IN THIS ISSUE

London
Journals Press

IMAGE: COMPUTERS & TECHNOLOGY IN A MODERN AIRCRAFT

www.journalspress.com

LONDON JOURNAL OF RESEARCH IN COMPUTER SCIENCE AND TECHNOLOGY

Volume 20 | Issue 2 | Compilation 1.0

Print ISSN 2514-863X
Online ISSN 2514-8648
DOI 10.17472/LJCST

COMPILED IN UNITED KINGDOM

ENGLISH

London Journals Press

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

PUBLISHER

London Journals Press
1210th, Waterside Dr,Opposite Arlington Building, Theale, Reading

SUBSCRIPTION

Frequency: Quarterly

Print subscription
$280USD for 1 year
$500USD for 2 year
(color copies including taxes and international shipping with TSA approved)

Find more details at https://journalspress.com/journals/subscription

ENVIRONMENT

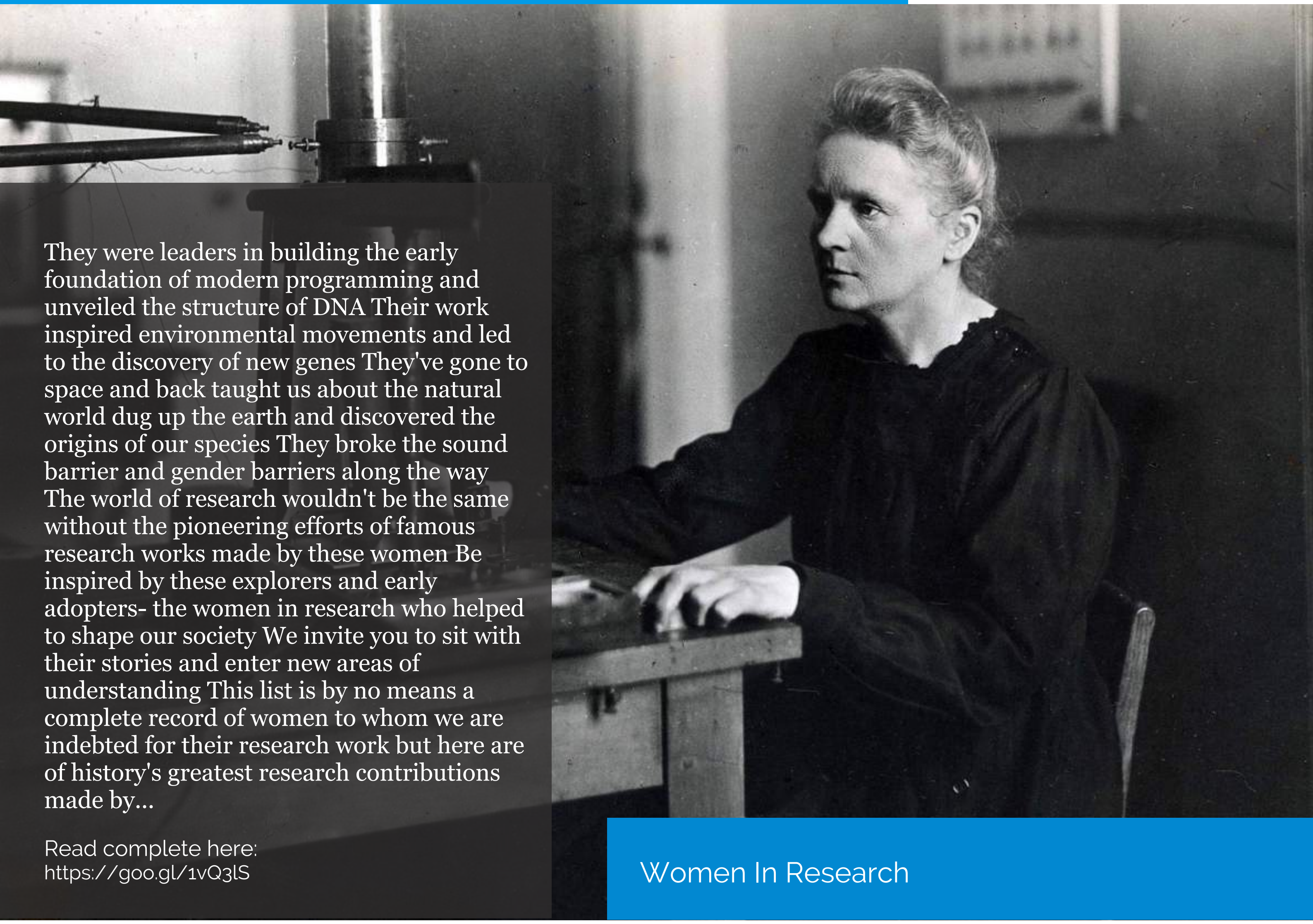
London Journals Press is intended about protecting the environment. This journal is printed using led free environmental friendly ink and acid-free papers that are 100% recyclable.

London Journals Press Headquaters

1210th, Waterside Dr,
Opposite Arlington
Building, Theale, Reading
Phone:+444 0118 965 4033
Pin: RG7-4TY
United Kingdom

Featured Blog Posts

blog.journalspress.com



Women In Research

They were leaders in building the early foundation of modern programming and unveiled the structure of DNA Their work inspired environmental movements and led to the discovery of new genes They've gone to space and back taught us about the natural world dug up the earth and discovered the origins of our species They broke the sound barrier and gender barriers along the way The world of research wouldn't be the same without the pioneering efforts of famous research works made by these women Be inspired by these explorers and early adopters- the women in research who helped to shape our society We invite you to sit with their stories and enter new areas of understanding This list is by no means a complete record of women to whom we are indebted for their research work but here are of history's greatest research contributions made by...

Read complete here:
https://goo.gl/1vQ3lS

Writing great research...

Prepare yourself before you start Before you start writing your paper or you start reading other...

Read complete here:
https://goo.gl/qKfHht

Computing in the cloud!

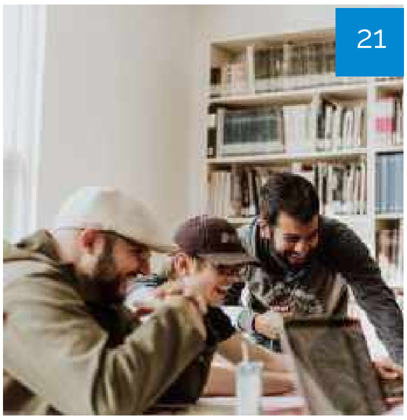
Cloud Computing is computing as a Service and not just as a Product Under Cloud Computing...

Read complete here:
https://goo.gl/H3EWK2

Journal Content

In this Issue

London
Journals Press



Editorial Board

Curated board members

Dr. Robert Caldelli

CNIT - National Interuniversity
Consortium for Telecommunications
Research Unit at MICC Media Integration and
Communication Center Ph.D.,
Telecommunications and Computer Science
Engineering, University of Florence, Italy

Dr. Xiaoxun Sunx

Australian Council
for Educational Research
Ph.D., Computer Science
University of Southern Queensland

Dariusz Jacek Jakóbczak

Department of Electronics and Computer Science,
Koszalin University of Technology, Koszalin,
Ph.D., Computer Science, Japanese Institute of
Information Technology,
Warsaw, Poland.

Dr. Yi Zhao

Harbin Institute of Technology Shenzhen
Graduate School, China Ph.D.,
The Hong Kong Polytechnic
University Hong Kong

Dr. Rafid Al-Khannak

Senior Lecturer Faculty of Design, Media and
Management Department of Computing Ph.D
Distributed Systems Buckinghamshire New
University, United Kingdom

Prof. Piotr Kulczycki

Centre of Information Technology for Data
Analysis Methods, Systems Research Institute,
Polish Academy of Sciences, Faculty of Physics
and Applied, Computer Science AGH University
of Science and Technology, Poland

Dr. Shi Zhou

Senior Lecturer, Dept of Computer Science,
Faculty of Engineering Science, Ph.D.,
Telecommunications Queen Mary,
University, London

Prof. Liying Zheng

School of Computer Science and Technology,
Professor for Computer Science, Ph.D.,
Control Theory and Control Engineering, Harbin
Engineering University, China

© 2020 London Journals Press

Dr. Saad Subair

College of Computer and Information Sciences,
Alazaeim Alazhari University, Khartoum North,
Sudan, Associate Professor of Computer Science
and Information Ph.D., Computer Science
Bioinformatics, University of Technology
Malasiya

Gerhard X Ritter

Emeritus Professor, Department of Mathematics,
Dept. of Computer & Information,
Science & Engineering Ph.D.,
University of Wisconsin-Madison, USA

Dr. Ikvinderpal Singh

Assistant Professor, P.G. Deptt. of Computer
Science & Applications, Trai Shatabdi GGS
Khalsa College, India

Prof. Sergey A. Lupin

National Research,
University of Electronic Technology Ph.D.,
National Research University of Electronic
Technology, Russia

Dr. Sharif H. Zein

School of Engineering,
Faculty of Science and Engineering,
University of Hull, UK Ph.D.,
Chemical Engineering Universiti Sains Malaysia,
Malaysia

Prof. Hamdaoui Oualid

University of Annaba, Algeria Ph.D.,
Environmental Engineering,
University of Annaba,
University of Savoie, France

Prof. Wen Qin

Department of Mechanical Engineering,
Research Associate, University of Saskatchewan,
Canada Ph.D., Materials Science,
Central South University, China

Luisa Molari

Professor of Structural Mechanics Architecture,
University of Bologna,
Department of Civil Engineering, Chemical,
Environmental and Materials, PhD in Structural
Mechanics, University of Bologna.

Prof. Chi-Min Shu

National Yunlin University of Science
and Technology, Chinese Taipei Ph.D.,
Department of Chemical Engineering University of
Missouri-Rolla (UMR) USA

Prof. Te-Hua Fang

Department of Mechanical Engineering,
National Kaohsiung University of Applied Sciences,
Chinese Taipei Ph.D., Department of Mechanical
Engineering, National Cheng Kung University,
Chinese Taipei

Dr. Fawad Inam

Faculty of Engineering and Environment,
Director of Mechanical Engineering,
Northumbria University, Newcastle upon Tyne,
UK, Ph.D., Queen Mary, University of London,
London, UK

Dr. Rocío Maceiras

Associate Professor for Integrated Science,
Defense University Center, Spain Ph.D., Chemical
Engineering, University of Vigo, SPAIN

Muhammad Hassan Raza

Postdoctoral Fellow, Department of Engineering
Mathematics and Internetworking,
Ph.D. in Internetworking Engineering,
Dalhousie University, Halifax Nova Scotia,
Canada

Rolando Salgado Estrada

Assistant Professor,
Faculty of Engineering, Campus of Veracruz,
Civil Engineering Department, Ph D.,
Degree, University of Minho, Portugal

Abbas Moustafa

Department of Civil Engineering,
Associate Professor, Minia University, Egypt, Ph.D
Earthquake Engineering and Structural Safety,
Indian Institute of Science

Dr. Babar shah

Ph.D., Wireless and Mobile Networks,
Department of Informatics,
Gyeongsang National University,
South Korea

Dr. Wael Salah

Faculty of Engineering,
Multimedia University Jalan Multimedia,
Cyberjaya, Selangor, Malaysia, Ph.D, Electrical and
Electronic Engineering, Power Electronics
and Devices, University Sians Malaysia

Prof. Baoping Cai

Associate Professor,
China University of Petroleum,
Ph.D Mechanical and Electronic Engineering,
China

Prof. Zengchang Qin

Beijing University of Aeronautics
and Astronautics Ph.D.,
University of Bristol,
United Kingdom

Dr. Manolis Vavalis

University of Thessaly,
Associate Professor, Ph.D.,
Numerical Analysis,
University of Thessaloniki, Greece

Dr. Mohammad Reza Shadnam

Canadian Scientific Research and Experimental
Development Manager-Science,
KPMG LLP, Canada, Ph.D., Nanotechnology,
University of Alberta, Canada

Dr. Gang Wang

HeFei University of Technology,
HeFei, China, Ph.D.,
FuDan University, China

Kao-Shing Hwang

Electrical Engineering Dept.,
Nationalsun-Yat-sen University Ph.D.,
Electrical Engineering and Computer Science,
Taiwan

Mu-Chun Su

Electronics Engineering,
National Chiao Tung University, Taiwan,
Ph.D. Degrees in Electrical Engineering,
University of Maryland, College Park

Zoran Gajic

Department of Electrical Engineering,
Rutgers University, New Jersey, USA
Ph.D. Degrees Control Systems,
Rutgers University, United States

Dr. Homero Toral Cruz

Telecommunications,
University of Quintana Roo, Ph.D.,
Telecommunications Center for Research
and Advanced Studies National Polytechnic
Institute, Mexico

Nagy I. Elkalashy

Electrical Engineering Department,
Faculty of Engineering,
Minoufiya University, Egypt

Vitoantonio Bevilacqua

Department of Electrical and Information
Engineering Ph.D., Electrical Engineering
Polytechnic of Bari, Italy

Dr. Sudarshan R. Nelatury

Pennsylvania State University USA Ph.D., Signal
Processing Department of Electronics and
Communications Engineering,
Osmania University, India

Prof. Qingjun Liu

Professor, Zhejiang University, Ph.D.,
Biomedical Engineering,
Zhejiang University, China

Scan to know paper details and
author's profile

Seasonal Variations in Physicochemical parameters and Benthic Microinvertebrates diversities of *Ntawogba* water quality Port Harcourt

Umunnakwe Johnbosco E., Aharanwa Bibiana C., Njoku Richard E
& Umunnakwe Bernadine A.

university of Technology Owerri Nigeria

ABSTRACT

The study determined the seasonal variation of Ntawogba creek's water quality on some physical-chemical parameters and the diversity of benthic macroinvertebrates as bioindicators of the river for twelve months Based on ecological niche of the system and anthropogenic activities in the area, three sampling stations and their coordinates were established along the stretch of the stream. The physicochemical parameters studied include temperature, pH, conductivity, salinity, turbidity, total dissolved solids, dissolved oxygen and biochemical oxygen demand according to standard methods as described by APHA, 1988. Sediment particle size was determined during wet and dry seasons by hydrometer method. Replicate samples of benthic macroinvertebrates were collected for each month randomly with Eckman's grab, and their diversity determined using Shannon-Wiener index equitability species.

Keywords: seasonal, parameters, dry, wet, benthos, macroinvertebrates.

Classification: J.2

Language: English

London
Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

Seasonal Variations in Physicochemical Parameters and Benthic Macroinvertebrates diversities of Ntawogba Water Quality Port Harcourt

Umunnakwe Johnbosco E.[α], Aharanwa Bibiana C.[σ], Njoku Richard E[ρ]
& Umunnakwe Bernadine A[Ѡ]

ABSTRACT

The study determined the seasonal variation of Ntawogba creek's water quality on some physical-chemical parameters and the diversity of benthic macroinvertebrates as bioindicators of the river for twelve months .Based on ecological niche of the system and anthropogenic activities in the area, three sampling stations and their coordinates were established along the stretch of the stream. The physicochemical parameters studied include temperature, pH, conductivity, salinity, turbidity, total dissolved solids, dissolved oxygen and biochemical oxygen demand according to standard methods as described by APHA, 1988. Sediment particle size was determined during wet and dry seasons by hydrometer method. Replicate samples of benthic macroinvertebrates were collected for each month randomly with Eckman's grab, and their diversity determined using Shannon-Wiener index equitability species. Physicochemical data from the monthly collections were subjected to 2-way analysis of variance and the difference among means was separated by Turkey-honest significance difference at 95% of probability. Percentage occurrence and relative numerical abundance of macroinvertebrates were calculated using excel descriptive statistical tools. The results of the physical – chemical variables revealed that the following parameters; temperature, conductivity, total dissolved solids, salinity were higher in dry season than in wet season, while turbidity and dissolved oxygen were higher in wet season due to increased water runoff. pH values alternated between slightly acidic and slightly alkalinity. The diversity and taxa richness were higher in the wet season (1.914; 3.960) than dry season (0.877; 2.425). The seasonal variation compares favorably with those of similar environments in the Niger Delta,Nigeria.

Keywords: seasonal, parameters, dry, wet, benthos, macroinvertebrates.

Corresponding Author: Dr Umunnakwe Johnbosco. E, *Ph.d*; is a senior lecturer in the department of Environmental Management, Federal University of Technology, Owerri , Nigeria. He holds a Ph.D in Environmental Management. Dr. Umunnakwe has published over twenty five papers in reputable international journals. He is the immediate past National Vice President, Nigerian Environmental Society. He is a public analyst licensed by the federal ministry of health in water qualities analysis.

Dr (mrs) Bibiana Chimezie Aharanwa, Ph.d; Is a chemist by training and is a senior lecturer in the department of Polymer and Textile Technology, Federal University of Technology, Owerri. She has conducted many researches in waste management and effluent monitoring. She is the current SIWES coordinator in her department.

Surv. Richard Ebere Njoku, Ph.d: Is a lecturer in the school of Environmental Sciences, Federal University of Technology Owerri. He is an expert in surveying, data management and Geoinformatics. He is a licensed surveyor of the Federal republic of Nigeria.

Mrs Bernadine Akuoma Umunnakwe Ph.d;: Is a biologist by training and an environmentalist with emphasis on water pollution. She is a staff of Rivers

State school Education board and a Ph.D researcher in Ignatius Ajuru University of Education Port Harcourt, Rivers State Nigeria.

Author α: Department of Environmental Management Federal university of Technology Owerri Nigeria.
σ: Department of Polymer and Textile Technology Federal university of Technology Owerri Nigeria.
ρ: Department of Surveying and Geoinformatics Federal university of Technology Owerri Nigeria.
Ѡ: Department of Science Education Ignatius Ajuru University of Education Port Harcourt Nigeria.

I. INTRODUCTION

Water is one of the most abundant and essential commodities of man, occupying about 70% of the earth’s surface and when it contains too much contamination by certain microorganisms or chemical compounds, it is rendered unsafe in its existing state for an intended use (Eja, 2002., Iloba & Ruejoma, 2014). The addition of various kinds of pollutants and nutrients through the agency of sewage, industrial effluents, agricultural runoff etc. into the *Ntawogba* creek has brought about a series of changes in the physicochemical characteristics of water (Umunnakwe *et al.,* 2020) which have been the subject of several investigations (Nevoh *et al.*,2015). The concentrations of physicochemical parameters are useful in the assessment of water qualities (Musa *et al.*, 2016). Macroinvertebrates are small animals found in water bodies such as insects, crustaceans, molluscs, annelids that do not have backbones and can be seen with the naked eyes (Adedipe & Nwankwo, 2016).They are used as bioindicators because of different chemical and physical changes caused by pollutants entering the water body (Ekokotu,2016). Their richness, abundance, composition can change as a result of seasonal changes in water quality which can be used to assess the health of the water (Yakub *et al.*,2016). Seasonal variations in quality of water generally refer to the change in components of water, which are to be present at the optimum level for suitable growth of plants and animals., which in turn have a direct or indirect influence over the planktonic population as well as living beings, (Oyewo &Don Pedro, 2003; Izonfue & Bariweni 2002). Many researchers have reported that effluent surface run-off from market houses and abattoirs, municipal gutters and sewage water draining into rivers affected the chemical components of the water which lead to environmental pollution that affected the food chain (Tyokumbur *et al.,* 2002. George, 2008). It has been stated generally by some environmental workers (Izonfuo and Bariweni, 2002 ; Zabbey, 2002; Zabbey & Hart, 2006 and Zabbey *et al.*, 2008) that in Niger Delta water quality varies seasonally.

II. MATERIALS AND METHODS

2.1 The Study Area

Ntawogba stream is a tributary of the Bonny River which is the largest river in the Niger Deltadelta of Nigeria with an average width of 0.5km and coordinates, latitude 4°48’, 4° 4’7 North and longitude 6° 58’, 6° 01‘East. It is a meandering stream that traverses almost the entire width of Port Harcourt and harbors several residential communities and industrial companies along its bank with different activities like waste dump, vehicle repair/car wash, sand mining, boat building/repairs, fishing, recreation and drainage construction, etc.

2.2 Sampling Stations

Three main sampling stations were established along the stretch of the stream. The stations were selected based on the ecological niche of the system, taking into cognizance human activities and accessibility to the area.

Station 1, is non tidal freshwater, draining through residential and commercial areas (Fig.1). The main activities here are bathing, car wash, mechanic work, welding and mining of sand and waste dump. This section of stream is walled with concrete embankment (Plate 1 and 2).

Station 2 is in a tidal fresh water zone walled with concrete embankment with floating algal mass (Plates 3 and 4). Human activities are car wash, welding, mechanic work and waste dumping.

Station 3 is a tidal brackish water ecological zone, occupied by residential and industrial houses with mangrove characteristics. The main human activities are fishing, bathing, car wash, sand mining/dredging, boat building, jetty operations, transportation by the use of canoe and motorized vessels. The area receives many loads of domestic and industrial waste materials input from the city *(Plates 5 and 6)*.

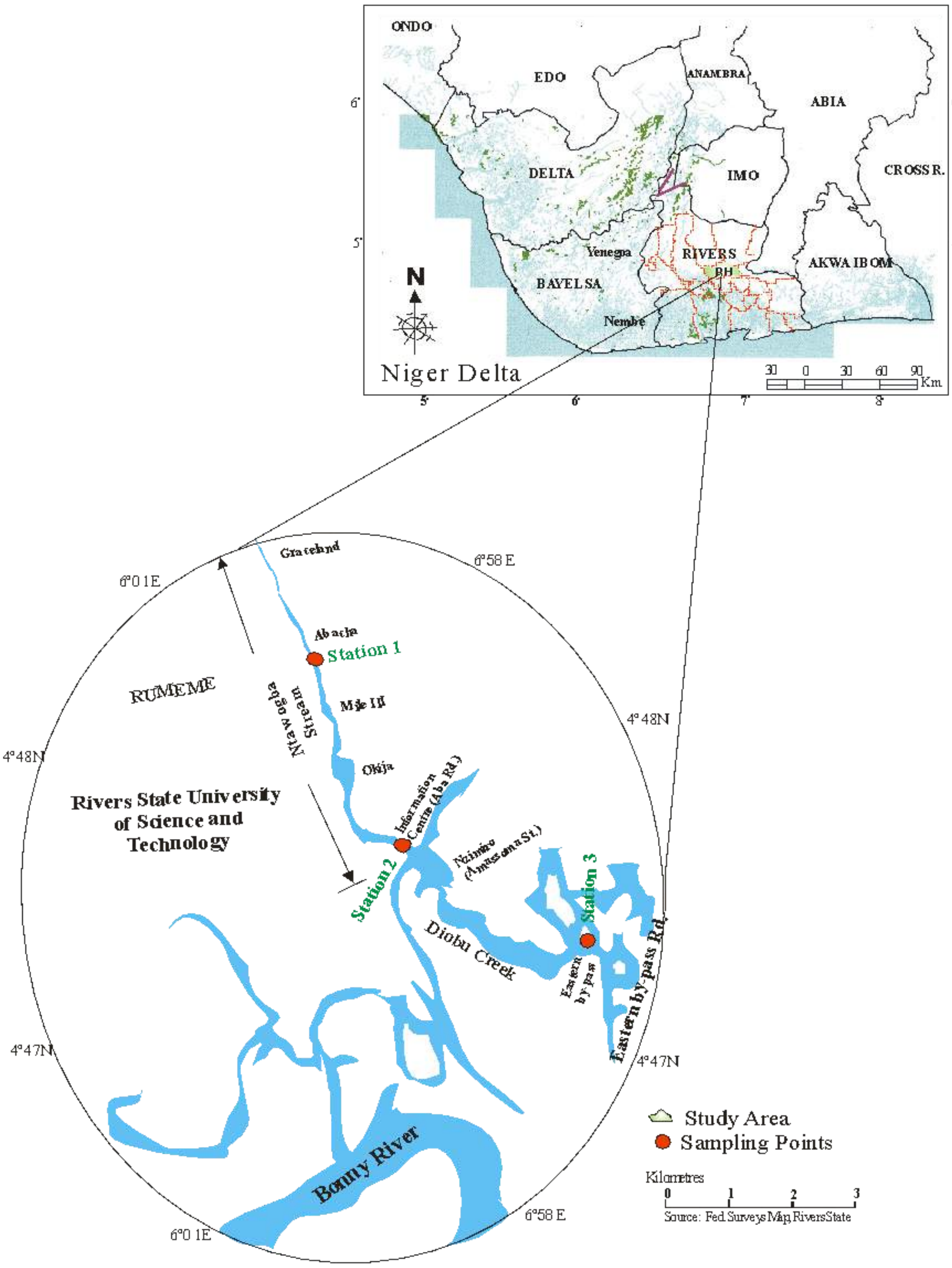


Fig 1: Map of Niger Delta showing Ntawogba Stream in Port Harcourt The Study Area

Plate 1: Station 1 - showing the Upstream part of the Study Area (Abacha axis)

Plate 2: Station 1 - showing the Downstream part of the Study Area (Abacha axis)

Plate 3: Station 2 - showing the Upstream part of the Study Area (Okija axis)

Plate 4: Station 2 -showing the Downstream part of the Study Area (Okijaaxis)

Plate 5: Station 3 -showing Upstream part of the Study Area (Eastern-Bypass)

Plate 6: Station 3 -showing the Downstream part of the Study Area (Eastern-by-pass Axis)

2.3 Sampling and Analysis

Rainfall data were retrieved from the records of the Department of Meteorological services, Port Harcourt. The methods used for the physical and chemical parameters studied were as described by APHA (1998).The water temperature was measured in situ in the field using mercury in glass thermometer (0-50°C), graduated at 0-01°C intervals. The water pH, electrical conductivity, Salinity Turbidity and Total dissolved solids in each of the sampled stations were measured in-situ directly in the field using a multiple-probe Horiba water checker (model U -10$^{\mu}$) by dipping the probe into the water sample. Surface water samples for the measurement of Dissolved Oxygen (DO) were collected and determined according to the modified Azide or Winkler's method (APHA, 1998).

Water sample for Biochemical Oxygen Demand (BOD) was collected in the same way as DO, but the BOD samples were diluted before incubation to determine the dilution factor and the BOD_5 calculated.

2.4 Sediment Study

Sediment particle size was determined by the hydrometer method (APHA 1998), by collecting samples at the stream bed from the upper 15cm layer at each of the study stations at both wet and dry seasons.

2.5 Benthic Macroinvertebrate

ReplicateReplicate samples of benthos were collected from each sampling station randomly, monthly with Eckmann's grab measuring $16cm^2$ x $16cm^2$ facilitated by lowering the grab with a long rope to the water bed, and retrieved by pulling. The samples from the stations were diluted with water, thoroughly mixed into slurry and preserved with 10% formalin to which rose Bengal dye had been added, to facilitate the sorting of organisms from the sample (Claudius *et al.*, 1979). The preserved samples were transported to the laboratory for subsequent analysis (Fakayode, 2005).

The organisms were identified taxonomically to the possible lowest levels (Sikoki & Zabbey 2006) using the best keys, such as Day (1967), Mellanby (1975); Young (1976); Powell (1980); Edmond (1978), and FAO (1990), and Pennark (1978)

2.6 Data Analysis

Physico-chemical data from the monthly sampling were subjected to 2 way analysis of variance. Differences among means were separated by Tukey Honest significant differences at 95% probability. Percentage occurrence and relative numerical abundance of macroinvertebrates macro benthos were calculated using Excel Descriptive Statistical Tools. Densities of the abundant species were analyzed for each of the sampled stations. Diversity of the benthic invertebrates was determined using Shannon Wiener Index, Equitability (E) of species (Sikoki, F.D. and Zabbey, N. 2006) and Margalef Diversity index.

III. RESULTS

The maximum rainfall was recorded in September (795.0mm) while the lowest rainfall (0.mm) was observed in December and January (both dry season months). Wet season value of rainfall was significantly higher (2,267.33mm) than dry season (48.5.5mm). Seasonally, mean temperature recorded was higher in dry season than in wet season (Tables 1 and 2). Highest pH value (7.83) was recorded in May (wet season) at station 1, while the lowest pH value (6.54) was measured in November (dry season) at Station 3.

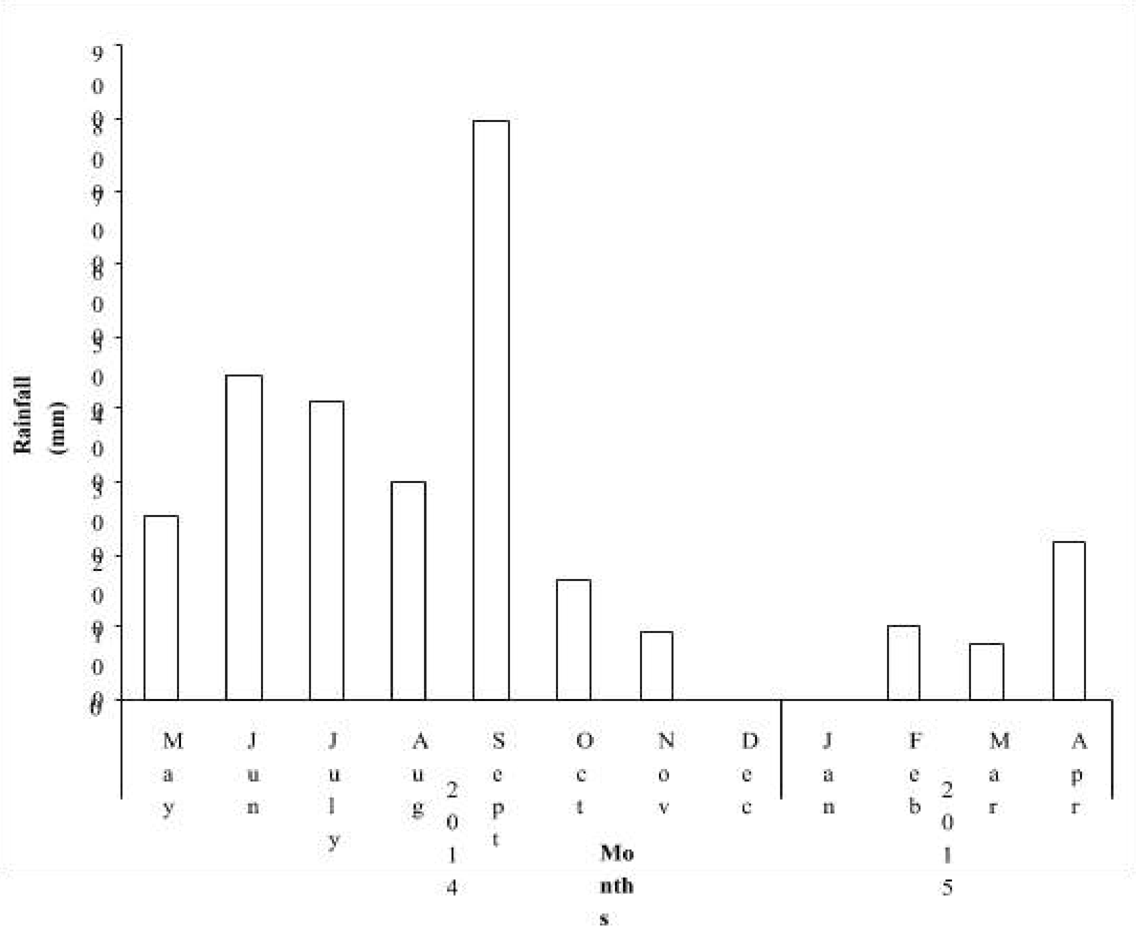


Fig. 2: Temporal Variation in Rainfall during the Study Period

Table 1: Monthly Range, Mean and Standard Deviation in Physico-chemical Parameters of Ntwogba Stream (May, 2014– April, 2015).

Month	Parameters							
	Temp. (°C)	pH	Cond. (µs/cm)	Turb. (NTU)	TDS (mg/l)	Sal (‰)	DO (mg/l)	BOD (mg/l)
May	30.07-31.05	7.63-7.83 7.73± 0.06 ^{a}	350-7,935 2,902.00 ± 16.58^{b}	11-782 332.33± 231.63^{a}	246-5555 2,031.33±1,761.64^{f}	0.1-0.5 1.57± 1.47^{d}	0-3.23 1.76 ± 0.94^{c}	12.12-18.20 14.62±2.15c

	30.72± 0.44^{a}							
June	29.02-30.07 29.71 ± 0.34^{a}	6.77-7.08 6.96 ± 0.10^{a}	229-12,615 4,422.00 ± 4,096.86^{f}	4-93 35.33 ± 28.87^{d}	190-8831 3,105.17 ±2,862.83^{f}	0.05-7.4 2.52 ± 2.44^{d}	1.62-7.11 3.72 ± 1.71^{a}	10.21-16.21 13.62±0.88c
July	27.04-28.06 28.10 ± 0.36	6.73-6.98 6.86 ± 0.07^{a}	239-12,155 4,238.28 ± 3,958.43^{f}	5-21 .11.33 ± 4.97^{e}	147-8509 2,959.67 ± 2,774.51^{f}	0.05-7.0 3.73 ± 2.31^{c}	1.62-7.11 3.73 ± 1.71 ^{a}	8.23-13.23 22.55±7.81b
August	26.57-29.04 27.88 ± 0.72^{a}	6.75-6.97 6.84 ± 0.07^{a}	234-1,504 728.33 ± 392.68^{i}	7-17 12.00 ± 2.89^{e}	164-6952 2,476.33 ± 2,238.25^{f}	0.05-5.7 1.95 ± 1.88^{d}	0.25-3.45 2.00 ± 0.94^{b}	6.10-8.15 17.29±1.59bc
September	26.04-29.06 27.14 ± 0.96^{a}	6.59-7.76 6.81 ± 0.14^{a}	187-9930 3,517.33 ± 3,207.13^{g}	8-20 13.33 ± 3.53^{e}	131-6951 2,462.33 ± 2,244.91^{f}	0.05-507 1.95 ± 1.88^{d}	1.23-3.66 2.85 ± 0.81^{ab}	5.34-6.22 16.74±2.23bc
October	29.55-30.54 29.72 ± 0.43^{a}	6.88-7.12 6.98 ± 0.07^{a}	238-7745 2,770.67 ± 2,487.31^{h}	25-60 45.00 ± 10.01^{c}	167-5422 1,939.67 ± 1,741.26^{f}	0.05-4.3 1.47 ± 1.39^{e}	2.61-3.25 2.71 ± 0.30^{ab}	7.06-8.16 17.82±3.66b
November	29..02-31.04 29.87 ± 0.61^{a}	6.54-7.25 6.90 ± 0.21^{a}	8037-33,985 2,0734.00 ± 7,495.66^{a}	8-84 56.67 ± 24.39^{b}	5626-23790 1,4514.00 ± 5,247.08^{a}	0.3-19.1 7.88 ± 5.70^{a}	0-3.93 2.26 ± 1.17^{ab}	13.10-16.14 15.16±4.30b

December	29.57-30.05 29.88 ± 0.16^{a}	6.67-6.93 6.77 ± 0.08^{a}	239-18,750 6,476.67 ± 6,136.94^{e}	6-16 10.33 ± 2.96^{e}	167-13125 4,533.33 ± 4,296.03^{e}	0.05-11.2 3.78 ± 3.71^{c}	0.02-3.67 2.38 ± 1.18^{ab}	13.02-14.22 13.21±4.24cb
January	31.04-33.05 32.04 ± 0.58^{a}	6.42-7.38 6.95 ± 0.28^{a}	4593-33,913 1,6949.00 ± 8,771.92^{b}	11-88 60.00 ± 24.58^{b}	3214-24439 1,2073.67 ± 6,372.92^{b}	0.4-18.5 8.08 ± 5.41^{a}	0-4.12 2.44 ± 1.25^{ab}	17.02-28.07 10.73±3.54d
February	30.03-31.04 31.87 ± 0.76^{a}	6.88-7.14 7.04 ± 0.08^{a}	749-25,000 8,919.67 ± 8,040.52^{d}	4-76 32.33 ± 22.15^{d}	306-17500 6,171.00 ± 5,665.68^{d}	0.5-13.3 7.15 ± 4.35^{a}	0.03.3.25 1.61 ± 0.93^{c}	16.16-18.41 7.13±1.45ec
March	31.04-32.52 31.054 ± 0.44^{a}	6.65-7.31 6.90 ± 0.21^{a}	629-21,450 7,632.67 ± 6,908.88^{e}	1-19 8.33 ± 5.46^{e}	439-15015 5,342.33 ± 4,836.49^{e}	0.2-13.1 4.52± 4.27^{b}	0-3.45 1.56 ± 1.01^{c}	15.16-18.32 5.78±0.62fd
April	30.52-33.02 31.52 ± 0.76^{a}	7.17-7.44 7.29 ± 0.08^{a}	579-33,845 1,1994.67 ± 1,0928.84^{c}	1-98 46.33 ± 28.18^{c}	407-18350 6,616.33 ± 5,870.16^{c}	0.2-14.3 5.07 ± 4.62^{b}	0-3.25 1.59 ± 0.94^{c}	15.23-20.41 7.61±0.78ed

Note: Means with different superscripts in the same column are significantly different @ $p > 0.05$ (Tukey HSD)

Table 2: Seasonal Range, Mean and Standard Deviation in Physico-chemical Parameters of Ntawogba Stream (May, 2014 – April, 2015)

Parameters	Wet Season Range, Mean ±SD	Dry Season Range , Mean ±SD
Temperature (°C)	26.04-31.55 28.88±0.61^{b}	29.02-33.05 31.12±0.48^{a}
pH	6.59-7.83 7.03±0.15^{a}	6.42-7.44 6.98±0.10^{b}

Conductivity (µS/cm)	187.00-12,615.00 3,090.80±568.40^{b}	239.00-33,985.00 11,232.28±2,540.54^{a}
Turbidity (NTU)	4.00-782.00 74.83±52.44^{a}	1.00-98.00 34.42±10.78^{b}
TDS (mg/l)	131.00-8,830.50 2,495.64±205.25^{a}	167.00-24,439.00 2,093.78±2,131.45^{b}
Salinity (%)	0.05-7.40 1.93±0.18^{b}	0.10-19.05 6.08±0.80^{a}
Do (mg/l)	0.0-7.11 2.80±0.52^{a}	0.0-3.93 1.93±0.18^{b}
BOD (mg/l)	5.34-18.20 11.70±4.90^{b}	13.02-28.07 19.26±4.81^{a}

$^{a-b}$ *Means with different superscripts in the same column are significantly different at* $p > 0.05$ *(Tukey HSD)*

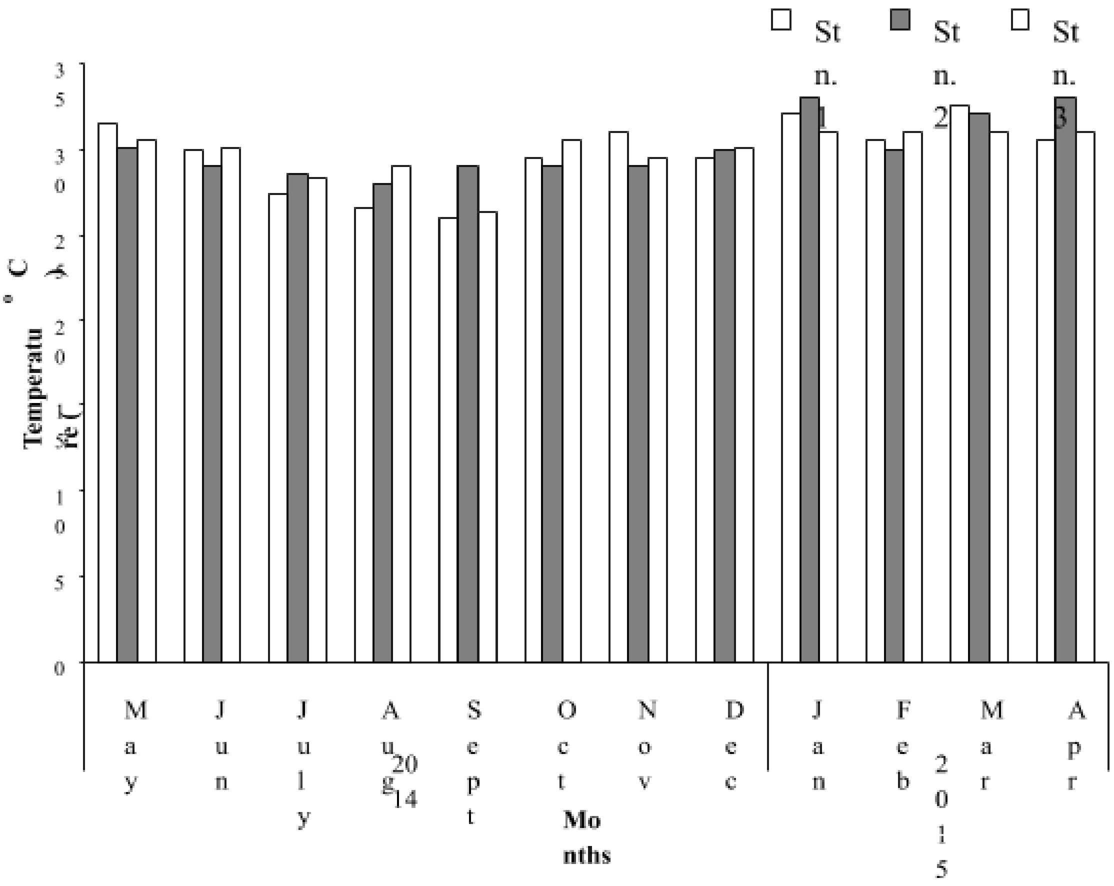


Fig. 3: Temporal Variation in Temperature of the Study Stations

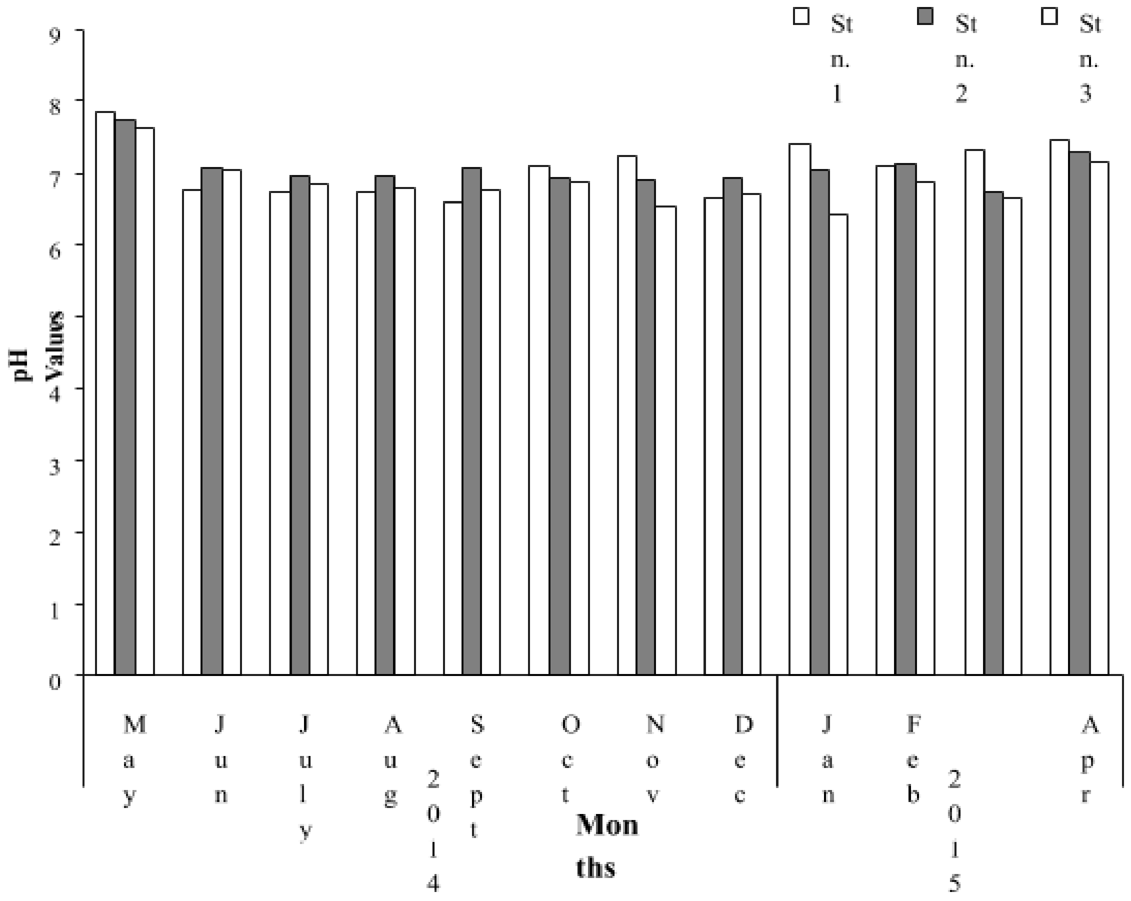


Fig. 4: Temporal Variation of pH in the Study Stations

Seasonally, conductivity varied, having higher value in the dry season (33,985.0mS/cm) than the wet season (8,830.0mS/cm). Seasonal variation was also observed in turbidity with higher values (4.00 to 782.00 NTU) during the wet season (34.42±10.78NTU) than the dry season (1.0-98.0 NTU). There was seasonal variation of total dissolved solids between dry season (2,.439. 0mg/l) and wet season (8,830.0mg/l) ;and higher salinity value was obtained in the dry season (19.05ppt) than the wet season (7.40ppt). The results of monthly dissolved oxygen values ranged between 0.0mg/l and 7.11mg/g(figure 5). Also, dissolved oxygen value was higher in the wet season than in dry season. Monthly variation of Biochemical Oxygen Demand (BOD) varied from 5.34mg/l to 28.07mg/l in September and January during the study period.

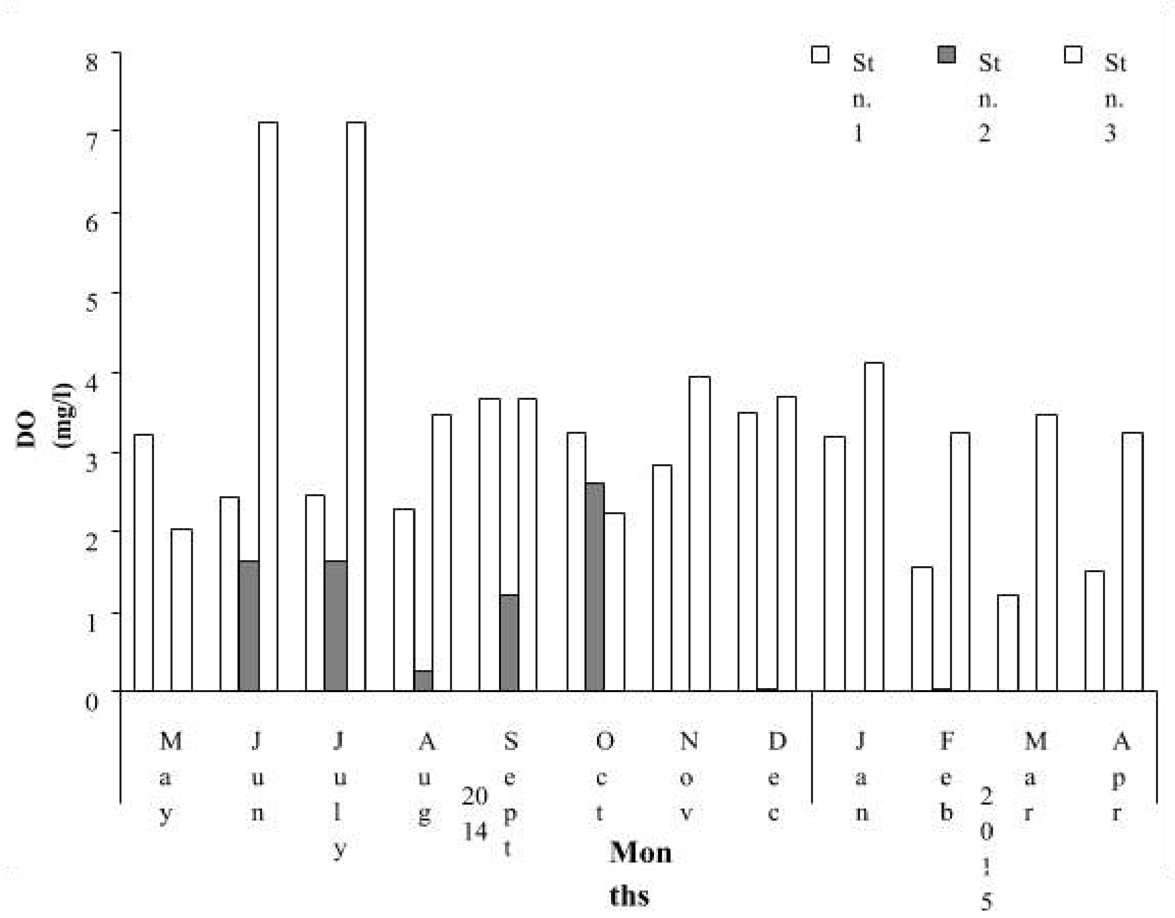


Fig. 5: Temporal Variation in DO of the Study Stations

Sediment Characteristics

Monthly value for sand ranged from 66.33% to 78.85%; the highest value was recorded in September and the lowest value in January, respectively. Seasonally, the wet season value was higher (78.85%) than dry season (74.76%). The monthly percentage silt ranged between 11.67 and 21.24% in May station 3 and January, station 2 across the stations. Dry season values were higher than wet season (21.24%) and 11.67%), having mean values of 117.03 ±0.61% and 15.94±0.74%, respectively. Clay content ranged between 3.66% and 15.73% in July and January across the stations. Meanwhile, dry season clay fraction was higher than wet season, varying significantly ($p < 0.05$).

Table 3: Monthly Range, Mean and Standard Deviation in Sediment Characteristics of Ntawoba Stream (May, 2014 – April, 2015).

Months	Clay (%) Mean ± SE	Silt (%) Mean ± SE	Sand (%) Mean ± SE
May	10.40-11.15 10.70 ± 0.23^{a}	11.67-14.20 12.85 ± 0.74^{a}	75.16-77.19 76.42 ± 0.64^{a}

June	5.38-10.70 7.76 ± 1.56^{bc}	14.56-18.60 16.92 ± 1.21^{ab}	74.20-77.19 75.38 ± 0.92^{a}
July	5.23-11.21 8.21 ± 1.72^{b}	13.38-19.30 16.85 ± 1.79^{ab}	75.40-75.42 74.44 ± 0.97^{a}
August	5.90-11.72 8.71 ± 1.68^{b}	14.08-19.30 16.88 ± 1.52^{ab}	72.20-76.86 74.42 ± 1.35^{a}
September	3.66-11.81 6.99 ± 2.47^{c}	12.34-16.40 16.80 ± 1.25^{ab}	75.30-78.85 76.67 ± 1.10^{a}
October	8.30-12.76 10.63 ± 1.29^{a}	13.89-19.20 17.34 ± 1.73^{a}	70.06-73.33 71.23 ± 1.05^{ab}
November	8.38-12.70 10.71 ± 1.28^{a}	13.94-19.49 17.57 ± 1.82^{a}	69.57-73.27 71.05 ± 1.13^{ab}
December	10.46-13.30 11.59 ± 0.88^{a}	14.44-19.47 17.74 ± 1.65^{a}	69.56-72.24 70.67 ± 0.81^{ab}
January	11.04-15.73 13.20 ± 1.30^{a}	21.17-15.10 19.17 ± 2.04^{a}	66.33-69.17 67.63 ± 0.83^{c}
February	10.09-13.57 11.98 ± 1.02^{a}	12.17-19.32 16.31 ± 2.14^{ab}	70.30-74.26 71.72 ± 1.28^{ab}
March	11.51-12.90 12.25 ± 0.40^{a}	12.75-18.25 16.13 ± 1.71^{ab}	70.24-74.36 71.63 ± 1.36^{ab}
April	8.22-12.80 10.44 ± 1.35^{a}	12.18-18.55 15.25 ± 1.84^{ab}	73.33-74.94 74.31 ± 0.54^{a}

Means with different superscript letters in the same column are significantly different $p > 0.05$ (Tukey HSD)

Table 4: Seasonal Range, Mean and Standard Deviation in Sediment Characteristics of Ntawogba Stream (May, 2014 – April, 2015).

Parameter **Parameters**	**Wet Season** **Range, Mean** ±SE	**Dry Season** **Range, Mean** ±SE
Clay (%)	3.66 -12.76 8.83± 0.77^{b}	8.22 – 15.75 11.69 ± 0.49^{a}
Silt (%)	11.67 – 19.30 15.94 ± 0.74^{b}	12.17 – 21.24 117.03 ± 0.61^{a}
Sand (%)	70.60 – 78.85 74.76 ± 0.86^{a}	66.33 – 74.76 71.17 ±0.92^{a}

Means with different *superscript letters in the same column are significantly different $p > 0.05$ (Tukey HSD)*

IV. BENTHIC MACROINVERTEBRATE

The pattern of seasonal distribution and relative abundance of benthic species is presented in Table 5. Generally, the highest density of 19,852 individuals/cm^2 (69.1%) of benthos was recorded in the dry season as compared to the wet season density of 8,878 individuals/cm^2 (30.9%). Meanwhile, the diversity and taxa richness were higher in wet (1.914; 3.960) than dry period (0.877; 2.425) for the organisms, while the evenness was higher in dry(0.627) than wet(0.583). The results of monthly density and distribution patterns of the benthos show that of the forty (40) species of benthos encountered in the area, variation in occurrence and distribution pattern was observed. The dry season month of December recorded the highest number of organisms with a total of 9,053 individuals (31.5%), closely followed by November, having total number 8,783 individuals (30.6%). The least number of individuals (217) was recorded in September (0.8%).

Table 5: Seasonal Variation in Benthic Organisms of Ntawogba Stream (May, 2014 – April, 2015)

	Species	Wet Season %	Dry Season %
ANNELIDA			
Oligochaeta	Lumbriculus sp	430	2,008
	Eiseniella tetrahidra	477	3,971
	Chaetogaster diastrophus	133	540
	Dero obtusa	174	1,577
	Nairs sp	195	78
	Ophidonais serpentina	1,650	3,231
	Paranais sp	153	394
	Stylaria lacustris	157	84
	Uncinais uncinata	1,022	3,463
	Total	4,391	15.345
	Percentage	15.3%	53.4 %
Polychaeta	Arenicola neamarina	8	-
	Capitella capitata	127	210
	Heteromastus filiformis	6	5
	Notomastus tenuis	30	8
	Notomastus abarans	569	-
	Notomastus latericieus	8	6
	Diopatra capitata	5	-
	Glycinde armingera	10	-
	Glycera capitata	2	-
	Glycera convoluta	3	-
	Nephthys hombergi	15	6
	Nereis diversicolor	49	13
	Nereis virens	26	-
	Nerieis pelagica	3	-
	Fabricia capensis	27	31
	Polydora capensis	3,367	4,096
	Syllis prolifera	5	-
	Syllis fulceligera	7	8
	Amphitrite neajohonstari	-	3
	Total	4,267	4,431
	Parentage	14.9%	15.4%
ARTHROPODA			
Crustacea	Ampithoe rubricata	-	8
	Gammarus lacusta	8	-
	Penaeus notialis	5	-
	Total	13	8
	Parentage	0%	0%
Insecta	Chironomus ablabiesmia	58	21
	Choaborus corethra	5	-
	Culex molestus	-	17
	Total	63	38
	Parentage	0.2%	0.1%
MOLLUSCA			
Gastropoda	Bullinus globuscus	2	3
	Bullinus forskali	5	-

	Pila ovata	23	12
	Planobis albus	5	-
	Physa fantinalis	106	59
	Total	141	74
	Parentage	0.5%	0.2%
Bivalvia	Tegalus andasonii	3	-
	Total	3	-
	Parentage	0%	0%
	Total	8,878	19,852
	Parentage	30.9%	69.1%
	Diversity	1.914	0.877
	Taxa richness	3.960	2.425
	Evenness	0.583	0.627

V. DISCUSSION

Rainfall records indicated a seven months wet season (April to October) period and five months dry season (November to March). The highest monthly rainfall (795.0mm) during the study period was recorded in September. NEDECO (1980) characterized this as a coastal rainfall pattern. From the variability in rainfall, some of the physico-chemical parameters of Ntawogba stream were inevitably influenced by rainfall which has been observed as one of the characteristics of Nigerian and tropical waters (Adebisi, 1980; Egborge, 1988 and 1999; George, 2008). The results of the physico-chemical condition of Ntawogba stream, Port Harcourt, Nigeria revealed that temperature values were generally higher in the dry season months due to direct solar radiation on the water body compared to wet season months. Low water temperature in the wet season may be attributed to numerous rainfall or precipitation which is typical of African inland waters (Wang et al., 2014). Also, pH values in various months of the surface water of Ntawogba Stream alternated between slightly acidic and slightly alkalinity. Bride et al., (2004) reported pH as an important ecological parameter that has a strong relationship with the physiology of most aquatic organisms. Electrical conductivity values observed in Ntawogba stream during the study was also noticed to be lower in wet season and generally higher in dry season. Most Nigerian inland waters have conductivity values below 500µS/cm at the peak of dry season and less in wet season (Egborge, 1994, Zabbey, 2002; George, 2008). Seasonal variation was also observed with higher turbidity mean values during the wet season than dry season. This was attributed to increased water velocity due to surface run off and industrialization (Chindh *et al.*, 2005).

There was also variation in total dissolved solids recorded between dry and wet seasons as a result of nature of water flow and continual discharge of effluent by humans in the area (Ajao & Fagade 2002; Zabbey 2002). Odokuma and Okpokwashili (1996) recorded higher values of TDS in dry season than rainy season in a previous work on the stream. Higher salinity values were obtained in the dry season than wet season. The lowest salinity values obtained in the rainy season are attributed to high rainfall during this period with high volumes of fresh water discharged into the water body thereby diluting and causing lower salinity. Solar radiation causes more evaporation and concentration of surface water to be more salty and increase in salinity value during the dry season (Manohar and Raghukumar (2013). The dry season months had lowest values of DO and highest in wet season. The higher values of dissolved oxygen obtained during the rainy season in Ntawogba could be due to increase in wet season rainfall and higher mixing rate. Yakubu *et al.* (1998) recorded higher values of dissolved oxygen in the early rainy season than dry season, due to increased rainfall and river runoffs resulting in increase in water current flow and high mixing rate. High BOD recorded in all seasons could be as a result of an increase in biodegradable organic substances in the study sites as noted by McNeely *et al.* (1979) . From the result of this study it may be suggested that differences in species composition, distribution and abundance recorded may be attributed to both differences in ecological, physiological, adaptations and environmental characteristics (Yakub *et al.*, 2016). Umeozor (1996) reported that during the rainy season, substrates were unstable thereby causing dislodgment of benthic invertebrates in the sediment while in dry season months, the substrate was stable and population built up. In this study, the number of macroinvertebrates increased from November and peaked in December, while the lowest numerical abundance value was recorded in September.

VI. CONCLUSIONS AND RECOMMENDATIONS

The seasonal variation was such that most of the parameters showed higher concentrations in dry season than in wet season. However Dissolved Oxygen value was higher in wet season than dry season indicating that the stream would support more life in wet season. Occurrence and distribution of the benthic macrofauna of Ntawogba stream showed marked seasonal differences. The diversity and taxa richness were higher in wet season than dry season. Species diversity of benthos in the study area compares favourably with similar environments in the Niger Delta.

REFERENCES

1. Adebisi, A.A. (1980). The physico-chemical hydrology of a tropical seasonal river upper Ogun River. Hydrologia, 79, 159 – 165.
2. Adedipe J.A& Nwankwo,D.I (2016).Effect of Rainfall cessation on the Plankton Abundance And Diversity In A Tidal Tropical Creek. Journal of Aquatic Sciences, 31(1), 115- 120..
3. Ajao, E.A. (1990). The influence of domestic and industrial effluents on population of sessile and benthic organisms in Lagos Lagoon, (Unpublished Doctoral Thesis), University of Ibadan.
4. Ajao, E.A & S.O. Fagade (2002). The benthic macrofauna of Lagos Lagoon. The Zoologist, 2, 1 –5.
5. American Public Health Association (APHA, 1998). Standard methods for the examination of water and wastewater. 20th edition, Washington DC, USA.
6. Amadi, E.N. & Chindah, A.C. (1997). The effect of municipal drainage on the Microflora of a black water stream in Port Harcourt, Nigeria,. Niger Delta Biologia, 2 ,25 - 39
7. Boltvosky, E. & Wright, R. (1976). Recent foraminifera. Junk Publishers, The Hague.
8. Braide, S.A., W.A.L. Izonfuo, P.U. Adiukwu, A.C. Chindah & C.C. Obunwo (2004). Water quality of miniweja stream. A swamp forest stream receiving non-point source waste discharges in Eastern Niger Delta, Nigeria, Journal of Scientia Africana, 3, 1 - 8.
9. Chindah, A.C; Braide, S.A. & Onwuteaka, J.N. (2005). Vertical distribution of Periphyton on woody substrate in the brackish wetland embayment of Bonny River, Niger Delta. Biologia, 5, 97-108.
10. Day, J.H. (1974). The ecology of the Morrumbene estuary. Mocambigue Transition Research Societies of South Africa,. 41, 43 - 97.
11. Dublin-Green, C.O. (1990). Seasonal variations in some physico-chemical parameters of the Bonny Estuary, Niger Delta, NIOMR Technical paper 59.
12. Egborge, A.B.M. Ezemonye, L.I. & Awoze G.E. (2003). Macroinvertebrate fauna of Udu-Ughievwen wetlands, Southern, Nigeria. Journal of Aquatic Science, 18, 1 - 8. Eja, M.A, (2002). Water pollution and sanitation for developing countries. Calabar: Seasprint (Nig.) Co.pp 9-13,125-129.
13. Ekokotu,F.A (2016).Diet composition and feeding inter-relationship in the Claroteid Catfish Clarotes laticeps in the Lower River

Niger ,Journal of Aquatic Sciences, 29(2B), 411-415.

14. Ekeh, I.B. & Sikoki, F.D. (2003). The state and seasonal variability of some physicochemical parameters in the New Calabar River, Nigeria. Acta Hydrobiolgia, 5, 45 – 60
15. Fakayode,S.O(2005). Impact of industrial effluents on water quality of the receiving water Alaro river in Ibadan,Nigeria, AJAEMA-RAGEE, z10(1),1-13.
16. Food & Agricultural Organisation (FAO, 1990). Field guide to the commercial marine resources of the gulf of Guinea. Wolfgang Schneider marine research.
17. George, A. D. I. (2008). The ecology of swimming crab (Callinectes amnicola) in Okpoka Creek, off upper Bonny River, Niger Delta, Nigeria. Thesis, Rivers State University of Science and Technology, Port Harcourt, Nigeria.
18. Hart, A.I. & N. Zabbey (2005). Physico-chemistry and benthic fauna of Woji Creek in the Lower Niger Delta, Nigeria Journal of Environment and Ecology, 23, 361 - 368.
19. ILoba, K. I., (2014).Composition and abundance of Rotifera in Ikpoba River, Benin, Edo State, Journal of Aquatic Sciences, 29(28),369 315.
20. ILoba, K. I. & Rue Joma, M.G.O. (2014). Physical characteristics and zooplankton of Kpan River Delta state, Nigeria, IJABR, 6(1),10-20.
21. Izonfuo.I.W.A and A.P.Bariweni(2002).The effect of urban runoff and human activities on some physicochemical parameters of Epie creek in the Niger Delta, Journal of Applied Sciences and Environmental Management, 5(1),47-55.
22. Mason, C.F. (1992). Biology of freshwater pollution 2nd Edition.Wiley and Sons-Incorporated. New York, 351.
23. Manohar,C.S& Raghukumar,C(2013).Fungal diversity from various marine habitats deduced through culture independent studies, FEMS Microbiological letters,341,69-78.
24. McNeely, R.N., Neimanis, V.P., & Dwyer, L. (1979). Environment Canada: Water quality sources book –A guide to water quality parameters.
25. Mellanby, H. (1975). Animal life in freshwater. A Guide to freshwater invertebrates. Chapman and Hall, London.
26. Musa, S. Onusirika B. C. & Emere, M. C. (2016). Physicochemical characteristics and abundance of aquatic macroinvertebrates in River Galma Zaria,Nigeria,31(2A),295-305.
27. NEDECO, (1980The waters of the Niger Delta report of an investigation by NEDECO (Netherlands Engineering Consultants). The Hague.
28. Nevoh. G.O, Akhionbare, S. M. O, & Uzomah, H. C. (2015).Water quality aspects of River Okpoka, in relation to watershed activities, Resources and Environment,5(4):124-133.
29. Odokuma, I.O. & Okpokwasili G.C. (1996). Seasonal influences of the New Calabar River, Nigeria. Environmental Monitoring and Assessment, 5, 1 - 14.
30. Oyewo, E.O. & Don-Pedro, K.N. (2003). Influence of Salinity Variability on heavy metal toxicity of three estuarine organisms. Journal of Nigeria Environmental. Science, 1, 141 - 155.
31. Pennak, R.W. (1978). Freshwater invertebrates of the United States, sound edition. John Wiley and Sons, New York.
32. Sikoki, F.D. and Zabbey, N. (2006). Environmental gradients and benthic community of the middle reaches of the Imo River, south-eastern Nigeria. Journal of Environmental Ecology. 24, 32 – 36.
33. Tyokumbur, E.T., Okorie, T.G., Ugwumba, A.O. (2002). Limnological assessment of the effect of effluents on Macroinvertebrates fauna in Awoba stream and reservoir, Ibadan, Nigeria, The Zoologist 1, 59 -69.
34. Umeozor, O.C. (1996). Influence of biotope, salinity and biochemical oxygen demand on the composition and distribution of aquatic insects in New Calabar River, Nigeria. Tropical FreshWater Biology, 5, 31 - 42.
35. Umunnakwe, J. E, Uyi, H. S, Umunnakwe, B. A. & Deekae, S. N.(2020). Benthic macroinvertebrates diversity and physical-chemical

parameters as indicators of the water qualities of Ntawogba creek Port Harcourt Nigeria. American Journal of Chemical & Biochemical Engineering, 4(11), 8-17.

36. Wang X,Singh P,Gao Z,Zhang X & Johnson,Z.I (2014). Distribution and Diversity of planktonic Fungi in the West Pacific warm pool,Plus One,9(7),10-15.
37. Yakub, A. S, Adedipe, J. A., Nkwoji, J. A., Abiodun, O. A., Bello, B. O., Ajijo, M. R. (2016). Distribution of Phytoplankton in a coastal Lagoon of Mahin, Ondo,Western Nigeria, Journal of aquatic sciences,31 (2A), 241-257.
38. Yakubu, A.F., Sikoki, F. D. and Horsfall, M. Jr. (1998). An investigation into the physico-chemical conditions and planktonic organisms of the lower reaches of the Nun River, Nigeria. Journal of Applied Science & Environmental Management, 1, 28 – 42.
39. Zabbey, N. (2002). An ecological survey of benthic macroinvertebrates of Woji Creek off the Bonny river system, Rivers State. (Unpublished Masters Thesis), University of Port Harcourt, Rivers State, Nigeria.
40. Zabbey, N. & Hart, A.I. (2006). Influence of some physico-chemical parameters on the composition and distribution of benthic fauna in Woji Creek, Niger Delta, Nigeria. Global Journal of Pure & Applied Science, 12, 1 – 5.
41. Zabbey, N., Sikoki, F.D. & Edoghotu, J. (2008). Plankton assemblages and environmental gradient in the middle reaches of the Imo River, Niger Delta, Nigeria. African Journal of Aquatic Science, 33, 241 – 248.

This page is intentionally left blank

Adoption of Open Educational Resource (OER) Materials by Faculty and Learners: A Framework to Increase Openness Adoption in Open and Distance Learning Programmes

Alimi, A. A, Adeyemo, I. A, Arulogun, O. T., Fajobi, D. T,Osunniyi, J. S,A deoye , M. A. & Ojo, O. S.

Ladoke Akintola University of Technology

ABSTRACT

In developing countries like Nigeria, distance learning programs have gained increased attention mainly due to the potential to provide equitable and accessible educational services for learners worldwide irrespective of location. This study specifically examined how frequent learners and faculty make use of OER materials. Also, this paper reports the investigation of faculty and learners' attitudes towards OER application and as well as identifying the purpose of using OER by them. Descriptive statistics were used as the research design and 1200 Open and Distance learning (ODL) students and staff of the Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria, were used as research population. 300 staff and students were sampled using Taro Yamane to determine the sample size. The finding indicated that OER was completely unknown to more than 60% of the respondents and that, among those who have some knowledge, less than 10% realized the importance of OER.

Keywords: open educational resource; distance learning; oer; materials; students; staff.

Classification: K.3.1

Language: English

London Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

Adoption of Open Educational Resource (OER) Materials by Faculty and Learners: A Framework to Increase Openness Adoption in Open and Distance Learning Programmes

Alimi, A . A[α], Adeyemo, I. A[σ], Arulogun, O. T.[ρ], Fajobi, D. T[Ѡ], Osunniyi, J. S[¥], Adeoye, M. A.[§]
& Ojo, O. S.[χ]

ABSTRACT

In developing countries like Nigeria, distance learning programs have gained increased attention mainly due to the potential to provide equitable and accessible educational services for learners worldwide irrespective of location. This study specifically examined how frequent learners and faculty make use of OER materials. Also, this paper reports the investigation of faculty and learners' attitudes towards OER application and as well as identifying the purpose of using OER by them. Descriptive statistics were used as the research design and 1200 Open and Distance learning (ODL) students and staff of the Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria, were used as research population. 300 staff and students were sampled using Taro Yamane to determine the sample size. The finding indicated that OER was completely unknown to more than 60% of the respondents and that, among those who have some knowledge, less than 10% realized the importance of OER. It can be concluded that OERs increasingly play an indisputable and fundamental role in providing cost-effective access to educational materials for both faculty and learners and becoming a valuable alternative to improve access to high-quality educational content released under open licenses by outstanding universities worldwide. Hence, distance learning programs should support the development of OER packages for improved teaching and learning.

Keywords: open educational resource; distance learning; oer; materials; students; staff.

Author: Ladoke Akintola University of Technology, Open and Distance Learning Centre, Ogbomoso, Nigeria.

I. INTRODUCTION

OERs have recently been described as digital materials that are freely and openly provided to educators, students, and self-learners for use and re-use in teaching, learning, and study. OER includes textbooks, course materials, manuals, videos, assessments, software, and all other resources,materials or techniques used to facilitate access to information, audio-video tutorials, sound and music lesson plans, quizzes, syllabi, instructional guides, models, content, software tools, licenses, and best practices. The 'open' conceptually implies the zero cost and non-restrictive nature of the materials (Rufai and Sani, 2017).

Numerous initiatives exist to intensify the adoption of open educational resources in tertiary institutions, yet its effect has not been greatly explored. Open Education can increase quality, access, and attractiveness of tertiary education, fostering "a more democratic and competitive higher education system, with the potential to improve access to education, develop and localize open educational services to suit local contexts, and enhance the integration of education into everyday lives as part of lifelong learning" (Fabio and Daniel, 2016).

In the past, a number of studies have investigated the reasons for the slow adoption of open access methods, focusing mainly on Open Policies and Open Educational Practices, but very few research projects have focused on the use of Open Educational Resource Materials (OER) for e-learners to open up to mainstream education, which is the need to empower learners to embrace open access. Kazmer and Haythornthwaite (2005) posit that e-learning comprises conventional training, such as courses, ad-hoc training, selected learning objects, formalization through document collections, and community formation which can be achieved via social software.

Recent studies suggest that the dearth of learning and instructional materials and the general underfunding of education, associated with the recent advancements in information and communication technology ease the application of OER in Nigeria educational enterprise (Rufai and Sani, 2017). The most important aspects of openness in OER have to do with free availability over the internet and as few restrictions as possible on the use of the resource (Clements, Pawlowski, and Manouselis, 2015). OER excludes technical barriers, price barriers, and a few legal permission barriers as possible (copyright and licensing restrictions) for the faculty and learners (Chen, Nasongkhla and Donaldson, 2014).

It is also a known fact that the students' eagerness and attitudinal change toward the internet and its applications in recent times is enough a spur for research of this kind. The intractable challenge of unawareness of students and tutors towards OER pedagogy has been the greatest challenge to learning from time immemorial. The review of studies on OER since this assertion was long made revealed that very scanty studies were carried out on this field. This is not still encouraging with regards to developing countries and especially in Nigeria when compared with developed nations, hence the need for this study.

II. AIM AND OBJECTIVES

This study aims to examine how frequent students and faculty apply OER materials while the specific objectives are to investigate the faculty and students' attitudes towards OER application and as well as determining the purpose of using OER by them.

III. LITERATURE REVIEW

Greenberg (1998) opined that contemporary distance learning is a learning experience or teaching plan that makes use of technologies in a wide spectrum to reach out to learners in which distance is not a barrier and it has been designed to encourage interaction between learners and certification of learning. In addition to this, Teaster and Blieszner (1999) posited that instructional methods have been used by various distance learning centers even though the teacher and the learner are separated in space and possibly time. Organization for Economic Co-operation and Development (OECD) 2007 reported that OER is materials that are digitized which are offered freely and openly to use and reuse for teaching learning and research by educators, students, and self-learners. It is deduced from this definition that accumulated assets involved in the development of individuals or capabilities for special understanding can be enjoyed without disparity/discrimination.

The UNESCO (2002) stressed that the words open and distance learning reflect solutions that focus on opening up access to education and training, freeing learners from time and space constraints, and creating flexible learning opportunities for individuals and groups of learners. Keats (2003), on the other hand, draws on lessons learned from open-source software creation and outlines a process model for collaborative content development.. Siemens (2003) listed several reasons for educators to share learning resources for free, including it, is free to share digital resources; it gives educators alternatives way and increases competition on the market; finally, it is democratic and a way to preserve public education.

IV. METHODOLOGY

A descriptive analysis was adopted. The descriptive method was used to ascertain the current status of the level of professional development among the learners who enrolled in LODLC degree programs. The study was conducted in Ogbomoso, Oyo State, Nigeria. The population of the study involved all the learners pursuing a degree program in LODLC across all the departments. LODLC's data records showed a total of 1,200 learners enrolled in a degree programme across four departments (Nursing, Accounting, Marketing, and Computer Science) operating in LODLC as of the 2017/2018 academic session. 300 samples consisting of learners in LODLC were randomly selected from the four departments i.e. computer science, marketing, accounting, and nursing with the used of the questionnaire using Taro Yamane formula with 95% confidence interval to determine the sample size, to reduce the sample size to a manageable size whose their ages range between 18 – 55 years. Thus, Taro-Yamane's expression was used:

$$n = \frac{N}{1+N(e^2)}\,, \qquad (1)$$

Where: n= Sample size, N= Total population, e= Margin of error disturbance. Therefore, given that N = 1200 (as stated above), and e is assumed to be 5%. Then sample size

$$n = \frac{1200}{1+1200(0.05^2)}\,,\quad n = \frac{1200}{1+1200(0.0025)}\,,\quad n = \frac{1200}{1+3}\,,\quad n = \frac{1200}{4}$$

$$= 300$$

Data collected was analyzed using simple tabulated frequency count and percentages.

V. RESULT AND DISCUSSIONS

The results are presented by analyzing the answers to each question. The distribution of responses regarding the awareness that students and staff have of OER platforms is presented in Table 1.

Table 1: Open Educational Resources Platforms (Awareness and unawareness)

Open Educational Resources Platforms	Not aware (%)	Aware (%)
MIT OpenCourseware	88	12
OpenCourseware Consortium	87	13
Open Learning Initiative	91	9
MERLOT	92	8
TOTAL	358	42
Percentage of the total	89.5%	10.5%

Source: Researcher computation, 2020

Based on the results directly related to OER and their platforms, it infers that the OER application is still unfamiliar within the teaching and learning context of higher education students, who simultaneously value the features of such resources with 89.5% not aware and 10.5% aware. Therefore, there is an urgent need to promote the dissemination and use of OER, so that students can benefit from resources which are mainly developed to be useful, freely available, charge-free, and used by anyone who wants to learn, regardless of their financial or social condition or of the place in the world where they happen to be.

Table 2: Cost associated with OER materials responded by students

Cost of OER materials	Frequency	%
Not costly	285	95.00
Costly	13	4.33
Very costly	2	1.8
Total	300	100

Source: Researcher computation, 2020

95% said OER does not cost them much compared to buying of textbooks that are related to their courses, so the adoption of OER reduces financial barriers to their course. However, there are financial savings in open textbook adoption that may likely reduce the rates of withdrawal students.

Table 3: Types of OER Sourced for by Students

Types of OER sourced for by students	Frequency	%
Textbooks	90	35.4
Video	120	47.24
Flashcards	40	15.75
Syllabus	4	1.57
Total	254	100

Source: Researcher computation, 2020

35.4% OER were of the type Textbook, 47.24% of Video, 15.75% of Flashcards and 1.57% of Syllabus are sourced for by students. This shows that Video is the highest type of OER being sourced for by students.

Table 4: Types of OER sourced for by Faculty

Types of OER sourced for by Faculty	Frequency	%
Textbooks	181	60.33
Video	101	33.67
Flashcards	18	6.00
Syllabus	0	0
Total	300	100.0

Source: Researcher computation, 2020

Textbooks have the highest percentage (60.33) been sourced for by the Faculty, this indicated that staff makes use of textbooks in OER to adapt to the materials that will be given to the students.

Table 5: Time Spent in studying with OER materials by the students

Time Spent	Frequency	%
Spend more than 24 hours per week	284	94.67
Spend less than 24 hours per week	16	5.33
Total	300	100.0

Source: Researcher computation, 2020

94.7 % of the student spent more than 24 hours per week studying because the study behaviours that they adopted in OER materials are easy to comprehend like the use of flashcards and reviewing lecture notes while 5.33% of the students spent less than 24 hours in studying.

Table 6: Awareness about OER materials by Faculty

Awareness about OER materials	N	%
I was not aware of Open Educational Resources	195	65.0
I was aware but I do not have an opinion about the potentialities of Open Educational Resources	45	15.0
I know the potentialities of Open Educational Resources but I have never used such resources	35	12.0
I have used Open Educational Resources and I consider them very useful in the teaching and learning process	21	7.0
I have used Open Educational Resources and I consider them of little relevance in the teaching and learning process	4	1.0
	300	100.0

Source: Researcher computation, 2020

The result above indicated that OER is completely unknown to more than 60% of the respondents and that, among those who have some knowledge, less than 10% give them importance. One of the concerns regarding the adoption of OER in Distance learning programmes lies in finding strategies that can make OER desirable to institutions, teachers and students. The results obtained give clear evidence of such a need. It is evident that in this study that a generalized adoption of OER did not occur.

VI. CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The results of this paper revealed that OERs are increasingly playing an indisputable and fundamental role in education for both faculty and learners. Interestingly, sharing educational resources openly and with no admission fee has ideological and financial justifications which creates a major demand for clarifying fundamental questions, such as who are the stakeholders in these processes, in what way are they involved and why, and how do they influence the development, use, and widening of OERs? However, print educational materials are tailored to address the educational needs of Nigerian students depending on the laid down objectives of the given institution and it is largely upon which the implementation of the national curriculum depends. The infrastructural decay of the print sections and the paradigm shift of the learning interest of the students is keeping them off the print section's register. The growing IT infrastructures, ease of access, and current contents and proximity associated with OER have made it an educational hotcake. Though print stocks remain relevant, their contribution to educational dispensation in Nigeria is still unsatisfactory.

6.2 Recommendations

To adequately address the problem associated with OER usage in the open and distance learning programme, the following policy recommendations are being suggested.

i. The government should undertake a novel project that would enable the development of our print materials, indigenous literature, and print artifacts in our academic libraries into OER to meet the students' needs and interests and at the same time meet up the trend of this globalized world.

ii. Distance learning programmes should support the development of OER packages for improved teaching and learning.

iii. Finally, the government should provide our librarians with in-service training to better their services toward their clients (students).

REFERENCES

1. Chen, S. H., Nasongkhla, J., & Donaldson, J. A. (2014). From vision to action – A strategic planning process model for open educational resources. Procedia - Social and Behavioral Sciences, 174, 3707–3714.
2. Clements, K., Pawlowski, J., & Manouselis, N. (2015). Open educational resources repositories literature review – Towards a comprehensive quality approaches framework. Computers in Human Behavior, 51, 1098–1106.
3. Fabio Nascimbeni & Daniel Burgos (2016). In Search for the Open Educator: Proposal of a Definition and a Framework to Increase Openness Adoption Among University Educators. International Review of Research in Open and Distributed Learning Volume 17, Number 6.
4. Rufai D, G & Sani M, A (2017).Use of Open Educational Resources and Print Educational Materials by Federal College of Education Katsina, Nigeria: A Study. DESIDOC Journal of Library & Information Technology, Vol. 37, No. 6, November 2017, pp. 437-442, DOI : 10. 14429/djlit.37.10628
5. Keats, D. W. (2003) "Towards a view of knowledge as the common heritage of humanity: mapping an Open Content strategy", In: M.A. Beebe, K.K. Magloire, B. Oyeyinka, and M. Rao (editors). Africa Dot Edu: Higher education and IT opportunities. New Delhi: Tata McGraw-Hill, 2003.
6. Kazmer, M. M. & Haythornthwaite, C. (2005). "Multiple perspectives on online learning", ACM Press, 25(1), pp.7-11
7. Siemens, G (2003) "Why We Should Share Learning Resources", available at: www.elearnspace.org/Articles/why_we_should_share.htm, 2003.
8. Teaster, P., & Blieszner, R. (1999). Promises and pitfalls of the interactive television approach to teaching adult development and aging. Educational Gerontology, 25 (8), 741-754.
9. UNESCO (2002) Open and Distance Learning: trends, policy, and strategy consideration. Paris, UNESCO
10. Rufai D, G & Sani M A.Use of Open Educational Resources and Print Educational Materials by Federal College of Education Katsina, Nigeria: A Study.

Scan to know paper details and
author's profile

An Enhanced Fraud Detection Model using Neural Networks for Telecommunications and Smart Cards in Nigeria

Onuodu, Friday Eleonu & Nnaa, Sunday Barikui

University of Port-Harcourt

ABSTRACT

Fraud, especially smartcard and telecommunication-based fraud always leave a grievous loss to its victims. The banking sector and the telecom companies has battled with this plague for years, fighting it with both technological and other security measures to eliminate its occurrence, but there are still open- problem despite all the efforts. Most of the systems developed are usually reactive instead of proactive, i.e. they detect the fraud after they have already occurred instead of preventing it, and others detect the fraud but do not have the mechanism to prevent it from occurring. Hence, the need for the development of an enhanced model that can detect Smartcard and telecom frauds in real-time and block the transaction while informing the relevant stakeholders (the account owner and the bank). In this work, we used the neural network to train a system using historical dataset of credit card fraudulent transactions and telecom fraud. This system could eliminate the inefficiencies of the existing systems and produced more efficient fraud detection and prevention using the Rule-Based approach to classify suspicious transactions and flag them if they contradict the rules. The result shows that fraud detection can now be made using well prepared datasets. Our model scored a performance accuracy mark of 94% as opposed to the existing system which had 65%.

Keywords: telecommunication, fraud, smart card, detection, neural network (ann).

Classification: F.1.1

Language: English

London
Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

An Enhanced Fraud Detection Model using Neural Networks for Telecommunications and Smart Cards in Nigeria

Onuodu, Friday Eleonu[α] & Nnaa, Sunday Barikui[σ]

ABSTRACT

Fraud, especially smartcard and telecommunication-based fraud always leave a grievous loss to its victims. The banking sector and the telecom companies has battled with this plague for years, fighting it with both technological and other security measures to eliminate its occurrence, but there are still open-problem despite all the efforts. Most of the systems developed are usually reactive instead of proactive, i.e. they detect the fraud after they have already occurred instead of preventing it, and others detect the fraud but do not have the mechanism to prevent it from occurring. Hence, the need for the development of an enhanced model that can detect Smartcard and telecom frauds in real-time and block the transaction while informing the relevant stakeholders (the account owner and the bank). In this work, we used the neural network to train a system using historical dataset of credit card fraudulent transactions and telecom fraud. This system could eliminate the inefficiencies of the existing systems and produced more efficient fraud detection and prevention using the Rule-Based approach to classify suspicious transactions and flag them if they contradict the rules. The result shows that fraud detection can now be made using well prepared datasets. Our model scored a performance accuracy mark of 94% as opposed to the existing system which had 65%. This work could be beneficial to telecom industries, to banks, to users of smartcards and POS and to every other person who carries out cashless transactions via the smartcards.

Indexterms: telecommunication, fraud, smart card, detection, neural network (ann).

Author α: Department of Computer Science, University of Port-Harcourt, Rivers State Nigeria.
σ: Department of Computer Science, Ignatius Ajuru University of Education, Rivers State, Nigeria.

I. BACKGROUND

Fraud is a major problem that the world is facing today. It can actually be classified as a disaster due to the devastating effects it has on its victims [1].Fraud has crept into all the sectors of the countries of the world and into almost all human activities and transactions with each other. From the activities of con artists, to forgery of documents, manipulation of records etc. over the years, several technologies for fraud detection have been developed in order to detect and prevent fraudulent activities in various sectors of the economy especially in the financial sector. In fact, once a technology emerges, the next step will be to build a strong fraud detection and prevention system that will work with the system to cushion and prevent fraudsters from accessing such technologies and carrying out fraudulent activities on them. Sometimes, most of the emerging technologies also have inbuilt fraud detecting mechanisms with them. However, some frauds are still hard to detect using the current technical security measures, especially in the telecommunication and smartcard based industries [2].

Telecommunication is a brilliant technology that has impacted the lives of almost all the citizens of the world as it aids communication across the globe, "In fact it is turning the world into a global village" [3]. This technology has experienced an increased acceptance across the world and this has led to a tremendous growth in the industry.

Innovations in the industry are also very frequent with new emerging technologies almost every day. The telecommunication community attends to the needs of two kinds of users, those who are provided with connections at an affordable rate also called the domestic users, and the second set are those who are provided with connections at a higher rate because of the higher scale of usage and subscription by these users, they are called the commercial users. However, over the years it has been noticed that the domestic users purchase subscriptions meant for the commercial users and vice-versa fraudulently. This is where the problem of fraud in the telecommunication industry and it has caused a great loss instead of profit to the sector. Another scenario of fraud is the telecom industry is when the users intentionally transmit voice data across a telecom network with the aim of avoiding or reducing the normal call charges. Abuse of voice and data networks are also fraudulent activities carried out on the telecom network.

In the case of credit card/smartcard fraud, it is normally and mostly noticed in the financial industries which inculcate technologies such as the electronic banking. The financial industry has suffered severe loss in the hand of the fraudsters over the years. Detecting the fraud has been a difficult task because, the pattern used in executing the fraud are not continuous, as they change from time to time. Once and again, a measure for detecting the financial fraud using the smartcards have been developed, but instead of curbing the attack, the fraudsters will always develop another pattern for carrying out fraudulent activities. The smartcard technology is another widely accepted technology due to its relevance in the implementation of a “cashless society” policy, however, these attacks on the users is posing a big threat to its continuous advancement as the customers are beginning to doubt the safety of their transactions using smartcards. The effect of fraud on smartcards affects both the users and the financial institutions, losses ranging from lawsuits by the customers who have been defrauded are becoming a normal occurrence in this industry. Research has it that the increase in technological advancement of the world is directly proportional to the increase in fraud in such sectors.

The use of artificial neural network (ANN) adds a true artificial intelligence to the security defences of the system, rather than the security measures that have been previously implemented. The ANN can be extremely helpful in modelling a complex transactional pattern. Therefore, it is suitable for a smartcard and telecommunication fraud detection system.

1.1 Aim and Objectives

The aim of this work is to develop an enhanced fraud detection model using neural networks for the telecommunication and smartcard in Nigeria. The specific objectives are to:

i. Design a secure smartcard and telecom transaction platform for new and existing users of the smartcard and telecom technologies.
ii. Develop fraud detection system that will diagnose and block suspected fraudulent activities via the smartcard and telecom platform
iii. Implement with Hypertext Pre-processor (PHP), JavaScript (JS), Hypertext Markup Language and MySQL as backend.
iv. Compare result with the existing system performance.

1.2 Credit Card Fraud

Fraud can be classified as any activity with the intent of deception to obtain financial gain by any manner without the knowledge of the cardholder and the issuer bank. Credit Card fraud can be done in numerous ways.

Credit card fraud has been causing many financial losses for the customer and the organization. In recent years this subject has been a growing line of research, techniques such as machine learning are used to detect and block fraudulent transactions.

According to the Federal Trade Commission of the United States, figures on fraud and credit card identity have been increasing in recent years, with 13 million of claims from 2012 to 2016 of its online database of consumer complaints. Statistics on Credit Card Fraud registered 1.3 million complaints in 2016, corresponding to 42 % of total complaints [4]

Recognizing misrepresentation or fraud is a difficult issue since fraudsters cause their conduct to seem real. Another trouble is that the quantity of real records is far more noteworthy than the quantity of false cases. Such unequal sets require extra safety measures from the information expert. The way to exact extortion location lies in the improvement of dynamic frameworks that can adjust to new fraud pattern or designs [5]

Fraud detection involves discovering a fraud activity in the midst of thousands of authentic ones, which could be very difficult and challenging. "With continued advancement in fraudulent strategies it is important to develop effective models to combat these frauds in their initial stage only, before they can take to completion" [6] The big challenge in developing such a model is that the number of fraudulent transactions among the total number of transaction is a very small number and hence the work of finding a fraudulent transaction in an effective and efficient way is quite perplexing.

Some common types of credit card fraud include:

- *Application Frauds:* This occurs when the fraudster gains control of the application system by accessing sensitive user details like password and username and open a fake account. It generally happens in relation to the identity theft. When the fraudster applies for credit or a new credit card altogether in the name of the cardholder. The fraudster steals the supporting documents in order to support or substantiate their fraudulent application.
- *Electronic or Manual Credit Card Imprints:* When the fraudster skims information that is placed on the magnetic strip of the card. This information is very confidential and by accessing it the fraudster may use it for fraudulent transactions in future
- *CNP (Card not Present):* When the fraudster knows the expiry date and account number of the card, the card can be used without its actual physical possession.
- *Counterfeit Card Fraud:* It is generally attempted through the process of skimming. A fake magnetic swipe card is made and it holds all the details of the original card. The fake card is fully functional and can be used to commit transactions in future.
- *Lost and Stolen Card Fraud:* In cases when the original card holder misplaces their card, it can get to the hands of fraudsters and they can then use it to make payments. It is hard to do this through machine as a pin number is required however; online transactions are easy enough for the fraudster.

1.3 Detecting Credit Card Fraud using Neural Networks

An artificial neural network (ANN) is a lot of interconnected hubs intended to copy the working of the human mind (Ghosh and Reilly, 1994). Every node has a weighted association with a few different nodes in contiguous layers. Singular nodes or hubs take the information got from associated nodes and utilize the loads together with a straightforward capacity to figure yield esteems. Neural systems or network architecture come in numerous shapes and designs. The Neural system design, including the quantity of shrouded layers, the quantity of nodes inside a particular concealed layer and their availability, most be indicated by client dependent on the intricacy of the issue.

The standard of neural network is roused by the elements of the brain particularly pattern acknowledgment or recognition and acquainted memory. The neural network perceives comparable patterns, predicts future qualities, values, or occasions dependent on the affiliated memory of the pattern it was learned. It is broadly applied in order and grouping. The benefit of neural network over different procedures is that these models can gain from an earlier time and hence, improve results over the long haul. They can likewise extricate rules and anticipate future movement dependent on the present circumstance. By utilizing neural systems, viably, banks can distinguish deceitful utilization of card, quicker and even more proficiently. Among the revealed credit card fraud considers most have concentrated on utilizing neural networks. In progressively reasonable terms, neural networks are non-direct factual data modelling apparatuses or tools. They can be used to model complex connections among information sources and yields or to discover designs in the information. [7].

There are two stages in neural network training and acknowledgment [7]. There are two types of neural network (NN) training methods regulated and unregulated methods. In regulated training method, samples of both fake and non-deceitful records used to create models. The regulated training method just looks for that transaction, which is generally divergent from the standard. On the other hand, the unregulated training methods do not need to bother with the past information on fraudulent and non-fraudulent transactions in the database. Neural networks (NNs) can deliver the best outcome for just enormous transaction dataset. They need long training dataset.

One of the merits of utilizing unsupervised neural networks over similar techniques is that these techniques can learn from data stream. The more information went to a SOM model, the more adjustment and enhancement for result is obtained. All the more explicitly, the SOM adjusts its model over the long haul. Accordingly, it can be utilized and updated online in banks or other money related enterprises. Subsequently, the fraudulent utilization of a card can be recognized quick and successfully.

In any case, neural systems have a few disadvantages and challenges, which are predominantly identified with determining appropriate engineering in one hand and over the top preparing required for coming to best execution in other hand.

1.4 Telecommunication Fraud

The telecommunication industry has extended significantly over the most recent couple of years with the advancement of reasonable cell phone innovation [8]. With the expanding number of cell phone supporters worldwide, cell phone fraud is likewise set to rise. It is an overall issue with significant yearly income misfortunes of numerous organizations. Media transmission fraudulent which is the centre is engaging especially to fraudsters as calling from the portable terminal is not bound to a physical area and it is anything but difficult to get a membership. This gives a way to illicit high benefit business for fraudsters requiring negligible speculation and moderately generally safe of being captured. Telecommunication fraud is defined as the unapproved use, altering or control of a cell phone or administration.

The procedure starts with social occasion verifiable information on fraudulent and non-fraudulent calls. This data is pre-processed to make it reasonable for neural network learning. Next, the neural network is prepared utilizing the pre-processed information to construct a model, which consolidates various examples of false conduct. The model is applied to approaching business where it adaptively learns new examples of misrepresentation improving its model as the kinds of extortion advances. Fraud discovery mechanism might be home blend, exclusive or a blend of both which is most likely the most advantageous methodology [9].

The self-learning framework or system furnishes the general system with the ability to take in new standards about fraudulent from the submission of fraudulent and non-fraudulent space explicit data and naturally distinguish sporadic perceptions in the data in this manner giving an input system to advancing and refreshing the vault of rules. Self-learning system coordinates reasonable calculations for measurable information examination and information mining undertakings that empower it to refresh, improve and expand existing misrepresentation discovery administers by dissecting submitted information. The extortion recognition territory is a functioning region of improvement for neural systems in broadcast communications. A large number of the best frameworks are crossover systems, which exploits the overall qualities of a few AI advances. Given the adjustments included, it is an application, which should come into routine use in the years ahead.

II. RELATED WORK

Jain [6] proposed a comparative analysis of various credit card fraud detection techniques. They discussed various credit card related frauds and provided a concrete review of the various techniques that are currently in place for detecting the frauds. Some of the techniques discussed include the Support Vector Machine (SVM), Artificial Neural Network (ANN), Bayesian Network etc. they analyzed the existing techniques based on quantitative measurements such as rate of detection and the rate of false alarm witnessed with the existing techniques. From their analysis, they arrived at a conclusion that the techniques were not able to detect all types of credit card fraud and when they did they detected the fraud after it has already been perpetrated and not in real time. Also these techniques were not properly trained in order to block fraudulent transactions. Finally, the techniques also lacked cross platform adaptability quality.

Kabari [3] proposed the telecommunications subscription fraud detection using the artificial neural network. They presented the design and the implementation of the fraud detection system in the work. They used Neuro-solutions for Excel to implement the ANN. Their system was subjected to performance testing and was discovered to be user friendly and recorded an 85.7% success and accuracy rate.

Johnson [10] proposed a Medicare fraud detection system. The system was implemented to detect the fraudulent activities carried out within the medical sector, in terms of falsification of health record and insurance records. They evaluated the performance of six deep-learning methods for addressing class imbalance using the CMS medical care data and LEIE fraud labels. They also considered a range of class distribution and studied the relationship between minority class size and the optimal division threshold.

Singh [11] proposed the electronic credit card detection system by collaboration of machine learning models. They focused on fraud activities that cannot be detected manually by carrying out research. They used a dataset of electronic payment card holders. Then they applied the machine learning techniques on the unstructured and process free data.

Malek [2] proposed a fraud detection and prevention in smart card based environments using Artificial intelligence. They considered the possibility of implementing neural network fraud engine on a smart card platform. The system was smart enough to suspend and block any suspicious or unusual fraudulent activity. The usage characteristics of the users were used as parameter for detecting fraud that is connected to the users. Java card 2.1 compliment was used for implementing the fraud engine. The system was tested on a simulator machine implemented on the PC. The engine had an execution time of 4.24 ms which was achieved bt recording 500 runs at 2.12 seconds. However, the execution time was too slow for effective fraud detection. This implies that a smart fraudster would successfully perpetrate his fraud and go uncaught before the system can detect it.

Oumar and Augustin [1] proposed credit card detection using ANN. In their work, data consisting of fraudulent and non-fraudulent activities were used as parameters to detect future fraud. The parameters were used to create a model that classifies the transaction with a high accuracy based on a machine learning technique.

ANN with Logistic Regression were used as a means of measurement and in order to achieve high accuracy, they refined the model using Back propagation which had proved to record high accuracy in time passed in order to help the model differentiate between fraudulent and non-fraudulent activities. However, their models are too ambiguous and not time and cost efficient.

Amanze and Onukwgha [12] proposed a review of the credit card fraud detection system for Nigerian banks using Adaptive mining and intelligent agents. They were able to design and implement a credit card fraud detection system for the banking industry in Nigeria. Their model consisted of a hybrid model which combines evidence from current and previous transaction behaviours of the customer in order to detect the suspicion level of each incoming transaction. Their research also gave a statistics of the fraud rate in Nigeria via credit card transaction. However, their fraud detection accuracy was low and called adequate improvement by the use of better models and algorithms for the system's implementation.

Amanze [13] proposed accredit card fraud detection system using intelligent agents and enhanced security measures. The aim of the system was to detect the fraud while it is going on by sending a token to the customer and for more security checks, the system would also ask the user secret questions which only the users can provide answers to. If the answers provided are correct, the transaction will be flagged as successful. But if the answers are wrong, the transaction will be tagged as fraudulent and an SMS will be sent to the customer and the bank to notify them of the fraudulent activity. However, this system does not automatically block confirmed fraudulent transactions but waits for the customer and the bank to take actions on the fraud detected. This system basically just provides information about proposed fraud and does nothing else about it.

Daliri [14] proposed a study that uses harmony search algorithm in neural networks to improve fraud detection in the banking system. In the proposed method, hidden patterns between the fraudulent and non-fraudulent customer's information were searched. This system detects the fraudulent activity and immediately blocks it before it takes place. The approach of this system was considered to be the best because of its proactive nature which is a booster for every fraud detection system. However, the system design of this system is ambiguous and time consuming, it is also not cost efficient for small scale developers.

Delamaire [15] proposed a study to discuss credit card fraud detection techniques. In their study, they pointed out different types of credit card fraud by reviewing several articles on credit card fraud. Then, they identified the available techniques at that time for detecting credit card fraud. Some of the fraud types identified were bankruptcy fraud, counterfeit fraud and behavioural fraud. Some of the prescribed techniques were the genetic algorithm, clustering techniques, neural networks, pair-wise matching and the decision tree. However, they could not build a suspicion scorecard that can predict fraudulent behaviours while taking into account the field of behaviour that can relate to the various types of credit card fraud.

Zanin [16] proposed credit card fraud detection through parenclitic network analysis. They presented a "a first hybrid data mining complex network classification algorithm", which had the capability to probe and discover fraudulent instances in a real card transaction dataset. Their decision to use that particular model was as a result of a network reconstruction algorithm that allowed representation of the derivation of one instance from a reference group to be created. They demonstrated how addition of features derived from a network data representation improved the score obtained by a standard neural network-based classification algorithm and additionally how this combined approach can out performs a commercial fraud detection system in certain operation niches. They also tested the feasibility of using complex networks to improve data mining model. However, their model could not detect credit card fraud such as bankruptcy fraud and other difficult credit card fraud that seem difficult to detect.

Zhang [17] proposed a model based on convolutional neural network for online transaction fraud detection. They presented a model for detecting fraud in the field of online transaction using the convolutionary neural network which constructs an input feature sequencing layer that implements the reorganization of raw features from different feature combination entering the convolution kernel to form different convolutional patterns. The model could collect low and non-derivative online data as input. The whole network consisted of a feature sequencing layer, four convolutional layers and pooling layers, and a fully connected layer. The system was tested using online transaction data from the commercial bank. Though their system produced a great accuracy rate in detecting fraud up to 94% according to their evaluation, however, they did not pay more attention to the discovery of sequence characteristics of transactions.

Sadgali [18] proposed fraud detection in credit card transaction using neural networks. They examined three advanced data mining techniques namely neural networks (NN), multiplayer perceptron layer (MPL) and convolutionary neural networks (CNN). They used a unique generic credit card dataset to evaluate the performance of their model. In the evaluation they compared and analysed each of the technique. After the analysis, they postulated that the MLP was best for detecting credit card fraud when applied to the generic dataset used. However they could not present a complete architecture of an adaptive model for credit card fraud detection using the result of their analysis.

Abakarim [19] proposed an efficient real time model for credit card fraud detection based on deep learning. Their model was based on an auto-coder that permits classification in real time credit card transaction as either genuine or fraudulent. Four different models classification models were used as a comparison to test the efficiency of their model. In terms of accuracy, recall and precision, benchmark showed better results for their model than those existing at that time since it recorded the F1 score.

III. MATERIALS AND METHODS

3.1 Methodology

We adopted System Development Lifecycle Methodology (SDLC) in this approach.

3.2 Analysis of the Existing System

Their dissertation focused on credit card application which was used to detect the fraudulent credit card activities on credit transaction. In this peculiar type, the pattern of current fraudulent usage of the credit card was analyzed with the previous transactions, by using the intelligent agent in data mining algorithm. Fig. 3.1 shows the architecture of the existing system model. The system has three data mining engines: customer/bank database, credit card transaction database and fraud detection database. The customer/bank database had the following: opening of account operation, withdrawal and deposit transaction and credit card transaction. Fraud techniques database gave details of attack attempts on customer's credit card. The credit card database contained all the previous credit card transactions carried out by the customer. The existing Credit Card Fraud Intelligent Agent Model (CCFIAM) was to detect the credit card fraud by analysing the spending patterns on every card and figure out any inconsistency with respect to the usual spending patterns. Intelligent agent was made use of these inputs (from user transaction input and past recorded credit fraud detection input) and watched ongoing transaction to check whether is fraudulent or not, beginning from the most recent attack methods of fraudsters and concentrating the most recent spending pattern of the transaction.

In the existing system, when a credit card transaction is initiated, the system verifies the user's pin code and username by validating it on the bank database. If the pin fails to validate after three consecutive attempts, the transaction was denied and fraud alert sent to the fraud database. But if the pin verification was successful, the system will capture the credit card transaction details and verify the credit card information before passing the information to data monitoring agent.

The monitoring agent used the last ten credit card transaction to build a transaction pattern for the customer and forward the pattern to the collating agent. The Monitoring agent also used data mining technique to retrieve previous credit card fraud patterns from the credit card database and also retrieve the customer details from the bank database. At monitoring agent, each of the agents focused on a particular type of credit card fraud, the agents ran in parallel and reported any suspicious attack to collating agent. However, the collating agent was responsible for communication with the diagnosing agent, which includes sending the task to be performed as input and providing the required data. The diagnosing agent matched the existing pattern of credit card transaction with the new transaction to check if there are variations in the pattern. If the transaction pattern does not match, the system will request for a secret question and answer from the user for more authentication. If the user fails the question, a fraud alert is send to the reporting agent. The reporting agent will then forward the extracted credit card transaction status to the database of the bank and the customer's phone and the transaction is denied. But where the credit card profile matched with the existing customer profile, the transaction is allowed to go through and the customer's account updated. At this, the transaction will be recorded on the credit card database and the fund transferred will be deducted from the customer's account balance.

3.2.1 Explanation of the Existing System Components

i. *Intelligent Agent:* These agents were responsible for ensuring that the transaction carried out via the credit card are by a valid user by ensuring that the details of the transaction are valid, detecting any suspicious activity and reporting them to the original owner of the account and the bank also.

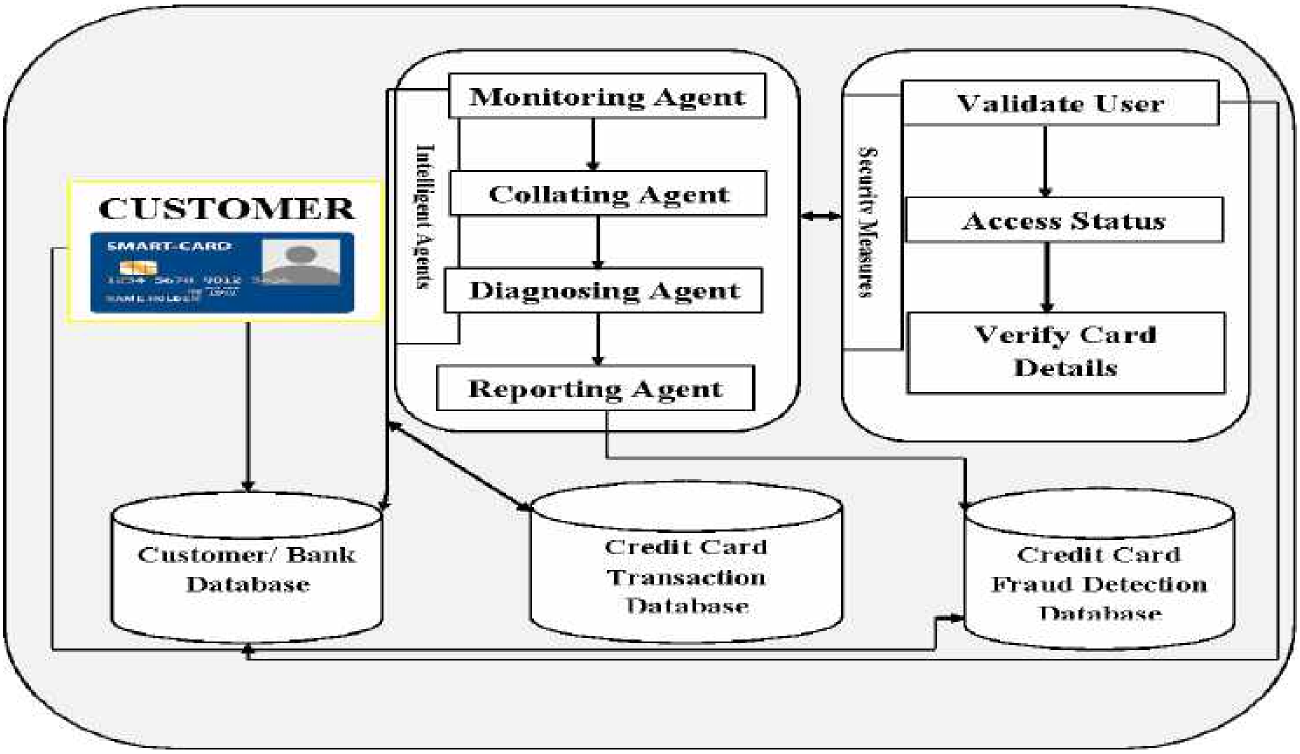


Fig.3.1: Cradit Card Fraud Detection System using Intel ligent Agents and Enhanced Security Features (Existing System.Source:Amanze[13])

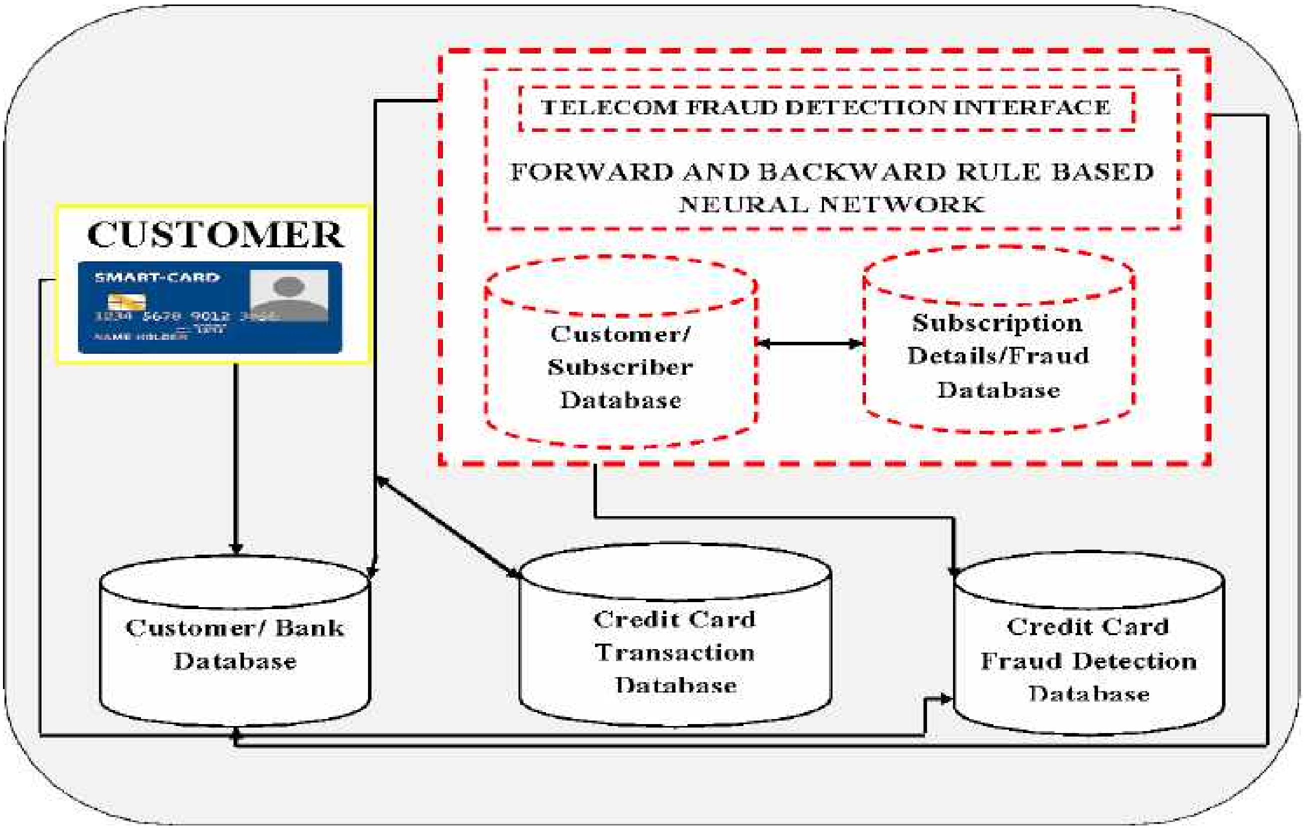


*Fig.3.2:*An Enhancedfraud Detection Model using Natural Networks for Telecom and Smartcards(Proposed System)

ii. *Security Measures:* These were measures put in place to aid the user interact with information on his/her account and to view the activities on his account. It also helps the system to confirm that the information provided by the user corresponds to the one previously stored.

iii. *Customer:* This phase introduces the user of the smartcard at the time. The user uses the smartcard to initiate a transaction which will either be flagged as successful or otherwise depending on the security check results of the system.

iv. *Customer Bank Database :* The customer/bank database has the following: opening of account operation, withdrawal and deposit transaction and credit card transaction. Details of the customer's personal information and secret pin are some of the data that are stored in this database.

v. *Credit Card Transaction Database:* The credit card database will contain all the previous credit card transactions carried out by the customer. Once a new transaction was carried out, whether successful or not, it was automatically be stored in this database.

vi. *Fraud Detection Database:* Fraud techniques database will gave details of attack attempts on customer's credit card. Details such as the time, date and at which service point the fraudulent activity was perpetrated will be stored in this database.

3.2.2 Advantages of the Existing System

i. The adaptive data mining and intelligent agent's model introduced a more secured communication channels for credit card transactions thereby preventing loss of money by the customers to credit card fraudsters.

ii. The existing system provided a kind of confidence in the customers that they are sending their personal information to legitimate banks' servers and not impostors. This helped to boost the electronic transactions thereby reducing the queue in the banking halls.

iii. The fraud detection system ensured that all critical data (credit card numbers, for example) were encrypted and that only authorized users have access to data in its entirety.

iv. The existing system was featured with alert system to enable e-commerce owners receive alert of fraudulent activities and automatically disable customer's (victims) account involved.

3.2.3 Disadvantages of the Existing System

i. The existing system's security measure for checking and detecting fraudulent activity is not effective enough to detect the recent forms of credit card fraud.

ii. The system does not provide any measure to detect telecommunication fraud in their system design.

iii. The execution time of the existing system is not fast enough to detect fraud in real time and the system does not include measures to block the credit card fraud account after two unsuccessful tries.

3.3 Analysis of the Proposed System

The proposed system is an enhanced fraud detection model using neural networks for telecommunication and smart cards. It is an improvement of the work carried out by Amanze [9]. the proposed system is a reliable smartcard and telecommunication fraud detector, due to the use of neural networks to train the system based on historical data from customer smartcard transaction, historical subscription details, and credit card fraud technique database. The proposed system follows the same principle of the existing system, but uses a better model based on neural network to detect fraudulent activities. The customer/bank database has the following: opening of account operation, withdrawal and deposit transaction and smartcard transaction. Fraud techniques database gave details of attack attempts on customer's smartcard. The smartcard database will contain all the previous credit card transactions carried out by the customer.

For the telecommunication fraud detector, the historical data is collected and normalized to reduce redundancy in the data and then used to train the system based on set of specified rules which will be specifies by the system. The rules are stored in database of rules and the system will always refer to these rules to classify a transaction as fraudulent or not. If a user initiates a subscription request, the user's details are checked against the details in the historical user details dataset, if the details match, then the user will be further verified to check if it's a domestic or commercial user. The domestic are usually provided with connections at an affordable rate, and the commercial are provided with connections at a higher rate because of the higher scale of usage and subscription by these users. This also implies that the commercial users will request for higher data subscription plans that the domestic users. Therefore, once the subscription request is launched by the subscriber, the system will use the information provided by the subscriber to verify if it's a domestic or commercial user. If the user is a domestic user and has requested for the connections and subscription of the commercial user, the system will flag the request as unsuccessful and will block the user's access to that particular service provider upon more than three retries.

On the other hand, for the smartcard user, the customer is assumed to have previously registered and has his/her details saved before being issued the smartcard by his/her bank. Also his transaction activities on that card has been monitored and saved. If the user launches a request to use the card either for withdrawal or to purchase an item, the secret pin will be required, if the user provides the correct pin in two trials, the transaction will be flagged successful. However, if the pin entered is incorrect twice, the transaction will first be flagged unsuccessful, and user will be referred to a page where he will be required to provide sensitive personal information that can only be known by the user, if the user successful supplies these answers, he/she will be required to reset his secret pin to the one he can remember and then retry the transaction which will now be successful. Else, the transaction will be tagged as fraudulent and blocked.

In this proposed system, the forward and backward rule based neural network is used to train the system to detect and block fraudulent activities on smartcard and telecom services in realtime before they even occur. The rule based approach works in the hidden layer of the neural network to work on the input available (historical dataset) to train the input in order to achieve accurate classification of transactions and detections if any is suspected to be fraudulent.

3.3.1 Explanation of the Proposed System Components

i. *Telecommunication Fraud Interface:* This provides an interface for the user to request for data subscriptions from his service provider. On the side of the service provider, i.e. the telecom service provider, they use this platform to verify the user and his request.

ii. *Neural Network:* This model helps to verify a users transactions using a set of laid down rules (**"ifs"**), it helps to train the system to classify user transactions as authentic or fake based on laid down rules already specified and used to train the system.

iii. *Customer / Subscriber Database:* This component consists of three vital phases known as the distribution phase which distributes letters, words and lines for the analysis and interpretation in a specific order. The disposition phase involves the disposition of texts within the page taking into account the four margins, (page margins; sides of the page each have a meaning) that must be large enough cared for with a harmonic heading. Furthermore, the proportion phase is the equilibrium of the dimensions of letters between each other. It means there is equilibrium in sense of humor and judgment upon judging.

iv. *Subscription Details / Fraud Database:* This database contains all the previous subscription activities of the user of the SIM. It also contains all the details of the fraudulent activities carried out via the SIM card.

3.3.2 Advantages of the Proposed System

The following advantages of the Proposed System include the:

i. The forward and backward rule based neural network model introduced a more secured transaction using telecommunication and smartcard based services to prevent losses to fraudsters..
ii. The proposed model boots the confidence of smartcard users due to the assurance that any suspected fraudulent activity on their account will be blocked without their participation and that their money will not be accessed without their permission.
iii. The fraud detection system ensures that all critical data (credit card numbers, for example) were protected and that only legitimate users have access to the data.
iv. The proposed system was featured with alert system to enable the users and the service providers (banks and telecom companies e.g. MTN) receive alert of fraudulent activities and then perform certain operation to ensure that the customer's data and resource is protected and safe.
v. The proposed system automatically blocks all suspected fraudulent activities in realtime before they can even occur. Hence, is the most efficient solution to telecom and smartcard based fraud.

3.4 Existing System Algorithm

Step 1:

To Identify The Profile Of Cardholder From Their Purchasing

Step 2:

The Probability Calculation Depends On The Amount Of Time That Has Elapsed Since Entry Into The Current State.

Step 3:

To Construct The Training Sequence

For Training Model

```
/*initialization*/
S = { };
For (a ∈ Accts) Do Cover [a] = 0;
For (r ∈ Rules) Do
Occur[r] = 0; /*number Of
Accounts In Which R Occurs* Acctsgen[r] = { }; /*set Of Accounts Generating R */
End For
Check The Previous Spending
Profile
For (a ∈ Accts) Do
Ra = Set Of Rules Generated
From A;
For (r ∈ Ra) Do
Occur[r] : = Occur[r] + 1;
Add A To Acctsgen[r];
End For; End For
If Transaction Is Outside Spending Profile The Send Alert To Monitoring Agent
For (a ∈ Accts) Do
Ra = Secret Questions;
Request For User To Supply Secret Question And Answer
While (cover [a] <trules) Do
R = Correct From Ra
Remove R From Ra
If (r ∉ S And Occur[r] ≥ Taccts ) Then
Add R To S; 24. For (a2 ∈acctsgen[r]) Do
Cover [a2] = Cover[a2] + 1; 26. End For; End If
End While; End For Intelligent Agents Report Back To Transaction Agent If Any Rule Is Broken Transaction Agent Stores The Alert Received Monitoring Agent Supervised By Manager Or Rollback The Transaction Before Being Committed To Database
```

Detection Phase: Fraud Detection

Step 1:

To Generate The Observation Symbol

Step 2:

To Form New Sequence By Adding In Existing Sequence

Step 3:

To Calculate The Probability Difference And Test The Result With Training Phase

Step 4:

Finally, If Both Are Same It Will Be A Normal Customer Else There Will Be Fraud Signal Will Be Provided.

3.5 Proposed System Algorithm

Step 1:

To Identify The Profile Of Cardholder From Their Purchasing

Step 2:

The Probability Calculation Depends On The Amount Of Time That Has Elapsed Since Entry Into The Current State.

Step 3:

To Construct The Training Sequence For Training Model

```
/*initialization*/
S = { };
For (a ∈ Accts) Do Cover [a] = 0;
Set Rules For Transactions
For (r ∈ Rules) Do
Occur[r] = 0; /*number Of Accounts In Which R Occurs*/
Acctsgen[r] = { }; /*set Of Accounts Generating R */
If Transaction Is Outside Spending Profile The Send Alert To Monitoring Agent
Request For Secret Pin To Proceed
If Secret Pin = Incorrect * 2
Flag Transaction As Unsucessful
For (a ∈ Accts) Do
Ra = Secret Questions;
Request For User To Supply Secret Question And Answer
While (cover[a] <trules) Do
Suggest Secret Pin Reset
Else
If (ra. Answer ∉ Answer) Then
Block Transaction In Realtime Before It Occurs While System Report Back To Transaction Agent If Any Rule Is Broken Transaction Agent Stores The Alert Received Monitoring Agent Supervised By Manager Or Rollback The Transaction Before Being Committed To Database
Else If Secret Pin = Correct Do
Execute Transaction
```

Step 4:

Store All Information In Their Respective Databases

Step 5:

Repeat Step 1-5 For New User Until New Transaction = 0.

Step 6:

Quit.

Telecom Fraud Detection Phase:

Step 1:

Login

Step 2:

Request For Connection Via Subscription

Step 3:

Verify Status Of User (domestic/commercial)

Step 4:

If Domestic User Request = Sm (small Connection) Then

Subscription = Successful. Else

Request = Unsuccessful

Step 5:

Flag Transaction As Fraudulent

Step 6:

Block Transaction.

Step 7:

Store Transaction In The Fraud Database

Step 8:

Quit.

IV. RESULTS AND DISCUSSION

4.1 Choice and Justification of Programming Language used

We implemented the Proposed System design with PHP, JavaScript Programming Language, Hypertext Markup Language, Cascading Style Sheet and MySQL Relational Database Management System. Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript.

The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers. JavaScript is a lightweight, interpreted programming language, designed for creating network-centric applications, complementary to and integrated with Java, complementary to and integrated with HTML, open and cross-platform

PHP is a programming language for building dynamic, interactive Web sites. As a general rule, PHP programs run on a Web server, and serve Web pages to visitors on request. One of the key features of PHP is that you can embed PHP code within HTML Web pages, making it very easy for you to create dynamic content quickly. PHP is a server-side, HTML-embedded scripting language that may be used to create dynamic Web pages. It is available for most operating systems and Web servers, and can access most common databases, including MySQL. PHP may be run as a separate program or compiled as a module for use with a Web server.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons: MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

4.2 Discussion of Results

Fig 4.1 shows the home page of the fraud detection system. It contains navigation buttons for accessing either of the two fraud detection system. From this page, a user can carry out either a telecom subscription transaction or a smartcard based transaction. Fig 4.2 contains the welcome page for the telecoms services. Users who intend to subscribe their phones and other devices will click on the manage subscription button to navigate to the subscription request page.

The request page contains a form that the user will be required to fill. Some of the labels of the form include the customer type and the amount of the subscription (in Naira). When a user submits this form, the system automatically verifies the information provided with that stored in the database, the first rule is to verify if the user actually exists, the second rule is to verify that the user is requesting for the subscription that is suited for his type of user.

If the user passes the second rule, the output as shown in Fig 4.3 is gotten and the transaction is the request is granted and tagged as a successful request.

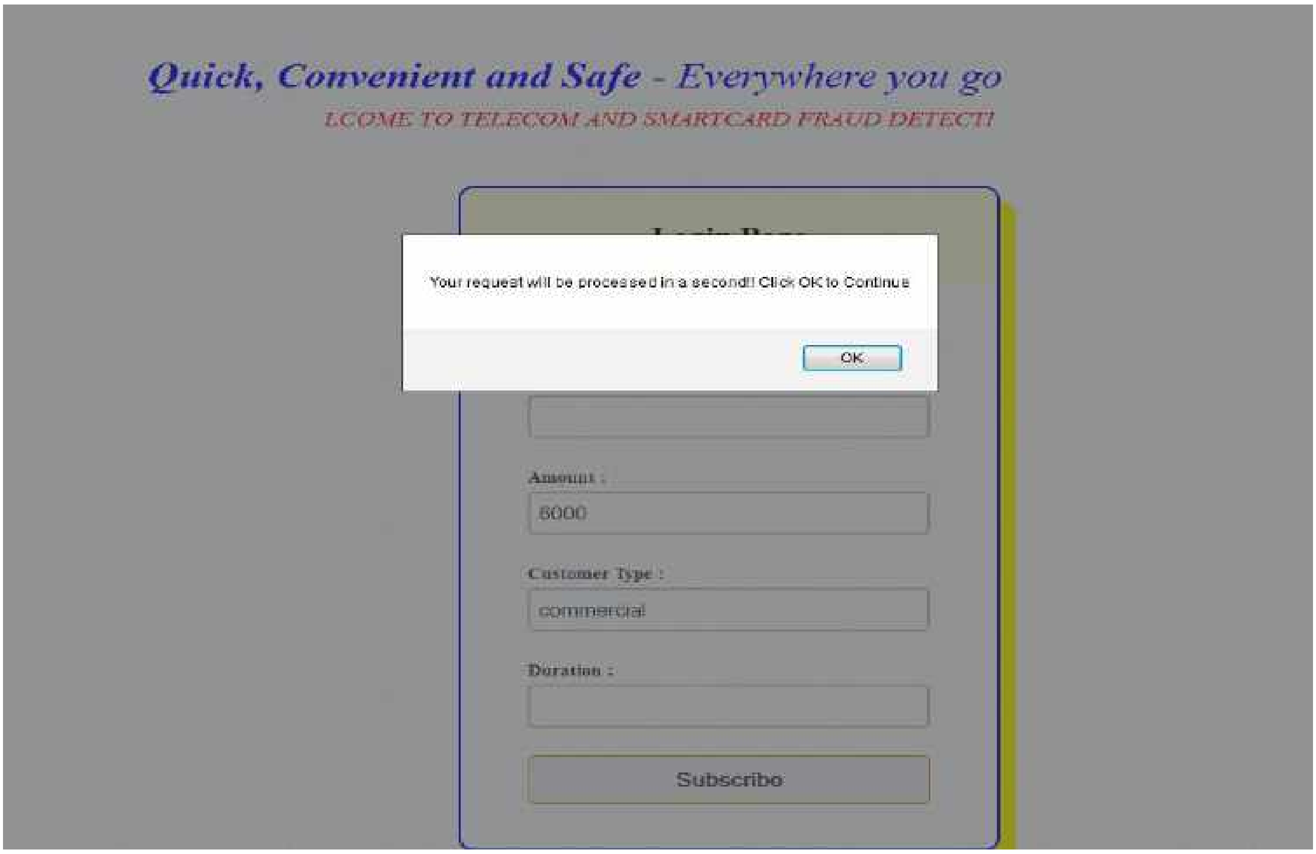
On the other hand, if the first rule is failed, the system is automatically navigated back to the homepage, and no transaction can be initiated. If the first rule checks and the second rule is flaunted, the user is notified that the requested subscription is not available for his/her type of user, the user is also given the opportunity to retry the request (Fig 4.4). However, if the user retries three more times and insists on the wrong subscription (Fig 4.5, Fig 4.6), the transaction is flagged as fraudulent and blocked automatically (Fig 4.7). The SIMcard of the user is blocked temporarily and can only be unblocked by the telecom company, after verifying the authenticity of the user who would have to provide certain legal documents before this can be implemented. These checks are important because of the losses incurred by the telecom sector as a result of users making subscription requests that are not within their service range or requesting for lower subscriptions than he should while still enjoying the quality of service that belongs to a higher user or subscriber.

Fig.4.1: Fraud Detection Home Page

Quick, Convenient and Safe - Everywhere you go

For All Your Data Needs

*Fig.4.2:*Telecom Welcome Page



Your request will be processed in a second!! Click OK to Continue

Amount : 6000

Customer Type : commercial

Duration :

Fig.4.3: Telecom Subscription Validation Page(1)

You have 2 attempts left.

User ID :

Amount : 20000

Customer Type : domestic

Duration :

Fig.4.4: Telecom Subscription Validation Page(2)

WELCOME TO MTN TELECOMS

Quick, Convenient and Safe - Everywhere you go

WELCOME TO TELECOM

You have 2 attempts left;

User ID :

Amount : 20000

Customer Type : domestic

Duration :

Fig.4.5: Telecom Subscription Validation Page(3)

WELCOME TO MTN TELECOMS

Quick, Convenient and Safe - Everywhere you go

WELCOME TO TELECOM AND SMARTCARD FRAUD

Your account has been temporarily Suspended due suspected fraudulent attempts

User ID :

Amount : 20000

Customer Type :

Fig.4.6: Telecom Transaction Blocking Output

FRAUD ALERT!!! TRANSACTION BLOCKED

Fig.4.7: Telecom Fraud Detection Output

SMARTCARD FRAUD DETECTION SYSTEM

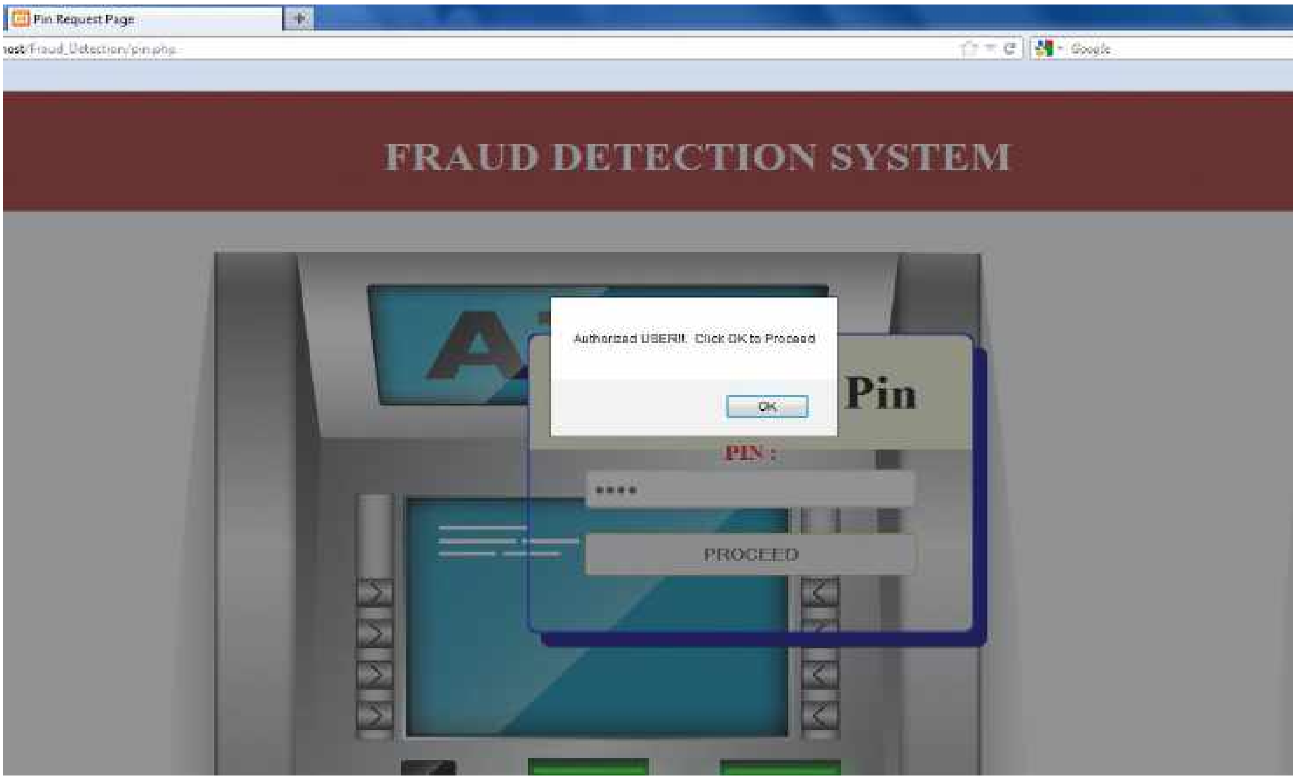
PLEASE INSERT YOUR SMARTCARD

Fig.4.8: Smartcard Initialization Page

Fig 4.9: Smartcard Transation Page

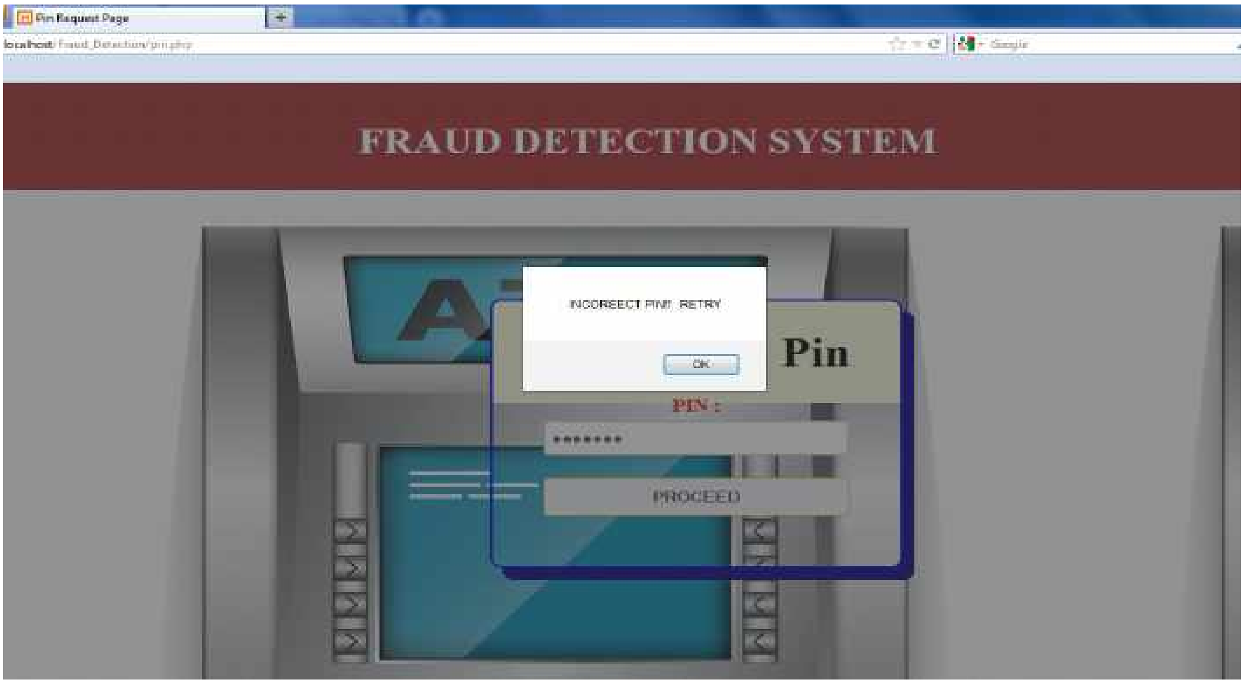
The smartcard transaction system as depicted in Fig 4.8, requires the use of a smartcard as the first input before any transaction can be iniatiated. Every user of the service is assumed to have pre-registered and thereby issued a smartcard to carry out certain transactions. Once the smartcard is inserted into the system: either the POS (Point of Sale) for purchase of goods and services, or the ATM (Automated Teller Machine) for withdrawal of cash and other services; the system recognizes the user and navigates him/her to the transaction page (Fig 4.9) . In this page, a list of transactions are listed out which the user can perform such as withrawal, quickmoni, transfer etc. any of the options chosen will require a secret pin from the user to perfom it. Fig 4.10 shows the output of a valid user pin. Thi means that the pin keyed in by the user is valid and the user is authentic.

However, if the user provides an invalid pin the output in Fig 4.11 will be displayed suspending tha transaction and giving the user one more retry (Fig 4.12) and if this also fails, the user is navigated to page where the final verification is carried out to check the authenticity of the user. the user is required to provide an answer to a tricky question (Fig 4.14) which only the user can have the answer to (such as the postion of a birthmark in your body, how many birthmarks you have etc.), if this answer is correct, the uer will be given an option to reset his/her secret pin to the one they can easily remember.the user can now perform the transaction sucessfully. But if once again, the tricky question's answer provided by the user is incorrect, the transaction is automatically classifies as fraudulent and saved in the smartcard fraud database, also the transaction is blocked and flagged as frudulent (Fig 4.16) while sending an alert to the user and the financial institution about the attempted fraudulent attack



Authorized USER!!. Click OK to Proceed

Fig 4.10: Smartcard User Validation Output(1)

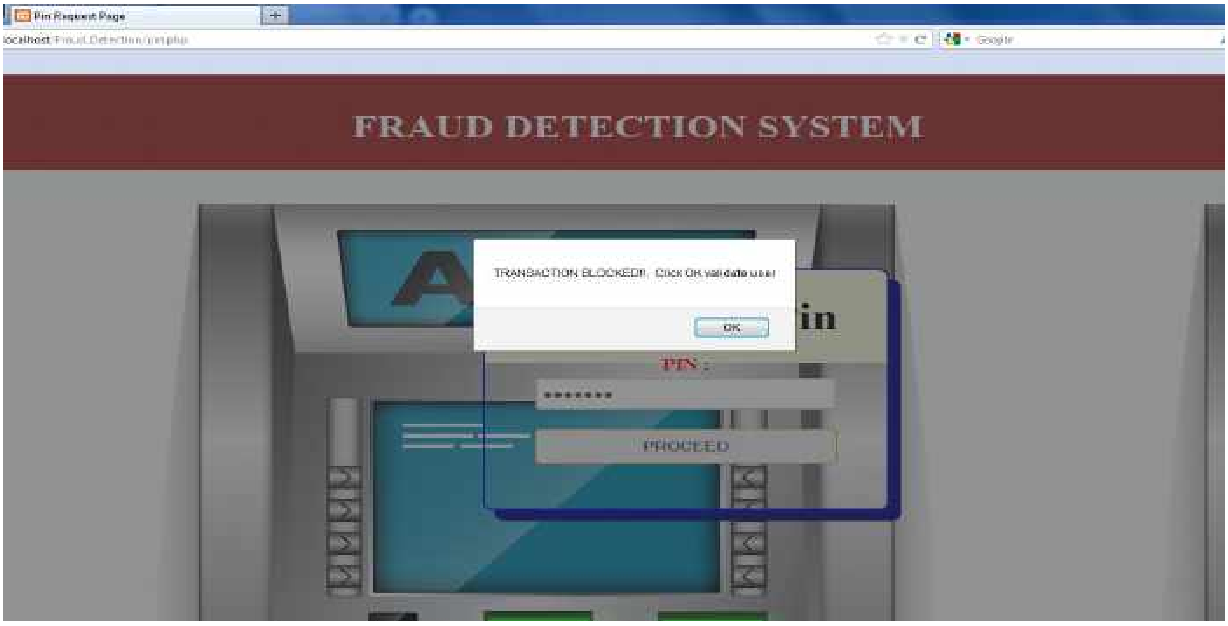


INCORRECT PIN!! RETRY

Fig 4.11: Smartcard User Validation Output(2)

You have 1 attempts left.

Fig 4.12: Smartcard User Validation Output(3)

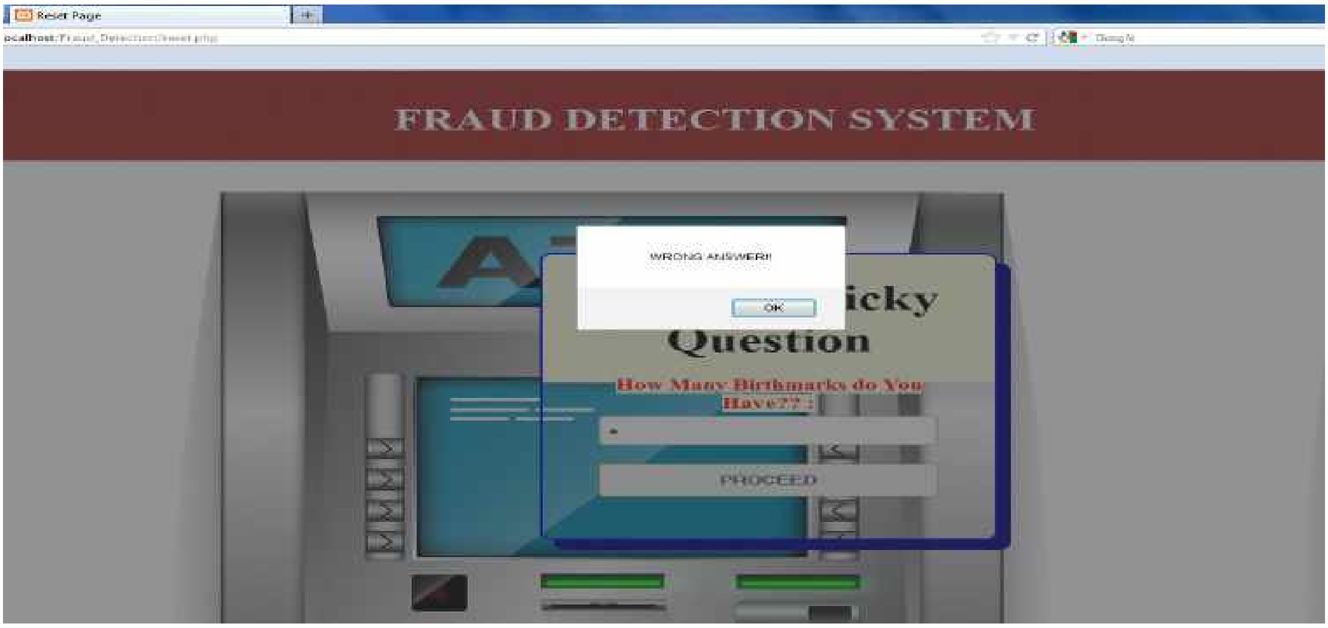


TRANSACTION BLOCKED!! Click OK validate user

Fig 4.13: Smartcard User Validation Output(4)

How Many Birthmarks do You Have?? :

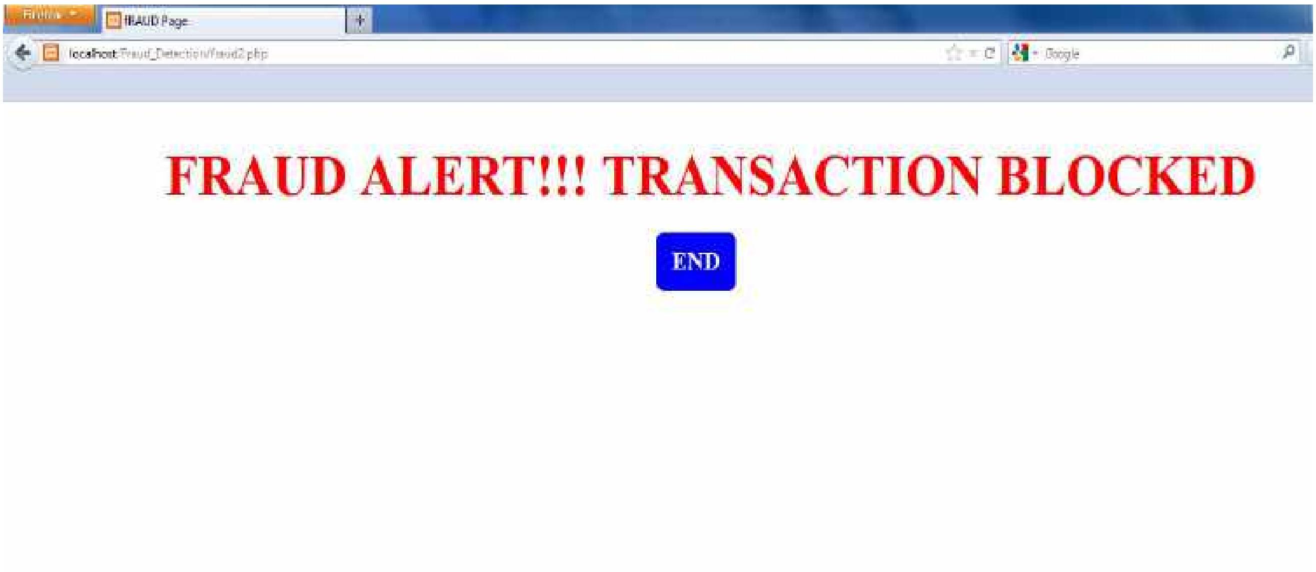
Fig 4.14: Smartcard User Pin Reset(1)



WRONG ANSWER!!

How Many Birthmarks do You Have?? :

Fig 4.15: Smartcard User Pin Reset(2)



FRAUD ALERT!!! TRANSACTION BLOCKED

Fig 4.16: Smartcard Fraud Detection Page

CORRECT!!! Click OK to RESET PIN

How Many Birthmarks do You Have??:

Fig 4.17: Smartcard User Pin Reset(3)

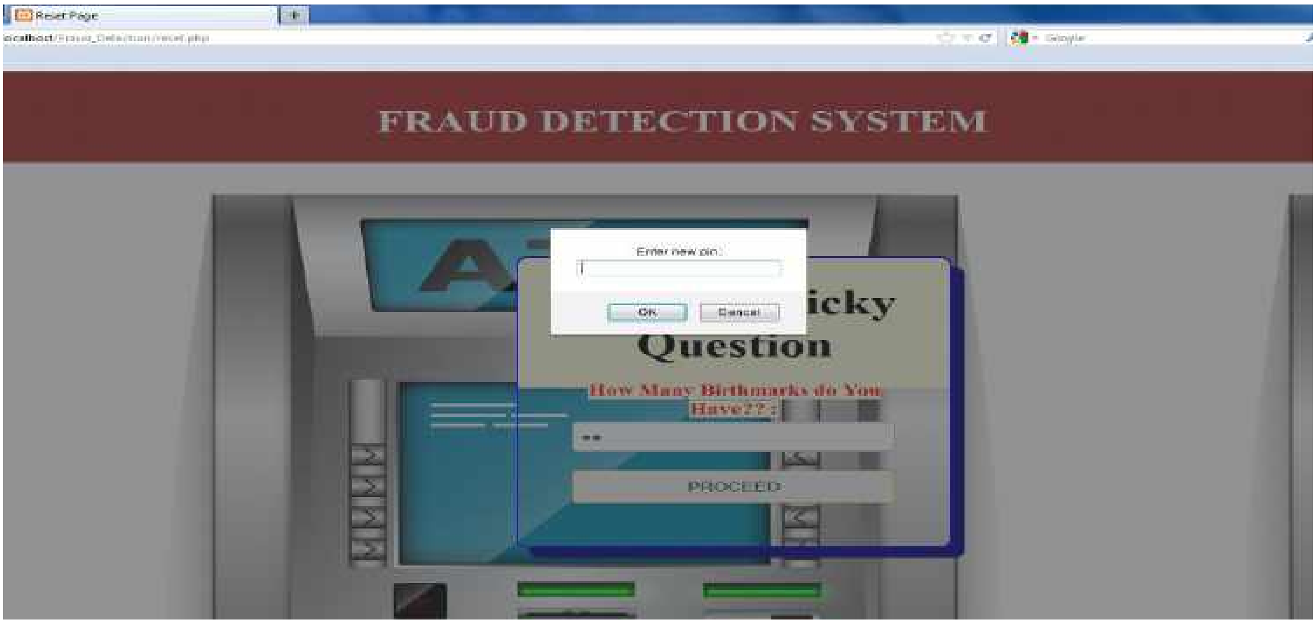


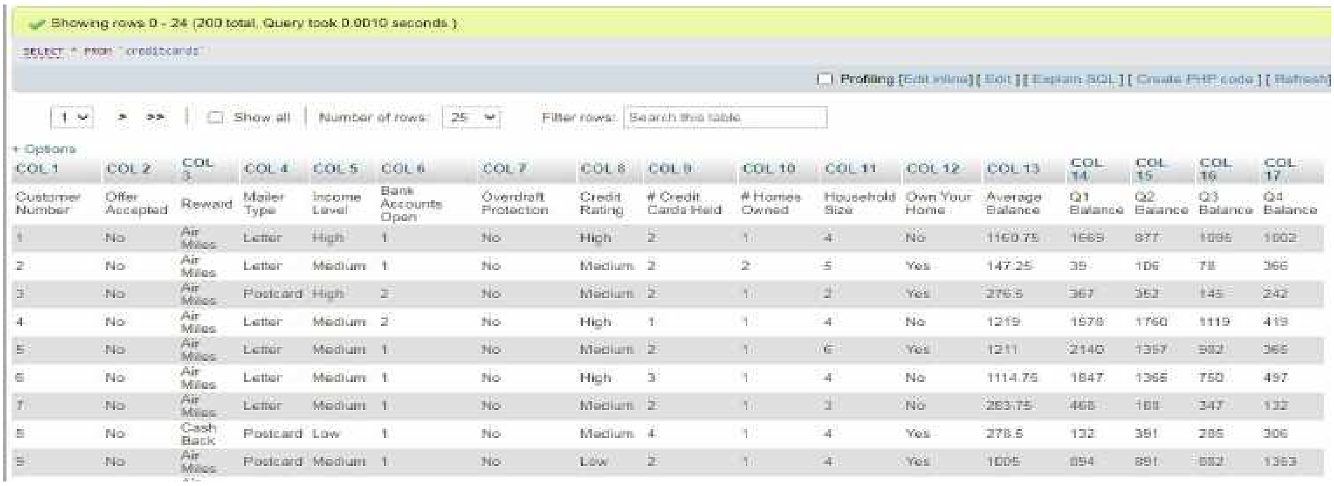
Fig 4.18: Smartcard User Pin Reset(4)

*Fig 4.19:*Select Amount Page

When compared to the existing system performance using certain parameters for evaluation, the proposed system proved more efficient an accuracy score of 94% as compared to the existing which has a score of 65%.

TRANSACTION SUCCESSFULL!!!

*Fig 4.20:*Successful Transation Page

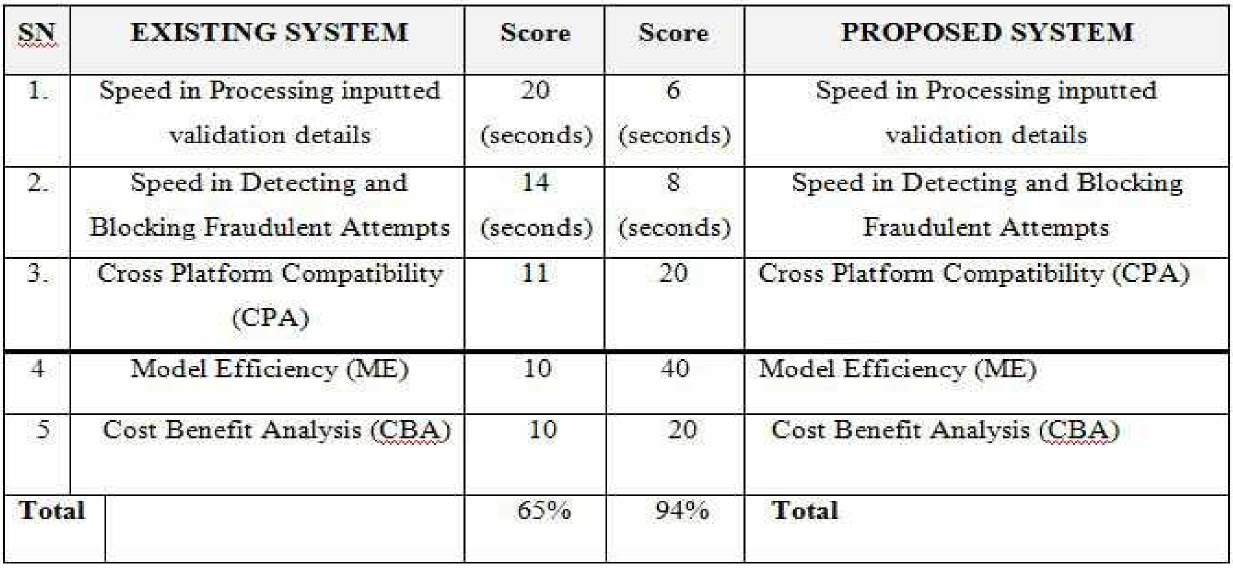


✓ Showing rows 0 - 24 (200 total, Query took 0.0010 seconds.)

```
SELECT * FROM `creditcards`
```

COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7	COL 8	COL 9	COL 10	COL 11	COL 12	COL 13	COL 14	COL 15	COL 16	COL 17
Customer Number	Offer Accepted	Reward	Mailer Type	Income Level	Bank Accounts Open	Overdraft Protection	Credit Rating	# Credit Cards Held	# Homes Owned	Household Size	Own Your Home	Average Balance	Q1 Balance	Q2 Balance	Q3 Balance	Q4 Balance
1	No	Air Miles	Letter	High	1	No	High	2	1	4	No	1160.75	1669	877	1095	1002
2	No	Air Miles	Letter	Medium	1	No	Medium	2	2	5	Yes	147.25	39	106	78	366
3	No	Air Miles	Postcard	High	2	No	Medium	2	1	2	Yes	276.5	367	352	145	242
4	No	Air Miles	Letter	Medium	2	No	High	1	1	4	No	1219	1578	1760	1119	419
5	No	Air Miles	Letter	Medium	1	No	Medium	2	1	6	Yes	1211	2140	1357	982	365
6	No	Air Miles	Letter	Medium	1	No	High	3	1	4	No	1114.75	1847	1365	750	497
7	No	Air Miles	Letter	Medium	1	No	Medium	2	1	3	No	283.75	468	188	347	132
8	No	Cash Back	Postcard	Low	1	No	Medium	4	1	4	Yes	278.5	132	391	285	306
9	No	Air Miles	Postcard	Medium	1	No	Low	2	1	4	Yes	1005	894	891	882	1363

Fig 4.21: Smartcard User Table



SN	EXISTING SYSTEM	Score	Score	PROPOSED SYSTEM
1.	Speed in Processing inputted validation details	20 (seconds)	6 (seconds)	Speed in Processing inputted validation details
2.	Speed in Detecting and Blocking Fraudulent Attempts	14 (seconds)	8 (seconds)	Speed in Detecting and Blocking Fraudulent Attempts
3.	Cross Platform Compatibility (CPA)	11	20	Cross Platform Compatibility (CPA)
4	Model Efficiency (ME)	10	40	Model Efficiency (ME)
5	Cost Benefit Analysis (CBA)	10	20	Cost Benefit Analysis (CBA)
Total		65%	94%	**Total**

*Fig 4.22:*Comparitive Analysis

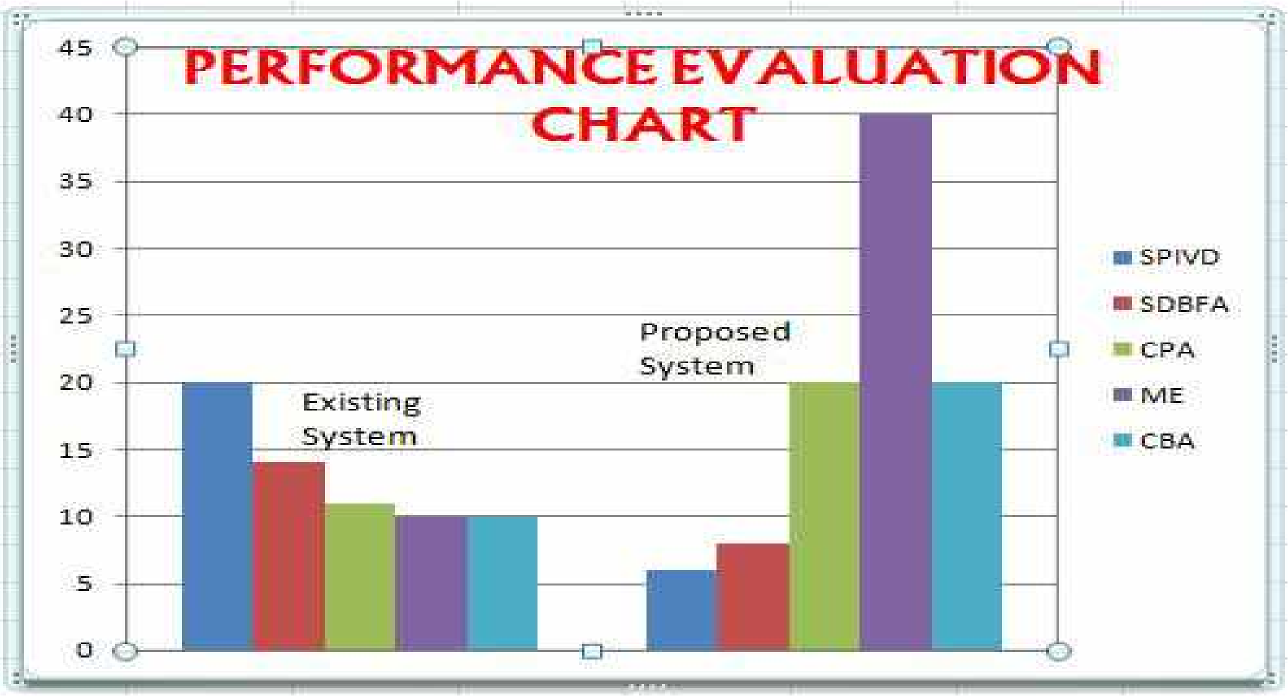


Fig 4.23: Performance Evolution Chart

V. CONCLUSION AND FUTURE WORK

This work described the development of an application for smartcard and telecommunications fraud detection. The development model involved a high security verification system to ensure that only the verified users are allowed to carry out transactions on their accounts. The security mechanisms used in classifying and detecting transactions based on the training data have been described and justified. Small parts of the developed code were presented as well as screenshots of the application.

We intend to expand the scope of the study to the development of a system that will detect other types of telecom frauds apart from the subscription fraud. Also, we intend to use a larger dataset for training the future system so that it will produce more efficient detection results. Also, we will like to introduce a hybrid model such as the neural network and the decision tree models in designing a fraud detection system.

REFERENCES

1. W.A. Oumar, and P.D. Augustin. “Credit card Detection using Artificial Neural Network,” International Journal of Innovation Technology and Exploring Engineering (IJITEE), vol.8, issue 7, pp.313-316. May 2019.
2. Z.W.W. Malek, K. Mayes, and K. Markantonakis. “Fraud Detection and Prevention in Smartcard Based Environments using Artificial Intelligence,” IFIP International Federation for Information Processing, 2008, pp. 118-132.
3. L.G. Kabari, N.D. Nanwin, and U.E. Nquoh. “Telecommunication Subscription Fraud detection using Artificial Neural Network,” Transaction on Machine Learning and Artificial Intelligence, vol.3, issue 6, pp.19-33. Dec 2015.
4. P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le. “Real Time Data-Driven Approaches for Credit Card Fraud Detection,” Proceedings of the 2018 International Conference on E-Business and Applications, 2018, pp. 6–9. Available: https://doi.org/10.1145/3194188.3194196
5. N. Carneiro, G. Figueira, and, M. Costa. “A Data Mining Based System for Credit-Card Fraud Detection in E-tail,”. Decision Support Systems, [Online] vol. 95, pp. 91–101. Available: *https://doi.org/10.1016/j.dss.2017.01.002. 2017*
6. Y. Jain, N. Tiwari, S. Dubey, and S. Jain. “A Comparative Analysis of Various Credit Card Fraud detection techniques,” International Journal of Recent Technology and Engineering (IJRTE), vol.7, issue 5S2, pp.402-407. Jan 2019.
7. R. Patidar, and L. Sharma. “Credit Card Fraud Detection using Neural Network,”. International Journal of Soft Computing and Engineering (IJSCE), vol.1, pp. 32–38. 2011.
8. J. Pieprzyk, H. Ghodosi, and E. Dawson.”Using Neural network for Credit Card Fraud Detection,” Information Security and Privacy, Proceedings of 12th Australasian Conference, ACISP 2007 vol. 4586. 2007.
9. A. Oodan, K. Savolaine, C. Daneshmand and P. Hoath. Telecommunications Quality of Service Management: From Legacy to Emerging Services. Let. 4(2), pp. 50-55. JUN 2003
10. M.I. Akhter, and M.G. Ahamad. “Detecting Telecommunication Fraud using Neural Networks Through Data Mining,” International Journal of Science and Engineering (IJSER), vol. 3 issue 3, pp. 1-5. MAR 2012.
11. M.J. Johnson and T.M. Khoshgoftaar. “Medicare Fraud Detection using Neural Networks,” Journal of Big Data, [Online] vol.6, issue 63, pp. 1-35. Available: *https://doi.org/10.1186/s40537-019-0225-0.*
12. S.S. Singh. “Electronic Credit Card Fraud Detection System by Collaboration of Machine Learning Models,” International Journal of Innovative Technology and Exploring Engineering (IJITEE), [Online] vol. 8, issue 12S, pp. 92-94. OCT 2019. Available: DOI:10.35940/ijitee.L1028.10812S19.
13. B.C. Amanze, and C.G. Onukwugha. “Credit Card Fraud Detection System in Nigeria Banks

using Adaptive Data Mining and Intelligent Agents: A Review," International Journal of Scientific Technology Research, vol. 7, issue 7, pp. 175-184. JUL 2018

14. B.C. Amanze, D.C. Asogwa, and C.I. Chukwuneke. "Credit Card Fraud Detection System using Intelligent Agents and Enhanced Security Features," International Journal of Trend in Research and Development (IJTRD), vol. 5, issue 3, pp. 524-530. MAY 2018.
15. S. Daliri. "Using Harmony Search Algorithm in Neural Network to Improve Fraud Detection in the Banking System," Computational Intelligence and Neuroscience, [Online] vol. 2020, pp. 1-5. Available: *https://doi.org/10.1155/2020/6503459.*
16. L. Delamire, H. Abdou, and J. Pointon. "Credit Card Fraud Detection Techniques: A Review," Banks and Banks System¸ vol. 4, issue 2, pp. 57-68. JAN 2009.
17. M. Zanin, M. Romance, S. Moral, and R. Criado. "Credit Card Fraud Detection through Parenclitic Network Analysis," Hihawi Complexity, [Online] pp. 1-9. Available: *https://doi.org/10.1155/2018/5764370. MAY 2018*
18. Z. Zhang, X. Zhou, X. Zhang, L. Wang, and P. Wang. "A Model Based on Convolutionary Neural Network for Online Transaction Fraud Detection," Hindawi Security and Communication Networks, [Online] pp. 1-9. Available: *https://doi.org/10.1155/2018/5680264.*
19. I. Sadgali, N. Sael. and F. Benabbou. "Fraud Detection in Credit Card Transaction using NeuralNetworks,' SAC2019, Casablanca, Morocco, 2019, pp. 1-4. Available: *https://doi.org/10.1145/3368756.3369082.*
20. Y. Abakarim, M. Lahhby, and A. Attioui. "An Efficient Real Time Model for Credit Card Fraud detection Based on Deep Learning," 12th International Conference on nIntelligent Systems: Theories and Application (STA'18), Rabat, Morocco, 2018, pp. 1-7, Available: *https://doi.org/10.1145/3289402.3289530.*

This page is intentionally left blank

Scan to know paper details and
author's profile

An Enhanced Rule-Based Expert System for Typhoid Fever Diagnosis using Fuzzy Logic

Onuodu, Friday Eleonu & Douglas, Minafa-a Thompson

University of Port Harcourt

ABSTRACT

This study deals with the design and development of a rule based expert system for diagnosis of typhoid fever, a common health disorder among Africans. This system interacts with users with plain English language based on some arranged rules. These rules, which are a typical collection of if/then rules, are extracted from experts in the medical fields in Nigeria. Using these rules, a knowledge base was designed for the expert system. Some programming codes were also written in PHP programming language for making deduction of new facts from rules in the knowledge base. The obtained results of the study showed that the proposed system outperforms the existing system in diagnostic speed and symptom processing. The evaluated parameters of the proposed system obtained values of 54 seconds and 45 seconds respectively for diagnostic speed and symptom processing when compared to the existing system which had values of 67 seconds and 84 seconds respectively. It is believed that this design can help to reduce the congestion we often see in our hospitals by providing solution for sick patients, irrespective of their locations.

IndexTerms: diagnosis, fuzzy-logic, improved, symptoms, typhoid fever.

Classification: I.5.1

Language: English

London
Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

An Enhanced Rule-Based Expert System for Typhoid Fever Diagnosis using Fuzzy Logic

Onuodu,Friday Eleonu [α] & Douglas, Minafa-a Thompson[σ]

ABSTRACT

This study deals with the design and development of a rule based expert system for diagnosis of typhoid fever, a common health disorder among Africans. This system interacts with users with plain English language based on some arranged rules. These rules, which are a typical collection of if/then rules, are extracted from experts in the medical fields in Nigeria. Using these rules, a knowledge base was designed for the expert system. Some programming codes were also written in PHP programming language for making deduction of new facts from rules in the knowledge base. The obtained results of the study showed that the proposed system outperforms the existing system in diagnostic speed and symptom processing. The evaluated parameters of the proposed system obtained values of 54 seconds and 45 seconds respectively for diagnostic speed and symptom processing when compared to the existing system which had values of 67 seconds and 84 seconds respectively. It is believed that this design can help to reduce the congestion we often see in our hospitals by providing solution for sick patients, irrespective of their locations.

IndexTerms: diagnosis, fuzzy-logic, improved, symptoms, typhoid fever.

Author α: Department of Computer Science, University of Port Harcourt, Rivers, Nigeria.

σ: Department of Computer Science, Ignatius Ajuru University of Education, Nigeria.

I. BACKGROUND

An expert system is a computer program designed to solve problems in a domain in which there is human expertise. The knowledge built into the system is usually obtained from experts in the field. Based on this knowledge, an expert system can replicate the thinking process of the human experts and make logical deductions accordingly. This paper focuses on the design and development of a rule based expert system for the diagnosis of typhoid fever. The two most common form of fever in Nigeria are malaria and typhoid. According to Malaria site (internet), malaria causes significant morbidity and mortality worldwide. In developing nations, scarce resources lead to inadequate diagnostic procedures. Malaria can result in anaemia (a decreased number of red blood cells). The remains of the destroyed red blood cells clump together and cause blockages in the blood vessels. This can result in brain damage or kidney damage, which is potentially fatal. A particularly serious, potentially life threatening, form of malaria parasite is called Plasmodium falciparum. Similarly, a bacterium called Salmonella typhi (S. typhi) is responsible for typhoid. S. typhi may be spread by consuming contaminated water, beverages and food, after which the bacteria enter the intestines and then the bloodstream, where they may spread to other body parts. Initial typhoid symptoms include malaise, headache, diarrhoea (or constipation), sore throat, fever as high as 104°F, as well as a rash. Diagnosis is carried out on the blood, bone marrow or stool cultures and with the Widal test.

In epidemics and less wealthy countries, after excluding malaria, dysentery or pneumonia, a therapeutic trial time with chloramphenicol is generally undertaken while awaiting the results of Widal test and cultures of the blood and stool. Apart from malaria and typhoid fever, the rule based expert system designed in this paper is developed to diagnose other types of fever such as pelebstein fever, leptospirosis fever, scarlet fever, dengue fever, rheumatic fever, hay fever and Lassa fever. There are many reasons for building an expert system to solve health related problems. Human experts may not always be available or may even be absent from a location. Also, by pooling knowledge of many experts, an expert system may be better than one human expert in its overall performance. An expert system does not get tired and are expected to be more consistent. It can also be used for training and passing on the knowledge derived from the human experts [1]

1.1 Statement of the Problem

This study addresses the problem of inability to diagnose and interpret typhoid fever. Typhoid fever is a disease with incomprehensive symptoms. From an in-depth feasibility study of related issues carried out, the mentioned problem is a system-based problem that requires improvement in order to accept different formats of datasets on typhoid fever and further diagnose the diseases in order to obtain the required treatment action. The Existing System is able to diagnose and interpret typhoid fever but in a delayed manner leading to preventive deaths In addition, the study will be divided into two levels: the first level will recognize precise typhoid fever and provide fuzzy symptoms; while the second level will take the fuzzy symptoms and finally obtain fuzzy treatment actions.

1.2 Aim and Objectives

The aim of this study is to develop an enhanced fuzzy-based model for diagnosing typhoid fever .using fuzzy logic The specific objectives of the study are to:

i. Design a Fuzzy-based Information System for Typhoid fever.
ii. Implement with PHP programming language and MySQL database as backend.
iii. Compare our results with the existing system for diagnosing typhoid fever.

1.3 Overview of Typhoid fever

Typhoid Fever Symptoms make it naturally difficult for a physician to establish the right diagnosis; it is referred to as typhoid fever in medical parlance. In order to diagnose typhoid fever properly, a diagnostic criterion for a particular disease is needed so as not to confuse it with other diseases because of shared symptoms. For a diagnosis to be effective in this regard, the target disease has to be recognized in a pool of typhoid fever and suggested two ways to handle this: by recognition of the combination of symptoms of the target disease or by exclusion of confusable disease as the cause of the symptoms .

1.4 Fuzzy Logic

Fuzzy logic is a branch of science that is extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Fuzzy logic may be applied to many fields, including control systems, neural networks and artificial intelligence (AI). Fuzzy logic can be used to describe how information is processed inside human brains. For example, it can be argued that humans do not know the difference between fat and thin. Five people may be fat and not have the same severity of fatness. Or, one person may appear thin, compared to another, while both are actually fat. Using fuzzy logic, you can assign different logic values for fatness, ranging from 0 to 1, according to severity of fatness. Variables between the extremes of zero and one are closer to the concept of probability, which means there is a major correlation between the science of probability and fuzzy logic. However, fuzzy logic refers to intensity of truth, while probability refers to likelihood.

Fuzzy logic is an extension of Boolean logic by Lotfi Zadeh in 1965 based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory.

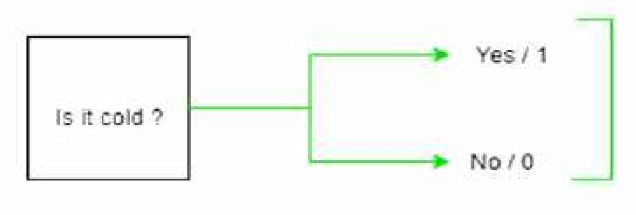


Fig. 2.1: Simple Illustration of Fuzzy Logic (Source: [1])

One advantage of fuzzy logic in order to formalize human reasoning is that the rules are set in natural language. Intuitively, it thus seems that the input variables like in this example are approximately appreciated by the brain, such as the degree of verification of a condition in fuzzy logic. Fuzzy logic can be conceptualized as a generalization of classical logic. Fuzzy logic can be used as an interpretation model for the properties of neural networks, as well as for giving a more precise description of their performance. We will show that fuzzy operators can be conceived as generalized output functions of computing units. Fuzzy logic can also be used to specify networks directly without having to apply a learning algorithm. An expert in a certain field can sometimes produce a simple set of control rules for a dynamical system with less effort than the work involved in training a neural network.

Fuzzy logic is now being used in many products of industrial and consumer electronics for which a good control system is sufficient and where the question of optimal control does not necessarily arises. The difference between crisp (i.e., classical) and fuzzy sets is established by introducing a membership function. Consider a finite set

$$X = \{x_1, x_2, \ldots, x_n\} \quad (1.1)$$

which will be considered the universal set in what follows. The subset A of X consisting of the single element x_1 can be described by then-dimensional membership vector

$$Z(A) = (1,0,0,\ldots,0), \quad (1.2)$$

where the convention has been adopted that a 1 at the ith position indicates that xi belongs to A. The set B composed of the elements x_1 and xn is described by the vector

$$Z(B) = (1,0,0, \ldots,1). \quad (1.3)$$

Any other crisp subset of X can be represented in the same way by an n-dimensional binary vector. But what happens if we lift the restriction to binary vectors? In that case we can define the fuzzy set C with the following vector description:

$$Z(C) = (0.5,0,0, \ldots,0) \quad (1.4)$$

In classical set theory such a set cannot be defined. An element belongs to a subset or it does not. In the theory of fuzzy sets we make a generalization and allow descriptions of this type. In our example the element x_1 belongs to the set C only to some extent. The degree of membership is expressed by areal number in the interval [0, 1], in this case 0.5. This interpretation of the degree of membership is similar to the meaning we assign to statements such as "person x_1 is an adult".

II. RELATED WORK

Matthias et al [2] looked at Expert System for Medical Diagnosis of Hypertension and Anaemia. The study implemented a model for diagnosing Hypertension and Anaemia using Bayesian Network technology which is efficient for modelling real-life problems and reasoning under certainties. The authors did a good job but failed to enhance the expert system with a hybrid of other expert systems and Artificial Neural Networks (ANN).

Soltan et al [3] researched on Diagnosis of some Diseases in Medicine via Computerized Expert System. The study designed an expert system that is aimed at providing patients with background for suitable diagnosis and treatment (especially Angina Pectoris and Myocarial Infection). The authors did a good job but failed to show an in-depth performance evaluation of their developed expert system especially in areas that involves time complexity, benchmarking and cost benefit analysis.

Zainab and Al Bakry [4] looked at medical diagnosis advisor system: a survey. The authors reviewed and classified knowledge-based techniques using four categories which included rule-based systems, case-based reasoning, knowledge-based systems and fuzzy logic. The authors did a good job but failed to implement the surveyed issue to a model for more clarification and understanding.

Ahmad et al [5] looked at an Online Expert System for Psychiatric Diagnosis. The authors developed a declarative online procedural rule-based expert model for psychological diseases diagnosis and classification. Furthermore, the developed model exploited computer as an intelligent and deductive tool to diagnose and treat four types of diseases which include depression, anxiety disorder, obsessive-compulsive disorder and hysteria. The authors did a good job but could not expand the knowledge-base capacity of their system to address more than the mentioned four (4) major mental health disorders. In addition, their model lacked an Improved Case-based Reasoning Technique (ICBR-RT) which uses Genetic Algorithm (GA) to recommend accurate diagnosis based on similar related-cases in quick response time.

Sanpreet [6] looked at a Fuzzy-rule based Expert System to Diagnostic the Mental Illness (MIDExS). The authors designed an Expert System to assist psychology doctors in diagnosing various mental illnesses related to humans. Furthermore, the developed expert system could be utilized to carry-out some evaluation of patient's physical and emotional symptom to diagnose the particular disorder. The author did a good job. However, performance evaluation of the developed system showed that it can only address four (4) major mental diseases.

Ayangbekun and Jimoh [7] looked at an Expert System for Diagnosis of Neurodegenerative Diseases. The scope of the work was extended to five brain disorders, namely Alzheimer's disease, Creutzfeldt Jakob disease, Huntington's disease, Multiple Sclerosis and Parkinson's disease. The computer programming language employed was the C#.NET programming language and Microsoft SQL Server 2012 served as the Relational Database Management System (RDBMS). The results obtained showed that the expert system was able to successfully diagnose these disorders corresponding to the selected symptoms entered as query. The authors did a good job. However, performance evaluation of their developed expert system showed that there was deficiency in time complexity and cost benefit analysis.

Ekong [8] looked at a Fuzzy Inference System for Predicting Depression Risk Levels. The study described research results in the development of a fuzzy driven system to determine the depression risk levels of patients. The system is implemented and simulated using MATLAB fuzzy tool box. The result of the system is consistent with an expert specialist's opinion on evaluating the performance of the system. The model lacked measures for long-term future upgrade.

Sumathi and Poorna [9] researched on Prediction of Mental Health Problems among children using Machine Learning techniques. The study identified eight machine learning techniques and has compared their performances on different measures of accuracy in diagnosing five basic mental health problems. A data set consisting of sixty cases is collected for training and testing the performance of the techniques. Twenty-five attributes have been identified as important for diagnosing the problem from the documents. The authors did a good job but could not implement the discussed issues to a model for more clarification and understanding.

Razzouk [10] researched on a Decision Support System for the Diagnosis of Schizophrenia Disorders. The work described the development of such a clinical decision support system for the diagnosis of schizophrenia spectrum disorders (SADDESQ). The development of the system was described in four stages: knowledge acquisition, knowledge organization, the development of a computer-assisted model, and the evaluation of the system's performance. The knowledge was extracted from an expert through open interviews. The interviews were aimed at exploring the expert's diagnostic decision-making process for the diagnosis of schizophrenia. The author did a good job. However, performance evaluation of their developed expert system showed that there was deficiency in time complexity and cost benefit analysis.

III. MATERIALS AND METHODS

3.1 Methodology

The Methodology adopted for the Improved Fuzzy-based Model for diagnosing typhoid fever is Software Development Lifecycle Methodology (SDLC). The Software Development Lifecycle Methodology (SDLC) involves a standardized set of task carried out in order to improve an Existing Project. It is also likened to the framework that is required to structure, plan and control the process of developing a new project. Furthermore, the Software Development Lifecycle Methodology (SDLC) comprises of models such as the waterfall model, the prototyping model, the incremental model and the spiral model.

3.2 Analysis of the Existing System

Majority of the players in the Health Sector of Nigeria operates the Paper-based health information system (see figure 3.1). Despite the documented benefits health information, most clinical encounters are still recorded by hand in a paper record. This is not without reason. The traditional paper record is still used due to its familiarity to users, portability, ease of recording "soft" or "subjective" findings, and its browsability for non-complex patients. There is also a sense of ownership of paper records, due to their being only one copy, which increases the sense of their security (although it will be noted below that this may be a false sense of security).

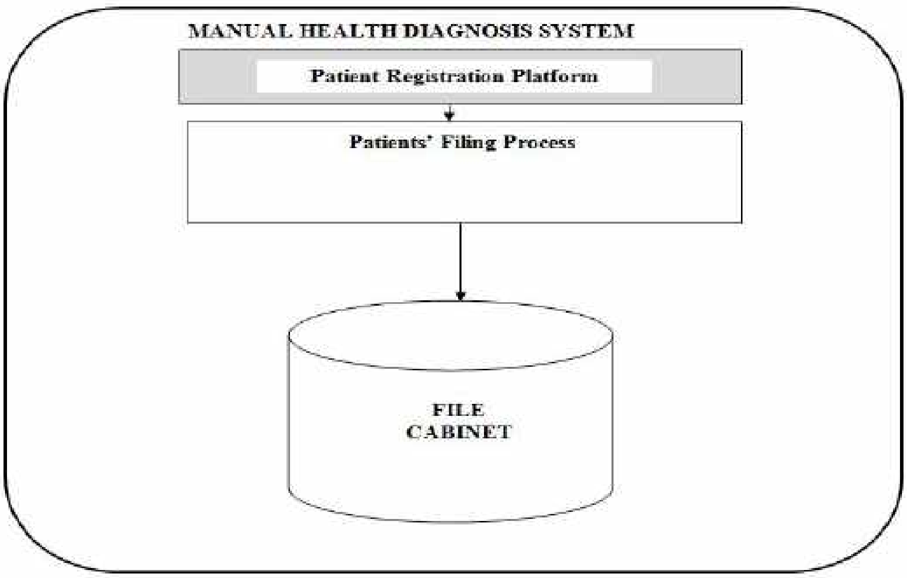
Nonetheless, there are many problems with paper-based medical records. The first is that the record can only be used in one place at one time. This is a problem for patients with complex medical problems, who interact with numerous specialists, nurses, physical therapists, etc. Another problem is that paper records can be much disorganized. Not only can they be fragmented across different physician offices and hospitals, as noted above, but the record at each location itself can often be disorganized, with little overall summary. In most paper records, pages are added to the record as they are generated chronologically, making the viewing of summarized data over time quite difficult. A final problem with the paper-based record is security and confidentiality. Although usually ascribed as a problem of the Paper-based system, there are attributes of the paper record that increase its vulnerability to access by non-privileged outsiders. Its difficulty in duplication leads to a great deal of photocopying and faxing among providers and institutions. In addition, the present system allows a provider's orders for diagnostic and treatment services (such as medications, laboratory, and other tests) to be entered electronically instead of being recorded on order sheets or prescription pads.

3.3 Explanation of the Existing System Components

The following components of the Existing System are:

i. *Patient Registration Platform:* This component illustrates a registration platform in which the patient inputs symptoms before the diagnosis process.

ii. *Patient Filing Process:* This component illustrates the organization of the inputted symptoms by the patient. In addition, it also illustrates office Filing Procedure which is a set of clearly defined and pan-organization followed practices in filing documents and important papers. Office Filing Procedures can include Employee Job Descriptions, Workplace Etiquette, as well as certain Confidentiality protocols.



(Source [6])

Fig.3.1: Existing System Architecture

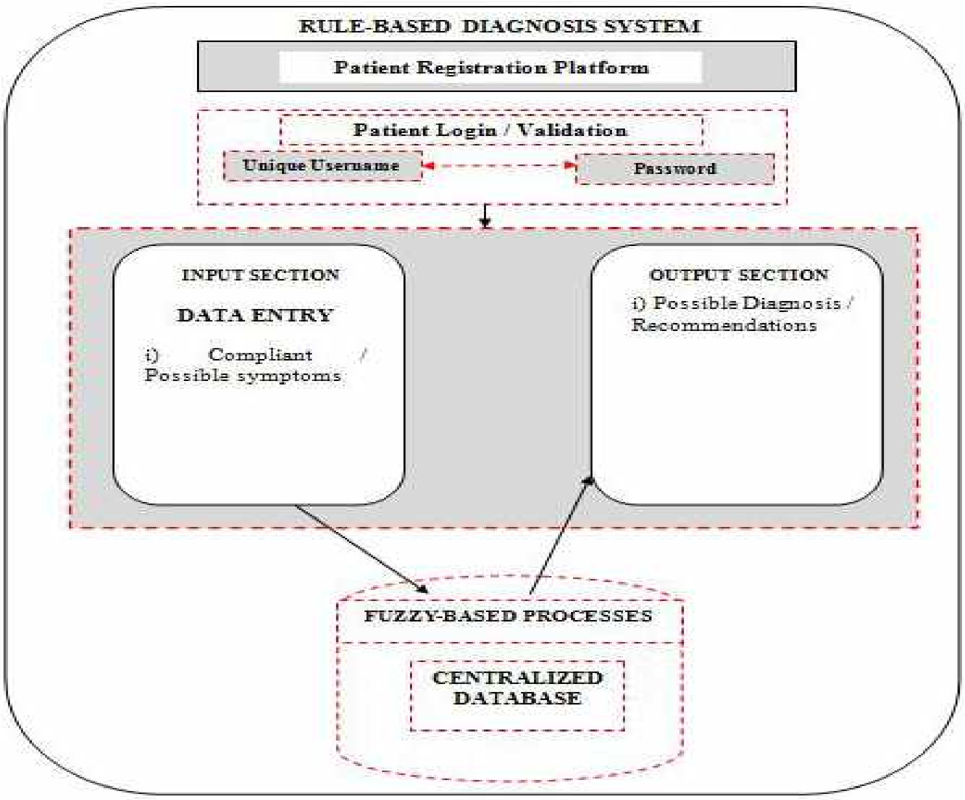


Fig.3.2: Proposed System Architecture

iii. *Filing Cabinet:* This component illustrates a piece of office furniture usually used to store paper documents in file folders. In the simplest context, it is an enclosure for drawers in which items are stored.

3.4 Disadvantages of the Existing System

The following disadvantages of the Existing System are:

i. *Editing and Communication Issue:* The paper-base system is cumbersome in that it does not allow users to easily edit file or send information to others. Paper files cannot be edited directly, forcing users to make new copies to update old files.

ii. *Order of Data Issue:* Data can get out of order in the paper-base system. If someone accidentally puts a file in the wrong place, or takes a file out of a cabinet and forgets to put it back, it can lead to data loss or the creation of additional copies of files.

3.5 Analysis of the Proposed System

The Proposed System of the study is an Improved Electronic Diagnosis System (IEDS) for Typhoid fever. The electronic medical record comprises a system of recording, processing, storing, recording and transferring health information electronically. Through the use of the IEDS, several limitations that are associated with the paper-based medical record system are clearly overcome. For example, in contrast to the paper record, the IEDS can play a larger role in medical decision-making, integrating the services of various departments, customizing care to the patients, reducing medical errors, improving quality, reducing costs, etc. In addition, the IEDS can effectively help to transfer patient information from one organization to another and in this way help in referrals and improving the access to healthcare. The use of IEDS in the healthcare system has demonstrated a number of advantages such that it can improve the quality of patient care, decrease healthcare costs, ensure adherence to government regulations and the standards of accrediting bodies, and improve the accessibility and the delivery of the healthcare.

3.6 Explanation of the Proposed System Components

The following component of the Proposed System is:

i. *Improved Fuzzy-based Diagnostic Model:* A control system based on fuzzy logic—a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively).

3.7 Advantages of the Proposed System

The following advantages of the Proposed System are:

i) Efficient User Graphical Interface for Patient-Physician Communication.
ii) Speed in acceptance and processing of diagnostic information from potential patients.
iii) The ability of the Improved Fuzzy-based System to update itself in the long-run.

3.8 Existing System Algorithm

```
Step 1:
START
Step 2:
DECLARE ALL VARIABLES
Step 3:
SR, UV, U, P, L, UR, QP, DSO. WHERE SR IS SERVER REQUEST, UV IS
USER VALIDATION, U IS USERNAME, P IS PASSWORD, L IS LOGIN, UR
USER REGISTRATION, QP IS QUERY PROCESS, DSO IS DIAGNOSIS SYSTEM OUTPUT
Step 4:
INITIATE SR
 Step 5:
 INITIATE UR
 Step 6:
 PROCESS UR
 Step 7:
 INITIATE UV
 Step 8:
 UV = L * (U + P)
 Step 9:
 INITIATE QP
 Step 10:
 OBTAIN DSO
 Step 11:
```

```
STOP
```

3.9 Proposed System Algorithm

```
Step 1:
START
Step 2:
DECLARE ALL VARIABLES
Step 3:
SR, UV, U, P, L, UR, QP, DSO, IFBM. WHERE SR IS SERVER REQUEST, UV IS USER VALIDATION, U IS USERNAME, P IS PASSWORD, L IS LOGIN, UR USER REGISTRATION, QP IS QUERY PROCESS, DSO IS DIAGNOSIS SYSTEM OUTPUT AND IFBM IS IMPROVED FUZZY-BASED MODEL
Step 4:
INITIATE SR
Step 5:
INITIATE UR
Step 6:
PROCESS UR
Step 7:
INITIATE UV
Step 8:
UV = L * (U + P)
Step 9:
INITIATE QP
Step 10:
ACTIVATE IFBM
Step 11:
INTERFACE IFBM-DATABASE
Step 12:
SEARCH FOR SIMILAR CASES RELATED TO THE USER'S REQUEST
Step 13:
ALERT MATCH FOUND
Step 14:
UPDATE SYSTEM DATABASE
Step 15:
OBTAIN DSO
Step 16:
STOP
```

IV. RESULTS AND DISCUSSION

4.1 Choice and Justification of Programming Language used

We implemented the Proposed System design with PHP, JavaScript Programming Language, Hypertext Markup Language, Cascading Style Sheet and MySQL Relational Database Management System. JavaScript is a server-side scripting language that is used for making web pages interactive. It is supported by all major web browsers. This is a programming language that is used by web developers for the creation of contents that communicate with databases. Secondly, PHP can be used for the development of web-based applications, system function performance; HTML is an acronym for Hypertext Markup Language and is used for structuring web pages. It consists of tags and is also supported by all major web browsers. Cascading Style Sheet (CSS) is a web development content that is used for styling and beautifying web pages. MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo and many more. Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

4.2 Discussion of Results

A rule based expert system called e-Diagnosis was designed to run under WINDOWS environment with PHP. This programming language is preferred due to its object oriented nature and its interactive capabilities with the users. Data were collected from different medical experts in order to build the new system, and special attention was given to understudy how a doctor attends to and diagnose patients. This new system is designed to visualize and simulate how patient diagnosis can be done with the use of computer. The new system is very easy to use and user friendly. Patients and medical practitioner that are already familiar with WINDOWS and WINDOWS based PACKAGES

like MS EXCEL, MS WORD, PageMaker etc. would be able to use this system with less stress. Labels, messages, boxes, dialogue boxes make the system even easier to use. The system also prompts and alerts the user if important data are omitted, as well as to save before exiting the application. The package worked on the simple logic of entering data, verifying data, validating data, editing, updating and computing. The program when run or compiled will display the welcome form which introduces the user to the application. The following are the procedures necessary for the diagnosis of a sick patient:
LOGIN FORM: This serves as an introductory screen to the application where the user is expected to supply username and password for authentication.

REGISTRATION FORM: This allows user to enter information about the patient's bio-data with reference to patient identity.

DIAGNOSIS PAGE: This allows user to enter information about the patient's illness i.e. the symptoms, diagnose the symptoms, prescribed drugs and print the report.

REPORTS PAGE: This displays the results of the diagnosis and also prescribes the necessary drugs.

Fig. 4.1: Login Page

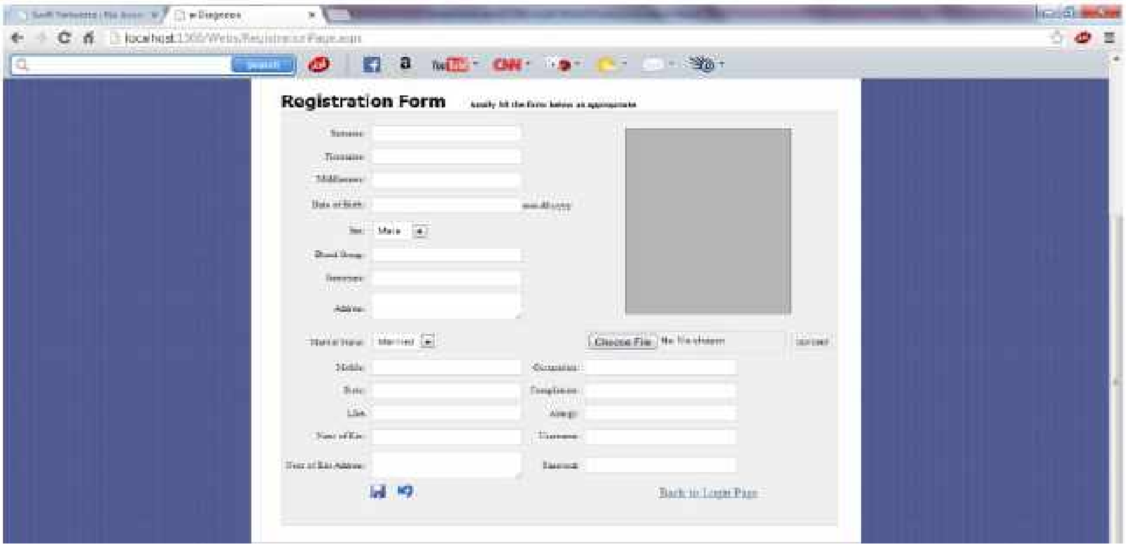
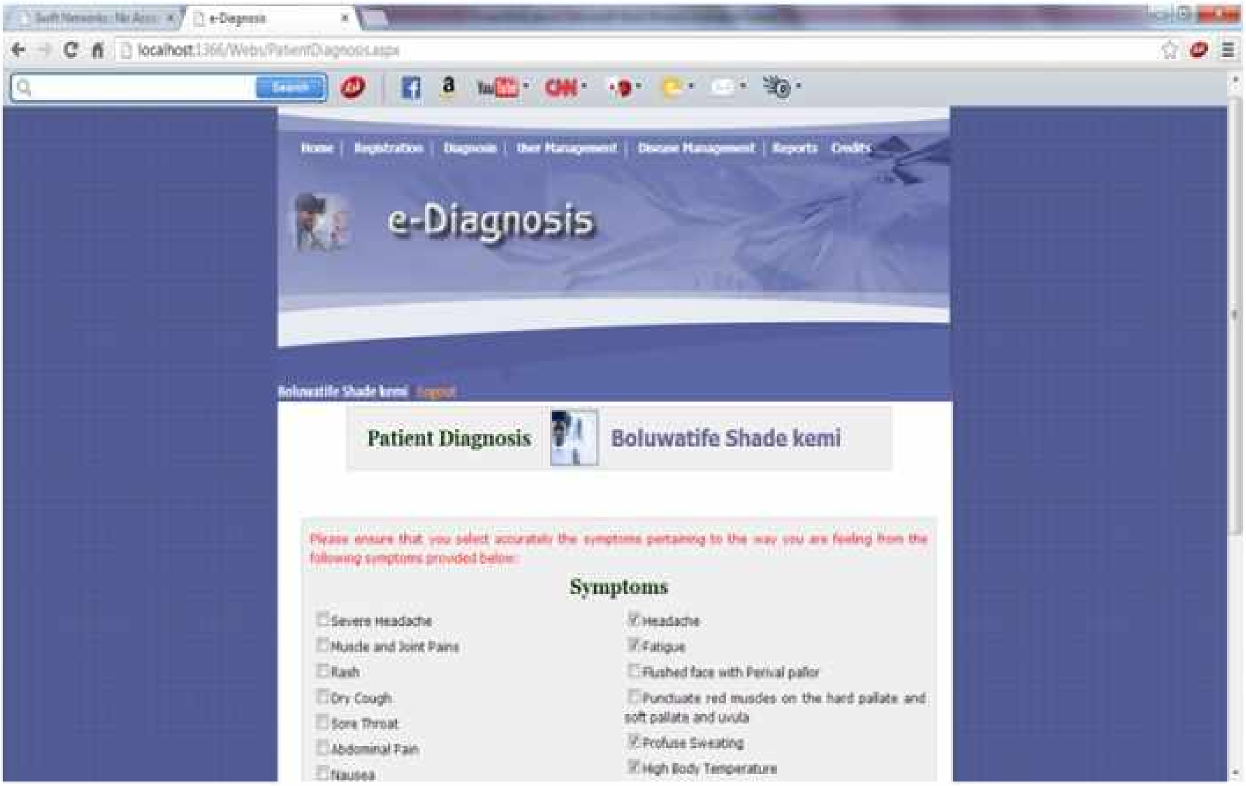


Fig. 4.2: Registration Page



Patient Diagnosis Boluwatife Shade kemi

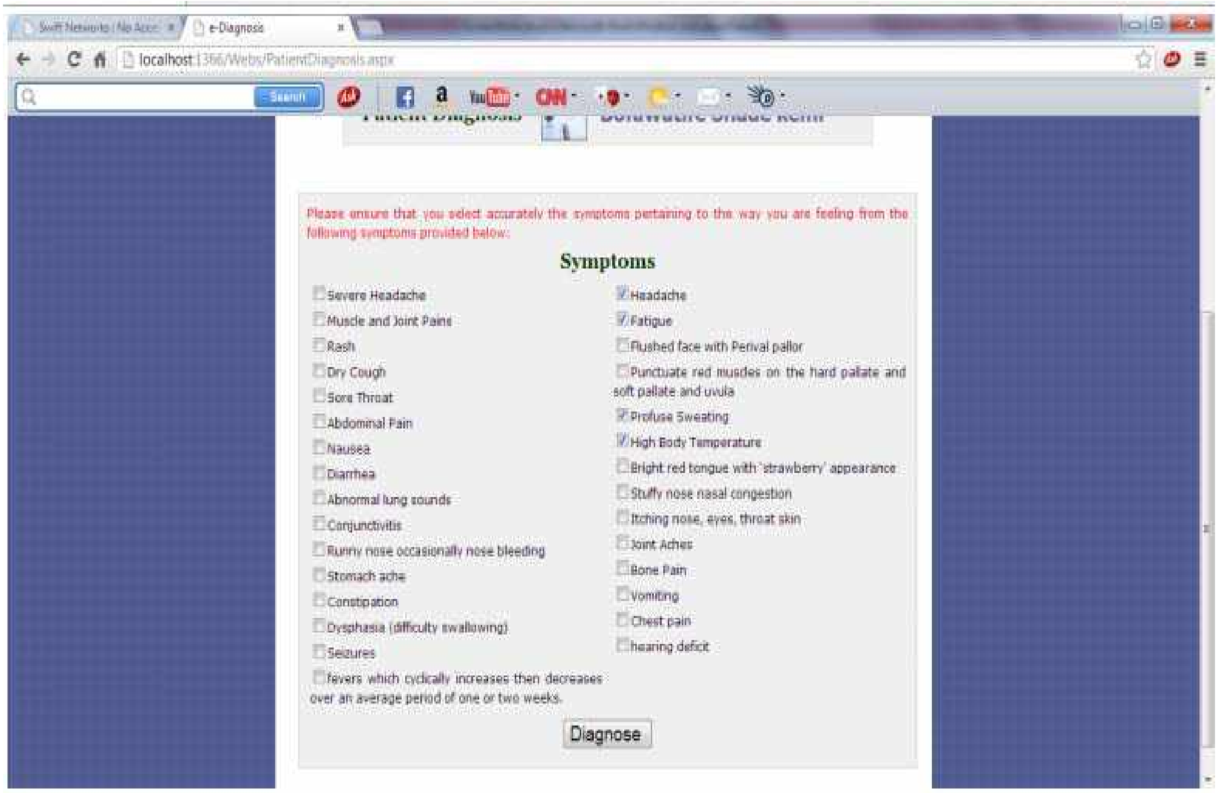
Please ensure that you select accurately the symptoms pertaining to the way you are feeling from the following symptoms provided below:

Symptoms

☐ Severe Headache
☐ Muscle and Joint Pains
☐ Rash
☐ Dry Cough
☐ Sore Throat
☐ Abdominal Pain
☐ Nausea

☑ Headache
☑ Fatigue
☐ Flushed face with Perival pallor
☐ Punctuate red muscles on the hard pallate and soft pallate and uvula
☑ Profuse Sweating
☑ High Body Temperature

Fig. 4.3: Diagnosis Page



Please ensure that you select accurately the symptoms pertaining to the way you are feeling from the following symptoms provided below:

Symptoms

☐ Severe Headache
☐ Muscle and Joint Pains
☐ Rash
☐ Dry Cough
☐ Sore Throat
☐ Abdominal Pain
☐ Nausea
☐ Diarrhea
☐ Abnomal lung sounds
☐ Conjunctivitis
☐ Runny nose occasionally nose bleeding
☐ Stomach ache
☐ Constipation
☐ Dysphasia (difficulty swallowing)
☐ Seizures
☐ fevers which cyclically increases then decreases over an average period of one or two weeks.

☑ Headache
☑ Fatigue
☐ Flushed face with Perival pallor
☐ Punctuate red muscles on the hard pallate and soft pallate and uvula
☑ Profuse Sweating
☑ High Body Temperature
☐ Bright red tongue with 'strawberry' appearance
☐ Stuffy nose nasal congestion
☐ Itching nose, eyes, throat skin
☐ Joint Aches
☐ Bone Pain
☐ Vomiting
☐ Chest pain
☐ hearing deficit

Fig. 4.4: Continuation of the Diagnosis Page

Patient Diagnosis

Boluwatife Shade kemi

Patient Diagnosis Result

Symptoms specified so far

Headache,Fatigue,Profuse Sweating,High Body Temperature,

Diagnosis Result

THYPOID FEVER FOUND,

Other Comments

Please you are advice to go for Medical Laboratory Test

Prescribed Drugs

Antibiotics, such as ampicilin , chloramphenicol trimethoprim sulfamthoxazole and ciprofloxacin.

Fig. 4.5: Continuation of the Diagnosis Page

4.2.1 Performance Evaluation

Table 4.1: Comparative Analysis

SN	EXISTING SYSTEM	Time in Seconds	Time in Seconds	PROPOSED SYSTEM
1.	Diagnostic Speed	54	67	Diagnostic Speed
2.	Symptoms Processing Speed	45	84	Symptoms Processing Speed

(**Source of data:** Program Implementation and Research Findings)

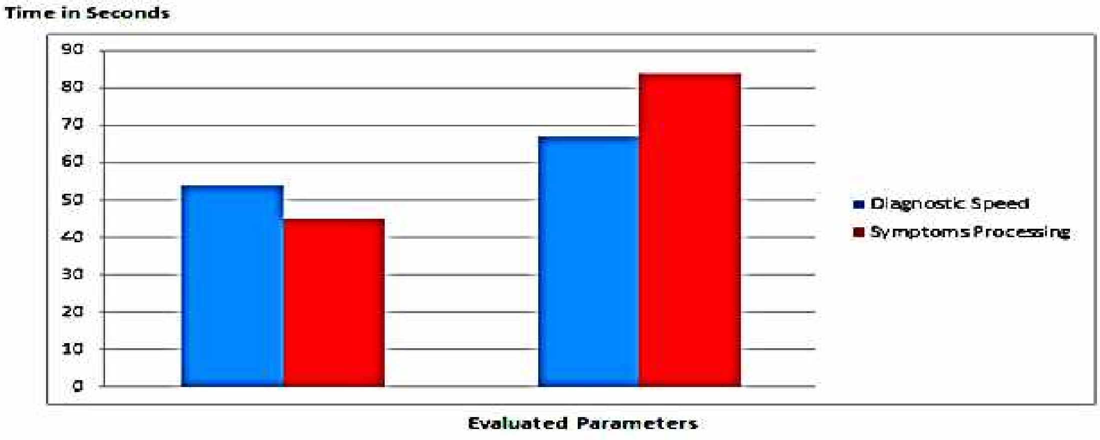


Fig. 4.6: Performance Evaluation Chart

V. CONCLUSION

This paper presents the design of a rule based expert system for the diagnosis of fever such as malaria fever, typhoid fever, pelebstein fever, leptospirosis fever, scarlet fever, dengue fever, rheumatic fever, hay fever and Lassa fever. The design, called e-Diagnosis, is a program developed in PHP programming language. This design shows the benefits of using a rule based expert system for diagnosing sick patients. The merits include reduction in congestion at the hospitals, cost reduction, reliability and accuracy of result. The designed system, e-Diagnosis, can also be updated if need arises.

REFERENCE

1. P. Asogbon, P. G. Aristo and F. Asharfa, Fuzzy Logic Fundamentals, International Journal of Engineering Technology (IJET), 4(8), 61 – 64, 2017.
2. D. Matthias and K. Obot, Expert System for Medical Diagnosis of Hypertension and Anaemia, MAYFEB Journal of Environmental Sciences, 3(2017), 12 – 19, 2017.
3. R.A Soltan, M.Z Rashad, B. El-Desouky, Diagnosis of some diseases in Medicine via Computerized Expert System, International Journal of Computer Science and Information Technology (IJCSIT), 5(5), 79 – 90, 2013.
4. T. Zainab and A. Al-Bakry, Medical Diagnosis Advisor System: A Survey, International Journal of Advanced Research in Computer Engineering and Technology, (IJARSET), 8 (1), 1 – 6, 2019.
5. A. Ahmad, A. Fatimah, A. Nouf, An Online Expert System for Psychiatric Diagnosis, International Journal of Artificial Intelligence and Applications, (IJAIA), 10(2), 59 – 76, 2019.
6. S. Sanpreet, A Fuzzy Rule-based Expert System to Diagnose Mental Illness (MIDExS), International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 3 (9), 8759 – 8764, 2019.
7. J. Anyangbekun and I. Jimoh, Expert System for Diagnosis of Neuro-generative diseases, International Journal of Computer Information Technology (IJCIT), 4(4), 694 – 698, 2015.
8. V. Ekong, A Fuzzy Inference System for predicting Depression risk levels, African Journal of Mathematics and Computer Science Research, 6(10), 197 – 204, 2014.
9. M. Sumathi and B. Poorna, Prediction of Mental Health Problems among Children Using Machine Learning Techniques, International Journal of Advanced Computer Science and Application, 7(1), 552 – 557, 2016.
10. D. Razzouk (2016), A Decision Support System for the Diagnosis of Schizophrenia Disorders, Brazilian Journal of Medical and Biological Research (2016), 39: 119 – 128.

This page is intentionally left blank

Scan to know paper details and
author's profile

Enhanced Mechanical and Morphological Properties of Nylon 6/SiO2 Nanocomposites

K. Shiva Kumar & A. Chennakesava Reddy

ABSTRACT

In this work, to enhance the mechanical and wear resistance of the Nylon-6/SiO2 nanoparticles, added to SiO2 at various volume fractions to Nylon-6 matrix material. The composite stirred continuously to maintain the homogeneity of the mixture in injection molding. Tensile test and hardness tests are conducted for specimens to study the influence of the filler content, load, sliding speed, and sliding distance the variation of mechanical properties. Scanning electron microscopic analysis is carried out for micro examination of the surfaces and to study the wear derbies. The result shows there is an influence of the filler material, and Nylon shows high bonding ability.

Keywords: nylon-6, nano SiO2, tensile, hardness, wear, SEM.

Classification: J.2

Language: English

London
Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

Enhanced Mechanical and Morphological Properties of Nylon 6/SiO2 Nanocomposites

K. Shiva Kumar[α] & A. Chennakesava Reddy[σ]

ABSTRACT

In this work, to enhance the mechanical and wear resistance of the Nylon-6/SiO2 nanoparticles, added to SiO2 at various volume fractions to Nylon-6 matrix material. The composite stirred continuously to maintain the homogeneity of the mixture in injection molding. Tensile test and hardness tests are conducted for specimens to study the influence of the filler content, load, sliding speed, and sliding distance the variation of mechanical properties. Scanning electron microscopic analysis is carried out for micro examination of the surfaces and to study the wear derbies. The result shows there is an influence of the filler material, and Nylon shows high bonding ability.

Keywords: nylon-6, nano SiO2, tensile, hardness, wear, SEM.

Author α: Research Scholar,Department of Mechanical Engineering JNTUH College of Engineering, Hyderabad – 500085, Telangana, India.

σ: Professor, Department of Mechanical Engineering JNTUH College of Engineering, Hyderabad – 500085, Telangana, India.

I. INTRODUCTION

PMMA composite with hydrophobic SiO2 nanoparticles enhances resistance to wear. The specimen shows hydrophilic behavior due to the PMMA matrix incorporation of the SiO2 nanoparticles[1]. Nylon 66 with CaCO3 filler materials increase the tensile strength of about 112 MPa.with the addition of CaCo3 to Nylon, and there is an enhancement in the tensile strength. RanjanMajhi et al. (2017[2]).Mechanical properties of the material varying with the pure metal. Material with filler enhances the performance of the content when GFRP (glass Fiber Reinforcement Plastic), Nylon with composites are investigated and found that the tensile strength is increasing with the concentration of the filler material Husain Mehdi et al. (2014)[3]. Nanocomposites of polypropylene (PP) and nano α-Al2O3 prepared, and the mechanical test carried out, Mechanical properties of the composites increased by α-Al2O3 particles and dispersant agents. During the Transmission and scanning electron microscopic observations, it observed that the surface is rougher with an increase in the filler content Mirjalili.F(2014)[4]. The hardness of the composite with the filler content is the first increase; then, it decreases with the filler content. Silica fume-filled Nylon composite shows superior properties compared to other filler materials Raja and Kumaravel(2015)[5]. To increase the wear resistance of the content and the pure material composites added with filler material. The filler material of carbon black added to PTFE composites to tribological behavior is studied and found that wear resistance of the material increased with the Carbon black filler material V.A. Shapovalov (2010)[6]. Glass fiber reinforced composite is filled with SiO2 nanoparticles in various proportions and examined the properties of the tensile strength and the impact strength. The results concluded that there is an impact of nanoparticles in the composites. The fractured surface morphology shows that the interfacial bonding increased with the filler content of the material Ramesh Chandra [7]. The addition of secondary filler to composites can influence the behavior of the material. The morphology of the material studied and found that there is a significant change in the properties. Stiffness and elongation of the material during a failure is high then material without filler content Francesco Silva et al. (2014)[8].Input process parameters such as the sliding distance load and sliding speed influence the output response of the composite. This input parameter can be optimized to increase the wear resistance and the tensile strength of the material. The amount of wear can be decreased with an increase in graphite content. An increase in the sliding content will increase the tensile strength of the composite[9]. Inorganic filler materials widely used, as the documents show better properties. The tribological behavior of the material with ABS composites increases the tensile strength and is much more useful than inorganic. SBF content in the ABS matrix shows matrix enhanced both the surface hardness and wear resistance. Elongation of the material decreased; as a result, it impacts the improvement of the tensile strength [10]. The flexural strength of composite with the addition of nanoparticles increased with an increase in the filler content. Maximum tensile strength observed at higher content of filler material. By the addition of okra, fiber-reinforced composites show an increase in the tensile strength[11]. A.C Reddy [12] concluded that mechanical properties and wear performance can be enhanced by adding filler materials to Nylon 6 composite.Nylon6 composites, with the addition of the Teflon, increases the tensile strength and hardness of the composite also composite exhibits reduction in the ductility [13].Nylon/Teflon composites with Nano Iron Oxide (γ-Fe_2O_3) particles mechanical properties found to be double to that actual tensile strength of the composite. Flow lines observed in Nylon 6/Teflon/Iron oxide [14]. The bulk density of the silica fume increased with an increase in the filler content. The hardness also found to be supplemented with an increase in filler content. Mechanical properties are improved and found to be high [15]. Vinyl Ester reinforced composite with a combination of SiO2 Particles are investigated, and fond found that youngs modules increases and tensile strength of the composite decreases.toughess of the material increases, which leads to an increase in the fracture energy.SEM analysis shows the mist zone, hackles, and step-like cracks observed.[16].The addition of 0.2%Wt Nano Silica Particles to the Thermoplastic PA6 improves the coefficient of the friction and wear resistance of the composite[17].

II. APPARATUS AND METHODOLOGY

Nylon-6 matrix material taken, and SiO2 Nanoparticles taken as the filler material. SiO2 partial size was approximately 80 nm. Process parameters also influence the mechanical properties, so the input process parameters such as filler content, load, sliding speed, and sliding distance are taken for carrying out the experiments Taguchi L9 orthogonal array is adopted, and test trails fixed. Filler content is ranging from 4% to 20% by weight. During the preparation of the composite SiO2 nanoparticles, the material is mixed in ME100LA mixer with Nylon at Temperature of 190°C for 20 min at 200 rpm of mixing blades. The mixture is heated to make composite soft and smooth. The mixer passed through the injection molding machine. Material forced into the mold cavity under the pressure of 70 Mpa to remove material shrinkage. The melt flow index of Nylon-6 was 12 g per 10 min. Input parameters vary, but other parameters such as injection pressure, the heating temperature of the charging barrel, and the cooling time of moldings were kept constant. Initially, Mould is at room temperature, Nylon-6 material would solidify below the glass transition temperature (105 °C).after some time specimens were ejected from the mold. The tensile test carried out for the fabricated composites materials.

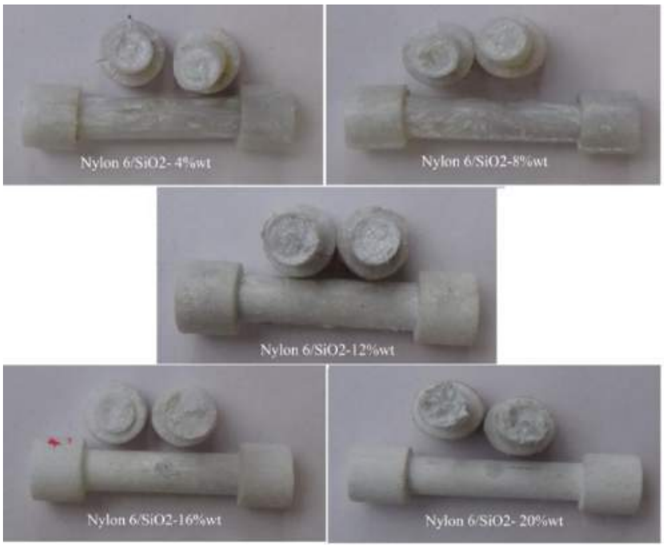


Figure 1: Tensile specimens of nylon-6/ SiO2

Tensometer Model PC-2000 (Fig. 2) used for the tensile test. After then test specimens were investigated using a scanning electron microscope, to study the wear behavior, wear test carried out using wear monitor (ASTM G99) with pin-on-disc type friction on a hardened ground steel (En32) disc emery paper (grade size of 400) fixed.

Figure 2: Tensometer

2.1 Design of experiments

Taguchi L9 design of trials for the factors of Normal Load (N), SiO2 (%wt.), Sliding speed (rpm), Sliding distance (m) given in Table 1. The Rockwell hardness test conducted for Nylon-6/ SiO2 and Scanning electron microscopy analysis carried out to study the composite material's fractography.

Table 1: Different levels of design factors

Factor	BN,wt%.	Normal Load, N	Sliding speed, rpm	Sliding distance, m
Symbol	A	B	C	D
Level–1	4	10	100	500
Level–2	12	15	200	750
Level–3	20	20	300	1000

Table 2: Orthogonal array (L9) and control parameters

Treat No.	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

III. TEST RESULTS AND DISCUSSIONS

Nylon-6/ SiO2 composites prepared with various compositions (0, 4, 8, 12, 16, and 20 wt %) of filler material and the tensile test done. During the tensile test, various load and displacement of the specimen recorded, as shown in Figure 3(a-f). It observed that there is an increase in the tensile strength later. There is a decrease in the tensile strength after composition reaches 8%. It is due to the interface bonding between the filler material, and the matrix is not suitable. This results in the optimum use of filler content are required to maximize tensile strength. The strain rate shows a sudden increase from 4%wt to 8%wt of SiO2 and a sudden decrease from 8% to 12%wt. From 12% to 16%, the strain rate is almost the same, and then it increases from 16% to 20%wt of SiO2.

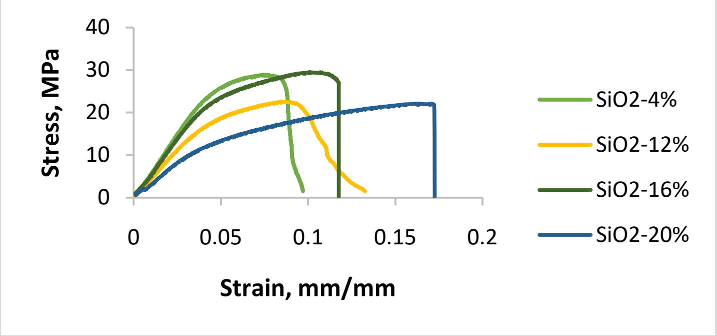


Fig 3: Stress-strain curves of NYLON-6/ SiO_2 polymer composites

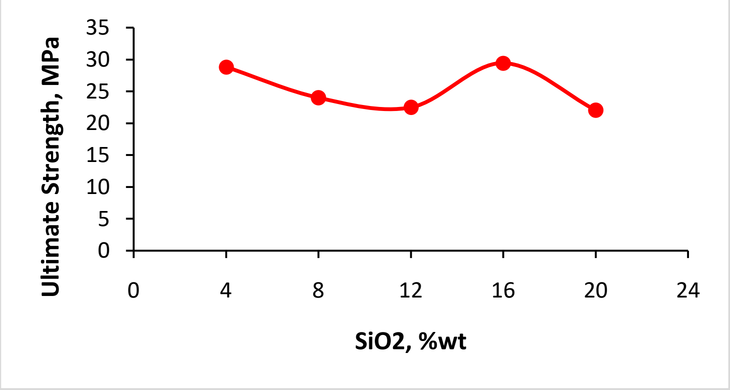


Fig 4: Ultimate strength (a) and corresponding strain (b) as a function of SiO_2.

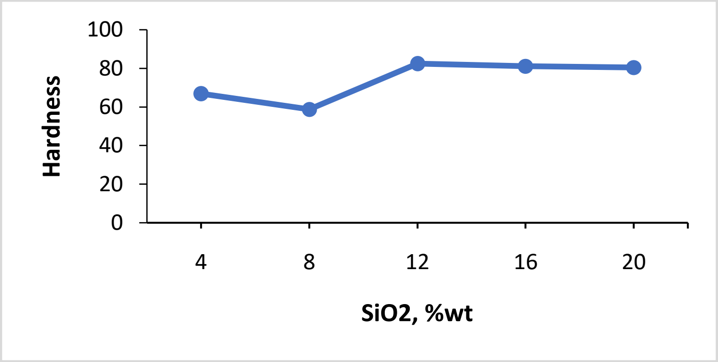


Fig 5: Hardness is a function of % SiO_2

3.1 Micro-Hardness Test Results

The hardness of the sample examined to graph plotted to vary the weight percentage of SiO2, as shown in Fig. 5. Hardness was measured using the Rockwell hardness test. By adding the filler content to the Nylon-6 composite, the hardness of the matrix material increases consistently. It observed that a change in %wt of SiO2 from 4% to 8% causes a slight decrease in hardness of the Nylon/ SiO2 polymer composite. Rockwell hardness increases from 8% to 12% and reaches the maximum value of 82.33 HRM at 12%wt of SiO2. The significant improvement in hardness may be attributed to the better distribution of SiO2 nanoparticles. There is no substantial change in the hardness of the composites from 12%wt to 20%wt of SiO2.

3.2 ANOVA

Parameter	Symbol	DOF	SS	MSS	P%
% SiO_2	A	2	680046	340023	24.30
Load	B	2	1107134	553567	39.56
Speed	C	2	350802	175401	12.53
Sliding Distance	D	2	660642	330321	23.61
Error	e	0	0	..	0
Total	T	8	2798624		100

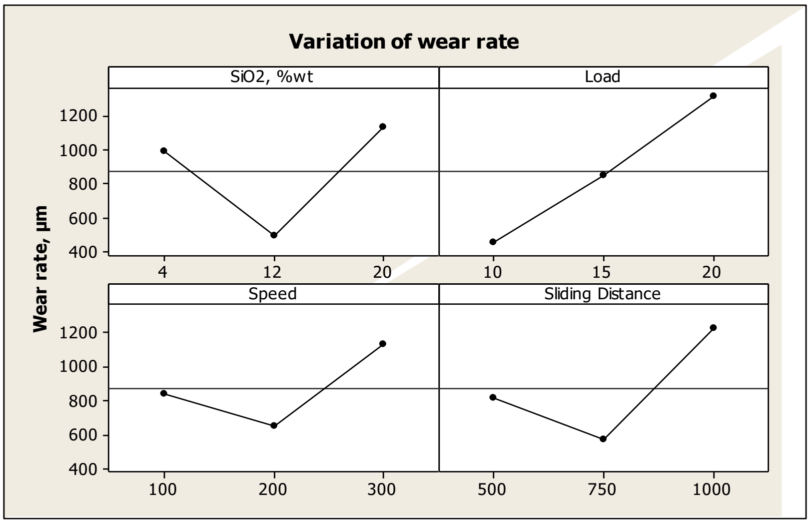


Figure 7: Variation of Wear Rate

Figure 7 shows that the Nylon/ SiO2 polymer composite with 20% wt of SiO2 has a higher wear rate. It observed that with an increase in loads in the range of 10N to 20N, the wear rate increases as the load increases, and the wear rate is maximum at 20N. For sliding speeds in the field of 100rpm to 300rpm, wear rate decreases from 100 rpm to 200 rpm and later increases from 200 rpm to 300 rpm reaching the maximum value at 300 rpm. For sliding distances in the range of 500 to 1000, the wear rate decreases from 500 to 750 and later increases from 750 to 1000.

3.3 Morphology of Fracture surfaces

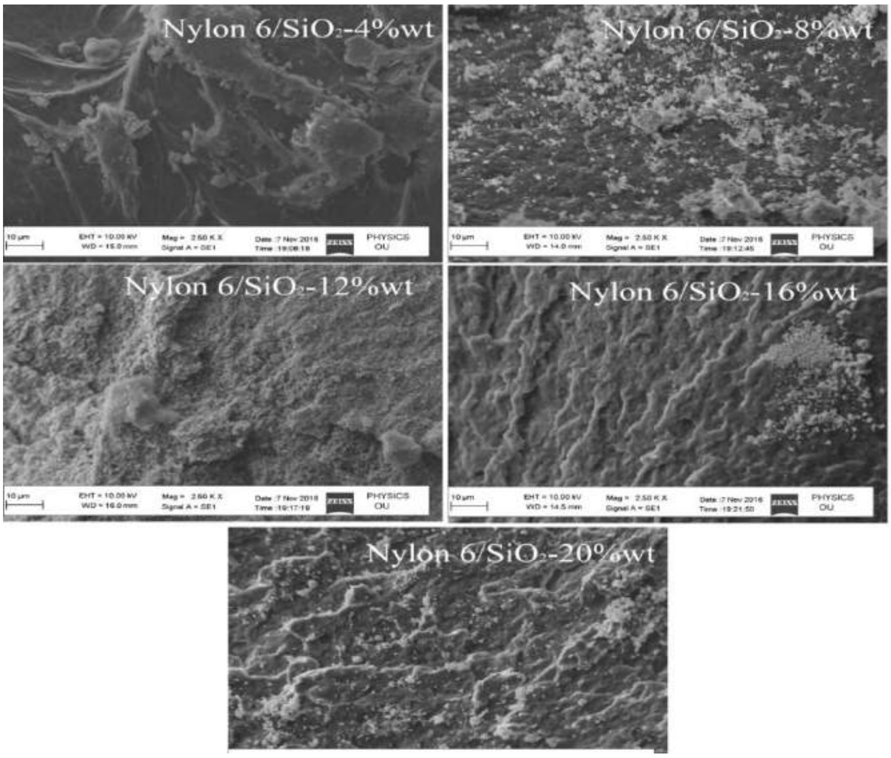


Figure 7: Fractography of Nylon-6/ SiO2polymer composites

Figure 7, Morphology of fracture surface of the material is studied using SEM analysis; it found that micro-cracks increase with an increase in the filler content of SiO2. The micro-cracks produced due to the interaction of the filler material with composite, which results in the strong inter bounding of the filler material with composite. As the bonding strength of materials is increasing, the tensile and wear resistance increases. The layers observed during crack with increasing filler content as the matrix material, and filler material creates a strong bond with each other.

3.4 SEM Analysis of Worn Surface

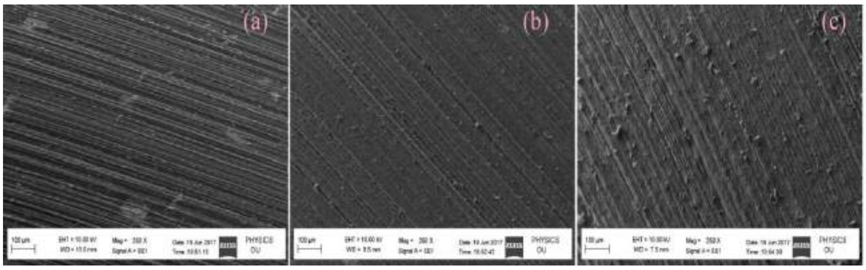


Figure 8: SEM image of worn surface for trial conditions of 1, 2, and 3.

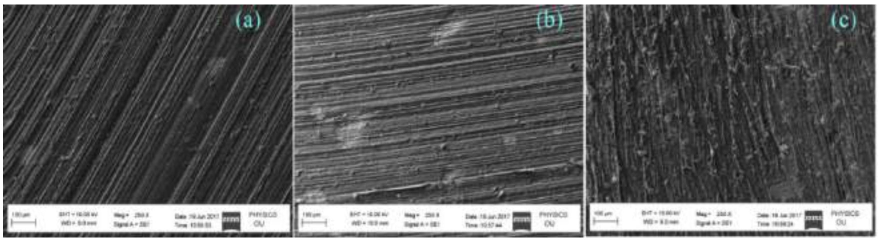


Figure 9: Worn surfaces of specimens for trial conditions of 4, 5, and 6.

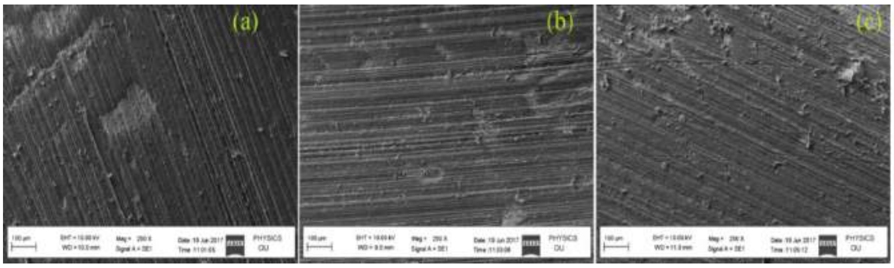


Figure 10: Worn surfaces of specimens for trial conditions of 7, 8, and 9.

Figure 8 to 10 SEM images shows that composite during the wear test. Micro-cracks increased with an increase in the filler content of the material. These micro-cracks are uniformly continuous throughout the surface. These groves are perpendicular to the sliding surface. The intensity of micro cracks and the wear test input parameters compared. It found that with an increase in the SiO2 percentage and load, there is an increase in the wear rate.

3.5. Wear Debris

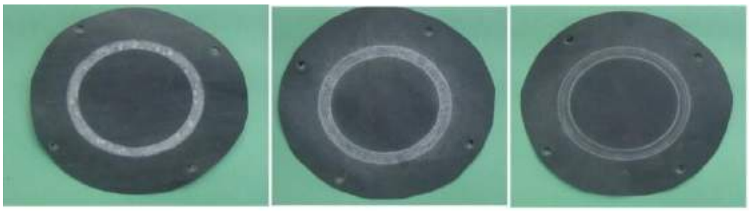


Fig. 9: Debris of specimens for trial conditions of 1, 2, and 3.

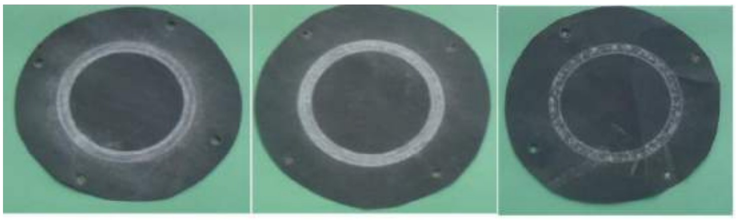


Fig. 10: Debris of specimens for trial conditions of 4, 5, and 6.

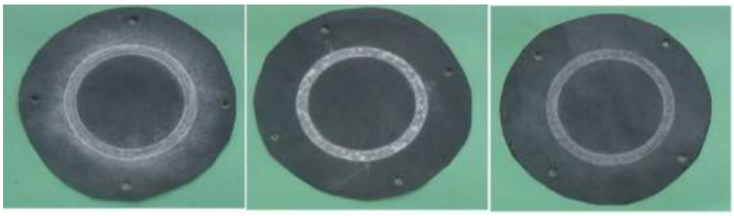


Fig. 11: Debris of specimens for trial conditions of 7, 8, and 9.

Figure 9-11, the wear debris is produced during the wear test were studied. It found that increase in SiO2 and load, size of flakes, and the number of chips also increased.

IV. CONCLUSIONS

In this study, the investigation carried out to observe the influence of SiO2 with Nylon composite material. The mechanical properties include the tensile test, wear test, the hardness of the content carried out, and the following conclusions.

- With the increase in SiO2 nanoparticle increase, the material's tensile strength up to a specific limit later, the material's tensile strength decreases with an increase in the SiO2 percentage.
- The hardness of the composite material increased with an increase in the filler content of the material.
- SiO2, load, and sliding distance are the primary process parameters. The percentage contribution of SiO2 is 24%, the load is 40%, and the sliding distance is 20 %.
- The composites SiO2 filled with nanoparticles showed lower wear rates at the combination of 12% wt SiO2, 10N load, 200 rpm speed, and 500 m sliding distance.
- Morphology of fracture materials is studied and found that roughness of the material increase as an increase in the load and filler content of the material.
- Wear debris of the material are investigated and found that the flakes with the increase in the SiO2.

REFERENCES

1. AI Martınez-Perez et al. (2019): Characterization and sliding wear performance of PMMA reinforced with SiO2 nanoparticles. Journal of Thermoplastic Composite, Materialspp1-15: DOI: 10.1177/ 0892705718815532.[1]
2. RanjanMajhi et al. (2017): Preparation and Study of Mechanical Properties of Nylon 66 / CaCO3 Engineering Thermoplastic Composite. International Journal of Latest Technology in Engineering, Management & Applied Science, 6(10),pp.31-34.[2]
3. Husain Mehdi et al. (2014): Experimental Analysis of Mechanical Properties of Composite Material Reinforced by Aluminium-Synthetic Fibers. International Journal of Mechanical Engineering, 4(2),pp.59-69.[3]
4. Mirjalili. F(2014) : Mechanical and Morphological Properties of Polypropylene/Nano α-Al2O3 Composites. The Scientific World Journal Volume 2014, Article ID 718765, 12 pages, as.[4].
5. V.L. Raja and A. Kumaravel(2015): Studies on Physical and Mechanical Properties of Silica Fume-Filled Nylon 66 Polymer Composites for Mechanical Components, Polymers & Polymer Composites, Vol. 23, No. 6, 2015,pp.427-434.[5]
6. V.N. Aderikha and V.A. Shapovalov(2010): Effect of filler surface properties on structure, mechanical, and tribological behavior of PTFE-carbon black composites. Wear, 268, pp. 1455–1464.[6]
7. Ramesh Chandra Yadav, Sachin Chaturvedi, Ashutosh Kumar, kamal Kumar Kannauji and Arpan Kumar Mondal, An Investigation of Mechanical and Sliding Wear Behavior of Glass Fiber Reinforced Polymer Composite With or Without Addition of Silica (SiO2).[7]
8. Francesco Silva et al. (2014): The Effect of Matrix and Reinforcement Material Selection on the Tensile Properties of Hybrid Composites.JSM Volume 1 Issue 1,pp.37-52.[8]
9. B. Stojanovic, M. Babi, S. Mitrovic, A. Vencl, N. Miloradovi, M. Panti "tribological characteristics of aluminum hybrid composites reinforced with silicon carbide and graphite. a review", Journal of the Balkan Tribological Association Vol. 19, No 1, 83–96 (2013). [25]
10. Mohammed Y. Abdellah, Hesham I. Fathi, Ayman M. M. Abdelhaleem, and Montasser Dewidar "Mechanical Properties and Wear Behavior of a Novel Composite of Acrylonitrile–Butadiene–Styrene Strengthened by Short Basalt Fiber," Journal

of composites science, 2, 34, pp2-12, 2018.[26]

11. Seelam Pichi Reddy, A. Chennakesava Reddy, "Tensile and Flexural Strength of Okra Fiber Reinforced Polymer Composites," International Journal of Engineering and Management Research, 6(1), pp. 491-495, 2016.[28]
12. A.C.Reddy "Characterization of Mechanical and Tribological Behavior of (Nylon6 Graphite+ Teflon) Nano Particulate Composite: Application Perspective", Int. Journal of Scientific & Engineering Research, 6(4), pp.378-386, 2015.[22]
13. A.C.Reddy "Characterization of Mechanical Behavior of Nylon/Teflon Nano Particulate Composites," International Journal of Advanced Research, 3(5), pp.1241-1246, 2015.[20]
14. Karan Agarwal, Nirmala Akhil, Regalla Srinivas, A.C Reddy "Enhancement in Mechanical Behavior of Nylon/Teflon Composites by Addition of Nano Iron Oxide (γ-Fe2O3)", International Journal of Science and Research (IJSR), 4(5), pp.927-932, 2015.[21].
15. V.L. Raja and A. Kumaravel "Studies on Physical and Mechanical Properties of Silica Fume-Filled Nylon 66 Polymer Composites for Mechanical Components", Polymers & Polymer Composites, Vol. 23, No. 6, 2015,pp.427-434.
16. D Dehghan Banani, S A J Jahromi and S MOJTABA ZEBARJAD∗ "A study on the role of nanosized SiO2 on deformation mechanism of vinyl ester" Bull. Mater. Sci., Vol. 37, No. 7, December 2014, pp. 1677–1683.
17. Monserrat Garcı, Matthijs de Rooij, Louis Winnubst, Werner E. van Zyl, Henk Verweij "Friction and Wear Studies on Nylon-6/SiO2 Nanocomposites" Journal of Applied Polymer Science, Vol. 92,2004, pp,1855–1862.
18. Yanhuai Ding, Ping Zhang, Yong Jiang, Fu Xu, Jiuren Yin, Yongde Zuo "Mechanical properties of nylon-6/SiO2 nanofibers prepared by electrospinning", Materials Letters, 63, pp. 34–36, 2009

This page is intentionally left blank

Scan to know paper details and
author's profile

Pairwise Spectral Correlation of SARS-Corona Viruses

Ethirajan Govindarajan

University of Petroleum

ABSTRACT

A set of complete genome data of nine SARS-Corona type viruses are analyzed for possible common features among them. A novel genetic features called '**Percentage Concentration of Nucleotides**' denoted as pA, pT, pG and pC are evaluated for each genome data and cross verified with other data whether all of them possess the same genetic features or not. It was found to be approximately the same for all genomes. This means, all SARS-Corona type viruses have the same structural genetic property, in the system biological sense. On the other hand, the genetic functional properties of SARS-Corona type viruses are different from one another. The scope of this paper is limited to study of spatial correlation among the genome data sets under consideration. Adjoints of a genome data are four independent binary sequences corresponding to the nucleotides of adenine, thymine, guanine and cytosine. For example, the adjoint of adenine of a genome sequence is a binary sequence consisting of 1's in the place of adenine in the genome sequence and 0's in all other places. Two-dimensional array of an adjoint of period 128 is obtained by dividing the binary sequence into segments of length 128 and placing all segments one below the other. One can try with periods of 64, 32, and so on and obtain respective two-dimensional arrays. This two-dimensional array is referred here as adjoint array. Adjoint arrays of all nine genome data are pair wise correlated and results reported in this paper.

Keywords: SARS-Corona Viruses, Artificial Intelligence, Machine Learning

Classification: J.3

Language: English

London
Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

Pairwise Spectral Correlation of SARS-Corona Viruses

Ethirajan Govindarajan

CAUTION

This is a theoretical investigation carried out on certain genome data of SARS-Corona type viruses archived in NCBI and results reported in this document are outcomes based on certain subjective definitions, notions and genome data analysis. Results presented here are to be deliberated and interpreted by Genetic Scientists, Agriculture Scientists, Biologists, Biotechnologists, Biochemists, Biophysicists and Microbiologists. This research report is meant for public use.

Chief Scientist and Team Leader

Ethirajan Govindarajan
Adjunct Professor, Department of Cybernetics, School of Computer Science
University of Petroleum and Energy Studies, Dehradun, Uttarakhand, India
Director, PRC Global Technologies Inc., Ontario, Canada
President, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
E-Mail: dr.rajaneg@gmail.com

Research Team

Dr. Manish Prateek, Dean, School of Computer Science, UPES, Dehradun, India
Dr. Amit Agarwal, Director, APJ Abdul Kalam, Institute of Technology, Dehradun, India
Dr. Michael Patrick Coyle, Chief Executive Officer, Avatar MedVision US LLC, NC, USA
Dr. Janarthanan Krishnamoorthy, Bio Technology Consultant, Singapore
Dr. K. Thiagarajan, Senior Research Officer, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
Parthasarathy Krishnan, CEO, Relay Pharma Corp., Sunnywale, California, USA

Project Team

Sathya Govindarajan, Director, PRC Global Tech., Inc., Ontario, Canada
Prashanthi Govindarajan, Director, PRC Global Tech., Inc., Ontario, Canada
Rahul Sharma, Senior Data Scientist, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
Smita Vempati, Research Officer, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
Gayatri Navle, Data Scientist Trainee, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
Pawan Kumar, Data Scientist Trainee, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
Yashaswi Vemuganti, Consultant, Avatar MedVision US LLC, NC, USA
S. Sree Hari Raju, Pallavi Engineering College, Hyderabad, Telangana, India

Advisory Committee Members

Dr. S. Nageswara Rao, Alchemist and Siddha Physician, Chennai, India
Dr. Andras Pellionisz, Chief Executive Officer, HolGenTech, California, USA
Dr. Jean Claude Perez, IBM European Research Center on Artificial Intelligence, France
Dr. Dharm Singh, Director General, Indian Medicinal Plants Marketing Federation (Retd.)
Dr. Ashok Alur, Director, Centre of Excellence for FPO, Govt. of Karnataka, India

Public Relations

H. B. Nayak, Vice President, Pentagram Research Centre Pvt. Ltd., Hyderabad, India
Dr. Hindupur Rajasimha, Chief Engineer, Indian Space Research Organization and IDBI (Retd.)

ABSTRACT

A set of complete genome data of nine SARS-Corona type viruses are analyzed for identifying possible common features among their spectra. A novel isomorphic transform called "**Discrete Rajan Transform**" introduced by **Ms. Prashanthi Govindarajan** in 2012 while she was in the Department of Electronics of Staffordshire University, United Kingdom, as an updated version of the homomorphic/isomorphic transform called "**Rajan Transform**" is proposed here to perform spectral analysis of the virus genome data. The purpose of spectral analysis is to extract finer and hidden details about the evolutionary nature of SARS-Corona viruses. It was found that all SARS-Corona type viruses evolve by random mutation but with their basic structural genetic property maintained in the system biological sense. This has already been verified by extracting common digital spatial patterns during a study on "**Pairwise Spatial Correlation of SARS-Corona Viruses**". However, the genetic functional properties of SARS-Corona type viruses are different for different strains. The scope of this paper is limited to the study of Rajan Transform based spectral correlation among the genome data under consideration. Every mutated virus strain contracted by an individual is likely to generate different strains in vitro, depending on the health conditions of that individual. Our study has indicated this possibility.

Curtain Raiser

सूक्ष्मयोनीनि भूतानि तर्क गम्यानि कानि चित
पक्ष्मणो ऽपि निपातेन येषां सयात सकन्धपर्ययः

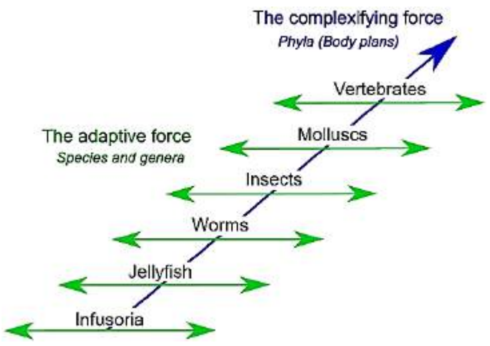
There are many creatures that are so minute that their existence can only be inferred.
With the falling of the eyelids alone, they are destroyed. (**Mahabharata**)

अन्नं न परिचक्षीत। तद्व्रतम्। आपोवा अन्नम्। ज्योतिरन्नादम्। अप्सु ज्योतिः प्रतिष्ठितम्। ज्योतिष्यापः प्रतिष्ठिताः। तदेतदन्नमन्ने प्रतिष्ठितम्। स य एतदन्नमन्ने प्रतिष्ठितं वेद प्रतितिष्ठति। अन्नवानन्नादो भवति। महान् भवति प्रजया पशुभिर्ब्रह्मवर्चसेन। महान् कीर्त्या।।

His vows that he should not discard food. Water indeed is consumable (food); fire is its consumer. Fire is in water. Water is in fire. Thus, one food is lodged in another food, meaning consumables become consumers and of course, consumers become consumables in a different context. He who knows thus that one food is lodged in another, gets firmly established. He becomes a possessor and a consumer. He becomes rich in progeny, cattle, and the luster of holiness and great in glory.

(**Thaitiriya Upanishad**)

RNAs and DNAs are the fundamental units used in this process of consumables and consumers interchanging their roles. Life Forms are constructs made up of these units. RNAs could be viewed as 'Viruses'. They are constructive as well as destructive. Viruses play a constructive role in the forward process of evolution when the hosts can consume them, whereas, the same viruses play a destructive role in devolution, the reverse process of evolution when the hosts are unable to consume them.



"Lamarck's theory of evolution involved a complexifying force that progressively drives animal body plans towards higher levels, creating a ladder of phyla, as well as an adaptive force that causes animals with a given body plan to adapt to circumstances. The idea of progress in such theories permits the opposite idea of decay, seen in devolution".

https://en.wikipedia.org/wiki/Devolution_(biology)

At present, SARS-Corona type viruses act as destructive viruses and they do their job. It is up to humans to understand them, their role and try to delay the process of becoming consumables for them in the reverse process.

This report is a result of intensive research carried out by devoted scientists who work with passion.

The team acknowledges **Ms. Prashanthi Govindarajan** (CEO, PRC Global Technologies Inc., Ontario, Canada) for her efforts in developing the theory of Discrete Rajan Transform and **Mr. Rahul Sharma, Mr. Pawan Kumar and Ms. Gayatri Navle** (Data Scientists, Pentagram Research Centre, Hyderabad, Telangana State, India) for their untiring effort in developing complex computational tools used for this study. The team also acknowledges **Professor Jean-Claude Perez** for his suggestion to study partially deleted génomes like strains evolved in WA, Seattle, USA referenced in his research paper titled "COVID-19, SARS and Bats Coronaviruses Genomes Unexpected Exogeneous RNA Sequences", (https://osf.io/d9e5g/).

Keywords: SARS-Corona Viruses, Spectral Correlations, Artificial Intelligence, Machine Learning

I. DATA USED FOR ANALYSIS

Nine complete genomes of viruses as given in the NCBI web site https://www.ncbi.nlm.nih.gov/genbank/sars-cov-2-seqs/ are considered here for data analysis. They are **(1) SARS coronavirus SZ16, complete genome,** GenBank: **AY304488.1; (2) SARS coronavirus ZS-C, complete genome,** GenBank: **AY395003.1; (3) Bat SARS coronavirus Rm1, complete genome,** GenBank: **DQ412043.1; (4) Bat SARS-like coronavirus isolate Rs4084, complete genome,** GenBank: **KY417144.1; (5) Wuhan seafood market, pneumonia virus isolate Wuhan-Hu-1, complete genome,** GenBank: **MN908947.1; (6) Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete genome,** GenBank: **MN908947.2; (7) Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome,** GenBank: **MN908947.3; (8) Bat SARS-like coronavirus isolate bat-SL-CoVZC45, complete genome,** GenBank: **MG772933.1; (9) Bat SARS-like coronavirus isolate bat-SL-CoVZXC21, complete genome.** These genome data are assumed genuine and analysis carried out on them.

Genome data is a characteristic sequence consisting of symbols A, T, G and C, and it is numerically coded as a sequence consisting of the numbers 1, 2, 3 and 4 (i.e., A ↔1, T ↔2, G ↔3 C ↔4). For example, the genome data having the NCBI ID: LC523807.1 and its numerical equivalent is given below.

```
aactggcagt   aaccagaatg   gagaacgcag   tggggcgcga   tcaaaacaac   gtcggcccca   aggtttaccc   aataatactg   cgtcttggtt   caccgctctc
actcaacatg   gcaaggaaga   ccttaaattc   cctcgaggac   aaggcgttcc   aattaacacc   aatagcagtc   cagatgacca   aattggctac   taccgaagag
ctaccagacg   aattcgtggt   ggtgacggta   aaatgaaaga   tctcagtcca   agatggtatt   tctactacct   aggaactggg   ccagaagctg   gacttcccta
tggtgctaac   aaagacggca   tcatatgggt   tgcaactgag   ggagccttga   atacacc

1142334132   1144131123   3131143413   2333343431   2411114114   3243344441   1332221444   1121121423   4324223322   4144342424
1424114123   3411331131   4422111224   4424313314   1133432244   1122114144   1121341324   4131231441   1122334214   2144311313
4214413143   1122432332   3323143321   1112311131   2424132441   1312332122   2421421442   1331142333   4413113423   3142244421
2332342114   1113143341   2412123332   2341142313   3313442231   1214144
```

One can code any genome sequence, as shown above.

II. ADJOINTS OF GENOME SEQUENCES AND THEIR ARRAY REPRESENTATIONS

Adjoint of a particular nucleotide in a genome sequence is the binary sequence obtained by substituting the particular nucleotides in the genome sequence by 1's and the others by 0's. Figs. 1-9 respectively portray the array images of the genomes AY304488.1, AY395003, DQ412043 and KY417144.1, MN908947.1, MN908947.2, MN908947.3, Mg772933 and Mg772934 and their adjoints arrays. All the arrays are of a width of 128 and a height of (LENGTH/128) + required number of 0's to complete the last row. For example, the genome 1: (AY304488.1) has a LENGTH of 29731. Its corresponding sequence array will have a width of 128 and a height of 238. Four arrays of the adjoints A (n), T (n), G (n) and C (n) of size 128x238 are also shown in Figs. 1-9. A color code has been assumed here as red for adenine, green for thymine, yellow for guanine, and blue for cytosine. That is how a genome array AY304488.1, for instance, is a multicolored array of size 128x238, and its adjoints arrays are in respective colors. Color-coding has been given to arrays in order to discriminate one from the rest.

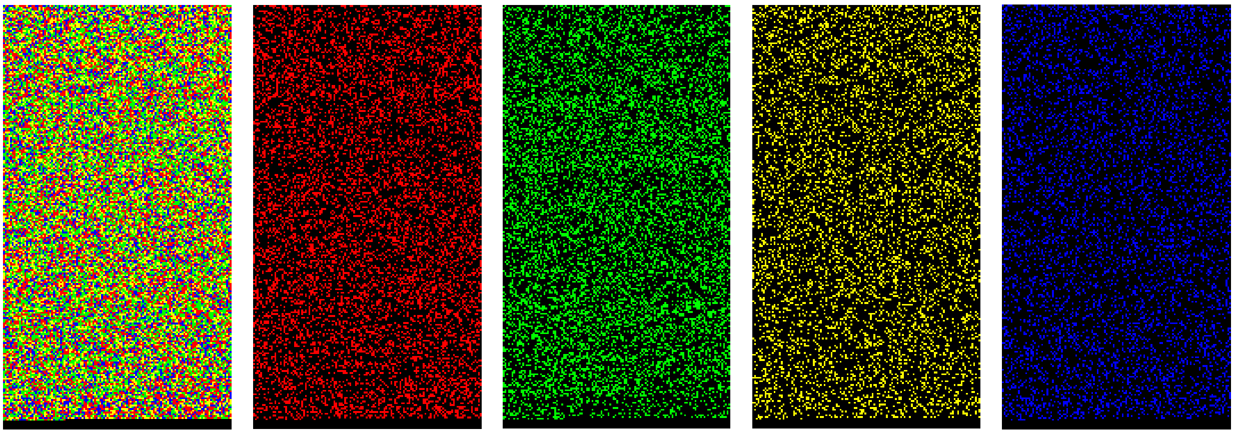

AY304488.1 A(n) T(n) G(n) C(n)

Fig. 1: Genome 1: AY304488.1 and its adjoints

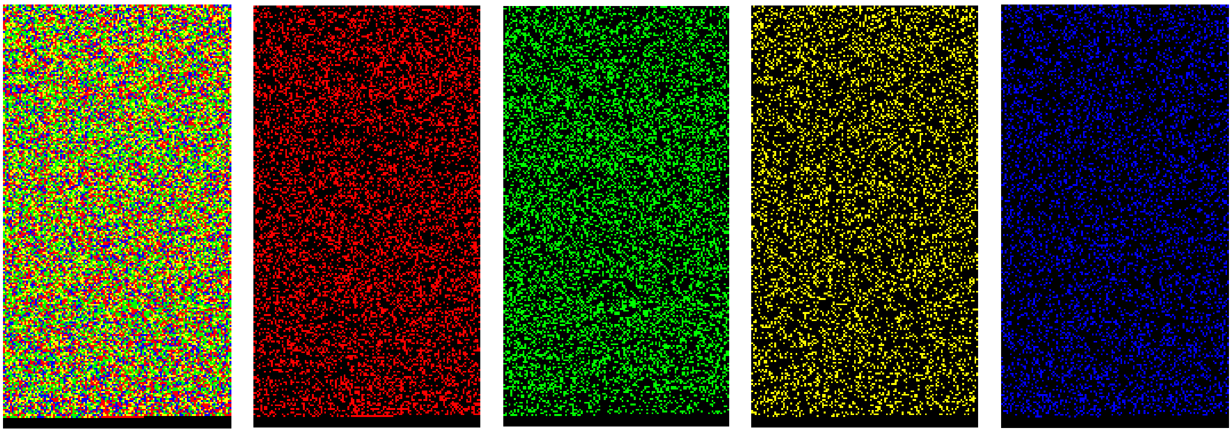
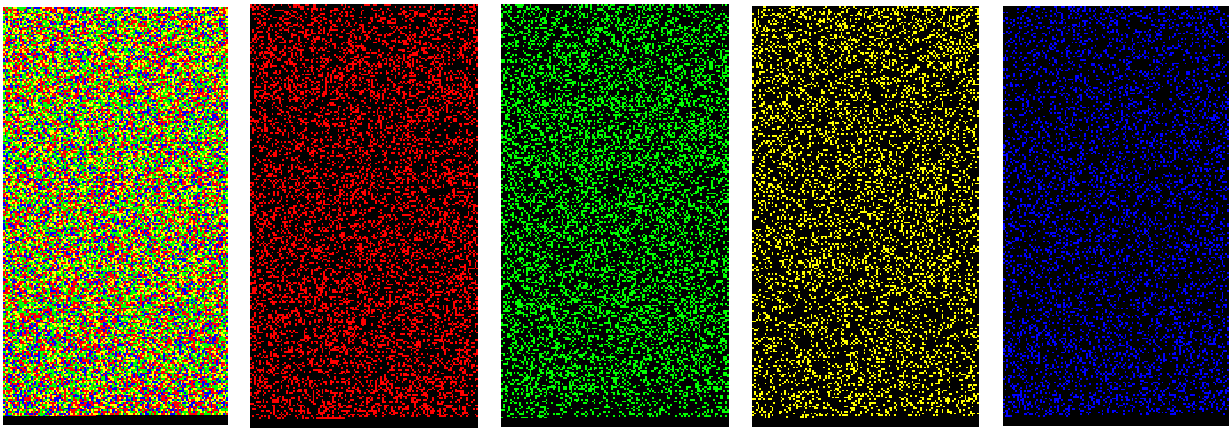
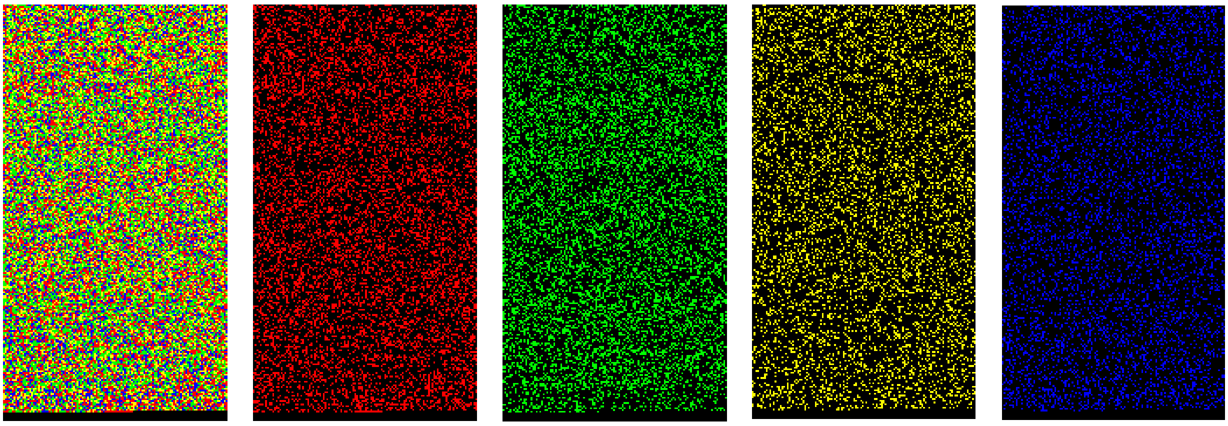

AY395003 A(n) T(n) G(n) C(n)

Fig.2: Genome 2: AY395003 and its adjoints



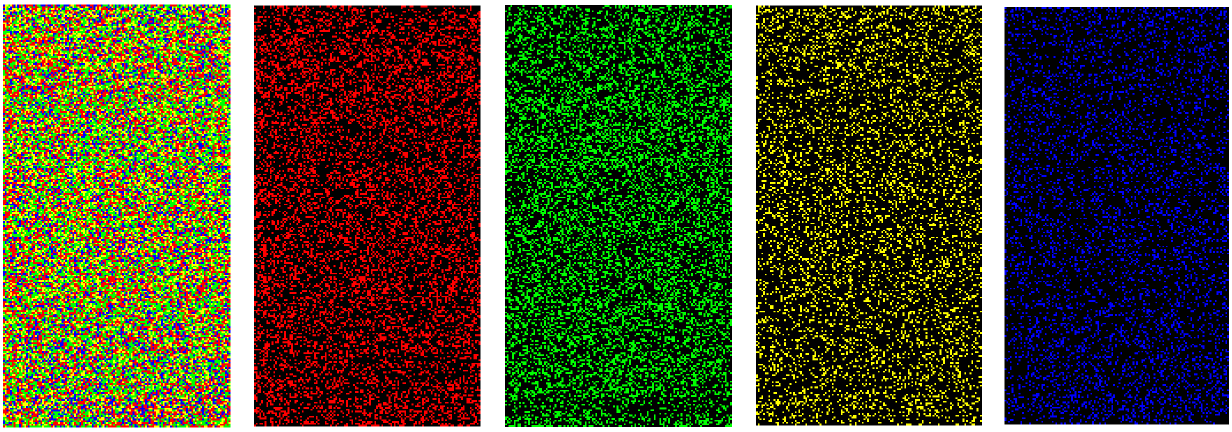
DQ412043 A(n) T(n) G(n) C(n)

Fig.3: Genome 3: DQ412043 and its adjoints



KY417144.1 A(n) T(n) G(n) C(n)

Fig.4: Genome 4: KY417144.1 and its adjoints



MN908947.1 A(n) T(n) G(n) C(n)

Fig.5: Genome 5: MN908947.1 and its adjoints

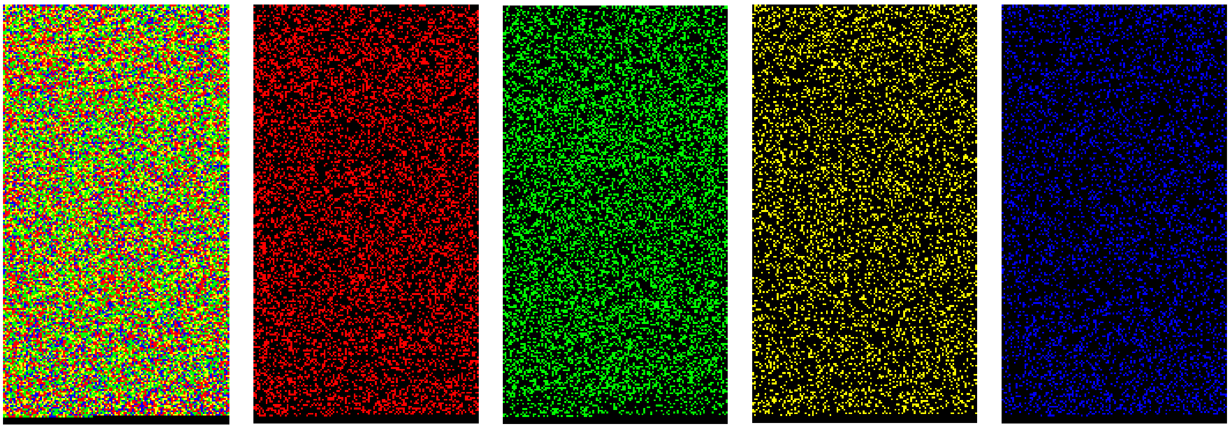

MN908947.2 A(n) T(n) G(n) C(n)

Fig.6: Genome 6: MN908947.2 and its adjoints

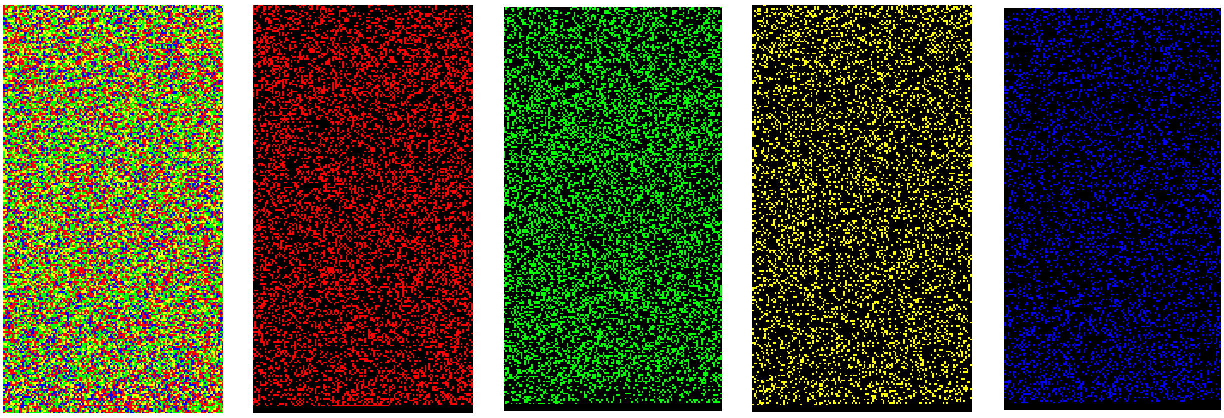

MN908947.3 A(n) T(n) G(n) C(n)

Fig.7: Genome 7: MN908947.3 and its adjoints

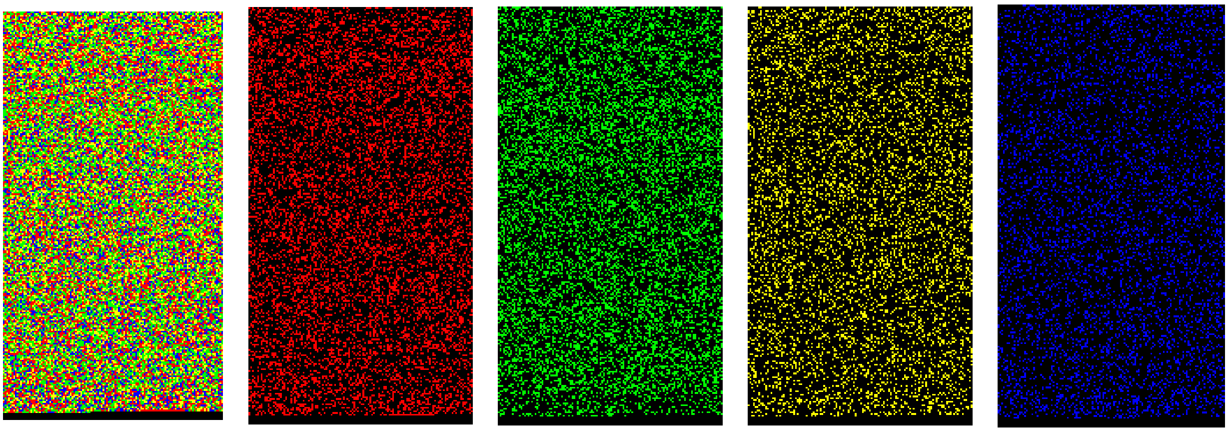
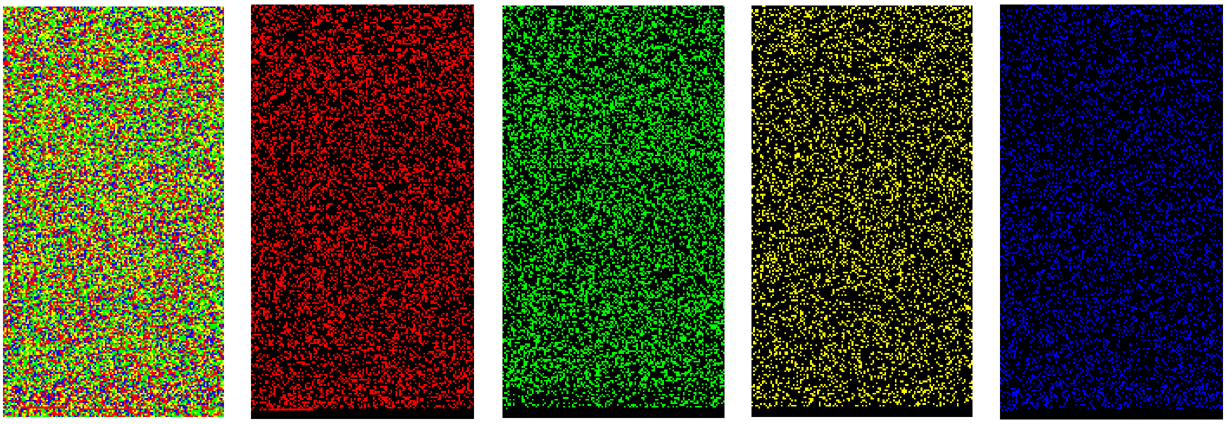

Mg772933 A(n) T(n) G(n) C(n)

Fig.8: Genome 8: Mg772933 and its adjoints



Mg772934 A(n) T(n) G(n) C(n)

Fig.9: Genome 9: Mg772934 and its adjoints

Total number of nucleotides in each genome

The total number of nucleotides in the genome is the total number of A, T, G, and C, respectively. These are listed below, along with the graph.

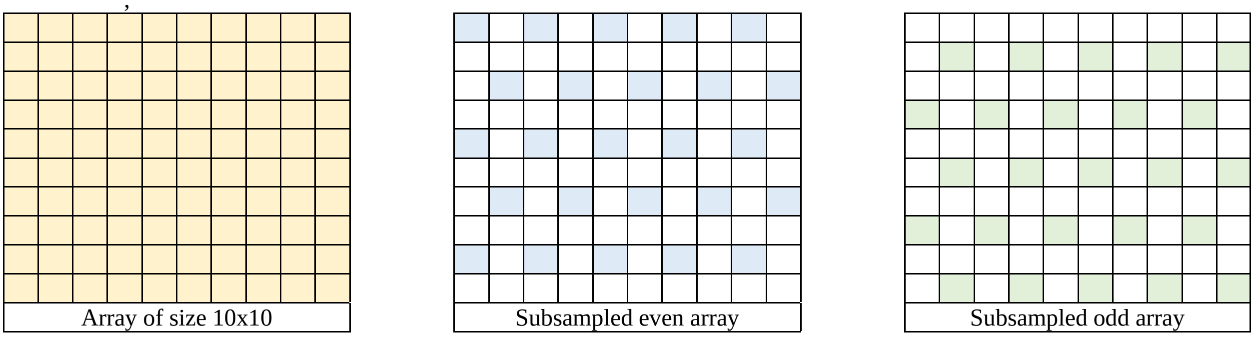
Genome	Genome ID	Total no. of Adenines	Total no. of Thymines	Total no. of Guanines	Total no. of Cytosines	Total no. of Nucleotides
1	AY304488.1	8461	9131	6191	5948	29731
2	AY395003	8454	9106	6167	5920	29647
3	DQ412043	8499	9019	6242	5989	29749
4	KY417144.1	8487	9094	6149	5995	29770
5	MN908947.1	9155	9802	5951	5605	30473
6	MN908947.2	8921	9592	5860	5502	29875
7	MN908947.3	8954	9594	5863	5492	29903
8	Mg772933	8740	9468	6020	5574	29802
9	Mg772934	8734	9455	5976	5567	29732

III. RECTANGULAR SUBSAMPLING OF ARRAYS OF ADJOINTS OF GENOME SEQUENCES

It is a usual practice in signal processing to subsample an analog signal, quantize the subsamples, convert them to digital bit sequences, and apply algorithms developed in the framework of discrete algebra and logic with the idea of processing the digital signals to extract hidden information and carry out pattern recognition. In the case of genomic signal processing also this kind of approach was tried mainly to extract hidden genetic information and carry out pattern recognition. The first step towards this is to convert a characteristic genome sequence into a number sequence by a coding morphism, for instance, A (adenine) is replaced by the number '1', T (thymine) is represented by the number '2', G (guanine) is represented by the number '3' and C (cytosine) is represented by the number '4'. Thus, a genomic characteristic sequence becomes a number sequence consisting of 1's, 2's, 3's, and 4's. Next, adjoints of nucleotides have to be obtained from the original genome sequences (Ref: Figs. 1 to 9). Now the adjoint arrays are subsampled using the following schema:

Rectangular checkerboard subsampling of an adjoint array for getting even and odd arrays

For example let us consider an array pattern given in Fig. 10. The even and odd arrays are obtained as shown in Fig. 11.

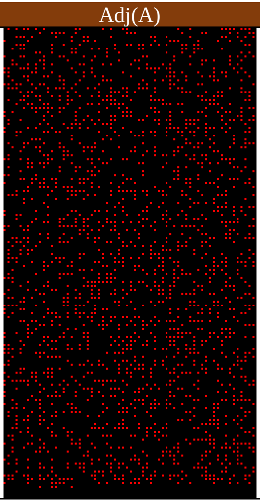
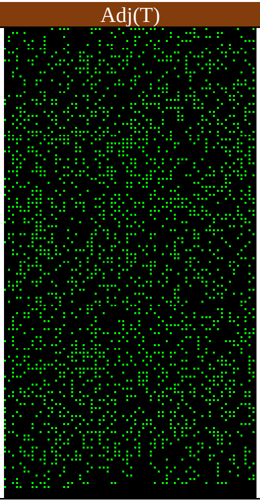
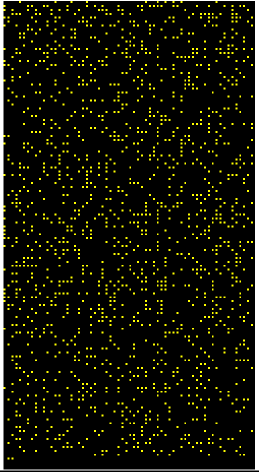
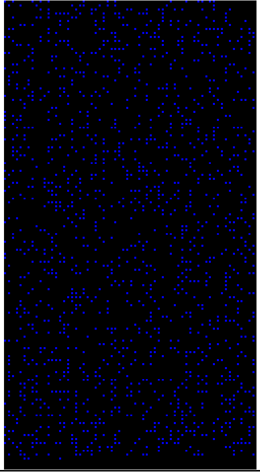
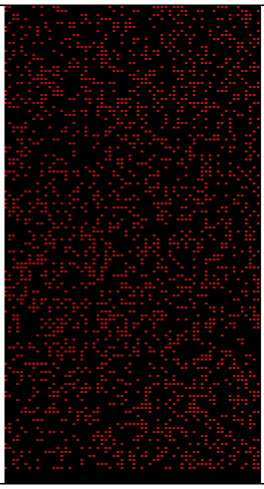
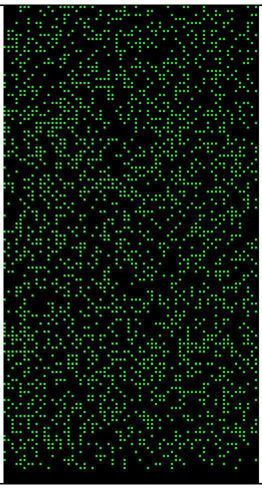
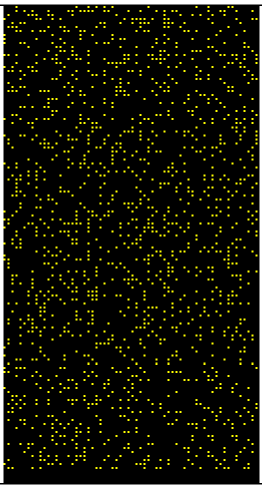
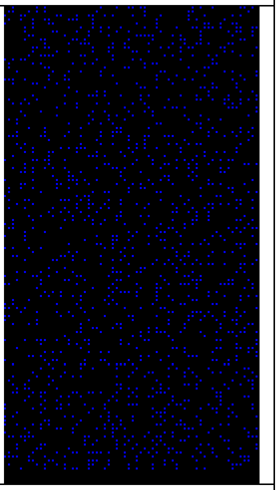
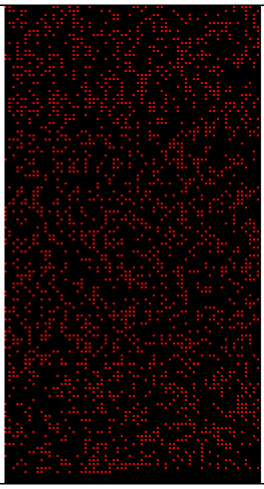
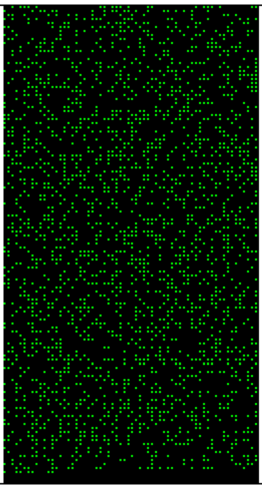
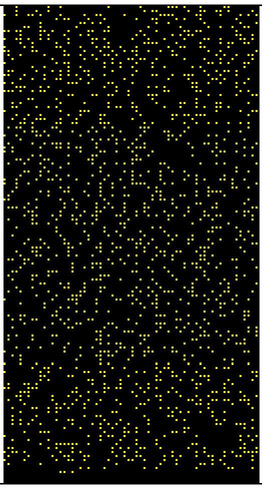
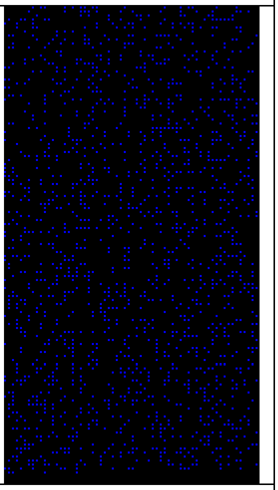


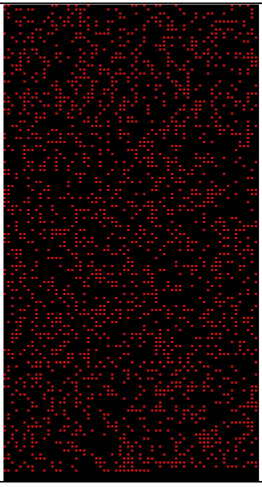
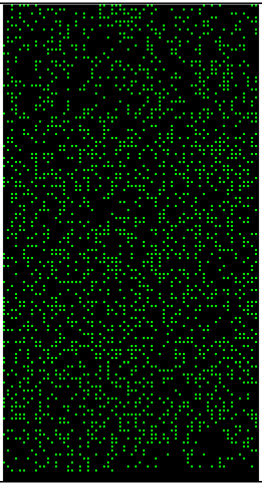
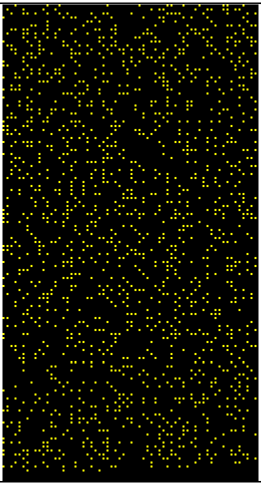
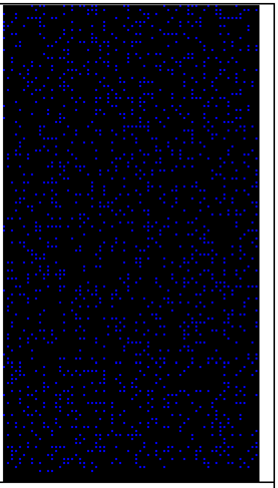
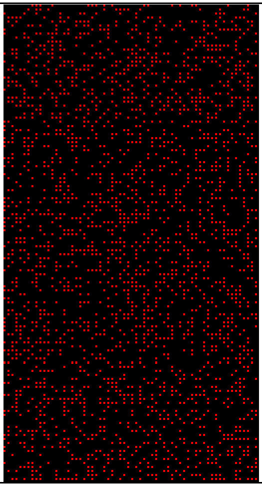
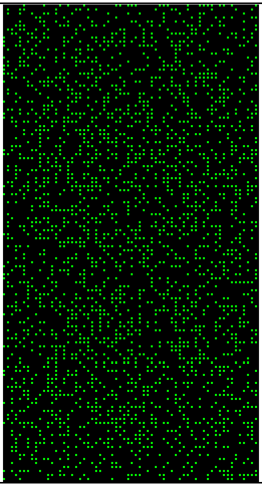
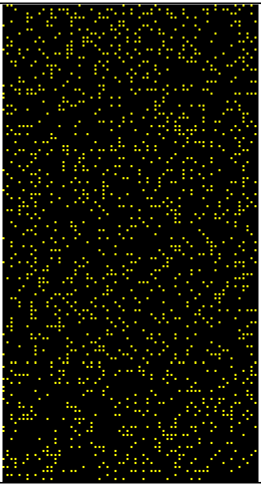
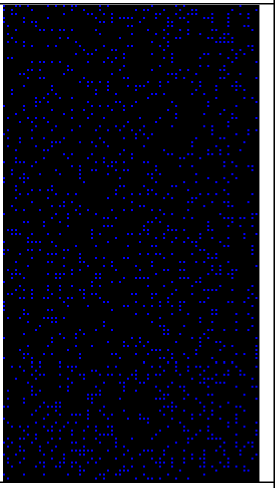
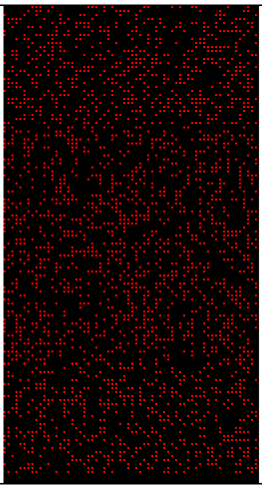
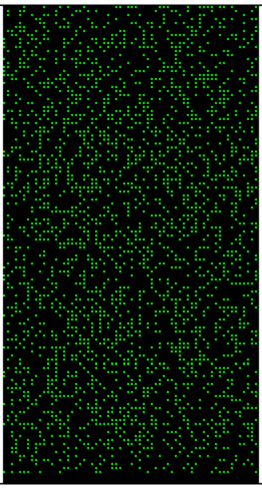
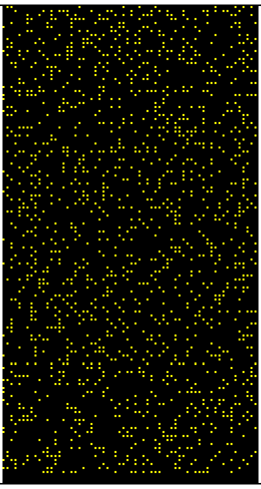
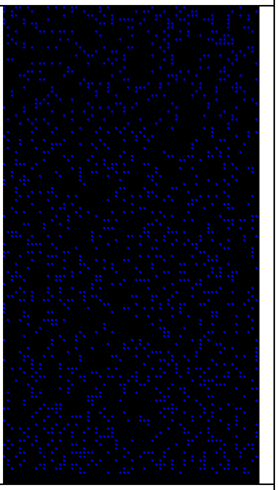
Array of size 10x10 Subsampled even array Subsampled odd array

Fig.10: Checker board and subsampled even and odd arrays

Adjoints arrays of nucleotides of all nine genomes are subsampled as shown above in Fig. 10, and the subsampled arrays are given below. Fig. 11 shows all even subsampled adjoints arrays. Fig. 12 shows all odd subsampled adjoints arrays.

Even arrays of adjoints of Genome 1 (AY304488.1), Genome 2 AY395003, Genome 3 DQ412043, Genome 4 KY417144.1, Genome 5 MN908947.1, Genome 6 MN908947.2, Genome 7 MN908947.3, Genome 8 Mg772933, Genome 9 Mg772934

Even subsampled adjoints arrays				
Genome	Adj(A)	Adj(T)	Adj(G)	Adj(C)
1				
2				
3				

4				
5				
6				

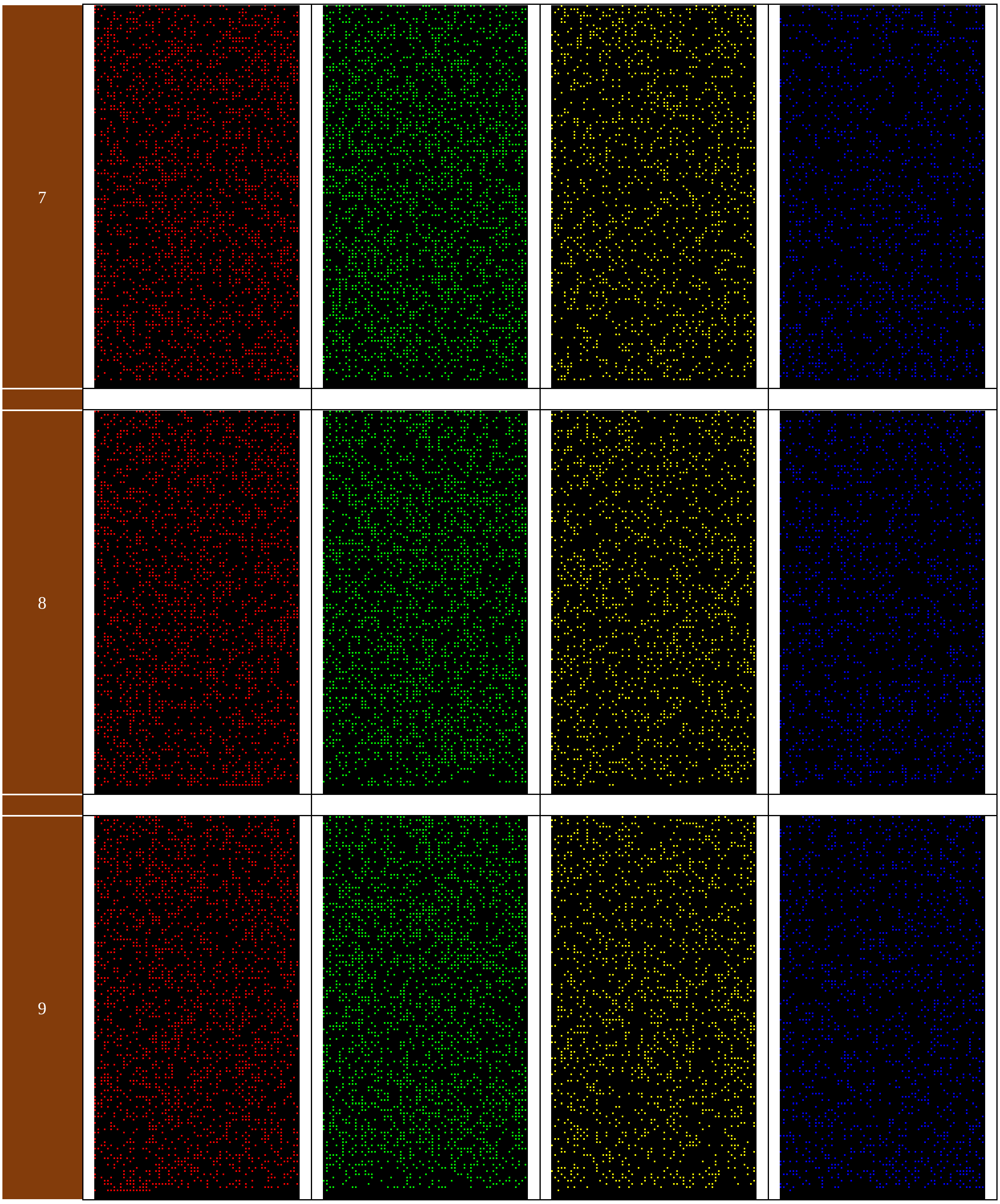


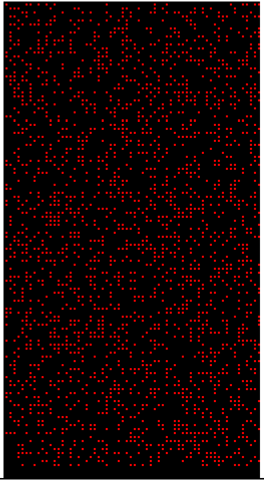
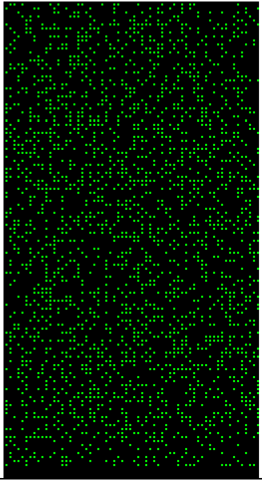
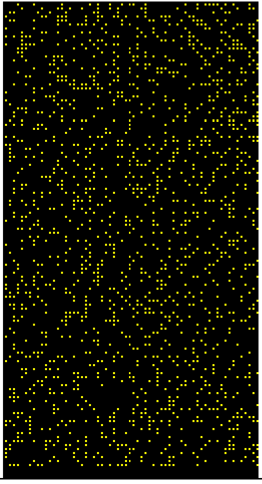
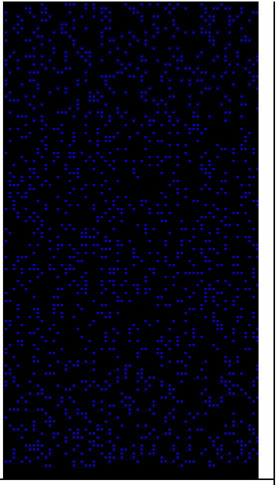
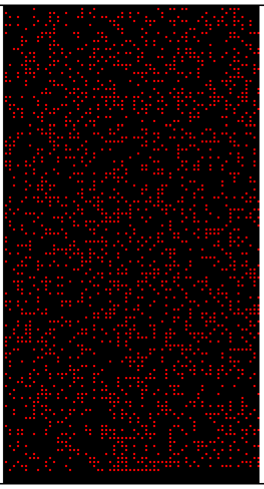
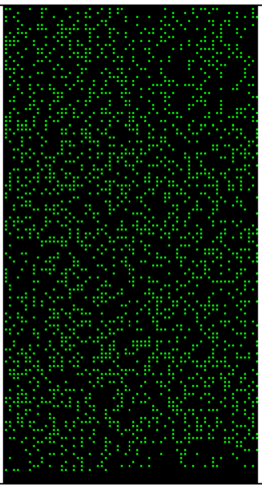
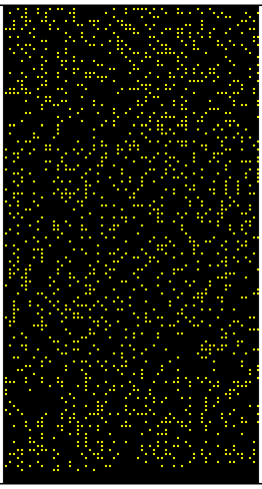
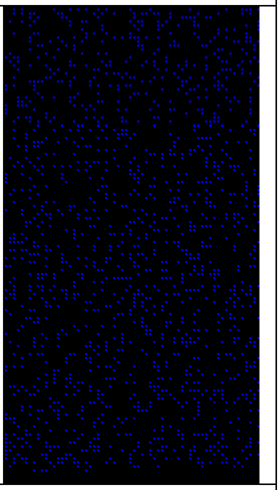
Fig.11: Even subsampled adjoints arrays

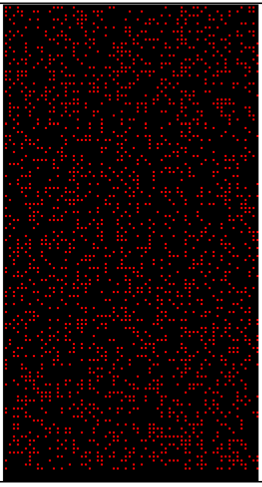
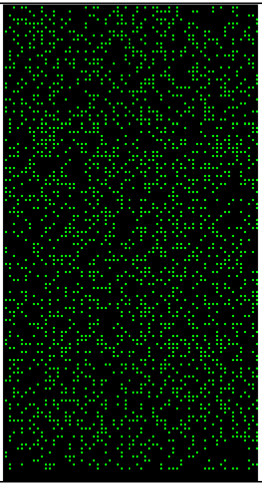
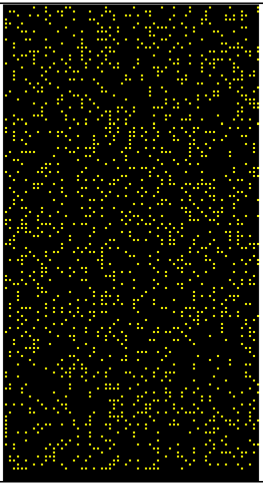
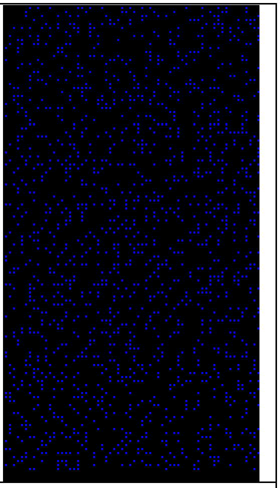
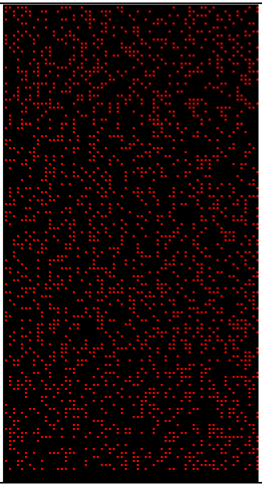
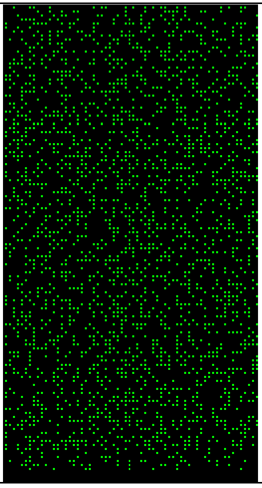
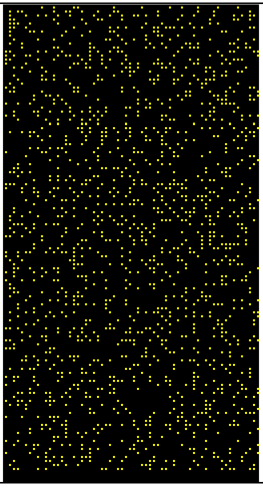
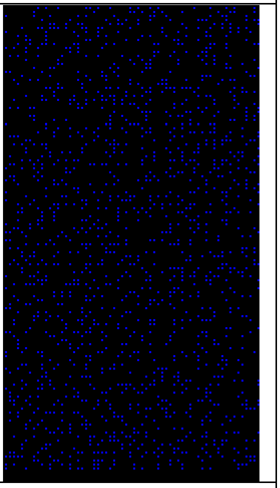
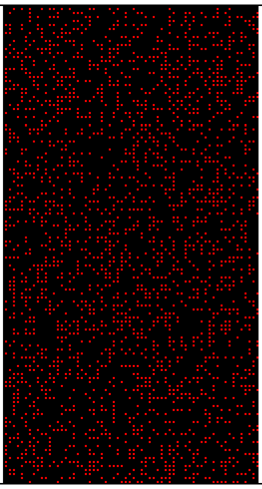
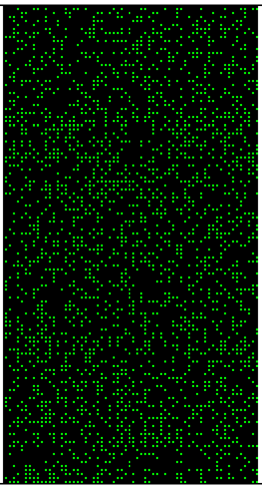
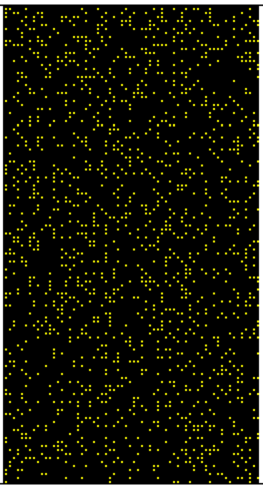
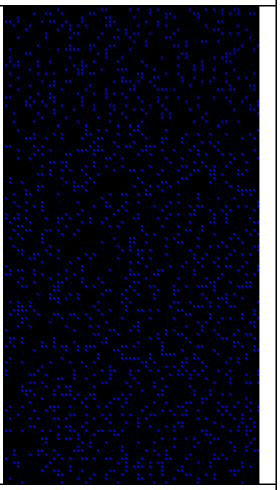
The number of nucleotides present in the even subsampled adjoints arrays of all nine genomes is given below.

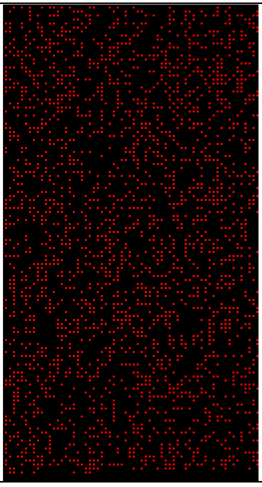
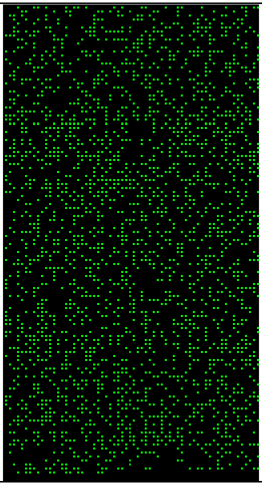
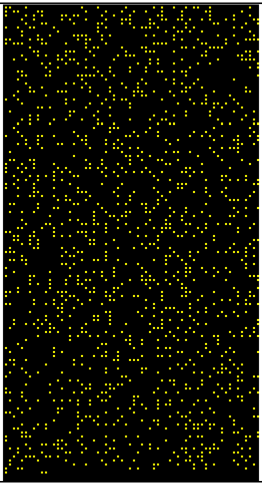
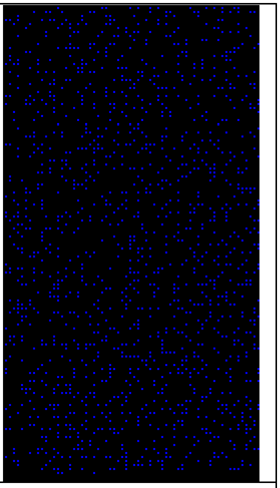
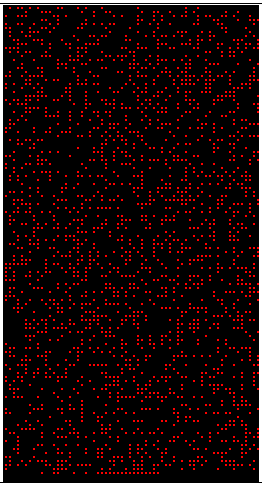
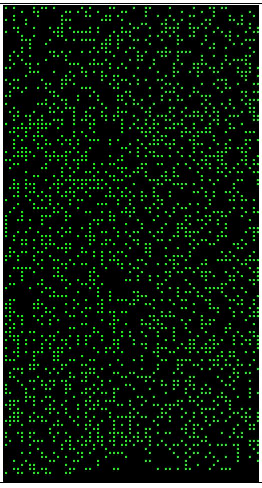
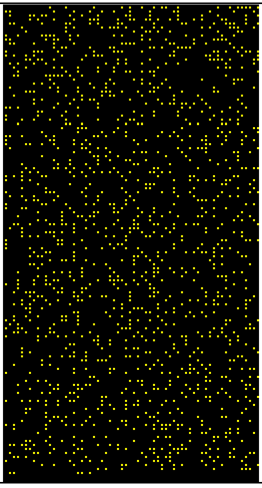
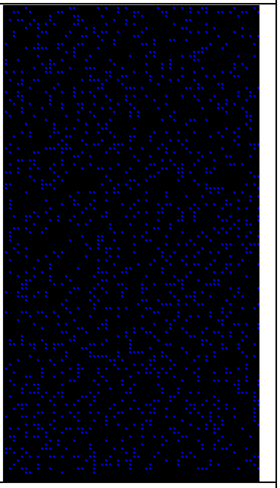
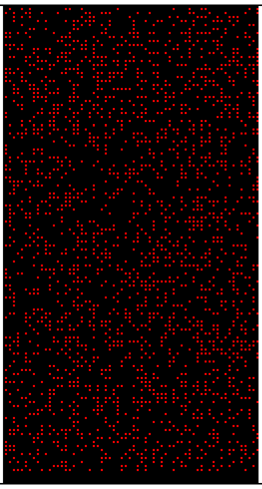
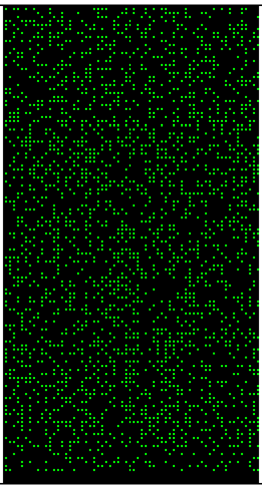
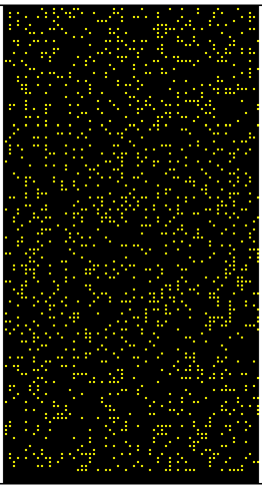
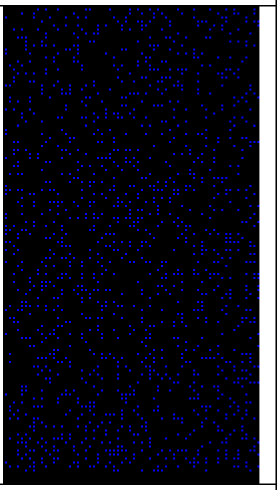
2142	1448	1558	2294
Genome 1 Adenine even subsampled	Genome 1 Cytosine even subsampled	Genome 1 Guanine even subsampled	Genome 1 Thymine even subsampled
2154	1454	1535	2281
Genome 2 Adenine even subsampled	Genome 2 Cytosine even subsampled	Genome 2 Guanine even subsampled	Genome 2 Thymine even subsampled
2096	1568	1591	2196

Genome 3 Adenine even subsampled	Genome 3 Cytosine even subsampled	Genome 3 Guanine even subsampled	Genome 3 Thymine even subsampled
2107	1553	1557	2244
Genome 4 Adenine even subsampled	Genome 4 Cytosine even subsampled	Genome 4 Guanine even subsampled	Genome 4 Thymine even subsampled
2297	1392	1502	2425
Genome 5 Adenine even subsampled	Genome 5 Cytosine even subsampled	Genome 5 Guanine even subsampled	Genome 5 Thymine even subsampled
2245	1371	1485	2387
Genome 6 Adenine even subsampled	Genome 6 Cytosine even subsampled	Genome 6 Guanine even subsampled	Genome 6 Thymine even subsampled
2254	1339	1476	2419
Genome 7 Adenine even subsampled	Genome 7 Cytosine even subsampled	Genome 7 Guanine even subsampled	Genome 7 Thymine even subsampled
2189	1419	1531	2338
Genome 8 Adenine even subsampled	Genome 8 Cytosine even subsampled	Genome 8 Guanine even subsampled	Genome 8 Thymine even subsampled
2179	1412	1522	2329
Genome 9 Adenine even subsampled	Genome 9 Cytosine even subsampled	Genome 9 Guanine even subsampled	Genome 9 Thymine even subsampled

Odd arrays of adjoints of Genome 1 (AY304488.1), Genome 2 AY395003, Genome 3 DQ412043, Genome 4 KY417144.1, Genome 5 MN908947.1, Genome 6 MN908947.2, Genome 7 MN908947.3, Genome 8 Mg772933, Genome 9 Mg772934

Odd subsampled adjoints arrays				
Genome	Adj(A)	Adj(T)	Adj(G)	Adj(C)
1				
2				

3				
4				
5				

6				
7				
8				

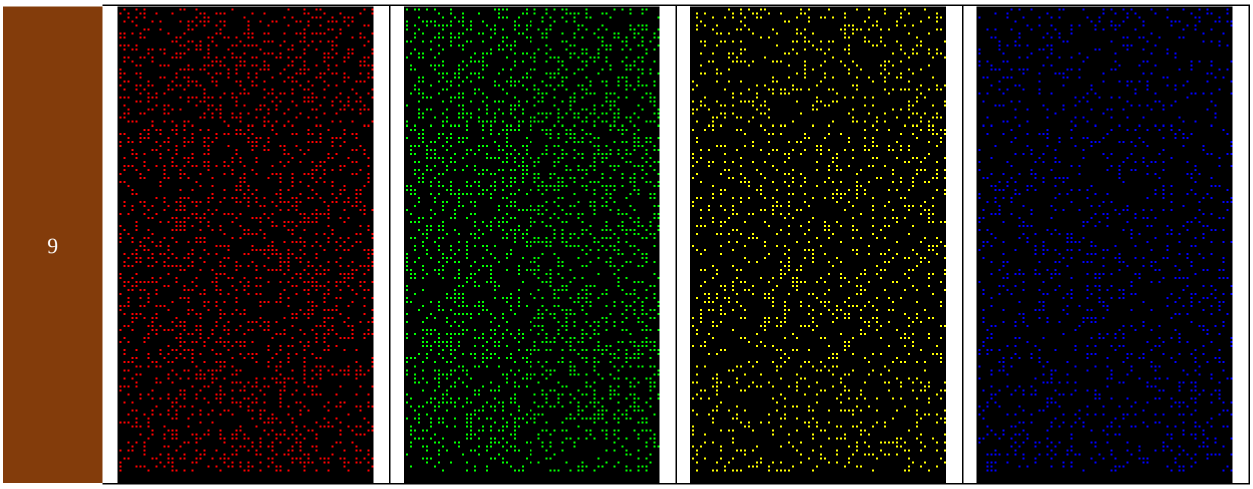


Fig.12: Odd subsampled adjoints arrays

The number of nucleotides present in the odd subsampled adjoints arrays of all nine genomes is given below.

2135	1445	1574	2270
Genome 1 Adenine odd subsampled	Genome 1 Cytosine odd subsampled	Genome 1 Guanine odd subsampled	Genome 1 Thymine odd subsampled
2126	1448	1565	2260
Genome 2 Adenine odd subsampled	Genome 2 Cytosine odd subsampled	Genome 2 Guanine odd subsampled	Genome 2 Thymine odd subsampled
2090	1506	1574	2254
Genome 3 Adenine odd subsampled	Genome 3 Cytosine odd subsampled	Genome 3 Guanine odd subsampled	Genome 3 Thymine odd subsampled
2106	1511	1541	2266
Genome 4 Adenine odd subsampled	Genome 4 Cytosine odd subsampled	Genome 4 Guanine odd subsampled	Genome 4 Thymine odd subsampled
2272	1429	1504	2411
Genome 5 Adenine odd subsampled	Genome 5 Cytosine odd subsampled	Genome 5 Guanine odd subsampled	Genome 5 Thymine odd subsampled
2218	1402	1480	2349
Genome 6 Adenine odd subsampled	Genome 6 Cytosine odd subsampled	Genome 6 Guanine odd subsampled	Genome 6 Thymine odd subsampled
2227	1408	1464	2364
Genome 7 Adenine odd subsampled	Genome 7 Cytosine odd subsampled	Genome 7 Guanine odd subsampled	Genome 7 Thymine odd subsampled
2144	1429	1469	2382
Genome 8 Adenine odd subsampled	Genome 8 Cytosine odd subsampled	Genome 8 Guanine odd subsampled	Genome 8 Thymine odd subsampled
2198	1371	1491	2364
Genome 9 Adenine odd subsampled	Genome 9 Cytosine odd subsampled	Genome 9 Guanine odd subsampled	Genome 9 Thymine odd subsampled

IV.PAIRWISE SPECTRAL CORRELATION OF SUBSAMPLED ADJOINTS ARRAYS

Spectrum $X(k)$ of a discrete function x(n) with duration N is obtained by applying Discrete Fourier Transform (DFT) to x(n).

$$X(k) = \sum_{n=0}^{N-1} x(n)\mathrm{e}^{-j\frac{2\pi}{N}nk}\;;\;\; k = 0, \dots, N-1$$

Now, the Inverse Discrete Fourier Transform (IDFT) is defined as

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\mathrm{e}^{\,j\frac{2\pi}{N}nk}\;;\;\; n = 0, \dots, N-1$$

The complex function $\mathrm{e}^{\,j\frac{2\pi}{N}nk}$ is the exponential form of the basis functions of sinusoids and cosinusoids.

Following the formulation of Discrete Fourier Transform (DFT), a tool for spectral analysis, several variants like Discrete Cosine Transform, Karhunen-Loeve Transform, Discrete Wavelet Transform, Walsh Transform, Hadamard Transform, to name a few, have been introduced as discrete transforms. All these discrete transforms are isomorphic maps connecting time/spatial domains and spectral domains. Isomorphic transforms are used for signal/data analysis. The question that arises here is whether it is possible to have discrete transforms/transformations for classification/pattern recognition. Classification/pattern recognition transforms are essentially homomorphic transforms/transformations. An effort was made by the team E. G. Rajan and his scholars and a novel homomorphic transformation was explored right in 1984, and finally formulated 'Rapid Transform' in 1997, which was named as '**Rajan Transform (RT)**' at a later date. RT is a variant of Hadamard Transform, and it functions as an isomorphic transform yielding spectrum of a data sequence along with its encrypted sequence. RT functions as a homomorphic transform yielding only the spectrum of a data sequence.

4.1 RAJAN TRANSFORM FUNDAMENTALS

Given a discrete sequence x(n) of length N=2^n (where n=1,2,3…), one can compute the RT spectrum X(k) using two approaches (i) using the isomorphic Discrete Rajan Transform (DRT) or (ii) using the homomorphic Rajan Transform (RT).

Discrete Rajan Transform Preliminaries (Prashanthi's Definition (Staffordshire University, 2014))

"Consider an N-dimensional signal vector $\boldsymbol{x}$ where $N = 2^n$ with *n* being a nonnegative integer. To obtain the final DRT transformed signal, *n* number of iterative stages are required. In each stage of the process, there is a unique operating matrix $\boldsymbol{R}_k$ of dimension $\left(\frac{N}{2^{k-1}} \times \frac{N}{2^{k-1}}\right)$ where $R_k = \begin{bmatrix} I_p & I_p \\ -e_k^1.I_p & e_k^1.I_p \end{bmatrix}$. I_p is the identity matrix of order p. At the k^{th} stage, the order of the identity matrix is $p_k = N/2^k$; $k \in \{1,2,\dots,n\}$ and e_k^i is the auxiliary information that defines the inherent phasor relationship between elements $\boldsymbol{x}_k^i(p_k+1)\ and\ \boldsymbol{x}_k^i(1)$ of the ith equilibrium segment of the input signal spectrum at the kth stage. For example, if the input signal sequence to an arbitrary stage $\bar{\boldsymbol{s}} = \{a,b,c,d,e,f,g,h\}$, the auxiliary information e_k is expected to contain inherent phasor relationship between a and e. This auxiliary information, in general, is written as $e_k^i = \begin{cases} -1, & \boldsymbol{x}_k^i(p_k+1) < \boldsymbol{x}_k^i(1) \\ 1, & otherwise \end{cases}$, where $i = \{1,2,\cdots,2^{k-1}\}$, indicates the segment number in the input sequence of kth stage. In each stage, the output signal sequence is denoted as $\boldsymbol{y}_k$and it is obtained from $\boldsymbol{y}_k = R_k\,\boldsymbol{x}_k = \begin{bmatrix} y_k^1 & y_k^2 \dots & y_k^i \end{bmatrix}$. Here Y_k has $2^k p_k$ elements in every stage. When $k = 1, \boldsymbol{x}_1 = \boldsymbol{x}$ (the original input sequence) and for $k > 1$, 2^{k-1} equilibrium segments exist in the input signal sequence for each stage. Each of these segments is to be associated with their respective auxiliary information. The elements of the vector $\boldsymbol{e}_k$ correspond to the auxiliary information of the respective equilibrium segments in kth stage. It is defined as $\boldsymbol{e}_k = \begin{bmatrix} e_k^1 & e_k^2 & \cdots & e_k^i \end{bmatrix}$. For example, if $\boldsymbol{y}_{k-1} = [a\ b\ c\ d\ e\ f\ g\ h]^T$ then $\bar{\mathbf{x}}_\mathrm{k} = [\bar{\boldsymbol{x}}_k^1 \quad \bar{\boldsymbol{x}}_k^2] = \begin{bmatrix} a & e \\ b & f \\ c & g \\ d & h \end{bmatrix}$ and, e_k^1 & e_k^2 are the auxiliary phasor information that corresponds to a & c and e & g respectively. The auxiliary phasor values in the vector $\boldsymbol{e}_k$ is obtained accordingly. In order to construct the common operator matrix $\mathbf{R}_\mathrm{k}$ in all the stages, auxiliary phasor information e_k^1 from $\boldsymbol{e}_k$ is used. Also, for k > 1, the transformed output from one stage is reshaped into equilibrium segments iteratively and it is given by $\boldsymbol{x}_k = \begin{bmatrix} \bar{\boldsymbol{x}}_k^1 & \mu_k^1 \cdot \bar{\boldsymbol{x}}_k^2 \cdots & \mu_k^{i-1} \cdot \bar{\boldsymbol{x}}_k^i \end{bmatrix}$, where $\boldsymbol{\mu}_k = \begin{bmatrix} \mu_k^1 & \mu_k^2 & \cdots & \mu_k^{i-1} \end{bmatrix}$ and $\mu_k^{i-1} = e_k^1 \times e_k^i$ for $k > 1$. In other words, $\bar{\mathbf{x}}_{\mathrm{k}+1} = \begin{bmatrix} \mathbf{y}_\mathbf{k}^\mathbf{1}(1) & \mathbf{y}_\mathbf{k}^\mathbf{1}(2) \cdots & \mathbf{y}_\mathbf{k}^\mathbf{1}(\mathrm{p}_\mathrm{k}) \\ \mathbf{y}_\mathbf{k}^\mathbf{2}(\mathrm{p}_\mathrm{k}+1) & \mathbf{y}_\mathbf{k}^\mathbf{2}(\mathrm{p}_\mathrm{k}+2) & \mathbf{y}_\mathbf{k}^\mathbf{2}(2\mathrm{p}_\mathrm{k}) \\ \vdots & \vdots & \vdots \\ \mathbf{y}_\mathbf{k}^\mathbf{i}(2^{\mathrm{k}-1}\mathrm{p}_\mathrm{k}+1) & \dots & \mathbf{y}_\mathbf{k}^\mathbf{i}(2^\mathrm{k}\mathrm{p}_\mathrm{k}) \end{bmatrix}^\mathrm{T} = \begin{bmatrix} \bar{\mathbf{x}}_{\mathrm{k}+1}^1 & \bar{\mathbf{x}}_{\mathrm{k}+1}^2 \cdots & \bar{\mathbf{x}}_{\mathrm{k}+1}^\mathrm{i} \end{bmatrix}$. The equation $\bar{\mathbf{x}}_{\mathrm{k}+1}$ represents the general expression for splitting of signal spectrum into equilibrium segments. This process is continued for *n* number of stages to obtain the final DRT transformed output. The first sample value in the final output of DRT is called Cumulative Point Index (CPI) and it represents the cumulative energy present in the input signal. Consider an example **x**= [1, 9, 6, 2, 3, 1, 5, 1]$^\mathrm{T}$. The length of the input is N=8, and thus, the number of iterative stages that are required to compute DRT transformed output is n=3 and k = {1, 2, 3}. In the first stage, k=1, p=4, and $\boldsymbol{x}_1 = \boldsymbol{x}$. Thus, the dimension of the operating matrix $\boldsymbol{R}_1$is (8×8). The auxiliary information is obtained by considering the condition and comparison is made between 1 and 3. Thus the value of $e_1^1$is 1. Operating matrix $\boldsymbol{R}_1$can be constructed as $\boldsymbol{R}_1 = \begin{bmatrix} I_4 & I_4 \\ -I_4 & I_4 \end{bmatrix}$. The transformed output from the first stage is calculated as $\boldsymbol{y}_1 = \boldsymbol{R}_1\,\boldsymbol{x}_1 =$ [4 10 11 3 2 -8 -1 -1]$^\mathrm{T}$. Now k=2, p=2 and $\boldsymbol{R}_2$is of dimension (4×4). And $\boldsymbol{x}_2$is obtained by reshaping $\boldsymbol{y}_1$. Thus, $[\bar{\boldsymbol{x}}_2^1 \quad \bar{\boldsymbol{x}}_2^2] = \begin{bmatrix} 4 & 2 \\ 10 & -8 \\ 11 & -1 \\ 3 & -1 \end{bmatrix}$. There are two equilibrium segments in the second stage i.e. $i = 2$ and hence $\boldsymbol{e}_2 = [e_2^1 \quad e_2^2\ ]$ and the values correspondingly are obtained as $\boldsymbol{e}_2 = [1 \quad -1]$, $\boldsymbol{\mu}_2 = [\mu_2^1]$ and $\mu_2^1 = 1 \times -1 = -1$. Thus, $\boldsymbol{x}_2$ is obtained as $\boldsymbol{x}_2 = \begin{bmatrix} 4 & -2 \\ 10 & 8 \\ 11 & 1 \\ 3 & 1 \end{bmatrix}$. To construct operator matrix $\boldsymbol{R}_2$, $e_2^1 = 1$ is used and $\boldsymbol{R}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$. The output of the second stage is calculated as $\boldsymbol{y}_2 = \boldsymbol{R}_2\,\boldsymbol{x}_2 = \begin{bmatrix} 15 & -1 \\ 13 & 9 \\ 7 & 3 \\ -7 & -7 \end{bmatrix}$. Similarly, for the last stage, k=3, p=1 and $\boldsymbol{R}_3$is of dimension (2×2). Now, $[\bar{\boldsymbol{x}}_3^1 \quad \bar{\boldsymbol{x}}_3^2 \quad \bar{\boldsymbol{x}}_3^3 \quad \bar{\boldsymbol{x}}_3^4] = \begin{bmatrix} 15 & 7 & -1 & 3 \\ 13 & -7 & 9 & -7 \end{bmatrix}$. The auxiliary information values can be obtained for the four equilibrium segments ($\because i = 4$) as $\boldsymbol{e}_3 = [e_3^1 \quad e_3^2 \quad e_3^3 \quad e_3^4] = [-1 \quad -1 \quad 1 \quad -1]$ and $\boldsymbol{\mu}_3 = [\mu_3^1 \quad \mu_3^2 \quad \mu_3^3] = [1 \quad -1 \quad 1]$.

The value of $\boldsymbol{x}_3$is thus calculated as $\boldsymbol{x}_3 = \begin{bmatrix} 15 & 7 & 1 & 3 \\ 13 & -7 & -9 & -7 \end{bmatrix}$. The final transformed DRT output is obtained as

$$\boldsymbol{y}_3 = \boldsymbol{R}_3\, \boldsymbol{x}_3 = \begin{bmatrix} 28 & 0 & -8 & -4 \\ 2 & 14 & 10 & 10 \end{bmatrix}$$

Thus after three iterative stages, the final DRT output is $\boldsymbol{y}_3$. In this case, the CPI value is 28.

Inverse Discrete Rajan Transform

The Inverse Discrete Rajan transform (IDRT) is used to retrieve the input signal from DRT spectrum in the presence of the auxiliary phasor information $\boldsymbol{e}_k^1$ and $\boldsymbol{\mu}_k$. Assuming that these values are known a priori, the DRT operator $\boldsymbol{R_k}$ is obtained. The expression used to retrieve back the input signal in each stage is $\widetilde{\boldsymbol{x}}_m = \frac{1}{2}\,[\boldsymbol{R}_m \boldsymbol{y}_m] = [\boldsymbol{x}_m^1 \quad \boldsymbol{x}_m^2 \ldots \quad \boldsymbol{x}_m^i]^{\boldsymbol{T}}$, where m = {k, k − 1, ..., 1}. In the forward process, the sequence was split into equilibrium segments, whereas in the inverse DRT process, they are recombined to retrieve the input sequence iteratively. When $m = k, \boldsymbol{y_m} = \boldsymbol{y}_k$ (the final DRT transformed output) and for $m < k$,

$$\boldsymbol{y}_{m-1} = \left[\overline{\boldsymbol{y}}_m(1) \quad \mu_k^1 \cdot \overline{\boldsymbol{y}}_m(2) \cdots \quad \mu_k^{i-1} \cdot \boldsymbol{y}_m(i)\right]$$

$$\overline{\boldsymbol{y}}_{m-1} = \begin{bmatrix} \boldsymbol{x}_m^i(1) & \boldsymbol{x}_m^i(2p_k+1) \cdots & \boldsymbol{x}_m^i(2^{k-1}p_k+1) \\ \boldsymbol{x}_m^i(2) & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \boldsymbol{x}_m^i(2p_k) & \boldsymbol{x}_m^i(2^2p_k) & \boldsymbol{x}_m^i(2^kp_k) \end{bmatrix}$$

After the final stage in the inverse process, the length of the output is the same as the input and it can be observed that $\mathbf{x} = \tilde{\mathbf{x}}$. Hence, Discrete Rajan Transform is proved to be an isomorphic transform."

Rajan Transform Preliminaries (E. G. Rajan's Definition (Indian Institute of Technology, Kanpur, 1984))

Rajan Transform is essentially a fast algorithm developed on the lines of Decimation-In-Frequency Fast Fourier Transform algorithms, but it is functionally different from the DIF-FFT algorithm. Given a number sequence x(n) of length N, which is a power of 2, first it is divided into the first half and the second half each consisting of (N/2) points so that the following holds good.

g(i) =x(i)+x(i+N/2) ; 0≤j≤N/2 ; 0≤i≤N/2
h(j) =|x(i)-x(i-N/2)| ; 0≤j≤N/2; 0≤i≤N/2

Now each (N/2) point segment is further divided into two halves each consisting of N/4 points so that the following holds good.

g1(k)=g(i)+g(j+(N/4)); 0≤k≤(N/4); 0≤j≤(N/4)
g2(k)=|g(i)-g(i-(N/4)| ; 0≤k≤(N/4); (N/4)≤j≤(N/2)
h1(k)=h(j)+h(j+(N/4)); 0≤k≤(N/4); 0≤j≤(N/4)
h2(k)=|h(j)-h(j-N/4)| ; 0≤k≤(N/4); 0≤j≤(N/4)

This process is continued until no more division is possible. The total number of stages thus turns out to be $\log_2 N$. Let us denote the sum and difference operators respectively as '+' and '~'. Then the signal flow graph for the Rajan Transform of a number sequence of length 16 would be of the form shown in Fig. 13. If x(n) is a number sequence of length $N=2^k$; k>0, then its Rajan Transform (RT) is denoted as X(k). RT is applicable to any number sequence and it induces an isomorphism in a class of sequence, that is, it maps a domain set consisting of the cyclic and dyadic permutations of a sequence on to a range set consisting of sequence of the form X(k)E(r) where X(k) denotes the permutation invariant RT and E(r) an encryption code corresponding to an element in the domain set. This map is a one-to-one and on to correspondence and an inverse map also exists. Hence, it is viewed as a transform. Fig. 13 shows a functional block diagram of a 16-point Rajan Transform algorithm.

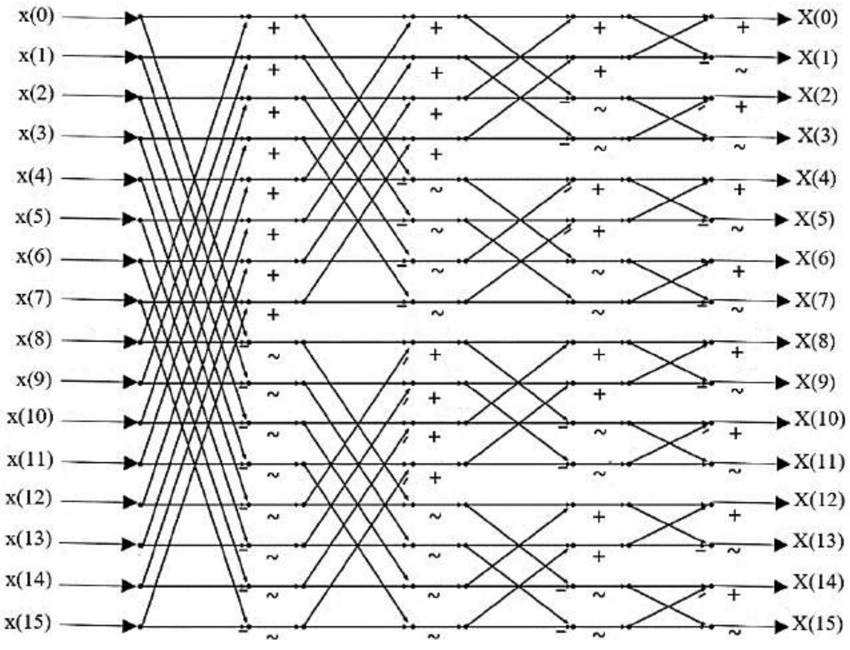


Fig.13: Signal Flow graph of Rajan Transform (Signal length is 16)

The operations of arithmetic addition is denoted by the symbol + and the arithmetic difference by the symbol ~ in the diagram. The number of stages of computation is 4. In general, if the number sequence is of length $N=2^n$, (where n is 1, 2, 3,…), then the number of stages of computation would be 'n'. For example, let us consider the sequence x(n) = 3,8,5,6,0,2,9,6. Then the RT spectrum of x(n) is X(k) = 39,5,13,9,13,1,7,5.

Algebraic Properties of Rajan Transform

Cyclic Shift Invariance Property

Let us consider the same sequence x(n) = 3,8,5,6,0,2,9,6. Now, one can generate seven more cyclic shifted versions such as $x_{c1}(n)$=6,3,8,5,6,0,2,9; $x_{c2}(n)$=9,6,3,8,5,6,0,2; $x_{c3}(n)$=2,9,6,3,8,5,6,0; $x_{c4}(n)$=0,2,9,6,3,8,5,6; $x_{c5}(n)$=6,0,2,9,6,3,8,5; $x_{c6}(n)$=5,6,0, 2,9,6,3,8 and $x_{c7}(n)$=8,5,6,0,2,9,6,3. The cyclic shifted version of $x_{c7}(n)$ is x(n) itself. One can easily verify that all these eight sequences to have the same X(k)=39,5,13,9,13,1,7,5, meaning Rajan Transform of a given sequence of length N would remain invariant for N cyclically permuted sequences.

Graphical Inverse Invariance Property

Let us consider x(n)=3,8,5,6,0,2,9,6. Its graphical inverse is $x^{-1}(n)$=6,9,2,0,6,5,8,3. Now, one can generate seven more cyclic shifted versions such as $x_{c1}^{-1}(n)$=3,6,9,2,0,6,5,8; $x_{c2}^{-1}(n)$=8,3,6,9,2,0,6,5; $x_{c3}^{-1}(n)$=5,8,3,6,9,2,0,6, $x_{c4}^{-1}(n)$=6,5,8,3,6,9,2,0; $x_{c5}^{-1}(n)$=0,6,5,8, 3,6,9,2; $x_{c6}^{-1}(n)$=2,0,6,5,8,3,6,9 and $x_{c7}^{-1}(n)$=9,2,0,6,5,8,3,6. The cyclic shifted version of $x_{c7}^{-1}(n)$ is $x^{-1}(n)$ itself. One can easily verify that all these eight sequences have the same X(k)=39,5,13,9,13,1,7,5, meaning Rajan Transform of a given sequence of length N would remain invariant for N graphically inverted sequence and its cyclically permuted sequences.

Dyadic Shift Invariance Property

The term 'dyad' refers to a group of two, and the term 'dyadic shift' to the operation of transposition of two blocks of elements in a sequence. For instance, let us consider x(n)=3,8,5,6,0,2,9,6 and transpose its first half with the second half. The resulting sequence $Td^{(2)}[x(n)]$=0,2,9,6,3,8,5,6 is the 2-block dyadic shifted version of x(n). The symbol $Td^{(2)}$ denotes the 2-block dyadic shift operator. In the same manner, one would obtain $Td^{(4)}[Td^{(2)}[x(n)]]$=9,6,0,2,5,6,3,8 and $Td^{(8)}[Td^{(4)}[Td^{(2)}[x(n)]]]$=6,9,2,0,6,5,8,3. Note that the graphical inverse of x(n) is $x^{-1}(n)$=(6,9,2,0,6,5,8,3) and it is the same as $Td^{(8)}[Td^{(4)}[Td^{(2)}[x(n)]]]$=6,9,2,0,6,5,8,3. One can easily verify that all these dyadic shifted sequences have the same X(k), that is, the sequence 39,5,13,9,13,7,5. There is yet another way of dyadic shifting the input sequence x(n) to $Td^{(2)}[Td^{(4)}[Td^{(8)}[x(n)]]]$. Consider the sequence x(n)=3,8,5,6,0,2,9,6 and one can obtain $Td^{(8)}[x(n)]$=8,3,6,5,2,0,6,9; $Td^{(4)}[Td^{(8)}[x(n)]]$=6,5,8,3,6,9,2,0 and $Td^{(2)}[Td^{(4)}[Td^{(8)}[x(n)]]]$=6,9,2,0,6,5,8,3 as dyadic shifts. Note that $Td^{(2)}[Td^{(4)}[Td^{(8)}[x(n)]]] = Td^{(8)}[Td^{(4)}[Td^{(2)}[x(n)]]]$.

Dual Class Invariance Property

Given a sequence x(n), one can construct another sequence y(n) consisting of at least one number which is not present in x(n) such that X(k)=Y(k). In such a case, y(n) is called the '**dual**' of x(n). Consider two sequences x(n)=2,4,2,2 and y(n)=3,1,3,3. Then X(k)=Y(k)=10,2,2,2. An underlying theorem to characterize a sequence of length $N=2^n$ to pair up with a dual sequence is "***A sequence is said to have a dual if and only if its CPI is an even number and is divisible by N/2***". This theorem advocates a necessary condition but not a sufficient condition. This means that all sequences that satisfy this theorem need not form valid dual pair with other sequence vide definition of dual. For example, consider the sequence x(n)=6,8,2,0. This indeed satisfies the theorem. That is, its CPI is 16 and the value of CPI/(N/2) is 8. Now its dual is computed as y(n)=2,0,6,8, which is not a dual of x(n) as per the definition. In such cases, they are called 'self dual' pairs. Some of the properties of dual sequences are: (i) if y(n) is a dual of x(n), then x(n) is also called the dual of y(n). Hence the pair <x(n),y(n)> is called 'dual pair'; (ii) dual of a sequence, say y(n) will necessarily exhibit geometric symmetry together with the original sequence x(n); (iii) each dual pair has a value called 'Differential Mean' (DM), which is equal to $(|x(i)\sim y(i)|)/2;\ 0\leq i\leq(N-1)$ about which the dual sequences are 'flip' symmetric. DM could be a real number. **This dual pair concept is potentially used in the study of viral sequences**.

Regenerative Property

As per duality theorem, any arbitrary sequence x(n) of length $N=2^n$ is said to have dual y(n) if and only of its CPI is an even number and is divisible by N/2. In other words, x(n) and y(n) form a dual pair. It was observed that the sequences |x(0)~y(0)|, |x(1)~y(1)|, |x(2)~y(2)|, ……, |x(N-2)~y(N-2)|, |x(N-1)~y(N-1)| also satisfies duality theorem and is eligible to form a dual pair with yet another sequence. For example let us consider a sequence x(n)=3,1,3,3. This satisfies duality theorem, since its CPI is 10 and it is divisible by N/2 that is, 2 yielding the value 5. Now one can obtain the dual sequence y(n)=2,4,2,2 by subtracting each element of x(n) from 5. Now x(n) and y(n) form a pair and yield their '**first generation sequence**', say $x_1(n)$=1,3,1,1, which is obtained by finding the point wise difference between the parent sequences x(n) and y(n). One can easily verify that $x_1(n)$ also satisfies duality theorem and forms a dual pair with $y_1(n)$=2,0,2,2. This property is termed here as 'regenerative property'. Another important observation follows $x_1(n)$ forms a dual pair with $y_1(n)$ but this pair does not produce an offspring sequence as it was done by the parent pair < x(n), y(n)>. This could be easily verified by the fact that the point wise difference between $x_1(n)$ and $y_1(n)$ yields $x_1(n)$ only. Hence for brevity we call the sequences $x_1(n)$ and $y_1(n)$ as a **'sterile pair'**. This property exhibits the possibility of further research on generative and sterile pairs. **This property is also potentially used in the study of viral sequences.**

Scalar Property

Let x(n) be a number sequence and λ be a scalar. Then the RT of λx(n) will be λX(k), where X(k) is the RT of x(n). For example, let us consider a sequence x(n)=1,3,1,2 and a scalar λ of value 2. Now the RT X(k) of x(n) is 7,3,1,1. The RT of λx(n)=2,6,2,4 is 14,6,2,2 which is nothing but λX(k).

Linearity Property

In general, RT does not satisfy the linearity property for all sequences. It was observed that for a pair x(n) and y(n) which are number sequences either in the increasing order or in the decreasing order, the linearity property works. That is, for λx(n)+my(n) where λ and m are scalars and x(n) and y(n) are two number sequences either in the increasing or decreasing order, the RT will be

$\lambda X(k)+mY(k)$, where X(k) and Y(k) are respectively the RTs of x(n) and y(n). A characterization theorem is yet to be established for categorizing pairs of sequences, which would satisfy linearity property.

Linear Pair forming Property

Two sequences x(n) and y(n) are said to form a linear pair when X(k)+Y(k) is the RT of x(n)+y(n), where X(k) and Y(k) are the RTs of x(n) and y(n) respectively. The symbol '+' denotes the point wise addition of two sequences. As outlined earlier, pairs of sequences consisting of increasing numbers of decreasing numbers only form linear pair. In general, arbitrary sequences need not form linear pairs. Consequently, RT could be viewed as a '**nonlinear transform**'. However, higher order RT spectra do form linear pairs, and this has been identified as a very useful property for pattern recognition purposes. For example, let us consider two arbitrary random sequences x(n) = 8, 2, 2, 1, 6, 2, 6, 1, 2, 3, 2, 9, 2, 5, 5, 4 and y(n) = 4, 5, 3, 1, 0, 1, 4, 6, 6, 8, 2, 0, 7, 9, 0, 5. Now, the pointwise summation of x(n) and y(n) is denoted as z(n) = 12, 7, 5, 2, 6, 3, 10, 7, 8, 11, 4, 0, 9, 14, 5, 9. Let $X_1(k)$ = 60, 6, 6, 0, 16, 2, 8, 2, 26, 4, 16, 2, 10, 4, 4, 2, $Y_1(k)$ = 61, 9, 19, 3, 17, 9, 9, 5, 27, 1, 13, 5, 13, 3, 7, 3 and $Z_1(k)$ = 112, 6, 28, 6, 26, 4, 14, 12, 32, 6, 12, 10, 12, 2, 10, 4 be the RT spectra of x(n), y(n) and z(n) respectively. In order to check whether RT satisfies linearity property, one should examine whether $Z_1(k) = X_1(k) + Y_1(k)$. It is observed $X_1(k) + Y_1(k)$ = 121, 15, 25, 3, 33, 11, 17, 7, 53, 5, 29, 7, 23, 7, 11, 5 and it is not equal to $Z_1(k)$. This means that the spectrum of sum is not equal to sum of spectra. Hence, RT does not satisfy linearity property. In this context, the notion of higher order RT spectra is proposed here with the idea of examining whether the second order RT satisfies linearity property or not. With reference to the example, the second order RT spectra are obtained by treating $X_1(k)$ and $Y_1(k)$ as inputs and RT applied to them. Let them be $X_2(k)$ and $Y_2(k)$. Let $Z_2(k)$ be the second order RT of the sum $X_2(k)+Y_2(k)$. In this second stage of computation, $X_2(k)$ = 168, 124, 88, 68, 76, 64, 52, 48, 60, 48, 28, 24, 36, 32, 24, 20 and $Y_2(k)$ = 204, 128, 76, 64, 76, 72, 44, 40, 64, 28, 40, 20, 36, 32, 28, 24. Now, $X_2(k)+Y_2(k)$ = 372, 252, 164, 132, 152, 136, 96, 88, 124, 76, 68, 44, 72, 64, 52, 44 and the RT of the sum $X_2(k)+Y_2(k)$ is $Z_2(k)$ = 372, 252, 164, 132, 148, 140, 92, 92, 108, 68, 76, 60, 68, 52, 60, 52. It is observed that $X_2(k)+Y_2(k)$ is not equal to $Z_2(k)$. So, second order RT does not satisfy linearity property. One could proceed with higher order RT spectra and check whether any of the higher order RT satisfies linearity property. The sequence x(n) and its higher order RT spectra are given in Table 1.

Table 1: **Input data x(n) and its higher order Rajan Transform Spectra**

x(n)	$X_1(k)$	$X_2(k)$	$X_3(k)$	$X_4(k)$	$X_5(k)$	$X_6(k)$	$X_7(k)$	$X_8(k)$	$X_9(k)$
8	60	168	960	2688	15360	43008	245760	688128	3932160
2	6	124	104	1984	1664	31744	26624	507904	425984
2	6	88	256	1408	4096	22528	65536	360448	1048576
1	0	68	40	1088	640	17408	10240	278528	163840
6	16	76	256	1216	4096	19456	65536	311296	1048576
2	2	64	56	1024	896	16384	14336	262144	229376
6	8	52	128	832	2048	13312	32768	212992	524288
1	2	48	24	768	384	12288	6144	196608	98304
2	26	60	416	960	6656	15360	106496	245760	1703936
3	4	48	56	768	896	12288	14336	196608	229376
2	16	28	96	448	1536	7168	24576	114688	393216
9	2	24	24	384	384	6144	6144	98304	98304
2	10	36	160	576	2560	9216	40960	147456	655360
5	4	32	40	512	640	8192	10240	131072	163840
5	4	24	64	384	1024	6144	16384	98304	262144
4	2	20	8	320	128	5120	2048	81920	32768

The sequence y(n) and its higher order RT spectra are given below.

Table 2: **Input data y(n) and its higher order Rajan Transform Spectra**

y(n)	$Y_1(k)$	$Y_2(k)$	$Y_3(k)$	$Y_4(k)$	$Y_5(k)$	$Y_6(k)$	$Y_7(k)$	$Y_8(k)$	$Y_9(k)$
4	61	204	976	3264	15616	52224	249856	835584	3997696
5	9	128	160	2048	2560	32768	40960	524288	655360
3	19	76	304	1216	4864	19456	77824	311296	1245184
1	3	64	80	1024	1280	16384	20480	262144	327680
0	17	76	272	1216	4352	19456	69632	311296	1114112
1	9	72	128	1152	2048	18432	32768	294912	524288
4	9	44	144	704	2304	11264	36864	180224	589824
6	5	40	80	640	1280	10240	20480	163840	327680
6	27	64	432	1024	6912	16384	110592	262144	1769472
8	1	28	32	448	512	7168	8192	114688	131072
2	13	40	208	640	3328	10240	53248	163840	851968
0	5	20	48	320	768	5120	12288	81920	196608
7	13	36	208	576	3328	9216	53248	147456	851968
9	3	32	32	512	512	8192	8192	131072	131072
0	7	28	112	448	1792	7168	28672	114688	458752
5	3	24	48	384	768	6144	12288	98304	196608

The sequence z(n) and its higher order RT spectra are given below.

Table 3: **Input data z(n) and its higher order Rajan Transform Spectra**

z(n)	$Z_1(k)$	$Z_2(k)$	$Z_3(k)$	$Z_4(k)$	$Z_5(k)$	$Z_6(k)$	$Z_7(k)$	$Z_8(k)$	$Z_9(k)$
12	112	372	1936	5952	30976	95232	495616	1523712	7929856
7	6	252	264	4032	4224	64512	67584	1032192	1081344
5	28	164	560	2624	8960	41984	143360	671744	2293760
2	6	132	120	2112	1920	33792	30720	540672	491520
6	26	148	528	2432	8448	38912	135168	622592	2162688
3	4	140	184	2176	2944	34816	47104	557056	753664
10	14	92	272	1536	4352	24576	69632	393216	1114112
7	12	92	104	1408	1664	22528	26624	360448	425984
8	32	108	848	1984	13568	31744	217088	507904	3473408
11	6	68	88	1216	1408	19456	22528	311296	360448
4	12	76	304	1088	4864	17408	77824	278528	1245184
0	10	60	72	704	1152	11264	18432	180224	294912
9	12	68	368	1152	5888	18432	94208	294912	1507328
14	2	52	72	1024	1152	16384	18432	262144	294912
5	10	60	176	832	2816	13312	45056	212992	720896
9	4	52	56	704	896	11264	14336	180224	229376

The sequence z(n) and its higher order RT spectra are given below.

Table 4: **Pointwise addtition of x(n) and y(n) and pointwise addition of their higher order Rajan Transform Spectra**

x(n) + y(n)	$X_1(k)$ + $Y_1(k)$	$X_2(k)$ + $Y_2(k)$	$X_3(k)$ + $Y_3(k)$	$X_4(k)$ + $Y_4(k)$	$X_5(k)$ + $Y_5(k)$	$X_6(k)$ + $Y_6(k)$	$X_7(k)$ + $Y_7(k)$	$X_8(k)$ + $Y_8(k)$	$X_9(k)$ + $Y_9(k)$
12	121	372	1936	5952	30976	95232	495616	1523712	7929856
7	15	252	264	4032	4224	64512	67584	1032192	1081344
5	25	164	560	2624	8960	41984	143360	671744	2293760
2	3	132	120	2112	1920	33792	30720	540672	491520
6	33	152	528	2432	8448	38912	135168	622592	2162688
3	11	136	184	2176	2944	34816	47104	557056	753664
10	17	96	272	1536	4352	24576	69632	393216	1114112
7	7	88	104	1408	1664	22528	26624	360448	425984
8	53	124	848	1984	13568	31744	217088	507904	3473408
11	5	76	88	1216	1408	19456	22528	311296	360448
4	29	68	304	1088	4864	17408	77824	278528	1245184
0	7	44	72	704	1152	11264	18432	180224	294912
9	23	72	368	1152	5888	18432	94208	294912	1507328
14	7	64	72	1024	1152	16384	18432	262144	294912
5	11	52	176	832	2816	95232	45056	212992	720896
9	5	44	56	704	896	64512	14336	180224	229376

Observations

1. Ref. Table 1. The following hold:
   $X_3(k) = 16X_1(k)$; $X_5(k) = 16X_3(k)$; $X_7(k) = 16X_5(k)$; $X_9(k) = 16X_7(k)$.
   $X_4(k) = 16X_2(k)$; $X_6(k) = 16X_4(k)$; $X_8(k) = 16X_6(k)$.
2. Ref. Table 2. The following hold:
   $Y_3(k) = 16Y_1(k)$; $Y_5(k) = 16Y_3(k)$; $Y_7(k) = 16Y_5(k)$; $Y_9(k) = 16Y_7(k)$.
   $Y_4(k) = 16Y_2(k)$; $Y_6(k) = 16Y_4(k)$; $Y_8(k) = 16Y_6(k)$.
3. Ref. Tables 3 and 4. The following hold:
   $Z_3(k) = X_3(k) + Y_3(k)$; $Z_4(k) = X_4(k) + Y_4(k)$; $Z_5(k) = X_5(k) + Y_5(k)$; $Z_6(k) = X_6(k) + Y_6(k)$;
   $Z_7(k) = X_7(k) + Y_7(k)$; $Z_8(k) = X_8(k) + Y_8(k)$; $Z_9(k) = X_9(k) + Y_9(k)$.
4. Given two sequences consisting of random numbers, Rajan Transform need not satisfy linearity property at the first level. On the other hand, repeated application of Rajan Transform yields higher order spectra highlighting the point that the sequences are regulated and they do not change randomly. In fact, n^{th} order RT spectral components are $(n-2)^{nd}$ order RT spectral components multiplied by N, where N is the length of the sequences. Thus, one can precisely evaluate higher order spectra without applying Rajan Transform repeatedly. For example, with reference to Table 1, one can directly evaluate $X_9(k)$ from $X_1(k)$ as 65,536 $X_1(k)$, that is $16^4X_1(k)$. Similarly, with reference to Table 2, one can directly evaluate $Y_9(k)$ as $65{,}536Y_1(k)$, that is $16^4Y_1(k)$. To generalize, let us assume two sequences of equal length of N. Let us also assume that at the r^{th} stage the sequences form linear pair, that is RT satisfies linearity property for the pair. This means that at the r^{th} stage, the pair enters into a '**Correlation Agreement**' (CA)', and we refere to this stage as '**Correlation Agreement Stage' (CAS)**' and the stage number as '**Correlation Agreement Stage Number**' (CASN)'

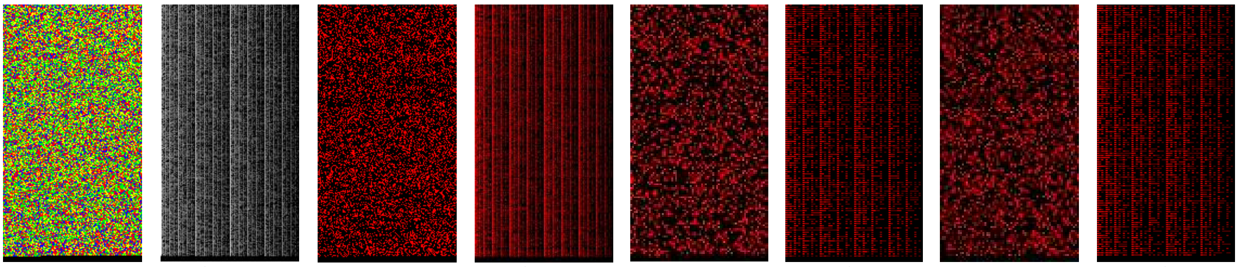
Inference

1. Higher order RT spectra can be evaluated for an N-point sequence right from its first order RT spectrum.
2. Two N-point sequences could be seen to form ‘Correlation Agreement’ at a particular stage of evaluation of higher order RT spectra. In the above example, the CASN is 3.
3. If two arbitrary N-point sequences are forced to change their contents as per rules of higher order RT computation, then the two sequences will get into ‘Corrleation Agreement (CA)’.
4. The question that arises here is whether one can apply this notion to randomly mutating virus genome sequences and possibly regulate the mutation process. If this conjecture were realizable, then one would be able to predict futuristic mutations of any virus, based on pairwise interactions among them in a host. A synthesized mutating RNA string with ACE2 receptacles when injected in a human body might force pairs of in vivo viruses equipped with stem proteins to get into self regulation and then neutralized by a vaccine.

The linear pair forming property of Rajan Transform is useful in getting the CASN for any pair of number sequences. This property coupled with ‘Regenerative Property’ might open up new avenues in devising methods for artificially synthesizing harmful RNA strains and their combating duals of harmless RNA strains. This is a hypothesis and it is subjected to fair scrutiny.

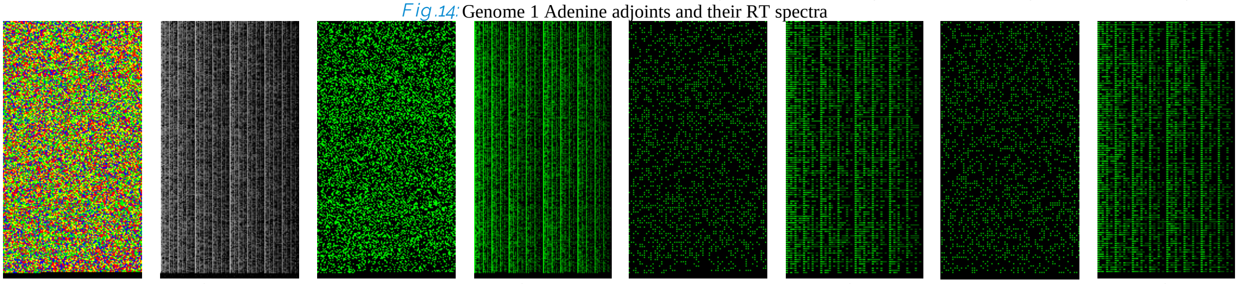
4.2 RAJAN TRANSFORM SPECTRA OF SARS-CORONA VIRUS GENOME SEQUENCES

Rajan Transform (RT) is applied to nine virus sequences (1) SARS coronavirus SZ16, complete genome, (2) SARS coronavirus ZS-C, complete genome, (3) Bat SARS coronavirus Rm1, complete genome, (4) Bat SARS-like coronavirus isolate Rs4084, complete genome, (5) Wuhan seafood market, pneumonia virus isolate Wuhan-Hu-1, complete genome, (6) Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete genome, (7) Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome, (8) Bat SARS-like coronavirus isolate bat-SL-CoVZC45, complete genome, (9) Bat SARS-like coronavirus isolate bat-SL-CoVZXC21, complete genome. These genome data are assumed to be genuine and analysis carried out on them.



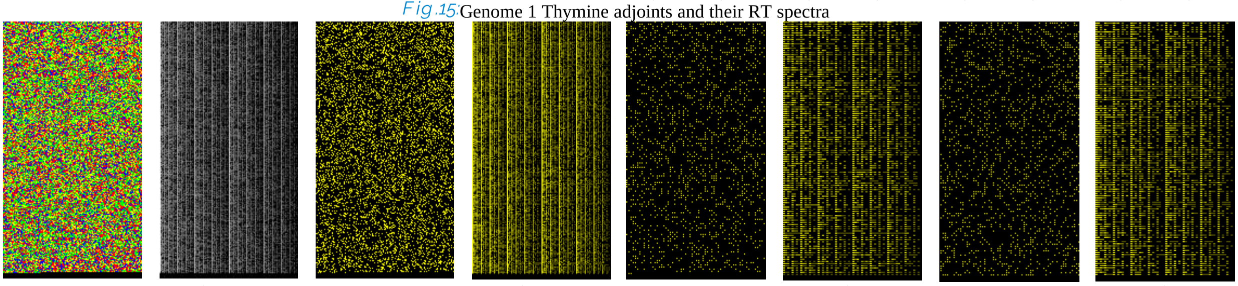
Genome 1: SARS corona virus SZ16
RT of Genome 1 SARS corona virus SZ16
Adenine Adjoint Array of Genome 1
RT of Adenine Adjoint Array of Genome 1
Even subsampled AdenineAdjoint
RT of even subsampled AdenineAdjoint
Odd subsampled AdenineAdjoint
RT of odd subsampled AdenineAdjoint

Fig.14: Genome 1 Adenine adjoints and their RT spectra



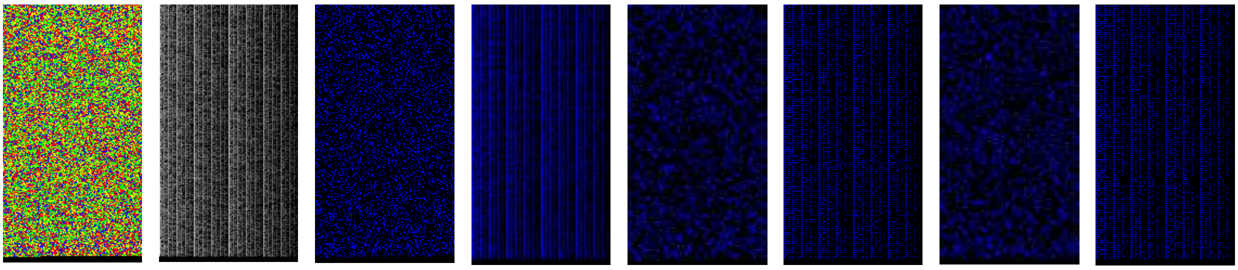
Genome 1: SARS corona virus SZ16
RT of Genome 1 SARS corona virus SZ16
Thymine Adjoint Array of Genome 1
RT of Thymine Adjoint Array of Genome 1
Even subsampled ThymineAdjoint
RT of even subsampled ThymineAdjoint
Odd subsampled ThymineAdjoint
RT of odd subsampled ThymineAdjoint

Fig.15: Genome 1 Thymine adjoints and their RT spectra



Genome 1: SARS corona virus SZ16
RT of Genome 1 SARS corona virus SZ16
Guanine Adjoint Array of Genome 1
RT of Guanine Adjoint Array of Genome 1
Even subsampled GuanineAdjoint
RT of even subsampled GuanneAdjoint
Odd subsampled GuanneAdjoint
RT of odd subsampled GuanineAdjoint

Fig.16: Genome 1 Guanine adjoints and their RT spectra



Genome 1: SARS corona virus SZ16

RT of Genome 1 SARS corona virus SZ16

Cytosine Adjoint Array of Genome 1

RT of Cytosine Adjoint Array of Genome 1

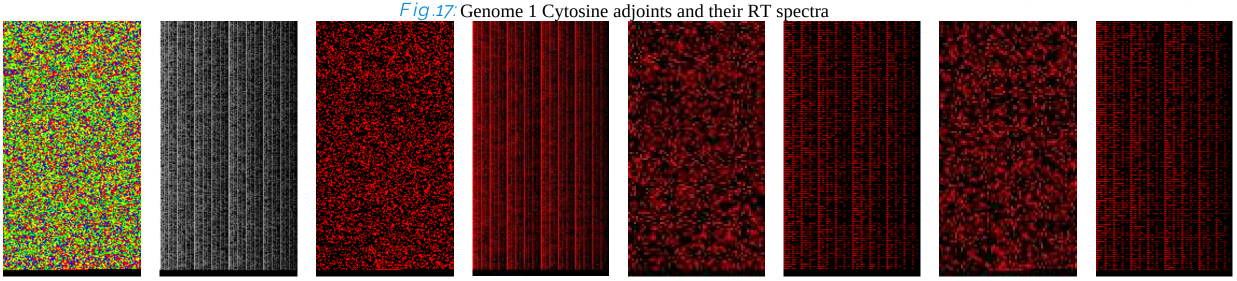
Even subsampled CytosineAdjoint

RT of even subsampled CytosineAdjoint

Odd subsampled CytosineAdjoint

RT of odd subsampled CytosineAdjoint

Fig.17: Genome 1 Cytosine adjoints and their RT spectra



Genome 2 SARS corona virus ZS-C

RT of Genome 2 SARS corona virus ZS-C

Adenine Adjoint Array of Genome 2

RT of Adenine Adjoint Array of Genome 2

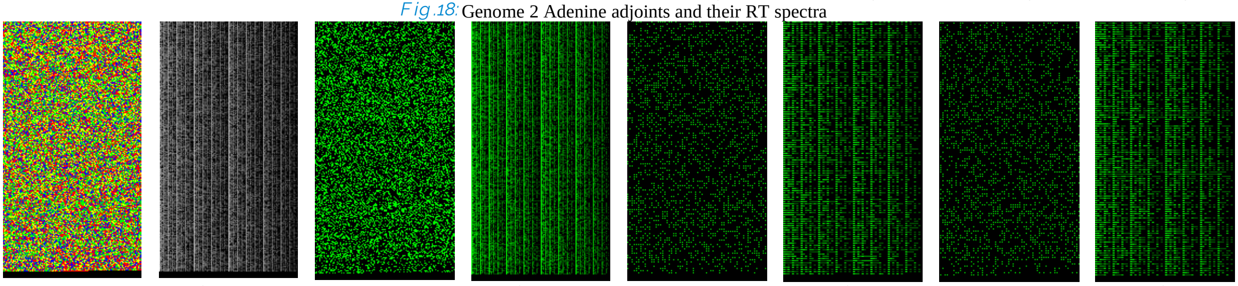
Even subsampled AdenineAdjoint

RT of even subsampled AdenineAdjoint

Odd subsampled AdenineAdjoint

RT of odd subsampled AdenineAdjoint

Fig.18: Genome 2 Adenine adjoints and their RT spectra



Genome 2 SARS corona virus ZS-C

RT of Genome 2 SARS corona virus ZS-C

Thymine Adjoint Array of Genome 2

RT of Thymine Adjoint Array of Genome 2

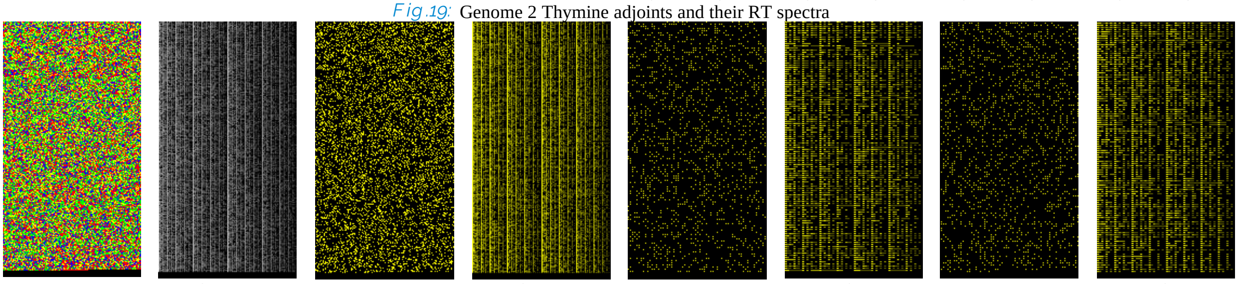
Even subsampled ThymineAdjoint

RT of even subsampled ThymineAdjoint

Odd subsampled ThymineAdjoint

RT of odd subsampled ThymineAdjoint

Fig.19: Genome 2 Thymine adjoints and their RT spectra



Genome 2 SARS corona virus ZS-C

RT of Genome 2 SARS corona virus ZS-C

Guanine Adjoint Array of Genome 2

RT of Guanine Adjoint Array of Genome 2

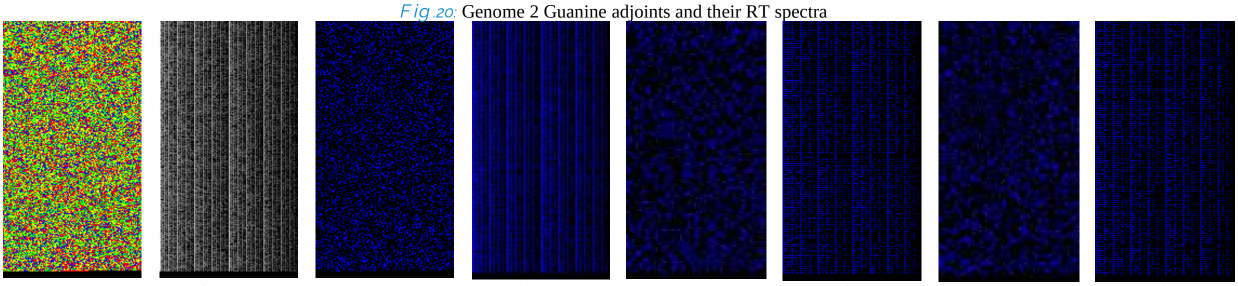
Even subsampled Guanine Adjoint

RT of even subsampled Guanine Adjoint

Odd subsampled Guanine Adjoint

RT of odd subsampled Guanine Adjoint

Fig.20: Genome 2 Guanine adjoints and their RT spectra



Genome 2 SARS corona virus ZS-C

RT of Genome 2 SARS corona virus ZS-C

Cytosine Adjoint Array of Genome 2

RT of Cytosine Adjoint Array of Genome 2

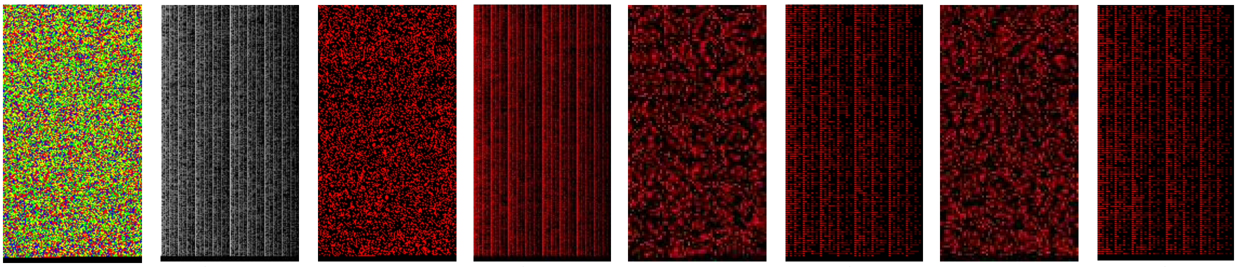
Even subsampled CytosineAdjoint

RT of even subsampled CytosineAdjoint

Odd subsampled CytosineAdjoint

RT of odd subsampled CytosineAdjoint

Fig.21: Genome 2 Cytosine adjoints and their RT spectra



Genome 3 Bat SARS corona virus Rm1

RT of Genome 3 SARS corona virus Rm1

Adenine Adjoint Array of Genome 3

RT of Adenine Adjoint Array of Genome 3

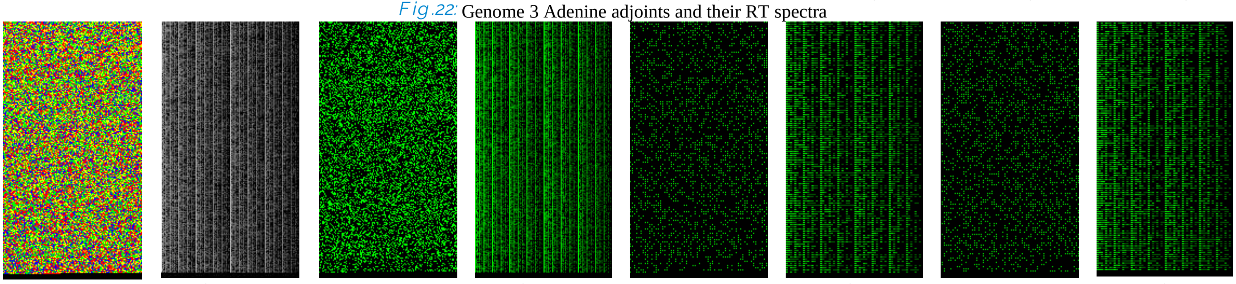
Even subsampled AdenineAdjoint

RT of even subsampled AdenineAdjoint

Odd subsampled AdenineAdjoint

RT of odd subsampled AdenineAdjoint

Fig.22: Genome 3 Adenine adjoints and their RT spectra



Genome 3 Bat SARS corona virus Rm1

RT of Genome 3 SARS corona virus Rm1

Thymine Adjoint Array of Genome 3

RT of Thymine Adjoint Array of Genome 3

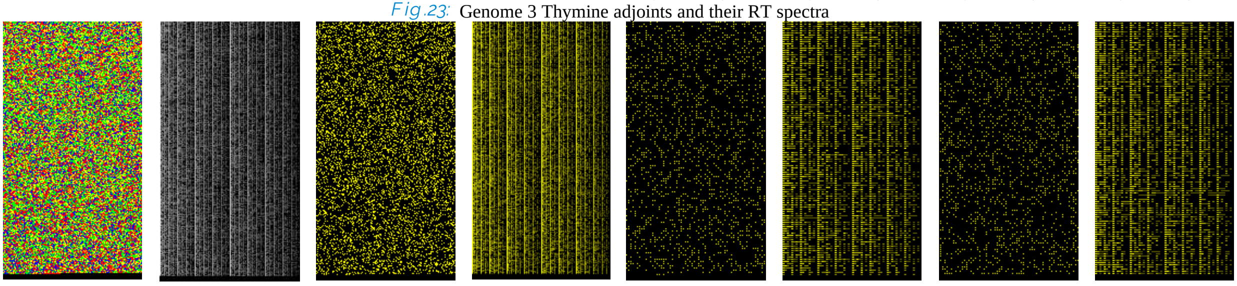
Even subsampled ThymineAdjoint

RT of even subsampled ThymineAdjoint

Odd subsampled ThymineAdjoint

RT of odd subsampled ThymineAdjoint

Fig.23: Genome 3 Thymine adjoints and their RT spectra



Genome 3 Bat SARS corona virus Rm1

RT of Genome 3 SARS corona virus Rm1

Guanine Adjoint Array of Genome 3

RT of Guanine Adjoint Array of Genome 3

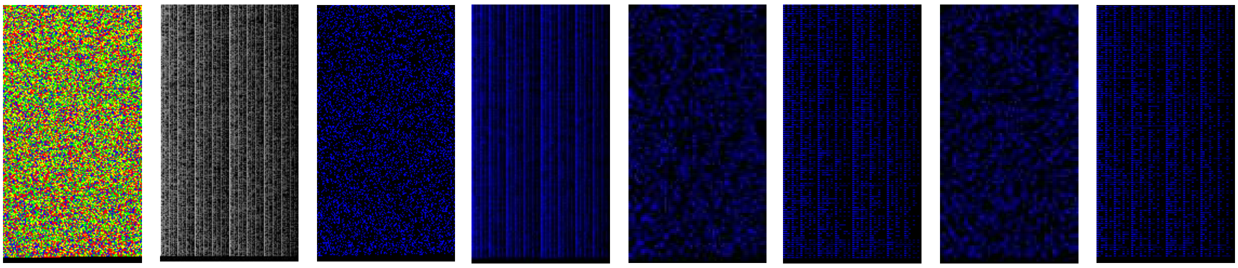
Even subsampled GuanineAdjoint

RT of even subsampled GuanineAdjoint

Odd subsampled GuanineAdjoint

RT of odd subsampled GuanineAdjoint

Fig.24: Genome 3 Guanine adjoints and their RT spectra



Genome 3 Bat SARS corona virus Rm1

RT of Genome 3 SARS corona virus Rm1

Cytosine Adjoint Array of Genome 3

RT of Cytosine Adjoint Array of Genome 3

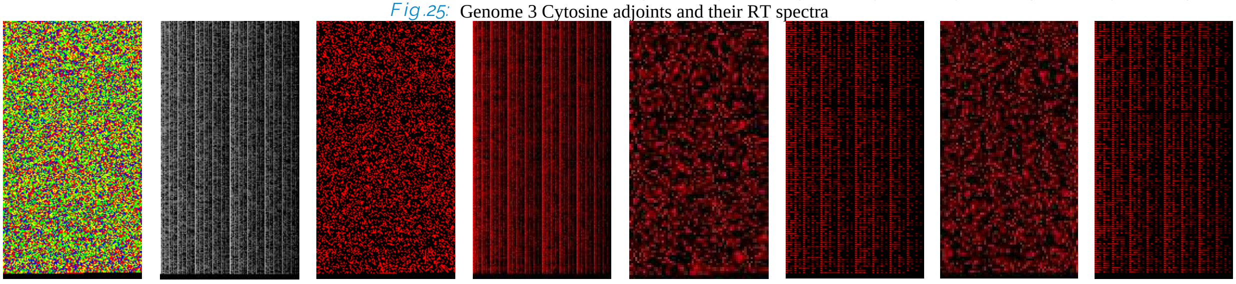
Even subsampled CytosineAdjoint

RT of even subsampled CytosineAdjoint

Odd subsampled CytosineAdjoint

RT of odd subsampled CytosineAdjoint

Fig.25: Genome 3 Cytosine adjoints and their RT spectra



Genome 4 Bat SARS corona virus Rs4084

RT of Genome 4 SARS corona virus Rs4084

Adenine Adjoint Array of Genome 4

RT of Adenine Adjoint Array of Genome 4

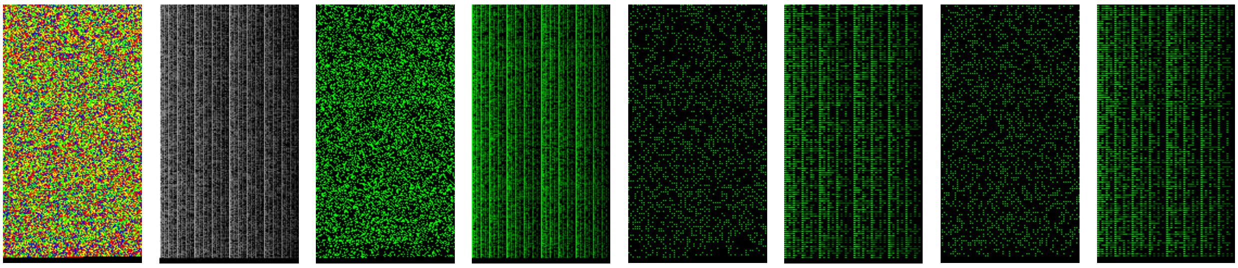
Even subsampled AdenineAdjoint

RT of even subsampled AdenineAdjoint

Odd subsampled AdenineAdjoint

RT of odd subsampled AdenineAdjoint

Fig.26: Genome 4 Adenine adjoints and their RT spectra



Genome 4 Bat SARS corona virus Rs4084

RT of Genome 4 SARS corona virus Rs4084

Thymine Adjoint Array of Genome 4

RT of Thymine Adjoint Array of Genome 4

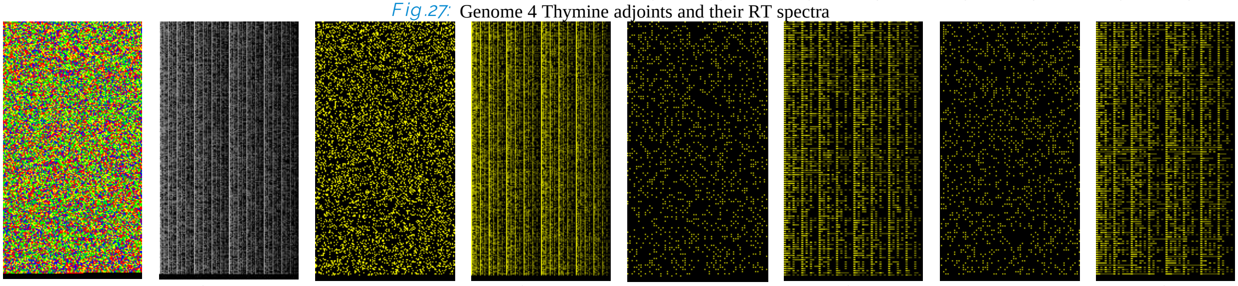
Even subsampled ThymineAdjoint

RT of even subsampled ThymineAdjoint

Odd subsampled ThymineAdjoint

RT of odd subsampled ThymineAdjoint

Fig.27: Genome 4 Thymine adjoints and their RT spectra



Genome 4 Bat SARS corona virus Rs4084

RT of Genome 4 SARS corona virus Rs4084

Guanine Adjoint Array of Genome 4

RT of Guanine Adjoint Array of Genome 4

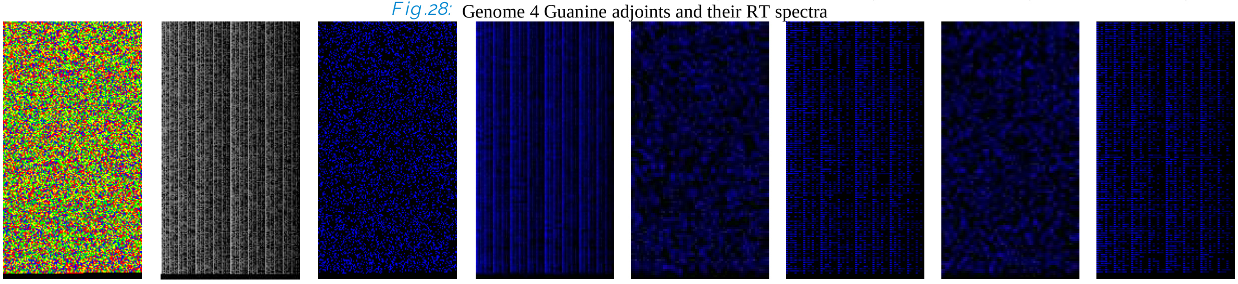
Even subsampled GuanineAdjoint

RT of even subsampled GuanineAdjoint

Odd subsampled GuanineAdjoint

RT of odd subsampled GuanineAdjoint

Fig.28: Genome 4 Guanine adjoints and their RT spectra



Genome 4 Bat SARS corona virus Rs4084

RT of Genome 4 SARS corona virus Rs4084

Cytosine Adjoint Array of Genome 4

RT of Cytosine Adjoint Array of Genome 4

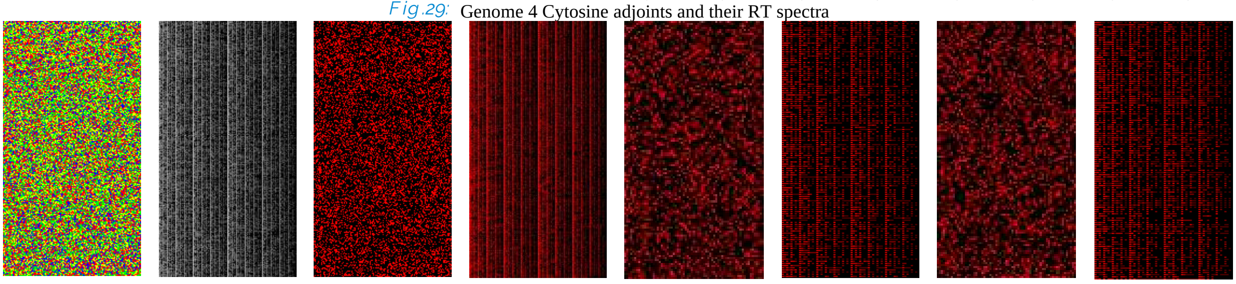
Even subsampled CytosineAdjoint

RT of even subsampled CytosineAdjoint

Odd subsampled CytosineAdjoint

RT of odd subsampled CytosineAdjoint

Fig.29: Genome 4 Cytosine adjoints and their RT spectra



Genome 5 Seafood Wuhan-Hu-1

RT of Genome 5 Seafood Wuhan-Hu-1

Adenine Adjoint Array of Genome 5

RT of Adenine Adjoint Array of Genome 5

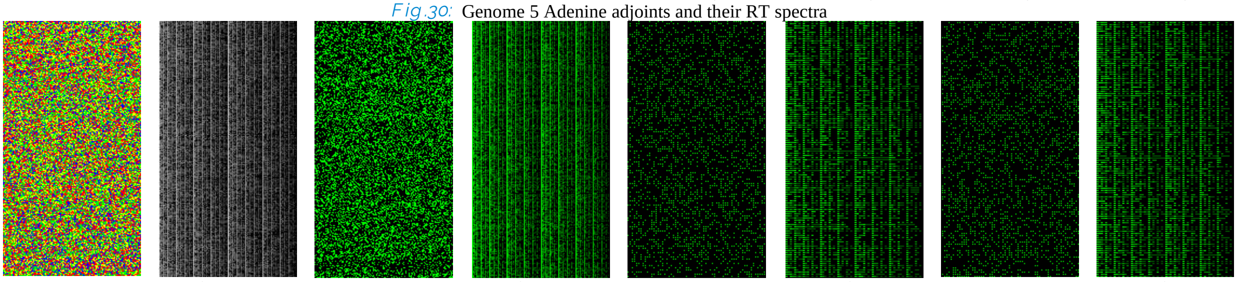
Even subsampled AdenineAdjoint

RT of even subsampled AdenineAdjoint

Odd subsampled AdenineAdjoint

RT of odd subsampled AdenineAdjoint

Fig.30: Genome 5 Adenine adjoints and their RT spectra



Genome 5 Seafood Wuhan-Hu-1

RT of Genome 5 Seafood Wuhan-Hu-1

Thymine Adjoint Array of Genome 5

RT of Thymine Adjoint Array of Genome 5

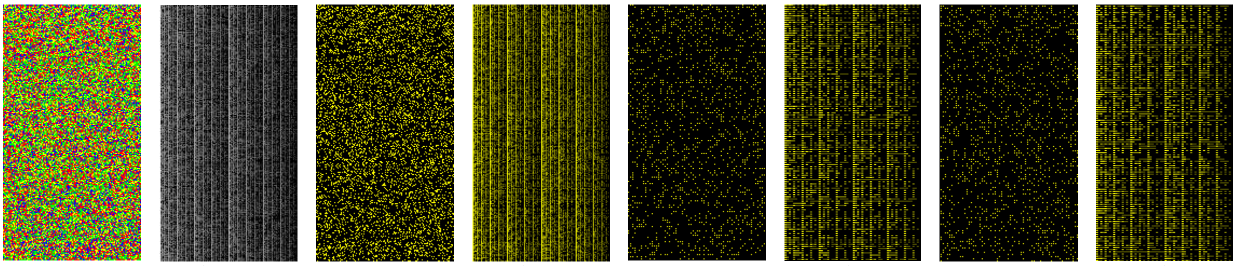
Even subsampled ThymineAdjoint

RT of even subsampled ThymineAdjoint

Odd subsampled ThymineAdjoint

RT of odd subsampled ThymineAdjoint

Fig.31: Genome 5 Thymine adjoints and their RT spectra



Genome 5 Seafood Wuhan-Hu-1

RT of Genome 5 Seafood Wuhan-Hu-1

Guanine Adjoint Array of Genome 5

RT of Guanine Adjoint Array of Genome 5

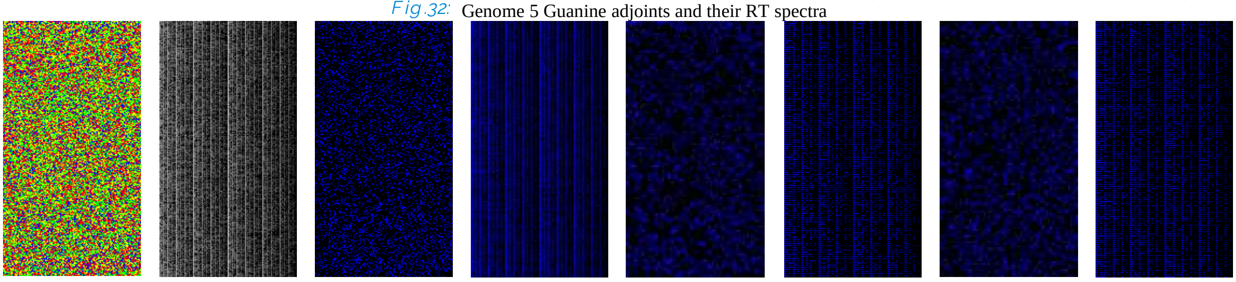
Even subsampled GuanineAdjoint

RT of even subsampled GuanineAdjoint

Odd subsampled GuanineAdjoint

RT of odd subsampled GuanineAdjoint

Fig.32: Genome 5 Guanine adjoints and their RT spectra



Genome 5 Seafood Wuhan-Hu-1

RT of Genome 5 Seafood Wuhan-Hu-1

Cytosine Adjoint Array of Genome 5

RT of Cytosine Adjoint Array of Genome 5

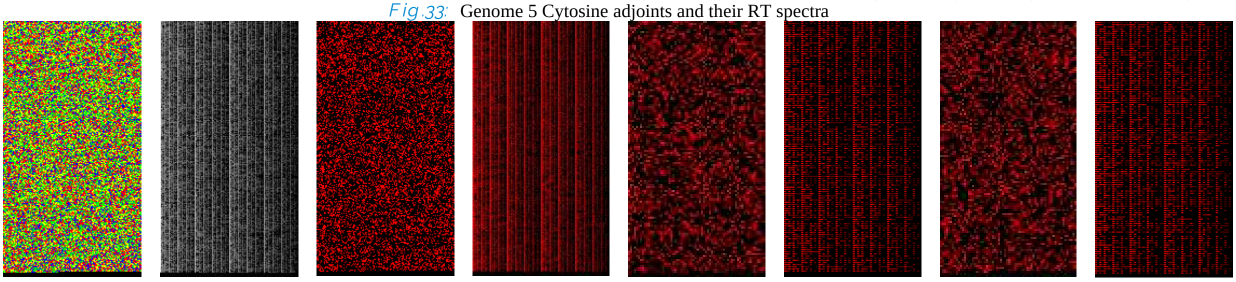
Even subsampled CytosineAdjoint

RT of even subsampled CytosineAdjoint

Odd subsampled CytosineAdjoint

RT of odd subsampled CytosineAdjoint

Fig.33: Genome 5 Cytosine adjoints and their RT spectra



Genome 6 Seafood Wuhan-Hu-1 Revised

RT of Genome 6 Seafood Wuhan-Hu-1 Revised

Adenine Adjoint Array of Genome 6

RT of Adenine Adjoint Array of Genome 6

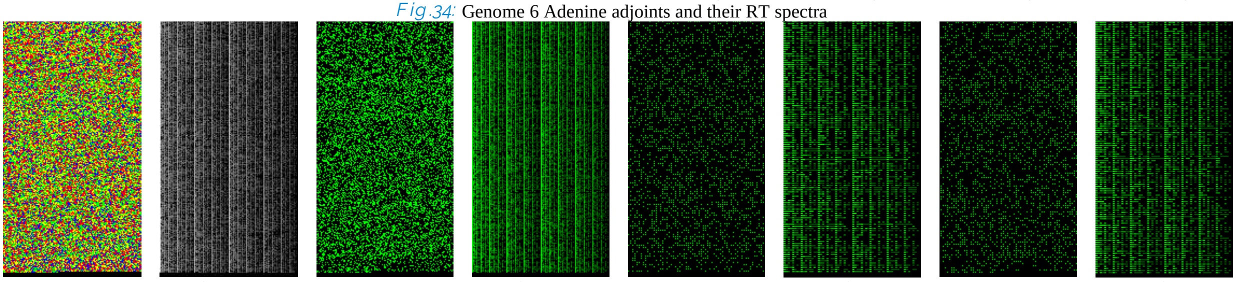
Even subsampled AdenineAdjoint

RT of even subsampled AdenineAdjoint

Odd subsampled AdenineAdjoint

RT of odd subsampled AdenineAdjoint

Fig.34: Genome 6 Adenine adjoints and their RT spectra



Genome 6 Seafood Wuhan-Hu-1 Revised

RT of Genome 6 Seafood Wuhan-Hu-1 Revised

Thymine Adjoint Array of Genome 6

RT of Thymine Adjoint Array of Genome 6

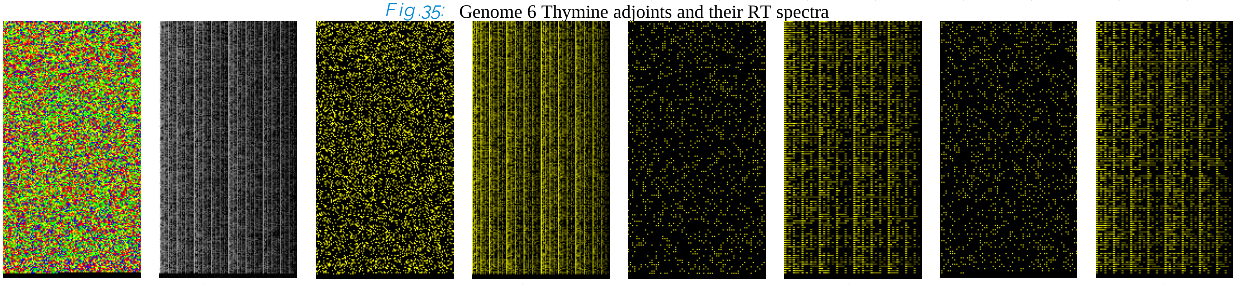
Even subsampled ThymineAdjoint

RT of even subsampled ThymineAdjoint

Odd subsampled ThymineAdjoint

RT of odd subsampled ThymineAdjoint

Fig.35: Genome 6 Thymine adjoints and their RT spectra



Genome 6 Seafood Wuhan-Hu-1 Revised

RT of Genome 6 Seafood Wuhan-Hu-1 Revised

Guanine Adjoint Array of Genome 6

RT of Guanine Adjoint Array of Genome 6

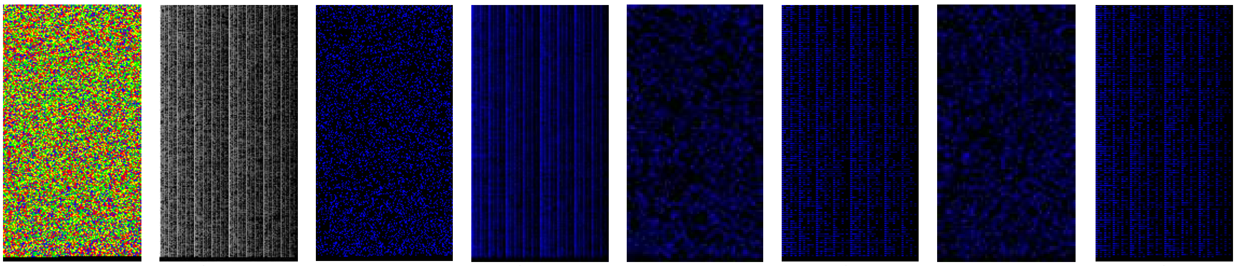
Even subsampled GuanineAdjoint

RT of even subsampled GuanineAdjoint

Odd subsampled GuanineAdjoint

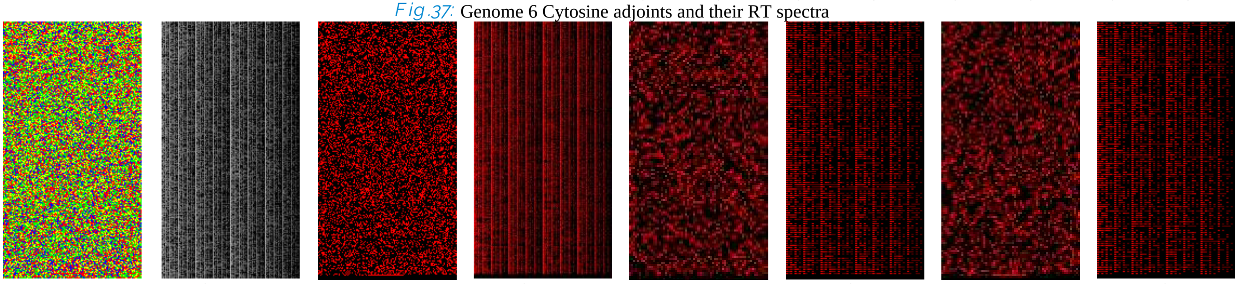
RT of odd subsampled GuanineAdjoint

Fig.36: Genome 6 Guanine adjoints and their RT spectra



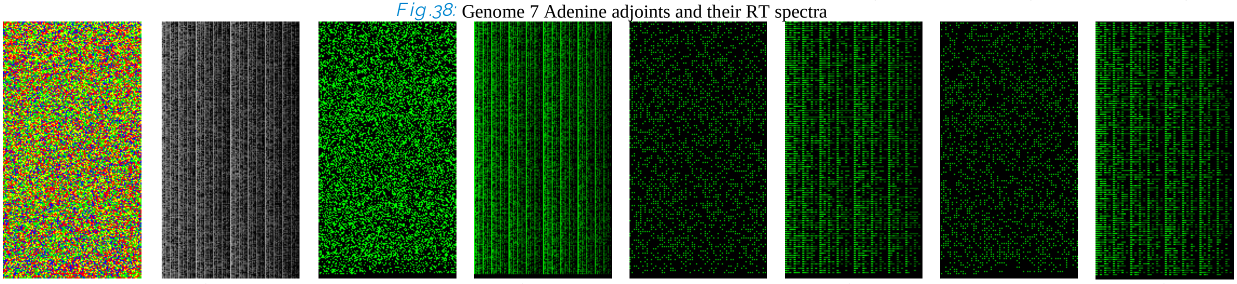
Genome 6 Seafood Wuhan-Hu-1 Revised
RT of Genome 6 Seafood Wuhan-Hu-1 Revised
Cytosine Adjoint Array of Genome 6
RT of Cytosine Adjoint Array of Genome 6
Even subsampled CytosineAdjoint
RT of even subsampled CytosineAdjoint
Odd subsampled CytosineAdjoint
RT of odd subsampled CytosineAdjoint

Fig.37: Genome 6 Cytosine adjoints and their RT spectra



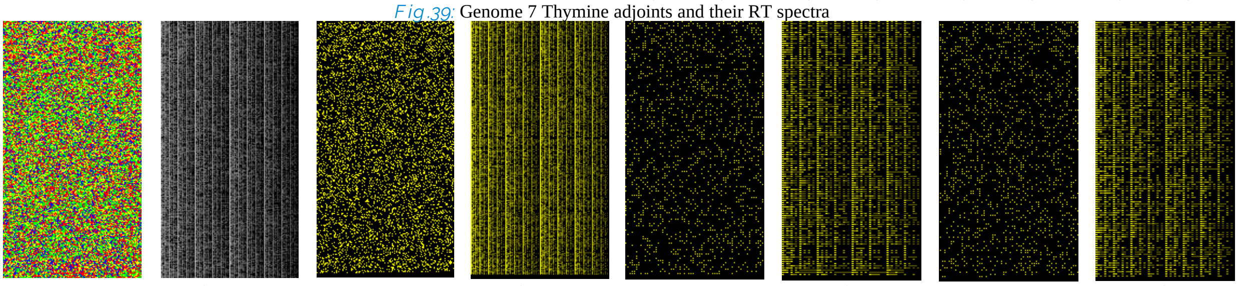
Genome 7 SARS-Corona Wuhan-Hu-1
RT of Genome 7 SARS-Corona Wuhan-Hu-1
Adenine Adjoint Array of Genome 7
RT of Adenine Adjoint Array of Genome 7
Even subsampled AdenineAdjoint
RT of even subsampled AdenineAdjoint
Odd subsampled AdenineAdjoint
RT of odd subsampled AdenineAdjoint

Fig.38: Genome 7 Adenine adjoints and their RT spectra



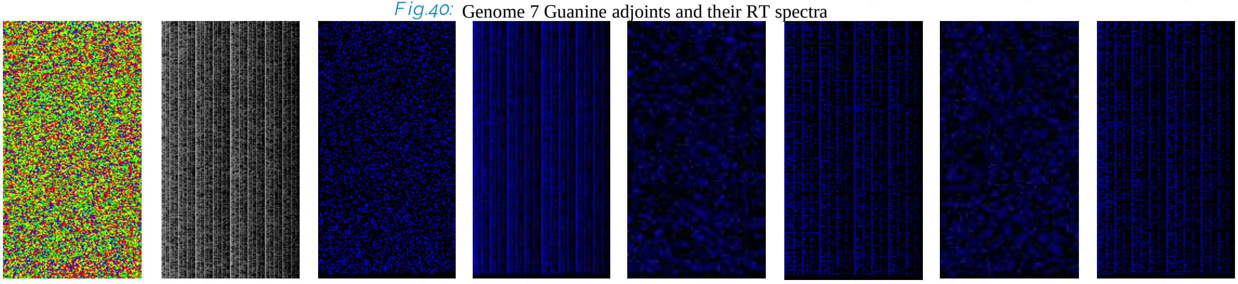
Genome 7 SARS-Corona Wuhan-Hu-1
RT of Genome 7 SARS-Corona Wuhan-Hu-1
Thymine Adjoint Array of Genome 7
RT of Thymine Adjoint Array of Genome 7
Even subsampled ThymineAdjoint
RT of even subsampled ThymineAdjoint
Odd subsampled ThymineAdjoint
RT of odd subsampled ThymineAdjoint

Fig.39: Genome 7 Thymine adjoints and their RT spectra



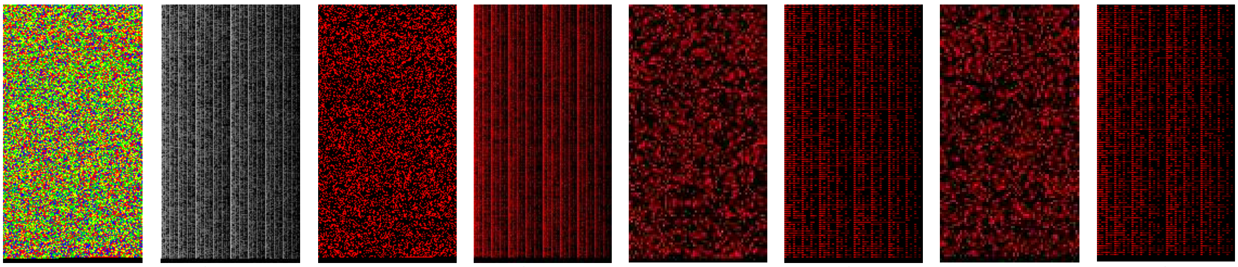
Genome 7 SARS-Corona Wuhan-Hu-1
RT of Genome 7 SARS-Corona Wuhan-Hu-1
Guanine Adjoint Array of Genome 7
RT of Guanine Adjoint Array of Genome 7
Even subsampled GuanineAdjoint
RT of even subsampled GuanineAdjoint
Odd subsampled GuanineAdjoint
RT of odd subsampled GuanineAdjoint

Fig.40: Genome 7 Guanine adjoints and their RT spectra



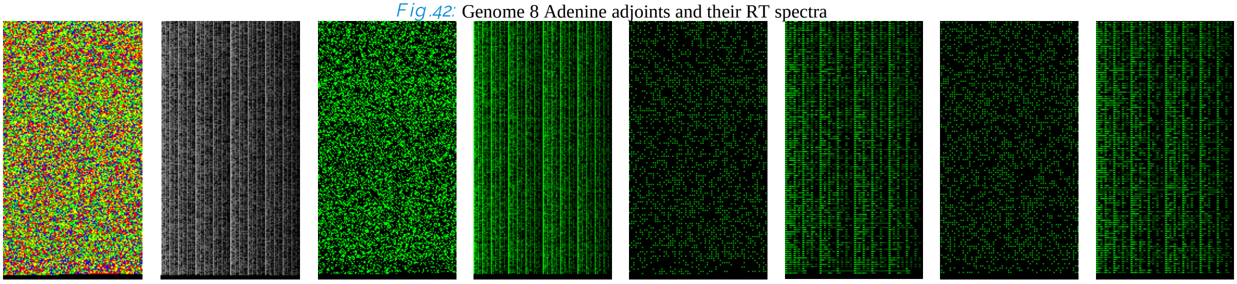
Genome 7 SARS-Corona Wuhan-Hu-1
RT of Genome 7 SARS-Corona Wuhan-Hu-1
Cytosine Adjoint Array of Genome 7
RT of Cytosine Adjoint Array of Genome 7
Even subsampled CytosineAdjoint
RT of even subsampled CytosineAdjoint
Odd subsampled CytosineAdjoint
RT of odd subsampled CytosineAdjoint

Fig.41: Genome 7 Cytosine adjoints and their RT spectra



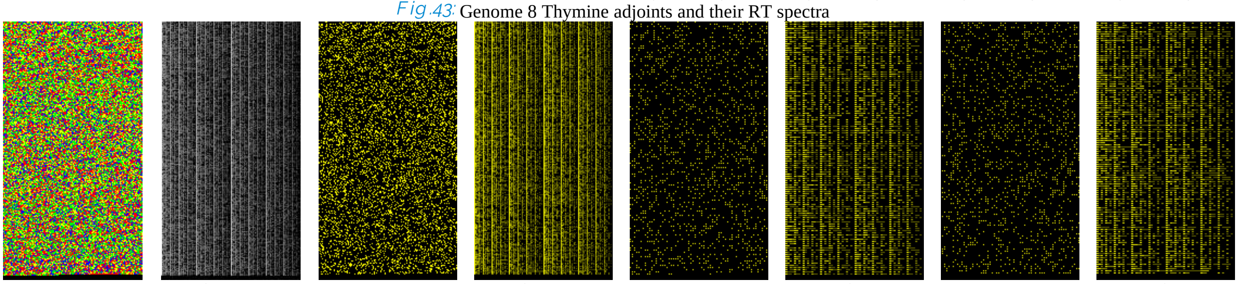
Genome 8 Bat-SL-CoVZC45
RT of Genome 8 Bat-SL-CoVZC45
Adenine Adjoint Array of Genome 8
RT of Adenine Adjoint Array of Genome 8
Even subsampled AdenineAdjoint
RT of even subsampled AdenineAdjoint
Odd subsampled AdenineAdjoint
RT of odd subsampled AdenineAdjoint

Fig.42: Genome 8 Adenine adjoints and their RT spectra



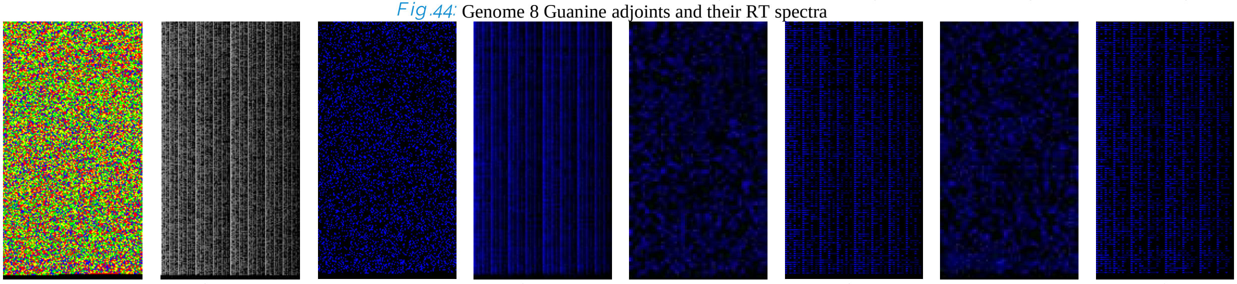
Genome 8 Bat-SL-CoVZC45
RT of Genome 8 Bat-SL-CoVZC45
Thymine Adjoint Array of Genome 8
RT of Thymine Adjoint Array of Genome 8
Even subsampled ThymineAdjoint
RT of even subsampled ThymineAdjoint
Odd subsampled ThymineAdjoint
RT of odd subsampled ThymineAdjoint

Fig.43: Genome 8 Thymine adjoints and their RT spectra



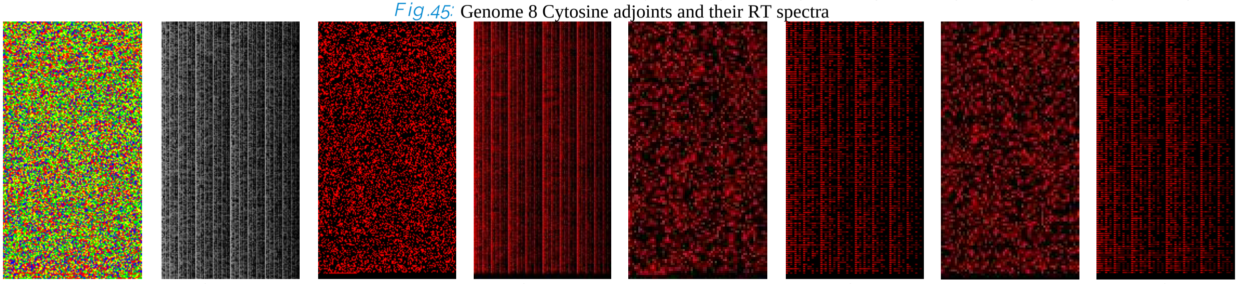
Genome 8 Bat-SL-CoVZC45
RT of Genome 8 Bat-SL-CoVZC45
Guanine Adjoint Array of Genome 8
RT of Guanine Adjoint Array of Genome 8
Even subsampled GuanineAdjoint
RT of even subsampled GuanineAdjoint
Odd subsampled GuanineAdjoint
RT of odd subsampled GuanineAdjoint

Fig.44: Genome 8 Guanine adjoints and their RT spectra



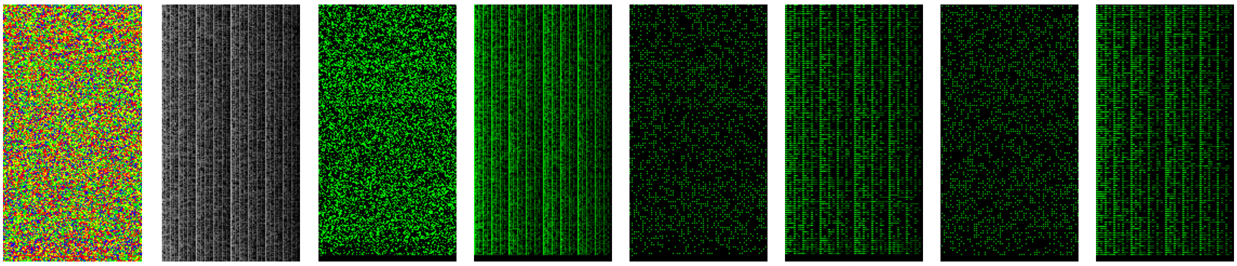
Genome 8 Bat-SL-CoVZC45
RT of Genome 8 Bat-SL-CoVZC45
Cytosine Adjoint Array of Genome 8
RT of Cytosine Adjoint Array of Genome 8
Even subsampled CytosineAdjoint
RT of even subsampled CytosineAdjoint
Odd subsampled CytosineAdjoint
RT of odd subsampled CytosineAdjoint

Fig.45: Genome 8 Cytosine adjoints and their RT spectra



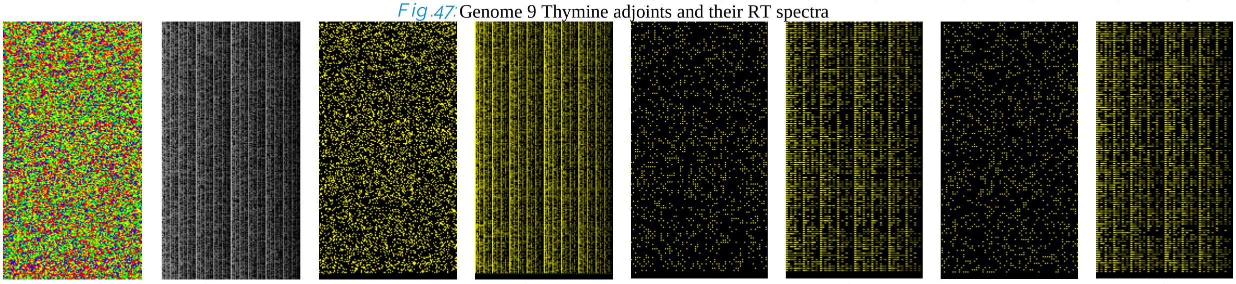
Genome 8 Bat-SL-CoVZXC21
RT of Genome 8 Bat-SL-CoVZXC21
Adenine Adjoint Array of Genome 9
RT of Adenine Adjoint Array of Genome 9
Even subsampled AdenineAdjoint
RT of even subsampled AdenineAdjoint
Odd subsampled AdenineAdjoint
RT of odd subsampled AdenineAdjoint

Fig.46: Genome 9 Adenine adjoints and their RT spectra



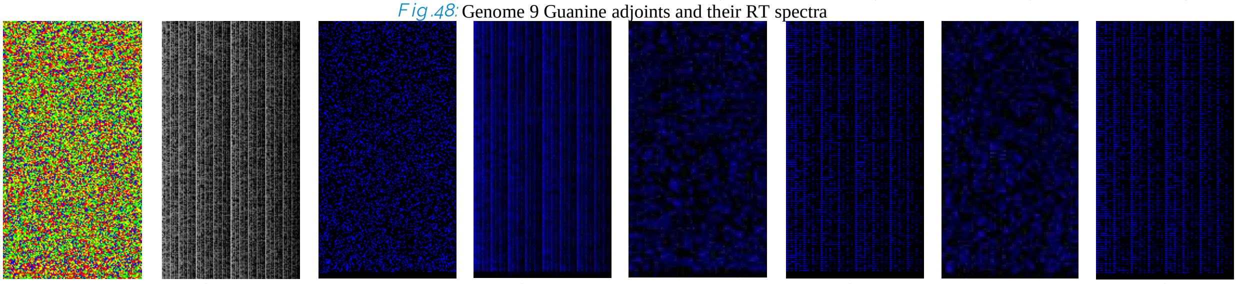
Genome 8 Bat-SL-CoVZXC21
RT of Genome 8 Bat-SL-CoVZXC21
Thymine Adjoint Array of Genome 9
RT of Thymine Adjoint Array of Genome 9
Even subsampled ThymineAdjoint
RT of even subsampled ThymineAdjoint
Odd subsampled ThymineAdjoint
RT of odd subsampled ThymineAdjoint

Fig.47: Genome 9 Thymine adjoints and their RT spectra



Genome 8 Bat-SL-CoVZXC21
RT of Genome 8 Bat-SL-CoVZXC21
Guanine Adjoint Array of Genome 9
RT of Guanine Adjoint Array of Genome 9
Even subsampled GuanineAdjoint
RT of even subsampled GuanineAdjoint
Odd subsampled GuanineAdjoint
RT of odd subsampled GuanineAdjoint

Fig.48: Genome 9 Guanine adjoints and their RT spectra



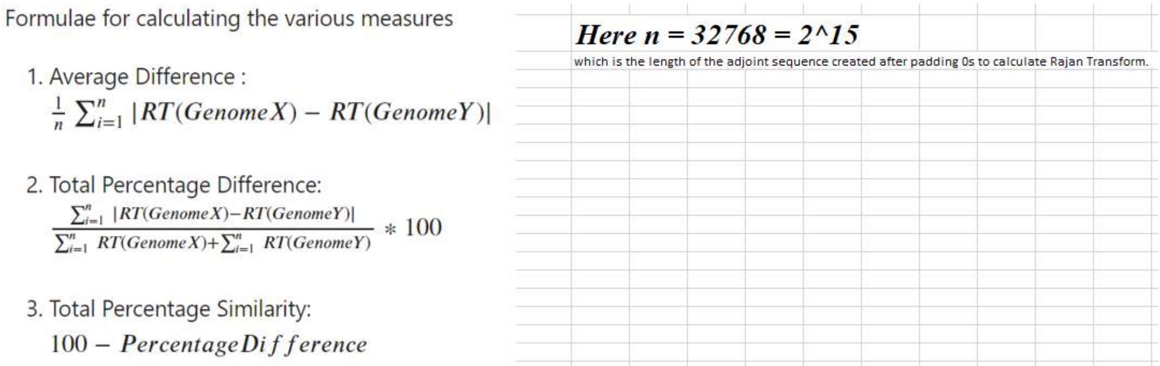
Genome 8 Bat-SL-CoVZXC21
RT of Genome 8 Bat-SL-CoVZXC21
Cytosine Adjoint Array of Genome 9
RT of Cytosine Adjoint Array of Genome 9
Even subsampled CytosineAdjoint
RT of even subsampled CytosineAdjoint
Odd subsampled CytosineAdjoint
RT of odd subsampled CytosineAdjoint

Fig.49: Genome 9 Cytosine adjoints and their RT spectra

The purpose of representing all genomes, their adjoints and subsampled arrays as two dimensional (colored) arrays is to give a visual perception of the sequences, and textual representations are not given here due to unwarranted space they consume. Moreover, a visual information is more meaningful than textual display, meaning pairwise correlations of sequences are better understood in terms of visual patterns rather than textual patterns.

4.3 PAIRWISE RAJAN TRANSFORM SPECTRAL CORRELATION OF SUBSAMPLED ADJOINTS ARRAYS

The calculation of (i) the % pairwise total absolute average difference in Rajan Transform Spectra of even and odd subsampled nucleotides adjoints of all nine genomes (ii) the % dissimilarity in Rajan Transform spectra of even and odd subsampled nucleotides adjoints of all nine genomes and (iii) the % similarity in Rajan Transform spectra of even and odd subsampled nucleotides adjoints of all nine genomes are carried out using the following formulas.



Formulae for calculating the various measures

1. Average Difference :
$$\frac{1}{n}\sum_{i=1}^{n}|RT(GenomeX)-RT(GenomeY)|$$

2. Total Percentage Difference:
$$\frac{\sum_{i=1}^{n}|RT(GenomeX)-RT(GenomeY)|}{\sum_{i=1}^{n}RT(GenomeX)+\sum_{i=1}^{n}RT(GenomeY)}*100$$

3. Total Percentage Similarity:
$$100-PercentageDifference$$

Here $n = 32768 = 2\text{^}15$

which is the length of the adjoint sequence created after padding 0s to calculate Rajan Transform.

Pairwise comparative study has been made among all nine virus genomes and results presented in Table 4

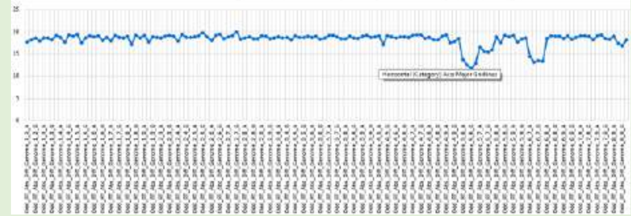
Table 4: Results of study of pairwise RT spectral correlations

Pairwise RT Spectral Correlations of Even Subsampled Nucleotides Adjoints				**Pairwise RT Spectral Correlations of Odd Subsampled Nucleotides Adjoints**			
Pairwise comparative analysis of even subsampled nucleotide adjoints arrays of all nine virus genome sequences	% Dissimilarity in Even Dyadic Rajan Transform Spectral Correlations of nine genomes	% Similarity in Even Dyadic Rajan Transform Spectral Correlations of nine genomes	% Pairwise Total bsolute Average Difference in Even Dyadic Rajan Transform Spectra of nine genomes	Pairwise comparative analysis of odd subsampled nucleotide adjoints arrays of all nine virus genome sequences	% Dissimilarity in Odd Dyadic Rajan Transform Spectral Correlations of nine genomes	% Similarity in Odd Dyadic Rajan Transform Spectral Correlations of nine genomes	% Pairwise Total bsolute Average Difference in Odd Dyadic Rajan Transform Spectra of nine genomes
Even_RT_Abs_Diff_Genome_1_2_A	3.6630859	82.18%	17.82%	Odd_RT_Abs_Diff_Genome_1_2_A	3.7165527	82.2801	17.7199
Even_RT_Abs_Diff_Genome_1_2_C	3.4545898	82.21%	17.79%	Odd_RT_Abs_Diff_Genome_1_2_C	3.5083008	81.8751	18.1249
Even_RT_Abs_Diff_Genome_1_2_G	3.5288086	81.85%	18.15%	Odd_RT_Abs_Diff_Genome_1_2_G	3.7294922	81.3509	18.6491
Even_RT_Abs_Diff_Genome_1_2_T	3.6884766	82.33%	17.67%	Odd_RT_Abs_Diff_Genome_1_2_T	3.7578125	82.143	17.857
Even_RT_Abs_Diff_Genome_1_3_A	3.8745117	81.10%	18.90%	Odd_RT_Abs_Diff_Genome_1_3_A	3.9733887	81.3345	18.6655
Even_RT_Abs_Diff_Genome_1_3_C	3.6186523	81.55%	18.45%	Odd_RT_Abs_Diff_Genome_1_3_C	3.5671387	81.4315	18.5685
Even_RT_Abs_Diff_Genome_1_3_G	3.7329102	81.08%	18.92%	Odd_RT_Abs_Diff_Genome_1_3_G	3.5849609	81.8263	18.1737
Even_RT_Abs_Diff_Genome_1_3_T	3.8251953	81.53%	18.47%	Odd_RT_Abs_Diff_Genome_1_3_T	4.0327148	80.8278	19.1722
Even_RT_Abs_Diff_Genome_1_4_A	3.7607422	81.53%	18.47%	Odd_RT_Abs_Diff_Genome_1_4_A	3.9116211	81.2755	18.7245
Even_RT_Abs_Diff_Genome_1_4_C	3.7646484	80.83%	19.17%	Odd_RT_Abs_Diff_Genome_1_4_C	3.4448242	82.3735	17.6265
Even_RT_Abs_Diff_Genome_1_4_G	3.5969238	81.57%	18.43%	Odd_RT_Abs_Diff_Genome_1_4_G	3.7905273	80.7351	19.2649
Even_RT_Abs_Diff_Genome_1_4_T	3.7001953	81.91%	18.09%	Odd_RT_Abs_Diff_Genome_1_4_T	3.9516602	81.0734	18.9266
Even_RT_Abs_Diff_Genome_1_5_A	3.9226074	81.12%	18.88%	Odd_RT_Abs_Diff_Genome_1_5_A	4.1115723	80.6098	19.3902
Even_RT_Abs_Diff_Genome_1_5_C	3.6669922	81.10%	18.90%	Odd_RT_Abs_Diff_Genome_1_5_C	3.3894043	82.4977	17.5023
Even_RT_Abs_Diff_Genome_1_5_G	3.5126953	81.95%	18.05%	Odd_RT_Abs_Diff_Genome_1_5_G	3.6650391	81.3427	18.6573
Even_RT_Abs_Diff_Genome_1_5_T	3.8713379	81.35%	18.65%	Odd_RT_Abs_Diff_Genome_1_5_T	4.0773926	80.8707	19.1293
Even_RT_Abs_Diff_Genome_1_6_A	3.7937012	81.54%	18.46%	Odd_RT_Abs_Diff_Genome_1_6_A	3.9272461	81.1761	18.8239
Even_RT_Abs_Diff_Genome_1_6_C	3.6137695	81.25%	18.75%	Odd_RT_Abs_Diff_Genome_1_6_C	3.7155762	80.8742	19.1258
Even_RT_Abs_Diff_Genome_1_6_G	3.628418	81.27%	18.73%	Odd_RT_Abs_Diff_Genome_1_6_G	3.5166016	81.9281	18.0719
Even_RT_Abs_Diff_Genome_1_6_T	3.7416992	81.93%	18.07%	Odd_RT_Abs_Diff_Genome_1_6_T	3.9807129	81.2213	18.7787
Even_RT_Abs_Diff_Genome_1_7_A	3.7250977	81.90%	18.10%	Odd_RT_Abs_Diff_Genome_1_7_A	3.7763672	82.0794	17.9206
Even_RT_Abs_Diff_Genome_1_7_C	3.6955566	80.92%	19.08%	Odd_RT_Abs_Diff_Genome_1_7_C	3.739502	80.7578	19.2422
Even_RT_Abs_Diff_Genome_1_7_G	3.4379883	82.12%	17.88%	Odd_RT_Abs_Diff_Genome_1_7_G	3.6542969	81.2577	18.7423
Even_RT_Abs_Diff_Genome_1_7_T	3.8574219	81.43%	18.57%	Odd_RT_Abs_Diff_Genome_1_7_T	3.9267578	81.4345	18.5655
Even_RT_Abs_Diff_Genome_1_8_A	3.824707	81.49%	18.51%	Odd_RT_Abs_Diff_Genome_1_8_A	4.012207	80.986	19.014
Even_RT_Abs_Diff_Genome_1_8_C	3.782959	80.62%	19.38%	Odd_RT_Abs_Diff_Genome_1_8_C	3.2475586	82.9291	17.0709
Even_RT_Abs_Diff_Genome_1_8_G	3.5625	81.67%	18.33%	Odd_RT_Abs_Diff_Genome_1_8_G	3.737793	80.84	19.16
Even_RT_Abs_Diff_Genome_1_8_T	3.8393555	81.59%	18.41%	Odd_RT_Abs_Diff_Genome_1_8_T	3.8857422	81.4321	18.5679
Even_RT_Abs_Diff_Genome_1_9_A	3.7993164	81.37%	18.63%	Odd_RT_Abs_Diff_Genome_1_9_A	4.0688477	80.7816	19.2184
Even_RT_Abs_Diff_Genome_1_9_C	3.6821289	81.14%	18.86%	Odd_RT_Abs_Diff_Genome_1_9_C	3.4023438	82.3035	17.6965
Even_RT_Abs_Diff_Genome_1_9_G	3.5581055	81.41%	18.59%	Odd_RT_Abs_Diff_Genome_1_9_G	3.6416016	81.1204	18.8796
Even_RT_Abs_Diff_Genome_1_9_T	3.8820801	81.08%	18.92%	Odd_RT_Abs_Diff_Genome_1_9_T	3.9584961	81.3134	18.6866
Even_RT_Abs_Diff_Genome_2_3_A	3.815918	81.56%	18.44%	Odd_RT_Abs_Diff_Genome_2_3_A	3.8583984	81.4341	18.5659
Even_RT_Abs_Diff_Genome_2_3_C	3.5234375	81.78%	18.22%	Odd_RT_Abs_Diff_Genome_2_3_C	3.6577148	81.091	18.909
Even_RT_Abs_Diff_Genome_2_3_G	3.6638184	81.58%	18.42%	Odd_RT_Abs_Diff_Genome_2_3_G	3.8200684	80.7651	19.2349
Even_RT_Abs_Diff_Genome_2_3_T	3.9182129	81.30%	18.70%	Odd_RT_Abs_Diff_Genome_2_3_T	3.9936523	81.0557	18.9443
Even_RT_Abs_Diff_Genome_2_4_A	3.7878418	81.57%	18.43%	Odd_RT_Abs_Diff_Genome_2_4_A	3.6557617	82.0666	17.9334
Even_RT_Abs_Diff_Genome_2_4_C	3.6320801	81.24%	18.76%	Odd_RT_Abs_Diff_Genome_2_4_C	3.8193359	80.5894	19.4106
Even_RT_Abs_Diff_Genome_2_4_G	3.4472656	82.48%	17.52%	Odd_RT_Abs_Diff_Genome_2_4_G	3.7150879	81.2462	18.7538
Even_RT_Abs_Diff_Genome_2_4_T	3.8276367	81.50%	18.50%	Odd_RT_Abs_Diff_Genome_2_4_T	3.9277344	81.2302	18.7698
Even_RT_Abs_Diff_Genome_2_5_A	3.9772949	81.03%	18.97%	Odd_RT_Abs_Diff_Genome_2_5_A	3.8964844	81.1757	18.8243
Even_RT_Abs_Diff_Genome_2_5_C	3.559082	81.39%	18.61%	Odd_RT_Abs_Diff_Genome_2_5_C	3.717041	80.9368	19.0632
Even_RT_Abs_Diff_Genome_2_5_G	3.798584	80.64%	19.36%	Odd_RT_Abs_Diff_Genome_2_5_G	3.9155273	80.2027	19.7973
Even_RT_Abs_Diff_Genome_2_5_T	3.6896973	82.43%	17.57%	Odd_RT_Abs_Diff_Genome_2_5_T	4.0349121	81.1116	18.8884
Even_RT_Abs_Diff_Genome_2_6_A	3.8769531	81.31%	18.69%	Odd_RT_Abs_Diff_Genome_2_6_A	3.6606445	82.0186	17.9814
Even_RT_Abs_Diff_Genome_2_6_C	3.498291	81.58%	18.42%	Odd_RT_Abs_Diff_Genome_2_6_C	3.7568359	80.7933	19.2067
Even_RT_Abs_Diff_Genome_2_6_G	3.5349121	81.90%	18.10%	Odd_RT_Abs_Diff_Genome_2_6_G	3.8156738	80.5253	19.4747
Even_RT_Abs_Diff_Genome_2_6_T	3.5146484	83.22%	16.78%	Odd_RT_Abs_Diff_Genome_2_6_T	3.9118652	81.5868	18.4132
Even_RT_Abs_Diff_Genome_2_7_A	3.8500977	81.46%	18.54%	Odd_RT_Abs_Diff_Genome_2_7_A	3.8691406	81.1882	18.8118
Even_RT_Abs_Diff_Genome_2_7_C	3.564209	81.33%	18.67%	Odd_RT_Abs_Diff_Genome_2_7_C	3.7402344	80.8849	19.1151
Even_RT_Abs_Diff_Genome_2_7_G	3.6101074	81.38%	18.62%	Odd_RT_Abs_Diff_Genome_2_7_G	3.9228516	80.0177	19.9823
Even_RT_Abs_Diff_Genome_2_7_T	3.5981445	82.88%	17.12%	Odd_RT_Abs_Diff_Genome_2_7_T	3.8813477	81.6898	18.3102
Even_RT_Abs_Diff_Genome_2_8_A	3.8498535	81.54%	18.46%	Odd_RT_Abs_Diff_Genome_2_8_A	3.824707	81.43	18.57
Even_RT_Abs_Diff_Genome_2_8_C	3.6696777	80.93%	19.07%	Odd_RT_Abs_Diff_Genome_2_8_C	3.6013184	81.201	18.799
Even_RT_Abs_Diff_Genome_2_8_G	3.3891602	82.71%	17.29%	Odd_RT_Abs_Diff_Genome_2_8_G	3.6130371	81.6059	18.3941
Even_RT_Abs_Diff_Genome_2_8_T	3.8879395	81.57%	18.43%	Odd_RT_Abs_Diff_Genome_2_8_T	3.8642578	81.5761	18.4239
Even_RT_Abs_Diff_Genome_2_9_A	3.9089355	81.01%	18.99%	Odd_RT_Abs_Diff_Genome_2_9_A	3.9316406	80.9758	19.0242

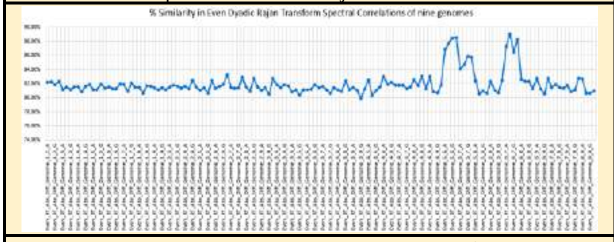
Even_RT_Abs_Diff_Genome_2_9_C	3.581543	81.40%	18.60%	Odd_RT_Abs_Diff_Genome_2_9_C	3.6738281	81.0228	18.9772
Even_RT_Abs_Diff_Genome_2_9_G	3.7624512	80.51%	19.49%	Odd_RT_Abs_Diff_Genome_2_9_G	3.5754395	81.5914	18.4086
Even_RT_Abs_Diff_Genome_2_9_T	3.5844727	82.73%	17.27%	Odd_RT_Abs_Diff_Genome_2_9_T	3.9555664	81.3684	18.6316
Even_RT_Abs_Diff_Genome_3_4_A	3.7316895	81.80%	18.20%	Odd_RT_Abs_Diff_Genome_3_4_A	3.9042969	81.1375	18.8625
Even_RT_Abs_Diff_Genome_3_4_C	3.6411133	81.39%	18.61%	Odd_RT_Abs_Diff_Genome_3_4_C	3.6459961	81.3323	18.6677
Even_RT_Abs_Diff_Genome_3_4_G	3.6286621	81.83%	18.17%	Odd_RT_Abs_Diff_Genome_3_4_G	3.6640625	81.2461	18.7539
Even_RT_Abs_Diff_Genome_3_4_T	3.7680664	81.65%	18.35%	Odd_RT_Abs_Diff_Genome_3_4_T	3.7836914	81.9101	18.0899
Even_RT_Abs_Diff_Genome_3_5_A	4.0065918	80.84%	19.16%	Odd_RT_Abs_Diff_Genome_3_5_A	4.0195313	80.8709	19.1291
Even_RT_Abs_Diff_Genome_3_5_C	3.6499023	81.11%	18.89%	Odd_RT_Abs_Diff_Genome_3_5_C	3.6318359	81.2338	18.7662
Even_RT_Abs_Diff_Genome_3_5_G	3.9174805	80.33%	19.67%	Odd_RT_Abs_Diff_Genome_3_5_G	3.6533203	81.2707	18.7293
Even_RT_Abs_Diff_Genome_3_5_T	3.940918	81.08%	18.92%	Odd_RT_Abs_Diff_Genome_3_5_T	4.0549316	81.0092	18.9908
Even_RT_Abs_Diff_Genome_3_6_A	3.907959	81.11%	18.89%	Odd_RT_Abs_Diff_Genome_3_6_A	3.8603516	81.3251	18.6749
Even_RT_Abs_Diff_Genome_3_6_C	3.6147461	81.16%	18.84%	Odd_RT_Abs_Diff_Genome_3_6_C	3.6816406	81.0367	18.9633
Even_RT_Abs_Diff_Genome_3_6_G	3.6074219	81.80%	18.20%	Odd_RT_Abs_Diff_Genome_3_6_G	3.5200195	81.7812	18.2188
Even_RT_Abs_Diff_Genome_3_6_T	3.8640137	81.40%	18.60%	Odd_RT_Abs_Diff_Genome_3_6_T	3.9602051	81.3507	18.6493
Even_RT_Abs_Diff_Genome_3_7_A	3.8144531	81.58%	18.42%	Odd_RT_Abs_Diff_Genome_3_7_A	4.0134277	80.7796	19.2204
Even_RT_Abs_Diff_Genome_3_7_C	3.644043	81.11%	18.89%	Odd_RT_Abs_Diff_Genome_3_7_C	3.7314453	80.7869	19.2131
Even_RT_Abs_Diff_Genome_3_7_G	3.8129883	80.62%	19.38%	Odd_RT_Abs_Diff_Genome_3_7_G	3.640625	81.1945	18.8055
Even_RT_Abs_Diff_Genome_3_7_T	3.8818359	81.38%	18.62%	Odd_RT_Abs_Diff_Genome_3_7_T	3.8969727	81.6076	18.3924
Even_RT_Abs_Diff_Genome_3_8_A	3.9287109	81.11%	18.89%	Odd_RT_Abs_Diff_Genome_3_8_A	3.831543	81.6757	18.3243
Even_RT_Abs_Diff_Genome_3_8_C	3.7141113	80.90%	19.10%	Odd_RT_Abs_Diff_Genome_3_8_C	3.6101074	81.0109	18.9891
Even_RT_Abs_Diff_Genome_3_8_G	3.5070801	82.37%	17.63%	Odd_RT_Abs_Diff_Genome_3_8_G	3.6096191	81.365	18.635
Even_RT_Abs_Diff_Genome_3_8_T	3.9467773	81.14%	18.86%	Odd_RT_Abs_Diff_Genome_3_8_T	3.8720703	81.5302	18.4698
Even_RT_Abs_Diff_Genome_3_9_A	3.8054199	81.47%	18.53%	Odd_RT_Abs_Diff_Genome_3_9_A	3.9692383	81.0808	18.9192
Even_RT_Abs_Diff_Genome_3_9_C	3.6884766	81.04%	18.96%	Odd_RT_Abs_Diff_Genome_3_9_C	3.6931152	80.7787	19.2213
Even_RT_Abs_Diff_Genome_3_9_G	3.932373	79.93%	20.07%	Odd_RT_Abs_Diff_Genome_3_9_G	3.5959473	81.2226	18.7774
Even_RT_Abs_Diff_Genome_3_9_T	3.8806152	81.16%	18.85%	Odd_RT_Abs_Diff_Genome_3_9_T	3.987793	81.208	18.792
Even_RT_Abs_Diff_Genome_4_5_A	3.6289063	82.53%	17.47%	Odd_RT_Abs_Diff_Genome_4_5_A	3.925293	80.9597	19.0403
Even_RT_Abs_Diff_Genome_4_5_C	3.824707	80.23%	19.77%	Odd_RT_Abs_Diff_Genome_4_5_C	3.361084	82.9263	17.0737
Even_RT_Abs_Diff_Genome_4_5_G	3.739502	81.01%	18.99%	Odd_RT_Abs_Diff_Genome_4_5_G	3.7041016	80.9612	19.0388
Even_RT_Abs_Diff_Genome_4_5_T	3.7946777	81.56%	18.44%	Odd_RT_Abs_Diff_Genome_4_5_T	3.9902344	81.1753	18.8247
Even_RT_Abs_Diff_Genome_4_6_A	3.4975586	82.99%	17.01%	Odd_RT_Abs_Diff_Genome_4_6_A	3.7705078	81.4026	18.5974
Even_RT_Abs_Diff_Genome_4_6_C	3.4833984	81.87%	18.13%	Odd_RT_Abs_Diff_Genome_4_6_C	3.7209473	81.1572	18.8428
Even_RT_Abs_Diff_Genome_4_6_G	3.4978027	82.16%	17.84%	Odd_RT_Abs_Diff_Genome_4_6_G	3.638916	81.1167	18.8833
Even_RT_Abs_Diff_Genome_4_6_T	3.746582	81.74%	18.26%	Odd_RT_Abs_Diff_Genome_4_6_T	3.9372559	81.3222	18.6778
Even_RT_Abs_Diff_Genome_4_7_A	3.7578125	81.74%	18.26%	Odd_RT_Abs_Diff_Genome_4_7_A	3.9387207	80.7718	19.2282
Even_RT_Abs_Diff_Genome_4_7_C	3.5224609	81.76%	18.24%	Odd_RT_Abs_Diff_Genome_4_7_C	3.8046875	80.7398	19.2602
Even_RT_Abs_Diff_Genome_4_7_G	3.6547852	81.22%	18.78%	Odd_RT_Abs_Diff_Genome_4_7_G	3.7375488	80.6436	19.3564
Even_RT_Abs_Diff_Genome_4_7_T	3.7988281	81.55%	18.45%	Odd_RT_Abs_Diff_Genome_4_7_T	3.890625	81.502	18.498
Even_RT_Abs_Diff_Genome_4_8_A	3.6030273	82.57%	17.43%	Odd_RT_Abs_Diff_Genome_4_8_A	3.8325195	81.3164	18.6836
Even_RT_Abs_Diff_Genome_4_8_C	3.5661621	81.68%	18.32%	Odd_RT_Abs_Diff_Genome_4_8_C	3.5068359	81.8714	18.1286
Even_RT_Abs_Diff_Genome_4_8_G	3.3295898	83.08%	16.92%	Odd_RT_Abs_Diff_Genome_4_8_G	3.5112305	81.8258	18.1742
Even_RT_Abs_Diff_Genome_4_8_T	3.878418	81.24%	18.76%	Odd_RT_Abs_Diff_Genome_4_8_T	3.953125	81.0029	18.9971
Even_RT_Abs_Diff_Genome_4_9_A	3.4692383	82.99%	17.01%	Odd_RT_Abs_Diff_Genome_4_9_A	3.9804688	80.6614	19.3386
Even_RT_Abs_Diff_Genome_4_9_C	3.7324219	80.84%	19.16%	Odd_RT_Abs_Diff_Genome_4_9_C	3.4179688	82.5136	17.4864
Even_RT_Abs_Diff_Genome_4_9_G	3.7397461	80.70%	19.30%	Odd_RT_Abs_Diff_Genome_4_9_G	3.3989258	82.2047	17.7953
Even_RT_Abs_Diff_Genome_4_9_T	3.7102051	81.76%	18.24%	Odd_RT_Abs_Diff_Genome_4_9_T	3.8974609	81.4983	18.5017
Even_RT_Abs_Diff_Genome_5_6_A	2.7392578	86.94%	13.06%	Odd_RT_Abs_Diff_Genome_5_6_A	2.8139648	86.3323	13.6677
Even_RT_Abs_Diff_Genome_5_6_C	2.3364258	87.69%	12.31%	Odd_RT_Abs_Diff_Genome_5_6_C	2.4594727	87.432	12.568
Even_RT_Abs_Diff_Genome_5_6_G	2.2614746	88.43%	11.57%	Odd_RT_Abs_Diff_Genome_5_6_G	2.2607422	88.249	11.751
Even_RT_Abs_Diff_Genome_5_6_T	2.3874512	88.53%	11.47%	Odd_RT_Abs_Diff_Genome_5_6_T	2.7780762	87.0882	12.9118
Even_RT_Abs_Diff_Genome_5_7_A	3.3391113	84.09%	15.91%	Odd_RT_Abs_Diff_Genome_5_7_A	3.4372559	83.4732	16.5268
Even_RT_Abs_Diff_Genome_5_7_C	2.9099121	84.74%	15.26%	Odd_RT_Abs_Diff_Genome_5_7_C	3.0444336	84.4483	15.5517
Even_RT_Abs_Diff_Genome_5_7_G	2.7495117	85.83%	14.17%	Odd_RT_Abs_Diff_Genome_5_7_G	2.9750977	84.5669	15.4331
Even_RT_Abs_Diff_Genome_5_7_T	2.9802246	85.74%	14.26%	Odd_RT_Abs_Diff_Genome_5_7_T	3.4138184	84.0987	15.9013
Even_RT_Abs_Diff_Genome_5_8_A	3.7143555	82.38%	17.62%	Odd_RT_Abs_Diff_Genome_5_8_A	3.9213867	81.1713	18.8287
Even_RT_Abs_Diff_Genome_5_8_C	3.737793	80.56%	19.44%	Odd_RT_Abs_Diff_Genome_5_8_C	3.3493652	82.5247	17.4753
Even_RT_Abs_Diff_Genome_5_8_G	3.7353516	80.96%	19.04%	Odd_RT_Abs_Diff_Genome_5_8_G	3.7114258	80.7579	19.2421
Even_RT_Abs_Diff_Genome_5_8_T	4.0654297	80.62%	19.38%	Odd_RT_Abs_Diff_Genome_5_8_T	4.0019531	81.163	18.837
Even_RT_Abs_Diff_Genome_5_9_A	3.6767578	82.33%	17.67%	Odd_RT_Abs_Diff_Genome_5_9_A	4.0048828	80.8351	19.1649
Even_RT_Abs_Diff_Genome_5_9_C	3.6474609	81.04%	18.96%	Odd_RT_Abs_Diff_Genome_5_9_C	3.4211426	82.3365	17.6635
Even_RT_Abs_Diff_Genome_5_9_G	3.7324219	80.68%	19.32%	Odd_RT_Abs_Diff_Genome_5_9_G	3.5148926	81.5669	18.4331
Even_RT_Abs_Diff_Genome_5_9_T	3.623291	82.44%	17.56%	Odd_RT_Abs_Diff_Genome_5_9_T	3.9892578	81.4466	18.5534
Even_RT_Abs_Diff_Genome_6_7_A	2.6489258	87.25%	12.75%	Odd_RT_Abs_Diff_Genome_6_7_A	2.9575195	85.5426	14.4574
Even_RT_Abs_Diff_Genome_6_7_C	2.0810547	89.01%	10.99%	Odd_RT_Abs_Diff_Genome_6_7_C	2.5712891	86.9064	13.0936
Even_RT_Abs_Diff_Genome_6_7_G	2.6064453	86.51%	13.49%	Odd_RT_Abs_Diff_Genome_6_7_G	2.578125	86.4965	13.5035
Even_RT_Abs_Diff_Genome_6_7_T	2.4472656	88.26%	11.74%	Odd_RT_Abs_Diff_Genome_6_7_T	2.8464355	86.6689	13.3311
Even_RT_Abs_Diff_Genome_6_8_A	3.628418	82.61%	17.39%	Odd_RT_Abs_Diff_Genome_6_8_A	3.7939453	81.4797	18.5203
Even_RT_Abs_Diff_Genome_6_8_C	3.3791504	82.31%	17.69%	Odd_RT_Abs_Diff_Genome_6_8_C	3.6569824	80.9807	19.0193
Even_RT_Abs_Diff_Genome_6_8_G	3.4597168	82.28%	17.72%	Odd_RT_Abs_Diff_Genome_6_8_G	3.6179199	81.061	18.939
Even_RT_Abs_Diff_Genome_6_8_T	3.9177246	81.27%	18.73%	Odd_RT_Abs_Diff_Genome_6_8_T	3.9987793	81.0737	18.9263

Even_RT_Abs_Diff_Genome_6_9_A	3.5541992	82.74%	17.26%	Odd_RT_Abs_Diff_Genome_6_9_A	3.7983398	81.5217	18.4783
Even_RT_Abs_Diff_Genome_6_9_C	3.5830078	81.25%	18.75%	Odd_RT_Abs_Diff_Genome_6_9_C	3.7114258	80.8984	19.1016
Even_RT_Abs_Diff_Genome_6_9_G	3.736084	80.57%	19.43%	Odd_RT_Abs_Diff_Genome_6_9_G	3.4484863	81.7379	18.2621
Even_RT_Abs_Diff_Genome_6_9_T	3.5561523	82.72%	17.28%	Odd_RT_Abs_Diff_Genome_6_9_T	4.0119629	81.2389	18.7611
Even_RT_Abs_Diff_Genome_7_8_A	3.8725586	81.45%	18.55%	Odd_RT_Abs_Diff_Genome_7_8_A	3.940918	80.9572	19.0428
Even_RT_Abs_Diff_Genome_7_8_C	3.4782715	81.88%	18.12%	Odd_RT_Abs_Diff_Genome_7_8_C	3.6674805	80.9329	19.0671
Even_RT_Abs_Diff_Genome_7_8_G	3.5874023	81.49%	18.51%	Odd_RT_Abs_Diff_Genome_7_8_G	3.6347656	81.0112	18.9888
Even_RT_Abs_Diff_Genome_7_8_T	3.9204102	81.32%	18.68%	Odd_RT_Abs_Diff_Genome_7_8_T	3.8168945	81.8942	18.1058
Even_RT_Abs_Diff_Genome_7_9_A	3.7678223	81.72%	18.28%	Odd_RT_Abs_Diff_Genome_7_9_A	3.9560547	80.9488	19.0512
Even_RT_Abs_Diff_Genome_7_9_C	3.6870117	80.80%	19.20%	Odd_RT_Abs_Diff_Genome_7_9_C	3.7421875	80.7469	19.2531
Even_RT_Abs_Diff_Genome_7_9_G	3.5991211	81.14%	18.86%	Odd_RT_Abs_Diff_Genome_7_9_G	3.4921875	81.5442	18.4558
Even_RT_Abs_Diff_Genome_7_9_T	3.5634766	82.75%	17.25%	Odd_RT_Abs_Diff_Genome_7_9_T	3.8857422	81.7891	18.2109
Even_RT_Abs_Diff_Genome_8_9_A	3.5922852	82.65%	17.35%	Odd_RT_Abs_Diff_Genome_8_9_A	3.9384766	81.0595	18.9405
Even_RT_Abs_Diff_Genome_8_9_C	3.7578125	80.59%	19.41%	Odd_RT_Abs_Diff_Genome_8_9_C	3.2885742	82.7161	17.2839
Even_RT_Abs_Diff_Genome_8_9_G	3.739502	80.62%	19.38%	Odd_RT_Abs_Diff_Genome_8_9_G	3.1938477	83.1304	16.8696
Even_RT_Abs_Diff_Genome_8_9_T	3.9423828	80.99%	19.01%	Odd_RT_Abs_Diff_Genome_8_9_T	3.8193359	81.9107	18.0893

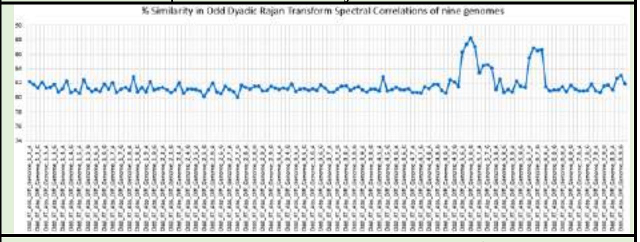
% Pairwise Total Absolute Average Difference in RT Spectra of Even Subsampled Nucleotides Adjoints of all Nine Genomes



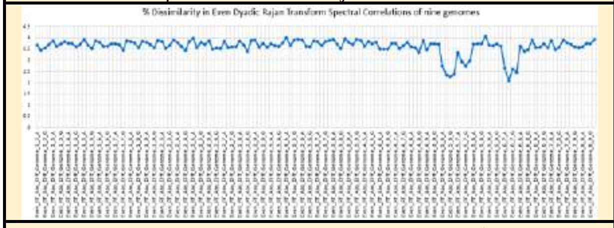
% Pairwise Total Absolute Average Difference in RT Spectra of Odd Subsampled Nucleotides Adjoints of all Nine Genomes



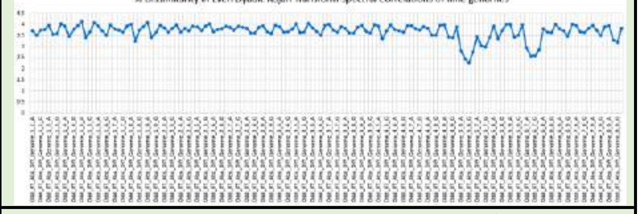
% Pairwise Similarity in RT Spectra of Even Subsampled Nucleotides Adjoints of all Nine Genomes



% Pairwise Similarity in RT Spectra of Odd Subsampled Nucleotides Adjoints of all Nine Genomes



% Pairwise Dissimilrity in RT Spectra of Even Subsampled Nucleotides Adjoints of all Nine Genomes



% Pairwise Dissimilarity in RT Spectra of Odd Subsampled Nucleotides Adjoints of all Nine Genomes

Observations

1. Subsampled adjoints of Genomes 5 and 6 exhibit maximum similarity (86.3323% to 88.53%) in their RT spectra.
1. Subsampled adjoints of Genomes 6 and 7 also exhibit high-level similarity (85.5426% to 89.01%) in their RT spectra.
2. Cytosine adjoint arrays of genomes 6 and 7 exhibit the highest level similarity (**89.01%**) in their RT spectra.

Inference

1. Even subsampled Cytosine adjoint arrays of Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete genome (Genome 6) and Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome (Genome 7) exhibit maximum similarity in their RT spectra, meaning their functional behavior is almost the same.

4.4 RAJAN TRANSFORM SPECTRAL REGULATION OF PAIRS OF SARS-CORONA VIRUS GENOME SEQUENCES

As outlined earlier, the linear pair forming property of Rajan Transform is useful in getting the CASN for any pair of number sequences. This property coupled with 'Regenerative Property' might open up new avenues in devising methods for artificially synthesizing harmful RNA strains and also their combating duals of harmless RNA strains. A thorough study was carried out on nine genomes considered for the entire research.

Basic experiment

The aim of this experiment is to examine the Linear Pair forming Property of Rajan Transform (RT) for different pairs of sequences. The lengths of these pairs are defined by 2^n where n = 3, 4, 5, 6, 7, 8 and one can generate sequences of lengths 8, 16, 32, 64, 128 and 512. The iteration denoted as CASN (Correlation Agreement Stage Number) at which the linear pair forming property holds true provides an idea about a formal RT spectral relationship between the two sequences. An effort was made to find out whether this CASN values bear a formal relation with the lengths of various pairs of sequences.

Result

Through some manual experiments, it was found that the CAS at which the linear pair forming property holds good for a pair of sequences is not directly related to the length of the pair of sequences. However, for pairs of sequences of a particular length, one CASN is found to be dominant. For instance, sequences of length 8 having random values from the closed interval [1, 100] satisfy linear pair forming property at iterations 1, 2, 3 and 4 but the iteration 3 is the dominant CASN, meaning most of the sequence pairs of length 8 satisfy the linear pair forming property at iteration 3.

Experimental outcomes of dominant CASNs

For random pairs of sequences of lengths 8, 16, 32, 64, 128, 256, 512, a total number of 1000 experiments per pair were carried out programmatically. Fig. 50 shows a graph connecting number of outcomes favorable to various CASN values.

Major breakthrough result

Any pair of sequences of length of 2^n, where n = 3, 4, 5, 6, 7…, does have a certain CASN.

Other significant results

It is to be noted that the sequence pairs formed for a particular length contains values between 1 and 100 chosen at random.

1. **Pairs of sequences with length 8:** Out of 1000 experiments on pairs of length 8, it was found that 807 experiments yielded iteration number (CASN) 3, where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 8 will satisfy linear pair forming property at iteration 3 with a probability of 0.81 approximately.
2. **Pairs of sequences with length 16:** Out of 1000 experiments for pairs of length 16, it was found that 551 experiments yielded iteration number 3, where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 16 will satisfy linear pair forming property at iteration 3 with a probability of 0.55 approximately. Iteration 4 is the second highest chance for length 16.
3. **Pairs of sequences with length 32:** Out of 1000 experiments for length 32 pairs, it was found that 439 experiments yielded iteration number 5 as the iteration where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 32 will satisfy linear pair forming property at iteration 5 with a probability of 0.44 approximately. Iteration 4 is the second highest chance for length 32.
4. **Pairs of sequences with length 64:** Out of 1000 experiments for length 64 pairs, it was found that 774 experiments yielded iteration number 5 as the iteration where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 64 will satisfy linear pair forming property at iteration 5 with a probability of 0.77 approximately.
5. **Pairs of sequences with length 128:** Out of 1000 experiments for length 128 pairs, it was found that 558 experiments yielded iteration number 5 as the iteration where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 128 will satisfy linear pair forming property at iteration 5 with a probability of 0.56 approximately. Iteration 6 is the second highest chance for length 128.
6. **Pairs of sequences with length 256:** Out of 1000 experiments for length 256 pairs, it was found that 529 experiments yielded iteration number 7 as the iteration where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 256 will satisfy linear pair forming property at iteration 7 with a probability of 0.53 approximately. Iteration 5 is the second highest chance for length 256.
7. **Pairs of sequences with length 512:** Out of 1000 experiments for length 512 pairs, it was found that 829 experiments yielded iteration number 7 as the iteration where linear pair forming property holds true. Therefore, one can say that pairs of sequences of length 512 will satisfy linear pair forming property at iteration 7 with a probability of 0.83 approximately

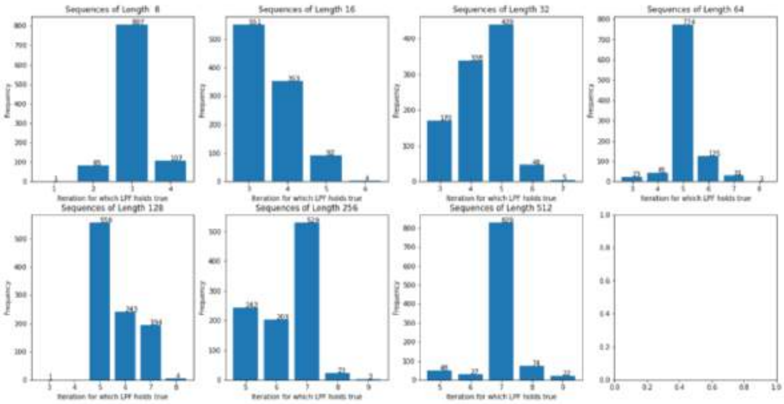


Fig.50: Number of outcomes favorable to various CASN values

A detailed study is required to arrive at a formal relationship between the lengths of sequences and the iteration at which the linear pair forming property holds true.

Case Study: Linear Pair Forming Property for first 128 values of Genome 1 and Genome 2

Consider two sequences x(n), the first 128 values of Genome 1 and y(n), the first 128 values of Genome 2. Rajan Transform is applied repeatedly and it was found that the linear pair forming property is satisfied at iteration 7. Let $X_n(k)$ be the n^{th} order Rajan Transform of x(n) and $Y_n(k)$ be the n^{th} order Rajan Transform of y(n). Let z(n) be the pointwise addition of x(n) and y(n). Table 5 shows various higher order RT s

Table 5: Higher order RT spectra pertaining to Genome 1 and Genome 2

Iteration 1	
x(n)	4, 2, 1, 4, 4, 4, 1, 3, 3, 1, 1, 1, 1, 3, 4, 4, 1, 1, 4, 4, 1, 1, 4, 4, 2, 4, 3, 1, 2, 4, 2, 4, 2, 2, 3, 2, 1, 3, 1, 2, 4, 2, 3, 2, 2, 4, 2, 4, 2, 1, 1, 1, 4, 3, 1, 1, 4, 2, 2, 2, 1, 1, 1, 1, 2, 4, 2, 3, 2, 3, 2, 1, 3, 4, 2, 3, 2, 4, 3, 4, 2, 4, 3, 3, 4, 2, 3, 4, 1, 2, 3, 4, 4, 2, 1, 3, 2, 3, 4, 1, 4, 4, 2, 1, 4, 3, 4, 1, 3, 2, 1, 2, 1, 1, 1, 4, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2
y(n)	1, 3, 1, 2, 4, 2, 3, 2, 2, 4, 2, 4, 2, 1, 1, 1, 4, 3, 1, 1, 4, 2, 2, 2, 1, 1, 1, 1, 2, 4, 2, 3, 2, 3, 2, 1, 3, 4, 2, 3, 2, 4, 3, 4, 2, 4, 3, 3, 4, 2, 3, 4, 1, 2, 3, 4, 4, 2, 1, 3, 2, 3, 4, 1, 4, 4, 2, 1, 4, 3, 4, 1, 3, 2, 1, 2, 1, 1, 1, 4, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 4, 2, 3, 2, 4, 3, 2, 2, 3, 1, 4, 1, 1, 3, 1, 1, 1, 4, 3, 1, 3, 2, 1, 1, 4, 2, 4, 3, 2, 4, 4, 4, 2, 4
$X_1(k)$	304, 14, 8, 2, 16, 2, 6, 4, 28, 6, 22, 0, 2, 0, 2, 0, 46, 4, 6, 4, 26, 4, 6, 0, 12, 6, 8, 2, 2, 0, 2, 0, 70, 4, 22, 0, 14, 4, 10, 4, 18, 0, 6, 4, 8, 2, 2, 0, 32, 2, 8, 2, 12, 2, 8, 2, 8, 2, 6, 0, 4, 2, 2, 0, 80, 6, 12, 6, 16, 2, 10, 0, 10, 4, 8, 6, 8, 6, 8, 6, 16, 2, 6, 0, 6, 4, 2, 0, 12, 2, 6, 4, 6, 0, 4, 2, 28, 10, 8, 2, 10, 4, 4, 2, 4, 2, 2, 0, 2, 0, 2, 0, 18, 0, 8, 2, 4, 2, 2, 0, 6, 4, 2, 0, 2, 0, 2, 0
$Y_1(k)$	301, 1, 9, 7, 17, 15, 3, 1, 29, 1, 17, 9, 7, 1, 3, 1, 47, 3, 13, 5, 15, 13, 1, 1, 13, 3, 3, 1, 7, 1, 3, 1, 61, 3, 5, 1, 7, 3, 3, 3, 23, 13, 1, 1, 7, 3, 1, 1, 27, 1, 11, 3, 11, 5, 3, 1, 7, 5, 5, 3, 3, 1, 1, 1, 85, 9, 1, 1, 9, 1, 5, 1, 15, 1, 5, 1, 9, 3, 5, 3, 19, 1, 5, 5, 5, 1, 1, 1, 11, 5, 5, 1, 3, 1, 1, 1, 35, 3, 7, 5, 9, 5, 3, 3, 15, 1, 3, 1, 5, 1, 3, 3, 13, 3, 3, 1, 7, 1, 3, 1, 5, 3, 3, 1, 3, 1, 1, 1
z(n) = x(n)+ y(n)	5, 5, 2, 6, 8, 6, 4, 5, 5, 5, 3, 5, 3, 4, 5, 5, 5, 4, 5, 5, 5, 3, 6, 6, 3, 5, 4, 2, 4, 8, 4, 7, 4, 5, 5, 3, 4, 7, 3, 5, 6, 6, 6, 6, 4, 8, 5, 7, 6, 3, 4, 5, 5, 5, 4, 5, 8, 4, 3, 5, 3, 4, 5, 2, 6, 8, 4, 4, 6, 6, 6, 2, 6, 6, 3, 5, 3, 5, 4, 8, 3, 5, 5, 4, 5, 4, 4, 5, 2, 4, 5, 6, 6, 3, 5, 5, 5, 5, 8, 4, 6, 6, 5, 2, 8, 4, 5, 4, 4, 3, 2, 6, 4, 2, 4, 6, 2, 2, 6, 3, 5, 5, 3, 5, 5, 6, 4, 6
$X_1(k)$+ $Y_1(k)$	605, 15, 17, 9, 33, 17, 9, 5, 57, 7, 39, 9, 9, 1, 5, 1, 93, 7, 19, 9, 41, 17, 7, 1, 25, 9, 11, 3, 9, 1, 5, 1, 131, 7, 27, 1, 21, 7, 13, 7, 41, 13, 7, 5, 15, 5, 3, 1, 59, 3, 19, 5, 23, 7, 11, 3, 15, 7, 11, 3, 7, 3, 3, 1, 165, 15, 13, 7, 25, 3, 15, 1, 25, 5, 13, 7, 17, 9, 13, 9, 35, 3, 11, 5, 11, 5, 3, 1, 23, 7, 11, 5, 9, 1, 5, 3, 63, 13, 15, 7, 19, 9, 7, 5, 19, 3, 5, 1, 7, 1, 5, 3, 31, 3, 11, 3, 11, 3, 5, 1, 11, 7, 5, 1, 5, 1, 3, 1
$Z_1(k)$	605, 15, 15, 11, 19, 3, 5, 5, 41, 5, 15, 5, 17, 7, 15, 9, 67, 13, 21, 5, 9, 3, 7, 5, 27, 9, 21, 7, 7, 1, 5, 3, 73, 21, 7, 3, 21, 13, 13, 5, 23, 1, 13, 5, 11, 1, 5, 1, 31, 3, 11, 1, 11, 3, 5, 3, 11, 7, 5, 1, 7, 7, 1, 1, 103, 13, 5, 1, 25, 7, 5, 5, 19, 1, 7, 5, 11, 5, 9, 7, 29, 1, 3, 1, 9, 1, 1, 1, 15, 7, 3, 3, 7, 3, 3, 1, 37, 15, 3, 1, 9, 7, 5, 3, 23, 5, 5, 1, 5, 3, 1, 1, 17, 3, 3, 1, 7, 1, 1, 1, 11, 3, 1, 1, 5, 1, 3, 1
Iteration 2	
$X_2(k)$	1196, 884, 668, 572, 668, 572, 508, 468, 644, 572, 548, 500, 436, 428, 420, 420, 512, 408, 368, 336, 428, 372, 372, 340, 400, 328, 360, 312, 368, 328, 360, 328, 464, 344, 304, 288, 336, 320, 272, 264, 320, 304, 280, 272, 264, 256, 240, 240, 340, 292, 268, 260, 300, 276, 252, 252, 300, 276, 260, 260, 272, 264, 256, 256, 540, 388, 324, 324, 332, 284, 292, 268, 328, 288, 288, 256, 272, 264, 256, 256, 352, 280, 248, 248, 256, 240, 200, 192, 240, 200, 208, 192, 184, 168, 176, 168, 288, 224, 200, 176, 200, 192, 176, 176, 224, 200, 200, 200, 188, 172, 180, 172, 212, 188, 156, 156, 176, 168, 144, 144, 184, 168, 168, 160, 164, 156, 156, 156
$Y_2(k)$	1148, 784, 736, 652, 660, 616, 548, 504, 572, 472, 520, 460, 476, 408, 440, 412, 500, 440, 408, 404, 416, 412, 400, 396, 352, 324, 340, 336, 336, 308, 324, 320, 444, 336, 328, 308, 340, 312, 316, 288, 344, 284, 308, 272, 316, 296, 288, 276, 332, 272, 256, 252, 300, 256, 256, 236, 296, 268, 252, 248, 276, 256, 240, 228, 484, 304, 328, 252, 340, 288, 252, 232, 324, 248, 288, 236, 272, 220, 260, 216, 276, 208, 204, 200, 244, 208, 204, 200, 204, 176, 184, 180, 184, 180, 176, 172, 300, 192, 232, 180, 220, 192, 196, 168, 240, 180, 220, 184, 216, 172, 196, 176, 236, 176, 184, 148, 212, 168, 176, 156, 192, 164, 172, 168, 176, 164, 168, 156
$z_1(n)$ = $X_1(k)$+ $Y_1(k)$	605, 15, 17, 9, 33, 17, 9, 5, 57, 7, 39, 9, 9, 1, 5, 1, 93, 7, 19, 9, 41, 17, 7, 1, 25, 9, 11, 3, 9, 1, 5, 1, 131, 7, 27, 1, 21, 7, 13, 7, 41, 13, 7, 5, 15, 5, 3, 1, 59, 3, 19, 5, 23, 7, 11, 3, 15, 7, 11, 3, 7, 3, 3, 1, 165, 15, 13, 7, 25, 3, 15, 1, 25, 5, 13, 7, 17, 9, 13, 9, 35, 3, 11, 5, 11, 5, 3, 1, 23, 7, 11, 5, 9, 1, 5, 3, 63, 13, 15, 7, 19, 9, 7, 5, 19, 3, 5, 1, 7, 1, 5, 3, 31, 3, 11, 3, 11, 3, 5, 1, 11, 7, 5, 1, 5, 1, 3, 1
$X_2(k)$+ $Y_2(k)$	2344, 1668, 1404, 1224, 1328, 1188, 1056, 972, 1216, 1044, 1068, 960, 912, 836, 860, 832, 1012, 848, 776, 740, 844, 784, 772, 736, 752, 652, 700, 648, 704, 636, 684, 648, 908, 680, 632, 596, 676, 632, 588, 552, 664, 588, 588, 544, 580, 552, 528, 516, 672, 564, 524, 512, 600, 532, 508, 488, 596, 544, 512, 508, 548, 520, 496, 484, 1024, 692, 652, 576, 672, 572, 544, 500, 652, 536, 576, 492, 544, 484, 516, 472, 628, 488, 452, 448, 500, 448, 404, 392, 444, 376, 392, 372, 368, 348, 352, 340, 588, 416, 432, 356, 420, 384, 372, 344, 464, 380, 420, 384, 404, 344, 376, 348, 448, 364, 340, 304, 388, 336, 320, 300, 376, 332, 340, 328, 340, 320, 324, 312
$Z_2(k)$	2344, 1668, 1404, 1224, 1328, 1188, 1016, 1012, 1200, 1060, 1052, 976, 908, 816, 888, 828, 992, 828, 796, 760, 824, 764, 788, 760, 756, 656, 688, 636, 708, 648, 688, 644, 872, 700, 624, 620, 660, 632, 588, 568, 652, 616, 576, 540, 576, 572, 516, 512, 628, 560, 488, 484, 576, 572, 500, 496, 580, 552, 520, 508, 560, 548, 520, 516, 936, 708, 652, 600, 644, 592, 568, 532, 612, 552, 568, 540, 524, 496, 504, 476, 588, 512, 488, 476, 444, 424, 420, 408, 444, 400, 376, 348, 376, 364, 344, 340, 552, 420, 420, 368, 428, 408, 360, 356, 460, 416, 392, 380, 400, 372, 364, 336, 420, 376, 336, 324, 356, 352, 320, 316, 368, 348, 332, 328, 336, 324, 320, 316
Iteration 3	
$X_3(k)$	38912, 2080, 3472, 1088, 3584, 992, 1584, 640, 3584, 832, 2160, 672, 1152, 384, 848, 480, 5888, 544, 1296, 320, 2048, 480, 880, 256, 1664, 640, 880, 352, 664, 296, 632, 312, 8960, 1056, 1296, 384, 1792, 416, 880, 256, 2304, 384, 1008, 352, 896, 0, 528, 224, 4096, 224, 848, 192, 1280, 224, 624, 64, 960, 320, 544, 240, 576, 64, 416, 80, 10240, 704, 1264, 224, 1024, 320, 592, 96, 1280, 416, 784, 192, 328, 88, 232, 136, 2368, 256, 912, 96, 1216, 128, 368, 96, 1144, 216, 696, 72, 520, 232, 200, 184, 3616, 480, 592, 64, 800, 160, 496, 0, 640, 48, 528, 112, 352, 16, 272, 48, 2272, 160, 656, 64, 1248, 224, 368, 0, 928, 192, 496, 32, 296, 120, 136, 24
$Y_3(k)$	38528, 2728, 2496, 1160, 2496, 1064, 768, 712, 3904, 744, 1536, 520, 1088, 616, 640, 520, 6016, 1336, 960, 344, 1600, 760, 384, 344, 1472, 248, 768, 216, 960, 440, 408, 256, 7808, 536, 704, 344, 1344, 440, 384, 248, 2752, 184, 896, 248, 704, 408, 280, 256, 3456, 776, 960, 264, 1216, 296, 256, 168, 1104, 376, 624, 216, 560, 312, 152, 144, 10880, 136, 704, 232, 1088, 168, 384, 264, 2112, 72, 384, 104, 704, 104, 384, 328, 2432, 216, 448, 184, 704, 248, 256, 216, 1216, 280, 384, 184, 576, 56, 256, 152, 4480, 312, 192, 120, 960, 248, 256, 120, 1728, 152, 512, 40, 576, 280, 256, 216, 1664, 296, 448, 232, 832, 168, 168, 128, 832, 200, 256, 136, 448, 136, 200, 128
$z_2(n)$ = $X_2(k)$+ $Y_2(k)$	2344, 1668, 1404, 1224, 1328, 1188, 1056, 972, 1216, 1044, 1068, 960, 912, 836, 860, 832, 1012, 848, 776, 740, 844, 784, 772, 736, 752, 652, 700, 648, 704, 636, 684, 648, 908, 680, 632, 596, 676, 632, 588, 552, 664, 588, 588, 544, 580, 552, 528, 516, 672, 564, 524, 512, 600, 532, 508, 488, 596, 544, 512, 508, 548, 520, 496, 484, 1024, 692, 652, 576, 672, 572, 544, 500, 652, 536, 576, 492,

	544, 484, 516, 472, 628, 488, 452, 448, 500, 448, 404, 392, 444, 376, 392, 372, 368, 348, 352, 340, 588, 416, 432, 356, 420, 384, 372, 344, 464, 380, 420, 384, 404, 344, 376, 348, 448, 364, 340, 304, 388, 336, 320, 300, 376, 332, 340, 328, 340, 320, 324, 312
$X_3(k)+Y_3(k)$	77440, 4808, 5968, 2248, 6080, 2056, 2352, 1352, 7488, 1576, 3696, 1192, 2240, 1000, 1488, 1000, 11904, 1880, 2256, 664, 3648, 1240, 1264, 600, 3136, 888, 1648, 568, 1624, 736, 1040, 568, 16768, 1592, 2000, 728, 3136, 856, 1264, 504, 5056, 568, 1904, 600, 1600, 408, 808, 480, 7552, 1000, 1808, 456, 2496, 520, 880, 232, 2064, 696, 1168, 456, 1136, 376, 568, 224, 21120, 840, 1968, 456, 2112, 488, 976, 360, 3392, 488, 1168, 296, 1032, 192, 616, 464, 4800, 472, 1360, 280, 1920, 376, 624, 312, 2360, 496, 1080, 256, 1096, 288, 456, 336, 8096, 792, 784, 184, 1760, 408, 752, 120, 2368, 200, 1040, 152, 928, 296, 528, 264, 3936, 456, 1104, 296, 2080, 392, 536, 128, 1760, 392, 752, 168, 744, 256, 336, 152
$Z_3(k)$	77440, 4808, 5968, 2248, 6080, 2056, 2352, 1352, 7488, 1576, 3696, 1192, 2240, 1000, 1488, 1000, 11904, 1880, 2256, 664, 3648, 1240, 1264, 600, 3136, 888, 1648, 568, 1600, 760, 912, 696, 16768, 1592, 2000, 728, 3136, 856, 1264, 504, 5056, 568, 1904, 600, 1600, 408, 784, 504, 7552, 1000, 1808, 456, 2496, 520, 880, 232, 2008, 720, 1176, 480, 1192, 352, 552, 208, 21120, 840, 1968, 456, 2112, 488, 976, 360, 3392, 488, 1168, 296, 960, 264, 688, 392, 4496, 360, 1376, 264, 2224, 488, 608, 328, 2608, 552, 1120, 328, 848, 232, 416, 264, 8064, 824, 816, 152, 1728, 440, 784, 88, 2240, 312, 1168, 40, 840, 368, 616, 160, 3968, 424, 1072, 328, 2112, 360, 464, 200, 1728, 424, 784, 136, 704, 296, 304, 136
Iteration 4	
$X_4(k)$	153168, 113072, 85488, 74800, 85424, 73488, 65552, 59856, 82512, 73136, 70000, 62576, 57392, 55312, 55312, 54544, 65968, 51792, 47632, 43792, 54352, 47856, 46704, 43632, 50768, 42416, 45552, 40176, 45872, 41104, 45072, 40976, 59312, 44112, 38928, 36048, 43088, 40688, 35184, 33072, 41040, 38832, 35824, 35632, 35376, 33680, 32336, 32272, 43792, 37424, 34288, 32688, 38128, 35472, 31952, 31760, 38128, 35280, 33296, 32848, 33632, 32800, 31968, 31552, 69184, 49600, 41632, 39936, 42432, 36608, 36704, 34944, 41920, 36928, 36704, 34304, 33216, 32896, 31520, 31232, 44992, 35904, 31584, 30400, 32832, 30464, 26144, 24384, 30784, 25536, 26784, 24896, 25152, 22400, 23904, 22592, 37072, 29008, 25712, 23952, 25392, 24048, 22032, 22000, 28464, 25264, 25488, 24176, 22608, 21904, 21648, 21616, 27360, 23296, 20288, 19840, 23168, 21600, 19488, 19232, 23744, 21856, 21600, 21216, 21600, 20928, 20480, 20416
$Y_4(k)$	147024, 100864, 94128, 82944, 84560, 78176, 70192, 65184, 73136, 60064, 66640, 59232, 60848, 52736, 56272, 52224, 64080, 55744, 52288, 52144, 53600, 52880, 50720, 50544, 45344, 41584, 43296, 42832, 42960, 39872, 41584, 40576, 56848, 43008, 42160, 39232, 43616, 39824, 40416, 36784, 44160, 36336, 39104, 35024, 40288, 37936, 36960, 35472, 42384, 34880, 32816, 32384, 38160, 33056, 32816, 30112, 38032, 34368, 32240, 31424, 35536, 32352, 30640, 29472, 62032, 39168, 42032, 31872, 43408, 36640, 32624, 29792, 41392, 31328, 36816, 30752, 34928, 28544, 32912, 27392, 35088, 26624, 26224, 25728, 31184, 26912, 25904, 25440, 26352, 22688, 23440, 22752, 23600, 22592, 22736, 22336, 38864, 24448, 30000, 22528, 28816, 23712, 25072, 21728, 30336, 23408, 27776, 23824, 26912, 22640, 25184, 22544, 29936, 22432, 23120, 19744, 26608, 22272, 22352, 19776, 24848, 20928, 22448, 20864, 23056, 20384, 21680, 20000
$z_3(n) = X_3(k)+Y_3(k)$	77440, 4808, 5968, 2248, 6080, 2056, 2352, 1352, 7488, 1576, 3696, 1192, 2240, 1000, 1488, 1000, 11904, 1880, 2256, 664, 3648, 1240, 1264, 600, 3136, 888, 1648, 568, 1624, 736, 1040, 568, 16768, 1592, 2000, 728, 3136, 856, 1264, 504, 5056, 568, 1904, 600, 1600, 408, 808, 480, 7552, 1000, 1808, 456, 2496, 520, 880, 232, 2064, 696, 1168, 456, 1136, 376, 568, 224, 21120, 840, 1968, 456, 2112, 488, 976, 360, 3392, 488, 1168, 296, 1032, 192, 616, 464, 4800, 472, 1360, 280, 1920, 376, 624, 312, 2360, 496, 1080, 256, 1096, 288, 456, 336, 8096, 792, 784, 184, 1760, 408, 752, 120, 2368, 200, 1040, 152, 928, 296, 528, 264, 3936, 456, 1104, 296, 2080, 392, 536, 128, 1760, 392, 752, 168, 744, 256, 336, 152
$X_4(k)+Y_4(k)$	300192, 213936, 179616, 157744, 169984, 151664, 135744, 125040, 155648, 133200, 136640, 121808, 118240, 108048, 111584, 106768, 130048, 107536, 99920, 95936, 107952, 100736, 97424, 94176, 96112, 84000, 88848, 83008, 88832, 80976, 86656, 81552, 116160, 87120, 81088, 75280, 86704, 80512, 75600, 69856, 85200, 75168, 74928, 70656, 75664, 71616, 69296, 67744, 86176, 72304, 67104, 65072, 76288, 68528, 64768, 61872, 76160, 69648, 65536, 64272, 69168, 65152, 62608, 61024, 131216, 88768, 83664, 71808, 85840, 73248, 69328, 64736, 83312, 68256, 73520, 65056, 68144, 61440, 64432, 58624, 80080, 62528, 57808, 56128, 64016, 57376, 52048, 49824, 57136, 48224, 50224, 47648, 48752, 44992, 46640, 44928, 75936, 53456, 55712, 46480, 54208, 47760, 47104, 43728, 58800, 48672, 53264, 48000, 49520, 44544, 46832, 44160, 57296, 45728, 43408, 39584, 49776, 43872, 41840, 39008, 48592, 42784, 44048, 42080, 44656, 41312, 42160, 40416
$Z_4(k)$	300192, 213936, 179616, 157744, 169984, 151664, 135744, 125040, 155488, 133360, 136800, 121648, 118400, 107888, 111424, 106928, 129600, 107984, 99328, 96528, 108128, 100560, 98144, 93456, 96192, 83856, 89600, 82320, 88352, 80720, 86304, 82640, 116064, 87216, 80992, 75376, 86528, 80688, 75584, 69872, 85152, 74928, 75168, 70704, 75904, 71664, 69184, 67568, 86144, 72336, 67264, 64912, 76512, 68304, 64608, 62032, 76160, 69648, 65344, 64464, 68768, 65552, 61984, 61648, 130912, 88880, 83552, 72112, 86016, 73264, 69312, 64560, 83616, 68144, 73632, 64752, 67968, 61424, 64448, 58800, 80320, 62480, 57856, 55888, 63904, 57296, 52128, 49936, 56896, 48272, 50176, 47888, 48864, 45072, 46560, 44816, 75424, 53552, 55200, 46192, 53760, 48496, 47040, 44464, 59232, 48496, 53856, 48368, 50048, 43888, 46784, 43376, 57408, 45968, 43520, 40208, 49760, 43664, 41440, 38800, 48064, 42960, 43520, 41872, 45088, 41104, 42912, 40272
Iteration 5	
$X_5(k)$	4980736, 268608, 444416, 137728, 458752, 127936, 202752, 84736, 458752, 111808, 276480, 86784, 150016, 54592, 108544, 60928, 753664, 67648, 165888, 42112, 262144, 60096, 112640, 32896, 212992, 76224, 112640, 44672, 91136, 32832, 80896, 37504, 1146880, 132800, 165888, 50688, 229376, 52288, 112640, 32512, 294912, 43840, 129024, 44288, 112128, 10944, 67584, 26624, 524288, 30656, 108544, 23424, 163840, 30016, 79872, 10624, 122880, 46656, 69632, 31104, 67584, 6080, 53248, 10112, 1310720, 92096, 161792, 27520, 131072, 35648, 75776, 14720, 163840, 45120, 100352, 24960, 44544, 22976, 29696, 17280, 303104, 30912, 116736, 13312, 155648, 21568, 47104, 12544, 146912, 35168, 88608, 9440, 54816, 18592, 26080, 20512, 462848, 59456, 75776, 9344, 102400, 25792, 63488, 128, 81920, 14272, 67584, 13952, 42496, 2624, 34816, 3712, 290816, 22336, 83968, 7168, 159744, 23488, 47104, 2304, 118784, 16576, 63488, 4352, 49152, 5440, 17408, 3072
$Y_5(k)$	4931584, 349472, 321088, 149600, 319488, 136992, 98752, 91872, 499712, 95520, 195008, 67680, 140288, 79136, 81472, 62432, 770048, 170720, 121280, 43552, 204800, 96480, 52288, 42656, 189440, 47840, 99904, 30240, 121856, 56032, 49088, 36256, 999424, 68896, 91712, 44512, 172032, 56608, 46016, 33120, 352256, 23840, 113088, 32224, 89088, 52000, 38976, 29280, 442368, 99040, 121280, 32672, 155648, 37600, 32320, 20768, 140288, 35552, 81472, 23456, 72704, 40160, 26048, 22560, 1392640, 17696, 91712, 30816, 139264, 22304, 48320, 34528, 270336, 9504, 47552, 14432, 90112, 13600, 49984, 37856, 311296, 27360, 55744, 23072, 90112, 30944, 34624, 26272, 154624, 25312, 50752, 20000, 73728, 6880, 30912, 22944, 573440, 40224, 26176, 15840,

	122880, 32032, 30912, 16736, 221184, 19744, 63936, 5600, 73728, 35616, 34624, 24160, 212992, 37600, 55744, 28576, 106496, 21216, 22336, 15648, 107520, 31456, 34368, 19360, 57344, 17632, 20512, 18624
$z_4(n) = X_4(k)+ Y_4(k)$	300192, 213936, 179616, 157744, 169984, 151664, 135744, 125040, 155648, 133200, 136640, 121808, 118240, 108048, 111584, 106768, 130048, 107536, 99920, 95936, 107952, 100736, 97424, 94176, 96112, 84000, 88848, 83008, 88832, 80976, 86656, 81552, 116160, 87120, 81088, 75280, 86704, 80512, 75600, 69856, 85200, 75168, 74928, 70656, 75664, 71616, 69296, 67744, 86176, 72304, 67104, 65072, 76288, 68528, 64768, 61872, 76160, 69648, 65536, 64272, 69168, 65152, 62608, 61024, 131216, 88768, 83664, 71808, 85840, 73248, 69328, 64736, 83312, 68256, 73520, 65056, 68144, 61440, 64432, 58624, 80080, 62528, 57808, 56128, 64016, 57376, 52048, 49824, 57136, 48224, 50224, 47648, 48752, 44992, 46640, 44928, 75936, 53456, 55712, 46480, 54208, 47760, 47104, 43728, 58800, 48672, 53264, 48000, 49520, 44544, 46832, 44160, 57296, 45728, 43408, 39584, 49776, 43872, 41840, 39008, 48592, 42784, 44048, 42080, 44656, 41312, 42160, 40416
$X_5(k)+ Y_5(k)$	9912320, 618080, 765504, 287328, 778240, 264928, 301504, 176608, 958464, 207328, 471488, 154464, 290304, 133728, 190016, 123360, 1523712, 238368, 287168, 85664, 466944, 156576, 164928, 75552, 402432, 124064, 212544, 74912, 212992, 88864, 129984, 73760, 2146304, 201696, 257600, 95200, 401408, 108896, 158656, 65632, 647168, 67680, 242112, 76512, 201216, 62944, 106560, 55904, 966656, 129696, 229824, 56096, 319488, 67616, 112192, 31392, 263168, 82208, 151104, 54560, 140288, 46240, 79296, 32672, 2703360, 109792, 253504, 58336, 270336, 57952, 124096, 49248, 434176, 54624, 147904, 39392, 134656, 36576, 79680, 55136, 614400, 58272, 172480, 36384, 245760, 52512, 81728, 38816, 301536, 60480, 139360, 29440, 128544, 25472, 56992, 43456, 1036288, 99680, 101952, 25184, 225280, 57824, 94400, 16864, 303104, 34016, 131520, 19552, 116224, 38240, 69440, 27872, 503808, 59936, 139712, 35744, 266240, 44704, 69440, 17952, 226304, 48032, 97856, 23712, 106496, 23072, 37920, 21696
$Z_5(k)$	9912320, 618080, 765504, 287328, 778240, 264928, 301504, 176608, 958464, 207328, 471488, 154464, 290304, 133728, 190016, 123360, 1523712, 238368, 287168, 85664, 466944, 156576, 164928, 75552, 402432, 124064, 212544, 74912, 212992, 88864, 129984, 73760, 2146304, 201696, 257600, 95200, 401408, 108896, 158656, 65632, 647168, 67680, 242112, 76512, 201216, 62944, 106560, 55904, 966656, 129696, 229824, 56096, 319488, 67616, 112192, 31392, 263168, 82208, 151104, 54560, 140288, 46240, 79296, 32672, 2703360, 109792, 253504, 58336, 270336, 57952, 124096, 49248, 434176, 54624, 147904, 39392, 134656, 36576, 79680, 55136, 614400, 58272, 172480, 36384, 245760, 52512, 81728, 38816, 301056, 60960, 139840, 28960, 129024, 24992, 56512, 43936, 1036288, 99680, 101952, 25184, 225280, 57824, 94400, 16864, 303104, 34016, 131520, 19552, 116224, 38240, 69440, 27872, 503808, 59936, 139712, 35744, 266240, 44704, 69440, 17952, 226304, 48032, 97856, 23712, 106496, 23072, 36032, 23584
Iteration 6	
$X_6(k)$	19605504, 14473216, 10942464, 9574400, 10934272, 9406464, 8390656, 7665408, 10561536, 9361408, 8960000, 8009728, 7346176, 7079936, 7079936, 6977792, 8443904, 6629376, 6096896, 5605376, 6957056, 6125568, 5978112, 5581056, 6498304, 5429248, 5830656, 5142528, 5871616, 5261312, 5769216, 5248768, 7591936, 5646336, 4982784, 4614144, 5515264, 5208064, 4503552, 4237056, 5253120, 4970496, 4585472, 4560896, 4528128, 4311040, 4139008, 4126976, 5606336, 4791232, 4387904, 4183104, 4879424, 4539456, 4090816, 4062400, 4879424, 4514880, 4262848, 4205504, 4305856, 4199360, 4090944, 4041536, 8855552, 6348800, 5328896, 5111808, 5431296, 4685824, 4698112, 4468992, 5365760, 4726784, 4698112, 4390912, 4251648, 4210688, 4034560, 4001536, 5758976, 4595712, 4042752, 3891200, 4202496, 3899392, 3346432, 3124992, 3940352, 3268608, 3428352, 3186688, 3219456, 2867200, 3059712, 2887936, 4745216, 3713024, 3291136, 3065856, 3250176, 3078144, 2820096, 2812160, 3643392, 3233792, 3262464, 3094528, 2893824, 2803712, 2770944, 2770688, 3502080, 2981888, 2596864, 2539520, 2965504, 2764800, 2494464, 2465536, 3039232, 2797568, 2764800, 2715648, 2764800, 2678784, 2621440, 2609408
$Y_6(k)$	18819072, 12914368, 12048384, 10613056, 10823680, 10002752, 8984576, 8347328, 9361408, 7684416, 8529920, 7585472, 7788544, 6753984, 7202816, 6680896, 8202240, 7131456, 6692864, 6678208, 6860800, 6772416, 6492160, 6465856, 5804032, 5326528, 5541888, 5478720, 5498880, 5099840, 5322752, 5197504, 7276544, 5501248, 5396480, 5025472, 5582848, 5101248, 5173248, 4704576, 5652480, 4654784, 5005312, 4479296, 5156864, 4852032, 4730880, 4544192, 5425152, 4468416, 4200448, 4141376, 4884480, 4227392, 4200448, 3858112, 4868096, 4395328, 4126720, 4026048, 4548608, 4144832, 3921920, 3768640, 7940096, 5009728, 5380096, 4083392, 5556224, 4693696, 4175872, 3809600, 5298176, 4013760, 4712448, 3932480, 4470784, 3649856, 4212736, 3509952, 4491264, 3411648, 3356672, 3289408, 3991552, 3440960, 3315712, 3260096, 3373056, 2900288, 3000320, 2916032, 3020800, 2895552, 2910208, 2855232, 4974592, 3133120, 3840000, 2879808, 3688448, 3031360, 3209216, 2784960, 3883008, 2992448, 3555328, 3053248, 3444736, 2901696, 3223552, 2881856, 3831808, 2867520, 2959360, 2531008, 3405824, 2854592, 2861056, 2527552, 3180544, 2682560, 2873344, 2666816, 2951168, 2605376, 2775040, 2563776
$Z_5(n) = X_5(k)+ Y_5(k)$	9912320, 618080, 765504, 287328, 778240, 264928, 301504, 176608, 958464, 207328, 471488, 154464, 290304, 133728, 190016, 123360, 1523712, 238368, 287168, 85664, 466944, 156576, 164928, 75552, 402432, 124064, 212544, 74912, 212992, 88864, 129984, 73760, 2146304, 201696, 257600, 95200, 401408, 108896, 158656, 65632, 647168, 67680, 242112, 76512, 201216, 62944, 106560, 55904, 966656, 129696, 229824, 56096, 319488, 67616, 112192, 31392, 263168, 82208, 151104, 54560, 140288, 46240, 79296, 32672, 2703360, 109792, 253504, 58336, 270336, 57952, 124096, 49248, 434176, 54624, 147904, 39392, 134656, 36576, 79680, 55136, 614400, 58272, 172480, 36384, 245760, 52512, 81728, 38816, 301536, 60480, 139360, 29440, 128544, 25472, 56992, 43456, 1036288, 99680, 101952, 25184, 225280, 57824, 94400, 16864, 303104, 34016, 131520, 19552, 116224, 38240, 69440, 27872, 503808, 59936, 139712, 35744, 266240, 44704, 69440, 17952, 226304, 48032, 97856, 23712, 106496, 23072, 37920, 21696
$X_6(k)+ Y_6(k)$	38424576, 27387584, 22990848, 20187456, 21757952, 19409216, 17375232, 16012736, 19922944, 17045824, 17489920, 15595200, 15134720, 13833920, 14282752, 13658688, 16646144, 13760832, 12789760, 12283584, 13817856, 12897984, 12470272, 12046912, 12302336, 10755776, 11372544, 10621248, 11370496, 10361152, 11091968, 10446272, 14868480, 11147584, 10379264, 9639616, 11098112, 10309312, 9676800, 8941632, 10905600, 9625280, 9590784, 9040192, 9684992, 9163072, 8869888, 8671168, 11031488, 9259648, 8588352, 8324480, 9763904, 8766848, 8291264, 7920512, 9747520, 8910208, 8389568, 8231552, 8854464, 8344192, 8012864, 7810176, 16795648, 11358528, 10708992, 9195200, 10987520, 9379520, 8873984, 8278592, 10663936, 8740544, 9410560, 8323392, 8722432, 7860544, 8247296, 7511488, 10250240, 8007360, 7399424, 7180608, 8194048, 7340352, 6662144, 6385088, 7313408, 6168896, 6428672, 6102720, 6240256, 5762752, 5969920, 5743168, 9719808, 6846144, 7131136, 5945664, 6938624, 6109504, 6029312, 5597120, 7526400, 6226240, 6817792, 6147776, 6338560, 5705408, 5994496, 5652544, 7333888, 5849408, 5556224, 5070528, 6371328, 5619392, 5355520, 4993088, 6219776, 5480128, 5638144, 5382464, 5715968, 5284160, 5396480, 5173184

$Z_6(k)$	38424576, 27387584, 22990848, 20187456, 21757952, 19409216, 17375232, 16012736, 19922944, 17045824, 17489920, 15595200, 15134720, 13833920, 14282752, 13658688, 16646144, 13760832, 12789760, 12283584, 13817856, 12897984, 12470272, 12046912, 12302336, 10755776, 11372544, 10621248, 11370496, 10361152, 11091968, 10446272, 14868480, 11147584, 10379264, 9639616, 11098112, 10309312, 9676800, 8941632, 10905600, 9625280, 9590784, 9040192, 9684992, 9163072, 8869888, 8671168, 11030528, 9258688, 8589312, 8325440, 9764864, 8767808, 8290304, 7919552, 9748480, 8911168, 8388608, 8230592, 8853504, 8343232, 8013824, 7811136, 16795648, 11358528, 10708992, 9195200, 10987520, 9379520, 8873984, 8278592, 10663936, 8740544, 9410560, 8323392, 8722432, 7860544, 8247296, 7511488, 10250240, 8007360, 7399424, 7180608, 8194048, 7340352, 6662144, 6385088, 7313408, 6168896, 6428672, 6102720, 6240256, 5762752, 5969920, 5743168, 9719808, 6846144, 7131136, 5945664, 6938624, 6109504, 6029312, 5597120, 7526400, 6226240, 6817792, 6147776, 6338560, 5705408, 5994496, 5652544, 7333888, 5849408, 5556224, 5070528, 6371328, 5619392, 5355520, 4993088, 6219776, 5480128, 5638144, 5382464, 5715968, 5284160, 5396480, 5173184
Iteration 7 (Linear Pair Forming Property is satisfied)	
$X_7(k)$	637534208, 34381824, 56885248, 17629184, 58720256, 16375808, 25952256, 10846208, 58720256, 14311424, 35389440, 11108352, 19202048, 6987776, 13908992, 7798784, 96468992, 8658944, 21233664, 5390336, 33554432, 7692288, 14417920, 4210688, 27262976, 9756672, 14417920, 5718016, 11665408, 4202496, 10339328, 4800512, 146800640, 16998400, 21233664, 6488064, 29360128, 6692864, 14417920, 4161536, 37748736, 5611520, 16515072, 5668864, 14352384, 1400832, 8635392, 3407872, 67108864, 3923968, 13893632, 2998272, 20971520, 3842048, 10223616, 1359872, 15728640, 5971968, 8912896, 3981312, 8650752, 778240, 6831104, 1294336, 167772160, 11788288, 20709376, 3522560, 16777216, 4562944, 9699328, 1884160, 20971520, 5775360, 12845056, 3194880, 5701632, 2940928, 3816448, 2211840, 38797312, 3956736, 14942208, 1703936, 19922944, 2760704, 6029312, 1605632, 18804736, 4501504, 11341824, 1208320, 7016448, 2379776, 3322880, 2625536, 59244544, 7610368, 9699328, 1196032, 13107200, 3301376, 8126464, 16384, 10485760, 1826816, 8650752, 1785856, 5439488, 335872, 4441088, 475136, 37224448, 2859008, 10747904, 917504, 20447232, 3006464, 6029312, 294912, 15204352, 2121728, 8126464, 557056, 6291456, 696320, 2243584, 393216
$Y_7(k)$	631242752, 44732416, 41099264, 19148800, 40894464, 17534976, 12640256, 11759616, 63963136, 12226560, 24961024, 8663040, 17956864, 10129408, 10428416, 7991296, 98566144, 21852160, 15523840, 5574656, 26214400, 12349440, 6692864, 5459968, 24248320, 6123520, 12787712, 3870720, 15597568, 7172096, 6283264, 4640768, 127926272, 8818688, 11739136, 5697536, 22020096, 7245824, 5890048, 4239360, 45088768, 3051520, 14475264, 4124672, 11403264, 6656000, 4988928, 3747840, 56623104, 12677120, 15523840, 4182016, 19922944, 4812800, 4136960, 2658304, 17956864, 4550656, 10428416, 3002368, 9306112, 5140480, 3334144, 2887680, 178257920, 2265088, 11739136, 3944448, 17825792, 2854912, 6184960, 4419584, 34603008, 1216512, 6086656, 1847296, 11534336, 1740800, 6397952, 4845568, 39845888, 3502080, 7135232, 2953216, 11534336, 3960832, 4431872, 3362816, 19791872, 3239936, 6496256, 2560000, 9437184, 880640, 3956736, 2936832, 73400320, 5148672, 3350528, 2027520, 15728640, 4100096, 3956736, 2142208, 28311552, 2527232, 8183808, 716800, 9437184, 4558848, 4431872, 3092480, 27262976, 4812800, 7135232, 3657728, 13631488, 2715648, 2859008, 2002944, 13762560, 4026368, 4399104, 2478080, 7340032, 2256896, 2625536, 2383872
$Z_6(n) = X_6(k)+ Y_6(k)$	38424576, 27387584, 22990848, 20187456, 21757952, 19409216, 17375232, 16012736, 19922944, 17045824, 17489920, 15595200, 15134720, 13833920, 14282752, 13658688, 16646144, 13760832, 12789760, 12283584, 13817856, 12897984, 12470272, 12046912, 12302336, 10755776, 11372544, 10621248, 11370496, 10361152, 11091968, 10446272, 14868480, 11147584, 10379264, 9639616, 11098112, 10309312, 9676800, 8941632, 10905600, 9625280, 9590784, 9040192, 9684992, 9163072, 8869888, 8671168, 11031488, 9259648, 8588352, 8324480, 9763904, 8766848, 8291264, 7920512, 9747520, 8910208, 8389568, 8231552, 8854464, 8344192, 8012864, 7810176, 16795648, 11358528, 10708992, 9195200, 10987520, 9379520, 8873984, 8278592, 10663936, 8740544, 9410560, 8323392, 8722432, 7860544, 8247296, 7511488, 10250240, 8007360, 7399424, 7180608, 8194048, 7340352, 6662144, 6385088, 7313408, 6168896, 6428672, 6102720, 6240256, 5762752, 5969920, 5743168, 9719808, 6846144, 7131136, 5945664, 6938624, 6109504, 6029312, 5597120, 7526400, 6226240, 6817792, 6147776, 6338560, 5705408, 5994496, 5652544, 7333888, 5849408, 5556224, 5070528, 6371328, 5619392, 5355520, 4993088, 6219776, 5480128, 5638144, 5382464, 5715968, 5284160, 5396480, 5173184
$X_7(k) + Y_7(k)$	1268776960, 79114240, 97984512, 36777984, 99614720, 33910784, 38592512, 22605824, 122683392, 26537984, 60350464, 19771392, 37158912, 17117184, 24337408, 15790080, 195035136, 30511104, 36757504, 10964992, 59768832, 20041728, 21110784, 9670656, 51511296, 15880192, 27205632, 9588736, 27262976, 11374592, 16622592, 9441280, 274726912, 25817088, 32972800, 12185600, 51380224, 13938688, 20307968, 8400896, 82837504, 8663040, 30990336, 9793536, 25755648, 8056832, 13624320, 7155712, 123731968, 16601088, 29417472, 7180288, 40894464, 8654848, 14360576, 4018176, 33685504, 10522624, 19341312, 6983680, 17956864, 5918720, 10165248, 4182016, 346030080, 14053376, 32448512, 7467008, 34603008, 7417856, 15884288, 6303744, 55574528, 6991872, 18931712, 5042176, 17235968, 4681728, 10214400, 7057408, 78643200, 7458816, 22077440, 4657152, 31457280, 6721536, 10461184, 4968448, 38596608, 7741440, 17838080, 3768320, 16453632, 3260416, 7279616, 5562368, 132644864, 12759040, 13049856, 3223552, 28835840, 7401472, 12083200, 2158592, 38797312, 4354048, 16834560, 2502656, 14876672, 4894720, 8872960, 3567616, 64487424, 7671808, 17883136, 4575232, 34078720, 5722112, 8888320, 2297856, 28966912, 6148096, 12525568, 3035136, 13631488, 2953216, 4869120, 2777088
$Z_7(k)$	1268776960, 79114240, 97984512, 36777984, 99614720, 33910784, 38592512, 22605824, 122683392, 26537984, 60350464, 19771392, 37158912, 17117184, 24337408, 15790080, 195035136, 30511104, 36757504, 10964992, 59768832, 20041728, 21110784, 9670656, 51511296, 15880192, 27205632, 9588736, 27262976, 11374592, 16622592, 9441280, 274726912, 25817088, 32972800, 12185600, 51380224, 13938688, 20307968, 8400896, 82837504, 8663040, 30990336, 9793536, 25755648, 8056832, 13624320, 7155712, 123731968, 16601088, 29417472, 7180288, 40894464, 8654848, 14360576, 4018176, 33685504, 10522624, 19341312, 6983680, 17956864, 5918720, 10165248, 4182016, 346030080, 14053376, 32448512, 7467008, 34603008, 7417856, 15884288, 6303744, 55574528, 6991872, 18931712, 5042176, 17235968, 4681728, 10214400, 7057408, 78643200, 7458816, 22077440, 4657152, 31457280, 6721536, 10461184, 4968448, 38596608, 7741440, 17838080, 3768320, 16453632, 3260416, 7279616, 5562368, 132644864, 12759040, 13049856, 3223552, 28835840, 7401472, 12083200, 2158592, 38797312, 4354048, 16834560, 2502656, 14876672, 4894720, 8872960, 3567616, 64487424, 7671808, 17883136, 4575232, 34078720, 5722112, 8888320, 2297856, 28966912, 6148096, 12525568, 3035136, 13631488, 2953216, 4869120, 2777088

Observation

1. It is found that $Z_7(k) = X_7(k) + Y_7(k)$ and thus the CASN is 7. One can repeat this experiment for various genome pairs.

4.5 RAJAN TRANSFORM SPECTRA BASED REGENERATION OF SHORT LENGTH DUAL PAIRS

For Anti-Mutation and Anti-Crossovers of SARS-Corona Virus Genome Sequences

As stated already, a sequence of length 2^n, where n=1,2,3,4… and its possible dual will have the same RT spectrum. A potent dual pair yields another sequence, which may or may not be potent to form a pair with another sequence. Such a sequence is called 'sterile' sequence. Given a long genome sequence, one can search for various short length potent sequences and identify their dual sequences. These dual forms would couple with potent sequences and neutralize them. This is a significant development achieved in this research. One can a priori consider all possible dual pairs of lengths 2^n, where n=2,3,4…., and carry out pattern matching of potent forms of dual pairs and try to neutralize segments of a genome.

Algorithm for identifying dual pairs

The algorithm for identifying all possible dual pairs in a sequence is the same as the one stated previously as a theorem but with a small additional constrint.

Algorithm

A pair of short length sequences of same length is called a dual pair if and only if arithmetic sum of the numbers in the sequence is either 10 or an integer multiple of 10. In any case, a dual form is obtained from a potent sequence by subtracting each number in sequence from 10. Resulting sequence is dual form of the potent sequence. This algorithm is restricted only to genome sequences.

Case Study

Consider sequences of length 4. The contents of a sequence are limited to 1, 2, 3 and 4 which represent adenine (A), thymine (T), guanine (G) and cytosine (C) respectively. One can construct 256 sequences of length 4 using the symbols A, T, G and C. They are called 4-tuple codons. It was found out of 256 such 4-tuple codons, one can precisely have 44 potent 4-tuple codons and 22 pairs of 4-tuple codons as dual pairs. They are listed below. This amounts to saying that 17.18% of total 4-tuple codons are potent.

List of 22 dual pairs of number equivalents of nucleotides

<1144, 4411>	<1234, 4321>	<1243, 4312>	<1324, 4231>	<1333, 4222>	<1342, 4213>
<1414, 4141>	<1423, 4132>	<1432, 4123>	<1441, 4114>	<2134, 3421>	<2143, 3412>
<2224, 3331>	<2233, 3322>	**<2242, 3313>**	<2314, 3241>	<2323, 3232>	<2332, 3223>
<2341, 3214>	<2413, 3142>	**<2422, 3133>**	<2431, 3124>		

List of 22 dual pairs of nucleotides

<AACC, CCAA>	<ATGC, CGTA>	<ATCG, CGAT>	<AGTC, CTGA>	<AGGG, CTTT>	<AGCT, CTAG>
<ACAC, CACA>	<ACTG, CAGT>	<ACGT, CATG>	<ACCA, CAAC>	<TAGC, GCTA>	<TACG, GCAT>
<TTTC, GGGA>	<TTGG, GGTT>	**<TTCT, GGAG>**	<TGAC, GTCA>	<TGTG, GTGT>	<TGGT, GTTG>
<TGCA, GTAC>	<TCAG, GACT>	**<TCTT, GAGG>**	<TCGA, GATC>		

Consider sequences of length 8. The contents of a sequence is limited to 1, 2, 3 and 4 which represent adenine (A), thymine (T), guanine (G) and cytosine (C) respectively. One can construct 65536 sequences of length 8 using the symbols A, T, G and C. They are called 8-tuple codons. It was found out of 65536 such 8-tuple codons, one can precisely have 8164 potent 8-tuple codons and 4082 pairs of 8-tuple codons as dual pairs. They are listed below. This means that 12.45% of total 8-tuple codons are potent.

List of 4082 dual pairs of number equivalents of nucleotides

<11111113, 44444442>	<11111122, 44444433>	<11111131, 44444424>	<11111212, 44444343>
<11111221, 44444334>	<11111311, 44444244>	<11112112, 44443443>	<11112121, 44443434>
<11112211, 44443344>	<11113111, 44442444>	<11114444, 44441111>	<11121112, 44434443>
<11121121, 44434434>	<11121211, 44434344>	<11122111, 44433444>	<11123444, 44432111>
<11124344, 44431211>	<11124434, 44431121>	<11124443, 44431112>	<11131111, 44424444>
<11132444, 44423111>	<11133344, 44422211>	<11133434, 44422121>	<11133443, 44422112>
<11134244, 44421311>	<11134334, 44421221>	<11134343, 44421212>	<11134424, 44421131>
<11134433, 44421122>	<11134442, 44421113>	<11141444, 44414111>	<11142344, 44413211>
<11142434, 44413121>	<11142443, 44413112>	<11143244, 44412311>	<11143334, 44412221>
<11143343, 44412212>	<11143424, 44412131>	<11143433, 44412122>	<11143442, 44412113>
<11144144, 44411411>	<11144234, 44411321>	<11144243, 44411312>	<11144324, 44411231>
<11144333, 44411222>	<11144342, 44411213>	<11144414, 44411141>	<11144423, 44411132>
<11144432, 44411123>	<11144441, 44411114>	<11211112, 44344443>	<11211121, 44344434>
<11211211, 44344344>	<11212111, 44343444>	<11213444, 44342111>	<11214344, 44341211>
<11214434, 44341121>	<11214443, 44341112>	<11221111, 44334444>	<11222444, 44333111>
<11223344, 44332211>	<11223434, 44332121>	<11223443, 44332112>	<11224244, 44331311>
<11224334, 44331221>	<11224343, 44331212>	<11224424, 44331131>	<11224433, 44331122>
<11224442, 44331113>	<11231444, 44324111>	<11232344, 44323211>	<11232434, 44323121>
<11232443, 44323112>	<11233244, 44322311>	<11233334, 44322221>	<11233343, 44322212>
<11233424, 44322131>	<11233433, 44322122>	<11233442, 44322113>	<11234144, 44321411>
<11234234, 44321321>	<11234243, 44321312>	<11234324, 44321231>	<11234333, 44321222>
<11234342, 44321213>	<11234414, 44321141>	<11234423, 44321132>	<11234432, 44321123>
<11234441, 44321114>	<11241344, 44314211>	<11241434, 44314121>	<11241443, 44314112>
<11242244, 44313311>	<11242334, 44313221>	<11242343, 44313212>	<11242424, 44313131>
<11242433, 44313122>	<11242442, 44313113>	<11243144, 44312411>	<11243234, 44312321>
<11243243, 44312312>	<11243324, 44312231>	<11243333, 44312222>	<11243342, 44312213>
<11243414, 44312141>	<11243423, 44312132>	<11243432, 44312123>	<11243441, 44312114>
<11244134, 44311421>	<11244143, 44311412>	<11244224, 44311331>	<11244233, 44311322>
<11244242, 44311313>	<11244314, 44311241>	<11244323, 44311232>	<11244332, 44311223>
<11244341, 44311214>	<11244413, 44311142>	<11244422, 44311133>	<11244431, 44311124>

<11311111, 44244444> <11312444, 44243111> <11313344, 44242211> <11313434, 44242121>
<11313443, 44242112> <11314244, 44241311> <11314334, 44241221> <11314343, 44241212>
<11314424, 44241131> <11314433, 44241122> <11314442, 44241113> <11321444, 44234111>
<11322344, 44233211> <11322434, 44233121> <11322443, 44233112> <11323244, 44232311>
<11323334, 44232221> <11323343, 44232212> <11323424, 44232131> <11323433, 44232122>
<11323442, 44232113> <11324144, 44231411> <11324234, 44231321> <11324243, 44231312>
<11324324, 44231231> <11324333, 44231222> <11324342, 44231213> <11324414, 44231141>
<11324423, 44231132> <11324432, 44231123> <11324441, 44231114> <11331344, 44224211>
<11331434, 44224121> <11331443, 44224112> <11332244, 44223311> <11332334, 44223221>
<11332343, 44223212> <11332424, 44223131> <11332433, 44223122> <11332442, 44223113>
<11333144, 44222411> <11333234, 44222321> <11333243, 44222312> <11333324, 44222231>
<11333333, 44222222> <11333342, 44222213> <11333414, 44222141> <11333423, 44222132>
<11333432, 44222123> <11333441, 44222114> <11334134, 44221421> <11334143, 44221412>
<11334224, 44221331> <11334233, 44221322> <11334242, 44221313> <11334314, 44221241>
<11334323, 44221232> <11334332, 44221223> <11334341, 44221214> <11334413, 44221142>
<11334422, 44221133> <11334431, 44221124> <11341244, 44214311> <11341334, 44214221>
<11341343, 44214212> <11341424, 44214131> <11341433, 44214122> <11341442, 44214113>
<11342144, 44213411> <11342234, 44213321> <11342243, 44213312> <11342324, 44213231>
<11342333, 44213222> <11342342, 44213213> <11342414, 44213141> <11342423, 44213132>
<11342432, 44213123> <11342441, 44213114> <11343134, 44212421> <11343143, 44212412>
<11343224, 44212331> <11343233, 44212322> <11343242, 44212313> <11343314, 44212241>
<11343323, 44212232> <11343332, 44212223> <11343341, 44212214> <11343413, 44212142>
<11343422, 44212133> <11343431, 44212124> <11344124, 44211431> <11344133, 44211422>
<11344142, 44211413> <11344214, 44211341> <11344223, 44211332> <11344232, 44211323>
<11344241, 44211314> <11344313, 44211242> <11344322, 44211233> <11344331, 44211224>
<11344412, 44211143> <11344421, 44211134> <11411444, 44144111> <11412344, 44143211>
<11412434, 44143121> <11412443, 44143112> <11413244, 44142311> <11413334, 44142221>
<11413343, 44142212> <11413424, 44142131> <11413433, 44142122> <11413442, 44142113>
<11414144, 44141411> <11414234, 44141321> <11414243, 44141312> <11414324, 44141231>
<11414333, 44141222> <11414342, 44141213> <11414414, 44141141> <11414423, 44141132>
<11414432, 44141123> <11414441, 44141114> <11421344, 44134211> <11421434, 44134121>
<11421443, 44134112> <11422244, 44133311> <11422334, 44133221> <11422343, 44133212>
<11422424, 44133131> <11422433, 44133122> <11422442, 44133113> <11423144, 44132411>
<11423234, 44132321> <11423243, 44132312> <11423324, 44132231> <11423333, 44132222>
<11423342, 44132213> <11423414, 44132141> <11423423, 44132132> <11423432, 44132123>
<11423441, 44132114> <11424134, 44131421> <11424143, 44131412> <11424224, 44131331>
<11424233, 44131322> <11424242, 44131313> <11424314, 44131241> <11424323, 44131232>
<11424332, 44131223> <11424341, 44131214> <11424413, 44131142> <11424422, 44131133>
<11424431, 44131124> <11431244, 44124311> <11431334, 44124221> <11431343, 44124212>
<11431424, 44124131> <11431433, 44124122> <11431442, 44124113> <11432144, 44123411>
<11432234, 44123321> <11432243, 44123312> <11432324, 44123231> <11432333, 44123222>
<11432342, 44123213> <11432414, 44123141> <11432423, 44123132> <11432432, 44123123>
<11432441, 44123114> <11433134, 44122421> <11433143, 44122412> <11433224, 44122331>
<11433233, 44122322> <11433242, 44122313> <11433314, 44122241> <11433323, 44122232>
<11433332, 44122223> <11433341, 44122214> <11433413, 44122142> <11433422, 44122133>
<11433431, 44122124> <11434124, 44121431> <11434133, 44121422> <11434142, 44121413>
<11434214, 44121341> <11434223, 44121332> <11434232, 44121323> <11434241, 44121314>
<11434313, 44121242> <11434322, 44121233> <11434331, 44121224> <11434412, 44121143>
<11434421, 44121134> <11441144, 44114411> <11441234, 44114321> <11441243, 44114312>
<11441324, 44114231> <11441333, 44114222> <11441342, 44114213> <11441414, 44114141>
<11441423, 44114132> <11441432, 44114123> <11441441, 44114114> <11442134, 44113421>
<11442143, 44113412> <11442224, 44113331> <11442233, 44113322> <11442242, 44113313>
<11442314, 44113241> <11442323, 44113232> <11442332, 44113223> <11442341, 44113214>
<11442413, 44113142> <11442422, 44113133> <11442431, 44113124> <11443124, 44112431>
<11443133, 44112422> <11443142, 44112413> <11443214, 44112341> <11443223, 44112332>
<11443232, 44112323> <11443241, 44112314> <11443313, 44112242> <11443322, 44112233>
<11443331, 44112224> <11443412, 44112143> <11443421, 44112134> <11444114, 44111441>
<11444123, 44111432> <11444132, 44111423> <11444141, 44111414> <11444213, 44111342>
<11444222, 44111333> <11444231, 44111324> <11444312, 44111243> <11444321, 44111234>
<11444411, 44111144> <12111112, 43444443> <12111121, 43444434> <12111211, 43444344>
<12112111, 43443444> <12113444, 43442111> <12114344, 43441211> <12114434, 43441121>
<12114443, 43441112> <12121111, 43434444> <12122444, 43433111> <12123344, 43432211>
<12123434, 43432121> <12123443, 43432112> <12124244, 43431311> <12124334, 43431221>
<12124343, 43431212> <12124424, 43431131> <12124433, 43431122> <12124442, 43431113>
<12131444, 43424111> <12132344, 43423211> <12132434, 43423121> <12132443, 43423112>
<12133244, 43422311> <12133334, 43422221> <12133343, 43422212> <12133424, 43422131>
<12133433, 43422122> <12133442, 43422113> <12134144, 43421411> <12134234, 43421321>
<12134243, 43421312> <12134324, 43421231> <12134333, 43421222> <12134342, 43421213>
<12134414, 43421141> <12134423, 43421132> <12134432, 43421123> <12134441, 43421114>
<12141344, 43414211> <12141434, 43414121> <12141443, 43414112> <12142244, 43413311>

<12142334, 43413221>	<12142343, 43413212>	<12142424, 43413131>	<12142433, 43413122>
<12142442, 43413113>	<12143144, 43412411>	<12143234, 43412321>	<12143243, 43412312>
<12143324, 43412231>	<12143333, 43412222>	<12143342, 43412213>	<12143414, 43412141>
<12143423, 43412132>	<12143432, 43412123>	<12143441, 43412114>	<12144134, 43411421>
<12144143, 43411412>	<12144224, 43411331>	<12144233, 43411322>	<12144242, 43411313>
<12144314, 43411241>	<12144323, 43411232>	<12144332, 43411223>	<12144341, 43411214>
<12144413, 43411142>	<12144422, 43411133>	<12144431, 43411124>	<12211111, 43344444>
<12212444, 43343111>	<12213344, 43342211>	<12213434, 43342121>	<12213443, 43342112>
<12214244, 43341311>	<12214334, 43341221>	<12214343, 43341212>	<12214424, 43341131>
<12214433, 43341122>	<12214442, 43341113>	<12221444, 43334111>	<12222344, 43333211>
<12222434, 43333121>	<12222443, 43333112>	<12223244, 43332311>	<12223334, 43332221>
<12223343, 43332212>	<12223424, 43332131>	<12223433, 43332122>	<12223442, 43332113>
<12224144, 43331411>	<12224234, 43331321>	<12224243, 43331312>	<12224324, 43331231>
<12224333, 43331222>	<12224342, 43331213>	<12224414, 43331141>	<12224423, 43331132>
<12224432, 43331123>	<12224441, 43331114>	<12231344, 43324211>	<12231434, 43324121>
<12231443, 43324112>	<12232244, 43323311>	<12232334, 43323221>	<12232343, 43323212>
<12232424, 43323131>	<12232433, 43323122>	<12232442, 43323113>	<12233144, 43322411>
<12233234, 43322321>	<12233243, 43322312>	<12233324, 43322231>	<12233333, 43322222>
<12233342, 43322213>	<12233414, 43322141>	<12233423, 43322132>	<12233432, 43322123>
<12233441, 43322114>	<12234134, 43321421>	<12234143, 43321412>	<12234224, 43321331>
<12234233, 43321322>	<12234242, 43321313>	<12234314, 43321241>	<12234323, 43321232>
<12234332, 43321223>	<12234341, 43321214>	<12234413, 43321142>	<12234422, 43321133>
<12234431, 43321124>	<12241244, 43314311>	<12241334, 43314221>	<12241343, 43314212>
<12241424, 43314131>	<12241433, 43314122>	<12241442, 43314113>	<12242144, 43313411>
<12242234, 43313321>	<12242243, 43313312>	<12242324, 43313231>	<12242333, 43313222>
<12242342, 43313213>	<12242414, 43313141>	<12242423, 43313132>	<12242432, 43313123>
<12242441, 43313114>	<12243134, 43312421>	<12243143, 43312412>	<12243224, 43312331>
<12243233, 43312322>	<12243242, 43312313>	<12243314, 43312241>	<12243323, 43312232>
<12243332, 43312223>	<12243341, 43312214>	<12243413, 43312142>	<12243422, 43312133>
<12243431, 43312124>	<12244124, 43311431>	<12244133, 43311422>	<12244142, 43311413>
<12244214, 43311341>	<12244223, 43311332>	<12244232, 43311323>	<12244241, 43311314>
<12244313, 43311242>	<12244322, 43311233>	<12244331, 43311224>	<12244412, 43311143>
<12244421, 43311134>	<12311444, 43244111>	<12312344, 43243211>	<12312434, 43243121>
<12312443, 43243112>	<12313244, 43242311>	<12313334, 43242221>	<12313343, 43242212>
<12313424, 43242131>	<12313433, 43242122>	<12313442, 43242113>	<12314144, 43241411>
<12314234, 43241321>	<12314243, 43241312>	<12314324, 43241231>	<12314333, 43241222>
<12314342, 43241213>	<12314414, 43241141>	<12314423, 43241132>	<12314432, 43241123>
<12314441, 43241114>	<12321344, 43234211>	<12321434, 43234121>	<12321443, 43234112>
<12322244, 43233311>	<12322334, 43233221>	<12322343, 43233212>	<12322424, 43233131>
<12322433, 43233122>	<12322442, 43233113>	<12323144, 43232411>	<12323234, 43232321>
<12323243, 43232312>	<12323324, 43232231>	<12323333, 43232222>	<12323342, 43232213>
<12323414, 43232141>	<12323423, 43232132>	<12323432, 43232123>	<12323441, 43232114>
<12324134, 43231421>	<12324143, 43231412>	<12324224, 43231331>	<12324233, 43231322>
<12324242, 43231313>	<12324314, 43231241>	<12324323, 43231232>	<12324332, 43231223>
<12324341, 43231214>	<12324413, 43231142>	<12324422, 43231133>	<12324431, 43231124>
<12331244, 43224311>	<12331334, 43224221>	<12331343, 43224212>	<12331424, 43224131>
<12331433, 43224122>	<12331442, 43224113>	<12332144, 43223411>	<12332234, 43223321>
<12332243, 43223312>	<12332324, 43223231>	<12332333, 43223222>	<12332342, 43223213>
<12332414, 43223141>	<12332423, 43223132>	<12332432, 43223123>	<12332441, 43223114>
<12333134, 43222421>	<12333143, 43222412>	<12333224, 43222331>	<12333233, 43222322>
<12333242, 43222313>	<12333314, 43222241>	<12333323, 43222232>	<12333332, 43222223>
<12333341, 43222214>	<12333413, 43222142>	<12333422, 43222133>	<12333431, 43222124>
<12334124, 43221431>	<12334133, 43221422>	<12334142, 43221413>	<12334214, 43221341>
<12334223, 43221332>	<12334232, 43221323>	<12334241, 43221314>	<12334313, 43221242>
<12334322, 43221233>	<12334331, 43221224>	<12334412, 43221143>	<12334421, 43221134>
<12341144, 43214411>	<12341234, 43214321>	<12341243, 43214312>	<12341324, 43214231>
<12341333, 43214222>	<12341342, 43214213>	<12341414, 43214141>	<12341423, 43214132>
<12341432, 43214123>	<12341441, 43214114>	<12342134, 43213421>	<12342143, 43213412>
<12342224, 43213331>	<12342233, 43213322>	<12342242, 43213313>	<12342314, 43213241>
<12342323, 43213232>	<12342332, 43213223>	<12342341, 43213214>	<12342413, 43213142>
<12342422, 43213133>	<12342431, 43213124>	<12343124, 43212431>	<12343133, 43212422>
<12343142, 43212413>	<12343214, 43212341>	<12343223, 43212332>	<12343232, 43212323>
<12343241, 43212314>	<12343313, 43212242>	<12343322, 43212233>	<12343331, 43212224>
<12343412, 43212143>	<12343421, 43212134>	<12344114, 43211441>	<12344123, 43211432>
<12344132, 43211423>	<12344141, 43211414>	<12344213, 43211342>	<12344222, 43211333>
<12344231, 43211324>	<12344312, 43211243>	<12344321, 43211234>	<12344411, 43211144>
<12411344, 43144211>	<12411434, 43144121>	<12411443, 43144112>	<12412244, 43143311>
<12412334, 43143221>	<12412343, 43143212>	<12412424, 43143131>	<12412433, 43143122>
<12412442, 43143113>	<12413144, 43142411>	<12413234, 43142321>	<12413243, 43142312>
<12413324, 43142231>	<12413333, 43142222>	<12413342, 43142213>	<12413414, 43142141>

<12413423, 43142132> <12413432, 43142123> <12413441, 43142114> <12414134, 43141421>
<12414143, 43141412> <12414224, 43141331> <12414233, 43141322> <12414242, 43141313>
<12414314, 43141241> <12414323, 43141232> <12414332, 43141223> <12414341, 43141214>
<12414413, 43141142> <12414422, 43141133> <12414431, 43141124> <12421244, 43134311>
<12421334, 43134221> <12421343, 43134212> <12421424, 43134131> <12421433, 43134122>
<12421442, 43134113> <12422144, 43133411> <12422234, 43133321> <12422243, 43133312>
<12422324, 43133231> <12422333, 43133222> <12422342, 43133213> <12422414, 43133141>
<12422423, 43133132> <12422432, 43133123> <12422441, 43133114> <12423134, 43132421>
<12423143, 43132412> <12423224, 43132331> <12423233, 43132322> <12423242, 43132313>
<12423314, 43132241> <12423323, 43132232> <12423332, 43132223> <12423341, 43132214>
<12423413, 43132142> <12423422, 43132133> <12423431, 43132124> <12424124, 43131431>
<12424133, 43131422> <12424142, 43131413> <12424214, 43131341> <12424223, 43131332>
<12424232, 43131323> <12424241, 43131314> <12424313, 43131242> <12424322, 43131233>
<12424331, 43131224> <12424412, 43131143> <12424421, 43131134> <12431144, 43124411>
<12431234, 43124321> <12431243, 43124312> <12431324, 43124231> <12431333, 43124222>
<12431342, 43124213> <12431414, 43124141> <12431423, 43124132> <12431432, 43124123>
<12431441, 43124114> <12432134, 43123421> <12432143, 43123412> <12432224, 43123331>
<12432233, 43123322> <12432242, 43123313> <12432314, 43123241> <12432323, 43123232>
<12432332, 43123223> <12432341, 43123214> <12432413, 43123142> <12432422, 43123133>
<12432431, 43123124> <12433124, 43122431> <12433133, 43122422> <12433142, 43122413>
<12433214, 43122341> <12433223, 43122332> <12433232, 43122323> <12433241, 43122314>
<12433313, 43122242> <12433322, 43122233> <12433331, 43122224> <12433412, 43122143>
<12433421, 43122134> <12434114, 43121441> <12434123, 43121432> <12434132, 43121423>
<12434141, 43121414> <12434213, 43121342> <12434222, 43121333> <12434231, 43121324>
<12434312, 43121243> <12434321, 43121234> <12434411, 43121144> <12441134, 43114421>
<12441143, 43114412> <12441224, 43114331> <12441233, 43114322> <12441242, 43114313>
<12441314, 43114241> <12441323, 43114232> <12441332, 43114223> <12441341, 43114214>
<12441413, 43114142> <12441422, 43114133> <12441431, 43114124> <12442124, 43113431>
<12442133, 43113422> <12442142, 43113413> <12442214, 43113341> <12442223, 43113332>
<12442232, 43113323> <12442241, 43113314> <12442313, 43113242> <12442322, 43113233>
<12442331, 43113224> <12442412, 43113143> <12442421, 43113134> <12443114, 43112441>
<12443123, 43112432> <12443132, 43112423> <12443141, 43112414> <12443213, 43112342>
<12443222, 43112333> <12443231, 43112324> <12443312, 43112243> <12443321, 43112234>
<12443411, 43112144> <12444113, 43111442> <12444122, 43111433> <12444131, 43111424>
<12444212, 43111343> <12444221, 43111334> <12444311, 43111244> <13111111, 42444444>
<13112444, 42443111> <13113344, 42442211> <13113434, 42442121> <13113443, 42442112>
<13114244, 42441311> <13114334, 42441221> <13114343, 42441212> <13114424, 42441131>
<13114433, 42441122> <13114442, 42441113> <13121444, 42434111> <13122344, 42433211>
<13122434, 42433121> <13122443, 42433112> <13123244, 42432311> <13123334, 42432221>
<13123343, 42432212> <13123424, 42432131> <13123433, 42432122> <13123442, 42432113>
<13124144, 42431411> <13124234, 42431321> <13124243, 42431312> <13124324, 42431231>
<13124333, 42431222> <13124342, 42431213> <13124414, 42431141> <13124423, 42431132>
<13124432, 42431123> <13124441, 42431114> <13131344, 42424211> <13131434, 42424121>
<13131443, 42424112> <13132244, 42423311> <13132334, 42423221> <13132343, 42423212>
<13132424, 42423131> <13132433, 42423122> <13132442, 42423113> <13133144, 42422411>
<13133234, 42422321> <13133243, 42422312> <13133324, 42422231> <13133333, 42422222>
<13133342, 42422213> <13133414, 42422141> <13133423, 42422132> <13133432, 42422123>
<13133441, 42422114> <13134134, 42421421> <13134143, 42421412> <13134224, 42421331>
<13134233, 42421322> <13134242, 42421313> <13134314, 42421241> <13134323, 42421232>
<13134332, 42421223> <13134341, 42421214> <13134413, 42421142> <13134422, 42421133>
<13134431, 42421124> <13141244, 42414311> <13141334, 42414221> <13141343, 42414212>
<13141424, 42414131> <13141433, 42414122> <13141442, 42414113> <13142144, 42413411>
<13142234, 42413321> <13142243, 42413312> <13142324, 42413231> <13142333, 42413222>
<13142342, 42413213> <13142414, 42413141> <13142423, 42413132> <13142432, 42413123>
<13142441, 42413114> <13143134, 42412421> <13143143, 42412412> <13143224, 42412331>
<13143233, 42412322> <13143242, 42412313> <13143314, 42412241> <13143323, 42412232>
<13143332, 42412223> <13143341, 42412214> <13143413, 42412142> <13143422, 42412133>
<13143431, 42412124> <13144124, 42411431> <13144133, 42411422> <13144142, 42411413>
<13144214, 42411341> <13144223, 42411332> <13144232, 42411323> <13144241, 42411314>
<13144313, 42411242> <13144322, 42411233> <13144331, 42411224> <13144412, 42411143>
<13144421, 42411134> <13211444, 42344111> <13212344, 42343211> <13212434, 42343121>
<13212443, 42343112> <13213244, 42342311> <13213334, 42342221> <13213343, 42342212>
<13213424, 42342131> <13213433, 42342122> <13213442, 42342113> <13214144, 42341411>
<13214234, 42341321> <13214243, 42341312> <13214324, 42341231> <13214333, 42341222>
<13214342, 42341213> <13214414, 42341141> <13214423, 42341132> <13214432, 42341123>
<13214441, 42341114> <13221344, 42334211> <13221434, 42334121> <13221443, 42334112>
<13222244, 42333311> <13222334, 42333221> <13222343, 42333212> <13222424, 42333131>
<13222433, 42333122> <13222442, 42333113> <13223144, 42332411> <13223234, 42332321>
<13223243, 42332312> <13223324, 42332231> <13223333, 42332222> <13223342, 42332213>
<13223414, 42332141> <13223423, 42332132> <13223432, 42332123> <13223441, 42332114>

<13224134, 42331421>
<13224143, 42331412>
<13224224, 42331331>
<13224233, 42331322>
<13224242, 42331313>
<13224314, 42331241>
<13224323, 42331232>
<13224332, 42331223>
<13224341, 42331214>
<13224413, 42331142>
<13224422, 42331133>
<13224431, 42331124>
<13231244, 42324311>
<13231334, 42324221>
<13231343, 42324212>
<13231424, 42324131>
<13231433, 42324122>
<13231442, 42324113>
<13232144, 42323411>
<13232234, 42323321>
<13232243, 42323312>
<13232324, 42323231>
<13232333, 42323222>
<13232342, 42323213>
<13232414, 42323141>
<13232423, 42323132>
<13232432, 42323123>
<13232441, 42323114>
<13233134, 42322421>
<13233143, 42322412>
<13233224, 42322331>
<13233233, 42322322>
<13233242, 42322313>
<13233314, 42322241>
<13233323, 42322232>
<13233332, 42322223>
<13233341, 42322214>
<13233413, 42322142>
<13233422, 42322133>
<13233431, 42322124>
<13234124, 42321431>
<13234133, 42321422>
<13234142, 42321413>
<13234214, 42321341>
<13234223, 42321332>
<13234232, 42321323>
<13234241, 42321314>
<13234313, 42321242>
<13234322, 42321233>
<13234331, 42321224>
<13234412, 42321143>
<13234421, 42321134>
<13241144, 42314411>
<13241234, 42314321>
<13241243, 42314312>
<13241324, 42314231>
<13241333, 42314222>
<13241342, 42314213>
<13241414, 42314141>
<13241423, 42314132>
<13241432, 42314123>
<13241441, 42314114>
<13242134, 42313421>
<13242143, 42313412>
<13242224, 42313331>
<13242233, 42313322>
<13242242, 42313313>
<13242314, 42313241>
<13242323, 42313232>
<13242332, 42313223>
<13242341, 42313214>
<13242413, 42313142>
<13242422, 42313133>
<13242431, 42313124>
<13243124, 42312431>
<13243133, 42312422>
<13243142, 42312413>
<13243214, 42312341>
<13243223, 42312332>
<13243232, 42312323>
<13243241, 42312314>
<13243313, 42312242>
<13243322, 42312233>
<13243331, 42312224>
<13243412, 42312143>
<13243421, 42312134>
<13244114, 42311441>
<13244123, 42311432>
<13244132, 42311423>
<13244141, 42311414>
<13244213, 42311342>
<13244222, 42311333>
<13244231, 42311324>
<13244312, 42311243>
<13244321, 42311234>
<13244411, 42311144>
<13311344, 42244211>
<13311434, 42244121>
<13311443, 42244112>
<13312244, 42243311>
<13312334, 42243221>
<13312343, 42243212>
<13312424, 42243131>
<13312433, 42243122>
<13312442, 42243113>
<13313144, 42242411>
<13313234, 42242321>
<13313243, 42242312>
<13313324, 42242231>
<13313333, 42242222>
<13313342, 42242213>
<13313414, 42242141>
<13313423, 42242132>
<13313432, 42242123>
<13313441, 42242114>
<13314134, 42241421>
<13314143, 42241412>
<13314224, 42241331>
<13314233, 42241322>
<13314242, 42241313>
<13314314, 42241241>
<13314323, 42241232>
<13314332, 42241223>
<13314341, 42241214>
<13314413, 42241142>
<13314422, 42241133>
<13314431, 42241124>
<13321244, 42234311>
<13321334, 42234221>
<13321343, 42234212>
<13321424, 42234131>
<13321433, 42234122>
<13321442, 42234113>
<13322144, 42233411>
<13322234, 42233321>
<13322243, 42233312>
<13322324, 42233231>
<13322333, 42233222>
<13322342, 42233213>
<13322414, 42233141>
<13322423, 42233132>
<13322432, 42233123>
<13322441, 42233114>
<13323134, 42232421>
<13323143, 42232412>
<13323224, 42232331>
<13323233, 42232322>
<13323242, 42232313>
<13323314, 42232241>
<13323323, 42232232>
<13323332, 42232223>
<13323341, 42232214>
<13323413, 42232142>
<13323422, 42232133>
<13323431, 42232124>
<13324124, 42231431>
<13324133, 42231422>
<13324142, 42231413>
<13324214, 42231341>
<13324223, 42231332>
<13324232, 42231323>
<13324241, 42231314>
<13324313, 42231242>
<13324322, 42231233>
<13324331, 42231224>
<13324412, 42231143>
<13324421, 42231134>
<13331144, 42224411>
<13331234, 42224321>
<13331243, 42224312>
<13331324, 42224231>
<13331333, 42224222>
<13331342, 42224213>
<13331414, 42224141>
<13331423, 42224132>
<13331432, 42224123>
<13331441, 42224114>
<13332134, 42223421>
<13332143, 42223412>
<13332224, 42223331>
<13332233, 42223322>
<13332242, 42223313>
<13332314, 42223241>
<13332323, 42223232>
<13332332, 42223223>
<13332341, 42223214>
<13332413, 42223142>
<13332422, 42223133>
<13332431, 42223124>
<13333124, 42222431>
<13333133, 42222422>
<13333142, 42222413>
<13333214, 42222341>
<13333223, 42222332>
<13333232, 42222323>
<13333241, 42222314>
<13333313, 42222242>
<13333322, 42222233>
<13333331, 42222224>
<13333412, 42222143>
<13333421, 42222134>
<13334114, 42221441>
<13334123, 42221432>
<13334132, 42221423>
<13334141, 42221414>
<13334213, 42221342>
<13334222, 42221333>
<13334231, 42221324>
<13334312, 42221243>
<13334321, 42221234>
<13334411, 42221144>
<13341134, 42214421>
<13341143, 42214412>
<13341224, 42214331>
<13341233, 42214322>
<13341242, 42214313>
<13341314, 42214241>
<13341323, 42214232>
<13341332, 42214223>
<13341341, 42214214>
<13341413, 42214142>
<13341422, 42214133>
<13341431, 42214124>
<13342124, 42213431>
<13342133, 42213422>
<13342142, 42213413>
<13342214, 42213341>
<13342223, 42213332>
<13342232, 42213323>
<13342241, 42213314>
<13342313, 42213242>
<13342322, 42213233>
<13342331, 42213224>
<13342412, 42213143>
<13342421, 42213134>
<13343114, 42212441>
<13343123, 42212432>
<13343132, 42212423>
<13343141, 42212414>
<13343213, 42212342>
<13343222, 42212333>
<13343231, 42212324>
<13343312, 42212243>
<13343321, 42212234>
<13343411, 42212144>
<13344113, 42211442>
<13344122, 42211433>
<13344131, 42211424>
<13344212, 42211343>
<13344221, 42211334>
<13344311, 42211244>
<13411244, 42144311>
<13411334, 42144221>
<13411343, 42144212>
<13411424, 42144131>
<13411433, 42144122>
<13411442, 42144113>
<13412144, 42143411>
<13412234, 42143321>
<13412243, 42143312>
<13412324, 42143231>
<13412333, 42143222>
<13412342, 42143213>
<13412414, 42143141>
<13412423, 42143132>
<13412432, 42143123>
<13412441, 42143114>
<13413134, 42142421>
<13413143, 42142412>
<13413224, 42142331>
<13413233, 42142322>
<13413242, 42142313>
<13413314, 42142241>
<13413323, 42142232>
<13413332, 42142223>
<13413341, 42142214>
<13413413, 42142142>
<13413422, 42142133>
<13413431, 42142124>
<13414124, 42141431>

<13414133, 42141422>	<13414142, 42141413>	<13414214, 42141341>	<13414223, 42141332>
<13414232, 42141323>	<13414241, 42141314>	<13414313, 42141242>	<13414322, 42141233>
<13414331, 42141224>	<13414412, 42141143>	<13414421, 42141134>	<13421144, 42134411>
<13421234, 42134321>	<13421243, 42134312>	<13421324, 42134231>	<13421333, 42134222>
<13421342, 42134213>	<13421414, 42134141>	<13421423, 42134132>	<13421432, 42134123>
<13421441, 42134114>	<13422134, 42133421>	<13422143, 42133412>	<13422224, 42133331>
<13422233, 42133322>	<13422242, 42133313>	<13422314, 42133241>	<13422323, 42133232>
<13422332, 42133223>	<13422341, 42133214>	<13422413, 42133142>	<13422422, 42133133>
<13422431, 42133124>	<13423124, 42132431>	<13423133, 42132422>	<13423142, 42132413>
<13423214, 42132341>	<13423223, 42132332>	<13423232, 42132323>	<13423241, 42132314>
<13423313, 42132242>	<13423322, 42132233>	<13423331, 42132224>	<13423412, 42132143>
<13423421, 42132134>	<13424114, 42131441>	<13424123, 42131432>	<13424132, 42131423>
<13424141, 42131414>	<13424213, 42131342>	<13424222, 42131333>	<13424231, 42131324>
<13424312, 42131243>	<13424321, 42131234>	<13424411, 42131144>	<13431134, 42124421>
<13431143, 42124412>	<13431224, 42124331>	<13431233, 42124322>	<13431242, 42124313>
<13431314, 42124241>	<13431323, 42124232>	<13431332, 42124223>	<13431341, 42124214>
<13431413, 42124142>	<13431422, 42124133>	<13431431, 42124124>	<13432124, 42123431>
<13432133, 42123422>	<13432142, 42123413>	<13432214, 42123341>	<13432223, 42123332>
<13432232, 42123323>	<13432241, 42123314>	<13432313, 42123242>	<13432322, 42123233>
<13432331, 42123224>	<13432412, 42123143>	<13432421, 42123134>	<13433114, 42122441>
<13433123, 42122432>	<13433132, 42122423>	<13433141, 42122414>	<13433213, 42122342>
<13433222, 42122333>	<13433231, 42122324>	<13433312, 42122243>	<13433321, 42122234>
<13433411, 42122144>	<13434113, 42121442>	<13434122, 42121433>	<13434131, 42121424>
<13434212, 42121343>	<13434221, 42121334>	<13434311, 42121244>	<13441124, 42114431>
<13441133, 42114422>	<13441142, 42114413>	<13441214, 42114341>	<13441223, 42114332>
<13441232, 42114323>	<13441241, 42114314>	<13441313, 42114242>	<13441322, 42114233>
<13441331, 42114224>	<13441412, 42114143>	<13441421, 42114134>	<13442114, 42113441>
<13442123, 42113432>	<13442132, 42113423>	<13442141, 42113414>	<13442213, 42113342>
<13442222, 42113333>	<13442231, 42113324>	<13442312, 42113243>	<13442321, 42113234>
<13442411, 42113144>	<13443113, 42112442>	<13443122, 42112433>	<13443131, 42112424>
<13443212, 42112343>	<13443221, 42112334>	<13443311, 42112244>	<13444112, 42111443>
<13444121, 42111434>	<13444211, 42111344>	<14111444, 41444111>	<14112344, 41443211>
<14112434, 41443121>	<14112443, 41443112>	<14113244, 41442311>	<14113334, 41442221>
<14113343, 41442212>	<14113424, 41442131>	<14113433, 41442122>	<14113442, 41442113>
<14114144, 41441411>	<14114234, 41441321>	<14114243, 41441312>	<14114324, 41441231>
<14114333, 41441222>	<14114342, 41441213>	<14114414, 41441141>	<14114423, 41441132>
<14114432, 41441123>	<14114441, 41441114>	<14121344, 41434211>	<14121434, 41434121>
<14121443, 41434112>	<14122244, 41433311>	<14122334, 41433221>	<14122343, 41433212>
<14122424, 41433131>	<14122433, 41433122>	<14122442, 41433113>	<14123144, 41432411>
<14123234, 41432321>	<14123243, 41432312>	<14123324, 41432231>	<14123333, 41432222>
<14123342, 41432213>	<14123414, 41432141>	<14123423, 41432132>	<14123432, 41432123>
<14123441, 41432114>	<14124134, 41431421>	<14124143, 41431412>	<14124224, 41431331>
<14124233, 41431322>	<14124242, 41431313>	<14124314, 41431241>	<14124323, 41431232>
<14124332, 41431223>	<14124341, 41431214>	<14124413, 41431142>	<14124422, 41431133>
<14124431, 41431124>	<14131244, 41424311>	<14131334, 41424221>	<14131343, 41424212>
<14131424, 41424131>	<14131433, 41424122>	<14131442, 41424113>	<14132144, 41423411>
<14132234, 41423321>	<14132243, 41423312>	<14132324, 41423231>	<14132333, 41423222>
<14132342, 41423213>	<14132414, 41423141>	<14132423, 41423132>	<14132432, 41423123>
<14132441, 41423114>	<14133134, 41422421>	<14133143, 41422412>	<14133224, 41422331>
<14133233, 41422322>	<14133242, 41422313>	<14133314, 41422241>	<14133323, 41422232>
<14133332, 41422223>	<14133341, 41422214>	<14133413, 41422142>	<14133422, 41422133>
<14133431, 41422124>	<14134124, 41421431>	<14134133, 41421422>	<14134142, 41421413>
<14134214, 41421341>	<14134223, 41421332>	<14134232, 41421323>	<14134241, 41421314>
<14134313, 41421242>	<14134322, 41421233>	<14134331, 41421224>	<14134412, 41421143>
<14134421, 41421134>	<14141144, 41414411>	<14141234, 41414321>	<14141243, 41414312>
<14141324, 41414231>	<14141333, 41414222>	<14141342, 41414213>	<14141414, 41414141>
<14141423, 41414132>	<14141432, 41414123>	<14141441, 41414114>	<14142134, 41413421>
<14142143, 41413412>	<14142224, 41413331>	<14142233, 41413322>	<14142242, 41413313>
<14142314, 41413241>	<14142323, 41413232>	<14142332, 41413223>	<14142341, 41413214>
<14142413, 41413142>	<14142422, 41413133>	<14142431, 41413124>	<14143124, 41412431>
<14143133, 41412422>	<14143142, 41412413>	<14143214, 41412341>	<14143223, 41412332>
<14143232, 41412323>	<14143241, 41412314>	<14143313, 41412242>	<14143322, 41412233>
<14143331, 41412224>	<14143412, 41412143>	<14143421, 41412134>	<14144114, 41411441>
<14144123, 41411432>	<14144132, 41411423>	<14144141, 41411414>	<14144213, 41411342>
<14144222, 41411333>	<14144231, 41411324>	<14144312, 41411243>	<14144321, 41411234>
<14144411, 41411144>	<14211344, 41344211>	<14211434, 41344121>	<14211443, 41344112>
<14212244, 41343311>	<14212334, 41343221>	<14212343, 41343212>	<14212424, 41343131>
<14212433, 41343122>	<14212442, 41343113>	<14213144, 41342411>	<14213234, 41342321>
<14213243, 41342312>	<14213324, 41342231>	<14213333, 41342222>	<14213342, 41342213>
<14213414, 41342141>	<14213423, 41342132>	<14213432, 41342123>	<14213441, 41342114>

<14214134, 41341421> <14214143, 41341412> <14214224, 41341331> <14214233, 41341322>
<14214242, 41341313> <14214314, 41341241> <14214323, 41341232> <14214332, 41341223>
<14214341, 41341214> <14214413, 41341142> <14214422, 41341133> <14214431, 41341124>
<14221244, 41334311> <14221334, 41334221> <14221343, 41334212> <14221424, 41334131>
<14221433, 41334122> <14221442, 41334113> <14222144, 41333411> <14222234, 41333321>
<14222243, 41333312> <14222324, 41333231> <14222333, 41333222> <14222342, 41333213>
<14222414, 41333141> <14222423, 41333132> <14222432, 41333123> <14222441, 41333114>
<14223134, 41332421> <14223143, 41332412> <14223224, 41332331> <14223233, 41332322>
<14223242, 41332313> <14223314, 41332241> <14223323, 41332232> <14223332, 41332223>
<14223341, 41332214> <14223413, 41332142> <14223422, 41332133> <14223431, 41332124>
<14224124, 41331431> <14224133, 41331422> <14224142, 41331413> <14224214, 41331341>
<14224223, 41331332> <14224232, 41331323> <14224241, 41331314> <14224313, 41331242>
<14224322, 41331233> <14224331, 41331224> <14224412, 41331143> <14224421, 41331134>
<14231144, 41324411> <14231234, 41324321> <14231243, 41324312> <14231324, 41324231>
<14231333, 41324222> <14231342, 41324213> <14231414, 41324141> <14231423, 41324132>
<14231432, 41324123> <14231441, 41324114> <14232134, 41323421> <14232143, 41323412>
<14232224, 41323331> <14232233, 41323322> <14232242, 41323313> <14232314, 41323241>
<14232323, 41323232> <14232332, 41323223> <14232341, 41323214> <14232413, 41323142>
<14232422, 41323133> <14232431, 41323124> <14233124, 41322431> <14233133, 41322422>
<14233142, 41322413> <14233214, 41322341> <14233223, 41322332> <14233232, 41322323>
<14233241, 41322314> <14233313, 41322242> <14233322, 41322233> <14233331, 41322224>
<14233412, 41322143> <14233421, 41322134> <14234114, 41321441> <14234123, 41321432>
<14234132, 41321423> <14234141, 41321414> <14234213, 41321342> <14234222, 41321333>
<14234231, 41321324> <14234312, 41321243> <14234321, 41321234> <14234411, 41321144>
<14241134, 41314421> <14241143, 41314412> <14241224, 41314331> <14241233, 41314322>
<14241242, 41314313> <14241314, 41314241> <14241323, 41314232> <14241332, 41314223>
<14241341, 41314214> <14241413, 41314142> <14241422, 41314133> <14241431, 41314124>
<14242124, 41313431> <14242133, 41313422> <14242142, 41313413> <14242214, 41313341>
<14242223, 41313332> <14242232, 41313323> <14242241, 41313314> <14242313, 41313242>
<14242322, 41313233> <14242331, 41313224> <14242412, 41313143> <14242421, 41313134>
<14243114, 41312441> <14243123, 41312432> <14243132, 41312423> <14243141, 41312414>
<14243213, 41312342> <14243222, 41312333> <14243231, 41312324> <14243312, 41312243>
<14243321, 41312234> <14243411, 41312144> <14244113, 41311442> <14244122, 41311433>
<14244131, 41311424> <14244212, 41311343> <14244221, 41311334> <14244311, 41311244>
<14311244, 41244311> <14311334, 41244221> <14311343, 41244212> <14311424, 41244131>
<14311433, 41244122> <14311442, 41244113> <14312144, 41243411> <14312234, 41243321>
<14312243, 41243312> <14312324, 41243231> <14312333, 41243222> <14312342, 41243213>
<14312414, 41243141> <14312423, 41243132> <14312432, 41243123> <14312441, 41243114>
<14313134, 41242421> <14313143, 41242412> <14313224, 41242331> <14313233, 41242322>
<14313242, 41242313> <14313314, 41242241> <14313323, 41242232> <14313332, 41242223>
<14313341, 41242214> <14313413, 41242142> <14313422, 41242133> <14313431, 41242124>
<14314124, 41241431> <14314133, 41241422> <14314142, 41241413> <14314214, 41241341>
<14314223, 41241332> <14314232, 41241323> <14314241, 41241314> <14314313, 41241242>
<14314322, 41241233> <14314331, 41241224> <14314412, 41241143> <14314421, 41241134>
<14321144, 41234411> <14321234, 41234321> <14321243, 41234312> <14321324, 41234231>
<14321333, 41234222> <14321342, 41234213> <14321414, 41234141> <14321423, 41234132>
<14321432, 41234123> <14321441, 41234114> <14322134, 41233421> <14322143, 41233412>
<14322224, 41233331> <14322233, 41233322> <14322242, 41233313> <14322314, 41233241>
<14322323, 41233232> <14322332, 41233223> <14322341, 41233214> <14322413, 41233142>
<14322422, 41233133> <14322431, 41233124> <14323124, 41232431> <14323133, 41232422>
<14323142, 41232413> <14323214, 41232341> <14323223, 41232332> <14323232, 41232323>
<14323241, 41232314> <14323313, 41232242> <14323322, 41232233> <14323331, 41232224>
<14323412, 41232143> <14323421, 41232134> <14324114, 41231441> <14324123, 41231432>
<14324132, 41231423> <14324141, 41231414> <14324213, 41231342> <14324222, 41231333>
<14324231, 41231324> <14324312, 41231243> <14324321, 41231234> <14324411, 41231144>
<14331134, 41224421> <14331143, 41224412> <14331224, 41224331> <14331233, 41224322>
<14331242, 41224313> <14331314, 41224241> <14331323, 41224232> <14331332, 41224223>
<14331341, 41224214> <14331413, 41224142> <14331422, 41224133> <14331431, 41224124>
<14332124, 41223431> <14332133, 41223422> <14332142, 41223413> <14332214, 41223341>
<14332223, 41223332> <14332232, 41223323> <14332241, 41223314> <14332313, 41223242>
<14332322, 41223233> <14332331, 41223224> <14332412, 41223143> <14332421, 41223134>
<14333114, 41222441> <14333123, 41222432> <14333132, 41222423> <14333141, 41222414>
<14333213, 41222342> <14333222, 41222333> <14333231, 41222324> <14333312, 41222243>
<14333321, 41222234> <14333411, 41222144> <14334113, 41221442> <14334122, 41221433>
<14334131, 41221424> <14334212, 41221343> <14334221, 41221334> <14334311, 41221244>
<14341124, 41214431> <14341133, 41214422> <14341142, 41214413> <14341214, 41214341>
<14341223, 41214332> <14341232, 41214323> <14341241, 41214314> <14341313, 41214242>
<14341322, 41214233> <14341331, 41214224> <14341412, 41214143> <14341421, 41214134>
<14342114, 41213441> <14342123, 41213432> <14342132, 41213423> <14342141, 41213414>
<14342213, 41213342> <14342222, 41213333> <14342231, 41213324> <14342312, 41213243>

<14342321, 41213234> <14342411, 41213144> <14343113, 41212442> <14343122, 41212433>
<14343131, 41212424> <14343212, 41212343> <14343221, 41212334> <14343311, 41212244>
<14344112, 41211443> <14344121, 41211434> <14344211, 41211344> <14411144, 41144411>
<14411234, 41144321> <14411243, 41144312> <14411324, 41144231> <14411333, 41144222>
<14411342, 41144213> <14411414, 41144141> <14411423, 41144132> <14411432, 41144123>
<14411441, 41144114> <14412134, 41143421> <14412143, 41143412> <14412224, 41143331>
<14412233, 41143322> <14412242, 41143313> <14412314, 41143241> <14412323, 41143232>
<14412332, 41143223> <14412341, 41143214> <14412413, 41143142> <14412422, 41143133>
<14412431, 41143124> <14413124, 41142431> <14413133, 41142422> <14413142, 41142413>
<14413214, 41142341> <14413223, 41142332> <14413232, 41142323> <14413241, 41142314>
<14413313, 41142242> <14413322, 41142233> <14413331, 41142224> <14413412, 41142143>
<14413421, 41142134> <14414114, 41141441> <14414123, 41141432> <14414132, 41141423>
<14414141, 41141414> <14414213, 41141342> <14414222, 41141333> <14414231, 41141324>
<14414312, 41141243> <14414321, 41141234> <14414411, 41141144> <14421134, 41134421>
<14421143, 41134412> <14421224, 41134331> <14421233, 41134322> <14421242, 41134313>
<14421314, 41134241> <14421323, 41134232> <14421332, 41134223> <14421341, 41134214>
<14421413, 41134142> <14421422, 41134133> <14421431, 41134124> <14422124, 41133431>
<14422133, 41133422> <14422142, 41133413> <14422214, 41133341> <14422223, 41133332>
<14422232, 41133323> <14422241, 41133314> <14422313, 41133242> <14422322, 41133233>
<14422331, 41133224> <14422412, 41133143> <14422421, 41133134> <14423114, 41132441>
<14423123, 41132432> <14423132, 41132423> <14423141, 41132414> <14423213, 41132342>
<14423222, 41132333> <14423231, 41132324> <14423312, 41132243> <14423321, 41132234>
<14423411, 41132144> <14424113, 41131442> <14424122, 41131433> <14424131, 41131424>
<14424212, 41131343> <14424221, 41131334> <14424311, 41131244> <14431124, 41124431>
<14431133, 41124422> <14431142, 41124413> <14431214, 41124341> <14431223, 41124332>
<14431232, 41124323> <14431241, 41124314> <14431313, 41124242> <14431322, 41124233>
<14431331, 41124224> <14431412, 41124143> <14431421, 41124134> <14432114, 41123441>
<14432123, 41123432> <14432132, 41123423> <14432141, 41123414> <14432213, 41123342>
<14432222, 41123333> <14432231, 41123324> <14432312, 41123243> <14432321, 41123234>
<14432411, 41123144> <14433113, 41122442> <14433122, 41122433> <14433131, 41122424>
<14433212, 41122343> <14433221, 41122334> <14433311, 41122244> <14434112, 41121443>
<14434121, 41121434> <14434211, 41121344> <14441114, 41114441> <14441123, 41114432>
<14441132, 41114423> <14441141, 41114414> <14441213, 41114342> <14441222, 41114333>
<14441231, 41114324> <14441312, 41114243> <14441321, 41114234> <14441411, 41114144>
<14442113, 41113442> <14442122, 41113433> <14442131, 41113424> <14442212, 41113343>
<14442221, 41113334> <14442311, 41113244> <14443112, 41112443> <14443121, 41112434>
<14443211, 41112344> <14444111, 41111444> <21111112, 34444443> <21111121, 34444434>
<21111211, 34444344> <21112111, 34443444> <21113444, 34442111> <21114344, 34441211>
<21114434, 34441121> <21114443, 34441112> <21121111, 34434444> <21122444, 34433111>
<21123344, 34432211> <21123434, 34432121> <21123443, 34432112> <21124244, 34431311>
<21124334, 34431221> <21124343, 34431212> <21124424, 34431131> <21124433, 34431122>
<21124442, 34431113> <21131444, 34424111> <21132344, 34423211> <21132434, 34423121>
<21132443, 34423112> <21133244, 34422311> <21133334, 34422221> <21133343, 34422212>
<21133424, 34422131> <21133433, 34422122> <21133442, 34422113> <21134144, 34421411>
<21134234, 34421321> <21134243, 34421312> <21134324, 34421231> <21134333, 34421222>
<21134342, 34421213> <21134414, 34421141> <21134423, 34421132> <21134432, 34421123>
<21134441, 34421114> <21141344, 34414211> <21141434, 34414121> <21141443, 34414112>
<21142244, 34413311> <21142334, 34413221> <21142343, 34413212> <21142424, 34413131>
<21142433, 34413122> <21142442, 34413113> <21143144, 34412411> <21143234, 34412321>
<21143243, 34412312> <21143324, 34412231> <21143333, 34412222> <21143342, 34412213>
<21143414, 34412141> <21143423, 34412132> <21143432, 34412123> <21143441, 34412114>
<21144134, 34411421> <21144143, 34411412> <21144224, 34411331> <21144233, 34411322>
<21144242, 34411313> <21144314, 34411241> <21144323, 34411232> <21144332, 34411223>
<21144341, 34411214> <21144413, 34411142> <21144422, 34411133> <21144431, 34411124>
<21211111, 34344444> <21212444, 34343111> <21213344, 34342211> <21213434, 34342121>
<21213443, 34342112> <21214244, 34341311> <21214334, 34341221> <21214343, 34341212>
<21214424, 34341131> <21214433, 34341122> <21214442, 34341113> <21221444, 34334111>
<21222344, 34333211> <21222434, 34333121> <21222443, 34333112> <21223244, 34332311>
<21223334, 34332221> <21223343, 34332212> <21223424, 34332131> <21223433, 34332122>
<21223442, 34332113> <21224144, 34331411> <21224234, 34331321> <21224243, 34331312>
<21224324, 34331231> <21224333, 34331222> <21224342, 34331213> <21224414, 34331141>
<21224423, 34331132> <21224432, 34331123> <21224441, 34331114> <21231344, 34324211>
<21231434, 34324121> <21231443, 34324112> <21232244, 34323311> <21232334, 34323221>
<21232343, 34323212> <21232424, 34323131> <21232433, 34323122> <21232442, 34323113>
<21233144, 34322411> <21233234, 34322321> <21233243, 34322312> <21233324, 34322231>
<21233333, 34322222> <21233342, 34322213> <21233414, 34322141> <21233423, 34322132>
<21233432, 34322123> <21233441, 34322114> <21234134, 34321421> <21234143, 34321412>
<21234224, 34321331> <21234233, 34321322> <21234242, 34321313> <21234314, 34321241>
<21234323, 34321232> <21234332, 34321223> <21234341, 34321214> <21234413, 34321142>
<21234422, 34321133> <21234431, 34321124> <21241244, 34314311> <21241334, 34314221>

<21241343, 34314212>
<21241424, 34314131>
<21241433, 34314122>
<21241442, 34314113>
<21242144, 34313411>
<21242234, 34313321>
<21242243, 34313312>
<21242324, 34313231>
<21242333, 34313222>
<21242342, 34313213>
<21242414, 34313141>
<21242423, 34313132>
<21242432, 34313123>
<21242441, 34313114>
<21243134, 34312421>
<21243143, 34312412>
<21243224, 34312331>
<21243233, 34312322>
<21243242, 34312313>
<21243314, 34312241>
<21243323, 34312232>
<21243332, 34312223>
<21243341, 34312214>
<21243413, 34312142>
<21243422, 34312133>
<21243431, 34312124>
<21244124, 34311431>
<21244133, 34311422>
<21244142, 34311413>
<21244214, 34311341>
<21244223, 34311332>
<21244232, 34311323>
<21244241, 34311314>
<21244313, 34311242>
<21244322, 34311233>
<21244331, 34311224>
<21244412, 34311143>
<21244421, 34311134>
<21311444, 34244111>
<21312344, 34243211>
<21312434, 34243121>
<21312443, 34243112>
<21313244, 34242311>
<21313334, 34242221>
<21313343, 34242212>
<21313424, 34242131>
<21313433, 34242122>
<21313442, 34242113>
<21314144, 34241411>
<21314234, 34241321>
<21314243, 34241312>
<21314324, 34241231>
<21314333, 34241222>
<21314342, 34241213>
<21314414, 34241141>
<21314423, 34241132>
<21314432, 34241123>
<21314441, 34241114>
<21321344, 34234211>
<21321434, 34234121>
<21321443, 34234112>
<21322244, 34233311>
<21322334, 34233221>
<21322343, 34233212>
<21322424, 34233131>
<21322433, 34233122>
<21322442, 34233113>
<21323144, 34232411>
<21323234, 34232321>
<21323243, 34232312>
<21323324, 34232231>
<21323333, 34232222>
<21323342, 34232213>
<21323414, 34232141>
<21323423, 34232132>
<21323432, 34232123>
<21323441, 34232114>
<21324134, 34231421>
<21324143, 34231412>
<21324224, 34231331>
<21324233, 34231322>
<21324242, 34231313>
<21324314, 34231241>
<21324323, 34231232>
<21324332, 34231223>
<21324341, 34231214>
<21324413, 34231142>
<21324422, 34231133>
<21324431, 34231124>
<21331244, 34224311>
<21331334, 34224221>
<21331343, 34224212>
<21331424, 34224131>
<21331433, 34224122>
<21331442, 34224113>
<21332144, 34223411>
<21332234, 34223321>
<21332243, 34223312>
<21332324, 34223231>
<21332333, 34223222>
<21332342, 34223213>
<21332414, 34223141>
<21332423, 34223132>
<21332432, 34223123>
<21332441, 34223114>
<21333134, 34222421>
<21333143, 34222412>
<21333224, 34222331>
<21333233, 34222322>
<21333242, 34222313>
<21333314, 34222241>
<21333323, 34222232>
<21333332, 34222223>
<21333341, 34222214>
<21333413, 34222142>
<21333422, 34222133>
<21333431, 34222124>
<21334124, 34221431>
<21334133, 34221422>
<21334142, 34221413>
<21334214, 34221341>
<21334223, 34221332>
<21334232, 34221323>
<21334241, 34221314>
<21334313, 34221242>
<21334322, 34221233>
<21334331, 34221224>
<21334412, 34221143>
<21334421, 34221134>
<21341144, 34214411>
<21341234, 34214321>
<21341243, 34214312>
<21341324, 34214231>
<21341333, 34214222>
<21341342, 34214213>
<21341414, 34214141>
<21341423, 34214132>
<21341432, 34214123>
<21341441, 34214114>
<21342134, 34213421>
<21342143, 34213412>
<21342224, 34213331>
<21342233, 34213322>
<21342242, 34213313>
<21342314, 34213241>
<21342323, 34213232>
<21342332, 34213223>
<21342341, 34213214>
<21342413, 34213142>
<21342422, 34213133>
<21342431, 34213124>
<21343124, 34212431>
<21343133, 34212422>
<21343142, 34212413>
<21343214, 34212341>
<21343223, 34212332>
<21343232, 34212323>
<21343241, 34212314>
<21343313, 34212242>
<21343322, 34212233>
<21343331, 34212224>
<21343412, 34212143>
<21343421, 34212134>
<21344114, 34211441>
<21344123, 34211432>
<21344132, 34211423>
<21344141, 34211414>
<21344213, 34211342>
<21344222, 34211333>
<21344231, 34211324>
<21344312, 34211243>
<21344321, 34211234>
<21344411, 34211144>
<21411344, 34144211>
<21411434, 34144121>
<21411443, 34144112>
<21412244, 34143311>
<21412334, 34143221>
<21412343, 34143212>
<21412424, 34143131>
<21412433, 34143122>
<21412442, 34143113>
<21413144, 34142411>
<21413234, 34142321>
<21413243, 34142312>
<21413324, 34142231>
<21413333, 34142222>
<21413342, 34142213>
<21413414, 34142141>
<21413423, 34142132>
<21413432, 34142123>
<21413441, 34142114>
<21414134, 34141421>
<21414143, 34141412>
<21414224, 34141331>
<21414233, 34141322>
<21414242, 34141313>
<21414314, 34141241>
<21414323, 34141232>
<21414332, 34141223>
<21414341, 34141214>
<21414413, 34141142>
<21414422, 34141133>
<21414431, 34141124>
<21421244, 34134311>
<21421334, 34134221>
<21421343, 34134212>
<21421424, 34134131>
<21421433, 34134122>
<21421442, 34134113>
<21422144, 34133411>
<21422234, 34133321>
<21422243, 34133312>
<21422324, 34133231>
<21422333, 34133222>
<21422342, 34133213>
<21422414, 34133141>
<21422423, 34133132>
<21422432, 34133123>
<21422441, 34133114>
<21423134, 34132421>
<21423143, 34132412>
<21423224, 34132331>
<21423233, 34132322>
<21423242, 34132313>
<21423314, 34132241>
<21423323, 34132232>
<21423332, 34132223>
<21423341, 34132214>
<21423413, 34132142>
<21423422, 34132133>
<21423431, 34132124>
<21424124, 34131431>
<21424133, 34131422>
<21424142, 34131413>
<21424214, 34131341>
<21424223, 34131332>
<21424232, 34131323>
<21424241, 34131314>
<21424313, 34131242>
<21424322, 34131233>
<21424331, 34131224>
<21424412, 34131143>
<21424421, 34131134>
<21431144, 34124411>
<21431234, 34124321>
<21431243, 34124312>
<21431324, 34124231>
<21431333, 34124222>
<21431342, 34124213>
<21431414, 34124141>
<21431423, 34124132>
<21431432, 34124123>
<21431441, 34124114>
<21432134, 34123421>
<21432143, 34123412>
<21432224, 34123331>
<21432233, 34123322>
<21432242, 34123313>
<21432314, 34123241>
<21432323, 34123232>
<21432332, 34123223>
<21432341, 34123214>
<21432413, 34123142>
<21432422, 34123133>
<21432431, 34123124>
<21433124, 34122431>
<21433133, 34122422>
<21433142, 34122413>
<21433214, 34122341>
<21433223, 34122332>
<21433232, 34122323>
<21433241, 34122314>
<21433313, 34122242>
<21433322, 34122233>
<21433331, 34122224>
<21433412, 34122143>
<21433421, 34122134>
<21434114, 34121441>
<21434123, 34121432>

```
<21434132, 34121423>    <21434141, 34121414>    <21434213, 34121342>    <21434222, 34121333>
<21434231, 34121324>    <21434312, 34121243>    <21434321, 34121234>    <21434411, 34121144>
<21441134, 34114421>    <21441143, 34114412>    <21441224, 34114331>    <21441233, 34114322>
<21441242, 34114313>    <21441314, 34114241>    <21441323, 34114232>    <21441332, 34114223>
<21441341, 34114214>    <21441413, 34114142>    <21441422, 34114133>    <21441431, 34114124>
<21442124, 34113431>    <21442133, 34113422>    <21442142, 34113413>    <21442214, 34113341>
<21442223, 34113332>    <21442232, 34113323>    <21442241, 34113314>    <21442313, 34113242>
<21442322, 34113233>    <21442331, 34113224>    <21442412, 34113143>    <21442421, 34113134>
<21443114, 34112441>    <21443123, 34112432>    <21443132, 34112423>    <21443141, 34112414>
<21443213, 34112342>    <21443222, 34112333>    <21443231, 34112324>    <21443312, 34112243>
<21443321, 34112234>    <21443411, 34112144>    <21444113, 34111442>    <21444122, 34111433>
<21444131, 34111424>    <21444212, 34111343>    <21444221, 34111334>    <21444311, 34111244>
<22111111, 33444444>    <22112444, 33443111>    <22113344, 33442211>    <22113434, 33442121>
<22113443, 33442112>    <22114244, 33441311>    <22114334, 33441221>    <22114343, 33441212>
<22114424, 33441131>    <22114433, 33441122>    <22114442, 33441113>    <22121444, 33434111>
<22122344, 33433211>    <22122434, 33433121>    <22122443, 33433112>    <22123244, 33432311>
<22123334, 33432221>    <22123343, 33432212>    <22123424, 33432131>    <22123433, 33432122>
<22123442, 33432113>    <22124144, 33431411>    <22124234, 33431321>    <22124243, 33431312>
<22124324, 33431231>    <22124333, 33431222>    <22124342, 33431213>    <22124414, 33431141>
<22124423, 33431132>    <22124432, 33431123>    <22124441, 33431114>    <22131344, 33424211>
<22131434, 33424121>    <22131443, 33424112>    <22132244, 33423311>    <22132334, 33423221>
<22132343, 33423212>    <22132424, 33423131>    <22132433, 33423122>    <22132442, 33423113>
<22133144, 33422411>    <22133234, 33422321>    <22133243, 33422312>    <22133324, 33422231>
<22133333, 33422222>    <22133342, 33422213>    <22133414, 33422141>    <22133423, 33422132>
<22133432, 33422123>    <22133441, 33422114>    <22134134, 33421421>    <22134143, 33421412>
<22134224, 33421331>    <22134233, 33421322>    <22134242, 33421313>    <22134314, 33421241>
<22134323, 33421232>    <22134332, 33421223>    <22134341, 33421214>    <22134413, 33421142>
<22134422, 33421133>    <22134431, 33421124>    <22141244, 33414311>    <22141334, 33414221>
<22141343, 33414212>    <22141424, 33414131>    <22141433, 33414122>    <22141442, 33414113>
<22142144, 33413411>    <22142234, 33413321>    <22142243, 33413312>    <22142324, 33413231>
<22142333, 33413222>    <22142342, 33413213>    <22142414, 33413141>    <22142423, 33413132>
<22142432, 33413123>    <22142441, 33413114>    <22143134, 33412421>    <22143143, 33412412>
<22143224, 33412331>    <22143233, 33412322>    <22143242, 33412313>    <22143314, 33412241>
<22143323, 33412232>    <22143332, 33412223>    <22143341, 33412214>    <22143413, 33412142>
<22143422, 33412133>    <22143431, 33412124>    <22144124, 33411431>    <22144133, 33411422>
<22144142, 33411413>    <22144214, 33411341>    <22144223, 33411332>    <22144232, 33411323>
<22144241, 33411314>    <22144313, 33411242>    <22144322, 33411233>    <22144331, 33411224>
<22144412, 33411143>    <22144421, 33411134>    <22211444, 33344111>    <22212344, 33343211>
<22212434, 33343121>    <22212443, 33343112>    <22213244, 33342311>    <22213334, 33342221>
<22213343, 33342212>    <22213424, 33342131>    <22213433, 33342122>    <22213442, 33342113>
<22214144, 33341411>    <22214234, 33341321>    <22214243, 33341312>    <22214324, 33341231>
<22214333, 33341222>    <22214342, 33341213>    <22214414, 33341141>    <22214423, 33341132>
<22214432, 33341123>    <22214441, 33341114>    <22221344, 33334211>    <22221434, 33334121>
<22221443, 33334112>    <22222244, 33333311>    <22222334, 33333221>    <22222343, 33333212>
<22222424, 33333131>    <22222433, 33333122>    <22222442, 33333113>    <22223144, 33332411>
<22223234, 33332321>    <22223243, 33332312>    <22223324, 33332231>    <22223333, 33332222>
<22223342, 33332213>    <22223414, 33332141>    <22223423, 33332132>    <22223432, 33332123>
<22223441, 33332114>    <22224134, 33331421>    <22224143, 33331412>    <22224224, 33331331>
<22224233, 33331322>    <22224242, 33331313>    <22224314, 33331241>    <22224323, 33331232>
<22224332, 33331223>    <22224341, 33331214>    <22224413, 33331142>    <22224422, 33331133>
<22224431, 33331124>    <22231244, 33324311>    <22231334, 33324221>    <22231343, 33324212>
<22231424, 33324131>    <22231433, 33324122>    <22231442, 33324113>    <22232144, 33323411>
<22232234, 33323321>    <22232243, 33323312>    <22232324, 33323231>    <22232333, 33323222>
<22232342, 33323213>    <22232414, 33323141>    <22232423, 33323132>    <22232432, 33323123>
<22232441, 33323114>    <22233134, 33322421>    <22233143, 33322412>    <22233224, 33322331>
<22233233, 33322322>    <22233242, 33322313>    <22233314, 33322241>    <22233323, 33322232>
<22233332, 33322223>    <22233341, 33322214>    <22233413, 33322142>    <22233422, 33322133>
<22233431, 33322124>    <22234124, 33321431>    <22234133, 33321422>    <22234142, 33321413>
<22234214, 33321341>    <22234223, 33321332>    <22234232, 33321323>    <22234241, 33321314>
<22234313, 33321242>    <22234322, 33321233>    <22234331, 33321224>    <22234412, 33321143>
<22234421, 33321134>    <22241144, 33314411>    <22241234, 33314321>    <22241243, 33314312>
<22241324, 33314231>    <22241333, 33314222>    <22241342, 33314213>    <22241414, 33314141>
<22241423, 33314132>    <22241432, 33314123>    <22241441, 33314114>    <22242134, 33313421>
<22242143, 33313412>    <22242224, 33313331>    <22242233, 33313322>    <22242242, 33313313>
<22242314, 33313241>    <22242323, 33313232>    <22242332, 33313223>    <22242341, 33313214>
<22242413, 33313142>    <22242422, 33313133>    <22242431, 33313124>    <22243124, 33312431>
<22243133, 33312422>    <22243142, 33312413>    <22243214, 33312341>    <22243223, 33312332>
<22243232, 33312323>    <22243241, 33312314>    <22243313, 33312242>    <22243322, 33312233>
<22243331, 33312224>    <22243412, 33312143>    <22243421, 33312134>    <22244114, 33311441>
<22244123, 33311432>    <22244132, 33311423>    <22244141, 33311414>    <22244213, 33311342>
```

<22244222, 33311333> <22244231, 33311324> <22244312, 33311243> <22244321, 33311234>
<22244411, 33311144> <22311344, 33244211> <22311434, 33244121> <22311443, 33244112>
<22312244, 33243311> <22312334, 33243221> <22312343, 33243212> <22312424, 33243131>
<22312433, 33243122> <22312442, 33243113> <22313144, 33242411> <22313234, 33242321>
<22313243, 33242312> <22313324, 33242231> <22313333, 33242222> <22313342, 33242213>
<22313414, 33242141> <22313423, 33242132> <22313432, 33242123> <22313441, 33242114>
<22314134, 33241421> <22314143, 33241412> <22314224, 33241331> <22314233, 33241322>
<22314242, 33241313> <22314314, 33241241> <22314323, 33241232> <22314332, 33241223>
<22314341, 33241214> <22314413, 33241142> <22314422, 33241133> <22314431, 33241124>
<22321244, 33234311> <22321334, 33234221> <22321343, 33234212> <22321424, 33234131>
<22321433, 33234122> <22321442, 33234113> <22322144, 33233411> <22322234, 33233321>
<22322243, 33233312> <22322324, 33233231> <22322333, 33233222> <22322342, 33233213>
<22322414, 33233141> <22322423, 33233132> <22322432, 33233123> <22322441, 33233114>
<22323134, 33232421> <22323143, 33232412> <22323224, 33232331> <22323233, 33232322>
<22323242, 33232313> <22323314, 33232241> <22323323, 33232232> <22323332, 33232223>
<22323341, 33232214> <22323413, 33232142> <22323422, 33232133> <22323431, 33232124>
<22324124, 33231431> <22324133, 33231422> <22324142, 33231413> <22324214, 33231341>
<22324223, 33231332> <22324232, 33231323> <22324241, 33231314> <22324313, 33231242>
<22324322, 33231233> <22324331, 33231224> <22324412, 33231143> <22324421, 33231134>
<22331144, 33224411> <22331234, 33224321> <22331243, 33224312> <22331324, 33224231>
<22331333, 33224222> <22331342, 33224213> <22331414, 33224141> <22331423, 33224132>
<22331432, 33224123> <22331441, 33224114> <22332134, 33223421> <22332143, 33223412>
<22332224, 33223331> <22332233, 33223322> <22332242, 33223313> <22332314, 33223241>
<22332323, 33223232> <22332332, 33223223> <22332341, 33223214> <22332413, 33223142>
<22332422, 33223133> <22332431, 33223124> <22333124, 33222431> <22333133, 33222422>
<22333142, 33222413> <22333214, 33222341> <22333223, 33222332> <22333232, 33222323>
<22333241, 33222314> <22333313, 33222242> <22333322, 33222233> <22333331, 33222224>
<22333412, 33222143> <22333421, 33222134> <22334114, 33221441> <22334123, 33221432>
<22334132, 33221423> <22334141, 33221414> <22334213, 33221342> <22334222, 33221333>
<22334231, 33221324> <22334312, 33221243> <22334321, 33221234> <22334411, 33221144>
<22341134, 33214421> <22341143, 33214412> <22341224, 33214331> <22341233, 33214322>
<22341242, 33214313> <22341314, 33214241> <22341323, 33214232> <22341332, 33214223>
<22341341, 33214214> <22341413, 33214142> <22341422, 33214133> <22341431, 33214124>
<22342124, 33213431> <22342133, 33213422> <22342142, 33213413> <22342214, 33213341>
<22342223, 33213332> <22342232, 33213323> <22342241, 33213314> <22342313, 33213242>
<22342322, 33213233> <22342331, 33213224> <22342412, 33213143> <22342421, 33213134>
<22343114, 33212441> <22343123, 33212432> <22343132, 33212423> <22343141, 33212414>
<22343213, 33212342> <22343222, 33212333> <22343231, 33212324> <22343312, 33212243>
<22343321, 33212234> <22343411, 33212144> <22344113, 33211442> <22344122, 33211433>
<22344131, 33211424> <22344212, 33211343> <22344221, 33211334> <22344311, 33211244>
<22411244, 33144311> <22411334, 33144221> <22411343, 33144212> <22411424, 33144131>
<22411433, 33144122> <22411442, 33144113> <22412144, 33143411> <22412234, 33143321>
<22412243, 33143312> <22412324, 33143231> <22412333, 33143222> <22412342, 33143213>
<22412414, 33143141> <22412423, 33143132> <22412432, 33143123> <22412441, 33143114>
<22413134, 33142421> <22413143, 33142412> <22413224, 33142331> <22413233, 33142322>
<22413242, 33142313> <22413314, 33142241> <22413323, 33142232> <22413332, 33142223>
<22413341, 33142214> <22413413, 33142142> <22413422, 33142133> <22413431, 33142124>
<22414124, 33141431> <22414133, 33141422> <22414142, 33141413> <22414214, 33141341>
<22414223, 33141332> <22414232, 33141323> <22414241, 33141314> <22414313, 33141242>
<22414322, 33141233> <22414331, 33141224> <22414412, 33141143> <22414421, 33141134>
<22421144, 33134411> <22421234, 33134321> <22421243, 33134312> <22421324, 33134231>
<22421333, 33134222> <22421342, 33134213> <22421414, 33134141> <22421423, 33134132>
<22421432, 33134123> <22421441, 33134114> <22422134, 33133421> <22422143, 33133412>
<22422224, 33133331> <22422233, 33133322> <22422242, 33133313> <22422314, 33133241>
<22422323, 33133232> <22422332, 33133223> <22422341, 33133214> <22422413, 33133142>
<22422422, 33133133> <22422431, 33133124> <22423124, 33132431> <22423133, 33132422>
<22423142, 33132413> <22423214, 33132341> <22423223, 33132332> <22423232, 33132323>
<22423241, 33132314> <22423313, 33132242> <22423322, 33132233> <22423331, 33132224>
<22423412, 33132143> <22423421, 33132134> <22424114, 33131441> <22424123, 33131432>
<22424132, 33131423> <22424141, 33131414> <22424213, 33131342> <22424222, 33131333>
<22424231, 33131324> <22424312, 33131243> <22424321, 33131234> <22424411, 33131144>
<22431134, 33124421> <22431143, 33124412> <22431224, 33124331> <22431233, 33124322>
<22431242, 33124313> <22431314, 33124241> <22431323, 33124232> <22431332, 33124223>
<22431341, 33124214> <22431413, 33124142> <22431422, 33124133> <22431431, 33124124>
<22432124, 33123431> <22432133, 33123422> <22432142, 33123413> <22432214, 33123341>
<22432223, 33123332> <22432232, 33123323> <22432241, 33123314> <22432313, 33123242>
<22432322, 33123233> <22432331, 33123224> <22432412, 33123143> <22432421, 33123134>
<22433114, 33122441> <22433123, 33122432> <22433132, 33122423> <22433141, 33122414>
<22433213, 33122342> <22433222, 33122333> <22433231, 33122324> <22433312, 33122243>
<22433321, 33122234> <22433411, 33122144> <22434113, 33121442> <22434122, 33121433>

<22434131, 33121424>
<22434212, 33121343>
<22434221, 33121334>
<22434311, 33121244>
<22441124, 33114431>
<22441133, 33114422>
<22441142, 33114413>
<22441214, 33114341>
<22441223, 33114332>
<22441232, 33114323>
<22441241, 33114314>
<22441313, 33114242>
<22441322, 33114233>
<22441331, 33114224>
<22441412, 33114143>
<22441421, 33114134>
<22442114, 33113441>
<22442123, 33113432>
<22442132, 33113423>
<22442141, 33113414>
<22442213, 33113342>
<22442222, 33113333>
<22442231, 33113324>
<22442312, 33113243>
<22442321, 33113234>
<22442411, 33113144>
<22443113, 33112442>
<22443122, 33112433>
<22443131, 33112424>
<22443212, 33112343>
<22443221, 33112334>
<22443311, 33112244>
<22444112, 33111443>
<22444121, 33111434>
<22444211, 33111344>
<23111444, 32444111>
<23112344, 32443211>
<23112434, 32443121>
<23112443, 32443112>
<23113244, 32442311>
<23113334, 32442221>
<23113343, 32442212>
<23113424, 32442131>
<23113433, 32442122>
<23113442, 32442113>
<23114144, 32441411>
<23114234, 32441321>
<23114243, 32441312>
<23114324, 32441231>
<23114333, 32441222>
<23114342, 32441213>
<23114414, 32441141>
<23114423, 32441132>
<23114432, 32441123>
<23114441, 32441114>
<23121344, 32434211>
<23121434, 32434121>
<23121443, 32434112>
<23122244, 32433311>
<23122334, 32433221>
<23122343, 32433212>
<23122424, 32433131>
<23122433, 32433122>
<23122442, 32433113>
<23123144, 32432411>
<23123234, 32432321>
<23123243, 32432312>
<23123324, 32432231>
<23123333, 32432222>
<23123342, 32432213>
<23123414, 32432141>
<23123423, 32432132>
<23123432, 32432123>
<23123441, 32432114>
<23124134, 32431421>
<23124143, 32431412>
<23124224, 32431331>
<23124233, 32431322>
<23124242, 32431313>
<23124314, 32431241>
<23124323, 32431232>
<23124332, 32431223>
<23124341, 32431214>
<23124413, 32431142>
<23124422, 32431133>
<23124431, 32431124>
<23131244, 32424311>
<23131334, 32424221>
<23131343, 32424212>
<23131424, 32424131>
<23131433, 32424122>
<23131442, 32424113>
<23132144, 32423411>
<23132234, 32423321>
<23132243, 32423312>
<23132324, 32423231>
<23132333, 32423222>
<23132342, 32423213>
<23132414, 32423141>
<23132423, 32423132>
<23132432, 32423123>
<23132441, 32423114>
<23133134, 32422421>
<23133143, 32422412>
<23133224, 32422331>
<23133233, 32422322>
<23133242, 32422313>
<23133314, 32422241>
<23133323, 32422232>
<23133332, 32422223>
<23133341, 32422214>
<23133413, 32422142>
<23133422, 32422133>
<23133431, 32422124>
<23134124, 32421431>
<23134133, 32421422>
<23134142, 32421413>
<23134214, 32421341>
<23134223, 32421332>
<23134232, 32421323>
<23134241, 32421314>
<23134313, 32421242>
<23134322, 32421233>
<23134331, 32421224>
<23134412, 32421143>
<23134421, 32421134>
<23141144, 32414411>
<23141234, 32414321>
<23141243, 32414312>
<23141324, 32414231>
<23141333, 32414222>
<23141342, 32414213>
<23141414, 32414141>
<23141423, 32414132>
<23141432, 32414123>
<23141441, 32414114>
<23142134, 32413421>
<23142143, 32413412>
<23142224, 32413331>
<23142233, 32413322>
<23142242, 32413313>
<23142314, 32413241>
<23142323, 32413232>
<23142332, 32413223>
<23142341, 32413214>
<23142413, 32413142>
<23142422, 32413133>
<23142431, 32413124>
<23143124, 32412431>
<23143133, 32412422>
<23143142, 32412413>
<23143214, 32412341>
<23143223, 32412332>
<23143232, 32412323>
<23143241, 32412314>
<23143313, 32412242>
<23143322, 32412233>
<23143331, 32412224>
<23143412, 32412143>
<23143421, 32412134>
<23144114, 32411441>
<23144123, 32411432>
<23144132, 32411423>
<23144141, 32411414>
<23144213, 32411342>
<23144222, 32411333>
<23144231, 32411324>
<23144312, 32411243>
<23144321, 32411234>
<23144411, 32411144>
<23211344, 32344211>
<23211434, 32344121>
<23211443, 32344112>
<23212244, 32343311>
<23212334, 32343221>
<23212343, 32343212>
<23212424, 32343131>
<23212433, 32343122>
<23212442, 32343113>
<23213144, 32342411>
<23213234, 32342321>
<23213243, 32342312>
<23213324, 32342231>
<23213333, 32342222>
<23213342, 32342213>
<23213414, 32342141>
<23213423, 32342132>
<23213432, 32342123>
<23213441, 32342114>
<23214134, 32341421>
<23214143, 32341412>
<23214224, 32341331>
<23214233, 32341322>
<23214242, 32341313>
<23214314, 32341241>
<23214323, 32341232>
<23214332, 32341223>
<23214341, 32341214>
<23214413, 32341142>
<23214422, 32341133>
<23214431, 32341124>
<23221244, 32334311>
<23221334, 32334221>
<23221343, 32334212>
<23221424, 32334131>
<23221433, 32334122>
<23221442, 32334113>
<23222144, 32333411>
<23222234, 32333321>
<23222243, 32333312>
<23222324, 32333231>
<23222333, 32333222>
<23222342, 32333213>
<23222414, 32333141>
<23222423, 32333132>
<23222432, 32333123>
<23222441, 32333114>
<23223134, 32332421>
<23223143, 32332412>
<23223224, 32332331>
<23223233, 32332322>
<23223242, 32332313>
<23223314, 32332241>
<23223323, 32332232>
<23223332, 32332223>
<23223341, 32332214>
<23223413, 32332142>
<23223422, 32332133>
<23223431, 32332124>
<23224124, 32331431>
<23224133, 32331422>
<23224142, 32331413>
<23224214, 32331341>
<23224223, 32331332>
<23224232, 32331323>
<23224241, 32331314>
<23224313, 32331242>
<23224322, 32331233>
<23224331, 32331224>
<23224412, 32331143>
<23224421, 32331134>
<23231144, 32324411>
<23231234, 32324321>
<23231243, 32324312>
<23231324, 32324231>
<23231333, 32324222>
<23231342, 32324213>
<23231414, 32324141>
<23231423, 32324132>
<23231432, 32324123>
<23231441, 32324114>
<23232134, 32323421>
<23232143, 32323412>
<23232224, 32323331>
<23232233, 32323322>
<23232242, 32323313>
<23232314, 32323241>
<23232323, 32323232>
<23232332, 32323223>
<23232341, 32323214>
<23232413, 32323142>
<23232422, 32323133>
<23232431, 32323124>
<23233124, 32322431>
<23233133, 32322422>
<23233142, 32322413>
<23233214, 32322341>
<23233223, 32322332>
<23233232, 32322323>
<23233241, 32322314>
<23233313, 32322242>
<23233322, 32322233>
<23233331, 32322224>
<23233412, 32322143>
<23233421, 32322134>
<23234114, 32321441>
<23234123, 32321432>
<23234132, 32321423>
<23234141, 32321414>
<23234213, 32321342>

<23234222, 32321333>
<23234411, 32321144>
<23241233, 32314322>
<23241332, 32314223>
<23241431, 32314124>
<23242214, 32313341>
<23242313, 32313242>
<23242421, 32313134>
<23243141, 32312414>
<23243312, 32312243>
<23244122, 32311433>
<23244311, 32311244>
<23311424, 32244131>
<23312234, 32243321>
<23312342, 32243213>
<23312441, 32243114>
<23313233, 32242322>
<23313332, 32242223>
<23313431, 32242124>
<23314214, 32241341>
<23314313, 32241242>
<23314421, 32241134>
<23321324, 32234231>
<23321423, 32234132>
<23322143, 32233412>
<23322314, 32233241>
<23322413, 32233142>
<23323133, 32232422>
<23323232, 32232323>
<23323331, 32232224>
<23324123, 32231432>
<23324222, 32231333>
<23324411, 32231144>
<23331233, 32224322>
<23331332, 32224223>
<23331431, 32224124>
<23332214, 32223341>
<23332313, 32223242>
<23332421, 32223134>
<23333141, 32222414>
<23333312, 32222243>
<23334122, 32221433>
<23334311, 32221244>
<23341214, 32214341>
<23341313, 32214242>
<23341421, 32214134>
<23342141, 32213414>
<23342312, 32213243>
<23343122, 32212433>
<23343311, 32212244>
<23411144, 32144411>
<23411333, 32144222>
<23411432, 32144123>
<23412224, 32143331>
<23412323, 32143232>
<23412422, 32143133>
<23413142, 32142413>
<23413241, 32142314>
<23413412, 32142143>
<23414132, 32141423>
<23414231, 32141324>
<23421134, 32134421>
<23421242, 32134313>
<23421341, 32134214>
<23422124, 32133431>
<23422223, 32133332>
<23422322, 32133233>
<23423114, 32132441>
<23423213, 32132342>
<23423321, 32132234>
<23234231, 32321324>
<23241134, 32314421>
<23241242, 32314313>
<23241341, 32314214>
<23242124, 32313431>
<23242223, 32313332>
<23242322, 32313233>
<23243114, 32312441>
<23243213, 32312342>
<23243321, 32312234>
<23244131, 32311424>
<23311244, 32244311>
<23311433, 32244122>
<23312243, 32243312>
<23312414, 32243141>
<23313134, 32242421>
<23313242, 32242313>
<23313341, 32242214>
<23314124, 32241431>
<23314223, 32241332>
<23314322, 32241233>
<23321144, 32234411>
<23321333, 32234222>
<23321432, 32234123>
<23322224, 32233331>
<23322323, 32233232>
<23322422, 32233133>
<23323142, 32232413>
<23323241, 32232314>
<23323412, 32232143>
<23324132, 32231423>
<23324231, 32231324>
<23331134, 32224421>
<23331242, 32224313>
<23331341, 32224214>
<23332124, 32223431>
<23332223, 32223332>
<23332322, 32223233>
<23333114, 32222441>
<23333213, 32222342>
<23333321, 32222234>
<23334131, 32221424>
<23341124, 32214431>
<23341223, 32214332>
<23341322, 32214233>
<23342114, 32213441>
<23342213, 32213342>
<23342321, 32213234>
<23343131, 32212424>
<23344112, 32211443>
<23411234, 32144321>
<23411342, 32144213>
<23411441, 32144114>
<23412233, 32143322>
<23412332, 32143223>
<23412431, 32143124>
<23413214, 32142341>
<23413313, 32142242>
<23413421, 32142134>
<23414141, 32141414>
<23414312, 32141243>
<23421143, 32134412>
<23421314, 32134241>
<23421413, 32134142>
<23422133, 32133422>
<23422232, 32133323>
<23422331, 32133224>
<23423123, 32132432>
<23423222, 32132333>
<23423411, 32132144>
<23234312, 32321243>
<23241143, 32314412>
<23241314, 32314241>
<23241413, 32314142>
<23242133, 32313422>
<23242232, 32313323>
<23242331, 32313224>
<23243123, 32312432>
<23243222, 32312333>
<23243411, 32312144>
<23244212, 32311343>
<23311334, 32244221>
<23311442, 32244113>
<23312324, 32243231>
<23312423, 32243132>
<23313143, 32242412>
<23313314, 32242241>
<23313413, 32242142>
<23314133, 32241422>
<23314232, 32241323>
<23314331, 32241224>
<23321234, 32234321>
<23321342, 32234213>
<23321441, 32234114>
<23322233, 32233322>
<23322332, 32233223>
<23322431, 32233124>
<23323214, 32232341>
<23323313, 32232242>
<23323421, 32232134>
<23324141, 32231414>
<23324312, 32231243>
<23331143, 32224412>
<23331314, 32224241>
<23331413, 32224142>
<23332133, 32223422>
<23332232, 32223323>
<23332331, 32223224>
<23333123, 32222432>
<23333222, 32222333>
<23333411, 32222144>
<23334212, 32221343>
<23341133, 32214422>
<23341232, 32214323>
<23341331, 32214224>
<23342123, 32213432>
<23342222, 32213333>
<23342411, 32213144>
<23343212, 32212343>
<23344121, 32211434>
<23411243, 32144312>
<23411414, 32144141>
<23412134, 32143421>
<23412242, 32143313>
<23412341, 32143214>
<23413124, 32142431>
<23413223, 32142332>
<23413322, 32142233>
<23414114, 32141441>
<23414213, 32141342>
<23414321, 32141234>
<23421224, 32134331>
<23421323, 32134232>
<23421422, 32134133>
<23422142, 32133413>
<23422241, 32133314>
<23422412, 32133143>
<23423132, 32132423>
<23423231, 32132324>
<23424113, 32131442>
<23234321, 32321234>
<23241224, 32314331>
<23241323, 32314232>
<23241422, 32314133>
<23242142, 32313413>
<23242241, 32313314>
<23242412, 32313143>
<23243132, 32312423>
<23243231, 32312324>
<23244113, 32311442>
<23244221, 32311334>
<23311343, 32244212>
<23312144, 32243411>
<23312333, 32243222>
<23312432, 32243123>
<23313224, 32242331>
<23313323, 32242232>
<23313422, 32242133>
<23314142, 32241413>
<23314241, 32241314>
<23314412, 32241143>
<23321243, 32234312>
<23321414, 32234141>
<23322134, 32233421>
<23322242, 32233313>
<23322341, 32233214>
<23323124, 32232431>
<23323223, 32232332>
<23323322, 32232233>
<23324114, 32231441>
<23324213, 32231342>
<23324321, 32231234>
<23331224, 32224331>
<23331323, 32224232>
<23331422, 32224133>
<23332142, 32223413>
<23332241, 32223314>
<23332412, 32223143>
<23333132, 32222423>
<23333231, 32222324>
<23334113, 32221442>
<23334221, 32221334>
<23341142, 32214413>
<23341241, 32214314>
<23341412, 32214143>
<23342132, 32213423>
<23342231, 32213324>
<23343113, 32212442>
<23343221, 32212334>
<23344211, 32211344>
<23411324, 32144231>
<23411423, 32144132>
<23412143, 32143412>
<23412314, 32143241>
<23412413, 32143142>
<23413133, 32142422>
<23413232, 32142323>
<23413331, 32142224>
<23414123, 32141432>
<23414222, 32141333>
<23414411, 32141144>
<23421233, 32134322>
<23421332, 32134223>
<23421431, 32134124>
<23422214, 32133341>
<23422313, 32133242>
<23422421, 32133134>
<23423141, 32132414>
<23423312, 32132243>
<23424122, 32131433>

<23424131, 32131424>
<23424212, 32131343>
<23424221, 32131334>
<23424311, 32131244>
<23431124, 32124431>
<23431133, 32124422>
<23431142, 32124413>
<23431214, 32124341>
<23431223, 32124332>
<23431232, 32124323>
<23431241, 32124314>
<23431313, 32124242>
<23431322, 32124233>
<23431331, 32124224>
<23431412, 32124143>
<23431421, 32124134>
<23432114, 32123441>
<23432123, 32123432>
<23432132, 32123423>
<23432141, 32123414>
<23432213, 32123342>
<23432222, 32123333>
<23432231, 32123324>
<23432312, 32123243>
<23432321, 32123234>
<23432411, 32123144>
<23433113, 32122442>
<23433122, 32122433>
<23433131, 32122424>
<23433212, 32122343>
<23433221, 32122334>
<23433311, 32122244>
<23434112, 32121443>
<23434121, 32121434>
<23434211, 32121344>
<23441114, 32114441>
<23441123, 32114432>
<23441132, 32114423>
<23441141, 32114414>
<23441213, 32114342>
<23441222, 32114333>
<23441231, 32114324>
<23441312, 32114243>
<23441321, 32114234>
<23441411, 32114144>
<23442113, 32113442>
<23442122, 32113433>
<23442131, 32113424>
<23442212, 32113343>
<23442221, 32113334>
<23442311, 32113244>
<23443112, 32112443>
<23443121, 32112434>
<23443211, 32112344>
<23444111, 32111444>
<24111344, 31444211>
<24111434, 31444121>
<24111443, 31444112>
<24112244, 31443311>
<24112334, 31443221>
<24112343, 31443212>
<24112424, 31443131>
<24112433, 31443122>
<24112442, 31443113>
<24113144, 31442411>
<24113234, 31442321>
<24113243, 31442312>
<24113324, 31442231>
<24113333, 31442222>
<24113342, 31442213>
<24113414, 31442141>
<24113423, 31442132>
<24113432, 31442123>
<24113441, 31442114>
<24114134, 31441421>
<24114143, 31441412>
<24114224, 31441331>
<24114233, 31441322>
<24114242, 31441313>
<24114314, 31441241>
<24114323, 31441232>
<24114332, 31441223>
<24114341, 31441214>
<24114413, 31441142>
<24114422, 31441133>
<24114431, 31441124>
<24121244, 31434311>
<24121334, 31434221>
<24121343, 31434212>
<24121424, 31434131>
<24121433, 31434122>
<24121442, 31434113>
<24122144, 31433411>
<24122234, 31433321>
<24122243, 31433312>
<24122324, 31433231>
<24122333, 31433222>
<24122342, 31433213>
<24122414, 31433141>
<24122423, 31433132>
<24122432, 31433123>
<24122441, 31433114>
<24123134, 31432421>
<24123143, 31432412>
<24123224, 31432331>
<24123233, 31432322>
<24123242, 31432313>
<24123314, 31432241>
<24123323, 31432232>
<24123332, 31432223>
<24123341, 31432214>
<24123413, 31432142>
<24123422, 31432133>
<24123431, 31432124>
<24124124, 31431431>
<24124133, 31431422>
<24124142, 31431413>
<24124214, 31431341>
<24124223, 31431332>
<24124232, 31431323>
<24124241, 31431314>
<24124313, 31431242>
<24124322, 31431233>
<24124331, 31431224>
<24124412, 31431143>
<24124421, 31431134>
<24131144, 31424411>
<24131234, 31424321>
<24131243, 31424312>
<24131324, 31424231>
<24131333, 31424222>
<24131342, 31424213>
<24131414, 31424141>
<24131423, 31424132>
<24131432, 31424123>
<24131441, 31424114>
<24132134, 31423421>
<24132143, 31423412>
<24132224, 31423331>
<24132233, 31423322>
<24132242, 31423313>
<24132314, 31423241>
<24132323, 31423232>
<24132332, 31423223>
<24132341, 31423214>
<24132413, 31423142>
<24132422, 31423133>
<24132431, 31423124>
<24133124, 31422431>
<24133133, 31422422>
<24133142, 31422413>
<24133214, 31422341>
<24133223, 31422332>
<24133232, 31422323>
<24133241, 31422314>
<24133313, 31422242>
<24133322, 31422233>
<24133331, 31422224>
<24133412, 31422143>
<24133421, 31422134>
<24134114, 31421441>
<24134123, 31421432>
<24134132, 31421423>
<24134141, 31421414>
<24134213, 31421342>
<24134222, 31421333>
<24134231, 31421324>
<24134312, 31421243>
<24134321, 31421234>
<24134411, 31421144>
<24141134, 31414421>
<24141143, 31414412>
<24141224, 31414331>
<24141233, 31414322>
<24141242, 31414313>
<24141314, 31414241>
<24141323, 31414232>
<24141332, 31414223>
<24141341, 31414214>
<24141413, 31414142>
<24141422, 31414133>
<24141431, 31414124>
<24142124, 31413431>
<24142133, 31413422>
<24142142, 31413413>
<24142214, 31413341>
<24142223, 31413332>
<24142232, 31413323>
<24142241, 31413314>
<24142313, 31413242>
<24142322, 31413233>
<24142331, 31413224>
<24142412, 31413143>
<24142421, 31413134>
<24143114, 31412441>
<24143123, 31412432>
<24143132, 31412423>
<24143141, 31412414>
<24143213, 31412342>
<24143222, 31412333>
<24143231, 31412324>
<24143312, 31412243>
<24143321, 31412234>
<24143411, 31412144>
<24144113, 31411442>
<24144122, 31411433>
<24144131, 31411424>
<24144212, 31411343>
<24144221, 31411334>
<24144311, 31411244>
<24211244, 31344311>
<24211334, 31344221>
<24211343, 31344212>
<24211424, 31344131>
<24211433, 31344122>
<24211442, 31344113>
<24212144, 31343411>
<24212234, 31343321>
<24212243, 31343312>
<24212324, 31343231>
<24212333, 31343222>
<24212342, 31343213>
<24212414, 31343141>
<24212423, 31343132>
<24212432, 31343123>
<24212441, 31343114>
<24213134, 31342421>
<24213143, 31342412>
<24213224, 31342331>
<24213233, 31342322>
<24213242, 31342313>
<24213314, 31342241>
<24213323, 31342232>
<24213332, 31342223>
<24213341, 31342214>
<24213413, 31342142>
<24213422, 31342133>
<24213431, 31342124>
<24214124, 31341431>
<24214133, 31341422>
<24214142, 31341413>
<24214214, 31341341>
<24214223, 31341332>
<24214232, 31341323>
<24214241, 31341314>
<24214313, 31341242>
<24214322, 31341233>
<24214331, 31341224>
<24214412, 31341143>
<24214421, 31341134>
<24221144, 31334411>
<24221234, 31334321>
<24221243, 31334312>
<24221324, 31334231>
<24221333, 31334222>
<24221342, 31334213>
<24221414, 31334141>
<24221423, 31334132>
<24221432, 31334123>
<24221441, 31334114>
<24222134, 31333421>
<24222143, 31333412>
<24222224, 31333331>
<24222233, 31333322>
<24222242, 31333313>
<24222314, 31333241>
<24222323, 31333232>
<24222332, 31333223>
<24222341, 31333214>
<24222413, 31333142>
<24222422, 31333133>
<24222431, 31333124>
<24223124, 31332431>
<24223133, 31332422>
<24223142, 31332413>
<24223214, 31332341>
<24223223, 31332332>
<24223232, 31332323>
<24223241, 31332314>
<24223313, 31332242>

<24223322, 31332233>	<24223331, 31332224>	<24223412, 31332143>	<24223421, 31332134>
<24224114, 31331441>	<24224123, 31331432>	<24224132, 31331423>	<24224141, 31331414>
<24224213, 31331342>	<24224222, 31331333>	<24224231, 31331324>	<24224312, 31331243>
<24224321, 31331234>	<24224411, 31331144>	<24231134, 31324421>	<24231143, 31324412>
<24231224, 31324331>	<24231233, 31324322>	<24231242, 31324313>	<24231314, 31324241>
<24231323, 31324232>	<24231332, 31324223>	<24231341, 31324214>	<24231413, 31324142>
<24231422, 31324133>	<24231431, 31324124>	<24232124, 31323431>	<24232133, 31323422>
<24232142, 31323413>	<24232214, 31323341>	<24232223, 31323332>	<24232232, 31323323>
<24232241, 31323314>	<24232313, 31323242>	<24232322, 31323233>	<24232331, 31323224>
<24232412, 31323143>	<24232421, 31323134>	<24233114, 31322441>	<24233123, 31322432>
<24233132, 31322423>	<24233141, 31322414>	<24233213, 31322342>	<24233222, 31322333>
<24233231, 31322324>	<24233312, 31322243>	<24233321, 31322234>	<24233411, 31322144>
<24234113, 31321442>	<24234122, 31321433>	<24234131, 31321424>	<24234212, 31321343>
<24234221, 31321334>	<24234311, 31321244>	<24241124, 31314431>	<24241133, 31314422>
<24241142, 31314413>	<24241214, 31314341>	<24241223, 31314332>	<24241232, 31314323>
<24241241, 31314314>	<24241313, 31314242>	<24241322, 31314233>	<24241331, 31314224>
<24241412, 31314143>	<24241421, 31314134>	<24242114, 31313441>	<24242123, 31313432>
<24242132, 31313423>	<24242141, 31313414>	<24242213, 31313342>	<24242222, 31313333>
<24242231, 31313324>	<24242312, 31313243>	<24242321, 31313234>	<24242411, 31313144>
<24243113, 31312442>	<24243122, 31312433>	<24243131, 31312424>	<24243212, 31312343>
<24243221, 31312334>	<24243311, 31312244>	<24244112, 31311443>	<24244121, 31311434>
<24244211, 31311344>	<24311144, 31244411>	<24311234, 31244321>	<24311243, 31244312>
<24311324, 31244231>	<24311333, 31244222>	<24311342, 31244213>	<24311414, 31244141>
<24311423, 31244132>	<24311432, 31244123>	<24311441, 31244114>	<24312134, 31243421>
<24312143, 31243412>	<24312224, 31243331>	<24312233, 31243322>	<24312242, 31243313>
<24312314, 31243241>	<24312323, 31243232>	<24312332, 31243223>	<24312341, 31243214>
<24312413, 31243142>	<24312422, 31243133>	<24312431, 31243124>	<24313124, 31242431>
<24313133, 31242422>	<24313142, 31242413>	<24313214, 31242341>	<24313223, 31242332>
<24313232, 31242323>	<24313241, 31242314>	<24313313, 31242242>	<24313322, 31242233>
<24313331, 31242224>	<24313412, 31242143>	<24313421, 31242134>	<24314114, 31241441>
<24314123, 31241432>	<24314132, 31241423>	<24314141, 31241414>	<24314213, 31241342>
<24314222, 31241333>	<24314231, 31241324>	<24314312, 31241243>	<24314321, 31241234>
<24314411, 31241144>	<24321134, 31234421>	<24321143, 31234412>	<24321224, 31234331>
<24321233, 31234322>	<24321242, 31234313>	<24321314, 31234241>	<24321323, 31234232>
<24321332, 31234223>	<24321341, 31234214>	<24321413, 31234142>	<24321422, 31234133>
<24321431, 31234124>	<24322124, 31233431>	<24322133, 31233422>	<24322142, 31233413>
<24322214, 31233341>	<24322223, 31233332>	<24322232, 31233323>	<24322241, 31233314>
<24322313, 31233242>	<24322322, 31233233>	<24322331, 31233224>	<24322412, 31233143>
<24322421, 31233134>	<24323114, 31232441>	<24323123, 31232432>	<24323132, 31232423>
<24323141, 31232414>	<24323213, 31232342>	<24323222, 31232333>	<24323231, 31232324>
<24323312, 31232243>	<24323321, 31232234>	<24323411, 31232144>	<24324113, 31231442>
<24324122, 31231433>	<24324131, 31231424>	<24324212, 31231343>	<24324221, 31231334>
<24324311, 31231244>	<24331124, 31224431>	<24331133, 31224422>	<24331142, 31224413>
<24331214, 31224341>	<24331223, 31224332>	<24331232, 31224323>	<24331241, 31224314>
<24331313, 31224242>	<24331322, 31224233>	<24331331, 31224224>	<24331412, 31224143>
<24331421, 31224134>	<24332114, 31223441>	<24332123, 31223432>	<24332132, 31223423>
<24332141, 31223414>	<24332213, 31223342>	<24332222, 31223333>	<24332231, 31223324>
<24332312, 31223243>	<24332321, 31223234>	<24332411, 31223144>	<24333113, 31222442>
<24333122, 31222433>	<24333131, 31222424>	<24333212, 31222343>	<24333221, 31222334>
<24333311, 31222244>	<24334112, 31221443>	<24334121, 31221434>	<24334211, 31221344>
<24341114, 31214441>	<24341123, 31214432>	<24341132, 31214423>	<24341141, 31214414>
<24341213, 31214342>	<24341222, 31214333>	<24341231, 31214324>	<24341312, 31214243>
<24341321, 31214234>	<24341411, 31214144>	<24342113, 31213442>	<24342122, 31213433>
<24342131, 31213424>	<24342212, 31213343>	<24342221, 31213334>	<24342311, 31213244>
<24343112, 31212443>	<24343121, 31212434>	<24343211, 31212344>	<24344111, 31211444>
<24411134, 31144421>	<24411143, 31144412>	<24411224, 31144331>	<24411233, 31144322>
<24411242, 31144313>	<24411314, 31144241>	<24411323, 31144232>	<24411332, 31144223>
<24411341, 31144214>	<24411413, 31144142>	<24411422, 31144133>	<24411431, 31144124>
<24412124, 31143431>	<24412133, 31143422>	<24412142, 31143413>	<24412214, 31143341>
<24412223, 31143332>	<24412232, 31143323>	<24412241, 31143314>	<24412313, 31143242>
<24412322, 31143233>	<24412331, 31143224>	<24412412, 31143143>	<24412421, 31143134>
<24413114, 31142441>	<24413123, 31142432>	<24413132, 31142423>	<24413141, 31142414>
<24413213, 31142342>	<24413222, 31142333>	<24413231, 31142324>	<24413312, 31142243>
<24413321, 31142234>	<24413411, 31142144>	<24414113, 31141442>	<24414122, 31141433>
<24414131, 31141424>	<24414212, 31141343>	<24414221, 31141334>	<24414311, 31141244>
<24421124, 31134431>	<24421133, 31134422>	<24421142, 31134413>	<24421214, 31134341>
<24421223, 31134332>	<24421232, 31134323>	<24421241, 31134314>	<24421313, 31134242>
<24421322, 31134233>	<24421331, 31134224>	<24421412, 31134143>	<24421421, 31134134>
<24422114, 31133441>	<24422123, 31133432>	<24422132, 31133423>	<24422141, 31133414>
<24422213, 31133342>	<24422222, 31133333>	<24422231, 31133324>	<24422312, 31133243>

<24422321, 31133234>
<24422411, 31133144>
<24423113, 31132442>
<24423122, 31132433>
<24423131, 31132424>
<24423212, 31132343>
<24423221, 31132334>
<24423311, 31132244>
<24424112, 31131443>
<24424121, 31131434>
<24424211, 31131344>
<24431114, 31124441>
<24431123, 31124432>
<24431132, 31124423>
<24431141, 31124414>
<24431213, 31124342>
<24431222, 31124333>
<24431231, 31124324>
<24431312, 31124243>
<24431321, 31124234>
<24431411, 31124144>
<24432113, 31123442>
<24432122, 31123433>
<24432131, 31123424>
<24432212, 31123343>
<24432221, 31123334>
<24432311, 31123244>
<24433112, 31122443>
<24433121, 31122434>
<24433211, 31122344>
<24434111, 31121444>
<24441113, 31114442>
<24441122, 31114433>
<24441131, 31114424>
<24441212, 31114343>
<24441221, 31114334>
<24441311, 31114244>
<24442112, 31113443>
<24442121, 31113434>
<24442211, 31113344>
<24443111, 31112444>
<24444444, 31111111>

List of 4082 dual pairs of nucleotides

<AAAAAAAG, CCCCCCCT>
<AAAAAATT, CCCCCCGG>
<AAAAAAGA, CCCCCCTC>
<AAAAATAT, CCCCCGCG>
<AAAAATTA, CCCCCGGC>
<AAAAAGAA, CCCCCTCC>
<AAAATAAT, CCCCGCCG>
<AAAATATA, CCCCGCGC>
<AAAATTAA, CCCCGGCC>
<AAAAGAAA, CCCCTCCC>
<AAAACCCC, CCCCAAAA>
<AAATAAAT, CCCGCCCG>
<AAATAATA, CCCGCCGC>
<AAATATAA, CCCGCGCC>
<AAATTAAA, CCCGGCCC>
<AAATGCCC, CCCGTAAA>
<AAATCGCC, CCCGATAA>
<AAATCCGC, CCCGAATA>
<AAATCCCG, CCCGAAAT>
<AAAGAAAA, CCCTCCCC>
<AAAGTCCC, CCCTGAAA>
<AAAGGGCC, CCCTTTAA>
<AAAGGCGC, CCCTTATA>
<AAAGGCCG, CCCTTAAT>
<AAAGCTCC, CCCTAGAA>
<AAAGCGGC, CCCTATTA>
<AAAGCGCG, CCCTATAT>
<AAAGCCTC, CCCTAAGA>
<AAAGCCGG, CCCTAATT>
<AAAGCCCT, CCCTAAAG>
<AAACACCC, CCCACAAA>
<AAACTGCC, CCCAGTAA>
<AAACTCGC, CCCAGATA>
<AAACTCCG, CCCAGAAT>
<AAACGTCC, CCCATGAA>
<AAACGGGC, CCCATTTA>
<AAACGGCG, CCCATTAT>
<AAACGCTC, CCCATAGA>
<AAACGCGG, CCCATATT>
<AAACGCCT, CCCATAAG>
<AAACCACC, CCCAACAA>
<AAACCTGC, CCCAAGTA>
<AAACCTCG, CCCAAGAT>
<AAACCGTC, CCCAATGA>
<AAACCGGG, CCCAATTT>
<AAACCGCT, CCCAATAG>
<AAACCCAC, CCCAAACA>
<AAACCCTG, CCCAAAGT>
<AAACCCGT, CCCAAATG>
<AAACCCCA, CCCAAAAC>
<AATAAAAT, CCGCCCCG>
<AATAAATA, CCGCCCGC>
<AATAATAA, CCGCCGCC>
<AATATAAA, CCGCGCCC>
<AATAGCCC, CCGCTAAA>
<AATACGCC, CCGCATAA>
<AATACCGC, CCGCAATA>
<AATACCCG, CCGCAAAT>
<AATTAAAA, CCGGCCCC>
<AATTTCCC, CCGGGAAA>
<AATTGGCC, CCGGTTAA>
<AATTGCGC, CCGGTATA>
<AATTGCCG, CCGGTAAT>
<AATTCTCC, CCGGAGAA>
<AATTCGGC, CCGGATTA>
<AATTCGCG, CCGGATAT>
<AATTCCTC, CCGGAAGA>
<AATTCCGG, CCGGAATT>
<AATTCCCT, CCGGAAAG>
<AATGACCC, CCGTCAAA>
<AATGTGCC, CCGTGTAA>
<AATGTCGC, CCGTGATA>
<AATGTCCG, CCGTGAAT>
<AATGGTCC, CCGTTGAA>
<AATGGGGC, CCGTTTTA>
<AATGGGCG, CCGTTTAT>
<AATGGCTC, CCGTTAGA>
<AATGGCGG, CCGTTATT>
<AATGGCCT, CCGTTAAG>
<AATGCACC, CCGTACAA>
<AATGCTGC, CCGTAGTA>
<AATGCTCG, CCGTAGAT>
<AATGCGTC, CCGTATGA>
<AATGCGGG, CCGTATTT>
<AATGCGCT, CCGTATAG>
<AATGCCAC, CCGTAACA>
<AATGCCTG, CCGTAAGT>
<AATGCCGT, CCGTAATG>
<AATGCCCA, CCGTAAAC>
<AATCAGCC, CCGACTAA>
<AATCACGC, CCGACATA>
<AATCACCG, CCGACAAT>
<AATCTTCC, CCGAGGAA>
<AATCTGGC, CCGAGTTA>
<AATCTGCG, CCGAGTAT>
<AATCTCTC, CCGAGAGA>
<AATCTCGG, CCGAGATT>
<AATCTCCT, CCGAGAAG>
<AATCGACC, CCGATCAA>
<AATCGTGC, CCGATGTA>
<AATCGTCG, CCGATGAT>
<AATCGGTC, CCGATTGA>
<AATCGGGG, CCGATTTT>
<AATCGGCT, CCGATTAG>
<AATCGCAC, CCGATACA>
<AATCGCTG, CCGATAGT>
<AATCGCGT, CCGATATG>
<AATCGCCA, CCGATAAC>
<AATCCAGC, CCGAACTA>
<AATCCACG, CCGAACAT>
<AATCCTTC, CCGAAGGA>
<AATCCTGG, CCGAAGTT>
<AATCCTCT, CCGAAGAG>
<AATCCGAC, CCGAATCA>
<AATCCGTG, CCGAATGT>
<AATCCGGT, CCGAATTG>
<AATCCGCA, CCGAATAC>
<AATCCCAG, CCGAAACT>
<AATCCCTT, CCGAAAGG>
<AATCCCGA, CCGAAATC>
<AAGAAAAA, CCTCCCCC>
<AAGATCCC, CCTCGAAA>
<AAGAGGCC, CCTCTTAA>
<AAGAGCGC, CCTCTATA>
<AAGAGCCG, CCTCTAAT>
<AAGACTCC, CCTCAGAA>
<AAGACGGC, CCTCATTA>
<AAGACGCG, CCTCATAT>
<AAGACCTC, CCTCAAGA>
<AAGACCGG, CCTCAATT>
<AAGACCCT, CCTCAAAG>
<AAGTACCC, CCTGCAAA>
<AAGTTGCC, CCTGGTAA>
<AAGTTCGC, CCTGGATA>
<AAGTTCCG, CCTGGAAT>
<AAGTGTCC, CCTGTGAA>
<AAGTGGGC, CCTGTTTA>
<AAGTGGCG, CCTGTTAT>
<AAGTGCTC, CCTGTAGA>
<AAGTGCGG, CCTGTATT>
<AAGTGCCT, CCTGTAAG>
<AAGTCACC, CCTGACAA>
<AAGTCTGC, CCTGAGTA>
<AAGTCTCG, CCTGAGAT>
<AAGTCGTC, CCTGATGA>
<AAGTCGGG, CCTGATTT>
<AAGTCGCT, CCTGATAG>
<AAGTCCAC, CCTGAACA>
<AAGTCCTG, CCTGAAGT>
<AAGTCCGT, CCTGAATG>
<AAGTCCCA, CCTGAAAC>
<AAGGAGCC, CCTTCTAA>
<AAGGACGC, CCTTCATA>
<AAGGACCG, CCTTCAAT>
<AAGGTTCC, CCTTGGAA>
<AAGGTGGC, CCTTGTTA>
<AAGGTGCG, CCTTGTAT>
<AAGGTCTC, CCTTGAGA>
<AAGGTCGG, CCTTGATT>
<AAGGTCCT, CCTTGAAG>
<AAGGGACC, CCTTTCAA>
<AAGGGTGC, CCTTTGTA>
<AAGGGTCG, CCTTTGAT>
<AAGGGGTC, CCTTTTGA>
<AAGGGGGG, CCTTTTTT>
<AAGGGGCT, CCTTTTAG>
<AAGGGCAC, CCTTTACA>
<AAGGGCTG, CCTTTAGT>
<AAGGGCGT, CCTTTATG>
<AAGGGCCA, CCTTTAAC>
<AAGGCAGC, CCTTACTA>
<AAGGCACG, CCTTACAT>
<AAGGCTTC, CCTTAGGA>
<AAGGCTGG, CCTTAGTT>
<AAGGCTCT, CCTTAGAG>
<AAGGCGAC, CCTTATCA>
<AAGGCGTG, CCTTATGT>
<AAGGCGGT, CCTTATTG>
<AAGGCGCA, CCTTATAC>
<AAGGCCAG, CCTTAACT>
<AAGGCCTT, CCTTAAGG>
<AAGGCCGA, CCTTAATC>
<AAGCATCC, CCTACGAA>
<AAGCAGGC, CCTACTTA>
<AAGCAGCG, CCTACTAT>
<AAGCACTC, CCTACAGA>
<AAGCACGG, CCTACATT>
<AAGCACCT, CCTACAAG>
<AAGCTACC, CCTAGCAA>
<AAGCTTGC, CCTAGGTA>
<AAGCTTCG, CCTAGGAT>
<AAGCTGTC, CCTAGTGA>
<AAGCTGGG, CCTAGTTT>
<AAGCTGCT, CCTAGTAG>
<AAGCTCAC, CCTAGACA>
<AAGCTCTG, CCTAGAGT>
<AAGCTCGT, CCTAGATG>
<AAGCTCCA, CCTAGAAC>
<AAGCGAGC, CCTATCTA>
<AAGCGACG, CCTATCAT>
<AAGCGTTC, CCTATGGA>
<AAGCGTGG, CCTATGTT>
<AAGCGTCT, CCTATGAG>
<AAGCGGAC, CCTATTCA>
<AAGCGGTG, CCTATTGT>
<AAGCGGGT, CCTATTTG>
<AAGCGGCA, CCTATTAC>
<AAGCGCAG, CCTATACT>
<AAGCGCTT, CCTATAGG>
<AAGCGCGA, CCTATATC>
<AAGCCATC, CCTAACGA>
<AAGCCAGG, CCTAACTT>
<AAGCCACT, CCTAACAG>
<AAGCCTAC, CCTAAGCA>
<AAGCCTTG, CCTAAGGT>
<AAGCCTGT, CCTAAGTG>
<AAGCCTCA, CCTAAGAC>
<AAGCCGAG, CCTAATCT>
<AAGCCGTT, CCTAATGG>
<AAGCCGGA, CCTAATTC>
<AAGCCCAT, CCTAAACG>
<AAGCCCTA, CCTAAAGC>
<AACAACCC, CCACCAAA>
<AACATGCC, CCACGTAA>
<AACATCGC, CCACGATA>
<AACATCCG, CCACGAAT>
<AACAGTCC, CCACTGAA>
<AACAGGGC, CCACTTTA>
<AACAGGCG, CCACTTAT>
<AACAGCTC, CCACTAGA>
<AACAGCGG, CCACTATT>
<AACAGCCT, CCACTAAG>
<AACACACC, CCACACAA>
<AACACTGC, CCACAGTA>
<AACACTCG, CCACAGAT>
<AACACGTC, CCACATGA>
<AACACGGG, CCACATTT>
<AACACGCT, CCACATAG>
<AACACCAC, CCACAACA>
<AACACCTG, CCACAAGT>
<AACACCGT, CCACAATG>
<AACACCCA, CCACAAAC>
<AACTAGCC, CCAGCTAA>
<AACTACGC, CCAGCATA>
<AACTACCG, CCAGCAAT>
<AACTTTCC, CCAGGGAA>
<AACTTGGC, CCAGGTTA>
<AACTTGCG, CCAGGTAT>
<AACTTCTC, CCAGGAGA>
<AACTTCGG, CCAGGATT>
<AACTTCCT, CCAGGAAG>
<AACTGACC, CCAGTCAA>
<AACTGTGC, CCAGTGTA>
<AACTGTCG, CCAGTGAT>
<AACTGGTC, CCAGTTGA>
<AACTGGGG, CCAGTTTT>
<AACTGGCT, CCAGTTAG>
<AACTGCAC, CCAGTACA>
<AACTGCTG, CCAGTAGT>
<AACTGCGT, CCAGTATG>
<AACTGCCA, CCAGTAAC>
<AACTCAGC, CCAGACTA>
<AACTCACG, CCAGACAT>
<AACTCTTC, CCAGAGGA>

<AACTCTGG, CCAGAGTT> <AACTCTCT, CCAGAGAG> <AACTCGAC, CCAGATCA> <AACTCGTG, CCAGATGT>
<AACTCGGT, CCAGATTG> <AACTCGCA, CCAGATAC> <AACTCCAG, CCAGAACT> <AACTCCTT, CCAGAAGG>
<AACTCCGA, CCAGAATC> <AACGATCC, CCATCGAA> <AACGAGGC, CCATCTTA> <AACGAGCG, CCATCTAT>
<AACGACTC, CCATCAGA> <AACGACGG, CCATCATT> <AACGACCT, CCATCAAG> <AACGTACC, CCATGCAA>
<AACGTTGC, CCATGGTA> <AACGTTCG, CCATGGAT> <AACGTGTC, CCATGTGA> <AACGTGGG, CCATGTTT>
<AACGTGCT, CCATGTAG> <AACGTCAC, CCATGACA> <AACGTCTG, CCATGAGT> <AACGTCGT, CCATGATG>
<AACGTCCA, CCATGAAC> <AACGGAGC, CCATTCTA> <AACGGACG, CCATTCAT> <AACGGTTC, CCATTGGA>
<AACGGTGG, CCATTGTT> <AACGGTCT, CCATTGAG> <AACGGGAC, CCATTTCA> <AACGGGTG, CCATTTGT>
<AACGGGGT, CCATTTTG> <AACGGGCA, CCATTTAC> <AACGGCAG, CCATTACT> <AACGGCTT, CCATTAGG>
<AACGGCGA, CCATTATC> <AACGCATC, CCATACGA> <AACGCAGG, CCATACTT> <AACGCACT, CCATACAG>
<AACGCTAC, CCATAGCA> <AACGCTTG, CCATAGGT> <AACGCTGT, CCATAGTG> <AACGCTCA, CCATAGAC>
<AACGCGAG, CCATATCT> <AACGCGTT, CCATATGG> <AACGCGGA, CCATATTC> <AACGCCAT, CCATAACG>
<AACGCCTA, CCATAAGC> <AACCAACC, CCAACCAA> <AACCATGC, CCAACGTA> <AACCATCG, CCAACGAT>
<AACCAGTC, CCAACTGA> <AACCAGGG, CCAACTTT> <AACCAGCT, CCAACTAG> <AACCACAC, CCAACACA>
<AACCACTG, CCAACAGT> <AACCACGT, CCAACATG> <AACCACCA, CCAACAAC> <AACCTAGC, CCAAGCTA>
<AACCTACG, CCAAGCAT> <AACCTTTC, CCAAGGGA> <AACCTTGG, CCAAGGTT> <AACCTTCT, CCAAGGAG>
<AACCTGAC, CCAAGTCA> <AACCTGTG, CCAAGTGT> <AACCTGGT, CCAAGTTG> <AACCTGCA, CCAAGTAC>
<AACCTCAG, CCAAGACT> <AACCTCTT, CCAAGAGG> <AACCTCGA, CCAAGATC> <AACCGATC, CCAATCGA>
<AACCGAGG, CCAATCTT> <AACCGACT, CCAATCAG> <AACCGTAC, CCAATGCA> <AACCGTTG, CCAATGGT>
<AACCGTGT, CCAATGTG> <AACCGTCA, CCAATGAC> <AACCGGAG, CCAATTCT> <AACCGGTT, CCAATTGG>
<AACCGGGA, CCAATTTC> <AACCGCAT, CCAATACG> <AACCGCTA, CCAATAGC> <AACCCAAC, CCAAACCA>
<AACCCATG, CCAAACGT> <AACCCAGT, CCAAACTG> <AACCCACA, CCAAACAC> <AACCCTAG, CCAAAGCT>
<AACCCTTT, CCAAAGGG> <AACCCTGA, CCAAAGTC> <AACCCGAT, CCAAATCG> <AACCCGTA, CCAAATGC>
<AACCCCAA, CCAAAACC> <ATAAAAAT, CGCCCCCG> <ATAAAATA, CGCCCCGC> <ATAAATAA, CGCCCGCC>
<ATAATAAA, CGCCGCCC> <ATAAGCCC, CGCCTAAA> <ATAACGCC, CGCCATAA> <ATAACCGC, CGCCAATA>
<ATAACCCG, CGCCAAAT> <ATATAAAA, CGCGCCCC> <ATATTCCC, CGCGGAAA> <ATATGGCC, CGCGTTAA>
<ATATGCGC, CGCGTATA> <ATATGCCG, CGCGTAAT> <ATATCTCC, CGCGAGAA> <ATATCGGC, CGCGATTA>
<ATATCGCG, CGCGATAT> <ATATCCTC, CGCGAAGA> <ATATCCGG, CGCGAATT> <ATATCCCT, CGCGAAAG>
<ATAGACCC, CGCTCAAA> <ATAGTGCC, CGCTGTAA> <ATAGTCGC, CGCTGATA> <ATAGTCCG, CGCTGAAT>
<ATAGGTCC, CGCTTGAA> <ATAGGGGC, CGCTTTTA> <ATAGGGCG, CGCTTTAT> <ATAGGCTC, CGCTTAGA>
<ATAGGCGG, CGCTTATT> <ATAGGCCT, CGCTTAAG> <ATAGCACC, CGCTACAA> <ATAGCTGC, CGCTAGTA>
<ATAGCTCG, CGCTAGAT> <ATAGCGTC, CGCTATGA> <ATAGCGGG, CGCTATTT> <ATAGCGCT, CGCTATAG>
<ATAGCCAC, CGCTAACA> <ATAGCCTG, CGCTAAGT> <ATAGCCGT, CGCTAATG> <ATAGCCCA, CGCTAAAC>
<ATACAGCC, CGCACTAA> <ATACACGC, CGCACATA> <ATACACCG, CGCACAAT> <ATACTTCC, CGCAGGAA>
<ATACTGGC, CGCAGTTA> <ATACTGCG, CGCAGTAT> <ATACTCTC, CGCAGAGA> <ATACTCGG, CGCAGATT>
<ATACTCCT, CGCAGAAG> <ATACGACC, CGCATCAA> <ATACGTGC, CGCATGTA> <ATACGTCG, CGCATGAT>
<ATACGGTC, CGCATTGA> <ATACGGGG, CGCATTTT> <ATACGGCT, CGCATTAG> <ATACGCAC, CGCATACA>
<ATACGCTG, CGCATAGT> <ATACGCGT, CGCATATG> <ATACGCCA, CGCATAAC> <ATACCAGC, CGCAACTA>
<ATACCACG, CGCAACAT> <ATACCTTC, CGCAAGGA> <ATACCTGG, CGCAAGTT> <ATACCTCT, CGCAAGAG>
<ATACCGAC, CGCAATCA> <ATACCGTG, CGCAATGT> <ATACCGGT, CGCAATTG> <ATACCGCA, CGCAATAC>
<ATACCCAG, CGCAAACT> <ATACCCTT, CGCAAAGG> <ATACCCGA, CGCAAATC> <ATTAAAAA, CGGCCCCC>
<ATTATCCC, CGGCGAAA> <ATTAGGCC, CGGCTTAA> <ATTAGCGC, CGGCTATA> <ATTAGCCG, CGGCTAAT>
<ATTACTCC, CGGCAGAA> <ATTACGGC, CGGCATTA> <ATTACGCG, CGGCATAT> <ATTACCTC, CGGCAAGA>
<ATTACCGG, CGGCAATT> <ATTACCCT, CGGCAAAG> <ATTTACCC, CGGGCAAA> <ATTTTGCC, CGGGGTAA>
<ATTTTCGC, CGGGGATA> <ATTTTCCG, CGGGGAAT> <ATTTGTCC, CGGGTGAA> <ATTTGGGC, CGGGTTTA>
<ATTTGGCG, CGGGTTAT> <ATTTGCTC, CGGGTAGA> <ATTTGCGG, CGGGTATT> <ATTTGCCT, CGGGTAAG>
<ATTTCACC, CGGGACAA> <ATTTCTGC, CGGGAGTA> <ATTTCTCG, CGGGAGAT> <ATTTCGTC, CGGGATGA>
<ATTTCGGG, CGGGATTT> <ATTTCGCT, CGGGATAG> <ATTTCCAC, CGGGAACA> <ATTTCCTG, CGGGAAGT>
<ATTTCCGT, CGGGAATG> <ATTTCCCA, CGGGAAAC> <ATTGAGCC, CGGTCTAA> <ATTGACGC, CGGTCATA>
<ATTGACCG, CGGTCAAT> <ATTGTTCC, CGGTGGAA> <ATTGTGGC, CGGTGTTA> <ATTGTGCG, CGGTGTAT>
<ATTGTCTC, CGGTGAGA> <ATTGTCGG, CGGTGATT> <ATTGTCCT, CGGTGAAG> <ATTGGACC, CGGTTCAA>
<ATTGGTGC, CGGTTGTA> <ATTGGTCG, CGGTTGAT> <ATTGGGTC, CGGTTTGA> <ATTGGGGG, CGGTTTTT>
<ATTGGGCT, CGGTTTAG> <ATTGGCAC, CGGTTACA> <ATTGGCTG, CGGTTAGT> <ATTGGCGT, CGGTTATG>
<ATTGGCCA, CGGTTAAC> <ATTGCAGC, CGGTACTA> <ATTGCACG, CGGTACAT> <ATTGCTTC, CGGTAGGA>
<ATTGCTGG, CGGTAGTT> <ATTGCTCT, CGGTAGAG> <ATTGCGAC, CGGTATCA> <ATTGCGTG, CGGTATGT>
<ATTGCGGT, CGGTATTG> <ATTGCGCA, CGGTATAC> <ATTGCCAG, CGGTAACT> <ATTGCCTT, CGGTAAGG>
<ATTGCCGA, CGGTAATC> <ATTCATCC, CGGACGAA> <ATTCAGGC, CGGACTTA> <ATTCAGCG, CGGACTAT>
<ATTCACTC, CGGACAGA> <ATTCACGG, CGGACATT> <ATTCACCT, CGGACAAG> <ATTCTACC, CGGAGCAA>
<ATTCTTGC, CGGAGGTA> <ATTCTTCG, CGGAGGAT> <ATTCTGTC, CGGAGTGA> <ATTCTGGG, CGGAGTTT>
<ATTCTGCT, CGGAGTAG> <ATTCTCAC, CGGAGACA> <ATTCTCTG, CGGAGAGT> <ATTCTCGT, CGGAGATG>
<ATTCTCCA, CGGAGAAC> <ATTCGAGC, CGGATCTA> <ATTCGACG, CGGATCAT> <ATTCGTTC, CGGATGGA>
<ATTCGTGG, CGGATGTT> <ATTCGTCT, CGGATGAG> <ATTCGGAC, CGGATTCA> <ATTCGGTG, CGGATTGT>
<ATTCGGGT, CGGATTTG> <ATTCGGCA, CGGATTAC> <ATTCGCAG, CGGATACT> <ATTCGCTT, CGGATAGG>
<ATTCGCGA, CGGATATC> <ATTCCATC, CGGAACGA> <ATTCCAGG, CGGAACTT> <ATTCCACT, CGGAACAG>
<ATTCCTAC, CGGAAGCA> <ATTCCTTG, CGGAAGGT> <ATTCCTGT, CGGAAGTG> <ATTCCTCA, CGGAAGAC>
<ATTCCGAG, CGGAATCT> <ATTCCGTT, CGGAATGG> <ATTCCGGA, CGGAATTC> <ATTCCCAT, CGGAAACG>
<ATTCCCTA, CGGAAAGC> <ATGAACCC, CGTCCAAA> <ATGATGCC, CGTCGTAA> <ATGATCGC, CGTCGATA>
<ATGATCCG, CGTCGAAT> <ATGAGTCC, CGTCTGAA> <ATGAGGGC, CGTCTTTA> <ATGAGGCG, CGTCTTAT>
<ATGAGCTC, CGTCTAGA> <ATGAGCGG, CGTCTATT> <ATGAGCCT, CGTCTAAG> <ATGACACC, CGTCACAA>
<ATGACTGC, CGTCAGTA> <ATGACTCG, CGTCAGAT> <ATGACGTC, CGTCATGA> <ATGACGGG, CGTCATTT>
<ATGACGCT, CGTCATAG> <ATGACCAC, CGTCAACA> <ATGACCTG, CGTCAAGT> <ATGACCGT, CGTCAATG>
<ATGACCCA, CGTCAAAC> <ATGTAGCC, CGTGCTAA> <ATGTACGC, CGTGCATA> <ATGTACCG, CGTGCAAT>
<ATGTTTCC, CGTGGGAA> <ATGTTGGC, CGTGGTTA> <ATGTTGCG, CGTGGTAT> <ATGTTCTC, CGTGGAGA>
<ATGTTCGG, CGTGGATT> <ATGTTCCT, CGTGGAAG> <ATGTGACC, CGTGTCAA> <ATGTGTGC, CGTGTGTA>
<ATGTGTCG, CGTGTGAT> <ATGTGGTC, CGTGTTGA> <ATGTGGGG, CGTGTTTT> <ATGTGGCT, CGTGTTAG>
<ATGTGCAC, CGTGTACA> <ATGTGCTG, CGTGTAGT> <ATGTGCGT, CGTGTATG> <ATGTGCCA, CGTGTAAC>
<ATGTCAGC, CGTGACTA> <ATGTCACG, CGTGACAT> <ATGTCTTC, CGTGAGGA> <ATGTCTGG, CGTGAGTT>
<ATGTCTCT, CGTGAGAG> <ATGTCGAC, CGTGATCA> <ATGTCGTG, CGTGATGT> <ATGTCGGT, CGTGATTG>
<ATGTCGCA, CGTGATAC> <ATGTCCAG, CGTGAACT> <ATGTCCTT, CGTGAAGG> <ATGTCCGA, CGTGAATC>
<ATGGATCC, CGTTCGAA> <ATGGAGGC, CGTTCTTA> <ATGGAGCG, CGTTCTAT> <ATGGACTC, CGTTCAGA>

<ATGGACGG, CGTTCATT>
<ATGGTTCG, CGTTGGAT>
<ATGGTCAC, CGTTGACA>
<ATGGGAGC, CGTTTCTA>
<ATGGGTCT, CGTTTGAG>
<ATGGGGCA, CGTTTTAC>
<ATGGCATC, CGTTACGA>
<ATGGCTTG, CGTTAGGT>
<ATGGCGTT, CGTTATGG>
<ATGCAACC, CGTACCAA>
<ATGCAGGG, CGTACTTT>
<ATGCACGT, CGTACATG>
<ATGCTTTC, CGTAGGGA>
<ATGCTGTG, CGTAGTGT>
<ATGCTCTT, CGTAGAGG>
<ATGCGACT, CGTATCAG>
<ATGCGTCA, CGTATGAC>
<ATGCGCAT, CGTATACG>
<ATGCCAGT, CGTAACTG>
<ATGCCTGA, CGTAAGTC>
<ATCAAGCC, CGACCTAA>
<ATCATGGC, CGACGTTA>
<ATCATCCT, CGACGAAG>
<ATCAGGTC, CGACTTGA>
<ATCAGCTG, CGACTAGT>
<ATCACACG, CGACACAT>
<ATCACGAC, CGACATCA>
<ATCACCAG, CGACAACT>
<ATCTAGGC, CGAGCTTA>
<ATCTACCT, CGAGCAAG>
<ATCTTGTC, CGAGGTGA>
<ATCTTCTG, CGAGGAGT>
<ATCTGACG, CGAGTCAT>
<ATCTGGAC, CGAGTTCA>
<ATCTGCAG, CGAGTACT>
<ATCTCAGG, CGAGACTT>
<ATCTCTGT, CGAGAGTG>
<ATCTCGGA, CGAGATTC>
<ATCGATGC, CGATCGTA>
<ATCGAGCT, CGATCTAG>
<ATCGACCA, CGATCAAC>
<ATCGTTGG, CGATGGTT>
<ATCGTGGT, CGATGTTG>
<ATCGTCGA, CGATGATC>
<ATCGGTAC, CGATTGCA>
<ATCGGGAG, CGATTTCT>
<ATCGGCTA, CGATTAGC>
<ATCGCACA, CGATACAC>
<ATCGCGAT, CGATATCG>
<ATCCAACG, CGAACCAT>
<ATCCAGAC, CGAACTCA>
<ATCCACAG, CGAACACT>
<ATCCTAGG, CGAAGCTT>
<ATCCTTGT, CGAAGGTG>
<ATCCTGGA, CGAAGTTC>
<ATCCGATG, CGAATCGT>
<ATCCGTTT, CGAATGGG>
<ATCCGCAA, CGAATACC>
<ATCCCTAT, CGAAAGCG>
<AGAATCCC, CTCCGAAA>
<AGAACTCC, CTCCAGAA>
<AGAACCGG, CTCCAATT>
<AGATTCGC, CTCGGATA>
<AGATGGCG, CTCGTTAT>
<AGATCACC, CTCGACAA>
<AGATCGGG, CTCGATTT>
<AGATCCGT, CTCGAATG>
<AGAGACCG, CTCTCAAT>
<AGAGTCTC, CTCTGAGA>
<AGAGGTGC, CTCTTGTA>
<AGAGGGCT, CTCTTTAG>
<AGAGGCCA, CTCTTAAC>
<AGAGCTGG, CTCTAGTT>
<AGAGCGGT, CTCTATTG>
<AGAGCCGA, CTCTAATC>
<AGACACTC, CTCACAGA>
<AGACTTGC, CTCAGGTA>
<AGACTGCT, CTCAGTAG>
<AGACTCCA, CTCAGAAC>
<AGACGTGG, CTCATGTT>
<ATGGACCT, CGTTCAAG>
<ATGGTGTC, CGTTGTGA>
<ATGGTCTG, CGTTGAGT>
<ATGGGACG, CGTTTCAT>
<ATGGGGAC, CGTTTTCA>
<ATGGGCAG, CGTTTACT>
<ATGGCAGG, CGTTACTT>
<ATGGCTGT, CGTTAGTG>
<ATGGCGGA, CGTTATTC>
<ATGCATGC, CGTACGTA>
<ATGCAGCT, CGTACTAG>
<ATGCACCA, CGTACAAC>
<ATGCTTGG, CGTAGGTT>
<ATGCTGGT, CGTAGTTG>
<ATGCTCGA, CGTAGATC>
<ATGCGTAC, CGTATGCA>
<ATGCGGAG, CGTATTCT>
<ATGCGCTA, CGTATAGC>
<ATGCCACA, CGTAACAC>
<ATGCCGAT, CGTAATCG>
<ATCAACGC, CGACCATA>
<ATCATGCG, CGACGTAT>
<ATCAGACC, CGACTCAA>
<ATCAGGGG, CGACTTTT>
<ATCAGCGT, CGACTATG>
<ATCACTTC, CGACAGGA>
<ATCACGTG, CGACATGT>
<ATCACCTT, CGACAAGG>
<ATCTAGCG, CGAGCTAT>
<ATCTTACC, CGAGGCAA>
<ATCTTGGG, CGAGGTTT>
<ATCTTCGT, CGAGGATG>
<ATCTGTTC, CGAGTGGA>
<ATCTGGTG, CGAGTTGT>
<ATCTGCTT, CGAGTAGG>
<ATCTCACT, CGAGACAG>
<ATCTCTCA, CGAGAGAC>
<ATCTCCAT, CGAGAACG>
<ATCGATCG, CGATCGAT>
<ATCGACAC, CGATCACA>
<ATCGTAGC, CGATGCTA>
<ATCGTTCT, CGATGGAG>
<ATCGTGCA, CGATGTAC>
<ATCGGATC, CGATTCGA>
<ATCGGTTG, CGATTGGT>
<ATCGGGTT, CGATTTGG>
<ATCGCAAC, CGATACCA>
<ATCGCTAG, CGATAGCT>
<ATCGCGTA, CGATATGC>
<ATCCATTC, CGAACGGA>
<ATCCAGTG, CGAACTGT>
<ATCCACTT, CGAACAGG>
<ATCCTACT, CGAAGCAG>
<ATCCTTCA, CGAAGGAC>
<ATCCTCAT, CGAAGACG>
<ATCCGAGT, CGAATCTG>
<ATCCGTGA, CGAATGTC>
<ATCCCAAG, CGAAACCT>
<ATCCCTTA, CGAAAGGC>
<AGAAGGCC, CTCCTTAA>
<AGAACGGC, CTCCATTA>
<AGAACCCT, CTCCAAAG>
<AGATTCCG, CTCGGAAT>
<AGATGCTC, CTCGTAGA>
<AGATCTGC, CTCGAGTA>
<AGATCGCT, CTCGATAG>
<AGATCCCA, CTCGAAAC>
<AGAGTTCC, CTCTGGAA>
<AGAGTCGG, CTCTGATT>
<AGAGGTCG, CTCTTGAT>
<AGAGGCAC, CTCTTACA>
<AGAGCAGC, CTCTACTA>
<AGAGCTCT, CTCTAGAG>
<AGAGCGCA, CTCTATAC>
<AGACATCC, CTCACGAA>
<AGACACGG, CTCACATT>
<AGACTTCG, CTCAGGAT>
<AGACTCAC, CTCAGACA>
<AGACGAGC, CTCATCTA>
<AGACGTCT, CTCATGAG>
<ATGGTACC, CGTTGCAA>
<ATGGTGGG, CGTTGTTT>
<ATGGTCGT, CGTTGATG>
<ATGGGTTC, CGTTTGGA>
<ATGGGGTG, CGTTTTGT>
<ATGGGCTT, CGTTTAGG>
<ATGGCACT, CGTTACAG>
<ATGGCTCA, CGTTAGAC>
<ATGGCCAT, CGTTAACG>
<ATGCATCG, CGTACGAT>
<ATGCACAC, CGTACACA>
<ATGCTAGC, CGTAGCTA>
<ATGCTTCT, CGTAGGAG>
<ATGCTGCA, CGTAGTAC>
<ATGCGATC, CGTATCGA>
<ATGCGTTG, CGTATGGT>
<ATGCGGTT, CGTATTGG>
<ATGCCAAC, CGTAACCA>
<ATGCCTAG, CGTAAGCT>
<ATGCCGTA, CGTAATGC>
<ATCAACCG, CGACCAAT>
<ATCATCTC, CGACGAGA>
<ATCAGTGC, CGACTGTA>
<ATCAGGCT, CGACTTAG>
<ATCAGCCA, CGACTAAC>
<ATCACTGG, CGACAGTT>
<ATCACGGT, CGACATTG>
<ATCACCGA, CGACAATC>
<ATCTACTC, CGAGCAGA>
<ATCTTTGC, CGAGGGTA>
<ATCTTGCT, CGAGGTAG>
<ATCTTCCA, CGAGGAAC>
<ATCTGTGG, CGAGTGTT>
<ATCTGGGT, CGAGTTTG>
<ATCTGCGA, CGAGTATC>
<ATCTCTAC, CGAGAGCA>
<ATCTCGAG, CGAGATCT>
<ATCTCCTA, CGAGAAGC>
<ATCGAGTC, CGATCTGA>
<ATCGACTG, CGATCAGT>
<ATCGTACG, CGATGCAT>
<ATCGTGAC, CGATGTCA>
<ATCGTCAG, CGATGACT>
<ATCGGAGG, CGATTCTT>
<ATCGGTGT, CGATTGTG>
<ATCGGGGA, CGATTTTC>
<ATCGCATG, CGATACGT>
<ATCGCTTT, CGATAGGG>
<ATCGCCAA, CGATAACC>
<ATCCATGG, CGAACGTT>
<ATCCAGGT, CGAACTTG>
<ATCCACGA, CGAACATC>
<ATCCTTAC, CGAAGGCA>
<ATCCTGAG, CGAAGTCT>
<ATCCTCTA, CGAAGAGC>
<ATCCGACA, CGAATCAC>
<ATCCGGAT, CGAATTCG>
<ATCCCATT, CGAAACGG>
<ATCCCGAA, CGAAATCC>
<AGAAGCGC, CTCCTATA>
<AGAACGCG, CTCCATAT>
<AGATACCC, CTCGCAAA>
<AGATGTCC, CTCGTGAA>
<AGATGCGG, CTCGTATT>
<AGATCTCG, CTCGAGAT>
<AGATCCAC, CTCGAACA>
<AGAGAGCC, CTCTCTAA>
<AGAGTGGC, CTCTGTTA>
<AGAGTCCT, CTCTGAAG>
<AGAGGGTC, CTCTTTGA>
<AGAGGCTG, CTCTTAGT>
<AGAGCACG, CTCTACAT>
<AGAGCGAC, CTCTATCA>
<AGAGCCAG, CTCTAACT>
<AGACAGGC, CTCACTTA>
<AGACACCT, CTCACAAG>
<AGACTGTC, CTCAGTGA>
<AGACTCTG, CTCAGAGT>
<AGACGACG, CTCATCAT>
<AGACGGAC, CTCATTCA>
<ATGGTTGC, CGTTGGTA>
<ATGGTGCT, CGTTGTAG>
<ATGGTCCA, CGTTGAAC>
<ATGGGTGG, CGTTTGTT>
<ATGGGGGT, CGTTTTTG>
<ATGGGCGA, CGTTTATC>
<ATGGCTAC, CGTTAGCA>
<ATGGCGAG, CGTTATCT>
<ATGGCCTA, CGTTAAGC>
<ATGCAGTC, CGTACTGA>
<ATGCACTG, CGTACAGT>
<ATGCTACG, CGTAGCAT>
<ATGCTGAC, CGTAGTCA>
<ATGCTCAG, CGTAGACT>
<ATGCGAGG, CGTATCTT>
<ATGCGTGT, CGTATGTG>
<ATGCGGGA, CGTATTTC>
<ATGCCATG, CGTAACGT>
<ATGCCTTT, CGTAAGGG>
<ATGCCCAA, CGTAAACC>
<ATCATTCC, CGACGGAA>
<ATCATCGG, CGACGATT>
<ATCAGTCG, CGACTGAT>
<ATCAGCAC, CGACTACA>
<ATCACAGC, CGACACTA>
<ATCACTCT, CGACAGAG>
<ATCACGCA, CGACATAC>
<ATCTATCC, CGAGCGAA>
<ATCTACGG, CGAGCATT>
<ATCTTTCG, CGAGGGAT>
<ATCTTCAC, CGAGGACA>
<ATCTGAGC, CGAGTCTA>
<ATCTGTCT, CGAGTGAG>
<ATCTGGCA, CGAGTTAC>
<ATCTCATC, CGAGACGA>
<ATCTCTTG, CGAGAGGT>
<ATCTCGTT, CGAGATGG>
<ATCGAACC, CGATCCAA>
<ATCGAGGG, CGATCTTT>
<ATCGACGT, CGATCATG>
<ATCGTTTC, CGATGGGA>
<ATCGTGTG, CGATGTGT>
<ATCGTCTT, CGATGAGG>
<ATCGGACT, CGATTCAG>
<ATCGGTCA, CGATTGAC>
<ATCGGCAT, CGATTACG>
<ATCGCAGT, CGATACTG>
<ATCGCTGA, CGATAGTC>
<ATCCAAGC, CGAACCTA>
<ATCCATCT, CGAACGAG>
<ATCCAGCA, CGAACTAC>
<ATCCTATC, CGAAGCGA>
<ATCCTTTG, CGAAGGGT>
<ATCCTGTT, CGAAGTGG>
<ATCCGAAC, CGAATCCA>
<ATCCGTAG, CGAATGCT>
<ATCCGGTA, CGAATTGC>
<ATCCCAGA, CGAAACTC>
<AGAAAAAA, CTCCCCCC>
<AGAAGCCG, CTCCTAAT>
<AGAACCTC, CTCCAAGA>
<AGATTGCC, CTCGGTAA>
<AGATGGGC, CTCGTTTA>
<AGATGCCT, CTCGTAAG>
<AGATCGTC, CTCGATGA>
<AGATCCTG, CTCGAAGT>
<AGAGACGC, CTCTCATA>
<AGAGTGCG, CTCTGTAT>
<AGAGGACC, CTCTTCAA>
<AGAGGGGG, CTCTTTTT>
<AGAGGCGT, CTCTTATG>
<AGAGCTTC, CTCTAGGA>
<AGAGCGTG, CTCTATGT>
<AGAGCCTT, CTCTAAGG>
<AGACAGCG, CTCACTAT>
<AGACTACC, CTCAGCAA>
<AGACTGGG, CTCAGTTT>
<AGACTCGT, CTCAGATG>
<AGACGTTC, CTCATGGA>
<AGACGGTG, CTCATTGT>

<AGACGGGT, CTCATTTG> <AGACGGCA, CTCATTAC> <AGACGCAG, CTCATACT> <AGACGCTT, CTCATAGG>
<AGACGCGA, CTCATATC> <AGACCATC, CTCAACGA> <AGACCAGG, CTCAACTT> <AGACCACT, CTCAACAG>
<AGACCTAC, CTCAAGCA> <AGACCTTG, CTCAAGGT> <AGACCTGT, CTCAAGTG> <AGACCTCA, CTCAAGAC>
<AGACCGAG, CTCAATCT> <AGACCGTT, CTCAATGG> <AGACCGGA, CTCAATTC> <AGACCCAT, CTCAAACG>
<AGACCCTA, CTCAAAGC> <AGTAACCC, CTGCCAAA> <AGTATGCC, CTGCGTAA> <AGTATCGC, CTGCGATA>
<AGTATCCG, CTGCGAAT> <AGTAGTCC, CTGCTGAA> <AGTAGGGC, CTGCTTTA> <AGTAGGCG, CTGCTTAT>
<AGTAGCTC, CTGCTAGA> <AGTAGCGG, CTGCTATT> <AGTAGCCT, CTGCTAAG> <AGTACACC, CTGCACAA>
<AGTACTGC, CTGCAGTA> <AGTACTCG, CTGCAGAT> <AGTACGTC, CTGCATGA> <AGTACGGG, CTGCATTT>
<AGTACGCT, CTGCATAG> <AGTACCAC, CTGCAACA> <AGTACCTG, CTGCAAGT> <AGTACCGT, CTGCAATG>
<AGTACCCA, CTGCAAAC> <AGTTAGCC, CTGGCTAA> <AGTTACGC, CTGGCATA> <AGTTACCG, CTGGCAAT>
<AGTTTTCC, CTGGGGAA> <AGTTTGGC, CTGGGTTA> <AGTTTGCG, CTGGGTAT> <AGTTTCTC, CTGGGAGA>
<AGTTTCGG, CTGGGATT> <AGTTTCCT, CTGGGAAG> <AGTTGACC, CTGGTCAA> <AGTTGTGC, CTGGTGTA>
<AGTTGTCG, CTGGTGAT> <AGTTGGTC, CTGGTTGA> <AGTTGGGG, CTGGTTTT> <AGTTGGCT, CTGGTTAG>
<AGTTGCAC, CTGGTACA> <AGTTGCTG, CTGGTAGT> <AGTTGCGT, CTGGTATG> <AGTTGCCA, CTGGTAAC>
<AGTTCAGC, CTGGACTA> <AGTTCACG, CTGGACAT> <AGTTCTTC, CTGGAGGA> <AGTTCTGG, CTGGAGTT>
<AGTTCTCT, CTGGAGAG> <AGTTCGAC, CTGGATCA> <AGTTCGTG, CTGGATGT> <AGTTCGGT, CTGGATTG>
<AGTTCGCA, CTGGATAC> <AGTTCCAG, CTGGAACT> <AGTTCCTT, CTGGAAGG> <AGTTCCGA, CTGGAATC>
<AGTGATCC, CTGTCGAA> <AGTGAGGC, CTGTCTTA> <AGTGAGCG, CTGTCTAT> <AGTGACTC, CTGTCAGA>
<AGTGACGG, CTGTCATT> <AGTGACCT, CTGTCAAG> <AGTGTACC, CTGTGCAA> <AGTGTTGC, CTGTGGTA>
<AGTGTTCG, CTGTGGAT> <AGTGTGTC, CTGTGTGA> <AGTGTGGG, CTGTGTTT> <AGTGTGCT, CTGTGTAG>
<AGTGTCAC, CTGTGACA> <AGTGTCTG, CTGTGAGT> <AGTGTCGT, CTGTGATG> <AGTGTCCA, CTGTGAAC>
<AGTGGAGC, CTGTTCTA> <AGTGGACG, CTGTTCAT> <AGTGGTTC, CTGTTGGA> <AGTGGTGG, CTGTTGTT>
<AGTGGTCT, CTGTTGAG> <AGTGGGAC, CTGTTTCA> <AGTGGGTG, CTGTTTGT> <AGTGGGGT, CTGTTTTG>
<AGTGGGCA, CTGTTTAC> <AGTGGCAG, CTGTTACT> <AGTGGCTT, CTGTTAGG> <AGTGGCGA, CTGTTATC>
<AGTGCATC, CTGTACGA> <AGTGCAGG, CTGTACTT> <AGTGCACT, CTGTACAG> <AGTGCTAC, CTGTAGCA>
<AGTGCTTG, CTGTAGGT> <AGTGCTGT, CTGTAGTG> <AGTGCTCA, CTGTAGAC> <AGTGCGAG, CTGTATCT>
<AGTGCGTT, CTGTATGG> <AGTGCGGA, CTGTATTC> <AGTGCCAT, CTGTAACG> <AGTGCCTA, CTGTAAGC>
<AGTCAACC, CTGACCAA> <AGTCATGC, CTGACGTA> <AGTCATCG, CTGACGAT> <AGTCAGTC, CTGACTGA>
<AGTCAGGG, CTGACTTT> <AGTCAGCT, CTGACTAG> <AGTCACAC, CTGACACA> <AGTCACTG, CTGACAGT>
<AGTCACGT, CTGACATG> <AGTCACCA, CTGACAAC> <AGTCTAGC, CTGAGCTA> <AGTCTACG, CTGAGCAT>
<AGTCTTTC, CTGAGGGA> <AGTCTTGG, CTGAGGTT> <AGTCTTCT, CTGAGGAG> <AGTCTGAC, CTGAGTCA>
<AGTCTGTG, CTGAGTGT> <AGTCTGGT, CTGAGTTG> <AGTCTGCA, CTGAGTAC> <AGTCTCAG, CTGAGACT>
<AGTCTCTT, CTGAGAGG> <AGTCTCGA, CTGAGATC> <AGTCGATC, CTGATCGA> <AGTCGAGG, CTGATCTT>
<AGTCGACT, CTGATCAG> <AGTCGTAC, CTGATGCA> <AGTCGTTG, CTGATGGT> <AGTCGTGT, CTGATGTG>
<AGTCGTCA, CTGATGAC> <AGTCGGAG, CTGATTCT> <AGTCGGTT, CTGATTGG> <AGTCGGGA, CTGATTTC>
<AGTCGCAT, CTGATACG> <AGTCGCTA, CTGATAGC> <AGTCCAAC, CTGAACCA> <AGTCCATG, CTGAACGT>
<AGTCCAGT, CTGAACTG> <AGTCCACA, CTGAACAC> <AGTCCTAG, CTGAAGCT> <AGTCCTTT, CTGAAGGG>
<AGTCCTGA, CTGAAGTC> <AGTCCGAT, CTGAATCG> <AGTCCGTA, CTGAATGC> <AGTCCCAA, CTGAAACC>
<AGGAAGCC, CTTCCTAA> <AGGAACGC, CTTCCATA> <AGGAACCG, CTTCCAAT> <AGGATTCC, CTTCGGAA>
<AGGATGGC, CTTCGTTA> <AGGATGCG, CTTCGTAT> <AGGATCTC, CTTCGAGA> <AGGATCGG, CTTCGATT>
<AGGATCCT, CTTCGAAG> <AGGAGACC, CTTCTCAA> <AGGAGTGC, CTTCTGTA> <AGGAGTCG, CTTCTGAT>
<AGGAGGTC, CTTCTTGA> <AGGAGGGG, CTTCTTTT> <AGGAGGCT, CTTCTTAG> <AGGAGCAC, CTTCTACA>
<AGGAGCTG, CTTCTAGT> <AGGAGCGT, CTTCTATG> <AGGAGCCA, CTTCTAAC> <AGGACAGC, CTTCACTA>
<AGGACACG, CTTCACAT> <AGGACTTC, CTTCAGGA> <AGGACTGG, CTTCAGTT> <AGGACTCT, CTTCAGAG>
<AGGACGAC, CTTCATCA> <AGGACGTG, CTTCATGT> <AGGACGGT, CTTCATTG> <AGGACGCA, CTTCATAC>
<AGGACCAG, CTTCAACT> <AGGACCTT, CTTCAAGG> <AGGACCGA, CTTCAATC> <AGGTATCC, CTTGCGAA>
<AGGTAGGC, CTTGCTTA> <AGGTAGCG, CTTGCTAT> <AGGTACTC, CTTGCAGA> <AGGTACGG, CTTGCATT>
<AGGTACCT, CTTGCAAG> <AGGTTACC, CTTGGCAA> <AGGTTTGC, CTTGGGTA> <AGGTTTCG, CTTGGGAT>
<AGGTTGTC, CTTGGTGA> <AGGTTGGG, CTTGGTTT> <AGGTTGCT, CTTGGTAG> <AGGTTCAC, CTTGGACA>
<AGGTTCTG, CTTGGAGT> <AGGTTCGT, CTTGGATG> <AGGTTCCA, CTTGGAAC> <AGGTGAGC, CTTGTCTA>
<AGGTGACG, CTTGTCAT> <AGGTGTTC, CTTGTGGA> <AGGTGTGG, CTTGTGTT> <AGGTGTCT, CTTGTGAG>
<AGGTGGAC, CTTGTTCA> <AGGTGGTG, CTTGTTGT> <AGGTGGGT, CTTGTTTG> <AGGTGGCA, CTTGTTAC>
<AGGTGCAG, CTTGTACT> <AGGTGCTT, CTTGTAGG> <AGGTGCGA, CTTGTATC> <AGGTCATC, CTTGACGA>
<AGGTCAGG, CTTGACTT> <AGGTCACT, CTTGACAG> <AGGTCTAC, CTTGAGCA> <AGGTCTTG, CTTGAGGT>
<AGGTCTGT, CTTGAGTG> <AGGTCTCA, CTTGAGAC> <AGGTCGAG, CTTGATCT> <AGGTCGTT, CTTGATGG>
<AGGTCGGA, CTTGATTC> <AGGTCCAT, CTTGAACG> <AGGTCCTA, CTTGAAGC> <AGGGAACC, CTTTCCAA>
<AGGGATGC, CTTTCGTA> <AGGGATCG, CTTTCGAT> <AGGGAGTC, CTTTCTGA> <AGGGAGGG, CTTTCTTT>
<AGGGAGCT, CTTTCTAG> <AGGGACAC, CTTTCACA> <AGGGACTG, CTTTCAGT> <AGGGACGT, CTTTCATG>
<AGGGACCA, CTTTCAAC> <AGGGTAGC, CTTTGCTA> <AGGGTACG, CTTTGCAT> <AGGGTTTC, CTTTGGGA>
<AGGGTTGG, CTTTGGTT> <AGGGTTCT, CTTTGGAG> <AGGGTGAC, CTTTGTCA> <AGGGTGTG, CTTTGTGT>
<AGGGTGGT, CTTTGTTG> <AGGGTGCA, CTTTGTAC> <AGGGTCAG, CTTTGACT> <AGGGTCTT, CTTTGAGG>
<AGGGTCGA, CTTTGATC> <AGGGGATC, CTTTTCGA> <AGGGGAGG, CTTTTCTT> <AGGGGACT, CTTTTCAG>
<AGGGGTAC, CTTTTGCA> <AGGGGTTG, CTTTTGGT> <AGGGGTGT, CTTTTGTG> <AGGGGTCA, CTTTTGAC>
<AGGGGGAG, CTTTTTCT> <AGGGGGTT, CTTTTTGG> <AGGGGGGA, CTTTTTTC> <AGGGGCAT, CTTTTACG>
<AGGGGCTA, CTTTTAGC> <AGGGCAAC, CTTTACCA> <AGGGCATG, CTTTACGT> <AGGGCAGT, CTTTACTG>
<AGGGCACA, CTTTACAC> <AGGGCTAG, CTTTAGCT> <AGGGCTTT, CTTTAGGG> <AGGGCTGA, CTTTAGTC>
<AGGGCGAT, CTTTATCG> <AGGGCGTA, CTTTATGC> <AGGGCCAA, CTTTAACC> <AGGCAAGC, CTTACCTA>
<AGGCAACG, CTTACCAT> <AGGCATTC, CTTACGGA> <AGGCATGG, CTTACGTT> <AGGCATCT, CTTACGAG>
<AGGCAGAC, CTTACTCA> <AGGCAGTG, CTTACTGT> <AGGCAGGT, CTTACTTG> <AGGCAGCA, CTTACTAC>
<AGGCACAG, CTTACACT> <AGGCACTT, CTTACAGG> <AGGCACGA, CTTACATC> <AGGCTATC, CTTAGCGA>
<AGGCTAGG, CTTAGCTT> <AGGCTACT, CTTAGCAG> <AGGCTTAC, CTTAGGCA> <AGGCTTTG, CTTAGGGT>
<AGGCTTGT, CTTAGGTG> <AGGCTTCA, CTTAGGAC> <AGGCTGAG, CTTAGTCT> <AGGCTGTT, CTTAGTGG>
<AGGCTGGA, CTTAGTTC> <AGGCTCAT, CTTAGACG> <AGGCTCTA, CTTAGAGC> <AGGCGAAC, CTTATCCA>
<AGGCGATG, CTTATCGT> <AGGCGAGT, CTTATCTG> <AGGCGACA, CTTATCAC> <AGGCGTAG, CTTATGCT>
<AGGCGTTT, CTTATGGG> <AGGCGTGA, CTTATGTC> <AGGCGGAT, CTTATTCG> <AGGCGGTA, CTTATTGC>
<AGGCGCAA, CTTATACC> <AGGCCAAG, CTTAACCT> <AGGCCATT, CTTAACGG> <AGGCCAGA, CTTAACTC>
<AGGCCTAT, CTTAAGCG> <AGGCCTTA, CTTAAGGC> <AGGCCGAA, CTTAATCC> <AGCAATCC, CTACCGAA>
<AGCAAGGC, CTACCTTA> <AGCAAGCG, CTACCTAT> <AGCAACTC, CTACCAGA> <AGCAACGG, CTACCATT>
<AGCAACCT, CTACCAAG> <AGCATACC, CTACGCAA> <AGCATTGC, CTACGGTA> <AGCATTCG, CTACGGAT>
<AGCATGTC, CTACGTGA> <AGCATGGG, CTACGTTT> <AGCATGCT, CTACGTAG> <AGCATCAC, CTACGACA>

<AGCATCTG, CTACGAGT>
<AGCAGACG, CTACTCAT>
<AGCAGGAC, CTACTTCA>
<AGCAGCAG, CTACTACT>
<AGCACAGG, CTACACTT>
<AGCACTGT, CTACAGTG>
<AGCACGGA, CTACATTC>
<AGCTATGC, CTAGCGTA>
<AGCTAGCT, CTAGCTAG>
<AGCTACCA, CTAGCAAC>
<AGCTTTGG, CTAGGGTT>
<AGCTTGGT, CTAGGTTG>
<AGCTTCGA, CTAGGATC>
<AGCTGTAC, CTAGTGCA>
<AGCTGGAG, CTAGTTCT>
<AGCTGCTA, CTAGTAGC>
<AGCTCACA, CTAGACAC>
<AGCTCGAT, CTAGATCG>
<AGCGAACG, CTATCCAT>
<AGCGAGAC, CTATCTCA>
<AGCGACAG, CTATCACT>
<AGCGTAGG, CTATGCTT>
<AGCGTTGT, CTATGGTG>
<AGCGTGGA, CTATGTTC>
<AGCGGATG, CTATTCGT>
<AGCGGTTT, CTATTGGG>
<AGCGGCAA, CTATTACC>
<AGCGCTAT, CTATAGCG>
<AGCCAAGG, CTAACCTT>
<AGCCATGT, CTAACGTG>
<AGCCAGGA, CTAACTTC>
<AGCCTATG, CTAAGCGT>
<AGCCTTTT, CTAAGGGG>
<AGCCTCAA, CTAAGACC>
<AGCCGTAT, CTAATGCG>
<AGCCCATA, CTAAACGC>
<ACAATCGC, CACCGATA>
<ACAAGGCG, CACCTTAT>
<ACAACACC, CACCACAA>
<ACAACGGG, CACCATTT>
<ACAACCGT, CACCAATG>
<ACATACCG, CACGCAAT>
<ACATTCTC, CACGGAGA>
<ACATGTGC, CACGTGTA>
<ACATGGCT, CACGTTAG>
<ACATGCCA, CACGTAAC>
<ACATCTGG, CACGAGTT>
<ACATCGGT, CACGATTG>
<ACATCCGA, CACGAATC>
<ACAGACTC, CACTCAGA>
<ACAGTTGC, CACTGGTA>
<ACAGTGCT, CACTGTAG>
<ACAGTCCA, CACTGAAC>
<ACAGGTGG, CACTTGTT>
<ACAGGGGT, CACTTTTG>
<ACAGGCGA, CACTTATC>
<ACAGCTAC, CACTAGCA>
<ACAGCGAG, CACTATCT>
<ACAGCCTA, CACTAAGC>
<ACACAGTC, CACACTGA>
<ACACACTG, CACACAGT>
<ACACTACG, CACAGCAT>
<ACACTGAC, CACAGTCA>
<ACACTCAG, CACAGACT>
<ACACGAGG, CACATCTT>
<ACACGTGT, CACATGTG>
<ACACGGGA, CACATTTC>
<ACACCATG, CACAACGT>
<ACACCTTT, CACAAGGG>
<ACACCCAA, CACAAACC>
<ACTATTCC, CAGCGGAA>
<ACTATCGG, CAGCGATT>
<ACTAGTCG, CAGCTGAT>
<ACTAGCAC, CAGCTACA>
<ACTACAGC, CAGCACTA>
<ACTACTCT, CAGCAGAG>
<ACTACGCA, CAGCATAC>
<ACTTATCC, CAGGCGAA>
<ACTTACGG, CAGGCATT>
<ACTTTTCG, CAGGGGAT>

<AGCATCGT, CTACGATG>
<AGCAGTTC, CTACTGGA>
<AGCAGGTG, CTACTTGT>
<AGCAGCTT, CTACTAGG>
<AGCACACT, CTACACAG>
<AGCACTCA, CTACAGAC>
<AGCACCAT, CTACAACG>
<AGCTATCG, CTAGCGAT>
<AGCTACAC, CTAGCACA>
<AGCTTAGC, CTAGGCTA>
<AGCTTTCT, CTAGGGAG>
<AGCTTGCA, CTAGGTAC>
<AGCTGATC, CTAGTCGA>
<AGCTGTTG, CTAGTGGT>
<AGCTGGTT, CTAGTTGG>
<AGCTCAAC, CTAGACCA>
<AGCTCTAG, CTAGAGCT>
<AGCTCGTA, CTAGATGC>
<AGCGATTC, CTATCGGA>
<AGCGAGTG, CTATCTGT>
<AGCGACTT, CTATCAGG>
<AGCGTACT, CTATGCAG>
<AGCGTTCA, CTATGGAC>
<AGCGTCAT, CTATGACG>
<AGCGGAGT, CTATTCTG>
<AGCGGTGA, CTATTGTC>
<AGCGCAAG, CTATACCT>
<AGCGCTTA, CTATAGGC>
<AGCCAACT, CTAACCAG>
<AGCCATCA, CTAACGAC>
<AGCCACAT, CTAACACG>
<AGCCTAGT, CTAAGCTG>
<AGCCTTGA, CTAAGGTC>
<AGCCGAAG, CTAATCCT>
<AGCCGTTA, CTAATGGC>
<AGCCCTAA, CTAAAGCC>
<ACAATCCG, CACCGAAT>
<ACAAGCTC, CACCTAGA>
<ACAACTGC, CACCAGTA>
<ACAACGCT, CACCATAG>
<ACAACCCA, CACCAAAC>
<ACATTTCC, CACGGGAA>
<ACATTCGG, CACGGATT>
<ACATGTCG, CACGTGAT>
<ACATGCAC, CACGTACA>
<ACATCAGC, CACGACTA>
<ACATCTCT, CACGAGAG>
<ACATCGCA, CACGATAC>
<ACAGATCC, CACTCGAA>
<ACAGACGG, CACTCATT>
<ACAGTTCG, CACTGGAT>
<ACAGTCAC, CACTGACA>
<ACAGGAGC, CACTTCTA>
<ACAGGTCT, CACTTGAG>
<ACAGGGCA, CACTTTAC>
<ACAGCATC, CACTACGA>
<ACAGCTTG, CACTAGGT>
<ACAGCGTT, CACTATGG>
<ACACAACC, CACACCAA>
<ACACAGGG, CACACTTT>
<ACACACGT, CACACATG>
<ACACTTTC, CACAGGGA>
<ACACTGTG, CACAGTGT>
<ACACTCTT, CACAGAGG>
<ACACGACT, CACATCAG>
<ACACGTCA, CACATGAC>
<ACACGCAT, CACATACG>
<ACACCAGT, CACAACTG>
<ACACCTGA, CACAAGTC>
<ACTAAGCC, CAGCCTAA>
<ACTATGGC, CAGCGTTA>
<ACTATCCT, CAGCGAAG>
<ACTAGGTC, CAGCTTGA>
<ACTAGCTG, CAGCTAGT>
<ACTACACG, CAGCACAT>
<ACTACGAC, CAGCATCA>
<ACTACCAG, CAGCAACT>
<ACTTAGGC, CAGGCTTA>
<ACTTACCT, CAGGCAAG>
<ACTTTGTC, CAGGGTGA>

<AGCATCCA, CTACGAAC>
<AGCAGTGG, CTACTGTT>
<AGCAGGGT, CTACTTTG>
<AGCAGCGA, CTACTATC>
<AGCACTAC, CTACAGCA>
<AGCACGAG, CTACATCT>
<AGCACCTA, CTACAAGC>
<AGCTAGTC, CTAGCTGA>
<AGCTACTG, CTAGCAGT>
<AGCTTACG, CTAGGCAT>
<AGCTTGAC, CTAGGTCA>
<AGCTTCAG, CTAGGACT>
<AGCTGAGG, CTAGTCTT>
<AGCTGTGT, CTAGTGTG>
<AGCTGGGA, CTAGTTTC>
<AGCTCATG, CTAGACGT>
<AGCTCTTT, CTAGAGGG>
<AGCTCCAA, CTAGAACC>
<AGCGATGG, CTATCGTT>
<AGCGAGGT, CTATCTTG>
<AGCGACGA, CTATCATC>
<AGCGTTAC, CTATGGCA>
<AGCGTGAG, CTATGTCT>
<AGCGTCTA, CTATGAGC>
<AGCGGACA, CTATTCAC>
<AGCGGGAT, CTATTTCG>
<AGCGCATT, CTATACGG>
<AGCGCGAA, CTATATCC>
<AGCCATAC, CTAACGCA>
<AGCCAGAG, CTAACTCT>
<AGCCACTA, CTAACAGC>
<AGCCTACA, CTAAGCAC>
<AGCCTGAT, CTAAGTCG>
<AGCCGATT, CTAATCGG>
<AGCCGGAA, CTAATTCC>
<ACAAACCC, CACCCAAA>
<ACAAGTCC, CACCTGAA>
<ACAAGCGG, CACCTATT>
<ACAACTCG, CACCAGAT>
<ACAACCAC, CACCAACA>
<ACATAGCC, CACGCTAA>
<ACATTGGC, CACGGTTA>
<ACATTCCT, CACGGAAG>
<ACATGGTC, CACGTTGA>
<ACATGCTG, CACGTAGT>
<ACATCACG, CACGACAT>
<ACATCGAC, CACGATCA>
<ACATCCAG, CACGAACT>
<ACAGAGGC, CACTCTTA>
<ACAGACCT, CACTCAAG>
<ACAGTGTC, CACTGTGA>
<ACAGTCTG, CACTGAGT>
<ACAGGACG, CACTTCAT>
<ACAGGGAC, CACTTTCA>
<ACAGGCAG, CACTTACT>
<ACAGCAGG, CACTACTT>
<ACAGCTGT, CACTAGTG>
<ACAGCGGA, CACTATTC>
<ACACATGC, CACACGTA>
<ACACAGCT, CACACTAG>
<ACACACCA, CACACAAC>
<ACACTTGG, CACAGGTT>
<ACACTGGT, CACAGTTG>
<ACACTCGA, CACAGATC>
<ACACGTAC, CACATGCA>
<ACACGGAG, CACATTCT>
<ACACGCTA, CACATAGC>
<ACACCACA, CACAACAC>
<ACACCGAT, CACAATCG>
<ACTAACGC, CAGCCATA>
<ACTATGCG, CAGCGTAT>
<ACTAGACC, CAGCTCAA>
<ACTAGGGG, CAGCTTTT>
<ACTAGCGT, CAGCTATG>
<ACTACTTC, CAGCAGGA>
<ACTACGTG, CAGCATGT>
<ACTACCTT, CAGCAAGG>
<ACTTAGCG, CAGGCTAT>
<ACTTTACC, CAGGGCAA>
<ACTTTGGG, CAGGGTTT>

<AGCAGAGC, CTACTCTA>
<AGCAGTCT, CTACTGAG>
<AGCAGGCA, CTACTTAC>
<AGCACATC, CTACACGA>
<AGCACTTG, CTACAGGT>
<AGCACGTT, CTACATGG>
<AGCTAACC, CTAGCCAA>
<AGCTAGGG, CTAGCTTT>
<AGCTACGT, CTAGCATG>
<AGCTTTTC, CTAGGGGA>
<AGCTTGTG, CTAGGTGT>
<AGCTTCTT, CTAGGAGG>
<AGCTGACT, CTAGTCAG>
<AGCTGTCA, CTAGTGAC>
<AGCTGCAT, CTAGTACG>
<AGCTCAGT, CTAGACTG>
<AGCTCTGA, CTAGAGTC>
<AGCGAAGC, CTATCCTA>
<AGCGATCT, CTATCGAG>
<AGCGAGCA, CTATCTAC>
<AGCGTATC, CTATGCGA>
<AGCGTTTG, CTATGGGT>
<AGCGTGTT, CTATGTGG>
<AGCGGAAC, CTATTCCA>
<AGCGGTAG, CTATTGCT>
<AGCGGGTA, CTATTTGC>
<AGCGCAGA, CTATACTC>
<AGCCAATC, CTAACCGA>
<AGCCATTG, CTAACGGT>
<AGCCAGTT, CTAACTGG>
<AGCCTAAC, CTAAGCCA>
<AGCCTTAG, CTAAGGCT>
<AGCCTGTA, CTAAGTGC>
<AGCCGAGA, CTAATCTC>
<AGCCCAAT, CTAAACCG>
<ACAATGCC, CACCGTAA>
<ACAAGGGC, CACCTTTA>
<ACAAGCCT, CACCTAAG>
<ACAACGTC, CACCATGA>
<ACAACCTG, CACCAAGT>
<ACATACGC, CACGCATA>
<ACATTGCG, CACGGTAT>
<ACATGACC, CACGTCAA>
<ACATGGGG, CACGTTTT>
<ACATGCGT, CACGTATG>
<ACATCTTC, CACGAGGA>
<ACATCGTG, CACGATGT>
<ACATCCTT, CACGAAGG>
<ACAGAGCG, CACTCTAT>
<ACAGTACC, CACTGCAA>
<ACAGTGGG, CACTGTTT>
<ACAGTCGT, CACTGATG>
<ACAGGTTC, CACTTGGA>
<ACAGGGTG, CACTTTGT>
<ACAGGCTT, CACTTAGG>
<ACAGCACT, CACTACAG>
<ACAGCTCA, CACTAGAC>
<ACAGCCAT, CACTAACG>
<ACACATCG, CACACGAT>
<ACACACAC, CACACACA>
<ACACTAGC, CACAGCTA>
<ACACTTCT, CACAGGAG>
<ACACTGCA, CACAGTAC>
<ACACGATC, CACATCGA>
<ACACGTTG, CACATGGT>
<ACACGGTT, CACATTGG>
<ACACCAAC, CACAACCA>
<ACACCTAG, CACAAGCT>
<ACACCGTA, CACAATGC>
<ACTAACCG, CAGCCAAT>
<ACTATCTC, CAGCGAGA>
<ACTAGTGC, CAGCTGTA>
<ACTAGGCT, CAGCTTAG>
<ACTAGCCA, CAGCTAAC>
<ACTACTGG, CAGCAGTT>
<ACTACGGT, CAGCATTG>
<ACTACCGA, CAGCAATC>
<ACTTACTC, CAGGCAGA>
<ACTTTTGC, CAGGGGTA>
<ACTTTGCT, CAGGGTAG>

<ACTTTCAC, CAGGGACA>
<ACTTGAGC, CAGGTCTA>
<ACTTGTCT, CAGGTGAG>
<ACTTGGCA, CAGGTTAC>
<ACTTCATC, CAGGACGA>
<ACTTCTTG, CAGGAGGT>
<ACTTCGTT, CAGGATGG>
<ACTGAACC, CAGTCCAA>
<ACTGAGGG, CAGTCTTT>
<ACTGACGT, CAGTCATG>
<ACTGTTTC, CAGTGGGA>
<ACTGTGTG, CAGTGTGT>
<ACTGTCTT, CAGTGAGG>
<ACTGGACT, CAGTTCAG>
<ACTGGTCA, CAGTTGAC>
<ACTGGCAT, CAGTTACG>
<ACTGCAGT, CAGTACTG>
<ACTGCTGA, CAGTAGTC>
<ACTCAAGC, CAGACCTA>
<ACTCATCT, CAGACGAG>
<ACTCAGCA, CAGACTAC>
<ACTCTATC, CAGAGCGA>
<ACTCTTTG, CAGAGGGT>
<ACTCTGTT, CAGAGTGG>
<ACTCGAAC, CAGATCCA>
<ACTCGTAG, CAGATGCT>
<ACTCGGTA, CAGATTGC>
<ACTCCAGA, CAGAACTC>
<ACGAATCC, CATCCGAA>
<ACGAACGG, CATCCATT>
<ACGATTCG, CATCGGAT>
<ACGATCAC, CATCGACA>
<ACGAGAGC, CATCTCTA>
<ACGAGTCT, CATCTGAG>
<ACGAGGCA, CATCTTAC>
<ACGACATC, CATCACGA>
<ACGACTTG, CATCAGGT>
<ACGACGTT, CATCATGG>
<ACGTAACC, CATGCCAA>
<ACGTAGGG, CATGCTTT>
<ACGTACGT, CATGCATG>
<ACGTTTTC, CATGGGGA>
<ACGTTGTG, CATGGTGT>
<ACGTTCTT, CATGGAGG>
<ACGTGACT, CATGTCAG>
<ACGTGTCA, CATGTGAC>
<ACGTGCAT, CATGTACG>
<ACGTCAGT, CATGACTG>
<ACGTCTGA, CATGAGTC>
<ACGGAAGC, CATTCCTA>
<ACGGATCT, CATTCGAG>
<ACGGAGCA, CATTCTAC>
<ACGGTATC, CATTGCGA>
<ACGGTTTG, CATTGGGT>
<ACGGTGTT, CATTGTGG>
<ACGGGAAC, CATTTCCA>
<ACGGGTAG, CATTTGCT>
<ACGGGGTA, CATTTTGC>
<ACGGCAGA, CATTACTC>
<ACGCAATC, CATACCGA>
<ACGCATTG, CATACGGT>
<ACGCAGTT, CATACTGG>
<ACGCTAAC, CATAGCCA>
<ACGCTTAG, CATAGGCT>
<ACGCTGTA, CATAGTGC>
<ACGCGAGA, CATATCTC>
<ACGCCAAT, CATAACCG>
<ACCAATGC, CAACCGTA>
<ACCAAGCT, CAACCTAG>
<ACCAACCA, CAACCAAC>
<ACCATTGG, CAACGGTT>
<ACCATGGT, CAACGTTG>
<ACCATCGA, CAACGATC>
<ACCAGTAC, CAACTGCA>
<ACCAGGAG, CAACTTCT>
<ACCAGCTA, CAACTAGC>
<ACCACACA, CAACACAC>
<ACCACGAT, CAACATCG>
<ACCTAACG, CAAGCCAT>
<ACCTAGAC, CAAGCTCA>
<ACTTTCTG, CAGGGAGT>
<ACTTGACG, CAGGTCAT>
<ACTTGGAC, CAGGTTCA>
<ACTTGCAG, CAGGTACT>
<ACTTCAGG, CAGGACTT>
<ACTTCTGT, CAGGAGTG>
<ACTTCGGA, CAGGATTC>
<ACTGATGC, CAGTCGTA>
<ACTGAGCT, CAGTCTAG>
<ACTGACCA, CAGTCAAC>
<ACTGTTGG, CAGTGGTT>
<ACTGTGGT, CAGTGTTG>
<ACTGTCGA, CAGTGATC>
<ACTGGTAC, CAGTTGCA>
<ACTGGGAG, CAGTTTCT>
<ACTGGCTA, CAGTTAGC>
<ACTGCACA, CAGTACAC>
<ACTGCGAT, CAGTATCG>
<ACTCAACG, CAGACCAT>
<ACTCAGAC, CAGACTCA>
<ACTCACAG, CAGACACT>
<ACTCTAGG, CAGAGCTT>
<ACTCTTGT, CAGAGGTG>
<ACTCTGGA, CAGAGTTC>
<ACTCGATG, CAGATCGT>
<ACTCGTTT, CAGATGGG>
<ACTCGCAA, CAGATACC>
<ACTCCTAT, CAGAAGCG>
<ACGAAGGC, CATCCTTA>
<ACGAACCT, CATCCAAG>
<ACGATGTC, CATCGTGA>
<ACGATCTG, CATCGAGT>
<ACGAGACG, CATCTCAT>
<ACGAGGAC, CATCTTCA>
<ACGAGCAG, CATCTACT>
<ACGACAGG, CATCACTT>
<ACGACTGT, CATCAGTG>
<ACGACGGA, CATCATTC>
<ACGTATGC, CATGCGTA>
<ACGTAGCT, CATGCTAG>
<ACGTACCA, CATGCAAC>
<ACGTTTGG, CATGGGTT>
<ACGTTGGT, CATGGTTG>
<ACGTTCGA, CATGGATC>
<ACGTGTAC, CATGTGCA>
<ACGTGGAG, CATGTTCT>
<ACGTGCTA, CATGTAGC>
<ACGTCACA, CATGACAC>
<ACGTCGAT, CATGATCG>
<ACGGAACG, CATTCCAT>
<ACGGAGAC, CATTCTCA>
<ACGGACAG, CATTCACT>
<ACGGTAGG, CATTGCTT>
<ACGGTTGT, CATTGGTG>
<ACGGTGGA, CATTGTTC>
<ACGGGATG, CATTTCGT>
<ACGGGTTT, CATTTGGG>
<ACGGGCAA, CATTTACC>
<ACGGCTAT, CATTAGCG>
<ACGCAAGG, CATACCTT>
<ACGCATGT, CATACGTG>
<ACGCAGGA, CATACTTC>
<ACGCTATG, CATAGCGT>
<ACGCTTTT, CATAGGGG>
<ACGCTCAA, CATAGACC>
<ACGCGTAT, CATATGCG>
<ACGCCATA, CATAACGC>
<ACCAATCG, CAACCGAT>
<ACCAACAC, CAACCACA>
<ACCATAGC, CAACGCTA>
<ACCATTCT, CAACGGAG>
<ACCATGCA, CAACGTAC>
<ACCAGATC, CAACTCGA>
<ACCAGTTG, CAACTGGT>
<ACCAGGTT, CAACTTGG>
<ACCACAAC, CAACACCA>
<ACCACTAG, CAACAGCT>
<ACCACGTA, CAACATGC>
<ACCTATTC, CAAGCGGA>
<ACCTAGTG, CAAGCTGT>
<ACTTTCGT, CAGGGATG>
<ACTTGTTC, CAGGTGGA>
<ACTTGGTG, CAGGTTGT>
<ACTTGCTT, CAGGTAGG>
<ACTTCACT, CAGGACAG>
<ACTTCTCA, CAGGAGAC>
<ACTTCCAT, CAGGAACG>
<ACTGATCG, CAGTCGAT>
<ACTGACAC, CAGTCACA>
<ACTGTAGC, CAGTGCTA>
<ACTGTTCT, CAGTGGAG>
<ACTGTGCA, CAGTGTAC>
<ACTGGATC, CAGTTCGA>
<ACTGGTTG, CAGTTGGT>
<ACTGGGTT, CAGTTTGG>
<ACTGCAAC, CAGTACCA>
<ACTGCTAG, CAGTAGCT>
<ACTGCGTA, CAGTATGC>
<ACTCATTC, CAGACGGA>
<ACTCAGTG, CAGACTGT>
<ACTCACTT, CAGACAGG>
<ACTCTACT, CAGAGCAG>
<ACTCTTCA, CAGAGGAC>
<ACTCTCAT, CAGAGACG>
<ACTCGAGT, CAGATCTG>
<ACTCGTGA, CAGATGTC>
<ACTCCAAG, CAGAACCT>
<ACTCCTTA, CAGAAGGC>
<ACGAAGCG, CATCCTAT>
<ACGATACC, CATCGCAA>
<ACGATGGG, CATCGTTT>
<ACGATCGT, CATCGATG>
<ACGAGTTC, CATCTGGA>
<ACGAGGTG, CATCTTGT>
<ACGAGCTT, CATCTAGG>
<ACGACACT, CATCACAG>
<ACGACTCA, CATCAGAC>
<ACGACCAT, CATCAACG>
<ACGTATCG, CATGCGAT>
<ACGTACAC, CATGCACA>
<ACGTTAGC, CATGGCTA>
<ACGTTTCT, CATGGGAG>
<ACGTTGCA, CATGGTAC>
<ACGTGATC, CATGTCGA>
<ACGTGTTG, CATGTGGT>
<ACGTGGTT, CATGTTGG>
<ACGTCAAC, CATGACCA>
<ACGTCTAG, CATGAGCT>
<ACGTCGTA, CATGATGC>
<ACGGATTC, CATTCGGA>
<ACGGAGTG, CATTCTGT>
<ACGGACTT, CATTCAGG>
<ACGGTACT, CATTGCAG>
<ACGGTTCA, CATTGGAC>
<ACGGTCAT, CATTGACG>
<ACGGGAGT, CATTTCTG>
<ACGGGTGA, CATTTGTC>
<ACGGCAAG, CATTACCT>
<ACGGCTTA, CATTAGGC>
<ACGCAACT, CATACCAG>
<ACGCATCA, CATACGAC>
<ACGCACAT, CATACACG>
<ACGCTAGT, CATAGCTG>
<ACGCTTGA, CATAGGTC>
<ACGCGAAG, CATATCCT>
<ACGCGTTA, CATATGGC>
<ACGCCTAA, CATAAGCC>
<ACCAAGTC, CAACCTGA>
<ACCAACTG, CAACCAGT>
<ACCATACG, CAACGCAT>
<ACCATGAC, CAACGTCA>
<ACCATCAG, CAACGACT>
<ACCAGAGG, CAACTCTT>
<ACCAGTGT, CAACTGTG>
<ACCAGGGA, CAACTTTC>
<ACCACATG, CAACACGT>
<ACCACTTT, CAACAGGG>
<ACCACCAA, CAACAACC>
<ACCTATGG, CAAGCGTT>
<ACCTAGGT, CAAGCTTG>
<ACTTTCCA, CAGGGAAC>
<ACTTGTGG, CAGGTGTT>
<ACTTGGGT, CAGGTTTG>
<ACTTGCGA, CAGGTATC>
<ACTTCTAC, CAGGAGCA>
<ACTTCGAG, CAGGATCT>
<ACTTCCTA, CAGGAAGC>
<ACTGAGTC, CAGTCTGA>
<ACTGACTG, CAGTCAGT>
<ACTGTACG, CAGTGCAT>
<ACTGTGAC, CAGTGTCA>
<ACTGTCAG, CAGTGACT>
<ACTGGAGG, CAGTTCTT>
<ACTGGTGT, CAGTTGTG>
<ACTGGGGA, CAGTTTTC>
<ACTGCATG, CAGTACGT>
<ACTGCTTT, CAGTAGGG>
<ACTGCCAA, CAGTAACC>
<ACTCATGG, CAGACGTT>
<ACTCAGGT, CAGACTTG>
<ACTCACGA, CAGACATC>
<ACTCTTAC, CAGAGGCA>
<ACTCTGAG, CAGAGTCT>
<ACTCTCTA, CAGAGAGC>
<ACTCGACA, CAGATCAC>
<ACTCGGAT, CAGATTCG>
<ACTCCATT, CAGAACGG>
<ACTCCGAA, CAGAATCC>
<ACGAACTC, CATCCAGA>
<ACGATTGC, CATCGGTA>
<ACGATGCT, CATCGTAG>
<ACGATCCA, CATCGAAC>
<ACGAGTGG, CATCTGTT>
<ACGAGGGT, CATCTTTG>
<ACGAGCGA, CATCTATC>
<ACGACTAC, CATCAGCA>
<ACGACGAG, CATCATCT>
<ACGACCTA, CATCAAGC>
<ACGTAGTC, CATGCTGA>
<ACGTACTG, CATGCAGT>
<ACGTTACG, CATGGCAT>
<ACGTTGAC, CATGGTCA>
<ACGTTCAG, CATGGACT>
<ACGTGAGG, CATGTCTT>
<ACGTGTGT, CATGTGTG>
<ACGTGGGA, CATGTTTC>
<ACGTCATG, CATGACGT>
<ACGTCTTT, CATGAGGG>
<ACGTCCAA, CATGAACC>
<ACGGATGG, CATTCGTT>
<ACGGAGGT, CATTCTTG>
<ACGGACGA, CATTCATC>
<ACGGTTAC, CATTGGCA>
<ACGGTGAG, CATTGTCT>
<ACGGTCTA, CATTGAGC>
<ACGGGACA, CATTTCAC>
<ACGGGGAT, CATTTTCG>
<ACGGCATT, CATTACGG>
<ACGGCGAA, CATTATCC>
<ACGCATAC, CATACGCA>
<ACGCAGAG, CATACTCT>
<ACGCACTA, CATACAGC>
<ACGCTACA, CATAGCAC>
<ACGCTGAT, CATAGTCG>
<ACGCGATT, CATATCGG>
<ACGCGGAA, CATATTCC>
<ACCAAACC, CAACCCAA>
<ACCAAGGG, CAACCTTT>
<ACCAACGT, CAACCATG>
<ACCATTTC, CAACGGGA>
<ACCATGTG, CAACGTGT>
<ACCATCTT, CAACGAGG>
<ACCAGACT, CAACTCAG>
<ACCAGTCA, CAACTGAC>
<ACCAGCAT, CAACTACG>
<ACCACAGT, CAACACTG>
<ACCACTGA, CAACAGTC>
<ACCTAAGC, CAAGCCTA>
<ACCTATCT, CAAGCGAG>
<ACCTAGCA, CAAGCTAC>

<ACCTACAG, CAAGCACT>
<ACCTTAGG, CAAGGCTT>
<ACCTTTGT, CAAGGGTG>
<ACCTTGGA, CAAGGTTC>
<ACCTGATG, CAAGTCGT>
<ACCTGTTT, CAAGTGGG>
<ACCTGCAA, CAAGTACC>
<ACCTCTAT, CAAGAGCG>
<ACCGAAGG, CAATCCTT>
<ACCGATGT, CAATCGTG>
<ACCGAGGA, CAATCTTC>
<ACCGTATG, CAATGCGT>
<ACCGTTTT, CAATGGGG>
<ACCGTCAA, CAATGACC>
<ACCGGTAT, CAATTGCG>
<ACCGCATA, CAATACGC>
<ACCCAAGT, CAAACCTG>
<ACCCATGA, CAAACGTC>
<ACCCTAAG, CAAAGCCT>
<ACCCTTTA, CAAAGGGC>
<ACCCGTAA, CAAATGCC>
<TAAAATAA, GCCCCGCC>
<TAAACCGC, GCCCAATA>
<TAATGGCC, GCCGTTAA>
<TAATCGGC, GCCGATTA>
<TAATCCCT, GCCGAAAG>
<TAAGTCCG, GCCTGAAT>
<TAAGGCTC, GCCTTAGA>
<TAAGCTGC, GCCTAGTA>
<TAAGCGCT, GCCTATAG>
<TAAGCCCA, GCCTAAAC>
<TAACTTCC, GCCAGGAA>
<TAACTCGG, GCCAGATT>
<TAACGTCG, GCCATGAT>
<TAACGCAC, GCCATACA>
<TAACCAGC, GCCAACTA>
<TAACCTCT, GCCAAGAG>
<TAACCGCA, GCCAATAC>
<TATAAAAA, GCGCCCCC>
<TATAGCCG, GCGCTAAT>
<TATACCTC, GCGCAAGA>
<TATTTGCC, GCGGGTAA>
<TATTGGGC, GCGGTTTA>
<TATTGCCT, GCGGTAAG>
<TATTCGTC, GCGGATGA>
<TATTCCTG, GCGGAAGT>
<TATGACGC, GCGTCATA>
<TATGTGCG, GCGTGTAT>
<TATGGACC, GCGTTCAA>
<TATGGGGG, GCGTTTTT>
<TATGGCGT, GCGTTATG>
<TATGCTTC, GCGTAGGA>
<TATGCGTG, GCGTATGT>
<TATGCCTT, GCGTAAGG>
<TATCAGCG, GCGACTAT>
<TATCTACC, GCGAGCAA>
<TATCTGGG, GCGAGTTT>
<TATCTCGT, GCGAGATG>
<TATCGTTC, GCGATGGA>
<TATCGGTG, GCGATTGT>
<TATCGCTT, GCGATAGG>
<TATCCACT, GCGAACAG>
<TATCCTCA, GCGAAGAC>
<TATCCCAT, GCGAAACG>
<TAGATCGC, GCTCGATA>
<TAGAGGCG, GCTCTTAT>
<TAGACACC, GCTCACAA>
<TAGACGGG, GCTCATTT>
<TAGACCGT, GCTCAATG>
<TAGTACCG, GCTGCAAT>
<TAGTTCTC, GCTGGAGA>
<TAGTGTGC, GCTGTGTA>
<TAGTGGCT, GCTGTTAG>
<TAGTGCCA, GCTGTAAC>
<TAGTCTGG, GCTGAGTT>
<TAGTCGGT, GCTGATTG>
<TAGTCCGA, GCTGAATC>
<TAGGACTC, GCTTCAGA>
<TAGGTTGC, GCTTGGTA>
<TAGGTGCT, GCTTGTAG>

<ACCTACTT, CAAGCAGG>
<ACCTTACT, CAAGGCAG>
<ACCTTTCA, CAAGGGAC>
<ACCTTCAT, CAAGGACG>
<ACCTGAGT, CAAGTCTG>
<ACCTGTGA, CAAGTGTC>
<ACCTCAAG, CAAGACCT>
<ACCTCTTA, CAAGAGGC>
<ACCGAACT, CAATCCAG>
<ACCGATCA, CAATCGAC>
<ACCGACAT, CAATCACG>
<ACCGTAGT, CAATGCTG>
<ACCGTTGA, CAATGGTC>
<ACCGGAAG, CAATTCCT>
<ACCGGTTA, CAATTGGC>
<ACCGCTAA, CAATAGCC>
<ACCCAACA, CAAACCAC>
<ACCCAGAT, CAAACTCG>
<ACCCTATT, CAAAGCGG>
<ACCCTGAA, CAAAGTCC>
<ACCCCAAA, CAAAACCC>
<TAAATAAA, GCCCGCCC>
<TAAACCCG, GCCCAAAT>
<TAATGCGC, GCCGTATA>
<TAATCGCG, GCCGATAT>
<TAAGACCC, GCCTCAAA>
<TAAGGTCC, GCCTTGAA>
<TAAGGCGG, GCCTTATT>
<TAAGCTCG, GCCTAGAT>
<TAAGCCAC, GCCTAACA>
<TAACAGCC, GCCACTAA>
<TAACTGGC, GCCAGTTA>
<TAACTCCT, GCCAGAAG>
<TAACGGTC, GCCATTGA>
<TAACGCTG, GCCATAGT>
<TAACCACG, GCCAACAT>
<TAACCGAC, GCCAATCA>
<TAACCCAG, GCCAAACT>
<TATATCCC, GCGCGAAA>
<TATACTCC, GCGCAGAA>
<TATACCGG, GCGCAATT>
<TATTTCGC, GCGGGATA>
<TATTGGCG, GCGGTTAT>
<TATTCACC, GCGGACAA>
<TATTCGGG, GCGGATTT>
<TATTCCGT, GCGGAATG>
<TATGACCG, GCGTCAAT>
<TATGTCTC, GCGTGAGA>
<TATGGTGC, GCGTTGTA>
<TATGGGCT, GCGTTTAG>
<TATGGCCA, GCGTTAAC>
<TATGCTGG, GCGTAGTT>
<TATGCGGT, GCGTATTG>
<TATGCCGA, GCGTAATC>
<TATCACTC, GCGACAGA>
<TATCTTGC, GCGAGGTA>
<TATCTGCT, GCGAGTAG>
<TATCTCCA, GCGAGAAC>
<TATCGTGG, GCGATGTT>
<TATCGGGT, GCGATTTG>
<TATCGCGA, GCGATATC>
<TATCCTAC, GCGAAGCA>
<TATCCGAG, GCGAATCT>
<TATCCCTA, GCGAAAGC>
<TAGATCCG, GCTCGAAT>
<TAGAGCTC, GCTCTAGA>
<TAGACTGC, GCTCAGTA>
<TAGACGCT, GCTCATAG>
<TAGACCCA, GCTCAAAC>
<TAGTTTCC, GCTGGGAA>
<TAGTTCGG, GCTGGATT>
<TAGTGTCG, GCTGTGAT>
<TAGTGCAC, GCTGTACA>
<TAGTCAGC, GCTGACTA>
<TAGTCTCT, GCTGAGAG>
<TAGTCGCA, GCTGATAC>
<TAGGATCC, GCTTCGAA>
<TAGGACGG, GCTTCATT>
<TAGGTTCG, GCTTGGAT>
<TAGGTCAC, GCTTGACA>

<ACCTACGA, CAAGCATC>
<ACCTTTAC, CAAGGGCA>
<ACCTTGAG, CAAGGTCT>
<ACCTTCTA, CAAGGAGC>
<ACCTGACA, CAAGTCAC>
<ACCTGGAT, CAAGTTCG>
<ACCTCATT, CAAGACGG>
<ACCTCGAA, CAAGATCC>
<ACCGATAC, CAATCGCA>
<ACCGAGAG, CAATCTCT>
<ACCGACTA, CAATCAGC>
<ACCGTACA, CAATGCAC>
<ACCGTGAT, CAATGTCG>
<ACCGGATT, CAATTCGG>
<ACCGGGAA, CAATTTCC>
<ACCCAAAC, CAAACCCA>
<ACCCATAG, CAAACGCT>
<ACCCAGTA, CAAACTGC>
<ACCCTAGA, CAAAGCTC>
<ACCCGAAT, CAAATCCG>
<TAAAAAAT, GCCCCCCG>
<TAAAGCCC, GCCCTAAA>
<TAATAAAA, GCCGCCCC>
<TAATGCCG, GCCGTAAT>
<TAATCCTC, GCCGAAGA>
<TAAGTGCC, GCCTGTAA>
<TAAGGGGC, GCCTTTTA>
<TAAGGCCT, GCCTTAAG>
<TAAGCGTC, GCCTATGA>
<TAAGCCTG, GCCTAAGT>
<TAACACGC, GCCACATA>
<TAACTGCG, GCCAGTAT>
<TAACGACC, GCCATCAA>
<TAACGGGG, GCCATTTT>
<TAACGCGT, GCCATATG>
<TAACCTTC, GCCAAGGA>
<TAACCGTG, GCCAATGT>
<TAACCCTT, GCCAAAGG>
<TATAGGCC, GCGCTTAA>
<TATACGGC, GCGCATTA>
<TATACCCT, GCGCAAAG>
<TATTTCCG, GCGGGAAT>
<TATTGCTC, GCGGTAGA>
<TATTCTGC, GCGGAGTA>
<TATTCGCT, GCGGATAG>
<TATTCCCA, GCGGAAAC>
<TATGTTCC, GCGTGGAA>
<TATGTCGG, GCGTGATT>
<TATGGTCG, GCGTTGAT>
<TATGGCAC, GCGTTACA>
<TATGCAGC, GCGTACTA>
<TATGCTCT, GCGTAGAG>
<TATGCGCA, GCGTATAC>
<TATCATCC, GCGACGAA>
<TATCACGG, GCGACATT>
<TATCTTCG, GCGAGGAT>
<TATCTCAC, GCGAGACA>
<TATCGAGC, GCGATCTA>
<TATCGTCT, GCGATGAG>
<TATCGGCA, GCGATTAC>
<TATCCATC, GCGAACGA>
<TATCCTTG, GCGAAGGT>
<TATCCGTT, GCGAATGG>
<TAGAACCC, GCTCCAAA>
<TAGAGTCC, GCTCTGAA>
<TAGAGCGG, GCTCTATT>
<TAGACTCG, GCTCAGAT>
<TAGACCAC, GCTCAACA>
<TAGTAGCC, GCTGCTAA>
<TAGTTGGC, GCTGGTTA>
<TAGTTCCT, GCTGGAAG>
<TAGTGGTC, GCTGTTGA>
<TAGTGCTG, GCTGTAGT>
<TAGTCACG, GCTGACAT>
<TAGTCGAC, GCTGATCA>
<TAGTCCAG, GCTGAACT>
<TAGGAGGC, GCTTCTTA>
<TAGGACCT, GCTTCAAG>
<TAGGTGTC, GCTTGTGA>
<TAGGTCTG, GCTTGAGT>

<ACCTTATC, CAAGGCGA>
<ACCTTTTG, CAAGGGGT>
<ACCTTGTT, CAAGGTGG>
<ACCTGAAC, CAAGTCCA>
<ACCTGTAG, CAAGTGCT>
<ACCTGGTA, CAAGTTGC>
<ACCTCAGA, CAAGACTC>
<ACCGAATC, CAATCCGA>
<ACCGATTG, CAATCGGT>
<ACCGAGTT, CAATCTGG>
<ACCGTAAC, CAATGCCA>
<ACCGTTAG, CAATGGCT>
<ACCGTGTA, CAATGTGC>
<ACCGGAGA, CAATTCTC>
<ACCGCAAT, CAATACCG>
<ACCCAATG, CAAACCGT>
<ACCCATTT, CAAACGGG>
<ACCCACAA, CAAACACC>
<ACCCTTAT, CAAAGGCG>
<ACCCGATA, CAAATCGC>
<TAAAAATA, GCCCCCGC>
<TAAACGCC, GCCCATAA>
<TAATTCCC, GCCGGAAA>
<TAATCTCC, GCCGAGAA>
<TAATCCGG, GCCGAATT>
<TAAGTCGC, GCCTGATA>
<TAAGGGCG, GCCTTTAT>
<TAAGCACC, GCCTACAA>
<TAAGCGGG, GCCTATTT>
<TAAGCCGT, GCCTAATG>
<TAACACCG, GCCACAAT>
<TAACTCTC, GCCAGAGA>
<TAACGTGC, GCCATGTA>
<TAACGGCT, GCCATTAG>
<TAACGCCA, GCCATAAC>
<TAACCTGG, GCCAAGTT>
<TAACCGGT, GCCAATTG>
<TAACCCGA, GCCAAATC>
<TATAGCGC, GCGCTATA>
<TATACGCG, GCGCATAT>
<TATTACCC, GCGGCAAA>
<TATTGTCC, GCGGTGAA>
<TATTGCGG, GCGGTATT>
<TATTCTCG, GCGGAGAT>
<TATTCCAC, GCGGAACA>
<TATGAGCC, GCGTCTAA>
<TATGTGGC, GCGTGTTA>
<TATGTCCT, GCGTGAAG>
<TATGGGTC, GCGTTTGA>
<TATGGCTG, GCGTTAGT>
<TATGCACG, GCGTACAT>
<TATGCGAC, GCGTATCA>
<TATGCCAG, GCGTAACT>
<TATCAGGC, GCGACTTA>
<TATCACCT, GCGACAAG>
<TATCTGTC, GCGAGTGA>
<TATCTCTG, GCGAGAGT>
<TATCGACG, GCGATCAT>
<TATCGGAC, GCGATTCA>
<TATCGCAG, GCGATACT>
<TATCCAGG, GCGAACTT>
<TATCCTGT, GCGAAGTG>
<TATCCGGA, GCGAATTC>
<TAGATGCC, GCTCGTAA>
<TAGAGGGC, GCTCTTTA>
<TAGAGCCT, GCTCTAAG>
<TAGACGTC, GCTCATGA>
<TAGACCTG, GCTCAAGT>
<TAGTACGC, GCTGCATA>
<TAGTTGCG, GCTGGTAT>
<TAGTGACC, GCTGTCAA>
<TAGTGGGG, GCTGTTTT>
<TAGTGCGT, GCTGTATG>
<TAGTCTTC, GCTGAGGA>
<TAGTCGTG, GCTGATGT>
<TAGTCCTT, GCTGAAGG>
<TAGGAGCG, GCTTCTAT>
<TAGGTACC, GCTTGCAA>
<TAGGTGGG, GCTTGTTT>
<TAGGTCGT, GCTTGATG>

<TAGGTCCA, GCTTGAAC>
<TAGGGTGG, GCTTTGTT>
<TAGGGGGT, GCTTTTTG>
<TAGGGCGA, GCTTTATC>
<TAGGCTAC, GCTTAGCA>
<TAGGCGAG, GCTTATCT>
<TAGGCCTA, GCTTAAGC>
<TAGCAGTC, GCTACTGA>
<TAGCACTG, GCTACAGT>
<TAGCTACG, GCTAGCAT>
<TAGCTGAC, GCTAGTCA>
<TAGCTCAG, GCTAGACT>
<TAGCGAGG, GCTATCTT>
<TAGCGTGT, GCTATGTG>
<TAGCGGGA, GCTATTTC>
<TAGCCATG, GCTAACGT>
<TAGCCTTT, GCTAAGGG>
<TAGCCCAA, GCTAAACC>
<TACATTCC, GCACGGAA>
<TACATCGG, GCACGATT>
<TACAGTCG, GCACTGAT>
<TACAGCAC, GCACTACA>
<TACACAGC, GCACACTA>
<TACACTCT, GCACAGAG>
<TACACGCA, GCACATAC>
<TACTATCC, GCAGCGAA>
<TACTACGG, GCAGCATT>
<TACTTTCG, GCAGGGAT>
<TACTTCAC, GCAGGACA>
<TACTGAGC, GCAGTCTA>
<TACTGTCT, GCAGTGAG>
<TACTGGCA, GCAGTTAC>
<TACTCATC, GCAGACGA>
<TACTCTTG, GCAGAGGT>
<TACTCGTT, GCAGATGG>
<TACGAACC, GCATCCAA>
<TACGAGGG, GCATCTTT>
<TACGACGT, GCATCATG>
<TACGTTTC, GCATGGGA>
<TACGTGTG, GCATGTGT>
<TACGTCTT, GCATGAGG>
<TACGGACT, GCATTCAG>
<TACGGTCA, GCATTGAC>
<TACGGCAT, GCATTACG>
<TACGCAGT, GCATACTG>
<TACGCTGA, GCATAGTC>
<TACCAAGC, GCAACCTA>
<TACCATCT, GCAACGAG>
<TACCAGCA, GCAACTAC>
<TACCTATC, GCAAGCGA>
<TACCTTTG, GCAAGGGT>
<TACCTGTT, GCAAGTGG>
<TACCGAAC, GCAATCCA>
<TACCGTAG, GCAATGCT>
<TACCGGTA, GCAATTGC>
<TACCCAGA, GCAAACTC>
<TTAAAAAA, GGCCCCCC>
<TTAAGCCG, GGCCTAAT>
<TTAACCTC, GGCCAAGA>
<TTATTGCC, GGCGGTAA>
<TTATGGGC, GGCGTTTA>
<TTATGCCT, GGCGTAAG>
<TTATCGTC, GGCGATGA>
<TTATCCTG, GGCGAAGT>
<TTAGACGC, GGCTCATA>
<TTAGTGCG, GGCTGTAT>
<TTAGGACC, GGCTTCAA>
<TTAGGGGG, GGCTTTTT>
<TTAGGCGT, GGCTTATG>
<TTAGCTTC, GGCTAGGA>
<TTAGCGTG, GGCTATGT>
<TTAGCCTT, GGCTAAGG>
<TTACAGCG, GGCACTAT>
<TTACTACC, GGCAGCAA>
<TTACTGGG, GGCAGTTT>
<TTACTCGT, GGCAGATG>
<TTACGTTC, GGCATGGA>
<TTACGGTG, GGCATTGT>
<TTACGCTT, GGCATAGG>
<TTACCACT, GGCAACAG>

<TAGGGAGC, GCTTTCTA>
<TAGGGTCT, GCTTTGAG>
<TAGGGGCA, GCTTTTAC>
<TAGGCATC, GCTTACGA>
<TAGGCTTG, GCTTAGGT>
<TAGGCGTT, GCTTATGG>
<TAGCAACC, GCTACCAA>
<TAGCAGGG, GCTACTTT>
<TAGCACGT, GCTACATG>
<TAGCTTTC, GCTAGGGA>
<TAGCTGTG, GCTAGTGT>
<TAGCTCTT, GCTAGAGG>
<TAGCGACT, GCTATCAG>
<TAGCGTCA, GCTATGAC>
<TAGCGCAT, GCTATACG>
<TAGCCAGT, GCTAACTG>
<TAGCCTGA, GCTAAGTC>
<TACAAGCC, GCACCTAA>
<TACATGGC, GCACGTTA>
<TACATCCT, GCACGAAG>
<TACAGGTC, GCACTTGA>
<TACAGCTG, GCACTAGT>
<TACACACG, GCACACAT>
<TACACGAC, GCACATCA>
<TACACCAG, GCACAACT>
<TACTAGGC, GCAGCTTA>
<TACTACCT, GCAGCAAG>
<TACTTGTC, GCAGGTGA>
<TACTTCTG, GCAGGAGT>
<TACTGACG, GCAGTCAT>
<TACTGGAC, GCAGTTCA>
<TACTGCAG, GCAGTACT>
<TACTCAGG, GCAGACTT>
<TACTCTGT, GCAGAGTG>
<TACTCGGA, GCAGATTC>
<TACGATGC, GCATCGTA>
<TACGAGCT, GCATCTAG>
<TACGACCA, GCATCAAC>
<TACGTTGG, GCATGGTT>
<TACGTGGT, GCATGTTG>
<TACGTCGA, GCATGATC>
<TACGGTAC, GCATTGCA>
<TACGGGAG, GCATTTCT>
<TACGGCTA, GCATTAGC>
<TACGCACA, GCATACAC>
<TACGCGAT, GCATATCG>
<TACCAACG, GCAACCAT>
<TACCAGAC, GCAACTCA>
<TACCACAG, GCAACACT>
<TACCTAGG, GCAAGCTT>
<TACCTTGT, GCAAGGTG>
<TACCTGGA, GCAAGTTC>
<TACCGATG, GCAATCGT>
<TACCGTTT, GCAATGGG>
<TACCGCAA, GCAATACC>
<TACCCTAT, GCAAAGCG>
<TTAATCCC, GGCCGAAA>
<TTAACTCC, GGCCAGAA>
<TTAACCGG, GGCCAATT>
<TTATTCGC, GGCGGATA>
<TTATGGCG, GGCGTTAT>
<TTATCACC, GGCGACAA>
<TTATCGGG, GGCGATTT>
<TTATCCGT, GGCGAATG>
<TTAGACCG, GGCTCAAT>
<TTAGTCTC, GGCTGAGA>
<TTAGGTGC, GGCTTGTA>
<TTAGGGCT, GGCTTTAG>
<TTAGGCCA, GGCTTAAC>
<TTAGCTGG, GGCTAGTT>
<TTAGCGGT, GGCTATTG>
<TTAGCCGA, GGCTAATC>
<TTACACTC, GGCACAGA>
<TTACTTGC, GGCAGGTA>
<TTACTGCT, GGCAGTAG>
<TTACTCCA, GGCAGAAC>
<TTACGTGG, GGCATGTT>
<TTACGGGT, GGCATTTG>
<TTACGCGA, GGCATATC>
<TTACCTAC, GGCAAGCA>

<TAGGGACG, GCTTTCAT>
<TAGGGGAC, GCTTTTCA>
<TAGGGCAG, GCTTTACT>
<TAGGCAGG, GCTTACTT>
<TAGGCTGT, GCTTAGTG>
<TAGGCGGA, GCTTATTC>
<TAGCATGC, GCTACGTA>
<TAGCAGCT, GCTACTAG>
<TAGCACCA, GCTACAAC>
<TAGCTTGG, GCTAGGTT>
<TAGCTGGT, GCTAGTTG>
<TAGCTCGA, GCTAGATC>
<TAGCGTAC, GCTATGCA>
<TAGCGGAG, GCTATTCT>
<TAGCGCTA, GCTATAGC>
<TAGCCACA, GCTAACAC>
<TAGCCGAT, GCTAATCG>
<TACAACGC, GCACCATA>
<TACATGCG, GCACGTAT>
<TACAGACC, GCACTCAA>
<TACAGGGG, GCACTTTT>
<TACAGCGT, GCACTATG>
<TACACTTC, GCACAGGA>
<TACACGTG, GCACATGT>
<TACACCTT, GCACAAGG>
<TACTAGCG, GCAGCTAT>
<TACTTACC, GCAGGCAA>
<TACTTGGG, GCAGGTTT>
<TACTTCGT, GCAGGATG>
<TACTGTTC, GCAGTGGA>
<TACTGGTG, GCAGTTGT>
<TACTGCTT, GCAGTAGG>
<TACTCACT, GCAGACAG>
<TACTCTCA, GCAGAGAC>
<TACTCCAT, GCAGAACG>
<TACGATCG, GCATCGAT>
<TACGACAC, GCATCACA>
<TACGTAGC, GCATGCTA>
<TACGTTCT, GCATGGAG>
<TACGTGCA, GCATGTAC>
<TACGGATC, GCATTCGA>
<TACGGTTG, GCATTGGT>
<TACGGGTT, GCATTTGG>
<TACGCAAC, GCATACCA>
<TACGCTAG, GCATAGCT>
<TACGCGTA, GCATATGC>
<TACCATTC, GCAACGGA>
<TACCAGTG, GCAACTGT>
<TACCACTT, GCAACAGG>
<TACCTACT, GCAAGCAG>
<TACCTTCA, GCAAGGAC>
<TACCTCAT, GCAAGACG>
<TACCGAGT, GCAATCTG>
<TACCGTGA, GCAATGTC>
<TACCCAAG, GCAAACCT>
<TACCCTTA, GCAAAGGC>
<TTAAGGCC, GGCCTTAA>
<TTAACGGC, GGCCATTA>
<TTAACCCT, GGCCAAAG>
<TTATTCCG, GGCGGAAT>
<TTATGCTC, GGCGTAGA>
<TTATCTGC, GGCGAGTA>
<TTATCGCT, GGCGATAG>
<TTATCCCA, GGCGAAAC>
<TTAGTTCC, GGCTGGAA>
<TTAGTCGG, GGCTGATT>
<TTAGGTCG, GGCTTGAT>
<TTAGGCAC, GGCTTACA>
<TTAGCAGC, GGCTACTA>
<TTAGCTCT, GGCTAGAG>
<TTAGCGCA, GGCTATAC>
<TTACATCC, GGCACGAA>
<TTACACGG, GGCACATT>
<TTACTTCG, GGCAGGAT>
<TTACTCAC, GGCAGACA>
<TTACGAGC, GGCATCTA>
<TTACGTCT, GGCATGAG>
<TTACGGCA, GGCATTAC>
<TTACCATC, GGCAACGA>
<TTACCTTG, GGCAAGGT>

<TAGGGTTC, GCTTTGGA>
<TAGGGGTG, GCTTTTGT>
<TAGGGCTT, GCTTTAGG>
<TAGGCACT, GCTTACAG>
<TAGGCTCA, GCTTAGAC>
<TAGGCCAT, GCTTAACG>
<TAGCATCG, GCTACGAT>
<TAGCACAC, GCTACACA>
<TAGCTAGC, GCTAGCTA>
<TAGCTTCT, GCTAGGAG>
<TAGCTGCA, GCTAGTAC>
<TAGCGATC, GCTATCGA>
<TAGCGTTG, GCTATGGT>
<TAGCGGTT, GCTATTGG>
<TAGCCAAC, GCTAACCA>
<TAGCCTAG, GCTAAGCT>
<TAGCCGTA, GCTAATGC>
<TACAACCG, GCACCAAT>
<TACATCTC, GCACGAGA>
<TACAGTGC, GCACTGTA>
<TACAGGCT, GCACTTAG>
<TACAGCCA, GCACTAAC>
<TACACTGG, GCACAGTT>
<TACACGGT, GCACATTG>
<TACACCGA, GCACAATC>
<TACTACTC, GCAGCAGA>
<TACTTTGC, GCAGGGTA>
<TACTTGCT, GCAGGTAG>
<TACTTCCA, GCAGGAAC>
<TACTGTGG, GCAGTGTT>
<TACTGGGT, GCAGTTTG>
<TACTGCGA, GCAGTATC>
<TACTCTAC, GCAGAGCA>
<TACTCGAG, GCAGATCT>
<TACTCCTA, GCAGAAGC>
<TACGAGTC, GCATCTGA>
<TACGACTG, GCATCAGT>
<TACGTACG, GCATGCAT>
<TACGTGAC, GCATGTCA>
<TACGTCAG, GCATGACT>
<TACGGAGG, GCATTCTT>
<TACGGTGT, GCATTGTG>
<TACGGGGA, GCATTTTC>
<TACGCATG, GCATACGT>
<TACGCTTT, GCATAGGG>
<TACGCCAA, GCATAACC>
<TACCATGG, GCAACGTT>
<TACCAGGT, GCAACTTG>
<TACCACGA, GCAACATC>
<TACCTTAC, GCAAGGCA>
<TACCTGAG, GCAAGTCT>
<TACCTCTA, GCAAGAGC>
<TACCGACA, GCAATCAC>
<TACCGGAT, GCAATTCG>
<TACCCATT, GCAAACGG>
<TACCCGAA, GCAAATCC>
<TTAAGCGC, GGCCTATA>
<TTAACGCG, GGCCATAT>
<TTATACCC, GGCGCAAA>
<TTATGTCC, GGCGTGAA>
<TTATGCGG, GGCGTATT>
<TTATCTCG, GGCGAGAT>
<TTATCCAC, GGCGAACA>
<TTAGAGCC, GGCTCTAA>
<TTAGTGGC, GGCTGTTA>
<TTAGTCCT, GGCTGAAG>
<TTAGGGTC, GGCTTTGA>
<TTAGGCTG, GGCTTAGT>
<TTAGCACG, GGCTACAT>
<TTAGCGAC, GGCTATCA>
<TTAGCCAG, GGCTAACT>
<TTACAGGC, GGCACTTA>
<TTACACCT, GGCACAAG>
<TTACTGTC, GGCAGTGA>
<TTACTCTG, GGCAGAGT>
<TTACGACG, GGCATCAT>
<TTACGGAC, GGCATTCA>
<TTACGCAG, GGCATACT>
<TTACCAGG, GGCAACTT>
<TTACCTGT, GGCAAGTG>

<TTACCTCA, GGCAAGAC>
<TTACCCAT, GGCAAACG>
<TTTATCGC, GGGCGATA>
<TTTAGGCG, GGGCTTAT>
<TTTACACC, GGGCACAA>
<TTTACGGG, GGGCATTT>
<TTTACCGT, GGGCAATG>
<TTTTACCG, GGGGCAAT>
<TTTTTCTC, GGGGGAGA>
<TTTTGTGC, GGGGTGTA>
<TTTTGGCT, GGGGTTAG>
<TTTTGCCA, GGGGTAAC>
<TTTTCTGG, GGGGAGTT>
<TTTTCGGT, GGGGATTG>
<TTTTCCGA, GGGGAATC>
<TTTGACTC, GGGTCAGA>
<TTTGTTGC, GGGTGGTA>
<TTTGTGCT, GGGTGTAG>
<TTTGTCCA, GGGTGAAC>
<TTTGGTGG, GGGTTGTT>
<TTTGGGGT, GGGTTTTG>
<TTTGGCGA, GGGTTATC>
<TTTGCTAC, GGGTAGCA>
<TTTGCGAG, GGGTATCT>
<TTTGCCTA, GGGTAAGC>
<TTTCAGTC, GGGACTGA>
<TTTCACTG, GGGACAGT>
<TTTCTACG, GGGAGCAT>
<TTTCTGAC, GGGAGTCA>
<TTTCTCAG, GGGAGACT>
<TTTCGAGG, GGGATCTT>
<TTTCGTGT, GGGATGTG>
<TTTCGGGA, GGGATTTC>
<TTTCCATG, GGGAACGT>
<TTTCCTTT, GGGAAGGG>
<TTTCCCAA, GGGAAACC>
<TTGATTCC, GGTCGGAA>
<TTGATCGG, GGTCGATT>
<TTGAGTCG, GGTCTGAT>
<TTGAGCAC, GGTCTACA>
<TTGACAGC, GGTCACTA>
<TTGACTCT, GGTCAGAG>
<TTGACGCA, GGTCATAC>
<TTGTATCC, GGTGCGAA>
<TTGTACGG, GGTGCATT>
<TTGTTTCG, GGTGGGAT>
<TTGTTCAC, GGTGGACA>
<TTGTGAGC, GGTGTCTA>
<TTGTGTCT, GGTGTGAG>
<TTGTGGCA, GGTGTTAC>
<TTGTCATC, GGTGACGA>
<TTGTCTTG, GGTGAGGT>
<TTGTCGTT, GGTGATGG>
<TTGGAACC, GGTTCCAA>
<TTGGAGGG, GGTTCTTT>
<TTGGACGT, GGTTCATG>
<TTGGTTTC, GGTTGGGA>
<TTGGTGTG, GGTTGTGT>
<TTGGTCTT, GGTTGAGG>
<TTGGGACT, GGTTTCAG>
<TTGGGTCA, GGTTTGAC>
<TTGGGCAT, GGTTTACG>
<TTGGCAGT, GGTTACTG>
<TTGGCTGA, GGTTAGTC>
<TTGCAAGC, GGTACCTA>
<TTGCATCT, GGTACGAG>
<TTGCAGCA, GGTACTAC>
<TTGCTATC, GGTAGCGA>
<TTGCTTTG, GGTAGGGT>
<TTGCTGTT, GGTAGTGG>
<TTGCGAAC, GGTATCCA>
<TTGCGTAG, GGTATGCT>
<TTGCGGTA, GGTATTGC>
<TTGCCAGA, GGTAACTC>
<TTCAATCC, GGACCGAA>
<TTCAACGG, GGACCATT>
<TTCATTCG, GGACGGAT>
<TTCATCAC, GGACGACA>
<TTCAGAGC, GGACTCTA>
<TTCAGTCT, GGACTGAG>

<TTACCGAG, GGCAATCT>
<TTACCCTA, GGCAAAGC>
<TTTATCCG, GGGCGAAT>
<TTTAGCTC, GGGCTAGA>
<TTTACTGC, GGGCAGTA>
<TTTACGCT, GGGCATAG>
<TTTACCCA, GGGCAAAC>
<TTTTTTCC, GGGGGGAA>
<TTTTTCGG, GGGGGATT>
<TTTTGTCG, GGGGTGAT>
<TTTTGCAC, GGGGTACA>
<TTTTCAGC, GGGGACTA>
<TTTTCTCT, GGGGAGAG>
<TTTTCGCA, GGGGATAC>
<TTTGATCC, GGGTCGAA>
<TTTGACGG, GGGTCATT>
<TTTGTTCG, GGGTGGAT>
<TTTGTCAC, GGGTGACA>
<TTTGGAGC, GGGTTCTA>
<TTTGGTCT, GGGTTGAG>
<TTTGGGCA, GGGTTTAC>
<TTTGCATC, GGGTACGA>
<TTTGCTTG, GGGTAGGT>
<TTTGCGTT, GGGTATGG>
<TTTCAACC, GGGACCAA>
<TTTCAGGG, GGGACTTT>
<TTTCACGT, GGGACATG>
<TTTCTTTC, GGGAGGGA>
<TTTCTGTG, GGGAGTGT>
<TTTCTCTT, GGGAGAGG>
<TTTCGACT, GGGATCAG>
<TTTCGTCA, GGGATGAC>
<TTTCGCAT, GGGATACG>
<TTTCCAGT, GGGAACTG>
<TTTCCTGA, GGGAAGTC>
<TTGAAGCC, GGTCCTAA>
<TTGATGGC, GGTCGTTA>
<TTGATCCT, GGTCGAAG>
<TTGAGGTC, GGTCTTGA>
<TTGAGCTG, GGTCTAGT>
<TTGACACG, GGTCACAT>
<TTGACGAC, GGTCATCA>
<TTGACCAG, GGTCAACT>
<TTGTAGGC, GGTGCTTA>
<TTGTACCT, GGTGCAAG>
<TTGTTGTC, GGTGGTGA>
<TTGTTCTG, GGTGGAGT>
<TTGTGACG, GGTGTCAT>
<TTGTGGAC, GGTGTTCA>
<TTGTGCAG, GGTGTACT>
<TTGTCAGG, GGTGACTT>
<TTGTCTGT, GGTGAGTG>
<TTGTCGGA, GGTGATTC>
<TTGGATGC, GGTTCGTA>
<TTGGAGCT, GGTTCTAG>
<TTGGACCA, GGTTCAAC>
<TTGGTTGG, GGTTGGTT>
<TTGGTGGT, GGTTGTTG>
<TTGGTCGA, GGTTGATC>
<TTGGGTAC, GGTTTGCA>
<TTGGGGAG, GGTTTTCT>
<TTGGGCTA, GGTTTAGC>
<TTGGCACA, GGTTACAC>
<TTGGCGAT, GGTTATCG>
<TTGCAACG, GGTACCAT>
<TTGCAGAC, GGTACTCA>
<TTGCACAG, GGTACACT>
<TTGCTAGG, GGTAGCTT>
<TTGCTTGT, GGTAGGTG>
<TTGCTGGA, GGTAGTTC>
<TTGCGATG, GGTATCGT>
<TTGCGTTT, GGTATGGG>
<TTGCGCAA, GGTATACC>
<TTGCCTAT, GGTAAGCG>
<TTCAAGGC, GGACCTTA>
<TTCAACCT, GGACCAAG>
<TTCATGTC, GGACGTGA>
<TTCATCTG, GGACGAGT>
<TTCAGACG, GGACTCAT>
<TTCAGGAC, GGACTTCA>

<TTACCGTT, GGCAATGG>
<TTTAACCC, GGGCCAAA>
<TTTAGTCC, GGGCTGAA>
<TTTAGCGG, GGGCTATT>
<TTTACTCG, GGGCAGAT>
<TTTACCAC, GGGCAACA>
<TTTTAGCC, GGGGCTAA>
<TTTTTGGC, GGGGGTTA>
<TTTTTCCT, GGGGGAAG>
<TTTTGGTC, GGGGTTGA>
<TTTTGCTG, GGGGTAGT>
<TTTTCACG, GGGGACAT>
<TTTTCGAC, GGGGATCA>
<TTTTCCAG, GGGGAACT>
<TTTGAGGC, GGGTCTTA>
<TTTGACCT, GGGTCAAG>
<TTTGTGTC, GGGTGTGA>
<TTTGTCTG, GGGTGAGT>
<TTTGGACG, GGGTTCAT>
<TTTGGGAC, GGGTTTCA>
<TTTGGCAG, GGGTTACT>
<TTTGCAGG, GGGTACTT>
<TTTGCTGT, GGGTAGTG>
<TTTGCGGA, GGGTATTC>
<TTTCATGC, GGGACGTA>
<TTTCAGCT, GGGACTAG>
<TTTCACCA, GGGACAAC>
<TTTCTTGG, GGGAGGTT>
<TTTCTGGT, GGGAGTTG>
<TTTCTCGA, GGGAGATC>
<TTTCGTAC, GGGATGCA>
<TTTCGGAG, GGGATTCT>
<TTTCGCTA, GGGATAGC>
<TTTCCACA, GGGAACAC>
<TTTCCGAT, GGGAATCG>
<TTGAACGC, GGTCCATA>
<TTGATGCG, GGTCGTAT>
<TTGAGACC, GGTCTCAA>
<TTGAGGGG, GGTCTTTT>
<TTGAGCGT, GGTCTATG>
<TTGACTTC, GGTCAGGA>
<TTGACGTG, GGTCATGT>
<TTGACCTT, GGTCAAGG>
<TTGTAGCG, GGTGCTAT>
<TTGTTACC, GGTGGCAA>
<TTGTTGGG, GGTGGTTT>
<TTGTTCGT, GGTGGATG>
<TTGTGTTC, GGTGTGGA>
<TTGTGGTG, GGTGTTGT>
<TTGTGCTT, GGTGTAGG>
<TTGTCACT, GGTGACAG>
<TTGTCTCA, GGTGAGAC>
<TTGTCCAT, GGTGAACG>
<TTGGATCG, GGTTCGAT>
<TTGGACAC, GGTTCACA>
<TTGGTAGC, GGTTGCTA>
<TTGGTTCT, GGTTGGAG>
<TTGGTGCA, GGTTGTAC>
<TTGGGATC, GGTTTCGA>
<TTGGGTTG, GGTTTGGT>
<TTGGGGTT, GGTTTTGG>
<TTGGCAAC, GGTTACCA>
<TTGGCTAG, GGTTAGCT>
<TTGGCGTA, GGTTATGC>
<TTGCATTC, GGTACGGA>
<TTGCAGTG, GGTACTGT>
<TTGCACTT, GGTACAGG>
<TTGCTACT, GGTAGCAG>
<TTGCTTCA, GGTAGGAC>
<TTGCTCAT, GGTAGACG>
<TTGCGAGT, GGTATCTG>
<TTGCGTGA, GGTATGTC>
<TTGCCAAG, GGTAACCT>
<TTGCCTTA, GGTAAGGC>
<TTCAAGCG, GGACCTAT>
<TTCATACC, GGACGCAA>
<TTCATGGG, GGACGTTT>
<TTCATCGT, GGACGATG>
<TTCAGTTC, GGACTGGA>
<TTCAGGTG, GGACTTGT>

<TTACCGGA, GGCAATTC>
<TTTATGCC, GGGCGTAA>
<TTTAGGGC, GGGCTTTA>
<TTTAGCCT, GGGCTAAG>
<TTTACGTC, GGGCATGA>
<TTTACCTG, GGGCAAGT>
<TTTTACGC, GGGGCATA>
<TTTTTGCG, GGGGGTAT>
<TTTTGACC, GGGGTCAA>
<TTTTGGGG, GGGGTTTT>
<TTTTGCGT, GGGGTATG>
<TTTTCTTC, GGGGAGGA>
<TTTTCGTG, GGGGATGT>
<TTTTCCTT, GGGGAAGG>
<TTTGAGCG, GGGTCTAT>
<TTTGTACC, GGGTGCAA>
<TTTGTGGG, GGGTGTTT>
<TTTGTCGT, GGGTGATG>
<TTTGGTTC, GGGTTGGA>
<TTTGGGTG, GGGTTTGT>
<TTTGGCTT, GGGTTAGG>
<TTTGCACT, GGGTACAG>
<TTTGCTCA, GGGTAGAC>
<TTTGCCAT, GGGTAACG>
<TTTCATCG, GGGACGAT>
<TTTCACAC, GGGACACA>
<TTTCTAGC, GGGAGCTA>
<TTTCTTCT, GGGAGGAG>
<TTTCTGCA, GGGAGTAC>
<TTTCGATC, GGGATCGA>
<TTTCGTTG, GGGATGGT>
<TTTCGGTT, GGGATTGG>
<TTTCCAAC, GGGAACCA>
<TTTCCTAG, GGGAAGCT>
<TTTCCGTA, GGGAATGC>
<TTGAACCG, GGTCCAAT>
<TTGATCTC, GGTCGAGA>
<TTGAGTGC, GGTCTGTA>
<TTGAGGCT, GGTCTTAG>
<TTGAGCCA, GGTCTAAC>
<TTGACTGG, GGTCAGTT>
<TTGACGGT, GGTCATTG>
<TTGACCGA, GGTCAATC>
<TTGTACTC, GGTGCAGA>
<TTGTTTGC, GGTGGGTA>
<TTGTTGCT, GGTGGTAG>
<TTGTTCCA, GGTGGAAC>
<TTGTGTGG, GGTGTGTT>
<TTGTGGGT, GGTGTTTG>
<TTGTGCGA, GGTGTATC>
<TTGTCTAC, GGTGAGCA>
<TTGTCGAG, GGTGATCT>
<TTGTCCTA, GGTGAAGC>
<TTGGAGTC, GGTTCTGA>
<TTGGACTG, GGTTCAGT>
<TTGGTACG, GGTTGCAT>
<TTGGTGAC, GGTTGTCA>
<TTGGTCAG, GGTTGACT>
<TTGGGAGG, GGTTTCTT>
<TTGGGTGT, GGTTTGTG>
<TTGGGGGA, GGTTTTTC>
<TTGGCATG, GGTTACGT>
<TTGGCTTT, GGTTAGGG>
<TTGGCCAA, GGTTAACC>
<TTGCATGG, GGTACGTT>
<TTGCAGGT, GGTACTTG>
<TTGCACGA, GGTACATC>
<TTGCTTAC, GGTAGGCA>
<TTGCTGAG, GGTAGTCT>
<TTGCTCTA, GGTAGAGC>
<TTGCGACA, GGTATCAC>
<TTGCGGAT, GGTATTCG>
<TTGCCATT, GGTAACGG>
<TTGCCGAA, GGTAATCC>
<TTCAACTC, GGACCAGA>
<TTCATTGC, GGACGGTA>
<TTCATGCT, GGACGTAG>
<TTCATCCA, GGACGAAC>
<TTCAGTGG, GGACTGTT>
<TTCAGGGT, GGACTTTG>

<TTCAGGCA, GGACTTAC>
<TTCACATC, GGACACGA>
<TTCACTTG, GGACAGGT>
<TTCACGTT, GGACATGG>
<TTCTAACC, GGAGCCAA>
<TTCTAGGG, GGAGCTTT>
<TTCTACGT, GGAGCATG>
<TTCTTTTC, GGAGGGGA>
<TTCTTGTG, GGAGGTGT>
<TTCTTCTT, GGAGGAGG>
<TTCTGACT, GGAGTCAG>
<TTCTGTCA, GGAGTGAC>
<TTCTGCAT, GGAGTACG>
<TTCTCAGT, GGAGACTG>
<TTCTCTGA, GGAGAGTC>
<TTCGAAGC, GGATCCTA>
<TTCGATCT, GGATCGAG>
<TTCGAGCA, GGATCTAC>
<TTCGTATC, GGATGCGA>
<TTCGTTTG, GGATGGGT>
<TTCGTGTT, GGATGTGG>
<TTCGGAAC, GGATTCCA>
<TTCGGTAG, GGATTGCT>
<TTCGGGTA, GGATTTGC>
<TTCGCAGA, GGATACTC>
<TTCCAATC, GGAACCGA>
<TTCCATTG, GGAACGGT>
<TTCCAGTT, GGAACTGG>
<TTCCTAAC, GGAAGCCA>
<TTCCTTAG, GGAAGGCT>
<TTCCTGTA, GGAAGTGC>
<TTCCGAGA, GGAATCTC>
<TTCCCAAT, GGAAACCG>
<TGAATGCC, GTCCGTAA>
<TGAAGGGC, GTCCTTTA>
<TGAAGCCT, GTCCTAAG>
<TGAACGTC, GTCCATGA>
<TGAACCTG, GTCCAAGT>
<TGATACGC, GTCGCATA>
<TGATTGCG, GTCGGTAT>
<TGATGACC, GTCGTCAA>
<TGATGGGG, GTCGTTTT>
<TGATGCGT, GTCGTATG>
<TGATCTTC, GTCGAGGA>
<TGATCGTG, GTCGATGT>
<TGATCCTT, GTCGAAGG>
<TGAGAGCG, GTCTCTAT>
<TGAGTACC, GTCTGCAA>
<TGAGTGGG, GTCTGTTT>
<TGAGTCGT, GTCTGATG>
<TGAGGTTC, GTCTTGGA>
<TGAGGGTG, GTCTTTGT>
<TGAGGCTT, GTCTTAGG>
<TGAGCACT, GTCTACAG>
<TGAGCTCA, GTCTAGAC>
<TGAGCCAT, GTCTAACG>
<TGACATCG, GTCACGAT>
<TGACACAC, GTCACACA>
<TGACTAGC, GTCAGCTA>
<TGACTTCT, GTCAGGAG>
<TGACTGCA, GTCAGTAC>
<TGACGATC, GTCATCGA>
<TGACGTTG, GTCATGGT>
<TGACGGTT, GTCATTGG>
<TGACCAAC, GTCAACCA>
<TGACCTAG, GTCAAGCT>
<TGACCGTA, GTCAATGC>
<TGTAACCG, GTGCCAAT>
<TGTATCTC, GTGCGAGA>
<TGTAGTGC, GTGCTGTA>
<TGTAGGCT, GTGCTTAG>
<TGTAGCCA, GTGCTAAC>
<TGTACTGG, GTGCAGTT>
<TGTACGGT, GTGCATTG>
<TGTACCGA, GTGCAATC>
<TGTTACTC, GTGGCAGA>
<TGTTTTGC, GTGGGGTA>
<TGTTTGCT, GTGGGTAG>
<TGTTTCCA, GTGGGAAC>
<TGTTGTGG, GTGGTGTT>
<TTCAGCAG, GGACTACT>
<TTCACAGG, GGACACTT>
<TTCACTGT, GGACAGTG>
<TTCACGGA, GGACATTC>
<TTCTATGC, GGAGCGTA>
<TTCTAGCT, GGAGCTAG>
<TTCTACCA, GGAGCAAC>
<TTCTTTGG, GGAGGGTT>
<TTCTTGGT, GGAGGTTG>
<TTCTTCGA, GGAGGATC>
<TTCTGTAC, GGAGTGCA>
<TTCTGGAG, GGAGTTCT>
<TTCTGCTA, GGAGTAGC>
<TTCTCACA, GGAGACAC>
<TTCTCGAT, GGAGATCG>
<TTCGAACG, GGATCCAT>
<TTCGAGAC, GGATCTCA>
<TTCGACAG, GGATCACT>
<TTCGTAGG, GGATGCTT>
<TTCGTTGT, GGATGGTG>
<TTCGTGGA, GGATGTTC>
<TTCGGATG, GGATTCGT>
<TTCGGTTT, GGATTGGG>
<TTCGGCAA, GGATTACC>
<TTCGCTAT, GGATAGCG>
<TTCCAAGG, GGAACCTT>
<TTCCATGT, GGAACGTG>
<TTCCAGGA, GGAACTTC>
<TTCCTATG, GGAAGCGT>
<TTCCTTTT, GGAAGGGG>
<TTCCTCAA, GGAAGACC>
<TTCCGTAT, GGAATGCG>
<TTCCCATA, GGAAACGC>
<TGAATCGC, GTCCGATA>
<TGAAGGCG, GTCCTTAT>
<TGAACACC, GTCCACAA>
<TGAACGGG, GTCCATTT>
<TGAACCGT, GTCCAATG>
<TGATACCG, GTCGCAAT>
<TGATTCTC, GTCGGAGA>
<TGATGTGC, GTCGTGTA>
<TGATGGCT, GTCGTTAG>
<TGATGCCA, GTCGTAAC>
<TGATCTGG, GTCGAGTT>
<TGATCGGT, GTCGATTG>
<TGATCCGA, GTCGAATC>
<TGAGACTC, GTCTCAGA>
<TGAGTTGC, GTCTGGTA>
<TGAGTGCT, GTCTGTAG>
<TGAGTCCA, GTCTGAAC>
<TGAGGTGG, GTCTTGTT>
<TGAGGGGT, GTCTTTTG>
<TGAGGCGA, GTCTTATC>
<TGAGCTAC, GTCTAGCA>
<TGAGCGAG, GTCTATCT>
<TGAGCCTA, GTCTAAGC>
<TGACAGTC, GTCACTGA>
<TGACACTG, GTCACAGT>
<TGACTACG, GTCAGCAT>
<TGACTGAC, GTCAGTCA>
<TGACTCAG, GTCAGACT>
<TGACGAGG, GTCATCTT>
<TGACGTGT, GTCATGTG>
<TGACGGGA, GTCATTTC>
<TGACCATG, GTCAACGT>
<TGACCTTT, GTCAAGGG>
<TGACCCAA, GTCAAACC>
<TGTATTCC, GTGCGGAA>
<TGTATCGG, GTGCGATT>
<TGTAGTCG, GTGCTGAT>
<TGTAGCAC, GTGCTACA>
<TGTACAGC, GTGCACTA>
<TGTACTCT, GTGCAGAG>
<TGTACGCA, GTGCATAC>
<TGTTATCC, GTGGCGAA>
<TGTTACGG, GTGGCATT>
<TGTTTTCG, GTGGGGAT>
<TGTTTCAC, GTGGGACA>
<TGTTGAGC, GTGGTCTA>
<TGTTGTCT, GTGGTGAG>
<TTCAGCTT, GGACTAGG>
<TTCACACT, GGACACAG>
<TTCACTCA, GGACAGAC>
<TTCACCAT, GGACAACG>
<TTCTATCG, GGAGCGAT>
<TTCTACAC, GGAGCACA>
<TTCTTAGC, GGAGGCTA>
<TTCTTTCT, GGAGGGAG>
<TTCTTGCA, GGAGGTAC>
<TTCTGATC, GGAGTCGA>
<TTCTGTTG, GGAGTGGT>
<TTCTGGTT, GGAGTTGG>
<TTCTCAAC, GGAGACCA>
<TTCTCTAG, GGAGAGCT>
<TTCTCGTA, GGAGATGC>
<TTCGATTC, GGATCGGA>
<TTCGAGTG, GGATCTGT>
<TTCGACTT, GGATCAGG>
<TTCGTACT, GGATGCAG>
<TTCGTTCA, GGATGGAC>
<TTCGTCAT, GGATGACG>
<TTCGGAGT, GGATTCTG>
<TTCGGTGA, GGATTGTC>
<TTCGCAAG, GGATACCT>
<TTCGCTTA, GGATAGGC>
<TTCCAACT, GGAACCAG>
<TTCCATCA, GGAACGAC>
<TTCCACAT, GGAACACG>
<TTCCTAGT, GGAAGCTG>
<TTCCTTGA, GGAAGGTC>
<TTCCGAAG, GGAATCCT>
<TTCCGTTA, GGAATGGC>
<TTCCCTAA, GGAAAGCC>
<TGAATCCG, GTCCGAAT>
<TGAAGCTC, GTCCTAGA>
<TGAACTGC, GTCCAGTA>
<TGAACGCT, GTCCATAG>
<TGAACCCA, GTCCAAAC>
<TGATTTCC, GTCGGGAA>
<TGATTCGG, GTCGGATT>
<TGATGTCG, GTCGTGAT>
<TGATGCAC, GTCGTACA>
<TGATCAGC, GTCGACTA>
<TGATCTCT, GTCGAGAG>
<TGATCGCA, GTCGATAC>
<TGAGATCC, GTCTCGAA>
<TGAGACGG, GTCTCATT>
<TGAGTTCG, GTCTGGAT>
<TGAGTCAC, GTCTGACA>
<TGAGGAGC, GTCTTCTA>
<TGAGGTCT, GTCTTGAG>
<TGAGGGCA, GTCTTTAC>
<TGAGCATC, GTCTACGA>
<TGAGCTTG, GTCTAGGT>
<TGAGCGTT, GTCTATGG>
<TGACAACC, GTCACCAA>
<TGACAGGG, GTCACTTT>
<TGACACGT, GTCACATG>
<TGACTTTC, GTCAGGGA>
<TGACTGTG, GTCAGTGT>
<TGACTCTT, GTCAGAGG>
<TGACGACT, GTCATCAG>
<TGACGTCA, GTCATGAC>
<TGACGCAT, GTCATACG>
<TGACCAGT, GTCAACTG>
<TGACCTGA, GTCAAGTC>
<TGTAAGCC, GTGCCTAA>
<TGTATGGC, GTGCGTTA>
<TGTATCCT, GTGCGAAG>
<TGTAGGTC, GTGCTTGA>
<TGTAGCTG, GTGCTAGT>
<TGTACACG, GTGCACAT>
<TGTACGAC, GTGCATCA>
<TGTACCAG, GTGCAACT>
<TGTTAGGC, GTGGCTTA>
<TGTTACCT, GTGGCAAG>
<TGTTTGTC, GTGGGTGA>
<TGTTTCTG, GTGGGAGT>
<TGTTGACG, GTGGTCAT>
<TGTTGGAC, GTGGTTCA>
<TTCAGCGA, GGACTATC>
<TTCACTAC, GGACAGCA>
<TTCACGAG, GGACATCT>
<TTCACCTA, GGACAAGC>
<TTCTAGTC, GGAGCTGA>
<TTCTACTG, GGAGCAGT>
<TTCTTACG, GGAGGCAT>
<TTCTTGAC, GGAGGTCA>
<TTCTTCAG, GGAGGACT>
<TTCTGAGG, GGAGTCTT>
<TTCTGTGT, GGAGTGTG>
<TTCTGGGA, GGAGTTTC>
<TTCTCATG, GGAGACGT>
<TTCTCTTT, GGAGAGGG>
<TTCTCCAA, GGAGAACC>
<TTCGATGG, GGATCGTT>
<TTCGAGGT, GGATCTTG>
<TTCGACGA, GGATCATC>
<TTCGTTAC, GGATGGCA>
<TTCGTGAG, GGATGTCT>
<TTCGTCTA, GGATGAGC>
<TTCGGACA, GGATTCAC>
<TTCGGGAT, GGATTTCG>
<TTCGCATT, GGATACGG>
<TTCGCGAA, GGATATCC>
<TTCCATAC, GGAACGCA>
<TTCCAGAG, GGAACTCT>
<TTCCACTA, GGAACAGC>
<TTCCTACA, GGAAGCAC>
<TTCCTGAT, GGAAGTCG>
<TTCCGATT, GGAATCGG>
<TTCCGGAA, GGAATTCC>
<TGAAACCC, GTCCCAAA>
<TGAAGTCC, GTCCTGAA>
<TGAAGCGG, GTCCTATT>
<TGAACTCG, GTCCAGAT>
<TGAACCAC, GTCCAACA>
<TGATAGCC, GTCGCTAA>
<TGATTGGC, GTCGGTTA>
<TGATTCCT, GTCGGAAG>
<TGATGGTC, GTCGTTGA>
<TGATGCTG, GTCGTAGT>
<TGATCACG, GTCGACAT>
<TGATCGAC, GTCGATCA>
<TGATCCAG, GTCGAACT>
<TGAGAGGC, GTCTCTTA>
<TGAGACCT, GTCTCAAG>
<TGAGTGTC, GTCTGTGA>
<TGAGTCTG, GTCTGAGT>
<TGAGGACG, GTCTTCAT>
<TGAGGGAC, GTCTTTCA>
<TGAGGCAG, GTCTTACT>
<TGAGCAGG, GTCTACTT>
<TGAGCTGT, GTCTAGTG>
<TGAGCGGA, GTCTATTC>
<TGACATGC, GTCACGTA>
<TGACAGCT, GTCACTAG>
<TGACACCA, GTCACAAC>
<TGACTTGG, GTCAGGTT>
<TGACTGGT, GTCAGTTG>
<TGACTCGA, GTCAGATC>
<TGACGTAC, GTCATGCA>
<TGACGGAG, GTCATTCT>
<TGACGCTA, GTCATAGC>
<TGACCACA, GTCAACAC>
<TGACCGAT, GTCAATCG>
<TGTAACGC, GTGCCATA>
<TGTATGCG, GTGCGTAT>
<TGTAGACC, GTGCTCAA>
<TGTAGGGG, GTGCTTTT>
<TGTAGCGT, GTGCTATG>
<TGTACTTC, GTGCAGGA>
<TGTACGTG, GTGCATGT>
<TGTACCTT, GTGCAAGG>
<TGTTAGCG, GTGGCTAT>
<TGTTTACC, GTGGGCAA>
<TGTTTGGG, GTGGGTTT>
<TGTTTCGT, GTGGGATG>
<TGTTGTTC, GTGGTGGA>
<TGTTGGTG, GTGGTTGT>

<TGTTGGGT, GTGGTTTG>	<TGTTGGCA, GTGGTTAC>	<TGTTGCAG, GTGGTACT>	<TGTTGCTT, GTGGTAGG>
<TGTTGCGA, GTGGTATC>	<TGTTCATC, GTGGACGA>	<TGTTCAGG, GTGGACTT>	<TGTTCACT, GTGGACAG>
<TGTTCTAC, GTGGAGCA>	<TGTTCTTG, GTGGAGGT>	<TGTTCTGT, GTGGAGTG>	<TGTTCTCA, GTGGAGAC>
<TGTTCGAG, GTGGATCT>	<TGTTCGTT, GTGGATGG>	<TGTTCGGA, GTGGATTC>	<TGTTCCAT, GTGGAACG>
<TGTTCCTA, GTGGAAGC>	<TGTGAACC, GTGTCCAA>	<TGTGATGC, GTGTCGTA>	<TGTGATCG, GTGTCGAT>
<TGTGAGTC, GTGTCTGA>	<TGTGAGGG, GTGTCTTT>	<TGTGAGCT, GTGTCTAG>	<TGTGACAC, GTGTCACA>
<TGTGACTG, GTGTCAGT>	<TGTGACGT, GTGTCATG>	<TGTGACCA, GTGTCAAC>	<TGTGTAGC, GTGTGCTA>
<TGTGTACG, GTGTGCAT>	<TGTGTTTC, GTGTGGGA>	<TGTGTTGG, GTGTGGTT>	<TGTGTTCT, GTGTGGAG>
<TGTGTGAC, GTGTGTCA>	<TGTGTGTG, GTGTGTGT>	<TGTGTGGT, GTGTGTTG>	<TGTGTGCA, GTGTGTAC>
<TGTGTCAG, GTGTGACT>	<TGTGTCTT, GTGTGAGG>	<TGTGTCGA, GTGTGATC>	<TGTGGATC, GTGTTCGA>
<TGTGGAGG, GTGTTCTT>	<TGTGGACT, GTGTTCAG>	<TGTGGTAC, GTGTTGCA>	<TGTGGTTG, GTGTTGGT>
<TGTGGTGT, GTGTTGTG>	<TGTGGTCA, GTGTTGAC>	<TGTGGGAG, GTGTTTCT>	<TGTGGGTT, GTGTTTGG>
<TGTGGGGA, GTGTTTTC>	<TGTGGCAT, GTGTTACG>	<TGTGGCTA, GTGTTAGC>	<TGTGCAAC, GTGTACCA>
<TGTGCATG, GTGTACGT>	<TGTGCAGT, GTGTACTG>	<TGTGCACA, GTGTACAC>	<TGTGCTAG, GTGTAGCT>
<TGTGCTTT, GTGTAGGG>	<TGTGCTGA, GTGTAGTC>	<TGTGCGAT, GTGTATCG>	<TGTGCGTA, GTGTATGC>
<TGTGCCAA, GTGTAACC>	<TGTCAAGC, GTGACCTA>	<TGTCAACG, GTGACCAT>	<TGTCATTC, GTGACGGA>
<TGTCATGG, GTGACGTT>	<TGTCATCT, GTGACGAG>	<TGTCAGAC, GTGACTCA>	<TGTCAGTG, GTGACTGT>
<TGTCAGGT, GTGACTTG>	<TGTCAGCA, GTGACTAC>	<TGTCACAG, GTGACACT>	<TGTCACTT, GTGACAGG>
<TGTCACGA, GTGACATC>	<TGTCTATC, GTGAGCGA>	<TGTCTAGG, GTGAGCTT>	<TGTCTACT, GTGAGCAG>
<TGTCTTAC, GTGAGGCA>	<TGTCTTTG, GTGAGGGT>	<TGTCTTGT, GTGAGGTG>	<TGTCTTCA, GTGAGGAC>
<TGTCTGAG, GTGAGTCT>	<TGTCTGTT, GTGAGTGG>	<TGTCTGGA, GTGAGTTC>	<TGTCTCAT, GTGAGACG>
<TGTCTCTA, GTGAGAGC>	<TGTCGAAC, GTGATCCA>	<TGTCGATG, GTGATCGT>	<TGTCGAGT, GTGATCTG>
<TGTCGACA, GTGATCAC>	<TGTCGTAG, GTGATGCT>	<TGTCGTTT, GTGATGGG>	<TGTCGTGA, GTGATGTC>
<TGTCGGAT, GTGATTCG>	<TGTCGGTA, GTGATTGC>	<TGTCGCAA, GTGATACC>	<TGTCCAAG, GTGAACCT>
<TGTCCATT, GTGAACGG>	<TGTCCAGA, GTGAACTC>	<TGTCCTAT, GTGAAGCG>	<TGTCCTTA, GTGAAGGC>
<TGTCCGAA, GTGAATCC>	<TGGAATCC, GTTCCGAA>	<TGGAAGGC, GTTCCTTA>	<TGGAAGCG, GTTCCTAT>
<TGGAACTC, GTTCCAGA>	<TGGAACGG, GTTCCATT>	<TGGAACCT, GTTCCAAG>	<TGGATACC, GTTCGCAA>
<TGGATTGC, GTTCGGTA>	<TGGATTCG, GTTCGGAT>	<TGGATGTC, GTTCGTGA>	<TGGATGGG, GTTCGTTT>
<TGGATGCT, GTTCGTAG>	<TGGATCAC, GTTCGACA>	<TGGATCTG, GTTCGAGT>	<TGGATCGT, GTTCGATG>
<TGGATCCA, GTTCGAAC>	<TGGAGAGC, GTTCTCTA>	<TGGAGACG, GTTCTCAT>	<TGGAGTTC, GTTCTGGA>
<TGGAGTGG, GTTCTGTT>	<TGGAGTCT, GTTCTGAG>	<TGGAGGAC, GTTCTTCA>	<TGGAGGTG, GTTCTTGT>
<TGGAGGGT, GTTCTTTG>	<TGGAGGCA, GTTCTTAC>	<TGGAGCAG, GTTCTACT>	<TGGAGCTT, GTTCTAGG>
<TGGAGCGA, GTTCTATC>	<TGGACATC, GTTCACGA>	<TGGACAGG, GTTCACTT>	<TGGACACT, GTTCACAG>
<TGGACTAC, GTTCAGCA>	<TGGACTTG, GTTCAGGT>	<TGGACTGT, GTTCAGTG>	<TGGACTCA, GTTCAGAC>
<TGGACGAG, GTTCATCT>	<TGGACGTT, GTTCATGG>	<TGGACGGA, GTTCATTC>	<TGGACCAT, GTTCAACG>
<TGGACCTA, GTTCAAGC>	<TGGTAACC, GTTGCCAA>	<TGGTATGC, GTTGCGTA>	<TGGTATCG, GTTGCGAT>
<TGGTAGTC, GTTGCTGA>	<TGGTAGGG, GTTGCTTT>	<TGGTAGCT, GTTGCTAG>	<TGGTACAC, GTTGCACA>
<TGGTACTG, GTTGCAGT>	<TGGTACGT, GTTGCATG>	<TGGTACCA, GTTGCAAC>	<TGGTTAGC, GTTGGCTA>
<TGGTTACG, GTTGGCAT>	<TGGTTTTC, GTTGGGGA>	<TGGTTTGG, GTTGGGTT>	<TGGTTTCT, GTTGGGAG>
<TGGTTGAC, GTTGGTCA>	<TGGTTGTG, GTTGGTGT>	<TGGTTGGT, GTTGGTTG>	<TGGTTGCA, GTTGGTAC>
<TGGTTCAG, GTTGGACT>	<TGGTTCTT, GTTGGAGG>	<TGGTTCGA, GTTGGATC>	<TGGTGATC, GTTGTCGA>
<TGGTGAGG, GTTGTCTT>	<TGGTGACT, GTTGTCAG>	<TGGTGTAC, GTTGTGCA>	<TGGTGTTG, GTTGTGGT>
<TGGTGTGT, GTTGTGTG>	<TGGTGTCA, GTTGTGAC>	<TGGTGGAG, GTTGTTCT>	<TGGTGGTT, GTTGTTGG>
<TGGTGGGA, GTTGTTTC>	<TGGTGCAT, GTTGTACG>	<TGGTGCTA, GTTGTAGC>	<TGGTCAAC, GTTGACCA>
<TGGTCATG, GTTGACGT>	<TGGTCAGT, GTTGACTG>	<TGGTCACA, GTTGACAC>	<TGGTCTAG, GTTGAGCT>
<TGGTCTTT, GTTGAGGG>	<TGGTCTGA, GTTGAGTC>	<TGGTCGAT, GTTGATCG>	<TGGTCGTA, GTTGATGC>
<TGGTCCAA, GTTGAACC>	<TGGGAAGC, GTTTCCTA>	<TGGGAACG, GTTTCCAT>	<TGGGATTC, GTTTCGGA>
<TGGGATGG, GTTTCGTT>	<TGGGATCT, GTTTCGAG>	<TGGGAGAC, GTTTCTCA>	<TGGGAGTG, GTTTCTGT>
<TGGGAGGT, GTTTCTTG>	<TGGGAGCA, GTTTCTAC>	<TGGGACAG, GTTTCACT>	<TGGGACTT, GTTTCAGG>
<TGGGACGA, GTTTCATC>	<TGGGTATC, GTTTGCGA>	<TGGGTAGG, GTTTGCTT>	<TGGGTACT, GTTTGCAG>
<TGGGTTAC, GTTTGGCA>	<TGGGTTTG, GTTTGGGT>	<TGGGTTGT, GTTTGGTG>	<TGGGTTCA, GTTTGGAC>
<TGGGTGAG, GTTTGTCT>	<TGGGTGTT, GTTTGTGG>	<TGGGTGGA, GTTTGTTC>	<TGGGTCAT, GTTTGACG>
<TGGGTCTA, GTTTGAGC>	<TGGGGAAC, GTTTTCCA>	<TGGGGATG, GTTTTCGT>	<TGGGGAGT, GTTTTCTG>
<TGGGGACA, GTTTTCAC>	<TGGGGTAG, GTTTTGCT>	<TGGGGTTT, GTTTTGGG>	<TGGGGTGA, GTTTTGTC>
<TGGGGGAT, GTTTTTCG>	<TGGGGGTA, GTTTTTGC>	<TGGGGCAA, GTTTTACC>	<TGGGCAAG, GTTTACCT>
<TGGGCATT, GTTTACGG>	<TGGGCAGA, GTTTACTC>	<TGGGCTAT, GTTTAGCG>	<TGGGCTTA, GTTTAGGC>
<TGGGCGAA, GTTTATCC>	<TGGCAATC, GTTACCGA>	<TGGCAAGG, GTTACCTT>	<TGGCAACT, GTTACCAG>
<TGGCATAC, GTTACGCA>	<TGGCATTG, GTTACGGT>	<TGGCATGT, GTTACGTG>	<TGGCATCA, GTTACGAC>
<TGGCAGAG, GTTACTCT>	<TGGCAGTT, GTTACTGG>	<TGGCAGGA, GTTACTTC>	<TGGCACAT, GTTACACG>
<TGGCACTA, GTTACAGC>	<TGGCTAAC, GTTAGCCA>	<TGGCTATG, GTTAGCGT>	<TGGCTAGT, GTTAGCTG>
<TGGCTACA, GTTAGCAC>	<TGGCTTAG, GTTAGGCT>	<TGGCTTTT, GTTAGGGG>	<TGGCTTGA, GTTAGGTC>
<TGGCTGAT, GTTAGTCG>	<TGGCTGTA, GTTAGTGC>	<TGGCTCAA, GTTAGACC>	<TGGCGAAG, GTTATCCT>
<TGGCGATT, GTTATCGG>	<TGGCGAGA, GTTATCTC>	<TGGCGTAT, GTTATGCG>	<TGGCGTTA, GTTATGGC>
<TGGCGGAA, GTTATTCC>	<TGGCCAAT, GTTAACCG>	<TGGCCATA, GTTAACGC>	<TGGCCTAA, GTTAAGCC>
<TGCAAACC, GTACCCAA>	<TGCAATGC, GTACCGTA>	<TGCAATCG, GTACCGAT>	<TGCAAGTC, GTACCTGA>
<TGCAAGGG, GTACCTTT>	<TGCAAGCT, GTACCTAG>	<TGCAACAC, GTACCACA>	<TGCAACTG, GTACCAGT>
<TGCAACGT, GTACCATG>	<TGCAACCA, GTACCAAC>	<TGCATAGC, GTACGCTA>	<TGCATACG, GTACGCAT>
<TGCATTTC, GTACGGGA>	<TGCATTGG, GTACGGTT>	<TGCATTCT, GTACGGAG>	<TGCATGAC, GTACGTCA>
<TGCATGTG, GTACGTGT>	<TGCATGGT, GTACGTTG>	<TGCATGCA, GTACGTAC>	<TGCATCAG, GTACGACT>
<TGCATCTT, GTACGAGG>	<TGCATCGA, GTACGATC>	<TGCAGATC, GTACTCGA>	<TGCAGAGG, GTACTCTT>
<TGCAGACT, GTACTCAG>	<TGCAGTAC, GTACTGCA>	<TGCAGTTG, GTACTGGT>	<TGCAGTGT, GTACTGTG>
<TGCAGTCA, GTACTGAC>	<TGCAGGAG, GTACTTCT>	<TGCAGGTT, GTACTTGG>	<TGCAGGGA, GTACTTTC>
<TGCAGCAT, GTACTACG>	<TGCAGCTA, GTACTAGC>	<TGCACAAC, GTACACCA>	<TGCACATG, GTACACGT>
<TGCACAGT, GTACACTG>	<TGCACACA, GTACACAC>	<TGCACTAG, GTACAGCT>	<TGCACTTT, GTACAGGG>
<TGCACTGA, GTACAGTC>	<TGCACGAT, GTACATCG>	<TGCACGTA, GTACATGC>	<TGCACCAA, GTACAACC>
<TGCTAAGC, GTAGCCTA>	<TGCTAACG, GTAGCCAT>	<TGCTATTC, GTAGCGGA>	<TGCTATGG, GTAGCGTT>
<TGCTATCT, GTAGCGAG>	<TGCTAGAC, GTAGCTCA>	<TGCTAGTG, GTAGCTGT>	<TGCTAGGT, GTAGCTTG>
<TGCTAGCA, GTAGCTAC>	<TGCTACAG, GTAGCACT>	<TGCTACTT, GTAGCAGG>	<TGCTACGA, GTAGCATC>
<TGCTTATC, GTAGGCGA>	<TGCTTAGG, GTAGGCTT>	<TGCTTACT, GTAGGCAG>	<TGCTTTAC, GTAGGGCA>
<TGCTTTTG, GTAGGGGT>	<TGCTTTGT, GTAGGGTG>	<TGCTTTCA, GTAGGGAC>	<TGCTTGAG, GTAGGTCT>

<TGCTTGTT, GTAGGTGG>
<TGCTGAAC, GTAGTCCA>
<TGCTGTAG, GTAGTGCT>
<TGCTGGTA, GTAGTTGC>
<TGCTCAGA, GTAGACTC>
<TGCGAATC, GTATCCGA>
<TGCGATTG, GTATCGGT>
<TGCGAGTT, GTATCTGG>
<TGCGTAAC, GTATGCCA>
<TGCGTTAG, GTATGGCT>
<TGCGTGTA, GTATGTGC>
<TGCGGAGA, GTATTCTC>
<TGCGCAAT, GTATACCG>
<TGCCAATG, GTAACCGT>
<TGCCATTT, GTAACGGG>
<TGCCACAA, GTAACACC>
<TGCCTTAT, GTAAGGCG>
<TGCCGATA, GTAATCGC>
<TCAAACGC, GACCCATA>
<TCAATGCG, GACCGTAT>
<TCAAGACC, GACCTCAA>
<TCAAGGGG, GACCTTTT>
<TCAAGCGT, GACCTATG>
<TCAACTTC, GACCAGGA>
<TCAACGTG, GACCATGT>
<TCAACCTT, GACCAAGG>
<TCATAGCG, GACGCTAT>
<TCATTACC, GACGGCAA>
<TCATTGGG, GACGGTTT>
<TCATTCGT, GACGGATG>
<TCATGTTC, GACGTGGA>
<TCATGGTG, GACGTTGT>
<TCATGCTT, GACGTAGG>
<TCATCACT, GACGACAG>
<TCATCTCA, GACGAGAC>
<TCATCCAT, GACGAACG>
<TCAGATCG, GACTCGAT>
<TCAGACAC, GACTCACA>
<TCAGTAGC, GACTGCTA>
<TCAGTTCT, GACTGGAG>
<TCAGTGCA, GACTGTAC>
<TCAGGATC, GACTTCGA>
<TCAGGTTG, GACTTGGT>
<TCAGGGTT, GACTTTGG>
<TCAGCAAC, GACTACCA>
<TCAGCTAG, GACTAGCT>
<TCAGCGTA, GACTATGC>
<TCACATTC, GACACGGA>
<TCACAGTG, GACACTGT>
<TCACACTT, GACACAGG>
<TCACTACT, GACAGCAG>
<TCACTTCA, GACAGGAC>
<TCACTCAT, GACAGACG>
<TCACGAGT, GACATCTG>
<TCACGTGA, GACATGTC>
<TCACCAAG, GACAACCT>
<TCACCTTA, GACAAGGC>
<TCTAAGCG, GAGCCTAT>
<TCTATACC, GAGCGCAA>
<TCTATGGG, GAGCGTTT>
<TCTATCGT, GAGCGATG>
<TCTAGTTC, GAGCTGGA>
<TCTAGGTG, GAGCTTGT>
<TCTAGCTT, GAGCTAGG>
<TCTACACT, GAGCACAG>
<TCTACTCA, GAGCAGAC>
<TCTACCAT, GAGCAACG>
<TCTTATCG, GAGGCGAT>
<TCTTACAC, GAGGCACA>
<TCTTTAGC, GAGGGCTA>
<TCTTTTCT, GAGGGGAG>
<TCTTTGCA, GAGGGTAC>
<TCTTGATC, GAGGTCGA>
<TCTTGTTG, GAGGTGGT>
<TCTTGGTT, GAGGTTGG>
<TCTTCAAC, GAGGACCA>
<TCTTCTAG, GAGGAGCT>
<TCTTCGTA, GAGGATGC>
<TCTGATTC, GAGTCGGA>
<TCTGAGTG, GAGTCTGT>
<TGCTTGGA, GTAGGTTC>
<TGCTGATG, GTAGTCGT>
<TGCTGTTT, GTAGTGGG>
<TGCTGCAA, GTAGTACC>
<TGCTCTAT, GTAGAGCG>
<TGCGAAGG, GTATCCTT>
<TGCGATGT, GTATCGTG>
<TGCGAGGA, GTATCTTC>
<TGCGTATG, GTATGCGT>
<TGCGTTTT, GTATGGGG>
<TGCGTCAA, GTATGACC>
<TGCGGTAT, GTATTGCG>
<TGCGCATA, GTATACGC>
<TGCCAAGT, GTAACCTG>
<TGCCATGA, GTAACGTC>
<TGCCTAAG, GTAAGCCT>
<TGCCTTTA, GTAAGGGC>
<TGCCGTAA, GTAATGCC>
<TCAAACCG, GACCCAAT>
<TCAATCTC, GACCGAGA>
<TCAAGTGC, GACCTGTA>
<TCAAGGCT, GACCTTAG>
<TCAAGCCA, GACCTAAC>
<TCAACTGG, GACCAGTT>
<TCAACGGT, GACCATTG>
<TCAACCGA, GACCAATC>
<TCATACTC, GACGCAGA>
<TCATTTGC, GACGGGTA>
<TCATTGCT, GACGGTAG>
<TCATTCCA, GACGGAAC>
<TCATGTGG, GACGTGTT>
<TCATGGGT, GACGTTTG>
<TCATGCGA, GACGTATC>
<TCATCTAC, GACGAGCA>
<TCATCGAG, GACGATCT>
<TCATCCTA, GACGAAGC>
<TCAGAGTC, GACTCTGA>
<TCAGACTG, GACTCAGT>
<TCAGTACG, GACTGCAT>
<TCAGTGAC, GACTGTCA>
<TCAGTCAG, GACTGACT>
<TCAGGAGG, GACTTCTT>
<TCAGGTGT, GACTTGTG>
<TCAGGGGA, GACTTTTC>
<TCAGCATG, GACTACGT>
<TCAGCTTT, GACTAGGG>
<TCAGCCAA, GACTAACC>
<TCACATGG, GACACGTT>
<TCACAGGT, GACACTTG>
<TCACACGA, GACACATC>
<TCACTTAC, GACAGGCA>
<TCACTGAG, GACAGTCT>
<TCACTCTA, GACAGAGC>
<TCACGACA, GACATCAC>
<TCACGGAT, GACATTCG>
<TCACCATT, GACAACGG>
<TCACCGAA, GACAATCC>
<TCTAACTC, GAGCCAGA>
<TCTATTGC, GAGCGGTA>
<TCTATGCT, GAGCGTAG>
<TCTATCCA, GAGCGAAC>
<TCTAGTGG, GAGCTGTT>
<TCTAGGGT, GAGCTTTG>
<TCTAGCGA, GAGCTATC>
<TCTACTAC, GAGCAGCA>
<TCTACGAG, GAGCATCT>
<TCTACCTA, GAGCAAGC>
<TCTTAGTC, GAGGCTGA>
<TCTTACTG, GAGGCAGT>
<TCTTTACG, GAGGGCAT>
<TCTTTGAC, GAGGGTCA>
<TCTTTCAG, GAGGGACT>
<TCTTGAGG, GAGGTCTT>
<TCTTGTGT, GAGGTGTG>
<TCTTGGGA, GAGGTTTC>
<TCTTCATG, GAGGACGT>
<TCTTCTTT, GAGGAGGG>
<TCTTCCAA, GAGGAACC>
<TCTGATGG, GAGTCGTT>
<TCTGAGGT, GAGTCTTG>
<TGCTTCAT, GTAGGACG>
<TGCTGAGT, GTAGTCTG>
<TGCTGTGA, GTAGTGTC>
<TGCTCAAG, GTAGACCT>
<TGCTCTTA, GTAGAGGC>
<TGCGAACT, GTATCCAG>
<TGCGATCA, GTATCGAC>
<TGCGACAT, GTATCACG>
<TGCGTAGT, GTATGCTG>
<TGCGTTGA, GTATGGTC>
<TGCGGAAG, GTATTCCT>
<TGCGGTTA, GTATTGGC>
<TGCGCTAA, GTATAGCC>
<TGCCAACA, GTAACCAC>
<TGCCAGAT, GTAACTCG>
<TGCCTATT, GTAAGCGG>
<TGCCTGAA, GTAAGTCC>
<TGCCCAAA, GTAAACCC>
<TCAATTCC, GACCGGAA>
<TCAATCGG, GACCGATT>
<TCAAGTCG, GACCTGAT>
<TCAAGCAC, GACCTACA>
<TCAACAGC, GACCACTA>
<TCAACTCT, GACCAGAG>
<TCAACGCA, GACCATAC>
<TCATATCC, GACGCGAA>
<TCATACGG, GACGCATT>
<TCATTTCG, GACGGGAT>
<TCATTCAC, GACGGACA>
<TCATGAGC, GACGTCTA>
<TCATGTCT, GACGTGAG>
<TCATGGCA, GACGTTAC>
<TCATCATC, GACGACGA>
<TCATCTTG, GACGAGGT>
<TCATCGTT, GACGATGG>
<TCAGAACC, GACTCCAA>
<TCAGAGGG, GACTCTTT>
<TCAGACGT, GACTCATG>
<TCAGTTTC, GACTGGGA>
<TCAGTGTG, GACTGTGT>
<TCAGTCTT, GACTGAGG>
<TCAGGACT, GACTTCAG>
<TCAGGTCA, GACTTGAC>
<TCAGGCAT, GACTTACG>
<TCAGCAGT, GACTACTG>
<TCAGCTGA, GACTAGTC>
<TCACAAGC, GACACCTA>
<TCACATCT, GACACGAG>
<TCACAGCA, GACACTAC>
<TCACTATC, GACAGCGA>
<TCACTTTG, GACAGGGT>
<TCACTGTT, GACAGTGG>
<TCACGAAC, GACATCCA>
<TCACGTAG, GACATGCT>
<TCACGGTA, GACATTGC>
<TCACCAGA, GACAACTC>
<TCTAATCC, GAGCCGAA>
<TCTAACGG, GAGCCATT>
<TCTATTCG, GAGCGGAT>
<TCTATCAC, GAGCGACA>
<TCTAGAGC, GAGCTCTA>
<TCTAGTCT, GAGCTGAG>
<TCTAGGCA, GAGCTTAC>
<TCTACATC, GAGCACGA>
<TCTACTTG, GAGCAGGT>
<TCTACGTT, GAGCATGG>
<TCTTAACC, GAGGCCAA>
<TCTTAGGG, GAGGCTTT>
<TCTTACGT, GAGGCATG>
<TCTTTTTC, GAGGGGGA>
<TCTTTGTG, GAGGGTGT>
<TCTTTCTT, GAGGGAGG>
<TCTTGACT, GAGGTCAG>
<TCTTGTCA, GAGGTGAC>
<TCTTGCAT, GAGGTACG>
<TCTTCAGT, GAGGACTG>
<TCTTCTGA, GAGGAGTC>
<TCTGAAGC, GAGTCCTA>
<TCTGATCT, GAGTCGAG>
<TCTGAGCA, GAGTCTAC>
<TGCTTCTA, GTAGGAGC>
<TGCTGACA, GTAGTCAC>
<TGCTGGAT, GTAGTTCG>
<TGCTCATT, GTAGACGG>
<TGCTCGAA, GTAGATCC>
<TGCGATAC, GTATCGCA>
<TGCGAGAG, GTATCTCT>
<TGCGACTA, GTATCAGC>
<TGCGTACA, GTATGCAC>
<TGCGTGAT, GTATGTCG>
<TGCGGATT, GTATTCGG>
<TGCGGGAA, GTATTTCC>
<TGCCAAAC, GTAACCCA>
<TGCCATAG, GTAACGCT>
<TGCCAGTA, GTAACTGC>
<TGCCTAGA, GTAAGCTC>
<TGCCGAAT, GTAATCCG>
<TCAAAGCC, GACCCTAA>
<TCAATGGC, GACCGTTA>
<TCAATCCT, GACCGAAG>
<TCAAGGTC, GACCTTGA>
<TCAAGCTG, GACCTAGT>
<TCAACACG, GACCACAT>
<TCAACGAC, GACCATCA>
<TCAACCAG, GACCAACT>
<TCATAGGC, GACGCTTA>
<TCATACCT, GACGCAAG>
<TCATTGTC, GACGGTGA>
<TCATTCTG, GACGGAGT>
<TCATGACG, GACGTCAT>
<TCATGGAC, GACGTTCA>
<TCATGCAG, GACGTACT>
<TCATCAGG, GACGACTT>
<TCATCTGT, GACGAGTG>
<TCATCGGA, GACGATTC>
<TCAGATGC, GACTCGTA>
<TCAGAGCT, GACTCTAG>
<TCAGACCA, GACTCAAC>
<TCAGTTGG, GACTGGTT>
<TCAGTGGT, GACTGTTG>
<TCAGTCGA, GACTGATC>
<TCAGGTAC, GACTTGCA>
<TCAGGGAG, GACTTTCT>
<TCAGGCTA, GACTTAGC>
<TCAGCACA, GACTACAC>
<TCAGCGAT, GACTATCG>
<TCACAACG, GACACCAT>
<TCACAGAC, GACACTCA>
<TCACACAG, GACACACT>
<TCACTAGG, GACAGCTT>
<TCACTTGT, GACAGGTG>
<TCACTGGA, GACAGTTC>
<TCACGATG, GACATCGT>
<TCACGTTT, GACATGGG>
<TCACGCAA, GACATACC>
<TCACCTAT, GACAAGCG>
<TCTAAGGC, GAGCCTTA>
<TCTAACCT, GAGCCAAG>
<TCTATGTC, GAGCGTGA>
<TCTATCTG, GAGCGAGT>
<TCTAGACG, GAGCTCAT>
<TCTAGGAC, GAGCTTCA>
<TCTAGCAG, GAGCTACT>
<TCTACAGG, GAGCACTT>
<TCTACTGT, GAGCAGTG>
<TCTACGGA, GAGCATTC>
<TCTTATGC, GAGGCGTA>
<TCTTAGCT, GAGGCTAG>
<TCTTACCA, GAGGCAAC>
<TCTTTTGG, GAGGGGTT>
<TCTTTGGT, GAGGGTTG>
<TCTTTCGA, GAGGGATC>
<TCTTGTAC, GAGGTGCA>
<TCTTGGAG, GAGGTTCT>
<TCTTGCTA, GAGGTAGC>
<TCTTCACA, GAGGACAC>
<TCTTCGAT, GAGGATCG>
<TCTGAACG, GAGTCCAT>
<TCTGAGAC, GAGTCTCA>
<TCTGACAG, GAGTCACT>

<TCTGACTT, GAGTCAGG>
<TCTGTACT, GAGTGCAG>
<TCTGTTCA, GAGTGGAC>
<TCTGTCAT, GAGTGACG>
<TCTGGAGT, GAGTTCTG>
<TCTGGTGA, GAGTTGTC>
<TCTGCAAG, GAGTACCT>
<TCTGCTTA, GAGTAGGC>
<TCTCAACT, GAGACCAG>
<TCTCATCA, GAGACGAC>
<TCTCACAT, GAGACACG>
<TCTCTAGT, GAGAGCTG>
<TCTCTTGA, GAGAGGTC>
<TCTCGAAG, GAGATCCT>
<TCTCGTTA, GAGATGGC>
<TCTCCTAA, GAGAAGCC>
<TCGAAGTC, GATCCTGA>
<TCGAACTG, GATCCAGT>
<TCGATACG, GATCGCAT>
<TCGATGAC, GATCGTCA>
<TCGATCAG, GATCGACT>
<TCGAGAGG, GATCTCTT>
<TCGAGTGT, GATCTGTG>
<TCGAGGGA, GATCTTTC>
<TCGACATG, GATCACGT>
<TCGACTTT, GATCAGGG>
<TCGACCAA, GATCAACC>
<TCGTATGG, GATGCGTT>
<TCGTAGGT, GATGCTTG>
<TCGTACGA, GATGCATC>
<TCGTTTAC, GATGGGCA>
<TCGTTGAG, GATGGTCT>
<TCGTTCTA, GATGGAGC>
<TCGTGACA, GATGTCAC>
<TCGTGGAT, GATGTTCG>
<TCGTCATT, GATGACGG>
<TCGTCGAA, GATGATCC>
<TCGGATAC, GATTCGCA>
<TCGGAGAG, GATTCTCT>
<TCGGACTA, GATTCAGC>
<TCGGTACA, GATTGCAC>
<TCGGTGAT, GATTGTCG>
<TCGGGATT, GATTTCGG>
<TCGGGGAA, GATTTTCC>
<TCGCAAAC, GATACCCA>
<TCGCATAG, GATACGCT>
<TCGCAGTA, GATACTGC>
<TCGCTAGA, GATAGCTC>
<TCGCGAAT, GATATCCG>
<TCCAAAGC, GAACCCTA>
<TCCAATCT, GAACCGAG>
<TCCAAGCA, GAACCTAC>
<TCCATATC, GAACGCGA>
<TCCATTTG, GAACGGGT>
<TCCATGTT, GAACGTGG>
<TCCAGAAC, GAACTCCA>
<TCCAGTAG, GAACTGCT>
<TCCAGGTA, GAACTTGC>
<TCCACAGA, GAACACTC>
<TCCTAATC, GAAGCCGA>
<TCCTATTG, GAAGCGGT>
<TCCTAGTT, GAAGCTGG>
<TCCTTAAC, GAAGGCCA>
<TCCTTTAG, GAAGGGCT>
<TCCTTGTA, GAAGGTGC>
<TCCTGAGA, GAAGTCTC>
<TCCTCAAT, GAAGACCG>
<TCCGAATG, GAATCCGT>
<TCCGATTT, GAATCGGG>
<TCCGACAA, GAATCACC>
<TCCGTTAT, GAATGGCG>
<TCCGGATA, GAATTCGC>
<TCCCAATT, GAAACCGG>
<TCCCAGAA, GAAACTCC>
<TCCCGAAA, GAAATCCC>
<TCTGACGA, GAGTCATC>
<TCTGTTAC, GAGTGGCA>
<TCTGTGAG, GAGTGTCT>
<TCTGTCTA, GAGTGAGC>
<TCTGGACA, GAGTTCAC>
<TCTGGGAT, GAGTTTCG>
<TCTGCATT, GAGTACGG>
<TCTGCGAA, GAGTATCC>
<TCTCATAC, GAGACGCA>
<TCTCAGAG, GAGACTCT>
<TCTCACTA, GAGACAGC>
<TCTCTACA, GAGAGCAC>
<TCTCTGAT, GAGAGTCG>
<TCTCGATT, GAGATCGG>
<TCTCGGAA, GAGATTCC>
<TCGAAACC, GATCCCAA>
<TCGAAGGG, GATCCTTT>
<TCGAACGT, GATCCATG>
<TCGATTTC, GATCGGGA>
<TCGATGTG, GATCGTGT>
<TCGATCTT, GATCGAGG>
<TCGAGACT, GATCTCAG>
<TCGAGTCA, GATCTGAC>
<TCGAGCAT, GATCTACG>
<TCGACAGT, GATCACTG>
<TCGACTGA, GATCAGTC>
<TCGTAAGC, GATGCCTA>
<TCGTATCT, GATGCGAG>
<TCGTAGCA, GATGCTAC>
<TCGTTATC, GATGGCGA>
<TCGTTTTG, GATGGGGT>
<TCGTTGTT, GATGGTGG>
<TCGTGAAC, GATGTCCA>
<TCGTGTAG, GATGTGCT>
<TCGTGGTA, GATGTTGC>
<TCGTCAGA, GATGACTC>
<TCGGAATC, GATTCCGA>
<TCGGATTG, GATTCGGT>
<TCGGAGTT, GATTCTGG>
<TCGGTAAC, GATTGCCA>
<TCGGTTAG, GATTGGCT>
<TCGGTGTA, GATTGTGC>
<TCGGGAGA, GATTTCTC>
<TCGGCAAT, GATTACCG>
<TCGCAATG, GATACCGT>
<TCGCATTT, GATACGGG>
<TCGCACAA, GATACACC>
<TCGCTTAT, GATAGGCG>
<TCGCGATA, GATATCGC>
<TCCAAACG, GAACCCAT>
<TCCAAGAC, GAACCTCA>
<TCCAACAG, GAACCACT>
<TCCATAGG, GAACGCTT>
<TCCATTGT, GAACGGTG>
<TCCATGGA, GAACGTTC>
<TCCAGATG, GAACTCGT>
<TCCAGTTT, GAACTGGG>
<TCCAGCAA, GAACTACC>
<TCCACTAT, GAACAGCG>
<TCCTAAGG, GAAGCCTT>
<TCCTATGT, GAAGCGTG>
<TCCTAGGA, GAAGCTTC>
<TCCTTATG, GAAGGCGT>
<TCCTTTTT, GAAGGGGG>
<TCCTTCAA, GAAGGACC>
<TCCTGTAT, GAAGTGCG>
<TCCTCATA, GAAGACGC>
<TCCGAAGT, GAATCCTG>
<TCCGATGA, GAATCGTC>
<TCCGTAAG, GAATGCCT>
<TCCGTTTA, GAATGGGC>
<TCCGGTAA, GAATTGCC>
<TCCCAAGA, GAAACCTC>
<TCCCTAAT, GAAAGCCG>
<TCCCCCCC, GAAAAAAA>
<TCTGTATC, GAGTGCGA>
<TCTGTTTG, GAGTGGGT>
<TCTGTGTT, GAGTGTGG>
<TCTGGAAC, GAGTTCCA>
<TCTGGTAG, GAGTTGCT>
<TCTGGGTA, GAGTTTGC>
<TCTGCAGA, GAGTACTC>
<TCTCAATC, GAGACCGA>
<TCTCATTG, GAGACGGT>
<TCTCAGTT, GAGACTGG>
<TCTCTAAC, GAGAGCCA>
<TCTCTTAG, GAGAGGCT>
<TCTCTGTA, GAGAGTGC>
<TCTCGAGA, GAGATCTC>
<TCTCCAAT, GAGAACCG>
<TCGAATGC, GATCCGTA>
<TCGAAGCT, GATCCTAG>
<TCGAACCA, GATCCAAC>
<TCGATTGG, GATCGGTT>
<TCGATGGT, GATCGTTG>
<TCGATCGA, GATCGATC>
<TCGAGTAC, GATCTGCA>
<TCGAGGAG, GATCTTCT>
<TCGAGCTA, GATCTAGC>
<TCGACACA, GATCACAC>
<TCGACGAT, GATCATCG>
<TCGTAACG, GATGCCAT>
<TCGTAGAC, GATGCTCA>
<TCGTACAG, GATGCACT>
<TCGTTAGG, GATGGCTT>
<TCGTTTGT, GATGGGTG>
<TCGTTGGA, GATGGTTC>
<TCGTGATG, GATGTCGT>
<TCGTGTTT, GATGTGGG>
<TCGTGCAA, GATGTACC>
<TCGTCTAT, GATGAGCG>
<TCGGAAGG, GATTCCTT>
<TCGGATGT, GATTCGTG>
<TCGGAGGA, GATTCTTC>
<TCGGTATG, GATTGCGT>
<TCGGTTTT, GATTGGGG>
<TCGGTCAA, GATTGACC>
<TCGGGTAT, GATTTGCG>
<TCGGCATA, GATTACGC>
<TCGCAAGT, GATACCTG>
<TCGCATGA, GATACGTC>
<TCGCTAAG, GATAGCCT>
<TCGCTTTA, GATAGGGC>
<TCGCGTAA, GATATGCC>
<TCCAATTC, GAACCGGA>
<TCCAAGTG, GAACCTGT>
<TCCAACTT, GAACCAGG>
<TCCATACT, GAACGCAG>
<TCCATTCA, GAACGGAC>
<TCCATCAT, GAACGACG>
<TCCAGAGT, GAACTCTG>
<TCCAGTGA, GAACTGTC>
<TCCACAAG, GAACACCT>
<TCCACTTA, GAACAGGC>
<TCCTAACT, GAAGCCAG>
<TCCTATCA, GAAGCGAC>
<TCCTACAT, GAAGCACG>
<TCCTTAGT, GAAGGCTG>
<TCCTTTGA, GAAGGGTC>
<TCCTGAAG, GAAGTCCT>
<TCCTGTTA, GAAGTGGC>
<TCCTCTAA, GAAGAGCC>
<TCCGAACA, GAATCCAC>
<TCCGAGAT, GAATCTCG>
<TCCGTATT, GAATGCGG>
<TCCGTGAA, GAATGTCC>
<TCCGCAAA, GAATACCC>
<TCCCATAT, GAAACGCG>
<TCCCTATA, GAAAGCGC>
<TCTGTAGG, GAGTGCTT>
<TCTGTTGT, GAGTGGTG>
<TCTGTGGA, GAGTGTTC>
<TCTGGATG, GAGTTCGT>
<TCTGGTTT, GAGTTGGG>
<TCTGGCAA, GAGTTACC>
<TCTGCTAT, GAGTAGCG>
<TCTCAAGG, GAGACCTT>
<TCTCATGT, GAGACGTG>
<TCTCAGGA, GAGACTTC>
<TCTCTATG, GAGAGCGT>
<TCTCTTTT, GAGAGGGG>
<TCTCTCAA, GAGAGACC>
<TCTCGTAT, GAGATGCG>
<TCTCCATA, GAGAACGC>
<TCGAATCG, GATCCGAT>
<TCGAACAC, GATCCACA>
<TCGATAGC, GATCGCTA>
<TCGATTCT, GATCGGAG>
<TCGATGCA, GATCGTAC>
<TCGAGATC, GATCTCGA>
<TCGAGTTG, GATCTGGT>
<TCGAGGTT, GATCTTGG>
<TCGACAAC, GATCACCA>
<TCGACTAG, GATCAGCT>
<TCGACGTA, GATCATGC>
<TCGTATTC, GATGCGGA>
<TCGTAGTG, GATGCTGT>
<TCGTACTT, GATGCAGG>
<TCGTTACT, GATGGCAG>
<TCGTTTCA, GATGGGAC>
<TCGTTCAT, GATGGACG>
<TCGTGAGT, GATGTCTG>
<TCGTGTGA, GATGTGTC>
<TCGTCAAG, GATGACCT>
<TCGTCTTA, GATGAGGC>
<TCGGAACT, GATTCCAG>
<TCGGATCA, GATTCGAC>
<TCGGACAT, GATTCACG>
<TCGGTAGT, GATTGCTG>
<TCGGTTGA, GATTGGTC>
<TCGGGAAG, GATTTCCT>
<TCGGGTTA, GATTTGGC>
<TCGGCTAA, GATTAGCC>
<TCGCAACA, GATACCAC>
<TCGCAGAT, GATACTCG>
<TCGCTATT, GATAGCGG>
<TCGCTGAA, GATAGTCC>
<TCGCCAAA, GATAACCC>
<TCCAATGG, GAACCGTT>
<TCCAAGGT, GAACCTTG>
<TCCAACGA, GAACCATC>
<TCCATTAC, GAACGGCA>
<TCCATGAG, GAACGTCT>
<TCCATCTA, GAACGAGC>
<TCCAGACA, GAACTCAC>
<TCCAGGAT, GAACTTCG>
<TCCACATT, GAACACGG>
<TCCACGAA, GAACATCC>
<TCCTATAC, GAAGCGCA>
<TCCTAGAG, GAAGCTCT>
<TCCTACTA, GAAGCAGC>
<TCCTTACA, GAAGGCAC>
<TCCTTGAT, GAAGGTCG>
<TCCTGATT, GAAGTCGG>
<TCCTGGAA, GAAGTTCC>
<TCCGAAAC, GAATCCCA>
<TCCGATAG, GAATCGCT>
<TCCGAGTA, GAATCTGC>
<TCCGTAGA, GAATGCTC>
<TCCGGAAT, GAATTCCG>
<TCCCAAAG, GAAACCCT>
<TCCCATTA, GAAACGGC>
<TCCCTTAA, GAAAGGCC>

Observations

1. The fundamentally accepted notion in genetic science is that the nucleotides Adenine and Thymine form a base pair and Guanine and Cytosine form another base pair. An ordered finite length sequence of certain base pairs amounts to what is called a double stranded DNA. In other words, nucleotides are concatenated by phospho-diester bonds to form a long strand, which couples with its complementary strand to form a stable DNA. So, the constructive base pairs <A,T>, <T,A>, <G,C> and <C,G>, which are structurally connected by hydrogen bonds form pairs of nucleotides in the DNA structure.
2. One can observe from the study of '**Dual Pairs**' the ordered pairs of nucleotides <A,C>, <C,A>, <T,G> and <G,T> are destructive in the sense that they do not form a bonded structure. On the other hand, they may neutralize each other.
3. A virus genome consisting of finite number of nucleotides may have quadruples, which are dual sequences that pair with other dual sequencs. Fortyfour quadruple dual sequences were tested for their frequency of occurrences in all nine virus genome sequences. Similarly, 8164 8-tuples were tested for their frequency of occurrences in all nine virus genome sequences. It was found that the more the frequency, more the possibility of the virus being neutralized.

Case Study:

Nine virus sequences (1) SARS coronavirus SZ16, complete genome, (2) SARS coronavirus ZS-C, complete genome, (3) Bat SARS coronavirus Rm1, complete genome, (4) Bat SARS-like coronavirus isolate Rs4084, complete genome, (5) Wuhan seafood market, pneumonia virus isolate Wuhan-Hu-1, complete genome, (6) Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete genome, (7) Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome, (8) Bat SARS-like coronavirus isolate bat-SL-CoVZC45, complete genome, (9) Bat SARS-like coronavirus isolate bat-SL-CoVZXC21, complete genome, obtained from NCBI genebank are considered here for this experiment.

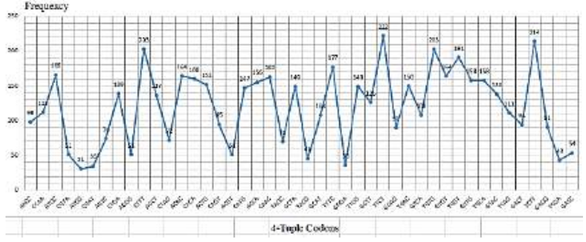
4-tuple codons to be searched in nine genome virus sequences

AACC, CCAA, ATGC, CGTA, ATCG, CGAT, AGTC, CTGA, AGGG, CTTT, AGCT, CTAG, ACAC, CACA, ACTG, CAGT, ACGT, CATG, ACCA, CAAC, TAGC, GCTA, TACG, GCAT, TTTC, GGGA, TTGG, GGTT, TTCT, GGAG, TGAC, GTCA,

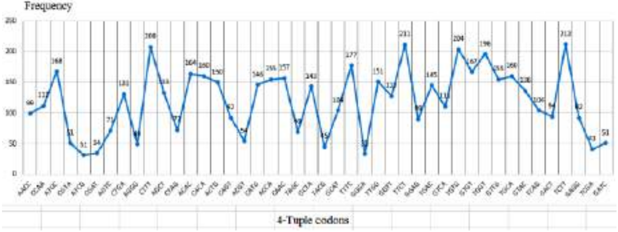
Table 6: Frequency of occurrence of 4 tuple potent codons

Genome 1		Genome 2		Genome 3		Genome 4		Genome 5		Genome 6		Genome 7		Genome 8		Genome 9	
Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency	Potent 4-tuple odons	frequency
AACC	98	AACC	99	AACC	95	AACC	90	AACC	113	AACC	112	AACC	112	AACC	95	AACC	99
CCAA	112	CCAA	112	CCAA	134	CCAA	113	CCAA	108	CCAA	107	CCAA	104	CCAA	106	CCAA	118
ATGC	166	ATGC	168	ATGC	158	ATGC	173	ATGC	151	ATGC	150	ATGC	140	ATGC	149	ATGC	143
CGTA	51	CGTA	51	CGTA	44	CGTA	51	CGTA	41	CGTA	40	CGTA	41	CGTA	51	CGTA	48
ATCG	31	ATCG	31	ATCG	34	ATCG	31	ATCG	20	ATCG	20	ATCG	20	ATCG	17	ATCG	20
CGAT	35	CGAT	34	CGAT	45	CGAT	33	CGAT	29	CGAT	29	CGAT	28	CGAT	25	CGAT	33
AGTC	74	AGTC	71	AGTC	67	AGTC	72	AGTC	72	AGTC	71	AGTC	74	AGTC	71	AGTC	68
CTGA	139	CTGA	131	CTGA	135	CTGA	135	CTGA	112	CTGA	111	CTGA	109	CTGA	131	CTGA	124
AGGG	51	AGGG	49	AGGG	61	AGGG	49	AGGG	52	AGGG	51	AGGG	48	AGGG	54	AGGG	52
CTTT	203	CTTT	208	CTTT	184	CTTT	202	CTTT	243	CTTT	239	CTTT	241	CTTT	231	CTTT	237
AGCT	137	AGCT	133	AGCT	134	AGCT	140	AGCT	140	AGCT	138	AGCT	137	AGCT	123	AGCT	127
CTAG	72	CTAG	72	CTAG	77	CTAG	73	CTAG	84	CTAG	82	CTAG	84	CTAG	72	CTAG	78
ACAC	164	ACAC	164	ACAC	161	ACAC	159	ACAC	161	ACAC	159	ACAC	156	ACAC	162	ACAC	162
CACA	160	CACA	160	CACA	158	CACA	157	CACA	150	CACA	149	CACA	143	CACA	159	CACA	167
ACTG	152	ACTG	150	ACTG	154	ACTG	152	ACTG	154	ACTG	151	ACTG	150	ACTG	158	ACTG	149
CAGT	95	CAGT	92	CAGT	110	CAGT	97	CAGT	111	CAGT	108	CAGT	107	CAGT	115	CAGT	111
ACGT	51	ACGT	54	ACGT	58	ACGT	51	ACGT	63	ACGT	63	ACGT	62	ACGT	58	ACGT	60
CATG	147	CATG	146	CATG	140	CATG	144	CATG	118	CATG	116	CATG	114	CATG	129	CATG	123
ACCA	156	ACCA	155	ACCA	158	ACCA	153	ACCA	143	ACCA	142	ACCA	141	ACCA	153	ACCA	153
CAAC	163	CAAC	157	CAAC	167	CAAC	160	CAAC	191	CAAC	187	CAAC	187	CAAC	175	CAAC	170
TAGC	70	TAGC	69	TAGC	68	TAGC	75	TAGC	78	TAGC	76	TAGC	79	TAGC	70	TAGC	73
GCTA	149	GCTA	143	GCTA	136	GCTA	147	GCTA	134	GCTA	131	GCTA	135	GCTA	127	GCTA	128
TACG	46	TACG	45	TACG	48	TACG	43	TACG	48	TACG	48	TACG	47	TACG	50	TACG	59
GCAT	108	GCAT	104	GCAT	109	GCAT	116	GCAT	78	GCAT	77	GCAT	71	GCAT	98	GCAT	94
TTTC	177	TTTC	177	TTTC	160	TTTC	168	TTTC	159	TTTC	155	TTTC	150	TTTC	160	TTTC	169
GGGA	36	GGGA	33	GGGA	46	GGGA	42	GGGA	31	GGGA	31	GGGA	33	GGGA	40	GGGA	43
TTGG	149	TTGG	151	TTGG	148	TTGG	152	TTGG	161	TTGG	160	TTGG	157	TTGG	154	TTGG	145
GGTT	126	GGTT	127	GGTT	123	GGTT	122	GGTT	139	GGTT	138	GGTT	142	GGTT	132	GGTT	133
TTCT	222	TTCT	211	TTCT	197	TTCT	199	TTCT	210	TTCT	206	TTCT	203	TTCT	209	TTCT	203
GGAG	90	GGAG	90	GGAG	92	GGAG	90	GGAG	58	GGAG	58	GGAG	61	GGAG	64	GGAG	65
TGAC	150	TGAC	145	TGAC	152	TGAC	143	TGAC	132	TGAC	130	TGAC	131	TGAC	129	TGAC	140
GTCA	108	GTCA	111	GTCA	110	GTCA	115	GTCA	94	GTCA	91	GTCA	94	GTCA	97	GTCA	98
TGTG	203	TGTG	204	TGTG	197	TGTG	198	TGTG	185	TGTG	183	TGTG	182	TGTG	192	TGTG	196
GTGT	164	GTGT	167	GTGT	177	GTGT	172	GTGT	189	GTGT	187	GTGT	178	GTGT	183	GTGT	186
TGGT	191	TGGT	196	TGGT	184	TGGT	191	TGGT	236	TGGT	235	TGGT	232	TGGT	227	TGGT	219
GTTG	158	GTTG	155	GTTG	147	GTTG	150	GTTG	187	GTTG	185	GTTG	198	GTTG	179	GTTG	180
TGCA	158	TGCA	160	TGCA	158	TGCA	169	TGCA	156	TGCA	155	TGCA	154	TGCA	158	TGCA	152
GTAC	138	GTAC	136	GTAC	123	GTAC	131	GTAC	129	GTAC	125	GTAC	124	GTAC	131	GTAC	130
TCAG	111	TCAG	104	TCAG	119	TCAG	100	TCAG	112	TCAG	108	TCAG	108	TCAG	104	TCAG	95
GACT	94	GACT	94	GACT	116	GACT	109	GACT	97	GACT	95	GACT	89	GACT	100	GACT	105
TCTT	214	TCTT	212	TCTT	191	TCTT	193	TCTT	215	TCTT	207	TCTT	199	TCTT	224	TCTT	225
GAGG	91	GAGG	92	GAGG	95	GAGG	92	GAGG	62	GAGG	62	GAGG	63	GAGG	75	GAGG	77
TCGA	43	TCGA	41	TCGA	45	TCGA	33	TCGA	20	TCGA	20	TCGA	21	TCGA	27	TCGA	24
GATC	54	GATC	51	GATC	54	GATC	57	GATC	62	GATC	58	GATC	56	GATC	53	GATC	51

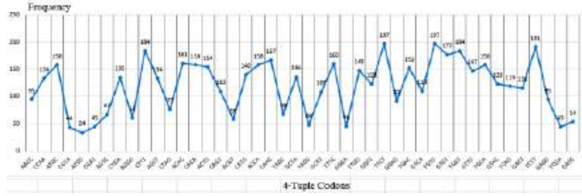
Fig. 51 shows frequency of occurrence graphs of 4-tuple potent codons in all nine virus genomes.



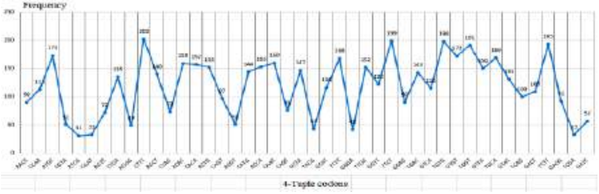
SARS coronavirus SZ16, complete Genome 1



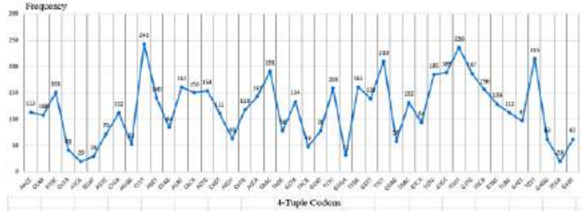
SARS coronavirus ZS-C, complete Genome 2



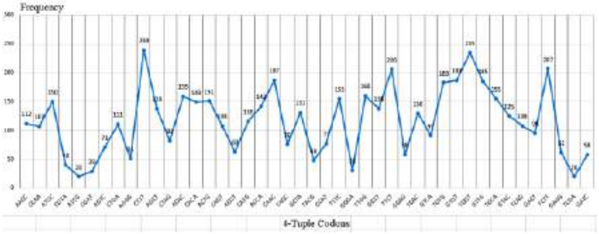
Bat SARS coronavirus Rm1, complete Genome 3



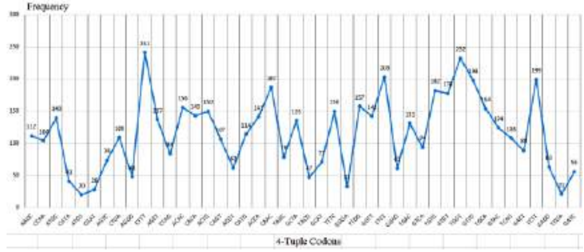
Bat SARS-like coronavirus isolate Rs4084, complete Genome 4



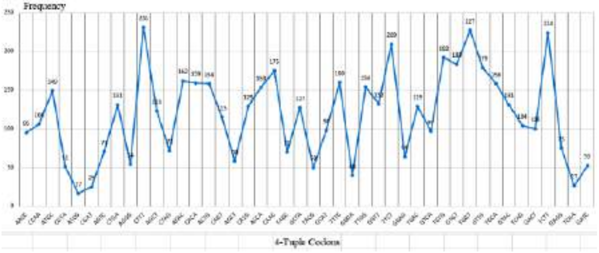
Wuhan seafood market, pneumonia virus isolate
Wuhan-Hu-1, complete Genome 5



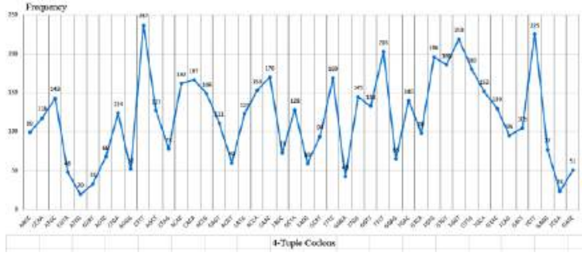
Wuhan seafood market pneumonia virus isolate
Wuhan-Hu-1, complete Genome 6



Wuhan seafood market pneumonia virus isolate
Wuhan-Hu-1, complete Genome 7



Bat SARS-like coronavirus isolate bat-SL-CoVZC45
complete Genome 8



Bat SARS-like coronavirus isolate bat-SL-CoVZXC21

Blank Space

Fig.51: Frequency of occurrences of 4 -tuple potent codons in all nine virus genomes

8-tuple codons to be searched in nine genome virus sequences

One can construct 65536 sequences of length 8 using the symbols A, T, G and C. They are called 8-tuple codons. It was found, out of 65536 such 8-tuple codons one can precisely have 8164 potent 8-tuple codons and 4082 pairs of 8-tuple potent codons as dual pairs. Listing of all 8164 potent codons and their frequencies of occurrences in nine genomes is not presented here due to space constraints. However, the frequencies of occurrences in nine virus genomes are portrayed in Fig. 52 to Fig. 60.

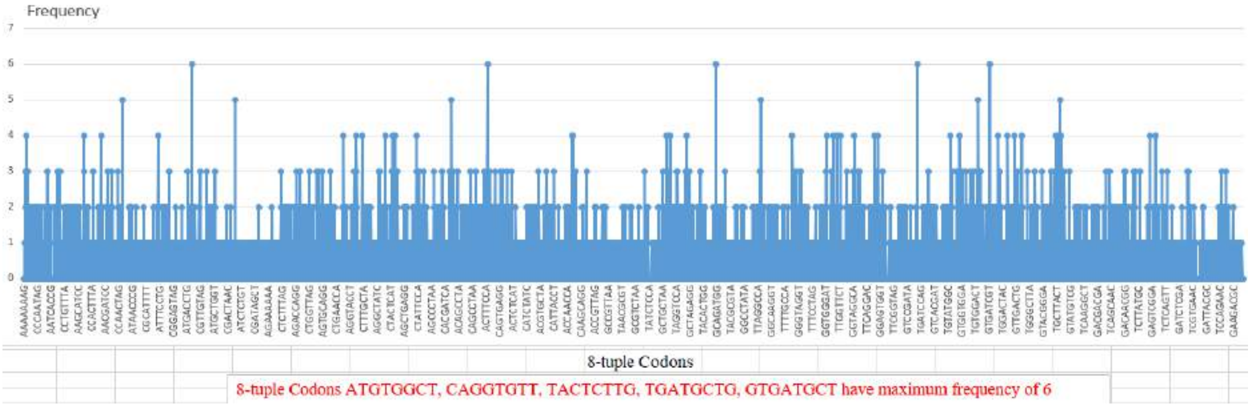


Fig.52: Maximum frequency shown by 8_tuple codons in SARS coronavirus SZ16, complete Genome 1

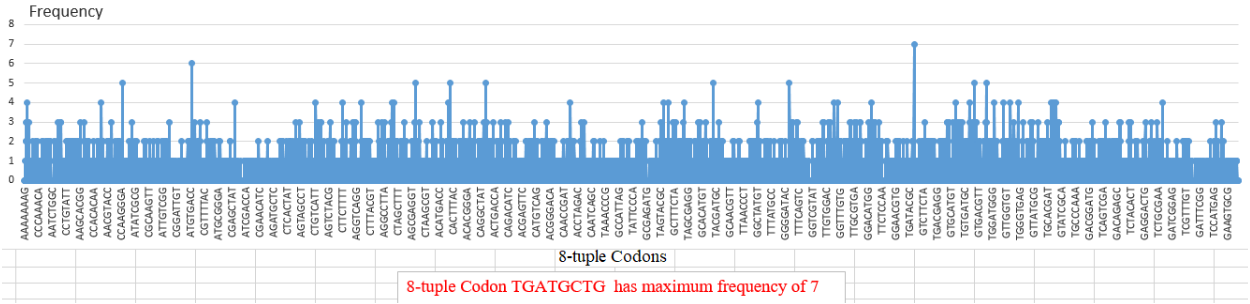


Fig.53: Maximum frequency shown by 8-tuple codons in SARS coronavirus SZ16, complete Genome 2

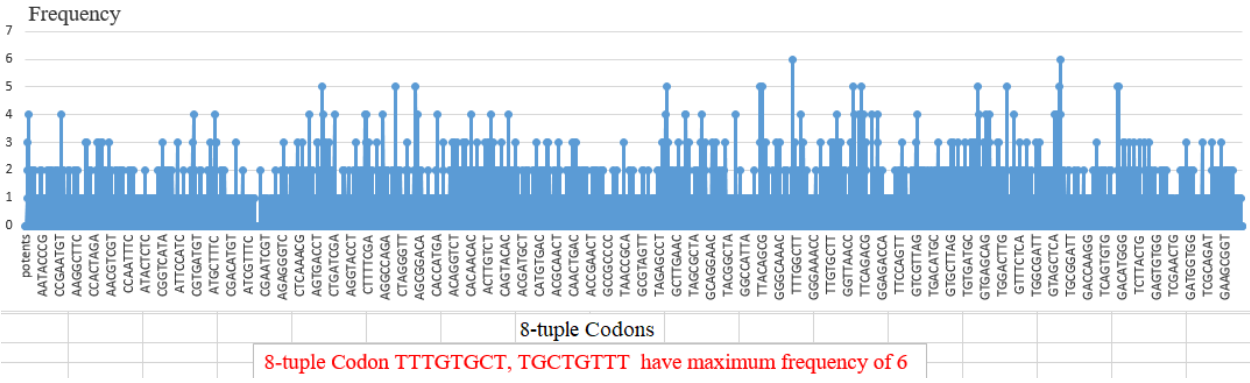


Fig.54: Maximum frequency shown by 8-tuple codons in Bat SARS coronavirus Rm1, complete Genome 3

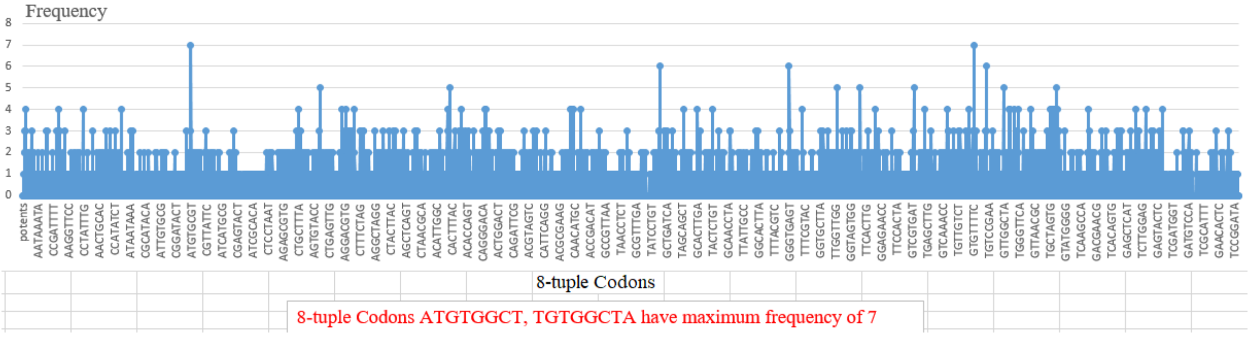


Fig.55: Maximum frequency shown by 8-tuple codons in Bat SARS-like coronavirus isolate Rs4084, complete Genome 4

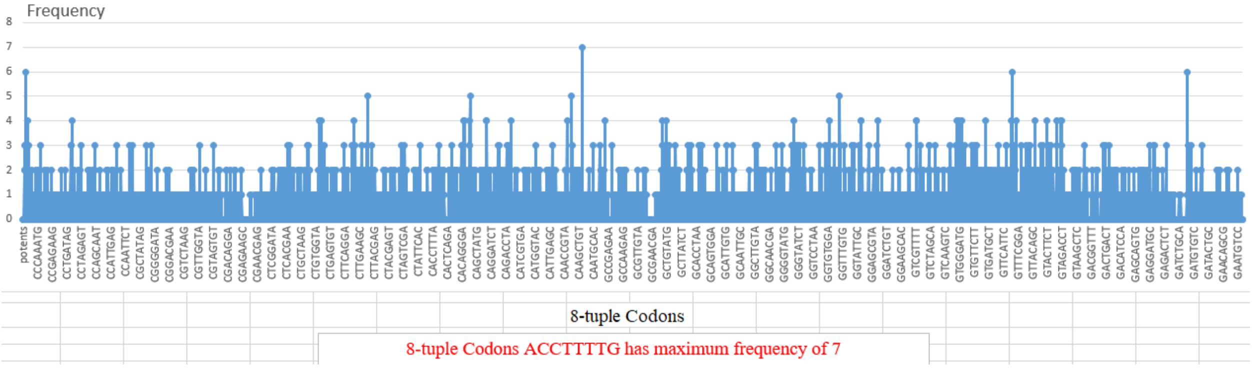


Fig.56: Maximum frequency shown by 8-tuple codons in Wuhan seafood market, pneumonia virus isolate Wuhan-Hu-1, complete Genome 5

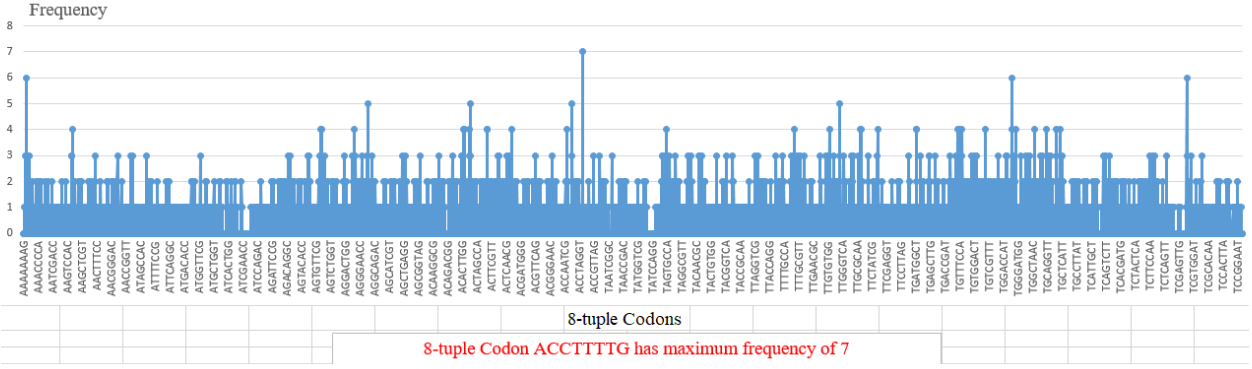


Fig.57: Maximum frequency shown by 8-tuple codons in Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete Genome 6

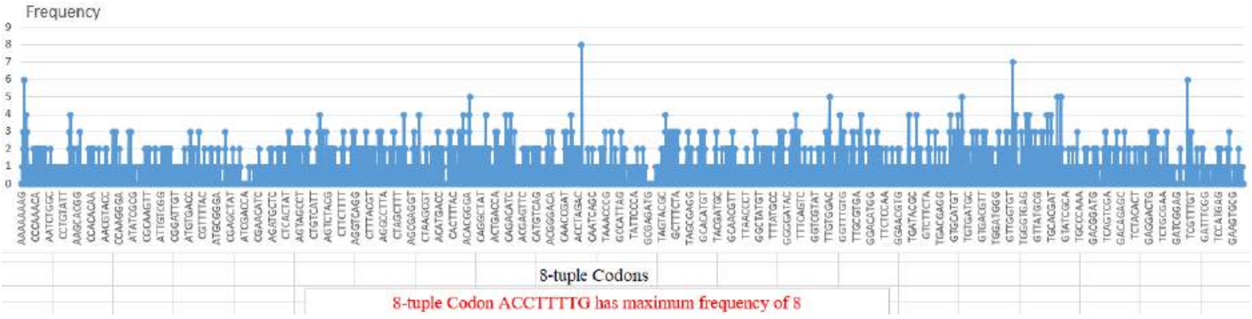


Fig.58: Maximum frequency shown by 8-tuple codons in Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete Genome 7

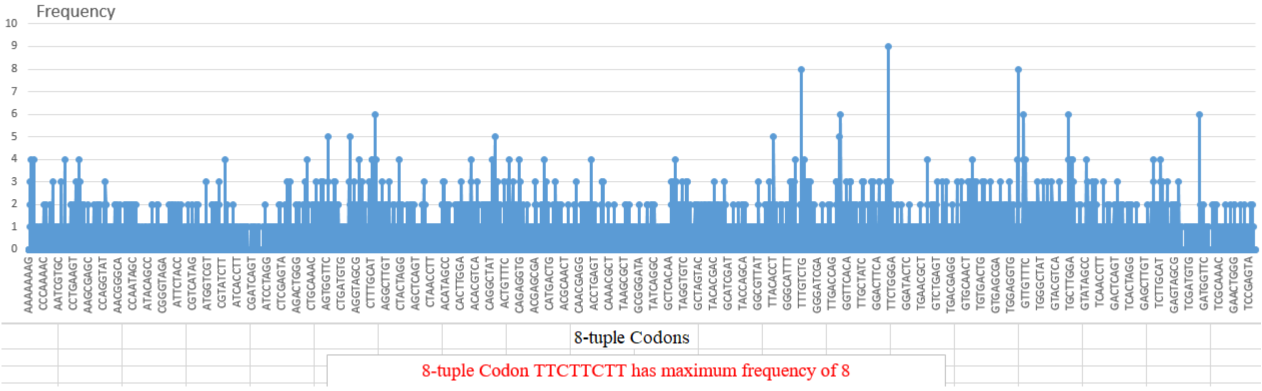


Fig.59: Maximum frequency shown by 8-tuple codons in Bat SARS-like coronavirus isolate bat-SL-CoVZC45 complete Genome 8

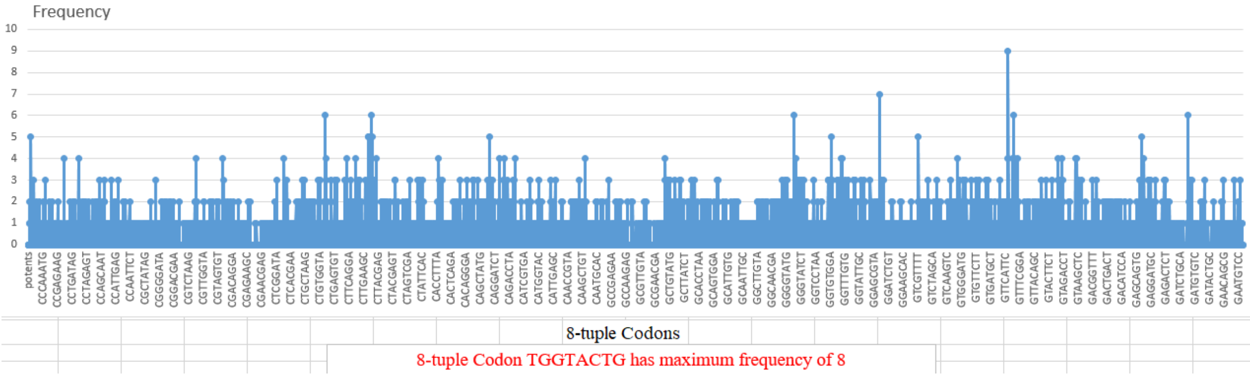


Fig.60: Maximum frequency shown by 8-tuple codons in
Bat SARS-like coronavirus isolate bat-SL-CoVZXC21 complete Genome 9

V. CONSOLIDATED OBSERVATIONS AND CONCLUDING REMARKS

Study of pairwise spectral correlations of nine SARS-Corona virus genomes was undertaken and results reported in this paper. A homomorphic transform called Rajan Transform was used as the main tool for the spectral analysis of virus genomes. Nucleotides of genomes are numerically represented and the transform applied on the equivalent number sequences. Adjoints of nucelotides are considered here for spectral analysis and pairwise spectral correlations among the nucleotide adjoints were explored. In addition, linear pair forming property of higher order Rajan Transform was tested on subsampled nucleotides adjoints and frequency of occurrences of potent 4-tuple and 8-tuple codons in all the original characteristic sequences of nine virus genomes were evaluated.

Significant results obtained are summarized in what follows.

1. Subsampled adjoints of Genomes 5 and 6 exhibit maximum similarity (86.3323% to 88.53%) in their RT spectra.
2. Subsampled adjoints of Genomes 6 and 7 also exhibit high-level similarity (85.5426% to 89.01%) in their RT spectra.
3. Cytosine adjoint arrays of genomes 6 and 7 exhibit the highest level similarity (**89.01%**) in their RT spectra.
4. To be more specific, even subsampled Cytosine adjoint arrays of Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete genome (Genome 6) and Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome (Genome 7) exhibit maximum similarity in their RT spectra, meaning their functional behavior is almost the same.
5. Rajan Transform is, by default, a nonlinear transform and so it does not satisfy linearity property at the first level. On the other hand, repeated application of Rajan Transform yields higher order spectra highlighting the point that the sequences are regulated and they do not change randomly. In fact, n^{th} order RT spectral components are $(n-2)^{nd}$ order RT spectral components multiplied by N, where N is the length of the sequences. Thus, one can precisely evaluate higer order spectra without applying Rajan Transform repeatedly. To generalize, let us assume two sequences of equal length of N. Let us also assume that at the r^{th} stage the sequences form linear pair, that is RT satisfies linearity property for that pair. This means that at the r^{th} stage, the pair enters into a '**Correlation Agreement**' (CA)', and we refere to this stage as '**Correlation Agreement Stage' (CAS)**' and the stage number as '**Correlation Agreement Stage Number**' (CASN)'. The question that arises here is whether one can apply this notion to randomly mutating virus genome sequences and possibly regulate the mutation process. If this conjecture were realizable, then one would be able to predict futuristic mutations of any virus, based on pairwise interactions among them in a host. A synthesized mutating RNA string with ACE2 receptacles when injected in a human body might force pairs of in vivo viruses equipped with stem proteins to get into self regulation and then neutralized by a vaccine. For example, the CASN for Genome 1 has been found to be 7. One can repeat this experiment for various genome pairs.
6. It was observed during the study of '**Dual Pairs**', the ordered pairs of nucleotides <A,C>, <C,A>, <T,G> and <G,T> are destructive in the sense that they do not form a bonded structure. On the other hand, they may neutralize each other. A virus genome consisting of finite number of nucleotides may have quadruples, which are dual sequences that pair with other dual sequencs. Forty four quadruple dual sequences were tested for their frequency of occurrences in all nine virus genome sequences. Similarly, 8164 8-tuples were tested for their frequency of occurrences in all nine virus genome sequences. It was found that the more the frequency, more the possibility of the virus being neutralized.
7. It was found that the virus genomes (1) SARS coronavirus SZ16, complete genome, (2) SARS coronavirus ZS-C, complete genome, (3) Bat SARS coronavirus Rm1, complete genome and (4) Bat SARS-like coronavirus isolate Rs4084, complete genome were in the primitive stage of evolution. On the other hand, the virus genomes (5) Wuhan seafood market, pneumonia virus isolate Wuhan-Hu-1, complete genome, (6) Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete genome, (7) Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome, (8) Bat SARS-like coronavirus isolate bat-SL-CoVZC45, complete genome, (9) Bat SARS-like coronavirus isolate bat-SL-CoVZXC21, complete genome are completely evolved versions and their functional behavior has been observed to be very much similar.

8. Maximum frequencies of 4-tuple potent codons (yellow colored) in all nine virus genomes vary from 197 to 241.
9. Maximum frequencies of 4-tuple dual codons (green colored) in all nine virus genomes vary from 52 to 177.
10. Maximum frequencies of 8-tuple potent codons in all nine virus genomes vary from 6 to 9.
11. As a result, 4-tuple dual codons should be considered for neutralizing mutating virus genomes.

Question

Is the present day Covid-19 pandemic situation ‘natural or man-made’ ?

Research Finding

The research carried out and reported in this document points out to the possibility that the pandemic situation could be fabricated. However, this research shows a direction as to how to combat such situations in future and develop anti-viral RNA forms, which would combat any unwarranted threat.

ॐ सह नाववतु । सह नौ भुनक्तु । सह वीर्यं करवावहै । तेजस्वि नावधीतमस्तु मा विद्विषावहै ।
ॐ शान्तिः शान्तिः शान्तिः ॥

Om, Together may we two, the teacher and the taught move; Together may we two relish our research;
Together may we perform our research with vigour; May what has been studied by us be filled with abundant knowledge;
May it not give rise to lack of understanding;
Om Peace, Peace, Peace.

An Activity of Digital Health Initiative Started by the Team

Scan to know paper details and
author's profile

Knowledge Discovery with Data Mining

R. Romela Elizabeth Preena, Mrs. G. Priya Darshini & D. Jeya Rani

ABSTRACT

The potential returns are colossal. Creative associations overall are now utilizing Data mining to find and request to higher-esteem clients, to reconfigure their item contributions to expand deals, and to limit misfortunes because of mistake or misrepresentation. Data mining is a procedure that utilizes an assortment of Data investigation instruments to find examples and connections in Data that might be utilized to make substantial forecasts. The first and easiest diagnostic advance in quite a while mining is to depict the Data its measurable traits, outwardly survey it utilizing diagrams and charts, and search for conceivably significant connections among factors. As accentuated in the area on The Data Mining Process, gathering, investigating and choosing the correct Data are basically significant.

In any case, Data portrayal alone can't give an activity plan. It should assemble a prescient model dependent on designs decided from known out comes, at that point test that model on results outside the first example. A decent model ought to never be mistaken for the real world; hoitver it tends to be a valuable manual for understanding its business. The last advance is to exactly check the model.

Keywords: NA

Classification: H.2.8

Language: English

London
Journals Press

LJP Copyright ID: 975841
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 20 | Issue 2 | Compilation 1.0

Knowledge Discovery with Data Mining

R. Romela Elizabeth Preena[α], Mrs. G. Priya Darshini,[σ] & D. Jeya Rani[ρ]

ABSTRACT

The potential returns are colossal. Creative associations overall are now utilizing Data mining to find and request to higher-esteem clients, to reconfigure their item contributions to expand deals, and to limit misfortunes because of mistake or misrepresentation. Data mining is a procedure that utilizes an assortment of Data investigation instruments to find examples and connections in Data that might be utilized to make substantial forecasts. The first and easiest diagnostic advance in quite a while mining is to depict the Data its measurable traits, outwardly survey it utilizing diagrams and charts, and search for conceivably significant connections among factors. As accentuated in the area on The Data Mining Process, gathering, investigating and choosing the correct Data are basically significant.

In any case, Data portrayal alone can't give an activity plan. It should assemble a prescient model dependent on designs decided from known out comes, at that point test that model on results outside the first example. A decent model ought to never be mistaken for the real world; hoitver it tends to be a valuable manual for understanding its business. The last advance is to exactly check the model.

Author α: M. Sc, M.Phil., Department of Computer Application,KG College of Arts and Science.
σ: M.C.A, M.Phil. Department of Computer Applications KG College of Arts and Science.
ρ: M. Sc, M.Phil Department of Computer Applications KG College of Arts and Science.

I. INTRODUCTION

Data mining is an apparatus, not an enchantment wand. It won't sit in its database watching what occurs and send it email to stand out enough to be noticed when it sees an intriguing example. It doesn't take out the need to know its business, to comprehend its Data, or to comprehend investigative strategies. Data mining helps business experts with discovering examples and connections in the Data it doesn't reveal to it the estimation of the examples to the association. Moreover, the examples revealed by Data mining must be checked in reality.

To guarantee important outcomes, it's imperative that it comprehend its Data. The nature of its yield will regularly be delicate to anomalies, immaterial sections or segments that change together, the manner in which it encode its Data, and the Data it leave in and the Data it prohibit. Calculations shift in their affectability to such Data issues, yet it is incautious to rely upon an Data mining item to settle on quite a few choices all alone.

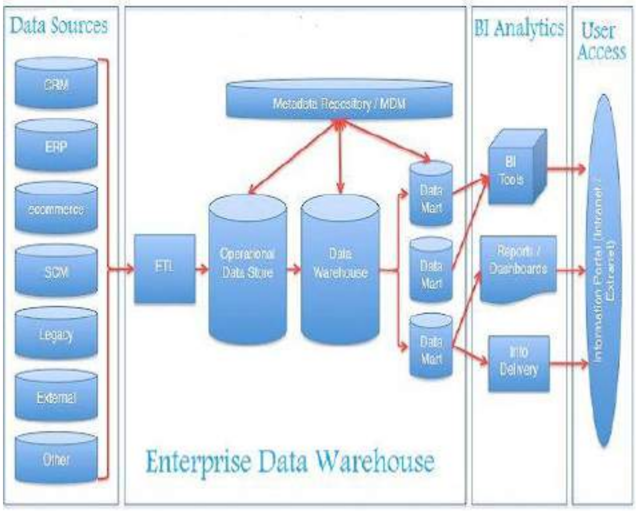
Data mining won't naturally find arrangements without direction. Instead of defining the dubious objective, "Help improve the reaction to my standard mail sales," it may utilize Data mining to discover the qualities of individuals who (1) react to its sales, or (2) react AND make an enormous buy. The examples Data digging finds for those two objectives might be altogether different. Albeit a decent Data mining device covers it from the complexities of factual procedures, it expects it to comprehend the functions of the apparatuses it pick and the calculations on which they are based.

The decisions it make in setting up its Data mining device and the advancements it pick will influence the exactness and speed of its models. Data mining doesn't supplant gifted business experts or chiefs, but instead gives them a ground-breaking new device to improve the activity they are doing. Any organization that knows its business and its clients is now mindful of numerous significant, high-result designs that its representatives have seen over the long time.

What Data mining can do is affirm such experimental perceptions and find new, inconspicuous examples that yield consistent gradual improvement.

II. DATA MINING AND DATA WAREHOUSING

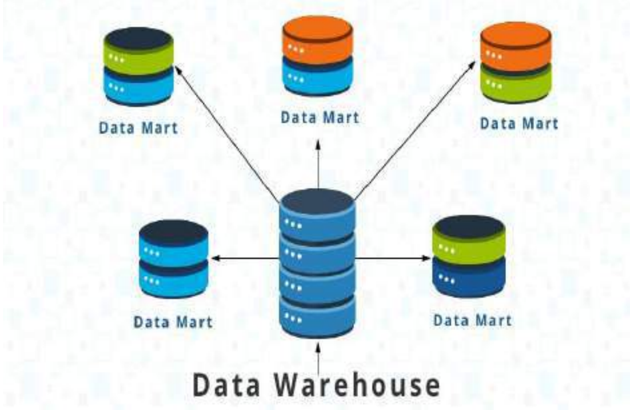
Every now and again, the Data to be mined is first separated from a venture Data stockroom into an Data mining database or Data store. There is some genuine advantage if its Data is as of now part of an Data distribution center. As it will see later on, the issues of purifying Data for a Data distribution center and for Data mining are fundamentally the same as. In the event that the Data has just been purified for a Data distribution center, at that point it in all likelihood won't need further cleaning so as to be mined. Besides, it will have just tended to huge numbers of the issues of Data combination and set up support techniques.



A Data distribution center isn't a prerequisite for Data mining. Setting up a huge Data distribution center that merges Data from numerous sources, settle Data honesty issues, and loads the Data into a question database can be a colossal errand, now and again taking years and costing a large number of dollars. It could, in any case, mine Data from at least one operational or value-based database by just extricating it into a read-just database. These new database capacities as a sort of Data bazaar.

III. DATA MINING AND OLAP

One of the most widely recognized inquiries from Data handling experts is about the contrast betiten Data mining and OLAP which represents On-Line Analytical Processing. As it will see, they are altogether different instruments that can supplement one another. OLAP is a piece of the range of choice help apparatuses. Conventional question and report apparatuses portray what is in a database. OLAP goes further; it's utilized to ansits why certain things are valid. The client frames a speculation about a relationship and checks it with a progression of questions against the Data. For instance, an examiner should decide the variables that lead to advance defaults.



Data mining is unique in relation to OLAP on the grounds that instead of check theoretical examples, it utilizes the Data itself to reveal such examples. It is basically an inductive procedure. For instance, assume the examiner who needed to recognize the hazard factors for credit default it's to utilize a Data mining device. The Data mining instrument may find that individuals with high obligation and low livelihoods its awful credit dangers, yet it may go further and furthermore find an example the investigator didn't think to attempt, for example, that age is additionally a determinant of hazard.

The examiner has to comprehend what the budgetary ramifications would be of utilizing the found example to oversee who gets credit. The OLAP instrument can permit the expert to respond to those sorts of inquiries. Besides, OLAP is additionally correlative in the beginning phases of the Data revelation process since it can assist it with investigating its Data, for example by concentrating consideration on significant factors, recognizing special cases, or discovering communications. This is significant in light of the fact that the better it comprehend its Data, the more compelling the Data disclosure procedure will be.

IV. DATA MINING, AI AND MEASUREMENTS

Data mining exploits progresses in the fields of man-made brainpoits (AI) and insights. The two orders have been taking a shot at issues of example acknowledgment and arrangement. The two networks have made extraordinary commitments to the comprehension and use of neural nets and choice trees.

Data mining doesn't supplant conventional factual methods. Or maybe, it is an expansion of factual techniques that is to a limited extent the consequence of a significant change in the insights network. The advancement of most factual strategies was, as of not long ago, in light of exquisite hypothesis and explanatory techniques that worked very till on the unobtrusive measures of Data being broke down. The expanded intensity of PCs and their loits cost, combined with the need to dissect colossal Dataal collections with a huge number of columns, have permitted the advancement of new methods dependent on a beast poits investigation of potential arrangements.

New strategies incorporate generally ongoing calculations like neural nets and choice trees, and new ways to deal with more established calculations, for example, discriminant investigation. By excellence of bringing to hold up under the expanded PC poits on the colossal volumes of accessible Data, these strategies can inexact practically any useful structure or cooperation all alone. Customary measurable procedures depend on the modeler to indicate the utilitarian structure and collaborations.

The key point is that Data mining is the utilization of these and other AI and measurable procedures to basic business issues in a manner that makes these strategies accessible to the gifted Data laborer just as the prepared insights proficient. Data digging is an apparatus for expanding the profitability of individuals attempting to manufacture prescient models.

V. DATA MINING AND EQUIPMENT / PROGRAMMING PATTERNS

The drop in the expense of PC preparing has been similarly emotional. Every age of chips significantly builds the intensity of the CPU, while permitting further drops on the cost bend. This is additionally reflected in the cost of RAM, where the expense of a megabyte has dropped from several dollars to around a dollar in only a couple of years. PCs routinely have 64 megabytes or a greater amount of RAM, and workstations may have 256 megabytes or more, while servers with gigabytes of fundamental memory are not an irregularity.

While the intensity of the individual CPU has extraordinarily expanded, the genuine advances in adaptability come from equal PC structures. For all intents and purposes all servers today bolster various CPUs utilizing symmetric multi-handling, and bunches of these SMP servers can be made that permit hundreds of CPUs to deal with discovering designs in the Data.

VI. DATA MINING APPLICATIONS

Data mining is progressively itll known in light of the significant commitment it can make. It tends to be utilized to control costs just as add to income increments. Numerous associations are utilizing Data mining to help deal with all periods of the client life cycle, including procuring new clients, expanding income from existing clients, and holding great clients. By deciding attributes of good clients, an organization can target possibilities with comparable qualities. By profiling clients who have purchased a specific item it can concentrate consideration on comparative clients who have not purchased that item. By profiling clients who have left, an organization can act to hold clients who are in danger of leaving, since it is for the most part far more affordable to hold a client than gain another one.

Data mining offers an incentive over a wide range of ventures. Media communications and Mastercard organizations are two of the pioneers in applying Data mining to distinguish fake utilization of their administrations. Insurance agencies and stock trades are likewise keen on applying this innovation to lessen misrepresentation. Clinical applications are another productive region: Data mining can be utilized to anticipate the viability of surgeries, clinical tests or drugs. Organizations dynamic in the money related markets use Data mining to decide market and industry attributes just as to foresee singular organization and stock execution. Retailers are utilizing Data mining to choose which items to stock specifically stores, just as to survey the adequacy of advancements and coupons. Pharmaceutical firms are mining huge databases of concoction mixes and of hereditary material to find substances that may be contender for improvement as operators for the medicines of illness.

VII. FRUITFUL DATA MINING

There are two keys to accomplishment in Data mining. First is concocting an exact plan of the difficult it are attempting to fathom. An engaged articulation for the most part brings about the best result. The subsequent key is utilizing the correct Data. In the wake of browsing the Data accessible to it, or maybe purchasing outside Data, it may need to change and consolidate it in noteworthy manners.

The more the model manufacturer can "play" with the Data, assemble models, assess results, and work with the Data some more, the better the subsequent model will be. Thus, how much an Data mining apparatus bolsters this intuitive Data investigation is a higher priority than the calculations it employments.

VIII. DATA DESCRIPTION FOR DATA MINING

Before it can assemble great prescient models, it should comprehend its Data. Start by social affair an assortment of numerical synopses and taking a gander at the dispersion of the Data. It might need to deliver cross arrangements for multidimensional Data.

The capacity to include a third, overlay variable incredibly builds the helpfulness of certain sorts of charts. Representation works since it abuses the more extensive data transfer speed of illustrations rather than text or numbers. It permits individuals to see the timberland and focus in on the trees. Examples, connections, excellent qualities and missing qualities are frequently simpler to see when indicated graphically, instead of as arrangements of numbers and text.

8.1 Grouping

Bunching isolates a database into various gatherings. The objective of bunching is to discover bunches that are totally different from one another, and whose individuals are fundamentally the same as one another. In contrast to order, it don't have the foggiest idea what the groups will be the point at which it start, or by which properties the Data will be bunched. Thusly, somebody who is educated in the business must decipher the groups. Regularly it is important to alter the bunching by barring factors that have been utilized to aggregate occasions, on the grounds that upon assessment the client recognizes them as immaterial or not significant. After it have discovered groups that sensibly portion its database, these bunches may then be utilized to order new Data. A portion of the basic calculations used to perform bunching incorporate Kohonen highlight guides and K-implies.

8.2 Connection investigation

Connection investigation is an engaging way to deal with investigating Data that can help distinguish connections among values in a database. The two most regular ways to deal with interface examination are affiliation revelation and succession disclosure. Affiliation disclosure discovers decides about things that show up together in an event such as a buy exchange. Market-crate investigation is a notable case of affiliation revelation. Succession disclosure is fundamentally the same as, in that an arrangement is an affiliation related after some time.

Affiliations are composed as A Þ B, where an is known as the predecessor or left-hand side (LHS), and B is known as the ensuing or right-hand side (RHS). For instance, in the affiliation rule "In the event that individuals purchase ahammer, at that point they purchase nails;" the precursor is "purchase a sledge" and the ensuing is "purchase nails." It's anything but difficult to decide the extent of exchanges that contain a specific thing or thing set: basically check them. The recurrence with which a specific affiliation shows up in the database is called its help or pervasiveness. On the off chance that, state, 15 exchanges out of 1,000 comprise of "sledge and nails," the help for this affiliation would be 1.5%. A low degree of help may show that the specific affiliation isn't significant or it might demonstrate the nearness of awful Data.

To find significant standards, be that as it may, it should likewise take a gander at the overall recurrence of event of the things and their mixes. Given the event of thing A (the forerunner), how regularly does thing B (the resulting) happen? That is, what is the restrictive consistency of B, given A? Utilizing the above model, this would mean asking "When individuals purchase a sledge, how regularly do they likewise purchase nails?" Another expression for this restrictive consistency is certainty. Certainty is determined as a proportion: (recurrence of An and B)/(recurrence of A).

Let's specify our hypothetical database in more detail to illustrate these concepts:

Total hardware-store transactions: 1,000
Number which include —hammer‖: 50
Number which include —nails‖: 80
Number which include —lumber‖: 20
Number which include —hammer‖ *and* —nails‖: 15
Number which include —nails‖ *and* —lumber‖: 10
Number which include —hammer‖ *and* —lumber‖: 10
Number which include —hammer,‖ —nails‖ *and*

It can now calculate:

Support for —hammer and nails‖ = 1.5% (15/1,000)
Support for —hammer, nails and lumber‖ = 0.5% (5/1,000)
Confidence of —hammer Þ nails‖ = 30% (15/50)
Confidence of —nails Þ hammer‖ = 19% (15/80)
Confidence of —hammer and nailsÞ lumber ‖ = 33% (5/15)
Confidence of —lumberÞ hammer and nails ‖ = 25% (5/20)

Thus it can see that the likelihood that a hammer buyer will also purchase nails (30%) is greater than the likelihood that someone buying nails will also purchase a hammer (19%). The prevalence of this hammer-and-nails association (the support is 1.5%) is high enough to suggest a meaningful rule. *Lift* is another measure of the poits of an association. The greater the lift, the greater the influence that the occurrence of A has on the likelihood that B will occur. Lift is calculated as the ratio (confidence of A Þ B)/ (frequency of B). From our example:

Lift of "hammer Þ nails": 3.75 (30%/8%)

Lift of "hammer and nails Þ lumber": 16.5 (33%/2%)

Affiliation calculations discover these guidelines by doing what could be compared to arranging the Data while checking events with the goal that they can compute certainty and backing. The productivity with which they can do this is one of the differentiators among calculations. This is particularly significant as a result of the combinatorial blast that outcomes in colossal quantities of rules, in any event, for advertise bins in the express path. A few calculations will make a database of rules, certainty factors, and bolster that can be.

Another basic trait of affiliation rule generators is the capacity to indicate a thing chain of command. In our model it has seen all nails and sledges, not singular sorts. It is imperative to pick a legitimate degree of total or it'll be probably not going to discover relationship of intrigue. A thing progressive system permits it to control the degree of accumulation and examination with various levels. Recollect that affiliation or succession rules are not so much guidelines, yet rather portrayals of connections in a specific database. There is no conventional testing of models on other Data to build the prescient intensity of these guidelines. Or maybe there is an understood suspicion that the previous conduct will proceed later on.

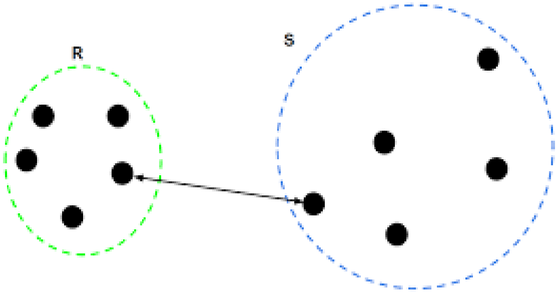
It is frequently hard to choose how to manage affiliation rules it've found. In store arranging, for instance, putting related things genuinely near one another may diminish the all out estimation of market containers clients may purchase less by and large since they no longer get impromptu things while strolling through the store looking for the ideal things. Knowledge, examination and experimentation are normally required to accomplish any profit by affiliation rules.

Graphical strategies may likewise be valuable in observing the structure of connections. Every one of the circles speaks to a worth or an occasion. The lines interfacing them show a connection. The thicker lines speak to more grounded or increasingly visit linkages, subsequently accentuating conceivably progressively significant connections, for example, affiliations. For example, taking a gander at a protection database to distinguish potential misrepresentation may uncover that a specific specialist and legal advisor cooperate on an uncommonly huge number of cases.

IX. PRESCIENT DATA MINING

9.1 A Chain of command of Ddecisions

The objective of Data mining is to deliver new Data that the client can follow up on. It does this by building a model of this present reality dependent on Data gathered from an assortment of sources which may incorporate corporate exchanges, client narratives and segment data, process control Data, and pertinent outer databases, for example, credit authority data or climate Data. The consequence of the model structure is a portrayal of examples and connections in the Data that can be unquestionably utilized for forecast.



To abstain from confounding the various parts of Data mining, it assists with imagining a chain of command of the decisions and choices
it have to make before it start:

- Business objective
- Type of expectation
- Model sort
- Algorithm
- Product

Business objective: what is a definitive motivation behind mining this Data? For instance, looking for designs in its Data to assist it with holding great clients, it may manufacture one model to foresee client productivity and a subsequent model to distinguish clients liable to leave (itaring down). Its insight into its association's needs and targets will direct it in planning the objective of its models.

9.2 Kind of forecast

(1) Grouping: anticipating into what classification or class a case falls,
(2) Relapse: foreseeing what number worth a variable will have. In the model above, it may utilize relapse to figure the measure of benefit, and order to foresee which clients may leave.

Model sort: A neural net to play out the relapse, maybe, and a choice tree for the grouping. There are additionally conventional measurable models to browse, for example, strategic relapse, discriminate examination, or general straight models. The most significant model sorts for Data mining are depicted in the following area, on Data Mining Models and Algorithms.

Calculations are accessible to assemble its models. It may assemble the neural net utilizing back propagation or spiral premise capacities. For the choice tree, it may pick among CART, C5.0, Quest, or CHAID.

Item: They for the most part have various executions of a specific calculation in any event, when they distinguish it with a similar name. These execution contrasts can influence operational attributes, for example, memory utilization and Data stockpiling, just as execution qualities, for example, speed and exactness. Other key contemplations to remember are shrouded later in the segment on selecting data mining products. Numerous business objectives are best met by building various model sorts utilizing an assortment of calculations. It will most likely be unable to figure out which model sort is best until it've attempted a few methodologies.

9.3 Some terminology

In prescient models, the qualities or classes it are anticipating are known as the reaction, ward or target factors. The qualities used to make the expectation are known as the indicator or free factors. Prescient models are fabricated, or prepared, utilizing Data for which the estimation of the reaction variable is as of now known. This sort of preparing is once in a while alluded to as regulated learning, in light of the fact that determined or assessed values are contrasted and the known outcomes.

9.4 Grouping

Arrangement issues expect to distinguish the qualities that demonstrate the gathering to which each case has a place. This example can be utilized both to comprehend the current Data and to foresee how new cases will act. For instance, it might need to anticipate whether people can be arranged as likely to react to a regular postal mail requesting, pointless against exchanging over to a contending long distance telephone administration, or a decent possibility for a surgery.

Data mining makes order models by inspecting previously arranged Data (cases) and inductively finding a prescient example. These current cases may originate from an authentic database, for example, individuals who have just experienced a specific clinical treatment or moved to another long distance administration. They may originate from a test where an example of the whole database is tried in reality and the outcomes used to make a classifier. For instance, an example of a mailing rundown would be sent an offer, and the consequences of the mailing used to build up an order model to be applied to the whole database. Some of the time a specialist orders an example of the database, and this arrangement is then used to make the model which will be applied to the whole database.

9.5 Relapse

Relapse utilizes existing qualities to figure what different qualities will be. In the least complex case, relapse utilizes standard measurable methods, for example, straight relapse. Lamentably, some certifiable issues are not just straight projections of past qualities. For example, deals volumes, stock costs, and item disappointment rates are largely hard to anticipate on the grounds that they may rely upon complex communications of various indicator factors. Along these lines, increasingly complex procedures might be important to figure future qualities.

A similar model sorts can regularly be utilized for both relapse and grouping. For instance, the CART choice tree calculation can be utilized to fabricate both arrangement trees and relapse trees. Neural nets also can make both arrangement and relapse models.

9.6 Time arrangement

Time arrangement determining predicts unknown future values dependent on a period differing arrangement of indicators. Like relapse, it utilizes realized outcomes to control its forecasts. Models must consider the particular properties of time, particularly the progressive system of periods (counting such fluctuated definitions as the five-or seven-day work itek, the thirteen-"month" year, and so forth.), irregularity, schedule impacts, for example, occasions, date math, and uncommon contemplations, for example, the amount of the past is important.

Most items use varieties of calculations that have been distributed in software engineering or measurements diaries, with their particular executions titaked to meet the individual seller's objective. For instance, numerous merchants sell renditions of the CART or CHAID choice trees with improvements to deal with equal PCs. A few sellers have restrictive calculations which, while not augmentations or improvements of any distributed methodology, may work very itll.

The vast majority of the models and calculations talked about in this area can be thought of as speculations of the standard workhorse of demonstrating, the direct relapse model. Much exertion has been exhausted in the insights, software engineering, man-made reasoning and building networks to defeat the confinements of this essential model. The basic trait of a significant number of the more up to date advances it will consider is that the example discovering component is Data driven instead of client driven. That is, the connections are found inductively by the product itself dependent on the current Data instead of requiring the modeler to indicate the useful structure and collaborations.

Maybe the most significant thing to recollect is that nobody model or calculation can or ought to beused only. For some random issue, the idea of the Data itself will influence the decision of modelsand calculations it pick. There is no "best" model or calculation. Therefore, it will require an assortment of devices and innovations so as to locate the most ideal model.

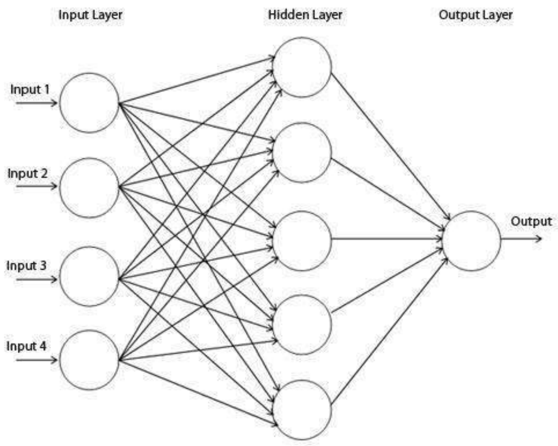
9.7 Neural Networks

Neural systems are specifically compelling in light of the fact that they offer methods for proficiently displaying huge and complex issues in which there might be several indicator factors that have numerous collaborations. Neural nets might be utilized in grouping issues or for relapses. After the Data layer, every hub takes in a lot of sources of info, increases them by an association itight Wxy, includes them together, applies a capacity to them, and passes the yield to the node(s) in the following layer. The engineering (or geography) of a neural system is the quantity of hubs and shrouded layers, and how they are associated. In structuring a neural system, either the client or the product must pick the quantity of concealed hubs and shrouded layers, the initiation capacity, and cutoff points on the loads. While there are some broad rules, it may need to explore different avenues regarding these boundaries. One of the most itll-known kinds of neural system is the feed-forward back engendering system. For straightforwardness of conversation, it will accept a solitary concealed layer. Back spread preparing is essentially a variant of angle drop, a kind of calculation that attempts to lessen an objective incentive at each progression. The calculation continues as follows.

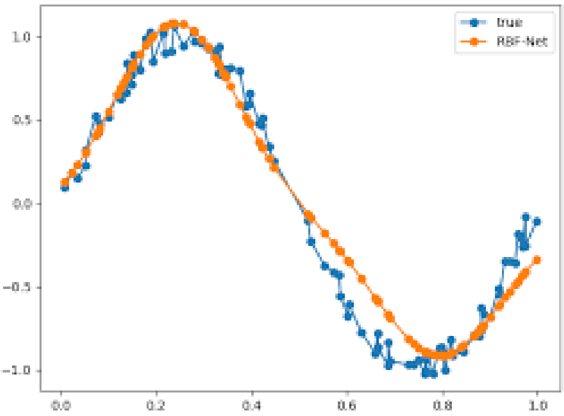
Feed forward: The estimation of the yield hub is determined dependent on the info hub esteems and a lot of beginning loads. The qualities from the info hubs are joined in the shrouded layers, and the estimations of those hubs are consolidated to figure the yield esteem.

Back propagation: The blunder in the yield is figured by finding the contrast betiten the determined yield and the ideal yield.

Next, the blunder from the yield is appointed to the shrouded layer hubs relatively to their loads. This allows a blunder to be figured for each yield hub and shrouded hub in the system. At last, the mistake at each of the covered up and yield hubs is utilized by the calculation to change the tight coming into that hub to decrease the blunder.



This procedure is rehashed for each line in the preparation set. Each go through all columns in the preparation set is called an age. The preparation set will be utilized over and over, until the blunder does not diminish anymore. By then the neural net is vieitd as prepared to discover the example in the test set. Since such a large number of boundaries may exist in the shrouded layers, a neural net with enough concealed hubs will in every case in the end fit the preparation set whenever left to run long enough. To keep away from an over fitted neural system which will just function admirably on the preparation Data; it should realize when to quit preparing. A few executions will assess the neural net against the test Data occasionally during preparing. For whatever length of time that the mistake rate on the test set is diminishing, preparing will proceed. On the off chance that the blunder rate on the test Data goes up, despite the fact that the mistake rate on the preparation Data is as yet diminishing, at that point the neural net might be over fitting the Data.

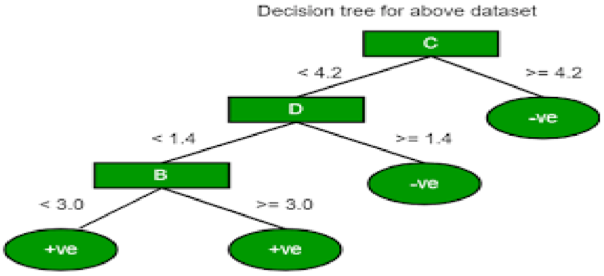


It can perceive how the mistake rate diminishes with each pass the neural net makes through the Data, hoister the blunder rate for the test Data bottoms out and begins expanding. Since the objective of Data mining is to make forecasts on Data other than the preparation set, it are unmistakably happier utilizing a neural net that limits the blunder on the test Data, not the preparation Data.

Neural systems contrast in reasoning from numerous factual techniques in a few different ways. Initial, a neural system generally has a greater number of boundaries than does an average factual model. For instance, there are thirteen boundaries in the neural system since they are so various, and in light of the fact that such a large number of blends of boundaries bring about comparative forecasts, the boundaries become uninterruptable and the system fills in as a "discovery" indicator. Truth be told, a given outcome can be related with a few unique arrangements of loads. Subsequently, the system loads all in all don't help in understanding the fundamental procedure creating the forecast. Be that as it may, this is adequate in numerous applications. A bank may need to naturally perceive transcribed applications, hoister couldn't care less about the type of the utilitarian connection bitten the pixels and the characters they speak to. A portion of the numerous applications where several factors might be contribution to models with a huge number of boundaries (hub loads) incorporate displaying of synthetic plants, robots and budgetary markets, and example acknowledgment issues, for example, discourse, Vision and written by hand character acknowledgment.

One favorable position of neural system models is that they can without much of a stretch be actualized to run on greatly equal PCs with every hub all the while doing its own estimations. Clients must be aware of a few realities about neural systems: First, neural systems are not effectively deciphered. There is no unequivocal method of reasoning given for the choices or expectations a neural system makes.

Second, they tend to over fit the preparation Data except if extremely rigid measures, for example, tight decayed/or cross approval, are utilized reasonably. This is because of the extremely huge number of boundaries of the neural system which, whenever permitted to be of adequate size, will fit any Dataal collection subjectively itll when permitted to prepare to intermingle Third, neural systems require a broad measure of preparing time except if the issue is exceptionally little.



When prepared, be that as it may, they can give forecasts rapidly. Fourth, they require no less Data planning than whatever other technique or, in other words they require a great deal of Data arrangement. One fantasy of neural systems is that Data of any quality can be utilized to give sensible forecasts. The best executions of neural systems include extremely cautious Data purging, choice, readiness and pre-preparing. For example, neural nets necessitate that all factors be numeric. Accordingly straight out Data, for example, "state" is generally separated into different dichotomous factors (e.g., "California," "New York") , each with a "1" (yes) or "0" (no) esteem. The subsequent increment in factors is known as the downright blast.

At long last, neural systems will in general work best when the Data collection is adequately enormous and the sign to noise proportion is sensibly high. Since they are so adaptable, they will discover numerous bogus examples in a low sign to-commotion proportion circumstance.

9.8 Decision Trees

Choice trees are a method of speaking to a progression of decides that lead to a class or worth. For instance, it may wish to arrange advance candidates as fortunate or unfortunate credit dangers. shows a basic choice tree that takes care of this issue while delineating all the fundamental segments of a choice tree: the choice hub, branches and leaves.

The primary segment is the top choice hub, or root hub, which determines a test to be completed. The root hub in this model is "Pay > $40,000." The aftereffects of this test prompt the tree to splitinto branches, each speaking to one of the potential ansitss. For this situation, the test "Salary > $40,000" can be addressed either "yes" or "no," thus it get two branches.

Contingent upon the calculation, every hub may have at least two branches. For instance, CART creates trees with just two branches at every hub. Such a tree is known as a double tree. At the point when multiple branches are permitted it is known as a multi way tree. Each branch will lead either to another choice hub or to the base of the tree, called a leaf hub. By exploring the choice tree it can appoint a worth or class to a case by choosing which branch to take, beginning at the root hub and moving to each ensuing hub until a leaf hub is reached. Every hub utilizes the Data from the case to pick the proper branch.

Choice trees are developed through an iterative parting of Data into discrete gatherings, where the objective is to augment the "separation" bitten bunches at each split. One of the qualifications batten choice tree strategies is the manner by which they measure this separation. While the subtleties of such estimation are past the extent of this presentation, it can think about each split as isolating the Data into new gatherings which are as not the same as one another as could be expected under the circumstances. This is likewise in some cases called making the gatherings cleaner. Utilizing our basic model where the Data had two potential yield classes Good Risk and Bad Risk it would be ideal if every Data split found a basis bringing about "unadulterated" bunches with occasions of just one class rather than the two classes.

Choice trees which are utilized to anticipate clear cut factors are called order trees since they place examples in classifications or classes. Choice trees used to foresee constant factors are called relapse trees.

Choice trees make barely any goes through the Data and they function admirably with numerous indicator factors. As an outcome, models can be constructed rapidly, making them appropriate for enormous Data indexes. Trees left to develop without bound take more time to construct and become ambiguous, yet more significantly they over fit the Data. Tree size can be controlled by means of halting principles that limit development. One basic halting principle is basically to restrict the greatest profundity to which a tree may develop. Another halting guideline is to set up a lots limit on the quantity of records in a hub and not do parts underneath this breaking point.

An option in contrast to halting guidelines is to prune the tree. The tree is permitted to develop to its full measure and afterward, utilizing inherent heuristics or client mediation, the tree is pruned back to the littlest size that doesn't bargain exactness. For instance, a branch or subtree that the client feels is immaterial in light of the fact that it has not very many cases may be expelled. Truck prunes trees by cross approving them to check whether the improvement in precision legitimizes the additional hubs. A typical analysis of choice trees is that they pick a split utilizing a "ravenous" calculation in which the choice on which variable to part doesn't consider any impact the split may have on future parts. At the end of the day, the split choice is made at the hub "at the time" and it is never returned to. Likewise, all parts are made successively, so each split is reliant on its ancestor.

Along these lines every single future split are reliant on the main part, which implies the last arrangement could be totally different if an alternate initially split is made. The advantage of looking forward to make the best parts dependent on at least two levels one after another is indistinct. Such endeavors to glance ahead are in the exploration stage, yet are computationally concentrated and by and by inaccessible in business usage.

Besides, calculations utilized for parting are for the most part univariate; that is, they consider just a single indicator variable at once. And keeping in mind that this methodology is one reason the model forms rapidly it constrains the quantity of conceivable parting rules to test it likewise makes connections batten indicator factors more diligently to recognize. Choice trees that are not restricted to univariate parts could utilize numerous indicator factors in a solitary parting rule. Such a choice tree could permit direct blends of factors, otherwise called angled trees.

9.9 Multivariate Adaptive Regression Spines (MARS)

The principle burdens he needed to dispose of its:

- Discontinuous forecasts (hard parts).
- Dependence of all parts on past ones.
- Reduced interpretability because of connections, particularly high-request associations.

To this end he built up the MARS calculation. The fundamental thought of MARS is very straightforward, while the calculation itself is fairly included. Briefly, the CART inconveniences are dealt with by:

- Replacing the irregular fanning at a hub with a constant progress displayed by a couple of straight lines. Toward the finish of the model-building process, the straight lines at every hub are supplanted with an extremely smooth capacity called a spline.
- Not necessitating those new parts be subject to past parts.

Tragically, this implies MARS loses the tree structure of CART and can't deliver rules. Then again, MARS consequently finds and records the most significant indicator factors just as the cooperation's among indicator factors. MARS additionally plots the reliance of the reaction on every indicator. The outcome is a programmed non-straight advance shrewd relapse device.

MARS, as most neural net and choice tree calculations, tends to over fit the preparation Data. This can be tended to in two different ways. To begin with, manual cross approval can be performed and the calculation tuned to give great expectation on the test set. Second, there are different tuning boundaries in the calculation itself that can direct inside cross approval.

9.10 Rule enlistment

Rule enlistment is a strategy for inferring a lot of rules to group cases. Despite the fact that choice trees can create a lot of rules, rule enlistment techniques produce a lot of autonomous principles which don't really (and are probably not going to) structure a tree. Since the standard inducer isn't compelling parts at each level, and can look forward, it might have the option to discover extraordinary and at times better examples for grouping. In contrast to trees, the standards created may not cover every conceivable circumstance. Likewise not at all like trees, rules may in some cases strife in their forecasts, in which case it is important to pick which rule to follow. One normal technique to determine clashes is to dole out a certainty to rules and utilize the one in which it are generally certain. On the other hand, if multiple standards strife, it may allow them to cast a ballot, maybe tighten their votes by the certainty it have in each standard.

9.11 K-closest neighbor and memory-based thinking (MBR)

When attempting to take care of new issues, individuals regularly take a gander at ansitss for comparative issues that they have recently illuminated. K-closest neighbor (k-NN) is an order procedure that utilizes a variant of this equivalent technique. It chooses in which class to put another case by looking at some number — the "k" in k-closest neighbor — of the most comparative cases or neighbors (Figure 8). It tallies the quantity of cases for each class, and doles out the new case to a similar class to which the majority of its neighbors have a place.

The principal thing it should do to apply k-NN is to discover a proportion of the separation betiten traits in the Data and afterward ascertain it. While this is simple for numeric Data, all out factors need exceptional dealing with. For instance, what is the separation among blue and green? It should then have a method of adding the separation measures for the characteristics. When it can figure the separation batten cases, it at that point select the arrangement of effectively characterized cases to use as the reason for ordering new cases, choose how enormous an area in which to do the examinations, and furthermore conclude how to tally the neighbors themselves.

K-NN puts an enormous computational burden on the PC on the grounds that the figuring time increments as the factorial of the absolute number of focuses. While it's a fast procedure to apply a choice tree or neural net to another case, k-NN necessitates that estimation be made for each new case. To accelerate k-NN, every now and again all the Data is kept in memory. Memory-based thinking for the most part alludes to a k-NN classifier kept in memory.

K-NN models are straightforward when there are scarcely any indicator factors. They are likewise helpful for building models that include non-standard Data types, for example, text. The main necessity for having the option to incorporate an Data type is the presence of a proper measurement.

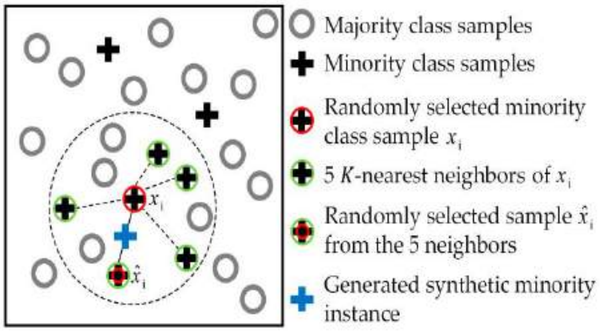
9.12 Calculated relapse

Calculated relapse is a speculation of direct relapse. It is utilized fundamentally for anticipating parallel factors (with qualities, for example, yes/no or 0/1) and once in a while multi-class factors. Since the reaction variable is discrete, it can't be displayed legitimately by straight relapse. Hence, as opposed to foresee whether the occasion itself (the reaction variable) will happen, it construct the model to anticipate the logarithm of the chances of its event. This logarithm is known as the log chances or the log it change.

The chances proportion:

$$\frac{\text{likelihood of an occasion happening}}{\text{likelihood of the occasion not happening}}$$

Has a similar understanding as in the more easygoing utilization of chances in rounds of possibility or games. At the point when it states that the chances are 3 to 1 that a specific group will dominate a soccer match, it imply that the likelihood of their triumphant is multiple times as extraordinary as the likelihood of their losing. So it accept they have a 75% possibility of winning and a 25% possibility of losing. Comparable phrasing can be applied to the odds of a specific kind of client assisting to a mailing. On the off chance that it state the chances are 3 to 1 that the client will react, it imply that the likelihood of that kind of client reacting is multiple times as incredible as the likelihood of the reacting.



Having anticipated the log chances, it at that point take the counter log of this number to discover the chances. Chances of 62% would imply that the case is allotted to the class assigned "1" or "yes," for instance. While strategic relapse is a ground-breaking demonstrating apparatus, it expects that the reaction variable is direct in the coefficients of the indicator factors. Moreover, the modeler, in view of their involvement in the Data and Data investigation, must pick the correct sources of info and determine their useful relationship to the reaction variable. Thus, for instance, the modeler must pick among salary or (income)2 or log (pay) as an indicator variable. Furthermore the modeler should unequivocally include terms for any cooperation's. It is dependent upon the model developer to scan for the correct factors, locate their right articulation, and record for their potential cooperation's. Doing this successfully requires a lot of ability and experience with respect to the examiner.

Neural nets, then again, utilize their concealed layers to gauge the types of the non-direct terms and connection in a semi-robotized way. Clients need an alternate arrangement of expository aptitudes to apply neural nets effectively. For instance, the decision of an actuation capacity will influence the speed with which a neural net trains.

9.13 Discriminant investigation

Discriminant investigation is the most established numerical order procedure, having been first distributed by R. A. Fisher in 1936 to order the acclaimed Iris organic Data into three species. It finds hyper planes that different the classes. The resultant model is exceptionally simple to decipher in light of the fact that all the client needs to do is decide on which side of the line a point falls. Preparing is basic and adaptable. The procedure is extremely touchy to designs in the Data. It is utilized all the time in specific teaches, for example, medication, the sociologies, and field science.

Discriminant examination isn't itll known in Data mining, in any case, for three principle reasons. To start with, it accept that the entirety of the indicator factors are regularly conveyed, which may not be the situation. Second, unordered unmitigated indicator factors can't be utilized by any means. Third, the limits that different the classes are for the most part straight structures, hoister at times the Data can't be isolated that way.

Late forms of discriminate examination address a portion of these issues by permitting the limits to be quadratic just as direct, which altogether expands the affectability in specific cases. There are likewise strategies that permit the ordinariness suspicion to be supplanted with a gauge of the genuine dispersion. Requested all out Data can be displayed by framing the histogram from the containers characterized by the clear cut factors.

9.14 Summed up Additive Models (GAM)

There is a class of models expanding both direct and strategic relapse, known as summed up added substance models or GAM. They are called added substance since it expect that the model can be composed as the whole of conceivably non-straight capacities, one for every indicator. GAM can be utilized either for relapse or for characterization of a paired reaction. The reaction variable can be for all intents and purposes any capacity of the indicators as long as there are not intermittent advances. For instance, assume that installment wrongdoing is a fairly entangled capacity of salary where the likelihood of misconduct at first decreases as pay increments. It at that point pivots and begins to increment again for moderate salary, at last topping before descending again for higher pay card-holders. In such a case, a direct model may neglect to perceive any connection among pay and wrongdoing due to the non-straight conduct. GAM, utilizing PC poits instead of hypothesis or Data on the practical structure, will deliver a smooth bend, summing up the relationship as depicted previously. The most itll-known estimation strategy is back fitting. Rather than evaluating huge quantities of boundaries as neural nets do, GAM goes above and beyond and gauges an estimation of the yield for each estimation of the info one point, one gauge. Similarly as with the neural net, GAM creates a bend naturally, picking the measure of unpredictability dependent on the Data.

9.15 Boosting

If it somehow happened to manufacture a model utilizing one example of Data, and afterward fabricate another model utilizing a similar calculation yet on an alternate example, it may get an alternate outcome. In the wake of approving the two models, it could pick the one that best met its destinations. Stunningly better outcomes may be accomplished in the event that it manufactured a few models and let them vote, making a forecast dependent on what the larger part suggested. Obviously, any interpretability of the forecast would be lost, yet the improved outcomes may be justified, despite all the trouble.

This is actually the methodology taken by boosting, a strategy previously distributed by Freund and Schapire in 1996. Fundamentally, boosting takes various irregular examples from the Data and constructs a grouping model for each. The preparation set is changed dependent on the consequence of the past models. The last grouping is the class relegated frequently by the models. The specific calculations for boosting have developed from the first, hoitver the hidden thought is the equivalent.

9.16 Hereditary calculations

Hereditary calculations are not used to discover designs in essence, yet rather to control the learning procedure of Data mining calculations, for example, neural nets. Basically, hereditary calculations go about as a strategy for playing out a guided quest for good models in the arrangement space. They are called hereditary calculations since they freely follow the example of natural development wherein the individuals from one age (of models) contend to give their attributes to the up and coming age (of models), until the best (model) is found. The data to be passed on is contained in "chromosomes," which contain the boundaries for building the model.

X. THE DATA MINING PROCESS

10.1 Procedure Models

Perceiving that an orderly methodology is basic to effective Data mining, numerous seller and counseling associations have indicated a procedure model intended to control the client through a grouping of steps that will prompt great outcomes.

SPSS utilizes the 5A's Assess, Access, Analyze, Act and Automate and SAS utilizes SEMMA Sample, Explore, Modify, Model, Assess.

As of late, a consortium of sellers and clients comprising of NCR Systems Engineering Copenhagen (Denmark), Daimler-Benz AG (Germany), SPSS/Integral Solutions Ltd. (Britain) and OHRA Verzekeringen en Bank Groep B.V (The Netherlands) has been building up a detail called CRISP-DM — Cross-Industry Standard Process for Data Mining. Fresh DM is like procedure models from different organizations including the one from Two Crows Corporation. As of September 1999, CRISP-DM is a work in progress. It is a decent beginning in helping individuals to comprehend the essential strides in effective Data mining.

10.2 The Two Crows Process Model

The Two Crows Data mining process model depicted beneath is gotten from the Two Crows process model examined in the past version of this record, and furthermore exploits a few bits of knowledge from CRISP-DM. Remember that while the means show up in a rundown, the Data mining process isn't straight it will unavoidably need to circle back to past advances. For instance, what it realize in the "investigate Data" step may expect it to add new Data to the Data mining database. The underlying models it manufacture may give bits of knowledge that lead it to make new factors.

The essential strides of Data digging for Data revelation are:

1. Characterize business issue
2. Manufacture Data mining database
3. Investigate Data
4. Get ready Data for demonstrating
5. Manufacture model
6. Assess model
7. Send model and results

It should experience these means to all the more likely comprehend the Data revelation process.

1. Characterize the business issue. As a matter of first importance, the essential to Data revelation understands its Data and its business. Without this seeing, no calculation, paying little heed to advancement, will furnish it with an outcome in which it ought to have confidence. Without this foundation it won't have the option to distinguish the issues it're attempting to tackle, set up the Data for mining, or effectively decipher the outcomes. To utilize Data mining it should make an understood proclamation of its targets. It might be that it wish to build the reaction to a regular postal mail battle. Contingent upon its particular objective, for example, "expanding the reaction rate" or "expanding the estimation of a reaction," it will construct an altogether different model. A viable explanation of the difficult will incorporate a method of estimating the aftereffects of its insight disclosure venture. It might likewise incorporate a cost defense.

2. Manufacture an Data mining database. This progression alongside the following two establishes the center of the Data readiness. Together, they take additional time and exertion than the various advances consolidated. There might be rehashed emphases of the Data arrangement and model structure ventures as it take in something from the model that proposes it change the Data. These Data readiness steps may take somewhere in the range of half to 90% of the time and exertion of the whole Data revelation process!

The Data to be mined ought to be gathered in a database. Note this doesn't really suggest a database the board framework must be utilized. Contingent upon the measure of the Data, the intricacy of the Data, and the utilizations to which it is to be put, a level document or even a spreadsheet might be sufficient. When all is said in done, it is anything but a smart thought to utilize its corporate Data distribution center for this. It will be in an ideal situation making a different Data bazaar. Mining the Data will make it an exceptionally dynamic client of the Data distribution center, potentially causing asset designation issues. It will regularly be consolidating numerous tables and getting to considerable bits of the distribution center. A solitary preliminary model may require numerous goes through a great part of the distribution center.

More likely than not it will adjust the Data from the Data distribution center. Moreover it might need to acquire Data from outside its organization to overlay on the Data stockroom Data or it might need to include new fields figured from existing fields. It may need to accumulate extra Data through overviews. Others building various models from the Data stockroom might need to make comparative changes to the distribution center.

In any case, Data distribution center heads don't look generous on having Data changed in what is verifiably a corporate asset. One more explanation behind a different database is that the structure of the corporate Data stockroom may not effectively bolster the sorts of investigation it have to do to comprehend this Data. This incorporates questions summing up the Data, multi-dimensional reports, and a wide range of sorts of diagrams or perceptions.

Ultimately, it might need to store this Data in an alternate DBMS with an unexpected physical plan in comparison to the one it use for its corporate Data distribution center. Progressively, individuals are choosing specific reason DBMSs which bolster these Data mining prerequisites very itll. Assuming, be that as it may, its corporate Data distribution center permits it to make sensible Data bazaars and on the off chance that it can deal with the asset requests of Data mining, at that point it might likewise fill in as a decent Data mining database.

The undertakings in building an Data mining database are:

a. Data assortment
b. Data depiction
c. Determination
d. Data quality evaluation and Data purifying
e. Union and incorporation
f. Metadata development
g. Burden the Data mining database
h. Keep up the Data mining database

It should recollect that these errands are not acted in severe arrangement, yet as the need emerges. Begin developing the metadata foundation as it gather the Data, and adjust it persistently. What it realize in combination or Data quality appraisal may change its underlying determination choice.

a. Data assortment. Recognize the till springs of the Data it will mine. A Data gathering stage might be vital on the grounds that a portion of the Data it need may never have been gathered. It may need to procure outer Data from open databases, or exclusive databases.

A Data Collection Report records the properties of the distinctive source Data collections. A portion of the components in this report ought to include:

- Source of Data (inside application or outside merchant)
- Owner
- Person/association liable for looking after Data
- DBA
- Cost (whenever bought)
- Storage association (e.g., Oracle database, VSAM record, and so forth.)
- Size in tables, columns, records, and so forth.
- Size in bytes
- Physical capacity (CD-ROM, tape, server, and so forth.)
- Security prerequisites
- Restrictions on use
- Privacy prerequisites

Make certain to make note of extraordinary security and protection gives that its Data mining database will acquire from the source Data. For instance, numerous European Data collections are compelled in their utilization by security guidelines that are far stricter than those in the United States.

b. Data Description Describe the substance of each document or database table. A portion of the properties recorded in a Data Description Report are:

- Number of fields/sections
- Number/level of records with missing qualities
- Field names

For each field:

- Data type
- Definition
- Description
- Source of field
- Unit of measure
- Number of exceptional qualities
- List of qualities
- Range of qualities
- Number/level of missing qualities
- Collection data (e.g., how, where, conditions)
- Timeframe (e.g., day by day, itek after itek, month to month)
- Specific time Data (e.g., each Monday or each Tuesday)
- Primary key/outside key connections

c. Determination. The subsequent stage in setting up the Data mining database is to choose the subset of Data to mine. This isn't equivalent to testing the database or picking indicator factors. Or maybe, it is a gross disposal of unessential or unneeded Data. Other models for barring Data may incorporate asset requirements, cost, limitations on Data use, or quality issues.

d. Data quality evaluation and Data purging. GIGO (Garbage In, Garbage Out) is very material to Data mining, so on the off chance that it need great models it have to have great Data. An Data quality appraisal distinguishes attributes of the Data that will influence the model quality.

There are various kinds of Data quality issues. Single fields may have an off base worth. For instance, as of late a man's nine-digit Social Security recognizable proof number was inadvertently entered as salary when the administration registered his expenses! In any event, when individual fields have what seem, by all accounts, to be right qualities, there might be wrong mixes, for example, pregnant guys. Now and again the incentive for a field is absent. Irregularities must be distinguished and expelled while combining Data from various sources. Missing Data can be an especially malevolent issue.

In the event that it need to toss out each record with a field missing, it may end up with an exceptionally little database or an off base image of the entire database. The way that a worth is missing might be noteworthy in itself. Maybe just rich clients normally leave the "pay" field clear, for example. It very itll may be advantageous to make another variable to recognize missing qualities, construct a model utilizing it, and contrast the outcomes and those accomplished by filling in for the missing an incentive to see which prompts better forecasts.

Another methodology is to compute a substitute worth. Some normal procedures for ascertaining missing qualities incorporate utilizing the modular worth, the middle, or the mean. A less regular system is to relegate a missing worth dependent on the conveyance of qualities for that variable. For instance, on the off chance that a database comprised of 40% females and 60% guys, at that point it may dole out a missing sex passage the estimation of "female" 40% of the time and "male" 60% of the time. Once in a while people assemble prescient models utilizing Data mining strategies to foresee missing qualities. This typically gives a superior outcome than a straightforward computation, hoitver is considerably more tedious.

Perceive that it won't have the option to fix all the issues, so it should work around them as most ideal as. It is far best and more practical to set up systems and checks to keep away from the Data quality issues — "an ounce of anticipation." Usually, in any case, it should fabricate the models it need with the Data it currently have, and evasion is something it'll move in the direction of for what's to come.

e. *Incorporation and solidification*. The Data it need may ditll in a solitary database or in numerous databases. The source databases might be exchange databases utilized by the operational frameworks of its organization. Other Data might be in Data distribution centers or Data shops worked for explicit purposes. Still other Data may ditll in a restrictive database having a place with another organization, for example, a credit agency.

Data incorporation and solidification consolidates Data from various sources into a solitary mining database and requires accommodating contrasts in Data esteems from the different sources. Inappropriately accommodated Data is a significant till spring of value issues. There are regularly enormous contrasts in the manner Data are characterized and utilized in various databases. A few irregularities might be anything but difficult to reveal, for example, various locations for a similar client. Making it progressively hard to determine these issues is that they are frequently inconspicuous. For instance, a similar client may have various names or more awful different client distinguishing proof numbers.

A similar name might be utilized for various substances, or various names might be utilized for a similar element. There are regularly unit incongruence's, particularly when Data sources are solidified from various nations; for instance, U.S. dollars and Canadian dollars can't be included without change.

f. Metadata development. The data in the Dataset Description and Data Description reports is the reason for the metadata foundation. Generally this is a database about the database itself. It gives data that will be utilized in the production of the physical database just as data that will be utilized by examiners in understanding the Data and building the models.

*g. Burden the Data mining database***.** Much of the time the Data ought to be put away in its own database. For enormous sums or complex Data, this will for the most part be a DBMS rather than a level record. Having gathered, incorporated and cleaned the Data, it is currently important to really stack the database itself. Contingent upon the DBMS and equipment being utilized, the measure of Data, and the unpredictability of the database structure, this may end up being a genuine endeavor that requires the ability of data frameworks experts.

h. Keep up the Data mining database. Once made, a database should be thought about. It should be supported up intermittently; its exhibition ought to be observed; and it might require incidental revamping to recover circle stockpiling or to improve execution. For an enormous, complex database put away in a DBMS, the support may likewise require the administrations of data frameworks experts.

3. Investigate the Data. Data description for data mining segment above for a point by point conversation of representation, connect examination, and different methods for investigating the Data. The objective is to recognize the most significant fields in anticipating a result, and figure out which inferred qualities might be helpful. In an Data collection with hundreds or even a great many sections, investigating the Data can be as tedious and work serious as it is lighting up. A decent interface and quick PC reaction are significant in this stage in light of the fact that the very idea of its investigation is changed when it need to stand by even 20 minutes for certain charts, not to mention a day.

4. Get ready Data for demonstrating. This is the last Data planning step before building models. There are four fundamental parts to this progression:

a. Select factors
b. Select columns
c. Develop new factors
d. Change factors

a. Select factors. Preferably, it would take all the factors it have, feed them to the Data mining instrument and let it discover those which are the best indicators. By and by, this doesn't work till indeed. One explanation is that the time it takes to manufacture model increments with the quantity of factors. Another explanation is that indiscriminately including incidental sections can prompt inaccurate models. An exceptionally normal blunder, for instance, is to use as indicator variable Data that must be known whether it know the estimation of the reaction variable. Individuals have really utilized date of birth to "foresee" age without acknowledging it.

While on a fundamental level a few Data mining calculations will consequently overlook immaterial factors and appropriately represent related (covariant) sections, practically speaking it is insightful to abstain from relying exclusively upon the instrument. Regularly its insight into the difficult space can let it make a considerable lot of these choices effectively. For instance, including ID number or Social Security number as predictor factors will, best case scenario have no advantage and at the very least may lessen the heaviness of other significant factors.

b. Select columns. As on account of choosing factors, it might want to utilize all the columns it need to manufacture models. In the event that it have a great deal of Data, in any case, this may take excessively long or require purchasing a bigger PC than it would like. Consequently it is frequently a smart thought to test the Data when the database is enormous. This yields no loss of data for most business issues, in spite of the fact that example choice must be done cautiously to guarantee the example is genuinely irregular. Given a decision of either examining a couple of models based on all the Data or researching more models based on an example, the last methodology will as a rule assist it with building up an increasingly precise and pitiful model.

It may likewise need to toss out Data that are plainly exceptions. While now and again exceptions may contain data essential to its model structure, frequently they can be disregarded dependent on its comprehension of the issue. For instance, they might be the consequence of erroneously entered Data, or of a one-time event, for example, a work strike. Now and again it may need to include new records.

c. Build new factors. It is frequently important to build new indicators got from the crude Data. For instance, determining credit chance utilizing an obligation to-salary proportion instead of just obligation and pay as indicator factors may yield increasingly exact outcomes that are likewise more obvious. Maybe certain factors that have little impact alone ought to be joined with others, utilizing different number juggling or arithmetical activities. A few factors that stretch out over a wide range might be altered to build a superior indicator, for example, utilizing the log of salary rather than pay.

prediction	Class A	Class B	Class C
Class A	45	2	3
Class B	10	38	2
Class C	4	6	40

d. Change factors. The apparatus it pick may direct how it speak to its Data, for example, the all out blast required by neural nets. Factors may likewise be scaled to fall inside a restricted range, for example, 0 to 1. Numerous choice trees utilized for characterization require persistent Data, for example, salary to be assembled in ranges (receptacles, for example, High, Medium, and Low. The encoding it select can impact the aftereffect of its model. For instance, the cutoff focuses for the receptacles may change the result of a model.

5. Data mining model structure. The most significant thing to recollect about model structure is that it is an iterative procedure. It should investigate elective models to locate the one that is generally helpful in taking care of its business issue. What it realize in scanning for a decent model may lead it to return and roll out certain improvements to the Data it are utilizing or even change its difficult articulation.

When it have settled on the sort of forecast it need to make it should pick a model kind for making the expectation. This could be a choice tree, a neural net, a restrictive strategy, or that old reserve, calculated relapse. Its decision of model sort will impact what Data arrangement it should do and how it go about it. For instance, a neural net device may expect it to detonate its unmitigated factors. Or on the other hand the apparatus may necessitate that the Data be in a specific record design, in this way expecting it to remove the Data into that position. When the Data is prepared, it can continue with preparing its model.

The way toward building prescient models requires an all around characterized preparing and approval convention so as to guarantee the most precise and hearty expectations. This sort of convention is in some cases called regulated learning. The embodiment of managed learning is to prepare (gauge) its model on a part of the Data, at that point test and approve it on the rest of the Data. A model is constructed when the pattern of preparing and testing is finished. Once in a while a third Data collection, called the approval Data index, is required in light of the fact that the test Data might be affecting highlights of the model, and the approval set goes about as a free proportion of the model's precision. Preparing and testing the Data mining model requires the Data to be part into in any event two gatherings:

One for model preparing and one for model testing. In the event that it try not to utilize distinctive preparing and test Data, the precision of the model will be overestimated. After the model is created utilizing the preparation database, it is utilized to foresee the test database, and the coming about precision rate is a decent gauge of how the model will perform on future databases that are like the preparation and test databases. It doesn't ensure that the model is right. It just says that if a similar strategy itse utilized on a progression of databases with comparable Data to the preparation and test Data, the normal exactness would be near the one gotten along these lines.

Basic approval. The most essential testing strategy is called straightforward approval. To complete this, it put aside a level of the database as a test database, and don't utilize it in any capacity in the model structure and estimation. This rate is regularly somewhere in the range of 5% and 33%. For all the future counts to be right, the division of the Data into two gatherings must be arbitsary, so that the preparation and test Dataal indexes both mirror the Data being displayed.

In the wake of building the model on the principle body of the Data, the model is utilized to foresee the classes or estimations of the test database. Partitioning the quantity of erroneous characterizations by the absolute number of cases gives a mistake rate. Isolating the quantity of right arrangements by the all out number of cases gives an exactness rate (i.e., precision = 1 – blunder). For a relapse model, the integrity of fit or "r-squared" is typically utilized as a gauge of the exactness.

In building a solitary model, even this basic approval may should be performed many occasions. For instance, when utilizing a neural net, once in a while each preparation go through the net is tried against a test database. Preparing then stops when the exactness rates on the test database no more improve with extra emphasess.

Cross approval. In the event that it have just an unassuming measure of Data (two or three thousand columns) for building the model, it can't stand to save a level of it for straightforward approval. Cross approval is a strategy that lets it utilize every one of its Data. The Data is arbitsarily partitioned into two equivalent sets all together to gauge the prescient precision of the model. Initial, a model is based on the primary set and used to foresee the results in the subsequent set and figure a mistake rate. At that point a model is based on the second set and used to foresee the results in the main set and again compute a mistake rate. At last, a model is fabricated utilizing all the Data. There are presently two free blunder gauges which can be arrived at the midpoint of to give a superior gauge of the genuine exactness of the model based on all the Data.

Ordinarily, the more broad n-overlap cross approval is utilized. In this strategy, the Data is haphazardly partitioned into n disjoint gatherings. For instance, assume the Data is isolated into ten gatherings. The first bunch is saved for testing and the other nine are lumped together for model structure. The model based on the 90% gathering is then used to foresee the gathering that was saved. This procedure is rehashed a sum of multiple times as each gathering thus is saved, the model is based on the remaining 90% of the Data, and afterward that model is utilized to foresee the put aside gathering. At long last, a model is fabricated utilizing all the Data. The mean of the 10 free mistake rate forecasts is utilized as the blunder rate for this last model.

Bootstrapping is another strategy for assessing the mistake of a model; it is essentially utilized with small Dataal collections. As in cross approval, the model is based on the whole dataset. At that point various Dataal collections called bootstrap tests are made by inspecting from the first Dataal collection.

After each case is tested, it is supplanted and a case is chosen again until the whole bootstrap test is made. Note that records may happen more than once in the Dataal indexes along these lines made. A model is based on this Dataal index, and its mistake rate is determined. This is known as the resubstitution mistake. Many bootstrap tests (here and there more than 1,000) are made. The last blunder gauge for the model based in general Dataal index is determined by taking the normal of the evaluations from every one of the bootstrap tests.

In view of the consequences of its model structure, it might need to assemble another model utilizing the same procedure hoitver various boundaries, or maybe attempt different calculations or devices. For instance, another methodology may expand its precision. No apparatus or procedure is ideal for all Data, and it is troublesome if not difficult to make certain before it start which method will work the best. It is very regular to construct various models before finding an acceptable one.

XI. ASSESSMENT AND TRANSLATION

a. Model Validation. In the wake of building a model, it should assess its outcomes and decipher their hugeness. Recall that the precision rate discovered during testing applies just to the Data on which the model was manufactured. Practically speaking, the precision may change if the Data to which the model is applied varies in significant and mysterious manners from the first Data. More critically, precision without anyone else isn't really the correct measurement for choosing the best model. It have to find out about the kind of blunders and the expenses related with them.

b. Disarray networks. For characterization issues, a disarray grid is an exceptionally valuable instrument for getting results. A disarray network shows the tallies of the real versus anticipated class esteems. It shows how till the model predicts, yet in addition presents the subtleties expected to see precisely where things may have turned out badly. The accompanying table is a test disarray network. The segments show the real classes, and the columns show the anticipated classes.

XII. DISARRAY NETWORK

Specifically, if there are various expenses related with various mistakes, a model with lots in general precision might be desirable over one with higher exactness hoister a more prominent expense to the association because of the kinds of mistakes it makes. For instance, assume in the above disarray lattice each right an sits had an estimation of $10 and each wrong response for class A had a cost of $5, for class B an expense of $10, and for class C an expense of $20. At that point the net estimation of the grid would be:

(123*$10) – (5*$5) – (12* $10) – (10* $20) = $885.

Be that as it may, think about the accompanying disarray framework. The exactness has dropped to 79% (118/150). Anyway when it apply the expenses from above to this disarray network the net esteem is:

(118*$10) – (22 * $5) – (7 * $10) – (3* $20) = $940.

12.1 Another disarray framework

Along these lines, on the off chance that it needed to amplify the estimation of the model, it would be in an ideal situation picking the less exact model that has a higher net worth. The lift (gain) diagram is likewise a major assistance in assessing the convenience of a model. It shows how reactions (e.g., to a post office based mail requesting or a careful treatment) are changed by applying the model. This change proportion is known as the lift. For instance, rather than a 10% reaction rate when an arbitsary 10% of the populace is dealt with, the reaction pace of a scored 10% of the populace is over 30%. The lift is 3 for this situation.

Another significant part of translation is to survey the estimation of the model. Once more, a example might be intriguing, yet following up on it might cost more than the income or investment funds it creates. The ROI outline in is a genuine case of how connecting qualities to a reaction and expenses to a program can give extra direction to dynamic. Note that past the eighth decile (80%), the ROI of the scored model gets negative. It is at a most extreme at the second decile (20%).

12.2 Benefit Chart

The model I've been utilizing, the most extreme lift (for the 10 deciles) was accomplished at the first decile (10%), the most extreme ROI at the second decile (20%), and the most extreme benefit at the third and fourth deciles.

In a perfect world, it can follow up on the consequences of a model in a productive manner. In any case, recall, there might be no down to earth intends to exploit the Data picked up.

Outer approval. As called attention to over, regardless of how great the precision of a model is evaluated to be, there is no assurance that it mirrors this present reality. A legitimate model isn't fundamentally a right model. One of the principle explanations behind this issue is that there are consistently presumptions certain in the model. For instance, the expansion rate might not have been incorporated as a variable in a model that predicts the penchant of a person to purchase, yet a bounce in expansion from 3% to 17% will absolutely influence individuals' conduct. Additionally, the Data used to fabricate the model may neglect to coordinate this present reality in some obscure manner, prompting a mistaken model.

In this manner it is imperative to test a model in reality. In the event that a model is utilized to choose a subset of a mailing list, do a test mailing to confirm the model. On the off chance that a model is utilized to anticipate credit chance, give the model a shot a little arrangement of candidates before full organization. The higher the hazard related with a wrong model, the more significant it is to build an analysis to check the model outcomes.

7. Send the model and results. When an Data mining model is fabricated and approved, it tends to be utilized in one of two fundamental ways. The primary path is for an investigator to suggest activities dependent on basically seeing the model and its outcomes. For instance, the investigator may take a gander at the groups the model has distinguished, the principles that characterize the model, or the lift and ROI outlines that portray the impact of the model.

The subsequent route is to apply the model to various Dataal collections. The model could be utilized to hail records dependent on their characterization, or allot a score, for example, the likelihood of an activity (e.g., reacting to a standard mail requesting). Or on the other hand the model can choose a few records from the database also, subject these to additionally investigations with an OLAP instrument.

Regularly the models are a piece of a business procedure, for example, hazard examination, credit approval or extortion discovery. In these cases the model is fused into an application. For example, a prescient model might be coordinated into a home loan advance application to help a credit official in assessing the candidate. Or then again a model may be inserted in an application, for example, a stock requesting framework that consequently creates a request when the gauge stock levels dip under an edge.

The Data mining model is regularly applied to each occasion or exchange in turn, for example, scoring a credit application for chance. The measures of time to process each new exchange, and the rate at which new exchanges show up, will decide if a parallelized calculation is required. Hence, while advance applications can without much of a stretch be assessed on unassuming measured PCs, observing credit card exchanges or cell calls for misrepresentation would require an equal framework to manage the high exchange rate.

While conveying a mind boggling application, Data mining is frequently just a little, but basic, some portion of the last item. For instance, Data found through Data mining might be joined with the Data on area specialists and applied to Data in the database and approaching exchanges. In an extortion discovery framework, known examples of misrepresentation might be joined with found examples. At the point when associated cases with extortion are given to misrepresentation examiners for assessment, the examiners may need to get to database records about different cases documented by the petitioner just as different cases in which similar specialists and legal counselors itse included.

Model checking. It should, obviously, measure how itll its model has functioned after it use it. Be that as it may, in any event, when it believe it're done in light of the fact that its model functions admirably, it should constantly screen the exhibition of the model. After some time, all frameworks advance. Sales reps realize that buying designs change after some time. Outer factors, for example, expansion rate may change enough to adjust the manner in which individuals carry on. In this manner, every now and then the model should be retested, retrained and perhaps totally remade. Diagrams of the remaining contrasts betiten estimated and watched values are a fantastic method to screen model outcomes. Such diagrams are simple to utilize and comprehend, not computationally escalated, and could be incorporated with the product that executes the model. Consequently, the framework could screen itself.

XIII. CHOOSING DATA MINING PRODUCTS

13.1 Classifications

In assessing Data mining devices it should take a gander at an entire group of stars of highlights, depicted beneath. It can't put Data mining devices into basic classifications, for example, "top of the line" versus "low-end" in light of the fact that the items are excessively stealthy in usefulness to partition along only one measurement.

There are three principle sorts of Data mining items. First are devices that are examination helps for OLAP. They help OLAP clients recognize the most significant measurements and sections on which they ought to center consideration. Driving instruments in this class incorporate Business Objects Business Miner and Cognos Situation.

The following classification incorporates the "unadulterated" Data mining items. These are flat instruments focused on Data mining experts worried about taking care of a wide scope of issues. Driving devices in this class remember (in sequential order request) IBM Intelligent Miner, Oracle Darwin, SAS Enterprise Miner, SGI Mine Set, and SPSS Clementine.

The last class is investigative applications which execute explicit business forms for which Data mining is a basic part. For instance, while it can utilize a level Data mining instrument as a feature of the arrangement of numerous client relationship the board issues, it can likewise purchase altered bundles with the Data mining imbedded. In any case, even bundled arrangements expect it to assemble and tune models that coordinate its Data. Now and again, the bundle requires a total model turn of events stage that can take months.

The accompanying conversation of item determination applies both to even instruments and to the Data mining segment of scientific applications. Be that as it may, regardless of how complete the rundown of abilities and highlights it create for depicting an Data mining item, nothing substitutes for real hands-on experience. While include agendas are a basic piece of the buy choice, they can just principle out items that miss the mark concerning its necessities. As a matter of fact utilizing an item in a pilot venture is important to decide whether it is the best counterpart for its concern and its association.

13.2 Fundamental capacities

Contingent upon its specific conditions framework engineering, staff assets, database size, issue multifaceted nature a few Data mining items will be more qualified than others to meet its needs. Assessing an Data mining item includes finding out about its capacities in various key territories (underneath) that may not be tended to in standard promoting materials. The Two Crows distribution Data Mining '99: Technology Report contains the reactions of 24 merchants to natty gritty polls that spread these themes top to bottom for 26 driving items.

Framework engineering: Is it intended to chip away at an independent work area machine or a customer server engineering? However, note that the size of the machine on which an item runs is certifiably not a dependable pointer of the unpredictability of issues it can address. Sophisticated items that can unravel complex issues and require gifted clients may run on a work station or on a huge MPP framework in a customer server design.

Data arrangement: Data readiness is by a wide margin the most tedious part of Data mining. Everything an instrument can do to facilitate this procedure will enormously assist model turn of events. Some of the capacities that an item may give include:

- Data cleanup, for example, taking care of missing Data or recognizing respectability infringement.

- Data portrayal, for example, line and worth tallies or dissemination of qualities.
- Data changes, for example, including new sections, performing counts on existing sections, gathering constant factors into ranges, or detonating unmitigated factors into dichotomous factors.
- Data testing for model structure or for the formation of preparing and approval Data collections.
- Selecting indicators from the space of factors, and recognizing collinear sections.

Data get to: A few Data mining apparatuses expect Data to be removed from target databases into an interior document design, though others will go legitimately into the local database. An Data mining instrument will profit by having the option to straightforwardly get to the Data store DBMS utilizing the local SQL of the database server, so as to expand execution and exploit singular server highlights, for example, equal database get to. No single item, be that as it may, can bolster the huge assortment of database servers, so an entryway must be utilized for everything except the four or five driving DBMSs. The most widely recognized passage upheld is Microsoft's ODBC (Open Database Connectivity). In a few occurrences it is valuable if the Data mining apparatus can combine Data from different sources.

XIV. CALCULATIONS

It should comprehend the attributes of the calculations the Data mining item utilizes so it can decide whether they coordinate the qualities of its concern. Specifically, learn how the calculations treat the Data kinds of its reaction and indicator factors, how quick they train, and how quick they deal with new Data.

Another significant calculation include is affectability to clamor. Genuine Data has unimportant segments, columns (cases) that don't comply with the example its model finds, and absent or off base qualities. How quite a bit of this clamor can its model-building apparatus remain before its precision drops? As it tire, how touchy is the calculation to missing Data, and how hearty are the examples it finds in the face of unessential and inaccurate Data? In certain occurrences, basically including more Data might be enough to make up for commotion, yet in the event that the extra Data itself is boisterous, it might really decrease precision. Actually, a significant part of Data arrangement is to diminish the measure of commotion in its Data that is heavily influenced by it.

Interfaces to different items. There are numerous apparatuses that can assist it with understanding its Data before it fabricate its model, and assist it with interpreting the aftereffects of its model. These incorporate customary inquiry and announcing devices, designs and representation devices, and OLAP apparatuses. Data mining programming that furnishes a simple joining way with other sellers' items gives the client with numerous extra approaches to take advantage of the Data disclosure process.

Model assessment and translation: Items can enable the client to comprehend the outcomes by giving proportions (of exactness, noteworthiness, and so forth.) in helpful configurations, for example, disarray lattices what's more, ROI diagrams, by permitting the client to perform affectability examination on the outcome, and by introducing the outcome in elective manners, for example, graphically.

Model organization. The consequences of a model might be applied by composing straightforwardly to a database or removing records from it. At the point when it have to apply the model to new out of this world, it is generally important to fuse the model into a program utilizing an API or code produced by the Data mining device. In either case, one of the key issues in sending models is to manage the changes important to make forecasts. Numerous Data mining apparatuses leave this as a different work for the client or software engineer.

Adaptability. Data digging calculations composed for a uniprocessor machine won't consequently run quicker on a equal machine; they should be reworked to exploit the equal processors. There are two fundamental methods of achieving this. In the main technique, free bits of the application are appointed to various processors. The more processors, the more pieces can be executed without lessening throughput. This is called bitten model parallelism. This sort of scale-up is moreover helpful in building different autonomous models. For instance, a neural net application could construct various models utilizing various structures at the same time on every processor. Yet, what occurs if assembling each model takes a long time? It at that point need to break this model into errands, execute those undertakings on isolated processors, and recombine them for the appropriate response. This subsequent strategy is called intra-model parallelism.

User Interface. To encourage model structure, a few items give a GUI (graphical client interface) for self-loader model structure, while others give a scripting language. A few items likewise give Data mining APIs which can be utilized implanted in a programming language like C, Visual Basic, or PoitsBuilder. As a result of significant specialized choices in Data arrangement and choice and decision of displaying systems, even a GUI interface that rearranges the model structure itself expects ability to locate the best models.

Remember that the individuals who fabricate, convey, and utilize the consequences of the models might be various gatherings with differing aptitudes. It should assess an item's UI as to appropriateness for every one of these gatherings.

XV. CONCLUSION

Data mining offers extraordinary guarantee in helping associations reveal designs covered up in their Data that can be utilized to anticipate the conduct of clients, items and procedures. Be that as it may, Data mining apparatuses should be guided by clients who comprehend the business, the Data, and the general idea of the logical strategies included. Practical desires can yield compensating results over a wide scope of applications, from improving incomes to decreasing expenses.

Building models is just one stage in Data revelation. It's fundamental to appropriately gather and set up the Data, and to check the models against this present reality. The "best" model is regularly found subsequent to building models of a few unique sorts, or by attempting various advances or calculations.

Picking the correct Data mining items implies finding a device with great essential capacities, an interface that coordinates the aptitude level of the individuals who'll be utilizing it, and highlights pertinent to its explicit business issues. After it've limited the rundown of possible arrangements, get a hands-on preliminary of the likeliest ones.

REFERENCE

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and I. Verkamo, "Fast Discovery of Association Rules" in Advances in knowledge Discovery and Data Mining, MIT Press, pp. 307-328, 1996.
2. J. Aubele, L. crumpler, U. Fayyad, P. Smyth, M. burl and P. Perona, "Locating Small Volcanoes on Venus using a Scientist-Trainable Analysis System", *Proc. 26th Lunar and Planetary Science Conference*, pp. 1458, 1995.
3. M. C. Burl, U. Fayyad, P. Perona, P. Smyth and M. P. Burl, "Automating the Hunt for Volcanoes on Venus", *proc. of Computer Vision and Pattern Recognition Conference (CVPR-94)*, pp. 302-308, 1994.
4. E. F. Codd, *Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate*, 1993.
5. J. D. Kennefick, R. R. De Carvalho, S. G. Djorgovski, M. M. wilber, E. S. Dickinson, N. Weir, et al., "The Discovery of Five Quasars at z¿4 Using the Second Palomar Sky Survey", *Astronomical Journal*, vol. 110, no. 1, pp. 78-86, 1995.
6. "Communications of The ACM", *special issue on Data Mining*, vol. 39, no. 11.
7. R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, New York:John Wiley and Sons, 1973.

8. U. Fayyad, D. Haussler and P. Stolorz, "Mining Science Data", *Communications of ACM*, vol. 39, no. 11, 1996.
9. Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.
10. U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From Data Mining to Knowledge Discovery: An Overview" in Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.
11. C. Glymour, D. Madigan, D. Pregibon and P. Smyth, "Statistical Themes and Lessons for Data Mining", *Data Mining and Knowledge Discovery*, vol. 1, no. 1, 1997.
12. D. Heckerman, "Bayesian Networks for Data Mining", *Data Mining and Knowledge Discovery*, vol. 1, no. 1, 1997.
13. J. Kettenring and D. Pregibon, *Statistics and Massive Data Sets Report to the Committee on Applied and Theoretical Statistics*, 1996.
14. Leamer and E. Edward, Specification searches: cd hoc inference with non-experimental data, Wiley, 1978.
15. M. Mehta, R. Agrawal and J. Rissanen, "SLIQ: a fast scalable classifier for data mining", *Proceedings of EDBT-96*, 1996.

London Journal Press Membership

For Authors, subscribers, Boards and organizations

London Journals Press membership is an elite community of scholars, researchers, scientists, professionals and institutions associated with all the major disciplines. London Journals Press memberships are for individuals, research institutions, and universities. Authors, subscribers, Editorial Board members, Advisory Board members, and organizations are all part of member network.

Read more and apply for membership here:
https://journalspress.com/journals/membership

For Authors

For Institutions

For Subscribers

Author Membership provide access to scientific innovation, next generation tools, access to conferences/seminars /symposiums/webinars, networking opportunities, and privileged benefits.
Authors may submit research manuscript or paper without being an existing member of LJP. Once a non-member author submits a research paper he/she becomes a part of "Provisional Author Membership".

Society flourish when two institutions come together." Organizations, research institutes, and universities can join LJP Subscription membership or privileged "Fellow Membership" membership facilitating researchers to publish their work with us, become peer reviewers and join us on Advisory Board.

Subscribe to distinguished STM (scientific, technical, and medical) publisher. Subscription membership is available for individuals universities and institutions (print & online). Subscribers can access journals from our libraries, published in different formats like Printed Hardcopy, Interactive PDFs, EPUBs, eBooks, indexable documents and the author managed dynamic live web page articles, LaTeX, PDFs etc.