Minimal siRNA Set Cover Heuristic for Gene Family Knockdown

Xiaoguang Li, Alioune Ngom, Luis Rueda

[1] School of Computer Science, University of Windsor, 5115 Lambton Tower,
401 Sunset Avenue, Windsor, N9B 3P4, Ontario, Canada
{li1111m, angom, lrueda}@uwindsor.ca

Abstract. RNA interference (RNAi) is widely used as an important tool for genomic and therapeutic applications. Small interfering RNA (siRNA) is involved in the RNA interference process and knocks down the expression of a specific gene. During this process, messenger RNA (mRNA) is degraded by siRNA, the function of a harmful gene can be inhibited. 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. In this paper, we explore some 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 problems.

Keywords: Minimal siRNA Cover, Set Cover, Gene Family Knockdown.

1 Introduction

RNA interference (RNAi) is a highly evolutionally conserved process of post-transcriptional gene silencing (PTGS) by double stranded RNA (dsRNA). When introduced into a cell, it will cause sequence-specific degradation of homologous mRNA sequences. It 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

M. Chetty, S. Ahmad, A. Ngom, S. W. Teng (Eds.): PRIB2008,
Supplementary Conf. Proc., pp.37-48, 2008.

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 paper, 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 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].

The rest of the paper is organized as follows. In Section 2, we give a brief introduction to minimal siRNA set cover problem and formulate it as an integer linear programming problem. The set cover problem has been well studied and a number of exact and approximate methods exist for it, including the exact branch-and-bound algorithm [9], probabilistic greedy algorithm [9], LP relaxation [13] and genetic algorithm [14, 15]. In Section 3, we describe one dominated target covering heuristic with some modifications based on Wang et al. [16], one dominant siRNA selection heuristic one genetic algorithm heuristic improved from [14] and one novel deterministic greedy heuristic called forward selection for the minimal siRNA set cover problem. Experimental results are discussed in Section 4 and we conclude in Section 5.

2 Minimal siRNA Set Cover Problem

Given a siRNA set, $S = \{s_1,\ldots,s_N\}$, and a gene set, $G = \{g_1,\ldots g_K\}$, 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 1 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 shows the matrix without off target effects. In this paper, we select the off target genes randomly.

Table 1. Example of a matrix with off target effects.

		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. Example of a matrix without 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

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, 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 is that, given a N×K matrix W with a siRNA set, $S= \{s_1,\ldots,s_N\}$ and a gene set $G= \{g_1,\ldots g_K\}$, 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:

$$\text{Minimize: } \sum_{j=1}^{N} x_j \tag{1}$$

$$\text{Subject to: } \sum_{j=1}^{N} w_{ij} x_j \geq 1 \qquad i=1,\ldots,K \tag{2}$$

$$x_j \in \{0,1\} \qquad j=1,\ldots,N \tag{3}$$

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

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

3 Heuristic Methods for Minimal siRNA Set Cover Problem

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.

3.1 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:

$$\mathrm{cov}(s_j, g_i) = w_{ij} \times \frac{1}{|S_{g_i}|} \quad s_j \in S_{g_i},\ g_i \in G \tag{4}$$

where $0 \leqslant cov(s_j,g_i) \leqslant 1$ and S_{gi} 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 (4), the coverage is calculated as:

$$C(s_j) = \max\{\mathrm{cov}(s_j, g_i) \mid 1 \le i \le k\} \quad g_i \in G_{sj} \tag{5}$$

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 3, which derives from Table 2.

Table 3. Example of a coverage function table.

	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

When $C(s_j) = 1$, we consider s_j as an essential siRNA since any feasible solution has to include it. In Table 3, 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}

3.2 Dominant siRNA Selection Heuristic (DSS)

We also want to satisfy the selection of dominant siRNAs; s_j dominates s_l if $Gs_l \subset Gs_j$. In Table 2, 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_l only if $C(s_j)>C(s_l)$. In Table 2, 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 (4) 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 3 with an amount that takes into account the number of coverd genes. The new *cov* function has the form as follows:

$$\mathrm{cov}(s_j, g_i) = w_{ij} \times \frac{1}{|Sg_i|} \times \frac{1}{m-|Gs_j|+1} \qquad s_j \in S_{g_i}, \quad g_i \in G \qquad (\mathbf{6})$$

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 (6), 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 3.1 only with the exception that function C is defined by using Equation (6) instead of Equation (4). 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.

3.3 Genetic Algorithm for Minimal siRNA Set Cover Problem

Beasley et al. [14] presented a genetic algorithm-based heuristic for set covering problem. Based on this method, our GA inherits the siRNA selection function defined in Section 3.2. The improved genetic approach can be illustrated in details as follows.

3.3.1 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 $S = \{s_1,\ldots,s_N\}$, 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:

$$f_i = \sum_{j=1}^{N} s_{ij} \qquad (7)$$

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

3.3.2 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.

3.3.3 Crossover Operator

We implement the *fusion operator* of [14] 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 ∈ [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 $1 - p$

3.3.4 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 [14]. It essentially depends on the rate of the GA convergrence 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 [14].

$$Num_{mut} = \left\lceil \frac{m_f}{1 + \exp\left(-4m_g\left(t - m_c\right)/m_f\right)} \right\rceil \qquad (8)$$

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.

3.3.5 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.

3.3.6 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).

3.4 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 (6) to select a s_j with the best value. 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.

4 Computational Experiments

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 in Java, the hardware platform is a workstation with Intel(R) Xeon(TM) CPUs 3.20GHz and 3.19GHz with 8.00GB of RAM and the operating system is Microsoft Windows XP, Professional x64 Edition.
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].

We design the siRNA sequences for above 3 gene families by a software [16]. It generates 96 siRNA sequences in family 1, 277 in family 2 and 1,339 in family 3.
Experiment results show that our heuristics are able to select less number of siRNAs than the methods mentioned in [9]. When the number of siRNA increases, DSS, GA_DSS and FS give much better results than other methods. It can be expected that this will provide a great help for RNAi interference experiments. Table 4, 5 and 6 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).

Table 4. Results for Target Family 1.

size of target set	G=2 S=22	G=3 S=27	G=4 S=30	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. Results for Target Family 2.

size of target set	G=5 S=40	G=10 S=79	G=20 S=137	G=30S =170	G=40 S=201	G=5 S=254	G=53 S=277
PG	5	11	24	36	51	33	30
BB	5	11	24	37	51	34	30
DTC	5	10	15	21	26	20	18
DSS	5	9	15	20	25	19	17
GA_DSS	5	9	15	20	25	19	17
FS	5	9	15	20	25	19	17

Table 6. Results for Target Family 3.

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	14	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

5 Conclusions and Future Work

In this paper, 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 other three improved methods. Experiments showed that these methods are able to obtain near minimal solutions which are still comparable to the known heuristics [9] for this problem. We plan to study different variations of our heuristic feasibility operator. Future research will include designing a much larger dataset than we used in this paper, since there is no such remarkable difference between the results of DSS, GA_DSS and FS for a relatively small gene family.

Acknowledgments. We would like to thank Dr. Coen M. Adema for providing us gene name and gene accession number of target family 1.

References

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. Beasley, J.E.: An algorithm for set covering problems. European Journal of Operational Research. Vol. 30, pp. 85--93 (1987)
14. Beasley, J.E., Chu, P.C.: A genetic algorithm for the set covering problem. European Journal of Operational Research. Vol. 94, pp. 392--404 (1996)
15. Eremeev, A.V.: A genetic algorithm with a non-binary representation for the set covering problem. Proc. Of Operational Research, Springer-Verlag, pp. 175--181 (1998)
16. siRNA sequence design software: https://rnaidesigner.invitrogen.com/rnaiexpress/
17. Adema, C.M., Hertel, L.A., Miller, R.D., Loker, E.S.: A family of fibrinogen-related proteins that precipitates parasite-derived molecules is produced by an invertebrate after infection. Proc. Natl. Acad. Sci.U.S.A. Vol.94, pp. 8691—8996 (1997)
18. Crompton, D.W.T.: How much human helminthiasis is there in the world? J. Parasitol. Vol.85, pp. 397—403 (1999)