Kyoto University

Doctoral Thesis

Identification and Analysis of Critical Sites in RNA/Protein Sequences and Biological Networks

Author:
Bao Yu

Supervisor:
Prof. Tatsuya Akutsu

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Laboratory of Mathematical Bioinformatics
Department of Intelligent Science and Technology

August 31, 2018

Declaration of Authorship

I, Bao Yu, declare that this thesis titled, "Identification and Analysis of Critical Sites in RNA/Protein Sequences and Biological Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Abstract

Bao Yu

Identification and Analysis of Critical Sites in RNA/Protein Sequences and Biological Networks

Critical objects identification and analysis are fundamental parts of Bioinformatics. For RNA/protein analysis, critical objects often denote cleavage sites identification for ribosomes which include endonuclease/Exoribonuclease, and proteases such as apoptosis-related enzymes including caspase family. For biological network analysis, critical objects often mean a node set that plays a critical role in controlling the whole network, as an example Nacher and Akutsu develop a minimum dominant set approach to divide nodes in a biological network into critical nodes/intermittent nodes/redundant nodes according to roles in controlling a biological network.

As a fundamental component of systems biology, proteolytic cleavage is involved in nearly all aspects of cellular activities, from gene regulation to cell life-cycle regulation. Among the various proteolytic cleavages, caspases/granzyme B cleavages are considered as essential parts in the execution of apoptosis and inflammation processes. Although a number of substrates for both types have been experimentally identified, the complete repertoire of caspases and granzyme B substrates remain to be fully understood.

As another essential type of cleavage enzyme, Dicer is necessary for the process of mature microRNA (miRNA) formation because the Dicer enzyme cleaves pre-miRNA correctly to generate miRNA with correct seed regions. Here MicroRNAs (miRNAs) are a type of small RNAs with the length of ∼22 nt, which perform the function of suppressing gene expression at the post-transcriptional level. Usually in vivo, a gene of a miRNA is transcribed to produce a long, primary miRNA (pri-miRNA) transcript, which is then processed into a ∼65-nt-long hairpin structure via cleavage by the Drosha (DGCR8) enzyme. Nonetheless, the mechanism underlying the selection of a Dicer cleavage site is also not fully understood.

Besides proteolytic/RNA cleavage enzymes, the controllability of complex networks is also of great importance in wide-ranging research fields. Recently, a new approach

based on the minimum feedback vertex set (MFVS) has been proposed to find such vertices in directed networks in which the target states are restricted to steady ones. However, multiple MFVS configurations may exist and thus the selection of vertices may depend on algorithms and input data representations. This creates an urgent need for novel frames and algorithms to deal with this difficulty.

In this research we focus on the problem of proteolytic/RNA cleavage as well as the controllability of complex networks. and propose novel methods/survey on already existed tools on three types of problems: (i) Dicer cleavage site prediction, (ii) Evaluation on existed Caspase/Granzyme B prediction tools, (iii) MFVS based networks controllability.

Acknowledgements

First of all, I would like to express my sincere appreciation to my supervisor, Professor Tatsuya Akutsu, for his patient guidance, invariably giving me useful and valuable comments and advice throughout the course of this work. He patiently provided invaluable advice, constructive comments and warm encouragement for me through the doctoral program and writing of this thesis.

I shall extend my thanks to Professor Jose C. Nacher who has joined the study of Critical/redundant minimum feedback vertex analysis, for his kind and valuable help.

I am deeply grateful to Professor Jiangning Song who advised the study of Caspase prediction tool survey, for his great comments and suggestions.

My heartfelt thanks also goes to Associate Professor Morihiro Hayashida for his helpful advice, technological support, discussion and comments during the study.

Furthermore, I would like to thank the members in Akutsu Laboratory for their kindness during this work and thank my friends for giving me so many wonderful memories and their encouragement will always be remembered.

Finally, I would like to show my love and deepest gratitude to my parents for their financial and spiritual supports, without which I can not finish my study in Kyoto University.

Contents

List of Figures

List of Tables

List of Abbreviations

MiRNA	Micro RNA
SVM	Support Vector Machine
DGCR8	Di**G**eorge syndrome Chromosomal Region 8
PAZ domain	PIWI AGO Zwille domain
RISC	RNA Induced Silencing Complex
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SDS-PAGE	Sodium Dodecyl Sulfate PolyAcrylamide Gel Electrophoresis
TBS	Tris-Buffered Saline
TTBS	TBS-Tween
ROC curve	Receiver Operating Characteristic curve
AUC	Area Under the Curve of ROC
RBF	Radial Basis Function
SVR	Support Vector Regression
DS	Dominant Set
MDS	Minimum Dominant Set
ILP	Integer Linear Programming
FVS	Feedback Vertex Set
MFVS	Minimum Feedback Vertex Set
CMFVS	Critical Minimum Feedback Vertex Set
IMFVS	Intermittent Minimum Feedback Vertex Set
RMFVS	Redundant Minimum Feedback Vertex Set

Dedicated to my mother, Ying
Zhu. Her support,
encouragement, and constant
love have sustained me
throughout my life.

Chapter 1

Introduction

1.1 Background

1.1.1 Dicer and RNA interference

RNA interference is a canonical gene-silencing process that is essential in various cellular functions such as viral defense, developmental timing and stem cell maintenance. Dicer, which is a multidomain ribonuclease, plays an important role in this process. Dicer first cuts dsRNAs into small fragments which are typically divided into two kinds: microRNAs (miRNAs) and short interfering RNAs (siRNAs). Then Dicer assists these small fragment RNAs to transfer into another important complex called RNA-induced silencing complexes (RISC). RISC is then activated to lead these small RNAs to perform gene silencing.

Since Dicer is essential in the microRNA/short interfering RNA generation process, it is important to elucidate the mechanism of the selection of Dicer cleavage sites. Recently Gu *et al.* (2012) show that the loop/bulge parts are important in Dicer cleavage site selection. In accordance with this breakthrough, a support vector machine (SVM)-based method called PHDCleav was developed to predict Dicer cleavage sites which outperforms other methods based on random forest and naive Bayes. PHDCleav, however,PHDCleav tests only whether a position in the shift window belongs to a loop/bulge structure.

1.1.2 Apoptosis and caspases

Apoptosis is a kind of programmed cell death which is different from necrosis. The concept of apoptosis is discovered in 1972, to describe a morphologically distinct form of cell death. Recently tremendous researches about the mechanism of apoptosis have been presented which prove that this programmed cell death process is conserved from worm to human. This process is mainly induced by three kinds of proteins: the caspase family proteins, the Bcl-2 family proteins and the Apaf-1/CED-4 protein.

The caspase family is mainly consisted of three components: the apoptosis-related caspases including caspase 2, 3, 6, 7, 8, 9, 10, the Pyroptosis related caspases including caspase 1, 4, 5, 11, 12. and caspases not functioned in programmed cell death or not found in *Homo sapiens* including caspase 14.
All apoptotic caspases exist in normal cells as inactive enzymes. When the program cell death begins, these caspases are activated by some sequential proteolytic events. these events mainly cleave the single peptide precursor into fragments that could finally leading to the generation of active enzymes.
Though caspases are essential in processing programmed cell death, the native substrates of caspases still remains to be fully explored. Since experimental identification and characterization of caspase substrates are often time-consuming, expensive and difficult, various computation tools have been developed to predict caspase substrates to avoid such kind of demerit.

1.1.3 Controlling of biological networks

Analysis of biological networks has emerged to be an important aspect of the computational biology. Several kinds of interaction networks such as PPI (protein-protein interaction) networks and metabolic networks could be analyzed through various algorithms and tools. In these algorithms, we focus and introduce the concept of dominating set and minimum dominating set. These two concepts are developed in the early 1990s and have been applied to a rich variety of problems from computer and wireless communication networks to social systems.
Based on this approach Nacher and Akutsu (2014) develop an MDS based framework to identify critical vertices in biological networks, which has the advatage that significantly limits the number of critical vertices compared with the previous algorithm.

1.2 Motivation

Critical objects are important in the analysis in various bioinformatics problems. For various problems the critical objects denote various meanings. In this thesis we evaluate three kinds of critical objects including Dicer cleavage site prediction, whose critical objects are Dicer cleavage sites, Caspase cleavage site prediction, whose critical objects are Caspase cleavage sites and finally Controllability in directed biological networks, whose critical objects are critical vertices in biological networks.
In Chapter 2 we develop an improved algorithm named LBSizeCleav which is also

based on support vector machine framework and considers the effect of length of the loop/bulge structures. Since both PHDCleav and LBSizeCleav are based on support vector machine, we will give a brief introduction of support vector classification in the next section.

In Chapter 3 we make a summary to the state-of-the-art tools for caspase substrates prediction. This approach provides a way for bioinformaticians who are interested in designing and developing next-generation approaches for caspase/granzyme B cleavage prediction.

In Chapter 4 we develop a minimum feedback vertex set (MFVS) based framework for critical vertices analysis. Compared with the MDS based framework for critical vertices analysis, the MFVS based framework based algorithm significantly decreases the size of critical vertices, which enables us to control a network with fewer vertices. Since our MFVS based algorithm is developed after the MDS based framework, we will give a brief introduction of the MDS based framework in the next section.

1.3 Preliminaries

1.3.1 Support vector classification

We can define the classification problem to the classification of a two-class problem without the loss of generality. This problem is defined as the followings: we need to divide the two classes by a function that is deduced by some existed examples, and this function can be generalized to the other examples that do not belong to the examples used to generate this function. Generally there are many functions that could separate the data, but there is only one function that could separate the data and maximize the margin (i.e. maximize the distance between this function and the nearest data point of each class, Figure 1.1), this optimal separating function, or hyperplane, is defined below.

Given the problem of separating the set of training examples from two classes,

$$D = \{(x^1, y^1), ..., (x^l, y^l)\}, x \in R^n, y \in \{-1, 1\} \tag{1.1}$$

with a function/hyperplane:

$$< \omega, x > +b = 0 \tag{1.2}$$

and the examples are said to be optimally separated if they are separated without misclassification and the distance between the closest example to the hyperplane is maximized. Usually a separating hyperplane in canonical form must satisfy the

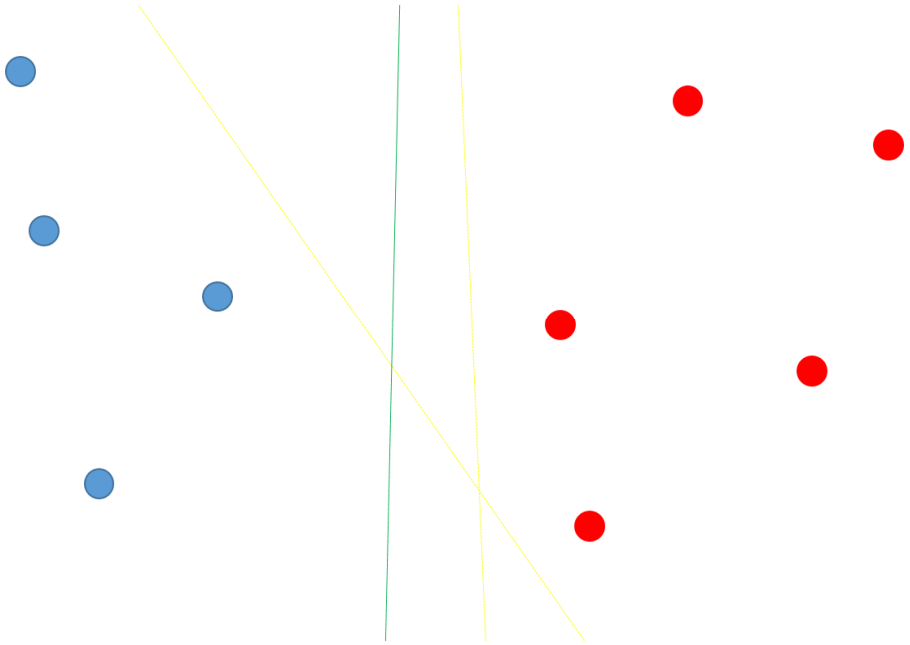


FIGURE 1.1: The optimal separating function and those not, where the green line shows the optimal separating function.

following equation:

$$y^i[< \omega, x^i > +b] \geq 1 \tag{1.3}$$

$$d = \frac{< \omega, x^i > +b}{\|\omega\|} \tag{1.4}$$

For support vector classification the optimal hyperplane is given by maximizing the margin, and this margin is given by the following equation:

$$\rho(\omega, b) = \min_{x^i : y^i = -1} d(\omega, b; x^i) + \min_{x^i : y^i = 1} d(\omega, b; x^i) \tag{1.5}$$

Since the deduction for how to maximize the margin is not used in the following part, we will only introduce the definitions of this problem.

Feature space mapping

Kernel function could be used as a way to construct a mapping from the low dimensional space to the higher dimensional space, which is a solution to the problem of the curse of dimensionality. The commonly used feature space mappings are listed as below:

Polynomial Function

The polynomial mapping is defined as:

$$K(x, x^{`}) =< x, x' >^{d} \tag{1.6}$$

Gaussian Radial Basis Function

The Gaussian RBF is a widely used mapping and the common form is defined as:

$$content...K(x, x') = exp(-\frac{\|x - x'\|^2}{2\sigma^2}) \tag{1.7}$$

Exponential Radial Basis Function

Exponential RBF of the form:

$$content...K(x, x') = exp(-\frac{\|x - x'\|}{2\sigma^2}) \tag{1.8}$$

could provide a piecewise linear solution which is useful when discontinuities are acceptable.
In Chapter 2 we utilize LIBSVM for support vector classification and select RBF as its kernel function. Here LIBSVM is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It supports multi-class classification.

1.3.2 Minimum dominating set

The minimum dominating set is a well-known concept in graph theory, we define a graph $G(V, E)$ consists of a set of vertices V and a set of edges E, each edge in the graph is directed. For this Graph a subset of vertices $S \in V$ is called a dominating set if every vertex in V is either an element of S or is adjacent to an element of S. A dominating set (DS) with a minimum number of elements is called a minimum dominating set. A minimum dominating set (MDS) is not necessarily uniquely determined for one graph G, as Figure 1.2 shows.

Computation of an MDS

Although MDS is an essential concept in graph theory, it has been proved that calculating the MDS is an NP-hard problem, which denotes that no polynomial-time algorithm that could efficiently compute an accurate minimum dominant set exists. however, NP-hardness does not necessarily mean that there is no fast algorithm to

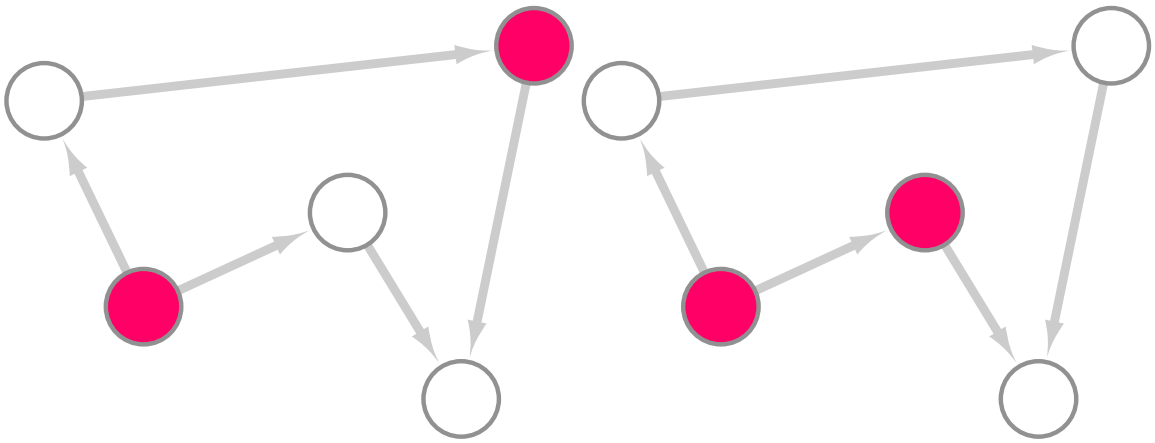


FIGURE 1.2: Examples for two different MDS configurations of one graph.

compute a minimum dominant set. (even it is not polynomial time computable) In fact, there are many algorithms that are recommended to accurately calculate the MDS with an acceptable time consuming. Besides many heuristic methods are also developed to improve the efficiency for minimum dominant set computation. In the exact calculation algorithms, the most widely used method is based on Integer linear programming (ILP).

Nacher and Akutsu (2014) transform the problem of computing an MDS into an ILP based form in a simple way and use CPLEX to perform the computation, which is a tool that is able to solve large-scale ILP instances. In this method they assign a 0-1 variable y_v to each vertex $v \in V$, where $y_v = 1$ denotes that v is in an MDS and $y_V = 0$ denoted that v is not. Then for a given graph $G(V, E)$, the following integer linear programming is constructed to compute an MDS for G:

$$\begin{aligned} &Minimize \quad \sum_{v \in V} y_v, \\ &Subject \quad to \quad y_v + \sum_{(u,v) \in E} y_u \geq 1 \quad for \quad all \quad v \in V, \\ &y_v \in \{0, 1\} \quad for \quad all \quad v \in V. \end{aligned} \tag{1.9}$$

and an MDS is given by the set $S = \{v | y_v = 1\}$.

The relationship between MDS and the structural controllability

Network controllability is one of an important research topics in complex networks. If we can define a set of vertices as driver vertices and drive the network using this set from any initial states to any desired states in finite time, then we say that this network is controllable. Nacher and Akutsu (2014) show the relationship between

MDS and structural controllability for linear systems by identifying vertices called critical vertices.

Critical/intermittent/redundant vertices

As discussed in the previous part, vertices in an MDS are considered to have important roles in controlling the whole network. However since MDS is not uniquely determined in a graph G, Nacher and Akutsu (2014) overcome this issue by dividing vertices in the whole network into three categories: critical vertices, intermittent vertices, and redundant vertices. They also show that the set of critical vertices can be computed by the following procedure:

1. Compute an MDS M for $G(V, E)$ using ILP.
2. Let C_{MDS} be an empty set.
3. Repeat steps 4–5 for all $v \in M$.
4. Make an ILP instance I_v by adding a constraint of $y_v \leq 0$ to the instance given by Eqs. 1.9.
5. If I_v does not have a feasible solution or $|M_v| > |M|$ holds where $M_v = \{v|y_v = 1\}$, then let $C_{MDS} \leftarrow C_{MDS} \cup \{v\}$.
6. Return C_{MDS}.

Similarly they also show that the redundant vertices can be computed by:

1. Compute an MDS M for $G(V, E)$ using ILP.
2. Let R_{MDS} be an empty set.
3. Repeat steps 4–5 for all $v \in V - M$.
4. Make an ILP instance I_v by adding a constraint of $y_v \geq 1$ to the instance given by Eqs. 1.9.
5. If I_v does not have a feasible solution or $|M_v| > |M|$ holds where $M_v = \{v|y_v = 1\}$, then let $R_{MDS} \leftarrow R_{MDS} \cup \{v\}$.
6. Return R_{MDS}.

1.4 Organization of the thesis

In Chapter 2 we use the length of loop/bulge structures (in addition to their presence or absence) to develop an improved method, LBSizeCleav, for predicting Dicer cleavage sites.

To evaluate our method, we perform prediction on 810 empirically validated sequences of human pre-miRNAs and perform fivefold cross-validation. In both 5p and 3p arms of pre-miRNAs, LBSizeCleav shows greater prediction accuracy than PHDCleav do. This result suggests that the length of loop/bulge structures is useful for prediction of Dicer cleavage sites.

In Chapter 3 we review and benchmark 12 state-of-the-art sequence-based bioinformatics approaches and tools for caspases/granzyme B cleavage prediction. We evaluate and compare these methods in terms of their input/output, algorithms used, prediction performance, validation methods, and software availability and utility. In addition, we construct independent datasets consisting of caspases/granzyme B substrates from different species and accordingly assess the predictive power of these different predictors for the identification of cleavage sites.

In Chapter 4, similar to the MDS based framework for analyzing critical vertices/intermittent vertices/redundant vertices, we present an algorithm as well as its implementation to compute and evaluate the critical, intermittent, and redundant vertices under the MFVS-based framework, where these three categories include vertices belonging to all MFVSs, some (but not all) MFVSs, and none of the MFVSs, respectively.

Chapter 2

LBSizeCleav: improved support vector machine (SVM)-based prediction of Dicer cleavage sites using loop/bulge length

2.1 Introduction

MicroRNAs (miRNAs) are a type of small RNAs with the length ∼22 nt, which perform the function of suppressing gene expression at the post-transcriptional level (Bartel, 2004, Bernstein *et al.*, 2001). Usually in vivo, a gene of a miRNA is transcribed to produce a long, primary miRNA (pri-miRNA) transcript, which is then processed into a ∼65-nt-long hairpin structure via cleavage by the Drosha (DGCR8) enzyme. Then, the resulting pre-miRNA is cleaved by another enzyme (termed Dicer) to generate a mature miRNA, which is ∼22 nt long (Lee *et al.*, 2002). Finally, the generated miRNA can be combined with an Argonaute protein to form the protein–miRNA complex, which can control various cellular progresses including development, cell death, and metabolism (Elbashir *et al.*, 2001, Hammond *et al.*, 2000, Zamore *et al.*, 2000).

Dicer is a 1922-amino acid multidomain protein that belongs to the RNase III family. Dicer generally contains several domains including ATPase–helicase, DUF283 (a double-stranded-RNA–binding domain), PAZ (Piwi–Argonaute–Zwille) domain, two RNase III domains, and a dsRBD (MacRae *et al.*, 2006). Dicer in various species may contain a different combination of these domains. Among these domains, the PAZ domain, RNase III domain, and dsRND are responsible for the function of substrate cleavage (Lau *et al.*, 2012). The cleavage occurs near the end of the terminal loop of pre-miRNA, introducing a cut into the hairpin.

Structural analysis of human Dicer revealed that the PAZ domain contains a 5p phosphate-binding pocket, which may be necessary for selection of a Dicer cleavage site (Park *et al.*, 2011). There are also studies showing that the loop/bulge structure also determines the accuracy of cleavage activity (Feng *et al.*, 2012, Gu *et al.*, 2012). MacRae et al. reported that the 3p-terminal nucleotide of single-stranded RNA can affect Dicer binding (MacRae *et al.*, 2007). In addition, Jin and Lee found that a single nucleotide polymorphism may be associated with miRNA regulation (Jin and Lee, 2013). All these studies revealed that secondary structures of both the Dicer enzyme and cleavage substrates are essential for cleavage site determination.

With a better understanding of the features of selection of a Dicer cleavage site, researchers may be able to elucidate the mechanism of action of enzymes in the RNA III family as well as the processes of RNA inference. Thus, it is imperative to explore the factors affecting the accuracy of Dicer cleavage to gain better insights into the mechanism of Dicer cleavage. Recently, a support vector machine (SVM)-based method (PHDCleav) was developed to predict selection of Dicer cleavage sites (Ahmed *et al.*, 2013). They proposed feature space mappings from pre-miRNA nucleotide sequences on the basis of existence of a predicted loop/bulge structure. SVM is a state-of-the-art machine learning technology (Drucker *et al.*, 1999) that has been applied to various areas of pattern recognition in many biological fields such as protein classification (Bhasin and Raghava, 2004, Cai *et al.*, 2003, Zavaljevski *et al.*, 2002), prediction of RNA secondary structure (Kumar *et al.*, 2008, Ng and Mishra, 2007), and drug–nondrug classification (Burbidge *et al.*, 2001, Byvatov *et al.*, 2003).

In this chapter, we made use of the length of loop/bulge structures and proposed a novel algorithm of feature space mapping called *LBSizeCleav*. To evaluate our method, we used 810 empirically valid sequences of pre-miRNAs from miRBase and performed fivefold cross-validation. In the 5p arm of pre-miRNAs, the proposed method attained higher accuracy (87.4%), whereas the best prediction result of PHDCleav corresponded to the accuracy of 84.0% (an extended binary pattern, a window of 14-nt size). In addition, in the 3p arm, the average prediction accuracy of our method reached 83.0%, whereas PHDCleav achieved up to 79.1% prediction accuracy. These results suggest that our method LBSizeCleav outperforms binary patterns of PHDCleav in predicting the position of Dicer cleavage sites. The better performance may in turn serve as the evidence that the features utilized by these two methods are necessary for Dicer cleavage selection.

2.2 Methods

In this section, we provide a brief description of feature space mapping algorithms of PHDCleav using sequences and secondary structures and propose a novel algorithm for feature space mapping, LBSizeCleav, based on the length of a loop/bulge structure.

2.2.1 Feature space mapping procedures of PHDCleav

Given a pre-miRNA sequence, a site between two successive nucleotides is mapped to a binary vector. In PHDCleav, a window is generated for each input sequence where for the positive pattern the center of the window is exactly located at the cleavage site of 5p (3p) arm and for the negative pattern the center of the window is located 6 nt away from the cleavage site of 5p(3p) arm. Since this is based on the assumption that a cleavage site can shift slightly (1-2 nt in biological experiments) but the chance is rare that Dicer cuts in the middle of mature miRNA, 6 nt could be changed under the principle that the center of the negative pattern is far enough from the real cleavage site. PHDCleav has shown that there is little affect to the accuracy of prediction even with the shifting of negative windows among the whole sequence of pre-miRNA.

A nucleotide in a window having the site at the center is converted to a four-dimensional vector as [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], and [0, 0, 0, 1], for A, U, C, and G, respectively (see Table 2.1). Let w denote the size of the window, where w is a positive even number. Then, a $4w$-dimensional vector is generated for the site.

There are many loops/bulges in the secondary structure of pre-miRNA where one arm contains extra nucleotides without counterparts in the other arm (Lyngsø *et al.*, 1999). A recent study indicated that these loops/bulges play an important role in the selection of a Dicer cleavage site (Gu *et al.*, 2012). This observation suggests that the loop/bulge structure may be a feature that is useful for prediction of a Dicer cleavage site. The extended binary pattern of PHDCleav was developed on the basis of this assumption.

After obtaining the secondary structure from a given sequence by some prediction methods, in the extended binary pattern of PHDCleav, a nucleotide is converted to a five-dimensional vector as [1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, 1, 0], and [0, 0, 0, 0, 1], for A, U, C, G, and L, respectively, where L indicates that the corresponding nucleotide is predicted to be in a loop/bulge structure. Just as the nucleotides in the window, its complementary nucleotides are also converted to a feature vector. After that, the dimensionality of the vector is $10w$.

TABLE 2.1: Binary patterns for nucleotides, A, U, C, G, and a loop/bulge structure, denoted by L, in PHDCleav (Ahmed *et al.*, 2013) and LBSizeCleav with k ones based on sequences and predicted secondary structures.In PHDCleav binary patterns each nucleotide is represented by a 4-dimensional vector, and in PHDCleav Extended patterns each nucleotide is represented by a 5-dimensional vector, while in LBSizeCleav the dimension of the vector is $3+k+N$, in which N denotes the maximum number of length of loop/bulges among all the pre-miRNAs in the training dataset.

mapping		sequence	structure
PHDCleav	A	$[1,0,0,0]$	$[1,0,0,0]$
	U	$[0,1,0,0]$	$[0,1,0,0]$
	C	$[0,0,1,0]$	$[0,0,1,0]$
	G	$[0,0,0,1]$	$[0,0,0,1]$
	L	–	$[0,0,0,0]$
Extended PHDCleav	A	–	$[1,0,0,0,0]$
	U		$[0,1,0,0,0]$
	C		$[0,0,1,0,0]$
	G		$[0,0,0,1,0]$
	L		$[0,0,0,0,1]$
LBSizeCleav	A	–	$[1,0,0,0,0,\ \ldots\ 0]$
	U		$[0,1,0,0,0,\ \ldots\ 0]$
	C		$[0,0,1,0,0,\ \ldots\ 0]$
	G		$[0,0,0,1,0,\ \ldots\ 0]$
	L		$[0,0,0,0,0,\ldots,0,\overbrace{1,\ldots,1}^{k},0,\ldots]$

2.2.2 Feature space mapping of LBSizeCleav

It is reasonable to consider not only the position but also the length of loop/bulge structures. Therefore, we propose novel feature space mapping (LBSizeCleav) by introducing the length of a loop/bulge structure into the algorithm.

The binary pattern of LBSizeCleav is an extension of that of PHDCleav. Let M be the maximal length of loops and bulges of all the pre-miRNAs in a dataset, and suppose L_l indicates that the corresponding nucleotide is in a loop/bulge structure of length l. Here we introduce a new parameter named k into LBSizeCleav, which is a positive integer representing the effect of length of loops and bulges to the kernel computation. Then, we designate a nucleotide without any loop/bulge structure for k as a $(M+k+3)$-dimensional vector, namely, [1, 0, 0, 0, ..., 0], [0, 1, 0, 0, ..., 0], [0, 0, 1, 0, 0, ..., 0], [0, 0, 0, 1, 0, ..., 0] for A, U, C, and G, respectively (see Table 2.1). A nucleotide in a loop/bulge structure of length l is represented as [0, ..., 0, 1, ..., 1, 0, ...], where k ones appear from the $(4+l)$-th element to the $(k+3+l)$-th element. Thus, for window size w, a $2w(M+k+3)$-dimensional vector is generated.

Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be binary patterns of L_{l_1} and L_{l_2}, respectively. If we use the inner product for kernel computation, then the inner product between the binary patterns

is $\boldsymbol{x}_1 \cdot \boldsymbol{x}_2 = \max\{k - |l_1 - l_2|, 0\}$. If we use the radial basis function (RBF) kernel, $\exp\{-\gamma||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2\} = \exp\{-4\gamma \min\{(l_1 - l_2)^2, k^2\}\}$, where $\gamma > 0$. These values assume the maximum when $l_1 = l_2$ and decrease according to the difference $|l_1 - l_2|$ and k, while k gets larger, the value changes of kernel function is more sensitive to the size of $|l_1 - l_2|$, in this way by controlling the value k we could control the sensitivity of our method to length of loops and bulges. Since PHDCleav used radial basis function (RBF), we also selected RBF as our kernel function.

Figure 2.1 illustrates the feature space mapping of LBSizeCleav for the pre-miRNA of the miRBase ID hsa-miR-200c with a predicted secondary structure, where nucleotides in the region removed by Dicer are shown as lowercase letters. CD-5p and CD-3p denote cleavage sites in 5p and 3p arms, respectively. Sequences in the red rectangles denote sequences used to generate feature vectors representing 5p and 3p arms, which are selected by the principle that the cleavage site is located at the center of the sequence. Here, we generate the feature vector of LBSizeCleav at $k = 3$ and $w = 6$ for the site CD-5p and for the site 6 nt away from CD-5p. The nucleotides in the window in the 5p arm are UGGgug, and loop/bulge structures are detected at two positions. As a result, L_1GGgL_1g is converted to the $6(M + 6)$-dimensional binary vector, where loop/bulge structures L_l are inserted. For the 3p arm, CGUCAU is converted in accordance with Table 2.1.

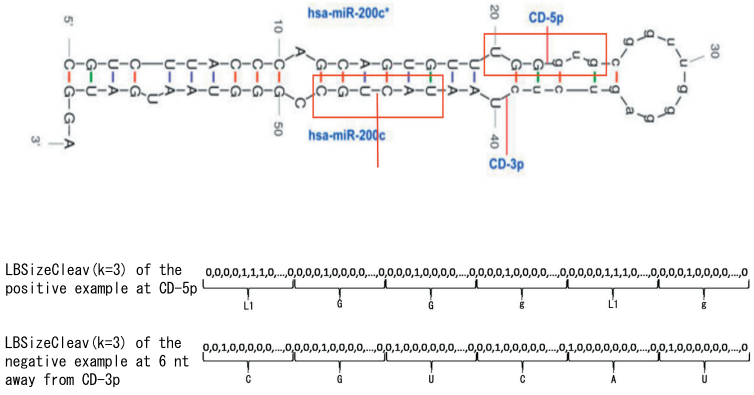


FIGURE 2.1: Illustration on the feature space mapping of LBSizeCleav. CD-5p and CD-3p denote cleavage sites in 5p and 3p arms, respectively. For two sites of CD-5p and six nucleotides far from CD-3p, the feature vectors of LBSizeCleav with $k = 3$ and $w = 6$ are shown, the red rectangles represent the window of the positive pattern of CD-5p and the window of the negative pattern of CD-3p.

2.3 Results

We retrieved 810 empirically validated sequences of pre-miRNAs from miRBase (version 21) (Griffiths-Jones *et al.*, 2006), where cleavage sites CD-5p and CD-3p are both defined for each pre-miRNA. The pre-miRNAs were selected under the principle that both the cleavage sites of CD-5p and CD-3p are experimentally validated. (i.e. only precursors with cleavage sites at both CD-5p and CD-3p are selected, we made this choice to let our dataset be generated the in same way as dataset of PHDCleav) All the pre-miRNAs are selected from human pre-miRNAs.

We used these cleavage sites as positive examples using windows of size 8,10,12,14 nt, where each window was selected so that a cleavage site is located at the center of the window, and we generated negative examples on the same sequence so that both centers of the positive and negative examples were 6 nt away from each other, as in the previous study (Ahmed *et al.*, 2013). This approach is based on the assumption that for most pre-miRNAs, the Dicer cleavage site is seldom selected at the center of the hairpin structure. In PHDCleav, two secondary structure predictors, quikfold (Markham and Zuker, 2008) and RNAFold from ViennaRNA (Hofacker, 2003) were used, hence, we used both the RNAFold from ViennaRNA and the quikfold server (version 3.0, http://mfold.rna.albany.edu/?q=DINAMelt/Quikfold) for prediction of RNA secondary structures. The results were given in Tables 2.2 and 2.3. Because in PHDCleav, the accuracy of prediction by nucleotide composition was worse than that by binary patterns, we compared our method with the binary patterns of PHDCleav. We used the libSVM 3.18 package (Chang and Lin, 2011) with the RBF kernel to utilize SVM because the RBF kernel was used in PHDCleav.

The performance of prediction methods was assessed by means of sensitivity, specificity, accuracy, and the Matthews correlation coefficient (MCC), defined as follows:

$$sensitivity = \frac{TP}{TP+FN}, \tag{2.1}$$

$$specificity = \frac{TN}{TN+FP}, \tag{2.2}$$

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN}, \tag{2.3}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \tag{2.4}$$

where TP, TN, FP, FN denote the number of true positive, true negative, false positive, and false negative results, respectively.

We performed fivefold cross-validation, and used the average sensitivity, specificity, accuracy, and MCC. We examined size w of a window from 8 to 14 and the number k of ones in LBSizeCleav from 1 to 5 for 5p and 3p arms of pre-miRNAs.

TABLE 2.2: Results on average specificity, sensitivity, accuracy, and MCC for both 5p and 3p arms by five-fold cross-validation using PHDCleav and LBSizeCleav ($k = 1, \cdots, 5$) with window sizes $8, 10, 12, 14$ based on sequences and secondary structures predicted by quikfold server. Sn, Sp, Ac, and MCC denote sensitivity, specificity, accuracy, and Matthews correlation coefficient, respectively.

method	window size	5p arm				3p arm			
		Sn	Sp	Ac	MCC	Sn	Sp	Ac	MCC
PHDCleav (sequence)	8	0.602	0.503	0.552	0.105	0.662	0.625	0.644	0.287
	10	0.541	0.573	0.557	0.115	0.661	0.642	0.652	0.303
	12	0.560	0.555	0.557	0.115	0.660	0.656	0.658	0.316
	14	0.539	0.572	0.555	0.111	0.654	0.702	0.678	0.356
PHDCleav (structure)	8	0.753	0.814	0.784	0.568	0.670	0.661	0.665	0.330
	10	0.784	0.827	0.806	0.612	0.702	0.719	0.710	0.421
	12	0.790	0.842	0.816	0.633	0.739	0.764	0.752	0.503
	14	0.799	0.857	0.828	0.657	0.779	0.783	0.781	0.562
Extended PHDCleav	8	0.750	0.798	0.774	0.548	0.652	0.716	0.684	0.369
	10	0.779	0.827	0.803	0.607	0.674	0.783	0.729	0.460
	12	0.809	0.845	0.827	0.654	0.714	0.790	0.752	0.506
	14	0.813	0.868	0.840	0.682	**0.781**	0.801	0.791	0.582
LBSizeCleav ($k = 1$)	8	0.668	0.924	0.796	0.612	0.630	0.684	0.657	0.315
	10	0.709	0.947	0.828	0.675	0.651	0.776	0.713	0.430
	12	0.774	0.945	0.859	0.730	0.686	0.847	0.766	0.540
	14	0.808	0.933	0.871	0.747	0.758	0.874	0.816	0.637
LBSizeCleav ($k = 2$)	8	0.662	**0.954**	0.808	0.645	0.626	0.723	0.674	0.351
	10	0.725	0.946	0.835	0.688	0.642	0.806	0.724	0.455
	12	0.784	0.938	0.861	0.731	0.665	0.882	0.773	0.560
	14	0.820	0.925	0.872	**0.749**	0.734	0.916	0.825	0.661
LBSizeCleav ($k = 3$)	8	0.692	0.949	0.821	0.664	0.619	0.735	0.677	0.356
	10	0.752	0.941	0.846	0.706	0.618	0.822	0.720	0.450
	12	0.803	0.932	0.867	0.741	0.635	0.914	0.774	0.571
	14	0.825	0.912	0.869	0.740	0.719	**0.942**	**0.830**	**0.678**
LBSizeCleav ($k = 4$)	8	0.695	0.949	0.822	0.667	0.614	0.736	0.675	0.353
	10	0.767	0.938	0.853	0.716	0.621	0.835	0.728	0.467
	12	0.815	0.927	0.871	0.747	0.639	0.912	0.776	0.573
	14	0.835	0.909	0.872	0.746	0.723	0.924	0.823	0.660
LBSizeCleav ($k = 5$)	8	0.700	0.947	0.824	0.668	0.594	0.771	0.682	0.371
	10	0.772	0.936	0.854	0.717	0.578	0.862	0.720	0.459
	12	0.821	0.924	0.872	**0.749**	0.634	0.921	0.777	0.579
	14	**0.838**	0.909	**0.874**	**0.749**	0.724	0.932	0.828	0.671

Table 2.2 shows the results of PHDCleav and LBSizeCleav ($k = 1, \cdots, 5$) based on sequences and secondary structures predicted by the quikfold server. In terms of prediction performance in the 5p arm of pre-miRNA, the best result of PHDCleav corresponded to the accuracy of 84.0%, whereas LBSizeCleav at $k = 5$ achieved the accuracy of 87.4%. In addition, the values of prediction accuracy of LBSizeCleav at $w = 12, 14$ were higher than those of PHDCleav. As for prediction performance in the 3p arm of pre-miRNA, the best result of PHDCleav corresponded to the accuracy of 79.1%, whereas LBSizeCleav achieved the accuracy of 83.0%.

Table 2.3 shows the results of PHDCleav and LBSizeCleav ($k = 1, \cdots, 5$) based on sequences and secondary structures predicted by the RNAFold. In terms of prediction performance in the 5p arm of pre-miRNA, the best result of PHDCleav corresponded

TABLE 2.3: Results on average specificity, sensitivity, accuracy, and MCC for both 5p and 3p arms by five-fold cross-validation using PHDCleav and LBSizeCleav ($k = 1, \cdots, 5$) with window sizes $8, 10, 12, 14$ based on secondary structures predicted by RNAFold. Sn, Sp, Ac, and MCC denote sensitivity, specificity, accuracy, and Matthews correlation coefficient, respectively.

method	window size	5p arm				3p arm			
		Sn	Sp	Ac	MCC	Sn	Sp	Ac	MCC
Extended PHDCleav	8	0.746	0.744	0.745	0.490	0.772	0.750	0.761	0.522
	10	0.792	0.783	0.787	0.575	0.779	0.800	0.790	0.580
	12	0.798	0.799	0.798	0.597	0.785	0.830	0.808	0.616
	14	0.778	0.813	0.795	0.591	0.805	**0.853**	0.829	0.659
LBSizeCleav ($k = 1$)	8	0.739	0.805	0.772	0.545	0.785	0.790	0.788	0.576
	10	0.798	0.820	0.809	0.618	0.795	0.815	0.805	0.610
	12	0.792	0.815	0.803	0.607	0.822	0.840	0.831	0.662
	14	0.815	**0.822**	**0.819**	**0.638**	0.851	0.852	**0.851**	**0.703**
LBSizeCleav ($k = 2$)	8	0.753	0.788	0.771	0.542	0.792	0.788	0.790	0.580
	10	0.816	0.795	0.806	0.612	0.811	0.794	0.803	0.606
	12	0.836	0.784	0.810	0.621	0.814	0.803	0.808	0.617
	14	0.845	0.769	0.807	0.616	0.867	0.800	0.834	0.669
LBSizeCleav ($k = 3$)	8	0.751	0.794	0.773	0.546	0.784	0.797	0.790	0.581
	10	0.808	0.808	0.808	0.615	0.795	0.813	0.804	0.608
	12	0.822	0.800	0.811	0.623	0.808	0.835	0.821	0.643
	14	0.816	0.803	0.809	0.619	0.853	0.838	0.846	0.692
LBSizeCleav ($k = 4$)	8	0.764	0.772	0.768	0.536	0.809	0.772	0.790	0.581
	10	0.824	0.762	0.793	0.587	0.824	0.766	0.795	0.590
	12	0.841	0.737	0.789	0.581	0.842	0.756	0.799	0.600
	14	0.871	0.678	0.774	0.559	0.898	0.697	0.797	0.607
LBSizeCleav ($k = 5$)	8	0.782	0.747	0.764	0.529	0.822	0.744	0.783	0.568
	10	0.836	0.732	0.784	0.572	0.829	0.726	0.777	0.558
	12	0.867	0.699	0.783	0.574	0.864	0.682	0.773	0.556
	14	**0.899**	0.626	0.763	0.546	**0.917**	0.619	0.768	0.562

to the accuracy of 81.3%, whereas LBSizeCleav at $k = 1$ achieved the accuracy of 82.2%. As for prediction performance in the 3p arm of pre-miRNA, the best result of PHDCleav corresponded to the accuracy of 82.9%, whereas LBSizeCleav achieved the accuracy of 85.1%.

To better evaluate the performance we also calculated the variance of each prediction result in Table 2.4. Figure 2.2, 2.3 show the results of LBSizeCleav and PHDCleav on receiver-operator characteristic (ROC) curves at window size $w = 14$ in 5p and 3p arms. Judging by the performance evaluation, our newly developed method outperformed the binary patterns of PHDCleav; this finding was suggestive of efficiency of the feature representing the length of loop/bulge structures.

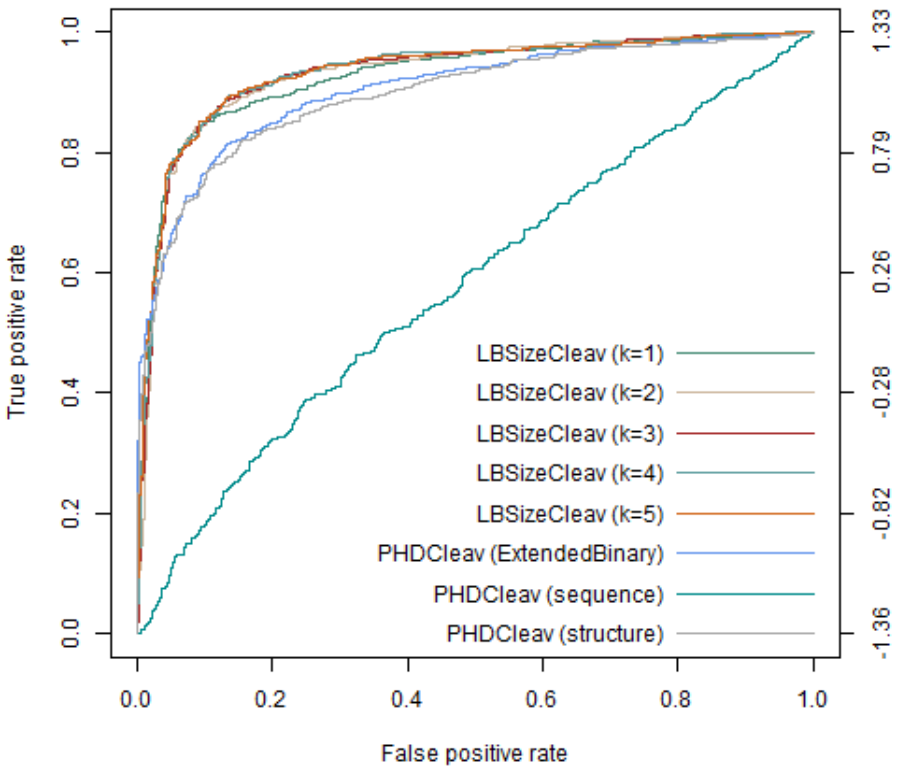
Since in our results, LBSizeCleav with parameters of $w = 14, k = 5$ outperforms the others, we selected these parameters as our parameters for prediction model. For an input sequence, we created a shift window of size 14 nt shifting from the 5p arm to the 3p arm. For each shift window we performed an SVM regression analysis using our model. Here we randomly selected 2 precursors from the dataset and showed

TABLE 2.4: Variances of specificity, sensitivity, accuracy, and MCC for both 5p and 3p arms by five-fold cross-validation using PHDCleav and LBSizeCleav ($k = 1, \cdots, 5$) with window sizes $8, 10, 12, 14$ based on sequences and secondary structures predicted by quikfold server. Sn, Sp, Ac, and MCC denote sensitivity, specificity, accuracy, and Matthews correlation coefficient, respectively.

feature extraction method	window size	CD-5p				CD-3p			
		Sn	Sp	Ac	Mc	Sn	Sp	Ac	Mc
PHDCleav (sequence)	8	0.0137	0.0008	0.0074	0.0036	0.0072	0.0015	0.0094	0.0066
	10	0.0184	0.0004	0.0111	0.0018	0.0044	0.0005	0.0072	0.0024
	12	0.0208	0.0001	0.0223	0.0003	0.0078	0.0011	0.0031	0.0046
	14	0.0293	0.0009	0.0174	0.0037	0.0065	0.0007	0.0048	0.0029
PHDCleav (structure)	8	0.0042	0.0039	0.0067	0.0155	0.0187	0.0013	0.0091	0.0062
	10	0.0026	0.0043	0.0088	0.0177	0.0100	0.0014	0.0050	0.0059
	12	0.0042	0.0027	0.0034	0.0109	0.0051	0.0011	0.0024	0.0045
	14	0.0047	0.0031	0.0034	0.0125	0.0039	0.0012	0.0014	0.0047
Extended PHDCleav	8	0.0029	0.0032	0.0063	0.0128	0.0123	0.0025	0.0043	0.0103
	10	0.0030	0.0038	0.0061	0.0154	0.0064	0.0019	0.0016	0.0075
	12	0.0040	0.0033	0.0050	0.0136	0.0054	0.0015	0.0011	0.0059
	14	0.0059	0.0027	0.0016	0.0108	0.0032	0.0013	0.0010	0.0052
LBSizeCleav($k = 1$)	8	0.0030	0.0025	0.0044	0.0115	0.0100	0.0004	0.0074	0.0019
	10	0.0022	0.0015	0.0013	0.0064	0.0078	0.0011	0.0015	0.0042
	12	0.0050	0.0024	0.0010	0.0090	0.0077	0.0015	0.0002	0.0051
	14	0.0075	0.0035	0.0010	0.0132	0.0036	0.0007	0.0002	0.0026
LBSizeCleav($k = 2$)	8	0.0042	0.0018	0.0008	0.0066	0.0053	0.0010	0.0036	0.0041
	10	0.0038	0.0020	0.0009	0.0076	0.0034	0.0010	0.0010	0.0039
	12	0.0051	0.0029	0.0016	0.0115	0.0042	0.0017	0.0008	0.0063
	14	0.0051	0.0028	0.0012	0.0107	0.0043	0.0008	0.0002	0.0024
LBSizeCleav($k = 3$)	8	0.0025	0.0013	0.0008	0.0050	0.0070	0.0010	0.0021	0.0042
	10	0.0039	0.0022	0.0012	0.0086	0.0064	0.0013	0.0006	0.0048
	12	0.0063	0.0031	0.0012	0.0119	0.0039	0.0015	0.0005	0.0055
	14	0.0060	0.0033	0.0015	0.0130	0.0073	0.0016	0.0003	0.0046
LBSizeCleav($k = 4$)	8	0.0029	0.0016	0.0009	0.0064	0.0066	0.0020	0.0030	0.0078
	10	0.0046	0.0025	0.0011	0.0095	0.0071	0.0014	0.0006	0.0049
	12	0.0061	0.0032	0.0014	0.0124	0.0033	0.0011	0.0007	0.0041
	14	0.0051	0.0030	0.0015	0.0120	0.0088	0.0025	0.0002	0.0082
LBSizeCleav($k = 5$)	8	0.0029	0.0017	0.0009	0.0066	0.0062	0.0021	0.0031	0.0082
	10	0.0055	0.0029	0.0013	0.0113	0.0032	0.0011	0.0005	0.0042
	12	0.0051	0.0029	0.0015	0.0114	0.0029	0.0011	0.0009	0.0044
	14	0.0047	0.0029	0.0016	0.0116	0.0076	0.0023	0.0002	0.0076

the score of the extended binary pattern of PHDCleav and LBSizeCleav with $k = 5$. From the result we could see that although both tools have predicted the cleavage site correctly, LBSizeCleav predicted more true negatives than extended binary pattern of PHDCleav, which indicates a better performance in identifying negative patterns of LBSizeCleav (see in Figure 2.4).

We also compared the performance of our tools with another state-of-art method, a recent published paper introduced an easy way named SGL (Simple Geometric Locater) to calculate the cleavage site of miRNA which outperforms other methods. We generated a benchmark to compare our method as well as PHDCleav with SGL of prediction in CD-5p, which result is shown in Figure 2.5. In this benchmark we selected the threshold (0.0) of LBSizeCleav as well as PHDCleav and calculated the EAEs(End Absolute Error, the absolute error of the predicted minus the true position



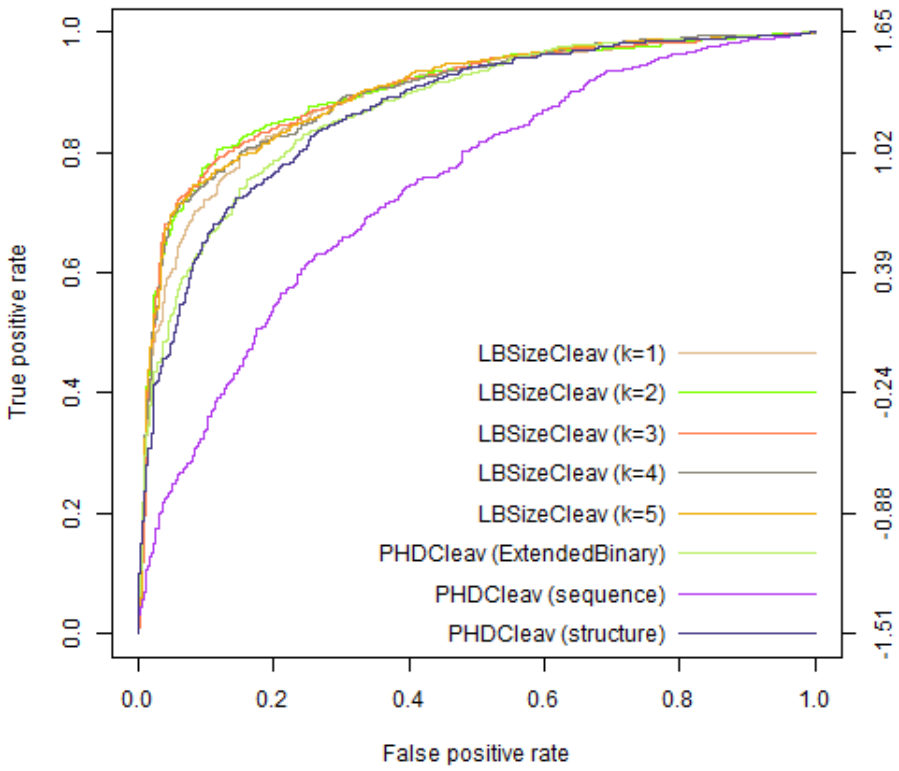
(A) 5p arm

FIGURE 2.2: Results on ROC curves by LBSizeCleav and PHDCleav with window size $w = 14$ for 5p arm. From the figure we could see that the ROC curve of LBSizeCleav from $k = 1$ to $k = 5$ is significantly better than binary Pattern and extended binary pattern of PHDCleav for both 5p and 3p arms.

for a specific duplex end) from the true cleavage site and compared it with the SGL method. From the result we could see that although at high EAEs PHDCleav outperforms LBSizeCleav, LBSizeCleav outperforms both PHDCleav and SGL at EAE 1, which indicates that LBSizeCleav predicted less false positives than PHDCleav.

2.4 Discussion

There were several pre-miRNAs, such as pre-mir221, pre-mir138-1, and pre-mir-15a, that were identified by LBSizeCleav but were not identified by PHDCleav in the prediction results from the 5p arm of a pre-miRNA with a shift window of 14 nt (see Figure 2.6). By comparing these three pre-miRNAs, we found that all of them contain a part of loop/bulge structures that is more than 1 nt long in their mature parts. This result indicates that the length of a loop/bulge structure is an



(A) 3p arm

FIGURE 2.3: Results on ROC curves by LBSizeCleav and PHDCleav with window size $w = 14$ for 3p arm. From the figure we could see that the ROC curve of LBSizeCleav from $k = 1$ to $k = 5$ is significantly better than binary Pattern and extended binary pattern of PHDCleav for both 5p and 3p arms.

important determinant of a cleavage site. Careful analysis revealed that pre-mir221 and pre-mir138-1 contain their loop/bulge structures in their bulge parts, whereas pre-mir-15a has its loop/bulge structure in its loop part, proving that both loop and bulge structures can affect the cleavage site selection. To further evaluate the effect of the length of a loop/bulge structure in affecting the cleavage site selection we calculated the number of pre-miRNAs which LBSizeCleav identified successfully while PHDCleav failed to identify and vice versa (see Table 2.5). From the result we could see that for the positive patterns our method performs almost the same as PHDCleav, but for negative patterns our method shows a significant improvement in accuracy. This result indicates that our method has a better resolution in identifying negative patterns.

TABLE 2.5: Number of patterns predicted only by LBSizeCleav ($k = 1, 4$)/PHDCleav (extended binary) using secondary structure predicted by quikfold.

		5'-arm	3'-arm
Positive	Only predicted by LBSizeCleav ($k = 1$) compared with PHDCleav (extended binary)	39	39
	Only predicted by PHDCleav (extended binary) compared with LBSizeCleav ($k = 1$)	57	38
Negative	Only predicted by LBSizeCleav ($k = 1$) compared with PHDCleav (extended binary)	82	65
	Only predicted by PHDCleav (extended binary) compared with LBSizeCleav ($k = 1$)	23	12
Positive	Only predicted by LBSizeCleav ($k = 4$) compared with PHDCleav(extended binary)	39	39
	Only predicted by PHDCleav compared with LBSizeCleav ($k = 4$)	57	38
Negative	Only predicted by LBSizeCleav ($k = 4$) compared with PHDCleav(extended binary)	82	65
	Only predicted by PHDCleav (extended binary) compared with LBSizeCleav ($k = 4$)	23	12

FIGURE 2.4: Regression analysis examples of LBSizeCleav($k = 5$) compared with PHDCleav extended binary.

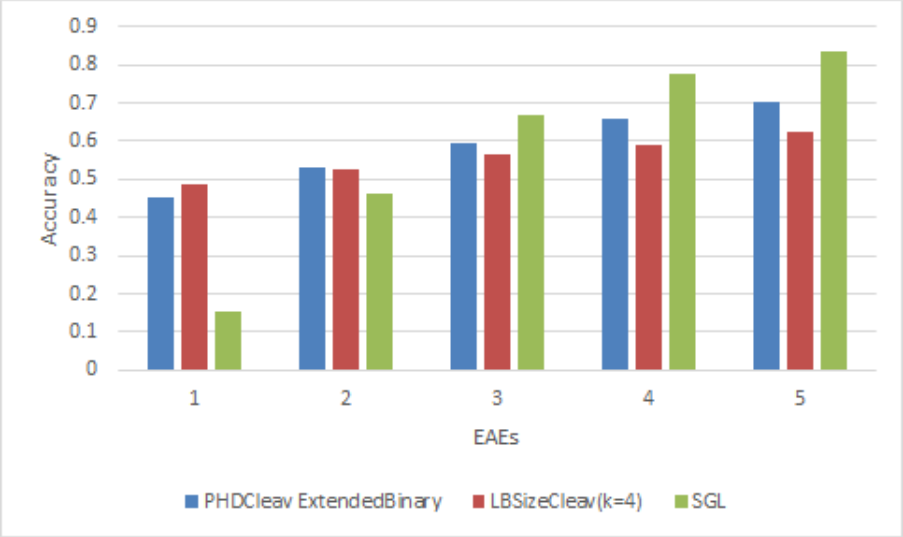


FIGURE 2.5: Result on accuracy of LBSizeCleav($k = 5$) compared with PHDCleav extended binary and SGL of prediction in CD-5p.

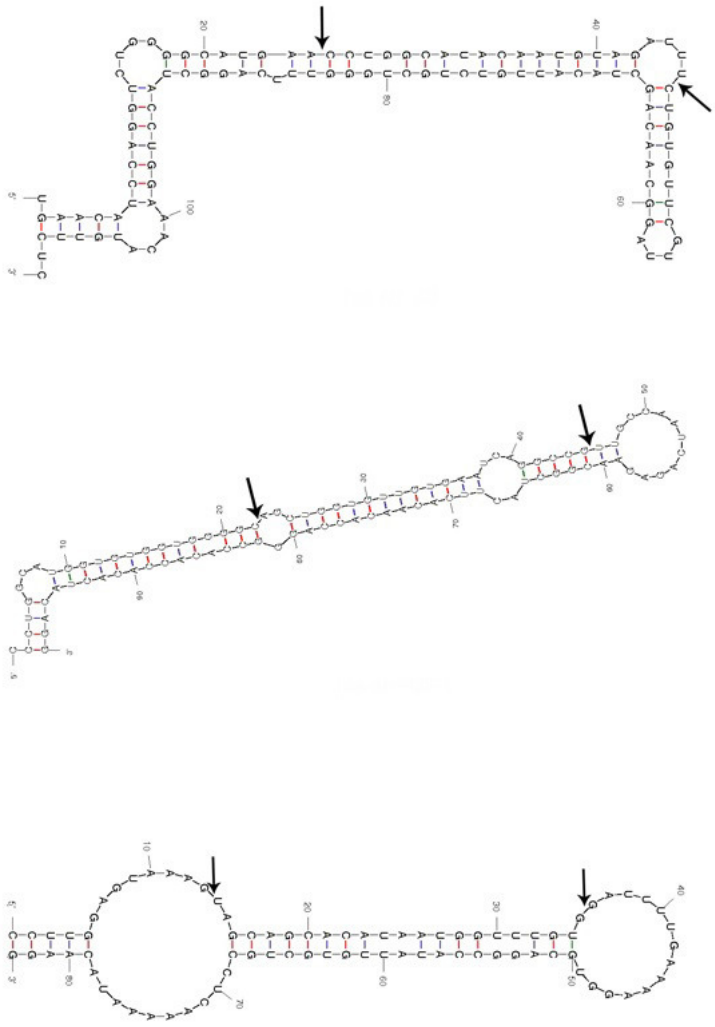


FIGURE 2.6: Secondary structures of hsa-mir-221, hsa-mir-138-1, hsr-mir-15a predicted by quikfold server. The black arrow means the cleavage site validated by biological experiments.

Chapter 3

Towards more accurate prediction of caspase cleavage sites: a comparative study of current methods, tools and features.

3.1 Introduction

Proteases are proteolytic enzymes that catalyze the breakdown of protein or peptide substrates by hydrolysis of peptide bonds (Adam, 1996, Adams, 2004, Anwar and Saleemuddin, 2001, Bonfil and Cher, 2011, duVerle and Mamitsuka, 2011, Lohmüller *et al.*, 2003, Mizianty and Kurgan, 2011, Nicholson and Thornberry, 1997, Wang *et al.*, 2016). They represent 2% (at least 500-600 proteases) of all gene products in human and are involved in the functional regulation of a large number of important physiological processes such as cell cycle (Jenal and Fuchs, 1998, Li and Hochstrasser, 1999), cell proliferation (Gerdes *et al.*, 1991), programmed cell death (Boldin *et al.*, 1996, Cardone *et al.*, 1998, Krajewska *et al.*, 1997), DNA replication (McGarry and Kirschner, 1998), tissue remodeling (Pellettieri *et al.*, 2010), and immune response (Franchi *et al.*, 2009, Muruve *et al.*, 2008). The members in this class of enzymes recognize specific substrate protein sequences and catalyze the hydrolysis of peptide bonds to activate or degrade the substrate proteins. The effects of the hydrolysis reactions are frequently amplified, resulting in a rapid and substantial change of the biological systems through modulating the balance of proteomic dynamics. Such highly orchestrated sequence of events are thus termed 'proteolytic cascades' (Cerenius *et al.*, 2010).

Caspases are a family of proteases that can be subdivided functionally into those involved in either apoptosis or inflammation (Cohen, 1997). In 1993, researchers found that the *Caenorhabditis elegans* cell death gene, ced-3, has a remarkable sequence similarity to interleukin-1b-converting enzyme (caspase-1), a mammalian protease responsible for proteolytic maturation of pro-interleukin-1b (Yuan *et al.*, 1993). This finding elucidated the first two members of the caspase family and provided evidence that these proteinases might play an essential role in apoptosis. Subsequent studies of these proteinases drove the identification of several other caspase family members important for apoptosis or inflammation.

Apoptosis, or programmed cell death, is a fundamental process that occurs in all tissues during development, homeostasis, and disease (Ashkenazi *et al.*, 1998, Barry and Bleackley, 2002, Bortner *et al.*, 1995, Rathmell and Thompson, 2002). On the other hand, the inflammatory response is triggered by innate immune sensors following cellular damage, infection, or stress, and serves to clear the harmful stimulus and initiate healing (Dostert *et al.*, 2008, Everett and McFadden, 1999).

To date, at least 15 mammalian caspases have been identified (Earnshaw *et al.*, 1999) and they are categorized into three groups, based on their substrate specificities: group I caspases (caspase-1, 4, 5 and 13) prefer bulky hydrophobic amino acids at the P4 site and cleave the peptide sequence (W/L)EHD, group II caspases (caspase-2, 3 and 7) preferentially cleave the sequence motif DEXD, whereas group III caspases (caspase-6, 8, 9 and 10) cleave the motif (I/V/L)E(H/T)D. In contrast to the caspases, granzyme B, another essential serine protease for apoptosis/inflammation, prefers to cleave the sequence motif IEXD (Thornberry *et al.*, 1997).

For caspases not falling into groups I, II and III, caspase-11 is considered as an orthologue of caspase-4 and 5 found in Murine. Activation of caspase-11 leads to septic shock, pyroptosis, and often organismal death. Caspase-12 is closely related to caspase-1 and the activating form of caspase-12 is only found in people of African descent in *Homo Sapiens*. Caspase-14 is enriched in human skin and mainly expressed in the upper layers of the epidermis. The protein is mainly localized to the cytosol according to the Cell Atlas.

Caspases are essential to coordinating and integrating signals which lead to apoptosis, inflammation and other forms of programmed death, including pyroptosis and necroptosis (Lauber *et al.*, 2003, Salvesen and Dixit, 1999). This view is supported by observations that proteins involved in apoptosis and inflammation contain common conserved domains, including caspase-associated recruitment domains (CARDs) and death effector domains (DEDs), which are also present in caspases. Recent findings

have indicated that classically 'apoptotic' caspases have essential roles in initiating inflammation, both directly and via inflammatory cell death pathways (Creagh *et al.*, 2003).

The specificity of proteases like caspases depends primarily on their active sites, whose selectivity depends on preferences for a number of specific amino acids at defined positions. In addition to the primary amino acid sequence of the substrate, the substrate specificity of a protease is also influenced by the three-dimensional conformation of its substrates. In particular, proteases preferentially cleave substrates within extended loop regions, while residues that are buried within the interior of the protein substrate are usually inaccessible to the protease active site.

Identification of native substrates of caspases and granzyme B is the key to the understanding of their physiological roles, implicated in the pathological processes contributing to proteolytic cascades, and leading to apoptotic cell death. Identification of native substrates also means to find potential substrates that can serve as viable therapeutic targets. Although the application of advanced large-scale high-throughput proteomic techniques has significantly increased the number of experimentally verified caspase and granzyme B substrates, the complete repertoire of the native substrates remains to be discovered, and furthermore, many other cleavage sites within the known substrates are not fully experimentally identified. Moreover, experimental identification and characterization of protease substrates are often time-consuming, expensive and requiring extensively trained personnel. Therefore, bioinformatic prediction of caspase and granzyme B substrates may provide valuable and experimentally testable information regarding novel potential cleavage sites or putative substrates, i.e., ranking the candidate protein target list according to their likeliness, narrowing it down to a reasonable number to be validated in the test tube.

Sequence and structural analysis of substrates of caspases and granzyme B has enabled the development of computational approaches for prediction of potential cleavage sites and putative substrates from sequence alone (Song *et al.*, 2011, Wee *et al.*, 2006, 2009) using techniques for analyzing protein sequences (Bhasin and Raghava, 2005, Chauhan *et al.*, 2010, Gromiha and Ou, 2013, Suresh *et al.*, 2015). However, the rapid growth in prediction approaches since the last comprehensive comparison (Wang *et al.*, 2013), which was reported almost five years ago, creates a need to critically assess and compare the expanding and diverse bundle of prediction methods. In this chapter, therefore, we present a comprehensive review of 12 sequence-based methods for caspases/granzyme B cleavage prediction, offering insights into the nature of different predictors and facilitating potential improvement of caspases/granzyme B

cleavage prediction. All predictors are critically reviewed in terms of input/output, algorithm, prediction performance, validation method and software utility i.e. whether a standalone software is available. To evaluate the performance of caspases/granzyme B cleavage predictors, we assembled independent testing datasets containing substrates of caspases/granzyme B of various species with carefully collected and curated data.

To address whether the predicted caspase substrate is really cleaved by caspase or not, we selected top-scoring sequences and test cleavage in vitro with a caspase assay. These sequences are top-ranked by prediction tools showing outstanding performances in our independent testing datasets.

3.2 Methods

3.2.1 Existing tools reviewed in this study

We briefly summarize the key aspects of the 12 tools evaluated for caspase/granzyme B cleavage prediction in Table 3.1. The tools included in the benchmarking analysis are GraBCas (Backes *et al.*, 2005), CaSPredictor (Garay-Malpartida *et al.*, 2005), PoPS (Boyd *et al.*, 2005), SitePrediction (Verspurten *et al.*, 2009), Cascleave (Song *et al.*, 2010), Cascleave 2.0 (Wang *et al.*, 2013), Pripper (Piippo *et al.*, 2010), PCSS (Barkan *et al.*, 2010), CASVM (Wee *et al.*, 2007), CAT3 (Ayyash *et al.*, 2012), PROSPER (Song *et al.*, 2012) and Blast (Altschul *et al.*, 1997).

TABLE 3.1: A summary of key features of each tool evaluated in this chapter. These features include applicable species, whether webserver exists, algorithm utilized, whether the batch prediction option is available, whether threshold is adjustable, whether stand alone software exists, programming language used to implement the program, the origins of training dataset, ratio of positive and negative samples, sliding window size (if exists), computing time to process one sequence, and whether solvent accessibility (SA) and secondary structure (SS) is considered. The '-' option means not available or not mentioned in the original paper.

Tools	SitePrediction	Cascleave	PoPS	Pripper
Species	Multi-species	Multi-Species	Multi-Species	Multi-Species
Webserver availability	http://www.dmbr.ugent.be/prx/bioit2-public/SitePrediction/	http://sunflower.kuicr.kyoto-u.ac.jp/~sjn/Cascleave/	http://pops.csse.monash.edu.au/	No server
Algorithm	Combination of frequency score representing amino acids occurrence and position similarity	BEAA(Binary encoding amino acid sequence profiles) trained and tested SVR model	PSSM matrix	Combination of SVM/random forest and J48 algorithm
Option of batch prediction	Yes	No	Yes	Yes
Adjustment of prediction thresholds	No	No	Yes	No
Standalone software availability	No	No	Yes	No

Language implemented	C++	Perl	Java	Java
Dataset origin	Data from MEROPS	Multiple resources	Data from MEROPS	Data from EBI (Apweiler *et al.*, 2001)
Ratio of positive to negative samples	-	1:3	-	1:1
Sliding window size	-	16 amino acids	-	10 amino acids
Computing time for processing a sequence	Within a second	5 min	Within a second	Within a second
Whether structural information considered	Secondary structure prediction, solvent accessibility and PEST sequence occurrence considered	Secondary structure , solvent accessibility and natively disordered regions considered	Secondary or tertiary structure of the substrate considered	Not considered
Types of caspases applicable	Specific training sets corresponding to caspases 1,3,6,7,8	Mixed training sets for all caspases	Mixed training sets for all caspases	Mixed training sets for all caspases

Tools	CAT3	PCSS	Blast	PROSPER
Species	Multi-species	Multi-Species	N.A.	Multi-Species
Webserver availability	No webserver	http://salilab.org /peptide	N.A.	https://prosper.erc. monash.edu.au /webserver.html
Algorithm	PSSM matrix	SVM with RBF kernel	N.A.	BEAA trained and tested SVR model with RBF kernel combined with MDGI feature selection
Option of batch prediction	Yes	Yes	N.A.	No
Adjustment of prediction thresholds	Yes	Yes	N.A.	No
Standalone software availability	Yes	No	N.A.	No
Language implemented	Perl	-	N.A.	Perl
Dataset origin	Data from PubMed (Acland *et al.*, 2014)	Multiple resources	N.A.	Data from MEROPS, CutDB and PMAP (Igarashi *et al.*, 2008)
Ratio of positive to negative samples	-	-	N.A.	1:3
Sliding window size	-	-	N.A.	6 amino acids
Computing time for processing a sequence	Within a second	A few minutes	N.A.	A few minutes
Whether Structure information considered	Not considered	Regular secondary structure considered	N.A.	Secondary structure, solvent accessibility and native disorder considered
Types of caspases applicable	Training sets corresponding to caspases-3	Separated training sets for caspases and granzyme B	N.A.	Mixed training sets for all caspases

Tools	GraBCas	CasPredictor	CASVM	Cascleave 2.0
Species	Multi-species	Multi-Species	Multi-Species	Multi-Species
Webserver availability	Not available	Not available	Not available	Not available
Algorithm	Scoring matrices	BLOSUM 62 Substitution Matrix -based CCSearcher algorithm	SVM	Maximum relevance minimum redundancy and forward feature selection techniques trained SVM model
Option of batch prediction	-	-	-	-
Adjustment of prediction thresholds	-	-	-	-
Standalone software availability	-	-	-	-
Language implemented	Java	Visual Basic	Perl	Java
Dataset origin	-	Various databases including SwissProt (Boeckmann *et al.*, 2003), InterDom (Ng *et al.*, 2003), and Pfam (Bateman *et al.*, 2004)	Various resources	MEROPS
Ratio of positive to negative samples	-	-	-	1:1
Sliding window size	-	-	Three scanning window sizes are available: P4P1, P4P2' and P14P10'	-
Computing time for processing a sequence	-	-	-	-
Whether Structure information considered	Not considered	Not Considered	Not considered	Secondary structure, solvent accessibility and natively disordered regions considered
Types of caspases applicable	Specific training sets corresponding to caspases-3 and granzyme B	Mixed training sets for all caspases	Mixed training sets for all caspases	Mixed training sets for all caspases

3.2.2 Model input

In machine learning, the dataset is often divided into training and testing datasets. The training datasets are used to build a computational model to learn hidden patterns in the data. For caspase/granzyme B substrates, the data are usually collected from various databases, such as MEROPS peptidase database (Rawlings *et al.*, 2013), which contains over 410,000 listed cleavage peptidases as well as 28,000 inhibitors (physiological and non-physiological). Other databases are CutDB (Igarashi *et al.*,

2006), CaMPDB (duVerle *et al.*, 2010), TopFIND (Lange and Overall, 2011) and Degrabase (Crawford *et al.*, 2013). CutDB integrates 3,070 proteolytic events for 470 different proteases captured from public archives. CaMPDB contains sequences of calpains, substrates and inhibitors as well as substrate cleavage sites, collected from the literature. TopFIND contains more than 290,000 N- / C-termini and more than 33,000 cleavage sites. Degrabase comprises about 8000 unique N termini from more than 3200 proteins directly identified in subtiligase-based positive enrichment mass spectrometry experiments in healthy and apoptotic human cell lines.

The issue of biased prediction often occurs when there is an extensive overlap between the training and testing datasets. To avoid such bias, tools such as Pripper constructed training datasets according to careful selection criteria of the data (including removing the sequence redundancy between the training and test datasets, controlling the ratio of positive data to negative data, as well as performing multiple rounds of randomization tests, e.g. 10 repeats of 10-fold cross-validation). Apart from the sequence overlap between the training dataset and validation dataset, the issue of data imbalance also needs to be addressed. Use of an unbalanced dataset often leads to biased models that favor the prediction of the 'majority' class of samples. Most tools solve this issue by selecting the positive dataset from experimental databases and manually generating/sampling the negative dataset with different approaches. For example, Cascleave/Cascleave 2.0 address this issue by generating positive and negative datasets from substrate sequences using a local sliding window approach surrounding the experimentally verified cleavage sites and other residues that are found not to be cleaved by caspases, respectively. Cascleave also employs an under-sampling approach by reducing the size of the over-represented negative samples. Pripper creates the negative dataset by selecting negative training sequences generated from the same substrate sequences that are used for positive sequences. Both Cascleave/Cascleave 2.0 and PROSPER set the ratio of the positive to negative data to approximately 1:3. Pripper sets the ratio of the positive data to negative data to approximately 1:1. Similar efforts to minimize the unbalanced data (such as controlling the ratio of positive and negative data), though not explained in detail, can be observed in the development of other prediction tools such as PoPS.

3.2.3 Models construction and development

Early tools for caspase/granzyme B cleavage site prediction predict caspase/granzyme B cleavage sites from sequence information only, while more sophisticated tools developed more recently consider additional information such as secondary structure

information, hydrophilicity/hydrophobicity, as well as solvent accessibility and protein native disorder information. The prediction methods can be generally classified into two types, machine learning-based algorithms and statistical scoring method-based algorithms. Machine learning-based tools include CASVM, Pripper, PCSS, PROSPER, Cascleave, and Cascleave 2.0. While Statistical scoring method-based tools include GraBCas, CaSPredictor, PoPS, SitePrediction, CAT3 and Blast.

GraBCas is a scoring method based on position specific scoring matrices (PSSM). The PSSM is constructed based on experimentally determined substrate specificities. For computing the score in PSSM, GraBCas screens for tetrapeptides with Asp (D) at their last position (P1) in a given amino acid sequence. Given the tetrapeptide A4A3A2D (P4P3P2P1) of a potential cleavage site, its cleavage score for a given endopeptidase is computed by multiplying the corresponding matrix entries of A2 at position P2, A3 at position P3 and A4 at position P4. To improve the performance, GraBCas analyzes the amino acid distribution of known granzyme B and caspase-3 cleavage sites at positions P6-P2' (where P and P' mean residues C-terminal to the cleavage site as prime (P') site and N-terminal peptide residues as nonprime (P) site) taken from the literature. CaSPredictor, a tool published at the same time with GraBCas, developed a scoring algorithm named CCSearch (Caspase Cleavage Site searcher) which is based on three parameters. The first parameter is calculated from the BLOSUM62 Substitution Matrix. The second parameter is the relative frequency *f(i)* for each amino acid residue at position i (P4-P1) from annotated sequences. The last parameter is the PEST index, calculated by giving a value of 1 to the amino acids in the following set: Ser (S), Thr (T), Pro (P), Glu or Asp (E/D), Asn (N), and Gln (Q), which are the residues of PEST regions (Rogers *et al.*, 1986). There is an evidence that PEST-like sequences, rich in the aforementioned amino acids, if located in the upstream or downstream of the cleavage site, may contribute to the specificity for at least 60% verified caspase substrates (Rechsteiner and Rogers, 1996, Rogers *et al.*, 1986). PoPS is a tool based on a computational model built from three components. The first component is the number of subsites within the active site of the protease. The second component is the specificity profile of each subsite, assigning a value to each of the 20 amino acids based on the relative contribution of the amino acid at that subsite to the overall substrate specificity of the protease. The last component is the weight of the subsite (Boyd *et al.*, 2005). SitePrediction is a tool based on the idea that besides the occurrences of fixed consensus cleavage sites in the substrate sequence, a second score is calculated to improve the performance. This score is based on the similarity of the potential cleavage sites to the known

sites used. SitePrediction also makes use of extra features including PEST sequences, solvent accessibility and secondary structure which may also provide contribution to the prediction performance (Verspurten *et al.*, 2009). CAT3 is a PSSM matrix-based method developed in 2012 (Ayyash *et al.*, 2012). CAT3 exploits positional specific frequency matrices from the multiple sequence alignments of the relevant set of peptides. Each matrix consists of 14 rows, representing positions P9...P1P1'...P5', where a D amino acid is at the position P1. The 20 columns of the matrix represent the frequencies of each amino acid. CAT3 also utilizes two weighting systems in order to correct the probability of overrepresented and underrepresented amino acids in the frequency matrices to establish the scoring matrices: Calculating log odd ratio and Subtraction of negative control background, which also contributes to the accuracy of CAT3. Blast is an aligning tool ubiquitously utilized in proteomics. Assuming that target substrates share a similar sequence, it can be used as a rudimentary prediction tool (Song *et al.*, 2010). For our assessment, the cleavage score of a query test protein corresponds to the highest Blast bit score (a normalized aligned value, independent from sequence length and database size) with the known substrates in the training set. PROSPERous (Song *et al.*, 2018b) is a recently developed tool which uses a combination of various scoring functions as the input, including Nearest Neighbor Similarity (NNS), Amino Acid Frequency (AAF), WebLogo-based Sequence conservation (WLS), BLOSUM62 Substitution Index (BSI) as well as pairs of these function, namely AAF+NNS, WLS+BSI and NNS+WLS. More recently, an advanced version of PROSPER, termed iProt-Sub (Song *et al.*, 2018a), was developed to provide optimized cleavage site prediction models with a larger coverage of more proteases (up to 4 major protease families and 38 different proteases). iProt-Sub uses 11 different sequence encoding schemes in combination with a two-step feature selection procedure to remove the redundant features and improve the accuracy (Song *et al.*, 2018a).

Most of the recently developed tools for predicting caspase/granzyme B cleavage sites are based on the SVM algorithm. These tools include CASVM, Pripper, PCSS, Cascleave, Cascleave 2.0, PROSPER and iProt-Sub.

SVMs are classifiers that based on the maximization of the margin between classes. The data are considered as n-dimensional vectors and the algorithm finds a hyper-plane that separates vectors in different classes with a maximal margin. A kernel function can be used to map vectors of the original feature space to a higher dimensional space in which the data can always be linearly separated. Note that the selection of training data greatly affects the performance of an SVM classifier, therefore we will provide a detailed description of the way to select training data for each tool. CASVM

is trained with sequences from a dataset containing unique caspase cleavage sites, obtained from experimentally verified caspase substrates, and an equal number of 'non-cleavage' sites, i.e., random tetrapeptide sequences extracted elsewhere on the same substrate. The tetrapeptide sequences are selected with the upstream 10 residues up to P14 position and downstream 10 residues up to P10' position (i.e. the classifier is trained on a local window size of P14-P10' sites) from the substrates. The SVM model of Pripper is trained on a balanced dataset containing positive cleavage site samples gathered from 358 substrate proteins, and negative sequences generated from substrate sequences containing positive ones, specifically by selecting Asp (D) positions that have not been detected as caspase cleavage sites. Feature vectors consist of a fixed number of amino acids, encoded in a numerical form, incorporating both sides of the cleavage site. Each amino acid in the sequence is represented as an array of length 20 representing the 20 different amino acids. Only one element is set to one, identifying the amino acid in question, while the rest is set to zero. PCSS model is based on single cleavage sequences, each with eight features representing oligopeptides. To each residue a feature number by the formula *(n*20+i)* is assigned, where n represents the zero-based position in the peptide sequence of the residue, and i represents the position of the residue in line with a zero-based alphabetical ordering of all residues. In addition, PCSS also considers secondary structure features, native disorder feature, solvent accessibility feature calculated by DSSP (Kabsch and Sander, 1983), Disopred (Ward *et al.*, 2004) and PSI-PRED (Jones, 1999), respectively. Cascleave utilizes a feature extraction method named binary encoding amino acid sequence profiles (BEAA) and its extension to include relevant structural features. In BEAA of which is encoded, substrate sequences are transformed into n-dimensional vectors using an orthonormal encoding scheme, in which each amino acid is represented by a 20-dimensional binary vector composed of either zero or one elements. Similarly to PCSS, the structural information predicted by state-of-the-art algorithms, specifically, secondary structures, solvent accessibility and natively unstructured regions are incorporated into the model to improve the performance. Cascleave also employs a novel approach named Bi-profile Bayesian signature, which is reported to significantly improve performance in methylation sites prediction (Shao *et al.*, 2009). Similarly to Cascleave, Cascleave 2.0 considers various structural information, including (but not limited to) secondary structures, solvent accessibility, disordered region, and amino acid index (AAindex (Kawashima and Kanehisa, 2000)). AAindex consists of a list of amino acid indices representing various physicochemical and biochemical properties.

Cascleave 2.0 also involves an over- and under-represented feature enrichment analysis, . The rationale is that for each protein substrate, the set of various heterogeneous features generated above are highly dimensional, heterogeneous, noisy and redundant and thus removing redundant features and employing more relevant features might be useful for improving the predictive performance. Inclusion of noisy and redundant leads to a time-consuming practice to train classifiers, thereby resulting in possible biased model training and prediction. Cascleave 2.0 automatically estimates and eliminates noisy features. PROSPER, like Cascleave, is an SVM-based method; it can be applied to a broader range of proteases. Compared to Cascleave, PROSPER utilizes a feature selection method called mean decrease Gini index (MDGI) within the random forest algorithm which can generate a score quantifying the importance and contribution of the individual element of a feature vector for correctly classifying a residue into a cleavage site or non-cleavage site. The MGDI feature selection step has proven useful for improving the prediction accuracy (Ebina *et al.*, 2010), and is particularly useful for large training datasets.

3.2.4 Performance evaluation

To assess the performance of the compared methods, several cross-validation approaches are usually utilized, including N-fold, leave-one-out, and leave-family-out. In addition, we look into prediction details by performing a case study. Cross-validation is typically exploited to avoid over-fitting the training dataset. Cross-validation consists in splitting the dataset into N folds and combine N-1 folds as the training dataset, while the left dataset is regarded as test dataset. Leave-one-out and leave-family-out are specific cases of N-fold cross-validation. Given a dataset with D data samples, leave-one-out cross-validation combines D-1 samples to form the training dataset and leaves the remaining one sample as the test sample. By iteratively selecting test sample, each sample in the dataset is used as a test sample once. On the other hand, in the leave-family- out cross-validation, if the dataset is collected from different species/families, each subset from the same species/family is iteratively selected and regarded as test datasets once, while other subsets will be combined to form the training dataset. As each sample/subset is iteratively selected as the test set, we need to perform the prediction many times with different combination of training datasets. We then average these results (usually accuracy) and acquire the final performance for cross-validation tests. Among the evaluated tools, Cascleave, Cascleave 2.0, PROSPER utilize the 5-fold cross-validation, while Cascleave 2.0 utilizes the leave-one-out cross-validation (LOOCV) to assess their performance. Performing an

independent test is another way to evaluate the performance of bioinformatics tools. In particular, it involves applying the algorithm to an independent test dataset with a different data distribution, e.g. data obtained from other experiments. Finally, case study, or experimental validation of predictions, is another effective way to test the performance of a prediction tool in real-world applications, providing useful information of the scalability and usefulness of a tool on unknown data. Here, we perform both independent test and case study to assess and compare different methods.

Experimental Validation

Recombinant target substrate proteins are incubated with active caspase-3 (Genetex) or active caspase-8 (Biovision) at 37°C for 2h. After incubation, proteins are separated on 10% SDSPAGE gels and transferred to nitrocellurose membrane. Transferred membrane is blocked with 4% blockace at 4°C for 2h and incubated with anti-myc antibody (9E10) for 1:2,000 in 0.2% blockace in TBS containing 0.02% tween 80 (TTBS). After incubation, membrane is incubated with peroxidase conjugated anti-mouse Ig for 1:5,000 in 0.2% blockace in TTBS for 1h. Bound antidoby is visualized by supersignal west pico (PIERCE) acoording to manufacture's instruction and LAS4000 mini (Fuji).

Predictor utility

An important consideration for developing practically useful predictors in the biological research community is to provide a user-friendly web interface or a local software tool, to enable non-bioinformaticians to apply the model directly to their own data. The usefulness of bioinformatics tools depends on three main factors, i.e. the web interface, the output and interpretation of prediction results and the availability of local executable software. A user-friendly interface can provide appropriate guidance and instructions for users to avoid making potential mistakes when exploiting the web server. This is particularly important when parameter settings are required before conducting prediction tasks. Among the predictors we tested, SitePrediction, Cascleave, PCSS and Blast have implemented web servers. All these tools require to provide parameters regarding penalty, prediction algorithm, error handling as well as email address where the prediction results will be sent. Specifically, SitePrediction requires an input sequence be submitted in the FASTA format and the type of protease can be optionally selected, and multiple substrate protein sequences can be predicted at the same time. Cascleave requires users to input substrate sequences that need to be predicted, as well as algorithms used for performing the prediction.

TABLE 3.2: Detailed description of the eight test datasets used in this study.

Test set name	Positive or negative	Test set description
Cas1-all	Positive set	Combination of caspase-1 substrates from *Homo sapiens*, *Mus musculus* and *Escherichia coli* extracted from MEROPS.
	Negative set	Combination of protein from *Homo sapiens*, *Mus musculus* and *Escherichia coli* excluding caspase-1 substrates.
Cas3-all	Positive set	Combination of caspase-3 substrates from *Homo sapiens*, *Mus musculus* and *Escherichia coli* extracted from MEROPS.
	Negative set	Combination of protein from *Homo sapiens*, *Mus musculus* and *Escherichia coli* excluding caspase-3 substrates.
Cas1-homo	Positive set	Caspase-1 substrates from *Homo sapiens* extracted from MEROPS.
	Negative set	Protein excluding caspase-1 substrates from *Homo sapiens*.
Cas3-homo	Positive set	Caspase-3 substrates from *Homo sapiens* extracted from MEROPS.
	Negative set	Protein excluding caspase-3 substrates from *Homo sapiens*.
Cas1-mus	Positive set	Caspase-1 substrates from *Mus musculus* extracted from MEROPS.
	Negative set	Protein excluding caspase-1 substrates from *Mus musculus*.
Cas3-mus	Positive set	Caspase-3 substrates from *Mus musculus* extracted from MEROPS.
	Negative set	Protein excluding caspase-3 substrates from *Mus musculus*.
Cas1-coli	Positive set	Caspase-1 substrates from *Escherichia coli* extracted from MEROPS.
	Negative set	Protein excluding caspase-1 substrates from *Escherichia coli*.
Cas3-coli	Positive set	Caspase-3 substrates from *Escherichia coli* extracted from MEROPS.
	Negative set	Protein excluding caspase-3 substrates from *Escherichia coli*.

In addition, the user also needs to specify the email address to acquire the prediction results. Since Cascleave dose not support batch submission, users can submit only one protein sequence to Cascleave each time. PCSS requires users to specify the training data as well as classifying data to perform the prediction, and also lists some pre-generated models for executing a quick prediction.

On the other hand, stand-alone software allows users to perform predictions for a large amount of sequences on local machines, offering an advantage over web servers. However, there exists also a burden of installing (and perhaps compiling) the software locally, along with the dependent libraries. In addition, if the input sequence data is too large, there exists a possibility that the local resources may not be sufficient enough to run the program properly. Among the predictors we tested, Pripper and CAT3 are stand-alone software tools written using Java whereas PoPS provides a JNLP file for downloading and local usage.

3.3 Results

3.3.1 Independent test and performance evaluation

In this section, to assess the performance of the reviewed tools in an objective and fair manner, we constructed independent test sets of caspase-1 and 3 substrates for *Homo sapiens*. To evaluate the performance of these tools on other species, we also constructed independent test sets of caspase-1 and 3 substrates for *Mus musculus* and *Escherichia coli*.

Note that since some of the tools are not accessible (i.e., not implemented as web servers, nor downloadable), we were forced to limit our assessment to the available ones: PoPS, SitePrediction, Cascleave, Pripper, PCSS, CAT3, and Blast.

Test dataset construction

For each of the three species, we extracted all the fasta sequences from MEROPS of release 12.0 (Rawlings *et al.*, 2013). Training datasets for each tool and independent test datasets should have a minimum overlap, because a large overlap will likely result in an overestimation of the performance and biased prediction outcome. We therefore eliminated sequences that were overlapped in the training datasets of prediction tools, including Cascleave and Cascleave 2.0 from the independent test datasets. Both of these tools are recently developed, and thus it is understandable that their training dataset covered most of the extracted sequences (especially when compared with training datasets of tools developed in the early years). Our analysis showed that more than half of the extracted sequences were discarded for this reason, leaving 66 caspase-1 substrate sequences and 121 caspase-3 substrate sequences, respectively. In total, for all three species. For the negative datasets, we randomly selected proteins excluding those identified as substrates of caspase-1 or 3 of each species. To avoid biased performance evaluation, the size of negative datasets was set as the same as of positive datasets. These constructed independent test datasets are named as Cas1-all and Cas3-all, respectively.

We further divided these datasets according to the corresponding species these substrates belong to, resulting in another six datasets corresponding to caspase-1 and caspase-3 substrates of the three species. Each of these sets is named as Cas1-homo, Cas3-homo, Cas1-mus, Cas3-mus, Cas1-coli and Cas3-coli, respectively. We notice that for caspase-1 the sizes of cas1-mus and cas1-coli were too small to be used for an effective ROC evaluation, and thus we skipped these two datasets when drawing ROC curves. We also notice that CAT3 was designed only for predicting caspase-3 substrates, and thus we only performed the evaluation of CAT3 on the sets that comprised of caspase 1 substrates. The detailed description of the test sets used is shown in Table 3.2.

Performance comparison

Among the reviewed predictors, since PoPS only has one parameter (threshold), we set the threshold as 0 to obtain more available results (as a lower threshold leads to a larger number of predicted potential cleavage sites (Boyd *et al.*, 2005)). For Cascleave, several prediction models (or the combination of models) such as BEAA, BPBAA and BPBDISO were tested. Since the combination of BEAA, BPBAA and BPBDISO achieved the best in terms of the ROC curve, we chose this option for evaluation. For SitePrediction, different predefined training databases, corresponding

to various species, exist. It is therefore possible to choose a particular database for each test set. For caspase-1 substrates prediction, we chose 'caspase-1 for all species' training database, and for caspase-3 substrates prediction, we chose the 'caspase 3 for all species' training database to perform the evaluation. For Pripper, since vote option gets the highest performance on the ROC curve testing, we chose the 'vote' option and cut option as 'Full cut'. For PCSS, we chose the caspase option in the pre-generated model and selected the training iteration as 100. For Blast we used the default parameters to perform the prediction. Then we performed the prediction on each of the constructed independent test sets described in the test dataset construction section, and for each set we evaluated the performance using AUC values.

Figures 3.1 and 3.2 show the ROC curves of different tools assessed using the Cas1-all and Cas3-all test datasets, respectively. Cascleave, PoPS and Pripper outperformed other tools and achieved the best AUC values on the Cas1-all set (with an AUC value of 0.796 for Cascleave, 0.739 for PoPS and 0.655 for Pripper, respectively), while tools such as Blast which depends on the sequence similarity performed poorly in ROC performance. While on the Cas3-all set Cascleave, SitePrediction, and CAT3 achieved the best AUC values (with an AUC value of 0.693 for Cascleave, 0.711 for CAT3 and 0.754 for SitePrediction, respectively).

Figure 3.3 shows the ROC curves of different tools on the Cas1-homo set. PoPS, Cascleave and Pripper achieved the highest AUC values (PoPS with AUC value of 0.744, Cascleave with AUC value of 0.771 and Pripper with AUC value of 0.663, respectively). Figure 3.4 shows the ROC curves for the prediction result of the Cas3-homo set, for *homo sapiens*. SitePrediction, CAT3 and Cascleave achieved the best AUC values (SitePrediction with AUC value of 0.787, CAT3 with AUC value of 0.703 and Cascleave with AUC value of 0.745, respectively).

Figure 3.5 shows the ROC curves on the Cas3mus set. The ROC curves show that for the Cas3-mus set, SitePrediction, PoPS and Cascleave achieved the best performance in terms of AUC value, each with AUC value of 0.760, 0.712 and 0.729.

Figure 3.6 shows the ROC curves on the Cas3coli set. We can see from the ROC curves that SitePrediction, PoPS and CAT3 achieved the best AUC value (SitePrediction with AUC value of 0.702, PoPS with AUC value of 0.627 and CAT3 with AUC value of 0.638, respectively).

Combining the tool evaluation results in Table 3.1 and performance benchmarking results, we can draw the following conclusions:

SitePrediction achieves a better performance for general prediction (i.e. it provides a better performance for predicting caspase substrates from species excluding *Homo*

sapiens). This is perhaps due to the separation of the training sets provided by SitePrediction. The performance results in turn show that it is better to construct independent training sets for each species than to mix all sequences into a single training set. Considering the faster computing speed of SitePrediction, users are recommended to use SitePrecition to predict species excluding *Homo sapiens*.

While SitePrediction possesses such a merit, it is also flawed when it has to address predictions on *Homo sapiens*. From Table 3.3 we can see that while SitePrediction achieves the highest performance for caspase-3 substrate prediction, it performs poorly for caspase-1 substrate prediction. This indicates that it is generally better to use Cascleave to achieve the best performance on caspase-1 substrate prediction.

Although Cascleave provides the best performance on caspase-1 substrate prediction and an acceptable performance on caspase-3 prediction, the computational cost for Cascleave is a little higher compared with other tools. Moreover, in addition to the requirement of submitting fasta files, Cascleave only allows the submission of one sequence to the server each time, thereby limiting the batch prediction for Cascleave.

If users want to perform a caspase-specific (i.e. the types of the caspase substrate is limited) substrate prediction offline, they are advised to utilize CAT3 for caspase-3 substrate prediction and Pripper for caspase-1 substrate prediction, since these two tools come with an implemented local package, while the other tools with better performance such as Cascleave and SitePrediction can only be utilized online.

We also notice that Blast performs poorly for almost all the training set, this is perhaps because that the Blast predictions are not aiming at identifying specific cleavage sites, but in general work by identifying homologous protein sequences as a whole that are similar to known cleavable sequences. The result of Blast is specifically generated according to the following steps: for a given caspase, a positive set and a test protein, the test protein is aligned using Blast against all the proteins in the positive set, then the best E-value is taken as the Blast prediction score, based on which Blast can predict the test protein to be a target substrate for the given caspase according to the highest similarity to the known targets.

On the other hand, the Blast calculation steps also indicate that Blast may perform better than specialized algorithms in certain cases (Fig. 6); however, in terms of false positive rates, it can not identify the cleavage site. Furthermore, even if not designed for identifying small motifs, Blast can find an overall sequence similarity which might result from a common ancestry, and therefore share a common function, such as the whole protein being a cleavage target for caspases.

The best AUC values for each prediction tool on each test set are summarized

TABLE 3.3: Summary of the top three tools that achieved the highest performance of AUC values for each set evaluated. The datasets used include are Cas1-all, Cas3-all, Cas1-homo, Cas3-homo, Cas3-mus and Cas3-coli.

Dataset	Top three tools of the highest performance of AUC values		
Cas1-all	Cascleave (0.796)	PoPS (0.739)	Pripper (0.655)
Cas3-all	SitePrediction (0.754)	CAT3 (0.711)	Cascleave (0.693)
Cas1-homo	Cascleave (0.771)	PoPS (0.744)	Pripper (0.663)
Cas3-homo	SitePrediction (0.787)	Cascleave (0.745)	CAT3 (0.703)
Cas3-mus	SitePrediction (0.760)	Cascleave (0.729)	PoPS (0.712)
Cas3-coli	SitePrediction (0.702)	CAT3 (0.638)	PoPS (0.627)

in Table 3.3. In summary, performance comparison analysis on the independent test indicates that SitePrediction and Cascleave are the two best-performing tools and generally provide an overall best performance among all the tools compared.

Although the AUC value shows that the state-of-the-art tools are still limited in terms of their predictive performance, there is room to further improve the performance by including newly discovered caspase substrates into training sets, integrating new informative features and developing novel models.

3.3.2 Case study: caspase-3 and 8 substrate cleavage prediction

Selection of potential caspase substrates

To further evaluate each tool described in this research, we selected a number of potential substrates with high scores predicted by most tools in the human proteome, and experimentally validated these potential substrates for caspase-3 and caspase-8 by performing caspase assay experiments. In the independent test step the AUC values of caspase-1 and caspase-3 for *Homo sapiens* showed that among all the tools tested Cascleave, PoPS and SitePrediction achieved better performances than other tools (Table 3.3). Among these three tools Cascleave and SitePrediction achieved the best AUC values. We further notice that while on the Cas3-homo set SitePrediction achieved the best AUC value, but performed poorly on the Cas1-homo set (even not included in the top three tools), suggesting that SitePrediction might be more suitable for predicting caspase-3 substrates.

We first made a consensus-based decision for caspase prediction using the predictors that are capable of discriminating caspase substrates from non-caspase substrate based on the result described above. The predictors used for this purpose were SitePrediction, Cascleave and PoPS. During the prediction we notice that it took a longer time for Cascleave to complete the prediction process for a single protein sequence, thus it is reasonable to utilize SitePrediction and PoPS first to perform

a rough discrimination of the sequences in the proteome of *Homo sapiens* and apply Cascleave to predict specific cleavage sites within the proteins selected out. The detailed procedures can be found in Figure 3.7.

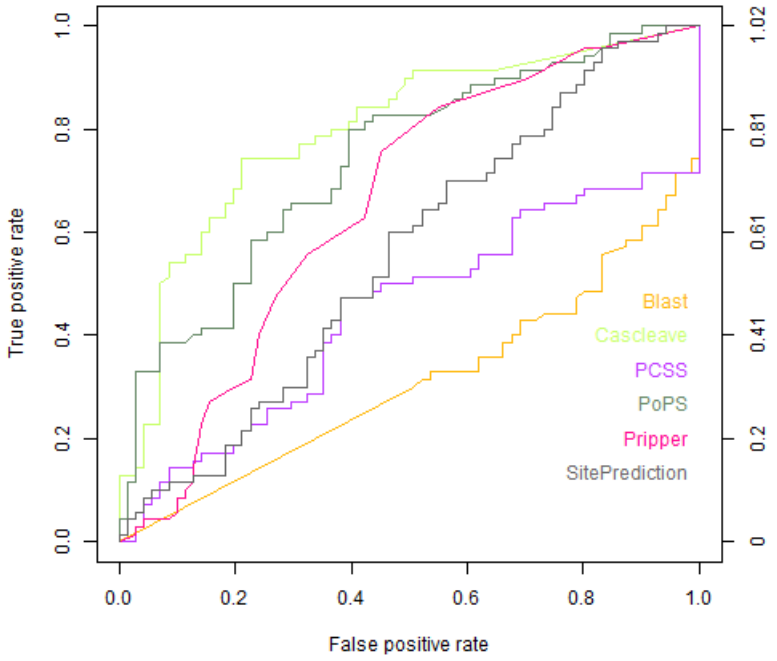


FIGURE 3.1: ROC curves of Blast, Cascleave, PCSS, PoPS, Pripper and SitePrediction on the Cas1-all set.

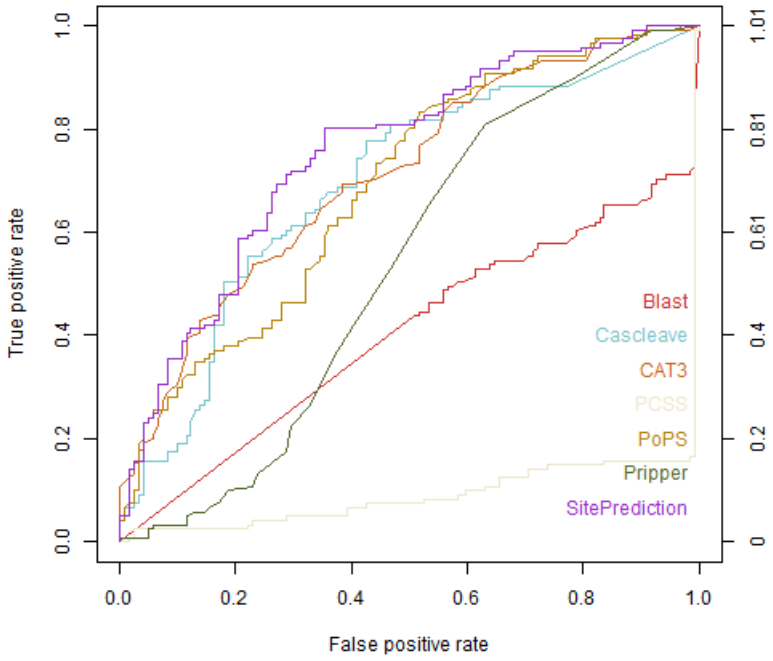


FIGURE 3.2: ROC curves of Blast, Cascleave, PCSS, PoPS, Pripper, CAT3 and SitePrediction on the Cas3-all set.

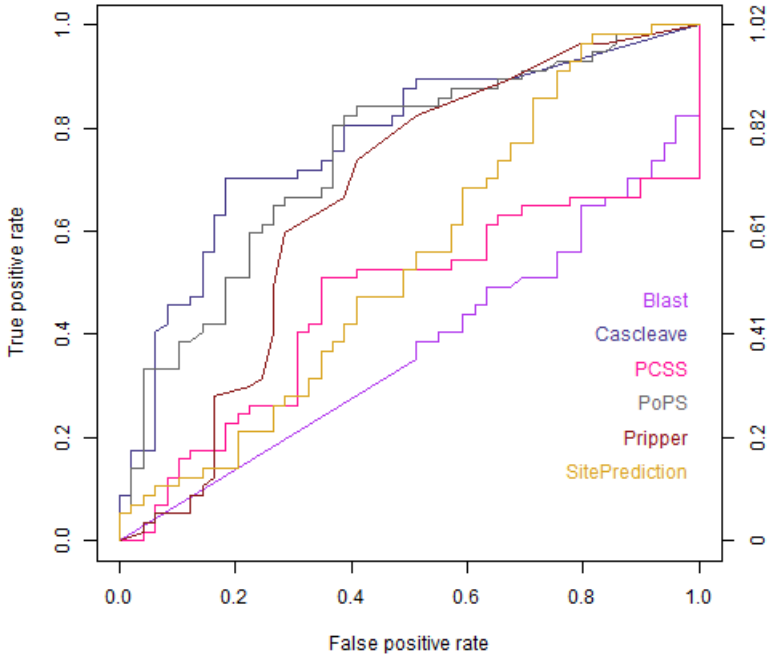


FIGURE 3.3: ROC curves of Blast, Cascleave, PCSS, PoPS, Pripper and SitePrediction on the Cas1-homo set.

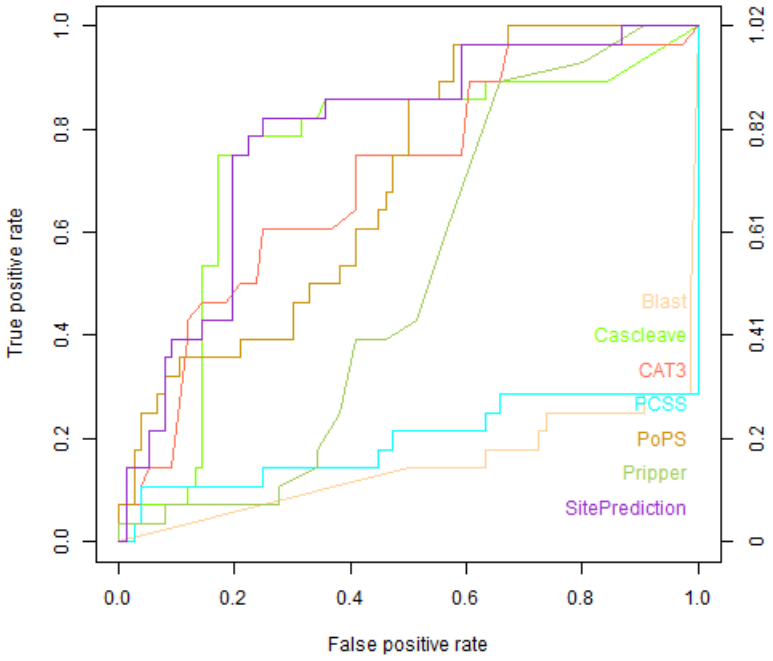


FIGURE 3.4: ROC curves of Blast, Cascleave, PCSS, PoPS, Pripper, CAT3 and SitePrediction on the Cas3-homo set.

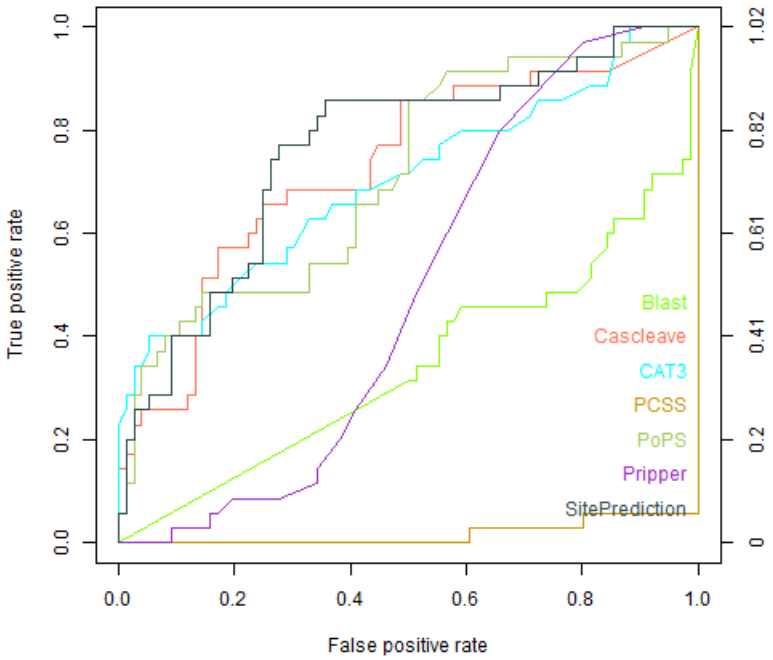


FIGURE 3.5: ROC curves of Blast, Cascleave, PCSS, PoPS, Pripper, CAT3, SitePrediction on the Cas3-mus set.

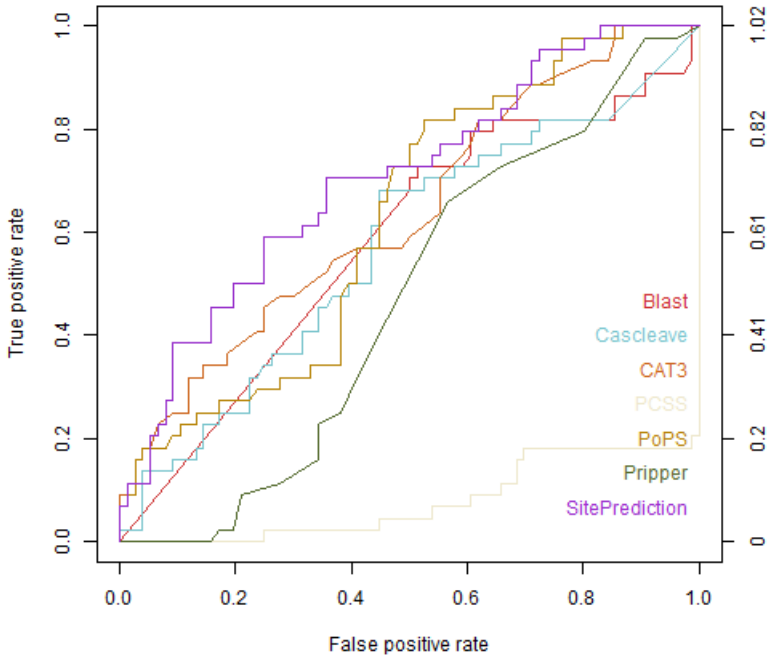


FIGURE 3.6: ROC curves of Blast, Cascleave, PCSS, PoPS, Pripper, CAT3, SitePrediction on the Cas3-coli set.

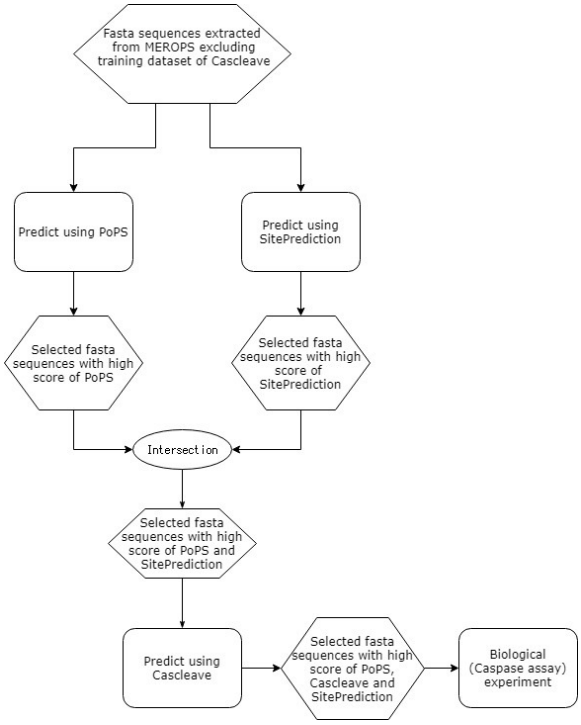


FIGURE 3.7: A flowchart of the procedures for caspase-3 and caspase-8 substrate cleavage site prediction of the human proteome.

Considering the fact that Cascleave performed best in discriminating caspase substrates from non-caspase substrates, we sorted the final predicted caspase substrates

TABLE 3.4: The caspase cleavage assay results of predicted potential caspase-3 substrates by PoPS, SitePrediction and Cascleave. "○" indicates the sequence is cleaved in the cleavage assay experiment while "×" indicates the sequence is not cleaved in the cleavage assay experiment.

Predicted caspase-3 substrate cleavage site	PoPS score	SitePrediction score	Cascleave score	Experimental result	Corresponding annotations in MEROPS
DVVD|GADT	21.32	1560.39	1.578	○	-
EEVD|GSSP	20.08	1515.922	1.461	○	-
EEVD|GSQG	20.08	1515.922	1.461	○	C14 homologue
DETD|SGAG	21.77	3400.88	1.345	○	C14.003: caspase-3, C14.005: caspase-6
EEVD|GAPR	20.08	1888.277	1.307	○	C14.005: caspase-6
DSVD|GSLT	21.26	1909.074	1.21	○	-
DDTD|GLTP	17.79	791.345	1.157	○	C14.005: caspase-6, C14.006: caspase-2
AEVD|GVDE	19.93	295.25	1.061	×	C14 homologue
DDPD|SAYL	18.08	680.822	1.058	○	-
SEVD|GNDS	20.05	449.294	1.039	○	C14.004: caspase-7, C14.006: caspase-2
AEVD|GATP	19.94	623.306	1.034	○	-
EEPD|GGFR	16.97	414.973	0.969	○	-
TEPD|SPSP	Non-cleavage	Non-cleavage	0.961	×	-
SEID|GLKG	18.7	220.873	0.911	○	-
EEPD|SANS	17.14	761.635	0.82	○	C14.005: caspase-6, C14.006: caspase-2, C14 homologue
NEVD|GSNE	20.01	223.501	0.766	○	-
EETD|GLDP	16.86	886.89	0.747	○	C14.001: caspase-1, C14.005: caspase-6, C14.006: caspase-2, C14 homologue
EETD|GLHE	16.86	886.89	0.747	○	-
GEVD|GKAI	19.85	271.729	0.691	○	-
TEMD|SETL	Non-cleavage	Non-cleavage	0.632	×	-
LESD|SESL	Non-cleavage	Non-cleavage	0.585	×	-

based on predicted cleavage probability the score of Cascleave, and then performed the caspase cleavage assay experiments to validate these potential caspase substrates.

Caspase cleavage assay

The caspase cleavage assay is conducted to experimentally validate the potential substrates selected out in the previous section. THe detailed information about CAspase cleavage assay could be seen in the Appendix.

Figure 3.8 illustrates the caspase substrate cleavage results based on western blotting. Recombinant proteins encoding GST-myc-GFP with IETD linker between GST and GFP (Right) or without linker (Left) were digested by active caspase-8 (+) or control (-). After digestion, proteins were analyzed by western blotting using anti-myc. Recombinant GST-mycGFP, 75 kDa band were detected in both conditions with and without caspase-8 treatment. In the recombinant GST-IETD-mycGFP protein case, a 75 kDa band was detected in caspase-8 non-treated condition, and in contrast a 50 kDa protein band was detected in caspase-8 treated recombinant GSTIETD-mycGFP, indicating that IETD linker was cleaved by caspase-8.

TABLE 3.5: The caspase cleavage assay results of predicted potential caspase-8 substrates by PoPS, SitePrediction and Cascleave. "○" indicates the sequence is cleaved in the cleavage assay experiment while "×" indicates the sequence is not cleaved in the cleavage assay experiment.

Predicted caspase-8 substrate cleavage site	PoPS score	SitePrediction score	Cascleave score	Experimental result	Corresponding annotations in MEROPS
DVVD|GADT	17.9	206.97	1.578	○	-
EEVD|GSSP	21.24	771.98	1.461	○	-
EEVD|GSQG	21.24	771.98	1.461	○	C14 homologue
DETD|SGAG	17.77	3194.444	1.345	○	C14.003: caspase-3, C14.005: caspase-6
EEVD|GAPR	21.24	1621.17	1.307	○	C14.005: caspase-6
DEVD|GAND	22.46	3371.648	1.261	○	-
DETD|SPTV	21.14	4921.875	1.236	○	C14.005: caspase-6, C14.006: caspase-2
DSVD|GSLT	17.59	525.68	1.21	○	C14 homologue
AEVD|GVDE	22.46	1010.936	1.061	○	-
SEVD|GNDS	Non-cleavage	Non-cleavage	1.039	○	C14.004: caspase-7, C14.006: caspase-2
AEVD|GATP	21.21	1268.26	1.034	○	-
TETD|SVGT	20.01	854.701	0.999	○	-
EEPD|GGFR	Non-cleavage	Non-cleavage	0.969	○	-
TEPD|SPSP	17.38	92.307	0.961	×	-
LEMD|SVLK	19.27	412.088	0.935	○	C14.005: caspase-6, C14.006: caspase-2, C14 homologue
EEPD|SANS	Non-cleavage	Non-cleavage	0.82	○	-
EETD|GLDP	22.17	559.69	0.747	○	C14.001: caspase-1, C14.005: caspase-6, C14.006: caspase-2, C14 homologue
EETD|GLHE	22.17	559.69	0.747	○	-
TEED|SVSV	18.61	275.71	0.714	○	-
TEMD|SETL	19.27	167.993	0.632	×	-
LESD|SESL	18.58	526.556	0.585	×	-

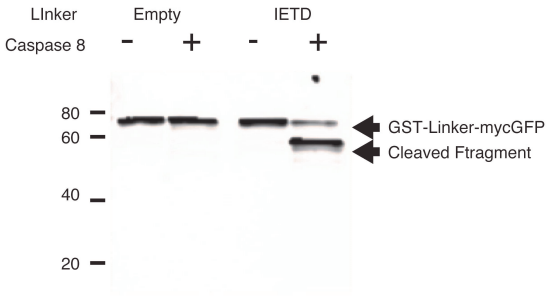


FIGURE 3.8: Western blotting of caspase assay analysis. Recombinant GST-mycGFP, 75 kDa band was detected in both conditions with and without caspase-8 protein treatment. In the recombinant GST-IETD-mycGFP protein case, a 75 kDa band was detected in caspase-8 non-treated condition, while in contrast a 50 kDa protein band was detected in caspase-8 treated recombinant GSTIETD-mycGFP, indicating that the IETD linker was cleaved by caspase-8.

3.3.3 Caspase assay result discussion

Caspase cleavage assay results are summarized in Tables 3.4 and 3.5. As we can see, the experimental results clearly show that the majority of predicted potential substrates were cleaved. These results indicate that the tools tested in the evaluation step demonstrate an excellent performance for predicting both caspase-3 and 8 target substrates. In addition there also exist a number of cleavage sites which can not be identified by most of the tools, highlighting that it remains a challenging task

to develop more reliable and accurate caspase cleavage site prediction methods. See Supplemental Information for details of protein expression.

From the result we can also see that some predicted substrates are both cleaved by caspases in the group II and group III, such as DETD|SGAG (both cleaved by caspase-3 and caspase-6) and DDTD|GLTP (both cleaved by caspase-2 and caspase-6). This is the reason why many substrates cleaved by caspase-3 are also cleaved by caspase-8. The result also indicates that there is a tendency that PoPS, SitePrediction and Cascleave prefer to predict substrates cleaved by caspases from the groups II and III rather than group I.

3.4 Discussion

Due to the functional significance of caspase substrate identification problem, computational biologists are motivated to develop more accurate and reliable predictors for caspase substrate prediction. Aiming at providing a comprehensive review of the status quo of caspase substrate predictors to non-bioinformaticians, this chapter describes and compares a number of widely used caspase substrate predictors in terms of their input/output, model construction and development, model performance evaluation as well as predictor utility. Benchmarking analysis on the independent test datasets revealed that Cascleave and SitePrediction achieve the overall highest AUC value when used for predicting caspase substrates in different species, especially for *Homo sapiens*. In particular, SitePrediction achieved the highest AUC value when used for predicting caspase substrates for species other than *Homo sapiens*. Detailed case studies of 21 caspase-3 substrate sequences and 21 caspase-8 substrate sequences demonstrate that while Cascleave, PoPS and SitePrediction achieved acceptable performance, there still exist some sequences that most currently available tools failed to predict. We conclude that caspase substrate prediction remains a challenging task and we expect that more powerful next-generation algorithms with improved prediction performance will emerge with the increasing availability of caspase substrate cleavage data that can be used as high-quality training data for constructing the prediction models.

As a potential way to improve the prediction performance, we need to integrate information from various aspects, including but not limited to secondary structure information, position information of amino acids out of caspase substrate region. Besides we can introduce next generation machine learning approaches such as deep learning.

Chapter 4

Analysis of critical and redundant vertices in controlling directed complex networks using feedback vertex sets

4.1 Introduction

It have been shown that the controllability for large scale networks remains be a challenging topic for various fields including the Internet and WiFi communication in telecommunications engineering, metabolite networks in Biology and economic science. However, in real world large scale networks are always an effective way to model complex systems, thus it is imperative to construct novel algorithms/tools to solve the problem for controlling large scale networks (Albert and Barabási, 2002), especially on networks of scale free structure.

There are a large number of algorithms/tools developed to solve solve the problem for controlling large scale networks including master equations (Prigogine and Nicolis, 1977), Monte-Carlo methods (Gillespie, 1977), stochastic model (Cai and Yuan, 2009), ordinary differential equations (ODE) (Goodwin *et al.*, 1963, Novak *et al.*, 2001, Tyson *et al.*, 2001), boolean networks (Akutsu *et al.*, 2012, Melkman and Akutsu, 2013) as well as other mathematical models. Most of these algorithms/tools share a common feature that these algorithms induce the simulated network of real system to a given state in finite time by changing the status of a subset of this network. This is done by selecting a number of vertices in this network as the so called control vertices or driver vertices. Liu *et al.* (2011) have shown that determining structural controllability of a network with linear dynamics is equivalent to computing the maximum matching of the corresponding bipartite graph. Such kind of controlling frameworks have been

utilized in many fields (Cowan *et al.*, 2012, Wang *et al.*, 2012) and various types of networks (Nacher and Akutsu, 2012). These researches have shown that while a random network need only a small number to be controlled, a scale free network needs a relative large number of vertices to control it (more than 80% with the parameter of $gamma = 2$).

Nacher and Akutsu (2014) presented a novel research to undertake this problem by introducing the MDS (minimum dominating set) concept, they also develop the MDS based framework to handle the control problem in complex networks. The dominating set is defined as followings: In a graph G, a dominating set (DS) is a set of vertices S with the property that every vertex in G is either an element of S or belongs to the adjacent list of a vertex in S. But utilizing this approach, they have successfully reduce the amount of control vertices to less than 20%. Many other researches utilized this framework including PPI networks (Ishitsuka *et al.*, 2016, Nacher and Akutsu, 2016) as well as identifying relationships between human diseases and non-coding RNAs (Kagami *et al.*, 2015).

Mochizuki *et al.* (2013) proposed another approach based on the feedback vertex set (FVS). In a graph G, an FVS is a set of vertices S with the property that removal of all the vertices in S results in G becoming acyclic, and an FVS with the minimum number of elements is known as a minimum feedback vertex set (MFVS). They showed that any FVS can be a set of control vertices if the target states are restricted to steady states and periodically steady states (i.e., attractors) for a wide class of networks, where it was previously known that a close relationship between an FVS and steady states in Boolean networks exist (Tatsuya, 2018). Since the target states are restricted in the MFVS-based approach, it is expected in many cases that much smaller sets of driver vertices are required than in the MM and MDS-based approaches. Although the MFVS-based approach by Mochizuki *et al.* (2013) seems to be useful, the approach has two drawbacks: no efficient algorithm for selecting the MFVS is shown there, and an MFVS is not necessarily uniquely determined as in the MM and MDS-based cases.

In this research we develop another framework that mapping the structural controllability problem on a directed network via MFVS (minimum feedback vertex set) approach instead of MDS (minimum dominating set) approach. For MDS based framework, the calculation algorithm is defined as followings: In a graph G, an FVS denotes a set of vertices S with the property that removing all the vertices in S makes G acyclic, while an MFVS indicates the smallest FVS for a given graph G.

For MDS based framework, the calculation algorithm does not generate a unique configuration of the driver vertices, instead, they divide the driver vertices into three

categories, critical driver vertices (or critical vertices), intermittent driver vertices (or intermittent vertices) and redundant driver vertices (or redundant vertices), these three categories of driver vertices are defined as followings: For a graph G several configurations with the same number of driver vertices that could control the whole networks could be generated, the critical vertices is defined as vertices that appear in all these configurations, intermittent vertices is defined as vertices appear in some of these configurations and redundant vertices is defined as vertices that do not appear in any of these configurations.

Before continue to discuss our algorithm, we must first notice a fact that as far as we know that MFVS problem belongs to the class of NP-complete problems, thus we can not find an efficient algorithm with a polynomial time complexity, however this does not mean that we can not find an algorithm that runs with an acceptable time consuming. besides with some heuristic algorithm named graph contraction we could reduce the size of the original problem by identifying that some of the vertices in G must be/can't be in any MFVSs and directly removing these vertices from G to decrease the graph size (Levy and Low, 1988, Lin and Jou, 2000, Smith and Walford, 1975, Wang *et al.*, 1985), which means that we can speed up our program if we apply graph contraction as pre-processing procedure to our MFVS calculation step. Among the graph contraction approaches Levy and Low (1988) presented a canonical graph contraction procedure consisting steps named IN0/OUT0/IN1/OUT1/LOOP, and this approach has been successfully applied to various areas such as determining Scan Flip-Flop (Ashar and Malik, 1994, Cheng and Agrawal, 1990) and building energy system simulation (Sowell and Haves, 2001).

We would also like to utilize this approach to speed up our algorithm. However since our goal is to calculate critical/redundant MFVS driver vertices, we would like to make some modifications to this contraction algorithm to suit our situation.

In this work, we firstly present our modified contraction algorithm which is suitable for critical/redundant vertices calculation via MFVS approach. Then we present our main algorithm for calculating critical/redundant vertices. In addition, we performed some computational simulations using random directed networks of Erdős-Rényi structure as well as random directed networks of scale free structure. Finally we applied our critical/redundant vertices calculation algorithm to real-world biological networks.

4.2 Methods

In this section we introduce 3 steps of our algorithm: our pre-processing procedure, our main algorithm for the calculation of critical/redundant vertices via an MFVS calculating algorithm and finally a way to determine the critical/redundant/intermittent status of some vertices that are ignored in the pre-processing procedure. First, we would like to give the definition of the MFVS as well as the definitions of CMFVS/RMFVS/IMFVS.

Definition 1. *Minimum Feedback Vertex Set (MFVS): Given a directed graph $G = (V, E)$, an MFVS of G is a minimum subset of V containing at least one vertex from every directed cycle in G.*

We denote an instance of a set of MFVS for G as $S_{MFVS}(G)$ or S_{MFVS}. Since there are multiple MFVS configurations for one graph, we denote the set of all MFVS configurations in G as S_{AM}, which means $S_{AM} = \{S^1_{MFVS}, S^2_{MFVS}, ..., S^M_{MFVS}\}$, where M is the number of all possible MFVSs in G. According to the definition of MFVS, because the size of all $S_{MFVS} \in S_{AM}$ are equal, we define the size of S_{MFVS} as $|S_{MFVS}|$.

Definition 2. *Critical/Intermittent/Redundant Minimum Feedback Vertex Set (CMFVS / IMFVS / RMFVS): Given a directed graph $G = (V, E)$, a critical MFVS (S_{CMFVS}) is defined by $\{v \in V \mid \forall\, S_{MFVS} \in S_{AM}\,(v \in S_{MFVS})\}$, a redundant MFVS ($S_{RMFVS}$) is defined by $\{v \in V \mid \forall\, S_{MFVS} \in S_{AM}\,(v \notin S_{MFVS})\}$, and an intermittent MFVS (S_{IMFVS}) is defined as $V - S_{CMFVS} - S_{RMFVS}$.*

Here we should notice that although there may be multiple MFVS configurations for a graph G, there is only one CMFVS/RMFVS/IMFVS configuration for each graph.

Figure 4.1 gives an example of critical vertices (red rounded vertices), redundant vertices (purple rounded vertices), intermittent vertices (blue rounded vertices). In this graph there are two MFVSs, {*vertex* 2, *vertex* 5}, {*vertex* 2, *vertex* 6}. Since in this graph vertex 2 belongs to all MFVSs, vertex 2 is confirmed as a critical vertex. Since vertices 5, 6 are in some but not all MFVSs, vertices 5, 6 are confirmed as intermittent vertices. Since vertices 1, 3, 4 are not in any MFVS, vertices 1, 3, 4 are confirmed as redundant vertices. Thus, in this graph $|S_{CMFVS}| = 1$, $|S_{RMFVS}| = 3$ and $|S_{IMFVS}| = 2$, $S_{AM} = \{\{vertex\ 2, vertex\ 5\}, \{vertex\ 2, vertex\ 6\}\}$, and $M = 2$.

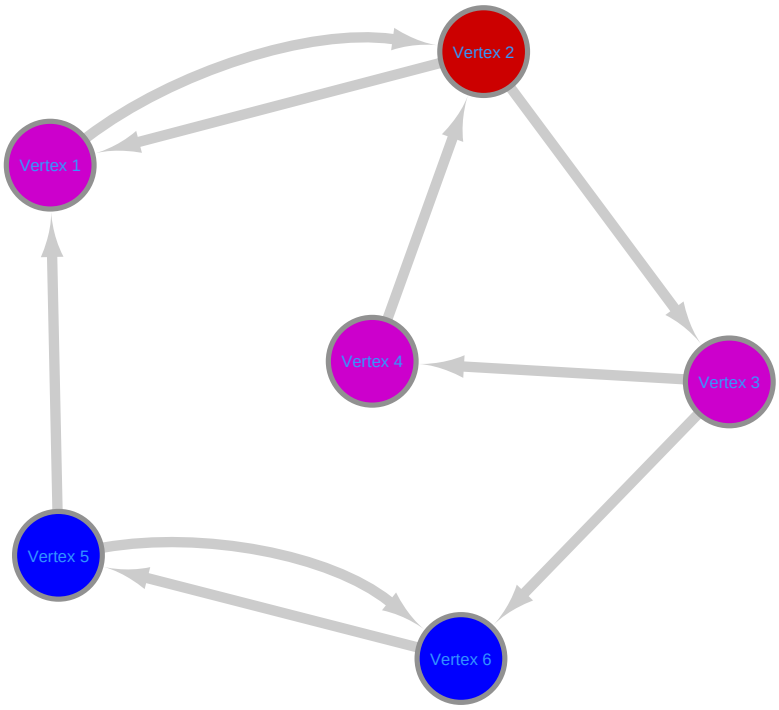


FIGURE 4.1: Illustrative example of critical vertices (red rounded vertices), redundant vertices (purple rounded vertices), and intermittent vertices (blue rounded vertices). In this graph there are two MFVSs, $\{vertex\ 2,\ vertex\ 5\}$, $\{vertex\ 2,\ vertex\ 6\}$. Since in this graph vertex 2 belongs to all MFVSs, vertex 2 is confirmed as a critical vertex. Since vertices 5, 6 are in some but not all MFVSs, vertices 5, 6 are confirmed as intermittent vertices. Since vertices 1, 3, 4 are not in any MFVS, vertices 1, 3, 4 are confirmed as redundant vertices. Thus, in this graph $|S_{CMFVS}| = 1$, $|S_{RMFVS}| = 3$ and $|S_{IMFVS}| = 2$, $S_{AM} = \{\{vertex\ 2, vertex\ 5\}, \{vertex\ 2, vertex\ 6\}\}$, and $M = 2$.

Our algorithm contains three parts: the pre-processing procedure (modified graph contraction procedure) of the critical/redundant MFVS calculation, the main algorithm of the critical/redundant MFVS calculation, and determination of the critical/redundant status of the remaining vertices of unknown status. The pre-processing section gives a description of the modification to the graph contraction procedure developed by Levy and Low (1988). to suit our critical/redundant MFVS calculation, and the pseudocode of our modified preprocessing algorithm is also provided as $IN0\_OUT0\_Modified$, $LOOP\_Modified$, $IN1\_Modified$ and $OUT1\_Modified$ in Algorithm 1 and 2. In the second part of this section, we give a description about our main algorithm to calculate critical/redundant vertices to generate the contracted graph after considering of the output of the pre-processing procedure as input. The key point for the pre-processing step is that the pre-processing step selects some vertices of which the critical/redundant status are determined via other vertices, thus for these vertices the main step is skipped to accelerate the computation. The last part of this section contains a description of the determination of the status of the

remaining vertices and we present ways to process these vertices.

4.2.1 Pre-processing for critical/redundant vertex calculation

Graph contraction algorithm for MFVS calculation

Studies on graph contraction for calculating MFVS have already been discussed in some previous researches, and we start this section by briefly introduce the most widely known graph contraction steps described in Levy and Low (1988):

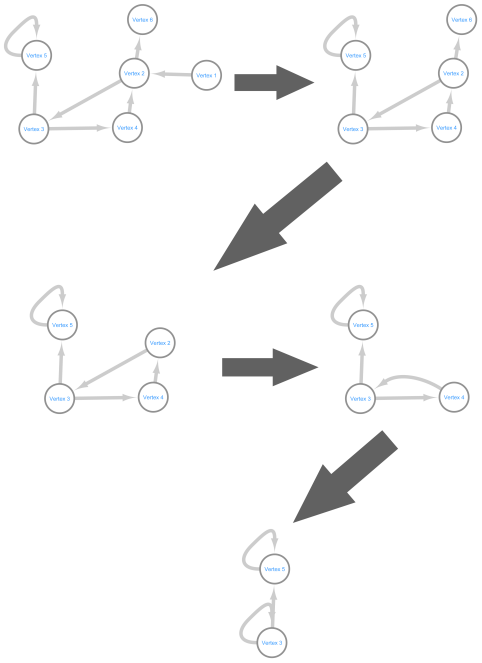


FIGURE 4.2: Example of a well-known graph contraction procedure. In the original graph (top left) vertex 1 is contracted by the IN0 procedure. Next, vertex 6 is contracted by the OUT0 procedure (top right), after which vertices 2, 4 are contracted by the IN1/OUT1 procedure (middle two graphs). Finally, vertices 3, 5 are contracted by a LOOP procedure (bottom).

- **IN0 (v)**: For a vertex v with indegree $(v) = 0$, G is contracted by removing v and all the out edges from v, where indegree(v) signifies the number of inward directed graph edges from v in a directed graph.

- **OUT0 (v)**: For a vertex v with outdegree $(v) = 0$, G is contracted by removing v and all the in edges to v, where outdegree(v) means the number of outward directed graph edges from v in a directed graph.

- **IN1 (v)**: For a vertex v if indegree $(v) = 1$ and u $(u \neq v)$ is the predecessor of v, G is contracted by merging v into u as follows: For every edge e from v to v', remove e and add an edge u to v'. Remove v and the edge u to v. Remove all parallel edges created by this transformation.

- **OUT1 (v)**: For a vertex v if outdegree $(v) = 1$ and u $(u \neq v)$ is the successor of v, G is contracted by merging v into u as follows: For every edge e from v' to v, remove e and add an edge v' to u. Remove v and the edge v to u. Remove all parallel edges created by this transformation.

- **LOOP (v)**: if $v - v$ is a self-loop edge in G, G is contracted by removing v and all the edges incident to v.

In Figure 4.2, we illustrate the algorithm of the graph contraction procedure developed by Levy et al by a figure example. In the original graph (left top graph) vertex 1 is contracted by IN0 procedure. After vertex 1 is contracted, vertex 6 is contracted by OUT0 procedure (right top graph). After vertices 1, 6 are contracted, vertices 2, 4 are contracted by IN1/OUT1 procedure (middle two graphs). Finally, vertices 3, 5 are contracted by LOOP procedure (bottom graph).

The graph contraction algorithm illustrated in Figure 4.2 is able to let us reduce the size of the graph without affecting the MFVS status.

As the graph contraction algorithm will be used for the critical/redundant MFVS calculation, we need to alter this algorithm to accommodate our situation.

IN0/OUT0/LOOP for critical/redundant MFVS calculation

In this section our modification to IN0/OUT0/LOOP is introduced. some theorems/corollaries and the pseudocode of the modified algorithm is shown as $IN0\_OUT0\_Modified$ and $LOOP\_Modified$ in Algorithm 1.

Theorem 1. *All the vertices contracted by the IN0/OUT0 procedure are redundant vertices, and these vertices are removed from G directly without affecting the critical/redundant status of other vertices.*

Proof. Given a directed graph $G = (V, E)$, let v be a vertex in G, if a vertex v is contracted by the IN0/OUT0 procedure, v has neither an out-degree nor an in-degree,

which means v is not included in any loops of G; thus, v is not in any MFVS, i.e., v is a redundant vertex. □

In addition, Corollary 1 is provided as a generalization of Theorem 1:

Corollary 1. *Given a directed graph $G = (V, E)$, let v be a vertex in G, if no loop in G passes v, v is a redundant vertex. Hence, v is removed from G directly without affecting the critical/redundant status of other vertices.*

Theorem 2. *If vertex v is a vertex in the original G (not contracted by other procedures) and is contracted by the LOOP procedure, v is a critical vertex, and v is removed from G directly without affecting the critical/redundant status of other vertices.*

Proof. If a vertex v is contracted by the LOOP procedure, self-loop(s) containing v is necessarily detected, which is exist in G before graph contraction. For any MFVS, vertex v must be selected into this MFVS to break the cycle $v - v$, which means that v is included in every possible MFVS; thus, v is a critical vertex. □

The pseudocode of the modified pre-processing algorithms for the IN0/OUT0/LOOP procedure of the critical/redundant MFVS calculation is shown in Algorithm 1.

IN1/OUT1 for critical/redundant MFVS calculation

In this section we continue introducing our modification to IN1/OUT1. As in the previous section we firstly give some theorems/corollaries before we provide a detailed introduction to our modified algorithm for IN1/OUT1. The pseudocode is provided as $IN1\_Modified$ as well as $OUT1\_Modified$ in Algorithm 2.

Definition 3. *Given a directed graph $G = (V, E)$, let v be a vertex in G; if there is a vertex u adjacent to v and all loops passing v also pass u, we say that u covers v.*

Theorem 3. *If u is a vertex that covers v, v is not a critical vertex.*

Proof. First, the proof that u and v cannot appear in the same instance of S_{MFVS} is provided. Suppose u and v are found in the same instance of S_{MFVS}, since all the loops passing v also pass u, v can be removed from this S_{MFVS} to obtain a new S'_{MFVS}. Because breaking all cycles passing v is done by selecting u into S_{MFVS},

Algorithm 1 Modified IN0/OUT0/LOOP for critical/redundant MFVS calculation

INPUT: A graph G=(V, E)
OUTPUT: A contracted graph G'=(V', E')

```
procedure IN0_OUT0_Modified
    while True do
        removed_vertex ← 0
        for vertex v_i ∈ V do
            if indegree(v_i) = 0 ∨ outdegree(v_i) = 0 then
                add v_i into S_RMFVS
                remove v_i from G
                removed_vertex ← removed_vertex + 1
        if removed_vertex = 0 then
            Break
procedure LOOP_Modified
    while True do
        removed_vertex ← 0
        for vertex v_i ∈ V do
            if v_i has a self-loop then
                add v_i into S_CMFVS
                remove v_i from G
                removed_vertex ← removed_vertex + 1
        if removed_vertex = 0 then
            Break
```

there is an S'_{MFVS} with a smaller size than S_{MFVS}, which means S_{MFVS} is not an instance of the MFVS of G. Thus u and v is not in the same instance of S_{MFVS}.

Assume that v is a critical vertex. Then, for any S_{MFVS}, there is $v \in S_{MFVS}$. Since u is a vertex covering v, which means that all loops passing v in G also pass u, v is replaced by u of this S_{MFVS} to acquire a new S_{MFVS} that does not contain v; thus, v is not in every possible MFVS, indicating that it is not a critical vertex. □

Before introducing the next Theorem, we will give the definition of a Chain Map (CHM).

Definition 4. *Chain Map (CHM): Given a directed graph $G = (V, E)$, if the critical/redundant property of a vertex v_1 is determined by another vertex v_2, vertex pair (v_2, v_1) belongs to a chain map. Further, v_2 and v_1 are both chain vertices. An instance of CHM of G is denoted as M_{CHM}.*

Theorem 4 and Corollary 2, 3 are given as follows:

Theorem 4. *Let u be a vertex that covers v. If v is also a vertex that covers u, u and v share the same critical/redundant status and neither u nor v are critical vertices, u and v are equal.*

Proof. We can prove that u and v are not in the same instance of S_{MFVS} as in the previous proof.

According to Theorem 3, neither u nor v are critical vertices.

Suppose v is a redundant vertex and u is assumed to be not a redundant vertex, then there are some MFVSs that contain u. Notice that since all the loops passing u also pass v, a new MFVS is created by replacing u with v, which means that there is an MFVS that contains v. Since we already know that v is not in any MFVS, u is a redundant vertex.

Suppose v is an intermittent vertex, i.e., v is contained by some MFVSs of G. Thus all loops passing v also pass u, then a new MFVS is obtained by replacing v with u, which implies that u is also in some of the MFVSs. Since u is not a critical vertex, u is an intermittent vertex. □

Corollary 2. *If u is a vertex that equals v, either pair (u, v) or pair (v, u) is added to M_{CHM}, and either u into v or v is merged into u.*

Corollary 3. *For each pair (u, v) in M_{CHM}, if u is an intermittent vertex, v is also an intermittent vertex, and if u is a redundant vertex, v is also a redundant vertex.*

Then, we give the introduction of our modified algorithm for IN1/OUT1 critical/redundant MFVS calculation.

For the OUT1 critical/redundant MFVS calculation, v is the vertex with out-degree 1 and u is the successor of v. Then two states are given for u depending on whether u is also a predecessor of v.

- 1: u is a predecessor of v.
- 2: u is not a predecessor of v.

For case 1, in which vertex u is a predecessor of v, three possible cases are given depending on $outdegree(u)$ and $indegree(u)$ (Figure 4.3):

- 1-1: If $outdegree(u) = 1$, subgraph $\{u, v\}$ is removed from G without affecting the MFVS status of other parts of the graph. Since either u or v is selected into an MFVS to break the cycle between u and v, both u and v are intermittent vertices.
- 1-2: If $indegree(u) = 1$, u covers v. According to Theorem 3 neither u nor v is critical vertex. Since either u or v is selected into an MFVS to break the cycle between u and v, both u and v are intermittent vertices. After breaking the cycle between u and v, both u and v become vertices without in-degree or out-degree, thus are removed from G.

- 1-3: If $outdegree(u) \neq 1 \wedge indegree(u) \neq 1$, this case is solved by the following process.

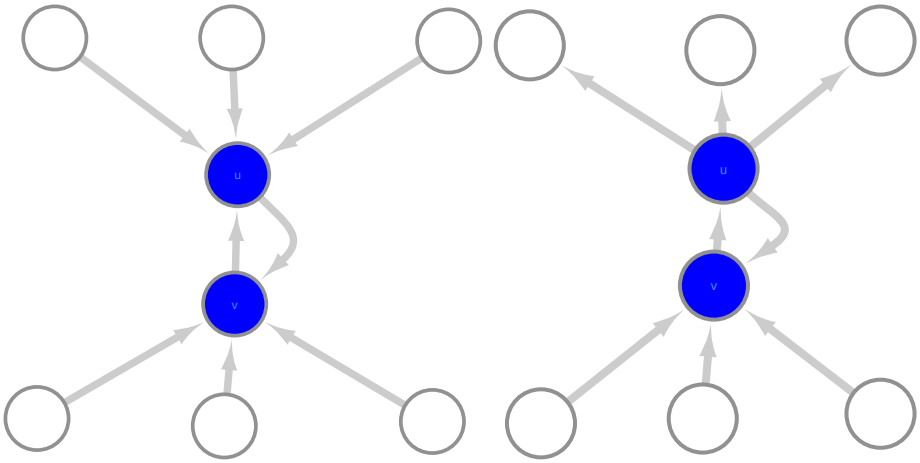


FIGURE 4.3: Illustration for cases 1-1 and 1-2. In these two cases both u and v are intermittent vertices (blue rounded vertices), and u and v are removed from the graph.

Before discussing the solution to case 1-3, firstly we provide the definition of non-critical MFVS as well as a self-connection (SC) (Figure 4.4).

Definition 5. *Non-critical feedback vertex set (NCMFVS): Given a directed graph $G = (V, E)$, a non-critical MFVS is defined as $\{v \in V | v \notin S_{CMFVS}\}$. An instance of a non-critical MFVS is defined as S_{NCMFVS}.*

Definition 6. *The self-connection of u except v (SC(u, v)): Given a directed graph $G = (V, E)$, where v and u are vertices in G, we firstly define temporarily removing v from the graph as temporarily eliminating all the out-pointed/in-pointed edges of v, and we define u to be self-connected except v if there is at least a path in G from u to u after we temporarily remove v from G.*

Then the case 1-3 is divided into the following 2 cases according to the condition whether $SC(u, v) = True$ (Figure 4.5):

- 1-3-1: If $SC(u, v) = True$, u covers v. According to Theorem 3, vertex v is not a critical vertex and add v into S_{NCMFVS}.
- 1-3-2: If $SC(u, v) = False$, u and v are equal. According to Theorem 4, u and v share the same critical/redundant status. Since the cycle between u and v needs to be broken, either u or v is selected into MFVS, thus both u and v are intermittent vertices, and are removed from the graph.

For case 2 where vertex u is not a predecessor of v, there are also three cases according to $outdegree(u)$ and $indegree(u)$ (Figure 4.6):

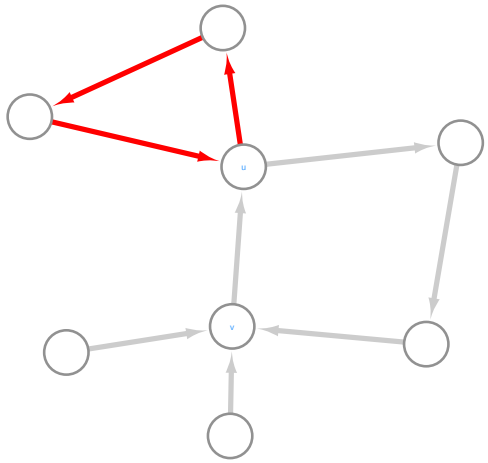


FIGURE 4.4: Illustration of self-connection of u except v (SC(u, v)). The SC(u, v) is defined as red path from u to u even if we remove v and all the edges connecting v from the graph.

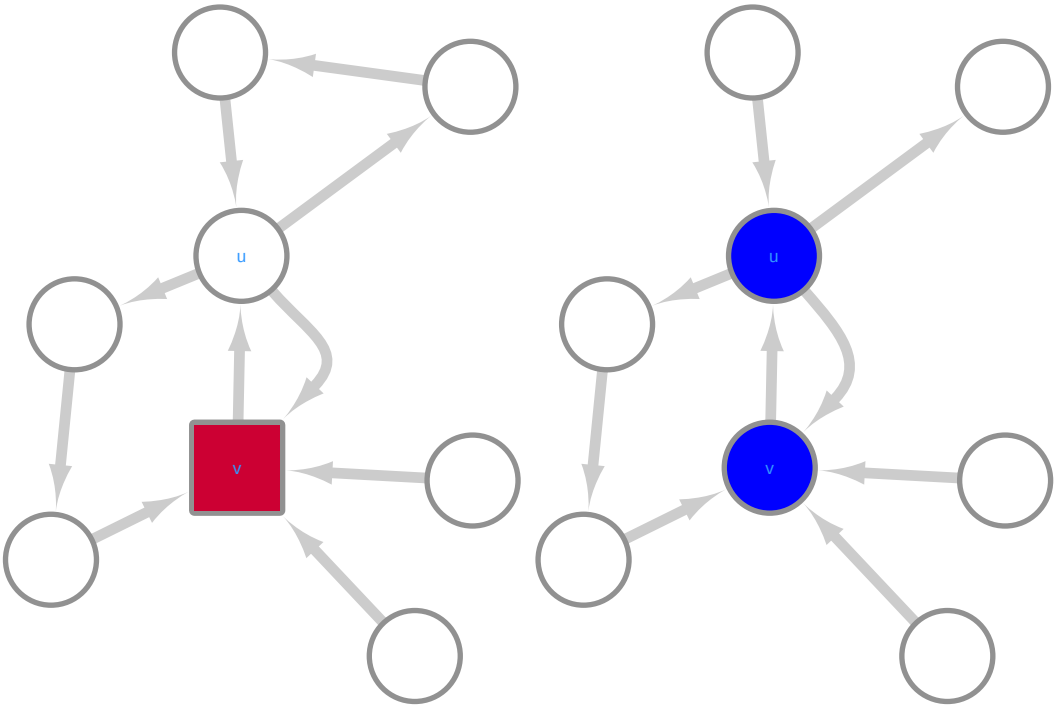


FIGURE 4.5: Illustration of case 1-3-1 and case 1-3-2. In the case on the left (case 1-3-1), v is covered by u and is represented by a red rectangular vertex, which means that v is not a critical vertex. In the case on the right (case 1-3-2), both u and v are marked as intermittent vertices and are removed from the graph (blue rounded vertex)

- 2-1: if $outdegree(u) = 0$, u and v are not in any loops of this graph. According to Corollary 1, both u and v are redundant vertices, and is added into S_{RMFVS} and removed from the graph.

- 2-2: if $indegree(u) = 1 \wedge outdegree(u) \neq 0$, u and v are equal. According to Theorem 4 and Corollary 2, neither u nor v are critical vertices and either

(u, v) or (v, u) belongs to M_{CHM}. Thus (u, v) is added into M_{CHM}, add u into S_{NCMFVS}, and v is merged into u.

- 2-3: if $outdegree(u) \neq 0 \wedge indegree(u) \neq 1$, the solution is provided as followings.

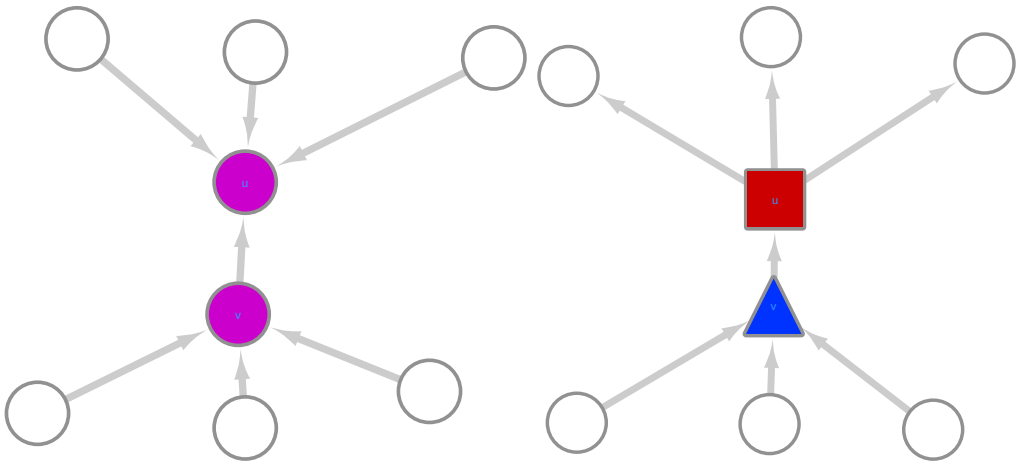


FIGURE 4.6: Illustration of case 2-1 (on the left) and 2-2 (on the right). In the first case both u and v are redundant vertices (purple rounded vertices) and removed from the graph. In the second case v (blue triangular vertex) is chained by u (red rectangular vertex).

Before discussing case 2-3 we will give the definition of a loop connection (LC) (Figure 4.7):

Definition 7. *Loop connection of u and v ($LC(u, v)$): Given a directed graph $G = (V, E)$, in which u and v are two adjacent vertices in G, the loop connection of u and v is defined as that there is at least a path from u to v after temporarily remove edges connecting v and u, i.e., an edge starts at v and ends at u and an edge starts at u and ends at v.*

Then case 2-3 can also be further divided into four cases according to whether $SC(u, v)/LC(u, v)$ are True (Figure 4.8):

- 2-3-1: if $SC(u, v) = True \wedge LC(u, v) = True$, u covers v. According to Theorem 3, v is not a critical vertex, thus v is added into S_{NCMFVS}.
- 2-3-2: if $SC(u, v) = True \wedge LC(u, v) = False$, v is not in any of the loops of this graph. According to Corollary 1, v is a redundant vertex, thus v is added into S_{RMFVS} and removed from the graph.
- 2-3-3: if $SC(u, v) = False \wedge LC(u, v) = True$, u and v are equal. According to Corollary 2, neither u nor v is critical vertices and u, v share the same

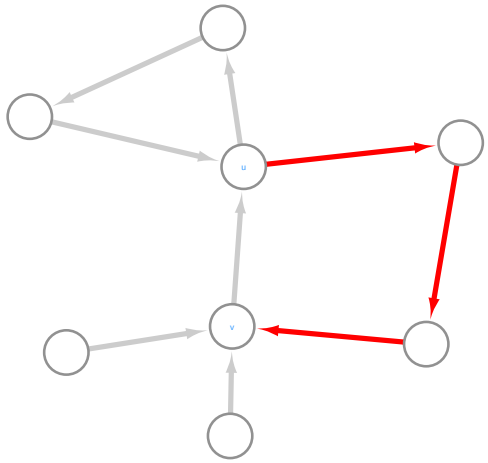


FIGURE 4.7: Illustration of a loop connection between u and v (LC(u, v)). The red path in LC(u, v) shows the path from u to v even if the edge between v and u. is removed

critical/redundant status, thus (u, v) is added into M_{CHM}, u is added into S_{NCMFVS} and v is merged into u.

- 2-3-4: if $SC(u, v) = False \wedge LC(u, v) = False$, neither u nor v is in any of the loops of this graph. According to Corollary 1, both u and v are redundant vertices and both u and v are added into S_{RMFVS} and removed from the graph.

The $IN1\_Modified$ procedure is generated symmetrically to $OUT1\_Modified$.

The pseudocode of the modified IN1/OUT1 is shown in Algorithm 2.

Then the modified IN0, OUT0, IN1, OUT1, LOOP is iteratively utilized to contract the graph until no vertex can be contracted.

4.2.2 Critical/Redundant MFVS calculation algorithm

After graph contraction, let all the vertices remaining in the contracted graph be $S_{contracted}$. the vertices with unknown status (i.e., $S_{unknown} = S_{contracted} - S_{NCMFVS}$ in the critical vertex calculation step, and $S_{unknown} = S_{contracted}$ in the redundant vertex calculation step) are selected in the contracted graph and let them be $S_{unknown}$. Then the critical/redundant MFVS calculation is processed by the following steps:

1. Calculate the MFVS of the contracted graph and let it be S_{MFVS}.
2. Let S_{CMFVS} be a set consisting of vertices confirmed as critical vertices in the previous graph contraction procedure.
3. Repeat steps 4-6 for all v for which $v \in S_{unknown} \cap S_{MFVS}$.
4. Create an ILP instance I_v by adding a constraint of $v_i \leq 0$ to the instance given in the following section.

Algorithm 2 Modified IN1/OUT1 for critical/redundant MFVS calculation

INPUT: A graph G=(V, E)
OUTPUT: A contracted graph G'=(V', E')

procedure $IN1\_Modified$
symmetrical operation to $OUT1\_Modified$ for critical/redundant MFVS calculation

procedure $OUT1\_Modified$
while $True$ **do**
$removed\_vertex \leftarrow 0$
for vertex $v_i \in V$ **do**
$adjacent_list \leftarrow$the out-pointed vertices of v_i
if $|adjacent\_list|$=1 and this vertex $\neq v_i$ **then**
$v_{target} \leftarrow$ successor of v_i
$adjacent\_list\_target\_in \leftarrow$ the in pointed vertices of v_{target}
$adjacent\_list\_target\_out \leftarrow$ the out pointed vertices of v_{target}
if $v_i \in adjacent\_list\_target\_out$ **then**
if $|adjacent\_list\_target\_in| = 1 \vee |adjacent\_list\_target\_out| = 1$ **then**
remove v_i and v_{target} from the graph
$removed\_vertex \leftarrow removed\_vertex + 2$
else
if $SC(v_{target}, v_i)$ **then**
add v_i into S_{NCMFVS}
else
remove v_i and v_{target} from the graph
$removed\_vertex \leftarrow removed\_vertex + 2$
else
if $|adjacent\_list\_target\_out| = 0$ **then**
add v_{target} and v_i into S_{RMFVS}
remove these two vertices from the graph
$removed\_vertex \leftarrow removed\_vertex + 2$
else if $|adjacent\_list\_target\_in| = 1$ **then**
add (v_{target}, v_i) into M_{CHM}
merge v_i into v_{target}
$removed\_vertex \leftarrow removed\_vertex + 1$
else
if $SC(v_{target}, v_i)$ **then**
if $LC(v_{target}, v_i)$ **then**
add v_i into S_{NCMFVS}
else
add v_i into S_{RMFVS}
remove v_i from the graph
$removed\_vertex \leftarrow removed\_vertex + 1$
else
if $LC(v_{target}, v_i)$ **then**
add (v_{target}, v_i) into M_{CHM}
merge v_i into v_{target}
$removed\_vertex \leftarrow removed\_vertex + 1$
else
add v_{target} and v_i into S_{RMFVS}
remove these two vertices from the graph
$removed\_vertex \leftarrow removed\_vertex + 2$
if $removed\_vertex = 0$ **then**
Break

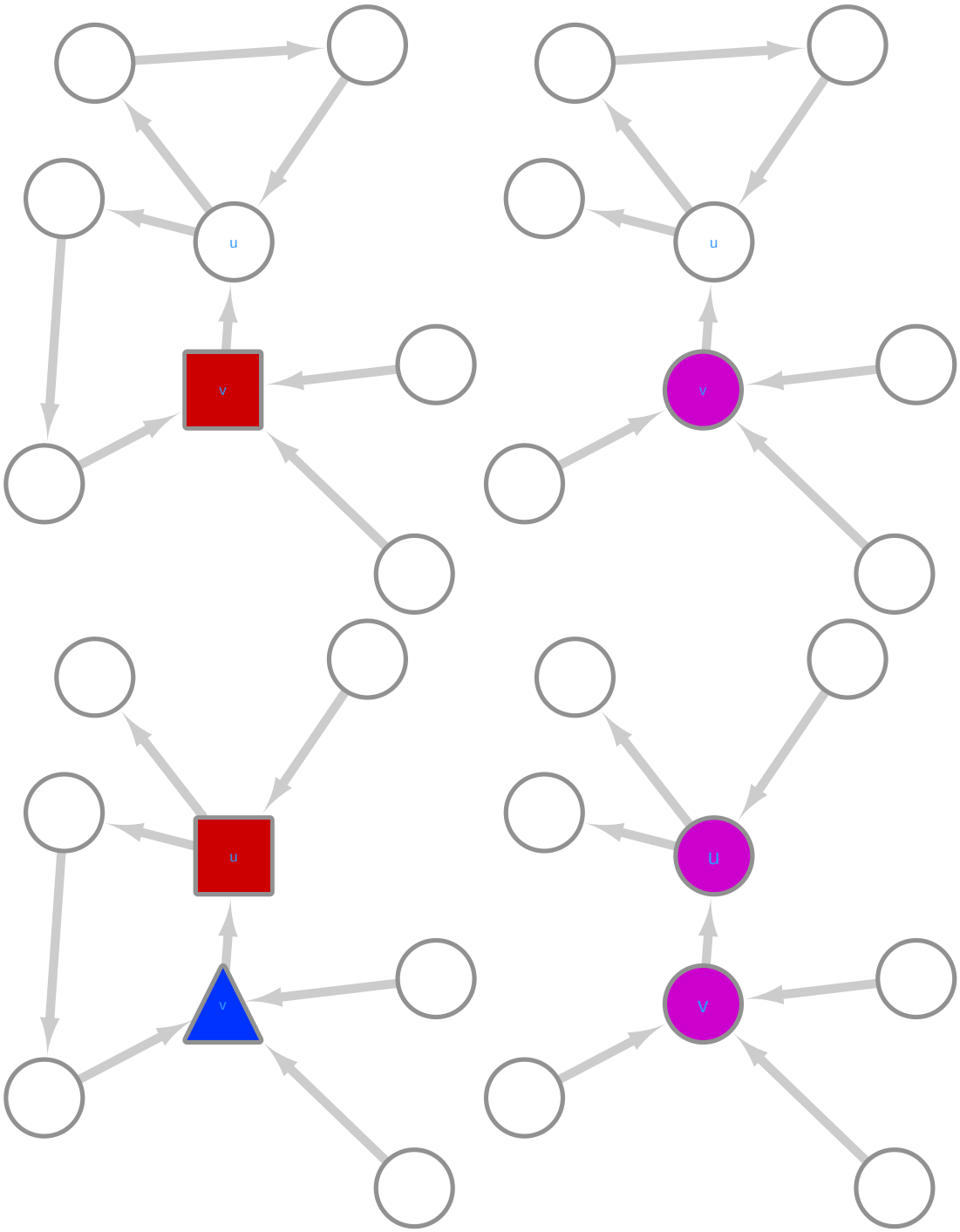


FIGURE 4.8: Description of case 2-3-1, 2-3-2, 2-3-3 and 2-3-4. In case 2-3-1 (top left), v (red rectangular vertex) is covered by u, which means that v is a non-critical vertex. In case 2-3-3 (bottom left), v (blue triangular vertex) and u belongs to chain map and v merged into u, and u is a non-critical vertex (red rectangular vertex). In case 2-3-2 (top right), since v (purple rounded vertex) is not in any loops of G, v is a redundant vertex and removed from the G. In case 2-3-4 (bottom right), since neither u nor v (purple rounded vertices) are in any loops of G, they are marked as redundant vertices and removed from G.

5. Obtain the solution of I_v and let $M_v = \{v_i | v_i = 1\}$.
6. If $I_v = \perp$ or $|M_v| > |S_{MFVS}|$, add v into S_{CMFVS}.
7. Return S_{CMFVS}.

In addition, we perform the redundant calculation via the following steps:

1. Calculate the MFVS of the contracted graph and let it be S_{MFVS}.
2. Let S_{RMFVS} be a set consisting of vertices confirmed as redundant vertices in the previous graph contraction procedure.
3. Repeat steps 4-6 for all v for which $v \in S_{unknown} \cap (S_{contracted} - S_{MFVS})$.
4. Create an ILP instance I_v by adding a constraint of $v_i \geq 1$ to the instance given in the following section.
5. Obtain the solution of I_v and let $M_v = \{v_i | v_i = 1\}$.
6. If $I_v = \perp$ or $|M_v| > |S_{MFVS}|$, add v into S_{RMFVS}.
7. Return S_{RMFVS}.

ILP formalization for calculating MFVS

Since the critical/redundant driver vertices calculation algorithm described above utilizes the MFVS calculation algorithm, we will further introduce the way to calculate an MFV set for a graph.

Given a directed graph $G = (V, E)$, we calculate the MFVS of G by utilizing the ILP formalization developed in Chakradhar *et al.* (1994) and the ILP formalization is as follows:

$$\text{Minimize}$$
$$\sum x_i$$
$$\text{Subject To}$$
$$w_i - w_j + n x_i \geq 1 \ \forall \ (v_i \rightarrow v_j) \in E$$

where $0 \leq w_i \leq n - 1$ and x_i is Boolean.

Then we can make usage of the above ILP formalization to perform the MFVS calculation in the critical/redundant MFVS calculation algorithm.

4.2.3 Determination of the critical/redundant status of the remaining vertices with unknown status

According to Corollary 3, for each pair (u, v) in M_{CHM}, if $u \in S_{RMFVS}$, add v into S_{RMFVS}. Note that there is a situation in which vertex u is not found in the contracted graph, which means that u is also a chain vertex of another vertex. In this situation there is a pair (u', u) in M_{CHM} and we firstly determine the critical/redundant status of vertex u by vertex u' and then determine the critical/redundant status of vertex v by vertex u.

Until now all the vertices in S_{CMFVS} are critical vertices, all the vertices in S_{RMFVS} are redundant vertices, and $S_{IMFVS} = V - S_{CMFVS} - S_{RMFVS}$.

4.3 Results

4.3.1 Computational analysis using artificial networks

In this section we give the test result for some artificial networks of the Erdős-Rényi structure as well as artificial networks of the scale-free structure. The definition for a scale-free network is: a network of which the degree distribution follows a power law. That is, the fraction $P(k)$ of nodes in the network having k connections to other nodes is valid for large values of k as $P(k) \sim k^{-\gamma}$, where γ is a parameter. Our algorithm is implemented using $C++$, and the test is conducted on a computer with a Linux operating system, an Intel (R) Xeon (R) CPU E5-2690, and memory of 32 GB. The source code is listed at http://sunflower.kuicr.kyoto-u.ac.jp/~houu/CMFVS_directed/index.html.

Firstly, the relationship between $|S_{MDS}|$, $|S_{MFVS}|$ as well as $|S_{CMFVS}|$, $|S_{RMFVS}|$, $|S_{IMFVS}|$ and the graph density are tested. The connection probability p_c is utilized between two vertices to define the graph density with the Erdős-Rényi graph, and a graph with a higher p_c generates a denser graph. For the scale-free networks, the ratio of the number of edges divided by the number of vertices are used to evaluate the density of a graph, and a graph with a higher edges/vertices fraction generates a denser graph.

The graph size $|V|$ of Erdős-Rényi graphs is set to 100 and $p_c \in \{0.01, 0.011, ..., 0.09\}$. *Boost Graph from Boost 1.61.0 in C++* is used to generate random graph. For scale free graphs, $|V| = 500$, the number of edges $|E| \in \{500, 550, 600, ..., 1900\}$, and $\gamma = 2.5$. *iGraph of version 1.0.2 in R* is used to generate the random graph.

Since we need to perform an MFVS calculation using an ILP optimization, an ILP optimizer needs to be selected to perform this task. In our implementation

we provided two kinds of ILP optimizing models, which are CPLEX from the IBM CPLEX Optimizer as well as glpsol from GLPK (GNU Linear Programming Kit) package, since the optimizing speed of CPLEX is much faster than that of glpsol, in the following test CPLEX is selected as our ILP optimizer.

The results are shown in Figure 4.9 and Figure 4.10. The results show that for both of the two kinds of graph structures, $|S_{MDS}|$ decreases and $|S_{MFVS}|$ increases as the graph becomes denser. This can be explained by the assumption that, as the density of the graph increases, each vertex is connected with additional vertices; thus, more edges need to be broken to break a cycle during the MFVS calculation and fewer vertices remaining in the graph are selected into an MDS if one vertex is selected into an MDS. Furthermore, for $|S_{CMFVS}|$, $|S_{RMFVS}|$, $|S_{IMFVS}|$, considered less significant, $|S_{RMFVS}|$ starts from a large value and continues decreasing as the graph becomes denser. This means that as the graph grows denser, $|S_{CMFVS}| + |S_{IMFVS}|$ becomes larger, which indicates that more control vertices need to be selected to control a denser network.

4.3.2 Efficiency of pre-processing of critical/redundant MFVS calculation for networks with different structure

We continue to evaluate the efficiency of the pre-processing procedure of our algorithm. For a graph generated via the Erdős-Rényi model, we evaluated the time cost of our algorithm with and without the pre-processing procedure using random graphs with $p_c \in \{0.01, 0.011, ..., 0.05\}$, and the graph size $|V| = 100$. For a graph with a scale-free structure, we used random graphs with a graph size of $|V| = 500$, the number of edges $|E| \in \{500, 550, 600, ..., 1900\}$, and $\gamma = 2.5$, the results are shown in Figure 4.11.

The results show that as the graph becomes denser, the efficiency of the pre-processing procedure is reduced for both Erdős-Rényi graphs and scale free graphs. This can be explained by the assumption that, as the graph becomes denser, there are fewer vertices in the graph that meet the condition for graph contraction, thus fewer vertices are contracted by the pre-processing procedure, and at some point the two lines representing the computational time with and without pre-processing coincide with each other, meaning that no vertices are being contracted by the pre-processing procedure.

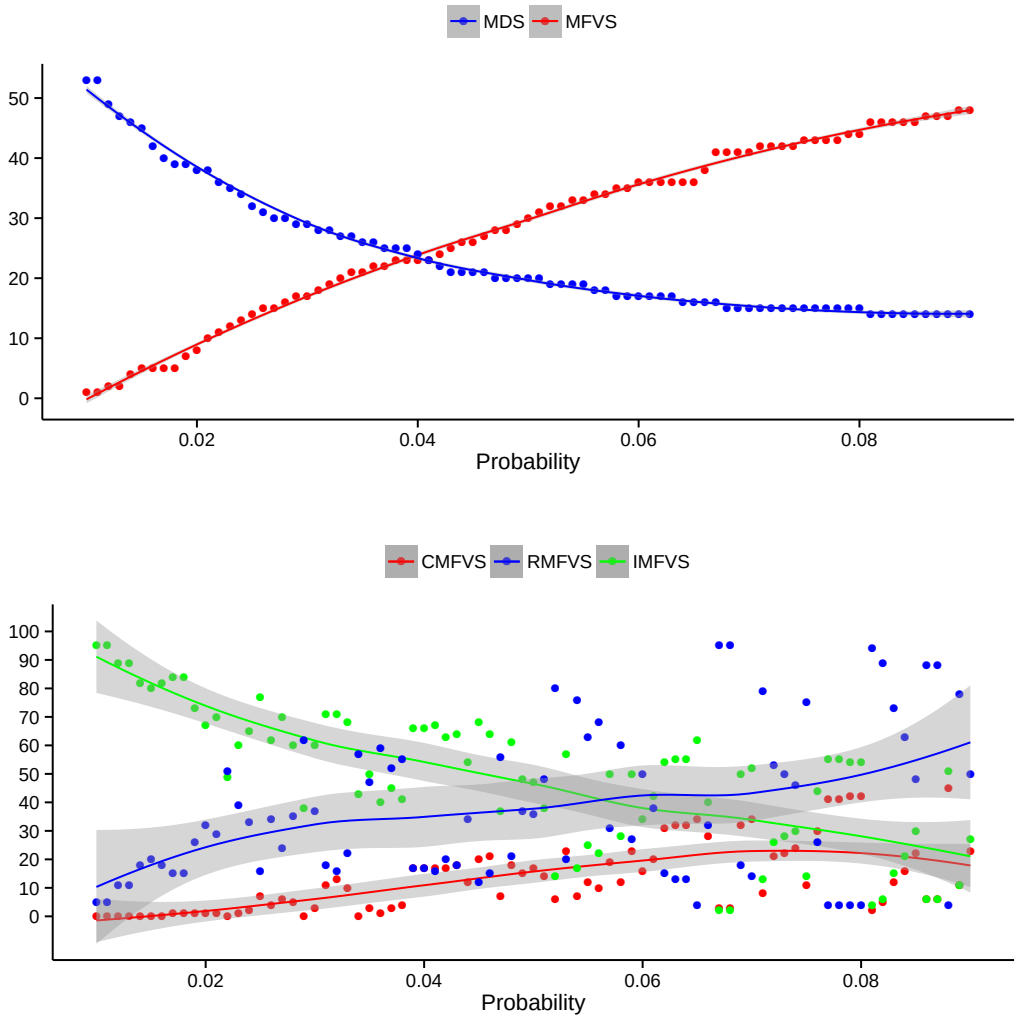


FIGURE 4.9: Relationships between p_c and $|MFVS|$, $|MDS|$ (top) as well as $|CMFVS|$, $|RMFVS|$, $|IMFVS|$ (bottom) for graphs with the Erdős-Rényi structure. The graph size $|V|$ is fixed at 100, and the x- and y-axes show the value of p_c and the result obtained for $|MFVS|$, $|MDS|$, $|CMFVS|$, $|RMFVS|$, and $|IMFVS|$, respectively. The results show that as p_c increases, $|MFVS|$ increases and $|MDS|$ decreases. And as p_c increases, $|RMFVS|$ decreases.

4.3.3 Computational analysis using real-world networks

In this section we utilize the technologies described above to explore the impact of the minimum feedback vertex sets on real biological systems such as signaling networks. Intracellular signaling pathways are important to coordinate developmental processes and cell responses to environmental changes in multi-cellular organisms composed of various tissues and organs. These organisms are referred to as metazoans (Pires-daSilva and Sommer, 2003). A recent paradigm shift has fueled the biological role and importance of these pathways by expanding the conceptual signaling functionality beyond simple linear cascades (Papin *et al.*, 2005). The resulting

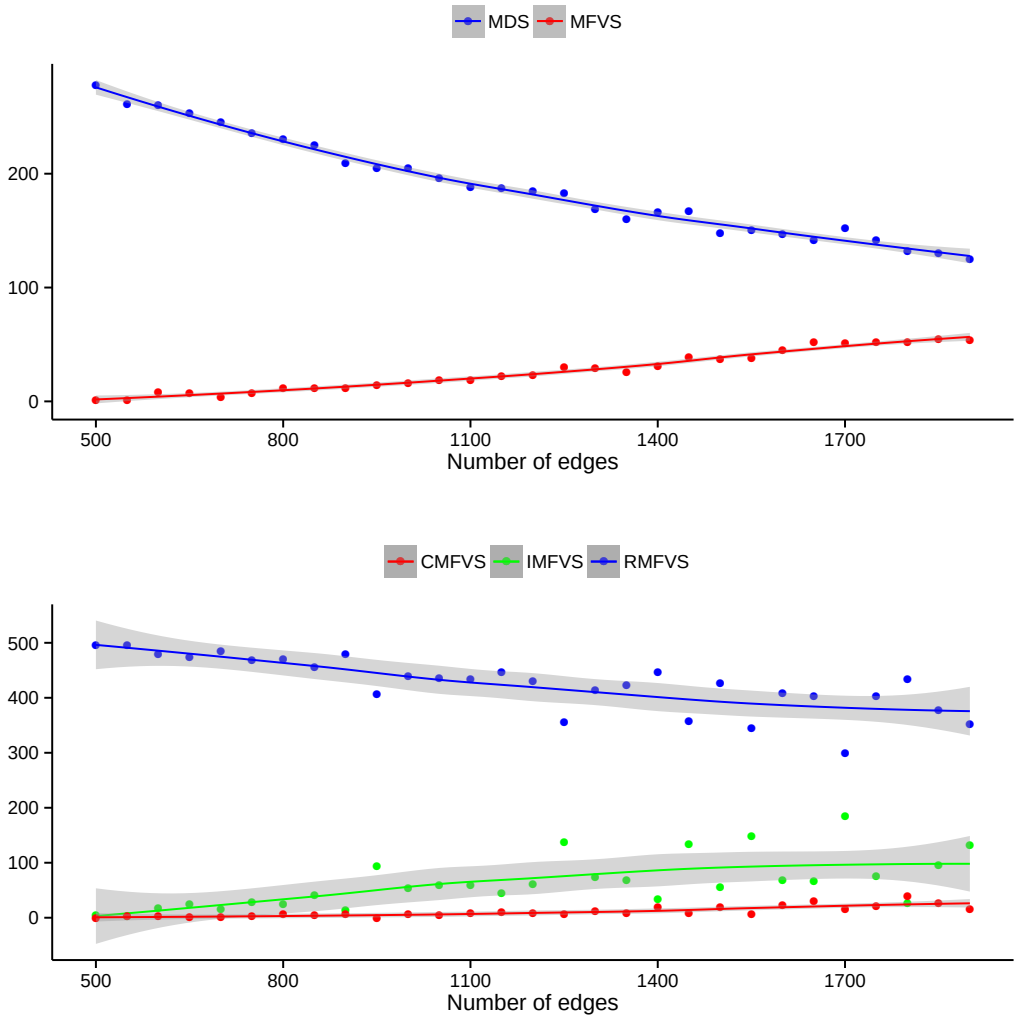


FIGURE 4.10: Relationships between the number of edges and $|MFVS|$, $|MDS|$ as well as $|CMFVS|$, $|RMFVS|$, $|IMFVS|$ for graphs with a scale-free structure. The graph size $|V|$ is fixed at 500, and the x- and y-axes show the number of edges and the results obtained for $|MFVS|$, $|MDS|$, $|CMFVS|$, $|RMFVS|$, and $|IMFVS|$. The results show that as the number of edges increases, $|MFVS|$ increases and $|MDS|$ decreases. Further, as the number of edges increases, $|RMFVS|$ decreases.

modern view also considers the complex interactions between pathways (cross-talks) as a functional part of the interdependent complex signaling network. Although the emergence of this new signaling network has encouraged further computational and statistical analysis, its importance has also been stressed by discoveries that link cancer, diabetes, and neurodegenerative disorders to specific failures in signaling pathways (Bauer-Mehren *et al.*, 2009).

Here, we used the SignaLink2.0 database to compile data for signaling proteins and directed signaling interactions between pairs of proteins of Caenorhabditis elegans, Drosophila melanogaster, and Homo sapiens organisms (Korcsmáros *et al.*, 2010). The

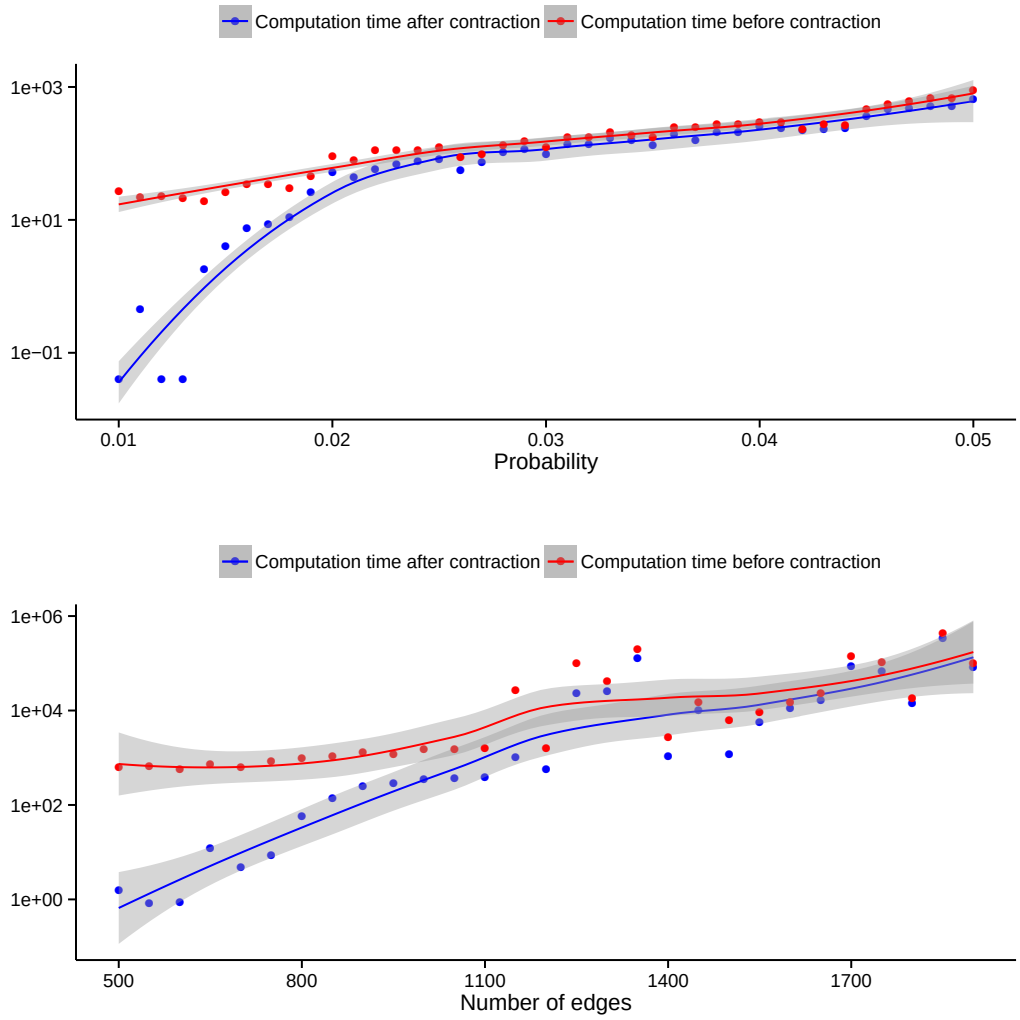


FIGURE 4.11: Relationships between computational time and density of the graph. The top and bottom figures show the results for graphs with the Erdős-Rényi and scale-free structures, respectively. The results indicate that as the graph becomes sparse, the computational time of both kinds of graph is satisfactory with the pre-processing procedure; however, as the graph becomes denser, the efficiency of pre-processing for both kinds of graph diminishes until at some point there is no difference between the computational time with and without the pre-processing procedure.

H. sapiens data correspond to the core of signaling proteins essential for transmitting the signal of its pathway. The multilayer structure of the SignaLink2.0 allows us to investigate several types of networks independently. Here, we considered a directed signaling protein interaction network, a network composed of transcriptional factors, and the integration of both of these networks. Table 4.1 shows the network statistics for each organism.

The computation of the MFVS and the associated control categories CMFVS, IMFVS, and RMFVS was performed on the three metazoans organisms and three types

of networks mentioned above. Fig. 4.12 shows that the MFVS fraction represents less than 10% of nodes in the directed PPI network and even around 5% or less for the integrated signaling network. This fraction decreases with biological complexity from human to *C. elegans*. Second, together with the absence of critical proteins in *C. elegans*, we observed that the critical fraction remains very small in *D. melanogaster* and *H. sapiens* organisms, less than 3% and 2%, respectively.

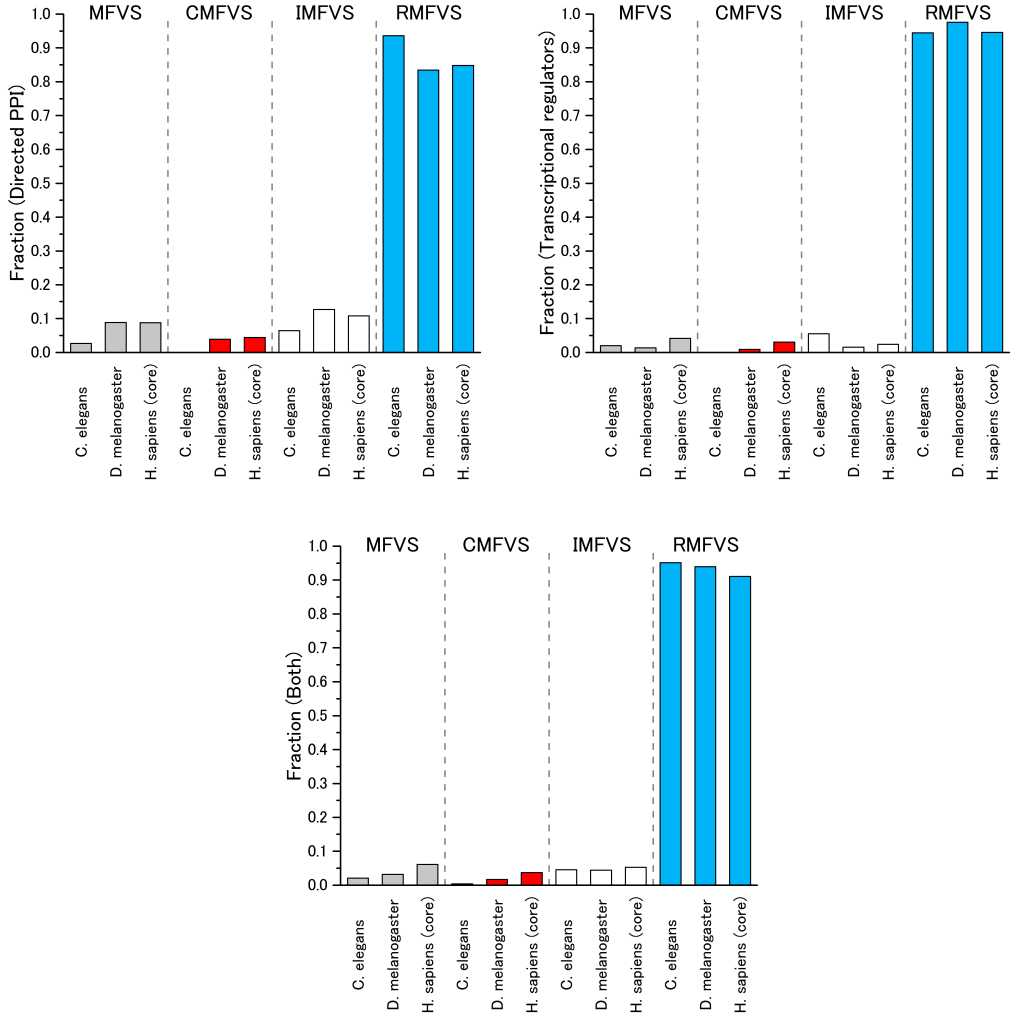


FIGURE 4.12: Fraction of nodes in each MFV set control category for each analyzed organism. The results are also classified by each type of network: (left) directed PPI network, (right) transcriptional regulators and (bottom) integrated network

Because the signaling network is responsible for continuous developmental processes, it is interesting to investigate the associations between essential genes and those proteins responsible for critical control. The enrichment results of essential genes associated to the CMFV and IMFV sets are given in Table 4.1. The results indicate that those proteins responsible for critical control tend to be significantly

enriched by essential genes in both *D. melanogaster* and *H. sapiens* organisms.

Moreover, for each network type and organism, we divided the datasets into seven signaling pathways as follows: RTK (including EGF/MAPK and Insulin/IGF), TGF-ß, Wingless/WNT, Hedgehog, JAK/STAT, Notch, and NHR (Nuclear hormone receptor). These pathways play important roles in cell development as well as in reprogramming different cellular responses to extracellular signals. Subsequently, the enrichment of each control category was computed for each pathway. Fig. 4.13 shows that a fraction of the pathways, namely Wingless/WNT, TGF-ß, and Hedgehog show a conserved enrichment of critical nodes (CFMVS) for the integrated directed PPI network with transcriptional regulators. Among the major functions shared by these three pathways, we found cell-fate and polarity determination, organ development, cell-fate determination, and proliferation. With the exception of the Notch pathway, the remaining pathways tend to have different major functions: EGF/MAPK involves growth, survival, differentiation, and cell-fate determination. JAK/STAT relates to proliferation, cell development, differentiation, apoptosis, and native and adaptive immune processes. NHR corresponds to development and growth, sex determination, regulation of cell metabolism. Finally, the Notch pathway is associated to cell-fate determination, proliferation, apoptosis, and organ development.

4.4 Discussion

In this work, we developed and implemented an algorithm to identify and evaluate the critical/redundant fraction of vertices in directed complex networks under the MFVS-based framework. One of the key feature of our development is to introduce the novel graph contraction operations for critical/redundant vertices to reduce the size of the original network.

The computational analysis we presented showed that, as a graph becomes denser, the fraction of redundant vertices decreases. The efficiency result of our pre-processing procedure showed that when the graph is sparse, our pre-processing step contributed to the performance significantly by reducing the computational time to 1/1000 – 1/100; however, while the graph becomes denser the efficiency of our pre-processing step was reduced dramatically. Therefore, improvements of our pre-processing procedure for dense graph are important future work.

The results of the analysis of the signaling networks showed that the fraction of MFVS vertices is very small for all three of the examined organisms. Furthermore, the results revealed that proteins corresponding to critical vertices tend to be enriched by essential genes in both *D. melanogaster* and *H. sapiens* organisms, especially in

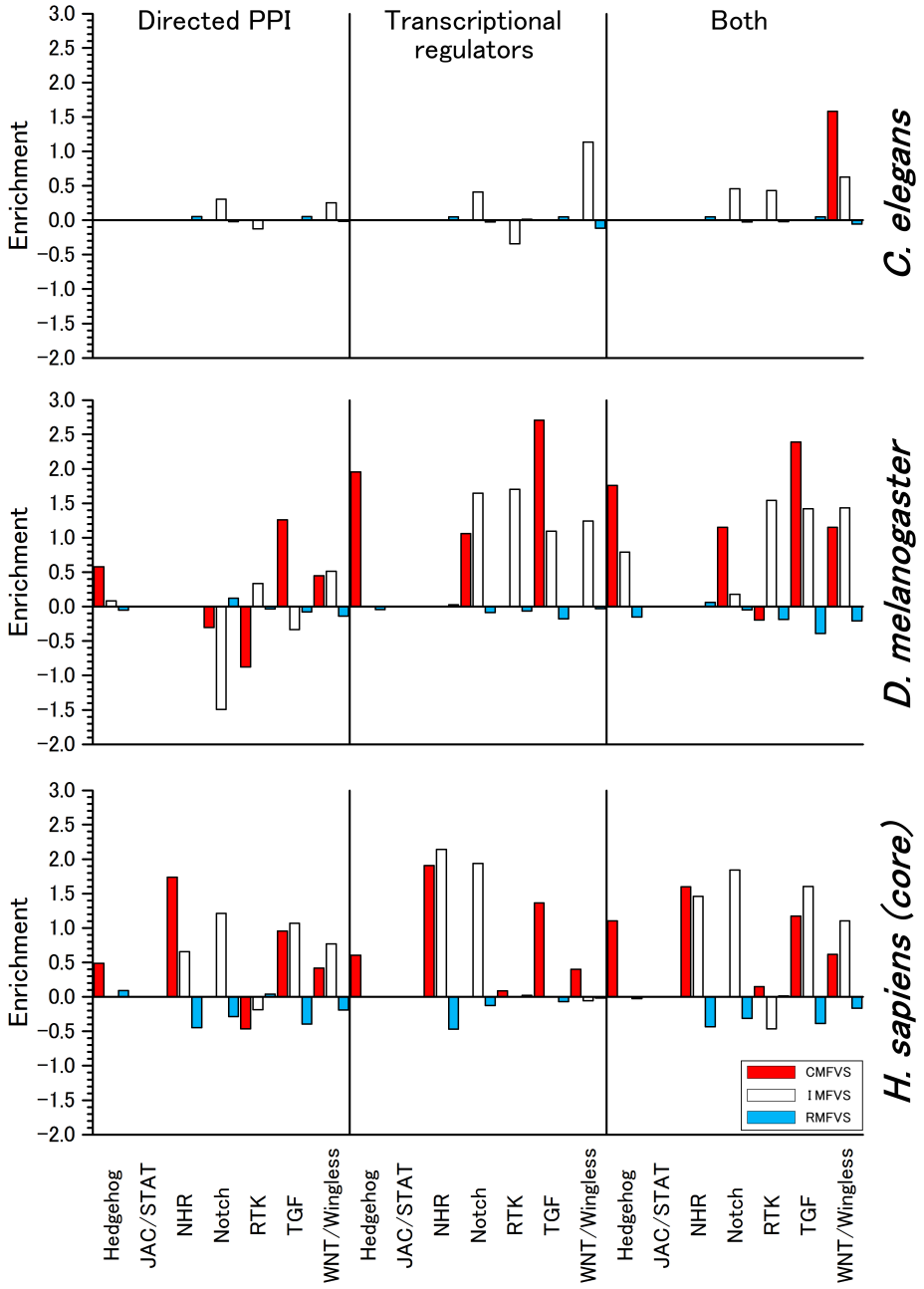


FIGURE 4.13: Enrichment results of each MFV set control category for each signaling pathway. The results are shown for three analyzed organisms: *C. elegans*, *D. melanogaster*, and *H. sapiens*. The enrichment results are also displayed in three types of networks: directed PPI network, transcriptional regulators, and the integrated network. A functional description of each signaling pathway can be found in the main text.

D. melanogaster. The results also showed that proteins corresponding to intermittent vertices tend to be enriched in these two organisms. A detailed analysis of the specific sub-pathways indicated that critical vertices are enriched by essential genes in pathways related to cell-fate and polarity determination, organ development, and cell-fate determination and proliferation. This finding suggests that our proposed approach

TABLE 4.1: **Statistics of the biological organisms and signaling networks analyzed in this work.** This table also shows the enrichment of the fraction of essential genes that are also identified as critical ($En_E(C)$) and intermittent control nodes ($En_E(I)$) by the MFV set algorithm. The statistical significance was assessed by performing Fisher's exact test and the two-tailed p-value is shown next to each enrichment value.

Directed PPI	*C. elegans*	*D. melanogaster*	*H. sapiens* (core)
#Node	190	181	296
#Edge	272	367	687
#CMFVS	0	7	13
#IMFVS	10	23	32
#RMFVS	180	151	251
#Essential	8	35	203
#CMFVS_Essential	0	2	12
#IMFVS_Essential	0	5	27
Enrichment(C)		0.39039	0.29711
p-value(C)		0.62210	0.06970
Enrichment(I)		0.11709	0.20725
p-value(I)		0.77880	0.04459

Transcriptional regulators	*C. elegans*	*D. melanogaster*	*H. sapiens* (core)
#Node	285	658	625
#Edge	402	5128	6966
#CMFVS	0	6	19
#IMFVS	14	10	15
#RMFVS	271	642	591
#Essential	10	69	405
#CMFVS_Essential	0	3	14
#IMFVS_Essential	1	2	13
Enrichment(C)		1.56195	0.12848
p-value(C)		0.01753	0.47420
Enrichment(I)	0.71085	0.64566	0.29076
p-value(I)		0.28260	0.09949

Both networks Combined	*C. elegans*	*D. melanogaster*	*H. sapiens* (core)
#Node	287	660	625
#Edge	447	5309	7270
#CMFVS	1	11	23
#IMFVS	13	29	33
#RMFVS	273	620	569
#Essential	10	69	405
#CMFVS_Essential	0	5	19
#IMFVS_Essential	1	6	27
Enrichment(C)		1.46968	0.24281
p-value(C)		0.00304	0.07711
Enrichment(I)	0.79195	0.68260	0.23319
p-value(I)	0.37570	0.19620	0.03913

based on the critical/redundant MFVS is useful for finding proteins and genes that have important control roles in directed biological networks.

Chapter 5

Conclusion

In Chapter 2 we proposed a novel method—LBSizeCleav—for prediction of Dicer cleavage sites. By integrating information on the length of a loop/bulge structure of a pre-miRNA (as predicted by the quikfold server), we developed novel feature space mapping. We performed fivefold cross-validation for validated pre-miRNA sequences from miRBase. In both 5p and 3p arms, the proposed method shows better performance than in the binary patterns of PHDCleav. This study shows a new way of feature evaluation; moreover, the better performance of our method points to the effectiveness of analysis of loop/bulge length at detecting Dicer cleavage sites. However in this study we only utilized the most commonly used RBF kernel as the kernel funcion of support vector classification, if we conbination various kernels together we may acquire better result compared with LBSizeCleav.

In Chapter 3 we reviewed and benchmarked 12 state-of-the-art sequence-based bioinformatics approaches and tools for caspases/granzyme B cleavage prediction. We evaluated and compared these methods in terms of their input/output, algorithms used, prediction performance, validation methods, and software availability and utility. In addition, we constructed independent datasets consisting of caspases/granzyme B substrates from different species and accordingly assessed the predictive power of these different predictors for the identification of cleavage sites. We found that the prediction results are highly variable among different predictors. Furthermore, we experimentally validated the predictions of a case study by performing caspase cleavage assay. We anticipate that this comprehensive review and survey analysis will provide an insightful resource for biologists and bioinformaticians who are interested in using and/or developing tools for caspase/granzyme B cleavage prediction.

In Chapter 4 we presented an algorithm as well as its implementation to compute and evaluate the critical, intermittent, and redundant vertices under the MFVS-based

framework, where these three categories include vertices belonging to all MFVSs, some (but not all) MFVSs, and none of the MFVSs, respectively. The results of computational experiments using artificially generated networks and real-world biological networks suggest that the proposed algorithm is useful for identifying these three kinds of vertices for relatively large-scale networks, and that the fraction of critical and intermittent vertices is considerably small. Moreover, an analysis of the signal pathways indicates that critical and intermittent MFVSs tend to be enriched by essential genes.

As a part of future work, more machine learning technologies such as kernel method shall be introduced into LBSizeCleav to improve the performance. For caspase/granzyme B cleavage prediction, more tools/algorithms shall be included to give a better overview and guidance for Bioinformaticians who research on capsase/granzyme B cleavage prediction. For criticality and redundancy MFVS prediction, since our algorithm is of low efficiency when the network is dense, it is imperative to develop algorithms effective in dense networks.

Publication list

Chapter 2 is based on an abstract presented at conference IBSB 2016 and paper Bao *et al.* (2016):

Chapter 3 is based on an abstract presented at conference SIGBIO 2017 and paper:
Y. Bao, S. Marini, T. Tamura, M. Kamada, S. Maegawa, H. Hosokawa, J. Song, T. Akutsu, Towards more accurate prediction of caspase cleavage sites: a comprehensive review of current methods, tools and features. Briefings in Bioinformatics, in press.

Chapter 4 is based on an abstract presented at conference IBSB 2017 and paper:
Y. Bao, M. Hayashida, P. Liu, M. Ishitsuka, JC. Nacher, T. Akutsu, Analysis of critical and redundant vertices in controlling directed complex networks using feedback vertex sets, Journal of Computational Biology, in press.

Appendix A

Protein Expression in Caspase Cleavage Assay

Both gluathione S transferase (GST) sequence and 5x myc tagged EGFP sequence were subcloned into pT7IRESmyc (pT7GSTmycEGFP). The DNA se- quences corresponding to candidate caspase target peptide sequences were hy- bridized and subcloned into pT7GSTmycEGFP. The fusion protein, containing candidate caspase target sequence, was made by using with human cell- free protein expression system (Clontech) according to manufactures instruction. Primer pairs used in this study are as follows:

DEVD:

GGGGCCCCTGGGATCCGACGAGGTGGACGGATCCCATCGATTTA

CCCCGGGGACCCTAGGCTGCTCCACCTGCCTAGGGTAGCTAAAT

IETD:

GGGGCCCCTGGGATCCATCGAGACCGACGGATCCCATCGATTTA

CCCCGGGGACCCTAGGTAGCTCTGGCTGCCTAGGGTAGCTAAAT

DSVDGSLT:

GGGGCCCCTGGGATCCGACAGCGTCGACGGCAGCCTGACCGGATCCCATCGATTT

CCCCGGGGACCCTAGGCTGTCGCAGCTGCCGTCGGACTGGCCTAGGGTAGCTAAAT

EEVDGAPR:

GGGGCCCCTGGGATCCGAGGAGGTGGACGGCGCTCCTCGGGGATCCCATCGATTTA

CCCCGGGGACCCTAGGCTCCTCCACCTGCCGCGAGGAGCCCCTAGGGTAGCTAAAT

DVVDGADT:

GGGGCCCCTGGGATCCGACGTGGTGGACGGCGCCGACACCGGATCCCATCGATTTA

CCCCGGGGACCCTAGGCTGCACCACCTGCCGCGGCTGTGGCCTAGGGTAGCTAAAT

EEVDGSSP:

GGGGCCCCTGGGATCCGAGGAGGTGGACGGCAGCAGCCCCGGATCCCATCGATTTA

CCCCGGGGACCCTAGGCTCCTCCACCTGCCGTCGTCGGGGCCTAGGGTAGCTAAAT

EEVDSGQG:
GGGGCCCCTGGGATCCGAGGAGGTGGACGGCAGCCAGGGCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTCCTCCACCTGCCGTCGGTCCCGCCTAGGGTAGCTAAAT
EETDGLDP:
GGGGCCCCTGGGATCCGAGGAGACCGACGGCCTGGACCCCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTCCTCTGGCTGCCGGACCTGGGGCCTAGGGTAGCTAAAT
EETDGLHE:
GGGGCCCCTGGGATCCGAGGAGACCGACGGCCTGCACGAGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTCCTCTGGCTGCCGGACGTGCTCCCTAGGGTAGCTAAAT
EEPDSANS:
GGGGCCCCTGGGATCCGAGGAGCCCGACAGCGCCAACAGCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTCCTCGGGCTGTCGCGGTTGTCGCCTAGGGTAGCTAAAT
AEVDGATP:
GGGGCCCCTGGGATCCGCCGAGGTGGACGGCGCCACCCCTGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCGGCTCCACCTGCCGCGGTGGGGACCTAGGGTAGCTAAAT
SEVDGNDS:
GGGGCCCCTGGGATCCAGCGAGGTGGACGGCAACGACAGCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTCGCTCCACCTGCCGTTGCTGTCGCCTAGGGTAGCTAAAT
EEPDGGFR:
GGGGCCCCTGGGATCCGAGGAGCCCGACGGCGGCTTCCGGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTCCTCGGGCTGCCGCCGAAGGCCCCTAGGGTAGCTAAAT
DDPDSAYL:
GGGGCCCCTGGGATCCGACGACCCTGACAGCGCCTACCTGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTGCTGGGACTGTCGCGGATGGACCCTAGGGTAGCTAAAT
NEVDGSNE:
GGGGCCCCTGGGATCCAACGAGGTGGACGGCAGCAACGAGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTTGCTCCACCTGCCGTCGTTGCTCCCTAGGGTAGCTAAAT
SEIDGLKG:
GGGGCCCCTGGGATCCAGCGAGATCGACGGCCTGAAGGGCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTCGCTCTAGCTGCCGGACTTCCCGCCTAGGGTAGCTAAAT
GEVDGKAI:
GGGGCCCCTGGGATCCGGCGAGGTGGACGGCAAGGCCATCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCCGCTCCACCTGCCGTTCCGGTAGCCTAGGGTAGCTAAAT
DDTDGLTP:
GGGGCCCCTGGGATCCGACGACACCGACGGCCTGACCCCTGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTGCTGTGGCTGCCGGACTGGGGACCTAGGGTAGCTAAAT

LESDSESL:
GGGGCCCCTGGGATCCCTGGAGAGCGACAGCGAGAGCCTGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGGACCTCTCGCTGTCGCTCTCGGACCCTAGGGTAGCTAAAT
TEPDSPSP:
GGGGCCCCTGGGATCCACCGAGCCCGACAGCCCCAGCCCCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTGGCTCGGGCTGTCGGGGTCGGGGCCTAGGGTAGCTAAAT
AEVDGVDE:
GGGGCCCCTGGGATCCGCCGAGGTGGACGGCGTGGACGAGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCGGCTCCACCTGCCGCACCTGCTCCCTAGGGTAGCTAAAT
DETDSGAG:
GGGGCCCCTGGGATCCGACGAGACCGACAGCGGCGCCGGCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTGCTCTGGCTGTCGCCGCGGCCGCCTAGGGTAGCTAAAT
TEMDSETL:
GGGGCCCCTGGGATCCACCGAGATGGACAGCGAGACCCTGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTGGCTCTACCTGTCGCTCTGGGACCCTAGGGTAGCTAAAT
LEMDSVLK:
GGGGCCCCTGGGATCCCTGGAGATGGACAGCGTGCTGAAGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGGACCTCTACCTGTCGCACGACTTCCCTAGGGTAGCTAAAT
TETDSVGT:
GGGGCCCCTGGGATCCACCGAGACCGACAGCGTGGGCACCGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTGGCTCTGGCTGACGCACCCGTGGCCTAGGGTAGCTAAAT
TEEDSVSV:
GGGGCCCCTGGGATCCACCGAGGAGGACAGCGTGAGCGTGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGTGGCTCCTCCTGTCGCACTCGCACCCTAGGGTAGCTAAAT
DETDSPTV:
GGGGCCCCTGGGATCCGACGAGACCGACAGCCCTACCGTGGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTGCTCTGGCTGTCGGGATGGCACCCTAGGGTAGCTAAAT
DEVDGAND:
GGGGCCCCTGGGATCCGACGAGGTGGACGGCGCCAACGACGGATCCCATCGATTTA
CCCCGGGGACCCTAGGCTGCTCCACCTGCCGCGGTTGCTGCCTAGGGTAGCTAAAT

Bibliography

Acland, A., Agarwala, R., Barrett, T., et al. 2014. Database resources of the national center for biotechnology information. *Nucleic Acids Research*. **42**(Database issue), D7.

Adam, Z. 1996. Protein stability and degradation in chloroplasts. *Plant Molecular Biology*. **32**(5), 773–783.

Adams, J. 2004. The proteasome: a suitable antineoplastic target. *Nature Reviews Cancer*. **4**(5), 349–360.

Ahmed, F., Kaundal, R., 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**(Suppl 14), S9.

Akutsu, T., Yang, Z., Hayashida, M., Tamura, T. 2012. Integer programming-based approach to attractor detection and control of boolean networks. *IEICE TRANSACTIONS on Information and Systems*. **95**(12), 2960–2970.

Albert, R., Barabási, A.L. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics*. **74**(1), 47–97.

Altschul, S.F., Madden, T.L., Schäffer, A.A., et al. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*. **25**(17), 3389–3402.

Anwar, A., Saleemuddin, M. 2001. Regulation of digestive proteolytic activity in the larvae of Spilosoma obliqua (Lep., Arctiidae). *Journal of Applied Entomology*. **125**(9-10), 577–582.

Apweiler, R., Biswas, M., Fleischmann, W., et al. 2001. Proteome Analysis Database: online application of interPro and CluSTr for the functional classification of proteins in whole genomes. *Nucleic Acids Research*. **29**(1), 44–48.

Ashar, P., Malik, S. 1994. Implicit computation of minimum-cost feedback-vertex sets for partial scan and other applications. In *Proceedings of the 31st annual Design Automation Conference*, pages 77–80. ACM.

Ashkenazi, A., Dixit, V.M., et al. 1998. Death receptors: signaling and modulation. *Science*. **281**(5381), 1305–1308.

Ayyash, M., Tamimi, H., Ashhab, Y. 2012. Developing a powerful in silico tool for the discovery of novel caspase-3 substrates: a preliminary screening of the human proteome. *BMC Bioinformatics*. **13**(1), 14.

Backes, C., Kuentzer, J., Lenhof, H.P., et al. 2005. GraBCas: a bioinformatics tool for score-based prediction of Caspase-and Granzyme B-cleavage sites in protein sequences. *Nucleic Acids Research*. **33**(suppl_2), W208–W213.

Bao, Y., Hayashida, M., Akutsu, T. 2016. Lbsizecleav: improved support vector machine (svm)-based prediction of dicer cleavage sites using loop/bulge length. *BMC bioinformatics*. **17**(1), 487.

Barkan, D.T., Hostetter, D.R., Mahrus, S., et al. 2010. Prediction of protease substrates using sequence and structure features. *Bioinformatics*. **26**(14), 1714–1722.

Barry, M., Bleackley, R.C. 2002. Cytotoxic T lymphocytes: all roads lead to death. *Nature Reviews Immunology*. **2**(6), 401–409.

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

Bateman, A., Coin, L., Durbin, R., et al. 2004. The Pfam protein families database. *Nucleic Acids Research*. **32**(suppl_1), D138–D141.

Bauer-Mehren, A., Furlong, L.I., Sanz, F. 2009. Pathway databases and tools for their exploitation: benefits, current limitations and challenges. *Molecular systems biology*. **5**(1), 290.

Bernstein, E., Caudy, A.A., Hammond, S.M., Hannon, G.J. 2001. Role for a bidentate ribonuclease in the initiation step of RNA interference. *Nature*. **409**(6818), 363–366.

Bhasin, M., Raghava, G. 2004. ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic Acids Research*. **32**(suppl 2), W414–W419.

Bhasin, M., Raghava, G. 2005. Pcleavage: an SVM based method for prediction of constitutive proteasome and immunoproteasome cleavage sites in antigenic sequences. *Nucleic Acids Research*. **33**(suppl_2), W202–W207.

Boeckmann, B., Bairoch, A., Apweiler, R., et al. 2003. The SWISS-PROT protein knowledgebase and its supplement Trembl in 2003. *Nucleic Acids Research*. **31**(1), 365–370.

Boldin, M.P., Goncharov, T.M., Goltseve, Y.V., et al. 1996. Involvement of MACH, a novel MORT1/FADD-interacting protease, in Fas/APO-1-and TNF receptor–induced cell death. *Cell*. **85**(6), 803–815.

Bonfil, R.D., Cher, M.L. 2011. The role of proteolytic enzymes in metastatic bone disease. *IBMS BoneKEy*. **8**(1), 16–36.

Bortner, C.D., Oldenburg, N.B., Cidlowski, J.A. 1995. The role of DNA fragmentation in apoptosis. *Trends in Cell Biology*. **5**(1), 21–26.

Boyd, S.E., Pike, R.N., Rudy, G.B., et al. 2005. PoPS: a computational tool for modeling and predicting protease specificity. *Journal of Bioinformatics and Computational Biology*. **3**(03), 551–585.

Burbidge, R., Trotter, M., Buxton, B., Holden, S. 2001. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computers & Chemistry*. **26**(1), 5–14.

Byvatov, E., Fechner, U., Sadowski, J., Schneider, G. 2003. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of Chemical Information and Computer Sciences*. **43**(6), 1882–1889.

Cai, C., Han, L., Ji, Z.L., Chen, X., Chen, Y.Z. 2003. SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Research*. **31**(13), 3692–3697.

Cai, X., Yuan, Z.M. 2009. Stochastic modeling and simulation of the p53-mdm2/mdmx loop. *Journal of Computational Biology*. **16**(7), 917–933.

Cardone, M.H., Roy, N., Stennicke, H.R., et al. 1998. Regulation of cell death protease caspase-9 by phosphorylation. *Science*. **282**(5392), 1318–1321.

Cerenius, L., Kawabata, S.i., Lee, B.L., et al. 2010. Proteolytic cascades and their involvement in invertebrate immunity. *Trends in Biochemical Sciences*. **35**(10), 575–583.

Chakradhar, S.T., Balakrishnan, A., Agrawal, V.D. 1994. An exact algorithm for selecting partial scan flip-flops. In *Proceedings of the 31st annual Design Automation Conference*, pages 81–86. ACM.

Chang, C.C., Lin, C.J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*. **2**(3), 27.

Chauhan, J.S., Mishra, N.K., Raghava, G.P. 2010. Prediction of GTP interacting residues, dipeptides and tripeptides in a protein from its evolutionary information. *BMC Bioinformatics*. **11**(1), 301.

Cheng, K.T., Agrawal, V.D. 1990. A partial scan method for sequential circuits with feedback. *IEEE Transactions on Computers*. **39**(4), 544–548.

Cohen, G.M. 1997. Caspases: the executioners of apoptosis. *Biochemical Journal*. **326**(1), 1–16.

Cowan, N.J., Chastain, E.J., Vilhena, D.A., Freudenberg, J.S., Bergstrom, C.T. 2012. Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PloS one*. **7**(6), e38398.

Crawford, E.D., Seaman, J.E., Agard, N., et al. 2013. The DegraBase: a database of proteolysis in healthy and apoptotic human cells. *Molecular & Cellular Proteomics*. **12**(3), 813–824.

Creagh, E.M., Conroy, H., Martin, S.J. 2003. Caspase-activation pathways in apoptosis and immunity. *Immunological Reviews*. **193**(1), 10–21.

Dostert, C., Pétrilli, V., Van Bruggen, R., et al. 2008. Innate immune activation through Nalp3 inflammasome sensing of asbestos and silica. *Science*. **320**(5876), 674–677.

Drucker, H., Wu, D., Vapnik, V.N. 1999. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*. **10**(5), 1048–1054.

duVerle, D., Takigawa, I., Ono, Y., et al. 2010. CaMPDB: a resource for calpain and modulatory proteolysis. In *Genome informatics. International Conference on Genome Informatics*, volume 22, pages 202–213.

duVerle, D.A., Mamitsuka, H. 2011. A review of statistical methods for prediction of proteolytic cleavage. *Briefings in Bioinformatics*. **13**(3), 337–349.

Earnshaw, W.C., Martins, L.M., Kaufmann, S.H. 1999. Mammalian caspases: structure, activation, substrates, and functions during apoptosis. *Annual Review of Biochemistry*. **68**(1), 383–424.

Ebina, T., Toh, H., Kuroda, Y. 2010. DROP: an SVM domain linker predictor trained with optimal features selected by random forest. *Bioinformatics*. **27**(4), 487–494.

Elbashir, S.M., Harborth, J., Lendeckel, W., Yalcin, A., Weber, K., Tuschl, T. 2001. Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells. *Nature*. **411**(6836), 494–498.

Everett, H., McFadden, G. 1999. Apoptosis: an innate immune response to virus infection. *Trends in Microbiology*. **7**(4), 160–165.

Feng, Y., Zhang, X., Graves, P., Zeng, Y. 2012. A comprehensive analysis of precursor microRNA cleavage by human Dicer. *RNA*. **18**(11), 2083–2092.

Franchi, L., Eigenbrod, T., Muñoz-Planillo, R., et al. 2009. The inflammasome: a caspase-1-activation platform that regulates immune responses and disease pathogenesis. *Nature Immunology*. **10**(3), 241–247.

Garay-Malpartida, H.M., Occhiucci, J.M., Alves, J., et al. 2005. CaSPredictor: a new computer-based tool for caspase substrate prediction. *Bioinformatics*. **21**(suppl_1), i169–i176.

Gerdes, J., Li, L., Schlueter, C., et al. 1991. Immunobiochemical and molecular biologic characterization of the cell proliferation-associated nuclear antigen that is defined by monoclonal antibody Ki-67. *The American Journal of Pathology*. **138**(4), 867.

Gillespie, D.T. 1977. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*. **81**(25), 2340–2361.

Goodwin, B.C et al. 1963. *Temporal organization in cells. A dynamic theory of cellular control processes.* London and New York: Academic Press.

Griffiths-Jones, S., Grocock, R.J., Van Dongen, S., Bateman, A., Enright, A.J. 2006. miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic Acids Research*. **34**(suppl 1), D140–D144.

Gromiha, M.M., Ou, Y.Y. 2013. Bioinformatics approaches for functional annotation of membrane proteins. *Briefings in Bioinformatics*. **15**(2), 155–168.

Gu, S., Jin, L., Zhang, Y., Huang, Y., Zhang, F., Valdmanis, P.N., Kay, M.A. 2012. The loop position of shRNAs and pre-miRNAs is critical for the accuracy of dicer processing in vivo. *Cell*. **151**(4), 900–911.

Hammond, S.M., Bernstein, E., Beach, D., Hannon, G.J. 2000. An RNA-directed nuclease mediates post-transcriptional gene silencing in Drosophila cells. *Nature*. **404**(6775), 293–296.

Hofacker, I.L. 2003. Vienna RNA secondary structure server. *Nucleic Acids Research*. **31**(13), 3429–3431.

Igarashi, Y., Eroshkin, A., Gramatikova, S., et al. 2006. CutDB: a proteolytic event database. *Nucleic Acids Research*. **35**(suppl_1), D546–D549.

Igarashi, Y., Heureux, E., Doctor, K.S., et al. 2008. PMAP: databases for analyzing proteolytic events and pathways. *Nucleic Acids Research*. **37**(suppl_1), D611–D618.

Ishitsuka, M., Akutsu, T., Nacher, J.C. 2016. Critical controllability in proteome-wide protein interaction network integrating transcriptome. *Scientific reports*. **6**, 23541.

Jenal, U., Fuchs, T. 1998. An essential protease involved in bacterial cell-cycle control. *The EMBO Journal*. **17**(19), 5658–5669.

Jin, Y., Lee, C.G. 2013. Single nucleotide polymorphisms associated with microRNA regulation. *Biomolecules*. **3**(2), 287–302.

Jones, D.T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*. **292**(2), 195–202.

Kabsch, W., Sander, C. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*. **22**(12), 2577–2637.

Kagami, H., Akutsu, T., Maegawa, S., Hosokawa, H., Nacher, J.C. 2015. Determining associations between human diseases and non-coding rnas with critical roles in network control. *Scientific reports*. **5**, 14577.

Kawashima, S., Kanehisa, M. 2000. AAindex: amino acid index database. *Nucleic Acids Research*. **28**(1), 374–374.

Korcsmáros, T., Farkas, I.J., Szalay, M.S., Rovó, P., Fazekas, D., Spiró, Z., Böde, C., Lenti, K., Vellai, T., Csermely, P. 2010. Uniformly curated signaling pathways reveal tissue-specific cross-talks and support drug target discovery. *Bioinformatics*. **26**(16), 2042–2050.

Krajewska, M., Wang, H.G., Krajewski, S., et al. 1997. Immunohistochemical analysis of in vivo patterns of expression of CPP32 (Caspase-3), a cell death protease. *Cancer Research*. **57**(8), 1605–1613.

Kumar, M., Gromiha, M.M., Raghava, G. 2008. Prediction of RNA binding sites in a protein using SVM and PSSM profile. *Proteins: Structure, Function, and Bioinformatics*. **71**(1), 189–194.

Lange, P.F., Overall, C.M. 2011. TopFIND, a knowledgebase linking protein termini with function. *Nature Methods*. **8**(9), 703–704.

Lau, P.W., Guiley, K.Z., De, N., Potter, C.S., Carragher, B., MacRae, I.J. 2012. The molecular architecture of human Dicer. *Nature Structural & Molecular Biology*. **19**(4), 436–440.

Lauber, K., Bohn, E., Kröber, S.M., et al. 2003. Apoptotic cells induce migration of phagocytes via caspase-3-mediated release of a lipid attraction signal. *Cell*. **113**(6), 717–730.

Lee, Y., Jeon, K., Lee, J.T., Kim, S., Kim, V.N. 2002. MicroRNA maturation: stepwise processing and subcellular localization. *The EMBO Journal*. **21**(17), 4663–4670.

Levy, H., Low, D.W. 1988. A contraction algorithm for finding small cycle cutsets. *Journal of algorithms*. **9**(4), 470–493.

Li, S.J., Hochstrasser, M. 1999. A new protease required for cell-cycle progression in yeast. *Nature*. **398**(6724), 246–251.

Lin, H.M., Jou, J.Y. 2000. On computing the minimum feedback vertex set of a directed graph by contraction operations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. **19**(3), 295–307.

Liu, Y.Y., Slotine, J.J., Barabási, A.L. 2011. Controllability of complex networks. *Nature*. **473**(7346), 167–173.

Lohmüller, T., Wenzler, D., Hagemann, S., et al. 2003. Toward computer-based cleavage site prediction of cysteine endopeptidases. *Biological Chemistry*. **384**(6), 899–909.

Lyngsø, R.B., Zuker, M., Pedersen, C.N. 1999. Internal loops in RNA secondary structure prediction. In *Proceedings of the third annual international conference on Computational molecular biology*, pages 260–267.

MacRae, I.J., Zhou, K., Li, F., Repic, A., Brooks, A.N., Cande, W.Z., Adams, P.D., Doudna, J.A. 2006. Structural basis for double-stranded RNA processing by Dicer. *Science*. **311**(5758), 195–198.

MacRae, I.J., Zhou, K., Doudna, J.A. 2007. Structural determinants of RNA recognition and cleavage by Dicer. *Nature Structural & Molecular Biology*. **14**(10), 934–940.

Markham, N.R., Zuker, M 2008. UNAFold. In *Bioinformatics*. Springer.

McGarry, T.J., Kirschner, M.W. 1998. Geminin, an inhibitor of DNA replication, is degraded during mitosis. *Cell*. **93**(6), 1043–1053.

Melkman, A.A., Akutsu, T. 2013. An improved satisfiability algorithm for nested canalyzing functions and its application to determining a singleton attractor of a boolean network. *Journal of Computational Biology*. **20**(12), 958–969.

Mizianty, M.J., Kurgan, L. 2011. Sequence-based prediction of protein crystallization, purification and production propensity. *Bioinformatics*. **27**(13), i24–i33.

Mochizuki, A., Fiedler, B., Kurosawa, G., Saito, D. 2013. Dynamics and control at feedback vertex sets. ii: A faithful monitor to determine the diversity of molecular activities in regulatory networks. *Journal of theoretical biology*. **335**, 130–146.

Muruve, D.A., Pétrilli, V., Zaiss, A.K., et al. 2008. The inflammasome recognizes cytosolic microbial and host DNA and triggers an innate immune response. *Nature*. **452**(7183), 103–107.

Nacher, J.C., Akutsu, T. 2012. Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control. *New Journal of Physics*. **14**(7), 073005.

Nacher, J.C., Akutsu, T. 2014. Analysis of critical and redundant nodes in controlling directed and undirected complex networks using dominating sets. *Journal of Complex Networks*. pages 394–412.

Nacher, J.C., Akutsu, T. 2016. Minimum dominating set-based methods for analyzing biological networks. *Methods*. **102**, 57–63.

Ng, K.L.S., Mishra, S.K. 2007. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*. **23**(11), 1321–1330.

Ng, S.K., Zhang, Z., Tan, S.H., et al. 2003. InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes. *Nucleic Acids Research*. **31**(1), 251–254.

Nicholson, D.W., Thornberry, N.A. 1997. Caspases: killer proteases. *Trends in Biochemical Sciences*. **22**(8), 299–306.

Novak, B., Pataki, Z., Ciliberto, A., Tyson, J.J. 2001. Mathematical model of the cell division cycle of fission yeast. *Chaos: An Interdisciplinary Journal of Nonlinear Science*. **11**(1), 277–286.

Papin, J.A., Hunter, T., Palsson, B.O., Subramaniam, S. 2005. Reconstruction of cellular signalling networks and analysis of their properties. *Nature reviews Molecular cell biology*. **6**(2), 99–111.

Park, J.E., Heo, I., Tian, Y., Simanshu, D.K., Chang, H., Jee, D., Patel, D.J., Kim, V.N. 2011. Dicer recognizes the 5 ‘ end of RNA for efficient and accurate processing. *Nature*. **475**(7355), 201–205.

Pellettieri, J., Fitzgerald, P., Watanabe, S., et al. 2010. Cell death and tissue remodeling in planarian regeneration. *Developmental Biology*. **338**(1), 76–85.

Piippo, M., Lietzén, N., Nevalainen, O.S., et al. 2010. Pripper: prediction of caspase cleavage sites from whole proteomes. *BMC Bioinformatics*. **11**(1), 320.

Pires-daSilva, A., Sommer, R.J. 2003. The evolution of signalling pathways in animal development. *Nature Reviews Genetics*. **4**(1), 39–49.

Prigogine, I., Nicolis, G. 1977. *Self Organization in Non-Equilibrium Systems*. J. Wiley and Sons, New York.

Rathmell, J.C., Thompson, C.B. 2002. Pathways of apoptosis in lymphocyte development, homeostasis, and disease. *Cell*. **109**(2), S97–S107.

Rawlings, N.D., Waller, M., Barrett, A.J., et al. 2013. MEROPS: the database of proteolytic enzymes, their substrates and inhibitors. *Nucleic Acids Research*. **42**(D1), D503–D509.

Rechsteiner, M., Rogers, S.W. 1996. PEST sequences and regulation by proteolysis. *Trends in Biochemical Sciences*. **21**(7), 267–271.

Rogers, S., Wells, R., Rechsteiner, M. 1986. Amino acid sequences common to rapidly degrade proteins: The PEST hypothesis. *Science*. **234**, 364–369.

Salvesen, G.S., Dixit, V.M. 1999. Caspase activation: the induced-proximity model. *Proceedings of the National Academy of Sciences*. **96**(20), 10964–10967.

Shao, J., Xu, D., Tsai, S.N., et al. 2009. Computational identification of protein methylation sites through bi-profile Bayes feature extraction. *PLoS ONE*. **4**(3), e4920.

Smith, G., Walford, R. 1975. The identification of a minimal feedback vertex set of a directed graph. *IEEE Transactions on Circuits and Systems*. **22**(1), 9–15.

Song, J., Tan, H., Shen, H., et al. 2010. Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics*. **26**(6), 752–760.

Song, J., Tan, H., Boyd, S.E., et al. 2011. Bioinformatic approaches for predicting substrates of proteases. *Journal of Bioinformatics and Computational Biology*. **9**(01), 149–178.

Song, J., Tan, H., Perry, A.J., et al. 2012. PROSPER: an integrated feature-based tool for predicting protease substrate cleavage sites. *PLoS ONE*. **7**(11), e50300.

Song, J., Wang, Y., Li, F., et al. 2018a. iprot-sub: a comprehensive tool for accurately mapping and predicting protease-specific substrates and cleavage sites. *Briefings in Bioinformatics*. **in press**.

Song, J., Li, F., Leier, A., et al. 2018b. PROSPERous: high-throughput prediction of substrate cleavage sites for 90 proteases with improved accuracy. *Bioinformatics*. **34**(4), 684–687.

Sowell, E.F., Haves, P. 2001. Efficient solution strategies for building energy system simulation. *Energy and buildings*. **33**(4), 309–317.

Suresh, M.X., Gromiha, M.M., Suwa, M. 2015. Development of a machine learning method to predict membrane protein-ligand binding residues using basic sequence information. *Advances in Bioinformatics*. **2015**.

Tatsuya, A. 2018. *Algorithms for Analysis, Inference, and Control of Boolean Networks*. World Scientific.

Thornberry, N.A., Rano, T.A., Peterson, E.P., et al. 1997. A combinatorial approach defines specificities of members of the caspase family and granzyme B Functional relationships established for key mediators of apoptosis. *Journal of Biological Chemistry*. **272**(29), 17907–17911.

Tyson, J.J., Chen, K., Novak, B. 2001. Network dynamics and cell physiology. *Nature Reviews Molecular Cell Biology*. **2**(12), 908–916.

Verspurten, J., Gevaert, K., Declercq, W., et al. 2009. SitePredicting the cleavage of proteinase substrates. *Trends in Biochemical Sciences*. **34**(7), 319–323.

Wang, C.C., Lloyd, E.L., Soffa, M.L. 1985. Feedback vertex sets and cyclically reducible graphs. *Journal of the ACM (JACM)*. **32**(2), 296–313.

Wang, H., Feng, L., Zhang, Z., et al. 2016. Crysalis: an integrated server for computational analysis and design of protein crystallization. *Scientific Reports*. **6**, 21383.

Wang, M., Zhao, X.M., Tan, H., et al. 2013. Cascleave 2.0, a new approach for predicting caspase and granzyme cleavage targets. *Bioinformatics*. **30**(1), 71–80.

Wang, W.X., Ni, X., Lai, Y.C., Grebogi, C. 2012. Optimizing controllability of complex networks by minimum structural perturbations. *Physical Review E*. **85**(2), 026115.

Ward, J.J., McGuffin, L.J., Bryson, K., et al. 2004. The DISOPRED server for the prediction of protein disorder. *Bioinformatics*. **20**(13), 2138–2139.

Wee, L.J., Tan, T.W., Ranganathan, S. 2006. SVM-based prediction of caspase substrate cleavage sites. *BMC Bioinformatics*. **7**(5), S14.

Wee, L.J., Tan, T.W., Ranganathan, S. 2007. CASVM: web server for SVM-based prediction of caspase substrates cleavage sites. *Bioinformatics*. **23**(23), 3241–3243.

Wee, L.J., Tong, J.C., Tan, T.W., et al. 2009. A multi-factor model for caspase degradome prediction. *BMC Genomics*. **10**(3), S6.

Yuan, J., Shaham, S., Ledoux, S., et al. 1993. The C. elegans cell death gene ced-3 encodes a protein similar to mammalian interleukin-1β-converting enzyme. *Cell*. **75**(4), 641–652.

Zamore, P.D., Tuschl, T., Sharp, P.A., Bartel, D.P. 2000. RNAi: double-stranded RNA directs the ATP-dependent cleavage of mRNA at 21 to 23 nucleotide intervals. *Cell*. **101**(1), 25–33.

Zavaljevski, N., Stevens, F.J., Reifman, J. 2002. Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*. **18**(5), 689–696.