Computational analysis of CpG site DNA methylation

Mohammadmersad Ghorbani

A thesis submitted for the degree of

Doctor of Philosophy

Brunel University

September 2013

School of information Systems, Computing and Mathematics

Abstract

Epigenetics is the study of factors that can change DNA and passed to next generation without change to DNA sequence. DNA methylation is one of the categories of epigenetic change. DNA methylation is the attachment of methyl group (CH3) to DNA. Most of the time it occurs in the sequences that G is followed by C known as CpG sites and by addition of methyl to the cytosine residue. As science and technology progress new data are available about individual's DNA methylation profile in different conditions. Also new features discovered that can have role in DNA methylation. The availability of new data on DNA methylation and other features of DNA provide challenge to bioinformatics and the opportunity to discover new knowledge from existing data. In this research multiple data series were used to identify classes of methylation DNA to CpG sites. These classes are a) Never methylated CpG sites,b) Always methylated CpG sites, c) Methylated CpG sites in cancer/disease samples and non-methylated in normal samples d) Methylated CpG sites in normal samples and non-methylated in cancer/disease samples. After identification of these sites and their classes, an analysis was carried out to find the features which can better classify these sites a matrix of features was generated using four applications in EMBOSS software suite. Features matrix was also generated using the gUse/WS-PGRADE portal workflow system. In order to do this each of the four applications were grid enabled and ported to BOINC platform. The gUse portal was connected to the BOINC project via 3G-bridge. Each node in the workflow created portion of matrix and then these portions were combined together to create final matrix. This final feature matrix used in a hill climbing workflow. Hill climbing node was a JAVA program ported to BOINC platform. A Hill climbing search workflow was used to search for a subset of features that are better at classifying the CpG sites using 5 different measurements and three different classification methods: support vector machine, naïve bayes and J48 decision tree. Using this approach the hill climbing search found the models which contain less than half the number of features and better classification results. It is also been demonstrated that using gUse/WS-PGRADE workflow system can provide a modular way of feature generation so adding new feature generator application can be done without changing other parts. It is also shown that using grid enabled applications can speedup both feature generation and feature subset selection. The approach used in this research for distributed workflow based feature generation is not restricted to this study and can be applied in other studies that involve feature generation. The approach also

needs multiple binaries to generate portions of features. The grid enabled hill climbing search application can also be used in different context as it only requires to follow the same format of feature matrix.

Declaration

I hereby declare that the research presented in this thesis is my own work except where otherwise stated, and has not been submitted for any other degree.

Acknowledgements

I would like to thank my family for their support in my study, my supervisors Dr Annette Payne and Dr Simon Taylor. I would like to thank students, academics and member of staff at department of information systems and computing at Brunel University for their help and advice in carrying out this research.

Supporting Publications

The following publications have resulted from the research presented in this thesis:

1) Taylor, S.J., Ghorbani, M., Mustafee, N., Turner, S.J., Kiss, T., Farkas, D., Kite, S. and Straßburger, S. (2011) "Distributed computing and modeling & simulation: Speeding up simulations and creating large models", *Simulation Conference (WSC), Proceedings of the 2011 Winter* IEEE, pp. 161.

2) Simon Taylor, Mohammadmersad Ghorbani, Tamas Kiss, Daniel Farkas and David Gilbert. Investigating Approaches to Speeding Up Systems Biology Using BOINC-Based Desktop Grids. Simon J E Taylor. Mohammadmersad Ghorbani. IWSG-Life 2011 science gateways for life sciences. 3rd international workshop on science gateways for life sciences London, United Kingdom 8-10 June 2011.

3) Taylor, Simon J.E. and Ghrobani, Mohammadmersad and Kiss, Tamas and Elder, Mark. (2012) Investigating volunteer computing for Simul8, a feasibility study. In: *Operational research society simulation workshop 2012 (SW12),* 27th - 28th March 2012, Worcestershire, England.

4) Mohammadmersad Ghorbani, David Gilbert, Mark Pook and Annette Payne Computational analysis of the differentially methylated DNA in the frataxin gene (FXN) seen in Friedreich's ataxia (FRDA) patients , 2012, Atax. Res. Conf. Abs.: 74.

5) Mohammadmersad Ghorbani, Annette Payne, Simon J. E. Taylor, Michael Themis Computational identification and Analysis of CpG Site Methylation in Cancer Cells *The Twelfth International Symposium on Intelligent Data Analysis (IDA2013) 17 - 19 October 2013 Royal Statistical Society, London, UK.*

6) Mohammadmersad Ghorbani, Simon J E Taylor, Mark A. Pook, and Annette Payne. 2013 Comparative (Computational) Analysis of the DNA Methylation Status of Trinucleotide Repeat Expansion Diseases . *Journal of Nucleic Acids Hindawi publication.*

Contents

List of Figures vi
List of Tables ix
Chapter 1 - Introduction 1
1 Overview 1
1.1 Rational and motivation 2
1.2 Aims and Objectives 3
1.3 Research Methodology 3
1.4 Outline 5
1.5 Contributions 6
1.6 Chapter Summary 7
Chapter 2 – Background 8
2 Introduction 8
2.1 Biological background 8
2.1.1 Epigenetic change biological perspective 11
2.1.2 DNA methylation 11
2.2 Methods of identifying DNA methylation 12
2.2.1 Bisulfite conversion 12
2.2.2 Microarray technology 14
2.2.3 Microarray platforms 15
2.3 Bioinformatics 17
2.4 Grid computing 18
2.4.1 Middleware software 19
2.4.2 Advantages of desktop grid 20
2.5 Desktop grid applications in bioinformatics 21
2.5.1 gLite 22
2.5.2 Globus 23
2.5.3 ARC 24
2.5.4 UNICORE 24
2.5.5 BOINC 25
2.5.6 XtremWeb 27
2.5.7 OurGrid 27

2.5.8 SZDG 28
2.5.9 HTCondor 29
2.6 Workflow Systems 30
2.6.1 Taverna 31
2.6.2 Galaxy 31
2.6.3 Kepler 31
2.6.4 Jopera 32
2.6.5 SQLshare 32
2.6.6 gUse 32
2.7 DNA methylation data sources 33
2.7.1 Published papers 33
2.7.2 General databases 34
2.7.3 Special purpose DNA methylation databases 34
2.8 Feature generation 35
2.8.1 MEME (Multiple EM for Motif Elicitation) 36
2.8.2 MAST (Motif Alignment & Search Tool) 36
2.8.3 Word Composition 36
2.8.4 EMBOSS 36
2.8.5 UCSC table browser 37
2.9 Machine learning 38
2.10 Classification methods in DNA methylation analysis 39
2.10.1 Problem definition 39
2.10.2 Tree models 40
2.10.3 Support vector machine 41
2.10.4 Naive Bayes classification 42
2.10.5 Neural Networks 43
2.11 Measurement of prediction outcome 44
2.11.1 Cross validation 44
2.11.2 Confusion matrix: 44
2.11.3 The true positive rate, precision and F-Measure 45
2.11.4 The area under ROC curve 46
2.11.5 Kappa 46
2.12 Searching the feature space 47
2.12.1 Heuristic search 48

2.12.2 Hill climbing 49

2.12.3 Simulated annealing 51

2.12.4 Genetic search algorithm 53

2.13 Summary 55

Chapter 3 - Comparative analysis of three Trinucleotide Repeated Diseases 57

3 Background 57

3.1 Method 59

3.1.1 Pattern Generation 59

3.2 Results 64

3.2.1 Frequency 64

3.2.2 Proportional frequency 65

3.2.3 WEKA J48 Analysis 66

3.2.4 Comparison with MEME software 67

3.2.5 CTCF binding 69

3.2.6 Comparison of patterns with DNA binding protein sites 69

3.3 Conclusion 70

Chapter 4 - CpG site identification 73

4 Background 73

4.1.1 Objectives 74

4.2 Methods 75

4.2.1 Datasets Selection 75

4.2.2 Raw data standardization for all series 75

4.2.3 Methods of Dataset selection 77

4.2.4 CpG sites identification 81

4.2.5 Motif Discovery 83

4.2.6 Using motifs for Classification 84

4.3 Results 84

4.3.1 CPG sites and their classes 84

4.3.2 Genes neighbouring the CpGs in Experiment A 85

4.3.3 MicroRNA results 89

4.3.4 Discovered motifs 90

4.3.5 Classification results 91

4.3.6 WEKA 91

4.3.7 Summary 93

Chapter 5 - Grid enabled workflow based feature generation and feature subset selection 94
5 Background 94
5.1 Objectives 95
5.2 Overview of chapter 96
5.3 BOINC desktop grid computing 96
5.4 Workunit submission 96
5.5 GenWrapper 97
5.6 gUse (Grid and Cloud User Support Environment) 99
5.7 Test grid Infrastructure preparation 100
5.8 Grid enabling selected EMBOSS applications 102
5.8.1 Banana 102
5.8.2 Btwisted 103
5.8.3 Wordcount 104
5.8.4 Jaspscan 106
5.9 Creating Feature Generation workflow 108
5.9.1 One application feature generation 110
5.9.2 Multiple Application feature generation 111
5.10 Workflow submission 114
5.11 Grid enabled hill climbing search 118
5.12 Hill climbing search master application 120
5.13 Hill climbing Search collector application 121
5.14 Hill climbing Search Workflow Generation 122
5.15 Hill climbing Search Job Submission 123
5.16 Performance test feature generating matrix 123
5.16.1 Performance test hill climbing search 126
5.17 Performance testing of each classification method 127
5.18 Conclusion 128
Chapter 6 - Features Classifying CpG sites 130
6 Background 130
6.1 Objectives 131
6.2 Feature subset selection. 131
6.3 All features results from classification with different classification methods in WEKA 133

6.4 All features results in each feature class type from classification with different classification methods in WEKA. 133

6.4.1 Structural features 133

6.4.2 Transcription factor binding features 134

6.4.3 Word count features 134

6.5 Feature subset selection 134

6.6 Hill climbing 136

6.7 Conclusion 141

Chapter 7 Summary and Future Work 143

7 Introduction 143

7.1 Findings 144

7.1.1 Distinguishing motifs in trinucleotide repeat disease 144

7.1.2 Identification of CpG sites in different classes 145

7.1.3 Fast and Scalable feature generation system, ...146

7.1.4 Predictive model of DNA methylation 147

7.2 Future work 148

7.2.1 Different methylation contexts 148

7.2.2 Improvement of feature generation 148

7.2.3 Improvement to hill climbing search 148

7.2.4 Conclusion 149

References 150

Appendix A: Ranked table of the features. 162

Appendix B Source codes 200

Ported wordcount source code 200

Ported jaspscan source code 202

List of Figures

Figure 1-1 Steps used in this research for creating predictive model of CpG site methylation..5
Figure 2-1 the figure shows the process of protein creation in from DNA to RNA to Protein 9
Figure 2-2 Table shows codes and amino acid related to each codon. Codons are three letter DNA base pairs. 10
Figure 2-3 Epigenetic changes 11
Figure 2-4 : Molecular view of "CpG site" methylation and process of DNA methylation by DNA methyltransferase. (hgu 2013) and (mpipsykl 2013)..................12
Figure 2-5 Process of bisulphite sequencing which can detect CpG site methylation. Two locations are cytosine after treatment so they are methylated in original sequence. 13
Figure 2-6 Hybridisation in microarray technology..14
Figure 2-7 increase in sequencing data stored at EBI over the years (Marx, 2013) 18
Figure 2-8 Type of different grid middlewares. 20
Figure 2-9 some of high ranked projects related to biological research which uses BOINC information from (boincstat 2013) and individual project web sites...............22
Figure 2-10 Job Flow in gLite (Burke Stephen et al., 2011)..23
Figure 2-11 Globus High level services (globus project 2013) 24
Figure 2-12 Architecture of UNICORE source (unicore project 2013)..........................25
Figure 2-13 Interaction between different BOINC processes and database processes in the rectangle are server side components. (Ries, Schroder and Grout, 2011) 26
Figure 2-14 OurGrid high level architecture (OurGrid 2013) 28

Figure 2-15 SZDG architecture (Taylor *et al.*, 2011) 29
Figure 2-16 Main processes in HTCondor system (Thain, Tannenbaum and Livny, 2005) 30
Figure 2-17 Different ways of presenting DNA methylation from top FMR1 methylation, DM1 methylation, FXN CpG sites methylation. 33
Figure 2-18 USCS table browser form and define regions form (UCSC Browser 2013) 38
Figure 2-19 shows support vector that separate dots from squares with hyperplane (Tufféry, 201, p. 503)..42
Figure 2-20 simple perceptron (Alpaydin E 2010) 43
Figure 2-21. An example of confusion matrix for four classes of CpG site methylation. 44
Figure 2-22 shows the true positive rate calculation from confusion matrix from Figure 2-21. 46
Figure 2-23 calculation of kappa statistic from confusion matrix 47
Figure 2-24 Hill climbing finds local optima..50
Figure 3-1 This figure shows the 3 gene regions under investigation: A) DMPK 3'UTR region, B) FMR1 5' UTR region and C) FXN intron 1 region. A scheme of the DNA sequence, transcriptional start site and the regions analysed are shown. The grey shading shows the always methylated (AM) regions, blue, the never methylated (NM) and the yellow area shows variably methylated (VM) regions. CpG sites are underlined and bold numbers at start and end of each line show base pair number in the sequence. A triangle shows the location of repeat expansion and the box above triangle shows the TNR repeats. The green highlighted region in the FMR1 gene indicates the promoter region. The CTCF binding sites are shown in red Expanded from (Pook 2012)............61
Figure 3-2 Decision tree created by WEKA package. AM is always methylated, NM is never methylated (NM) and (VM) is variably methylated..67
Figure 4-1 Summary of previous works on CpG island and CpG site methylation 74
Figure 4-2 shell script code for extracting samples raw data and creating one file per sample 76
Figure 4-3 Steps in finding Normal and Cancer samples in GEO database 79
Figure 4-4 CpG site identification steps...83
Figure 4-5 Proportion of CpG site to UCSC CpG Island to total number of CpG sites .85
Figure 4-6 proportion of each relation to total number of CpG sites in each class.........85
Figure 4-7 Unique microRNA in each class of CpG sites 90
Figure 4-8 Motifs discovered in DNA sequence of CpG sites.......................................91
Figure 4-9 This diagram illustrates that the m13C (TCCAAGGGACACC) motif doesn't occur in the flanking DNA sequences occurring in 50 out of 51 of the CpGs identified in experiment A and occurs in all 7 of the sequences surrounding CpGs identified in experiment C...92
Figure 5-1: Snapshot of BOINC output template file. 97
Figure 5-2. Diagram shows how different part of GenWrapper interacts with each other and communication with BOINC client (Marosi, Balaton and Kacsuk, 2009). 99
Figure 5-3. gUse server architecture. 100

Figure 5-4. Different software used in the client and server side of the system and their interaction with each other. 101
Figure 5-5. The client side view of program used for feature generation and feature subset selection....... 102
Figure 5-6 part of jaspscan result file....... 108
Figure 5-7. Input sequence file and Output feature file. 109
Figure 5-8. Workflow of one node banana contains generator application and collector. 110
Figure 5-9. workflow for two ported application btwisted and banana. 112
Figure 5-10. the completed workflow of the feature matrix generator needs two input files one determining the number of sequences per file and the other is the fasta file of all sequences. Individual application results could be downloaded as in the workflow they were configured as permanent .The final results were generated in the matrix generator node. 113
Figure 5-11 an example of Workflow configuration page 115
Figure 5-12 an example of a job Submission page, the “configure” button should be used to add input and output files and to define binaries for each workflow. 115
Figure 5-13. Details of the status of each job after workflow submission could be monitored in ws-pgrade portal. This figure shows the workflow submission for an input file which contained 20 sequences. It then indicated that each sequence should be submitted individually so eventually 20 jobs were submitted. 117
Figure 5-14. An example of BOINC results page. 118
Figure 5-15. Hill climbing search workflow graph in the graph editor shows the search node is the worker node, and generator and collector nodes are master nodes 122
Figure 5-16. BOINC client log snapshot shows details of tasks running on clients machines....... 123
Figure 5-17 speedup improvement by batching of more than 1 sequence per file 124
Figure 5-18 The speed-up result for 653 sequence in the dataset for different numbers of sequences per file. 125
Figure 5-19 The graph shows speedup results for two different kinds of configuration. Improvement could be seen as the number of “wu_in_progreess” was restricted. 127
Figure 5-20 shows speedup results for different classification methods in feature subset selection....... 128
Figure 6-1: feature subset selection method....... 135
Figure 6-2 Comparison of best random starting point, random starting point, all feature starting point, zero feature starting point kappa maximisation with hill climbing 136
Figure 7-1 Relationship between algorithms developed as part of this research 144
Figure 7-2 Feature generation and subset selection workflows 149

List of Tables

Table 2-1 List of general purpose public databases....34
Table 2-2 DNA methylation databases....35
Table 2-3 Hill Climbing....51
Table 2-4 Simulated Annealing....53
Table 2-5 Genetic Algorithm....55
Table 3-1 Table of discriminatory patterns....62
Table 3-2 Table of unique patterns in each region....63
Table 3-3 Pattern count algorithm....65
Table 3-4 Comparison of MEME results with patterns found in this study....68
Table 4-1 Dataset Selection Algorithm....78
Table 4-2 GEO data series....80
Table 4-3 CpG site identification....82
Table 4-4 Genes neighbouring expriment A....86
Table 4-5: The genes neighbouring the CpGs found in Experiment C....89
Table 5-1 Sequential feature generation algorithm....95
Table 5-2 Comparison of different way of porting application boinc platform (Marosi, Kovács and Kacsuk, 2013)....98
Table 5-3: Grid Enable banana program....103
Table 5-4: Grid enabled btwisted....104
Table 5-5: Grid enabled wordcount....105
Table 5-6 : Grid Enable Jaspscan....107
Table 5-7 Summary of ported applications....108
Table 5-8 Example of full set feature matrix.....119
Table 5-9 Example of feature subset matrix....119
Table 5-10 Speedup results for different number of sequences and different number sequence per files....124
Table 5-11 speedup results for different number of iterations of hill climbing for 20 to 100 iterations....126
Table 5-12 Shows the comparison of speed-up for different numbers of jobs and different configurations.....127

Table 5-13 details of speedup as programs progress 128
Table 6-1 Reported result of selected papers on Epigenetic analysis 130
Table 6-2 Hill Climbing Search 132
Table 6-3: results of applying different classification methods using all features........133
Table 6-4: results of applying classification method using only structural features.....133
Table 6-5: results of applying classification method using only transcription factors.134
Table 6-6 Results of applying classification method using only word counts.............134
Table 6-7 Kappa statistic strength of agreement...137
Table 6-8 Details of expriments and values of each expriment for the maximum result 138
Table 6-9 table shows direct and indirect improvement by hill climbing search. 139
Table 6-10 : Proportion of features from different type of features in the final subset. Total number of feature column shows the proportion of features from the initial full feature space. Table is sorted by kappa value. 140
Table 6-11 top 5 group of features that shared among different subsets. Percentage represents percentage of models that these features occurred in them details of each feature provided in Appendix A...141

Chapter 1 - Introduction

1 Overview

Advancement in technology and science leads to the increase in size and accumulation of biological data. These advancements make new challenges for analysis of data and also provide new opportunities to find new knowledge from existing data. These challenges include coping with new discoveries as the science progress and integrating this new knowledge to existing knowledge and processing of large datasets which needs some form of standardisation.

One of the new areas in biology which produce large datasets is Epigenetic. In epigenetic studies, factors that can pass to next generation without change to the DNA sequence are examined (Sharma, Kelly and Jones, 2010).

Firstly in a smaller scale study, DNA methylation in three different trinucleotide repeat diseases was examined. The data was gathered from charts and graph from publications. To compare this data, they were manually reproduced and standardised into 3 classes. This combination and the use of word pattern as features haven't been reported on these trinucleotide repeats disease. These disease are fragile X syndrome, myotonic dystrophy type I and Friedreich's ataxia (Pook M *et al* 2008) (Naumann *et al*., 2009) (Lopez Castel *et al*., 2011). Three different classes of regions which were analysed are described here:

a) Variably methylation CpG regions (more than one site) in normal and disease condition.
b) Non-variably methylated CpG regions in normal and disease condition.
c) Never methylated CpG regions in normal and disease condition.

Secondly methylation of CpG sites in DNA sequences were studied using the datasets which stored in GEO (Gene Expression Omnibus) database (Barrett *et al*., 2013) over the time using Illumine HumanMethylation450K microarray technology (Illumina, 2012c). This microarray platform is the most recent microarray technology with the largest coverage comparing with other platforms. The samples examined here are from cancer and normal tissue, with four possible classes of samples being examined. These classes are:

a) Never methylated CpG sites.
b) Always methylated CpG sites.
c) Methylated CpG sites in cancer samples and non-methylated in normal samples.
d) Methylated CpG sites in normal samples and non-methylated in cancer samples.

In order to address the challenge of integrating new discoveries, a scalable and modular system was designed. In this system each new module represents different kind of analysis on DNA sequence and can be added to or updated without change to another part of the system. In order to overcome computational challenge of analysing data, grid computing techniques were applied. Grid computing provides different methods and tools to share resources (CPUs, storage, instruments, data etc.) among users and to accelerate scientific research (Wilkinson, 2011). CpG sites identified in different classes were examined by classification and heuristic search methods, these methods found a feature subset that can better classify CpG sites. Finally predictive models provided good accuracy. It is possible to use these models as the pre-step of biological experiment design. For example, biologist may want to check set of CpG sites rather than all of them. They first examine these sets with the computational models and then choose those CpG sites that show better results in the computational test. So this research provides distributed modular system for feature generation and feature subset selection. Additionally identified CpG sites from examination of multiple datasets can have biomarker potential.

1.1 Rational and motivation

New technologies produce large amount of data about single location in DNA sequence in different conditions. Examples of these conditions are samples produced at different times, tissues, disease and etc. Accumulation of these datasets over the time provides opportunity to discover new biomarkers. Stable biomarkers among all samples can provide insight into the process which leads to that stability or better design of experiments which examines variable biomarkers. Variable biomarkers may have diagnostic value.

1.2 Aims and Objectives

The aim of this research is to find predictive models for the classification of different classes of CpG DNA methylation in human DNA.

In order to achieve this aim these objectives were completed:

1. Identify CpG sites with their classes from existing publically available data sets from normal and diseased individuals.
2. Identify features related to and classify these CpG sites using IDA and grid computing techniques.
3. Define and implement software that determines a feature subset (model) that predicts these classes of CpG sites accurately.

1.3 Research Methodology

The approach used in this research to analyse CpG sites methylation involves following specific steps. The proposed approach can be used as methodology, or roadmap, for DNA methylation studies. Each step's output can be used separately and the whole method can be used as a methodology to analyse DNA methylation.

The first step defines the target of the study, i.e., the type of disease. In this thesis, two types of disease are analysed, trinucleotide repeat diseases and cancer. Data was obtained from published papers or raw datasets that are publicly available. Three methylation classes are investigated here, namely never methylated, always methylated and variably methylated in the case of disease and normal DNA sequences. Methylation classes can be defined based on the tissue, cell type etc. These classes do not necessarily have to be defined as disease vs. normal. As it has been shown in previous research studies, DNA methylation varies among different cell and tissue types in normal situation and also over different time intervals (Sharma, Kelly and Jones, 2010).

Identifying regions or CpG sites based on the defined classes is next step. The CpG methylation is either defined in the literature or has to be calculated from raw data. This data can be provided as a dataset to further analyse attributes that can better classify these classes.

After identifying CpG sites, attributes related to them can be calculated based on the sequences near these sites by using applications like EMBOSS (Rice, Longden and Bleasby, 2000). They can be queried in other databases, like UCSC table tracks (UCSC Browser 2013), to identify the overlap with other genomic attributes. The attribute table provides a dataset for classification and data mining methods. It has been previously shown that these methods have effectively classified CpG sites or regions based on the identified classes. A grid-enabled workflow system simplifies the process of adding new nodes to the feature generation workflow. This system accelerates the feature generation and the feature subset selection by distributing jobs over desktop grid machines. Figure 1-1 shows each step, and the output of this step, of the methodology used in this research.

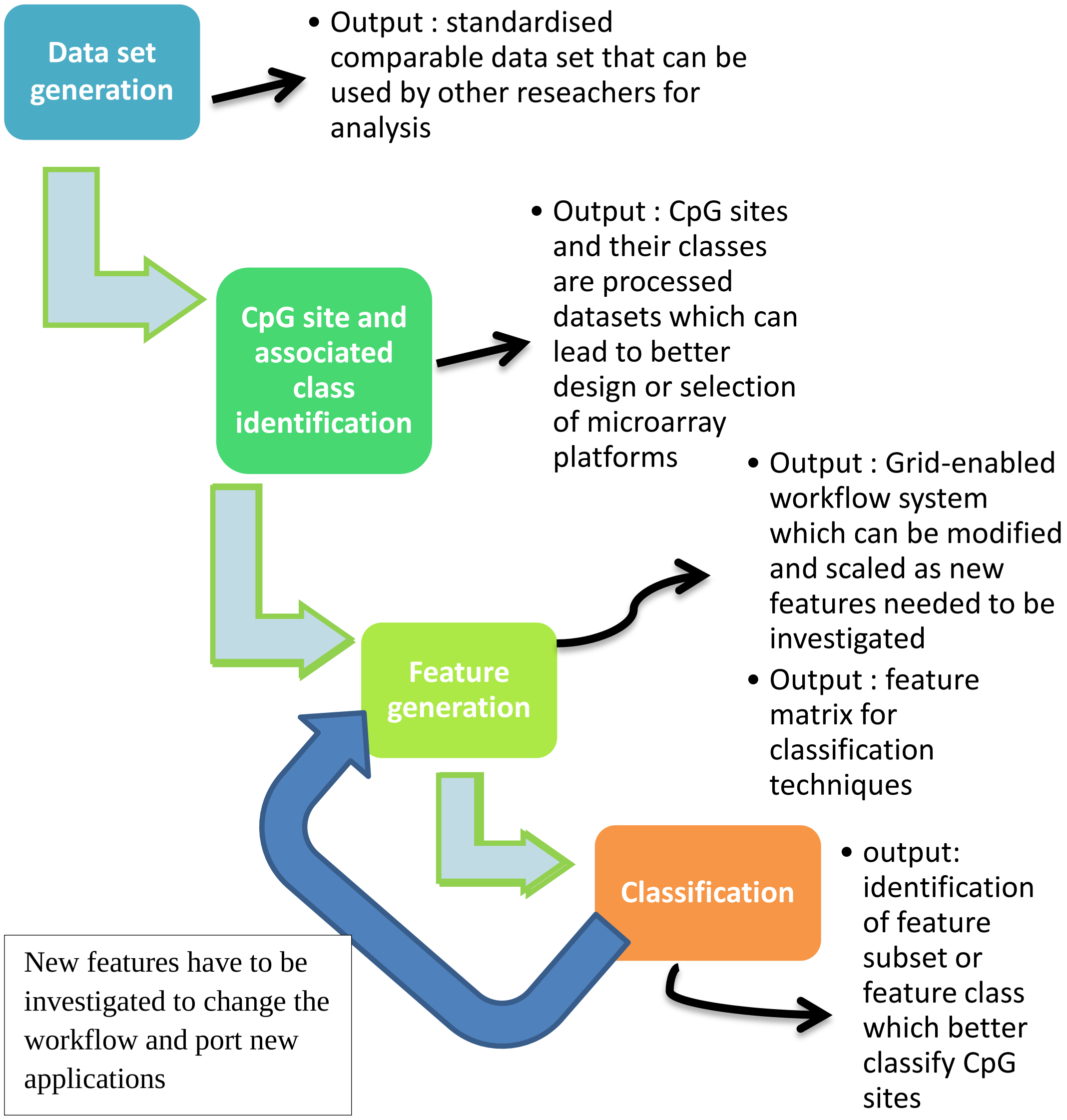

Data set generation
• Output : standardised comparable data set that can be used by other reseachers for analysis
CpG site and associated class identification
• Output : CpG sites and their classes are processed datasets which can lead to better design or selection of microarray platforms
Feature generation
• Output : Grid-enabled workflow system which can be modified and scaled as new features needed to be investigated
• Output : feature matrix for classification techniques
Classification
• output: identification of feature subset or feature class which better classify CpG sites
New features have to be investigated to change the workflow and port new applications

Figure 1-1 Steps used in this research for creating predictive model of CpG site methylation

1.4 Outline

Chapter 2 provides biological background to the problem that is studied here. It also provides outlook on new challenges made by data generation and needs for processing and analysis of data by new technologies in biological science. Grid computing is reviewed as the technology that accelerates the processing of the data. Also different workflow systems are examined as one of the methods to creating modular systems which integrate new discovery tools without the change to other part of the system.

Heuristic search and classification methods are reviewed as the method for selecting important features subsets that classify CpG sites.

Chapter 3 proposes the DNA motifs that distinguish DNA methylation region classes in in three trinucleotide repeat diseases. The data source came from manually examining each trinucleotide repeat diseases methylation patterns in published papers.

Chapter 4 provides the method of selecting datasets stored in GEO database and making datasets comparable. A CpG site class identification method and the discovered CpG sites are provided in this chapter.

Chapter 5 describes details of the development of the grid enabled workflow system for generating features related to DNA sequence around the CpG sites. The workflow for searching the subset of important features is provided in this chapter.

Chapter 6 describes methods and models for predicting the different classes of CpG site.

Chapter 7 provides future work proposed in the area of DNA methylation. It also provides the suggestion for improvement of the workflow system. It summarised the results obtained in the thesis.

1.5 Contributions

1) Identification of motifs that distinguish three trinucleotide repeats disease methylation region classes. (Chapter 3)

2) Identification of CpG sites in four classes of methylation in normal and cancer samples. These sites can be seen as the biomarker when they are variable in disease and normal sets. Also they could be used as the dataset to further analyse the features associated with them. (Chapter 4)

3) A modular Grid enabled workflow system for analysing identified CpG sites. (Chapter 5)

4) Generation of models predicting CpG site methylation classes and identification of features which are shared among all models. (Chapter 6)

5) A computational method for analysing CpG site methylation from dataset selection to feature subset selection is proposed. (Chapter 4,5,6)

1.6 Chapter Summary

In this chapter the research which was carried out in this thesis is briefly reviewed. It is stated in this chapter that existing data in the public database on DNA methylation have the potential of containing useful knowledge and computational tools can be applied to extract information from these data. In the process of achieving these tasks, a grid enabled web based system was developed that can be applied in other similar circumstances. The next chapter provides the background on DNA methylation and computational methods used in this research.

Chapter 2 – Background

2 Introduction

The research area in this thesis is in the field of bioinformatics and the use of different computational techniques to discover new knowledge from accumulated data from advancement in technology and science. This chapter provides biological background on the problem under investigation. It also provides an introduction to grid computing and a summary of common grid middleware. Grid computing techniques are used in this research to reduce the execution time of computational intensive programs that are usually employed in this type of studies. Finally, classification and heuristic search methods are summarised.

2.1 Biological background

Every living system's biological structure is organised in a hierarchy of organisation levels. The lowest unit in the cellular level is the cell. A group of similar cells creates different tissues like blood, bone, etc. In sequence, a functional group of tissues creates an organ. Organs that work together to deliver specific functionality, make an organ system and, finally, organism is a living system that can carry out different processes like growth and reproduction. Many multicellular organisms have similar hierarchy of organisation, although some basic organisms, like bacteria, include only one cell.

Inside the cells are molecules of deoxyribonucleic acid (DNA). DNA carries encoded information which is later translated to RNA and later to proteins and leads to functionality in upper level of organism.

DNA molecules are strings made of simpler nucleotide units. Each nucleotide is composed of four different kinds of nucleobases: adenine (A), cytosine (C) guanine (G) and thymine (T), and a backbone which consists of phosphate groups and alternating sugar (deoxyribose). The DNA structure is mostly in the form of a double helix in which C is matched with G and T with A, in each strand of DNA. DNA is not just a long string of continuous base pairs but it is organised in structural units which are called chromosomes. Inside chromosomes, DNA is wrapped around histone proteins.

DNA is copied to RNA molecules (transcription). RNA is single stranded that contains uracil (U) instead of thymine (T) and RNA molecules are translated to proteins. Proteins are made up of chains of amino acids. This sequential transfer of information is known as *central dogma of molecular genetics* and explains the flow of information from DNA to protein (see Figure 2-1).

> "*The sequence of nucleotides in a gene specifies the sequence of nucleotide in the messenger RNA in turn the sequence of nucleotides in the messenger RNA specifies the sequence of amino acids in the polypeptide chain*"
>
> (Hartl and Ruvolo, 2011 p. 23)

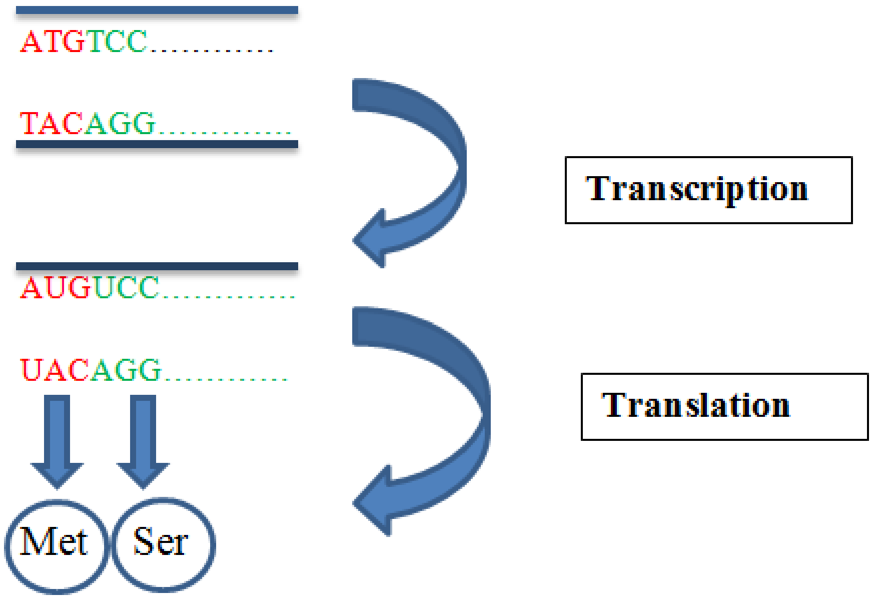
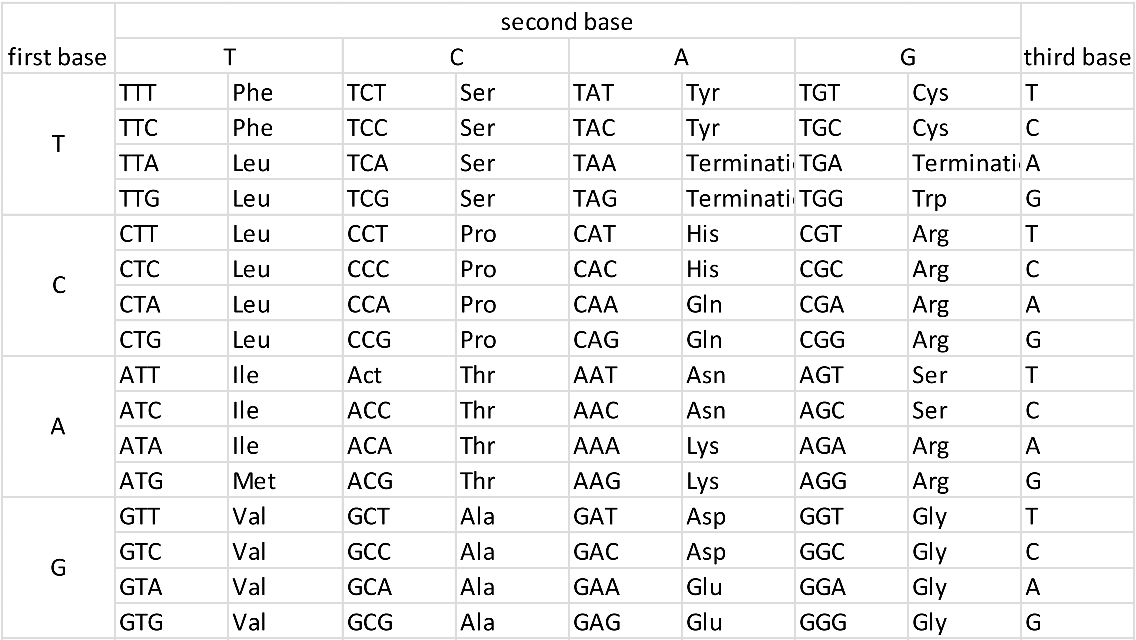


Figure 2-1 the figure shows the process of protein creation in from DNA to RNA to Protein

Proteins are important molecules in the cell and they are responsible for most of the cells functions like signalling, transportation, catalysis etc. Some proteins, like membrane proteins, are the building block of the structure of cells (Alberts, 2009, p 119).

Genes can be seen as subset of DNA sequence which finally translated to protein. Each three base pairs in DNA sequence finally translated to one amino acid in the protein. These triplets are called codon. Figure 2-2 shows standard genetic code and abbreviation of the translated amino acids. This is called genetic code and defines how the sequence of codons specifies the sequence of amino acid in a protein synthesis. As can be seen from the genetic code table, there are 64 codons in the code. This number derives from the number of the codon size, which is three, power the number of the base pairs, which are four. So, there are 3^4= 64 codons. Codons are not translated to amino acids in the one to one manner. Multiple codons can be translated to one amino acid in many cases.



	second base								
first base	T		C		A		G		third base
T	TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys	T
	TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys	C
	TTA	Leu	TCA	Ser	TAA	Terminati	TGA	Terminati	A
	TTG	Leu	TCG	Ser	TAG	Terminati	TGG	Trp	G
C	CTT	Leu	CCT	Pro	CAT	His	CGT	Arg	T
	CTC	Leu	CCC	Pro	CAC	His	CGC	Arg	C
	CTA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	A
	CTG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	G
A	ATT	Ile	Act	Thr	AAT	Asn	AGT	Ser	T
	ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	C
	ATA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	A
	ATG	Met	ACG	Thr	AAG	Lys	AGG	Arg	G
G	GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly	T
	GTC	Val	GCC	Ala	GAC	Asp	GGC	Gly	C
	GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly	A
	GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly	G

Figure 2-2 Table shows codes and amino acid related to each codon. Codons are three letter DNA base pairs.

Other mechanisms, which do not directly change the DNA sequence itself but modify the way DNA sequence accessed, can have influence on the way genes transcribed and later translated. The next section gives an overview of these changes.

2.1.1 Epigenetic change biological perspective

Epigenetics is the study of factors that can change DNA and pass to next generation without changing the DNA sequence. The more precise definition of epigenetics, provided in (Sharma, Kelly and Jones, 2010) is: “The study of heritable changes in gene expression that occur independent of changes in the primary DNA sequence”. They (Sharma, Kelly and Jones) described four main categories of epigenetics: DNA methylation, covalent histone modifications, non-covalent mechanisms, and non-coding RNAs.

In Figure 2-3, there is a diagrammatic representation of the different epigenetic changes. Part of this research involves the study of a single CpG site using multiple data series from GEO database, as highlighted in the diagram.

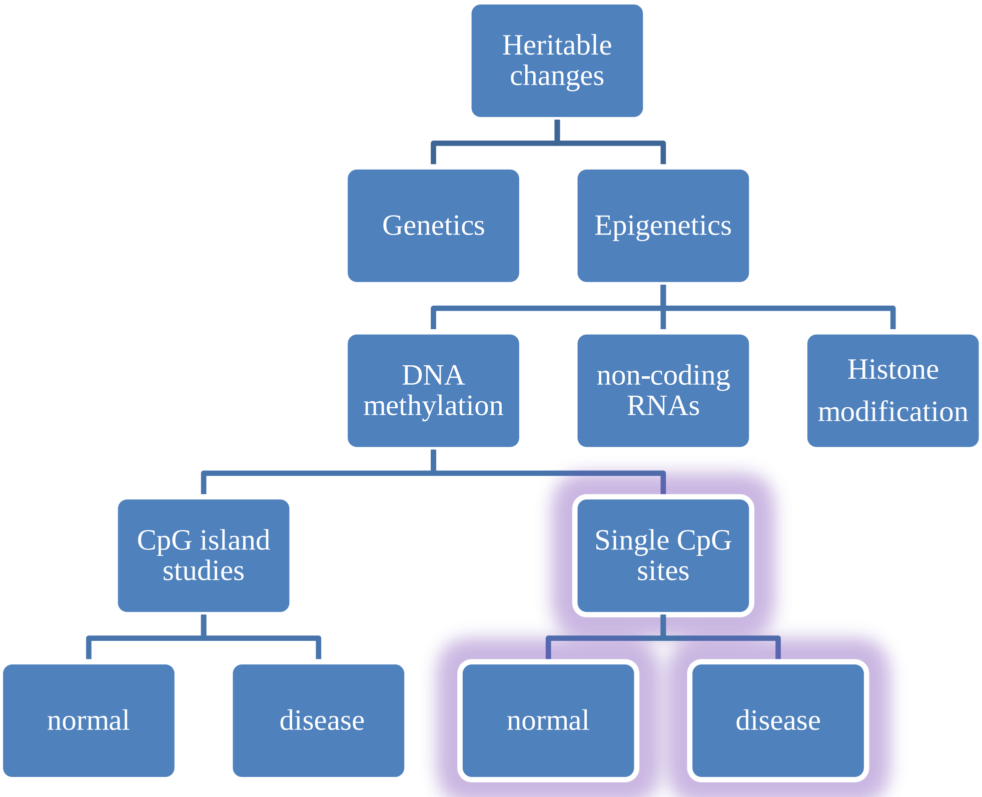


Figure 2-3 Epigenetic changes

2.1.2 DNA methylation

DNA methylation is the attachment of methyl group (CH_3) to DNA. Most of the time it occurs in the sequences in which a G nucleotide is followed by a C nucleotide, known as CpG sites, by the addition of methyl to cytosine (C). p indicates a phosphate group in DNA backbone that separates cytosine (C) and guanine (G). The process of DNA methylation is accomplished by DNA methyltransferase enzymes. It is known that methylation plays important biological role in many different areas. Sometimes it occurs naturally. However, when it occurs aberrantly, it is correlated with diseases like cancer and trinucleotide repeats diseases. The same DNA with similar sequences that altered by DNA methylation can have different function. Some genomic regions are rich in CpG sites and they are called CpG islands. CpG islands are known as important regulatory regions. DNA methylation of CpG islands can suppress genes and cause diseases. Figure 2-4 illustrates the CpG methylation.

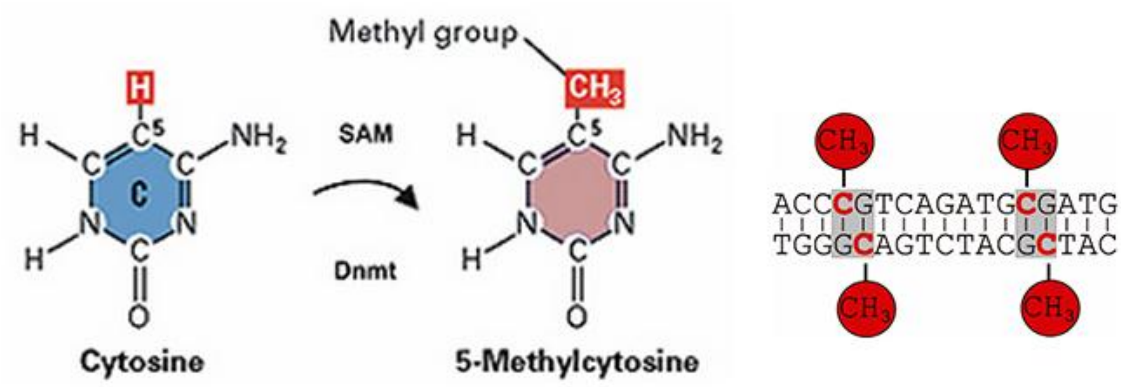


Figure 2-4 : Molecular view of “CpG site” methylation and process of DNA methylation by DNA methyltransferase. (hgu 2013) and (mpipsykl 2013)

2.2 Methods of identifying DNA methylation

There are different methods to detect DNA methylation, Peter W. Laird reviewed these methods (Laird, 2010), which are, among others, the array-based approach and the sequence based approach. The array based approach is better fitted when there are large numbers of samples, as concluded by Laird. This section provides a summary of bisulfite conversion as the “gold standard” in DNA methylation. Microarray technology and different kind of microarray platforms are reviewed here.

2.2.1 Bisulfite conversion

The procedure, developed by (Frommer *et al*., 1992), uses sodium bisulfite to convert cytosine to uracil. This conversion does not change methylated cytosine. A comparison of the original and the converted sequence can reveal the methylation status of CpG sites. This method is considered as the “gold standard” for detecting CpG site methylation (Lister and Ecker, 2009) and the process is shown in the Figure 2-5.

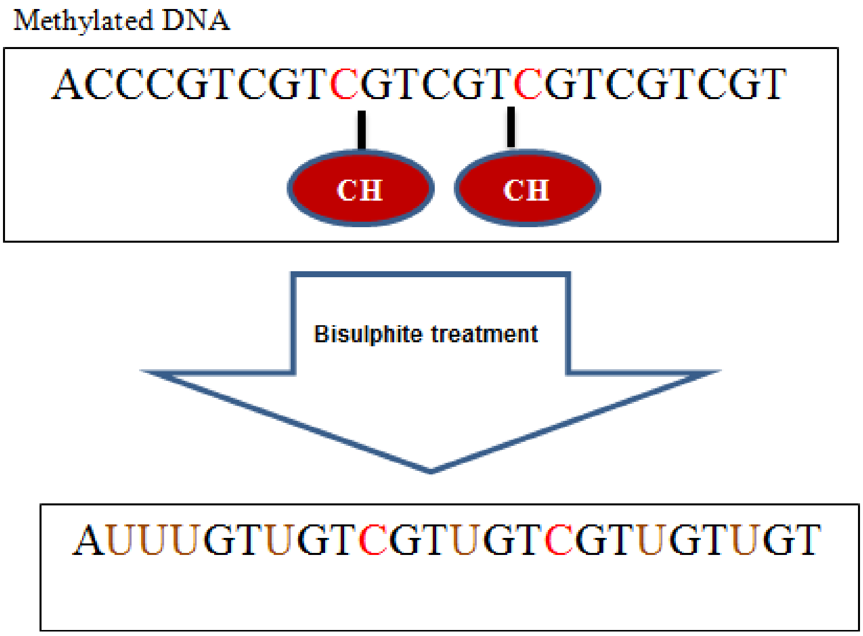


Figure 2-5 Process of bisulphite sequencing which can detect CpG site methylation. Two locations are cytosine after treatment so they are methylated in original sequence.

Bisulfite conversion is used in Human Epigenome Project (HEP). HEP determined the methylation value of 1.88 million CpG sites in chromosomes 6, 20, 22 and in 12 different tissues including biological and technical replications (Eckhardt *et al*., 2006).

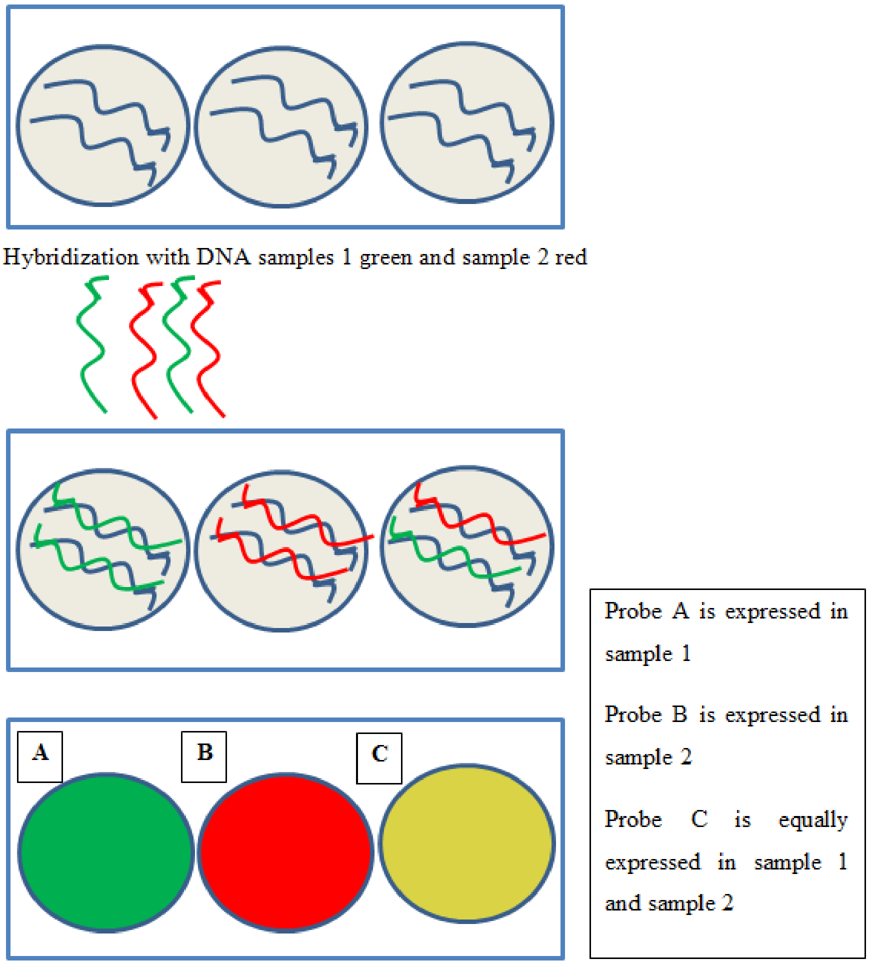
In another high resolution profiling, Zhang et al. used this technique to determine the methylation value of CpG sites in chromosome 21 and measured methylation state of 580,427 CpG sites in five human cell types (Zhang *et al*., 2009).

Lister et al. produced single base resolution maps of DNA methylation for two human cell lines, namely H1 human embryonic stem cells17 and IMR90 fetal lung fibroblasts1 (Lister *et al*., 2009). They have detected approximately 62 million methylcytosines in H1 and 45 million in IMR90 cell lines.

There is study currently carried out to map 1,000 human epigenomes with 100 blood cell types by BLUEPRINT project using bisulfite sequencing. The project includes other epigenetic changes, like histone marks (Adams et al., 2012).

2.2.2 Microarray technology

DNA microarray, known as DNA chip, consists of known DNA sequences of interest spotted on the chip. These microscopic DNA sequences are called probes. Sample DNAs are labelled with red or green fluorescent and they hybridised by the probe spots. Because the DNA samples, also called targets, are coloured when they hybridise, they show different colours depending on the level of hybridisation or expression in the spot. The level of expression can be detected by scanning the microarray. DNA microarray technology enables investigation of many DNA sites in parallel and batch mode. Chips can be designed for various purposes like gene expression or DNA methylation. Figure 2-6 shows the process of hybridisation in microarray technology for three probes.



Probe A is expressed in sample 1

Probe B is expressed in sample 2

Probe C is equally expressed in sample 1 and sample 2

Figure 2-6 Hybridisation in microarray technology

2.2.3 Microarray platforms

This section introduces some of the most popular DNA microarray platforms. It also provides information on the publicly available data sources on these platforms.

In microarray technology the probes are synthesised and then attached to a surface. Three of the microarray platforms used in microarray technology is based on the Infinium assays of the Illumina company and a fourth platform is based on ChIip technology. Both technologies are described in this section. In these microarray platforms two bead types are used for each location, methylated bead type and unmethylated bead type and these two bead types generate two signal intensities for each locus. The ratio of these two signals determines the DNA methylation. The Infinium II assay uses one bead type for assessing DNA methylation. This approach alongside with Infinium I is used in HumanMethylation450K, which is the platform of microarray datasets used in this study (Illumina, 2012b) . The fourth platform uses the chromatin immunoprecipitation to detect DNA methylation. The details of this method can be found in (Douglas Roberts, 2007).

2.2.3.1 GoldenGate Methylation Cancer Panel I

This methylation microarray platform covers 1,505 selected CpG sites from 807 genes. This platform has been used for cancer studies where 4,468 samples in 27 series submitted to GEO database (Accessed 04/07/2013). The access code for this platform in the ArrayExpress database is A-GEOD-9183. This platform is the predecessor of HumanMethylation27k and HumanMethylation450K. Because of lesser coverage, this platform is not used in this study.

2.2.3.2 HumanMethylation27K

HumanMetylation27k is a microarray platform which covers 27,578 CpG sites in Human genome. It covers 14,495 genes and has been widely used before HumanMethyltion450K which has larger coverage (Illumina, 2012a). The platform ID in GEO database is GPL8490 and currently there are 11,851 samples related to this platform in 202 series. In the ArrayExpress database, there are 201 experiments that use this array with A-GEOD-8490 ID. Some of these experiments are copied from the GEO database. This platform is not continued anymore. Nevertheless, because there are large numbers of samples in the GEO database, it can be used by data mining techniques to acquire new knowledge.

2.2.3.3 HumanMethylaion450K

HumanMethylation450K is one of the most widely used microarray platforms for analysing DNA methylation on single CpG site resolution. It covers 96% of CpG islands and, also, regions around CpG islands, like island shores and shelves. It also covers CpG sites in other areas of the genome. The total number of sites (including some none-CpG sites) is over 450,000 locations in human DNA. Nearly 90% of HumanMethylation27K platform, which is the predecessor of the HumanMethylation450K platform, is covered in this platform. This fact makes the platform very comprehensive in terms of coverage (Illumina, 2012c). There are more than 28 million CpG sites in the human genome and this represent 1% of total CpG sites. In the GEO database currently 99 data series are submitted that use this platform. This data contains a total of 4,889 samples.

2.2.3.4 Agilent Human CpG Island ChIP-on-Chip Microarray G4492A

This microarray uses different approach to the Illumina methylation microarray platform. It uses the method of chromatin immunoprecipitation on the microarray for 27,000 CpG islands with the method of tiling. It has lesser number of samples in the GEO public database (11 series and 110 samples) and less coverage than Illumina HumanMethylation450k (Douglas Roberts, 2007).

2.2.3.5 Other Custom Arrays

It is possible to create custom arrays to investigate specific sites. However, these arrays will be study-specific and because the CpG sites are varied among them, they need to be standardised so they can be compared.

2.3 Bioinformatics

In this research different bioinformatics methods are applied to analyse DNA methylation. These include retrieving raw data from databases and processing it to make it comparable and applying data mining tools on these data sets to drive new knowledge. In this section, a brief overview of bioinformatics is given.

Due to rapid advancement in the application of advanced technology in biology, large amount of data has been produced over the past decades. Consequently, computational tools and methods need to deal with this large amount of data. For example, the ENCODE project reported that it produced 15 terabytes of data (Birney, 2012). Generally, the term bioinformatics means the use of computer tools and methods in biology. Bioinformatics methods are used in many different ways in biological science, ranging from storing biological data, analysing this data, protein structure prediction, sequence alignment etc. The following definition for bioinformatics is provided by the dictionary of genetics:

> "A *new field in which computer hardware and software technologies are developed and used to gather, store, analyse, and disseminate biological data, images, and other information. Wide-scale, collaborative works in genomics and proteomics rely heavily on bioinformatics.*"
>
> (K.Mulligan,Robert C.King William D.Stansfield Pamela, 2007)

As mentioned earlier, large a amount of data is stored in public and private databases. European Bioinformatics Institute (EBI) currently stores 20 petabytes of data on genes. These trends will continue as sequencing technology is becoming cheaper and, therefore, individual genomes can be sequenced faster. Consequently, more data will be available and this will be the new challenge for bioinformatics. This data gives researcher the opportunity to get new knowledge by data processing and data mining tools. Figure 2-7 shows sequencing data storage trends at EBI. The graph can give an idea of the challenges computer scientist and engineers will face in the coming years in order to process and analyse these massive amounts of data (Marx, 2013).

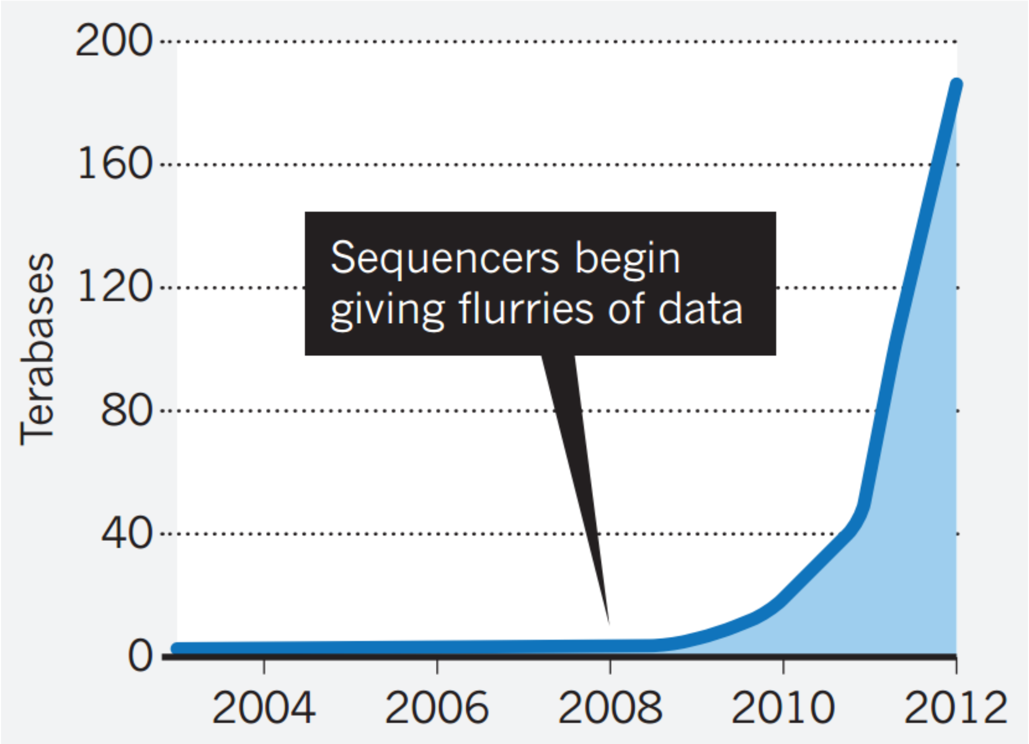


Figure 2-7 increase in sequencing data stored at EBI over the years (Marx, 2013)

Grid computing is one approach in dealing with the new challenges emerging by these trends in data generation and collection in biological science. In the next section, an overview of Grid computing in general is provided. Furthermore, section 2.4.1 discusses important middleware software in grid computing. Additionally, it states the benefits of using desktop grid computing.

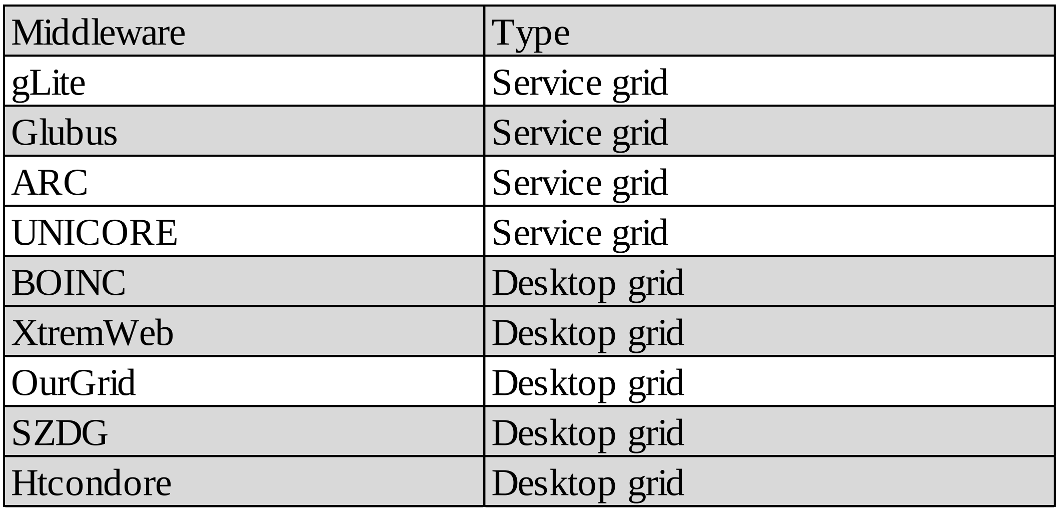
2.4 Grid computing

Grid computing is a form of distributed computing that provides access to resources, such as data, computing or instrument, via connected networks. The term 'grid computing' was first defined by Ian Foster and Carl Kesselman (1998) as: "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." Despite the fact that grid computing was first defined by Foster and Kesselman, the first example of grid computing can be traced back to Supercomputing 1995 conference where 17 supercomputer sites were connected together (Wilkinson, 2011). The term “grid” is adopted from the electrical grid that is used as a way to distribute electrical power according to demand. The definition of grid computing refined later to include policy issues among different organisations. The term “virtual organisation” (VO) was introduced to give more focus on the sharing of resources among different organisations. VO comprises individuals or institutions that have coordinated access of resources among institutions with the same goal (Foster, Kesselman and Tuecke, 2001).

Cloud computing which is another type of distributed computing became popular recently. Cloud computing emphasis is on virtualisation. There are some attempts to use grid computing techniques as the service provider of the cloud systems, such as volunteer cloud systems. Cloud computing can provide quality of service for grid resources via cloud resources. Other authors categorise grid and cloud differences in seven separate models, namely computation, architecture, virtualisation, monitoring, programming, application and security models (Abhishek Kalapatapu and Mahasweta Sarkar, 2011, pp. 20-22). Both grid and cloud computing model have some common characteristics in respect of using high speed networks for providing and sharing resources.

2.4.1 Middleware software

Grid computing middleware software products are computer programmes that support the sharing of resource. They facilitate the coordination of different tasks to achieve specific computing and problem solving goals. This section lists some common middleware implementations in grid computing. Grid computing middleware is not a single software product or service, but rather it is a set of software components and services that work together to provide sharing of resources to user communities. Although there are many grid middleware implementations available, in this thesis there are presented the ones that are widely used in multinational or country-specific projects. In the following subsections, there is an analysis of the selected middleware implementations. There are different types of grid middleware classification. One of these is service grid and desktop grid the middleware in service grid much more complex than desktop grids according to (Urbah *et al.*, 2009) service grids are shared infrastructure and facilities among scientific communities. Service grids have complex facilities for user authentication and authorisation. Desktop grids are simply created from desktop computers and provide large number of computers. Figure 2-8 shows the desktop grid and service grid middlewares which described in previous sections.



Middleware	Type
gLite	Service grid
Glubus	Service grid
ARC	Service grid
UNICORE	Service grid
BOINC	Desktop grid
XtremWeb	Desktop grid
OurGrid	Desktop grid
SZDG	Desktop grid
Htcondore	Desktop grid

Figure 2-8 Type of different grid middlewares.

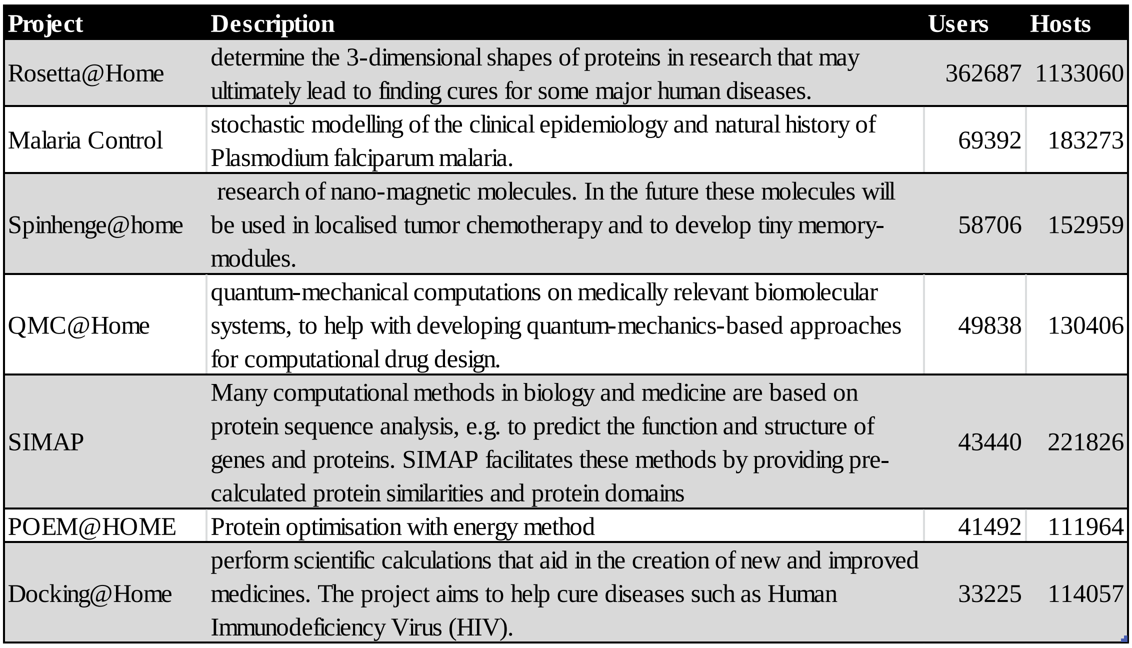
Both service grids and desktop grids have been used for scientific problem solving. The following two sections discuss advantage of desktop grid computing and some applications of desktop grid computing in bioinformatics. It is worth to mention that service grids also provide large number of services and facilities for bioinformatics. Then the middlewares are introduced in the order listed in Figure 2-8.

2.4.2 Advantages of desktop grid

Grid systems can be classified in different ways. One of the major classifications is service vs. desktop grids. Service grids provide guaranteed quality of service with small number of dedicated physical infrastructure, while desktop grids do not guarantee quality of service (QoS) but can attract large number of CPUs when they are not in use. Desktop grids do not need lots of investment and provides low cost solution for scientific research. Projects, like EDGeS, provide bridging technology to make both types of grids interoperable (Urbah *et al.*, 2009). Cloud computing can provide QoS to desktop grids. Another advantage of desktop grids over service grids is the minimum heat intensity. As one of the problems in large data centres and computer clusters is cooling and lots of energy is needed to cool down the system (Schott and Emmen, 2010), this property is really beneficial to grid systems. In order to expand the number of machines volunteers are needed in the case of biological studies, according to a survey by the international desktop grid federation (desktopgridfederation 2013), public opinion shows that some application, like medical and bioinformatics applications, can attract more voluntarily public resources to the projects than other applications. Good public opinion on the biological and medical projects is another advantage of using desktop grids in these projects.

2.5 Desktop grid applications in bioinformatics

Many bioinformatics applications applied desktop grid approach for solving computation intensive tasks. The project Folding@Home simulates protein folding and molecular dynamics. Folding@Home claimed that it is the largest distributed supercomputer (Folding 2013) with over 172,000 computational nodes participating in the project. Computational nodes include GPUs and PlayStations. Part of the software is proprietary and part is open-source. Similar to this project is the Rosseta@Home project which uses BOINC platform to submit jobs and download the results. Because BOINC platform is general purpose, many other bioinformatics applications use this platform. Currently, 13 BOINC projects are listed in the “Biology and Medicine” category of BOINC projects. Figure 2-9 provides a list of biology related projects using BOINC. The statistics derived from the BOINC popularity web site and the individual projects web sites.



Project	Description	Users	Hosts
Rosetta@Home	determine the 3-dimensional shapes of proteins in research that may ultimately lead to finding cures for some major human diseases.	362687	1133060
Malaria Control	stochastic modelling of the clinical epidemiology and natural history of Plasmodium falciparum malaria.	69392	183273
Spinhenge@home	research of nano-magnetic molecules. In the future these molecules will be used in localised tumor chemotherapy and to develop tiny memory-modules.	58706	152959
QMC@Home	quantum-mechanical computations on medically relevant biomolecular systems, to help with developing quantum-mechanics-based approaches for computational drug design.	49838	130406
SIMAP	Many computational methods in biology and medicine are based on protein sequence analysis, e.g. to predict the function and structure of genes and proteins. SIMAP facilitates these methods by providing pre-calculated protein similarities and protein domains	43440	221826
POEM@HOME	Protein optimisation with energy method	41492	111964
Docking@Home	perform scientific calculations that aid in the creation of new and improved medicines. The project aims to help cure diseases such as Human Immunodeficiency Virus (HIV).	33225	114057

Figure 2-9 some of high ranked projects related to biological research which uses BOINC information from (boincstat 2013) and individual project web sites.

2.5.1 gLite

gLite is a middleware for service grid and was developed for the EGEE (Enabling Grid for E-sciencE) project. It is built on service oriented architecture. gLite is similar to Globus and has been influenced by Globus. The main services that are provided by gLite are: data, job management, information and monitoring, security and access services (Laure *et al.*, 2004). Two main parts of the gLite architecture are the Computing Element (CE) and the Storage Element (SE). Computing Elements are any computing resources that are exposed as one entity to the grid system by grid gate interface. Storage Elements provide unified access to the storage units of the grid system. Figure 2-10 shows the job flow in gLite.

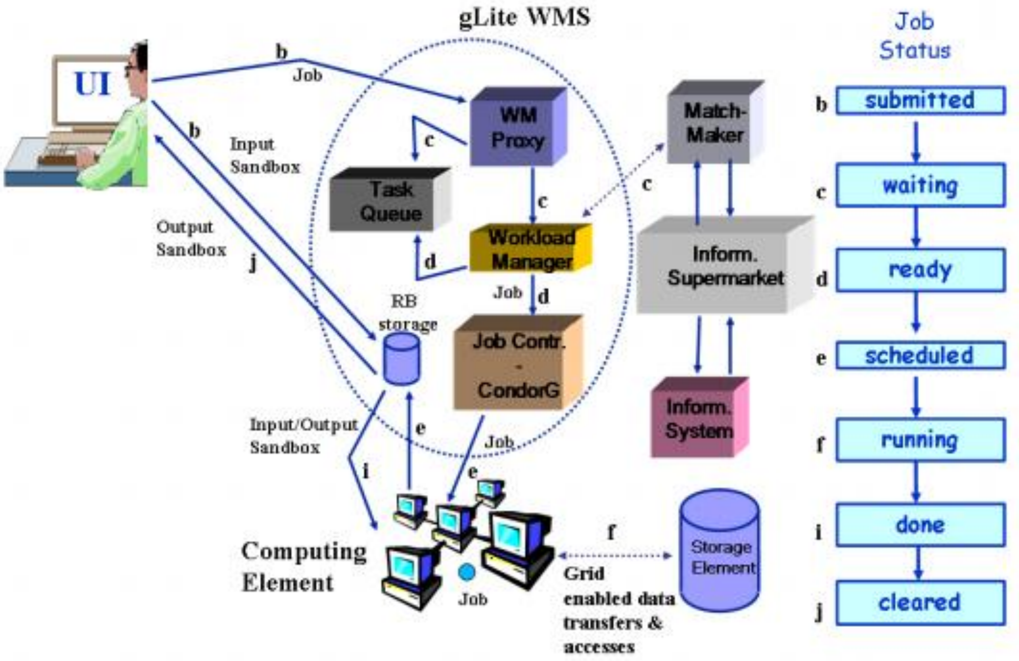


Figure 2-10 Job Flow in gLite (Burke Stephen et al., 2011)

2.5.2 Globus

Globus is an open-source grid middleware that was introduced in 1998. It has four major components. That is, security, data management, execution management and common run time (Globus, 2013a). Figure 2-11 shows the level components of the fifth version of Globus Toolkit (GT). GT provides tools to create grid environment. Recently Globus provides facilities to deploy fully functional Globus on Amazon EC2 cloud with Globus provision (Globus, 2013b). GT can be used to produce service grid and guaranteed quality of service.

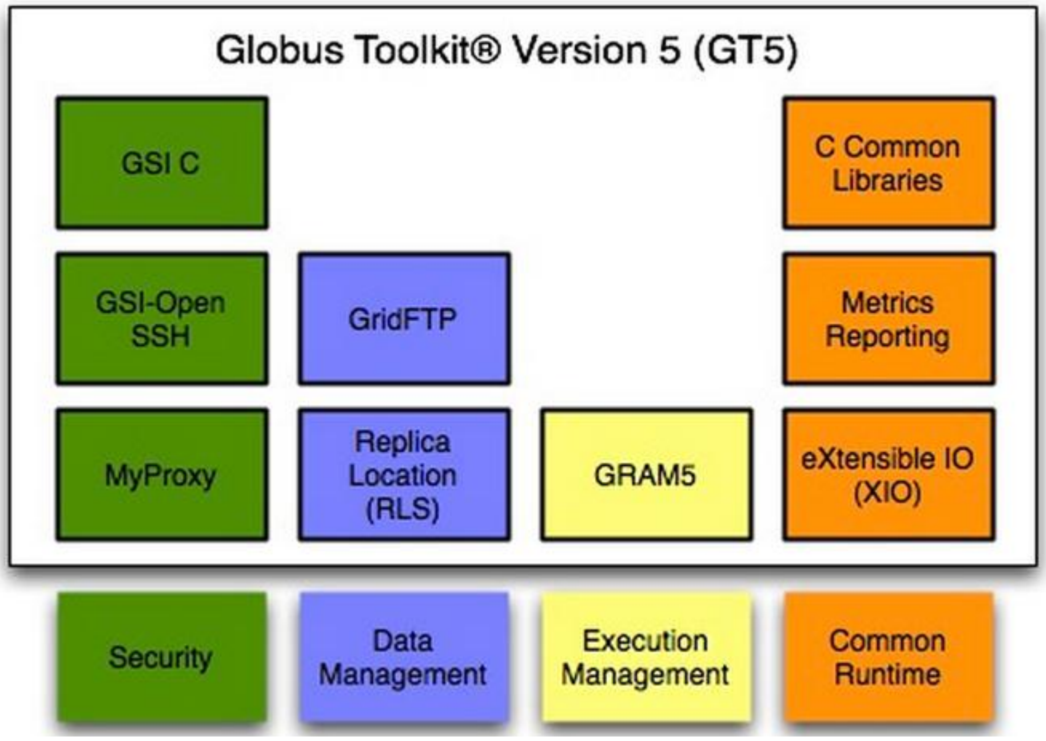


Figure 2-11 Globus High level services (globus project 2013)

2.5.3 ARC

ARC (Advanced Resource Connector) middleware was developed in 2002 to meet user demand of NurdoGrid user community. The design of ARC architecture was mainly influenced by requirements of Nordic ATLA user community. New versions of ARC gradually adopted service oriented architecture and provide their functionalities as web service. ARC, together with other grid platforms, is part of the EMI (European Middleware Initiative) software release (Ellert *et al.*, 2007).

2.5.4 UNICORE

UNICORE (Uniform Interface to Computing Resources) was first developed in 1997 to provide a uniform way of access to different resources that were shared among heterogeneous hardware and software facilities. UNICORE addresses issue of different policies among computer centres (Erwin, 2002). UNICORE architecture is three-layered. These three layers are the client layer, the system layer and the service layer, as shown in Figure 2-12. UNICORE is open-source software and is implemented in the

Java programming language. This middleware is an example of service grid. It has been used in different fields of scientific projects. UNICORE requires special administration for installation and support.

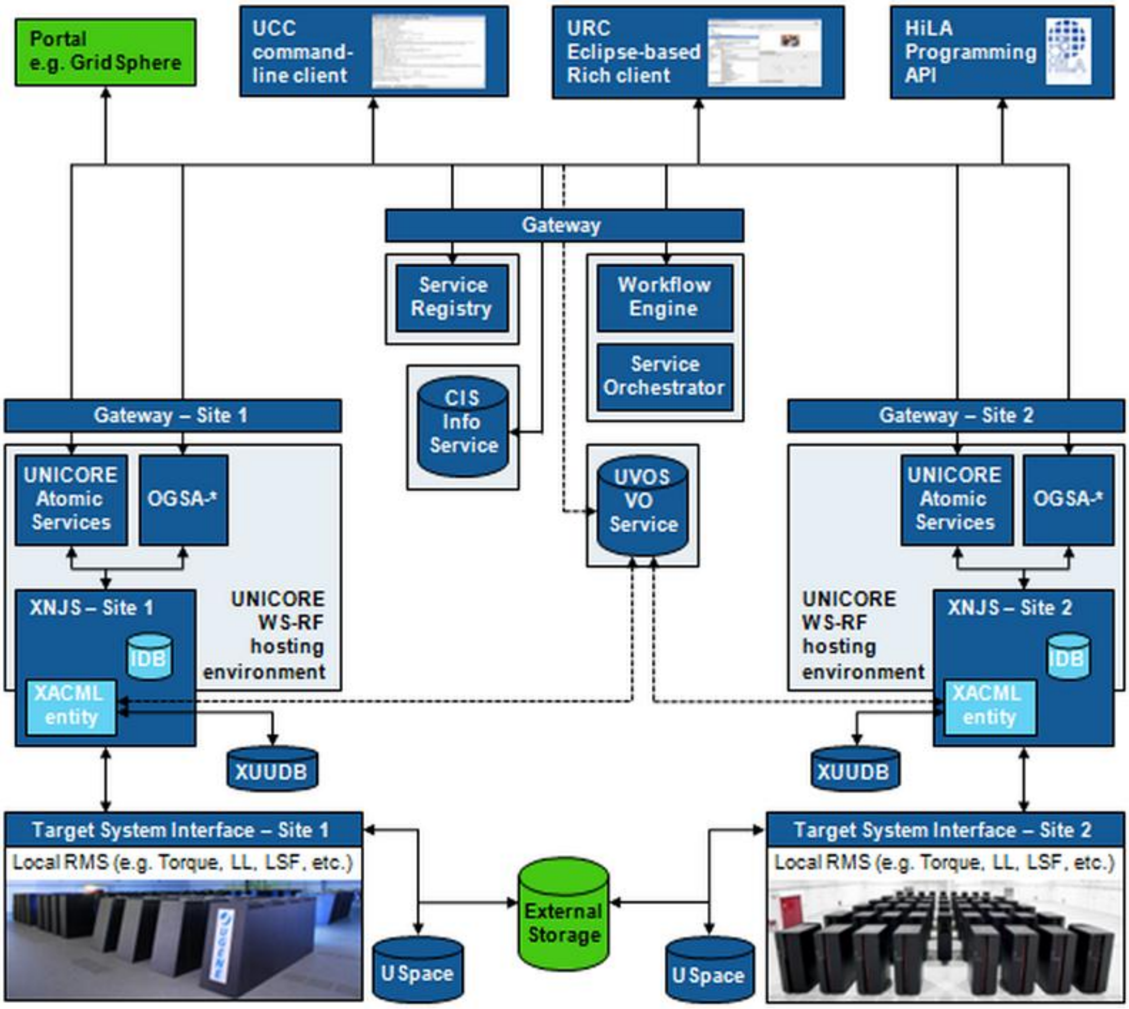


Figure 2-12 Architecture of UNICORE source (unicore project 2013)

2.5.5 BOINC

BOINC (Berkeley Open Infrastructure for Network Computing) is open-source software that was first developed for the SETI@HOME project and later became available as a general purpose software for projects that need computing power. BOINC project can be classified as a client-server system. Many research projects that require intensive computing resources use BOINC. List of biology related projects provided in Figure 2-9.

In BOINC, jobs are handled as follows: the *feeder* adds new work units to the *scheduler*, the *scheduler* handles the requests of the BOINC client for new work units. Whenever results are produced, a validator checks these results for redundancy, which is frequently the case. The task of the *validator* is to compare these redundant results for the same work unit and choose one of them. The *assimilator* then stores the results in the desired output file directory or parses the results to desired format and stores them in a database system. All of the aforementioned components are running as *daemons* in the server. The *feeder* and the *transitioner* are application independent. *Transitioner* observe the state of workunits and look if the state of workunits changed it may create new results based on the state or tag the workunits with error. *Work generator*, *assimilator* and *validator* are generally application specific. There are simple codes available which can be modified for assimilation, validation and work generation tasks. *File deleter* is another daemon which is used after results are returned back from clients assimilated. Normally file deleter is application independent and deletes input and output files after they finished and assimilated. These programs should be listed as *daemons* in each project configuration file. Other optional programs can be added as daemon for specific purpose. Jobs are described in template files. These files are in xml file format and describe input, output and other job parameters (Anderson, 2004),(Ries, Schroder and Grout, 2011). Figure 2-13 shows BOINC component relationship both for client side and server side. Components in the rectangle are server side components.

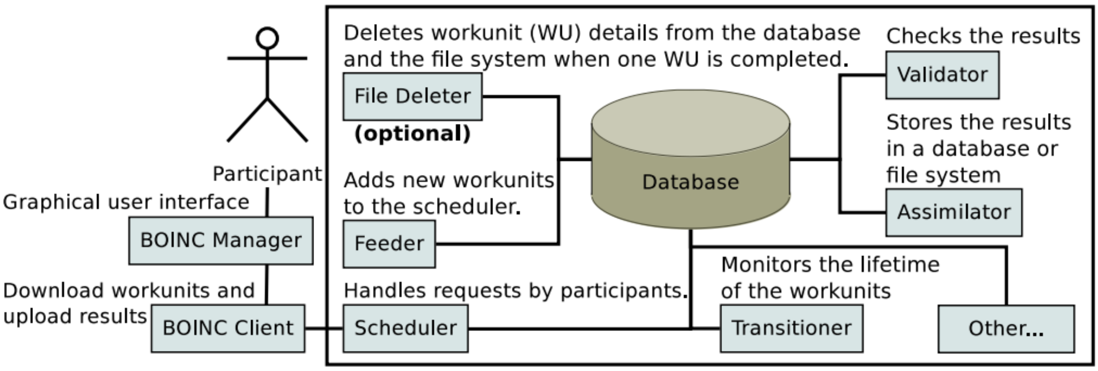

Participant
Graphical user interface
BOINC Manager
Download workunits and upload results
BOINC Client
Deletes workunit (WU) details from the database and the file system when one WU is completed.
File Deleter
(optional)
Adds new workunits to the scheduler.
Feeder
Handles requests by participants.
Scheduler
Database
Checks the results
Validator
Stores the results in a database or file system
Assimilator
Monitors the lifetime of the workunits
Transitioner
Other...

Figure 2-13 Interaction between different BOINC processes and database processes in the rectangle are server side components. (Ries, Schroder and Grout, 2011)

2.5.6 XtremWeb

XtremWeb is another desktop grid middleware. The main aim of the project was to develop a large scale distributed system similar to BOINC but, unlike BOINC, it uses decentralised peer-to-peer approach to achieve this gaol (Cappello *et al.*, 2005). XTREMWEB-CH is an improved version of XtremWeb middleware for peer-to-peer desktop grid computing. Main components of XtremWeb are: coordinator, warehouses and workers. All communication to the coordinator starts by workers using SOAP over HTTP. Files are retrieved from the warehouse component and the produced results are stored in them too (Abdennadher and Boesch, 2005). There is also a program to easily deploy worker nodes on the Amazon cloud (Abdennadher, 2012).

2.5.7 OurGrid

OurGrid is an open source middleware for grid computing based on P2P architecture. The project started at 2004. It is designed to speed-up jobs that need bag-of-tasks application. Bag-of-tasks jobs are jobs which are independent of each other and can run in parallel. OurGrid has four main components: broker, worker, peer and discover service. Broker is where a user can submit jobs and monitor them. Similar to HTCondor, jobs are described in the (JDF) file. Workers are execution machines. Peers are machines that determine which workers are available to run jobs and, in other words, they are worker providers. Discovery service connects multiple sites. In Figure 2-14, the architecture of OurGrid is depicted (OurGrid 2013).

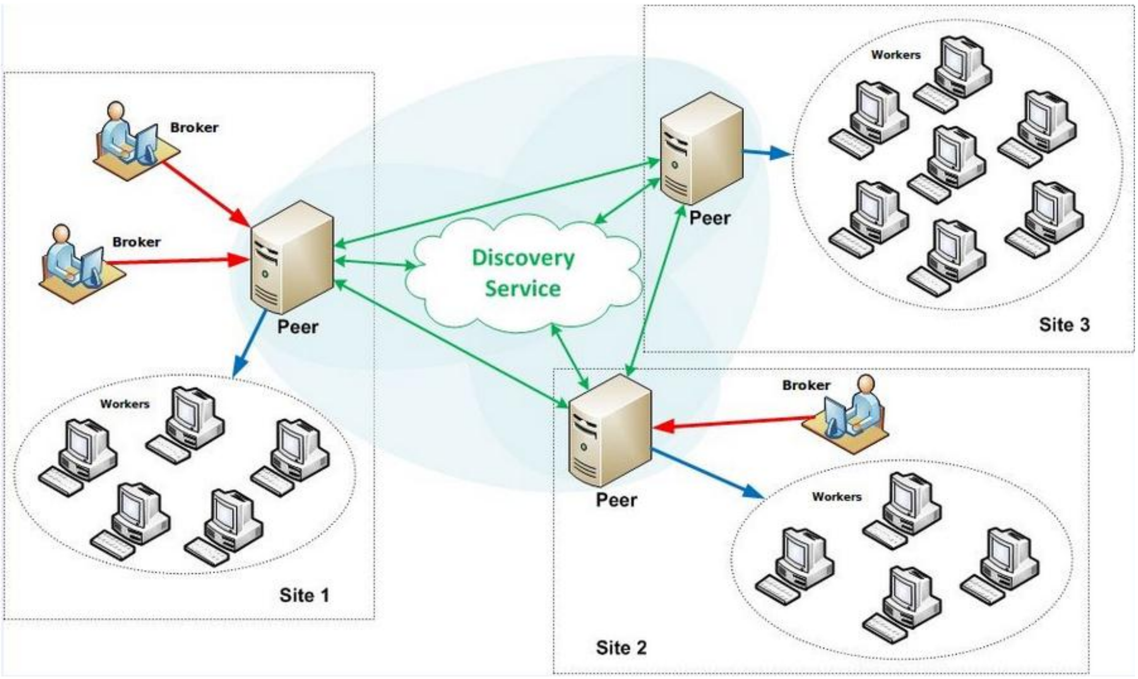


Figure 2-14 OurGrid high level architecture (OurGrid 2013)

2.5.8 SZDG

SZDG is an extension of the BOINC system. New components have been added to BOINC in order to make the BOINC projects interoperable and provide scalability to the system. Among the most important components are the 3g-bridge and the gUse desktop grid submitter. 3g-bridge, in SZDG context, provides connection of WS-PGRADE portal to 3g-bridge. Submitter’s task is to generate unique ID for jobs, place the information about input and output files in 3g-bridge database, and check the status of jobs periodically. Figure 2-15 shows how the different parts of SZDG work together (Kacsuk *et al.*, 2009), (Taylor *et al.*, 2011).

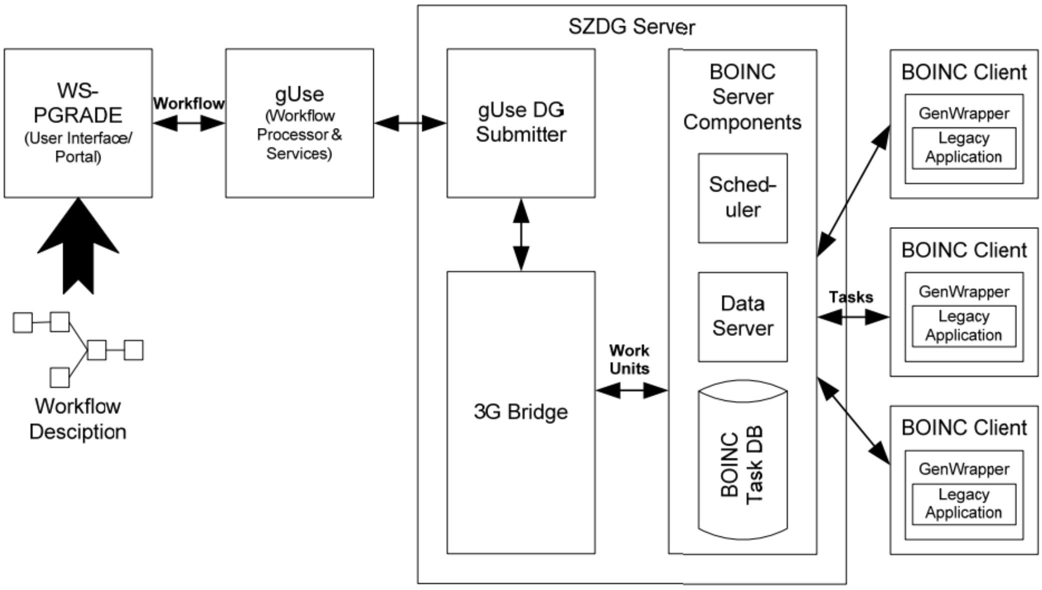


Figure 2-15 SZDG architecture (Taylor *et al.*, 2011)

2.5.9 HTCondor

HTCondor system was developed in 1984 at the University of Wisconsin, previously named as condor until 2012. HTCondor middleware harness the processing power of Idle CPUs to run jobs. Jobs are matched to machines according to their specifications which are defined by the job submit file. HTCondor job scheduling can be used as in master-worker paradigm or directed acyclic graph manager. Figure 2-16 shows the main processes that are involved in running jobs on HTCondor (Thain, Tannenbaum and Livny, 2005).

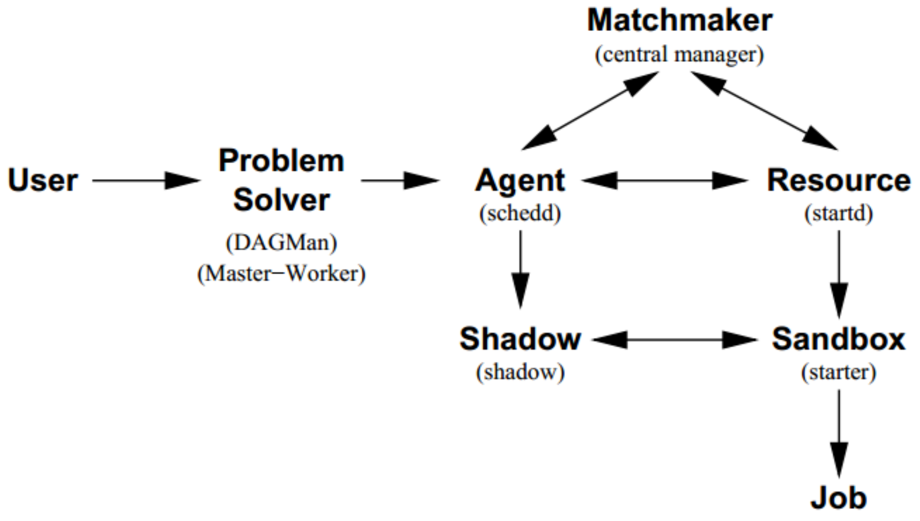


Figure 2-16 Main processes in HTCondor system (Thain, Tannenbaum and Livny, 2005)

2.6 Workflow Systems

Scientific workflows are directed graphs consisting of nodes and edges. Nodes represent tasks and edges represent dataflow, or control-flow, which directs the execution steps among tasks (Mehta *et al.*, 2012). Tasks, which are dependent on each other, can be local single programs, remote web services etc. Also, there can be embedded sub-workflows (Tiwari and Sekhar, 2007). Workflows can be shared among the user community, so users can modify and customise them. Furthermore, they may be included in other workflows. Because there are different workflow formats, projects like SHIWA (SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs 2013) investigate ways to connect those diverse formats.

Workflow systems became essential part of bioinformatics research as the number of tools and resources expanded in scientific research and discovery (Mehta *et al.*, 2012). They enable scientists to combine various tasks together to accomplish the study. Another reason for employing workflows in bioinformatics is the massive amount of data and tools. Having huge amount of accumulated sequencing data and ever expanding number of tools available for analysing this data, workflows enable researchers to create and modify new experiments with new data and tools as soon as they become available (Missier *et al.*, 2010).

The system developed in this work uses workflows for feature generation and feature subset selection. Using workflow in feature generation makes the system easier to develop and maintain. Further, the addition of new components to the existing ones does not change other parts of the system. The following sections examine the different types of workflow systems.

2.6.1 Taverna

Taverna is a workflow management system best known for its use in life sciences applications. It supports data-intensive models and service-oriented workflows Although it has been used in the grid environment, it is not created in a way to use heterogeneous grid services. Therefore, a plug in needs to be installed so it can be used in the target grid environment (taverna 2013). Workflows can be downloaded from myExpriment website directly in Taverna workflow management system for sharing and reusing with other research community members. Users can register in myExpriment website to use the social networking facilities to access and reuse workflows (Goble *et al.*, 2010), (Missier *et al.*, 2010).

2.6.2 Galaxy

Galaxy is a web based system which provides facilities for interactive data analysis. Data can be sent from UCSC table browser to galaxy or imported in a file. Common procedures are applied for achieving specific outputs. Customised galaxy web sites with specific applications provide target users with the most popular applications in target user community. Galaxy workflows can be shared with other users in galaxy website. Galaxy can be easily deployed on amazon EC2 cloud (Goecks *et al.*, 2010).

2.6.3 Kepler

Kepler is an open-source java-based workflow engine. The important aspect of Kepler workflow is its actor-oriented design. Actors are independent computation nodes which can be a web services, database queries etc. Kepler uses its own mark-up language for modelling (Altintas *et al.*, 2004a). It can use grid technology as well as native support for parallel processing (Altintas *et al.*, 2004b). Kepler is used in the Clothocard project which is a design environment for synthetic biology systems. It is also used in the real-time environment for analytical processing (REAP) project which combines other projects and use sensor data (Kepler, 2013).

2.6.4 Jopera

Jopera for eclipse is a service composition tool for creating distributed applications using web services or other computational blocks. Grid services or java and JavaScript can be called in workflow nodes. Jopera is used by a number of grid systems, like TeraGrid, and in climate modelling (Pautasso, Heinis and Alonso, 2006) (Chang *et al.*, 2008).

2.6.5 SQLshare

This workflow tool is a web based application that mainly focuses on querying conventional databases. It has been shown in (Howe *et al.*, 2013) that it is useful and can be used as a competitor to other scientific workflow systems.

2.6.6 gUse

gUSE is an open-source distributed computing infrastructure (DCI) gateway environment that enables users to access different DCI, like cloud and grid. It provides services to monitor different grid environments. One of the most important services the system provides to user is its workflow engine that supports different types of workflow designs and merges computational nodes to accomplish specific procedures (Kacsuk *et al.*, 2009) (Kacsuk *et al.*, 2012).

User friendly environment can be developed for specific scientific communities by application specific module API and Remote API. So, users do not need to know details of workflow design. Users may only see the web page forms to submit jobs and access the results.

gUSE is currently used for creating 22 scientific web portals. Five of these portals are specifically designed for bioinformatics and computational biology user communities. It is also used in other areas, like multimedia and animation. The highest number of registered users is currently in the field of animation with over 13,000 users in the RenderFarm.fi. This system is connected to BOINC platform (Available science gateways 2013).

2.7 DNA methylation data sources

2.7.1 Published papers

There are different resources for DNA methylation data. The least automation-friendly data sources are published papers, in which DNA methylation data is depicted in graphs. For comparison of the data, each CpG site should manually mapped to genome coordinates and the level of methylation is determined by examining the graph. Text mining methods can be used to find published papers on particular subjects. However, there is a need to manually convert data to machine readable format. Figure 2-17 shows three different examples of presenting CpG methylation data.

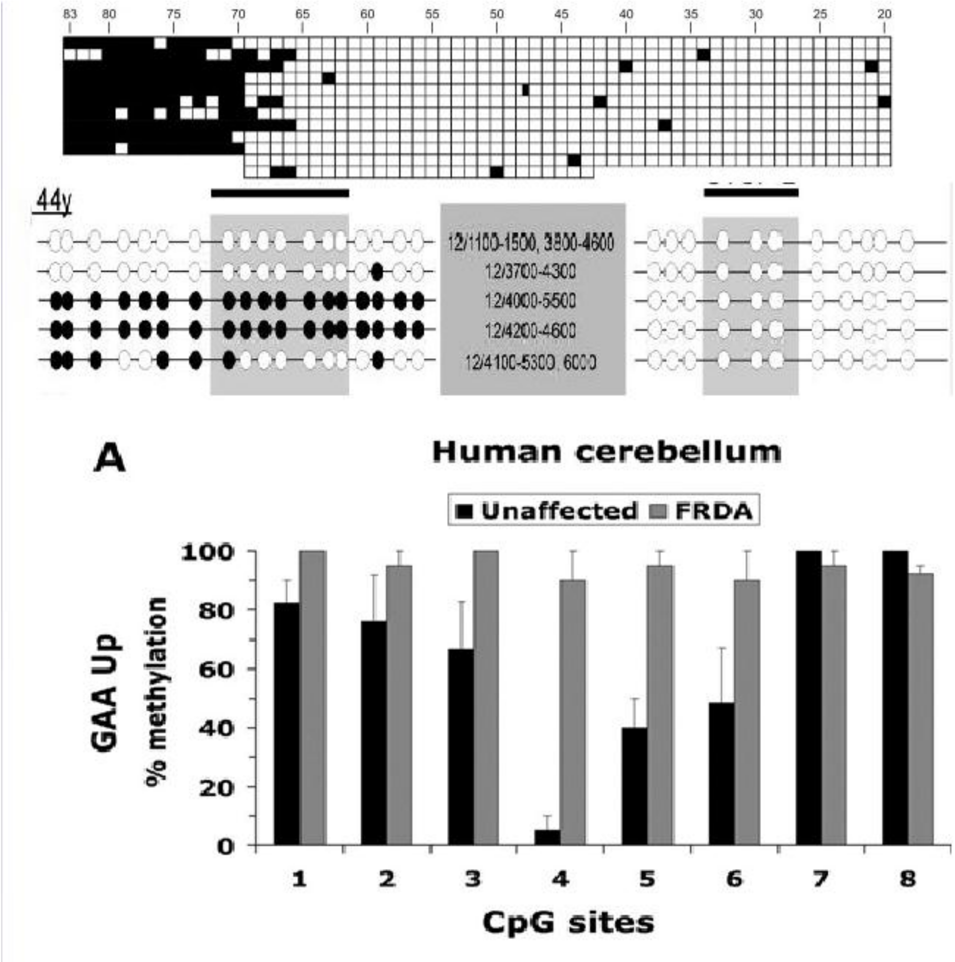


Figure 2-17 Different ways of presenting DNA methylation from top FMR1 methylation, DM1 methylation, FXN CpG sites methylation.

There are not yet standard ways of storing this methylation data to make the comparison easier among different types of diseases or characteristics, like tissue, cell type etc.

2.7.2 General databases

Alongside published papers, there are public databases which can be used to acquire DNA methylation data. Some of these databases are general databases and some are specialised in DNA methylation. Microarray data the most automation-friendly format and is used to get new knowledge on DNA methylation. In microarray data, the CpG sites are annotated with additional information and sequences. Table 2-1 lists the main general purpose databases.

Table 2-1 List of general purpose public databases

Name	Description	Address
EBI ArrayExpress	ArrayExpress is a database for array or sequence based data maintained by European bioinformatics institute.	http://www.ebi.ac.uk/arrayexpress/
GEO (Genome Expression Omnibus)	GEO is a database for array and sequence based data with nearly one million samples for 11,716 platforms.	http://www.ncbi.nlm.nih.gov/geo/
The Cancer Genomic Atlas (TCGA)	Includes DNA methylation data on different platforms as one of different data types provided.	https://tcga-data.nci.nih.gov

2.7.3 Special purpose DNA methylation databases

Some databases are specially keeping DNA methylation data. They might integrate data from various sources like GEO and ArrayExpress. In some cases they are gathered data from literature by text mining methods.

Table 2-2 DNA methylation databases

Database name	Description	Website
MethDB	DNA methylation content on various tissue and species.	http://www.methdb.de/
MethylomeDB	Methylation profile of CpG site in human and mouse brain.	http://www.neuralepigenomics.org/methylomedb/
DisEaseMeth	Gathered information from various sources and platforms mainly kept the information on the probe that is located in promoter sites.	http://202.97.205.78/diseasemeth/
MethyCancer	Gathered DNA methylation data from various sources for cancer.	http://methycancer.psych.ac.cn/
PubMeth	PubMeth is a database for DNA methylation in cancer. It uses text mining in literature and manually curates and annotates the data after text mining.	http://www.pubmeth.org/

2.8 Feature generation

The main purpose of feature generation is to define characteristics of data on some measurement. So, if there are n different data points and m different features for each data item, tabular data matrix of size n*m can be generated. This table then will be used in data mining methods. This data matrix can be later used to find out how well all features can classify objects or which feature subset better classifies data items. Sometimes features can be categorized into different groups on some criteria. These groups can be independently used in classifier. Classification performance on each group can indicate the importance of each feature group in the study. In the case of CpG sites, each CpG site represents one data item and the sequence around the CpG site is used for generating features. This approach has been used for CpG island for normal cell and tissue specific. Features can be divided into different groups. Examples of these feature groups are structural, regulatory (i.e., transcription factor), or only based on the sequence (i.e., motifs and word in the sequence). In this section different methods for generating features by sequence are reviewed.

2.8.1 MEME (Multiple EM for Motif Elicitation)

MEME is an approximation algorithm to identify motifs in a set of DNA or protein sequences. The method used in MEME is expectation maximisation to fit two component mixture models to the set of sequences. MEME gets the number of sequences and the width of sequence as input. The number of motif distribution in a DNA sequence can be determined with three options, i.e., any number repetitions (ANR) of motifs, one per sequence (OPS), or zero or one per sequence (ZOOPS). Identified motifs are sorted by the parameter E-value. This parameter shows expectation of finding similar motif in a random sequence. DNA sequence in each class can be used as an input of the MEME suit (Bailey *et al.*, 2006).

2.8.2 MAST (Motif Alignment & Search Tool)

The results of MEME software can be used in the MAST program with all sequences together to identify the number of hit for each motif in all sequences. The produced results can be used to create a feature matrix where each row represents the sequence and each column represents motifs. Entries of the matrix are the number of hits of each motif in the sequence (Bailey *et al.*, 2009). This approach has been used in (Feltus *et al.*, 2006) for two classes of methylation-prone and methylation-resistance sequence

2.8.3 Word Composition

Generating all possible permutation of four letters C,G,A,T (word) and counting the number of occurrences of each word by moving the sliding windows over sequence is another technique that is used for feature generation in DNA methylation. This can be done by applications like “wordcount” in EMBOSS. Extension of this approach, as will be discussed in chapter 3, can be used to allow wildcard in the word and generate more features. The number of features is 4^N, where N denotes the word size. This approach has been used in previous studies on DNA methylation for 2, 3, 4, and 5 base pair (Lu *et al.*, 2010) (Previti *et al.*, 2009).

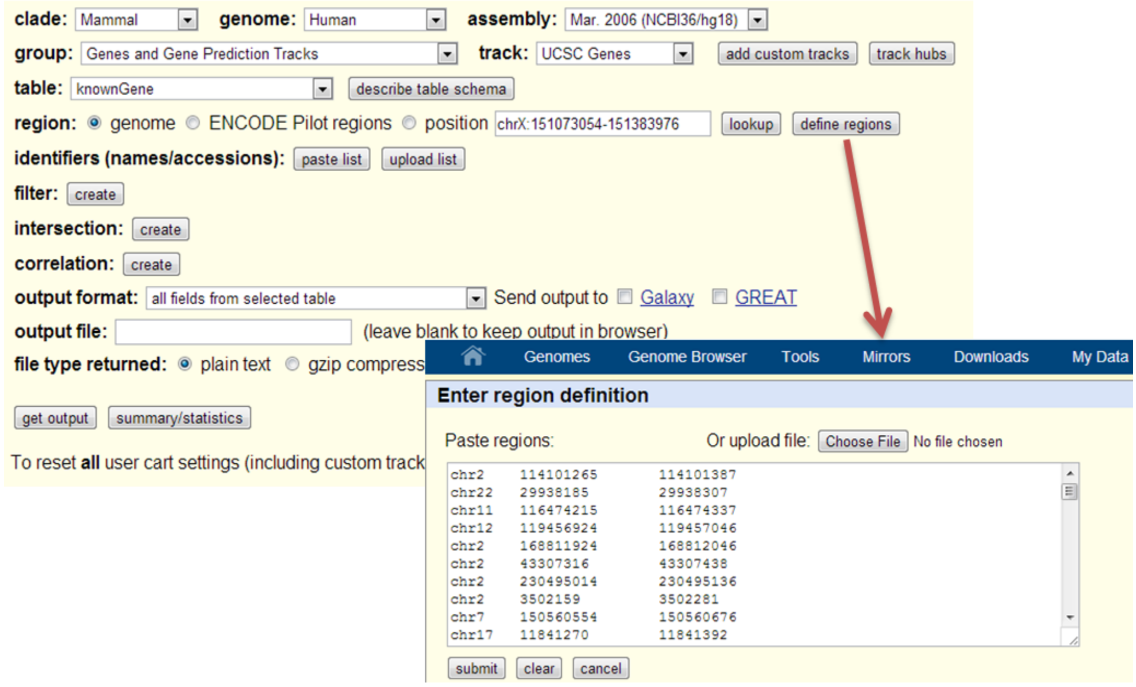
2.8.4 EMBOSS

EMBOSS is a set of different binaries commonly used to analyse DNA and protein sequences (Rice, Longden and Bleasby, 2000). Some applications in EMBOSS software suite are used for generating feature like “banana” and “btwisted” which use DNA sequence as input and generate structural feature of DNA sequence .

“Banana” can predict bending of a normal DNA double helix. Banana is used by Previti et al. for feature generation. “btwisted” calculates overall twist of the DNA sequence and the stacking energy. Similar to banana, it is used by Previti et al for structural feature generation (Previti *et al.*, 2009). In this research, “jaspscan” is used to generate feature matrix on transcription factor binding sites in JASPER database and “wordcount” is used for generating the number of words in the DNA sequence. Each of these programs outputs should be modified by an external application to make them usable for classification. Further details are provided in chapter 5.

2.8.5 UCSC table browser

Track files stored in UCSC tables can be queried by table browser tool to investigate whether specific genomic track has an overlap with the identified CpG sites. The number of overlaps or the size of overlaps has been used as feature of specific DNA sites. Regions which are required to be investigated should be defined in the BED browser extensible format. They normally have three fields in each line, namely chromosome number, start site and end site of the sequence. Specific queries can be applied on the track using filter option. Additionally, intersection to other tracks can be defined in the form, as shown in Figure 2-18. The results can be exported to galaxy main website (Karolchik *et al.*, 2003), (UCSC Browser 2013).



clade: Mammal genome: Human assembly: Mar. 2006 (NCBI36/hg18)

group: Genes and Gene Prediction Tracks track: UCSC Genes

table: knownGene

region: ● genome ○ ENCODE Pilot regions ○ position chrX:151073054-151383976

identifiers (names/accessions):

filter:

intersection:

correlation:

output format: all fields from selected table Send output to □ Galaxy □ GREAT

output file: (leave blank to keep output in browser)

file type returned: ● plain text ○ gzip compress

To reset **all** user cart settings (including custom track

Enter region definition

Paste regions: Or upload file: Choose File No file chosen

```
chr2    114101265      114101387
chr22   29938185       29938307
chr11   116474215      116474337
chr12   119456924      119457046
chr2    168811924      168812046
chr2    43307316       43307438
chr2    230495014      230495136
chr2    3502159        3502281
chr7    150560554      150560676
chr17   11841270       11841392
```

Figure 2-18 USCS table browser form and define regions form (UCSC Browser 2013)

2.9 Machine learning

Some problems don’t have known solutions, for example predicting the weather situation from previous measurement. We don’t know the algorithm for predicting the weather, but there are large number of examples, like days and their condition that can be used to predict the future weather. In these kinds of tasks we try to use the characteristic of one day weather condition to predict future condition. These types of problems are called supervised learning problem because we know which the category of weather example. Regression problem are type of supervised learning problems which there are real values instead of categorical values and we want to find the functions to estimate the value of one variable for example the amount of milk particular cow will produce based on the attributes and characteristic of the cow. (Hand, Mannila and Smyth ,2001a p. 13). Another example is estimating the price of a car given its attribute like its age brand engine etc. (Alpaydin 2010)

Another type of methods unlike supervised learning we only have input data and want to find some kind of pattern in the data. For example there are customers and we have their buying patterns we want to group similar customers according to their attribute this may provide useful information for the manager of the company to better approach those customers. There are many implementation of machine learning algorithms WEKA (Mark H et al. 2009) is an open source package written in java which is used in this research in another setting other method and tools can be used without major difference in the results.

2.10 Classification methods in DNA methylation analysis

The classification is process of using existing data or set of examples to predict the class of those examples, each example associated with some features. Lets assume there are set of days with information about the temperature, humidity and wind speed and each day is classified as either sunny or rainy. The classification algorithm should be able to discriminate between the days using different measurements. This gives us the model to predict the class of new days. The model could also provide insight about the weather for example which measurement can more precisely separate sunny day from rainy day.

Classification methods are used in this research. These methods create predictive models that can be applied to new instances. Classification methods are applied to evaluate how precisely all features can classify CpG sites and which features class or subset of features can classify CpG sites. These models can give insight on the factors that need more investigation. Classification algorithms have been used in (Feltus *et al.*, 2006) (Previti *et al.*, 2009) to classify CpG islands. There are many classification methods. For example, in the WEKA (Mark H et al. 2009) package there are more than seventy different classification methods. Here, we use some of the main classification methods. Each of these methods is in three main subfamilies of classification methods and is widely used for classification in the DNA methylation analysis.

2.10.1 Problem definition

Set of examples or instances can be seen as the matrix, in this matrix each row represents the instance and each column represents the feature related to that instance, one of the columns is the class variable or categorical variable (Equation 2-1). The last column in the matrix contains values for the class variable. The class variable values belong to one of the *K* number of classes. Here there are *m*-1 measurements related to each instance.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & c_{1m} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & c_{nm} \end{bmatrix}$$

$$c_{ij} \in \; C = \{C_i | i = 1, \dots, K\}, \; a_{ij} \in \mathbb{R} \; i = 1, \dots., n, j = 1, \dots. m - 1$$

Equation 2-1

The purpose of the classification algorithm is to learn the model from training data in this case the input matrix, and predict the class of new instances , so we want to find mapping f of measurements to class variable .

$$f : \mathbb{R}^n \mapsto C \quad Y = f(A), A = (a_{i1}, \dots, a_{in}) \; Y \in C = \{C_i | i = 1, \dots, K\}$$

Equation 2-2

In the case of CpG site dataset there are four classes of CpG sites and 740 measurements. Each section of the matrix is generated by four different applications.

2.10.2 Tree models

These models have been used by (Witten, Frank and Hall, 2011) to classify CpG islands based on their feature. The basic idea behind tree models is to split the feature space in a way that the majority of CpG sites in each divided cell belongs to one class. This process continues recursively until some stopping point. For example, one stopping point is the time that all data points in a node belong to one class. Tree models can visualise data points and can be easily understood. The basic form of tree models is the binary form where each branch can be tested on the value of feature. C4.5 is one of the most popular tree models in data mining. C4.5 algorithm has been described by Witten et al, and works as follows: “First, select an attribute to place at the root node, and make one branch for each possible value. This splits up the example set into subsets, one for every value of the attribute. Now the process can be repeated recursively for each branch, using only those instances that actually reach the branch. If at any time all instances at a node have the same classification, stop developing that part of the tree." (Witten, Frank and Hall, 2011, p. 99). J48 is WEKA (Mark H et al. 2009) implementation of C4.5 algorithm.

2.10.3 Support vector machine

This method assumes that data points can be split for each class and tries to find the hyperplane that maximises the distance between data points. The data points that are in the margin of hyperplane in each class are called support vectors. Figure 2-19 shows how the separating line, from all possible lines, separates two classes (Tufféry, 201, p. 503).

Let's say that we have training data we want to maximise the distance between two datasets in doing so we want to find the hyperplanes which their equation can be given by:

$w.x + b = 1 \; and \; w.x + b = -1$

We can prove that the distance between two hyperplanes can be found by following equation:

$$D = \frac{|b1 - b2|}{||w||^2}$$

In the equation the ||w|| is the norm of the vector perpendicular to hyperplane. In order to maximise this distance we can minimise the $||w||^2$, the following constraints should be applied as well.

$$w.x + b > \; 1 \; and \; w.x + b < -1$$

The full proof and more discussion can be found in the (Alpaydin 2011).

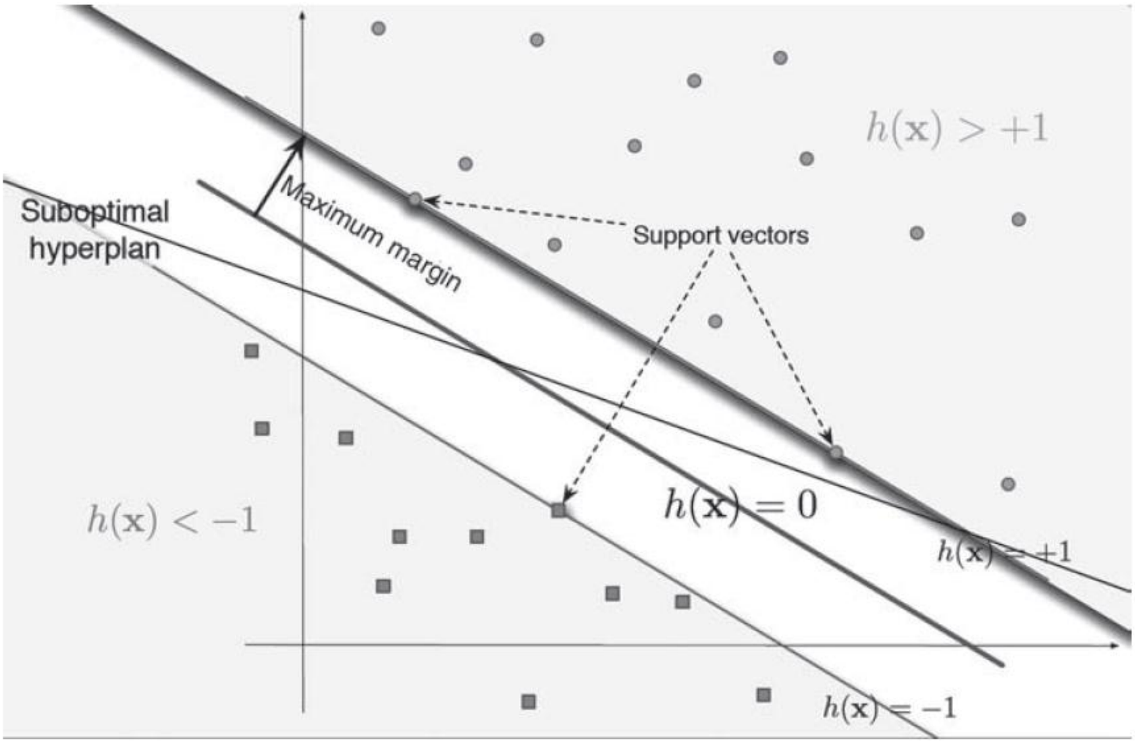


Figure 2-19 shows support vector that separate dots from squares with hyperplane (Tufféry, 201, p. 503).

2.10.4 Naive Bayes classification

Naive Bayes is a classification method which uses Bayes Theorem in probability theory. In this classification there is an assumption that all features are independent of each other. This is called first order Bayes assumption. By using this assumption, it is possible to approximate the probability by product of probability of each feature per class (see Equation 2-3).

$$P(F|c_k) = P(f_{1,} \ldots\ldots, f_n \; |c_k) = \prod_{i=1}^{n} p(f_i|c_k) \quad 1 \leq k \leq n$$

Equation 2-3

In this equation, $p(f_i\ |c_k)$ is conditional probability of each feature for the *k*th class denoted by c_k and f_i is the feature number i . In some situation the naïve assumption can be modified. This allows some assumption on the features dependency and the probability of two pairs of feature is calculated instead of one. These modifications provide little improvement to the classification prediction (Hand, Mannila and Smyth, 2001b).

2.10.5 Neural Networks

Neural networks are inspired from the human brain structure. In neural networks connected networks of neurons are used to generate prediction function. In neural networks input and output are connected together with layers of internal units then these units are adjusted in the loop in the way that they give good prediction performance. One example of neural networks is back propagation network which use the outcome of prediction and use error to adjust the internal layers of networks (Russell and Norvig, 2010). The simple perceptron is depicted in the following figure multi layer perceptron contains more than one layer of these weights, full description and discussion provided in (Alpaydin ,E 2010) output is calculated by Equation 2-4.

$$\boldsymbol{y = \sum_{i=1}^{n} w_i x_i + w_0}$$

Equation 2-4 (Alpaydin , E 2010)

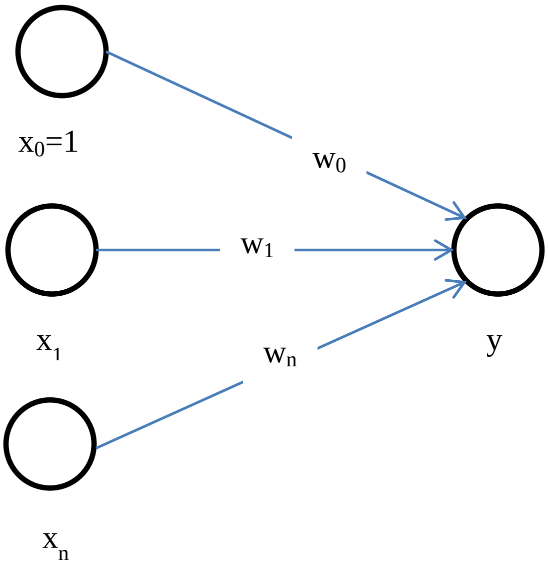


Figure 2-20 simple perceptron (Alpaydin E 2010)

2.11 Measurement of prediction outcome

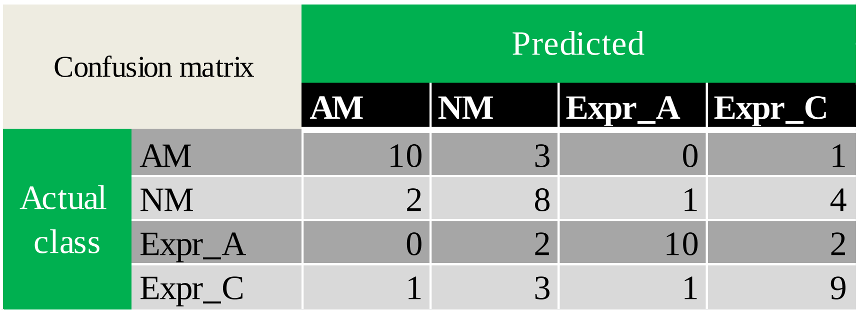
In this section metrics which is used to evaluate the classification outcome are discussed. In order to evaluate the classification we need training and test set. Cross validation is one of the methods to evaluate the classification which is first described in section 2.11.1. Section 2.11.2 describes confusion matrix, section 2.11.3 describe true positive rate, precision and f-measure. Finally the last two sections describe area under ROC curve and kappa.

2.11.1 Cross validation

The evaluation method used in this research was a 10-fold cross validation on each classification method. In this method, the data was divided into 10 groups. 9 groups are used for training. The remaining group is kept for the testing of the model.

2.11.2 Confusion matrix:

Each classification result contains the instances for the predicted class. The predictions for these classes in some cases were not correct. A matrix can be generated to contain the information of the result of the predictions. This matrix is called the confusion matrix. An example of a confusion matrix is shown in the Figure 2-21 The entries on the main diagonal shows the correct prediction, if these numbers are high for all entries when compared to other entries, it shows a good classification result. For example class AM instances were correctly predicted by the model in ***(10/ (10+3+0+1)) 71***% of cases and class Experiment C in 9/ (1+3+1+9) ***64%*** of cases and etc.



Confusion matrix		Predicted			
		AM	**NM**	**Expr_A**	**Expr_C**
Actual class	AM	10	3	0	1
	NM	2	8	1	4
	Expr_A	0	2	10	2
	Expr_C	1	3	1	9

Figure 2-21. An example of confusion matrix for four classes of CpG site methylation.

2.11.3 The true positive rate, precision and F-Measure

The true positive rate of a class is the first measure in evaluating the classification result. This measurement is determined by the number of correctly classified instances divided by all the instances (Equation 2-5). The Figure 2-22 shows that the model can better predict instances in AM and Experiment_A than other two classes.

Equation 2-5

$$TPR = \frac{Correctly\ predicted\ Instance\ in\ class}{All\ instance\ in\ class}$$

On the other hand precision is the number of instances truly predicted as one class over the entire instance predicted to be in the same class. This corresponds to sum of each column in the confusion table see Equation 2-6.

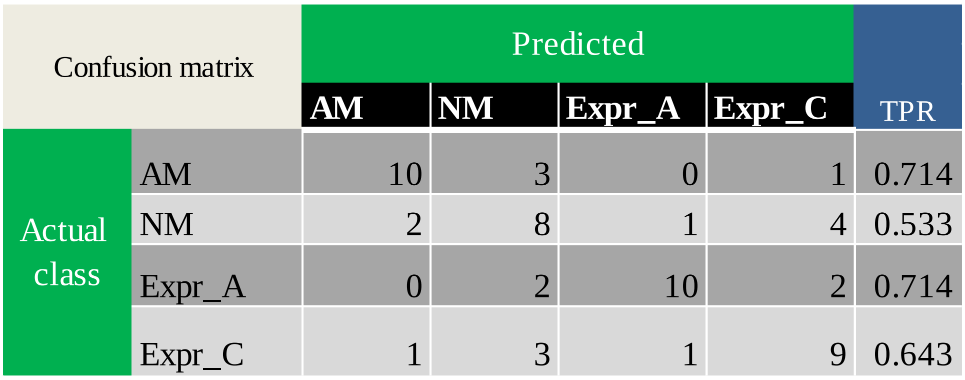
Equation 2-6

$$Precision = \frac{Correctly\ predicted\ Instance\ in\ class\ X}{All\ instance\ Predited\ as\ class\ X}$$

The F-measure is the combined measurement of precision and true positive rate. The best value is 1 and worst is 0, similar to the true positive rate and precision. Equation 2-7 shows how the F-measure was calculated.

Equation 2-7

$$F - Measure = 2.\frac{Precision.TPR}{Precision + TPR}$$



Confusion matrix		Predicted				TPR
		AM	NM	Expr_A	Expr_C	
Actual class	AM	10	3	0	1	0.714
	NM	2	8	1	4	0.533
	Expr_A	0	2	10	2	0.714
	Expr_C	1	3	1	9	0.643

Figure 2-22 shows the true positive rate calculation from confusion matrix from Figure 2-21.

2.11.4 The area under ROC curve

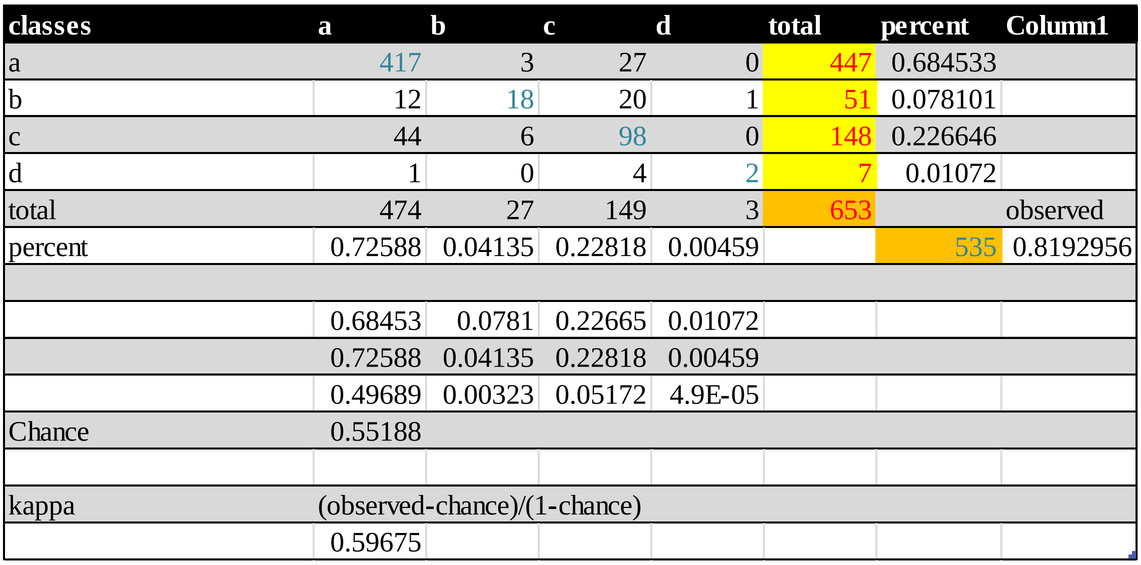
The area under the ROC (receiver operating characteristic) curves is another measurement that can be used to evaluate the performance of the classification. ROC curves are created by obtaining the true positive rate and false positive rate using different thresholds in the classification. If the area under curve near to 1 this is the best result and under 0.5 is not a good result. The curve is plotted using the true positive rate on Y axis and the false positive rate on X axis. This measurement is used alongside TPR, Precision and F-measure to evaluate the prediction performance. Area-under curve results alongside all of these measures are presented in the tables in section 6.3, 6.4 and 6.5.

2.11.5 Kappa

The Kappa statistic is another measurement for evaluating the classification results. The following figure shows the step used by the WEKA package to calculate the Kappa from the confusion matrix. Kappa is calculated according to the Equation 2-8. In this equation *P(A)* is proportion of times classification results are in agreement with the actual results. *P(E)* is the proportional of times that it is expected the classification results to be in agreement by chance (Landis and Koch, 1977), (Carletta 1996).

$$\boldsymbol{Kappa} = \frac{\boldsymbol{P(A)-P(E)}}{\boldsymbol{1-P(E)}}$$

Equation 2-8



classes	a	b	c	d	total	percent	Column1
a	417	3	27	0	447	0.684533	
b	12	18	20	1	51	0.078101	
c	44	6	98	0	148	0.226646	
d	1	0	4	2	7	0.01072	
total	474	27	149	3	653		observed
percent	0.72588	0.04135	0.22818	0.00459		535	0.8192956
	0.68453	0.0781	0.22665	0.01072			
	0.72588	0.04135	0.22818	0.00459			
	0.49689	0.00323	0.05172	4.9E-05			
Chance	0.55188						
kappa	(observed-chance)/(1-chance)						
	0.59675						

Figure 2-23 calculation of kappa statistic from confusion matrix

2.12 Searching the feature space

The ultimate goal of generating feature matrix is to investigate how features can classify the CpG sites. The following scenarios can be considered.

1. Using all features as an input to classification algorithms.
2. Using feature classes. Feature classes are the features that normally exist in one group, i.e., structural feature, transcription factors etc.
3. Using feature subsets from all features and creating smaller but more accurate predictive models.

The first and second scenarios can be easily implemented by providing machine learning software tools with the feature matrix file and choosing the algorithm. The third scenario needs some considerations. N features have total number of 2^N subsets and as the number of features increases, it is not computationally feasible to investigate all subsets. In order to implement feature subset selection, the following steps should be considered (Wang *et al.*, 2005).

1. Starting point: Selecting the start point in feature space a) all features can be considered first and gradually removing features to see if the classification will improve by new subset; b) choosing empty set and adding features; c) randomly selecting subset of features and evaluating them.

2. Evaluation: Evaluating subset can be done by providing new feature matrix to classification algorithm and examining the classification performance. There are different measures which can be applied for classification, i.e., number of correctly classified instance, f-measure, kappa measure, false positive rate and area under ROC curve.
3. Search method: Heuristic search algorithms are used in this research for searching the feature space. These algorithms are used when finding exact solution is impractical.
4. Stopping point: When the results are not improved anymore.

2.12.1 Heuristic search

As described in the previous section, some search spaces are too large and they cannot be searched exhaustively. Therefore, in this research other than the classic search methods has been used in order to search the feature space and reach the optimal solution. The technique that is used in this study is called *heuristic* method. Heuristic means some set of guidelines. In these search method we need some way to compare the solutions. For comparison first we need to evaluate the solutions in some way. These values are “fitness” of a solution. Ultimately we want to maximise or minimise fitness of the solution. The set that contains all possible solution is search space. Usually we can find some kind of distance between solutions that make them comparable. We have to define a representation for the solution. Representation can be seen as data structure that holds characteristic of solutions and each solution should uniquely mapped by representation. Representation should be able to represent all possible solutions.

These are the main components of the search

1. Representation for the solution.
2. Fitness function.
3. Comparison of fitness function to find desired value.
4. Methods for changing the solution.

Each heuristic search algorithm can have more components that specially designed for that method like genetic algorithms which have the mechanism of mutation and crossover for changing the solution and examining new solution.

Heuristic searches most of the time find local optimums. It means that if we start at some solution and looking for the solutions that are near that solution or neighbour solutions we can find better solutions and when we reach the optimum solution the program will stop to improve. At stopping point there may be still better solution in the search space that we haven't find. There are some methods to modify the solution to escape the local optima trap.

Hill climbing, Simulated Annealing, Genetic Algorithms, are among heuristic search algorithms. Genetic algorithm inspired from biological science and change in the genotype of organism (Whitely, D 1994). There are other biological inspired algorithms like swarm intelligence like ant colony optimisation, particle Swarm optimisation, bacteria colony optimisation (Karaboga, D and Akay, B 2009). Simulated annealing uses metaphor from material science (Kirkpatrick, S 1984). Hill climbing can be seen as climbing the graph in two dimensional spaces to reach the top of hill or local optima. Tabu search (Glover, F 1986) try to model human memory. It tries to use memory to escape from local optima. Iterated local search try to use another local search algorithm as part of the algorithm and it use previously found local optima to find new optima (Lourenço, Helena R., Olivier C. Martin, and Thomas Stutzle 2001).

2.12.2 Hill climbing

Hill climbing algorithm is a local search algorithm. Sometimes, it is also called greedy local search. In its simplest form, which is known as "steepest-ascent" version, the implementation of the algorithm is an iterative loop that directs the search. While the loop counters increments, the current optimal solution is compared with the next immediate neighbour solution. If the next solution is found to be better than the current optima, there is a replacement and the search continues until it reaches a pick. At this point the algorithm terminates the search. Because *hill climbing* looks only at immediate neighbour and compares that with the previously found optima, the pick point that the search terminates is not necessary the optimal solution, but rather a local optima, as can be seen from Figure 2-24. The success of hill climbing algorithm is mainly relies on the shape of search space.

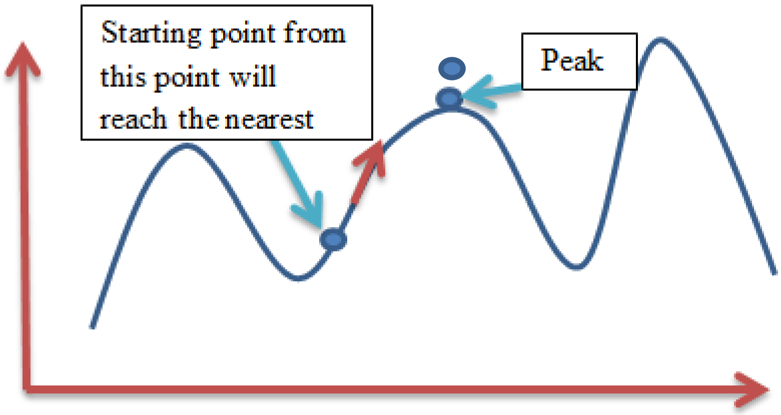


Figure 2-24 Hill climbing finds local optima

There are many variations of the hill climbing algorithm that mainly try to modify the algorithm to escape from local optima. One of these variations is the stochastic hill climbing where there is a random selection of steps towards the peak. Although it is slower, sometimes it finds better solution. Another one is the random restart hill climbing that restarts the algorithm at randomly different start points until it finds the optimal solution.

Hill climbing algorithms have been applied in a number of biological context studies. It is shown that this method can be effectively find grids in microarray images (Rueda and Vidyadharan, 2006). Hill climbing has been also used for multiple sequence alignment (Su, Lin and Ting, 2011). In another study, this search method has been applied for protein structure prediction (Kumar, Tyagi and Sharma, 2013). Hill climbing has been also used in combination with other methods, as it is shown in (Nunes *et al.*, 2004), to improve the performance of the genetic algorithm and avoid premature convergence. Additionally, it is used for gene subset selection in microarray data (González and Belanche, 2013).

Table 2-3 Hill Climbing

```
Algorithm 2-1    Hill Climbing

Input : Number of iteration

1. Initialise Random solution (Solution) for the problem and
   Random Fitness for the starting point (Fitness)
1. FOR number of iterations
2.        Choose random point near Solution (NewSolution)
          And calculate NewFitness
3.            IF NewFitness is better than Fitness
4.                  Solution = newSolution
5.                  Fitness = NewFitness
6.            EndIF
7. EndFor

Output : Solution
```

2.12.3 Simulated annealing

Simulated annealing is a probabilistic heuristic search method that aims to find the global optimal solution in a large search space. The idea and name of the method is borrowed from metallurgy. In metallurgy, the technique involves heating of the metals above their critical temperature and then cooling in order to decrease the material’s defects. Similarly, simulated annealing takes temperature as a parameter, starting with a high value or infinity and gradually the temperature parameter is reduced until it reaches a value of zero. Instead of selecting the best solution, it selects a random solution. If the solution is improved, it is always selected; otherwise the algorithm selects the solution with some probability. The probability of selecting decreases, when solutions with bad fitness are accepted. As the algorithm progresses, the temperature is cooling down and the probability of selecting bad solutions also goes down. This approach is used to make the algorithm escape from local optima and start the search again in different parts of the search space. Finally, when temperature reaches zero, it only looks at neighbour solutions to find the optimal solution.

Determining the rate of cooling is performed by a mathematical function and the parameter for that function should be defined. Similar to starting temperature, this can affect the way the algorithm behaves. For example, low cooling rate can cause the algorithm not to converge, on the other hand, high cooling rate can cause the algorithm to be similar to hill climbing and terminate the search at a local optimum. This is because it reaches zero temperature quickly and not enough random guess is allowed.

Simulated annealing search method has been used in many biological contexts. Some of them are described here. It has been applied to analyse gene expression data. It has also been used in QSAR (Quantitative structure–activity relationship) modelling for feature selection. QSAR is a method for predicting biological activity based on physiochemical activity (Ghosh and Bagchi, 2009). Simulated annealing has been used for gene subset selection by (Filippone, Masulli and Rovetta, 2006). Another application of the algorithm in biology was the study of genetic structure of the population by looking at genetic variance in the population. Recently, this method has been applied in new fields of synthetic biology, such as design of enhancers or cis-regulatory elements. This study explored DNA sequences for specific enhancers (Martinez *et al.*, 2013).

Table 2-4 Simulated Annealing

```
Algorithm 2-2    Simulated Annealing

Input : Number of iteration , Colling rate, Starting temperature

1. Choose some random solution for the problem (Solution)
2.   FOR the number of iterations
3.       Calculate the fitness of the Solution (Fitness)
4.       Find some solution close to the Solution
         (NewSolution) Calculate its fitness (NewFitness)
5.           IF NewFitness is worse than Fitness
6.               Using some probability based on
                 two fitness and temperature
                 choose NewSolution and Fitness
                 or keep Solution and Fitness
7.           Else  Solution=NewSolution
8.       Cool down temperature with cooling rate

Output : Solution
```

2.12.4 Genetic search algorithm

Genetic algorithm (GA) is heuristic search techniques inspired from biology. The method was developed by John Henry Holland in early 1970 and some improvements were added later. The basic idea is to represent possible solutions as chromosomes. Most of the time solutions are represented in binary string. These solutions are combined with each other to create new offspring. Fitter chromosomes are survived from one generation to another. There are mainly two methods for combining chromosomes (crossover): the uniform and the one-point crossover. In genetic algorithm the following definitions are used.

1. Generation: Number of times mating is occurred.
2. Population: Number of chromosomes alive at any time.
3. Crossover: Similar to biology this is the way chromosomes are recombined together to produce children chromosomes.
4. Mutation: This is random change of genes. For example one bit in the chromosome can randomly change from 1 to 0 or vice versa.

Each GA needs these parameters to be fixed there are some attempts to create parameter less genetic algorithms and make them simpler (Lobo and Goldberg, 2004).

1. Number of Generations
2. Population Size
3. Crossover Probability
4. Mutation Probability: Each gene given some probability to change in binary representation of the solution this is the rate conversion from 0 to 1 happens.
5. The number of bits (genes) making up each Chromosome.

The simple genetic algorithm is provided in the Table 2-5.

Table 2-5 Genetic Algorithm

```
Algorithm 2-3    Genetic Algorithm

Input : Number of Generations, Population Size,
        Crossover Probability , Mutation Probability
FOR Number of Generations
    Crossover Population
    Mutate all the Population,
    Kill off all Invalid Chromosomes
    Survival of the fittest 1
EndFor
Output : Chromosome with the best fitness in the last generation
```

2.13 Summary

In this chapter, the biology theoretical background of this research, and more specifically the background on DNA methylation, is provided. Furthermore, this chapter investigates the computational tools that can be used to study the DNA methylation. It starts with the biological background of DNA methylation and the importance of DNA methylation. It continues with the analysis of the existing methods for detecting DNA methylation. Then, it examines the challenges and opportunities raised by the advancements in technology and the availability of biological data. Also, this chapter visits the fields of bioinformatics.

Some computational tasks in bioinformatics have special requirements, like distributed nature of data and resources. Grid computing reviewed as one of the answers to these requirements. Different grid middleware implementations are examined in this chapter. From the reviewed grid middleware implementations, BOINC seems to have more advantages over the others. This is because it has the potential of using the voluntary resources. Furthermore, it has been used in a variety of bioinformatics projects. Workflow systems are studied here as the way to automate the procedures that contain multiple connected steps and, also, as the way to create modular systems where partly changes of the system do not affect other parts. Among different workflow systems, the gUse system shows the advantage of supporting different distributed systems that can be transparently connected to the workflow nodes.

Section 2.7 provides a review of different data sources that contain DNA methylation data series. Three of the main sources of data on DNA methylation, which were reviewed in this section, are published paper general purpose database and the rest are specialised database. Section 2.8 provides an overview of feature generation techniques. In this section MEME, MAST EMBOSS and USC table browser were identified as the main sources of feature generation in DNA methylation studies. Section 2.9 provides a review of the main classification methods used in this research. These methods are tree models, Naive Bayes and support vector machine. In section 2.10, an analysis of the heuristic search methods used to select subset of features is provided. These methods are hill climbing, simulated annealing and genetic search. Section 2.11 provides the methodology used in this research for analysing DNA methylation.

Chapter 3 - Comparative analysis of three Trinucleotide Repeated Diseases

3 Background

This section provides small scale problem of pattern generation for DNA methylation for small dataset steps are similar to large scale data analysis the analysis in this chapter has done on three trinucleotide repeat disease which has not reported to be compared yet in this way. Large scale data analysis provided in chapter 4 to 6 which is the extension of some steps which are used in this chapter in the distributed way.

DNA methylation involving the addition of a methyl group to a CpG sequence is one of the mechanisms of gene regulation commonly associated with transcriptional repression and is necessary for mammalian development, X inactivation and genomic imprinting (Pook, 2012). Gene silencing is a major biological consequence of DNA methylation. DNA methylation is reported in genes of both healthy cells, where it assists in regulating gene expression during development for example, and diseased cells, where it is associated with aberrant gene expression most notably in cancerous cells. One group of diseases in which DNA methylation is reported to have an important role is trinucleotide repeat (TNR) expansion diseases. Here we investigate the pattern of sequences in variably methylated (VM) and non-variably methylated (AM always methylated and NM never methylated) CpG sites of three TNR expansion diseases: FRDA, FRAXA and DM1 (Pook, 2012). Friedreich's ataxia (FRDA) is a disorder characterised by a GAA repeat expansion within intron 1 sequence of the FXN gene. The consequence of the expanded GAA repeats is to reduce the expression of the mitochondrial protein frataxin. Typically unaffected individuals have 5-32 GAA repeats and affected individuals have 66-1700 repeats. FRAXA is a mental retardation disorder associated with one of the seven folate-sensitive fragile sites that have been identified within human chromosomes. All of these sites have a large non-coding CGG repeat expansions. FRAXA is caused by repeat expansion with 5’ UTR of the FMR1 (fragile X mental retardation 1) gene. Unaffected individuals have 6-54 CGG repeats and affected individuals have 55-200 repeats (Pook, 2012). DM1 is an autosomal dominant disorder which characterised by clinical features such as muscle weakness, myotonia and heart defects. DM1 is characterised by expansion of CTG repeats within the 3’-UTR of the DMPK gene. Unaffected individuals have 5-37 CTG repeats and affected individuals have 90 to several thousand CTG repeats (Pook, 2012) .

Particular motifs have been identified which predict the methylation status of DNA sequences in normal cells. Notably methylation is more prevalent in regions of low CpG density, with regions of intermediate density being most variably methylated (Yuan, 2011). Computational methods have also been used showed that the frequencies of CpG, TpG, and CpA are different between unmethylated and methylated CpG islands (Bock and Lengauer, 2008). Further Yamada and Satou (Yamada and Satou, 2008) used machine learning by support vector machine and random forest using previously reported methylation data to analyse DNA sequence features to predict methylation status. They revealed that frequencies of sequences containing CG, CT or CA are different between unmethylated and methylated CpG islands. Ali and Seker (Ali and Seker, 2010) used an adapted K-nearest neighbour classifier to predict the methylation state on chromosomes 6, 20 and 22 in various tissues. They identified four feature sub-sets which were shown that the methylated CpG islands can be distinguished from the unmethylated CpG islands. Lastly Previti et al (Previti *et al.*, 2009) used a mining process in the absence of supervised data to cluster and then predict methylation status of CpG islands in different tissues and showed significant differences in the sequences of CpG islands (CGIs) that predispose them to such methylation. In their review of computational epigenetics, Bock and Lengauer (Bock and Lengauer, 2008) in their review “Computational Epigenetics” highlighted the fact that, although it is clear that much work has been done to document the epigenetic state of the genome (much of it reported in the ENCODE project (Encode project 2013)), work in the area of de novo DNA methylation prediction is to date limited. Aberrant methylation has been shown to be associated with mutations. Methylation in the MGMT promoter has been demonstrated to be closely associated with G:C to A:T mutations (Yuan, 2011). A few studies have attempted to search for motifs associated with aberrant methylation most notably Feltus et al (Feltus *et al.*, 2006) who suggest that the sequence surrounding a CpG can be used to predict aberrant methylation. In another study by McCabe et al (McCabe, Lee and Vertino, 2009) patterns were identified using machine learning techniques and used for pattern matching. DNA signatures and a co-occurrence with polycomb binding were found to predict aberrant CpG methylation in cancer cells.

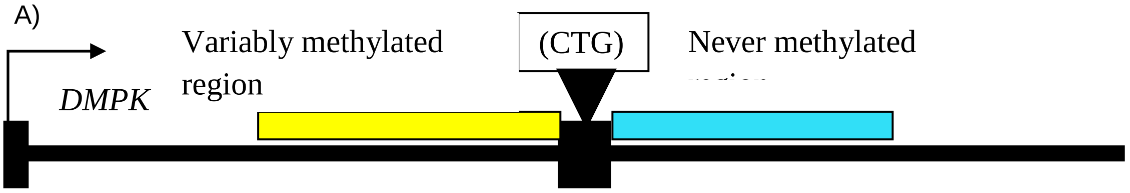
In this study a pattern searching algorithm was used to identify motifs in the DNA surrounding aberrantly methylated CpGs found in the DNA of patients with one of the three TNR expansion diseases: FRAXA, DM1 or FRDA. Sequences surrounding both the variably methylated (VM) CpGs, which are hypermethylated in patients compared with unaffected controls, and the non-variably methylated CpGs which remain either fully methylated (AM) or unmethylated (NM) in both patients and unaffected controls were examined. This expand the approach of Lu et al (Lu *et al.*, 2010) using a search window of 5bp allowing up to 3 mismatches. Any sequence with 4 mismatches is discounted because they represent only a one bp motif. Patterns identified therefore include mismatches, e.g. if the sequence contains GATCT it is counted in GA**T, GA*CT where * represents any base pair in the motif.

3.1 Method

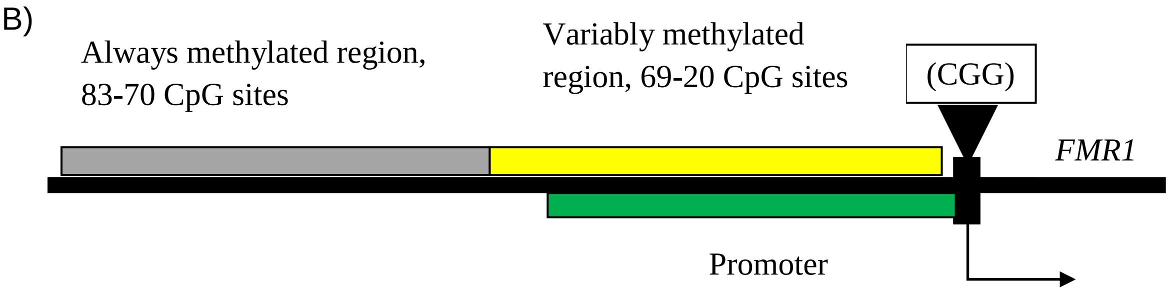
3.1.1 Pattern Generation

The three DNA sequences of FXN (Pook M *et al.*, 2008) FMR1 (Naumann *et al.*, 2009) and DMPK (Lopez Castel *et al.*, 2011) genes involved in the 3 TNR expansion diseases were examined. CpG sites for which their methylation statuses were available for both disease and normal cells were tagged. In order to identify patterns in the sequences of these CpGs all possible sequences of a window size of 5bp were generated in similar way to those used in the study by Lu et al 2010. DNA of 5bp length has been shown in the literature to have significant roles in biological functions, e.g. some of them are modifying sites and binding sites of enzymes and some are binding motifs of some transcription factors. It has also been shown that 5bp DNA lengths are important for DNA methylation where they are probably associated with the binding of DNA methyltransferase (Lu *et al.*, 2010). 5bp long sequences are important for the binding of many enzymes including the methyltransferase; both methylases (LlaDII and Bsp6I R/M) have two recognition sites (5’-GCGGC-3’ and 5’-GCCGC-3’) (27) and 5’-CCCGC-3’ is the recognition site of the DNA methyltransferase (methylase) FauIA (of the restriction-modification system FauI from Flavobacterium aquatile) (Chernukhin *et al* 2005).

Sequences from FXN, FMR1 and DMPK genes were divided into three classes of region; always methylated (AM), variably methylated (VM) and never methylated (NM), where the regions that are variably methylated are aberrantly methylated in patients and the always and never methylated regions are methylated and non-methylated respectively in both patients and controls. Significantly there is no overlap between regions. Both positive and negative strands were analysed. For each of these regions the 5bp window "slides along" from start to end of sequence and the pattern in that window is noted plus some additional information; the CpG site that patterns occurred near, location of pattern (using numbering as shown in Figure 3-1) and the exact sequence that occurred in the window.

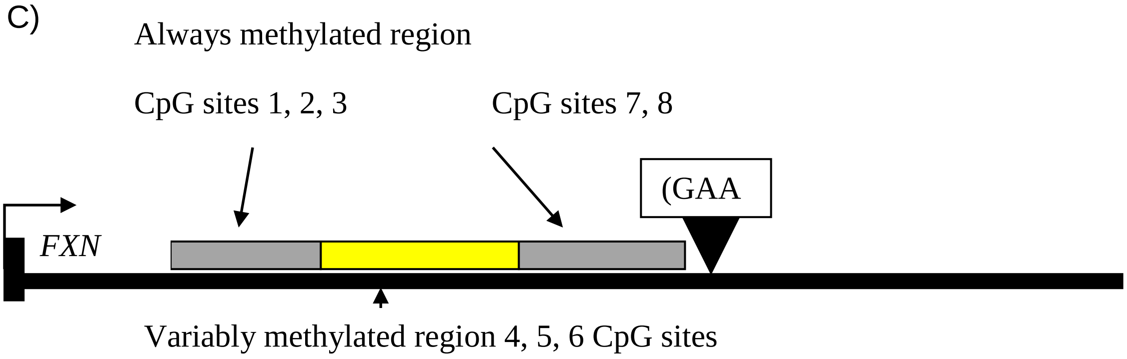


```
     ttgggttggt ggcccacgcc ggccaactca ccgcagtctg gcgccgccca ggagccgccc        60
     gcgctccctg aaccctagaa ctgtcttcga ctccggggcc ccgttggaag actgagtgcc       120
     cggggcacgg cacagaagcc gcgcccaccg cctgccagtt cacaaccgct ccgagcgtgg       180
     gtctccgccc agctccagtc ctgtgatccg ggcccgcccc ctagcggccg gggagggagg       240
     ggccgggtcc gcggccggcg aacggggctc gaagggtcct tgtagccggg aatg             294
ctgctgctgctgctgctgctgctgctgctg
     gggggga                                                                 6
     tcacagacca tttctttctt tcggccaggc tgaggccctg acgtggatgg gcaaactgca        66
     ggcctgggaa ggcagcaagc cgggccgtcc gtgttccatc ctccacgcac ccccacctat       126
     cgttggttcg caaagtgcaa agctttcttg tgcatgacgc cctgctctgg ggagcgtctg       186
     gcgcgatctc tgcctgctta ctcgggaaat ttgcttttgc caaa
```


B)
Always methylated region, 83-70 CpG sites
Variably methylated region, 69-20 CpG sites
(CGG)
FMR1
Promoter

```
tgttccgcca tcatggaagc gcctattctt catacccctt atcacagctg caactactca      60
tttacttgtc tgacaatttg atttatgtcc acctactttg ctaggtacta agttcaatgc     120
tggcagtcgt ttcttctttt tttttctttt ctgttttgct caccgatttc tcgttagcac     180
ttagcacagt gtctggcaca cgatagatgc tccgtcaact tctcagttgg ataccagcat     240
cccgaaggga acatggatta aggcagctat aagcacggtg taaaaacagg aataagaaaa     300
agttgaggtt tgtttcacag tggaatgtaa agggttgcaa ggaggtgcat cggcccctgt     360
ggacaggacg catgactgct acacacgtgt tcaccccacc ctctggcaca gggtgcacat     420
acagtagggg cagaaatgaa cctcaagtgc ttaacacaat ttttaaaaaa tatatagtca     480
agtgaaagta tgaaaatgag ttgaggaaag gcgagtacgt gggtcaaagc tgggtctgag     540
gtgaagagag agggcggggc cgaggggctg agcccgcggg gggagggaac agcgttgatc     600
acgtgacgtg gtttcagtgt ttacacccgc agcgggccgg gggttcggcc tcagtcaggc     660
gctcagctcc gtttcggttt c                                               681
```

C)



```
gaaggcggaa ggggatccct tcagagtggc tggtacgccg catgtattag gggagatgaa      60
agaggcaggc cacgtccaag ccatatttgt gttgctctcc ggagtttgta ctttaggctt     120
aaacttccca cacgtgttat ttggcccaca ttgtgtttga agaaactttg ggattggttg     180
ccagtgctta aaagttagga cttagaaaat ggatttcctg gcaggacgcg gtggc          235
```

Figure 3-1 This figure shows the 3 gene regions under investigation: A) DMPK 3'UTR region, B) FMR1 5' UTR region and C) FXN intron 1 region. A scheme of the DNA sequence, transcriptional start site and the regions analysed are shown. The grey shading shows the always methylated (AM) regions, blue, the never methylated (NM) and the yellow area shows variably methylated (VM) regions. CpG sites are underlined and bold numbers at start and end of each line show base pair number in the sequence. A triangle shows the location of repeat expansion and the box above triangle shows the TNR repeats. The green highlighted region in the FMR1 gene indicates the promoter region. The CTCF binding sites are shown in red Expanded from (Pook 2012).

The patterns identified were ranked in order of frequency in a region class and the proportional frequency in each region as calculated by dividing the frequency in that region by the length of the region in bp. The proportional frequencies of each pattern in each region class were calculated by adding all the regions in that class together giving the sum of that proportional frequency in that region class. Thus we are able to determine which patterns are most prevalent in each methylation region class. We were able to identify patterns that are not present in any one class and more prevalent in the other two classes using the sum of the other two classes' proportional frequencies (Table 3-1).

Table 3-1 Table of discriminatory patterns

top 10 patterns that separate AM class from NM and AM	top 10 patterns that separate VM class from AM and NM	top 10 patterns that separate NM class from AM and VM
(less frequent in AM than NM+VM)	(less frequent in VM than AM+NM)	(less frequent in NM than AM+VM)
ccgg[agct]{0,1}	[agct]{0,1}tttt	ta[agct]{0,2}a
g[agct]{0,1}gcg	t[agct]{0,1}cat	[agct]{0,1}gcgg
g[agct]{0,1}ctc	catg[agct]{0,1}	[agct]{0,1}ccgc
cgg[agct]{0,1}t	ga[agct]{0,1}at	c[agct]{0,1}ccg
ag[agct]{0,1}ct	at[agct]{0,1}ca	ag[agct]{0,1}gg
cg[agct]{0,1}tc	taa[agct]{0,1}t	c[agct]{0,1}gcg
cga[agct]{0,1}c	tct[agct]{0,1}a	ag[agct]{0,1}ac
t[agct]{0,1}cga	[agct]{0,1}tgca	tt[agct]{0,1}aa
gt[agct]{0,1}ac	ta[agct]{0,1}ta	ctt[agct]{0,1}a
tcga[agct]{0,1}	ac[agct]{0,1}ta	ttt[agct]{0,1}a

Patterns that are unique in one class and didn't occur in other two classes are shown in Table 3-2. This makes it possible to determine which pattern(s) best discriminated between the region classes.

Table 3-2 Table of unique patterns in each region

The 8 proportionally most frequently occurring patterns which are more prevalent in NM than VM and AM			
pattern	sum(VM)	sum(AM)	sum(NM)
a[agct]{0,1}gat	0.000	0.004	0.009
atc[agct]{0,1}t	0.000	0.004	0.009
[agct]{0,1}atcg	0.000	0.006	0.009
cgat[agct]{0,1}	0.000	0.006	0.009
atcg[agct]{0,1}	0.000	0.006	0.009
[agct]{0,1}cgat	0.000	0.006	0.009
[agct]{0,1}atct	0.000	0.002	0.004
agat[agct]{0,1}	0.000	0.002	0.004
Top 10 proportionally most frequently occurring patterns unique to AM			
pattern	sum(VM)	sum(AM)	sum(NM)
taa[agct]{0,1}t	0.000	0.044	0.000
tct[agct]{0,1}a	0.000	0.043	0.000
ta[agct]{0,1}ta	0.000	0.039	0.000
ga[agct]{0,1}aa	0.000	0.035	0.000
ta[agct]{0,1}at	0.000	0.030	0.000
t[agct]{0,1}tat	0.000	0.029	0.000
at[agct]{0,1}ag	0.000	0.027	0.000
[agct]{0,1}atac	0.000	0.025	0.000
a[agct]{0,1}tag	0.000	0.025	0.000
tat[agct]{0,1}a	0.000	0.025	0.000
Patterns unique to VM			
pattern	sum(VM)	sum(AM)	sum(NM)
gt[agct]{0,1}ac	0.014	0.000	0.000
g[agct]{0,1}gac	0.011	0.000	0.000
ga[agct]{0,1}tc	0.007	0.000	0.000

Further, to validate and compare our results we used MEME software (Bailey *et al.*, 2006). to identify patterns in these same regions. A 5bp window size was used and “any number of repetitions” was selected, all other settings were default.

The WEKA (Mark H et al. 2009) J48 classification technique was used to find the patterns that best classify the sequences in the three classes. The patterns of each region were used as attributes in the analysis rather than the sums of all the regions in the same class. The patterns were treated as attributes in WEKA and sequences as instances. We used the WEKA J48 algorithm (an implementation of the C4.5 algorithm) to generate a decision tree. Attributes are selected based on information gain so in our tree CCGG* has the highest information gain.

The WEKA package is used in this research for classification because it is open source and free and is compatible to other part of the system although other machine learning tools can be used another reason for the use of WEKA is because it is written in JAVA and the hill climbing search and pattern generation in this chapter are both written in JAVA it was easier to call WEKA libraries programmatically. In another setting other classification tools can be used.

To determine if any of the patterns have an identity to known DNA motifs for such DNA binding proteins as transcription factors we analysed the patterns identified by WEKA as distinguishing each region class using TOMTOM software (Bailey *et al.*, 2006) using the JASPER and Uniprobe databases for the TOMTOM search.

3.2 Results

3.2.1 Frequency

Of all the possible combinations of 5bp patterns where 2 or more of the 5bps are identical within a pattern e.g. CG*** is one pattern where patterns of CGTTG and CGTTA are the actual sequences found. 1584 different patterns where found in all the regions analysed. Most were found in all 3 genes in all the regions. 1454 patterns were found in the VM regions, 1563 in the AM regions and 1264 in the NM region. Two patterns are unique in the FMR1 gene in both regions. There are no unique patterns for FXN. One pattern is unique for the DMPK gene in both regions. Analysis of the patterns revealed that some patterns did not occur in some regions allowing the region classes to be separated these results shown in Table 3-2.

Table 3-3 Pattern count algorithm

```
Algorithm 3-1    Pattern Count

Input : DNA sequence , word size n

1. Generate all possible pattern of size n with gaps and
   mismatch
2.    FOR each DNA sequence
3.       FOR each windows slide of size n in the sequence
4.          FOR each pattern generated in Line 1
5.             IF the pattern matches the slide
6.                add one to the value of the $A_{ij}$
                  in the matrix A i is the index of
                  sequence and j is the index of the
                  pattern.
7.             EndIf
8.          EndFor
9.       EndFor
10.   EndFor

Output : Matrix A of pattern count
```

3.2.2 Proportional frequency

On calculating the sum of the proportional frequencies of patterns we found three patterns are unique for VM regions, 84 are unique to AM. There are no unique patterns for the NM region. The patterns which showed the greatest proportional differences between the regions are given in Table 3-2. This shows that there are patterns which are unique to VM and AM regions.

The summed proportional frequencies of each pattern for each region class showed a distinct difference in the frequencies of particular patterns in different class regions. Our results clearly show that some patterns are more prevalent in some region classes than others and therefore the methylation status of the regions around the repeats is influenced by the underlying DNA sequence as well as the length of the trinucleotide repeat.

3.2.3 WEKA J48 Analysis

The finding from the frequencies showing that some patterns could be used to distinguish the 3 class regions from each other was confirmed by J48 classification decision tree analysis using WEKA software. The results are given in Figure 3-2. The WEKA programme identified that two patterns are all that is necessary to classify our 3 regions, as shown in the decision tree.

Using the proportional frequencies of all regions (not the summed frequencies) we observed that AATT* distinguished between NM and VM + AM using the J48 algorithm where the proportional frequency of AATT* is more than 0.003697 in NM. This result mirrors the result of the frequency analysis reported above that there are no unique patterns for the NM region hence the distinction is based on frequency rather than the presence or absence of a pattern. AM can be distinguished from VM by the sequence CCGG* which is found in VM and not AM regions.

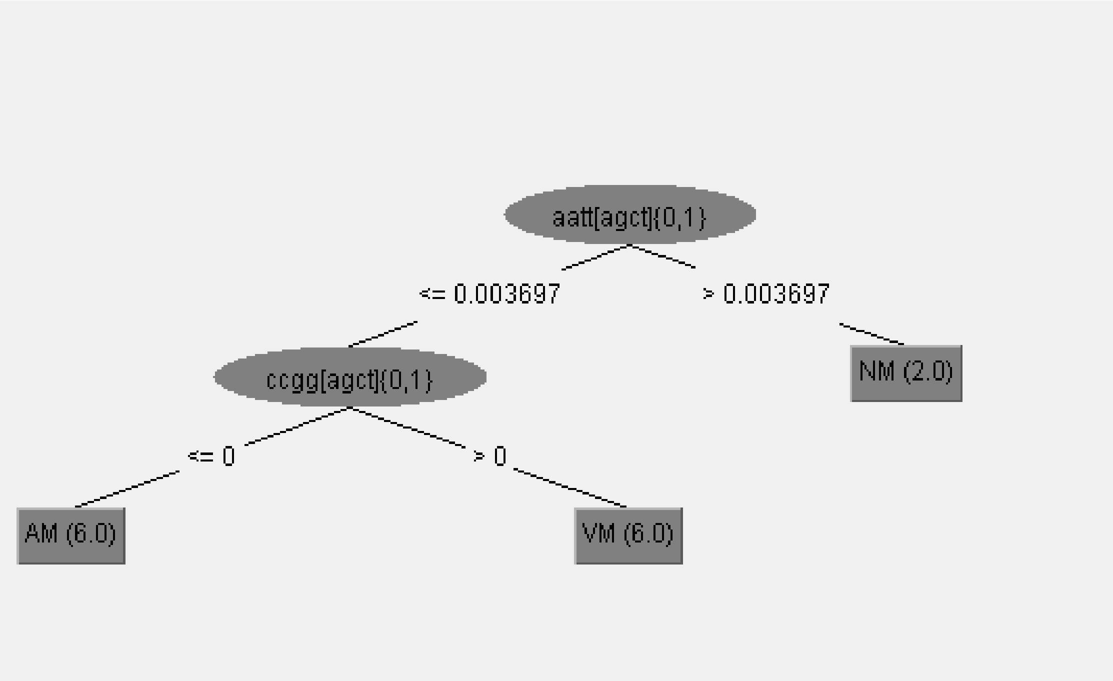
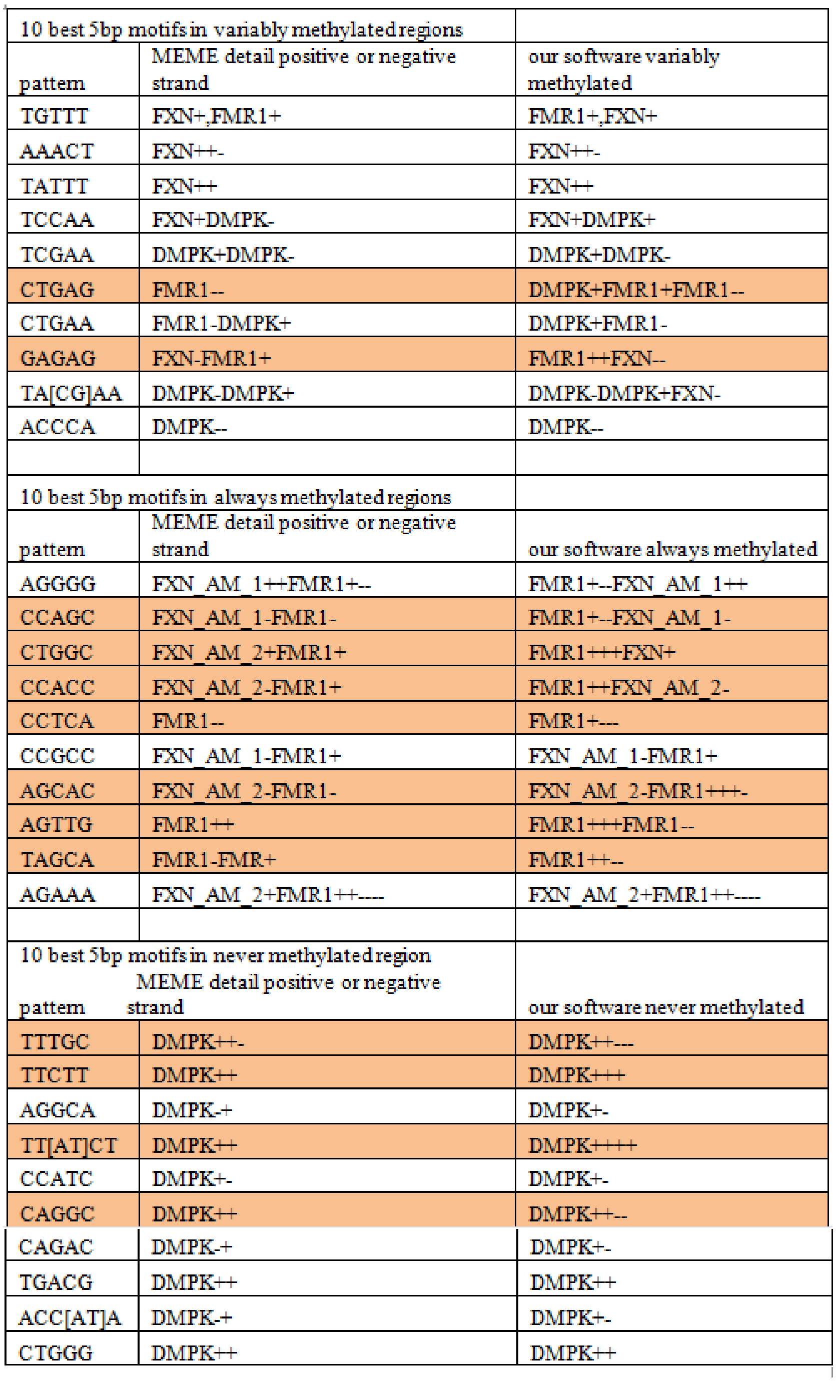


Figure 3-2 Decision tree created by WEKA package. AM is always methylated, NM is never methylated (NM) and (VM) is variably methylated.

3.2.4 Comparison with MEME software

In order to compare algorithm 3-1’s predictions to those generated by MEME we compared the 10 best distinguishing motifs identified by MEME with the patterns identified by algorithm 3-1. The results are given in Table 3-4. The results are comparable but notably algorithm 3-1 identified more patterns than the MEME software. No patterns found using the MEME software were missed by algorithm 3-1.

Table 3-4 Comparison of MEME results with patterns found in this study



10 best 5bp motifs in variably methylated regions		
pattern	MEME detail positive or negative strand	our software variably methylated
TGTTT	FXN+,FMR1+	FMR1+,FXN+
AAACT	FXN++-	FXN++-
TATTT	FXN++	FXN++
TCCAA	FXN+DMPK-	FXN+DMPK+
TCGAA	DMPK+DMPK-	DMPK+DMPK-
CTGAG	FMR1--	DMPK+FMR1+FMR1--
CTGAA	FMR1-DMPK+	DMPK+FMR1-
GAGAG	FXN-FMR1+	FMR1++FXN--
TA[CG]AA	DMPK-DMPK+	DMPK-DMPK+FXN-
ACCCA	DMPK--	DMPK--
10 best 5bp motifs in always methylated regions		
pattern	MEME detail positive or negative strand	our software always methylated
AGGGG	FXN_AM_1++FMR1+--	FMR1+--FXN_AM_1++
CCAGC	FXN_AM_1-FMR1-	FMR1+--FXN_AM_1-
CTGGC	FXN_AM_2+FMR1+	FMR1+++FXN+
CCACC	FXN_AM_2-FMR1+	FMR1++FXN_AM_2-
CCTCA	FMR1--	FMR1+---
CCGCC	FXN_AM_1-FMR1+	FXN_AM_1-FMR1+
AGCAC	FXN_AM_2-FMR1-	FXN_AM_2-FMR1+++-
AGTTG	FMR1++	FMR1+++FMR1--
TAGCA	FMR1-FMR+	FMR1++--
AGAAA	FXN_AM_2+FMR1++----	FXN_AM_2+FMR1++----
10 best 5bp motifs in never methylated region		
pattern	MEME detail positive or negative strand	our software never methylated
TTTGC	DMPK++-	DMPK++---
TTCTT	DMPK++	DMPK+++
AGGCA	DMPK-+	DMPK+-
TT[AT]CT	DMPK++	DMPK++++
CCATC	DMPK+-	DMPK+-
CAGGC	DMPK++	DMPK++--
CAGAC	DMPK-+	DMPK+-
TGACG	DMPK++	DMPK++
ACC[AT]A	DMPK-+	DMPK+-
CTGGG	DMPK++	DMPK++

Key

+ Positive strand

- Negative strand

FXN_AM_1 and FXN_AM_2 are two separated regions with always methylated CpG sites.

🟧 Motif that meme does not report all occurrences

3.2.5 CTCF binding

Since it has been reported that FRDA patients have depleted levels of CTCF and there is a suggestion that this protein could act to protect DNA from targeted methylation in healthy individuals (De Biase *et al.*, 2009), the regions were analysed for CTCF binding sites to determine if the differential methylation could be linked not only to DNA sequence but also CTCF binding. The diagrams in Figure 3-1 show the putative CTCF binding sites in the analysed regions. Since the bindings sites seem to be equally prevalent in all the region classes it would seem that simple depletion of CTCF levels may not be the explanation for the variability in the methylation in patients compared to controls unless there are other factors that influence the binding of CTCF to its site over and above just the binding site sequence.

3.2.6 Comparison of patterns with DNA binding protein sites

Hogart et al (Hogart *et al.*, 2012) identified overrepresented transcription factor consensus binding motifs in methylated sequences. This would suggest that the methylation-sensitive binding of DNA binding proteins plays an important part in regulating genes. Thus the variation in methylation seen in VM regions could be an important mechanism in these disease states due to the DNA binding proteins that bind to these regions. Further since the binding of DNA binding proteins such as transcription factors may influence aberrant methylation patterns we wished to compare our patterns with binding sites in the Jasper and Uniprobe database. TOMTOM analysis revealed that the CCGG pattern which is found in VM regions but not AM regions is part of the consensus binding site for 35 DNA binding proteins however not all are found in mammals. A human protein ELK4, which was found to bind, may be influenced by the degree of methylation in the promoter of some genes as demonstrated in the caldesmon gene (CALD1) by Cooper et al (Cooper *et al.*, 2007). Another, GABPA, whose binding sites are over represented in methylated regions of primary mouse hematopoietic stem cells (Hogart *et al.*, 2012), shows evidence of GABPA binding being methylation-sensitive as demonstrated by Lucas et al (Lucas et al 2009) who showed that the regulation of TMS1/ASC gene is controlled in such a way.

The AATT pattern which is found more frequently in NM matched with 151 DNA binding protein consensus sequences, although very few are found in mammals. However, there was a preponderance of homeobox domain proteins in the matches. One protein, PAX6, is inhibited from binding by methylation of its binding site (Wang *et al.*, 2011). Another, PAX7 results in H3K4 tri-methylation of surrounding chromatin stimulating transcriptional activation of target genes to regulate entry into the myogenic developmental programme in skeletal muscle (McKinnell *et al.*, 2008).

3.3 Conclusion

Our results showed that there are sequence patterns which can be used to distinguish between AM, VM and NM regions of these TNR genes. A single pattern can be used to distinguish the NM region from the other two. Furthermore, the fact that the VM regions show a few striking and unique patterns is particularly notable when the frequencies are summed and WEKA analysis of non-summed frequencies show that one pattern can be used to distinguish this region class from AM. This finding could point towards one mechanism which contributes to the methylation status of these regions of DNA in patients compared with controls.

The three genes however show differences based on our classification of the VM, AM and NM regions. There could be several explanations for this: For DMPK the VM region is upstream of the TNR and has the only NM region in any of the genes which is downstream of TNR. In FMR1 and FXN both VM and AM regions are upstream of the repeat region. DMPK and FMR1 are similar in the way that their AM region is continuous unlike FXN which has two disconnected AM regions. Further the nature of the TNRs in each of the genes is different; FXN has a TNR of GAAn, FMR1 has CGGn and DMPK has CTGn. Thus FXN is unique in having only purines in one strand of its repeat (and only pyrimidines in the other strand), while the other two repeats are mixtures of purines and pyrimidines in each strand.

In comparing algorithm 3-1 to identify patterns with MEME we have demonstrated that algorithm 3-1 identifies more patterns than MEME in these short DNA sequences. MEME software has been optimised to find patterns in much longer sequences thus may be not as good as algorithm 3-1 for detecting patterns in short sequences or using small window sizes. Further when the results from MEME alone on our sequences was analysed using WEKA. The software gave a less discriminating tree than the results from algorithm 3-1. Thus showing algorithm 3-1 is better at discriminating patterns than MEME.

There are many possible points of discussion that can be drawn from our data. The report that FRDA patients have depleted levels of CTCF suggested the possibility that this protein could act to protect DNA from targeted methylation in healthy individuals (De Biase *et al.*, 2009). However, the distribution of potential CTCF binding sites in the three genes examined here would suggest that this is not the sole cause of the variation in methylation seen in the different regions.

It is notable that the restriction sites of the two classical enzyme pairs HpaII - MspI (CCGG) and SmaI - XmaI (CCCGGG) used to analyse DNA sequences for methylation have CCGG at their core. Although not all CpG methylation occurs within these sequences much does. This illustrates the significance of discovering the CCGG pattern as a mark for VM regions.

DNA methyltransferases 3a and 3b (Dnmt3a/b) are the enzymes responsible for de novo DNA methylation in humans and the mouse. However, the mechanisms by which specific DNA sequences are targeted to be methylated are not known, nor are the signals that trigger this phenomenon.

The work of Hervouet et al (Hervouet, Vallette and Cartron, 2009) has shown that Dnmt3a and Dnmt3b have consensus sequences to methylate DNA (T/A/C)(A/T)(T/G/A)CG(T/G/C)G(G/C/A) and (A/C)(C/G/A)(A/G)CGT(C/G)(A/G).

Thus the propensity of a methylase to de novo methylate certain CpG may not due to the binding specificity of the enzyme itself since these sequences demonstrate the low specificity of these enzymes. Hervouet et al. go on to suggest that the mechanism is controlled by the interaction of Dnmt3a or 3b with specific transcription factors suggesting that the specificity comes from the binding or not of these transcription factors to specific sequences in the promoter regions of genes. It is also possible that there is an interaction of antisense RNA with specific DNA sequences or with the methylases themselves to molecules that may aid in the directing of de novo methylation. Epi-miRNAs have been demonstrated to regulate or possibly direct the epigenetic machinery as reported in a review by Lorio et al (Iorio, Piovan and Croce, 2010). Either of these mechanisms could lead to the more directed de novo methylation seen in vivo and therefore could explain the differences between the logos characterised for the 3 different genes investigated in this work.

Such aberrant methylation is well known to cause down regulation of genes resulting in disease states by very different mechanisms. In cancer the aberrant methylation is not under the influence of TNRs present near the genes, thus the mechanism giving the observed variation in methylation in these genes is probably subtly different. Furthermore, the resulting methylation may result in different effects. In FMR1, DNA methylation prevents the binding of the transcription factor α-Pal/NRF-1, whereas methylation of the FXN intron 1 region may be involved in the formation of a transient purine-purine-pyrimidine DNA triplex preventing transcriptional elongation (Grabczyk, Kumari and Usdin, 2001). Recently micro RNAs have been hypothesised to have a role in the down regulation of genes. It has been shown that micro RNA expression can be modulated by promoter methylation or histone acetylation, a phenomenon that is found in numerous diseases including FRDA. Also antisense RNAs may be more highly expressed. Interestingly, work by De Biase et al (De Biase *et al.*, 2009). shows the presence of increased amounts of a novel transcript FAST-1 (FXN Antisense Transcript – 1) in FRDA patients which may prove to be significant.

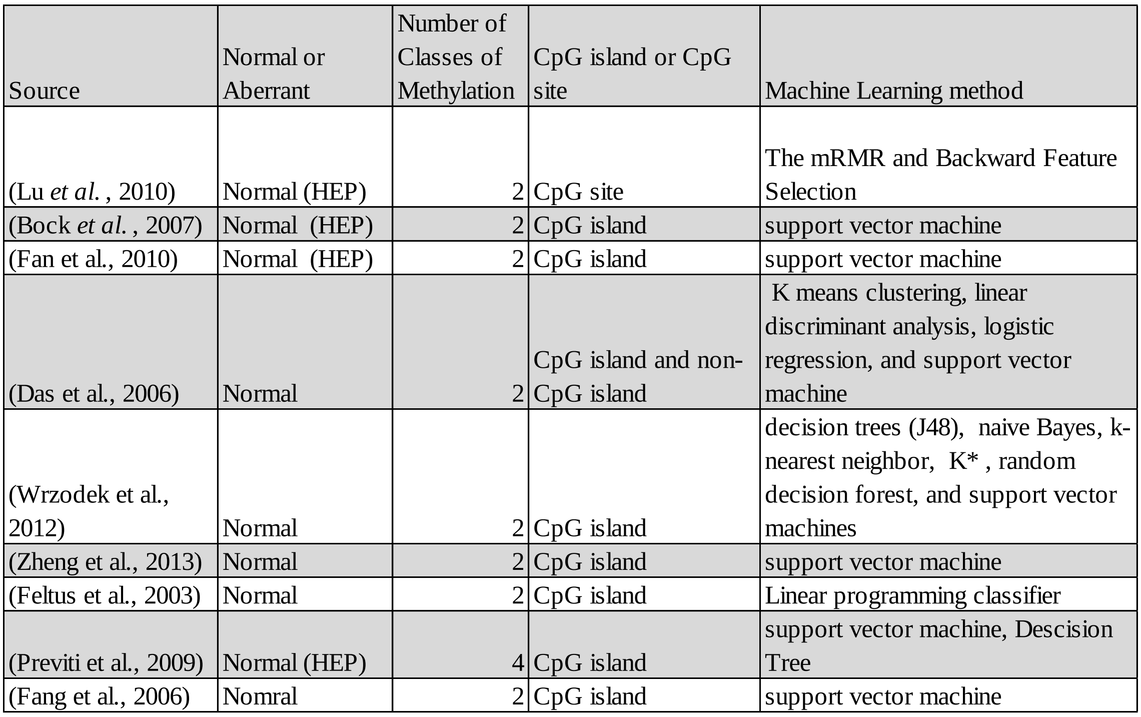
Thus the results presented here are evidence that the DNA sequence surrounding a CpG can influence its susceptibility to be de novo methylated in a disease state associated with a trinucleotide repeat. This supports the findings of other investigators who have made similar findings in cancer cells (Feltus *et al.*, 2006). Our results represent those from only three of the numerous trinucleotide repeat associated diseases. We acknowledge therefore that further work to elucidate the involvement of DNA methylation and then the DNA sequence around any methylated CpG islands in patients is required to build a complete picture of this phenomenon in this classification of diseases.

Chapter 4 - CpG site identification

4 Background

CpG dinucleotide methylation is one of the mechanisms of epigenetic regulation that has important role in cancer therefore differentially methylated CpG sites can act as a biomarker in identifying disease or can provide better understanding mechanisms of DNA methylation.

Most previous studies are based on the CpG islands and two classes of methylation prone and methylation resistant CpGs and mainly on the normal tissue. Lu et al used word composition to generate features and predict DNA methylation on normal CpG site (Lu *et al.*, 2010). Bock et al used support vector machines to find the attributes that predict CpG islands on human chromosome 21 and 22 they reported that there is correlation between Epigenetic factors and DNA factors (Bock *et al.,* 2007). Fen et al analysed HEP data and provides the model which outperformed similar models using support vector machines they use both DNA features and histone modification data as the features in their model (Fan *et al.,* 2010). Das et al use different machine learning algorithms including SVM and logistic regression to analyse and predict DNA methylation on the human brain (Das *et al.,* 2009). Wrzodek et al used CpG island data set from ENCODE project and generate different type of features including structural ,sequence based and CpG island properties to predict DNA methylation (Wrzodek *et al.,* 2012). Zheng et al used support vector machine and HEP dataset to create predictive model of DNA methylation with various features including transcription factor binding sites, DNA structure, and DNA sequence attributes (Zheng *et al.,* 2013). Feltus et al analysed DNA methylation in cell lines and DNA sequence patterns along side machine learning methods to predict aberrant DNA methylation (Feltus *et al.*, 2003). Preveti et al profile DNA methylation they use two more tissue specific classes to their prediction analysis they use HEP data as the dataset for analysing DNA methylation (Preveti *et al.,* 2009). Fang et al use human brain tissue dataset (Rollins *et al.,* 2006) using support vector machine and DNA sequence features like frequency of transcription factor binding sites , the number of Alu repeat that overlapped with the CpG island (Fang *et al.*, 2006).



Source	Normal or Aberrant	Number of Classes of Methylation	CpG island or CpG site	Machine Learning method
(Lu *et al.* , 2010)	Normal (HEP)	2	CpG site	The mRMR and Backward Feature Selection
(Bock *et al.* , 2007)	Normal (HEP)	2	CpG island	support vector machine
(Fan et al., 2010)	Normal (HEP)	2	CpG island	support vector machine
(Das et al., 2006)	Normal	2	CpG island and non-CpG island	K means clustering, linear discriminant analysis, logistic regression, and support vector machine
(Wrzodek et al., 2012)	Normal	2	CpG island	decision trees (J48), naive Bayes, k-nearest neighbor, K* , random decision forest, and support vector machines
(Zheng et al., 2013)	Normal	2	CpG island	support vector machine
(Feltus et al., 2003)	Normal	2	CpG island	Linear programming classifier
(Previti et al., 2009)	Normal (HEP)	4	CpG island	support vector machine, Descision Tree
(Fang et al., 2006)	Nomral	2	CpG island	support vector machine

Figure 4-1 Summary of previous works on CpG island and CpG site methylation

Most of these works are based on the HEP dataset which examines three chromosomes. With more data available on single CpG sites it is possible to investigate more classes of DNA methylation. Here we identified sites in four classes of change in DNA methylation, sites that are hypermethylated in normal and hypomethylated in cancer samples, hypomethylated in normal and hypermethylated in cancer, sites that are always Hypermethylated and sites that are always hypomethylated.

In this study we found find CpG sites that are always methylated or never methylated across all samples or differentially methylated in cancer and normal samples with the following objectives:

4.1.1 Objectives

1. Identify methylation sites that can act as biomarkers for all type of cancers from the laboratory data. These are the sites that are mostly hypermethylated in cancer and hypomethylated in normal or hypermethylated in normal and hypomethylated in cancer cells.

2. To identify DNA motifs in the DNA sequence surrounding a CpG that could reder a CpG prone to methylation in cancer cells.
3. Using these motifs, predict CpG sites that are differentially methylated between normal and cancer cells *in silico.*
4. Improve the selection of and reduce the number of CpG sites used for the determination of differential methylation.

4.2 Methods

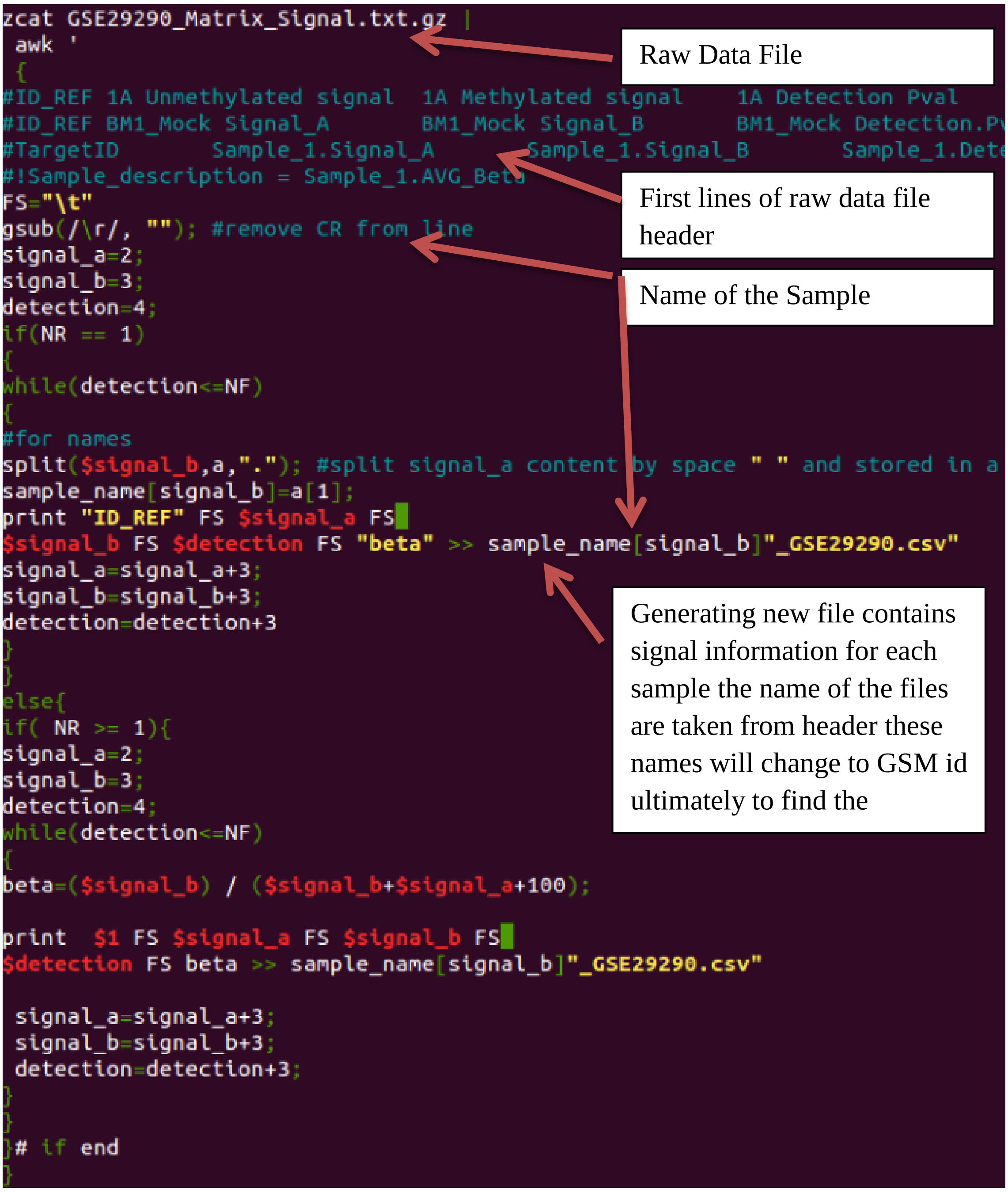
4.2.1 Datasets Selection

Most previous studies were focused on CpG islands but with the advance of technology data available for single CpG sites by using microarray technology one of the best candid for this purpose is HumanMethylation450k because of coverage and samples available for this platform.

At the time of writing (April 2013) there are 73 series submitted to GEO based on HumanMethylation450K platform. We have selected data series studying cancer cells and normal samples where the raw data is available for all CpG sites in HumanMethylation450K microarray. Series that did not have a published paper associated with them were excluded since we wished to use the most reliable data. We included one series that investigated only normal samples to allow the number of fewer normal samples to be as numerous as the cancer studies. The platform soft file was downloaded from GEO and converted to table format with custom program filtering which resulted in 16 data series and 535 data samples of which 301 of them are cancer samples and 234 are normal samples consisting of 259,783,695 data points.

4.2.2 Raw data standardization for all series

Series submitted to GEO contain processed and raw data. Here the only series considered were those that have raw data. Matching the series raw data to the characteristic of the samples was not standard as each raw data had to be treated separately because of the naming convention on each sample name. There were two general ways of raw data submission seen in the series submitted for HumanMethylation450K datasets. A) Signal intensities are in two separate files b) Signal intensities are in the same file. To extract the data and put all the samples in separate files which contained two signal intensity, the first few lines of each raw data file were examined and the appropriate awk script was generated to read the lines of series raw data. An example of awk script is provided here.



```
zcat GSE29290_Matrix_Signal.txt.gz |
 awk '
 {
#ID_REF 1A Unmethylated signal  1A Methylated signal     1A Detection Pval
#ID_REF BM1_Mock Signal_A       BM1_Mock Signal_B        BM1_Mock Detection.P
#TargetID       Sample_1.Signal_A       Sample_1.Signal_B       Sample_1.Dete
#!Sample_description = Sample_1.AVG_Beta
FS="\t"
gsub(/\r/, ""); #remove CR from line
signal_a=2;
signal_b=3;
detection=4;
if(NR == 1)
{
while(detection<=NF)
{
#for names
split($signal_b,a,"."); #split signal_a content by space " " and stored in a
sample_name[signal_b]=a[1];
print "ID_REF" FS $signal_a FS
$signal_b FS $detection FS "beta" >> sample_name[signal_b]"_GSE29290.csv"
signal_a=signal_a+3;
signal_b=signal_b+3;
detection=detection+3
}
}
else{
if( NR >= 1){
signal_a=2;
signal_b=3;
detection=4;
while(detection<=NF)
{
beta=($signal_b) / ($signal_b+$signal_a+100);

print  $1 FS $signal_a FS $signal_b FS
$detection FS beta >> sample_name[signal_b]"_GSE29290.csv"

 signal_a=signal_a+3;
 signal_b=signal_b+3;
 detection=detection+3;
}
}
}# if end
}
```

Raw Data File

First lines of raw data file header

Name of the Sample

Generating new file contains signal information for each sample the name of the files are taken from header these names will change to GSM id ultimately to find the

Figure 4-2 shell script code for extracting samples raw data and creating one file per sample

This script created separate files for each sample with the headers as the name. To make it useful it must contains information about the characteristic of the sample and its condition e.g. cancer, so each file must be mapped to the GSM id and the header title stored in the Sample_title field of GEO soft files. A Java program was developed to rename these files to the GSM files as the final file which will be used in other sections.

4.2.3 Methods of Dataset selection

Each platform in GEO database has the SOFT file for its series which contains information on series submitted to that platform the format of platform SOFT file is as follows.

The first task is to download the platform soft file from the GEO database .HumanMethylation450K platform id is GPL13534. In the platform file each series entries started with ^SERIES = GSE20945, so downloading all series was achieved by looping in the GPL file line and using the URL in java program, reading all the series of the GPL file. Now all series SOFT file are available, these files should be examined for their sample and each sample's characteristic. The characteristic lines start with "!Sample_characteristics_ch" and also the same platform id are extracted by "!Sample_platform_id", also series "Series_pubmed_id" and "Series_sample_id". Although the search start for the same platform for all samples the sample platform id should be extracted because in some cases samples belong to the same study but two different platforms and both platforms e.g. those that appear in the series file downloaded for GPL13534. All of this information is converted to a tab delimited file. This file provides all information in the tab delimited format for every sample in the platform and is used in Microsoft Excel to filter samples that did not have a Pubmed id as this indicated that they haven't published yet. Samples from platforms other than GPL13534 are removed as well. All 73 series were examined manually to find the series related to cancer. Finally the cancer related series that remain in dataset were used to distinguish samples with healthy or normal condition from cancers, by keywords like "health". These keywords were used to create new fields in the tab delimited file for every line. Finally the Excel program was used to separate the samples with empty healthy/normal and non-empty fields. This created two sets of samples, one for normal and the other one for cancer samples. This process generated two files with a total of 535 data samples 301 of them are cancer samples and 234 are normal samples.

Table 4-1 Dataset Selection Algorithm

```
Algorithm 4-1    Dataset Selection

Input : GEO Platform soft file "GPL13534"
For each Line in GPL file
    If Line starts with "^SERIES"
        Extract series name from the file.
        Download the series soft file using GEO database URL.
    EndIf
EndFor
For each series soft file GSE in directory
    For each Line in the GSE file
        If Line starts with
            "!Sample_characteristics_ch"
            or "!Sample_platform_id" or "Series_pubmed_id"
            or "Series_sample_id"
            Extract Sample id characteristic publication
            And platform id and insert to tab delimited
            file information from the Line
        EndIf
    EndFor
EndFor
Output : samples information tab delimited file
```

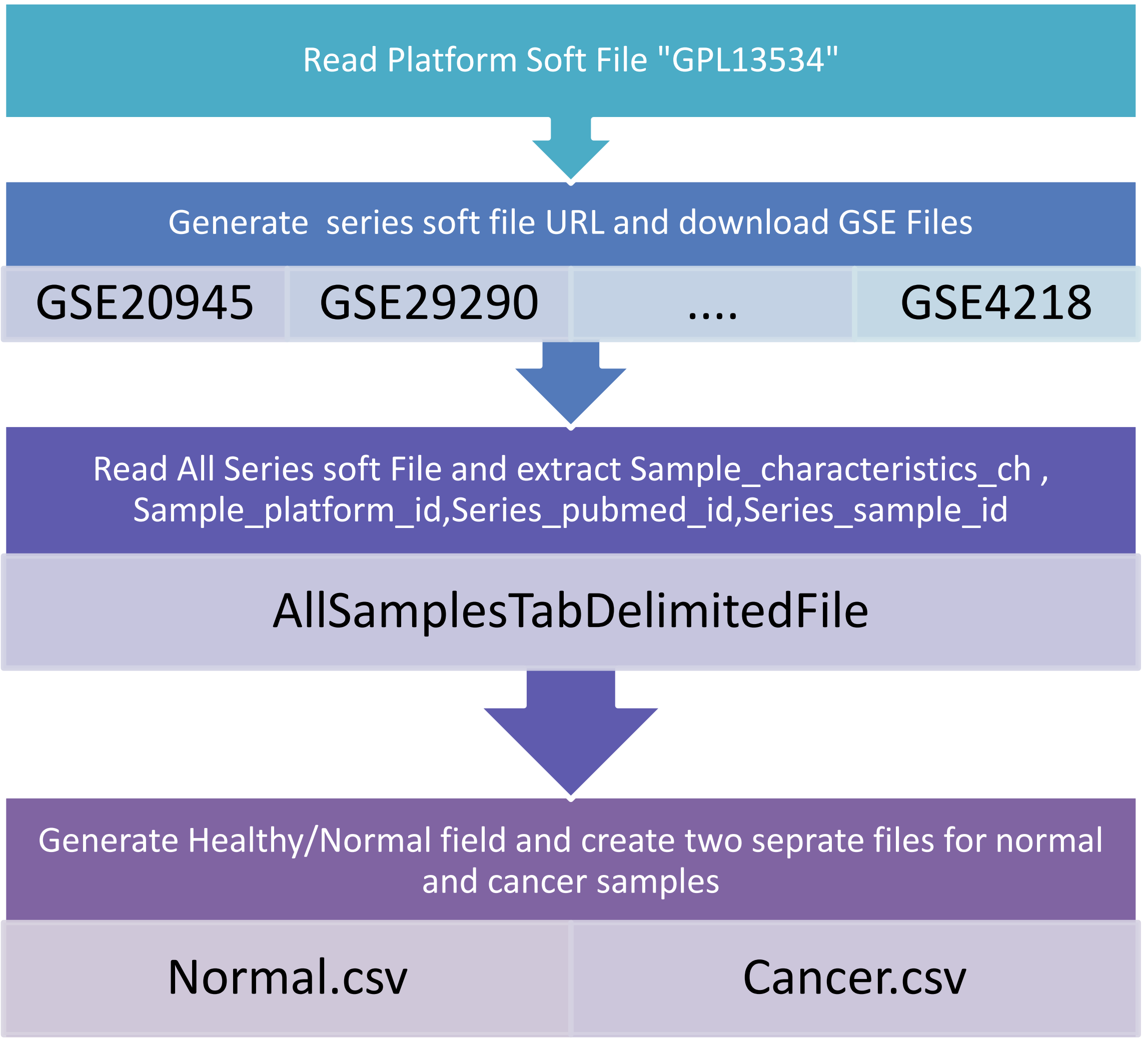

Read Platform Soft File "GPL13534"
Generate series soft file URL and download GSE Files
GSE20945
GSE29290
....
GSE4218
Read All Series soft File and extract Sample_characteristics_ch , Sample_platform_id,Series_pubmed_id,Series_sample_id
AllSamplesTabDelimitedFile
Generate Healthy/Normal field and create two seprate files for normal and cancer samples
Normal.csv
Cancer.csv

Figure 4-3 Steps in finding Normal and Cancer samples in GEO database

Series used in this study after examining platform soft file are listed in the Table 4-2.

Table 4-2 GEO data series

Series	Title
GSE20945	Transient low doses of DNA-demethylating agents exert durable antitumor effects on hematological and epithelial tumor cells
GSE29290	Evaluation of the Infinium Methylation 450K technology
GSE30338	IDH1 mutation is sufficient to establish the glioma hypermethylator phenotype
GSE36278	Hotspot mutations in H3F3A and IDH1 define distinct epigenetic and biological subgroups of glioblastoma
GSE37965	DNA methylation profiling in breast cancer discordant identical twins identifies DOK7 as novel epigenetic biomarker.
GSE38240	DNA methylation alterations exhibit intraindividual stability and interindividual heterogeneity in prostate cancer metastases
GSE38266, GSE38268	Identification and functional validation of HPV-mediated hypermethylation in head and neck squamous cell carcinoma
GSE30870	Distinct DNA methylomes of newborns and centenarians
GSE31848	Recurrent variations in DNA methylation in human pluripotent stem cells and their differentiated derivatives
GSE32148	Genome-wide peripheral blood leukocyte DNA methylation microarrays identified a single association with inflammatory bowel diseases
GSE33233	Distinct DNA methylomes of newborns and centenarians.
GSE34486	DNA methylation regulates lineage-specifying genes in primary lymphatic and blood endothelial cells
GSE36064	Age-associated DNA methylation in pediatric populations
GSE39141	Genome-wide DNA methylation profiling predicts relapse in childhood B-cell acute lymphoblastic leukaemia
GSE42118	DNA methylation changes are a late event in acute promyelocytic leukemia and coincide with loss of transcription factor binding

4.2.4 CpG sites identification

Selected samples from datasets were stored in two files which were read by a Java program to identify CpG sites with the following criteria. Each file was read read line by line and to produce vectors of M-values. Any vectors which satisfy following criteria were selected for further analysis. CpG sites which all sample's M-values were over 2.0 are selected as hypermethylated CpG sites and sites which their m-value were less than -2.0 were selected as hypomethylated as described in (Du. 2010) for variably methylated sites.

In order to identify four classes of CpG sites four experiments were designed for the selection of normal and cancer samples. For experiment "A", sites in which 60 percent of samples were hypermethylated in cancer, 60 percent hypomethylated in normal were selected for expriment C sites. Where 60 percent of samples were hypomethylated in normal and 60 percent hypermenthylated in cancer selected for always methylated sites and for never methylated sites we made the hypomethylation threshold the restrictor and sites that had 0.1 beta values or less were selected. CpG sites in each class with more than 50 percent overlap were removed.

The overall views of these three steps are:

1) Making raw data standard for all series.
2) Dataset Selection
3) CpG site identification

Table 4-3 CpG site identification

```
Algorithm 4-2    CpG site identification

Input : Selected data series , Illumina platform file
For each CpG site in Illumina platform file
    For each sample in all data series
        Calculate intensity ratio of CpG site for sample
    EndFor
    Assign class to CpG site
EndFor
Output : CpG sites with class
```

Figure 4-4 shows details of each section given in the previous steps.

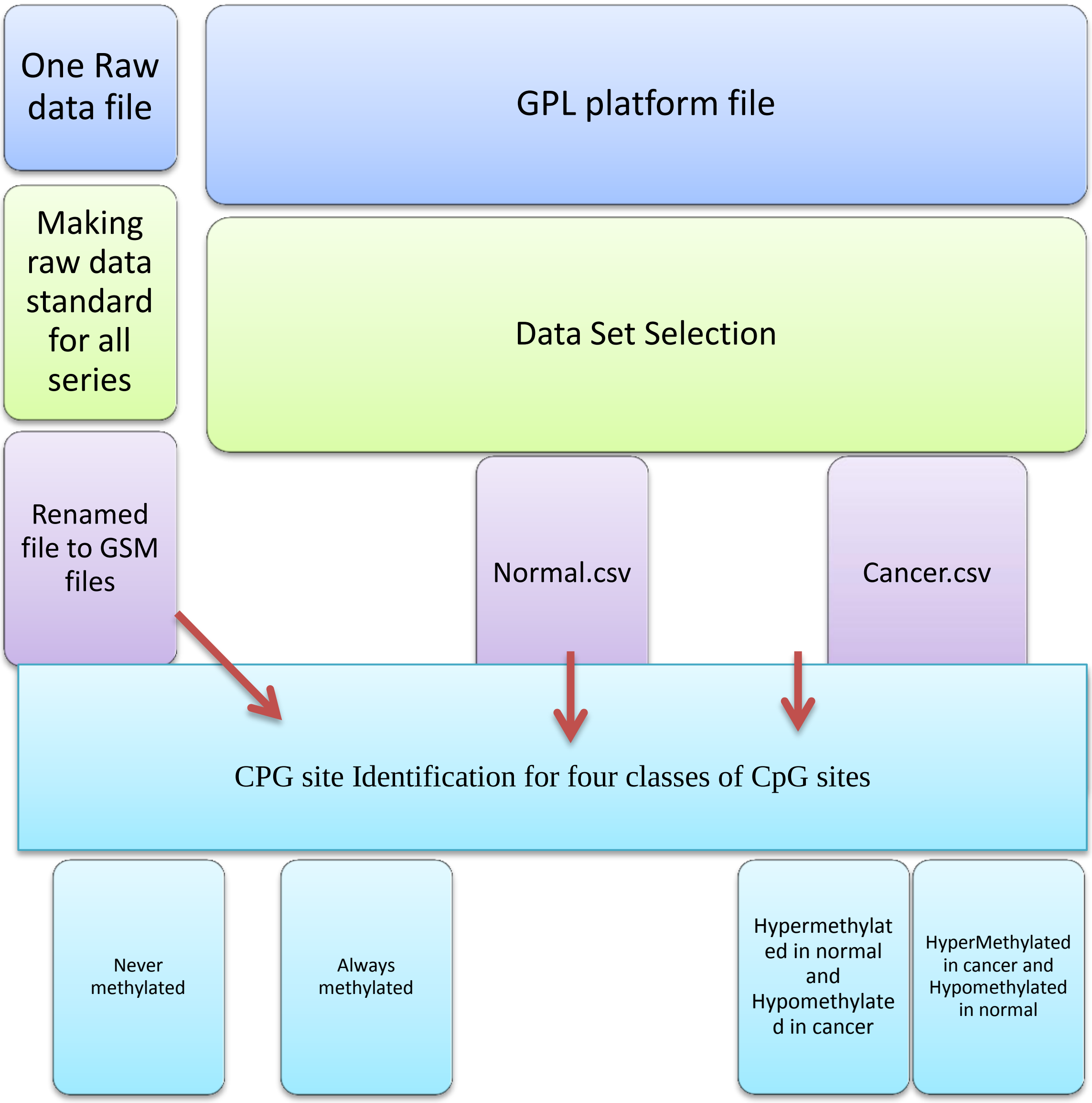

One Raw data file
GPL platform file
Making raw data standard for all series
Data Set Selection
Renamed file to GSM files
Normal.csv
Cancer.csv
CPG site Identification for four classes of CpG sites
Never methylated
Always methylated
Hypermethylated in normal and Hypomethylated in cancer
HyperMethylated in cancer and Hypomethylated in normal

Figure 4-4 CpG site identification steps

4.2.5 Motif Discovery

MEME (Multiple EM for motif Elicitation) suit (Bailey *et al.*, 2006) was used for discovering motifs. We used the default MEME settings with the ZOOPS (zero or one motif per sequence) parameter for discovering motifs for each class of identified CpG sites. Sixty base pairs up and down stream of each CpG site is used as input of the MEME for each class and the five best motifs according to their E-value in MEME and their probability matrix were selected for further analysis. Next these 20 motifs were used as input to the MAST tool with all sequences. The MAST program removed 2 motifs which have more than 60 percent overlap with others resulting in finally 18 motifs being selected by MAST used for further analysis.

4.2.6 Using motifs for Classification

A Java program was developed to convert MAST hit results to a feature matrix and the results used in the weka package to evaluate the potential of using these motifs for classification of four classes of CpG sites. J48, Naïve Bayes and support vector machines were used as a classification tools for this purpose.

4.3 Results

4.3.1 CPG sites and their classes

Using the method described, 653 CpG sites were identified: 447 CpG sites are in never methylated class, 148 sites are in always methylated, 51 hypomethylated in normal and hypermethylated in cancer (Experiment A) and 7 sites hypermethylated in normal and hypomethylated in cancer (Experiment C). We mapped the positional relationship of the CpG sites to CpG islands in the UCSC browser. 81 CpG sites were not in any positional relationship with a CpG Island. Never methylated sites are predominantly within islands. Most of the CpGs in the two classes of variably methylated sites have no relationship to any CpG islands. Always methylated CpGs are spread among different classes of positional relationships to UCSC CpG islands.

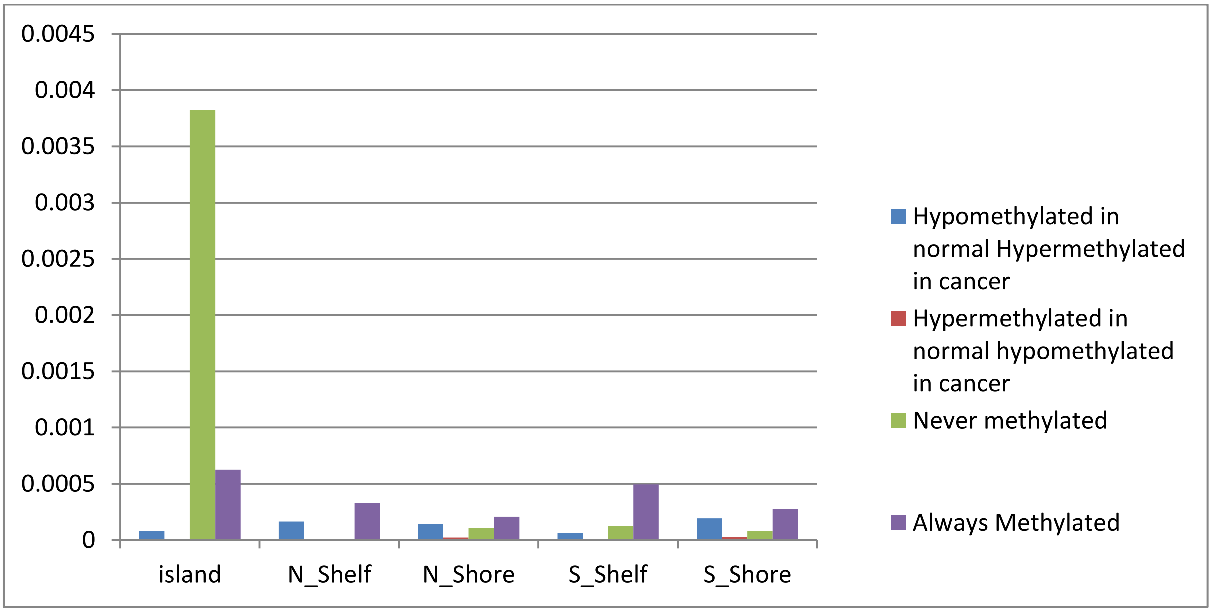


Figure 4-5 Proportion of CpG site to UCSC CpG Island to total number of CpG sites

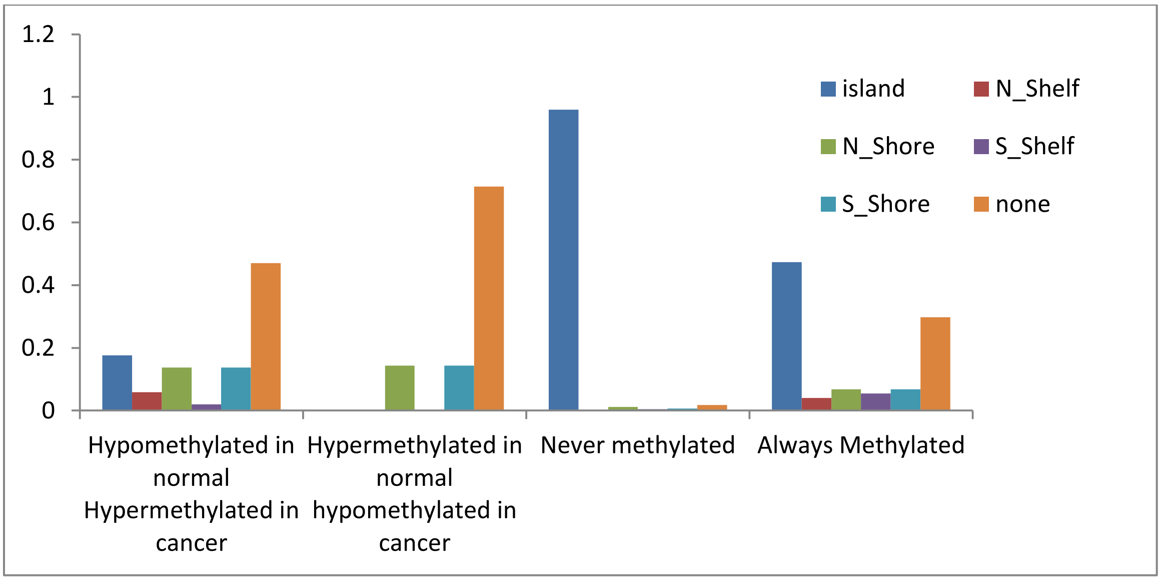


Figure 4-6 proportion of each relation to total number of CpG sites in each class

4.3.2 Genes neighbouring the CpGs in Experiment A

The genes neighbouring the CpGs found in Experiment A are listed in the Table 4-4 along with their function as described by Cormine software (Coremine 2013) .

Table 4-4 Genes neighbouring expriment A

UCSC_RefGene_Name	Function	Cancer Involvement
PPFIA1	Cell motility, apoptosis. invasion suppressor gene	Amplified breast and head and neck cancers (cell trying to avoid invasion)
EXD3	gene silencing activity	None known
PTPRCAP	Protein tyrosine phosphatase receptor, apoptosis	Hypermethylated in many cancers. Implicated in tumorigenesis
LOC100129637	Unknown	Unknown
TMC6	DNA repair	Variants seen in Cervical Cancer
BIN2	endocytosis	Abrogated in Myeloproliferative neoplasms
C17orf101	oxidoreductase activity	None known
PPFIA1	Cell division and chromosome partitioning, cell motility, apoptosis	Amplified breast and head and neck cancers (cell trying to avoid invasion)
MAP1D	aminopeptidase activity, phosphorylation	Over expressed in colon cancer
SORBS2	cytoskeletal protein, cell migration, apoptosis	Downregulated in pancreatic, thyroid and cervical cancer
ELMO1	endocytosis, phagocytosis, apoptosis cell migration	Promote cell invasion in ovary, colon and brain cancer
ERI3	exonuclease activity, cell division, signal transduction, DNA replication	Increased in breast cancer
LAG3	regulation of leukocyte activation, cell proliferation, apoptosis	Involved in many different cancers assisting in detection avoidance and resistance to apoptosis
PLCB2	phospholipase C activity, calcium ion binding, signal transduction apoptosis	Highly expressed in Breast cancer promoting mitosis and migration of tumour cells
SPN	regulation of inflammatory response to antigenic stimulus, induction of apoptosis by extracellular signals	Significantly expressed in lymphomas
PARP10	NAD+ ADP-ribosyltransferase activity, cell proliferation, apoptosis	Inhibits transformation of cells, in KEGG small cell lung cancer
MYO1G	myosin complex, cell division, DNA hypermethylation	Involved in survival leukaemia and breast

UCSC_RefGene_Name	Function	Cancer Involvement
		cancer cell
CD6	Cell Adhesion Molecule (CAM), apoptosis cell proliferation	Aberrantly expressed in leukemia
RAPGEF1	intracellular signaling cascade, small GTPase mediated signal transduction, cell proliferation, apoptosis	Upregulation in breast, lung, gastrointestinal and gynaecological cancers
NCKAP1L	Regulation of actin cytoskeleton, cell proliferation apoptosis	Down regulated in many cancers
TRAF5	MAPK signaling pathway, Apoptosis, RIG-I-like receptor signaling pathway, Adipocytokine signaling pathway,	Expressed in lymphomas and small cell lung cancer
C3orf21	regulation of protein amino acid phosphorylation	None known
CA6	carbonate dehydratase activity, zinc ion binding, cell proliferation, apoptosis	Expressed in ovarian and breast cancers
CCDC88C	regulation of protein amino acid phosphorylation, cell migration	Involved in tumour invasion
TNRC18	DNA binding, lipid transporter activity	None known
ANO8	chloride channel activity, embryo development	Over expressed in many cancers
PTPN7	protein tyrosine phosphatase activity, cell proliferation, apoptosis	Implicated in blood cancers
TBC1D16	regulation of Ras protein signal transduction, gene expression	Involved in melanoma progression
STK16	protein amino acid phosphorylation, cell growth, apoptosis	Over expressed in tumour cells
RFFL	zinc ion binding, RING type, apoptosis, DNA methylation	Involved in myeloma
SPN	negative regulation of adaptive immune response, positive regulation of cell death, apoptosis	Supressed in many tumours
PC	transcription repressor activity	Upregulated in many tumours (renal, small cell lung, sarcoma)
MIR365-1	Not known	Not known
RADIL	cell adhesion, forkhead and RAS associated	None Known
FBXL16	proteolysis, macromolecule catabolic process, cell proliferation, cell cycle	Down regulated in many cancers
LMNB2	lamin filament, cytoskeleton, cell cycle, methylation,	Down regulated in prostate, gastric, skin and

UCSC_RefGene_Name	Function	Cancer Involvement
	apoptosis	leukaemia cancers
JAK3	positive regulation of leukocyte activation, apoptosis, signal transduction, phosphorylation	Upregulated in many cancers
KCNJ8	ATP-activated inward rectifier potassium channel activity, vasodilation, apoptosis, gene expression	Upregulated in nasopharyngeal carcinoma

The genes were analysed for their predicted DNA binding protein sites including 5000bp up and down stream of the coding regions. The most enriched predicted binding site according to oPossum was MZF1_1-4 a zinc finger TF which may be one regulator of transcriptional events during hemopoietic development. It has been implicated in upregulating apoptosis. MZF-1, was revealed to interact with LDOC1 and enhance the activity of LDOC1 for inducing apoptosis (Inoue *et al.*, 2005). Thus if methylation in cancer prevents its binding this could affect the cells ability to enter apoptosis. Also MZF-1 has been shown to supress tumourgenicity (Hsieh *et al.*, 2007) The second most enriched, Klf4, contributes to the down-regulation of p53/TP53 transcription (Rowland *et al* 2005) which is important in tumourgenesis.

These genes are enriched for the E2F family of transcription factors as assessed by oPossum software (Sui *et al* 2007) 19 of the genes are predicted to bind (equivalent to 55.88%). This compares to 32.77% of all genes in the human genome.

On DAVID (Huang, Sherman and Lempicki, 2008) analysis of the genes in the most enriched cluster was one with a functional key word of “Apoptosis” indicating that a large proportion of these genes are involved or predicted to be involved in apoptosis.

The genes neighbouring the CpGs found in Experiment C are listed in the table below along with their function as described by Cormine software (Coremine 2013).

Table 4-5: The genes neighbouring the CpGs found in Experiment C

UCSC RefGene Name	Function	Cancer Involvement
RPTOR	Androgen receptor activity, kinase activity, telomerase activity, kinase activity, cell growth, cell cycle, insulin signalling	Up regulated in multiple cancers
C22orf9	Not Known	None Known
NOS1AP	Signal transduction, gene expression, cell migration, cell proliferation	Associated with breast cancer progression
RGS12	Signal transduction, cell cycle. RNA interference, apoptosis, SNAP receptor activity	Mutated in colorectal tumours

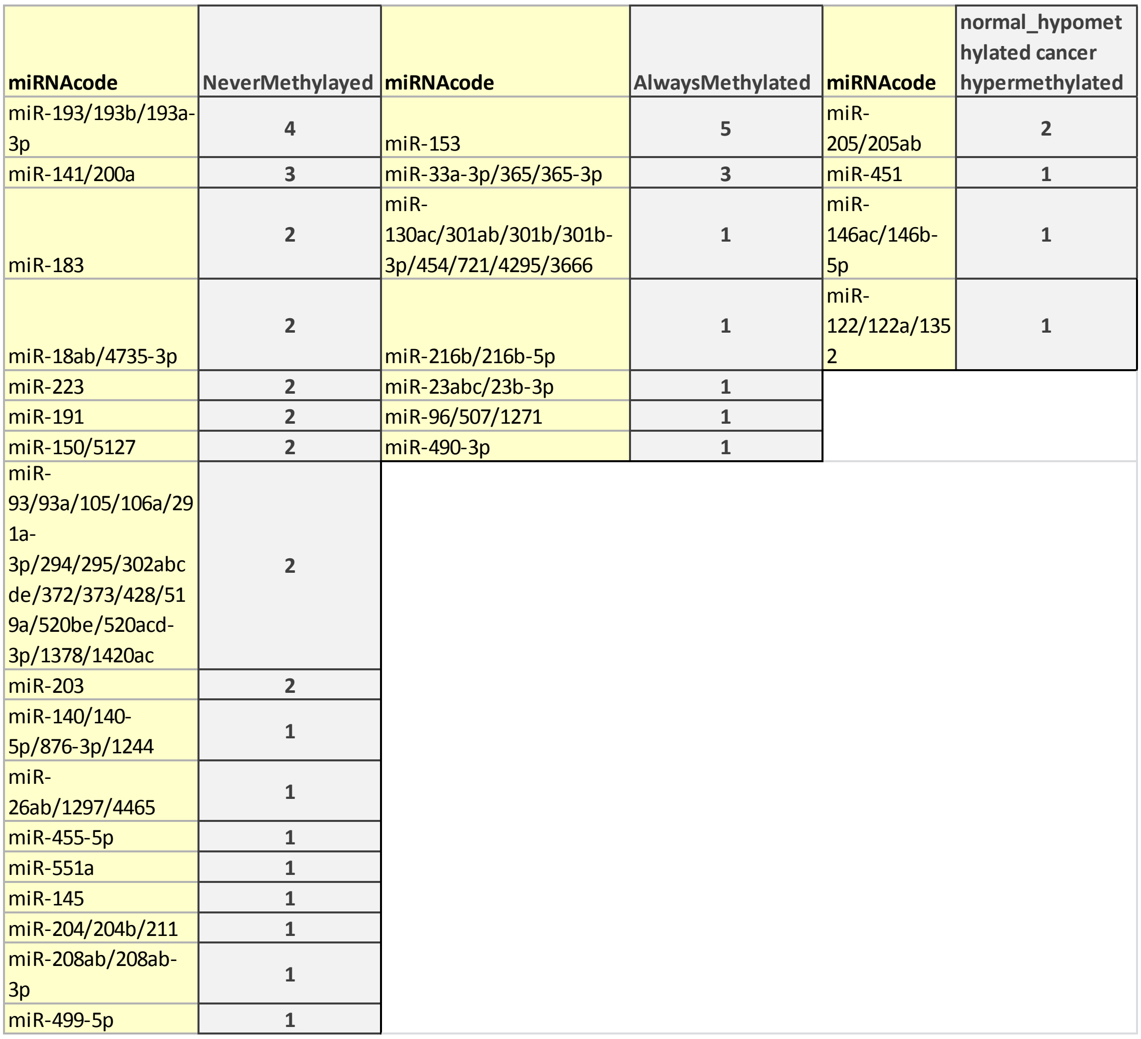
When analysed for transcription factor binding NOS1AP was the protein which had the most TF motifs associated with it and these include Sox2, RREB1, Evi1, NR3C1 with the highest z-score as determined by oPossum and notably E2F1. None of the other genes in this list were predicted to bind E2F type transcription factors.

There are too few genes in this list to analyse using DAVID functional clustering software.

4.3.3 MicroRNA results

In a query of UCSC tracks, track “miR Sites High” in table “miRcode Predicted MicroRNA Target Sites microRNA” was investigated. A total number of 241 overlaps with microRNAs were found. The results are depicted in chart need the number of the chart here. There were 148 overlaps with never methylated sites, 25 with Experiment A, 68 with always methylated sites and 0 in experiment C.

64 of these hits are unique .17 are unique to nevermethylated sites, 4 microRNA are unique to normal_hypomethylated and cancer hypermethylated (Expriment A), 7 are unique to always methylated sites. These are depicted in the following figure.

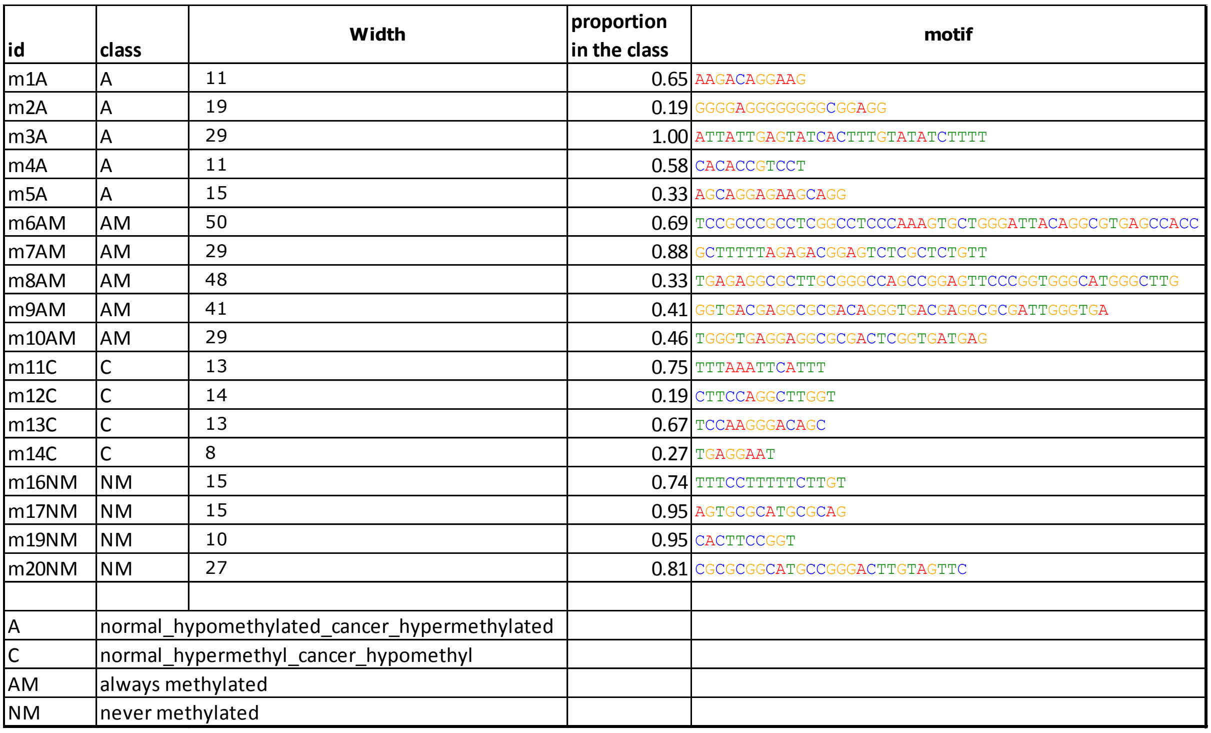


miRNAcode	NeverMethylayed	miRNAcode	AlwaysMethylated	miRNAcode	normal_hypomethylated cancer hypermethylated
miR-193/193b/193a-3p	4	miR-153	5	miR-205/205ab	2
miR-141/200a	3	miR-33a-3p/365/365-3p	3	miR-451	1
miR-183	2	miR-130ac/301ab/301b/301b-3p/454/721/4295/3666	1	miR-146ac/146b-5p	1
miR-18ab/4735-3p	2	miR-216b/216b-5p	1	miR-122/122a/1352	1
miR-223	2	miR-23abc/23b-3p	1		
miR-191	2	miR-96/507/1271	1		
miR-150/5127	2	miR-490-3p	1		
miR-93/93a/105/106a/291a-3p/294/295/302abcde/372/373/428/519a/520be/520acd-3p/1378/1420ac	2				
miR-203	2				
miR-140/140-5p/876-3p/1244	1				
miR-26ab/1297/4465	1				
miR-455-5p	1				
miR-551a	1				
miR-145	1				
miR-204/204b/211	1				
miR-208ab/208ab-3p	1				
miR-499-5p	1				

Figure 4-7 Unique microRNA in each class of CpG sites

4.3.4 Discovered motifs

Figure 4-8 show the motifs which were found in four classes of CpG sites and their width and classes using MEME.



id	class	Width	proportion in the class	motif
m1A	A	11	0.65	AAGACAGGAAG
m2A	A	19	0.19	GGGGAGGGGGGGGCGGAGG
m3A	A	29	1.00	ATTATTGAGTATCACTTTGTATATCTTTT
m4A	A	11	0.58	CACACCGTCCT
m5A	A	15	0.33	AGCAGGAGAAGCAGG
m6AM	AM	50	0.69	TCCGCCCGCCTCGGCCTCCCAAAGTGCTGGGATTACAGGCGTGAGCCACC
m7AM	AM	29	0.88	GCTTTTTAGAGACGGAGTCTCGCTCTGTT
m8AM	AM	48	0.33	TGAGAGGCGCTTGCGGGCCAGCCGGAGTTCCCGGTGGGCATGGGCTTG
m9AM	AM	41	0.41	GGTGACGAGGCGCGACAGGGTGACGAGGCGCGATTGGGTGA
m10AM	AM	29	0.46	TGGGTGAGGAGGCGCGACTCGGTGATGAG
m11C	C	13	0.75	TTTAAATTCATTT
m12C	C	14	0.19	CTTCCAGGCTTGGT
m13C	C	13	0.67	TCCAAGGGACAGC
m14C	C	8	0.27	TGAGGAAT
m16NM	NM	15	0.74	TTTCCTTTTTCTTGT
m17NM	NM	15	0.95	AGTGCGCATGCGCAG
m19NM	NM	10	0.95	CACTTCCGGT
m20NM	NM	27	0.81	CGCGCGGCATGCCGGGACTTGTAGTTC
A	normal_hypomethylated_cancer_hypermethylated			
C	normal_hypermethyl_cancer_hypomethyl			
AM	always methylated			
NM	never methylated			

Figure 4-8 Motifs discovered in DNA sequence of CpG sites

4.3.5 Classification results

We use the matrix obtained by MAST to evaluate the potential of using these motifs for classification of four classes of CpG sites. J48, Logistic and support vector machines were used as a classification tools for this purpose. And specifically the two classes of variably methylated sites.

4.3.6 WEKA

Using 10-fold cross validation methodology we used 3 algorithms to classify the CpG sites according to their class, based on their motifs. 1) a support vector machine algorithm resulted in 69.5253 % correctly classified 2) a logistic algorithm resulted in 73.9663 % 3) a J48 algorithm resulted in 71.2098 % correct prediction of each CpG site into one of the 4 classes (never methylated , always methylated classes, Hypermethylated in normal and Hypomethylated in cancer or vice versa).

Since the CpGs that distinguish between normal and cancer calls are of most interest we performed a similar classification analysis using the Hypermethylated in normal and Hypomethylated in cancer or vice versa results only.

Using 10-fold cross validation methodology we used the 3 algorithms to classify the distinguishing CpG sites according to their class based on their motifs. 1) a support vector machine algorithm resulted in 98.2759 % correctly classified 2) a logistic algorithm resulted in 96.5517 % 3) a J48 algorithm resulted in 94.8276 % prediction of each of the 2 classes of CpG, Hypermethylated in normal and Hypomethylated in cancer or vice versa.

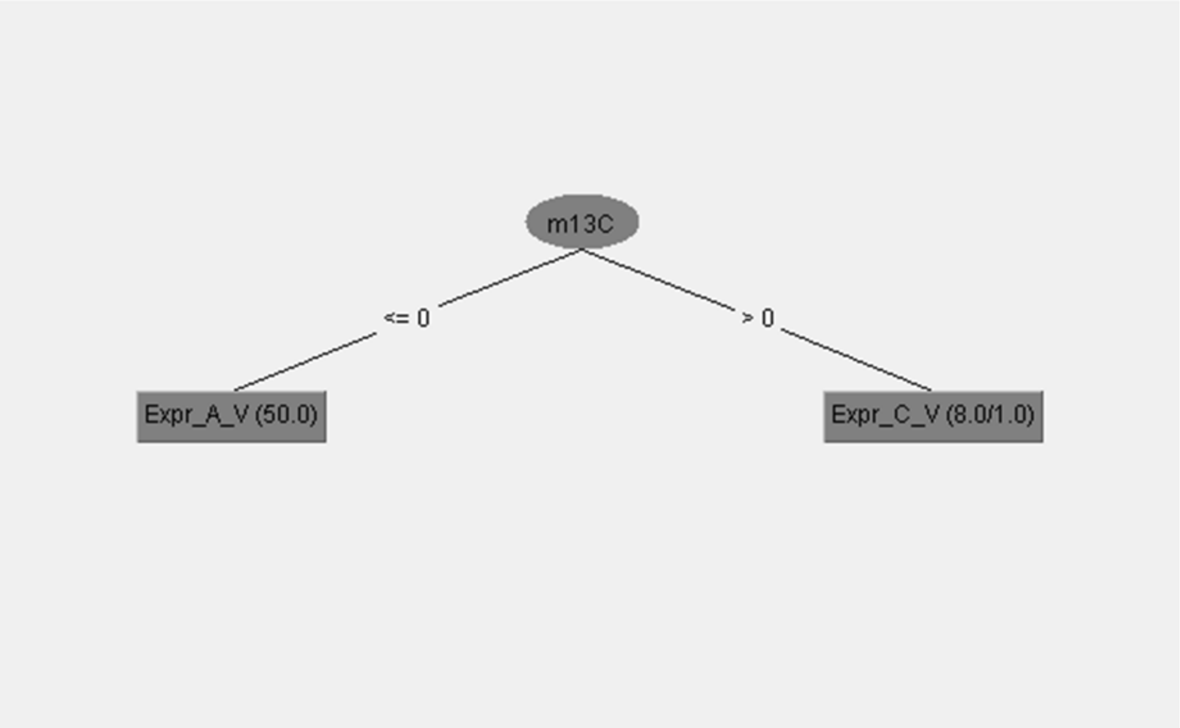


Figure 4-9 This diagram illustrates that the m13C (TCCAAGGGACACC) motif doesn't occur in the flanking DNA sequences occurring in 50 out of 51 of the CpGs identified in experiment A and occurs in all 7 of the sequences surrounding CpGs identified in experiment C.

The m13C motif contains the binding motif for the EBF1 and the RME transcription factors which have been shown to act as a tumour suppressor in multiple tumour types notably leukaemia's and colon cancer (Liao 2009 and Chen et al 2012). The NR2F1 binding motif is also present in m13C, another transcription factor with oestrogen response element binding which is down regulated in many tumour types (Thompson et al 2012). Also NR4A2 which is a nuclear orphan receptor involved in neoplasms and a potential therapy target binds to this sequence (Deutsch et al. 2012).

4.3.7 Summary

This chapter provides details of CpG sites identification by processing multiple data series. These series were obtained from GEO database. Because each of these data series submitted in different formats their formats has to be standardise. Stricter standards in raw data format to GEO and similar database can provide easier data reuse. Processing these data found 653 CpG sites in four classes of DNA methylation. Further analysis revealed motifs for each class these motifs shows potential discriminative advantage. Genes neighbouring identified CpG sites shows that they are related to cancer as it is described in section 4.3.2.

Chapter 5 - Grid enabled workflow based feature generation and feature subset selection

5 Background

In order to predict DNA methylation and creates a model for methylation, features related to DNA sequence of methylated CpG site should be identified first. Many features can influence DNA methylation and many new features have been discovered as technology and science progresses. A system is needed that generates these features or queries existing features, based on DNA sequences. This system should be scalable so if new features needed to be investigated the whole system does not change. Because generating features for each CpG site is independent of other sites these tasks can be distributed over multiple machines. The gUse workflow portal was selected to achieve this. gUse has a workflow management service and can be connected to a diverse range of distributed computing infrastructures. The workflow management service provides a modular way of feature generation. Connecting workflow to a distributed system provides the facility to speedup feature generation and feature subset selection tasks. Two main advantages of using gUse are application specific API (Application program interface) which enables the developer to develop a form-based interface for workflows. Therefore the user can easily interact with the application like any other web based application without the need to know the details of the workflow. BOINC was chosen as a grid middleware because it has the potential to expand worker machines to volunteer desktops although other similar middlewares with similar features can be used. BOINC was used in many biological projects as discussed in section 2.5 and is supported by gUse. The high level algorithm feature generation is provided in the Table 5-1.

Table 5-1 Sequential feature generation algorithm

```
Algorithm 5-1    Feature generation

Input : DNA sequences
1. For each DNA sequence
2.     For each feature generation program
3.         Run the feature generation program on the sequence
4.         Store the result in the feature matrix
5.     EndFor
6. EndFor
Output : Feature matrix
```

There are two for loops in the algorithm 5-1 in the Table 5-1. Because running feature generation on each sequence is independent of other sequences, we can distribute the tasks on these two loops on separate machines to achieve speedup. This chapter discusses this approach and the algorithms for four feature generation programs are provided. The current feature generation programs for the DNA methylation studies as mentioned in the previous chapter are using the sequential non-modular way of feature generation. Sometimes this task is done manually in spread sheets and then the feature matrix aggregated as an input to machine learning algorithm. This chapter provides the modular and distributed example of feature generation. The use of tools like BOINC and gUse does not influence the design of these kinds of systems, any system with similar properties can be used to achieve these tasks.

5.1 Objectives

The work in this chapter had two objectives:

a) The first objective was to create a scalable system. This ensures that generating new features has less of an impact on the other parts of the system.

b) The second objective was to accelerate feature generation and feature subset selection by connecting workflows to a desktop grid.

5.2 Overview of chapter

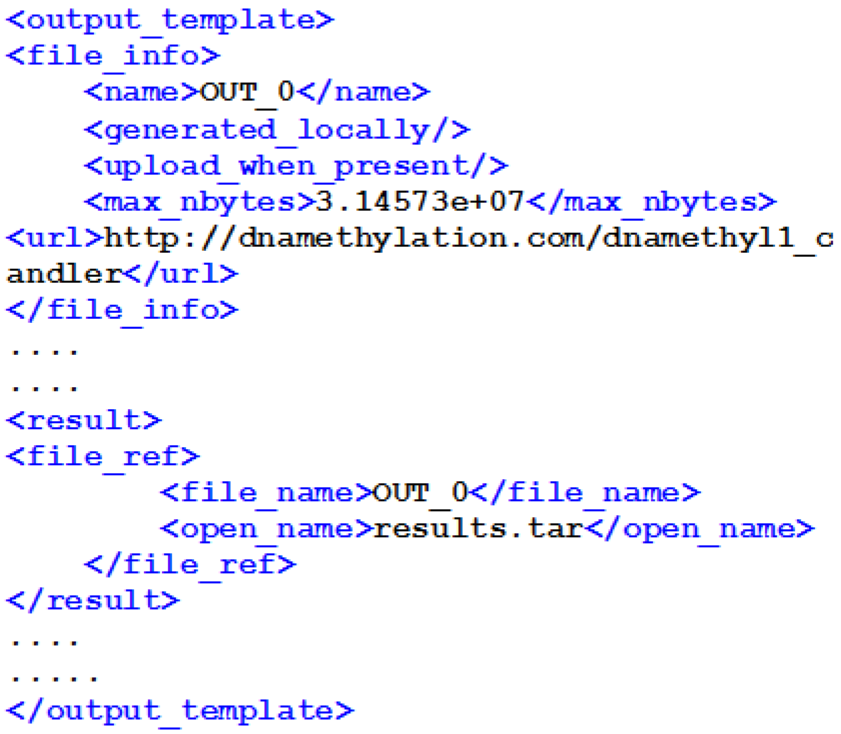
This chapter discusses the method used to port the five applications to BOINC infrastructure. Section 5.3 gives an overview of BOINC and its applications in the science. Section 5.4 explains the GenWrapper (Marosi, Balaton and Kacsuk, 2009) which is used here to port applications to BOINC and also other methods of porting applications are compared in this section. Section 5.5 gives details of porting four different EMBOSS (Rice, Longden and Bleasby, 2000) applications to BOINC. Section 5.6 gives details of porting the hill climbing feature search application.

5.3 BOINC desktop grid computing

BOINC is open source software which can be used for volunteer computing and grid computing. It has several subsystems which can be classified into two main groups, client side and server side. BOINC projects are created to solve specific computational problems. These projects cover many fields of science among them astrology, physics, chemistry, biology, mathematics (BOINC 2013). Client machines are connected to BOINC using the URL of the project. Workunits contain the necessary binaries and input files for achieving a specific task and generating output files. Workunits specifications are stored in two xml files. These xml files contain a list of input and output files.

5.4 Workunit submission

Generator program creates workunits. There are different methods for creating workunits. They can be created by BOINC API, DC – API, Generic master programs and 3G bridge (Marosi, Kovács and Kacsuk, 2013). BOINC also provides a workunit creator command line tool "create_work" which can be used for simple applications or test purposes. For work generation each workunit needs input and output template file or job template files, these are XML files which define the job description. Example of parameters defined in the template files are the name of the input and output files, the size of the output file, the numbers and other job related options. Figure 5-1 shows a portion of an output file. The input file is similar but it has a <workunit> element instead of <result> element. BOINC client on the worker machine uses two different directories for keeping the files and running workunits. Because of this, there are two names for files: one is a physical name defined in the <file_name>, and other is a logical name expected by worker program, which is defined in the <open_name>. Further details can be found in the BOINC project website (BOINC 2013).



```
<output_template>
<file_info>
    <name>OUT_0</name>
    <generated_locally/>
    <upload_when_present/>
    <max_nbytes>3.14573e+07</max_nbytes>
<url>http://dnamethylation.com/dnamethyl1_c
andler</url>
</file_info>
....
....
<result>
<file_ref>
        <file_name>OUT_0</file_name>
        <open_name>results.tar</open_name>
    </file_ref>
</result>
....
.....
</output_template>
```

Figure 5-1: Snapshot of BOINC output template file.

5.5 GenWrapper

Running a program under BOINC needs has some requirements. The binary code of a program should be able to to communicate with BOINC, either using API or other techniques which summarised in the Table 5-2.

Table 5-2 Comparison of different way of porting application boinc platform (Marosi, Kovács and Kacsuk, 2013)

	BOINC API	DC-API	BOINC Wrapper	GenWrapper	GBAC
Supported programming languages	C/C++/FORTRAN / Python	C/C++/Java/Python	Control-flow description in XML	POSIX shell scripting	None required
Legacy application support	No	No	Yes	Yes	Yes
Native application support	Yes	Yes	Partial	Partial	Partial
Application level checkpointing	Yes	Yes	Partial	Partial	Partial
Type	Native	Native	Native	Native	Virtualized
Requires client-side third party software	No	No	No	No	Yes (VirtualBox predeployed)

GenWrapper provides a generic method for porting legacy applications to BOINC, by using GenWrapper multiple binaries can be combined and they can be controlled by Linux shell script. GenWrapper was selected for the ported applications in this research because it facilitates legacy application’s files processing and easy way to use BOINC communication commands. Figure 5-2 shows how GenWrapper communicates with the BOINC client application (Marosi, Balaton and Kacsuk, 2009).

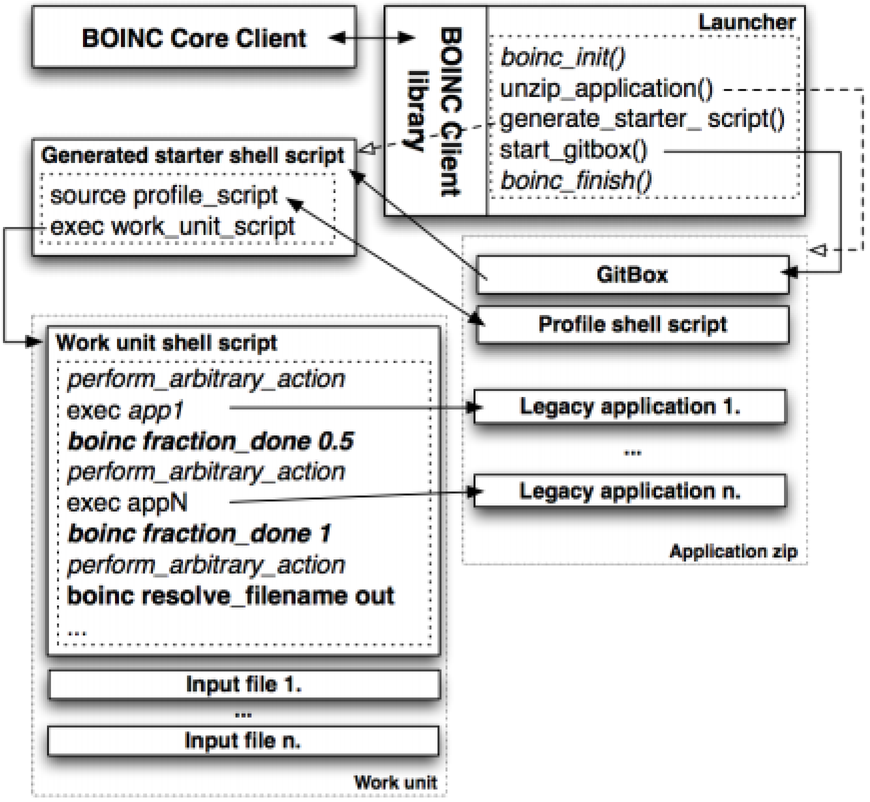


Figure 5-2. Diagram shows how different part of GenWrapper interacts with each other and communication with BOINC client (Marosi, Balaton and Kacsuk, 2009)**.**

5.6 gUse (Grid and Cloud User Support Environment)

gUse is a distributed computing infrastructure gateway framework. It provides the user with the access to the distributed computing infrastructure. gUse (Kacsuk *et al.*, 2012) is implemented as the portlet in the Liferay portal and benefits from the features provided by Liferay. From the architectural perspective, these features are web applications deployed in a web container. Liferay itself is an open source portal developed in java which can be deployed in the Tomcat servlet container. gUse uses the MySQL database management system. Figure 5-3 shows the simplified architecture of the system.

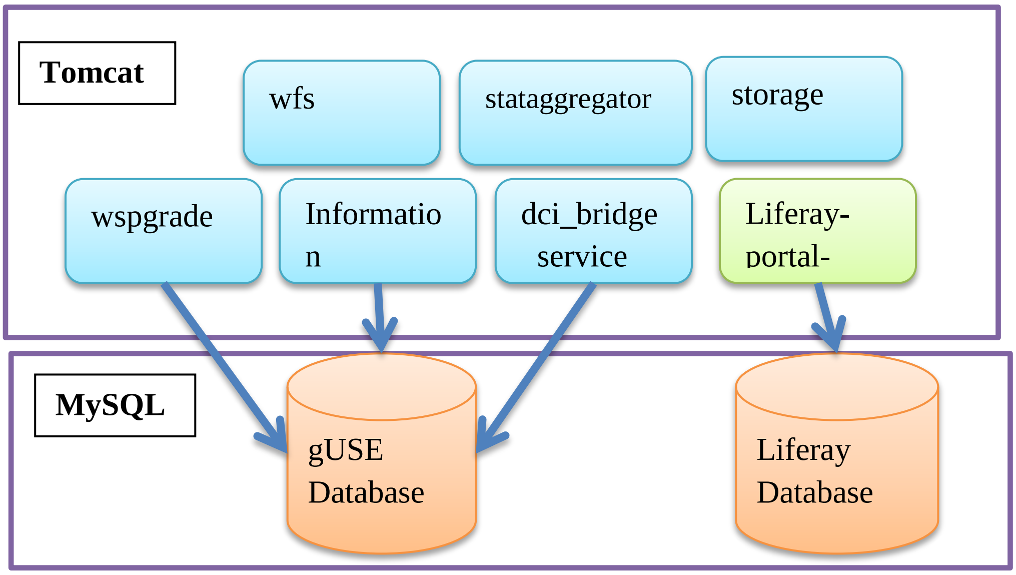


Figure 5-3. gUse server architecture.

5.7 Test grid Infrastructure preparation

This section provides the details of the infrastructure which was prepared to test the system.

The BOINC project was created in the Debian Linux 6.0 machine. The applications were ported to BOINC by GenWrapper, which was reviewed in the previous section.

gUse system was also installed into the same Debian machine and connected to BOINC via the 3g-bridge. The client machines were seven Microsoft Windows virtual machines on VMware. VMware contained the BOINC client connected to the project. Figure 5-5 shows how the client side subsystems are working with each other. Figure 5-4 shows client and server side of the test system.

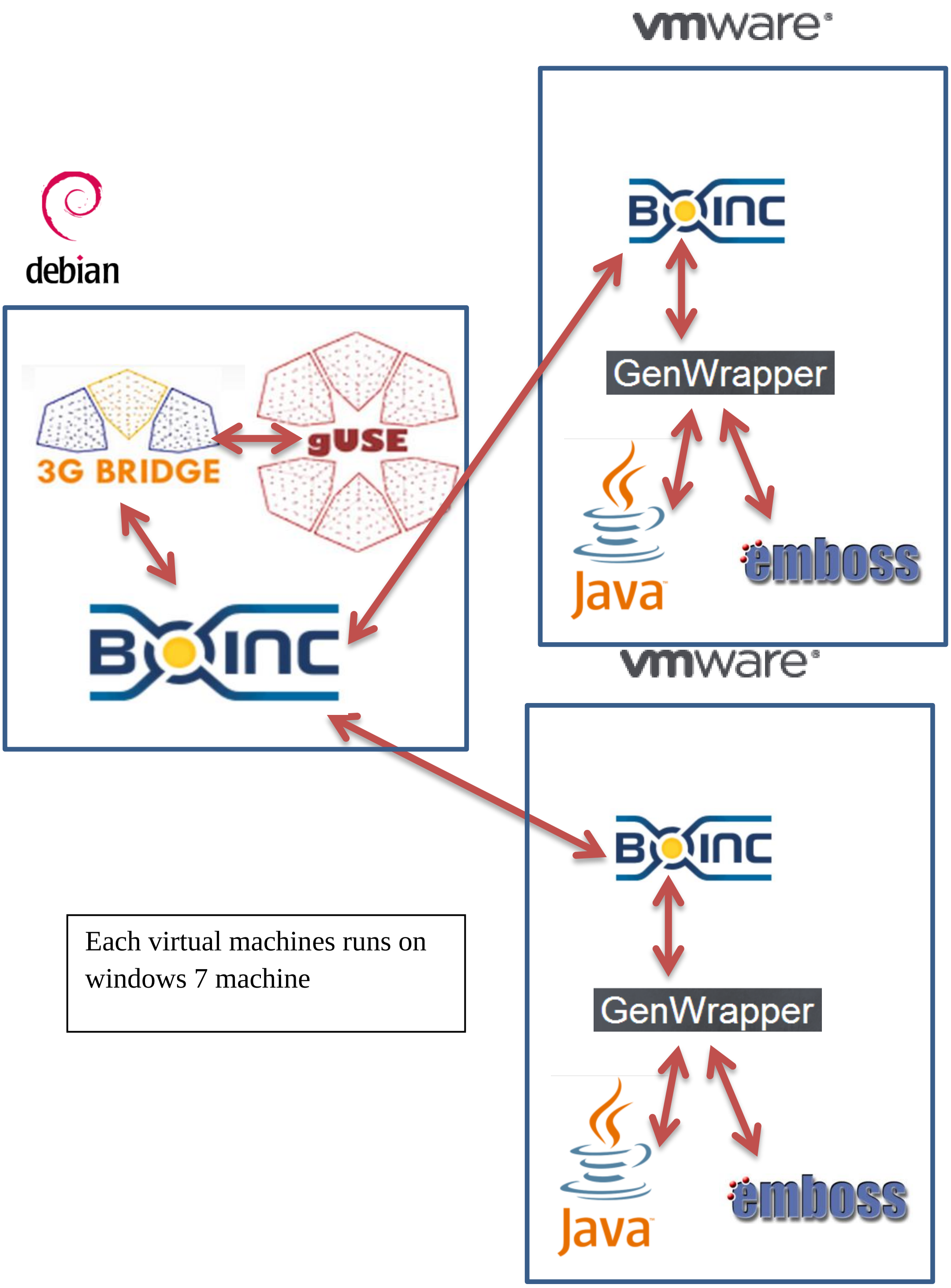


Figure 5-4. Different software used in the client and server side of the system and their interaction with each other.

Inside the virtual machine, when a job is available, BOINC calls the application and application calls GitBox to run the shell script provided by the application. Shell script then runs the programs in the virtual machine.

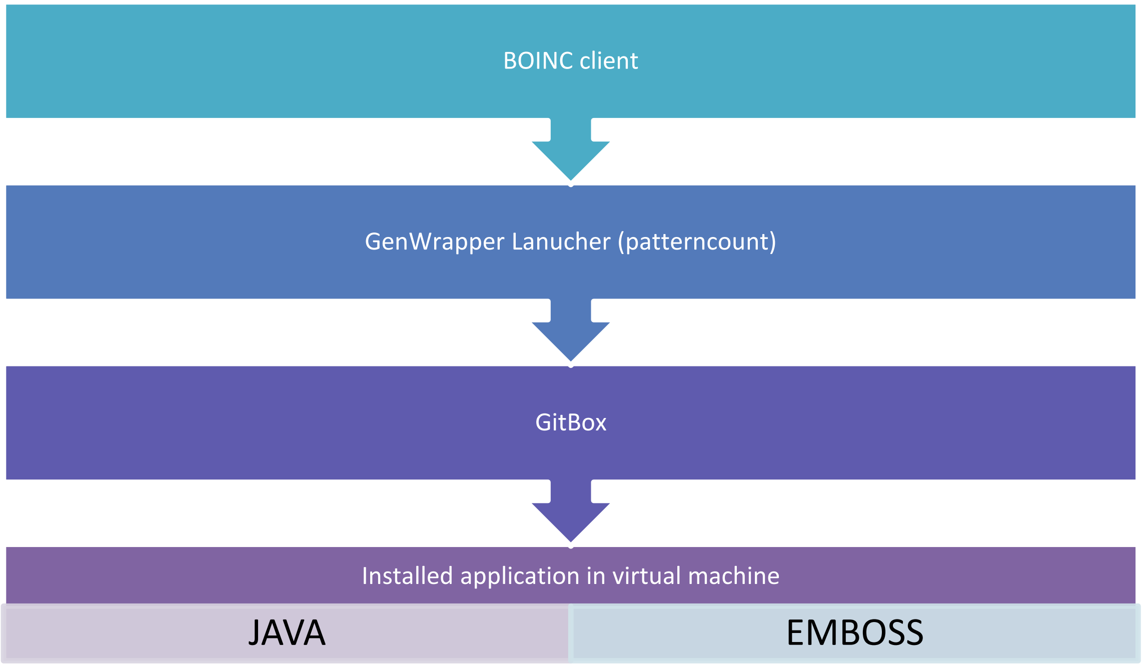

BOINC client
GenWrapper Lanucher (patterncount)
GitBox
Installed application in virtual machine
JAVA
EMBOSS

Figure 5-5. The client side view of program used for feature generation and feature subset selection.

5.8 Grid enabling selected EMBOSS applications

The following section provides details of the grid enabling 4 EMBOSS applications. All of these applications had sequences in FASTA format as their input. Outputs are tables with the sequence ids as rows and the columns are the feature names.

5.8.1 Banana

Banana is one of the programs of the EMBOSS application suite. It can predict the bending of a normal DNA double helix (Rice, Longden and Bleasby, 2000). Banana was used for generating two features. These features are in the class “structural features”. Genwrapper was used in order to port the application into the BOINC desktop grid environment. Banana generates one file for each sequence which had two values (one for bend and one for curve) for each base pair in the sequence. The average of these values was used as the feature for the sequence. Gitbox, which is part of GenWrapper supports “awk”, which was used to read the profile files generated by banana to produce data points in table. The following code shows how two values are extracted from profile files.

```
bend=`awk 'BEGIN{bend=0}{if(NR >1) bend=bend+$2} END{print bend/(NR-
1)}' < $f".profile"`
curve=`awk 'BEGIN{curve=0}{if(NR >1) curve=curve+$3} END{print
curve/(NR-1)}' < $f".profile"`
```

Table 5-3: Grid Enable banana program

```
Algorithm 5-2    Grid Enabled banana

Input : DNA sequences File

  1. Generate one file per sequence in the file
  2. For all file in directory
  3.      Run the banana program
  4. EndFor
  5. For all profile files in the directory
  6.      Use first field as bend value and second as curve and
          combine all values
  7. EndFor

Output : Feature matrix  banana A
```

5.8.2 Btwisted

Btwisted is an application which calculates the overall twist of the DNA sequence and the stacking energy (btwisted 2013). Similar to banana it generates one file per sequence. In each file there are five records: total twist, total turns, average base per turn, total stacking energy and average stacking energy.

Similarly GenWrapper was used to port the application to the BOINC desktop grid computing platform. Awk was used to read the btwisted file line by line and to generate the three attributes (turn, twist, stackenergy) using following awk commands.

```
turn=`awk 'BEGIN {FS=":"; } FNR==3 {print $2} ' < $f".out"`
twist=`awk 'BEGIN {FS=":"; } FNR==4 {print $2} ' < $f".out"`
stackenergy=`awk 'BEGIN {FS=":"; } FNR==6 {print $2} ' < $f".out"`
```

Table 5-4: Grid enabled btwisted

```
Algorithm 5-3    Grid Enabled btwisted

Input : DNA sequences File

Generate one file per sequence in the file
    For all file in directory
            Run the btwisted program
        For all btwisted output files in the directory
            Read all output file and combine second field of
            third fourth and sixth line of the file
        EndFor
    EndFor

Output : Feature matrix  btwisted A
```

5.8.3 Wordcount

Wordcount counts the number of string pattern or words in each sequence. Words are made of base pairs of sequence of specific size. It moves along the sequence and count the words. It generates one file per sequence similar to banana and btwisted. In each file there are two columns one is the word name for example AACG and the other is the number of occurrences of that word. First the shell script program generates all possible patterns of the string of 4 characters A, C, G and T. It then used these files to generate a feature matrix file containing the frequency of each word in the sequence. GenWrapper was used here to port the application to the BOINC. Full source code of the program can be seen in Appendix B. The main steps are as follows.

Table 5-5: Grid enabled wordcount

```
Algorithm 5-4    Grid enabled wordcount

Input : DNA sequences File

Generate one file per sequence in the file
Generate All possible comibination of wordsize n
  generate index file
For all file in directory
    Run the wordcount program
EndFor
For all output files in the directory
    Read all output file and combine files to one final
    matrix file using the index file of all words
EndFor

Output : Feature matrix  wordcount A
```

5.8.4 Jaspscan

Jaspscan scans the sequence for the motifs in the JASPER transcription factor motif database. This database contains the collection of transcription factor binding sites and is collated from published papers and has an open data access policy (Jasper 2013). It generates one file for all sequences in the input FASTA file.

Jaspscan generates one file for all sequences, unlike other ported applications which generates one file per sequence. Here we only need to find the section that has the information for each sequence, to generate the matrix file. An example output is shown in the Figure 5-6. Here each section is marked with “#Sequence”. The string “#===” is used to distinguish between different sequences. A full list of transcription factor binding motifs was provided to the program. In this example the value of the feature MA0002.2 is 1 because there is one entry in the matrix and MA0003.1 value is 5 because it occurred 5 times.

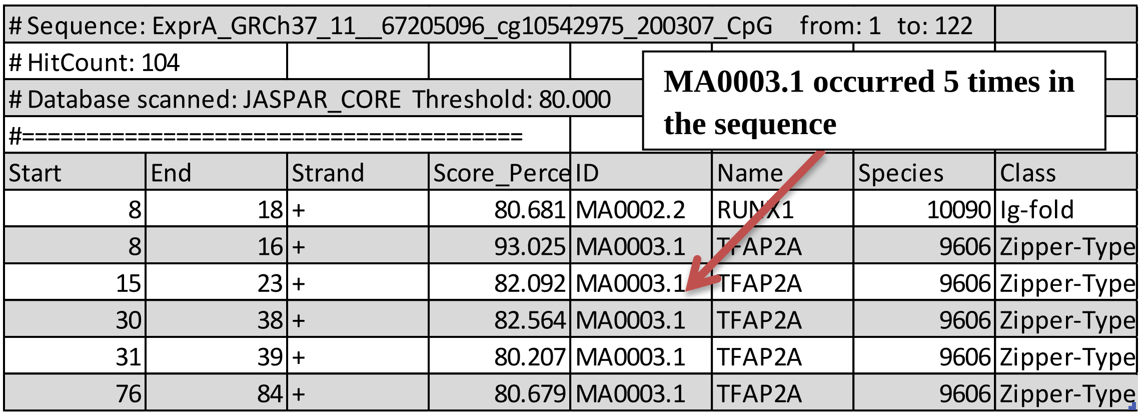
Table 5-6 : Grid Enable Jaspscan

```
Algorithm 5-5    Grid Enabled Jaspscan

Input : DNA sequences File

23.  Run Jaspscan program on DNA sequence File
24.  Initialise variable i for index of sequence j for index of feature
25.    While not end of  Jaspscan output file
26.        If Line contains "#Sequence"
27.            Find the index of sequence store in I
28.            Skip 4 lines
29.            Split Line by space and store the results in Array S
30.            Find index of transcription factor binding site by S[3] and store in j
31.          A[i][j]= A[i][j]+1
32.    EndWhile

Output : Feature matrix  A
```



# Sequence: ExprA_GRCh37_11__67205096_cg10542975_200307_CpG from: 1 to: 122
# HitCount: 104
# Database scanned: JASPAR_CORE Threshold: 80.000
#======================================

Start	End	Strand	Score_Perce	ID	Name	Species	Class
8	18	+	80.681	MA0002.2	RUNX1	10090	Ig-fold
8	16	+	93.025	MA0003.1	TFAP2A	9606	Zipper-Type
15	23	+	82.092	MA0003.1	TFAP2A	9606	Zipper-Type
30	38	+	82.564	MA0003.1	TFAP2A	9606	Zipper-Type
31	39	+	80.207	MA0003.1	TFAP2A	9606	Zipper-Type
76	84	+	80.679	MA0003.1	TFAP2A	9606	Zipper-Type

MA0003.1 occurred 5 times in the sequence

Figure 5-6 part of jaspscan result file.

Full code can be accessed in Appendix B. Table 5-7 is a list of ported applications and number of features they create.

Table 5-7 Summary of ported applications

	File generation	Number of features
banana	1 per sequence	2
btwisted	1 per sequence	3
wordcount	1 per sequence	256
jaspscan	1 for all sequences	467

5.9 Creating Feature Generation workflow

Applications that mentioned in the previous sections can be used individually to generate features and then merged together. The ultimate goal of the system was to create modular system for feature generation.

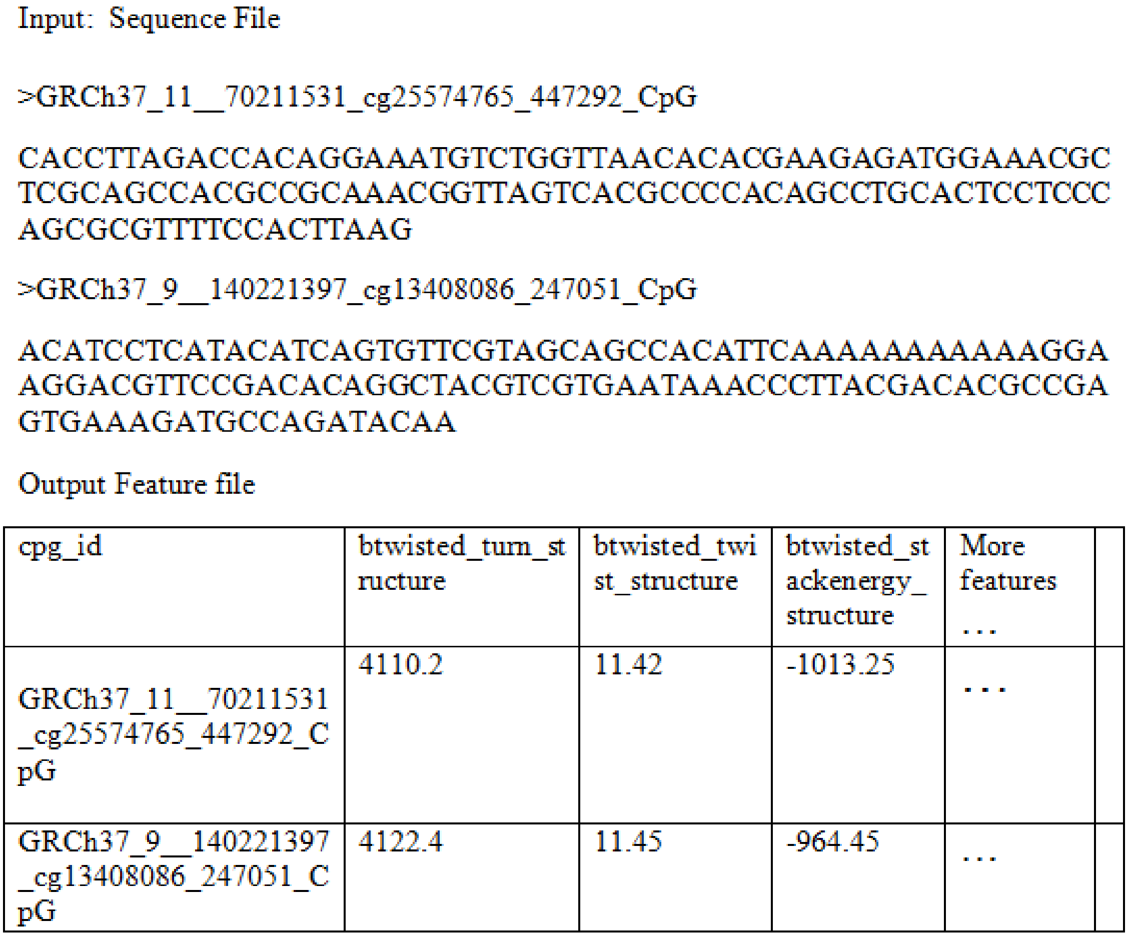
Each of this application in the ws-pgrade portal has three main programs

1. Generator: This program generates tasks or workunits which will eventually run on the worker machines. Some examples of these tasks are 1) splitting big files into smaller files 2) Processing large number of files. 3) Generating parameter combinations.
2. Worker: The workunit generated by generator should be submitted to the worker machines .The worker program processes the workunits and sends the results

back. For example finding the maximum of a function given some parameters.

3. Collector: The collector aggregates results returned from the worker machine. For example the different parts of a file processed by a worker should merge to give the final result, or in another example finding the maximum of all returned results from evaluated parameters on the worker machines.

This system should provide facility to add new feature generating part. gUSE system used for this goal. This system can be expanded based on the user demand. This section starts with the simplest form of feature generation with one application and continues with adding all parts to the system. The input to the workflow is the file contains DNA sequences with their ID and the output is the feature matrix contains all investigated features see Figure 5-7.



Input: Sequence File

>GRCh37_11__70211531_cg25574765_447292_CpG

CACCTTAGACCACAGGAAATGTCTGGTTAACACACGAAGAGATGGAAACGC
TCGCAGCCACGCCGCAAACGGTTAGTCACGCCCCACAGCCTGCACTCCTCCC
AGCGCGTTTTCCACTTAAG

>GRCh37_9__140221397_cg13408086_247051_CpG

ACATCCTCATACATCAGTGTTCGTAGCAGCCACATTCAAAAAAAAAAAGGA
AGGACGTTCCGACACAGGCTACGTCGTGAATAAACCCTTACGACACGCCGA
GTGAAAGATGCCAGATACAA

Output Feature file

cpg_id	btwisted_turn_structure	btwisted_twist_structure	btwisted_stackenergy_structure	More features ...	
GRCh37_11__70211531_cg25574765_447292_CpG	4110.2	11.42	-1013.25	...	
GRCh37_9__140221397_cg13408086_247051_CpG	4122.4	11.45	-964.45	...	

Figure 5-7. Input sequence file and Output feature file.

5.9.1 One application feature generation

The banana application which was the first program to be used for feature generation in the workflow was grid-enable as described in previous section. A workflow graph was created in the workflow editor. The generator split the sequence file, and then it generates a defined number of sequences per file. The number of sequences per file was determined in “splitnumberfile”. Banana was executed for each file in worker nodes and generated “tabdelimited” files similar to the matrix mentioned in the previous section see Figure 5-7. The collector pasted each new file, one after another, except the header of the file, and creates new file. Because the program created the same headers for each file in the same order, this does not create any inconsistency between the values. Figure 5-8 shows the graph of the workflow.

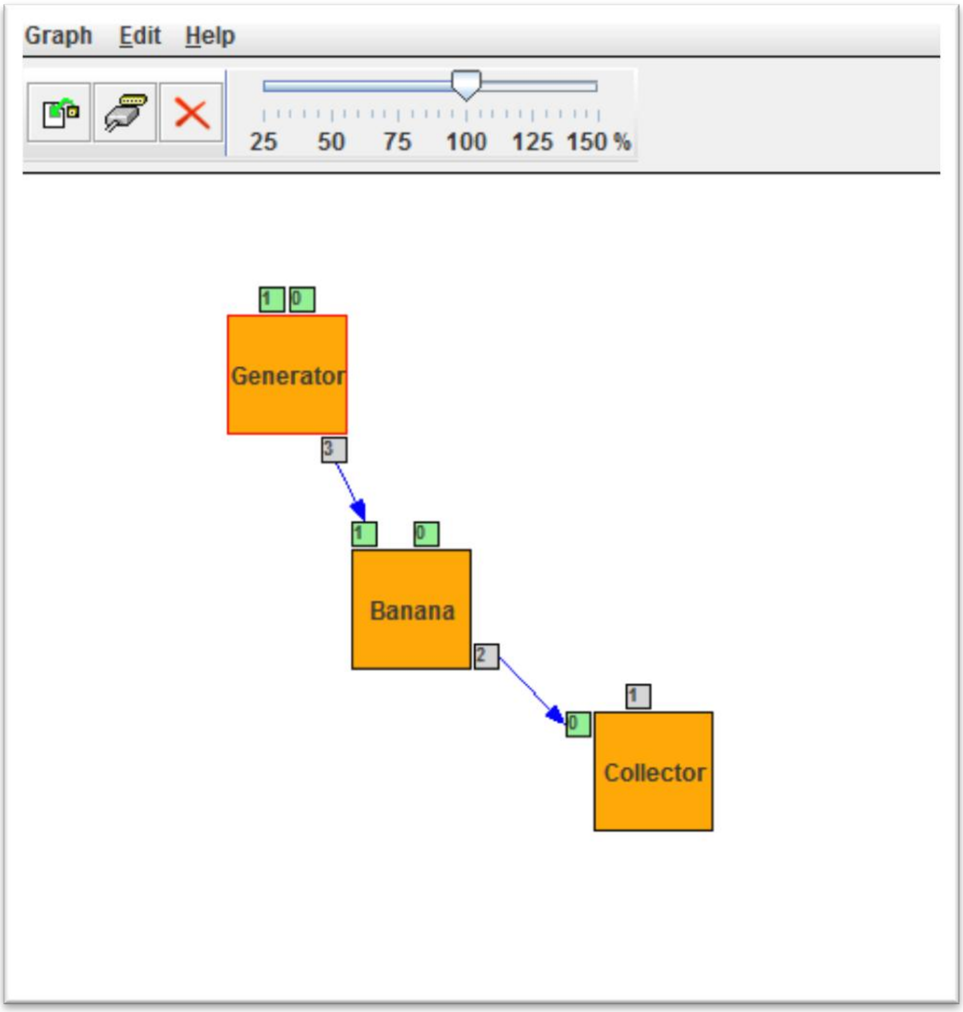
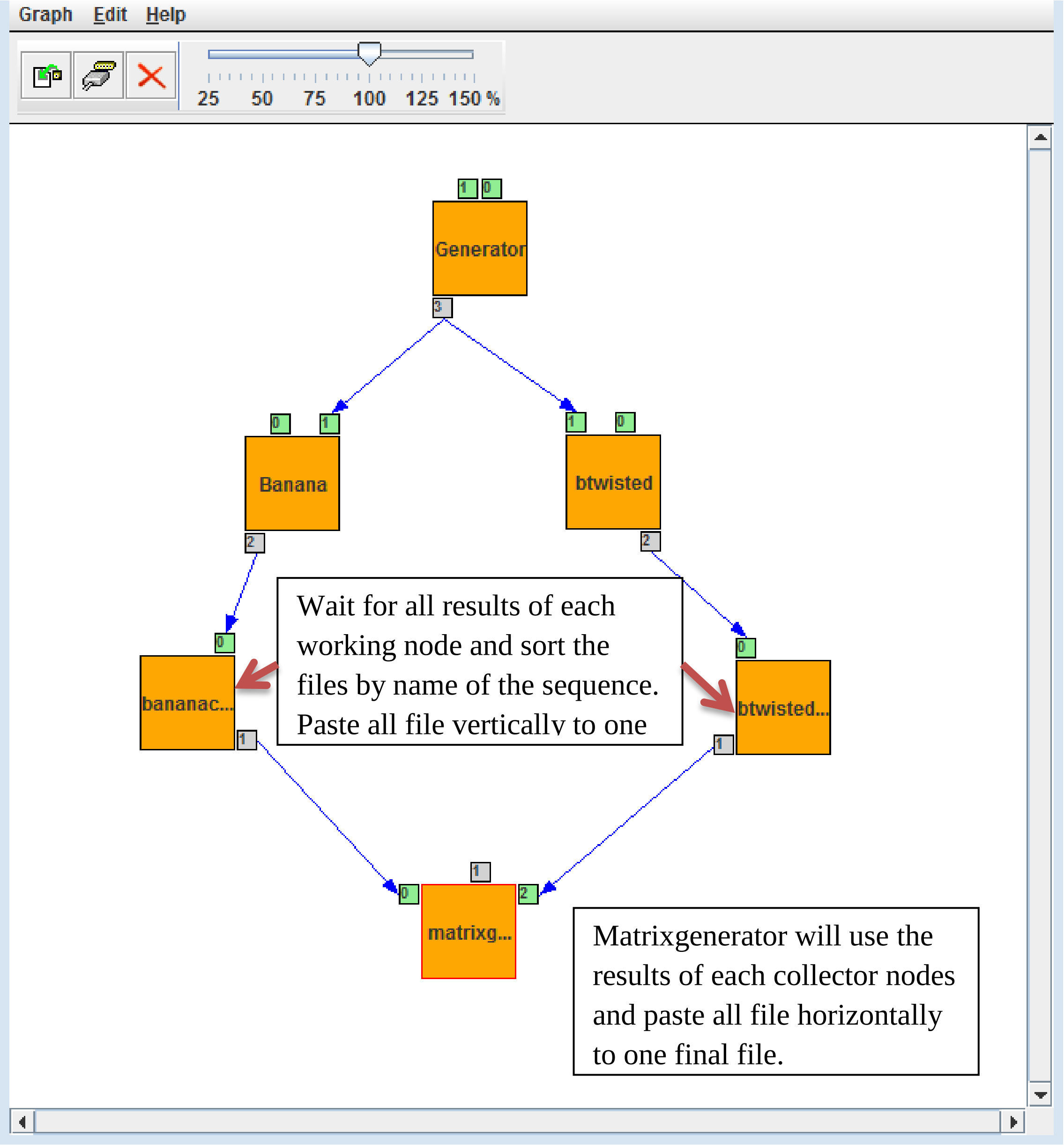


Figure 5-8. Workflow of one node banana contains generator application and collector.

5.9.2 Multiple Application feature generation

Adding btwisted to this workflow needed the additional step of combining the results of the banana collector with the results of the btwisted collector. The feature matrix headers were in order, so each application had its own collector. The source code for the collectors is similar, but they were used in separate nodes. This made it possible to download the results for each individual EMBOSSES applications. Because final results should be in order, they should be sorted by name in the collector node. Figure 5-9 shows the workflow after btwisted was added.



Wait for all results of each working node and sort the files by name of the sequence. Paste all file vertically to one

Matrixgenerator will use the results of each collector nodes and paste all file horizontally to one final file.

Figure 5-9. workflow for two ported application btwisted and banana.

At this stage any other application could be added to the workflow using the same guidelines, without making change to other parts of the workflow. Similarly, parts that are not required can be removed from workflow. The generator node and matrixgenerator node will be the same for all nodes. The final workflow designed in this work is shown in the Figure 5-10.

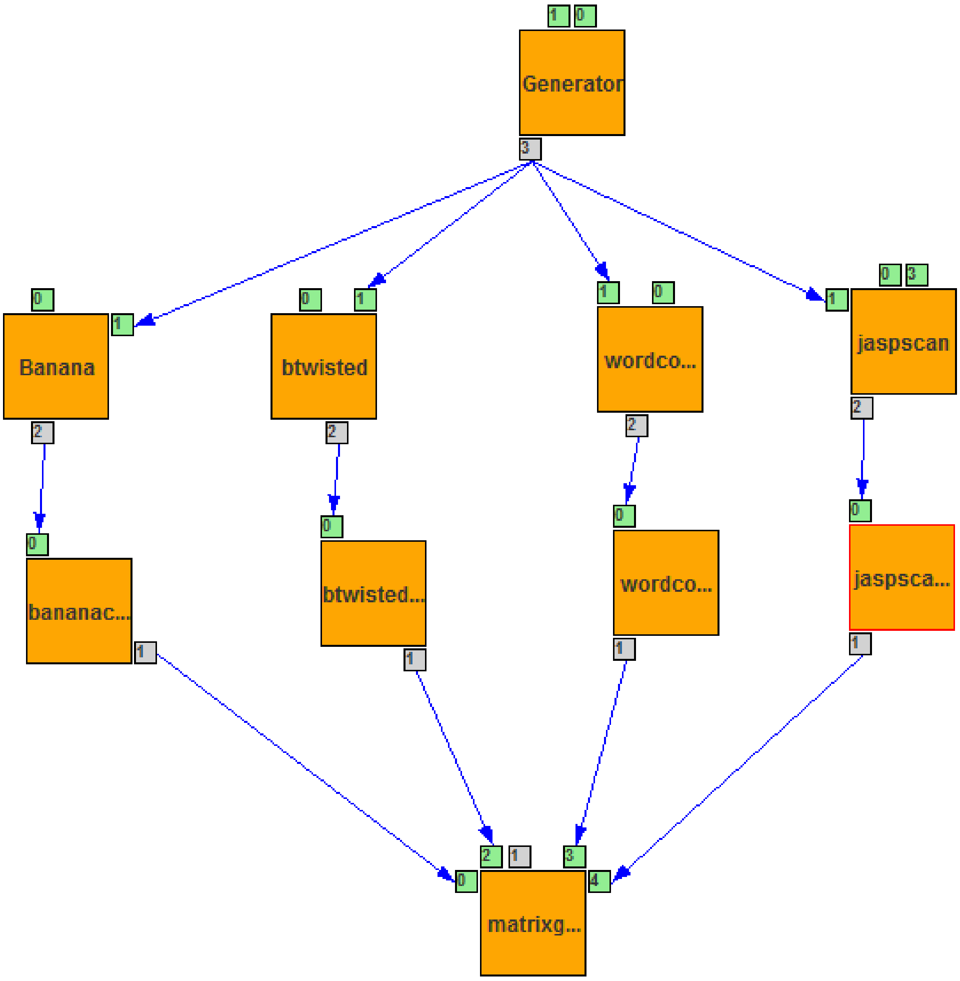


Figure 5-10. the completed workflow of the feature matrix generator needs two input files one determining the number of sequences per file and the other is the fasta file of all sequences. Individual application results could be downloaded as in the workflow they were configured as permanent .The final results were generated in the matrix generator node.

The following code shows how each EMBOSS application collector works. The collector application is a shell script which runs in the server similar to the generator.

Extract Headers

```
awk '{
    if (NR == 1)
       print $0 >> "tabresult_header_1"
    else
       if($1 != "cpg_id")
         print $0 >> "temptabresult_1"

    }' tabdelimited_*
```

Sort by name of sequence

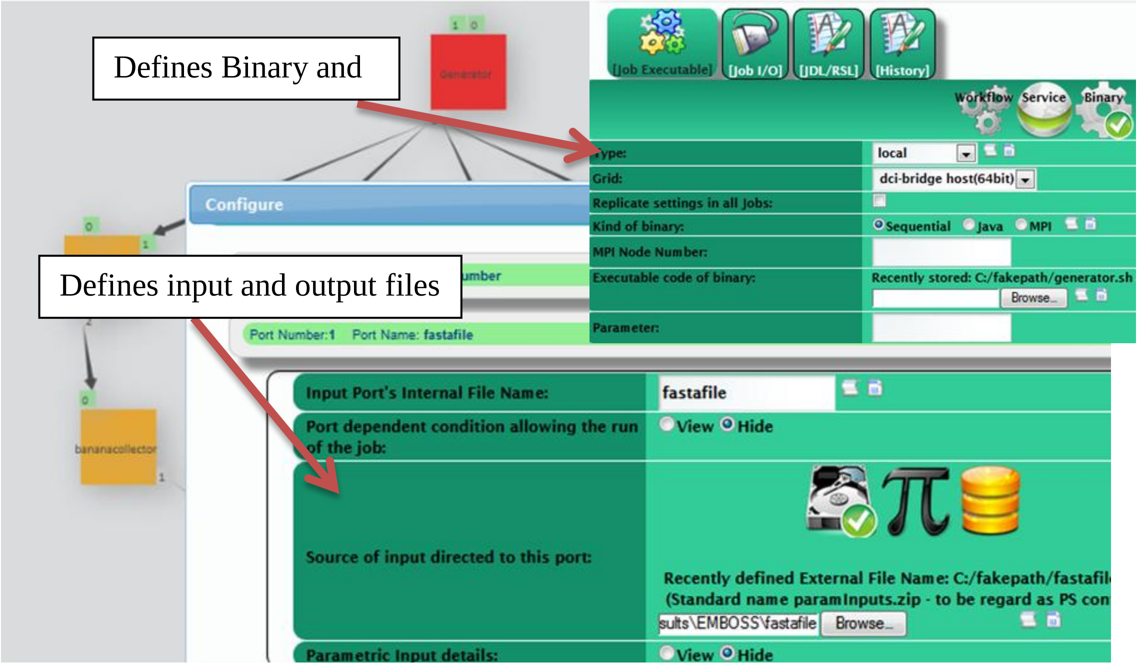
```
sort -s -k1  temptabresult_1 >
tempsortedtabresult_1 ;
awk '{print $0 > "allresult"}' tabresult_header_1;
awk '{print $0 >> "allresult"}'
tempsortedtabresult_1;
```

MatrixGenerator simply pasted all the files together. It used the dot product capability of the gUSE system to collect all the files.

```
paste allresult_* > matrixfeaturefile
```

5.10 Workflow submission

In order to configure and execute workflows in the gUse, the graph of the workflows should be use to create “concrete”. This task is done by creating a “concrete” workflow in the portal page under the workflow tab. Concrete was configured and the name of input and output files were defined. Also the resource and binaries that the node was going to run should be defined in this section. The output port of the Generator was defined as generator. The input port of the collector was defined as collector so it waited until it received all the results from the application node. The binary for the BOINC was set according to the resources defined in the information service tab of the system. The binary for the Generator and collector node were selected as local.



Defines Binary and

Defines input and output files

Type: local

Grid: dci-bridge host(64bit)

Replicate settings in all Jobs: □

Kind of binary: ● Sequential ○ Java ○ MPI

MPI Node Number:

Executable code of binary: Recently stored: C:/fakepath/generator.sh

Parameter:

Port Number:1 Port Name: fastafile

Input Port's Internal File Name: fastafile

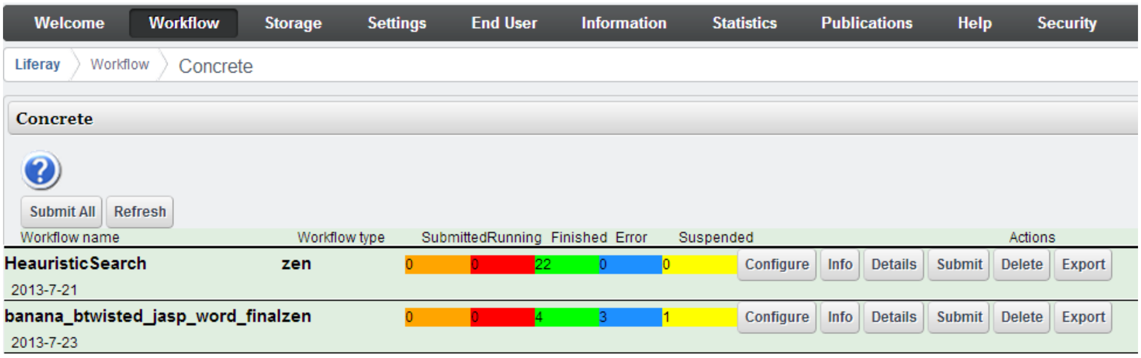
Port dependent condition allowing the run of the job: ○ View ● Hide

Source of input directed to this port: Recently defined External File Name: C:/fakepath/fastafil (Standard name paramInputs.zip - to be regard as PS con sults\EMBOSS\fastafile

Parametric Input details: ○ View ● Hide

Figure 5-11 an example of Workflow configuration page

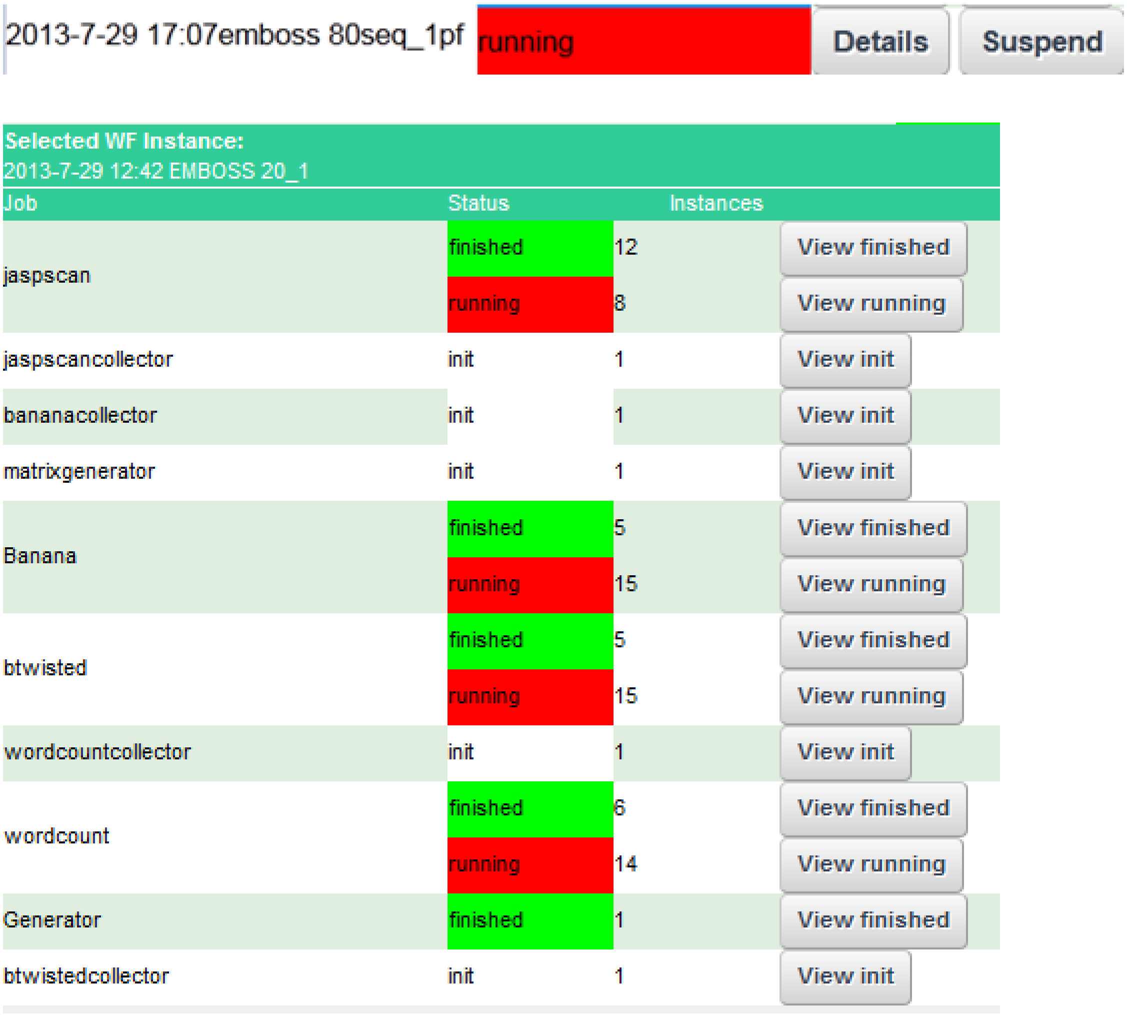
Workflows were submitted via the portal liferay/workflow/concrete tab. By choosing the concrete tab a list of generated workflows could be seen. In this page users can overview the overall status of the workflows and take appropriate actions. For each new instance of the workflow new files could be uploaded to the system. This task can be done by choosing the configure button in the actions section. The two input files (number of sequence per file and FASTA file) described in the previous sections should be uploaded in this page.



Workflow name	Workflow type	Submitted	Running	Finished	Error	Suspended	Actions
HeauristicSearch 2013-7-21	zen	0	0	22	0	0	Configure Info Details Submit Delete Export
banana_btwisted_jasp_word_final 2013-7-23	zen	0	0	4	3	1	Configure Info Details Submit Delete Export

Figure 5-12 an example of a job Submission page, the “configure” button should be used to add input and output files and to define binaries for each workflow.

The details of each workflow node progress could be examined in this page. Whenever workflow nodes finished their job, the results could be downloaded by clicking view all the “content” buttons.



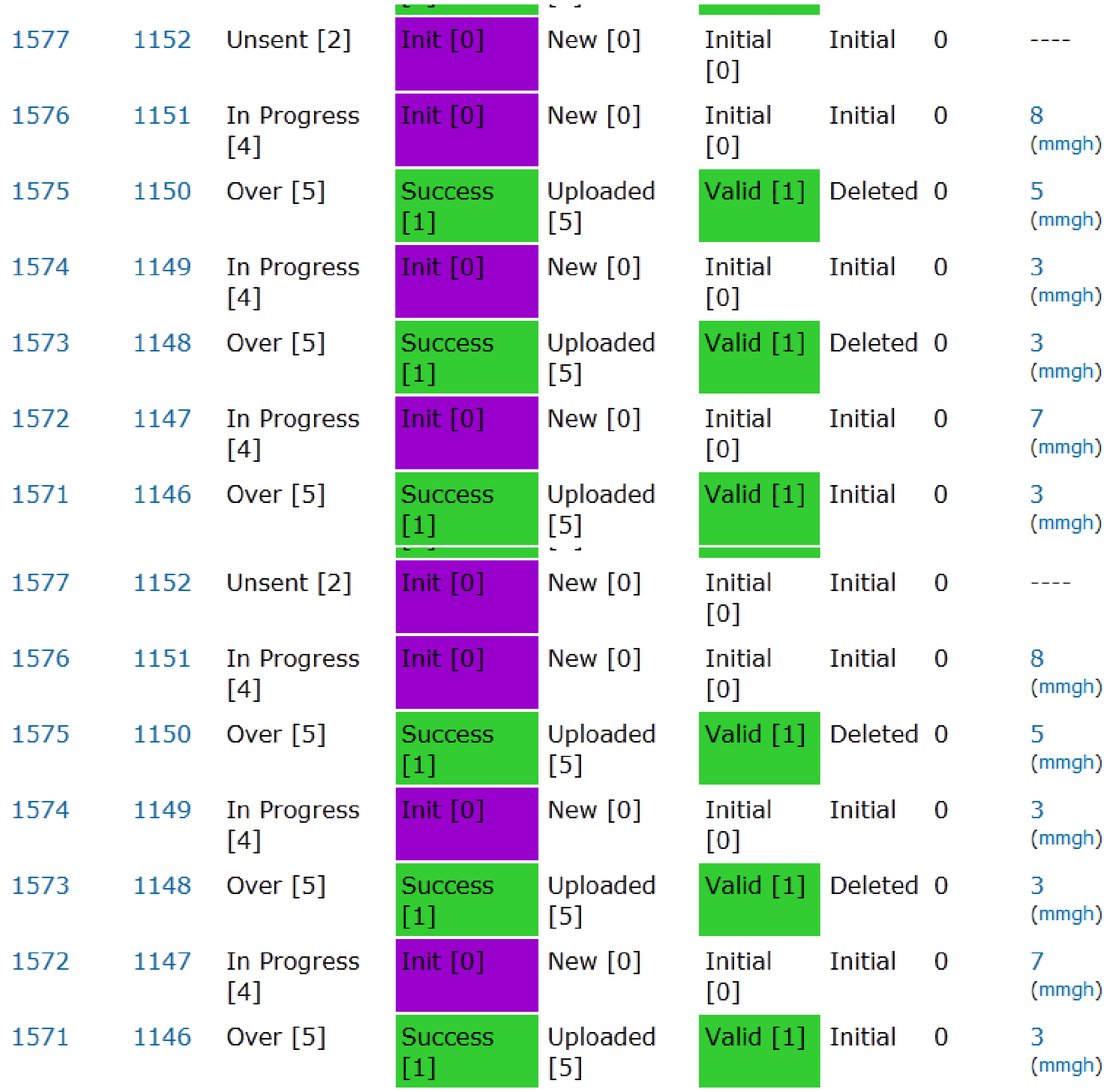
2013-7-29 17:07emboss 80seq_1pf running Details Suspend

Selected WF Instance:
2013-7-29 12:42 EMBOSS 20_1

Job	Status	Instances	
jaspscan	finished	12	View finished
	running	8	View running
jaspscancollector	init	1	View init
bananacollector	init	1	View init
matrixgenerator	init	1	View init
Banana	finished	5	View finished
	running	15	View running
btwisted	finished	5	View finished
	running	15	View running
wordcountcollector	init	1	View init
wordcount	finished	6	View finished
	running	14	View running
Generator	finished	1	View finished
btwistedcollector	init	1	View init

Figure 5-13. Details of the status of each job after workflow submission could be monitored in ws-pgrade portal. This figure shows the workflow submission for an input file which contained 20 sequences. It then indicated that each sequence should be submitted individually so eventually 20 jobs were submitted.

Another way to check the status of jobs was by querying the BOINC database. The details of the jobs running in the BOINC workers and whether or not they were sent to workers can be checked by querying the BOINC database. This can be done by examining the admin webpage of the BOINC project. Figure 5-14 shows that some jobs were not sent to the worker yet. Whenever the valid results were sent back to the BOINC server 3Gbridge daemon uploaded them to the upload folder and cancel the workunits.



1577	1152	Unsent [2]	Init [0]	New [0]	Initial [0]	Initial	0	----
1576	1151	In Progress [4]	Init [0]	New [0]	Initial [0]	Initial	0	8 (mmgh)
1575	1150	Over [5]	Success [1]	Uploaded [5]	Valid [1]	Deleted	0	5 (mmgh)
1574	1149	In Progress [4]	Init [0]	New [0]	Initial [0]	Initial	0	3 (mmgh)
1573	1148	Over [5]	Success [1]	Uploaded [5]	Valid [1]	Deleted	0	3 (mmgh)
1572	1147	In Progress [4]	Init [0]	New [0]	Initial [0]	Initial	0	7 (mmgh)
1571	1146	Over [5]	Success [1]	Uploaded [5]	Valid [1]	Initial	0	3 (mmgh)

1577	1152	Unsent [2]	Init [0]	New [0]	Initial [0]	Initial	0	----
1576	1151	In Progress [4]	Init [0]	New [0]	Initial [0]	Initial	0	8 (mmgh)
1575	1150	Over [5]	Success [1]	Uploaded [5]	Valid [1]	Deleted	0	5 (mmgh)
1574	1149	In Progress [4]	Init [0]	New [0]	Initial [0]	Initial	0	3 (mmgh)
1573	1148	Over [5]	Success [1]	Uploaded [5]	Valid [1]	Deleted	0	3 (mmgh)
1572	1147	In Progress [4]	Init [0]	New [0]	Initial [0]	Initial	0	7 (mmgh)
1571	1146	Over [5]	Success [1]	Uploaded [5]	Valid [1]	Initial	0	3 (mmgh)

Figure 5-14. An example of BOINC results page.

5.11 Grid enabled hill climbing search

The hill climbing algorithm as described in section 2.12.2 is a method for searching a large search space which cannot be searched exhaustively. In the problem of feature subset selection, the solution is defined by binary string. The length of the string shows total number of features. Selected features are represented by one. Non-selected features are represented by zero. If there is 1 at a position 2 it means that feature number two is selected. Table 5-8 shows an example of full table. For a solution like 0101, the resulting new matrix is shown in the Table 5-9.

.

Table 5-8 Example of full set feature matrix.

Cpg_id	Feature_0	Feature_1	Feature_2	Feature_3
Cg0001	0.6	0.4	0.1	0.3
Cg0002	0.2	0.6	0.1	0.5
Cg0003	0.7	0.8	0.5	0.3

Table 5-9 Example of feature subset matrix

Cpg_id	Feature_1	Feature_3
Cg0001	0.4	0.3
Cg0002	0.6	0.5
Cg0003	0.8	0.3

In order to search the feature space, small random changes were applied to the binary string and a new solution was generated. This solution was evaluated against the fitness function, and if the new fitness was better, it replaced the best solution. This process was repeated for fixed number of times and the final results would be the selected feature subset.

Because choosing one starting point can trap the hill climbing algorithm at local optima, the starting point for the algorithm can be any random number, as different starting points might lead to better solutions. These starting points were run in parallel and the results were compared. Finally the maximum value of the results was selected. Due to the nature of this problem it is good candidate to grid enable. This program was developed as a java program, and similar to the previous section, GenWrapper was used to call the java on the worker machine and also to zip the results in the case of batching more than one binary in one job. The inputs were two files; one file contained the string representations of the starting point solutions. The second file contained the parameters given to the program. These parameters defined the number of iterations, classification method, measurement index and the name of the file that contained starting point’s information.

5.12 Hill climbing search master application

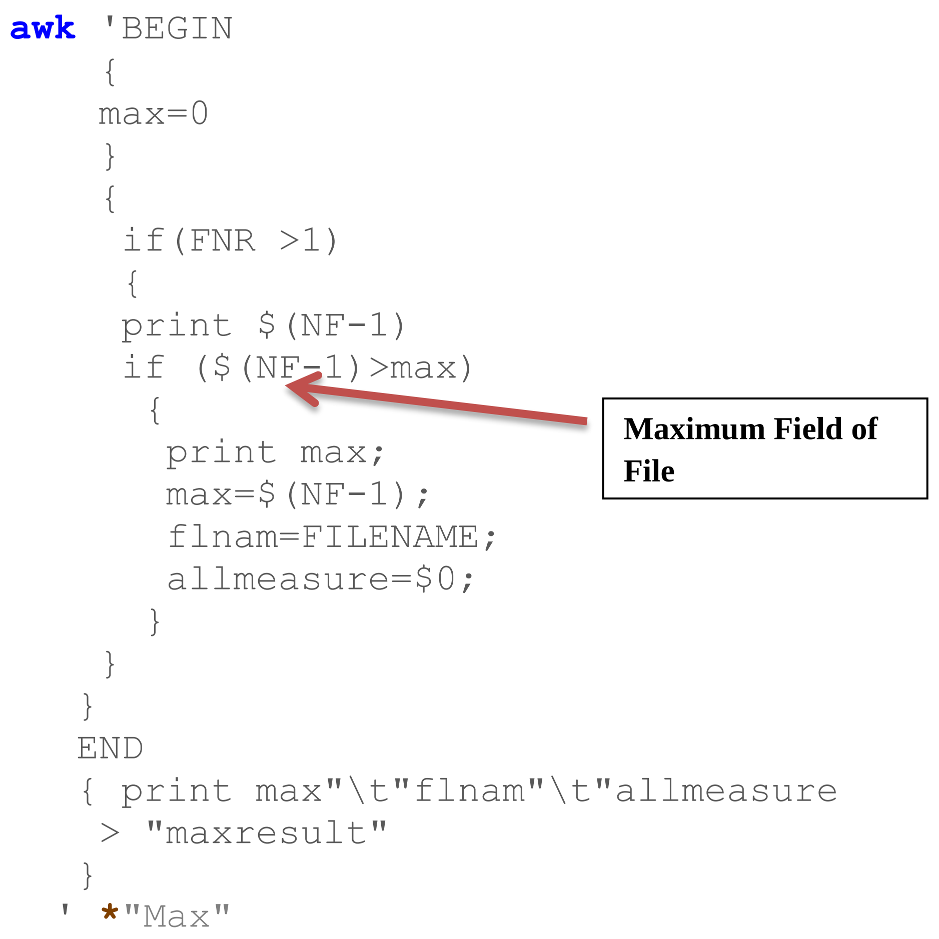
The master application for the hill climbing search was a shell script file which ran in the server. It took as an input “splitnumberfile” and “rankedlisttodistribute” files and split the ranked list according to the number of splits specified in the “splitnumberfile”. “splitnumberfile” provided the capability of batching two or more searches in a single job to overcome communication overhead in the cases where the computation time is fast. The awk program was used to read the “rankedlisttodistribute” and generated new files for each line. The generated files are named as partHS_<index> the index started from 0. These naming rules were applied because files should follow the rules expected by the gUSe workflow system for generator ports. The following code is the master program for the hill climbing search.

```
splitnumber=`awk '{if ($1="split") print $2}'
splitnumberfile`;
#number of binary per file defined here
echo $splitnumber
#split -L rankedlisttodistribute part
awk  -v spl=$splitnumber
 'BEGIN{
       i=-1
       }
 {
  if(spl!=1)
      {  #special case for 1
        if ((NR%spl)==1)
         {
         i=i+1;
         }
        x="partHS_"i;
        print $0 > x
      }
      else
      {
        print $0 > "partHS_"(NR-1)
      }

}' rankedlisttodistribute
```

5.13 Hill climbing Search collector application

The collector application was shell script that runs in the server. It extracted the received files. Each received file contained two files. One is the <FILENAME>Max, each line of the file contained information about when the maximum value is updated in the hill climbing search. Each line in the file <FILENAME> contained the measurement calculated and the binary representation of the solution in each iteration. Then the script searched all the "<FILENAME>Max" files to find the maximum in those files. It copied this file to a folder that contained all files. It then tar zipped all the files together as the final result. The portion of collector code that finds maximum is shown in the following code. It used the awk program to search through each line and the maximum field of the file.



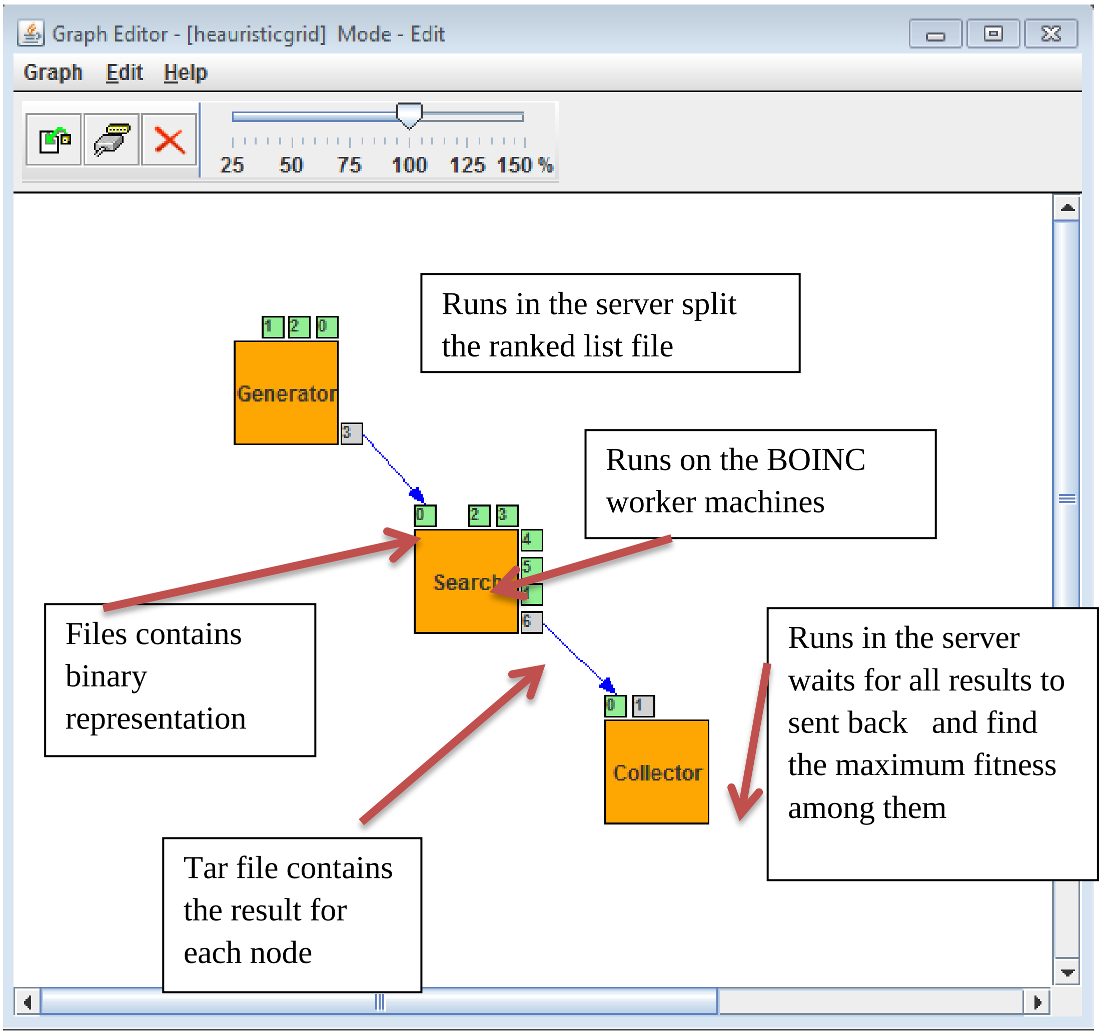
```
awk 'BEGIN
     {
     max=0
     }
     {
      if(FNR >1)
      {
      print $(NF-1)
      if ($(NF-1)>max)
        {
         print max;
         max=$(NF-1);
         flnam=FILENAME;
         allmeasure=$0;
        }
     }
    }
    END
    { print max"\t"flnam"\t"allmeasure
     > "maxresult"
    }
  ' *"Max"
```

Maximum Field of File

5.14 Hill climbing Search Workflow Generation

In this research the gUse portal workflow system was applied to submit jobs to BOINC. First the workflow topology was defined in the graph editor of the workflow. Each port in our example is the representation of the file and each node are the applications running in the system. The arrows between the ports represent file transitions. Green ports are input ports and grey ports represent output ports.

Generator and Collector applications were running in the server as mentioned in the previous section. In another setting it is possible to use another machine other than server to run these programs as long as they are supported by the gUse resources for node.



Runs in the server split the ranked list file

Runs on the BOINC worker machines

Files contains binary representation

Runs in the server waits for all results to sent back and find the maximum fitness among them

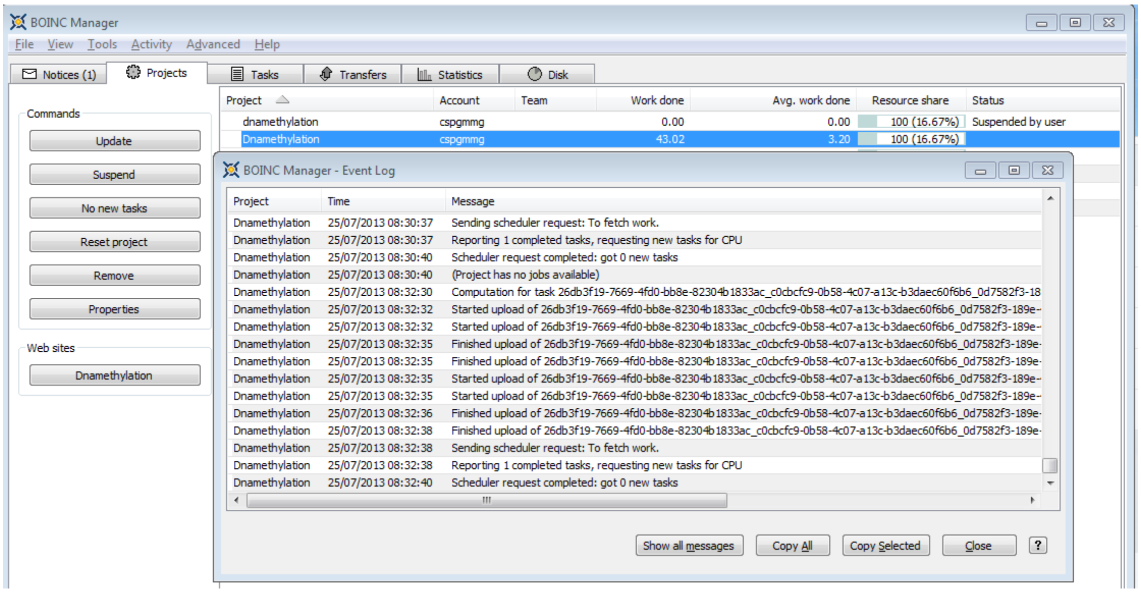
Tar file contains the result for each node

Figure 5-15. Hill climbing search workflow graph in the graph editor shows the search node is the worker node, and generator and collector nodes are master nodes

5.15 Hill climbing Search Job Submission

The workflow configuration was similar to the EMBOSS application. This was done for the search node with the parameter file and for the Generator node with the "rankedlisttodistribute" file. After submitting jobs by clicking the "details" button, this should indicate that it is running.

The log file of BOINC manager in the worker machine can indicate the progress of the works in the client side. The following snapshot shows the connection of the client to the BOINC project and one completed task.



Project	Account	Team	Work done	Avg. work done	Resource share	Status
dnamethylation	cspgmmg		0.00	0.00	100 (16.67%)	Suspended by user
Dnamethylation	cspgmmg		43.02	3.20	100 (16.67%)	

```
Project          Time                   Message
Dnamethylation   25/07/2013 08:30:37    Sending scheduler request: To fetch work.
Dnamethylation   25/07/2013 08:30:37    Reporting 1 completed tasks, requesting new tasks for CPU
Dnamethylation   25/07/2013 08:30:40    Scheduler request completed: got 0 new tasks
Dnamethylation   25/07/2013 08:30:40    (Project has no jobs available)
Dnamethylation   25/07/2013 08:32:30    Computation for task 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-18
Dnamethylation   25/07/2013 08:32:32    Started upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:32    Started upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:35    Finished upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:35    Finished upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:35    Started upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:35    Started upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:36    Finished upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:38    Finished upload of 26db3f19-7669-4fd0-bb8e-82304b1833ac_c0cbcfc9-0b58-4c07-a13c-b3daec60f6b6_0d7582f3-189e-
Dnamethylation   25/07/2013 08:32:38    Sending scheduler request: To fetch work.
Dnamethylation   25/07/2013 08:32:38    Reporting 1 completed tasks, requesting new tasks for CPU
Dnamethylation   25/07/2013 08:32:40    Scheduler request completed: got 0 new tasks
```

Figure 5-16. BOINC client log snapshot shows details of tasks running on clients machines.

5.16 Performance test feature generating matrix

Performance testing for the hill climbing search was carried out by increasing the number of the sequences in the "fasta file". Analysis of the initial test showed that because of the short running time of each sequence, there might be a communication overhead. For example 100 sequences and one sequence per file will cause 400 work units submissions for all algorithms. Increasing the number of sequences per file made the performance better. Figure 5-17 shows the speed-up improvement by the batching of more than one sequence per file. The batching help to reduce large number of jobs submission and also reduce communication overhead for example if the sending and receiving times are more than running single jobs it is desirable to put those two jobs together in this way the running time of two jobs will be longer than sending and receiving of them so we can gain speedup.

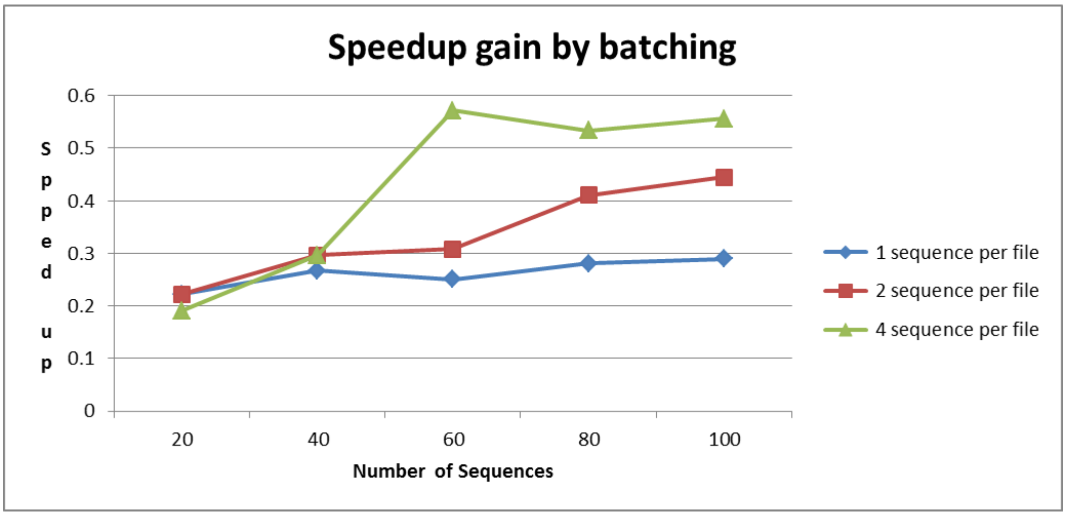


Figure 5-17 speedup improvement by batching of more than 1 sequence per file

The initial results showed no speed-up (<1) but it was improved by batching more than one sequence together, and increasing the number of sequences.

Table 5-10 Speedup results for different number of sequences and different number sequence per files

number of sequences	1 sequence per file	2 sequence per file	3 sequence per file
20	0.22	0.22	0.20
40	0.27	0.30	0.30
60	0.25	0.31	0.58
80	0.29	0.41	0.53
100	0.29	0.44	0.56

So the last experiment was done on the entire dataset derived from chapter 4 which contains 653 sequences. In this case the results shows speedup of up to 5 times more than sequential time which can be seen in the Figure 5-18. Whenever there are large number of sequences, in the case whole genome for example in the platform that was studied here the running time of feature generation increase dramatically. In order to estimate the running time of the Illumina platform which has nearly half a million site we can estimate the running time. Each job when all feature generation programs run takes nearly 68 seconds (wordcount+ banana+ btwisted+ jaspscan) on Intel Core(TM) 2 Duo CPU E7400 2.8 GHz. In large scale setting it for half million sequences and 2 million work units (without batching) it will take nearly one year (393 days) to run on one machine. Depending on the available machines we can get the result much faster for example for 1000 machines (In the volunteer computing settings sometimes there are 100 thousands machines Figure 2-9) even with 20 times speedup we can get the jobs done in less than a month. The overall running time for the best experiment for 653 sequence was nearly 148 minute.

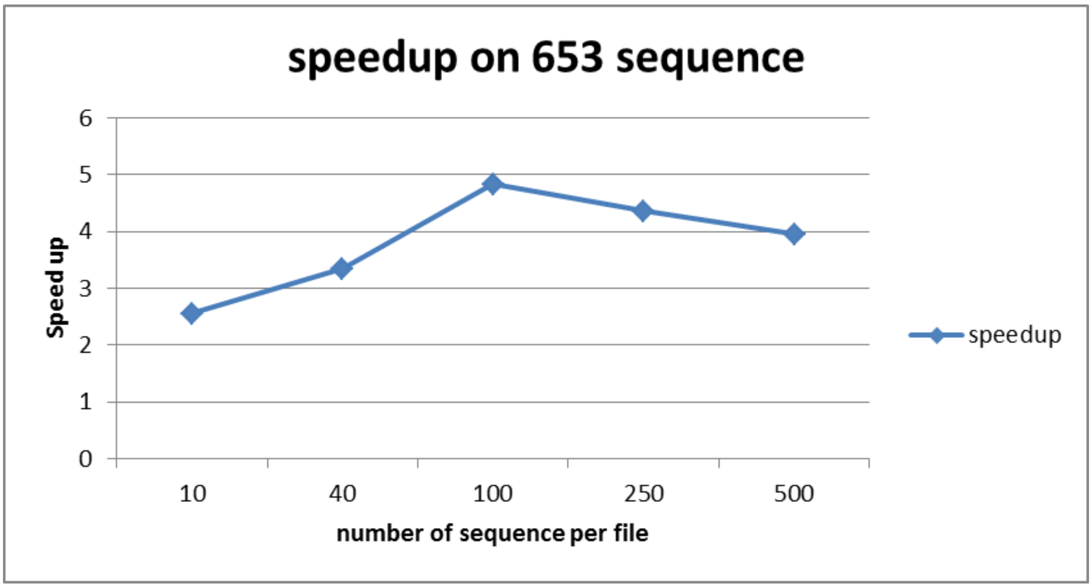


Figure 5-18 The speed-up result for 653 sequence in the dataset for different numbers of sequences per file.

5.16.1 Performance test hill climbing search

Initial tests on the hill climbing search were done with gradually increasing iterations for the same number of jobs, measuring the performance. In order to compare the run time with the sequential time, average computation time of one iteration of a model for 100 iterations calculated. The initial performance test was done using the naïve bayes classifier. The Table 5-11 shows the result for 20 sequences with an increasing number of jobs.

Table 5-11 speedup results for different number of iterations of hill climbing for 20 to 100 iterations

number of jobs	number of classifier calls	speedup
20	400	1.78
40	800	2.33
60	1200	2.67
80	1600	2.82
100	2000	2.44

This data shows that the speedup got better as jobs became more computationally intensive. The initial performance test gave nothing better than 2.8 times faster than sequential times, in the best case, with further investigation of the submitted jobs, it was realised that all jobs were assigned to a few available hosts, and some hosts didn't get any jobs. In order to improve this situation, the number of jobs per host was limited. This can be done in BOINC by setting the parameter "wu_in_progress" in the BOINC configuration file. Applying this restriction to one job in progress for each host made the performance better. The results can be seen in the Figure 5-19 and Table 5-12.

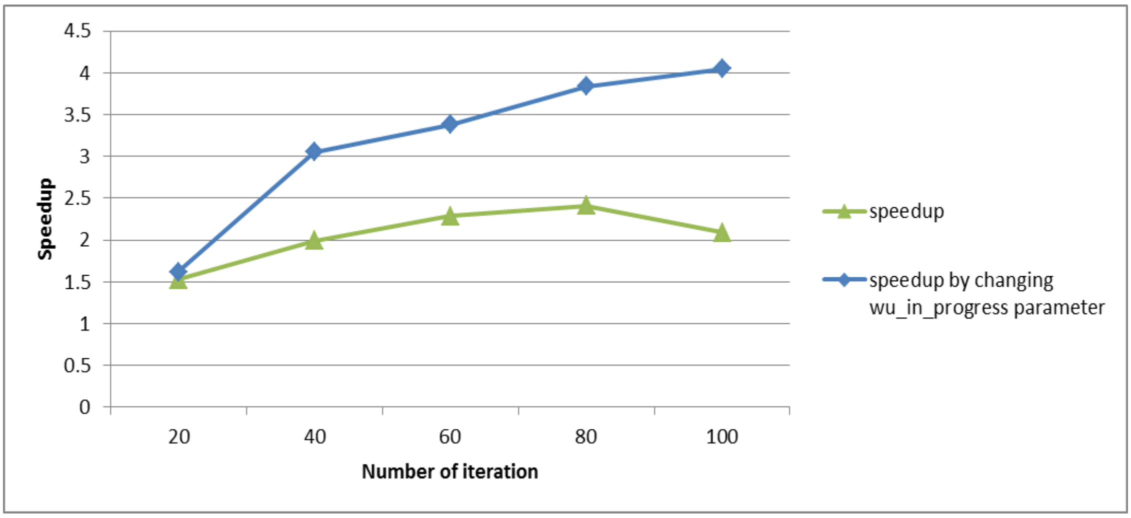


Figure 5-19 The graph shows speedup results for two different kinds of configuration. Improvement could be seen as the number of "wu_in_progreess" was restricted.

Table 5-12 Shows the comparison of speed-up for different numbers of jobs and different configurations.

number of jobs	speedup	speedup by changing wu_in_progress parameter
20	1.52	1.62
40	1.99	3.05
60	2.29	3.38
80	2.41	3.84
100	2.09	4.05

5.17 Performance testing of each classification method

After initial testing, the system was used for the hill climbing search on each classification method. The following results showed that this system achieved a maximum of 5.37 speedup out of a possible 7, since we had 7 machines in our test environment. The difference between the speedup results should be seen as communication overhead and run times of collector and generator nodes. Results are shown in Figure 5-20 and Table 5-13.

Table 5-13 details of speedup as programs progress

percent of work completed	J48	SVM	naïve bayes
10	0.97	0.99	0.97
20	1.84	1.94	1.94
30	2.74	2.80	2.83
40	3.59	3.72	3.68
50	4.40	4.62	4.57
60	5.14	5.30	5.38
70	5.93	5.78	5.38
80	4.07	3.90	3.90
90	4.14	4.38	4.37
100	4.44	4.75	4.84

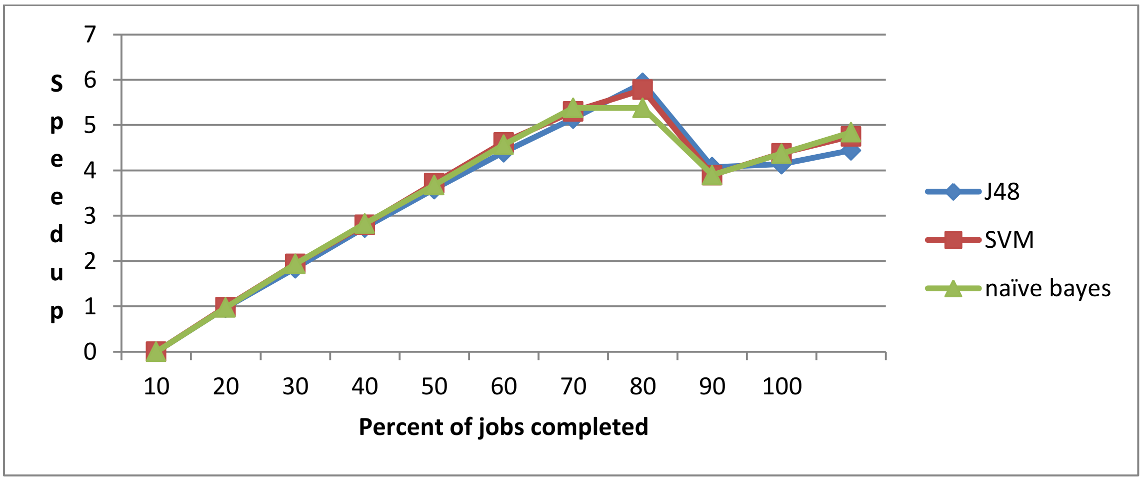


Figure 5-20 shows speedup results for different classification methods in feature subset selection.

5.18 Conclusion

This chapter provides details of system preparation and test of the system for two main tasks of feature generation and feature subset search on the grid. Thus the first objective which was to create the scalable system for feature generation has been completed by creating workflow .In the case of feature generation, workflow can provide modular feature generation. In this system adding new feature generating nodes did not change the whole system. Small number of the jobs in feature generating workflow connected to the BOINC does not provide any speed-up; Speed-up was increased by the growth in the number of sequences. In order to improve the speedup and reduce communication time, numbers of sequences per work units were increased. Additionally results showed that the hill climbing search is more computational intensive and benefits more from connecting to the BOINC platform in the test environment. The second objective which was to accelerate the execution time of feature generation and feature subset selection completed by connecting the system to the BOINC and improvement to the BOINC system.

Chapter 6 - Features Classifying CpG sites

6 Background

Machine learning algorithms were used to predict methylation of CpG Island and CpG sites. The Table 6-1 shows the results of similar works in this context more description is provided in the background section of chapter 4. This section discusses the result of applying machine learning and feature subset selection algorithms.

Table 6-1 Reported result of selected papers on Epigenetic analysis

Source	Reported Results	Method
(Lu *et al.* , 2010)	Accuracy 75%	Nearest neighbor
(Bock *et al.* , 2007)	Accuracy 79%	Support vector machine
(Fan et al., 2010)	Accuracy 87%	Support vector machine
(Das et al., 2006)	Accuracy 86%	Support vector machine
(Wrzodek et al., 2012)	Accuracy 91%	Support vector machine
(Zheng et al., 2013)	Accuracy 93%	Support vector machine
(Feltus et al., 2003)	Accuracy of 78%	Linear Discrimanent analysis
(Previti et al., 2009)	Accuracy of 89%	Decision tree
(Fang et al., 2006)	Specifity of 73%	Support vector machine

Because using all features for classification may not resulted in good prediction of methylation status of the CpG sites, we need to find smaller subset of features. This may lead to identification of more biologically important features that determine whether a specific CpG sites is methylated or not in a certain circumstance for example disease or non-disease. In order to find features which classifying the CpG sites better the hill climbing algorithm was used because the number of features is very high and we can't search the whole search space exhaustively. Set of experiments was designed to find feature subsets which measured by different metrics. The features that appeared in the final result of all this solutions are good candidates for further biological investigation they are important by the fact that they appear in all or most of the solutions.

6.1 Objectives

a) To find the smaller but more accurate classifying feature subset by hill climbing search.

b) To find features that shared in all experiments after applying hill climbing search.

6.2 Feature subset selection.

Features that were generated by the workflow system described in the previous chapter were used for classifying CpG sites. In this section, classification methods were applied to investigate which category of features gives better accuracy in predicting the class of CpG sites. In another set of experiments it was investigated which feature subsets better classify CpG sites, using hill climbing search algorithm. There are 737 features generated by the feature generating workflow, 19 were removed after initial filtering because the value of the feature was zero for all sites. The remaining features were used as an input for hill climbing search method.

The hill climbing algorithm encodes the feature subset with 0 and 1 as described in section 5.11 and creates new feature matrix from binary representation in each iterations. The program then used different metrics to search the feature space. Table 6-2 contains the pseudo code for this algorithm.

Table 6-2 Hill Climbing Search

```
Algorithm 6-1    Hill Climbing

Input : Feature matrix file, Best Starting point file , Number
of iterations , Classification method index, Measurement index

For solutions in Best Starting point file
    Extract binary representation
EndFor

For number of iterations
    Generate feature subset file from binary
    representation and feature matrix file
    Evaluate the feature subset file with
    cross validation on classification method defined
    by classification method index
    IF measurement index value is better tan previous
    value
        Bestvalue= newmeasurement
        BestBinaryrep= binary
        Make small change to binary representation
    EndIf
EndFor

Output : Bestbinaryrep
```

6.3 All features results from classification with different classification methods in WEKA

The WEKA package (Mark H et al. 2009) was used to analyse the all feature matrix obtained in the chapter 5. Table 6-3 shows the results for all features matrix. SVM shows the best performance when compared to the other two methods in terms of the true positive rate. Naïve Bayes shows a better performance than the other methods if we use kappa as measurement.

Table 6-3: results of applying different classification methods using all features.

All features	TP Rate	FP rate	Precisioı	Recall	F-Measure	ROC Area	Kappa
Naïve bayes	0.73	0.30	0.72	0.73	0.72	0.76	0.40
SVM	0.75	0.39	0.69	0.75	0.70	0.68	0.36
J48	0.66	0.36	0.65	0.66	0.65	0.65	0.26

6.4 All features results in each feature class type from classification with different classification methods in WEKA.

In this section each classification method was applied to the different types of features. These features were grouped based on common characteristics of the features.

6.4.1 Structural features

Banana and btwisted produced features that are related to structure of DNA. Using only these features generated the results which is shown in Table 6-4, this shows very poor performance using SVM and the best result was derived from J48. Using the structural features alone to do the classification gave the worse results than all features.

Table 6-4: results of applying classification method using only structural features.

Structural Features	TP Rate	FP Rate	Precisioı	Recall	F-Measure	ROCArea	Kappa
J48	0.72	0.38	0.68	0.72	0.70	0.72	0.34
SVM	0.68	0.68	0.53	0.68	0.56	0.50	0.00
Naïve Bayes	0.69	0.36	0.66	0.69	0.67	0.78	0.30

6.4.2 Transcription factor binding features

Transcription factor binding sites are features generated by counting the number of hits when scanning the CpG sequence with jaspscan. If there were over 80 percent overlap between the transcription factor binding site and the sequence 60 base pair up and down stream of CpG site they were selected. The results shows SVM performed better on this feature compared to other method. These set of features show slightly better results than those for all features using SVM and J48.

Table 6-5: results of applying classification method using only transcription factors.

Transcription factor binding site Features	TP Rate	FP Rate	Precisio	Recall	F-Measure	ROCArea	Kappa
J48	0.66	0.36	0.65	0.66	0.65	0.66	0.27
SVM	0.76	0.39	0.68	0.76	0.71	0.69	0.40
Naïve Bayes	0.69	0.34	0.68	0.69	0.68	0.73	0.32

6.4.3 Word count features

Word counts are features that were produced by counting words of size 4 base pairs in sliding windows of sequence in the sequence 60 base pairs up and down stream of the CpG sites. There are total of 256 features in this category. The overall results were better than structural and transcription factor binding sites. All classification methods work much better here than using all features specifically SVM.

Table 6-6 Results of applying classification method using only word counts.

Wordcounts features	TP Rate	FP Rate	Precisio	Recall	F-Measure	ROCArea	Kappa
J48	0.70	0.33	0.69	0.70	0.69	0.68	0.35
SVM	0.80	0.31	0.72	0.80	0.75	0.75	0.50
Naïve Bayes	0.73	0.29	0.73	0.73	0.71	0.81	0.39

6.5 Feature subset selection

Choosing a subset of features from all the features and optimising the measurements can lead to a better model. In order to examine this, initially 1000 random guesses made using a random binary representation of the feature and the top 10 binaries were selected for the 5 measurements of TP rate, Precision, F-Measure, ROC Area and

Kappa. These ranked-lists are chosen because in the hill-climbing we need some starting point and we choose the best starting points. These ranked lists were then run for 2000 iterations for each classification method (SVM, naïve bayes, and J48) by 10 fold cross validation. Neural network classification method was considered and implemented, but it showed very poor classification performance with hill climbing and was not used for further investigation. This made a total of 150 searches. Finally 15 maximum representations were selected for each ranked list. The following chart shows the experiment.

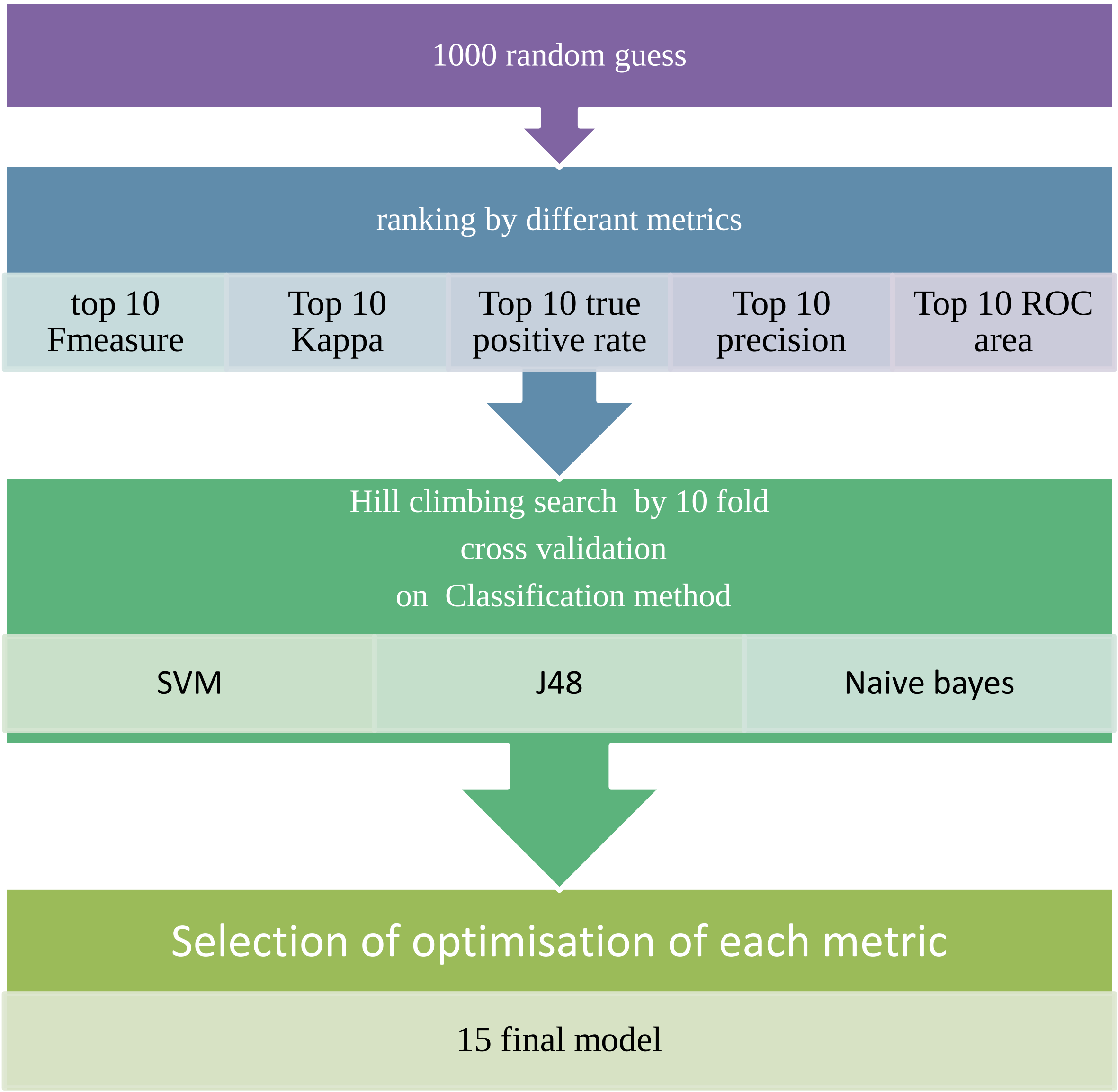

1000 random guess
ranking by differant metrics
top 10 Fmeasure
Top 10 Kappa
Top 10 true positive rate
Top 10 precision
Top 10 ROC area
Hill climbing search by 10 fold cross validation on Classification method
SVM
J48
Naive bayes
Selection of optimisation of each metric
15 final model

Figure 6-1: feature subset selection method.

The following chart shows that choosing best starting point was the right choice as it gives better results with the same number of iterations.

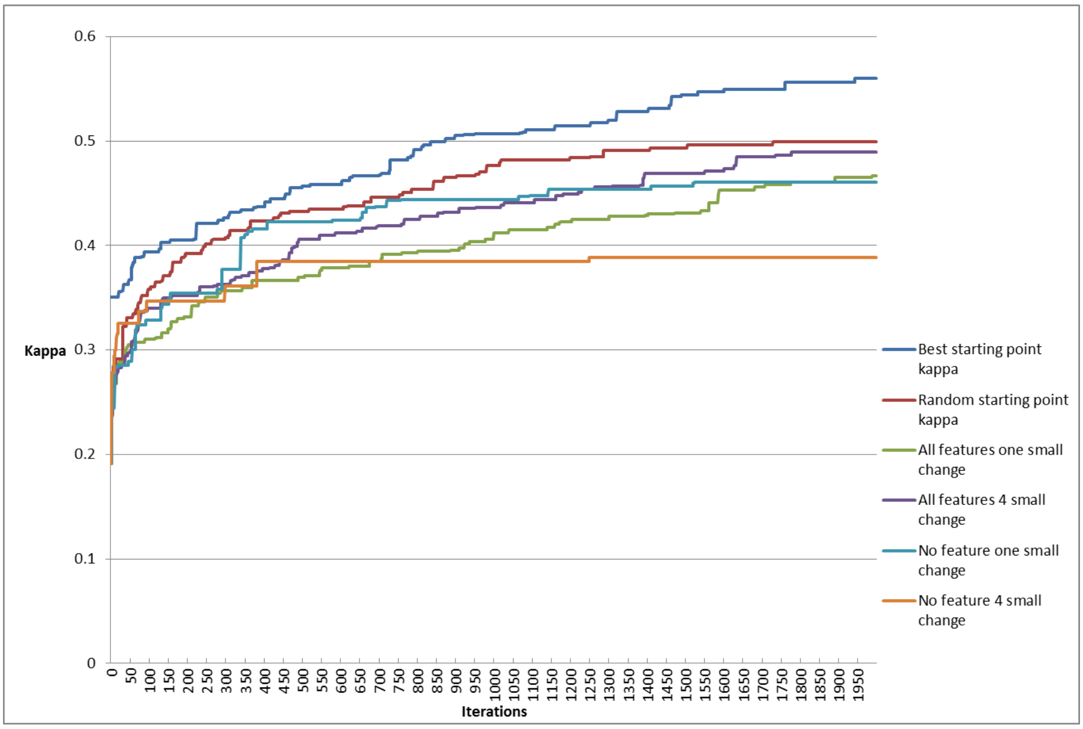


Figure 6-2 Comparison of best random starting point, random starting point, all feature starting point, zero feature starting point kappa maximisation with hill climbing

6.6 Hill climbing

By analysing the result it can be seen that some measurement methods indirectly improve other measurements for example choosing precision as the measurement and J48 as classification method, resulted in the model with the highest kappa of all models.

The improvement of each model from initial result is given in the Table 6-9. Interestingly highest improvement is reached by the kappa and the J48 classification method.

J48 initially had the lowest kappa among three classification methods by evaluating with all features in the matrix. This experiment shows maximum kappa, true positive rate and F-measure among all experiments. Lowest improvements were reached by SVM. The following table by (Landis and Koch, 1977) provides a suggestion on how the kappa statistics should be interpreted. Therefore according to this table the best result is near substantial.

Table 6-7 Kappa statistic strength of agreement

Kappa statistic	strength of agreement
<0	poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost perfect

By applying this table the improvement is very close to substantial (0.59) from nearly fair strength. The results from J48 experiments reached average of 0.55 kappa value.

Table 6-8 Details of expriments and values of each expriment for the maximum result

Expriment id	measurement	method	maximum of measurment	weightedTruePositiveRate	weightedFalsePositiveRate	weightedPrecision	weightedRecall	weightedFMeasure	weightedAreaUnderROC	kappa
top10_SVM_weightedRecall	TPR	SVM	0.81	0.81	0.29	0.73	0.81	0.77	0.76	0.54
top10_naivebayes_weightedRecall	TPR	nb	0.77	0.77	0.29	0.76	0.77	0.76	0.79	0.48
top10_J48_weightedRecall	TPR	J48	0.81	0.81	0.25	0.80	0.81	0.80	0.79	0.57
top10_SVM_weightedPrecision	Precision	SVM	0.75	0.77	0.36	0.75	0.77	0.74	0.71	0.44
top10_naivebayes_weightedPrecision	Precision	nb	0.78	0.78	0.24	0.78	0.78	0.77	0.80	0.51
top10_J48_weightedPrecision	Precision	J48	0.81	0.82	0.21	0.81	0.82	0.81	0.81	0.60
top10_SVM_KAPPA	kappa	SVM	0.55	0.81	0.26	0.73	0.81	0.77	0.77	0.55
top10_naivebayes_kappa	kappa	nb	0.52	0.78	0.24	0.78	0.78	0.77	0.80	0.52
top10_J48_kappa	kappa	J48	0.56	0.80	0.23	0.79	0.80	0.79	0.78	0.56
top10_SVM_weighteFmeasure	Fmeasure	SVM	0.77	0.81	0.28	0.73	0.81	0.77	0.76	0.54
top10_naivebayes_weighteFmeasure	Fmeasure	nb	0.77	0.78	0.24	0.78	0.78	0.77	0.79	0.51
top10_J48_weighteFmeasure	Fmeasure	J48	0.80	0.81	0.23	0.79	0.81	0.80	0.78	0.57
top10_SVM_weightedAreaUnderROC_list	Area under ROC curve	SVM	0.78	0.81	0.26	0.74	0.81	0.77	0.78	0.55
top10_naivebayes_weightedAreaUnderROC_list	Area under ROC curve	nb	0.92	0.80	0.18	0.82	0.80	0.79	0.92	0.57
top10_J48_weightedAreaUnderROC	Area under ROC curve	J48	0.83	0.76	0.24	0.75	0.76	0.75	0.83	0.48

Table 6-9 table shows direct and indirect improvement by hill climbing search.

id	measurement	method	initial	maximum of measurment	improvement from direct optimisation	maximum	Indirect improvement	source of maximum
top10_J48_weightedPrecision	Precision	J48	0.65	0.81	0.16	0.82	0.17	top10_nb_ROC_Area
top10_J48_weighteFmeasure	Fmeasure	J48	0.65	0.80	0.15	0.81	0.16	top10_J48_weightedPrecision
top10_J48_weightedRecall	TPR	J48	0.66	0.81	0.16	0.82	0.16	top10_J48_weightedPrecision
top10_J48_kappa	Kappa	J48	0.26	0.56	0.30	0.60	0.34	top10_J48_weightedPrecision
top10_J48_weightedAreaUnderROC	Area under ROC curve	J48	0.65	0.83	0.19	0.92	0.28	top10_naivebayes_weightedAreaUnderROC
top10_naivebayes_weightedAreaUnderROC	Area under ROC curve	nb	0.76	0.92	0.16	0.92	0.16	top10_naivebayes_weightedAreaUnderROC
top10_naivebayes_kappa	kappa	nb	0.40	0.52	0.12	0.60	0.20	top10_J48_weightedPrecision
top10_naivebayes_weightedPrec	Precision	nb	0.72	0.78	0.07	0.82	0.10	top10_nb_ROC_Area
top10_naivebayes_weighteFmea	Fmeasure	nb	0.72	0.77	0.06	0.81	0.09	top10_J48_weightedPrecision
top10_naivebayes_weightedReca	TPR	nb	0.73	0.77	0.05	0.82	0.09	top10_J48_weightedPrecision
top10_SVM_weightedAreaUnde	Area under ROC curve	SVM	0.68	0.78	0.10	0.92	0.24	top10_naivebayes_weightedAreaUnderROC
top10_SVM_KAPPA	kappa	SVM	0.36	0.55	0.18	0.60	0.23	top10_J48_weightedPrecision
top10_SVM_weightedRecall	TPR	SVM	0.75	0.81	0.06	0.82	0.07	top10_J48_weightedPrecision
top10_SVM_weighteFmeasure	Fmeasure	SVM	0.70	0.77	0.06	0.81	0.11	top10_J48_weightedPrecision
top10_SVM_weightedPrecision	Precision	SVM	0.69	0.75	0.06	0.82	0.13	top_10_nb_ROC_Area

By looking into the detail of each model It can be seen that by less than half the number of the features the program reach highest kappa and interestingly the second ranked model created by less than third of the original features in the feature matrix. So we get smaller model but better results. The three best results ordered by kappa have lowest number of features in the model comparing to original model. (See Table 6-10)

Table 6-10 : Proportion of features from different type of features in the final subset. Total number of feature column shows the proportion of features from the initial full feature space. Table is sorted by kappa value.

	structure	Transcription Factor binding site	wordcount	total feature	kappa
top10_J48_weightedPrecision	0.20	0.47	0.43	0.45	0.60
top10_naivebayes_weightedAreaUnderROC_list	0.20	0.24	0.38	0.29	0.57
top10_J48_weighteFmeasure	0.40	0.48	0.45	0.47	0.57
top10_J48_weightedRecall	0.00	0.48	0.45	0.47	0.57
top10_J48_kappa	0.00	0.48	0.45	0.47	0.56
top10_SVM_weightedAreaUnderROC_list	0.20	0.48	0.54	0.50	0.55
top10_SVM_KAPPA	0.20	0.48	0.55	0.50	0.55
top10_SVM_weightedRecall	0.20	0.48	0.54	0.50	0.54
top10_SVM_weighteFmeasure	0.20	0.49	0.53	0.50	0.54
top10_naivebayes_kappa	1.00	0.48	0.47	0.48	0.52
top10_naivebayes_weightedPrecision	0.80	0.50	0.47	0.49	0.51
top10_naivebayes_weighteFmeasure	0.60	0.48	0.48	0.48	0.51
top10_naivebayes_weightedRecall	0.00	0.48	0.53	0.49	0.48
top10_J48_weightedAreaUnderROC	0.20	0.47	0.48	0.47	0.48
top10_SVM_weightedPrecision	0.60	0.51	0.50	0.51	0.44

The Hamming distance between models calculated to find out how similar each models are from each other. The hamming distance is calculating the number of bits that are different between two binaries. All hamming distance for every pair of experiments calculated the result shows the average similarity is 0.55 with maximum 0.96 and minimum 0.44 for 105 pairs of experiments.

The final test is done with ranking the features using all models (feature subset) the final result shows the highest ranking features. The full list and details of features provided in the Appendix A.

Table 6-11 top 5 group of features that shared among different subsets. Percentage represents percentage of models that these features occurred in them details of each feature provided in Appendix A.

rank	percent	featurename
1	0.93	MA0035.1 MA0414.1 CCCC
2	0.86	MA0307.1 MA0119.1 MA0171.1 MA0117.1 MA0255.1 MA0250.1 MA0076.1 MA0128.1 MA0229.1 MA0427.1 MA0048.1 MA0121.1 MA0083.1 MA0287.1 MA0075.1 GCAG ACCT GCCT ATGC TGCC CTCA
3	0.8	MA0248.1 MA0112.1 MA0010.1 MA0297.1 MA0275.1 MA0077.1 MA0346.1 MA0219.1 MA0413.1 MA0244.1 MA0179.1 MA0245.1 MA0353.1 MA0266.1 MA0150.1 MA0440.1 MA0213.1 MA0348.1 MA0178.1 MA0089.1 MA0159.1 AACT AGAA AGCG ATAC ACCC ACAT GATT GAGT TGAC CGAT GTGA GTTG TCGC TTCG TCAG CTTA
4	0.73	MA0137.2 MA0438.1 MA0230.1 MA0049.1 MA0109.1 MA0384.1 MA0442.1 MA0302.1 MA0003.1 MA0294.1 MA0043.1 MA0007.1 MA0214.1 MA0021.1 MA0138.1 MA0233.1 MA0130.1 MA0406.1 MA0227.1 MA0152.1 MA0329.1 MA0443.1 MA0148.1 MA0351.1 MA0232.1 AGTT ATCC GCTG GTGC GCTC GCGC GGGT TACC GTCC GGCT GACT CACT CGGG CATG CAGC GTTC CGTT CGAA TCCG CGCC AAAT
5	0.66	MA0073.1 MA0094.2 MA0444.1 MA0202.1 MA0408.1 MA0432.1 MA0358.1 MA0320.1 MA0305.1 MA0270.1 MA0453.1 MA0293.1 MA0415.1 MA0381.1 MA0288.1 MA0424.1 MA0011.1 MA0201.1 MA0246.1 MA0390.1 MA0208.1 MA0388.1 MA0217.1 MA0342.1 MA0272.1 MA0199.1 MA0460.1 MA0095.1 MA0347.1 MA0037.1 MA0065.2 MA0215.1 MA0449.1 MA0402.1 MA0357.1 MA0435.1 MA0254.1 MA0336.1 MA0373.1 MA0165.1 MA0085.1 MA0457.1 MA0194.1 MA0389.1 MA0252.1 AAGT AACA ATAA ACCA ACAG GCTT GACG CAAG GGCC GTAC GCCC TCAT TGTA TCTC TTAA CACA CGCG CTAT TTTG AAGA CCTC TAAA

6.7 Conclusion

The results in this chapter show a hill climbing search can reduce the number of features and at the same time improve the classification results. The first and second best results ranked by kappa shows the number of features are less than half of original size. Some features that shared among most of the models are provided in Table 6-11. Most of them are transcription factor binding sites and word count features this shows that DNA sequence surrounding CpG is the most important in predicting the methylation status. These feature subsets with their algorithm provides predictive models of DNA methylation. Different models provided in this chapter could be used depending on the measurement required. Some features are shared among different models interestingly GGCC as a feature around CpG sites is shared among 10 out of 15 of the models. This motif is very similar to motif CCGG that is found in trinucleotide repeat diseases. It is shown in chapter 3 that this motif has discriminative value for TNR disease.

Chapter 7 Summary and Future Work

7 Introduction

The aim of this research was to analyse the less investigated epigenetic methylation changes in single CpG sites methylation using large scale datasets and create predictive model of DNA methylation. More specifically we looked at large scale datasets for illumine 450k experiments for cancer samples, and also compared of CpG regions in three trinucleotide repeat expansion diseases, using existing data generated by high throughput technologies or reported in papers. Grid enabled workflows were created to generate features related to the sequence of CpG sites. The goal of creating this workflow was to provide a way to add new feature generating components without changes to other parts of the system. Another goal was to provide a faster way of features generation and feature subset selection by connecting a workflow to a desktop grid. Subsets of these features then were selected with a hill climbing search. These gave better prediction of CpG methylation class by different measurement and classification methods. Features which were shared among all feature subset were identified. In order to identify similar features in cancer and trinucleotide repeat disease these feature were compared. The features that are shared amongst cancer and trinucleotide repeat diseases may have biological importance. The system and methodology developed for achieving these goals not restricted to the context of CpG site DNA methylation in the diseases studied here, but also to other diseases, conditions, aging and development studies. This chapter summarise the finding achieved in this research. The following diagram shows how each contribution achieved using different methods and algorithms.

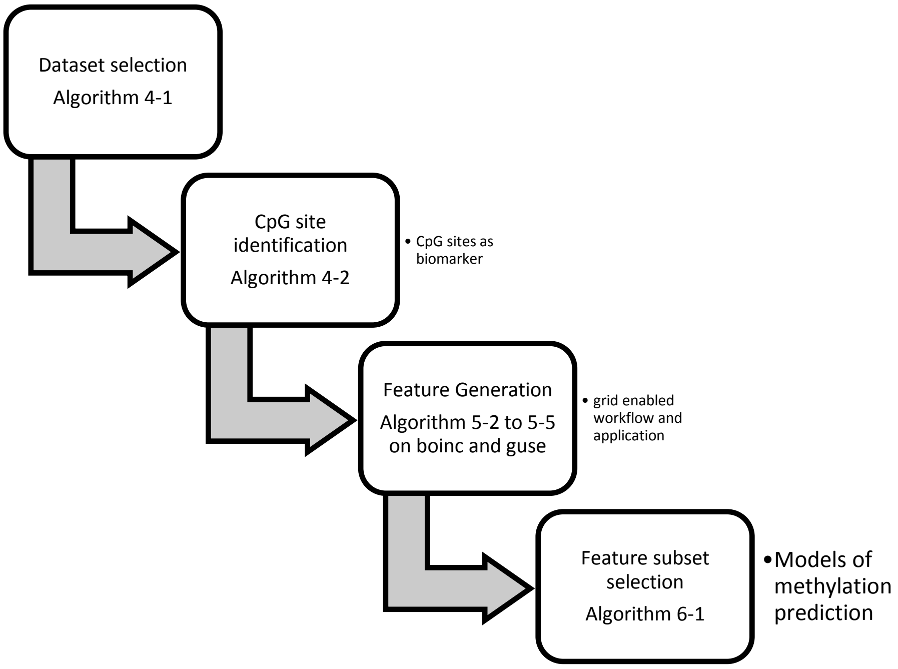


Figure 7-1 Relationship between algorithms developed as part of this research

7.1 Findings

7.1.1 Distinguishing motifs in trinucleotide repeat disease

In chapter 3 three trinucleotide expansion diseases were compared. The data for DNA methylation for each disease was generated in separate studies. This data was aggregated. Then a computational comparison of this data found motifs which distinguished different CpG regions in the diseases among these motifs. In this study all possible patterns of 5 base pairs were examined for short motifs this approach can be used and it showed that it has much better results than the MEME algorithm. Our results show that there are sequence patterns which can be used to distinguish between always methylated, variably methylated and never methylated regions of these TNR genes. A single pattern can be used to distinguish the never methylated region from the other two. As side issue this study reveals the need for standard representation of DNA methylation data in publication and storage of these data in public database for future comparison.

7.1.2 Identification of CpG sites in different classes

In chapter 4 datasets from one microarray platform for methylation detection which has the largest coverage and highest number of deposited samples in public databases was chosen for identifying four different classes of CpG sites. These sites were 1) hypermethylated in disease and hypomethylated in normal samples 2) hypomethylated in disease and hypermethylated in cancer, 3) sites which were never methylated in any samples 4) sites which were always methylated across all samples.

The samples used in the study were selected by reading the platform soft file from GEO database and samples were filtered by using keyword healthy normal, after filtering 535 data samples which 301 of them were cancer samples and 234 were normal samples were selected. These samples contained more than 450,000 CpG site and a total of 259,783,695 data points. Raw data of these samples were processed to generate CSV files.

By processing the samples 653 total CpG sites were selected from more than 450,000 sites. These sites were selected using ratio of the methylation and non-methylation intensity in the samples CSV files. This ratio is called beta-value. CpG sites which their beta value was more than 0.8 for all samples were classified as methylated sites. CpG sites with the beta values under 0.2 for all samples were classified as never methylated. CpG sites with the beta value under 0.2 for sixty percent of normal samples and beta value more than 0.8 for sixty percent of cancer samples, classified as hypermethylated in cancer and hypomethylated in normal. CpG sites with the beta value under 0.2 for sixty percent of cancer samples and beta value more than 0.8 for sixty percent of normal samples, classified as hypermethylated in normal and hypomethylated in cancer. 447 CpG sites are in never methylated class, 148 sites are in always methylated, 51 hypomethylated in normal and hypermethylated in cancer and 7 sites hypermethylated in normal and hypomethylated in cancer. These sites can be used as the biomarkers. They also provide dataset to further analyse the features associated in each to each CpG site classes.

Sixty base pair DNA sequence of upstream and downstream of these sites were used in the MEME software and motifs that shared among these classes were identified. All of these motifs were used in the MAST software to find the number of times motifs occurred in all sequences of CpG sites. The results from MAST program were processed to create a motif matrix. Since motifs which distinguish normal and cancer samples are of most interest. Motif matrix was used as an input to J48 classification algorithm which resulted in 94.82 correct classifications of CpG sites.

Genes near these sites were examined with DAVID software they showed that they are associated with the “Apoptosis” term in the GO ontology. This indicated that a large proportion of these genes are involved or predicted to be involved in apoptosis.

7.1.3 Fast and Scalable feature generation system,

In chapter 5 details of development of a grid enabled workflow system is discussed. This system contained gUse which is connected to the BOINC systems. Workflow proposed in chapter 5 was used to generate features related to DNA sequence around CpG sites, adding new software which generates features using DNA sequence can be achieved by adding new node to the workflow. These nodes can be any kind of node supported by gUse system and they do not necessarily need to be BOINC enabled. gUse provides easy to use web based user interface and gateway to the BOINC infrastructure.

Five applications were ported to the BOINC infrastructure by GenWrapper. These applications can be used in any BOINC project that has GenWrapper as their app provided they have proper submitter. For the submission of jobs gUse system and 3Gbridge used which provides web service to submit jobs to BOINC. The resulting matrix generated by this workflow have used in another workflow to search for subset of features that can better distinguish classes from each other using classification methods: J48 tree (WEKA of C4.5 decision tree algorithm), support vector machines, and naïve Bayes method.

The performance test of these two workflows showed up to point 5 times speed up by using 7 machines as the test environment. Because test environment runs on virtual machines, the machines can be easily ported to new physical machines, increasing the number of workers in the system. In the case of the feature generating workflow because running time of each sequence is very low. More than one sequence per file should be added to each file, for each jobs running on the workers to ensure speed-up is achieved. In this research the ported applications were programmed to handle this, although three of the original applications can’t handle more than one sequence, using shell script and calling applications multiple time resolved this problem. The user only needs to define the number of sequences per file in one of workflows generator nodes.

The heuristic search is more computational intensive and one execution of each run on single input for J48 and SVM methods takes more than 8 hours and since each experiment contains 10 inputs it takes 3 days to finish on single machine, but because each single input can be run independently they are very good candidate to run in the grid and BOINC environment. Nearly 4 times speedup was gained by using the BOINC environment and the run time of each experiment decreased to less than a day.

7.1.4 Predictive model of DNA methylation

Chapter 6 provides the results of the feature subset selection in 15 experiments by using all features as input to the classification method the results shows 0.72, 0.74, 0.65 true positive rate for naïve Bayes, SVM , J48 respectively for classification methods. The kappa did not show not promising results for all features. After feature subset selection, features subsets were found that can classify the CpG sites with 0.81 true positive rates and a kappa of 0.59 which is almost “substantial” using the kappa range suggested by (Landis and Koch, 1977) in Table 6-7 and Table 6-6. The kappa increased nearly 0.33 after optimisation with J48. It was also revealed that direct optimisation on kappa gains less than optimisation on precision measure, so it is recommended that for feature subset selection a range of experiments and measurements be considered. The number of features in the subsets with highest kappa was less than half of all features generated initially. This means that we found subset with much smaller size but more accurate prediction.

All models generated by the 15 experiments were compared with hamming distance. The results showed they have 0.55 similarity on average with maximum of 0.96 and minimum 0.45 similarity. Finally all these models added together to find the features occurrence in all models. 3 features were found that exists in 14 out of 15 models. Similarly 21, 37, 46, and 67 features found in 13,12,11,10 models respectively. The full list of ranked features and related information on these features are provided in the Appendix.

7.2 Future work

This section provides future works and improvement to this research.

7.2.1 Different methylation contexts

The goal of the grid enabled system provided in this research was to identify and analyse CpG site DNA methylation in cancer and normal cells. This system has the potential to use in other context. For example looking at differences in tissues or using cell lines. Similar to this study a workflow can be used for feature generation and feature subset selection. New experiments can be designed to investigate data deposited in public databases in these situations.

7.2.2 Improvement of feature generation

This research we used BOINC for accelerating the feature generation and feature subset selection. It can be seen in the workflow this system is very similar to “map reduce” paradigm. Hadoop system as implementation of “map-reduce” paradigm can be investigated and compared with BOINC (White, T 2012). It is possible to report the result to a database when they arrived at a server instead of collecting them in the file. The rows of database represent CpG sites and column represents features. In this way a user may check database and database entries can be visualised so the user can at some point decide to use the results and deal with the missing values by statistical methods. In this way all feature results are not dependant on single job failure. These suggestions can improve the system and accelerate feature generation task.

7.2.3 Improvement to hill climbing search

Because in BOINC systems there might be thousands of computers, the search space can be easily expanded and all initial random guesses can be expanded and run on a grid theoretically, but because each running time takes 8 hours for 2000 iterations, if one wants to expand this it needs to make checkpoints at some iterations and the workunits can start from the place which they ended. For example instead of running 2000 times, the jobs can be run 200 times on each machine, after each 200 times they returned the result back to BOINC and are rescheduled for new submission in this way if one job has failed we don’t lose all the iterations. It is also possible to parallelise the “small change” task in hill climbing search algorithm to different part of solution. For example 100 machines can work each on 1/100 of features and keeping the other parts unchanged. It is also possible to connect these to workflows together to have full automated process of feature generation and feature subset selection as it is depicted in the Figure 7-2.

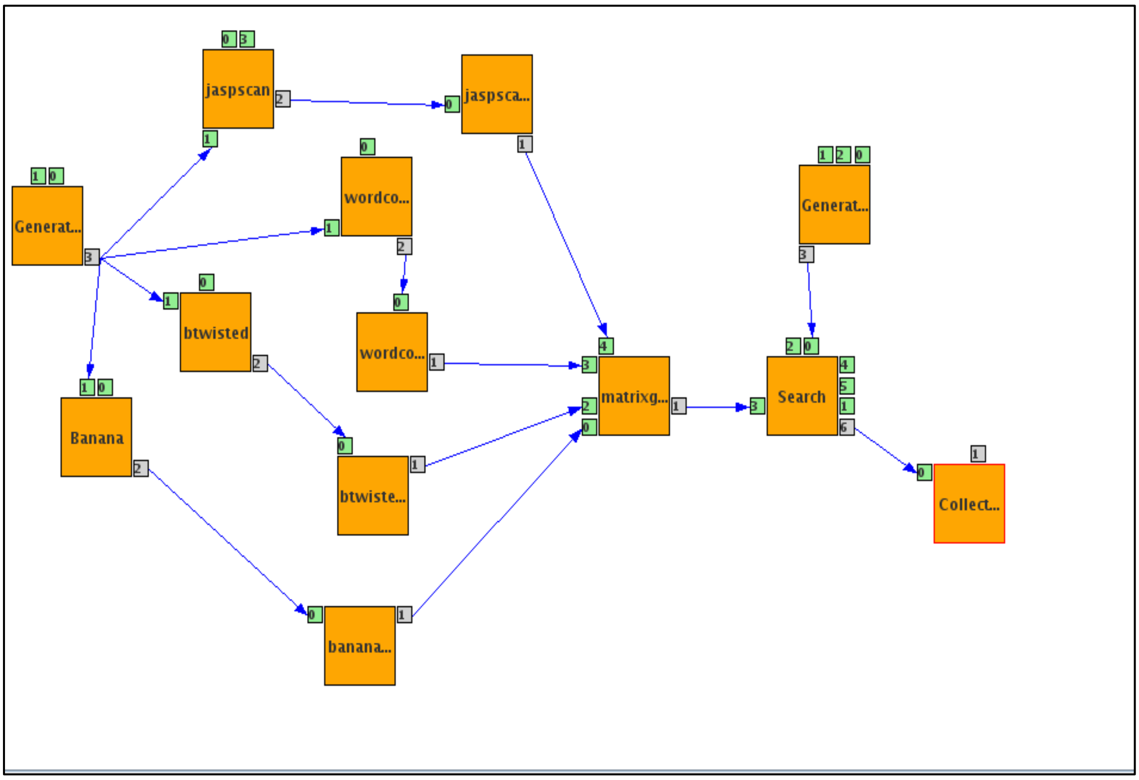


Figure 7-2 Feature generation and subset selection workflows

7.2.4 Conclusion

I have shown that computational methods can enhance and enrich the knowledge obtained from biological data. Further the system developed to carry out this research can be used in other contexts. This system provides user friendly fast and scalable environment for analysis of DNA methylation.

References

Abdennadher, N. (2012) *Combining Cloud, Grid and Volunteer computing*. Available at: http://wiki.scc.kit.edu/gridkaschool/index.php/Combining_Cloud,_Grid_and_Volunteer_computing (Accessed: 08/05 2013).

Abdennadher, N. and Boesch, R. (2005) "Towards a peer-to-peer platform for high performance computing", *High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Eighth International Conference on* IEEE, pp. 8 pp.

Abhishek Kalapatapu and Mahasweta Sarkar (2011) "Cloud Computing an overview" in *Cloud Computing: Methodology, Systems, and Applications*, eds. L. Wang, R. Ranjan, J. Chen and B. Benatallah, CRC Press (an imprint of Taylor & Francis), pp. 20-22.

Adams, D., Altucci, L., Antonarakis, S.E., Ballesteros, J., Beck, S., Bird, A., Bock, C., Boehm, B., Campo, E. and Caricasole, A. (2012) "BLUEPRINT to decode the epigenetic signature written in blood", *Nature biotechnology*, vol. 30, no. 3, pp. 224-226.

Alberts, B. (2009) "Protein structure and function" in *Essential cell biology* Garland Science, pp. 119.

Ali, I. and Seker, H. (2010) "A comparative study for characterisation and prediction of tissue-specific DNA methylation of CpG islands in chromosomes 6, 20 and 22", *Conference proceedings : ...Annual International Conference of the IEEE Engineering in Medicine and Biology Society.IEEE Engineering in Medicine and Biology Society.Conference*, vol. 2010, pp. 1832-1835.

Al-Mahdawi, S., Pinto, R.M., Ismail, O., Varshney, D., Lymperi, S., Sandi, C., Trabzuni, D. and Pook, M. (2008) "The Friedreich ataxia GAA repeat expansion mutation induces comparable epigenetic changes in human and transgenic mouse brain and heart tissues", *Human molecular genetics*, vol. 17, no. 5, pp. 735-746.

Alpaydin, E. (2010) *Introduction to machine learning*, MIT Press.

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludaescher, B. and Mock, S. (2004a) "Kepler: Towards a grid-enabled system for scientific workflows", *the Workflow in Grid Systems Workshop in GGF10-The Tenth Global Grid Forum, Berlin, Germany*.

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B. and Mock, S. (2004b) "Kepler: an extensible system for design and execution of scientific workflows", *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on* IEEE pp. 423.

Anderson, D.P. (2004) "Boinc: A system for public-resource computing and storage", *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on* IEEE, pp. 4.

Attila Csaba Maros (2010) *Application porting to BOINC – DC-API, GenWrapper, 3G Bridge*. Available at: http://desktopgridfederation.org/c/document_library/get_file?uuid=6d781120-40b5-4f4b-9593-b478c9e8ead1&groupId=10939 (Accessed: 08/10 2013).

Available science gateways, 2013. Available from: http://guse.hu/portals/sg . (Accessed August 2013).

Bailey, T.L., Boden, M., Buske, F.A., Frith, M., Grant, C.E., Clementi, L., Ren, J., Li, W.W. and Noble, W.S. (2009) "MEME SUITE: tools for motif discovery and searching", *Nucleic acids research*, vol. 37, no. suppl 2, pp. W202-W208.

Bailey, T.L., Williams, N., Misleh, C. and Li, W.W. (2006) "MEME: discovering and analyzing DNA and protein sequence motifs", *Nucleic acids research*, vol. 34, no. suppl 2, pp. W369-W373.

Barrett, T., Wilhite, S.E., Ledoux, P., Evangelista, C., Kim, I.F., Tomashevsky, M., Marshall, K.A., Phillippy, K.H., Sherman, P.M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C.L., Serova, N., Davis, S. and Soboleva, A. (2013) "NCBI GEO: archive for functional genomics data sets—update", *Nucleic acids research*, vol. 41, no. D1, pp. D991-D995.

Birney, E. (2012) "The making of ENCODE: Lessons for big-data projects", *Nature*, vol. 489, no. 7414, pp. 49-51.

Bock, C. and Lengauer, T. (2008) "Computational epigenetics", *Bioinformatics*, vol. 24, no. 1, pp. 1-10.

Bock, C., Walter, J., Paulsen, M. and Lengauer, T. (2007) "CpG Island Mapping by Epigenome Prediction", *PLoS Comput Biol*, vol. 3, no. 6, pp. e110.

BOINC 2013 : *Choosing BOINC project* Available at (http://boinc.berkeley.edu/projects.php). (Accessed April 2013)

Boincstat :*Projects stat info* Available at http://boincstats.com/en/stats/projectStatsInfo (Accessed July 2013)

Btwisted 2013: *Btwisted* Available at http://bioweb2.pasteur.fr/docs/EMBOSS/btwisted.html (Accessed April 2013)

Burke Stephen, Campana Simone, Lanciotti Elisa, Litmaath Maarten, Lorenzo Patricia Mendez ´, Vincenzo Miccio, Nater Christopher, Santinelli Roberto and Sciaba Andrea (2011) *GLITE 3.2 USER GUIDE*. Available at: http://www.isragrid.org.il/wp-content/uploads/2013/05/gLite-3-UserGuide.pdf (Accessed: 08/04 2013).

Cappello, F., Djilali, S., Fedak, G., Herault, T., Magniette, F., Néri, V. and Lodygensky, O. (2005) "Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with grid", *Future Generation Computer Systems*, vol. 21, no. 3, pp. 417-437.

Carletta, J. (1996) "Assessing agreement on classification tasks: the kappa statistic", *Computational linguistics*, vol. 22, no. 2, pp. 249-254.

Chang, H., Niyogi, D., Chen, F., Kumar, A., Song, C., Zhao, L., Govindaraju, R.S., Merwade, V., Lei, M. and Scheeringa, K. (2008) "Developing a TeraGrid Based Land Surface Hydrology and Weather Modeling Interface", *Proceedings of the TeraGrid 2008 Conference*.

Chen, F., Song, J., Di, J., Zhang, Q., Tian, H. and Zheng, J. (2012) "IRF1 suppresses Ki-67 promoter activity through interfering with Sp1 activation", *Tumor Biology*, vol. 33, no. 6, pp. 2217-2225.

Cho, D.H., Thienes, C.P., Mahoney, S.E., Analau, E., Filippova, G.N. and Tapscott, S.J. (2005) "Antisense transcription and heterochromatin at the DM1 CTG repeats are constrained by CTCF", *Molecular cell*, vol. 20, no. 3, pp. 483-489.

Chung, D.W., Rudnicki, D.D., Yu, L. and Margolis, R.L. (2011) "A natural antisense transcript at the Huntington's disease repeat locus regulates HTT expression", *Human molecular genetics*, vol. 20, no. 17, pp. 3467-3477.

Cooper, S.J., Trinklein, N.D., Nguyen, L. and Myers, R.M. (2007) "Serum response factor binding sites differ in three human cell types", *Genome research*, .

Coremine (2013) *Coremine medical*. Available at: http://www.coremine.com/medical/ (Accessed: April 2013).

Da Wei Huang, Brad T Sherman and Lempicki, R.A. (2008) "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources", *Nature protocols*, vol. 4, no. 1, pp. 44-57.

Das, R., Dimitrova, N., Xuan, Z., Rollins, R.A., Haghighi, F., Edwards, J.R., Ju, J., Bestor, T.H. and Zhang, M.Q. (2006) "Computational prediction of methylation status in human genomic sequences", *Proceedings of the National Academy of Sciences*, vol. 103, no. 28, pp. 10713-10716.

De Biase, I., Chutake, Y.K., Rindler, P.M. and Bidichandani, S.I. (2009) "Epigenetic silencing in Friedreich ataxia is associated with depletion of CTCF (CCCTC-binding factor) and antisense transcription", *PloS one*, vol. 4, no. 11, pp. e7914.

Desktopgridfederation: online (Accessed July 2013) http://desktopgridfederation.org/

Douglas Roberts, C.H. (2007) *Genome-wide Monitoring of CpG Island Methylation With Agilent's Tiling Microarray Technology*. Available at: http://www.chem.agilent.com/library/applications/5989-6838EN_FINAL_low.pdf (Accessed: 08/04 2013).

Du, P. (2010) "Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis", *BMC Bioinformatics,* vol. 11, pp. 587.

Eckhardt, F., Lewin, J., Cortese, R., Rakyan, V.K., Attwood, J., Burger, M., Burton, J., Cox, T.V., Davies, R. and Down, T.A. (2006) "DNA methylation profiling of human chromosomes 6, 20 and 22", *Nature genetics,* vol. 38, no. 12, pp. 1378-1385.

Ellert, M., Grønager, M., Konstantinov, A., Kónya, B., Lindemann, J., Livenson, I., Nielsen, J.L., Niinimäki, M., Smirnova, O. and Wäänänen, A. (2007) "Advanced Resource Connector middleware for lightweight computational Grids", *Future Generation Computer Systems,* vol. 23, no. 2, pp. 219-240.

Encode Project, (2013). Available at: http://www.nature.com/encode/#/threads (April 2013).

Erwin, D.W. (2002) "UNICORE—a Grid computing environment", *Concurrency and Computation: Practice and Experience,* vol. 14, no. 13-15, pp. 1395-1410.

Fan, S., Zou, J., Xu, H. and Zhang, X. (2010) "Predicted methylation landscape of all CpG islands on the human genome", *Chinese Science Bulletin,* vol. 55, no. 22, pp. 2353-2358.

Fang, F., Fan, S., Zhang, X. and Zhang, M.Q. (2006) "Predicting methylation status of CpG islands in the human brain", *Bioinformatics,* vol. 22, no. 18, pp. 2204-2209.

Feltus, F.A., Lee, E.K., Costello, J.F., Plass, C. and Vertino, P.M. (2006) "DNA motifs associated with aberrant CpG island methylation", *Genomics,* vol. 87, no. 5, pp. 572-579.

Feltus, F., Lee, E., Costello, J., Plass, C. and Vertino, P. (2003) "Predicting aberrant CpG island methylation", Proceedings of the National Academy of Sciences, vol. 100, no. 21, pp. 12253-12258.

Filippone, M., Masulli, F. and Rovetta, S. (2006) "Supervised classification and gene selection using simulated annealing", *Neural Networks, 2006. IJCNN'06. International Joint Conference on* IEEE, pp. 3566.

Folding, 2013. Available at: http://folding.stanford.edu (Accessed July 2013).

Foster, I. and Kesselman, C. (1998) *Computational Grids*. Available at: http://globusproject.com/alliance/publications/papers/chapter2.pdf (Accessed: 08/04 2013).

Foster, I., Kesselman, C. and Tuecke, S. (2001) "The anatomy of the grid: Enabling scalable virtual organizations", *International journal of high performance computing applications,* vol. 15, no. 3, pp. 200-222.

Frommer, M., McDonald, L.E., Millar, D.S., Collis, C.M., Watt, F., Grigg, G.W., Molloy, P.L. and Paul, C.L. (1992) "A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands.", *Proceedings of the National Academy of Sciences,* vol. 89, no. 5, pp. 1827-1831.

Fu, Y.H., Kuhl, D.P., Pizzuti, A., Pieretti, M., Sutcliffe, J.S., Richards, S., Verkerk, A.J., Holden, J.J., Fenwick, R.G.,Jr and Warren, S.T. (1991) "Variation of the CGG repeat at the fragile X site results in genetic instability: resolution of the Sherman paradox", *Cell,* vol. 67, no. 6, pp. 1047-1058.

Ghosh, P. and Bagchi, M. (2009) "QSAR modeling for quinoxaline derivatives using genetic algorithm and simulated annealing based feature selection", *Current medicinal chemistry,* vol. 16, no. 30, pp. 4032-4048.

Globus, (2013a) *About the globus toolkit*. Available at: http://www.globus.org/toolkit/about.html (Accessed: 08/04 2013).

Globus, (2013b) *what is Globus Provision?* Available at: http://globus.org/provision/ (Accessed: 08/04 2013).

Globus project: *About the Globus Toolkit* Available at http://www.globus.org/toolkit/about.html (Accessed July 2013)

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 533-549.

Goble, C.A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., Borkum, M., Bechhofer, S., Roos, M. and Li, P. (2010) "myExperiment: a repository and social network for the sharing of bioinformatics workflows", *Nucleic acids research,* vol. 38, no. suppl 2, pp. W677-W682.

Goecks, J., Nekrutenko, A., Taylor, J. and Team, T.G. (2010) "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences", *Genome Biol,* vol. 11, no. 8, pp. R86.

González, F. and Belanche, L.A. (2013) "Feature Selection for Microarray Gene Expression Data using Simulated Annealing guided by the Multivariate Joint Entropy", *arXiv preprint arXiv:1302.1733,* .

Grabczyk, E., Kumari, D. and Usdin, K. (2001) "Fragile X syndrome and Friedreich's ataxia: two different paradigms for repeat induced transcript insufficiency", *Brain research bulletin,* vol. 56, no. 3-4, pp. 367-373.

Grewal, S.I.S. and Jia, S. (2007) "Heterochromatin revisited", *Nature reviews. Genetics,* vol. 8, no. 1, pp. 35-46.

Hand, D.J., Mannila, H. and Smyth, P. (2001a) "Introducttion" in *Principles of Data Mining* MIT Press, pp. 13.

Hand, D.J., Mannila, H. and Smyth, P. (2001b) "Predictive Modelling for Classification" in *Principles of Data Mining* MIT Press, pp. 353-356.

Hartl, D.L. and Ruvolo, M. (2011) "Genes , Genomics and Genetic Analysis" in *Genetics analysis of genes and genomics* Jones & Bartlett Learning, pp. 23.

Hervouet, E., Vallette, F.M. and Cartron, P.F. (2009) "Dnmt3/transcription factor interactions as crucial players in targeted DNA methylation", *Epigenetics : official journal of the DNA Methylation Society*, vol. 4, no. 7, pp. 487-499.

Heyn, H., Carmona, F.J., Gomez, A., Ferreira, H.J., Bell, J.T., Sayols, S., Ward, K., Stefansson, O.A., Moran, S., Sandoval, J., Eyfjord, J.E., Spector, T.D. and Esteller, M. (2013) "DNA methylation profiling in breast cancer discordant identical twins identifies DOK7 as novel epigenetic biomarker", *Carcinogenesis*, vol. 34, no. 1, pp. 102-108.

hgu : *Medical research council human genetics unit* Available at: http://www.hgu.mrc.ac.uk/img/researchers_img/meehan/DNA_Methylation_in_Vertebrates_a.jpg (Accsessed August 2013)

Hogart, A., Lichtenberg, J., Ajay, S.S., Anderson, S., NIH Intramural Sequencing Center, Margulies, E.H. and Bodine, D.M. (2012) "Genome-wide DNA methylation profiles in hematopoietic stem and progenitor cells reveal overrepresentation of ETS transcription factor binding sites", *Genome research*, vol. 22, no. 8, pp. 1407-1418.

Howe, B., Ribalet, F., Chitnis, S., Armbrust, G. and Halperin, D. (2013) "SQLShare: Scientific Workflow Management via Relational View Sharing".

Hsieh, Y., Wu, T., Huang, C., Hsieh, Y. and Liu, J. (2007) "Suppression of tumorigenicity of human hepatocellular carcinoma cells by antisense oligonucleotide MZF-1", *Chinese Journal of Physiology*, vol. 50, no. 1, pp. 9-15.

Illumina (2012a) *DNA methylation analysis*. Available at: http://res.illumina.com/documents/products/datasheets/datasheet_dna_methylation_analysis.pdf (Accessed: 08/04 2013).

Illumina (2012b) *HumanMethylation450 BeadChip Achieves Breadth of Coverage Using Two Infinium Chemistries*. Available at: http://res.illumina.com/documents/products/technotes/technote_hm450_data_analysis_optimization.pdf (Accessed: 08/04 2013).

Illumina (2012c) *Illumina humanmethylation450 datasheet*. Available at: http://www.illumina.com/documents/products/datasheets/datasheet_humanmethylation450.pdf (Accessed: 08/04 2013).

Inoue, M., Takahashi, K., Niide, O., Shibata, M., Fukuzawa, M. and Ra, C. (2005) "LDOC1, a novel MZF-1-interacting protein, induces apoptosis", *FEBS letters*, vol. 579, no. 3, pp. 604-608.

Iorio, M.V., Piovan, C. and Croce, C.M. (2010) "Interplay between microRNAs and the epigenetic machinery: an intricate network", *Biochimica et biophysica acta*, vol. 1799, no. 10-12, pp. 694-701.

JA Deutsch, A., Angerer, H., E Fuchs, T. and Neumeister, P. (2012) "The nuclear orphan receptors NR4A as therapeutic target in cancer therapy",*Anti-Cancer Agents in Medicinal Chemistry-Anti-Cancer Agents),* vol. 12, no. 9, pp. 1001-1014.

Jasper 2013: *Jasper* available at http://jaspar.genereg.net/ (Accessed 2013)

K.Mulligan,Robert C.King William D.Stansfield Pamela (2007) *bioinformatics in (Dictionary of genetics)*. Available at: http://www.oxfordreference.com/view/10.1093/acref/9780195307610.001.0001/acref-9780195307610-e-0711 (Accessed: 08/04 2013).

Kacsuk, P., Farkas, Z., Kozlovszky, M., Hermann, G., Balasko, A., Karoczkai, K. and Marton, I. (2012) "WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities", *Journal of Grid Computing,* vol. 10, no. 4, pp. 601-630.

Kacsuk, P., Kovacs, J., Farkas, Z., Marosi, A.C., Gombas, G. and Balaton, Z. (2009) "SZTAKI desktop grid (SZDG): a flexible and scalable desktop grid system", *Journal of Grid Computing,* vol. 7, no. 4, pp. 439-461.

Karaboga, D., & Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31(1), 61-85.

Karolchik, D., Baertsch, R., Diekhans, M., Furey, T.S., Hinrichs, A., Lu, Y., Roskin, K.M., Schwartz, M., Sugnet, C.W. and Thomas, D.J. (2003) "The UCSC genome browser database", *Nucleic acids research,* vol. 31, no. 1, pp. 51-54.

Katoh, K. and Toh, H. (2010) "Parallelization of the MAFFT multiple sequence alignment program", *Bioinformatics,* vol. 26, no. 15, pp. 1899-1900.

Kepler (2013) *Projects Using Kepler*. Available at: https://kepler-project.org/users/projects-using-kepler (Accessed: 08/04 2013).

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6), 975-986.

Klose, R.J. and Bird, A.P. (2006) "Genomic DNA methylation: the mark and its mediators", *Trends in biochemical sciences,* vol. 31, no. 2, pp. 89-97.

Kumar, R., Tyagi, S. and Sharma, M. (2013) "Memetic Algorithm: Hybridization of Hill Climbing with Selection Operator", *International journal of Soft Computing and Enggineering,* vol. 3, no. 2, pp. 140-145.

Laird, P.W. (2010) "Principles and challenges of genomewide DNA methylation analysis", *Nat Rev Genet,* vol. 11, no. 3, pp. 191-203.

Landis, J.R. and Koch, G.G. (1977) "An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers", *Biometrics,* pp. 363-374.

Laure, E., Edlund, A., Pacini, F., Buncic, P., Beco, S., Prelz, F., Di Meglio, A., Mulmo, O., Barroso, M. and Kunszt, P.Z. (2004) *Middleware for the next generation Grid infrastructure*. Available at: http://cds.cern.ch/record/865715/files/p826.pdf?version=1 (Accessed: 08/05 2013).

Liao, D. (2009) "Emerging roles of the EBF family of transcription factors in tumor suppression", *Molecular Cancer Research,* vol. 7, no. 12, pp. 1893-1901.

Lister, R. and Ecker, J.R. (2009) "Finding the fifth base: genome-wide sequencing of cytosine methylation", *Genome research,* vol. 19, no. 6, pp. 959-966.

Lister, R., Pelizzola, M., Dowen, R.H., Hawkins, R.D., Hon, G., Tonti-Filippini, J., Nery, J.R., Lee, L., Ye, Z. and Ngo, Q. (2009) "Human DNA methylomes at base resolution show widespread epigenomic differences", *Nature,* vol. 462, no. 7271, pp. 315-322.

Lobo, F.G. and Goldberg, D.E. (2004) "The parameter-less genetic algorithm in practice", *Information Sciences,* vol. 167, no. 1, pp. 217-232.

Lopez Castel, A., Nakamori, M., Tome, S., Chitayat, D., Gourdon, G., Thornton, C.A. and Pearson, C.E. (2011) "Expanded CTG repeat demarcates a boundary for abnormal CpG methylation in myotonic dystrophy patient tissues", *Human molecular genetics,* vol. 20, no. 1, pp. 1-15.

Lourenço, Helena R., Olivier C. Martin, and Thomas Stutzle. "Iterated local search." *arXiv preprint math*/0102188 (2001).

Lu, L., Lin, K., Qian, Z., Li, H., Cai, Y. and Li, Y. (2010) "Predicting DNA methylation status using word composition", .

Lucas, M.E., Crider, K.S., Powell, D.R., Kapoor-Vazirani, P. and Vertino, P.M. (2009) "Methylation-sensitive regulation of TMS1/ASC by the Ets factor, GA-binding protein-alpha", *The Journal of biological chemistry,* vol. 284, no. 22, pp. 14698-14709.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; *SIGKDD Explorations, Volume 11, Issue 1.*

Marosi, A.C., Balaton, Z. and Kacsuk, P. (2009) "GenWrapper: a generic wrapper for running legacy applications on desktop grids", *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on* IEEE, pp. 1.

Marosi, A., Kovács, J. and Kacsuk, P. (2013) "Towards a volunteer cloud system", *Future Generation Computer Systems,* vol. 29, no. 6, pp. 1442-1451.

Martinez, C.A., Barr, K., Kim, A. and Reinitz, J. (2013) "A synthetic biology approach to the development of transcriptional regulatory models and custom enhancer design", *Methods,* .

Marx, V. (2013) "Biology: The big challenges of big data", *Nature,* vol. 498, no. 7453, pp. 255-260.

McCabe, M.T., Lee, E.K. and Vertino, P.M. (2009) "A multifactorial signature of DNA sequence and polycomb binding predicts aberrant CpG island methylation", *Cancer research,* vol. 69, no. 1, pp. 282-291.

McKinnell, I.W., Ishibashi, J., Le Grand, F., Punch, V.G., Addicks, G.C., Greenblatt, J.F., Dilworth, F.J. and Rudnicki, M.A. (2008) "Pax7 activates myogenic genes by recruitment of a histone methyltransferase complex", *Nature cell biology,* vol. 10, no. 1, pp. 77-84.

Mehta, G., Deelman, E., Knowles, J.A., Chen, T., Wang, Y., Vöckler, J., Buyske, S. and Matise, T. (2012) "Enabling data and compute intensive workflows in bioinformatics", *Euro-Par 2011: Parallel Processing Workshops*Springer pp. 23.

Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Williams, A., Oinn, T. and Goble, C. (2010) "Taverna, reloaded", *Scientific and Statistical Database Management* Springer, pp. 471.

Morris, K.V., Santoso, S., Turner, A.M., Pastori, C. and Hawkins, P.G. (2008) "Bidirectional transcription directs both transcriptional gene activation and suppression in human cells", *PLoS genetics,* vol. 4, no. 11, pp. e1000258.

Mpipsykl: online (Accessed August 2013) http://www.mpipsykl.mpg.de/en/research/themes/aging/figures/spengler_02_01.jpg

Naumann, A., Hochstein, N., Weber, S., Fanning, E. and Doerfler, W. (2009) "A distinct DNA-methylation boundary in the 5'- upstream sequence of the FMR1 promoter binds nuclear proteins and is lost in fragile X syndrome", *American Journal of Human Genetics,* vol. 85, no. 5, pp. 606-616.

Nunes, C.M., Britto Jr, Alceu de S, Kaestner, C.A. and Sabourin, R. (2004) "Feature subset selection using an optimized hill climbing algorithm for handwritten character recognition" in *Structural, Syntactic, and Statistical Pattern Recognition* Springer, pp. 1018-1025.

OurGrid, (2013) *About Ourgrid*. Available at: http://www.ourgrid.org/overview.php(Accessed: July 2013).

Pautasso, C., Heinis, T. and Alonso, G. (2006) "JOpera: Autonomic Service Orchestration.", *IEEE Data Eng.Bull.,* vol. 29, no. 3, pp. 32-39.

Pook, M. (2012) " DNA Methylation and Trinucleotide Repeat Expansion Diseases" in *DNA Methylation - From Genomics **to** Technology*, eds. T. Tatarinova and O. Kerton, InTech, Janeza Trdine 9, 51000 Rijeka, Croatia, pp. 193.

Previti, C., Harari, O., Zwir, I. and del Val, C. (2009) "Profile analysis and prediction of tissue-specific CpG island methylation classes", *BMC Bioinformatics,* vol. 10, no. 1, pp. 116.

Rice, P., Longden, I. and Bleasby, A. (2000) "EMBOSS: The European Molecular Biology Open Software Suite", *Trends Genet*, vol. 16, pp. 276-277.

Ries, C.B., Schroder, C. and Grout, V. (2011) "Approach of a UML profile for Berkeley Open Infrastructure for network computing (BOINC)", *Computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on* IEEE,pp. 483.

Rowland, B.D., Bernards, R. and Peeper, D.S. (2005) "The KLF4 tumour suppressor is a transcriptional repressor of p53 that acts as a context-dependent oncogene", *Nature cell biology,* vol. 7, no. 11, pp. 1074-1082.

Rueda, L. and Vidyadharan, V. (2006) "A hill-climbing approach for automatic gridding of cDNA microarray images", *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* vol. 3, no. 1, pp. 72.

Russell, S.J. and Norvig, P. (2010) "Learning from example" in *Artificial Intelligence: A Modern Approach*, eds. S.J. Russell and P. Norvig, Prentice Hall, pp. 727-737.

Schara, U. and Schoser, B.G. (2006) "Myotonic dystrophies type 1 and 2: a summary on current aspects", *Seminars in pediatric neurology,* vol. 13, no. 2, pp. 71-79.

Schott, B. and Emmen, A. (2010) "Green Methodologies in Desktop-Grid", *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on* IEEE, pp. 671.

Sharma, S., Kelly, T.K. and Jones, P.A. (2010) "Epigenetics in cancer", *Carcinogenesis,* vol. 31, no. 1, pp. 27-36.

SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs 2013,. Available from: < http://www.shiwa-workflow.eu >. [August 2013].

Su, S., Lin, C. and Ting, C. (2011) "An effective hybrid of hill climbing and genetic algorithm for 2D triangular protein structure prediction", *Proteome science,* vol. 9, no. Suppl 1, pp. S19.

Sui, S.J.H., Fulton, D.L., Arenillas, D.J., Kwon, A.T. and Wasserman, W.W. (2007) "oPOSSUM: integrated tools for analysis of regulatory motif over-representation", *Nucleic acids research,* vol. 35, no. suppl 2, pp. W245-W252.

Taverna 2013, Market Information. Available from: < http://www.taverna.org.uk/ >. [July 2013].

Taylor, S.J., Ghorbani, M., Mustafee, N., Turner, S.J., Kiss, T., Farkas, D., Kite, S. and Straßburger, S. (2011) "Distributed computing and modeling & simulation: Speeding up simulations and creating large models", *Simulation Conference (WSC), Proceedings of the 2011 Winter* IEEE, pp. 161.

Thain, D., Tannenbaum, T. and Livny, M. (2005) "Distributed computing in practice: The Condor experience", *Concurrency and Computation: Practice and Experience,* vol. 17, no. 2-4, pp. 323-356.

Thompson, V.C., Day, T.K., Bianco-Miotto, T., Selth, L.A., Han, G., Thomas, M., Buchanan, G., Scher, H.I., Nelson, C.C. and Greenberg, N.M. (2012) "A gene signature identified using a mouse model of androgen receptor-dependent prostate cancer predicts biochemical relapse in human disease", *International Journal of Cancer*, vol. 131, no. 3, pp. 662-672.

Tiwari, A. and Sekhar, A.K. (2007) "Workflow based framework for life science informatics", *Computational Biology and Chemistry*, vol. 31, no. 5, pp. 305-319.

Tufféry, S. (2011) "Classsification and prediction method" in *Data Mining and Statistics for Decision Making*, ed. S. Tufféry, Wiley , pp. 503-503.

UCSC Browser 2013. *Table Browser* Available at: http://genome.ucsc.edu/cgi-bin/hgTables?command=start (Accessed July 2013).

Unicore Project: *UNICORE 6 Architecture* online http://www.unicore.eu/unicore/architecture.php (Accessed August 2013)

Urbah, E., Kacsuk, P., Farkas, Z., Fedak, G., Kecskemeti, G., Lodygensky, O., Marosi, A., Balaton, Z., Caillat, G. and Gombas, G. (2009) "EDGeS: bridging EGEE to BOINC and XtremWeb", *Journal of Grid Computing*, vol. 7, no. 3, pp. 335-354.

Versweyveld Leslie, Tumiatti Fabio, Sukhoroslov Oleg, Lovas Robert, Schott Bernhard, Huang Vicky, Brasileiro Francisco Vilar and Shi Xuanhua (2012) *A roadmap desktop grid for e-science* . Available at: http://desktopgridfederation.org/documents/10508/57919/RoadMapGH.pdf?version=1.6 (Accessed: 08/06 2013).

VOORSLUYS WILLIAM, BROBERG JAMES and BUYYA RAJKUMAR (2010) "introduction to cloud computing" in *Cloud Computing*, eds. R. Buyya, J. Broberg and A.M. Goscinski, Wiley, pp. 7-7.

Wang, Y., Tetko, I.V., Hall, M.A., Frank, E., Facius, A., Mayer, K.F. and Mewes, H.W. (2005) "Gene selection from microarray data for cancer classification—a machine learning approach", *Computational biology and chemistry*, vol. 29, no. 1, pp. 37-46.

Wang, T., Chen, M., Liu, L., Cheng, H., Yan, Y.E., Feng, Y.H. and Wang, H. (2011) "Nicotine induced CpG methylation of Pax6 binding motif in StAR promoter reduces the gene expression and cortisol production", *Toxicology and applied pharmacology*, vol. 257, no. 3, pp. 328-337.

White, T. (2012) Hadoop: The Definitive Guide, O'Reilly Media.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4(2), 65-85.

Wilkinson, B. (2011) "Introduction to grid computing" in *Grid Computing: Techniques and Applications* Taylor & Francis, pp. 7-7.

Witten, I.H., Frank, E. and Hall, M.A. (2011) *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques,* Elsevier Science.

Wrzodek, C., Büchel, F., Hinselmann, G., Eichner, J., Mittag, F. and Zell, A. (2012) "Linking the Epigenome to the Genome: Correlation of Different Features to DNA Methylation of CpG Islands", *PLoS ONE*, vol. 7, no. 4, pp. e35327.

Yamada, Y. and Satou, K. (2008) "Prediction of genomic methylation status on CpG islands using DNA sequence features", *WSEAS Transactions on Biology and Biomedicine*, vol. 5, no. 7, pp. 153-162.

Yuan, G. (2011) "Prediction of Epigenetic Target Sites by Using Genomic DNA Sequence" in *Handbook of Research on Computational and Systems Biology: Interdisciplinary Applications* IGI Global, pp. 187-201.

Zhang, Y., Rohde, C., Tierling, S., Jurkowski, T.P., Bock, C., Santacruz, D., Ragozin, S., Reinhardt, R., Groth, M. and Walter, J. (2009) "DNA methylation analysis of chromosome 21 gene promoters at single base pair and single allele resolution", *PLoS genetics*, vol. 5, no. 3, pp. e1000438.

Zheng, H., Wu, H., Li, J. and Jiang, S. (2013) "CpGIMethPred: computational model for predicting methylation status of CpG islands in human genome", *BMC Medical Genomics*, vol. 6, no. Suppl 1, pp. S13

Appendix A: Ranked table of the features.

This table shows the occurrence of features in all prediction experiments first column is the feature name features starting with MA are the feature related to Jasper database . rank is the number of time feature appeared in the final selected subset. Occurrence shows the proportion of time feature occurred in all experiments. Species, name, class, family is only applied for jaspscan feature and will be empty for banana btwisted and wordcount.

Feature name	**Rank**	**Occurance**	**species**	**Name**	**class**	**Family**
MA0035.2	14	0.93	Mus musculus	Gata1	Zinc-coordinating	GATA
MA0414.1	14	0.93	Saccharomyces cerevisiae	XBP1	Ig-fold	Rel
CCCC	14	0.93				
MA0307.1	13	0.87	Saccharomyces cerevisiae	GLN3	Zinc-coordinating	GATA
MA0119.1	13	0.87	Homo sapiens	TLX1::NFIC	Helix-Turn-Helix::Other	Homeo::Nuclear Factor I-CCAAT-binding
MA0171.1	13	0.87	Drosophila melanogaster	CG11085	Helix-Turn-Helix	Homeo
MA0117.1	13	0.87	Rattus norvegicus	Mafb	Zipper-Type	Leucine Zipper
MA0255.1	13	0.87	Drosophila melanogaster	z	Helix-Turn-Helix	Zeste
MA0250.1	13	0.87	Drosophila melanogaster	unc-4	Helix-Turn-Helix	Homeo
MA0076.1	13	0.87	Homo	ELK4	Winged	Ets

Feature name	Rank	Occurance	species	Name	class	Family
			sapiens		Helix-Turn-Helix	
MA0128.1	13	0.87	Triticum aestivum	EmBP-1	Zipper-Type	Leucine Zipper
MA0229.1	13	0.87	Drosophila melanogaster	inv	Helix-Turn-Helix	Homeo
MA0427.1	13	0.87	Saccharomyces cerevisiae	YJL103C	Zinc-coordinating	Fungal Zn cluster
MA0048.1	13	0.87	Homo sapiens	NHLH1	Zipper-Type	Helix-Loop-Helix
MA0121.1	13	0.87	Arabidopsis thaliana	ARR10	Helix-Turn-Helix	Myb
MA0083.1	13	0.87	Homo sapiens	SRF	Other Alpha-Helix	MADS
MA0287.1	13	0.87	Saccharomyces cerevisiae	CUP2	Zinc-coordinating	Copper fist
MA0075.1	13	0.87	Mus musculus	Prrx2	Helix-Turn-Helix	Homeo
ACCT	13	0.87				
ATGC	13	0.87				
CTCA	13	0.87				
GCAG	13	0.87				
GCCT	13	0.87				
TGCC	13	0.87				
MA0248.1	12	0.80	Drosophila melanogaster	tup	Helix-Turn-Helix	Homeo
AACT	12	0.80				
MA0010.1	12	0.80	Drosophila melanogaster	br_Z1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0297.1	12	0.80	Saccharomyces cerevisiae	FKH2	Winged Helix-Turn-Helix	Forkhead
MA0275.1	12	0.80	Saccharomyces cerevisiae	ASG1	Zinc-coordinating	Fungal Zn cluster

Feature name	Rank	Occurance	species	Name	class	Family
MA0077.1	12	0.80	Homo sapiens	SOX9	Other Alpha-Helix	High Mobility Group
MA0346.1	12	0.80	Saccharomyces cerevisiae	NHP6B	Other Alpha-Helix	High Mobility Group
MA0219.1	12	0.80	Drosophila melanogaster	ems	Helix-Turn-Helix	Homeo
MA0413.1	12	0.80	Saccharomyces cerevisiae	USV1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0244.1	12	0.80	Drosophila melanogaster	slbo	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0179.1	12	0.80	Drosophila melanogaster	CG32532	Helix-Turn-Helix	Homeo
MA0245.1	12	0.80	Drosophila melanogaster	slou	Helix-Turn-Helix	Homeo
MA0353.1	12	0.80	Saccharomyces cerevisiae	PDR3	Zinc-coordinating	Fungal Zn cluster
MA0266.1	12	0.80	Saccharomyces cerevisiae	ABF2	Other Alpha-Helix	High Mobility Group
MA0150.1	12	0.80	Homo sapiens	NFE2L2	Zipper-Type	Leucine Zipper
MA0440.1	12	0.80	Saccharomyces cerevisiae	ZAP1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0213.1	12	0.80	Drosophila melanogaster	brk	Helix-Turn-Helix	Brinker
MA0348.1	12	0.80	Saccharomyces cerevisiae	OAF1	Zinc-coordinating	Fungal Zn cluster
MA0178.1	12	0.80	Drosophila melanogaster	CG32105	Helix-Turn-Helix	Homeo
MA0089.1	12	0.80	Gallus	NFE2	Zipper-	Leucine Zipper

Feature name	Rank	Occurance	species	Name	class	Family
			gallus	L1::MafG	Type	
MA0159.1	12	0.80	Homo sapiens	RXR::RAR_DR5	Zinc-coordinating	Hormone-nuclear Receptor
ACAT	12	0.80				
ACCC	12	0.80				
AGAA	12	0.80				
AGCG	12	0.80				
ATAC	12	0.80				
CGAT	12	0.80				
CTTA	12	0.80				
GAGT	12	0.80				
GATT	12	0.80				
GTGA	12	0.80				
GTTG	12	0.80				
MA0112.1	12	0.80	Homo sapiens ,Mus musculus ,Rattus norvegicus ,Gallus gallus ,Xenopus laevis ,Xenopus (Silurana) tropicalis ,Bos taurus ,Oryctolagus cuniculus	ESR1	Zinc-coordinating	Hormone-nuclear Receptor
TCAG	12	0.80				
TCGC	12	0.80				
TGAC	12	0.80				
TTCG	12	0.80				
AAAT	11	0.73				
MA0137.2	11	0.73	Homo sapiens	STAT1	Ig-fold	Stat

Feature name	Rank	Occurance	species	Name	class	Family
MA0438.1	11	0.73	Saccharomyces cerevisiae	YRM1	Zinc-coordinating	Fungal Zn cluster
MA0230.1	11	0.73	Drosophila melanogaster	lab	Helix-Turn-Helix	Homeo
MA0049.1	11	0.73	Drosophila melanogaster	hb	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0109.1	11	0.73	Oryctolagus cuniculus	Hltf	Zinc-coordinating	GATA
MA0384.1	11	0.73	Saccharomyces cerevisiae	SNT2	Helix-Turn-Helix	Myb
MA0442.1	11	0.73	Mus musculus	SOX10	Other Alpha-Helix	High Mobility Group
MA0302.1	11	0.73	Saccharomyces cerevisiae	GAT4	Zinc-coordinating	GATA
MA0003.1	11	0.73	Homo sapiens	TFAP2A	Zipper-Type	Helix-Loop-Helix
MA0294.1	11	0.73	Saccharomyces cerevisiae	EDS1	Zinc-coordinating	Fungal Zn cluster
MA0043.1	11	0.73	Homo sapiens	HLF	Zipper-Type	Leucine Zipper
MA0007.1	11	0.73	Rattus rattus	Ar	Zinc-coordinating	Hormone-nuclear Receptor
MA0214.1	11	0.73	Drosophila melanogaster	bsh	Helix-Turn-Helix	Homeo
MA0021.1	11	0.73	Zea mays	Dof3	Zinc-coordinating	Dof
MA0138.2	11	0.73	Homo sapiens	REST	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0233.1	11	0.73	Drosophila melanogaster	mirr	Helix-Turn-Helix	Homeo
MA0130.1	11	0.73	Homo sapiens	ZNF354C	Zinc-coordinatin	BetaBetaAlpha-zinc finger

Feature name	Rank	Occurance	species	Name	class	Family
					g	
MA0406.1	11	0.73	Saccharomyces cerevisiae	TEC1	Helix-Turn-Helix	Homeo
MA0227.1	11	0.73	Drosophila melanogaster	hth	Helix-Turn-Helix	Homeo
MA0152.1	11	0.73	Mus musculus	NFATC2	Ig-fold	Rel
MA0329.1	11	0.73	Saccharomyces cerevisiae	MBP1	Ig-fold	Rel
MA0443.1	11	0.73	Drosophila melanogaster	btd	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0148.1	11	0.73	Homo sapiens	FOXA1	Winged Helix-Turn-Helix	Forkhead
MA0351.1	11	0.73	Saccharomyces cerevisiae	DOT6	Helix-Turn-Helix	Myb
MA0232.1	11	0.73	Drosophila melanogaster	lbl	Helix-Turn-Helix	Homeo
AGTT	11	0.73				
ATCC	11	0.73				
CACT	11	0.73				
CAGC	11	0.73				
CATG	11	0.73				
CGAA	11	0.73				
CGCC	11	0.73				
CGGG	11	0.73				
CGTT	11	0.73				
GACT	11	0.73				
GCGC	11	0.73				
GCTC	11	0.73				
GCTG	11	0.73				
GGCT	11	0.73				
GGGT	11	0.73				
GTCC	11	0.73				
GTGC	11	0.73				
GTTC	11	0.73				
TACC	11	0.73				

Feature name	Rank	Occurance	species	Name	class	Family
TCCG	11	0.73				
AACA	10	0.67				
AAGA	10	0.67				
AAGT	10	0.67				
ACAG	10	0.67				
ACCA	10	0.67				
MA0073.1	10	0.67	Homo sapiens	RREB1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0094.2	10	0.67	Drosophila melanogaster	Ubx	Helix-Turn-Helix	Homeo
MA0444.1	10	0.67	Drosophila melanogaster	CG34031	Helix-Turn-Helix	Homeo
MA0202.1	10	0.67	Drosophila melanogaster	Rx	Helix-Turn-Helix	Homeo
MA0408.1	10	0.67	Saccharomyces cerevisiae	TOS8	Helix-Turn-Helix	Homeo
MA0432.1	10	0.67	Saccharomyces cerevisiae	YNR063W	Zinc-coordinating	Fungal Zn cluster
MA0358.1	10	0.67	Saccharomyces cerevisiae	PUT3	Zinc-coordinating	Fungal Zn cluster
MA0320.1	10	0.67	Saccharomyces cerevisiae	IME1	Other	Other
MA0305.1	10	0.67	Saccharomyces cerevisiae	GCR2	Other	Other
MA0270.1	10	0.67	Saccharomyces cerevisiae	AFT2	Other	Other
MA0453.1	10	0.67	Drosophila melanogaster	nub	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0293.1	10	0.67	Saccharomyces cerevisiae	ECM23	Zinc-coordinating	GATA

Feature name	**Rank**	**Occurance**	**species**	**Name**	**class**	**Family**
MA0415.1	10	0.67	Saccharomyces cerevisiae	YAP1	Zipper-Type	Leucine Zipper
MA0381.1	10	0.67	Saccharomyces cerevisiae	SKN7	Winged Helix-Turn-Helix	E2F
MA0288.1	10	0.67	Saccharomyces cerevisiae	CUP9	Helix-Turn-Helix	Homeo
MA0424.1	10	0.67	Saccharomyces cerevisiae	YER184C	Zinc-coordinating	Fungal Zn cluster
MA0011.1	10	0.67	Drosophila melanogaster	br_Z2	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0201.1	10	0.67	Drosophila melanogaster	Ptx1	Helix-Turn-Helix	Homeo
MA0246.1	10	0.67	Drosophila melanogaster	so	Helix-Turn-Helix	Homeo
MA0390.1	10	0.67	Saccharomyces cerevisiae	STB3	Other	Other
MA0208.1	10	0.67	Drosophila melanogaster	al	Helix-Turn-Helix	Homeo
MA0388.1	10	0.67	Saccharomyces cerevisiae	SPT23	Other	Other
MA0217.1	10	0.67	Drosophila melanogaster	caup	Helix-Turn-Helix	Homeo
MA0342.1	10	0.67	Saccharomyces cerevisiae	MSN4	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0272.1	10	0.67	Saccharomyces cerevisiae	ARG81	Zinc-coordinating	Fungal Zn cluster
MA0199.1	10	0.67	Drosophila	Optix	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
			melanogaster			
MA0460.1	10	0.67	Drosophila melanogaster	ttk	Other	AT-hook
MA0095.1	10	0.67	Homo sapiens	YY1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0347.1	10	0.67	Saccharomyces cerevisiae	NRG1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0037.1	10	0.67	Homo sapiens	GATA3	Zinc-coordinating	GATA
MA0065.2	10	0.67	Mus musculus	PPARG::RXRA	Zinc-coordinating	Hormone-nuclear Receptor
MA0215.1	10	0.67	Drosophila melanogaster	btn	Helix-Turn-Helix	Homeo
MA0449.1	10	0.67	Drosophila melanogaster	h	Zipper-type	Helix-Loop-Helix
MA0402.1	10	0.67	Saccharomyces cerevisiae	SWI5	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0357.1	10	0.67	Saccharomyces cerevisiae	PHO4	Zipper-Type	Helix-Loop-Helix
MA0435.1	10	0.67	Saccharomyces cerevisiae	YPR015C	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0254.1	10	0.67	Drosophila melanogaster	vvl	Helix-Turn-Helix	Homeo
MA0336.1	10	0.67	Saccharomyces cerevisiae	MGA1	Winged Helix-Turn-Helix	E2F
MA0373.1	10	0.67	Saccharomyces cerevisiae	RPN4	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0165.1	10	0.67	Drosophila melanog	Abd-B	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
			aster			
MA0085.1	10	0.67	Drosophila melanogaster	Su(H)	Other	LAG1
MA0457.1	10	0.67	Drosophila melanogaster	PHDP	Helix-Turn-Helix	Homeo
MA0194.1	10	0.67	Drosophila melanogaster	Lim1	Helix-Turn-Helix	Homeo
MA0389.1	10	0.67	Saccharomyces cerevisiae	SRD1	Zinc-coordinating	GATA
MA0252.1	10	0.67	Drosophila melanogaster	vis	Helix-Turn-Helix	Homeo
ATAA	10	0.67				
CAAG	10	0.67				
CACA	10	0.67				
CCTC	10	0.67				
CGCG	10	0.67				
CTAT	10	0.67				
GACG	10	0.67				
GCCC	10	0.67				
GCTT	10	0.67				
GGCC	10	0.67				
GTAC	10	0.67				
TAAA	10	0.67				
TCAT	10	0.67				
TCTC	10	0.67				
TGTA	10	0.67				
TTAA	10	0.67				
TTTG	10	0.67				
ACCG	9	0.60				
ACGG	9	0.60				
ACGT	9	0.60				
MA0239.1	9	0.60	Drosophila melanogaster	prd	Helix-Turn-Helix	Homeo
MA0093.1	9	0.60	Homo sapiens	USF1	Zipper-Type	Helix-Loop-Helix
MA0322.1	9	0.60	Saccharomyces cerevisi	INO4	Zipper-Type	Helix-Loop-Helix

Feature name	Rank	Occurance	species	Name	class	Family
			ae			
MA0387.1	9	0.60	Saccharomyces cerevisiae	SPT2	Other Alpha-Helix	High Mobility Group
MA0405.1	9	0.60	Saccharomyces cerevisiae	TEA1	Zinc-coordinating	Fungal Zn cluster
MA0207.1	9	0.60	Drosophila melanogaster	achi	Helix-Turn-Helix	Homeo
MA0260.1	9	0.60	Caenorhabditis elegans	che-1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0188.1	9	0.60	Drosophila melanogaster	Dr	Helix-Turn-Helix	Homeo
MA0314.1	9	0.60	Saccharomyces cerevisiae	HAP3	Other Alpha-Helix	NFY CCAAT-binding
MA0283.1	9	0.60	Saccharomyces cerevisiae	CHA4	Zinc-coordinating	Fungal Zn cluster
MA0168.1	9	0.60	Drosophila melanogaster	B-H1	Helix-Turn-Helix	Homeo
MA0086.1	9	0.60	Drosophila melanogaster	sna	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0315.1	9	0.60	Saccharomyces cerevisiae	HAP4	Other Alpha-Helix	NFY CCAAT-binding
MA0140.1	9	0.60	Mus musculus	Tal1::Gata1	Zipper-Type	Helix-Loop-Helix
MA0403.1	9	0.60	Saccharomyces cerevisiae	TBF1	Helix-Turn-Helix	Myb
MA0316.1	9	0.60	Saccharomyces cerevisiae	HAP5	Other Alpha-Helix	NFY CCAAT-binding
MA0063.1	9	0.60	Mus musculus	Nkx2-5	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
MA0323.1	9	0.60	Saccharomyces cerevisiae	IXR1	Other Alpha-Helix	High Mobility Group
MA0180.1	9	0.60	Drosophila melanogaster	Vsx2	Helix-Turn-Helix	Homeo
MA0024.1	9	0.60	Homo sapiens	E2F1	Winged Helix-Turn-Helix	E2F
MA0421.1	9	0.60	Saccharomyces cerevisiae	YDR026C	Helix-Turn-Helix	Myb
MA0115.1	9	0.60	Homo sapiens	NR1H2::RXRA	Zinc-coordinating	Hormone-nuclear Receptor
MA0031.1	9	0.60	Homo sapiens	FOXD1	Winged Helix-Turn-Helix	Forkhead
MA0069.1	9	0.60	Homo sapiens	Pax6	Helix-Turn-Helix	Homeo
MA0434.1	9	0.60	Saccharomyces cerevisiae	YPR013C	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0301.1	9	0.60	Saccharomyces cerevisiae	GAT3	Zinc-coordinating	GATA
MA0448.1	9	0.60	Drosophila melanogaster	H2.0	Helix-Turn-Helix	Homeo
MA0344.1	9	0.60	Saccharomyces cerevisiae	NHP10	Other Alpha-Helix	High Mobility Group
MA0008.1	9	0.60	Arabidopsis thaliana	HAT5	Helix-Turn-Helix	Homeo
MA0361.1	9	0.60	Saccharomyces cerevisiae	RDS1	Zinc-coordinating	Fungal Zn cluster
MA0185.1	9	0.60	Drosophila melanogaster	Deaf1	Other Alpha-Helix	Sand
MA0355.1	9	0.60	Saccharomyces cerevisi	PHD1	Other	KilA-N

Feature name	Rank	Occurance	species	Name	class	Family
			ae			
MA0274.1	9	0.60	Saccharomyces cerevisiae	ARR1	Zipper-Type	Leucine Zipper
MA0335.1	9	0.60	Saccharomyces cerevisiae	MET4	Zipper-Type	Leucine Zipper
MA0354.1	9	0.60	Saccharomyces cerevisiae	PDR8	Zinc-coordinating	Fungal Zn cluster
MA0027.1	9	0.60	Mus musculus	En1	Helix-Turn-Helix	Homeo
MA0167.1	9	0.60	Drosophila melanogaster	Awh	Helix-Turn-Helix	Homeo
MA0020.1	9	0.60	Zea mays	Dof2	Zinc-coordinating	Dof
CAAT	9	0.60				
CAGG	9	0.60				
CATC	9	0.60				
CCAC	9	0.60				
CCCA	9	0.60				
CCTT	9	0.60				
CGAC	9	0.60				
GAAA	9	0.60				
GAAC	9	0.60				
GATG	9	0.60				
GCAT	9	0.60				
GCCG	9	0.60				
GGTT	9	0.60				
GTGG	9	0.60				
MA0080.1	9	0.60	Homo sapiens	SPI1	Winged Helix-Turn-Helix	Ets
TAGC	9	0.60				
TATT	9	0.60				
TCAC	9	0.60				
TCCA	9	0.60				
TCCC	9	0.60				
TCCT	9	0.60				
TGTG	9	0.60				
TTAG	9	0.60				
TTGT	9	0.60				

Feature name	Rank	Occurance	species	Name	class	Family
AAAA	8	0.53				
AGAG	8	0.53				
AGGC	8	0.53				
AGTC	8	0.53				
ATAT	8	0.53				
ATCA	8	0.53				
ATTT	8	0.53				
CACC	8	0.53				
CAGA	8	0.53				
CCAT	8	0.53				
MA0105.1	8	0.53	Homo sapiens	NFKB1	Ig-fold	Rel
MA0060.1	8	0.53	Homo sapiens	NFYA	Other Alpha-Helix	NFY CCAAT-binding
MA0333.1	8	0.53	Saccharomyces cerevisiae	MET31	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0446.1	8	0.53	Drosophila melanogaster	fkh	Winged Helix-Turn-Helix	Forkhead
MA0407.1	8	0.53	Saccharomyces cerevisiae	THI2	Zinc-coordinating	Fungal Zn cluster
MA0280.1	8	0.53	Saccharomyces cerevisiae	CAT8	Zinc-coordinating	Fungal Zn cluster
MA0057.1	8	0.53	Homo sapiens	MZF1_5-13	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0223.1	8	0.53	Drosophila melanogaster	exex	Helix-Turn-Helix	Homeo
MA0216.1	8	0.53	Drosophila melanogaster	cad	Helix-Turn-Helix	Homeo
MA0452.1	8	0.53	Drosophila melanogaster	Kr	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0157.1	8	0.53	Mus musculus	FOXO3	Winged Helix-Turn-Helix	Forkhead
MA0379.1	8	0.53	Saccharomyces cerevisi	SIG1	Zinc-coordinating	BetaBetaAlpha-zinc finger

Feature name	Rank	Occurance	species	Name	class	Family
			ae			
MA0439.1	8	0.53	Saccharomyces cerevisiae	YRR1	Zinc-coordinating	Fungal Zn cluster
MA0277.1	8	0.53	Saccharomyces cerevisiae	AZF1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0002.2	8	0.53	Mus musculus	RUNX1	Ig-fold	Runt
MA0281.1	8	0.53	Saccharomyces cerevisiae	CBF1	Zipper-Type	Helix-Loop-Helix
MA0210.1	8	0.53	Drosophila melanogaster	ara	Helix-Turn-Helix	Homeo
MA0218.1	8	0.53	Drosophila melanogaster	ct	Helix-Turn-Helix	Homeo
CCGC	8	0.53				
MA0047.2	8	0.53	Mus musculus	Foxa2	Winged Helix-Turn-Helix	Forkhead
MA0340.1	8	0.53	Saccharomyces cerevisiae	MOT3	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0247.1	8	0.53	Drosophila melanogaster	tin	Helix-Turn-Helix	Homeo
MA0147.1	8	0.53	Mus musculus	Myc	Zipper-Type	Helix-Loop-Helix
MA0264.1	8	0.53	Caenorhabditis elegans	ceh-22	Helix-Turn-Helix	Homeo
MA0238.1	8	0.53	Drosophila melanogaster	pb	Helix-Turn-Helix	Homeo
MA0241.1	8	0.53	Drosophila melanogaster	ro	Helix-Turn-Helix	Homeo
MA0265.1	8	0.53	Saccharomyces cerevisiae	ABF1	Zinc-coordinating	BetaBetaAlpha-zinc finger

Feature name	Rank	Occurance	species	Name	class	Family
MA0116.1	8	0.53	Rattus norvegicus	Zfp423	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0118.1	8	0.53	Halocynthia roretzi	Macho-1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0151.1	8	0.53	Mus musculus	ARID3A	Helix-Turn-Helix	Arid
CCGT	8	0.53				
MA0285.1	8	0.53	Saccharomyces cerevisiae	CRZ1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0211.1	8	0.53	Drosophila melanogaster	bap	Helix-Turn-Helix	Homeo
MA0026.1	8	0.53	Drosophila melanogaster	Eip74EF	Winged Helix-Turn-Helix	Ets
MA0356.1	8	0.53	Saccharomyces cerevisiae	PHO2	Helix-Turn-Helix	Homeo
MA0235.1	8	0.53	Drosophila melanogaster	onecut	Helix-Turn-Helix	Homeo
MA0222.1	8	0.53	Drosophila melanogaster	exd	Helix-Turn-Helix	Homeo
MA0426.1	8	0.53	Saccharomyces cerevisiae	YHP1	Helix-Turn-Helix	Homeo
MA0092.1	8	0.53	Mus musculus	Hand1::Tcfe2a	Zipper-Type	Helix-Loop-Helix
MA0172.1	8	0.53	Drosophila melanogaster	CG11294	Helix-Turn-Helix	Homeo
MA0404.1	8	0.53	Saccharomyces cerevisiae	TBS1	Zinc-coordinating	Fungal Zn cluster
MA0183.1	8	0.53	Drosophila melanogaster	CG7056	Helix-Turn-Helix	Homeo
MA0401.1	8	0.53	Sacchar	SWI4	Ig-fold	Rel

Feature name	Rank	Occurance	species	Name	class	Family
			omyces cerevisiae			
MA0112.2	8	0.53	Homo sapiens	ESR1	Zinc-coordinating	Hormone-nuclear Receptor
MA0385.1	8	0.53	Saccharomyces cerevisiae	SOK2	Other	KilA-N
MA0326.1	8	0.53	Saccharomyces cerevisiae	MAC1	Zinc-coordinating	Copper fist
MA0352.1	8	0.53	Saccharomyces cerevisiae	PDR1	Zinc-coordinating	Fungal Zn cluster
MA0370.1	8	0.53	Saccharomyces cerevisiae	RME1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0256.1	8	0.53	Drosophila melanogaster	zen	Helix-Turn-Helix	Homeo
MA0204.1	8	0.53	Drosophila melanogaster	Six4	Helix-Turn-Helix	Homeo
MA0028.1	8	0.53	Homo sapiens	ELK1	Winged Helix-Turn-Helix	Ets
CCTG	8	0.53				
CGGC	8	0.53				
CTAG	8	0.53				
CTCG	8	0.53				
CTCT	8	0.53				
CTGG	8	0.53				
CTTT	8	0.53				
GCAC	8	0.53				
GGAA	8	0.53				
GGAT	8	0.53				
GGGA	8	0.53				
GTCG	8	0.53				
GTGT	8	0.53				
MA0104.1	8	0.53	Mus musculus	Mycn	Zipper-type	Helix-Loop-Helix
MA0108.2	8	0.53	Vertebrata	TBP	Beta-sheet	TATA-binding

Feature name	Rank	Occurance	species	Name	class	Family
MA0114.1	8	0.53	Vertebrata	HNF4A	Zinc-coordinating	Hormone-nuclear Receptor
TAAG	8	0.53				
TCAA	8	0.53				
TGGA	8	0.53				
TGGG	8	0.53				
TTGC	8	0.53				
AAGC	7	0.47				
ACAA	7	0.47				
ACAC	7	0.47				
ACTC	7	0.47				
AGCA	7	0.47				
AGTA	7	0.47				
ATGG	7	0.47				
ATGT	7	0.47				
ATTC	7	0.47				
banana_bend_structure	7	0.47				
CAAC	7	0.47				
CCAG	7	0.47				
CGGA	7	0.47				
CGGT	7	0.47				
CGTA	7	0.47				
CTCC	7	0.47				
CTGC	7	0.47				
MA0360.1	7	0.47	Saccharomyces cerevisiae	RDR1	Zinc-coordinating	Fungal Zn cluster
MA0321.1	7	0.47	Saccharomyces cerevisiae	INO2	Zipper-Type	Helix-Loop-Helix
MA0430.1	7	0.47	Saccharomyces cerevisiae	YLR278C	Zinc-coordinating	Fungal Zn cluster
MA0437.1	7	0.47	Saccharomyces cerevisiae	YPR196W	Zinc-coordinating	Fungal Zn cluster
MA0331.1	7	0.47	Saccharomyces cerevisiae	MCM1	Other Alpha-Helix	MADS
MA0191.1	7	0.47	Drosophila melanogaster	HGTX	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
MA0166.1	7	0.47	Drosophila melanogaster	Antp	Helix-Turn-Helix	Homeo
MA0253.1	7	0.47	Drosophila melanogaster	vnd	Helix-Turn-Helix	Homeo
MA0161.1	7	0.47	Homo sapiens	NFIC	Other	NFI CCAAT-binding
MA0412.1	7	0.47	Saccharomyces cerevisiae	UME6	Zinc-coordinating	Fungal Zn cluster
MA0009.1	7	0.47	Mus musculus	T	Beta-Hairpin-Ribbon	T
MA0006.1	7	0.47	Mus musculus	Arnt::Ahr	Zipper-Type	Helix-Loop-Helix
MA0423.1	7	0.47	Saccharomyces cerevisiae	YER130C	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0378.1	7	0.47	Saccharomyces cerevisiae	SFP1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0139.1	7	0.47	Mus musculus	CTCF	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0058.1	7	0.47	Homo sapiens	MAX	Zipper-Type	Helix-Loop-Helix
MA0362.1	7	0.47	Saccharomyces cerevisiae	RDS2	Zinc-coordinating	Fungal Zn cluster
MA0396.1	7	0.47	Saccharomyces cerevisiae	STP3	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0045.1	7	0.47	Pisum sativum	HMG-I/Y	Other Alpha-Helix	High Mobility Group
MA0251.1	7	0.47	Drosophila melanogaster	unpg	Helix-Turn-Helix	Homeo
MA0429.1	7	0.47	Saccharomyces cerevisiae	YLL054C	Zinc-coordinating	Fungal Zn cluster
MA0056.1	7	0.47	Homo sapiens	MZF1_1-4	Zinc-coordinatin	BetaBetaAlpha-zinc finger

Feature name	**Rank**	**Occurance**	**species**	**Name**	**class**	**Family**
					g	
MA0392.1	7	0.47	Saccharomyces cerevisiae	STB5	Zinc-coordinating	Fungal Zn cluster
MA0276.1	7	0.47	Saccharomyces cerevisiae	ASH1	Zinc-coordinating	GATA
MA0279.1	7	0.47	Saccharomyces cerevisiae	CAD1	Zipper-Type	Leucine Zipper
MA0298.1	7	0.47	Saccharomyces cerevisiae	FZF1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0278.1	7	0.47	Saccharomyces cerevisiae	BAS1	Helix-Turn-Helix	Myb
MA0197.1	7	0.47	Drosophila melanogaster	Oct	Helix-Turn-Helix	Homeo
MA0454.1	7	0.47	Drosophila melanogaster	odd	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0051.1	7	0.47	Homo sapiens	IRF2	Winged Helix-Turn-Helix	IRF
MA0258.1	7	0.47	Homo sapiens	ESR2	Zinc-coordinating	Hormone-nuclear Receptor
MA0309.1	7	0.47	Saccharomyces cerevisiae	GZF3	Zinc-coordinating	GATA
MA0332.1	7	0.47	Saccharomyces cerevisiae	MET28	Zipper-Type	Leucine Zipper
MA0162.1	7	0.47	Mus musculus	Egr1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0371.1	7	0.47	Saccharomyces cerevisiae	ROX1	Other Alpha-Helix	High Mobility Group
MA0099.2	7	0.47	Homo sapiens	AP1	Zipper-Type	Leucine Zipper
MA0136.1	7	0.47	Mus musculu	ELF5	Winged Helix-	Ets

Feature name	Rank	Occurance	species	Name	class	Family
			s		Turn-Helix	
MA0262.1	7	0.47	Caenorhabditis elegans	mab-3	Zinc-coordinating	DM
MA0296.1	7	0.47	Saccharomyces cerevisiae	FKH1	Winged Helix-Turn-Helix	Forkhead
MA0393.1	7	0.47	Saccharomyces cerevisiae	STE12	Helix-Turn-Helix	Homeo
MA0450.1	7	0.47	Drosophila melanogaster	hkb	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0198.1	7	0.47	Drosophila melanogaster	OdsH	Helix-Turn-Helix	Homeo
MA0410.1	7	0.47	Saccharomyces cerevisiae	UGA3	Zinc-coordinating	Fungal Zn cluster
MA0226.1	7	0.47	Drosophila melanogaster	hbn	Helix-Turn-Helix	Homeo
MA0015.1	7	0.47	Drosophila melanogaster	Cf2_II	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0419.1	7	0.47	Saccharomyces cerevisiae	YAP7	Zipper-Type	Leucine Zipper
MA0334.1	7	0.47	Saccharomyces cerevisiae	MET32	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0044.1	7	0.47	Pisum sativum	HMG-1	Other Alpha-Helix	High Mobility Group
MA0160.1	7	0.47	Mus musculus	NR4A2	Zinc-coordinating	Hormone-nuclear Receptor
MA0088.1	7	0.47	Xenopus laevis	znf143	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0144.1	7	0.47	Mus musculus	Stat3	Ig-fold	Stat
CTTC	7	0.47				

Feature name	Rank	Occurance	species	Name	class	Family
GACA	7	0.47				
GCGG	7	0.47				
GGAG	7	0.47				
GTCT	7	0.47				
MA0039.1	7	0.47	Mus musculus	Klf4	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0062.1	7	0.47	Homo sapiens	GABPA	Winged Helix-Turn-Helix	Ets
TAAC	7	0.47				
TATG	7	0.47				
TCGT	7	0.47				
TGAA	7	0.47				
TGTC	7	0.47				
TTGA	7	0.47				
TTTA	7	0.47				
AAAC	6	0.40				
AAAG	6	0.40				
AATC	6	0.40				
ACTT	6	0.40				
AGAC	6	0.40				
AGCT	6	0.40				
CAGT	6	0.40				
CATA	6	0.40				
CCAA	6	0.40				
CGAG	6	0.40				
CGCT	6	0.40				
CGTC	6	0.40				
CGTG	6	0.40				
CTAC	6	0.40				
GAAG	6	0.40				
GAGC	6	0.40				
GCAA	6	0.40				
GCCA	6	0.40				
GCGA	6	0.40				
MA0324.1	6	0.40	Saccharomyces cerevisiae	LEU3	Zinc-coordinating	Fungal Zn cluster
MA0107.1	6	0.40	Homo sapiens	RELA	Ig-fold	Rel
MA0395.1	6	0.40	Saccharomyces cerevisiae	STP2	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0014.1	6	0.40	Mus musculu	Pax5	Helix-Turn-Helix	Homeo

Feature name	**Rank**	**Occurance**	**species**	**Name**	**class**	**Family**
			s			
MA0292.1	6	0.40	Saccharomyces cerevisiae	ECM22	Zinc-coordinating	Fungal Zn cluster
MA0374.1	6	0.40	Saccharomyces cerevisiae	RSC3	Zinc-coordinating	Fungal Zn cluster
MA0096.1	6	0.40	Antirrhinum majus	bZIP910	Zipper-Type	Leucine Zipper
MA0177.1	6	0.40	Drosophila melanogaster	CG18599	Helix-Turn-Helix	Homeo
MA0228.1	6	0.40	Drosophila melanogaster	ind	Helix-Turn-Helix	Homeo
MA0033.1	6	0.40	Homo sapiens	FOXL1	Winged Helix-Turn-Helix	Forkhead
MA0035.2	6	0.40	Mus musculus	Gata1	Zinc-coordinating	GATA
MA0023.1	6	0.40	Drosophila melanogaster	dl_2	Ig-fold	Rel
MA0054.1	6	0.40	Petunia x hybrida	myb.Ph3	Helix-Turn-Helix	Myb
MA0380.1	6	0.40	Saccharomyces cerevisiae	SIP4	Zinc-coordinating	Fungal Zn cluster
MA0124.1	6	0.40	Homo sapiens	NKX3-1	Helix-Turn-Helix	Homeo
MA0163.1	6	0.40	Homo sapiens	PLAG1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0133.1	6	0.40	Homo sapiens	BRCA1	Other	Other
MA0308.1	6	0.40	Saccharomyces cerevisiae	GSM1	Zinc-coordinating	Fungal Zn cluster
MA0164.1	6	0.40	Mus musculus	Nr2e3	Zinc-coordinating	Hormone-nuclear Receptor
MA0041.1	6	0.40	Rattus norvegic	Foxd3	Winged Helix-	Forkhead

Feature name	Rank	Occurance	species	Name	class	Family
			us		Turn-Helix	
MA0397.1	6	0.40	Saccharomyces cerevisiae	STP4	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0365.1	6	0.40	Saccharomyces cerevisiae	RFX1	Winged Helix-Turn-Helix	RFX
MA0046.1	6	0.40	Vertebrata	HNF1A	Helix-Turn-Helix	Homeo
MA0349.1	6	0.40	Saccharomyces cerevisiae	OPI1	Zipper-Type	Leucine Zipper
MA0441.1	6	0.40	Saccharomyces cerevisiae	ZMS1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0113.1	6	0.40	Homo sapiens	NR3C1	Zinc-coordinating	Hormone-nuclear Receptor
MA0064.1	6	0.40	Zea mays	PBF	Zinc-coordinating	Dof
MA0029.1	6	0.40	Mus musculus	Evi1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0318.1	6	0.40	Saccharomyces cerevisiae	HMRA2	Helix-Turn-Helix	Homeo
MA0132.1	6	0.40	Mus musculus	Pdx1	Helix-Turn-Helix	Homeo
MA0090.1	6	0.40	Homo sapiens	TEAD1	Helix-Turn-Helix	Homeo
MA0350.1	6	0.40	Saccharomyces cerevisiae	TOD6	Helix-Turn-Helix	Myb
MA0234.1	6	0.40	Drosophila melanogaster	oc	Helix-Turn-Helix	Homeo
MA0098.1	6	0.40	Homo sapiens	ETS1	Winged Helix-Turn-Helix	Ets
MA0271.1	6	0.40	Saccharomyces cerevisiae	ARG80	Other Alpha-Helix	MADS
MA0142.1	6	0.40	Mus musculu	Pou5f1	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
			s			
MA0145.1	6	0.40	Mus musculus	Tcfcp2l1	Other	CP2
MA0399.1	6	0.40	Saccharomyces cerevisiae	SUT1	Zinc-coordinating	Fungal Zn cluster
MA0236.1	6	0.40	Drosophila melanogaster	otp	Helix-Turn-Helix	Homeo
MA0126.1	6	0.40	Drosophila melanogaster	ovo	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0182.1	6	0.40	Drosophila melanogaster	CG4328	Helix-Turn-Helix	Homeo
MA0004.1	6	0.40	Mus musculus	Arnt	Zipper-Type	Helix-Loop-Helix
MA0205.1	6	0.40	Drosophila melanogaster	Trl	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0263.1	6	0.40	Caenorhabditis elegans	ttx-3::ceh-10	Helix-Turn-Helix	Homeo
MA0240.1	6	0.40	Drosophila melanogaster	repo	Helix-Turn-Helix	Homeo
MA0420.1	6	0.40	Saccharomyces cerevisiae	YBR239C	Zinc-coordinating	Fungal Zn cluster
MA0428.1	6	0.40	Saccharomyces cerevisiae	YKL222C	Zinc-coordinating	Fungal Zn cluster
MA0154.1	6	0.40	Mus musculus	EBF1	Zipper-Type	Helix-Loop-Helix
MA0339.1	6	0.40	Saccharomyces cerevisiae	MIG3	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0291.1	6	0.40	Saccharomyces cerevisiae	DAL82	Other	Other
MA0394.1	6	0.40	Sacchar	STP1	Zinc-	BetaBetaAlpha-

Feature name	Rank	Occurance	species	Name	class	Family
			omyces cerevisiae		coordinating	zinc finger
MA0013.1	6	0.40	Drosophila melanogaster	br_Z4	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0400.1	6	0.40	Saccharomyces cerevisiae	SUT2	Zinc-coordinating	Fungal Zn cluster
MA0409.1	6	0.40	Saccharomyces cerevisiae	TYE7	Zipper-Type	Helix-Loop-Helix
GGAC	6	0.40				
GGCA	6	0.40				
GGGG	6	0.40				
GGTA	6	0.40				
GTAA	6	0.40				
GTAG	6	0.40				
MA0002.1	6	0.40	Homo Sapiens	RUNX1	lg-fold	Runt
MA0018.1	6	0.40	Homo sapiens	CREB1	Zipper-Type	Leucine Zipper
MA00445.1	6	0.40	Drosophila melanogaster	D	other alpha-helix	High Mobility Group
MA0102.2	6	0.40	Vertebrata	CEBPA	Zipper-Type	Leucine Zipper
MA0137.1	6	0.40				
TAAT	6	0.40				
TAGA	6	0.40				
TAGG	6	0.40				
TCTG	6	0.40				
TGCA	6	0.40				
TGGC	6	0.40				
TTAT	6	0.40				
TTCA	6	0.40				
TTGG	6	0.40				
TTTC	6	0.40				
TTTT	6	0.40				
AAGG	5	0.33				
ATGA	5	0.33				
btwisted_turn_structure	5	0.33				
btwisted_twist_structure	5	0.33				
CATT	5	0.33				

Feature name	Rank	Occurance	species	Name	class	Family
CCCT	5	0.33				
CCGA	5	0.33				
CGCA	5	0.33				
CTTG	5	0.33				
GAAT	5	0.33				
GAGG	5	0.33				
GATA	5	0.33				
GTTA	5	0.33				
GTTT	5	0.33				
MA0306.1	5	0.33	Saccharomyces cerevisiae	GIS1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0249.1	5	0.33	Drosophila melanogaster	twi	Zipper-type	Helix-Loop-Helix
MA0155.1	5	0.33	Homo sapiens	INSM1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0192.1	5	0.33	Drosophila melanogaster	Hmx	Helix-Turn-Helix	Homeo
MA0186.1	5	0.33	Drosophila melanogaster	Dfd	Helix-Turn-Helix	Homeo
MA0039.2	5	0.33	Mus musculus	Klf4	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0436.1	5	0.33	Saccharomyces cerevisiae	YPR022C	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0242.1	5	0.33	Drosophila melanogaster	run::Bgb	Ig-fold	Runt
MA0224.1	5	0.33	Drosophila melanogaster	exex	Helix-Turn-Helix	Homeo
MA0459.1	5	0.33	Drosophila melanogaster	tll	Zinc-coordinating	Hormone-nuclear Receptor
MA0225.1	5	0.33	Drosophila melanogaster	ftz	Helix-Turn-Helix	Homeo
MA0455.1	5	0.33	Drosoph	OdsH	Helix-	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
			ila melanogaster		Turn-Helix	
MA0100.1	5	0.33	Mus musculus	Myb	Helix-Turn-Helix	Myb
MA0359.1	5	0.33	Saccharomyces cerevisiae	RAP1	Helix-Turn-Helix	Myb
MA0062.2	5	0.33	Mus musculus	GABPA	Winged Helix-Turn-Helix	Ets
MA0072.1	5	0.33	Homo sapiens	RORA_2	Zinc-coordinating	Hormone-nuclear Receptor
MA0131.1	5	0.33	Homo sapiens	MIZF	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0047.2	5	0.33	Mus musculus	Foxa2	Winged Helix-Turn-Helix	Forkhead
MA0447.1	5	0.33	Drosophila melanogaster	gt	Zipper-type	Leucine Zipper
MA0273.1	5	0.33	Saccharomyces cerevisiae	ARO80	Zinc-coordinating	Fungal Zn cluster
MA0141.1	5	0.33	Mus musculus	Esrrb	Zinc-coordinating	Hormone-nuclear Receptor
MA0259.1	5	0.33	Homo sapiens	HIF1A::ARNT	Zipper-Type	Helix-Loop-Helix
MA0284.1	5	0.33	Saccharomyces cerevisiae	CIN5	Zipper-Type	Leucine Zipper
MA0036.1	5	0.33	Homo sapiens	GATA2	Zinc-coordinating	GATA
MA0375.1	5	0.33	Saccharomyces cerevisiae	RSC30	Zinc-coordinating	Fungal Zn cluster
MA0367.1	5	0.33	Saccharomyces cerevisiae	RGT1	Zinc-coordinating	Fungal Zn cluster
MA0110.1	5	0.33	Arabidopsis thaliana	ATHB-5	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
MA0094.2	5	0.33	Drosophila melanogaster	Ubx	Helix-Turn-Helix	Homeo
MA0030.1	5	0.33	Homo sapiens	FOXF2	Winged Helix-Turn-Helix	Forkhead
MA0243.1	5	0.33	Drosophila melanogaster	sd	Helix-Turn-Helix	Homeo
MA0025.1	5	0.33	Homo sapiens	NFIL3	Zipper-Type	Leucine Zipper
MA0042.1	5	0.33	Homo sapiens	FOXI1	Winged Helix-Turn-Helix	Forkhead
MA0295.1	5	0.33	Saccharomyces cerevisiae	FHL1	Winged Helix-Turn-Helix	Forkhead
MA0176.1	5	0.33	Drosophila melanogaster	CG15696	Helix-Turn-Helix	Homeo
MA0005.1	5	0.33	Arabidopsis thaliana	AG	Other Alpha-Helix	MADS
MA0071.1	5	0.33	Homo sapiens	RORA_1	Zinc-coordinating	Hormone-nuclear Receptor
MA0190.1	5	0.33	Drosophila melanogaster	Gsc	Helix-Turn-Helix	Homeo
MA0433.1	5	0.33	Saccharomyces cerevisiae	YOX1	Helix-Turn-Helix	Homeo
MA0299.1	5	0.33	Saccharomyces cerevisiae	GAL4	Zinc-coordinating	Fungal Zn cluster
MA0019.1	5	0.33	Rattus norvegicus	Ddit3::Cebpa	Zipper-Type	Leucine Zipper
MA0267.1	5	0.33	Saccharomyces cerevisiae	ACE2	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0286.1	5	0.33	Saccharomyces cerevisi	CST6	Zipper-Type	Leucine Zipper

Feature name	Rank	Occurance	species	Name	class	Family
			ae			
MA0313.1	5	0.33	Saccharomyces cerevisiae	HAP2	Other Alpha-Helix	NFY CCAAT-binding
MA0319.1	5	0.33	Saccharomyces cerevisiae	HSF1	Winged Helix-Turn-Helix	E2F
MA0032.1	5	0.33	Homo sapiens	FOXC1	Winged Helix-Turn-Helix	Forkhead
MA0101.1	5	0.33	Homo sapiens	REL	Ig-fold	Rel
MA0122.1	5	0.33	Mus musculus	Nkx3-2	Helix-Turn-Helix	Homeo
MA0170.1	5	0.33	Drosophila melanogaster	C15	Helix-Turn-Helix	Homeo
MA0138.2	5	0.33	Homo sapiens	REST	Zinc-coordinating	BetaBetaAlpha-zinc finger
TAGT	5	0.33				
TATA	5	0.33				
AACG	4	0.27				
AATA	4	0.27				
AATG	4	0.27				
ACGC	4	0.27				
AGAT	4	0.27				
AGCC	4	0.27				
AGTG	4	0.27				
ATCT	4	0.27				
btwisted_stackenergy_structure	4	0.27				
CAAA	4	0.27				
CACG	4	0.27				
CCTA	4	0.27				
GACC	4	0.27				
GAGA	4	0.27				
GGTC	4	0.27				
GGTG	4	0.27				
MA0153.1	4	0.27	Homo sapiens	HNF1B	Helix-Turn-Helix	Homeo
MA0040.1	4	0.27	Rattus norvegicus	Foxq1	Winged Helix-Turn-Helix	Forkhead
MA0303.1	4	0.27	Sacchar	GCN4	Zipper-	Leucine Zipper

Feature name	Rank	Occurance	species	Name	class	Family
			omyces cerevisiae		Type	
MA0104.2	4	0.27	Mus musculus	Mycn	Zipper-Type	Helix-Loop-Helix
MA0311.1	4	0.27	Saccharomyces cerevisiae	HAL9	Zinc-coordinating	Fungal Zn cluster
MA0237.1	4	0.27	Drosophila melanogaster	pan	Other Alpha-Helix	High Mobility Group
MA0328.1	4	0.27	Saccharomyces cerevisiae	MATALPHA2	Helix-Turn-Helix	Homeo
MA0209.1	4	0.27	Drosophila melanogaster	ap	Helix-Turn-Helix	Homeo
MA0327.1	4	0.27	Saccharomyces cerevisiae	MATA1	Helix-Turn-Helix	Homeo
MA0289.1	4	0.27	Saccharomyces cerevisiae	DAL80	Zinc-coordinating	GATA
MA0084.1	4	0.27	Homo sapiens	SRY	Other Alpha-Helix	High Mobility Group
MA0257.1	4	0.27	Drosophila melanogaster	zen2	Helix-Turn-Helix	Homeo
MA0382.1	4	0.27	Saccharomyces cerevisiae	SKO1	Zipper-Type	Leucine Zipper
MA0050.1	4	0.27	Homo sapiens	IRF1	Winged Helix-Turn-Helix	IRF
MA0149.1	4	0.27	Homo sapiens	EWSR1-FLI1	Winged Helix-Turn-Helix	Ets
MA0067.1	4	0.27	Mus musculus	Pax2	Helix-Turn-Helix	Homeo
MA0012.1	4	0.27	Drosophila melanogaster	br_Z3	Zinc-coordinating	BetaBetaAlpha-zinc finger

Feature name	Rank	Occurance	species	Name	class	Family
MA0103.1	4	0.27	Gallus gallus	ZEB1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0317.1	4	0.27	Saccharomyces cerevisiae	HCM1	Winged Helix-Turn-Helix	Forkhead
MA0425.1	4	0.27	Saccharomyces cerevisiae	YGR067C	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0411.1	4	0.27	Saccharomyces cerevisiae	UPC2	Zinc-coordinating	Fungal Zn cluster
MA0169.1	4	0.27	Drosophila melanogaster	B-H2	Helix-Turn-Helix	Homeo
MA0221.1	4	0.27	Drosophila melanogaster	eve	Helix-Turn-Helix	Homeo
MA0053.1	4	0.27	Zea mays	MNB1A	Zinc-coordinating	Dof
MA0187.1	4	0.27	Drosophila melanogaster	Dll	Helix-Turn-Helix	Homeo
MA0080.2	4	0.27	Homo sapiens	SPI1	Winged Helix-Turn-Helix	Ets
MA0018.2	4	0.27	Rattus norvegicus	CREB1	Zipper-Type	Leucine Zipper
MA0391.1	4	0.27	Saccharomyces cerevisiae	STB4	Zinc-coordinating	Fungal Zn cluster
MA0206.1	4	0.27	Drosophila melanogaster	abd-A	Helix-Turn-Helix	Homeo
MA0174.1	4	0.27	Drosophila melanogaster	CG42234	Helix-Turn-Helix	Homeo
MA0055.1	4	0.27	Homo sapiens	Myf	Zipper-Type	Helix-Loop-Helix
MA0173.1	4	0.27	Drosophila melanogaster	CG11617	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
MA0196.1	4	0.27	Drosophila melanogaster	NK7.1	Helix-Turn-Helix	Homeo
MA0017.1	4	0.27	Homo sapiens	NR2F1	Zinc-coordinating	Hormone-nuclear Receptor
MA0417.1	4	0.27	Saccharomyces cerevisiae	YAP5	Zipper-Type	Leucine Zipper
MA0338.1	4	0.27	Saccharomyces cerevisiae	MIG2	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0074.1	4	0.27	Homo sapiens	RXRA::VDR	Zinc-coordinating	Hormone-nuclear Receptor
TACG	4	0.27				
TATC	4	0.27				
TCGA	4	0.27				
TCGG	4	0.27				
TGAG	4	0.27				
TTCT	4	0.27				
ACTA	3	0.20				
ACTG	3	0.20				
AGGA	3	0.20				
AGGG	3	0.20				
AGGT	3	0.20				
ATCG	3	0.20				
ATTG	3	0.20				
banana_curve_structure	3	0.20				
GATC	3	0.20				
GCGT	3	0.20				
GCTA	3	0.20				
GTCA	3	0.20				
MA0108.1	3	0.20	Vertebrata	TBP	Beta-sheet	TATA-binding
TACA	3	0.20				
MA0372.1	3	0.20	Saccharomyces cerevisiae	RPH1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0203.1	3	0.20	Drosophila melanogaster	Scr	Helix-Turn-Helix	Homeo
MA0189.1	3	0.20	Drosophila	E5	Helix-Turn-Helix	Homeo

Feature name	Rank	Occurance	species	Name	class	Family
			melanogaster			
MA0158.1	3	0.20	Mus musculus	HOXA5	Helix-Turn-Helix	Homeo
MA0099.2	3	0.20	Homo sapiens	AP1	Zipper-Type	Leucine Zipper
MA0341.1	3	0.20	Saccharomyces cerevisiae	MSN2	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0146.1	3	0.20	Mus musculus	Zfx	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0016.1	3	0.20	Drosophila melanogaster	usp	Zinc-coordinating	Hormone-nuclear Receptor
MA0220.1	3	0.20	Drosophila melanogaster	en	Helix-Turn-Helix	Homeo
MA0231.1	3	0.20	Drosophila melanogaster	lbe	Helix-Turn-Helix	Homeo
MA0022.1	3	0.20	Drosophila melanogaster	dl_1	Ig-fold	Rel
MA0416.1	3	0.20	Saccharomyces cerevisiae	YAP3	Zipper-Type	Leucine Zipper
MA0300.1	3	0.20	Saccharomyces cerevisiae	GAT1	Zinc-coordinating	GATA
MA0087.1	3	0.20	Mus musculus	Sox5	Other Alpha-Helix	High Mobility Group
MA0337.1	3	0.20	Saccharomyces cerevisiae	MIG1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0079.2	3	0.20	Homo sapiens	SP1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0038.1	3	0.20	Rattus norvegicus	Gfi	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0364.1	3	0.20	Saccharomyces cerevisi	REI1	Zinc-coordinating	BetaBetaAlpha-zinc finger

Feature name	Rank	Occurance	species	Name	class	Family
			ae			
MA0268.1	3	0.20	Saccharomyces cerevisiae	ADR1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0325.1	3	0.20	Saccharomyces cerevisiae	LYS14	Zinc-coordinating	Fungal Zn cluster
MA0261.1	3	0.20	Caenorhabditis elegans	lin-14	Other	Other
MA0129.1	3	0.20	Nicotiana sp.	TGA1A	Zipper-Type	Leucine Zipper
MA0398.1	3	0.20	Saccharomyces cerevisiae	SUM1	Other	AT-hook
MA0422.1	3	0.20	Saccharomyces cerevisiae	YDR520C	Zinc-coordinating	Fungal Zn cluster
MA0310.1	3	0.20	Saccharomyces cerevisiae	HAC1	Zipper-Type	Leucine Zipper
MA0193.1	3	0.20	Drosophila melanogaster	Lag1	Helix-Turn-Helix	Homeo
MA0091.1	3	0.20	Homo sapiens	TAL1::TCF3	Zipper-Type	Helix-Loop-Helix
MA0001.1	3	0.20	Arabidopsis thaliana	AGL3	Other Alpha-Helix	MADS
MA0061.1	3	0.20	Homo sapiens	NF-kappaB	Ig-fold	Rel
TCTT	3	0.20				
TGAT	3	0.20				
TGGT	3	0.20				
TGTT	3	0.20				
AATT	2	0.13				
ACGA	2	0.13				
ATAG	2	0.13				
CCCG	2	0.13				
CCGG	2	0.13				
CTAA	2	0.13				
CTGA	2	0.13				
CTGT	2	0.13				

Feature name	Rank	Occurance	species	Name	class	Family
GGCG	2	0.13				
GGGC	2	0.13				
MA0102.1	2	0.13	Mus musculus , Rattus norvegicus	CEBPA	Zipper-Type	Leucine Zipper
TACT	2	0.13				
TCTA	2	0.13				
TGCG	2	0.13				
TGCT	2	0.13				
MA0127.1	2	0.13	Pisum sativum	PEND	Zipper-Type	Leucine Zipper
MA0082.1	2	0.13	Antirrhinum majus	squamosa	Other Alpha-Helix	MADS
MA0366.1	2	0.13	Saccharomyces cerevisiae	RGM1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0431.1	2	0.13	Saccharomyces cerevisiae	YML081W	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0097.1	2	0.13	Antirrhinum majus	bZIP911	Zipper-Type	Leucine Zipper
MA0123.1	2	0.13	Zea mays	abi4	Beta-Hairpin-Ribbon	AP2 MBD-like
MA0343.1	2	0.13	Saccharomyces cerevisiae	NDT80	Ig-fold	NDT80/PhoG
MA0195.1	2	0.13	Drosophila melanogaster	Lim3	Helix-Turn-Helix	Homeo
MA0282.1	2	0.13	Saccharomyces cerevisiae	CEP3	Zinc-coordinating	Fungal Zn cluster
MA0181.1	2	0.13	Drosophila melanogaster	Vsx1	Helix-Turn-Helix	Homeo
MA0451.1	2	0.13	Drosophila melanogaster	kni	Zinc-coordinating	Hormone-nuclear Receptor
MA0081.1	2	0.13	Homo	SPIB	Winged	Ets

Feature name	Rank	Occurance	species	Name	class	Family
			sapiens		Helix-Turn-Helix	
MA0363.1	2	0.13	Saccharomyces cerevisiae	REB1	Helix-Turn-Helix	Myb
MA0458.1	2	0.13	Drosophila melanogaster	slp1	Winged Helix-Turn-Helix	Forkhead
MA0456.1	2	0.13	Drosophila melanogaster	opa	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0156.1	2	0.13	Rattus norvegicus	FEV	Winged Helix-Turn-Helix	Ets
MA0200.1	2	0.13	Drosophila melanogaster	Pph13	Helix-Turn-Helix	Homeo
MA0330.1	2	0.13	Saccharomyces cerevisiae	MBP1::SWI6	Ig-fold	Rel
MA0059.1	2	0.13	Homo sapiens	MYC::MAX	Zipper-Type	Helix-Loop-Helix
MA0184.1	2	0.13	Drosophila melanogaster	CG9876	Helix-Turn-Helix	Homeo
MA0120.1	2	0.13	Zea mays	id1	Zinc-coordinating	BetaBetaAlpha-zinc finger
TTAC	2	0.13				
TTCC	2	0.13				
AACC	1	0.07				
ATTA	1	0.07				
GTAT	1	0.07				
MA0368.1	1	0.07	Saccharomyces cerevisiae	RIM101	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0079.2	1	0.07	Homo sapiens	SP1	Zinc-coordinating	BetaBetaAlpha-zinc finger
MA0078.1	1	0.07	Mus musculus	Sox17	Other Alpha-Helix	High Mobility Group
MA0111.1	1	0.07	Mus musculus	Spz1	Other	Other
MA0125.1	1	0.07	Mus	Nobox	Helix-	Homeo

Feature name	**Rank**	**Occurance**	**species**	**Name**	**class**	**Family**
			musculus		Turn-Helix	
MA0034.1	1	0.07	Hordeum vulgare	Gamyb	Helix-Turn-Helix	Myb
MA0175.1	1	0.07	Drosophila melanogaster	CG13424	Helix-Turn-Helix	Homeo
MA0212.1	1	0.07	Drosophila melanogaster	bcd	Helix-Turn-Helix	Homeo
MA0376.1	1	0.07	Saccharomyces cerevisiae	RTG3	Zipper-Type	Helix-Loop-Helix

Appendix B Source codes

Ported wordcount source code

```
#!/bin/bash -l
#@ environment = $FILE;$EXEC


for a in A G C T
do
for g in A G C T
do
for c in A G C T
do
for t in A G C T
do
   echo $a$g$c$t >> wordcounts #
done
done
done
done
INPUT_SEQUENCE="fasta_input"  #defined in the input port


INPUTFILE=`boinc resolve_filename "${INPUT_SEQUENCE}"`
echo ${INPUTFILE} 1>&2; #for testing the name


echo "start of for looop to check which one is not expanding "

awk  '{
i=NR%2;
j=NR;
if (i == 1) print $0 >> j"_wordcount"  #id file
if (i == 0) {

print $0 >> (j-1)"_wordcount";  #sequence
close((j-1)"_wordcount");
}

}' < ${INPUTFILE}

EXECUTABLE_FILE="wordcount.exe";
echo "start of for looop to check which one is not expanding "
for f in *"_wordcount"
do

  options="-sequence "$f" -wordsize=4 -outfile "$f".wordcount";

 echo "START /B /WAIT" $EXECUTABLE_FILE " " $options > start.bat


 ./start.bat

 echo "" >> $f".wordcount"  #important for seeing end of file

done

echo "individual files are generated " 1>&2
```

```
WORD_LIST_FILE="wordcounts"  #contains list of motif in wordcounts
database
pat=`awk '{a=$1","a} END{ print a;}' < ${WORD_LIST_FILE}`
pat=`echo "${pat%?}"`
echo $pat > attrlist_awk  #space make problem here
 #split("auto-da-fe", a, "-")
#pat="sdf"
 echo "" >  "attrlist_awk_cpg" ;
 rm attrlist_awk_cpg;
 #some files are missing
 awk -v arra=$pat '
 #BEGIN {counter=0}
 {
 if (FNR==1)  #it is like begin for one file
 {
    split(arra,a,",");
    i=0;
for (word in a)
  {
  #print a[word] ;
  word_arr[a[word]]=0;i++;
  #print i
  }
n=i;
for(word in a)
  wordsname=wordsname"\t"a[word]
if(NR==1) print "FILENAME""\t"wordsname >> "attrlist_awk"
if(NR==1) print "cpg_id""\t"wordsname >> "attrlist_awk_cpg"

if (NR==1) counter=0;
t="";
}

 word_arr[$1]=$2; #just values
 if ( FNR == 1 ) {counter=counter+1;} #    print FILENAME; FNR starts
from 1

 #getline endoffile < FILENAME
 #print "ee"endoffile

 if($0 == "") { #check for end of file last line of the file is empty
with echo

 print counter"\t"FILENAME #check for last line of the current file
like END for one file
 for(word in a)
   t=t"\t"word_arr[a[word]];
 name[1]="";
 split(FILENAME,name,".");

 getline tr < name[1]  #read next record from file to variable tr
 close(name[1]);
 #tf=gsub(/\r/,"",tr) #return number of occurance carriage return and
change tr
 gsub(/\r/,"",tr) #change tr

 print FILENAME"\t"t >> "attrlist_awk";  #use filename as id
 #print FILENAME"\t"tr"\t"t >> "attrlist_awk_cpg";
 print tr"\t"t >> "attrlist_awk_cpg";

 }
 }
 '  *.wordcount  #"wordlist.txt"  reading aoutput files


  echo  > outp_o
 rm outp_o;
 awk '{
 for(i=1;i<=NF;i++)
{

 if(i!=2) printf $i"\t" >>  "outp_o" ;
 if(i==NF) printf $i >>  "outp_o" ;

 }
 print "" >>  "outp_o"}' attrlist_awk_cpg




 outp=`boinc resolve_filename "tabdelimited"`
 cp outp_o ${outp}

boinc finish 0
```

Ported jaspscan source code

```
INPUT_SEQUENCE="fasta_input"  #defined in the input port
INPUTFILE=`boinc resolve_filename "${INPUT_SEQUENCE}"`
echo ${INPUTFILE} 1>&2; #for testing the name

# tr -d '\015' < ${INPUTFILE} > OUTPUT.TXT #removing CR from file

i=0


EXECUTABLE_FILE="jaspscan.exe"; #name of the executable


outp=`boinc resolve_filename "tabdelimited"`
echo $outp 1>&2
MATRIX_LIST_NAME="matrix_list.txt"  #contains list of motif in JASPER
database
MATRIX_LIST_FILE=`boinc resolve_filename "${MATRIX_LIST_NAME}"`
MOTIF_EXTRACTED=`awk '{t=t "\t" $1}END{sub(/ /, "\t", t);print t}' <
${MATRIX_LIST_FILE}`

echo "entered while" 1>&2
echo "entered while"
#options="-threshold 80 -menu C -matrice all -outfile
"$INPUTFILE"jasper.out -sequence "$INPUTFILE;
options="-threshold 80 -menu C -matrice all -outfile
"$INPUT_SEQUENCE"jasper.out -sequence "$INPUTFILE;

 echo "START /B /WAIT" $EXECUTABLE_FILE " " $options > start.bat
#trying with start b wait
 ./start.bat  # run jasper
 #cp * c://snapshot
# exec 4< "Expr_A_Varmethylated_platform_1_removeoverlap.fajasper.out"
 awk '{
   if($0 ~ /# Sequence/) {cpgid=$3;print  $3;}
   else cpgid="";
   if (cpgid != "" ) {MSTR="";i=0;cpg_id=cpgid;}
   #if($0 ~ /MA/) {MA=$5;print  $5;}
   if($0 ~ /MA/) {MA=$5;}
   else MA="";
   if (MA != "" ) {MSTR=MSTR"\t"MA}
   if($0 ~ /#---------------------------------------/) {i=i+1;
      if (i==2)
   print cpg_id"\t"MSTR > "motifinline";
 }
  # else MA="";
 }' "fasta_inputjasper.out"
 echo rt;

 tr ' ' '\t' <motifinline > motiflinetab
  Stringindexallmotif="";
pat=`awk '{a=$1","a} END{ print a;}' < ${MATRIX_LIST_FILE}`
 pat=`echo "${pat%?}"`
 echo $pat > attrlist

 awk -v arra=$pat '
     BEGIN
    {
#motifsname="cpg_id"
#print arra;
    split(arra,a,",");
    i=0;
for (motif in a)
  {print a[motif] ; motif_arr[a[motif]]=0;i++;print i}
n=i;
for(motif in a)
   motifsname=motifsname"\t"a[motif]
#print $1"\t"motifsname >> "attrlist"
print "cpg_id""\t"motifsname >> "attrlist"
}
{
motifs="";
motifsname"";
for (motif in a)
  {motif_arr[a[motif]]=0;}
for(i=1;i<=NF;i++)
motif_arr[$i]=motif_arr[$i]+1;
for (motif in a)
  motifs=motifs"\t"motif_arr[a[motif]];
#if(NR=1) print "cpg_id""\t"motifs >> "attrlist"
#else
print $1"\t"motifs >> "attrlist"
}
END{ print "sd"}' < motiflinetab #adding all associative array



 #tr 't' '\t' <attrlist > ${outp}
 tr 't' '\t' < attrlist > outp_1

 echo  > attrlistfirstlineremoved1
 #tr -d '\n' < attrlistfirstlineremoved
 tr -d '\n' < attrlistfirstlineremoved1 > attrlistfirstlineremoved

 awk '{if (NR>1) print $0 >> "attrlistfirstlineremoved"}' outp_1
 echo  > outp_o
 rm outp_o;
 awk '{
```

```
 for(i=1;i<=NF;i++)
{

 if(i!=2) printf $i"\t" >>  "outp_o" ;
 if(i==NF) printf $i >>  "outp_o" ;

 }
 print "" >>  "outp_o"}' attrlistfirstlineremoved

 cp outp_o ${outp}

 echo $cpg_id 1>&2
boinc finish 0
```