MACHINE LEARNING ALGORITHMS FOR RNA MOTIF CLUSTERING IN LARGE-SCALE PLANT MIRNA DATA

Weiguang Zhou

A Dissertation submitted to
the School of Computing Sciences of The University of East Anglia
in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE.
NOVEMBER, 2016

© This Dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that no quotation from the dissertation, nor any information derived therefrom, may be published without the author or the supervisor's prior consent.

SUPERVISOR(S), MARKERS/CHECKER AND ORGANISER

The undersigned hereby certify that the markers have independently marked the dissertation entitled "**Machine learning algorithms for RNA motif clustering in large-scale plant miRNA data**" by **Weiguang Zhou**, and the external examiner has checked the marking, in accordance with the marking criteria and the requirements for the degree of **Master of Science**.

Supervisor: _____
Dr. Wenjia Wang, Dr. Steve Hayward

Markers: _____
Marker 1: Dr. Wenjia Wang

_____
Marker 2: Dr. Steve Hayward

External Examiner: _____
Checker/Moderator

Moderator: _____
Dr. Wenjia Wang

DISSERTATION INFORMATION AND STATEMENT

Dissertation Submission Date: **November, 2016**

Student:	**Weiguang Zhou**
Title:	**Machine learning algorithms for RNA motif clustering in large-scale plant miRNA data**
School:	**Computing Sciences**
Course:	**Advanced Computing Science**
Degree:	**M.Sc.**
Year:	**2016**
Organiser:	**Dr. Wenjia Wang**

STATEMENT:

Unless otherwise noted or referenced in the text, the work described in this dissertation is, to the best of my knowledge and belief, my own work. It has not been submitted, either in whole or in part for any degree at this or any other academic or professional institution.

Permission is herewith granted to The University of East Anglia to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

_____

Signature of Student

Abstract

Micro RNAs are small non-coding RNAs, with approximately 21 nucleotides. They regulate the expression of specific target genes at the post-transcriptional level, such as transcript cleavage. This project is the first exploration of the application of clustering algorithms to develop novel analysis pipelines to recognise in vivo RNA structure patterns at and around the micro RNA cleavage site.

Because novelty of this research, there was no tool to use. Therefore, I created the Integrated Clustering Analysis System (ICAS) to do the work. ICAS made data preprocessing organised, clear and reproducible. The ICAS clusters the cleavage sites into 2 to 15 clusters both by reactivity features and distance matrices. Then it performs ensemble clustering analysis on the individual results. For each of the clusterings, the ICAS generates a report with statistical and biological details, and a dozen of tables and charts.

When clustering by reactivity, clusterings with 2 or 3 clusters were discovered, in which each cluster was significantly different from others on cleavage efficiency. Most of them were not stable. However, when using k-mean with k equalled 2, a stable clustering was found. It was stable because every time k-means gave the identical clustering result.

When cleavage sites were clustered by distance matrices, some unstable clusterings with 2 or 3 clusters were found. However, the structures and distance matrices were not trustworthy. Some problems were addressed in the paper, suggestion given.

Ensemble clustering improved the performance in 60% of the cases and was able to discover more table clusterings with 2, 3 or 4 clusters than individual algorithms.

Acknowledgements

I would like to thank my supervisors, Dr. Wenjia Wang and Dr. Steve Hayward, for their many useful suggestions and constant support during this research. I would also like to thank Dr. Ji Zhou, Dr. Minglei Yang, Dr. Qi Liu, Dr Zhang Hang, Dr. Jitender and all members in the JIC for all the help provided related with biology and bioinformatics.

I would also like to express my gratitude to whoever sponsored my course in full or part.

Finally, I am very grateful to my family for their patience and *love*.

Weiguang Zhou

at Norwich, UK.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AGO Argonaute. 1

AMI Adjusted Mutual Information. 15

ANOVA Analysis of variance. 48, 62

cDNA Complementary DNA. 27

CS Cleavage Site. 44

CSV Comma-separated values. 34, 37

DBN Dot-Bracket Notation. 5, 6, 33, 35, 37

DNA Deoxyribonucleic acid. 25

FASTA Fast-all. 8, 27

GUI Graphical User Interface. vii, ix, xi, xii, 16, 17, 20, 33, 35, 63, 64

HPC High-performance Computing. xi, 17, 20, 48

ICAS Integrated Clusteing Analysis System. xi, 3, 16–18, 21, 33, 44, 46, 55, 56, 65, 70

IDE Integrated Development Environment. 16

JIC John Innes Centre. 2

miRISC miRNA-induced silencing complex. 1

miRNA micro RNA. iv, vi, 1–3, 5, 8–10, 25, 27, 30, 31

mRNA messenger RNA. 1, 2, 25, 29

NMI Normalised Mutual Information. 15

PARE parallel analysis of RNA ends. 27

PCA Principal component analysis. x, xi, 38, 41, 42, 45, 46, 49

PNG Portable Network Graphics. 37

RNA Ribonucleic acid. iv, x, 3, 5, 13, 25, 26, 33, 36, 51, 52, 54, 74

RPKM Reads Per Kilobase of transcript per Million mapped reads. 29

SVG Scalable Vector Graphics. 37

SVM Support Vector Machine. 2, 9, 73, 74

UEA University of East Anglia. 17

WT Wild type. xi, 26, 27, 32, 36, 37, 41–44, 47, 55

XRN4 5'-3' Exoribonuclease 4. xi, 26, 27, 30, 32, 36, 37, 41–44, 47, 55

Chapter 1

Introduction

1.1 Background

miRNAs are endogenous non-coding molecules that regulate the gene express at the transcriptional level (Ambros, 2004; Bartel, 2004). They are combined with Argonaute (AGO) proteins and then form into the miRNA-induced silencing complex (miRISC). (Fabian and Sonenberg, 2012) miRNAs silence messenger RNAs (mRNAs) via base-pairing with complementary sequences within mRNA molecule (Bartel, 2009). The cleavage site in the mRNA pairs with the miRNA. Figure 1.1 demonstrates two cleavage sites. EV006535 and EV142354 are two mRNAs. BNA-miR160a and Bna-miRC2 are two miRNAs. The lines between the miRNA and mRNA indicate pairing between them.

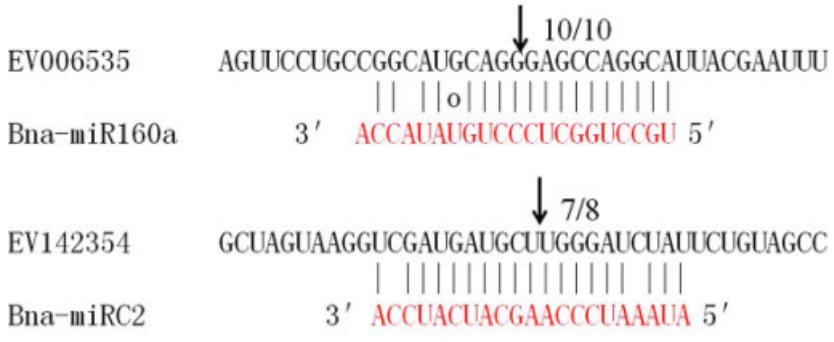

Figure 1.1: Cleavage site (Xu et al., 2012)

Many tools have been used to predict cleavage sites. Some focused on primary structure (pairing between the mRNA and the miRNA), whereas others also consider

the secondary structure.

Non machine learning tools include Targetfinder (Fahlgren et al., 2007), TAPIR (Bonnet et al., 2010), Target-align, Target Prediciton, psRNATarget, miRanda and etc. Behind these tools are a batch of algorithms, including FASTA, FASTA/RNAhybrid, Scan for matches, Smith-Waterman, Local Alignment and etc. Krek et al. (2005) used a pipeline of PicTar algorithm, RNAhybrid. Chi et al. (2012) used BLASTN and CleaveLand. Sun et al. (2011) used scan-for-match algorithm to find approximately matched target of microRNA, RNAhybrid to calculate the free energy.

Those tools are restricted by human observation of miRNA targets and machine learning algorithms can avoid human bias (Liu et al., 2008). Thus, recent researches involve more machine learning. Kim et al. (2006b); Mendoza et al. (2013); Kim et al. (2006a); Liu et al. (2008); Ahmed et al. (2013); Menor et al. (2014) used machine learning methods to identify cleavage sites. These algorithms were mainly classifiers, including Support Vector Machine (SVM), logistic regression, Fisher's linear discriminant analysis (FLDA), naïve Bayes (NB), the SVM and the random forest. Among them SVM was the most used one.

The client of this project is the JIC, who made the most decisions and directions concerning biology.

1.2 Need to study and purpose of the study

Jiang et al. (2009) stated that miRNAs were of crucial importance in human disease development, progression, prognosis, diagnosis and evaluation of treatment response. For example, let-7 inhibition introduces a new therapy for obesity and type 2 diabetes (Zhu et al., 2011; Frost and Olson, 2011). These studies suggest the importance of miRNA study in health care of human. Hu (2002) stated that study of RNA motifs is

helpful in understanding the regulation activity.

1.3 Aim and objectives

The aim of this project is to explore the application of machine learning algorithms to develop novel analysis pipelines to recognise in vivo RNA structure patterns at and around the miRNA target site.

1.4 Methods and results

This project focuses on the relationship between RNA secondary structure (or RNA motifs) and cleavage efficiency. clustering analysis was used to group the motifs. The structure was calculated by ViennaRNA (Lorenz et al., 2011). The cleavage efficiency was calculated from degradome sequencing data.

I have developed ICAS which applied six clustering algorithms, namely k-means, k-medoids, spectral clustering, hierarchical clustering, affinity propagation, mean shift to reactivity and RNA distance datasets. Several clusterings in which each cluster is statistically significantly different form others were found, although the labellings are not ideal.

1.5 Research questions

1. Is there a clustering in which each of the clusters is significantly different from others? If yes, how many clusters are there?

2. Are the clusters biologically meaningful?

3. Which method is better in terms of reliability and trustworthiness, the method using reactivity or the one using structure distance matrices?

4. Does dimensionality reduction improve the performance of clustering analysis?

5. Does ensemble clustering improve the performance of clustering analysis?

6. Does genetic clustering improve the performance of clustering analysis?

Chapter 2

Literature Review

This chapter reviews researches on RNA secondary structure, machine learning and non-machine learning algorithms used for cleavage site discovery, and finally clustering algorithms including ensemble clustering and evaluation. No research using clustering analysis on miRNA cleavage site prediction or cleavage efficiency has been found. This project will be the first.

2.1 RNA structure and distance

The Dot-Bracket Notation (DBN) is commonly used to present RNA secondary structure. Valid structures in DBN format are strings consisting of dots '.', opening '(' and closing ')' parentheses. Dotted positions are unpaired. The paring of opening and closing parentheses abides by the same way in a mathematics equation. A nucleotide noted by an opening parenthesis always pairs with another one noted by a closing parenthesis.

Figure 2.1 displays two motifs(structures).

Motif A is denoted as follow:

GCGUGAAGUCGCAGC
((. ((. . . . .))))

Motif B is denoted as follow:

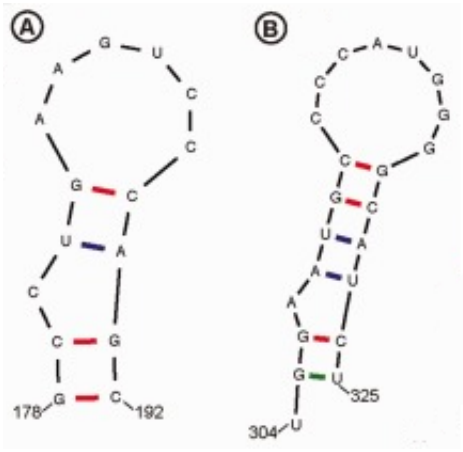

Figure 2.1: Two motifs(Pisapia et al., 2013)

UGGAAUGCCCCAUGGGGCAUCU
. ((. ((((. . . . .))))))

RNADistance in ViennaRNA (Lorenz et al., 2011) calculates the dissimilarity(distance) between structures presented by the DBN format.

2.2 Non-machine learning tools and algorithms for Target Prediction

2.2.1 PatScan

The first tool/algorithm we used in searching for homologous sequences is "scan for matches" (also known as PatScan) (Dsouza et al., 1997). It has been proved to be useful in searching for fuzzy matches in microRNA-messengerRNA duplexes (Lu et al., 2008; Sun et al., 2011; Li et al., 2016). It tests every possible site in the messenger RNA by reverse complementarity with the microRNA. It is easy to use. I have downloaded it and used it for searching microRNA and target duplexes in arabidopsis and rice. I compared the matches found by PatScan with parameters [7,0,0](7 substitutions, 0 insertion and 0 deletion) and [5,3,3](5 substitutions, 3 insertion and 3 deletion) and those by Addo-Quaye et al. (2008). The former was able to find most of the matches the latter found. However, because it uses a brute-force approach, it runs very slowly. it

took more than 1 hour to search for all the homologous sequences by a single microRNA with the parameters [5,3,3]. If I am to use this algorithm, it is better that I modify it and make it perform better. The algorithm separates edit distance and scoring. It cares only the edit distance when searching for putative targets. The score is to be given later by using other scoring systems. The benefit of that is it decouples edit distance and scoring. The demerit is that it is not efficient. The other problem which slows down the whole process is that it does not record edit distance. I have to use a global alignment algorithm to re-calculate the edit distance between the miRNA and the putative cleavage site. This is not necessary. This redundant step can be removed by recording the edit distance in the first place.

2.2.2 Smith-Waterman

Smith-Waterman (Smith and Waterman, 1981) is a local alignment algorithm. It uses a matrix with a scoring system to compare the query sequence and the genomic database. It traces back diagonally from the bottom of the matrix where a passing score is found. The scoring system can be customised.The same or different scores are given to substitutions, insertions, deletions, transversions, transitions and the positions of the above. It is much faster than PatScan, but the query sequence is still compared against every subsequence in the database.

2.2.3 FASTA and BLAST

FASTA (Pearson and Lipman, 1988) is the first fast sequence searching algorithm for comparing a query sequence and a genomic database. BLAST (Altschul et al., 1997) stands for Base Local Alignment Search Tool. It is an improvement of FASTA in terms of speed and usability. The basic idea of them is that a good alignment contains

subsequences of exact match. They firstly pinpoint those alignments and then extend to the right and left. Thus the query sequence is not needed to compare with every subsequence of the genomic database. This will greatly reduce the comparison between the query sequence and the genomic database.

2.2.4 Summary and Comparison

Srivastava et al. (2014) have compared tools and algorithms. Table 2.1 lists the mapping of tools and algorithms.

Table 2.1: Comparison of cleavage site prediction tools at their best performance with Arabidopsis

Tool	Algorithm	Score	Free energy	precision	recall
psRNATarget	Smith-Waterman	3	25	0.81	0.89
psRobot	Modified Smith-Waterman	2.8	-	0.87	0.84
Tapir fasta	Fast-all (FASTA)	4	0.7	0.89	0.9
Tapir hybrid	RNAHybrid	4	0.7	0.86	0.87
Target_Prediction	Scan for matches and RNA Hybrid	2.5	0.73	0.85	0.84
Targetfinder	FASTA	4	-	0.89	0.97

Srivastava et al. (2014) stated that Targetfinder performed the best among the selected tools (recall 88% and precision 97%) when it is tested against Arabidopsis. The ensemble of tools were also tested, but it resulted in marginal benefit. For non-Arabidopsis species, Targetfinder, psRNATarget and Tapirhyrid performed the best.

2.3 Machine Learning Methods for Target Prediction

Liu et al. (2008) stated that non-machine learning methods are restricted by human observation of miRNA targets. With machine learning algorithms, human bias can be eliminated in decision making process. Dai et al. (2011) point out that machine

learning-based methods have great potential in predicting miRNA targets. Some methods have been developed to predict miRNA targets in animals, such as RNA22 (Miranda et al., 2006), miTarget (Kim et al., 2006a) and GenMiR++ (Huang et al., 2007).

Mendoza et al. (2013) used random forest for cleavage site prediction. Kim et al. (2006a); Liu et al. (2008); Ahmed et al. (2013) used SVM. Menor et al. (2014) used logistic regression, Fisher's linear discriminant analysis (FLDA), naïve Bayes (NB), the SVM and the random forest.

2.3.1 Feature extraction and selection

Kim et al. (2006a) extracted 41 features in three categories by structure, thermodynamics and position.

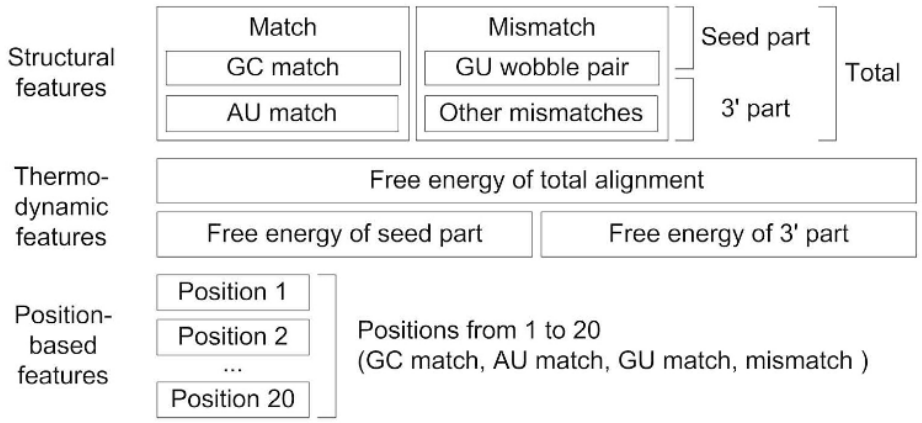

Figure 2.2: Three categories of SVM features (Kim et al., 2006a)

Menor et al. (2014) generated 151 site-level features and selected 12 (Table 2.2) by mutual information. Seven of the selected features focused on the seed region, which indicated its importance.

Although those features were used in classifiers, the feature extraction and selection step for classifiers and clusterers are the same. The methods can be reused in this study.

Table 2.2: Selected site-level features by correlation-based feature selection (Menor et al., 2014)

Feature	Description
miR_match_P01	Match status of miRNA position 1
miR_match_P03	Match status of miRNA position 3
miR_match_P04	Match status of miRNA position 4
miR_match_P08	Match status of miRNA position 8
miR_match_P15	Match status of miRNA position 15
Seed_bulge	Number of bulges in seed region
Total_AU	Number of AU matches in target site
Total_mismatch	Number of mismatches in target site
Total_bulge	Number of bulges in target site
Total_bulge_nt	Number of nucleotides within bulges in target site
Seed_P01_acc	Accessibility score position 1 of seed region
Seed_cons_score	Conservation score of seed region

2.3.2 Clustering and miRNA target prediction

No research using clustering analysis on miRNA target (or cleavage site) prediction has been found. However, some steps, such as feature selection and extraction, in other machine learning methods can be reused.

2.4 Cluster analysis

Cluster analysis or clustering is the task of clustering observations by similarity. There are more than 100 published clustering algorithms. Estivill-Castro (2002) stated that the most appropriate clustering algorithm for a particular problem often needs to be chosen by experiment, unless there is a mathematical reason.

Jain et al. (1999) emphasised the importance to distinguish between clustering and classification. The reason of ambiguity is caused by the similarity of the two. Both of them are used for data clustering. However, clustering is unsupervised whereas classification is supervised. In a classification problem, the labels of clusters are given when the model is being trained. On the contrast, in a clustering situation, no label has been given. The labels are gained from the data only.

Jain et al. (1999) stated that clustering is useful in exploratory pattern analysis and pattern classification.

Jain and Dubes (1988a) gave the steps of clustering analysis.

1. pattern representation
2. proximity/similarity definition
3. clustering
4. data abstraction
5. evaluation

Figure 2.3 demonstrates the steps.

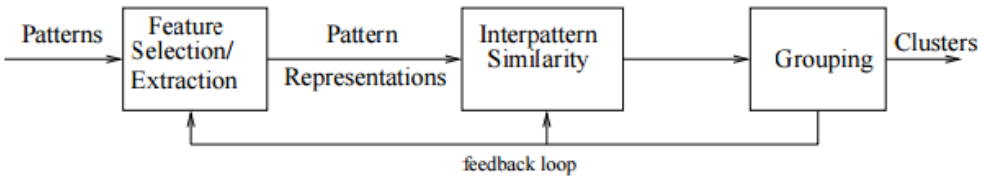

Figure 2.3: Clustering stages(Jain et al., 1999)

Pattern representation includes feature selection and extraction. Feature selection is to select the most important features. Feature extraction creates new features from the known ones (Jain et al., 1999).

Pattern similarity/proximity is measured by the distance between two observations. When the observations are points in 2D or 3D space, Euclidean distance is used without considering covariance between observations. However, most problems are not geometry, and Euclidean distance may not be the best similarity function. Thus, people also use Manhattan distance, Minkowski distance, etc.

In the clustering step, there are many algorithms. Vega-Pons and Ruiz-Shulcloper (2011) stated that when two clustering algorithms are applied to the same dataset, the

results can be very different. The two may be equally plausible when there is no prior knowledge to evaluate the results.

Berkhin (2006) categorised clustering algorithms into hierarchical methods, partition methods and six other groups. Hierarchical methods contain agglomerative algorithms and divisive algorithms whereas partition methods include relocation algorithms, probabilistic clustering, k-medoids (Kaufman and Rousseeuw, 1987) methods, k-means methods, density-based algorithms, etc.

The k-means method is widely used (Arthur and Vassilvitskii, 2007). It is the most popular clustering algorithm used in scientific and industrial applications (Berkhin, 2004). Firstly, k centroids are randomly initialised. The Euclidean distances of every observation to all the centroids are calculated and compared. The observations/points will be grouped to the nearest centroid and k groups are formed. The centroids of these groups are then calculated again, and new k centroids are generated. The above steps are then repeated until the centroids stop changing or converge.

The initial centroids might affect the final result of k-means. That is because this algorithm is only able to reach local optimum. Each time the algorithm runs, it may reach the different local optimum. Thus, a rule of thumb is the run it multiple times with different initial centroids.

Many modified versions of k-means are published to troubleshoot issues in k-means. The k-means algorithm require the group number k to be input in advance. Unfortunately, it is hard to know the group number in practice. Thus, some varied versions of k-means are developed, such as X-means(Pelleg et al., 2000) and G-means(Hamerly and Elkan, 2003). K-means uses Euclidean distance which does not cope with the covariance of data. Thus, another modified version Gaussian mixture (Press et al., 2007) has been developed. K-median (Jain and Dubes, 1988b; Bradley et al., 1997)

algorithm substitutes the mean value with median value.

K-medoids (Kaufman and Rousseeuw, 1987) is similar to k-means. Both attempt to minimise the distance between each data point and the centroid/medoid. In contrast to the k-means algorithm, k-medoids chooses data points as the centre(examplar, medoid) of the group. In some situations, the centre must be an existing data point. One example could be choosing a person to present a group of people. In RNA structure clustering, there is no average structure of a group of structures. The representative must be chosen from existing ones.

affinity propagation (Frey and Dueck, 2007) is based on the concept of "message passing" between data points until convergence. Examplars will be selected by this algorithm. each examplar is associated with a group of samples. The most important parametre is preference, which is positive correlated with the number of clusters. Damping factor controls the iteration and converge. Low damping factor leads to early converge. However, if it is too low, oscillation might occur.

Mean-shift (Comaniciu and Meer, 2002) is a centroid based algorithm. It locates the maxima of a density function. The Mean is shifted to the mean of the neighbourhood. The converge condition is that the new centroid does not have as many neighbours as the old.

Given a candidate centroid x_i, $N(x_i)$ being the neighbourhood of x_i and K being the kernel function. The candidate is updated according to Equation 2.4.1:

$$x_i = x_i + \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)} \tag{2.4.1}$$

For the kernel function. The most frequently used kernel profiles are flat and Gaussian. When Gaussian kernel is used, the bandwidth is the standard deviation σ.

Hierarchical clustering builds a hierarchy of clusters. Strategies can be either

bottom-up or top-down (Rokach and Maimon, 2005). Bottom-up or agglomerative strategy starts with each cluster containing one sample. Each of them searches for the most similar one and pair with it to form a higher level cluster. The strategy stops until every sample is in the same cluster. The top-down or divisive strategy is the opposite. Firstly, it puts every observation in a single cluster. It splits the clusters recursively until there is only one observation in every cluster.

In agglomerative clustering, the distance of clusters are measured by various ways. One way is to use the minimal distance between two clusters. The distance of each point of one cluster to another cluster is measured. The minimum value is used as the distance. The others way are to use the maximum or average of that. However, the most computational inexpensive way would be to calculate the centroids of the clusters firstly and then measure the distances of the centroids.

2.4.1 Ensemble clustering

Ensemble algorithms can also be used in clustering. An ensemble machine learning algorithm is a combination of several individual machine learning algorithms and can perform better than any individual ones when the individual ones are dissimilar. It is possible to construct good ensembles for statistical, presentational and computational reasons (Dietterich, 2000).

Clustering ensemble is also referred to as consensus function or aggregation of clusterings. It is an alternative to improve the quality of clustering algorithms which combines multiple partitionings of a set of objects without accessing the original data. (Strehl and Ghosh, 2002) There are two principal steps. The first is generation in which a set of partitions are created. The next step is consensus function which integrates all partitions generated in step one (Vega-Pons and Ruiz-Shulcloper, 2011).

People use ensemble clustering when it is impossible to go back to the original features or algorithms, such as clusterings provided by human experts instead of computational available algorithms. In practice, it might not be feasible to share the low level data due to computational, bandwidth and storage costs. Other constraints include security, privacy, the proprietary nature of data and the accompanying ownership issues, the need for fault tolerant distribution of data and services, real-time processing requirements or statutory constraints imposed by law (Prodromidis et al., 2000).

Many individual clustering algorithms focus on searching for a single optimal clustering by some specific criteria. Consensus clustering is able to outperform individual ones. Consensus clustering algorithms often generate better labellings, less sensitive to noise, outliers or sample variations (Nguyen and Caruana, 2007).

2.4.2 Clustering evaluation

Evaluation of the clustering quality is a non-trivial and ill-posed task (Slonim and Tishby, 1999). As a matter of fact, there are a great number of objective functions (Jain and Dubes, 1988a). Internal criteria, such as squared error and compactness, evaluate the result with respect to intrinsic data only.

External criteria, on the other hand, measure quality by external source. Many of these metrics are used to measure the agreement of two assignments, such as purity (Boley et al., 1999), Euclidean method, Adjusted Mutual Information (AMI), Normalised Mutual Information (NMI), Rand index (Rand, 1971), etc. These measures all use an external assignment to compare the clustering result.

Chapter 3

Analysis and proposed methods

Since this project is the first to use clustering algorithms on cleavage site analysis, there is no ready-to-use tools and metrics. I developed a new tool as a solution to pre-process raw data and perform clustering. I also defined a new metric, significant rate, to measure the clustering performance.

3.1 Integrated Clusteing Analysis System(ICAS)

This project is the first of its kind, and there is no customised tools available. Thus I have to develop a tool for this research.

Reproducibility is an important requirement for any research. One option would be to log each step thoroughly and the successor of the research follows a long step list and redo the steps. The other option is to provide future researchers with software that he or she can put the raw data in and rerun the steps quickly. The ICAS makes the research more reproducible.

With the progress of this project, a great number of Python, R and C-Sharp code have been written to deal with individual tasks before the ICAS. They are not organised and hard to use without a GUI. Every time, the client changed the raw data, I had to switch between different languages and Integrated Development Environments (IDEs). It was easy to skip steps or put wrong parameters for some scripts. In a word, it

was very tedious and error-prone. The ICAS organised code together and provided an integrated GUI.

User requirements and data were always changing. The client may do the experiments again and generate more accurate data in the future. With the ICAS, the client will be able to do data pre-processing and clustering easily.

Thus, I created the ICAS, a flexible and HPC-friendly clustering pipeline. It takes the raw biology data, performs individual and ensemble clustering and reports the best clustering. It also generates scripts for the HPC in the University of East Anglia (UEA).

3.1.1 Architecture

There are four major subsystems in the ICAS, namely data pre-process, clustering, ensemble and reporting systems (See Figure 3.1).

3.1.2 Data pre-processing

The data pre-processing subsystem consists of three components. The first is cleavage site prediction. The other two components produce two different datasets, namely reactivity and cleavage site distance (See Chapter 4).

3.1.3 Clustering

These datasets are consumed by various algorithms in the clustering subsystem. Reactivity datasets are used by k-means and mean shift, distance datasets by k-medoids and hierarchical clustering, and both by spectral clustering and affinity propagation (See Chapter 5 and 6).

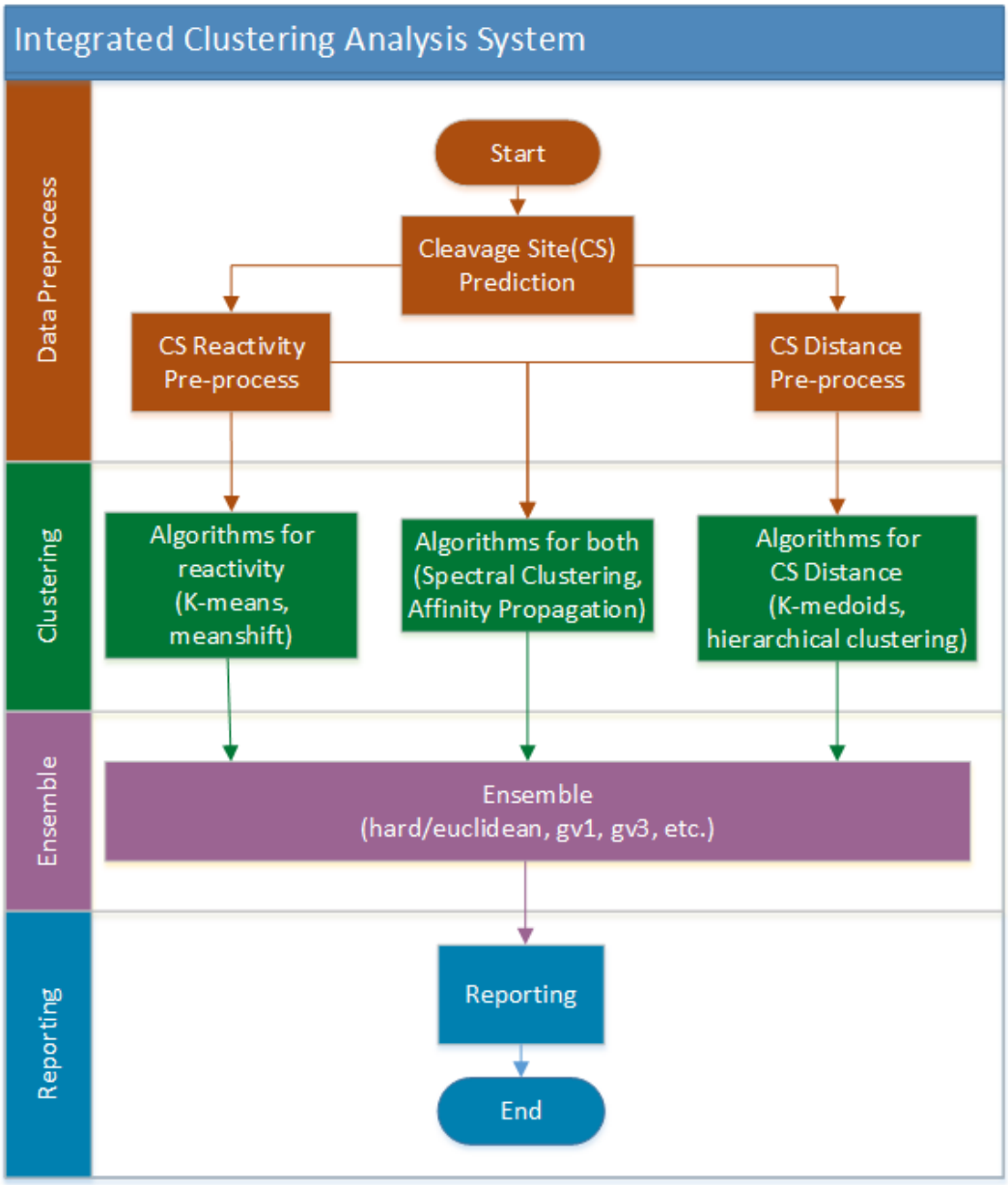

Figure 3.1: Functional flow chart of ICAS

3.1.4 Ensemble clustering

The individual clustering algorithms then pass the results to the ensemble sub-system, which takes clusterings of the same dataset and attempts to produce better results according to the evaluation criterion (See Chapter 7).

3.1.5 Reporting

The report contains three tables, the bar plot for the cluster size, the box plot for each cluster with cleavage efficiency, pairwise distance between medoids, and distances from

each data point to the medoids.

A dozen of charts are also included in the report. Most of the charts can be found in this paper (See Table 3.1).

Table 3.1: Charts in the report and this paper

Chart in the report	**Charts in this paper**
histogram for cluster size	5.4, 6.6, 7.3
box plot for each cluster	5.5
average reactivity for each cluster	5.6
reactivity variance for each cluster	
clustering centroids	5.3
structure of the medoids	
medoids and distance	6.7
structures in a cluster	6.8
distance of a cleavage site to each medoid	6.9

3.1.6 Quick clustering

The quick clustering fixates the dataset, algorithm and k(cluster count), so the algorithms may give a result very quickly. It does ensemble clustering(Hard/Euclidean) for the individual algorithm, too. It compares the results from both the individual and ensemble algorithms and chooses the best one. Finally, a report of the best clustering is generated. It is a light version of clustering, ensembling and reporting combined.

3.1.7 Flexibility

This study is exploratory, Therefore, changes happened frequently and are expected to continue happening in the future. Thus, I designed the system to be extendible and flexible.

Data flexibility

In machine learning, various algorithms are used to transform the data. This system provides the flexibility to add and remove datasets. I added some transformed datasets

to the system (See Section 5.3).

Algorithm flexibility

New algorithms are also welcome in this system, as long as the input and output meet certain standards.

The applications which run the algorithm scripts are also flexible. The system detects the script extension and chooses the application accordingly.

Implementation of flexibility

The user changes the dataset, algorithm and other settings by a batch of .csv files. The GUI is to be added in the future.

3.1.8 Support for High-performance Computing(HPC)

Clustering algorithms take a long time and tremendous computational resources to complete. This system helps generate the HPC job script, and the user can run them in the HPC. When the script finishes in the HPC, the user takes the results back and merges the results to the existing ones in the system.

The user also has the option to run without HPC, which gives users the opportunity to test the script locally first before submitting it to the HPC (See Figure 3.2).

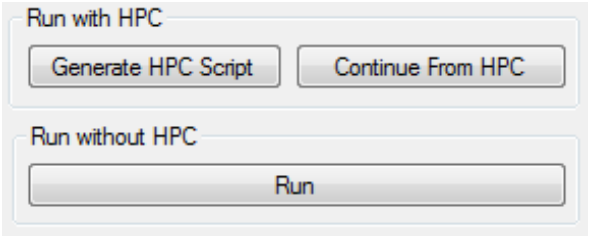

Figure 3.2: GUI for HPC

3.1.9 Validation and verification

The system has been tested internally and externally. Within the system, I created unit test cases before developing the concrete function. Outside the system, my first choice would always be published packages. Otherwise, I would check the package before using it.

Unit test

In computer science, unit testing is a software testing method by which individual units of source code, sets of program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use(Kolawa and Huizinga, 2007). I created 28 test cases. In each of the test case, there are always an actual result and an expected one. The test case compares the two results to check whether they are identical.

One example would be the test case for compactness calculation. There are four points in the test case, (0, 0), (0, 1), (1, 0) and (1, 1). The expected value was 0.24 (calculated by Equation 3.2.3). I called Function Metrix.Compactness and the rounded returned actual value was also 0.24. The expected value and the actual values were identical. Therefore, the test case was passed.

Third party packages

Seven third-party packages were used in ICAS (Figure 3.3).

Scikit-Learn (Pedregosa et al., 2011), CLUE (Hornik, 2005) and Vienna RNA (Lorenz et al., 2011) were found in published papers. Thus they are considered as reliable.

The k-medoid implementation(Alspaugh, 2011) was found from the Internet, and it was not as trustworthy as the three published ones. Therefore, I tested it before I

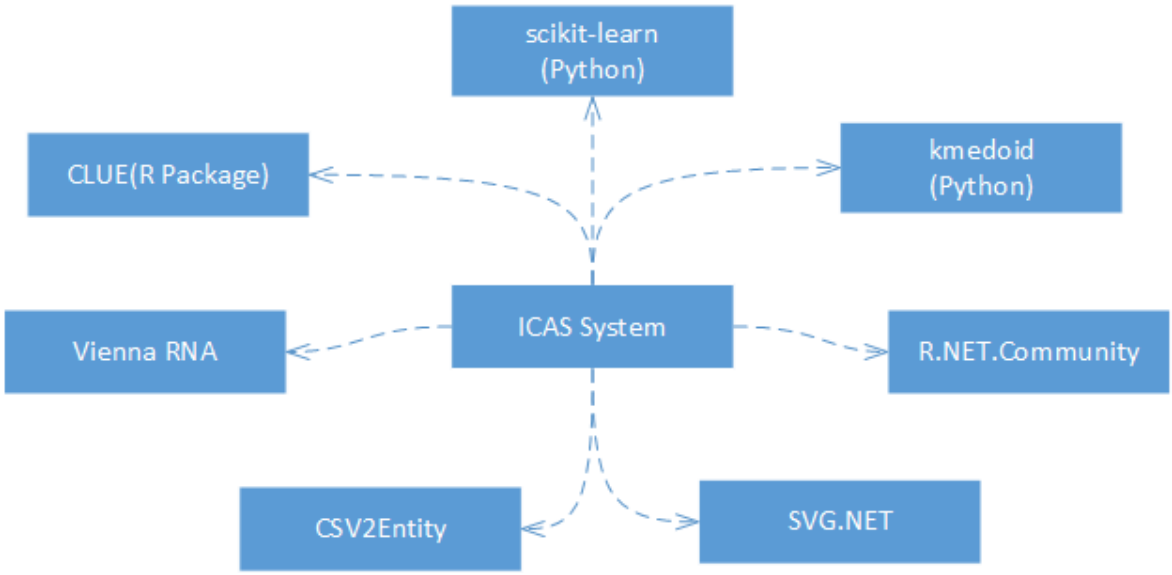

Figure 3.3: External dependency

used it. (See Section 6.3.1).

SVG.NET is an image format converting tool. I tested it by comparing the two formats with the same cleavage site. The images looked identical, so this package is credible.

R.NET.Community is a wrapper to R. I compared its results with the ones generated by R Studio. They were identical. Hence I have confidence in it.

CSV2Entity (Zhou, 2011) was built by myself and I used it in my previous projects. I have my faith in it.

3.2 Metrics

The purpose of this study is to find a clustering in which each cluster is significantly different from others. To measure clusterings with this aim, I introduce a new metric, significant rate. Two clusterings can have the same SR. Hence compactness was also used as the secondary metric to determine the best clustering.

The internal agreement of clusterings was also tested in this study with the Euclidean method. The agreement was used to check how stable the algorithms were.

For each algorithm, the agreement test was applied to test whether the algorithm gave different clusterings each time with the same parameters. The result was used to decide whether to repeat the algorithm with the same parameters.

3.2.1 Significant Rate

The ideal result of this study is that every cluster is significantly different from other gourps/clusters by its average. I defined significant rate between 0 and 100. The significant rate function is defined as the following.

$$r = \frac{s}{p} * 100 \tag{3.2.1}$$

Where p is the possible combinations of any 2 clusters from n clusters, $\binom{n}{2}$. Thus the significant rate function is defined as the following.

$$r = \frac{s}{\binom{n}{2}} * 100 \tag{3.2.2}$$

s is calculated by pairwise T test. Each cluster is to be compared with all other clusters and the p-value of the comparisons are given. The p-value adjustment is *Holm*. If the p-value is smaller than 0.05, I consider the two clusters are significantly different from each other by average. Pairs meet that criteria are counted as s.

Clusterings with significant rate 100 are referred to as SR 100 clusterings or clusterings which meet the metrics in the following content.

3.2.2 Compactness

Compactness is the secondary metric in deciding the best clustering. It measures the average pairwise distances between points in the same cluster. Given N data points, k

clusters, each cluster has n_k data points, the compactness function is written as follow:

$$Compactness = \frac{1}{N}\sum_{k=1}^{K} n_k(\frac{\sum_{i,j\in k} d(x_i, x_k)}{n_k(n_k-1)/2}) \tag{3.2.3}$$

Where $d(x_i, x_j)$ is the distance between x_i and x_j. For the reactivity features, this paper uses Euclidean distance. For the structure data, the distance matrix is precomputed in the data pre-processing phase and was used as the distance in the function above directly.

3.2.3 Agreement test with Euclidean method

This study used the Euclidean method for agreement test. Equation 3.2.4 gives the function.

$$1 - \frac{d}{m} \tag{3.2.4}$$

In this equation, d is the Euclidean dissimilarity of the memberships, i.e., the square root of the minimal sum of the squared differences of u and all column permutations of v, and m is an upper bound for the maximal Euclidean dissimilarity (Dimitriadou et al., 2002).

In this paper, the agreement test with the Euclidean method will be simply referred to as the agreement test.

3.3 Summary

In this study, Integrated Clustering Analysis System was developed to facilitate the clustering analysis. Significant rate and compactness were used to evaluate the clustering results.

Chapter 4

Data Pre-processing

This chapter describes the steps to pre-process the raw data. There are three types of raw data, namely the RNA data(mRNA and miRNA), the degradome data and the reactivity data. These raw datasets need to be pre-processed before carrying out further analysis. The basic tasks in pre-processing include predicting the cleavage site, generating the reactivity and cleavage efficiency of the cleavage site, calculating the structure and distance matrix.

4.1 Workflow

Figure 4.1 shows the data pre-processing details. The raw data is in orange, and the final data for clustering and reporting in green. The following sections give detailed explanation of the workflow.

4.2 Raw data

Twelve data files in the three types were provided by the JIC, and their basic information is given in Table 4.1.

The first two files store gene sequences of Deoxyribonucleic acids (DNAs) and RNAs, which are polymeric molecules assembled as a chain of nucleotides. In RNAs, the nucleotides are Adenines, Uracils, Cytosines, and Guanines. Usually, they are denoted

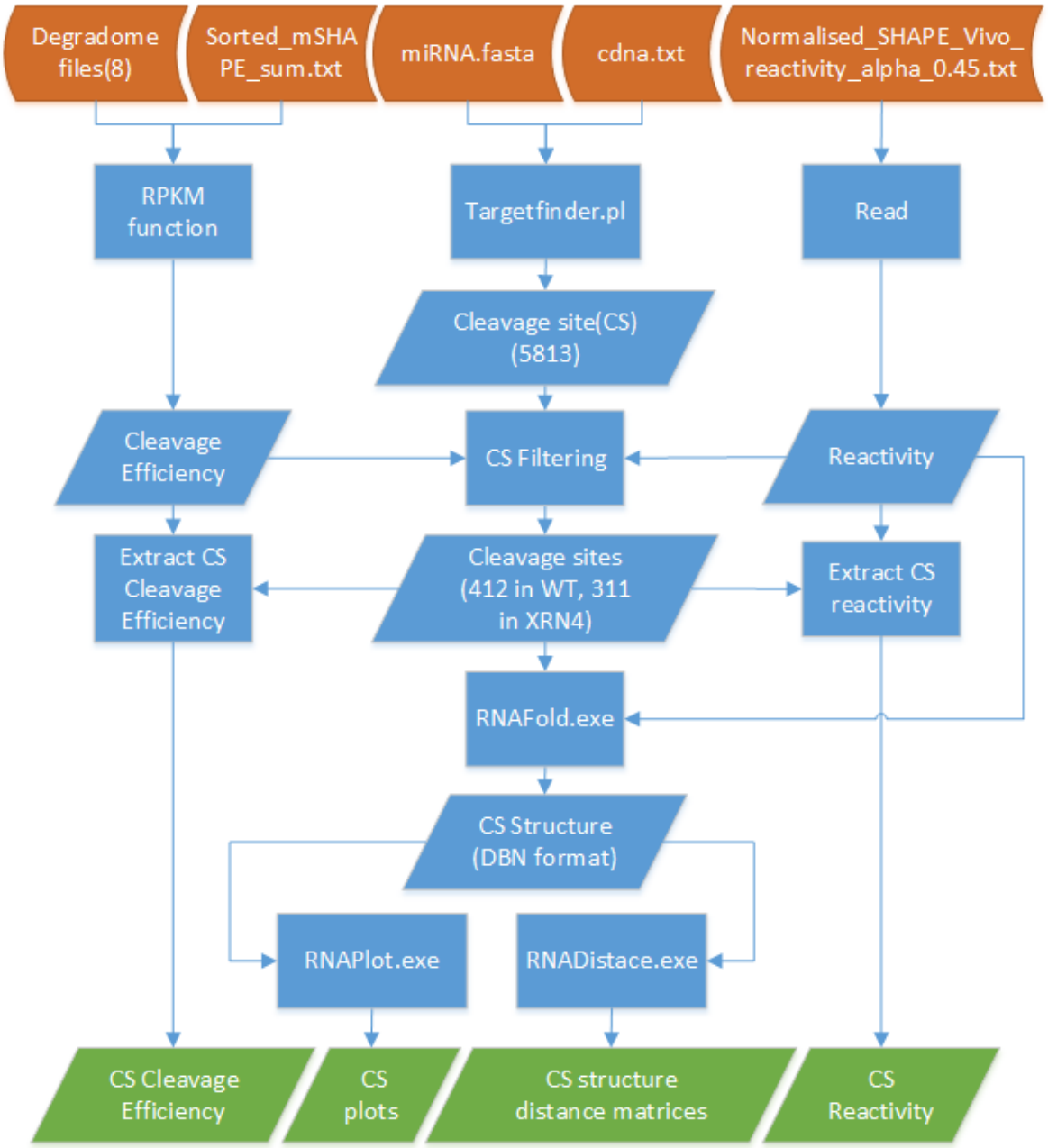

Figure 4.1: Data pre-processing workflow

Table 4.1: Raw data from the JIC

No.	File	Comments	Size	Records
1	miRNA.fasta	micro RNA for Arabidopsis thaliana	64k	1530
2	cdna.txt	Arabidopsis thaliana genes	70m	41667
3	normalized_SHAPE _vivo_reactivity _alpha_0.45.txt	Reactivity file	177m	20095
4-11	Degradome files	4 for WT, 4 for XRN4	964m	289879
12	sorted_mSHAPE_sum.txt	base file for cleavage efficiency	258m	40819

as As, Us, Cs and Gs. In DNA, the Uracil (U) is replaced by Thymine (T). Thus, an RNA/DNA molecule is presented as a string of these four letters in computer files.

Complementary DNA (cDNA) and miRNA data files are structured with FASTA format. In a FASTA file, a record is stored in a name line followed by some values lines. The name line starts with a larger-than sign (>) followed by the name and other information. The value lines represent the sequence. The record ends until it meets the start of another record or the end of the file. For the cDNA file, the first line is the chromosome name following the larger-than sign, followed lines of sequences. A sequence can be pages long.

The rest files store degradome and reactivity data. They use the same file format. Every line is a record that starts with the gene name followed by readings of every nucleotide in that gene. Each of the file could be explained as a ragged array, because the length of gene varies.

4.3 Degradome and cleavage efficiency

Degradome sequencing (Degradome-Seq) is also referred to as parallel analysis of RNA ends (PARE) (German et al., 2008; Addo-Quaye et al., 2008). It has been used to identify miRNA cleavage sites (Thomson et al., 2011).

4.3.1 Degradome combination

The degradome files are of two gene expressions, WT and XRN4. Each file represented an independent experiment, and might encounter some errors or bias. Combining the repeats together could reduce them. From statistic point of view, the sample size was enlarged, which gave the results more statistical power.

The eight degradome files were combined into only two in two steps. The first step was to combine degradome files of the same biological repeat but different technical repeats. The last step was to combine degradome files of the different biological repeats.

The first two files combined together were Degrad_wt_1_ATCACG_R1_T1 and Degrad_wt_1_ATCACG_R1_T2. Figure 4.2 shows the correlation of the two. Firstly, the reads of every gene was summed up into two dictionaries. The key was gene name, and the value read. The two dictionaries were sorted by their names. Genes existed in only one dictionary was filtered out. The values of the two dictionaries were extracted into two numeric array for Pearson's R test. The coefficient was 0.9983, indicating the two array (or degradome files) were strongly positively correlated. The p-value was smaller than 0.01, suggesting that the correlation was significant. Therefore, I was confident to combine the two degradome files. The same steps were applied to combine other files. In each of the steps, the coefficient was always larger than 0.95, and the p-value smaller than 0.01.

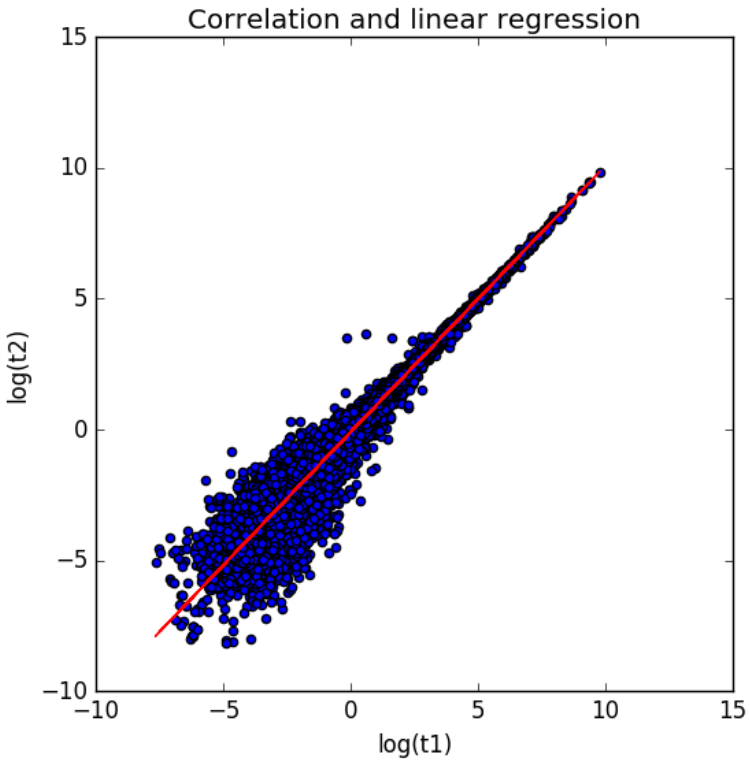

Figure 4.2: Correlation (blue dots) and linear regression (red line) between technical repeat 1 and technical repeat (t2) of WT biological repeat 1

Table 4.2 summarised the combination. Gene counts were in parentheses.

Table 4.2: Combination of degradome files

Initial Files	Step1	Step2
Degrad_wt_1_ATCACG_R1_T1(36611)	Degrad_wt_1 (38336)	Degrad_wt (39542)
Degrad_wt_1_ATCACG_R1_T2(37298)		
Degrad_wt_2_ACAGTG_R1_T1(34193)	Degrad_wt_2 (37247)	
Degrad_wt_2_ACAGTG_R1_T2(36074)		
Degrad_XRN4_1_GAGTGG_R1_T1(35763)	Degrad_XRN4_1 (38129)	Degradome_xrn4 (39883)
Degrad_XRN4_1_GAGTGG_R1_T2(36686)		
Degrad_XRN4_2_ATTCCT_R1_T1(36225)	Degrad_XRN4_2 (38380)	
Degrad_XRN4_2_ATTCCT_R1_T2(37030)		

4.3.2 From degradome to cleavage efficiency

The degradome data is the cleavage count. This data cannot be used as cleavage efficiency directly. I used Reads Per Kilobase of transcript per Million mapped reads (RPKM) as cleavage efficiency. RPKM is frequently used to measure mRNA abundance based on RNA-seq. It is calculated from the number of reads mapped to a particular gene region g, r_g, and the feature length, l_g , which is the number of nucleotides in a mappable region of a gene (Mortazavi et al., 2008). The function is defined as:

$$RPKM_g = \frac{r_g * 10^9}{l_g * R} \tag{4.3.1}$$

where R is the total number of reads from the sequencing run of that sample.

Another file (sorted_mSHAPE_sum.txt) was used to calculate the abundance of gene. The regional reads were from two degradome files, namely degradome_xrn4.txt and degradome_wt.txt. Thus the cleavage efficiency function could be presented as follows:

$$E = \frac{r_{nt}}{l_g * R_g} \tag{4.3.2}$$

Table 4.3 explains the variable definition of the function.

Shapiro-test was performed against the cleavage efficiency. The cleavage efficiency was not normally distributed. Logarithm transformation was then used to improve

Table 4.3: Parameters of cleavage efficiency function

Variable	definition
E	Cleavage Efficiency
r_{nt}	Reads of the nucleotide, from degradome_wt.txt or degradome_xrn4.txt
l_g	Gene length, from sorted_mSHAPE_sum.txt
R_g	Gene reads, adding up every nucleotide, from sorted_mSHAPE_sum.txt

normality. Figure 4.3 demonstrates the distribution. Shapiro-test p-values were printed in the title.

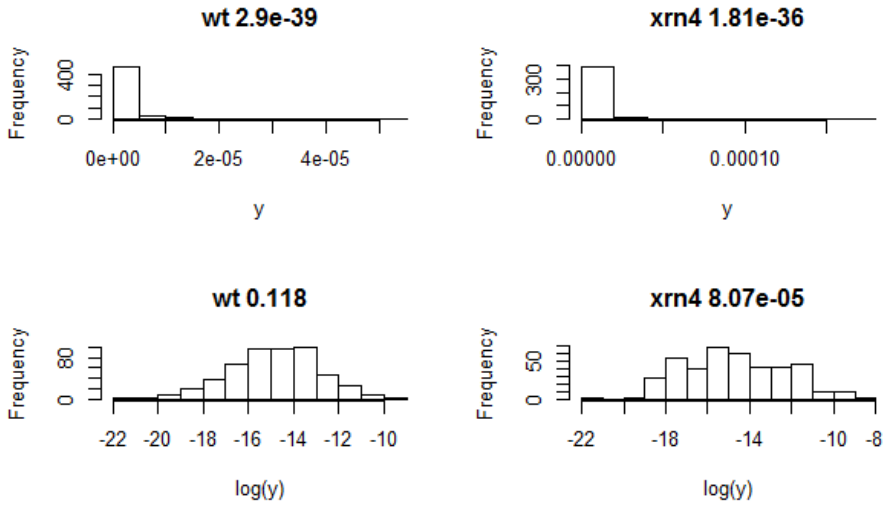

Figure 4.3: Cleavage efficiency distribution

From the Shapiro-test results, the wild-type cleavage efficiency after logarithm is considered as normally distributed. The XRN4 cleavage efficiency also became more normally distributed after logarithm transformation. The logarithm-transformed cleavage efficiency was used through this study.

4.4 Cleavage site prediction

Two files were used in cleavage site prediction, namely the miRNA file (miRNA.fasta) and the gene file (cdna.txt). Before the prediction, artificial miRNA in the miRNA.fasta file had been removed.

Srivastava et al. (2014) compared the performance of popular tools for cleavage site prediction against Arabidopsis thaliana. Table 2.1 demonstrates the results.

This study emphasised precision. False cleavage sites would lead to false conclusions while missing some of the true ones might not be so devastating. Targetfinder(Fahlgren et al., 2007) and Tapir/fasta (Bonnet et al., 2010) both stood as the top with precision 0.89. However, Targetfinder outperformed Tapir/fasta in terms of recall. Thus, Targetfinder was chosen for the prediction.

4.4.1 Cleavage site filtering

Targetfinder found 5832 cleavage sites. However, most of them were not to be used in this study. They had to meet some criteria. Firstly, these cleavage sites must be connected with cleavage efficiency and reactivity. The key links them together is the gene name. The same gene must be found in the cleavage site files, the degradome(efficiency) files and the reactivity files. There were 40739, 38444 and 20095 genes in the three respectively. The intersection of them gave 2704 genes. Cleavage sites with a gene name that cannot be found in the 2704 genes were ruled out and as a result, 3270 were left. After that, I combined the cleavage sites with the same gene and start/end points but with different miRNA. 3102 cleavage sites were left. The other factor that I must have taken into consideration was whether I could extend the cleavage sites to both end with 50 nucleotides. That meant if the starting position was smaller than 50, I would have to ignore the cleavage site and I would also remove the ones that could not extend to the left end of the gene sequence. Consequently, 2633 were left concerning extension.

Now, half of the cleavage sites had been abridged, but this was not the end. The other most important factor to be considered was the cleavage efficiency. Cleavage sites

with a zero cleavage efficiency were removed. At last, 412 cleavage sites were found in the WT dataset, and 311 in the XRN4 dataset.

Chart 4.4 demonstrates the cleavage site filtering process for the WT degradome. The process for XRN4 data differed only in the last step.

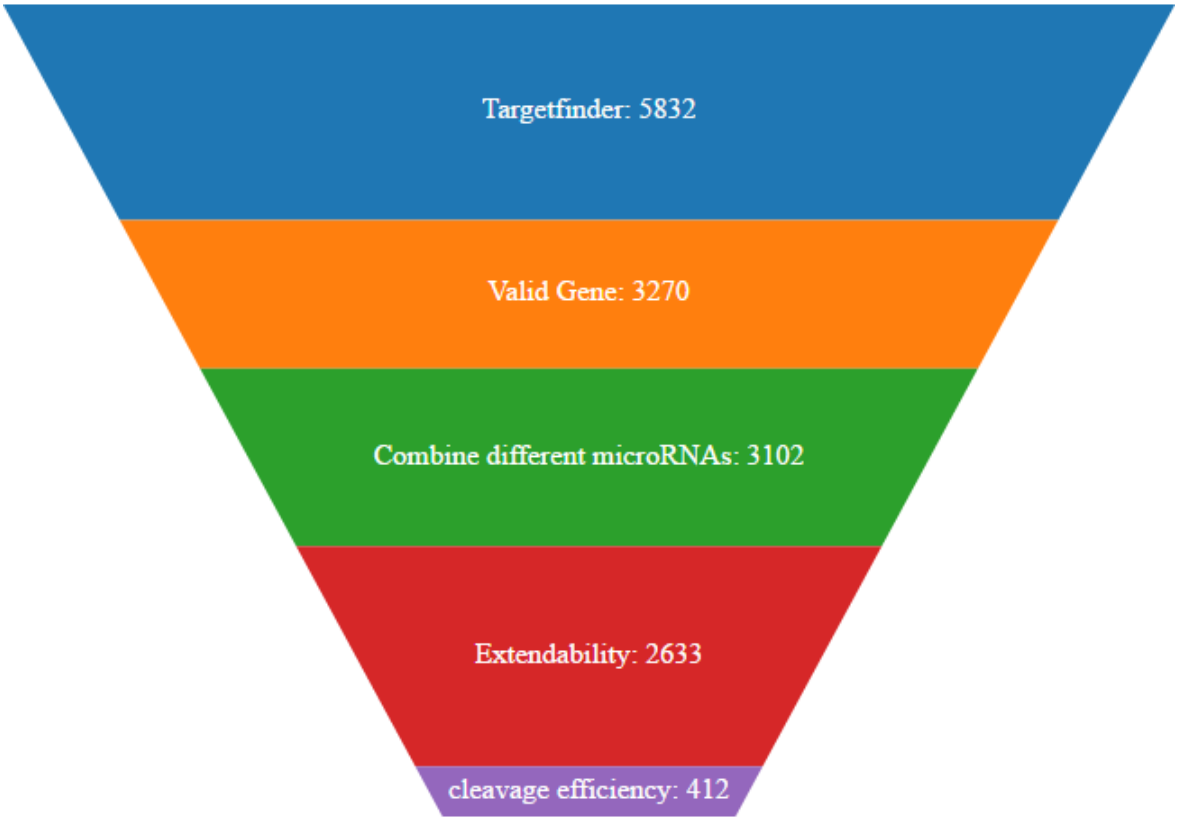

Figure 4.4: Cleavage site filter(Wild Type)

4.5 Reactivity and cleavage efficiency of the cleavage site

Reactivity is the probability for certain nucleotide to be modified by chemicals, such as dicarbomethoxyacetylene (DMA) or Selective 2-hydroxyl acylation analyzed by primer extension (SHAPE). Higher the reactivity is, the more possible it is modified and more possible it is in single strand.

I used the reactivity of every nucleotide in the cleavage site as the feature for clustering. Firstly, the reactivity of the 21 nucleotides in the cleavage site were generated. Then, they were extended symmetrically to 71 and 121 nucleotides. Therefore, there were 21, 71 or 121 features for clustering.

Cleavage efficiency is another important property for the cleavage site. I generated cleavage efficiency for every nucleotide in the degradome pre-processing. For the cleavage site, the cleavage efficiency was the summary of cleavage efficiency of the 10th and 11th nucleotides.

4.6 Structure computation and representation

RNAFold, RNADistance and RNAPlot in ViennaRNA 2.0 (Lorenz et al., 2011) were used for the three sub-tasks respectively. In the first step, ICAS generated the sequence and reactivity for every cleavage site. Then ICAS used RNAFold for the DBN. In the second step, the ICAS used RNADistance to compute the distance between every DBN string pair. The results were put into matrices. Finally, the structure plots were generated from the dot-bracket strings using RNAPlot in ViennaRNA.

4.7 Implementation of pre-processing program and GUI

Based on the literature review, I found there are some software packages, such as Targetfinder and RNA distance, that can be used in this work, but not for the pre-processing tasks involving degredome combination, cleavage efficiency and reactivity extraction for the cleavage site. I then had to implement some programs myself with Python and C#. In addition these programs need to be connected into a system to do

all the preprocessing tasks in a logical sequence, therefore, I designed a GUI.

4.7.1 Data pre-processing GUI

The data pre-processing GUI does most of the pre-processing tasks.

There are several tools under the *Tools* menu. One is the Targetfinder file converter. It converts combines every file generated by Targetfinder in one folder to a single Comma-separated values (CSV) file. Figure 4.5 demonstrates that.

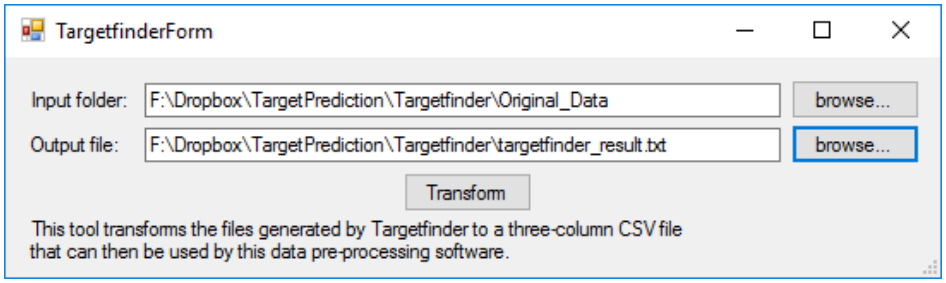

Figure 4.5: Targetfinder file converter

Another useful tool is the degradome merger. It firstly calculates the correlation of two degradome files, and then merge them into one single file. Figure 4.6 displays the merger.

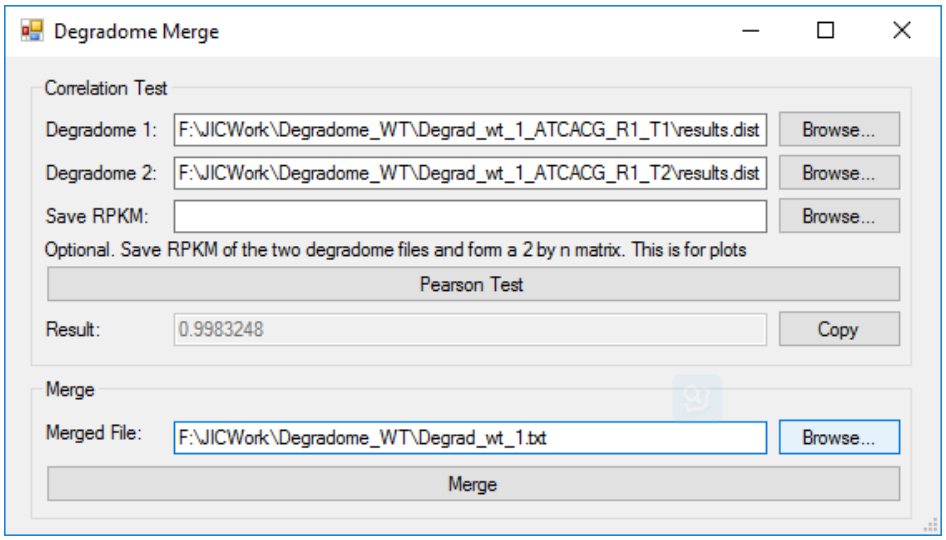

Figure 4.6: Degradome merger

After the cleavage sites and efficiency are ready, the user can just use the one-click pre-process function to generate all the files needed for data analysis. or go to the steps

menu to do them one by one.

This application divided deata pre-process into 10 consecutive steps. And the step to compute cleavage site distances will be implemented in the future.

Step 1 To get the valid genes. In this study, the genes in different data are different. There are cleavage site files, degradome files and reactivity files. If one gene appears in one file but not other, it is useless. This step is simple. It generates three files with genes appeared the three types of files, and then take the intersection of the three. The final file contains genes that appear everywhere.

Step 2 To convert the degradome to cleavage efficiency. It also stores the data dictionaries, because most of values are zero. A dictionary saves a great number of computer space.

Step 3 To convert the reactivity array to dictionary to save computer space.

Step 4 To read the genes and serialise them into a file.

Step 5 To filter the cleavage sites.

Step 6 To generate related files of the cleavage site. They are the cleavage efficiency of the cleavage site and the reactivity of cleavage site.

Step 7 To generate the average reactivity.

Step 8 To compute the structure of each cleavage site and represent the structure with dot-bracket format.

Step 9 To plot cleavage site structures.

Step 10 To calculate the distance between cleavage site structures (DBN).

Figure 4.7 gives the GUI of steps.

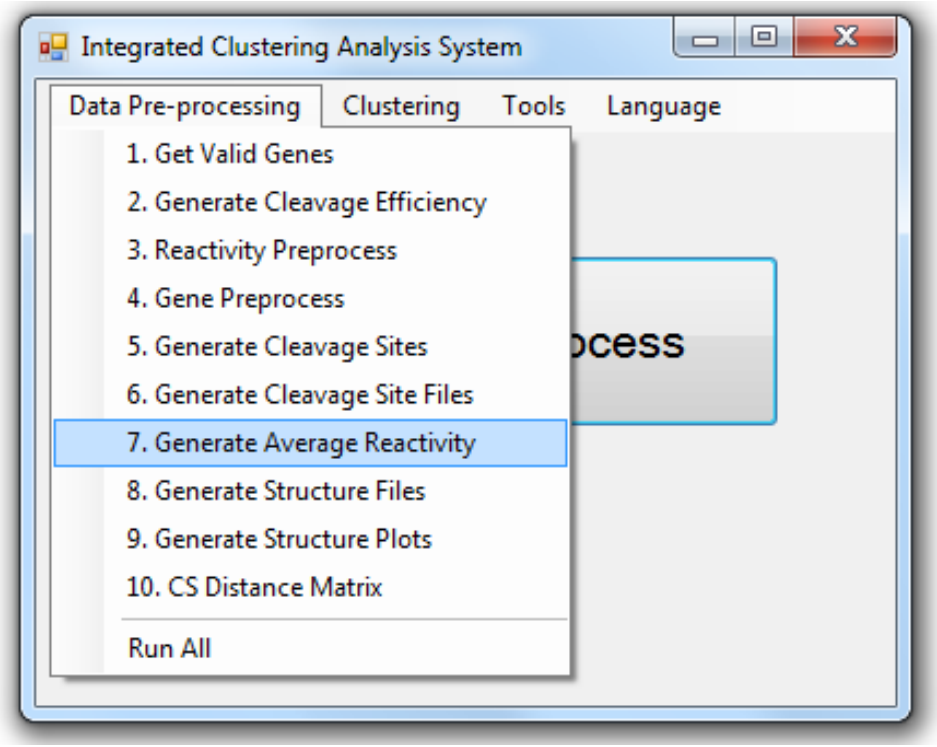

Figure 4.7: Data pre-process steps

4.8 Results

As a result, cleavage efficiency, plots, structure distance matrices and reactivity for cleavage sites were generated. The last two were datasets for clustering. These datasets had three dimensions, namely feature, degradome and cleavage site length. Figure 4.8 provides their hierarchical view.

4.8.1 Dataset naming convention

As in this stage, there were already ten datasets. It would be tedious referencing each of the datasets by description. Thus, it was necessary to name them. The pattern for the dataset name was Degradome_CSLength_FeatureType. Table 4.4 lists three of the datasets with their names. The other seven dataset names followed the same pattern.

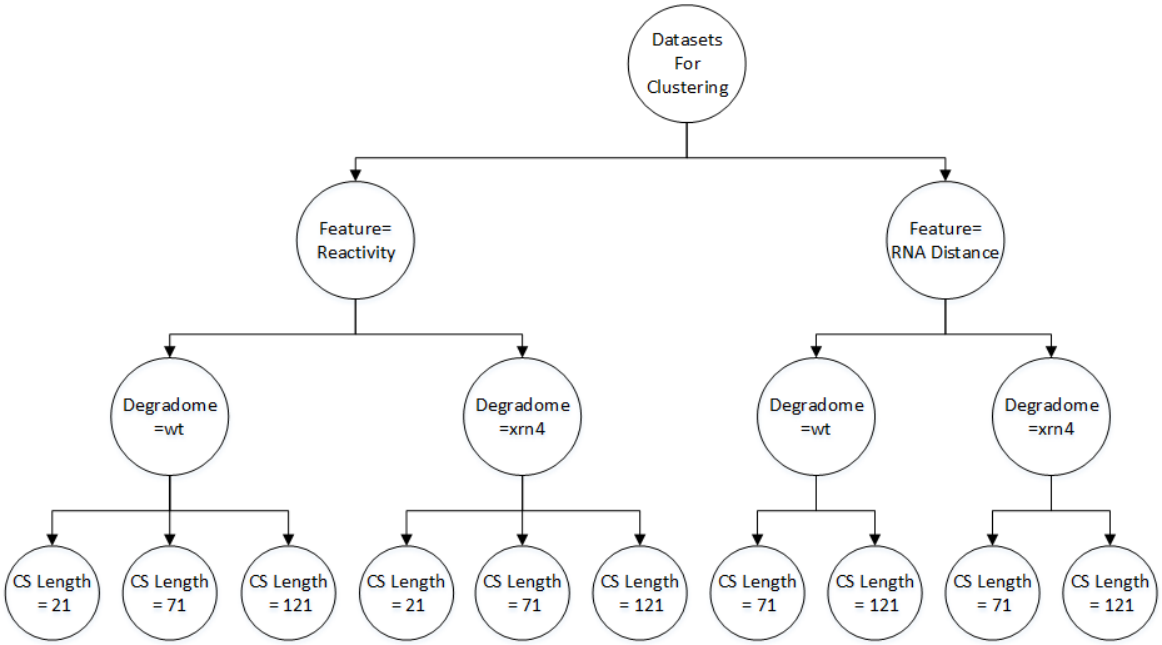

Figure 4.8: Data hierarchical view

Table 4.4: Dataset naming convention

Name	**Degradome**	**cleavage site Length**	**Reactivity/ RNA Distance**
WT_71_Reactivity	WT	71	Reactivity
WT_71_Distance	WT	71	RNA Distance
XRN4_121_Reactivity	XRN4	121	Reactivity

4.9 Summary

I searched for the best tool for cleavage site prediction. I tried some of them, and at last, I used Targetfinder. 5832 cleavage sites were found but not all of them met the requirement of this study. After filtering, only 311 were left for the WT degradome and 412 for XRN4.

I combined the degradome datasets, converted the degradome data to cleavage efficiency data, and finally calculated the cleavage efficiency for every cleavage site.

Six reactivity datasets for cleavage sites with length 21, 71 and 121 were generated. They were the features for clustering.

Cleavage site structures were generated, followed by four distance matrices for clustering and the graphical structures for reporting. The graphical cleavage site structures

were also generated.

Finally, I developed a system that does the above all together.

Chapter 5

Clustering by reactivity

5.1 Introduction

As designed in the research methodology and procedure described in Chapter 3 clustering of the cleavage sites will be carried out firstly by reactivity and then by structure. This chapter focuses on clustering by reactivity. I used four clustering algorithms: k-means, mean shift, affinity propagation and spectral clustering. I attuned the parameters of the algorithms and ran the algorithm with the same parameters a hundred times to generate sufficient clusterings. I generated datasets by feature selection using PCA, variance and linear regression. Linear regression was not processed further due to low r-squared values. The rest datasets along with the original datasets were fully tested with each of the four clustering algorithms.

There is no prior research found using reactivity for cleavage site clustering analysis.

5.2 Clustering with four algorithms

In each of the four clustering algorithms, I attempted to cluster the data into 3 to 15 clusters by changing the parameters. For k-means and spectral clustering, k or the cluster count was given directly. For affinity propagation and mean shift, I changed damping or bandwidth to limit the cluster number from 3 to 15. Other parameters were changed in a range, too. Listing 5.1 provides the pseudo code of the implementation.

Listing 5.1: Pseudo code for clustering algorithm implementation

```
//load reactivity data from original datasets, pca datasets
   or datasets selected by variance.
data = loadData(...)

for parameter1 in ["a","b","c"...]:
    for parameter2 in ["d","e","f"...]: #
        //apply to k-means and spectral clustering
        for k in [3,4,5, ... 15]: #k from 3 to 15
                //for k-means and spectral clustering
            //I run 100 times
            for round in [0,1,2 ... 99]:
                clusterer = new Clusterer(parameter1,
                   parameter2, k ...)
                clusterer.train(data)
                save(clusterer.labels,...)
```

For k-means and spectral clustering, I repeated the same combination of variables 100 times because every time the result was different. For affinity propagation and mean shift, I did not repeat because these algorithms were stable and the result was constant. Agreement by Euclidean method (Introduced in Section 3.2.3) had been used to test if the results were identical.

5.2.1 Select K

Given the fact that there is no prior knowledge to select the value of k, the cluster count was chosen from 2 to 15.

5.2.2 K-means

The initial centres of the clusters are randomly generated in k-means algorithms. That means every time this algorithm might generate different result(clusters). *Init* (Method for initialisation) was chosen between k-means++ and random. That means for each k, there would be 200 repeats.

5.2.3 Mean shift

The bandwidth(Radius) is the most important parameter in Mean-shift algorithm. The cluster number decreases when the bandwidth increases. Thus it is important to analyse it when using mean shift. I shrank or amplified the default bandwidth from 0.5 to 2 times. Table 5.1 demonstrates the results.

Table 5.1: Bandwidth in mean shift

Degradome	CS Length	Bandwidth transformation(Times)						
		0.5	0.75	1	1.25	1.5	1.75	2
wt	21	184	113	7	3	1	1	1
	71	250	154	58	3	1	1	1
	121	258	163	53	3	1	1	1
xrn4	21	146	82	40	7	1	1	1
	71	186	112	50	9	1	1	1
	121	196	124	42	6	1	1	1

To control k between 3 and 15, the bandwidth was set to be between 1 and 1.5 times of its default value and the interval was set to 0.001.

5.2.4 Affinity propagation

I tested the six reactivity datasets with damping from 0.5 to 1 at interval 0.1. There were 55 to 85 clusters in each clustering. K was too large in this algorithm. However, I ran this algorithm nonetheless and no clusters meet the metrics was found.

5.2.5 Spectral clustering

In this study, I used normalised cuts algorithm or Shi-Malik algorithm (Shi and Malik, 1997). The strategy to use to assign labels in the embedding space or $Assign_labels$ was chosen between *kmeans* and *discretize*. That means for each k, there would be 200 repeats.

5.3 Dimensionality reduction

Dimensionality reduction is the process of reducing the number of random variables under consideration via obtaining a set of principal variables (Roweis and Saul, 2000). In some cases, data analysis can be done in the reduced space more accurately than in the original space.

5.3.1 Principal component analysis (PCA)

PCA was invented by Peason (1901). It reduces the feature numbers to a set of principal components, which are linear combinations of the original features.

The PCAs were selected when their Eigenvalues were higher than 1.0. Table 5.2 displays the features count after PCA.

Table 5.2: Features count after PCA

Degradome	**CS Length**		
	21	**71**	**121**
WT	8	24	42
XRN4	7	24	42

Datasets generated by the PCA algorithms were suffixed with _PCA.

5.3.2 Feature selection by variance

Variance is the expectation of the squared deviation of a random variable from its mean. The more variant the feature is, the more influential the feature is. In contrast

to PCA, this method keeps the original features instead of creating new ones. It is easy to interpret the results.

I plotted the variance for the WT and XRN4 datasets. (Figure 5.1 and 5.2) Interestingly, both reported the reactivity of the 5th nucleotide in the cleavage site as the most variant one among all 121 nucleotides inside or near the cleavage site.

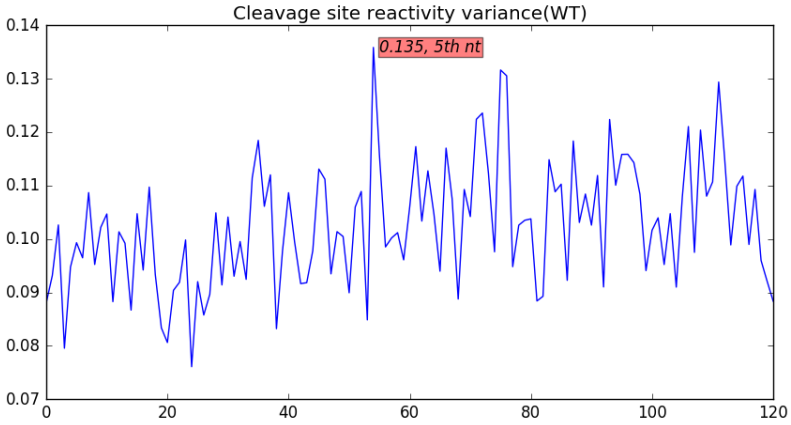

Figure 5.1: Cleavage site reactivity variance(WT)

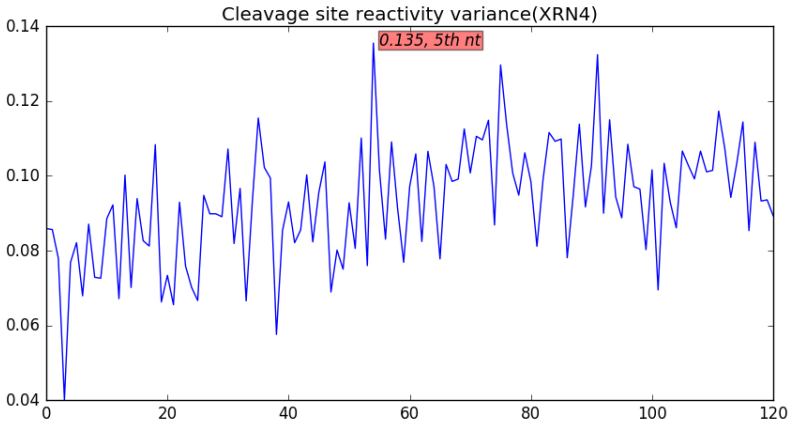

Figure 5.2: Cleavage site reactivity variance(XRN4)

The median is around 0.1 (0.103 for WT, 0.094 for XRN4). Thus, 0.1 was chosen as the threshold to cut off features. Features with a variance lower than 0.1 were discarded. The selected features are demonstrated in the Table 5.3.

Table 5.3: Features count after feature selection by variance

Degradome	CS Length		
	21	**71**	**121**
WT	15	45	68
XRN4	9	28	43

5.3.3 Feature selection by Linear Regression

Similar to the previous method, this method also keeps some original features and the clustering results will be easy to interpret.

In the linear regression, the independent value X was the reactivity of the cleavage sites while the dependent value y was the cleavage efficiency.

I tried different linear models, including ordinary least squares, ridge regression, lasso, orthogonal matching, Bayesian ridge regression. The ordinary least squares algorithm achieved the highest r squared values among all of them. Thus this algorithm was chosen.

Table 5.4 shows the r-squared values of different datasets. The r-squared values increased when there were more features. This could be seen as overfitting. Considering there were only 300 or 400 data points in each dataset, 121 features could be too many.

Table 5.4: Linear regression(ordinary least squares) r-squared values

Degradome	CS Length		
	21	**71**	**121**
WT	0.074	0.244	0.369
XRN4	0.11	0.358	0.566

The p-values of the features were calculated and those with a p-value that were smaller than 0.1 were selected. Table 5.5 shows the feature count after selection.

Table 5.5: Features selected

Degradome	CS Length		
	21	**71**	**121**
WT	6	17	31
XRN4	5	17	23

Finally, the selected features were multiplied with their corresponding coefficients and six new datasets were formed. The new r-squared values became much smaller than the old one (Table 5.6). Therefore, this method was not processed further.

Table 5.6: R-squared values after feature selection

Degradome	**CS Length**		
	21	**71**	**121**
WT	0.055	0.108	0.170
XRN4	0.076	0.160	0.182

5.4 Results

Table 5.7 shows the clustering with significant rate 100 found in k-means and spectral clustering. No clustering with significant rate 100 in affinity propagation and mean shift was found.

Table 5.7: SR 100 clusterings by algorithms and dimensionality reduction

Algorithm	K-means						Spectral Clustering					
Degradome	WT			XRN4			WT			XRN4		
Length	21	71	121	21	71	121	21	71	121	21	71	121
Original	2	2	2	-	-	-	-	2	2	2	2	2
PCA	2	2	2	-	2	2, 3	2	2, 3	2, 3	-	2	2, 3
Variance	-	2	2	-	2	2	2	2	2	-	-	2

Algorithms with randomised initialisation (k-means and spectral clustering) were more likely to find clusterings with significant rate 100 whereas algorithms give the same result for the same parameters (affinity propagation and mean shift) were more likely to fail this task.

Before dimensionality reduction, ICAS did not find any SR 100 clusterings with more than 2 clusters. PCA found SR 100 clusterings with 3 clusters. Feature selection by variance also found SR 100 clusterings where the original datasets failed to find the same.

5.4.1 Report the best-performed clustering

Here I looked into the report of the best-performed clustering (Table 5.8).

Table 5.8: One of the best performed clusterings

Algorithm	K-means	**Significant Rate**	100
Dataset	WT_21	**K(Groups)**	2
Compactness	0.00013		

Figure 5.3 shows two-dimensional distribution of the data after PCA. The data points were well clustered.

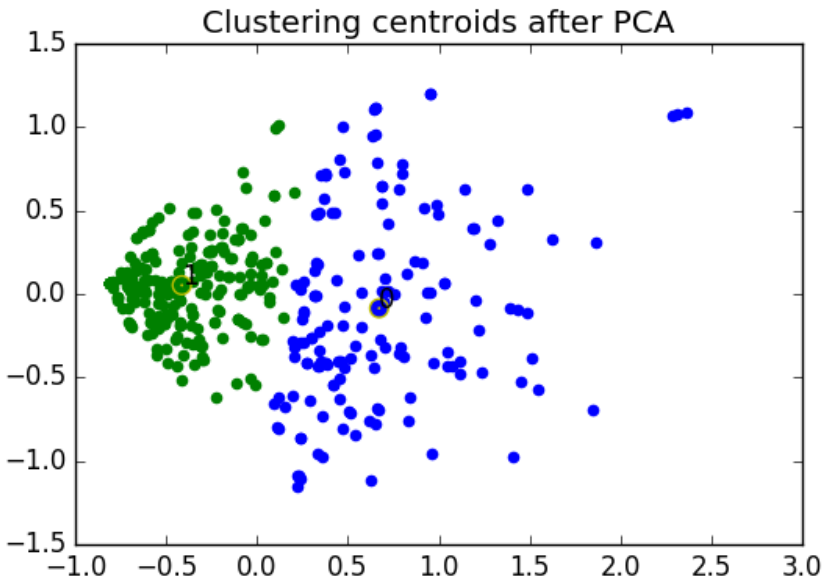

Figure 5.3: Two-dimensional distribution of the data after PCA

Figure 5.4 shows distribution of the data. The clusters were balanced.

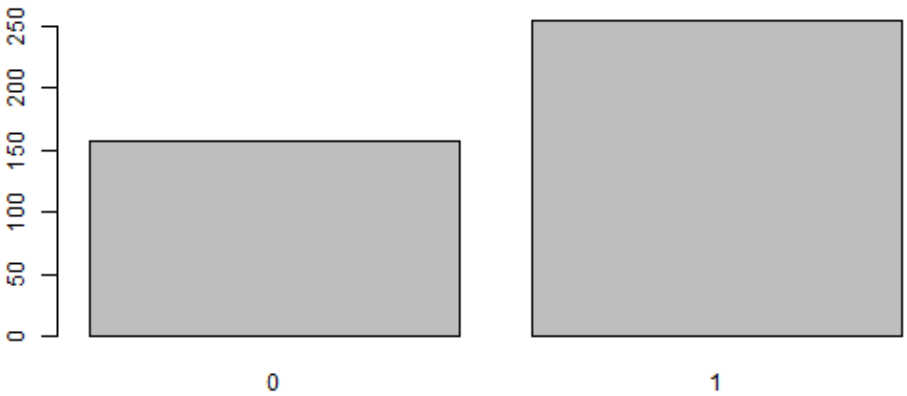

Figure 5.4: Two-dimensional distribution of the data after PCA

ICAS also provided the box plot (Figure 5.5), which clearly shows clusters with significantly different medians. The median logged cleavage efficiency for Group 0 and 1 were -14.84769 and -14.52951 respectively.

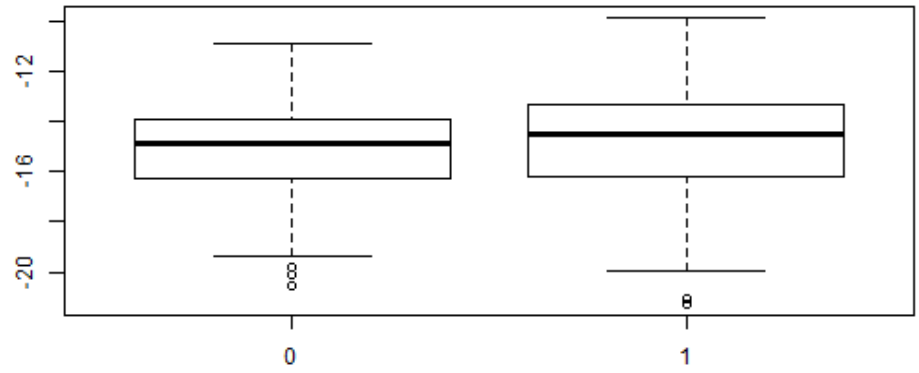

Figure 5.5: Box plot on average log cleavage efficiency

Figure 5.6 compares the average reactivity. The blue line represented the average reactivity of all 21 nucleotides. Compared this figure with Figure 5.5, the reactivity seemed to be negatively correlated with cleavage efficiency, Section 5.4.3 analyses it.

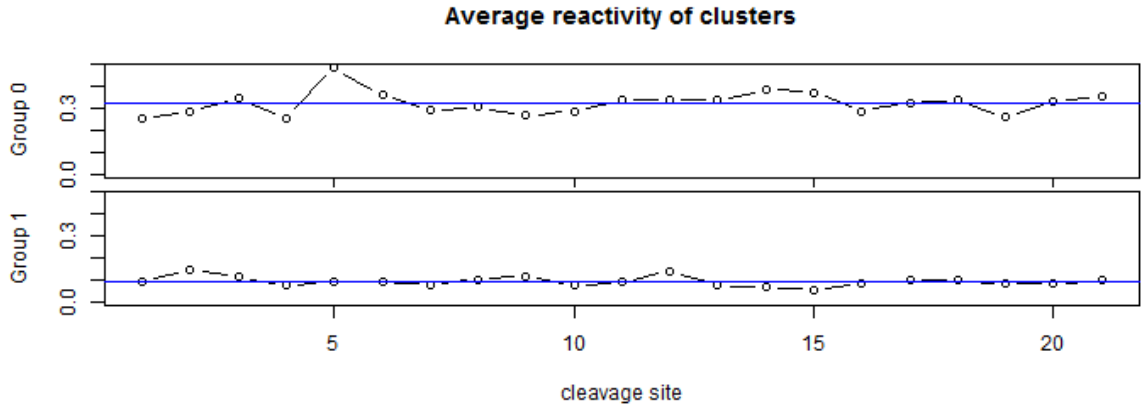

Figure 5.6: The average reactivity of clusters

5.4.2 Stability

I checked the stability of the best performed clustering mentioned in previous section. The agreement between clusterings with the same algorithms(K-means) and k were always 1. That means every time kmeans with k=2 generated the exact result. Thus, this k-means with k=2 was very stable and reliable.

I also checked the stability other SR 100 clusterings. Both k-means and spectral clustering were not stable.

5.4.3 Correlation between reactivity and cleavage efficiency

To test the correlation between reactivity and cleavage efficiency of the WT data, I summed up the reactivity of the same cleavage site such that I could do Pearson test. The null hypothesis was that the reactivity and cleavage efficiency were unrelated, whereas the alternative hypothesis was that they were correlated. PearsonâĂŹs correlation coefficient was -0.13, and p-value 0.0088. I rejected to null hypothesis. Reactivity and cleavage efficiency were significantly weekly negatively correlated.

I did the same with the XRN4 data. Pearson's correlation coefficient was 0.07, and p-value 0.20. I accept the null hypothesis that there was no correlation between reactivity and cleavage efficiency.

Because the conflict between the statistical results generated from the two degradome sequencing data, I could not conclude whether reactivity and cleavage efficiency were correlated or not, and whether they were positively or negatively correlated.

Figure 5.7 demonstrates the scatter plot and the linear regression of reactivity and cleavage efficiency of WT and XRN4. The data points were distant from the regression line. There were some outliers. It may change the correlation test results if the outliers are removed.

5.5 Implementation

Clustering algorithms were run in the HPC. For each algorithm, I created a job file revoking three scripts.

The first script was the specific algorithm in Python, such as k-mean script job_kmean.py,

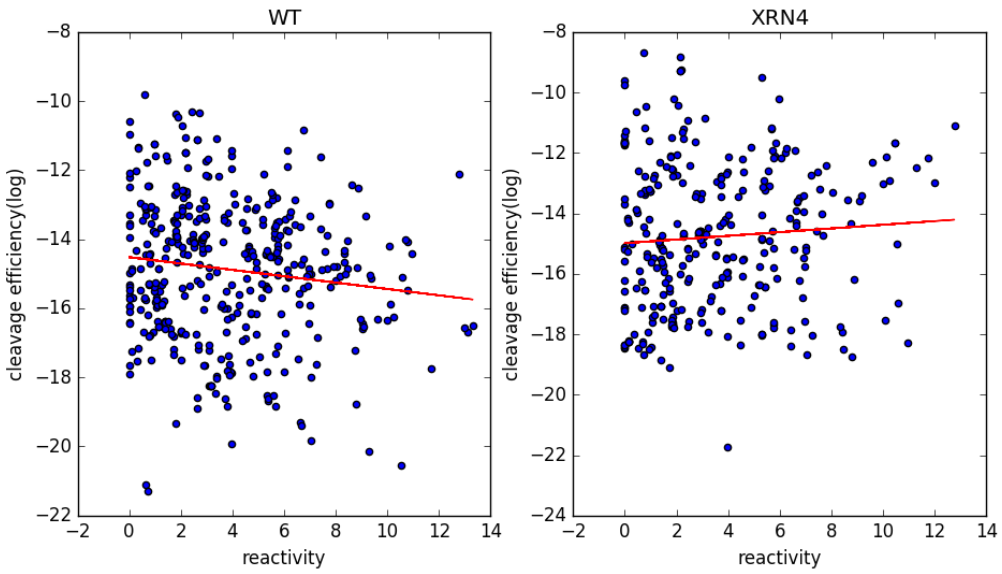

Figure 5.7: Scatter plot (blue dots) and the linear regression (red lines) of reactivity and cleavage efficiency of WT (left) and XRN4 (right)

which generated the clusterings including the labels, the centroids, the inertia and the Python kmean object. Thus each clustering consisted of four files. Multithreading was used to accelerate the script.

The second script was the *job_oneway.py* file. Just as its name implies, the script does the one-way Analysis of variance (ANOVA) test with clustering and the cleavage efficiency. An output file named oneway.csv would be created by oneway.py, listing all the candidates with a one-way ANOVA test p-value that are smaller than 0.05.

The last script was *job_stat.R*. Stat.R looked into the oneway.csv file and found out how many pairs were significantly different. The output file was *stat.csv*, listing results sorted by significant rate.

5.6 Summary

K-means, mean shift, affinity propagation and spectral clustering were used for the reactivity datasets which had been dimensionality reduced by PCA and variance. Some clusterings met the criteria were found. A very stable clustering was found when using

kmeans and k=2. However, when k was 3, the clusters were not balanced in terms of cluster size.

Dimensionality reduction helped to find more SR 100 clusterings, although they were not stable.

In conclusion, the best clustering was found with k-means when k=2.

Chapter 6

Clustering by structure

6.1 Introduction

In the last chapter, the work on clustering of the cleavage site data with reactivity was presented. This chapter focuses on clustering by distance matrices. It could be better if I extracted features from the structure and reuse the algorithms I used in the last chapter, but this requires in-depth biological knowledge and understanding of the structure which is beyond my scope. I then used RNA distance to produce distance matrix and performed clustering algorithms that use a distance matrix.

There is no prior research using cleavage site distance matrices for clustering. Admittedly, the results were not stable and reliable, and there are many to be improved.

The client abandoned this approach and informed me to switch back to reactivity after I had almost finished it.

6.2 The distance matrices

The distance matrices were generated during the data pre-processing stage(Section 4.6).

The average distance for the cleavage sites with 71 and 121 nucleotides are approximately 53 and 96 (Table 6.1).

Figure 6.1 and 6.2 provides the distribution of pair distances.

Table 6.1: RNA distance in cleavage sites

Degradome	cleavage site Length	Distance			
		Avg.	Median	Max	Sd
wt	71	53.2	52	104	12.7
xrn4	71	52.8	52	100	12.9
wt	121	95.7	96	172	16.7
xrn4	121	95.4	96	172	17.5

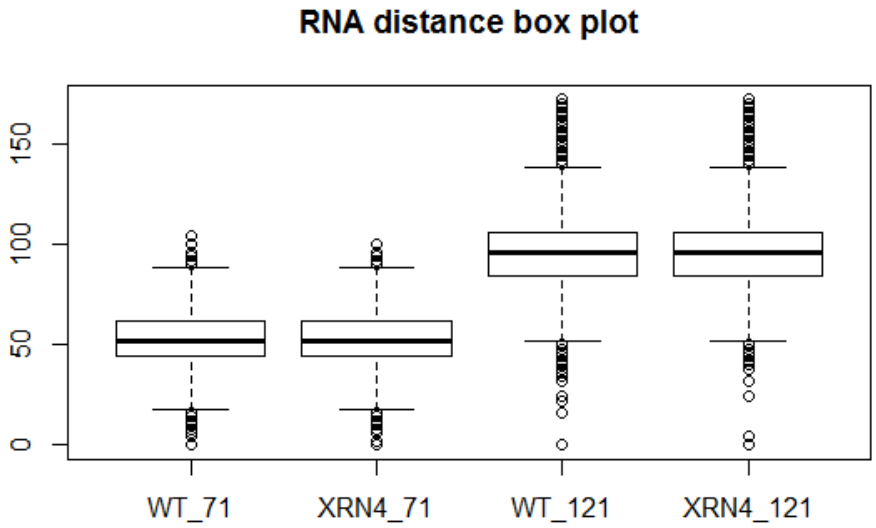

Figure 6.1: Cleavage site distance box plot

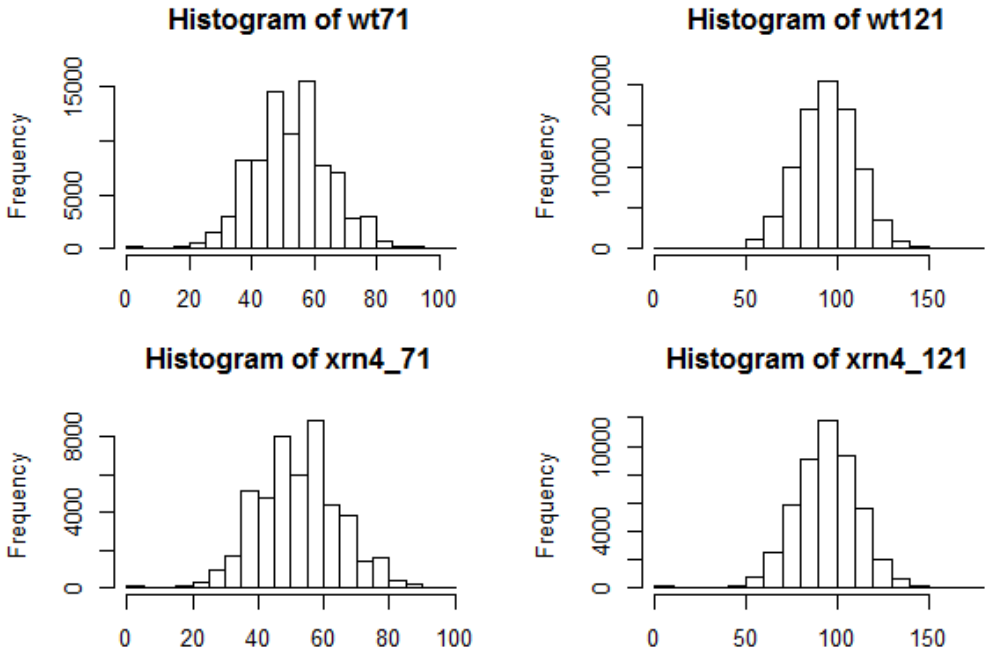

Figure 6.2: Cleavage site distance box plot

6.3 Clustering algorithms

The algorithms and implementation were similar to that introduced in Section 5.2. The difference was that k-means was replaced by k-medoids, mean shift by hierarchical

clustering, and the feature matrix by distance matrix. affinity propagation and hierarchical clustering generate the same results each time whereas k-medoids and spectral clustering produce changing ones.

6.3.1 K-medoids

K-medoids suited the RNA distance datasets very well. The centres are data points that exist in the dataset. Thus, it became the first choice for clustering by the RNA distance matrices.

Test against k-means

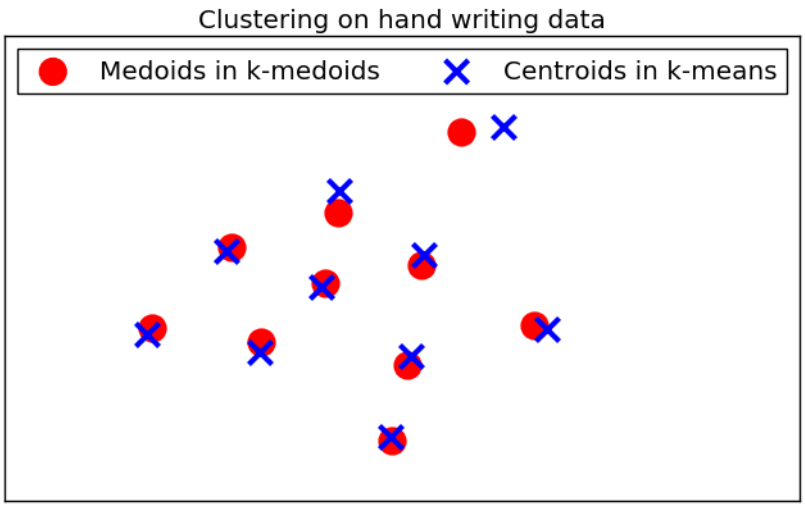

Figure 6.3: Hand writing clustering by k-means (red circle) and k-medoids (blue cross)

The k-medoids code was downloaded from the Internet (Alspaugh, 2011). It was tested against k-means to check whether they could generate similar centroids/medoids. I used a demo of K-Means clustering on the handwritten digits data from the scikit-learn official website. Figure 6.3 demonstrates the result. Nine out of ten centroids and medoids matched with each other. Thus, the k-medoids code was considered credible.

6.3.2 Hierarchical clustering

The hierarchical labelling was hard to use directly in this study. They need to be flattened. The hierarchy was flattened to k (2 to 15) clusters. Figure 6.4 demonstrates the steps.

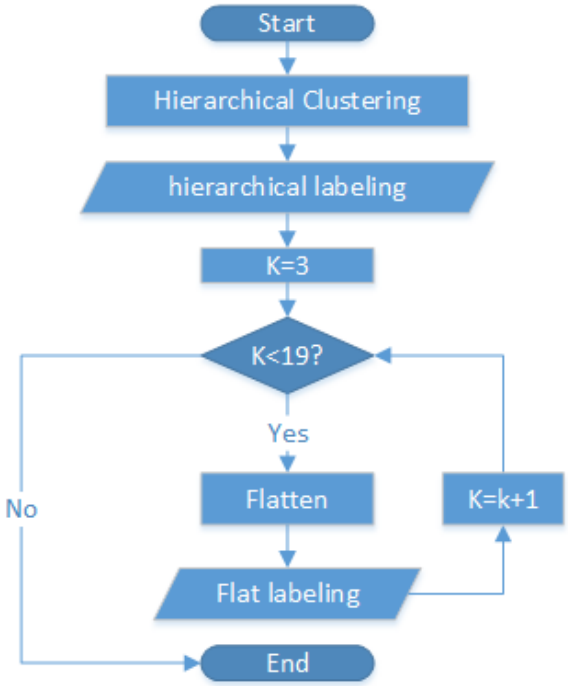

Figure 6.4: Hierarchical clustering workflow

No significant difference between the clusters in any clustering was found. I looked into the flat labelling and found they were unbalanced. Table 6.2 demonstrates the distribution with Dataset WT_71_Distance with k(cluster number) from 3 to 15. The other three datasets led to similar results.

The dendrograms were plotted for the hierarchical labelling for the cleavage site. The peak was always at the leftmost. Every time the hierarchy was cut, the leftmost with was cut off from the rest and the two parts were very unbalanced. Figure 6.5 demonstrates the dendrograms.

The data was unbalanced. If the dendrogram had had its peak in the middle, the labelling would have been more balanced.

Table 6.2: Distribution of clusters with Dataset WT_71_Distance

K	Distribution
2	411, 1
3	410, 1, 1
4	408, 1, 1, 2
5	402, 6, 1, 1, 2
6	396, 6, 6, 1, 1, 2
7	395, 6, 6, 1, 1, 2, 1
8	394, 6, 6, 1, 1, 2, 1, 1
9	393, 6, 6, 1, 1, 2, 1, 1, 1
10	392, 6, 6, 1, 1, 1, 2, 1, 1, 1
11	388, 6, 4, 6, 1, 1, 1, 2, 1, 1, 1
12	384, 6, 4, 4, 6, 1, 1, 1, 2, 1, 1, 1
13	383, 6, 4, 4, 6, 1, 1, 1, 1, 2, 1, 1, 1
14	382, 6, 4, 4, 6, 1, 1, 1, 1, 1, 2, 1, 1, 1
15	382, 6, 4, 4, 3, 3, 1, 1, 1, 1, 1, 2, 1, 1, 1

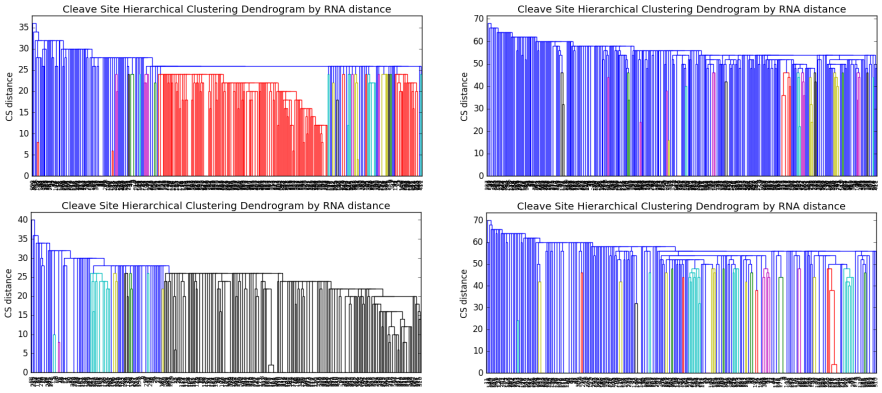

Figure 6.5: Dendrograms of hierarchical clustering with datasets WT_71_Distance(top left), WT_121_Distance(top right), XRN4_71_Distance(bottom left) and WT_121_Distance(bottom right)

6.3.3 Spectral clustering

The Euclidean distance matrix cannot be used directly in spectral clustering algorithm for RNA structure. Pedregosa et al. (2011) suggested The Gaussian function() was used for precomputing the affinity matrix. Equation 6.3.1 demonstrated the function, where s is standard deviation, and d is Euclidean distance.

$$g(x) = e\frac{-d^2}{2s^2} \tag{6.3.1}$$

6.3.4 Affinity propagation

affinity propagation used the same matrix as Spectral clustering. I tested the four distance datasets with damping from 0.5 to 0.9 at interval 0.1. There were 27 to 296 clusters in each clustering. It was not likely this algorithm would discover any significant different clusters. I ran the algorithm nonetheless, and no clustering which met the criteria was acquired.

6.4 Results and analysis

Four algorithms were used for clustering the structures. K-medoids and spectral clustering discovered some clusterings that met the criteria. The clusterings with significant rate 100 had 2 or 3 clusters. Table 6.3 shows the detailed summary of the clusters found when significant rate was 100.

Table 6.3: Cluster count when significant rate was 100

Degradome	**WT**		**XRN4**	
Length	**71**	**121**	**71**	**121**
K-medoids	2, 3	2, 3	2, 3	2
Spectral clustering	2	-	2, 3	2, 3

6.4.1 Report the best-performed clustering

Here I looked into one report of the best-performed clusterings (Table 6.4).

Table 6.4: One of the best-performed clustering

Algorithm	K-medoids	**Significant Rate**	100
Dataset	WT_71_Distance	**K(Groups)**	3

The ICAS plotted the histogram (Figure 6.6) for this clustering. There were 180, 82 and 150 structures in the three clusters. This clustering was far more balanced than the best clusterings discovered clustering algorithms with reactivity.

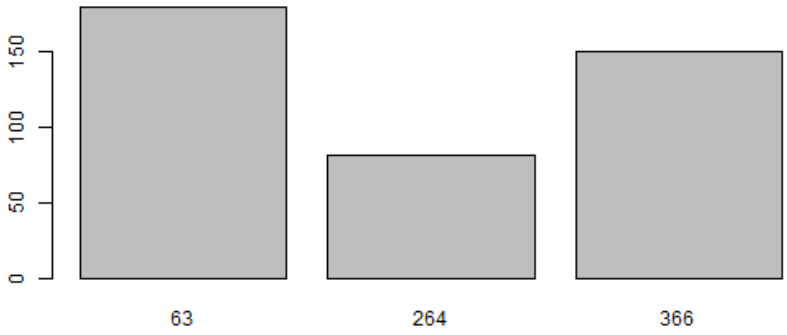

Figure 6.6: Data distribution of three clusters

The ICAS generated a chart of cluster medoids and the distance between them (Figure 6.7).

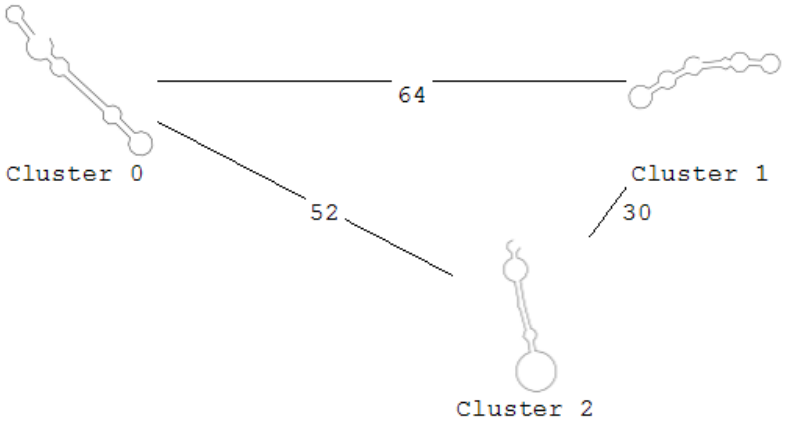

Figure 6.7: Medoids and the pairwise distance

6.4.2 cluster cohesion

The ICAS also plotted out the clusters. Figure 6.8 displays 20 structures of Group 0.

Within each of the clusters, the structures look quite different from each other. Most importantly, the seed areas were similar to each other and the seed area determines the cleavage activities (Section 2.3.1), thus I cannot consider the data points in the cluster as similar.

6.4.3 Disputed data points

In order to check the validation of the kmedoids algorithm, I checked the distance from each data point to each medoid. The data points were assigned to the medoid

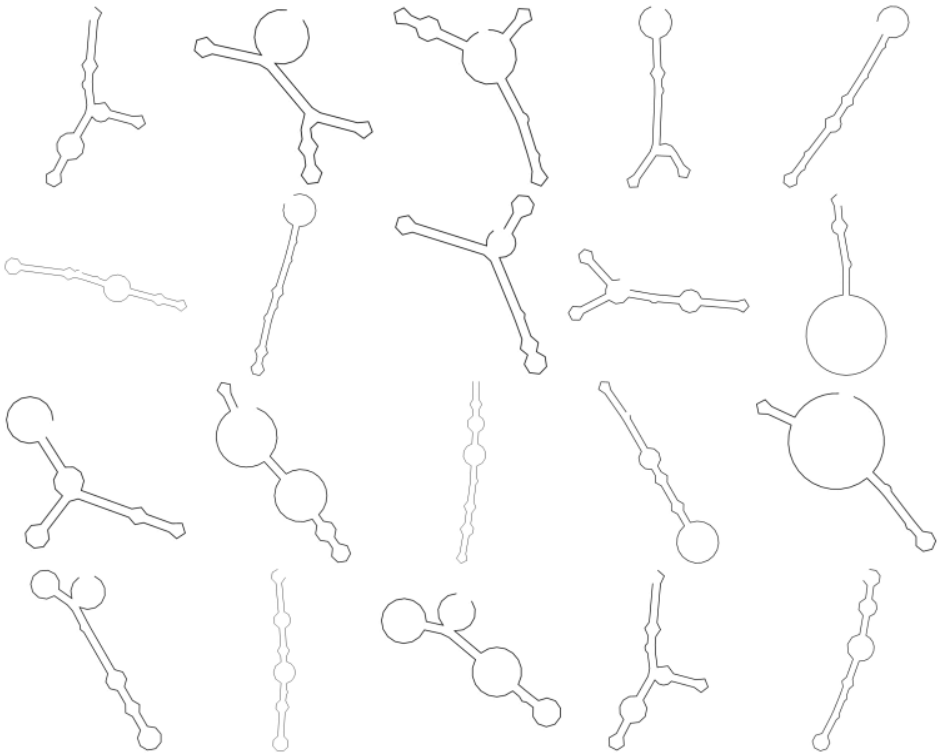

Figure 6.8: Some structures in Group 0

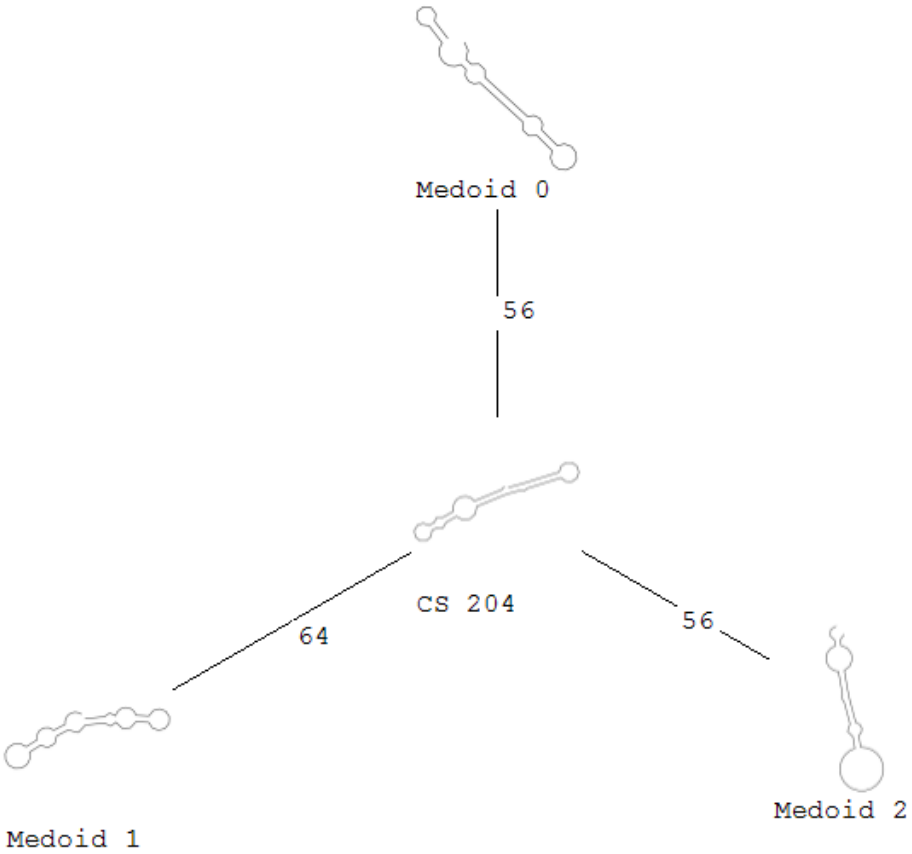

Figure 6.9: A disputed data point between Cluster 0 and 2

of the shortest distance as expected. Thus I did not find faults in the algorithm from the distances. However, I noticed that some data points had two or more nearest

medoids. And the data points were always assigned to the last medoid, which made the last cluster larger than it should be. An example is cleavage site 204 (Figure A). The distances to Medoid 0 and 2 were the same. In 412 data points, 27 or 6.6% of them had more than one nearest medoids. The cause of the problem was that the distance was integers, and two integers are more likely to be the same. In the future work, my suggestion is to randomly add a fraction which is far smaller than 1, so the controversial data points will not always be assigned to the last medoid.

6.4.4 Stability

Both k-medoids and spectral clustering were not stable. In 100 repeats, they might have one or two clusterings that met the evaluation criteria whereas hierarchical clustering and affinity propagation were very stable.

The instability could be caused by the distribution of the data. For example, if the data was evenly distributed, many data points might be a medoid or centroid, and the result might not be stable. In a real dataset such as the datasets in this project, there might be numerous clusterings, and one or two of them might meet some standard (such as the evaluation criteria in this paper) accidentally.

Figure 6.10 demonstrates instability with evenly distributed datasets. The same coloured hallow shapes were medoids in the same run.

6.4.5 Conflict between the structures with 71 and 121 nucleotides

The structures were calculated with 71 or 121 nucleotides without considering the nucleotides outside the cleavage site but still in the same messenger RNA. I compared the structure of the same cleavage site. The structures were always different with 71 or 121 nucleotides. My method was to cut off 25 nucleotides from each end of the 121

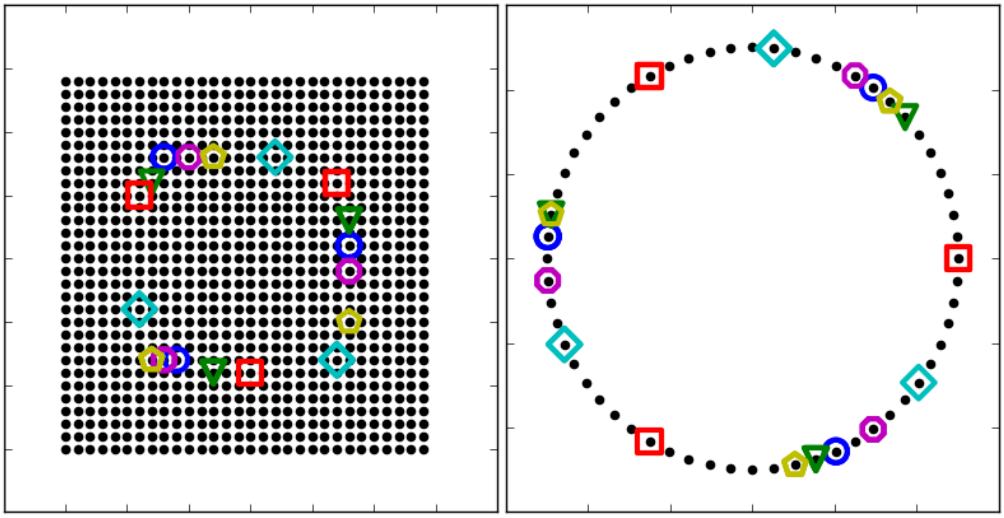

Figure 6.10: Evenly distributed data with medoids

structure and compare it with the corresponding cleavage site with 71 nucleotides, so each of them had 71 nucleotides. Figure 6.11 and 6.12 demonstrate the comparison. The left structure has 121 nucleotides. I erased the 25 nucleotides from both ends and turned it into the one in the middle. The right structure was the one generated with 71 nucleotides. The middle one and the right one were clearly different.

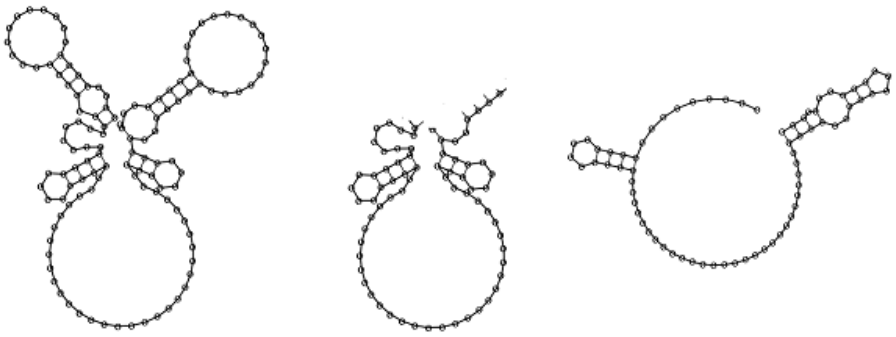

Figure 6.11: Trustworthiness comparison 1

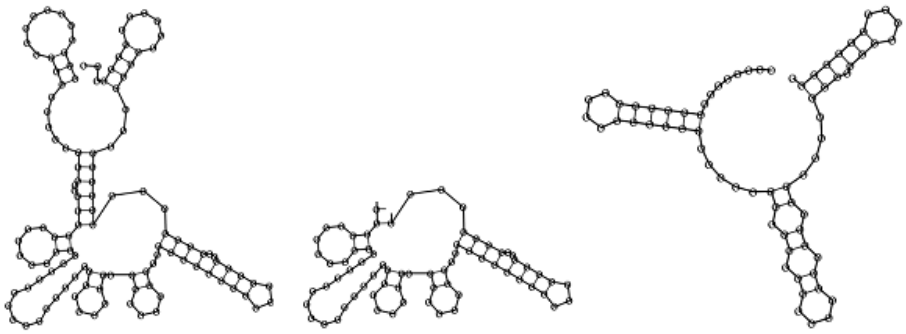

Figure 6.12: Trustworthiness comparison 2

I compared all the structures and Table 6.5 summarises the comparison.

Table 6.5: Structure comparison summary between cleavage site with 71 and 121 nucleotides

	WT	XRN4
Total Cleavage site	412	311
Match	9	7
Match percentage	2%	2%
Average edit distance(DBN)	24.1	24.75

From Table A, only 2% of the structures match. The average edit distance of the DBN between them was 24, which could be considered as a large distance given the string length was only 71.

In conclusion, the cleavage site structures are not trustworthy when generated locally in the cleavage site, they should be generated globally considering the whole messenger RNA.

6.5 Summary

After using k-medoids, hierarchical clustering, affinity propagation and spectral clustering, clusterings that met the evaluation criteria were finally discovered. However, the results were not stable; there were disputed data points between clusters; the data points in the same cluster did not look similar, and the structures with 71 and 121 nucleotides were contradictory.

In a net shell, although clusterings (k=2, 3) which abide by the evaluation standards were found, the result could not be fully trusted.

Chapter 7

Ensemble clustering

In the last two chapters, individual algorithms for reactivity and CS distance were introduced. This chapter presents the ensemble approach. An example is given which helps readers further understand. The performance of this approach is compared with that of individual algorithms in the last two chapters.

There is no prior research found using ensemble approach for cleavage site clustering analysis.

7.1 Ensemble approach

To improve the performance of the clustering, I proposed an ensemble approach. The key idea is to combine several clustering to generate a final, possibly better clustering.

7.1.1 Steps

Figure 7.1 displays the workflow of the ensemble process. The process can be divided into the following steps.

Step 1 Read the clusterings generated by individual clustering algorithms.

Step 2 Select the most dissimilar clusterings.

Step 3 Run consensus function(hard/Euclidean) 100 times.

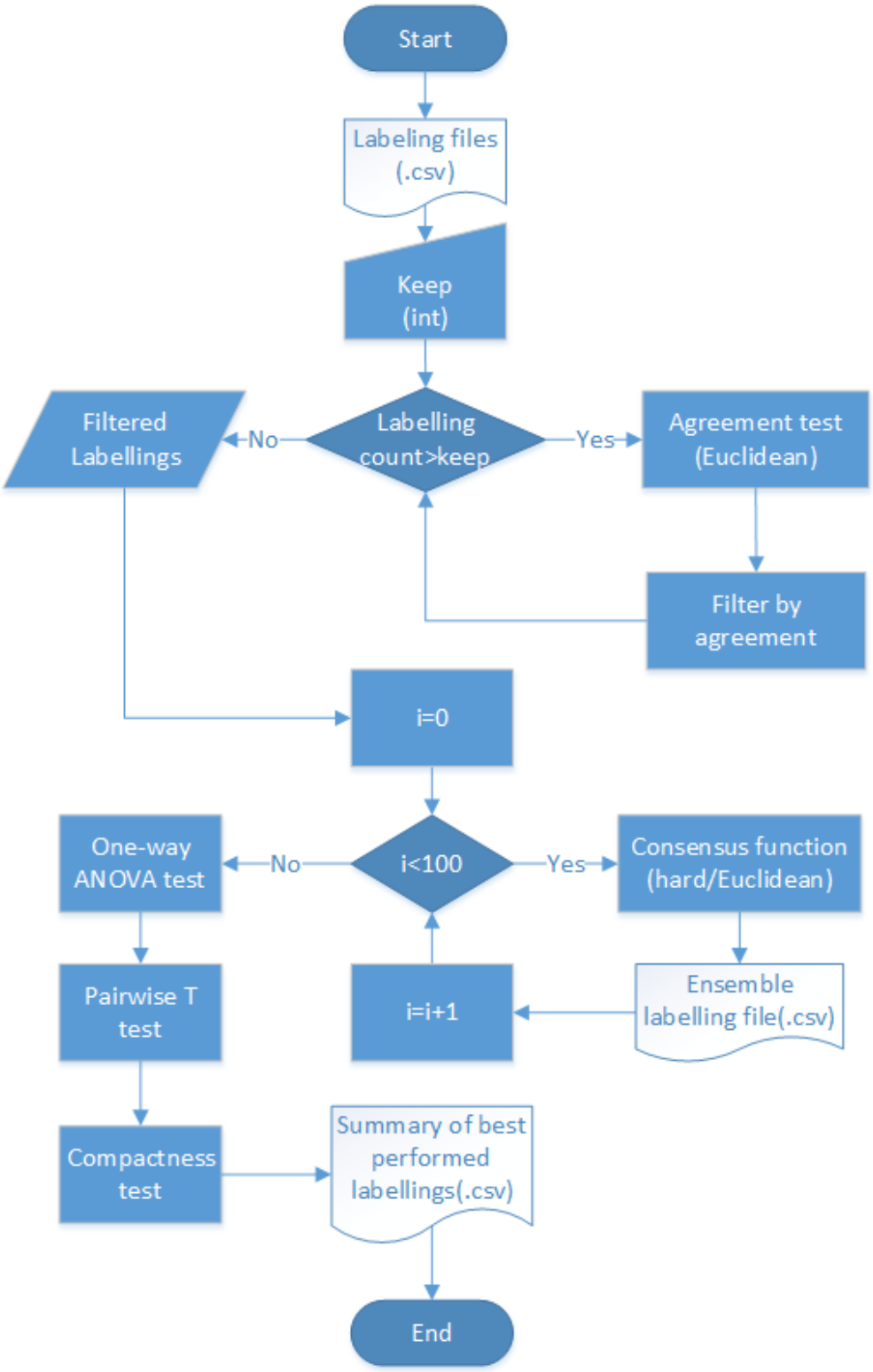

Figure 7.1: Working process of clustering ensemble

Step 4 Find the best-performed clusterings by Apply one-way ANOVA test, pairwise T test and compactness test.

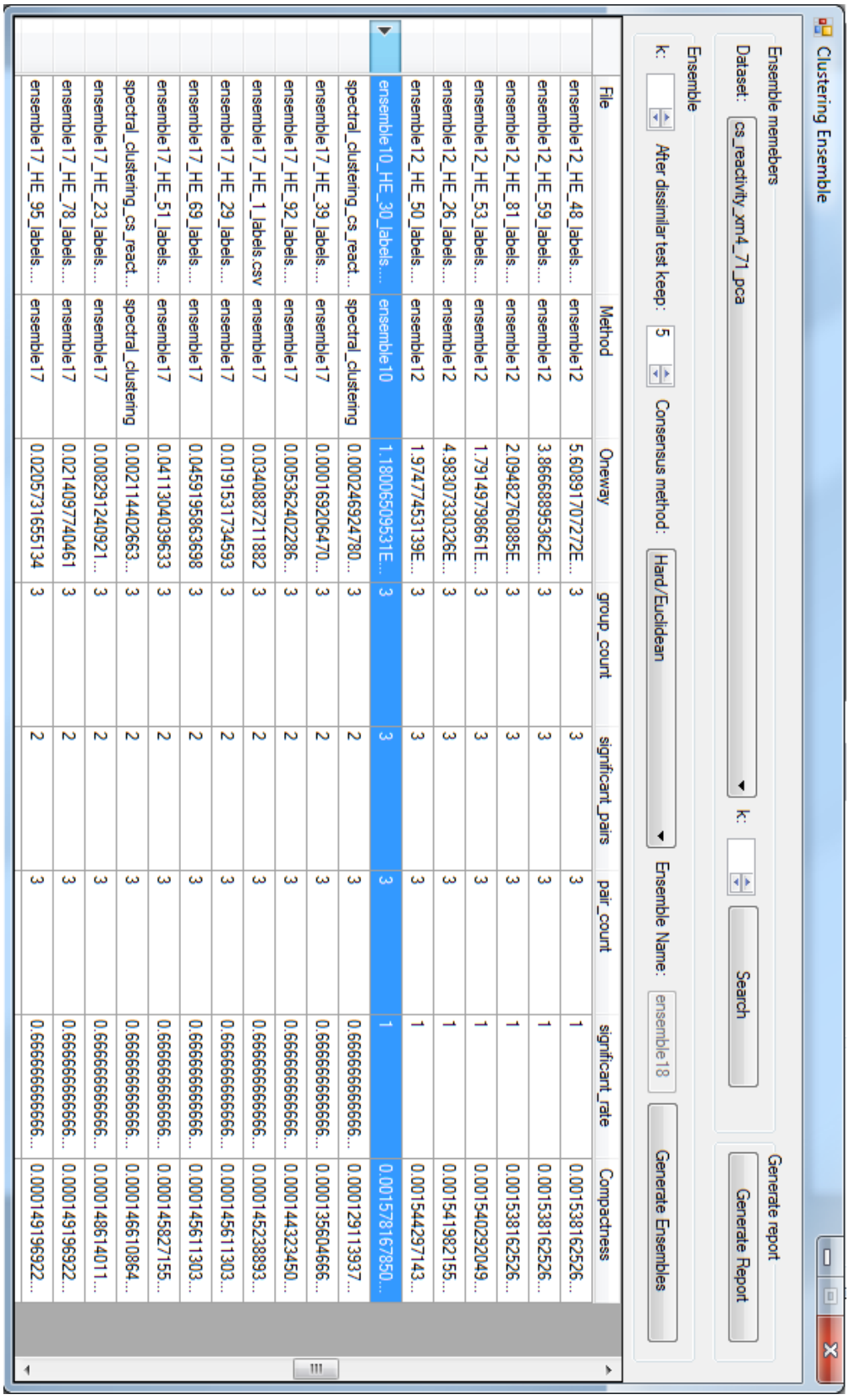

Figure 7.2: Ensemble form GUI

7.1.2 Ensemble GUI

There are four sections in the ensemble GUI (Figure 7.2), the ensemble members section, the report section, the ensemble section and the results section. The user selects

a dataset and the cluster numbers(k, optional) in the ensemble members section, and click the search button. The clustering which meet these search criteria will show in the results section. Next step is to generates ensembles in the ensemble section. The user selects the cluster numbers(k, optional), members to keep(default is 5, compulsory) and consensus method, and click the generate ensemble button. The ensemble process will start instantly. After a few seconds, the ensemble process will finish, and the result will be presented in the results section. The user can choose one of the results, usually the first one, to generate a report.

In the GUI, the only places that accept user input are two cluster number numeric up-down controls. The numeric up-down control is designed to accept only numbers, so the user will not put a non-number character and make the system crash.

7.1.3 Fixed k or not

In the ensemble approach, the clusters number can be fixed or not. If the user wants to fixate the number of clusters, he or she can input a number in the numeric up-down control. Otherwise, he or she can delete the number, and the system will select k value automatically.

7.1.4 Genetic algorithm

The genetic algorithm is implemented by conducting the ensemble approach with the ensemble results instead of that from individual algorithms. Thus the ensemble results using clusterings from individual algorithms are considered the first generation. The results of the first generation are utilised by the ensemble approach again to generate the second generation. This iteration goes on and the third, fourth and until n-th generations will be generated.

7.2 Results

I used the ensemble approach to 10 datasets. The results are summarised in Table 7.1.

Table 7.1: Ensemble results

Dataset	**K**	**Better?**	**Comments**
WT_21_Reactivity	3	No	Compactness decreased
WT_71_Reactivity	3	No	Compactness decreased
WT_121_Reactivity	4	Yes	SR increased
XRN4_21_Reactivity	5	Yes	Compactness increased
XRN4_71_Reactivity	3	Yes	SR increased
XRN4_121_Reactivity	3	Yes	Compactness increased
WT_71_Distance	4	No	Compactness decreased
WT_121_Distance	3	Yes	Compactness increased
XRN4_71_Distance	3	No	Compactness decreased
XRN4_121_Distance	4	Yes	Compactness increased

From the result, the ensemble approach was 60% likely to increase the performance, and it never decreased the significant rate.

7.2.1 Report one of the best-performed clusterings

In this section, the ensemble approach was used, and the performance of the individual clusterings were improved. Firstly, I chose 14 best-performed clusterings with Dataset XRN4_71_PCA. All of them had the same significant rate 67. K was fixed to 3. Six were from k-means, and the rest eight from spectral clustering. The pairwise agreement was calculated. The agreement was between 0.24 and 0.72. The ICAS repeatedly removed nine of them according to the highest pairwise agreement. Five were left, and the agreement between them was from 0.26 to 0.38. Table 7.2 displays the pairwise agreement.

The five were used by hard/Euclidean consensus function with 100 repeats. An ensemble result with significant rate 100 was discovered. I then performed the genetic algorithm. I chose six of the best-performed ensemble results and repeated the same process. The new generation ensemble results decreased the compactness by 20% from

Table 7.2: Pairwise agreements of individual clusterings

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1				
[2,]	0.360969	1			
[3,]	0.376247	0.276042	1		
[4,]	0.321909	0.294031	0.276042	1	
[5,]	0.321909	0.26066	0.258489	0.3032	1

0.0015 to 0.0012.

I also compared the balance of the two clusterings. Figure 7.3 compares the distribution of the two. It looked that the second generation was more balanced than the first. The standard deviations of member count in each cluster of the two clusterings were 59.1 and 52 .0. Thus, the second generation was more balanced regarding cluster size.

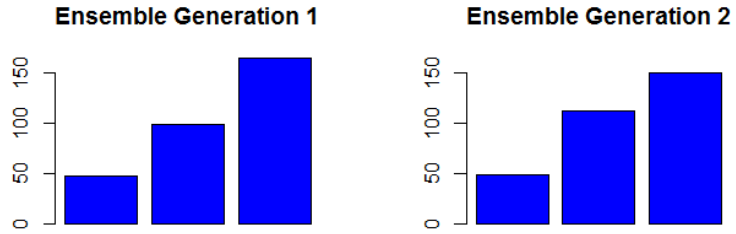

Figure 7.3: Distribution of Generations 1 and 2

I repeated the genetic approach, but the performance of the third generation was worse than the second generation. Thus, I stopped the iteration.

7.2.2 Stability

The performance of the ensemble approach was better than that of the individual algorithms. With k-means and spectral clustering algorithms applied to reactivity, clusterings with significant rate 100 were found. However, they were all unbalanced. On the contrast, the ensemble approach achieved significant rate 100 from a batch of individual ones with significant rate 67. The clusters were more balanced than that in the individual ones. Figure 7.4 compares the internal agreement between of k-means,

spectral clustering, the ensemble approach and the genetic approach.

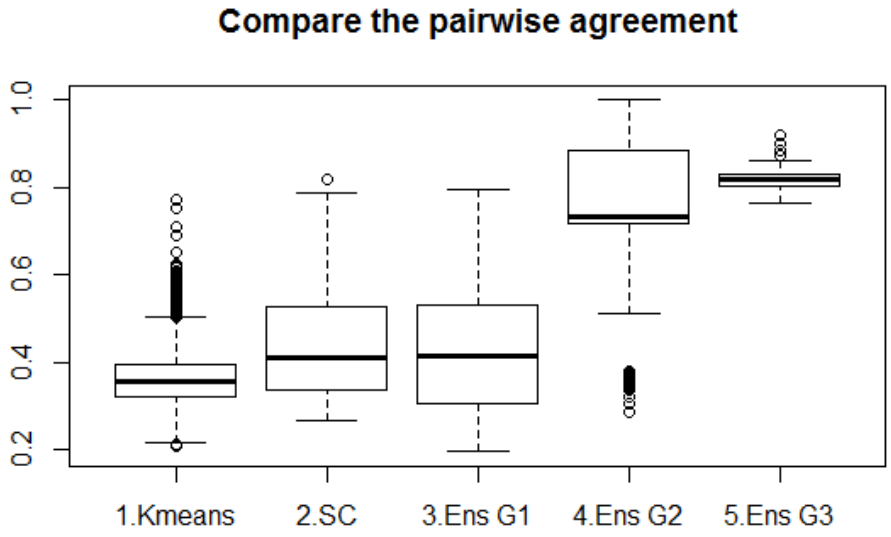

Figure 7.4: Median internal agreement (Dateset = XRN4_71_PCA, K = 3)

From Figure 7.4, the internal agreement of the clusterings tends to increase after iterations of ensemble clustering. A higher internal agreement indicates the clusterings were more similar to each other, and the results were more stable. Conclusively, ensemble clustering algorithms were more stable than individual ones. I did pairwise T test, too, which showed that each of the average agreement was significantly different from others. Ensemble approaches significantly increased agreement and stability after two or three generations.

7.3 Clustering with four clusters

Previously, in the individual clusterings, all clusterings with significant rate 100 had three clusters. The ensemble approach was able to find a clustering with four clusters and the significant rate was 100 in Dataset reactivity_WT_121. However, this clustering was not balanced.

7.4 Summary

The ensemble approach was applied to ten of the datasets, and 60% of them see an increase in performance.

Regarding SR and compactness, The ensemble approach could generate better results than the individual ones. The genetic approach could make the results even better.

The genetic approach makes the clustering results more stable after each iteration.

In conclusion, the ensemble clustering methods could improve the performance of individual ones.

Chapter 8

Discussion and conclusions

In the previous chapters, the raw data was pre-processed. Reactivity and cleavage site distance were used to cluster the cleavage sites with individual and ensemble algorithms. Reports were generated. Meanwhile, a cluster pipeline was developed to carry on the above tasks. In this final chapter, a conclusion is drawn based on the clustering results, and suggestions for possible further work are made.

8.1 Ever-changing raw data and requests

This is the first time that clustering analysis has been applied to cleavage site study, and there is no predecessor to follow. Thus this project is susceptible to change.

The first major change was the cleavage site prediction tool. I had been using this tool from January to June 2016 until I found the results were not reasonable.

Guided by the client, I had been using Scan-for-match from January 2016 to June 2016. This tool generates a high number of false cleavage sites. At that time the degradome data was not ready, and I was told that the degradome data would be used to filter out the false cleavage sites. When I got the raw data in June, I found the results were not expected and I started looking for new tools.

In July 2016, the client updated the degradome data, and I restarted pre-process and clustering.

Every time, the client made some changes with raw data, I had to restart the projects. And all the plots and clustering analysis became out of date immediately.

The methods were also changing. The features changed between reactivity and RNA distance, back and forth. Since the client was not sure which one was better, I decided to finish clustering for both in the hope of that one method would give the expected results. The other practical reason was that when the client asked me to abandon RNA distance and switch to reactivity, I had almost finished data pre-processing and clustering analysis, so I decided to write them down, even though the method was not well-orchestrated.

Without the assistance of ICAS, I would have spent more time to deal with changes.

8.2 Summary and discussion

The aim of this project was to develop a novel analysis pipeline to recognise in vivo RNA structure patterns at and around the miRNA target site. The ICAS was established to fulfil this requirement. Raw data was pre-processed. Individual and ensemble algorithms were applied to cluster the cleavage site structure presented with reactivity and then the DBN distance matrix. The ICAS generates reports with a dozen of charts for each clustering result.

The ICAS played a major role in this study. Without the ICAS, it would be difficult to handle frequent data and methods change. The system is designed to be flexible. It is easy to add new clustering algorithms and new datasets, and change other settings. Most importantly, this research has not found any expected results yet. I organised all the code into one single system for my successors so they can continue the research easily.

The metrics were set to determine the best clustering result. I proposed significant

rate, which is the count of significant different pairs of clusters divide by the cluster count (See Section 3.2.1), as the first measurement, followed by compactness. Agreement by Euclidean method was also used to measure the difference between clustering results.

I firstly clustered the data by reactivity of the cleavage sites. I found a stable clustering with two clusters by k-means. One cluster had high reactivity and low cleavage efficiency. The other was opposite (Section 5.4.1). However, after analysis, there was no correlation between them (Section 5.4.3).

Features were selected by PCA, variance and linear regression. Most of the time, PCA improved performance. Occasionally, variance-selected features performed better than the original features. Linear regression returned a very low r-squared value and were not studied further (Section 5.3).

Clustering analysis was also applied to structure data. Some results which met the metrics were found. However, the structures in the same cluster were too dissimilar to be clustered together (Section 6.4.2). Some data points had the same distance to their medoids and other medoids (Section 6.4.3). The performance was not reliable and stable (Section 6.4.4). Finally, the structure per se was not trustworthy (Section 6.4.5). Thus, I cannot trust the results of clustering by cleavage site distance matrices.

Ensemble and genetic methods took the individual algorithm results and improved them. Experiments showed that 60% of the case had the performance improved. Although, it is not guaranteed the performance will become better, 60% is still a strong indicator that ensemble methods are worth trying.

8.3 Conclusions

From the above discussion, the following conclusions are drawn.

1. This study is novel and no tool had been developed for it. I developed a flexible and HPC-friendly pipeline, ICAS, which helped data-preprocessing, clustering and report generating. It also made changing of raw data and methodologies much easier and faster. (Section 3.1).

2. A new metric, significant rate was introduced in this study. A clustering which meets the metrics is one in which each cluster is significantly different from others in terms of cleavage efficiency or the significant rate is 100 (Section 3.2.1).

3. Clustering by reactivity was performed. Clusterings with 2 or 3 clusters, which met the metrics, were found in this research. Specifically, when using k-means and k was 2 on the WT reactivity dataset with 21 features, the result was identical in each run. However, the rest clusterings were not stable. No clustering which met the metrics was found when k was from 4 to 15.

4. Clustering by cleavage structure distance matrices in this study was not reliable and trustworthy. Further studies are needed.

5. The cleavage site structures are not trustworthy when generated locally in the cleavage site, they should be generated globally considering the whole messenger RNA.

6. In this project, the method using reactivity outperformed that using cleavage site distance matrices in terms of stability.

7. Algorithms with random initialisation are more likely to find clustering schemes that meet some external metrics, but the results are not stable and reliable (Section 6.4).

8. Dimensionality reduction can change the performance of clustering algorithms

and are more likely to improve it. PCA was able to clusterings with two or three clusters in which each cluster was significantly different from others in terms of cleavage efficiency whereas without dimensionality reduction, only clusterings met the metrics with two clusters were found (Section 5.4).

9. Ensemble clustering improved the performance in 60% of the cases, and it was able to find four significantly different (by cleavage efficiency) clusters when clustering by reactivity (Section 7.2).

10. Ensemble and genetic methods may generate more stable results than individual algorithms such as k-means, k-medoids and spectral clustering. The pairwise agreement (Euclidean method) improved from 0.4 to 0.8 after 3 generations of ensemble clustering (Section 7.2.2).

8.4 Suggestion for further work

8.4.1 Clustering performance criteria

I found some significant clusterings by the current criteria. However, some of them had imbalanced clusters. In future studies, cluster balance should be taken into consideration and clusterings which does not meet this standard shall be removed in early stages.

8.4.2 Cleavage structure, distance and feature extraction

The cleavage site structure was not trustworthy in this study. That was because the nucleotides inside the cleavage site may pair with the ones outside of the cleavage site. The best way to determine this is to compute the whole messenger RNA such that the structure will be more reliable and trustworthy. The problem is that RNADistance

program only compares the whole structures. The successor needs to find a way that is biologically meaningful to calculate the distance between two cleavage sites which are part of the messenger RNA.

Another way is to extract digital features from the structure and do clustering with them. Although there is no prior research of clustering the cleavage site, features for other machine learning methods related with the cleavage site can be used in future work. Kim et al. (2006b) suggested three categories of features, namely structural features, thermo-dynamic features and position-based features. Those features were used in classification(SVM). I suggest the successor use them for clustering as well.

8.4.3 Weighted clustering

This study treated every nucleotide in the cleavage site as equal, which in reality might not be true. Future work may change the weight of each nucleotide to see whether the performance improves. However, this involves a tremendous amount of computation.

8.4.4 Consider supervised learning

In this study, I could have used supervised learning. Some function of the cleavage efficiency could be used with several parameters. After that, some machine learning algorithms can be applied to find out the possible values of the parameters by a criterion to minimise the error of the function.

Given cleavage efficiency function $ce = f(x)$, where x is the vector $[x_1, x_2, ...x_n]$. The possible value of n could be 21 if only 21 nucleotides in the cleavage site are to be used and x would be the reactivity of the 21 nucleotides. The linear function can be presented as Equation 8.4.1

$$f(x) = a_1x_1 + a_1x_2...a_nx_n \tag{8.4.1}$$

Thus, the problem would become searching for $a_1, a_2...a_n$, which minimise the squared error of the function.

beside linear regression, other supervised learning algorithms, such as logistic classification, Artificial neural network, SVM and decision tree, can also be used.

This study discarded a great number of clusterings because they did not meet the statistical standards relating to cleavage efficiency. The reason for that is the cleavage efficiency is not used when the algorithms try to converge to the local or global optimum. If supervised learning were used, the cleavage efficiency would be taken into consideration in the algorithm and the result would be more desirable.

Different form clustering which takes the cleavage site as a whole in considering the cleavage efficiency, most supervised learning algorithms evaluate the influence of every feature. They are two different approaches to finding the intrinsic rules. There is no golden standard of which one is better. Thus, applying both gives more opportunities to discover something significant.

One reason this study did not use supervised learning was that I was not able to extract features for the RNA structure data. Once the features have been extracted, the supervised learning can be applied.

Bibliography

Addo-Quaye, C., Eshoo, T. W., Bartel, D. P., and Axtell, M. J. (2008). Endogenous siRNA and miRNA targets identified by sequencing of the Arabidopsis degradome. *Current Biology*, 18(10):758–762.

Ahmed, F., Kaundal, R., and Raghava, G. P. (2013). Phdcleav: a svm based method for predicting human dicer cleavage sites using sequence and secondary structure of mirna precursors. *BMC bioinformatics*, 14(14):1.

Alspaugh, S. (2011). K-medoids code. `https://github.com/salspaugh/machine_learning`. [Online; accessed 05-November-2016].

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Ambros, V. (2004). The functions of animal microRNAs. *Nature*, 431(7006):350–355.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

Bartel, D. P. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *cell*, 116(2):281–297.

Bartel, D. P. (2009). MicroRNAs: target recognition and regulatory functions. *Cell*, 136(2):215–233.

Berkhin, P. (2004). Survey of clustering data mining techniques, 2002. *Accrue Software: San Jose, CA*.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

Boley, D., Gini, M., Gross, R., Han, E.-H. S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. (1999). Partitioning-based clustering for web document categorization. *Decision Support Systems*, 27(3):329–341.

Bonnet, E., He, Y., Billiau, K., and Van de Peer, Y. (2010). Tapir, a web server for the prediction of plant microRNA targets, including target mimics. *Bioinformatics*, 26(12):1566–1568.

Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1997). Clustering via concave minimization. *Advances in neural information processing systems*, pages 368–374.

Chi, S. W., Hannon, G. J., and Darnell, R. B. (2012). An alternative mode of microRNA target recognition. *Nature structural & molecular biology*, 19(3):321–327.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619.

Dai, X., Zhuang, Z., and Zhao, P. X. (2011). Computational analysis of miRNA targets in plants: current status and challenges. *Briefings in bioinformatics*, 12(2):115–121.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer.

Dimitriadou, E., Weingessel, A., and Hornik, K. (2002). A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(07):901–912.

Dsouza, M., Larsen, N., and Overbeek, R. (1997). Searching for patterns in genomic data. *Trends in Genetics*, 13(12):497–498.

Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75.

Fabian, M. R. and Sonenberg, N. (2012). The mechanics of miRNA-mediated gene silencing: a look under the hood of miRISC. *Nature structural & molecular biology*, 19(6):586–593.

Fahlgren, N., Howell, M. D., Kasschau, K. D., Chapman, E. J., Sullivan, C. M., Cumbie, J. S., Givan, S. A., Law, T. F., Grant, S. R., Dangl, J. L., et al. (2007). High-throughput sequencing of arabidopsis microRNAs: evidence for frequent birth and death of MIRNA genes. *PloS one*, 2(2):e219.

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976.

Frost, R. J. and Olson, E. N. (2011). Control of glucose homeostasis and insulin sensitivity by the Let-7 family of microRNAs. *Proceedings of the National Academy of Sciences*, 108(52):21075–21080.

German, M. A., Pillay, M., Jeong, D.-H., Hetawal, A., Luo, S., Janardhanan, P., Kannan, V., Rymarquis, L. A., Nobuta, K., German, R., et al. (2008). Global identification of microRNA–target RNA pairs by parallel analysis of RNA ends. *Nature biotechnology*, 26(8):941–946.

Hamerly, G. and Elkan, C. (2003). Learning the k in k-means. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 281–288. MIT Press.

Hornik, K. (2005). A clue for cluster ensembles. *Journal of Statistical Software*, 14(12):1–25.

Hu, Y.-J. (2002). Prediction of consensus structural motifs in a family of coregulated RNA sequences. *Nucleic acids research*, 30(17):3886–3893.

Huang, J. C., Morris, Q. D., and Frey, B. J. (2007). Bayesian inference of MicroRNA targets from sequence and expression data. *Journal of Computational Biology*, 14(5):550–563.

Jain, A. K. and Dubes, R. C. (1988a). *Algorithms for clustering data*. Prentice-Hall, Inc.

Jain, A. K. and Dubes, R. C. (1988b). *Algorithms for clustering data*. Prentice-Hall, Inc.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

Jiang, Q., Wang, Y., Hao, Y., Juan, L., Teng, M., Zhang, X., Li, M., Wang, G., and Liu, Y. (2009). miR2Disease: a manually curated database for microRNA deregulation in human disease. *Nucleic acids research*, 37(suppl 1):D98–D104.

Kaufman, L. and Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

Kim, S.-K., Nam, J.-W., Rhee, J.-K., Lee, W.-J., and Zhang, B.-T. (2006a). miTarget:

microRNA target gene prediction using a support vector machine. *BMC bioinformatics*, 7(1):411.

Kim, S.-K., Nam, J.-W., Rhee, J.-K., Lee, W.-J., and Zhang, B.-T. (2006b). miTarget: microRNA target gene prediction using a support vector machine. *BMC bioinformatics*, 7(1):411.

Kolawa, A. and Huizinga, D. (2007). Automated defect prevention: Best practices in software management. *Wiley-IEEE Computer Society Press*, 41:86.

Krek, A., Grün, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., et al. (2005). Combinatorial microRNA target predictions. *Nature genetics*, 37(5):495–500.

Li, Y., Deng, C., Shang, Q., Zhao, X., Liu, X., and Zhou, Q. (2016). Characterization of siRNAs derived from cucumber green mottle mosaic virus in infected cucumber plants. *Archives of virology*, 161(2):455–458.

Liu, H., Yue, D., Zhang, L., Gao, S.-J., and Huang, Y. (2008). A machine learning approach for miRNA target prediction. In *Genomic Signal Processing and Statistics, 2008. GENSiPS 2008. IEEE International Workshop on*, pages 1–3. IEEE.

Lorenz, R., Bernhart, S. H., Zu Siederdissen, C. H., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):1.

Lu, S., Sun, Y.-H., and Chiang, V. L. (2008). Stress-responsive microRNAs in Populus. *The Plant Journal*, 55(1):131–151.

Mendoza, M. R., da Fonseca, G. C., Loss-Morais, G., Alves, R., Margis, R., and Bazzan, A. L. (2013). Rfmirtarget: predicting human microrna target genes with a random forest classifier. *PloS one*, 8(7):e70153.

Menor, M., Ching, T., Zhu, X., Garmire, D., and Garmire, L. X. (2014). mirmark: a site-level and utr-level classifier for mirna target prediction. *Genome biology*, 15(10):1.

Miranda, K. C., Huynh, T., Tay, Y., Ang, Y.-S., Tam, W.-L., Thomson, A. M., Lim, B., and Rigoutsos, I. (2006). A pattern-based method for the identification of microrna binding sites and their corresponding heteroduplexes. *Cell*, 126(6):1203–1217.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5(7):621–628.

Nguyen, N. and Caruana, R. (2007). Consensus clusterings. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 607–612. IEEE.

Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448.

Peason, K. (1901). On lines and planes of closest fit to systems of point in space. *Philosophical Magazine*, 2:559–572.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pelleg, D., Moore, A. W., et al. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, volume 1.

Pisapia, L., Cicatiello, V., Barba, P., Malanga, D., Maffei, A., Hamilton, R. S., and Del Pozzo, G. (2013). Co-regulated expression of alpha and beta mRNAs encoding HLA-DR surface heterodimers is mediated by the MHCII RNA operon. *Nucleic acids research*, 41(6):3772–3786.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (2007). Gaussian mixture models and k-means clustering. *Numerical Recipes: The Art of Scientific Computing, 3rd ed. Cambridge University Press, New York*, pages 842–850.

Prodromidis, A., Chan, P., and Stolfo, S. (2000). Meta-learning in distributed data mining systems: Issues and approaches. *Advances in distributed and parallel knowledge discovery*, 3:81–114.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Rokach, L. and Maimon, O. (2005). Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 731–737. IEEE.

Slonim, N. and Tishby, N. (1999). Agglomerative information bottleneck. In Solla, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 617–623. The MIT Press.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

Srivastava, P. K., Moturu, T. R., Pandey, P., Baldwin, I. T., and Pandey, S. P. (2014). A comparison of performance of plant miRNA target prediction tools and the characterization of features for genome-wide target prediction. *BMC genomics*, 15(1):1.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Sun, Y.-H., Lu, S., Shi, R., and Chiang, V. L. (2011). Computational prediction of plant miRNA targets. *RNAi and Plant Gene Function Analysis: Methods and Protocols*, pages 175–186.

Thomson, D. W., Bracken, C. P., and Goodall, G. J. (2011). Experimental strategies for microRNA target identification. *Nucleic acids research*, 39(16):6845–6853.

Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372.

Xu, M. Y., Dong, Y., Zhang, Q. X., Zhang, L., Luo, Y. Z., Sun, J., Fan, Y. L., and Wang, L. (2012). Identification of mirnas and their targets from brassica napus by high-throughput sequencing and degradome analysis. *BMC genomics*, 13(1):1.

Zhou, W. (2011). CSV2Entity. `http://http://csv2entity.codeplex.com/`. [Online; accessed 05-November-2016].

Zhu, H., Shyh-Chang, N., Segrè, A. V., Shinoda, G., Shah, S. P., Einhorn, W. S., Takeuchi, A., Engreitz, J. M., Hagan, J. P., Kharas, M. G., et al. (2011). The lin28/let-7 axis regulates glucose metabolism. *Cell*, 147(1):81–94.