2009

Models for the optical dispersion relations of amorphous semiconductors

Wei Lu
University of Windsor

NOTE TO USERS

Minimal siRNA Set Cover Heuristic for Gene Family Knockdown

By

Xiaoguang Li

A Thesis Submitted to the Faculty of Graduate Studies through the School of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Science at the University of Windsor

Windsor, Ontario, Canada

2008

Your file Votre référence
ISBN: 978-0-494-47068-8
Our file Notre référence
ISBN: 978-0-494-47068-8

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:
[This thesis also incorporates the outcome under the supervision of professor Alioune Ngom and Luis Rueda . The collaboration is covered in Chapter 4 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the provision of some key ideas."]

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes [*1*] original papers that have been previously published/submitted for publication in peer reviewed conference, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter 4	*Minimal siRNA Set Cover Heuristic for Gene Family Knockdown*	*[accepted for publication]*

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University of Institution.

Abstract

RNA interference (RNAi) is a highly evolutionally conserved process of post-transcriptional gene silencing (PTGS) by which double stranded RNA (dsRNA), when introduced into a cell, causes sequence-specific degradation of homologous mRNA sequences, siRNA (small interfering RNA are a class of 20-25 nucleotide-long double-stranded RNA molecules) is involved in the RNA interference (RNAi) pathway where the siRNA interferes with the expression of a specific gene. We focus on the problem of gene family knockdown by using the minimal number of siRNAs. The problem is to determine the minimal number of siRNAs required to knockdown a family of genes targeted by these siRNAs. This is a minimal set covering problem, and hence it is NP-hard. In this thesis, we explore a number of heuristic optimization methods for the minimal siRNA covering problem. Such methods include evolutionary heuristics, as well as novel greedy methods, applied for the first time to the minimal siRNA cover problem. Preliminary experiments with genetic algorithms show significant reduction in the siRNA cover size, when compared with branch&bound and probabilistic greedy. We are currently implementing novel greedy methods which are variants of well-known feature subset selection algorithms. In such methods, we define criterion functions over a collection of siRNA subsets to help us decide which subset is best to be included in a candidate solution.

We use three gene families: the FREP genes from Biomphalaria glabrata and the olfactory genes from Caenorhabditis elegans. We also conducted experiments on one artificial data set.

Acknowledgements

First of all, I would like to thank Dr. Robert Kent, Dr. Kevin W. Li and Dr. Dan Wu, for their guidance, elaborate instruction and valuable technical assistance through the course of this thesis.

I also wish to express my deep sense of gratitude to Dr. Alioune Ngom, my advisor. His kind instruction and comments in the preparation of this thesis is in invaluable. I can not complete this thesis without his inspiring idea and assistance.

Also, I express my great appreciation to my parents, husband and friends for their encouragement and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

1. Introduction

RNA interference (RNAi) was first discovered in 1998 by Andrew Fire and Craig Mello in the nematode worm *Caenorhabditis elegans* and later found in a widely number of organisms, including mammals. RNA interference (RNAi) plays both regulatory and immunological roles in the eukaryotic genetic system [1, 2], and it also involved in both therapeutic and genomic applications because of its potentials in treatments for widely existed diseases such as HIV [3, 4], Huntington's diseases [5] and some certain types of cancers [6, 7]. RNA interference (RNAi) is a mechanism that inhibits gene expression at the stage of translation by hindering the transcription of specific genes. RNAi targets include RNA from viruses and transposons (significant for some forms of innate immune response), and work on regulating development and genome maintenance. Small interfering RNA strands (siRNA) play a key role in the RNAi process, and have complementary nucleotide sequences to the targeted RNA strand. Specific RNAi pathway proteins are guided by the siRNA to the targeted messenger RNA (mRNA), where they cleave the target, breaking it down into smaller portions which can not be translated into protein any more. A type of RNA transcribes from the genome itself, microRNA (miRNA), works in the same way [8].

Nowadays, RNAi research mainly focus on single gene knockdown. Gene knockdown relates to genetically modifying an organism whose goal is to have reduced expression of one or more genes in its chromosomes by inserting a reagent such as a short DNA or RNA oligonucleotide with a sequence complementary to an active gene or its mRNA

transcripts. This can lead to permanent modification of the chromosomal DNA to produce a "knockdown organism" or a temporary change in gene expression without modification of the chromosomal DNA molecules to knock down the function of a single gene. In this thesis, we want to knockdown a gene family with a minimal number of siRNAs because the efficacy of a specific siRNA in knocking down its target gene is determined by its homology to that gene. As the synthesis of individual siRNAs may cost hundreds or thousands of dollars, so using compact sets of siRNAs for gene family knockdown would have more advantages.

Following association with an RNAi silencing complex, siRNA targets mRNA transcripts that have sequence identity for destruction. A phenotype resulting from this knockdown of expression may inform about the function of the targeted gene. However, off-target effects compromise the specificity of RNAi if sequence identity between siRNA and random mRNA transcripts causes RNAi to knockdown expression of non-targeted genes. The chance for off-target RNAi increases with greater length of the initial dsRNA （double strand RNA is RNA with two complementary strands sequence）, inclusion into the analysis of available un-translated region sequences and allowing for mismatches between siRNA and target sequences. siRNA sequences from within 100 nucleotide of the 5' termini of coding sequences have low chances for off-target reactivity. This may be owing to coding constraints for signal peptide-encoding regions of genes relative to regions which encode for mature proteins. Off-target distribution varies along the chromosomes of *Caenorhabditis elegans*, apparently owing to the use of more unique sequences in gene-dense regions. Finally, biological and thermodynamical descriptors of

effective siRNA reduce the number of potential siRNAs compared with those identified by sequence identity alone, but off-target RNAi remains likely, with an off-target error rate of 10% [11]. In a word, we want to avoid off-target effects in which the siRNA causes unintended knockdown of an untargeted gene to which it incidentally has high homology. So our purpose is to select a minimal set of siRNAs that cover targeted genes in a family and do not cover any untargeted genes. This is a NP-Hard problem [9] since we can regard it as a set cover problem. In this paper, we introduce four heuristics for this problem: a genetic algorithm-based heuristic, a dominated target covering heuristic, a dominant siRNA selection heuristic and a forward selection heuristic. Our experiment results show that our methods significantly reduce the number of siRNA covers compared with other two algorithms: branch and bound, probabilistic greedy [9].

We implement our proposed methods on three gene families. The first family, which is the set of Fibrinogen-related protein (FREP) genes from the snail Biomphalaria glabrata are medically relevant because this snail is a model organism for infection by the human-affecting parasite Schistosoma mansoni [9]. The second family is another set of FREP genes like family 1[10]. And the data of third family, which is the olfactory genes of nematode Caenorhabditis elegans, is downloaded from NCBI [12].

Chapter 2

2. Gene Knockdown

Gene knockdown involves techniques through one or more of an organism's genes expression which is reduced by genetic modification (a change in the DNA of one of the

organism's chromosomes) or by treatment with a reagent, for instance: a short DNA or RNA oligonucleotide to either an mRNA transcript or a gene. The result is a: knockdown organism when genetic modification of DNA is done. If the change in gene expression is result from an oligonucleotide binding to an mRNA or temporarily binding to a gene, this leads to a temporary change in gene expression without modification of the chromosomal DNA and the result is as a: transient knockdown.

Transient knockdown decreased expression through blocking transcription, degradation of RNA transcript and blocking either mRNA translation, pre-mRNA splicing sites or nuclease cleavage sites used for maturation of other functional RNAs such as miRNA [37] by binding oligonucleotide to the active gene or its transcripts. Transient knockdowns for learning about a gene have been sequenced as the most direct use. It also has an unknown or incompletely known function, an experimental approach known as reverse genetics. Researchers infer how the knockdown differs from individuals in which the gene of interest is operational by an experimental approach. Since oligos can be injected into single-celled zygotes and will be present in the daughter cells of the injected cell through embryonic development [38], transient knockdowns are often used in developmental biology.

Heretofore, knockdown organisms with permanent alterations in their DNA have been applied mainly for research purposes. These organisms are most commonly used for reverse genetics such as mice or rats, because they cannot easily be applied through transient knockdown technologies.

2.1 RNA Interference

RNA interference refers to a mechanism in eukaryotic cells that leads to a specific posttranscriptional gene silencing in response to long double stranded RNA. In the cell the long dsRNA are cut into 21-25 bp pieces by an enzyme called DICER, a member of the RNase III family of dsRNA specific ribonucleases. The resulting small dsRNAs activates a large protein complex called the RNA-induced silencing complex (RISC) which, by using one of the strands as a guide, binds to the corresponding mRNA and degrades it, leading to a specific 'knock down' of protein expression. In vertebrate cells investigations of RNAi initially gave problems with the vertebrate interferon defense system, which by several complex pathways leads to an overall and therefore nonspecific cellular shut down of protein expression. As the interferon response is thought to be triggered only by dsRNAs longer than 30 bp, smaller dsRNAs about 21 base pairs long, termed small interfering RNAs or simply siRNAs, have been explored for the induction of specific RNAi in vertebrate cells. These smaller dsRNAs are thought to function as equivalents of the small dsRNAs cut by DICER.

RNAi is an RNA-dependent gene silencing process that is controlled by the RNA-induced silencing complex (RISC) and is initiated by short double-stranded RNA molecules in a cell's cytoplasm, where they interact with the catalytic RISC component argonaute [14]. When the dsRNA is exogenous (coming from infection by a virus with an RNA genome or laboratory manipulations), the RNA is imported directly into the cytoplasm and cleaved to short fragments by the enzyme dicer. The initiating dsRNA can also be endogenous (originating in the cell), as in pre-microRNAs expressed from RNA-coding genes in the genome. The primary transcripts from such genes are first processed to form

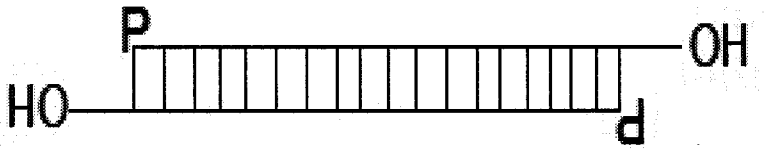
the characteristic stem-loop structure of pre-miRNA in the nucleus, then exported to the cytoplasm to be cleaved by dicer. Thus, the two dsRNA pathways, exogenous and endogenous, converge at the RISC complex [15]. .

2.1.2 Importance of RNAi

Gene expression is the process by which the information encoded in a gene is converted into amino acid sequences. When a gene is expressed, DNA is transcribed into mRNA which then acts as a template for the production of proteins. Thus, degradation and regulation of mRNA help govern cellular mRNA and, therefore, protein levels that result from gene expression. Complete genomes are being sequenced for several organisms and there is an increasing need for studying gene behaviors and functions. Changes in phenotype, resulting from RNAi, gives information about the functions of the targeted gene. Therefore, a mechanism like RNAi, which employs existing cell machinery, is highly useful. RNAi is also becoming increasingly important in developing therapeutic applications for a number of diseases due to its potential for specific targeted silencing [23]. During gene expression, there are three stages where genes causing diseases can be controlled – transcriptional, post-transcriptional, and post-translational intervention. Traditionally, drugs for disease control have been targeted towards proteins, which occur in the posttranslational phase. RNAi targets the protein-producing mRNA and can thereby control disease earlier - in the transcription phase. RNAi has been successfully used to target diseases such as AIDS [24], neurodegenerative diseases [25], cholesterol [26] and cancer [27] on mice with the hope of extending these approaches to treat humans.

2.2 siRNA

A nucleotide (nt) is a subunit of DNA or RNA and is made up of one of adenine (A), guanine (G), cytosine (C) or uracil (U) (in RNA) or thymine (T) (in DNA), along with a phosphate molecule, and a sugar molecule. The RNA molecule is formed from a sequence of these nucleotides. The complementary nucleotides of A, C, G and U are U, G, C and A respectively. When long dsRNA from an external source is introduced into the cell, it is recognized by Dicer, a member of the RNase III family of dsRNA-specific ribonucleases. Dicer cleaves the dsRNA to produce siRNA duplexes of lengths 19 - 21 nt [16]. Each siRNA strand has a 5' phosphate group and a 3' hydroxyl group and has a 2 nt overhang at the 3' end [17]. The siRNA duplex separates into sense and antisense strands and one of the strands is taken up by a RNA-protein complex, referred to as RISC [18].



Schematic representation of a siRNA molecule: a ~19-21basepair RNA core duplex that is followed by a 2 nucleotide 3' overhang on each strand. OH: 3' hydroxyl; P: 5' phosphate.

Figure 1.1 Anatomy of an siRNA

(From http://en.wikipedia.org/wiki/Image:SiRNA_structure2.jpg)

Activation of RISC requires an ATP-dependent unwinding of the siRNA duplex. Both the sense and antisense strands of the siRNA are capable of directing RNAi but specificity depends on the anti-sense strand. The active RISC then targets mRNA transcripts that

have sequence complementarity with the siRNA sequence. The targeted mRNA sequences are cleaved into smaller fragments which are then degraded. This results in sequencespecific removal of mRNA in targeted genes, which are then not expressed at the protein level. Figure 1.2 graphically illustrates the RNAi pathway initiated by the introduction of dsRNA. The knockdown effects induced by RNAi are usually transient but using vectorbased delivery methods, stable RNAi can be induced. RNAi is not immediate and there is a time course associated with the process. RNAi has also been shown to be inheritable in *C. elegans* [19]. In mammals, it was observed that long dsRNAs, with lengths more than 30 nt activate the PKR kinase pathway in the cell, also known as the interferon response. This causes non-specific degradation of mRNA, and leads to apoptosis or cell death. However, using synthesized siRNAs of lengths 21 - 23 nt [20] does not evoke the interferon response and provides effective silencing by RNAi.

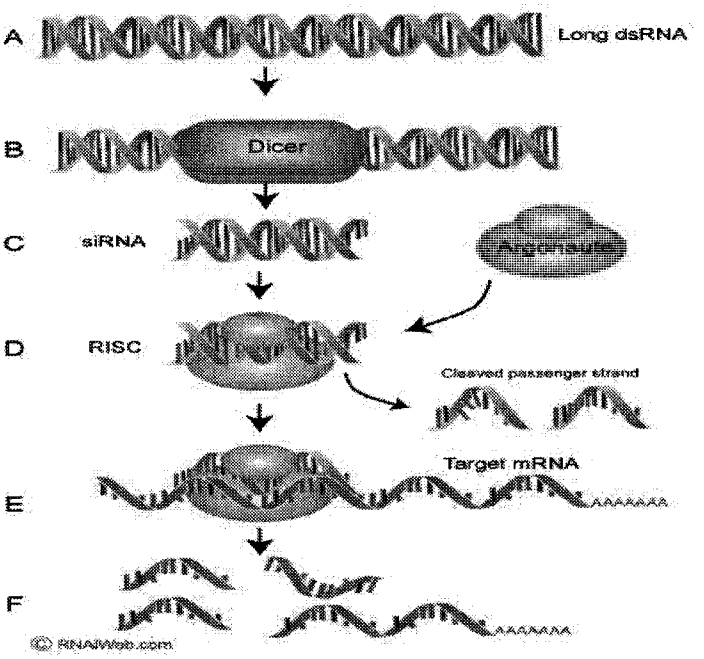


Figure 1.2 RNA Interference Pathway (From RNAiWeb.com)

In addition to siRNAs, gene silencing can also be caused by micro RNAs (miRNA). miRNAs are small RNAs, processed from double stranded hairpin structures that are encoded in the genome, and are believed to be involved in gene regulation. Unlike siRNAs, which work by mRNA degradation, miRNA work by suppressing translation of mRNA to protein. miRNAs have been shown to function as siRNAs by binding to perfectly complementary mRNA sequences to cause degradation. On the other hand, siRNAs can act as mRNAs with 3 - 4 nt mismatches and G-U mismatches [21]. This demonstrates that it is only the degree of complementarity which determines the functionality of the siRNA or miRNA sequence [22]. However, the effects of miRNA-like behavior of siRNAs on efficacy experiments have not been extensively studied so far. This work only deals with the siRNA and so this document will not delve in the details of miRNA, but it is important to note that they are closely related.

2.2.1 siRNA Sequence Design

Three different kinds of software are implemented in this thesis for the siRNA sequence design - BLOCK-iT™ RNAi Designer [28], siDirect [29] and TROD (T7 RNAi Oligo Designer) [30].

BLOCK-iT™ RNAi Designer [28] uses a patented algorithm. The reasonable scheme which is used by this software not only supposition but also based on the practical experiment results. The experiment certified that RNAi Designer can choose unique target sequences which will remarkably increase the probability of off-target genes. BLOCK-iT™ RNAi Designer identifies effective RNAi sequences then uses BLAST to

select from a widely organism specific database which can be ensured to eliminate the off-target RNAi sequences.

Rational siRNA design is used by siDirect [29]. The general guide lines [31] are shown as follows:

1. siRNA targeted sequence is usually 21 nt in length.

2. Avoid regions within 50-100 bp of the start codon and the termination codon

3. Avoid intron regions

4. Avoid stretches of 4 or more bases such as AAAA, CCCC

5. Avoid regions with GC content <30% or > 60%.

6. Avoid repeats and low complex sequence

7. Avoid single nucleotide polymorphism (SNP) sites

8. Perform BLAST homology search to avoid off-target effects on other genes or sequences

9. Always design negative controls by scrambling targeted siRNA sequence. The control RNA should have the same length and nucleotide composition as the siRNA but have at least 4-5 bases mismatched to the siRNA. Make sure the scrambling will not create new homology to other genes.

By experimentally analyzing the silencing efficiency of 180 siRNAs targeting the mRNA of two genes and correlating it with various sequence features of individual siRNAs, Reynolds et al [32] at Dharmacon, Inc identified eight characteristics associated with siRNA functionality. These characteristics are used by rational siRNA design algorithm to evaluate potential targeted sequences and assign scores to them. Sequences with higher

scores will have higher chance of success in RNAi. Table 2.1 below lists the 8 criteria and the methods of score assignment.

Criteria	Description	Score	
		Yes	No
1	Moderate to low (30%-52%) GC Content	1 point	
2	At least 3 A/Us at positions 15-19 (sense)	1 point /per A or U	
3	Lack of internal repeats (Tm*<20¡ãC)	1 point	
4	A at position 19 (sense)	1 point	
5	A at position 3 (sense)	1 point	
6	U at position 10 (sense)	1 point	
7	No G/C at position 19 (sense)		-1 point
8	No G at position 13 (sense)		-1 point

Table 2.1 Reynolds' rational design rules [32]

TROD (T7 RNAi Oligo Designer) [30] is a web application that facilitates the design of DNA oligonucleotides for the synthesis of short interfering RNAs (siRNAs) with T7 RNA polymerase. TROD looks for all occurrences of the sequence $N_2GN_{18}C$. The G and C nucleotides are required for efficient synthesis of the RNA by T7 RNA polymerase, since it strongly favours a G at the start site. From these sequences, the program generates both the sense and antisense strands of DNA oligos that will be used to produce the siRNAs *in vitro*. By definition, the 'antisense' DNA oligo encodes the antisense siRNA strand, and vice versa. The following (reverse complemented) T7 promoter is appended to the 3' ends of the DNA oligos: 5'-TATAGTGAGTCGTATTA-3'. After T7 transcription, the siRNA duplexes will look like the following: sense siRNA: 5'-GNNNNNNNNNNNNNNNNNUUU-3'

antisense siRNA: 3'-NNCNNNNNNNNNNNNNNNNG-5'

TROD, by default, appends an AA dinucleotide to the 5' end of the DNA 'sense' strand, since AT-richness is preferred at this site. The result (as shown above) is a UU dinucleotide overhang on the sense siRNA after T7 transcription. In addition, in order to destabilize the 5' end of the antisense siRNA (so as to direct it to the RISC complex), a U replaces the complementary C, producing a GU pair.

2.2.2 Efficaciousness of siRNA

From the current research, the result shows that not all possible siRNA can be synthesized against a specific target, but a subsection of them would take effect in causing any degradation [33] and more, all siRNA do not result in equal knockdown effects [33]. As we found out the efficacy of siRNA even in the same target mRNA differ in different target sites, here to select efficiency ones which are extremely functional causing a certain higher percentage of the target mRNA sequence to degrade. There is variance on the importance of each of these properties. Reynolds et al. [32] launched their siRNA knockdown experiments and concluded that properties of the target mRNA did not affect the efficacy of knockdown but be solely based on properties of the siRNA. However, other studies [34, 35] showed that secondary structure and thermodynamic features (related to stability) of the siRNA are also important determining factors of functionality. Up on most studies, siRNA with knockdown of greater than 80% of the target mRNA are considered highly efficient while threshold varies based on the required level of silencing. So to help designing siRNA sequences to highly efficient against target mRNA sequences, it is quite important to have siRNA efficacy prediction.

2.2.3 Specificity of siRNA

To better design siRNA sequences, the specificity of it is equaled important with potency or efficacy. While maximum degradation of target mRNA is required, silencing of non-target mRNA should be avoided. To achieve the objective of maximally degrade target mRNA, avoid silencing of non-target mRNA is needed. As siRNA gene silencing is considered to be highly sequence-specific, even one single base mismatch would ruin gene silencing [20], while we can show the evidence in cultured human cells with eleven complementary matches out of 19 nucleotides of a siRNA is enough causing silencing [36]. This shows siRNAs may interact with limited sequence similarity and mentions us to give necessary consideration on siRNA specificity in design algorithms. Qiu et al. [11] have examined the effects of siRNA lengths on off-target error rates.

2.3 Gene Family Knockdown

At present, the focus of RNAi research has been put on single gene knockdown, which refers to genetically modifying an organism with intention to reduce expression of single or multiple genes in their chromosomes by inserting a reagent with a sequence complementary to an active gene or its mRNA transcripts, the most typical represent is a short DNA or RNA oligonucleotide. The impacts of gene knockdown will permanently modify the chromosomal DNA to produce a "knockdown organism" or a temporary change in gene expression without modification of the chromosomal DNA molecules to knock down the function of a single gene. In this thesis, we are trying to solve the problem which is using a minimal number of siRNAs to knockdown a gene family. The reason is the efficacy of a specific siRNA in knocking down its targeted gene is

determined by its homology to that gene. Further more, the synthesis of individual siRNAs may cost large amount of money, so using compact sets of siRNAs for gene family knockdown would bring more efficiency and profit.

Because of its generally high specificity to a single target mRNA, RNAi has so far been primarily used to target and knock down the expression of individual genes in isolation. Often, however, it is useful to be able to knock down multiple genes simultaneously. For example, a family of closely related genes may have mutually redundant function; to observe any phenotypic change, it may be necessary to suppress the entire family simultaneously. For single gene knockdown, it usually suffices to select a substring of the target mRNA as the initiator siRNA. For families of genes, however, it is less clear how to design an optimal set of siRNAs to target the entire family.

2.4 Problem Statement

Given an siRNA covering problem instance {S, G, W}, we want to find the minimum set of siRNAs required to knock down all the target genes. We define the problem as follows:

2.4.1 Minimal siRNA Selection Problem

Given a siRNA set, $S = \{s_1,...,s_N\}$, and a gene set, $G = \{g_1,...g_K\}$, a N×K matrix $W=[w_{ij}]$ is generated such that $w_{ij}= 1$ if s_j cover g_i, otherwise $w_{ij}= 0$. By doing this, we can transfer the minimal siRNA set cover problem to simple set covering problem. Table 2.2 shows an example of a matrix with the number of siRNAs N=7 and the number of genes K=6. First, we generate this matrix from the original sequences of siRNAs. For example, g_1 and g_3

have the same siRNA sequences: s_1=CACUCUACUGCAGCAAAGC; g_2, g_3 and g_6 have the same siRNA sequences: s_2=GUGGGAGCGCGUGAUGAAC. Then for the first column: $w_{11}=1$, $w_{31}=1$, and $w_{i1}=0$ for other elements; for the second column: $w_{22}=1$, $w_{32}=1$, $w_{62}=1$, and $w_{i2}=0$ for other elements. With the off target effect genes: g_4, g_5 and g_6, we should not select column 2, 4 and 5, because those genes include s_2, s_4 and s_5. Table 2.3 shows the matrix without off target effects. In this thesis, we select the off target genes randomly.

		s_1	**s_2**	s_3	**s_4**	**s_5**	s_6	s_7
On target	g_1	1	**0**	0	**0**	**1**	1	0
	g_2	0	**1**	1	**0**	**1**	0	0
	g_3	1	**1**	1	**1**	**1**	0	1
Off target	g_4	0	**0**	0	**0**	**1**	0	0
	g_5	0	**0**	0	**1**	**0**	0	0
	g_6	0	**1**	0	**0**	**0**	0	0

Table 2.2 Example of a matrix with off target effects.

		s_1	s_3	s_6	s_7
On target	g_1	1	0	1	0
	g_2	0	1	0	0
	g_3	1	1	0	1

Table 1.3 Example of a matrix without off target effects.

Given a matrix *W*, the objective of the minimal siRNA set cover problem is to find a minimal set of siRNAs that can cover all the target genes without covering any off target

genes. In Table 2.2, for instance, $\{s_3, s_6\}$ is an optimal solution, while the solution $\{s_1, s_3, s_7\}$ is not, and therefore it is not cost effective.

The definition for minimal siRNA set cover problem is that, given a N×K matrix W with a siRNA set, $S= \{s_1,...,s_N\}$ and a gene set $G= \{g_1,...g_K\}$, the goal of the minimal siRNA set cover problem is to select a subset $S_{min} \subseteq S$ of siRNAs such that 1) S_{min} is minimal, and 2) S_{min} covers all the target genes without hitting any off target genes. In [9], this was proved to be an NP-hard problem by performing a reduction from the set covering problem.

This problem can be formulated as an integer linear programming (ILP) problem as follows:

$$\text{Minimize: } \sum_{j=1}^{N} x_j \qquad (1)$$

$$\text{Subject to: } \sum_{j=1}^{N} w_{ij}x_j \geq 1 \quad \text{i=1,...,K} \qquad (2)$$

$$x_j \in \{0,1\} \quad \text{j=1,...,N} \qquad (3)$$

Variables x_j=1 when siRNA j is selected, otherwise x_j=0.

In this thesis, we solve the above ILP problem by using three deterministic greedy heuristics and a genetic algorithm.

Chapter 3

3. Overview of Existing Techniques

This chapter gives an introduction for set cover problem. There is plenty of research has already done on set cover problem. A number of existing exact and approximate methods for it, including greedy heuristics [39, 40], Lagrangean relaxation [41], exact branch & bound, probabilistic greedy [9] and genetic algorithms [42, 43]. This section is followed by the discussion of these methods.

3.1 Chvatal Heuristic

This idea is to Let A be a binary matrix of size m X n, let c^T be a positive row vector of length n and let e be the column vector, all of whose m components are ones. The set-covering problem is to minimize $c^T x$ subject to $Ax \geqslant e$ and x binary. Then compare the value of the objective function at a feasible solution found by a simple greedy heuristic to the true optimum. It turns out that the ratio between the two grows at most logarithmically in the largest column sum of A.

3.2 Lagrangean relaxation

In Beasley [41], the author consider the set covering problem (SCP) which is the problem of covering the rows of a m-row, n-column, zero-one matrix(a_{ij}) by a subset of the columns at minimum cost. Formally, the problem can be defined as follows:

Let

$x_j = 1$ if column j (cost c_j) is in the solution,

$x_j = 0$ otherwise, then the program is:

minimise $\sum_{j=1}^{n} c_j x_j$ (4)

subject to $\sum_{j=1}^{n} a_{ij}x_j \geq 1$, i=1,...,m, (5)

$x_j \in (0,1)$, j=1,...,n. (6)

Equation (4) ensures that each row is covered by at least one column and equation (6) is the integrality constraint. The author presents a three stage algorithm consisting of:

(I) A dual ascent procedure:

Letting u_i (≥ 0, i=1, ···, m) be the dual variables associated with equation (5) then the dual of the linear programming relaxation of the SCP (DLSCP) is given by

minimise $\sum_{i=1}^{m} u_i$ (7)

subject to $\sum_{i=1}^{m} a_{ij}u_i \geq c_j$, i =1,...,n, (8)

$u_i \geq 0$, i =1,...,m. (9)

The author adopted a two pass procedure to the problem of generating a good lower bound for the SCP from a feasible solution for DLSCP. This procedure was modelled on the dual ascent procedure of Balas and Ho [44]. At the end of this procedure, there will be a set of dual variables feasible for DLSCP and a corresponding lower bound as given by equation (7).

(II) A sub gradient procedure starting from an initial set of lagrangean multipliers equal to the dual variables from stage (I):

(III) Solving the dual of the linear programming relaxation of the SCP:

(a) Generate an upper bound,

(b) Use the dual ascent procedure to calculate an initial set of lagrange multipliers,

(c) Use the subgradient procedure in an attempt to improve upon the bound derived from the dual ascent procedure,

(d) At the end of the subgradient procedure optimally solve DLSCP using a simplex algorithm.

Note here that the third stage is achievable computationally because a comprehensive set of problem reduction tests are used to remove some rows, and a large number of columns, from the problem.

3.3 Exact Branch and Bound Algorithm

The minimal siRNA cover problem can be solved using branch-and bound techniques. A search tree is constructed by iteratively picking an siRNA and branching on it. At each point the algorithm generates two subtrees with one corresponding to selecting the siRNA and the other to de-selecting the siRNA. When an siRNA is selected, deduction techniques are used to reduce the search space. During the search, the algorithm keeps track of the current best cover with the lowest number of siRNAs.

The Algorithm below shows the pseudocode for the exact, branch-and-bound process. S is the set of unselected siRNAs, and M is the set of uncovered target genes. The argument rank contains ranks for all the siRNAs. x is the current selection of siRNAs and b is the best selection found so far. The algorithm starts with (S, M, rank, 0, 1), where and represent vectors with all 0s and all 1s, respectively.

In step 4, we use Lemma 3 [9], which is proposed by Zhao et al. 2004, allows us to prune branches which cannot lead to any covers better than the known best one.

Branch and Bound Algorithm:

1. Branch_and_Bound (S, M, rank, x, b) {

2. Reduce S/M and update x

3. Sort the rest of the siRNAs in non-increasing order of uncovered genes they can cover

4. Compute the lower bound on the current path based on Lemma 3 [9]

5. if (the lower bound $>$ | b | or the low bound $>$*Number_From_Greedy*) {return b}

6. if (| M | = 0) {return x}

7. s1 = first (S)

8. y = Branch_and_Bound (S – {s1}, M – g(s1), rank, x, b)

9. if (| y | $<$ | b | or (| y | = | b | and avgRank(y, rank) $>$ avgRank(b, rank))= {b=y}

10. y = Branch_and_Bound (S – {s1}, M , rank, x, b)

11. if (| y | $<$ | b | or (| y | = | b | and avgRank(y, rank) $>$ avgRank(b, rank))= {b=y}

12. return b

13. }

During the reduction stage, the algorithm discards any dominated siRNA, which covers only a subset of genes also covered by another siRNA. An essential siRNA is an siRNA which is the only one that can cover a particular target gene. We add all essential siRNAs to the solution x since they must be in any complete cover. We update the uncovered genes M accordingly.

After reduction is done, the potential lower bound on the current path is estimated based on Lemma 3 [9]. If the current estimate requires more siRNAs than the current best cover

b, then we prune the current path. Otherwise, it generates two branches, one for selecting the next siRNA and the other for de-selecting the same siRNA. Simultaneously, the algorithm uses as the upper bound the number of siRNA in the cover obtained from our greedy algorithm to prune some paths since the minimal cover must be no worse than the one obtained from any greedy algorithm.

3.4 Probabilistic Greedy Algorithm

The probabilistic greedy algorithm [9] shares some common aspects with the randomized greedy algorithm [45], but it differs in an important way. The randomized greedy algorithm [45] uses a uniform probability distribution to select the next candidate to the cover, while this approach selects the next siRNA randomly according to a domain-specific non-uniform probability distribution. Let f(s) be the number of genes that an siRNAs covers, and h(g) be the number of siRNAs which cover a particular gene g. Based on the two features: f(s) and $\min_{g \in g(s)} h(g)$, the author define a selection metric for selecting siRNAs, and then construct a subset of the unselected siRNAs as the set of potential siRNAs based on the selection metric and limiting parameter a$\in$[0, 1]. They compute a selection probabilistic distribution over the potential siRNA set.

The selection metric for siRNA, s, found to be the most effective is:

$$m(s) = \left\lceil \frac{f^2(s)}{\min_{g \in g(s)} h(g)} \right\rceil$$

The selection probability distribution over the set of potential siRNAs S' is defined as:

$$P(s) = \left\lceil \frac{m(s)}{\sum_{s \in S'} m(s)} \right\rceil$$

The Algorithm below shows the pseudocode for the probabilistic greedy algorithm. S is a set of siRNAs, and M is the set of the target genes. The argument rank contains ranks for all the siRNAs:

```
1.  ProbGreedy (S, M, rank) {
2.  α = aValue        //α∈[0, 1]
3.  k = aValue        //Number of repetition
4.  opt = S
5.  iter = 0
6.  while (iter < k= {
7.     C = ø
8.     while (M ≠ø) {
9.    MAX = 0
10.   for each  s ∈S {
11.      Compute metric m(s)
12.      if ( MAX < m(s) ) MAX = m(s)
13.   }
14.   S'= ø
15.    for each  s ∈S
16.    if ( m(s) >α* MAX) S'= S' ∪ {s}
17.   Compute probability distribution  P(s) over S'
18.   Select s randomly from S' according to P(s)
19.   C = C  ∪  {s}
20.   S = S – {s}
```

21. M = M – g(s)

22. }

23. Remove redundant siRNAs from C

24. if (| C | < | opt | or (| C | = | opt | and avgRank(C, rank) > avgRank(opt, rank))

25. opt = C

26. }

27. return C

28. }

This procedure is repeated k times from line 6 to 27, and returns the best siRNA cover it finds. Whenever there is a tie between two siRNA covers, the algorithm uses their average ranks as a tie-break, and chooses the one with the higher average rank. Since there are up to min(| M | , | S |) siRNAs in an siRNA cover, thus, the inner while loop could have up to min(| M | , | S |) iterations, where each iteration requires O(S) time. Therefore, the algorithm requires O(k* | S | * min(| M | , | S |)) time. The probabilistic greedy algorithm reduces to standard greedy algorithm, when α=1.0 and k=1.

3.5 Genetic Algorithm

This Genetic algorithm [42] made some modifications to the basic genetic procedures including a new fitness-based crossover operator, a variable mutation rate and a heuristic feasibility operator accommodated specifically for the set cover problem.

A genetic algorithm (GA) can be understood as an "intelligent" probabilistic search algorithm which can be applied to a variety of combinational optimisation problems [45].

The theoretical foundations of GAs were originally developed by Holland [46]. The idea of GAs is based on the evolutionary process of biological organism in nature. During the course of the evolution, natural populations evolve according to the principles of natural selection and "survival of the fittest". Individuals which are more successful in adapting to their environment will have a better chance of surviving and reproducing, whilst individuals which are less fit will be eliminated. This means that the genes from the highly fit individuals will spread to an increasing number of individuals in each successive generation. The combination of good characteristics from highly adapted ancestors may produce even more fit offspring. In this way, species evolve to become more and more well adapted to their environment.

A GA simulates these processes by taking an initial population of individuals and applying genetic operators in each reproduction. In optimisation terms, each individual in the population is encoded into a string or chromosome which represents a possible solution to a given problem. The fitness of an individual is evaluated with respect to a given object function. Highly fit individuals or solutions are given opportunities to reproduce by exchanging pieces of their genetic information, in a crossover procedure, with other highly fit individuals. This produces new "offspring" solutions, which share some characteristics taken from both parents. Mutation is often applied after crossover by altering some genes in the strings. The offspring can either replace the whole population (generational approach) or replace less fit individuals (steady-state approach). This evaluation-selection-reproduction cycle is repeated until a satisfactory solution is found. The basic steps of a simple GA are shown below:

Generate an initial population;

Evaluate fitness of individuals in the population;

Repeat

Select parents from the population;

Recombine (mate) parents to produce children;

Evaluate fitness of the children;

Replace some or all of the population by the children;

Until a satisfactory solution has been found.

Chapter 4

4. Methodologies

It is well known that heuristic method is extremely important for the present and future developments of bioinformatics, since it can provide key solutions for the new challenges posed by the progressive transformation of biology into data analysis. There are four heuristic methods are presented in this paper to solve the minimal siRNA set cover problem.

4.1 Dominated Target Covering Heuristic (DTC)

To select a minimal number of siRNAs S_{min}-covering each target gene, DTC uses a function to evaluate each individual siRNA. Given a matrix W which is determined by a siRNA set $S=\{s_1, \cdots s_N\}$ and a gene set $G=\{g_1, \cdots g_K\}$, we define the cover function cov as follows:

$$\mathrm{cov}(s_j, g_i) = w_{ij} \times \frac{1}{|S_{g_i}|} \quad s_j \in S_{g_i},\ g_i \in G \tag{10}$$

where $0 \leqslant cov(s_j, g_i) \leqslant 1$ and S_{gi} (S_{gi} denotes the number of siRNAs generated by each gene, so it is impossible to be empty)is the set of siRNAs related to gene g_i. The value of

$cov(s_j,g_i)$ is considered as a ratio by which s_j contributes to the satisfaction of coverage constraint for gene g_i.

Since the minimal number of siRNAs is to be selected, it is suitable to take into consideration each siRNA with regard to its capability of satisfying coverage constraints. After applied Equation (10), the coverage is calculated as:

$$C(s_j) = \max\{\text{cov}(s_j, g_i) | 1 \le i \le k\} \quad g_i \in Gs_j \tag{11}$$

where Gs_j is the set of genes covered by s_j, $C(s_j)$ is the maximum contribution made by S_j according to each gene. This is illustrated in Table 2.3, which derives from Table 2.2.

	s_1	s_3	s_6	s_7
g_1	1/2	0	1/2	0
g_2	**0**	**1**	**0**	**0**
g_3	1/3	1/3	0	1/3
C	1/2	**1**	1/2	1/3

Table 4.1 Example of a coverage function table.

When $C(s_j) = 1$, we consider s_j as an essential siRNA since any feasible solution has to include it. In Table 4.1, it is obvious that s_3 is an essential siRNA.

This heuristic consists of three phases: initialization, construction and reduction. Initially, we calculate $C(s)$ for each siRNA $s \in S$ from the given matrix W. Then an initial non-feasible solution S_{ini} is created, which only contains essential siRNAs. We denote S as the set s_j, S_{sol} is the subset of S which contains selected s_j in the next phase. In the construction phase, we always select the high-ratio siRNAs s_j into S_{ini} by sorting $S \backslash S_{sol}$ in descending order of $C(s)$. Note that, when we select a $s_j \in S \backslash S_{sol}$ that covers g_i, we delete s_j from matrix W, and then we compute $C(s)$ from the reduced matrix W'. This step executes repeatedly until we get an initial feasible solution. In the reduction phase, S_{sol} is reduced

by repeatedly removing low-ratio siRNAs to achieve a feasible but near optimal solution S_{min} which is selected to cover all the target genes.

More precisely, the steps of the heuristic can be described as follows:

1. Initialization Phase

a) compute $C(s)$ for all $s \in S$

b) $S_{ini} = \{s \in S \mid C(s)=1\}$ {essential siRNAs in initial solution}

2. Construction Phase

c) $S_{sol} = S_{ini}$

d) sort $S \backslash S_{sol}$ in descending order of $C(s)$

e) for each gene g_i not covered by S_{sol}

$S_{sol} = S_{sol} \cup s_j$ {next highest-ratio $s_j \in S \backslash S_{sol}$ that covers g_i }

f) delete s_j from matrix W;

g) repeat step a) to step f)

3. Reduction Phase

h) $S_{min} = S_{sol}$

i) $W=W \mid S_{min}$ /*the restriction of matrix W to the siRNAs in S_{min} */

j) compute $C(s)$ for all $s \in S_{min}$

k) sort $S_{del} = \{s \in S_{min} \mid C(s)<1\}$ in ascending order of $C(s)$

l) if $S_{min} \backslash \{s\}$ is feasible for each $s \in S_{del}$ then

$S_{min} = S_{min} \backslash \{s\}$

m) return S_{min}

In the worst-case, the time complexity of DTC is dominated by the construction phase. The time complexity of the construction phase is dominated by sorting in step d) and the search for siRNAs to cover some genes in step e). Sorting the siRNAs in e) takes O (n logn). In step e), there are potentially K genes that are not yet covered by S_{sol}. Testing a gene for coverage takes at most n steps. Searching for the highest-degree siRNA s_j to cover a gene gi takes at most n steps. Therefore, step e) takes O (nk). The construction phase iterates n times at most (see step g)). Therefore, the construction phase runs in $O(n^2k) + O(n^2 \log n)$.

4.2 Dominant siRNA Selection Heuristic (DSS)

We also want to satisfy the selection of dominant siRNAs; s_j dominates s_1 if $Gs_1 \subset Gs_j$. In Table 2.3, for example, s_1 dominates s_6 since $Gs_6=\{g_1\} \subset Gs_1=\{g_1,g_3\}$. Selecting dominant siRNAs instead of dominated siRNAs covers more genes. In the example, however, we have $C(s_1)=C(s_6)$, and hence DTC will select s_1 for gene coverage rather than s_6 which depends on the particular order of the siRNAs. This is because DTC will select a dominant siRNA s_j over its dominated siRNA s_1 only if $C(s_j)>C(s_1)$. In Table 2.3, s_6 dominates s_7 and $C(s_6)>C(s_7)$, therefore s_6 will be selected first.

To satisfy the selection of a dominant siRNA that has the same degree as some of its dominated siRNAs, we modify Equation (10) in such a way that a dominant siRNA s_j will have a higher *C(s)* value than its dominated siRNAs. We solve this by adding a penalizing each entry in Table 4.1 with an amount that takes into account the number of coverd genes. The new *cov* function has the form as follows:

$$\mathrm{cov}(s_j, g_i) = w_{ij} \times \frac{1}{|Sg_i|} \times \frac{1}{m-|Gs_j|+1} \quad s_j \in S_{g_i}, \quad g_i \in G$$

(12)

where $0 \leqslant cov(sj,gi) \leqslant 1$, S_{gi} is the set of siRNAs related to gene gi, G_{sj} in the penalty term is the set of genes covered by s_j and m is the number of genes. In Equation (12), siRNAs that cover fewer genes are penalized more than those that cover more genes.

Dominant siRNA Selection (DSS) heuristic is similar to DTC heuristic described in Section 4.1 only with the exception that function C is defined by using Equation (12) instead of Equation (10). In DSS, siRNAs that cover dominated genes are selected first, as in DTC. Unlike DTC, dominant siRNAs among all such siRNAs will be selected first. These two greedy principles together allow a larger coverage at each selection step. So DSS is greedier than DTC.

4.3 Genetic Algorithm for Minimal siRNA Set Cover Problem

Beasley et al. [41] presented a genetic algorithm-based heuristic for set covering problem. Based on this method, our GA inherits the siRNA selection function defined in Section 4.2. The improved genetic approach can be illustrated in details as follows.

4.3.1 Representation and Fitness Function

To design a genetic algorithm, we have to devise a suitable representation scheme at first. Given the initial candidate siRNA set $S = \{s_1,...,s_N\}$, we want to find a feasible subset $S_{min} \subseteq S$ of minimal cardinality. Therefore, the search space is the power set of S, denoted by 2^S; that is the set of all subsets of S. The fitness of an individual s is related to its objective value, which corresponds to the number of siRNAs in its associated subset. So the fitness function is:

$$f_i = \sum_{j=1}^{N} s_{ij} \qquad (13)$$

where s_{ij} is the value of the j-th bit (column) in the string (row) corresponding to the i-th individual.

4.3.2 Parent Selection Operator

For the purpose of selecting the fittest individuals continuously, we apply a binary tournament selection which selects the best individual in any tournament. The chosen individual will be removed from the population, otherwise individuals can be selected more than once for the next generation.

4.3.3 Crossover Operator

We implement the *fusion operator* of [41] which regards both the structure and the relative fitness of each parent solution, and produces a single child only. This crossover focuses on the differences of the parents. So it will generate new solutions more efficiently when they have similar parents. Besides, the fittest parent will obtain more probability to contribute the fitness of the child. Let f_{p1}^s and f_{p2}^s be the scaled fitness values of the parents P_1 and P_2 respectively, and let C denote the child solution, then for each j $\in$ [1, N]:

1. IF $P_{1j} = P_{2j}$, THEN $C_j = P_{1j} = P_{2j}$;
2. ELSE
   (1) $C_j = P_{1j}$ with probability $p = \dfrac{f_{p2}^s}{f_{p1}^s + f_{p2}^s} s$
   (2) $C_j = P_{2j}$ with probability 1 - p

4.3.4 Mutation Operator and Variable Mutation Rate

In the next step of crossover, we use the mutation operator to change a number of bit positions randomly. The number of positions to mutate for a given solution depends on the mutation rate. We use the variable mutation rate in [41]. It essentially depends on the rate of the GA convergence which means lower mutation rates are used in early generations. When mutation increases to higher rates, the population converges, after that mutation stabilizes to a constant rate. The mutation schedule below specifies the number of bits to mutate [41].

$$Num_{mut} = \left\lceil \frac{m_f}{1+\exp\left(-4m_g\left(t-m_c\right)/m_f\right)} \right\rceil \qquad \textbf{(14)}$$

where t is the number of child solutions that have already been generated, m_f specifies the final stable mutation rate, m_c is the number of solutions that should be generated such that the mutation rate is $\frac{m_f}{2}$, and m_g specifies the gradient at $t = m_c$. The value of m_f is user-defined and the values of m_c and m_g are problem-dependent parameters.

4.3.5 Heuristic Feasibility Operator

Crossover and mutation operators can generate unfeasible solutions. Hence, we propose a heuristic feasibility operator that keeps the feasibility of solutions in the population. More over, the operator provides a local optimization method for fine-tuning the results generated from crossover and mutation operators. This operator consists of the last two phases of DSS heuristic: construction and reduction phases. GA has already generated a potentially good solution S_{sol} so we do not need to apply the initialization phase for this step. The construction phase starts with such a solution S_{sol} which is not a feasible

solution generated by GA. The feasibility operator is applied for unfeasible solutions only.

4.3.6 The Algorithm

This Genetic Algorithm can be summarized as follows:

1) Generate an initial population of N solutions. Set t:=0.

2) Select two solutions S_1 and S_2 from the population using binary tournament selection.

3) Produce a new solution C using the fusion crossover operator.

4) Mutate Num_{mut} randomly selected bits in C.

5) Make C feasible and remove redundant columns in C by using DSS heuristic operator.

6) If C is identical to any one of the solutions in the population, go to step 2; otherwise, set t:=t+1 and go to step 7.

7) Replace a randomly selected solution with an above average fitness in the population by C.

8) Repeat steps 2-7 until $t=P_s$. (t is the number of child solutions that have already been generated, P_s is the population size which is a user defined parameter).

4.4 Forward Selection Heuristic

Forward Selection begins from an empty set of features. It first evaluates all one-feature subsets and selects the one with the best performance. Then evaluates all two-feature subsets that include the feature already selected in the first step and selects the best one. This process will continue until extending the size of the current subset leads to a lower performance. The steps of forward selection heuristic are shown as follows in detail:

1) Use Equation (12) to select a s_j with the best value (the highest value of $C(s)$). For instance, s_1 is selected.

2) From all possible two-dimensional vectors that contain s_j form the fist step, that is, $[s_1, s_2]^T$, $[s_1, s_3]^T$, $[s_1, s_4]^T \ldots [s_1, s_j]^T$, compute the criterion value for each of them and select the best one, give an illustration: $[s_1, s_4]^T$.

3) Form all three-dimensional vectors generated from the two-dimensional winner ($[s_1, s_4]^T$), that is, $[s_1, s_4, s_2]^T$, $[s_1, s_4, s_3]^T$, $[s_1, s_4, s_5]^T \ldots [s_1, s_4, s_j]^T$ and select the best one.

4) Continue this procedure, until find a subset of S which can cover all the target genes with the minimal number of s_j.

5) In case that S may include redundant siRNAs, the last phase of DSS: reduction phase will be used in this step.

The time complexity for step 1) is O (n^2k), from step 2) to step 4) the time complexity is O (n). While, the reduction phase of DSS is implemented in step 5). So the total time complexity for forward selection is O ($n^2k+n+n\log n$).

Chapter 5

5. Computational Experiment Results

The experiments were conducted to find minimal siRNA sequences to cover all the target genes. As a set cover problem, we have a matrix and the goal is to select minimal columns to cover all the rows. We tested our heuristic methods on three gene families and an artificial matrix. During the experiments, each method was carried out consistently in the same environment using the same dataset.

5.1 Experiment Environment

We implemented all approaches, and experimental results show that our heuristic approaches are good alternatives for the minimal siRNA set cover selection problem heuristics: exact branch and bound algorithm, probabilistic greedy algorithm [9]. All heuristics were implemented by java programming language.

The hardware platform is as follows:

- A workstation with Intel(R) Xeon(TM) CPUs 3.20GHz and 3.19GHz
- 8.00GB of RAM
- The operating system is Microsoft Windows XP, Professional x64 Edition.

5.2 Dataset

In this thesis, we apply our methods to three gene families. The first family, the set of Fibrinogen-related protein (FREP) genes from the snail Biomphalaria glabrata, is of interest in human immunological studies because both humans and B.glabrata may become infected by the parasite Schistosoma mansoni [9]. The second family is also a set of FREP genes like family 1[10]. And for the third family, we downloaded the olfactory genes of nematode Caenorhabditis elegans from NCBI [12]. Fibrinogen-related proteins (FREPs) are in the hemolymph of the freshwater gastropod Biomphalaria glabrata. They are produced in hemocytes. Some categories of FREPs are modulated following infection with parasites such as the digenetic trematode Echinostoma paraensei. Some FREPs are capable of binding to parasite surfaces and can precipitate soluble parasite antigens, prompting hypothesis that they take into effect in internal defense [17]. The defense responses of B.glabrata are a relational concern since this snail is one of the most

important intermediate hosts for another digenetic trematode, Schistosoma mansoni, a parasite which infects about 83 million people [18]. Studying of molecules or genes involved in snail response to trematode infection will be very helpful for understanding the underlying mechanisms of the snail host and parasite interaction.

The actual target gene families used in our experiments are:

- Target family 1: 13 FREP genes from Zhao et al. [9].
- Target family 2: 53 fibrinogen (FBG) genes from the FREP family in Zhang et al. [10].
- Target family 3: 150 olfactory genes from NCBI [12].

5.3 Experiment Results

We design the siRNA sequences for above 3 gene families by three softwares [28, 29, 30].

5.3.1 Experiments on using BLOCK-iT™ RNAi Designer

Table 5.1, 5.2 and 5.3 show the number of siRNAs used for covering target genes. In these tables, G is the number of target genes, S is the number of siRNA sequences without off target gene effects. (Some abbreviations are used: PG=Probabilistic Greedy, BB=Branch & Bound, DTC=Dominated Target Covering, DSS=Dominant siRNA Selection, GA_DSS=Genetic Algorithm with Dominant siRNA Selection, FS=Forward Selection).

size of target set	G=2 S=22	G=3 S=27	G=4 S=34	G=5 S=30	G=6 S=36	G=7 S=45	G=8 S=51	G=9 S=65	G=10 S=75	G=11 S=81	G=12 S=84	G=13 S=96
PG	2	3	4	5	5	6	7	9	10	13	14	15
BB	2	3	4	5	5	6	7	9	10	13	14	15

DTC	2	3	4	5	5	6	7	7	8	9	10	11
DSS	2	3	4	5	5	6	7	7	8	8	9	10
GA_DSS	2	3	4	5	5	6	7	7	8	8	9	10
FS	2	3	4	5	5	6	7	7	8	8	9	10

Table 5.1 Results for Target Family 1 (BLOCK-iT™ RNAi Designer).

size of target set	**G=5 S=40**	**G=10 S=79**	**G=20 S=137**	**G=30 S=170**	**G=40 S=201**	**G=50 S=254**	**G=53 S=277**
PG	5	11	28	36	51	33	30
BB	5	11	28	37	51	34	30
DTC	5	10	19	21	26	20	18
DSS	5	9	19	20	25	19	17
GA_DSS	5	9	19	20	25	19	17
FS	5	9	19	20	25	19	17

Table 2.2 Results for Target Family 2 (BLOCK-iT™ RNAi Designer).

size of target set	**G=15 S=144**	**G=30 S=273**	**G=45 S=423**	**G=60 S=567**	**G=75 S=711**	**G=90 S=860**	**G=105 S=991**	**G=120 S=1097**	**G=135 S=1202**	**G=150 S=1339**
PG	16	33	48	63	79	95	110	122	120	153
BB	16	33	48	63	80	96	112	123	121	154
DTC	16	27	42	57	71	86	99	110	121	136
DSS	14	26	41	56	69	84	97	108	118	132
GA_DSS	14	26	41	56	69	84	97	108	118	132
FS	14	26	41	56	69	84	97	108	118	132

Table 5.3 Results for Target Family 3 (BLOCK-iT™ RNAi Designer).

5.3.2 Experiments on using siDirect

Table 5.4, 5.5 and 5.6 show the number of siRNAs used for covering target genes:

size of target set	G=2 S=2 2	G=3 S=2 7	G=4 S=3 5	G=5 S=3 7	G=6 S=4 4	G=7 S=5 5	G=8 S=6 3	G=9 S=7 2	G=1 0 S=8 4	G=1 1 S=9 5	G=12 S=10 3	G=13 S=11 0
PG	2	3	4	6	9	10	11	12	13	14	15	16
BB	2	3	4	6	9	10	11	12	13	14	15	16
DTC	2	3	4	5	6	8	9	9	10	11	11	12
DSS	2	3	4	5	5	6	7	9	10	11	11	12
GA_ DSS	2	3	4	5	5	6	7	9	10	11	11	12
FS	2	3	4	5	5	6	7	9	10	11	11	12

Table 5.4 Results for Target Family 1 (siDirect).

size of target set	G=5 S=50	G=10 S=93	G=20 S=15 8	G=30 S=19 4	G=40 S=22 5	G=50 S=27 3	G=53 S=29 9
PG	5	16	31	48	69	45	42
BB	5	16	31	48	69	45	42
DTC	5	13	19	30	37	30	25
DSS	5	11	19	29	36	29	24
GA_D SS	5	11	19	29	36	29	24
FS	5	11	19	29	36	29	24

Table 5.5 Results for Target Family 2 (siDirect).

size of targ et set	G=1 5 S=1 58	G=3 0 S=2 89	G=4 5 S=4 42	G=6 0 S=5 83	G=7 5 S=7 33	G=9 0 S=8 84	G=1 05 S=10 12	G=1 20 S=11 30	G=1 35 S=12 27	G=1 50 S=13 62
PG	21	40	58	73	90	109	125	138	143	175
BB	21	40	58	73	90	109	125	137	142	175
DT C	18	33	49	64	82	99	114	129	142	158
DS S	18	32	48	63	81	98	113	128	133	140
GA _D SS	18	32	48	63	81	98	113	128	133	140
FS	18	32	48	63	81	98	113	128	133	140

Table 5.6 Results for Target Family 3 (siDirect).

5.3.3 Experiments on using TROD (T7 RNAi Oligo Designer)

Table 5.7, 5.8 and 5.9 show the number of siRNAs used for covering target genes:

size of target set	G=2 S=22	G=3 S=27	G=4 S=33	G=5 S=35	G=6 S=41	G=7 S=50	G=8 S=59	G=9 S=69	G=10 S=79	G=11 S=89	G=12 S=98	G=13 S=106
PG	2	3	4	7	8	9	10	11	12	13	15	16
BB	2	3	4	7	8	9	10	11	12	13	15	16
DTC	2	3	4	5	6	7	8	8	9	10	10	11
DSS	2	3	4	4	5	6	7	8	9	10	10	11
GA_DSS	2	3	4	4	5	6	7	8	9	10	10	11
FS	2	3	4	4	5	6	7	8	9	10	10	11

Table 5.7 Results for Target Family 1 (TROD).

size of target set	G=5 S=49	G=10 S=92	G=20 S=153	G=30 S=185	G=40 S=220	G=50 S=271	G=53 S=297
PG	5	14	29	44	62	38	35
BB	5	14	29	44	62	37	35
DTC	5	12	17	28	32	26	21
DSS	5	10	17	27	31	25	20
GA_DSS	5	10	17	27	31	25	20
FS	5	10	17	27	31	25	20

Table 5.8 Results for Target Family 2 (TROD).

size of target set	G=15 S=152	G=30 S=282	G=45 S=433	G=60 S=575	G=75 S=720	G=90 S=872	G=105 S=1001	G=120 S=1118	G=135 S=1215	G=150 S=1350
PG	19	39	54	68	85	104	119	130	132	162
BB	19	39	54	68	84	104	119	129	131	162
DTC	16	32	46	60	76	96	109	120	131	143
DSS	16	31	45	59	75	94	108	118	127	138

GA_ DSS	16	31	45	59	75	94	108	118	127	138
FS	16	31	45	59	75	94	108	118	127	138

Table 5.9 Results for Target Family 3 (TROD).

Table 5.10 shows the result generated by an artificial big matrix.

size of target set	G=255 S=2792
PG	63
BB	63
DTC	15
DSS	13
GA_DSS	13
FS	13

Table 5.10 Results for an Artificial Matrix.

5.3.4 Summary of Experiments on Different Designed Sequence and Results

Experiment results show that our heuristics are able to select less number of siRNAs than the methods mentioned in [9]. Since the three siRNA software designers use different algorithms, we got different siRNA sequences from each designer. Therefore, the numbers of the siRNA sequences are all different. When the number of siRNA increases, DSS, GA_DSS and FS give much better results than other methods. From the artificial dataset, we can clearly notice that the contrast between DSS, GA_DSS, FS and PG, BB is extremely remarkable. It can be expected that this will provide a great help for RNAi interference experiments.

Chapter 6

6. Conclusion

6.1 Conclusions

In this thesis, we discussed some heuristic approaches for the minimal siRNA set cover problem which is important to gene family knockdown. We introduced a novel heuristic method: forward selection for set covering problem and we also implemented other three improved methods.

Experiment results showed that these methods are able to obtain relative minimal solutions which are still comparable to the known heuristics [9] for this problem. Hence, our methods can significantly reduce the number of siRNAs required in gene family knockdown experiments as compared to knocking down genes one by one.

The experiments showed that it is a nice application of set cover heuristics to a recent biological problem. The modifications made to suit the problem were well thought out and the novel forward selection heuristic was promising.

6.2 Future Work

There are some directions for further work. We summarize as follows:

- Applying some other siRNA sequence design software to compare the output sequences which are generated by these software.
- Implementing siRNA sequences from biological experiments instead of using siRNA sequence design software.

· Experiments on more gene families.

· Designing a more evolutionary heuristic method.

Reference:

1. Tuschl, T.: RNA interference and small interfering RNAs. Chembiochem 2 (4), 239--245 (2001)

2. Hannon, G.J.: RNA interference. Nature vol. 418, pp. 244--251 (2002)

3. Jacque, J.M., Triques, K., Stevenson, M.: Modulation of HIV-1 replication by RNA interference. Nature vol. 418, pp. 435--438 (2002)

4. Surabhi, R., Gaynor, R.: RNA interference directed against viral and cellular targets inhibits human immunodeficiency virus type 1 replication. Journal of Virology 76 (24) pp. 12963--12973 (2002)

5. Borkhardt, A.: Blocking oncogenes in malignant cells by RNA interference-new hope for a highly specific cancer treatment? Cancer Cell 2 (3) pp. 167--168 (2002)

6. Barik, S.: Development of gene-specific double-stranded rna drugs. Annals of Medicine36 (7) pp. 540--551 (2004)

7. Chi, J.T., Chang, H.Y., Wang, N.N., Chang, D.S., Dunphy, N., Brown, P.O.: Genomewide view of gene silencing by small interfering RNAs, PNAS 100 (11) pp. 6343--6346 (2003)

8. Morris, K.V.: RNA and the Regulation of Gene Expression: A Hidden Layer of Complexity. Caister Academic Press. ISBN 978-1-904455-25-7 (2008)

9. Zhao, W., Fanning, M.L., Lane, T.: Efficient RNAi-based gene family knockdown via set cover optimization. Artificial Intelligence in Medicine. Vol. 35, pp. 61--73 (2005)

10. Zhang, S.M., Loker, E.S.: Representation of an immune responsive gene family encoding fibrinogen related proteins in the freshwater mollusk biomphalaria glabrata, an intermediate host for schistosoma mansoni. Gene vol. 341, pp. 255--266 (2004)

11. Qiu, S., Adema, C.M., Lane, T.: A computational study of off-target effects of RNA interference. Nucleic Acids Research. Vol. 33(6) pp. 1834--1847 (2005)

12. National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov

13. Daneholt, Bertil. "Advanced Information: RNA interference". *The Nobel Prize in Physiology or Medicine 2006.* Retrieved on 2007-01-25.

14. Bagasra O, Prilliman KR (2004). "RNA interference: the molecular immune system". *J. Mol. Histol.* **35** (6): 545–53. doi:10.1007/s10735-004-2192-8. PMID 15614608.

15. R Agrawal and R Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

16. T Tuschl. RNA interference and small interfering RNAs. *Chembiochem*, 2(4):239–245, 2001.

17. J Martinez, A Patkaniowska, H Urlaub, R Luhrmann, and T Tuschl. Single-stranded antisense sirnas guide target rna cleavage in rnai. *Cell*, 110(5):563–574, 2002.

18. A Grishok, H Tabara, and CC Mello. Genetic requirements for inheritance of rnai in c. elegans. *Science*, 287(5462):2494–2497, 2000.

19. SM Elbashir, W Lendeckel, and T Tuschl. RNA interfernce is mediated by 21- and 20- nucleotide RNAs. *Genes and Development*, 15:188–200, 2001.

21. S Saxena, ZO Jonsson, and A Dutta. Small rnas with imperfect match to endogenous mrna repress translation. implications for off-target activity of small inhibitory rna in mammalian cells. *Journal of Biological Chemistry*, 278(45):44312–44319, 2003.

22. G Hutvagner and PD Zamore. A microrna in a multiple-turnover rnai enzyme complex. *Science*, 297:2056–2060, 2002.

23. H Shi, A Djikeng, T Mark, E Wirtz, C Tschudi, and E Ullu. Genetic interference in trypanosoma brucei by heritable and inducible double-stranded rna. *RNA*, 6(7):1069–1076, 2000.

24. MA Martinez, A Gutierrez, M Armand-Ugon, J Blanco, M Parera, J Gomez,

B Clotet, and JA Este. Suppression of chemokine receptor expression by rna interference allows for inhibition of hiv-1 replication. *AIDS*, 16(18):2385–2390, 2002.

25. H Xia, Q Mao, SL Eliason, SQ Harper, IH Martins, HT Orr, HL Paulson, L Yang, RM Kotin, and BL Davidson. Rnai suppresses polyglutamine-induced neurodegeneration in a model of spinocerebellar ataxia. *Nature Medicine*, 10:816–820, 2004.

26. J Soutschek, A Akinc, B Bramlage, K Charisse, R Constien, M Donoghue, S Elbashir, A Geick, P Hadwiger, J Harborth, MJohn, V Kesavan, G Lavine, RK Pandey, T Racie, KG Rajeev, I Rohl, I Toudjarska, G Wang, S Wuschko, D Bumcrot, V Koteliansky, S Limmer, M Manoharan, and HP Vornlocher. Therapeutic silencing of an endogenous gene by systemic administration of modified sirnas. *Nature*, 432:173–178, 2004.

27. A Borkhardt. Blocking oncogenes in malignant cells by RNA interference — new hope for a highly specific cancer treatment? *Cancer Cell*, 2(3):167–168, September 2002.

28. siRNA sequence design software: https://rnaidesigner.invitrogen.com/rnaiexpress/

29. siDirect: highly effective, target-specific siRNA design software for mammalian RNA interference. Yuki Naito, Tomoyuki Yamada, Kumiko Ui-Tei, Shinichi Morishita and Kaoru Saigo (2004). *Nucleic Acids Res.*, Vol. 32(Web Server issue): W124-W129.

30. Donzé, O. and Picard, D. (2002) RNA interference in mammalian cells using siRNAs synthesized with T7 RNA polymerase. *Nucleic Acids Research*, 30 (10): e46.

31. Elbashir SM et al. (2001) Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells. Nature. 411:494-498.

32. Reynolds A, Leake D, Boese Q, Scaringe S, Marshall WS, Khvorova A. Rational siRNA design for RNA interference. Nat Biotechnol. 2004 Mar;22(3):326-30.

33. T Holen, M Amarzguioui, MT Wiiger, E Babaie, and H Prydz. Positional effects of short interfering rnas targeting the human coagulation trigger tissue factor. *Nucleic Acids Res.*, 30(8):1757–1766, 2002.

34. BSE Heale, HS Soifer, C Bowers, and JJ Rossi. sirna target site secondary structure predictions using local stable substructures. *Nucleic Acids Research*, 33(3), 2005.

35. KQ Luo and DC Chang. The gene-silensing efficacy of siRNA is strongly dependent on the local structure of mRNA at the targeted region. *Biochemical and Biophysical Research Communications*, 318:303–310, 2004.

36. AL Jackson, SR Bartz, J Schelter, SV Kobayashi, J Burchard,MMao, B Li, G Cavet, and PS Linsley. Expression profiling reveals off-target gene regulation by rnai. *Nature Biotechnology*, 21:635–637, 2003.

37. Summerton, J (2007). "Morpholino, siRNA, and S-DNA Compared: Impact of Structure and Mechanism of Action on Off-Target Effects and Sequence Specificity" (Pubmed). *Med Chem.* **7** (7): 651–660.

38. Nasevicius, A; Ekker SC (2000). "Effective targeted gene 'knockdown' in zebrafish" (Pubmed). *Nature Genetics* **26** (2): 216–20. doi:10.1038/79951

39. Chvatal V. A greedy heuristic for the set covering problem. Math Oper Res 1979;4:233—5.

40. Feo TA, Resende MGC. A probabilistic heuristic for a computationally difficult set covering problem. Oper Res Lett 1989;8(2):67—71.

41. Beasley JE. An algorithm for set covering problems. Eur J Oper Res 1987;31:85—93.

42. Beasley JE, Chu PC. A genetic algorithm for the set covering problem. Eur J Oper Res 1996;94:392—404.

43. Eremeev AV. A genetic algorithm with a non-binary representation for the set covering problem. In: Proceedings of the Operational Research. Springer-Verlag; 1998. p. 175—81.

44. Balas, E., and Ho, A. (1980), "Set covering algorithms using cutting planes, heuristics, and subgradient optimisation: A computational study", Mathematical Programming Study

12, 37-60.

45. Feo TA, Resende MGC. A probabilistic heuristic for a computationally difficult set covering problem. Oper Res Lett 1989;8(2):67—71.

VITA AUCTORIS

NAME:	Xiaoguang Li
PLACE OF BIRTH	Beijing, China
YEAR OF BIRTH	1982
EDUCATION	Nanjing University of Technology, Nanjing, China 2000-2004 B.Sc. University of Windsor, Windsor, Ontario 2006-2008 M.Sc.