Filtering of Ineffective siRNAs and Improved siRNA Design Tool

Prudence WH Wong[1] **TW Lam**[1] **YC Mui**[1] **SM Yiu**[1]
HF Kung[2] **Marie Lin**[2] **YT Cheung**[2]

[1] Department of Computer Science, University of Hong Kong, Hong Kong
Email: {whwong, twlam, ycmui, smyiu}@cs.hku.hk
[2] Institute of Molecular Biology, University of Hong Kong, Hong Kong

Abstract

Short interfering RNAs (siRNAs) can be used to suppress gene expression and have many potential applications in therapy, yet how to design an effective siRNA is still not clear. Based on the MPI basic principles (Tuschl, Elbashir, Harborth & Weber 2003), a number of siRNA design tools have been developed in the past two years. The set of candidates output by these tools is usually large and often contains some ineffective siRNAs. In view of this, we initiate the study of filtering ineffective siRNAs. The contribution of this paper is two-fold. Firstly, we propose a fair scheme to compare existing design tools based on real data in the literature. Secondly, we attempt to improve the MPI principles and existing tools by an algorithm that can filter ineffective siRNAs. The algorithm is based on some new observations on the secondary structure, which we have verified by AI techniques (decision trees and support vector machines). We have tested our algorithm together with the MPI principles and the existing tools. The results show that our filtering algorithm is effective.

Keywords: siRNAs, secondary structure, decision trees

1 Introduction

Short interfering RNAs (siRNAs), of length about 21, can be used to suppress gene expression (Fire, Xu, Montgomery, Kostas, Driver & Mello 1998, Elbashir, Harborth, Lendeckel, Yalcin, Weber & Tuschl 2001, Elbashir, Lendeckel & Tuschl 2001, Caplen, Parrish, Imani, Fire & Morgan 2001) and have many potential applications in therapy, for example, it is believed that siRNAs can be used to suppress the HIV-1 replication in human cell lines (Jacque, Triques & Stevenson 2002). Different genes require different siRNAs to suppress the expression. An siRNA is, in fact, a DNA sequence that is formed by a substring of the mRNA of the target gene. However, not every substring of the target mRNA can form an effective siRNA (Holen, Amarzguioui, Wiiger, Babaie & Prydz 2002). A typical mRNA can have length of thousands. The number of potential candidates for siRNAs is therefore huge. To verify whether a given siRNA is effective, one must go through laboratory experiments. These experiments are both time-consuming and expensive. Yet how to design an effective siRNA (that is, to select the right substring from the mRNA for the construction of the siRNA) is still not clear.

As the first attempt to solve the problem, Tuschl et. al. (Tuschl et al. 2003) provided a set of guidelines, commonly known as the MPI principles, on how to design effective siRNAs. These principles try to capture some properties that an effective siRNA should have, for example, the GC-content[1] of an siRNA should be between 30% and 70%. However, there are two issues in these principles. The properties given in the principles are not exclusive to effective siRNAs. In fact, among 20 ineffective siRNAs that have been reported in the literature, 6 of them also follow the MPI principles. Another issue is that the principles are rather primitive and not selective, the number of candidates that follow the principles is usually large. We have tested the MPI principles using 41 mRNAs of average length 2338. The average number of candidates reported is 331.

In the past two years, several siRNA design tools have been developed by refining and extending the MPI principles. However, in general, the set of candidates reported by most of these tools is still large (hundreds) and often contains some ineffective siRNAs (see Table 1).

Our Contributions: The contribution of this paper is two fold.

1. **A Comparison Scheme:** Despite the fact that quite a number of design tools have been developed, there is no study on comparing these

[1] GC-content is the percentage of the nucleotides G and C on the length-21 siRNA.

Design tools	# of relevant cases	# of ineffective siRNAs reported before filtering	# of ineffective siRNAs reported after filtering
Ambion_AA	5	4	2
MIT_AA (default)	5	1	1
Dharmacon_NA (default)	9	1	1
Emboss_NA	9	8	3
JackLin_NA	9	4	2
MPI principles	9	6	1
Dharmacon_NN	20	1	1
MIT_NN	20	13	7
OptiRNAi_NN	20	0	0
Qiagen_NN	20	3	1

Table 1: Number of Ineffective siRNAs Filtered by Our Algorithm. (The tools are grouped by the starting nucleotides of the siRNAs reported.)

tools. In fact, it is not trivial to compare these tools directly as the numbers of candidate siRNAs reported by these tools vary a lot. It is desirable to have a fair scheme to evaluate these tools. In this paper, we propose a fair scheme to compare these tools based on the published siRNAs. The idea is to make use of a random selector that will randomly pick the candidates from the target mRNA. The number of candidates to be chosen by the random selector depends on the output size of the tool in concern. Based on the published siRNAs, we calculate some indices showing how much the choice of the tool is better than a random choice. The random selector actually acts as a reference (control) for comparison. Our aim is to filter ineffective siRNAs, so the focus of our comparison is mainly on published ineffective siRNAs, the comparison for effective siRNAs is used as a reference.

We have evaluated 7 existing tools and the MPI principles. The tools include Ambion (Ambion 2003), Dharmacon (Dharmacon 2003), Emboss (Williams 2002), Jack Lin (Lin 2002), MIT (MIT 2002), Qiagen (Qiagen 2003), and OptiRNAi (Cui, Ning, Naik & Ducan 2003). The result shows that in general, most of these tools still output quite a number of ineffective siRNAs and have a similar (if not worse) performance as the random selector. For effective siRNAs, Jack Lin seems to be the best based on the published data.

2. **A Filtering Algorithm:** Basically, most of these tools still try to identify a set of properties for selecting effective siRNAs. In this paper, we initiate the study of the properties that an *ineffective* siRNA would possess, which enable one to filter out the candidates that are unlikely to be an effective siRNA. We develop a filtering algorithm to improve the MPI principles and existing tools. The algorithm is based on some new observations on the secondary structure, which we have verified by AI techniques (decision trees and support vector machines). We have evaluated our filtering algorithm by applying it to the existing tools and the MPI principles. The results show that our filtering algorithm is effective. The number of ineffective siRNAs reported can be reduced by up to 83% while the number of effective siRNAs reported is only reduced by an average of 17%.

Organization of the paper: The rest of this paper is organized as follows. Section 2 presents the scheme for comparing existing siRNA design tools and the comparison result of 7 existing tools. In Section 3, we then present the filtering algorithm for filtering ineffective siRNAs and discuss the experimental results of applying our filtering algorithm on the 7 existing tools. In Section 4, we discuss how we find the filtering rule. Section 5 gives a summary and conclusion of our work.

2 The Comparison Scheme

Idea and results: In this section we compare the performance of existing siRNA design tools and the MPI principles using real data in the literature. From the literature, there are 55 effective siRNAs and 20 ineffective siRNAs for human genes. (The references for the real data can be found in the Appendix.) We compare the following tools: Ambion, Dharmacon, Emboss, Jack Lin, MIT, Qiagen, and OptiRNAi. Note that if a tool has options to restrict the selected siRNAs to have AA, NA, or NN as the starting two nucleotides, we tried the default and the NN options (where "N" stands for any nucleotide).

For a given mRNA, the numbers of candidate siRNAs reported by the tools can vary a lot. It is not trivial how one can compare these tools directly. We propose to use a random selector that randomly picks

Design tools	Against **INEFFECTIVE** siRNAs		
	actual %	expected %	net %
Ambion_AA	80%	72%	8%
MIT_AA (default)	20%	6%	14%
Dharmacon_NA (default)	11%	2%	9%
Emboss_NA	89%	67%	22%
JackLin_NA	44%	25%	19%
MPI principles	67%	38%	29%
Dharmacon_NN	5%	3%	2%
MIT_NN	65%	61%	4%
OptiRNAi_NN	0%	0%	0%
Qiagen_NN	15%	2%	13%

Table 2: The net percentages of various siRNA design tools against **ineffective** siRNAs.

candidates from the target mRNA as a reference for comparison. To handle the issue of different output sizes for the tools, we make sure that the number of candidates to be selected by the random selector would be the same as the number of candidates reported by the tool in concern. Also, if the tool only reports siRNAs starting with AA, the random selector will only select siRNAs starting with AA.

We then compare the two sets of candidates against the known siRNAs. For ineffective siRNAs, intuitively, if the tool reports less such siRNAs than the random selector, the choice of the tool is better than a random choice. We calculate the percentages of known ineffective siRNAs that have been reported by the tool and the random selector. The difference in these percentages, *the net percentage*, is used as index to show how much the choice of the tool is better than a random choice. Note that in calculating the percentages, if the tool only report siRNAs starting with AA, we only consider the known siRNAs starting with AA. In fact, we do not actually run a random selector. We compute the *expected* percentage of ineffective siRNAs reported by the random selector. The detailed calculation will be discussed below. The *net percentage* is then defined as the actual percentage of ineffective siRNAs reported by the tool minus the expected percentage of the random selector. Obviously, a good siRNA tool should have a negative net percentage against ineffective siRNAs.

Table 2 shows the net percentages of various design tools against ineffective siRNAs. We see that most of the tools have positive net percentage; in other words, these tools report more ineffective siRNAs than the random selector. So, their choices of candidates are no better than random choices with respective to ineffective siRNAs.

Computing the expected percentage for random selector: Now, we discuss the details of how to compute the expected percentage of the random selector. Consider a design tool T that reports siRNAs starting with AA. The other two cases for NA and NN are similar. Suppose M is the input mRNA. Let S_M be the set of ineffective siRNAs starting with AA that are reported in the literature and $\sigma_M = |S_M|$. Let n_M be the number of length-21 substrings of M that start with AA. Let k_M be the size of output of T for M. The random selector will select k_M siRNAs from the n_M candidates randomly. Let X_M denote the number of siRNAs reported by the random selector that are in S_M. Then the expected value of X_M can be computed as follows.

$$\begin{aligned} E(X_M) &= \sum_{1 \le i \le \sigma_M} i \cdot Pr(X_M = i) \\ &= \sum_{1 \le i \le \sigma_M} i \cdot \frac{\binom{\sigma_M}{i}\binom{n_M - \sigma_M}{k_M - i}}{\binom{n_M}{k_M}}, \end{aligned}$$

where $\binom{n}{r}$ denotes the number of combinations of choosing r items from n items. The expected number of ineffective siRNAs reported by the random selector equals to $\sum_M (E(X_M))$, and the expected percentage equals to $\sum_M (E(X_M))$ divided by the number of ineffective siRNAs in the literature that start with AA.

We have also performed the comparison of the tools against effective siRNAs. In this case, the actual percentage, the expected percentage, and the net percentage are defined on known effective siRNAs. A good tool should have a positive net percentage. Table 3 shows the net percentages of various design tools against effective siRNAs. All the tools have positive net percentages, meaning that they report more effective siRNAs than the random selector. Their choices are better than random choices. In particular, Jack Lin seems to be the best based on the published data.

To conclude, the existing tools perform well in selecting the effective siRNAs but are not good for filtering out the ineffective ones. In the next section, we show how to enhance these tools by a filtering algorithm that filters potential ineffective siRNAs.

Design tools	Against **EFFECTIVE** siRNAs		
	actual %	expected %	net %
Ambion_AA	87%	72%	15%
MIT_AA (default)	40%	3%	37%
Dharmacon_NA (default)	6%	3%	3%
Emboss_NA	94%	72%	22%
JackLin_NA	63%	20%	43%
MPI principles	83%	51%	32%
Dharmacon_NN	9%	6%	3%
MIT_NN	85%	64%	21%
OptiRNAi_NN	36%	1%	35%
Qiagen_NN	44%	3%	41%

Table 3: The net percentage of various siRNA design tools against **effective** siRNAs.

3 The Filtering Algorithm and Its Performance

3.1 Performance of the filtering algorithm

From the discussion in Section 2, we see that both the MPI principles and most design tools report a certain number of ineffective siRNAs. In view of this, we attempt to improve the MPI principles and existing tools by an algorithm that can filter ineffective siRNAs. The target of the filtering algorithm is to reduce the number of ineffective siRNAs reported, and more importantly, reduce the net percentage against ineffective siRNAs.

We have applied the filtering algorithm on the output of the design tools to filter potential ineffective candidates. We observe that the output size is reduced by about 23% on average. We have also shown in Table 1 that the number of ineffective siRNAs drops by a significant amount. Regarding the net percentage against ineffective siRNAs, Table 4 shows that the percentages decrease drastically for most of the tools (up to 46% for the MPI principles). In particular, the net percentages of five of them become negative, implying that the corresponding tools now report fewer ineffective siRNAs than the random selector. This shows that our filtering algorithm is effective. Note that the expected percentage of the random selector is based on the reduced size of the output after filtering.

For the net percentage against effective siRNAs, Table 5 shows that the percentages decrease after applying the filtering but by a smaller amount; precisely, the net percentage decreases by at most 11%.

3.2 Details of the filtering algorithm

In this section, we give the details of the filtering algorithm. Note that in the process of suppressing gene expression, the siRNA needs to approach the corresponding target site on the mRNA. One of the factors affecting the success is the accessibility of the mRNA near the target site. This motivates us to study the secondary structure of the mRNA in concern, i.e., the pairing of the bases of the mRNA.

Repelling loops and big repelling loops: Our filtering algorithm is based on the secondary structure properties that we call *repelling loops* and *big repelling loops*. The idea is as follows. The pairing of the bases may introduce loops (e.g. internal loop and multi-branched loop (Waterman 2000)). See Figure 1 for an example of a multi-branched loop. Suppose that a target site is hidden between two very close branches of a loop (because of the repelling force on the other side of the loop). Then it may not be easy for the corresponding siRNA to access the target site, so the siRNA has a high chance to be ineffective. We call such a loop a *repelling loop* with respect to that siRNA and the two branches are called the *enclosing branches*. Figure 1 shows an example of a repelling loop and the corresponding target site. Furthermore, if the repelling loop is big, i.e., the number of unpaired bases is large, the unpaired bases on the other sides of the loop have higher free energy, thus may interfere the siRNA activity. We call such a loop a *big repelling loop* (with respect to that siRNA).

Now we give precise definitions for repelling loops and big repelling loops. For a given target site, consider the loops that are near to the target site and with at least two branches. A target site is near to a loop if it overlaps with the loop or is within a short distance from the loop, say 10 nucleotides. Intuitively, for each loop, if the segment enclosed by the enclosing branches (with respect to the target site) is small relative to the total length of the loop, the target site is more difficult to be accessed. Therefore, we measure the ratio between the length of the segment enclosed by the enclosing branches (with respect to the target site) and the total length of the loop. If this ratio r is less than 0.5, we say that the loop is an r*-repelling loop*. For example, the loop in Figure 1 is a (2/20)-repelling loop. Furthermore, if

Design tools	Net % against **INEFFECTIVE** siRNAs		
	before filtering	after filtering	change
Ambion_AA	8%	-22%	-30%
MIT_AA (default)	14%	15%	+1%
Dharmacon_NA (default)	9%	9%	0%
Emboss_NA	22%	-20%	-42%
JackLin_NA	19%	2%	-17%
MPI principles	29%	-17%	-46%
Dharmacon_NN	2%	3%	+1%
MIT_NN	4%	-12%	-16%
OptiRNAi_NN	0%	0%	0%
Qiagen_NN	13%	3%	-10%

Table 4: Comparison of the net percentages against **ineffective** siRNAs before and after applying the filtering algorithm.

Design tools	Net % against **EFFECTIVE** siRNAs		
	before filtering	after filtering	change
Ambion_AA	15%	8%	-7%
MIT_AA (default)	37%	26%	-11%
Dharmacon_NA (default)	3%	4%	+1%
Emboss_NA	22%	17%	-5%
JackLin_NA	43%	34%	-9%
MPI principles	32%	27%	-5%
Dharmacon_NN	3%	5%	+2%
MIT_NN	21%	18%	-3%
OptiRNAi_NN	35%	33%	-2%
Qiagen_NN	41%	35%	-6%

Table 5: Comparison of the net percentages against **effective** siRNAs before and after applying the filtering algorithm.

the r-repelling loop has length ℓ, then we say that it is an *ℓ-big r-repelling loop*. In particular, our filtering algorithm considers 0.25-repelling loops and 15-big 0.25-repelling loops. The thresholds for repelling loops and big repelling loops are obtained by using AI techniques, which we will describe in Section 4.

The filtering algorithm: Based on the concept of repelling loops and big repelling loops, we have devised a filtering algorithm to filter potential ineffective siRNAs. The details are as follows. Consider any mRNA and a set of candidate siRNAs. We first obtain the secondary structures of the mRNA from Zuker's MFOLD algorithm (Zuker 2003). The MFOLD algorithm usually reports over ten secondary structures, each with a free energy indicates the stability of the corresponding structure. We focus on the five structures that have the lowest free energy, i.e., the five most stable structures. To determine whether to filter a candidate siRNA, we count, for each such structure, the number of 0.25-repelling loops and the number of 15-big 0.25-repelling loops with respect to that siRNA candidate. The filtering condition for an siRNA candidate is:

> If three out of the five most stable secondary structures contain at least one 15-big 0.25-repelling loop and at least a total of two 0.25-repelling loops with respect to the candidate, then the candidate is filtered.

The filtering algorithm checks the filtering condition for each candidate siRNA and report those that are not filtered.

4 Finding the Filtering Rule by AI techniques

In this section we discuss how the filtering rule is derived using decision tree learning (Quinlan 1987). Together with repelling loops, big repelling loops we mentioned in Section 3, we have also considered the following two factors that are related to our observations.

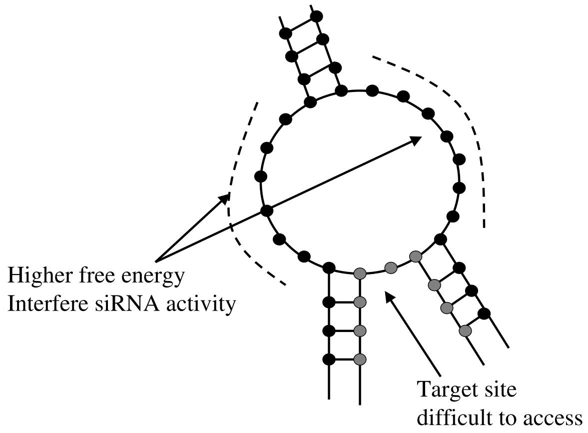

Figure 1: A 20-big 0.1-repelling loop.

- The number of branches in a repelling loop: intuitively, if the number of branches increases, branches will be closer together, so if a target site is enclosed by the branches, it may be difficult to access it.
- The free bases in the target site: Free base has higher free energy and may interfere siRNA activity. So, we also consider the number of free bases in the target site. To reflect the relative strength of a CG-bond and an AT-bond, we assign a weight of 2 to a free A or T base and a weight of 3 to a free C or G base. The total weight of the free bases will be used in the decision tree training.

Training Process: For repelling loops and big repelling loops, there are two parameters, r and ℓ, to consider. We repeatedly train the decision tree by fixing the repelling loop threshold in the range 0.05 to 0.45, incremented by 0.05 each time. For a particular repelling loop threshold r, we compute the following attributes for each siRNA. Recall that when considering the secondary structures, we use the five most stable structures reported by the MFOLD algorithm.

1. The largest number α such that at least 3 out of the 5 most stable secondary structures contain at least α r-repelling loops.
2. For ℓ equals 13, 14, up to 17, the largest number β_ℓ such that at least 3 out of the 5 most stable secondary structures contain at least β ℓ-big r-repelling loops.
3. The average number of branches of all the repelling loops in the three secondary structures having at least α r-repelling loops. Note that the two branches that enclose the target site are not counted.
4. The average of the total weight of free bases in the three secondary structures having at least α r-repelling loops.

We then train the decision trees by including different attributes as follows. We have designed six sets of experiments: the first five sets correspond to one of the value of the big loop threshold β_ℓ, and the last set correspond to all the big loop thresholds. For each experiment, we include the attributes α and the corresponding β('s), while the remaining two attributes may or may not be included. As a result, we have four combinations: including both the unpaired weight and the number of branches including either one of them, and including neither of them.

We train the decision tree using a subset of data from the literature, which contains 27 effective and 6 ineffective siRNAs that satisfy the following basic principles: (i) the position of the target site is at least 100 from the start codon of the corresponding mRNA, (ii) the GC-content of the siRNA is between 30% and 70%, and (iii) the siRNA does not contain consecutive long runs of four or more equal nucleotides, e.g., GGGG.

Results: Four decision trees are returned for each experiment. For each decision tree, we compute the rate of correct classification for ineffective and effective siRNAs. Since our target is to filter ineffective siRNAs, the ideal decision tree is one having 100% correct classification for ineffective siRNAs and a high correct classification rate for effective siRNAs. Figure 2 shows the classification rate of the decision trees for all the experiments. In each experiment, we only report the best decision tree, i.e., the one with the highest classification rate for ineffective siRNAs.

The results show that the best decision tree is the one for Experiments 3 and 6 with repelling loop threshold $r = 0.25$, which leads to the filtering rule used by our filtering algorithm. The classification rate for ineffective and effective siRNAs is 100% and 92.6%, respectively. In fact, when the repelling loop threshold is 0.25, all the eight decision trees for Experiments 3 and 6 are the same. This shows that even if we include the unpaired weight or the number of branches, the best decision tree only involves the attributes about the number of 0.25-repelling loops

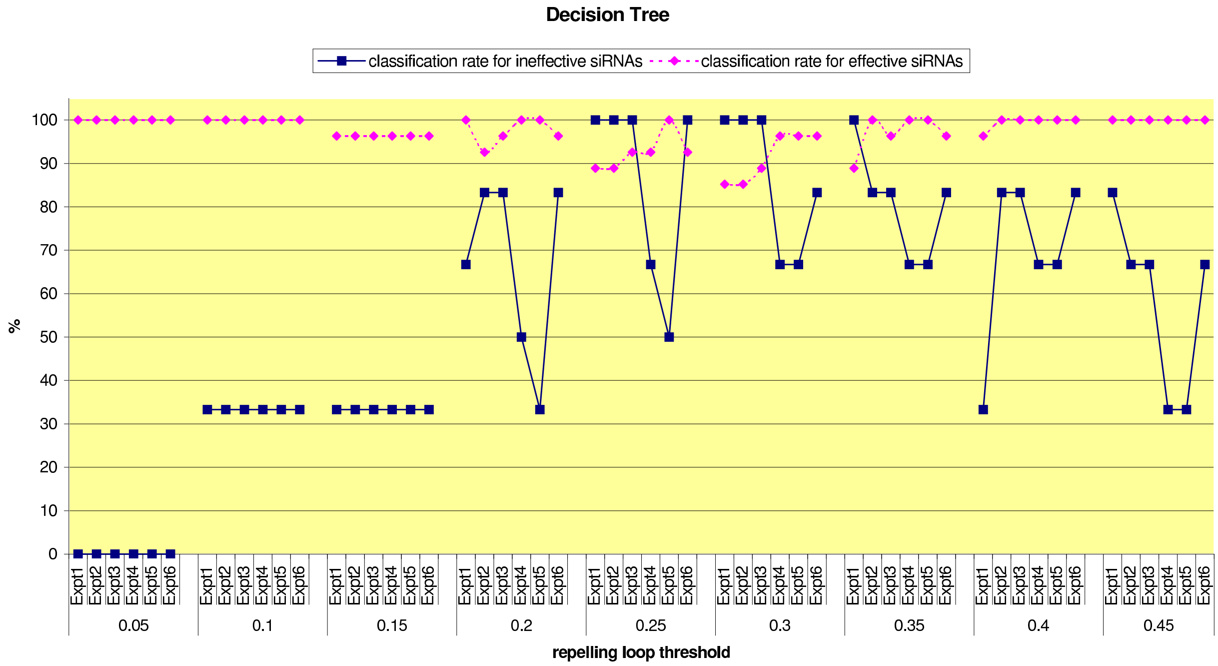

Figure 2: The classification rates of the decision trees in various experiments.

and the number of 15-big 0.25-repelling loops.

We also make use of the support vector machine (Joachims 1999) to see if the classification is consistent with that of the decision tree. We train the support vector machine learning module using the attributes α and β as in Experiments 3 and 6. The support vector machine obtained has a similar performance as the decision tree we obtained. Precisely, the classification rate for both ineffective and effective siRNAs are the same as that of the decision tree, and even further, the sets of siRNAs that are classified as ineffective and effective are the same for both the decision tree and the support vector machine. These results show that the attributes and the thresholds are selected appropriately.

5 Conclusion

In this paper, we have proposed a scheme to evaluate existing siRNA design tools based on published effective and ineffective siRNAs. In the scheme, the output of each design tool is compared to a set of randomly selected siRNA candidates. The results show that existing tools are not good at filtering ineffective siRNAs. We also propose a filtering algorithm to filter potential ineffective siRNA candidates from the output of existing tools. The algorithm is based on two observations, namely repelling loops and big repelling loops, on secondary structures of the target mRNA. The rule for classifying potential ineffective siRNAs from other candidates is generated with the help of AI techniques, in particular, the decision tree and support vector machine. The filtering algorithm is shown to be effective.

The results of this paper give evidence that secondary structures should be considered for the design of siRNA. We are in the process of designing laboratory experiments to further verify our observations on secondary structures.

References

Ambion (2003). Ambion siRNA Target Finder. URL: http://www.ambion.com/techlib/misc/siRNA_finder. html.

Caplen, N. J., Parrish, S., Imani, F., Fire, A. & Morgan, R. A. (2001), 'Specific inhibition of gene expression by small double-stranded rnas in invertebrate and vertebrate systems', *Proceedings of the National Academy of Science* **98**, 9742–9747.

Cui, W., Ning, J., Naik, U. P. & Ducan, M. K. (2003), OptiRNAi, a web-based program to select siRNA sequences, *in* 'Proceedings of The Computational Systems Bioinformatics Conference', pp. 433–434. URL: http://bioit.dbi.udel.edu/rnai.

Dharmacon (2003). Dharmacon siDESIGN Center. URL: http://design.dharmacon.com/rnadesign /default.aspx?SID=691011983.

Elbashir, S. M., Harborth, J., Lendeckel, W., Yalcin, A., Weber, K. & Tuschl, T. (2001), 'Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells', *Nature* **411**, 494–498.

Elbashir, S. M., Lendeckel, W. & Tuschl, T. (2001), 'RNA interference is mediated by 21- and 22-nucleotide RNAs', *Genes and Development* **15**, 188–200.

Fire, A., Xu, S., Montgomery, M. K., Kostas, S. A., Driver, S. A. & Mello, C. C. (1998), 'Potent and specific genetic interference by double-stranded rna in Caenorhabditis elegans', *Nature* **391**, 806–811.

Holen, T., Amarzguioui, M., Wiiger, M. T., Babaie, E. & Prydz, H. (2002), 'Positional effects of short interfering RNAs targeting the human coagulation trigger Tissue Factor', *Nucleic Acids Research* **30**(8), 1757–1766.

Jacque, J.-M., Triques, K. & Stevenson, M. (2002), 'Modulation of HIV-1 replication by RNA interference', *Nature* **418**, 435–438.

Joachims, T. (1999), Making large-Scale SVM Learning Practical., *in* B. Scholkopf, C. Burges & A. Smola, eds, 'Advances in Kernel Methods - Support Vector Learning', MIT-Press.

Lin, J. (2002). Jack Lin's siRNA Sequence Finder. URL: http://www.sinc.sunysb.edu/Stu/shilin/rnai.html.

MIT (2002). siRNA Selection Program. Whitehead Institute for Biomedical Research. URL: http://jura.wi.mit.edu/pubint/http://iona.wi.mit.edu/ siRNAext/.

Qiagen (2003). The siRNA design tool Qiagen. URL: http://python.penguindreams.net/Xeragon_Order_Entry/jsp/Index.jsp.

Quinlan, J. (1987). Decision Tree C4.5. URL: http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/c4.5/tutorial.html.

Tuschl, T., Elbashir, S., Harborth, J. & Weber, K. (2003). The siRNA user guide. URL: http://www.rockefeller.edu/labheads/tuschl/sirna.html.

Waterman, M. S. (2000), *Introduction to Computational Biology – Maps, sequences and genomes*, Chapman & Hall/CRC.

Williams, G. (2002). The siRNA design tool EMBOSS. URL: http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Apps/sirna.html.

Zuker, M. (2003), 'Mfold web server for nucleic acid folding and hybridization prediction', *Nucleic Acids Research* **31**(13), 3406–3415.

Appendix: References for the published data

X. Bai, D. Zhou, J. Brown, B. Crawford, T. Hennet, and J. Esko. Biosynthesis of the linkage region of glycosaminoglycans. *Journal of Biological Chemistry*, 276(51):48189–48195, 2001.

J. Bakker, X. Lin, and W. Nelson. Methyl-CpG binding domain protein 2 represses transcription from hypermethylated p-class glutathione-S-transferase gene promoters in hepatocellular carcinoma cells. *Journal of Biological Chemistry*, 277:22573–22580, 2002.

V. Chevrier, M. Piel, N. Collomb, Y. Saoudi, R. Frank, M. Paintrand, S. Narumiya, M. Bornens, and D. Job. The Rho-associated protein kinase p160ROCK is required for centrosome positioning. *Journal of Cell Biology*, 157:807–817, 2002.

D. Cortez, S. Guntuku, J. Qin, and S. J. Elledge. ATR and ATRIP: Partners in checkpoint signaling. *Science*, 294:1713–1716, 2001.

J. Harborth, S. M. Elbashir, K. Bechert, T. Tuschl, and K. Weber. Identification of essential genes in cultured mammalian cells using small interfering RNAs. *Journal of Cell Science*, 114:4557–4565, 2001.

E. Hewitt, L. Duncan, D. Mufti, J. Baker, P. Stevenson, and P. Lehner. Ubiquitylation of MHC class I by the K3 viral protein signals internalization and TSG101-dependent degradation. *European Molecular Biology Organization Journal*, 21(10):2418–2429, 2002.

H. Hirai and H.-G. Wang. A role of the C-terminal region of human Rad9 (hRad9) in nuclear transport of the hRad9 checkpoint complex. *Journal of Biological Chemistry*, 277(28):25722–25727, 2002.

T. Holen, M. Amarzguioui, M. T. Wiiger, E. Babaie, and H. Prydz. Positional effects of short interfering RNAs targeting the human coagulation trigger Tissue Factor. *Nucleic Acids Research*, 30(8):1757–1766, 2002.

X. Jiang, H.-E. Kim, H. Shu, Y. Zhao, H. Zhang, J. Kofron, J. Donnelly, D. Burns, S. chung Ng, S. Rosenberg, and X. Wang. Distinctive Roles of PHAP Proteins and Prothymosin- in a Death Regulatory Pathway. *Science*, 299:223–226, 2003.

M. Kisielow, S. Kleiner, M. Nagasawa, A. Faisal, and Y. Nagamine. Isoform-specific knockdown and expression of adapter protein ShcA using small interfering RNA. *Biochemical Journal*, 363:1–5, 2002.

P. Lassus, X. Opitz-Araya, and Y. Lazebnik. Requirement for Caspase-2 in Stress-Induced Apoptosis Before Mitochondrial Permeabilization. *Science*, 297:1352–1354, 2002.

J. Liu, F. Yao, R. Wu, M. Morgan, A. Thorburn, R. Finely, and Y. Chen. Mediation of the DCC apoptotic signal by DIP13 alpha. *Journal of Biological Chemistry*, 277(29):26281–26285, 2002.

N. Mailand, C. Lukas, B. K. Kaiser, P. K. Jackson, J. Bartek, and J. Lukas. Deregulated human Cdc14A phosphatase disrupts centrosome separation and chromosome segregation. *Nature Cell Biology*, 4:318–322, 2002.

S. Martin-Lluesma, V. Stucke, and E. Nigg. Role of Hec1 in spindle checkpoint signaling and kinetochore recruitment of Mad1/Mad2. *Science*, 297:2267–2270, 2002.

S. Prasanth, K. Prasanth, and B. Stillman. Orc6 involved in DNA replication, chromosome segregation, and cytokinesis. *Science*, 297:1026–1031, 2002.

Y. Shang and M. Brown. Molecular determinants for the tissue specificity of SERMs. *Science*, 295:2465–2468, 2002.

M. Surka, C. Tsang, and W. Trimble. The mammalian septin MSF localizes with microtubules and is required for completion of cytokinesis. *Molecular Biology of the Cell*, 13:3532–3545, 2002.

M. Tsuneoka, Y. Koda, M. Soejima, K. Teye, and H. Kimura. A novel Myc target gene, mina53, that is involved in cell proliferation. *Journal of Biological Chemistry*, 277(38):35450–35459, 2002.

L. Zou and S. J. Elledge. Sensing DNA Damage Through ATRIP Recognition of RPA-ssDNA Complexes. *Science*, 300:1542–1548, 2003.