Published in final edited form as:

IEEE/ACM Trans Comput Biol Bioinform. 2009 ; 6(2): 190–199. doi:10.1109/TCBB.2008.139.

A Framework for Multiple Kernel Support Vector Regression and Its Applications to siRNA Efficacy Prediction

Shibin Qiu and

Pathwork Diagnostics Inc., Sunnyvale, CA, USA. sqiu@pathworkdx.com, sqiu@cs.unm.edu

Terran Lane

Department of Computer Science, University of New Mexico, Albuquerque, NM, 87131, USA.

terran@cs.unm.edu

Abstract

The cell defense mechanism of RNA interference has applications in gene function analysis and promising potentials in human disease therapy. To effectively silence a target gene, it is desirable to select appropriate initiator siRNA molecules having satisfactory silencing capabilities. Computational prediction for silencing efficacy of siRNAs can assist this screening process before using them in biological experiments. String kernel functions, which operate directly on the string objects representing siRNAs and target mRNAs, have been applied to support vector regression for the prediction and improved accuracy over numerical kernels in multidimensional vector spaces constructed from descriptors of siRNA design rules. To fully utilize information provided by string and numerical data, we propose to unify the two in a kernel feature space by devising a multiple kernel regression framework where a linear combination of the kernels are used. We formulate the multiple kernel learning into a quadratically constrained quadratic programming (QCQP) problem, which although yields global optimal solution, is computationally demanding and requires a commercial solver package. We further propose three heuristics based on the principle of kernel–target alignment and predictive accuracy. Empirical results demonstrate that multiple kernel regression can improve accuracy, decrease model complexity by reducing the number of support vectors, and speed up computational performance dramatically. In addition, multiple kernel regression evaluates the importance of constituent kernels, which for the siRNA efficacy prediction problem, compares the relative significance of the design rules. Finally, we give insights into the multiple kernel regression mechanism and point out possible extensions.

Keywords

Multiple kernel learning; multiple kernel heuristics; support vector regression; QCQP optimization; RNA interference; siRNA efficacy

1 Introduction

RNA interference (RNAi) is a cell defense mechanism that represses the expression of viral genes by recognizing and destroying their mRNAs, preventing them from being translated into proteins. The gene *knockdown* in RNAi is induced by short interfering RNA (siRNA) of ~21 nucleotides (nt) long, processed from a double stranded RNA (dsRNA) by the enzyme

Dicer, or transfected directly. Target mRNA transcripts that are hybridized with the siRNA are destroyed by the RNA-induced silencing complex (RISC) [1], as shown in Figure 1. RNAi has widespread applications in biology and great potentials in disease therapy [2]. Although a dsRNA of a few hundred nucleotides long is introduced into plants and insects to initiate RNAi, dsRNAs longer than ~50 bp activate the interferon (IFN) pathway in mammalian cells [3]. Therefore, siRNA molecules are directly transfected in mammals to achieve specific gene knockdown. Due to therapeutical applications, siRNA silencing efficacy becomes the focuses of many biological and computational research [4].

One goal of a gene knockdown is to maximally silence the target gene. Unfortunately, if not carefully selected, an siRNA may lead to unsatisfactory or even unobservable target degradation, since siRNAs targeting different sites on the same mRNA have widely differing effectiveness and more than 70% of arbitrary siRNAs are not functional [5]. Therefore, it is crucial to predict the silencing *efficacy*, which is defined as the percentage of mRNA knockdown caused by an siRNA sequence, before using it in biological and therapeutical experiments. In this work, we develop a multiple kernel support vector learning model and apply it to the silencing efficacy prediction problem.

siRNA design rules characterizing structural and thermodynamic properties of the sequences have been proposed in the form of multidimensional numerical descriptors [5], [6], [7], [8], [9], [10], which lead to input spaces for learning models [11], [12]. Vector spaces can also be constructed using nucleic motifs [9] and subsequence patterns [13], [14]. Sparse encodings are often used to represent the sequences in a vector space [9], [15]. Using a threshold on the efficacy, 75% for example, an siRNA can be categorized into two classes: functional, if its efficacy is at least 75%, and non-functional otherwise. Then classification algorithms, such as SVM [11], [13], [14], decision tree [8], and neural network [10], can be applied. If an SVM is trained in a vector space, we call the kernel function (e.g Gaussian kernel) a numerical kernel as distinguished from a string kernel that operates directly on string objects.

To more accurately predict the efficacy than classification, regression methods have been used. Huesken *et al.* employed a neural network and reported prediction correlation coefficient, but error rates, such as mean squared error (*MSE*), were not shown [9]. Moreover, neural networks are trained by gradient search, which depends on initial values and does not guarantee global optimality. Vert *et al.* used a regularized linear regression to predict the efficacy and selected significant subsequences [15]. However, this work only used linear models and did not exploit nonlinearity. Since a support vector machine employs structural risk minimization and its convex programming leads to global optimization, it has better generalization capabilities than other learning models [16]. Qiu and Lane constructed multidimensional vector spaces from descriptors of siRNA design rules to employ numerical kernels in support vector regression and achieved significant accuracies in the efficacy prediction [12], [17]. They also developed and applied string kernels to the prediction and achieved higher accuracies than the numerical kernels [18], [19], [12], [17].

Since the similarities measured by a string kernel and a numerical kernel from rule descriptors model different aspects of the siRNA samples, they contain different information

and may even complement each other. Conceivably, learning models can be improved by utilizing information from both the string and the descriptors. However, the siRNAs are presented as strings, and the descriptors, vectors in an Euclidean space. There is not a direct way to combine these two types of data. We propose a multiple kernel regression framework to unify the information in the kernel feature space, where a linear combination of string and numerical kernels are used. Multiple kernel SVM classification has been studied by Lanckriet *et al.* [20]. Since regression is generally more complicated than classification and support vector regression has a more complex formulation than its classification counterpart, a multiple kernel regression model must be developed separately. We formulate the multiple kernel learning into a quadratically constrained quadratic programming (QCQP) problem. Empirical results demonstrated that multiple kernel regression improved prediction performance and simplified model complexity by reducing the number of support vectors. Although the QCQP formulation yields global optimal solution, it is computationally inefficient and requires a commercial solver. Therefore, we further propose heuristics for multiple kernel learning to speed up computation and simplify usage.

We measure the fitness between a kernel and target labels to develop our heuristics. One of our heuristic is developed from kernel–target alignment, which was proposed for single kernel classification [21]. We derive the other two heuristics from predictive accuracy. Tests on four biological data sets demonstrate that the heuristics for multiple kernel regression improve predictive accuracy and speed up computational performance. Furthermore, it gives insights into the kernel combination, which provides an additional benefit of comparing the relative significance of the design rules. Although the heuristics are developed for regression, they can easily be adapted for classification.

2 Multiple kernel support vector regression

For clarity of notations, we summarize the formulation of the standard, single kernel support vector (SV) regression before we develop the framework of multiple kernel support vector regression.

2.1 Support vector regression

Suppose we are given a set of l training examples

$$\{(x_1, z_1), (x_2, z_2), \dots, (x_l, z_l)\}$$

where x_i ($1 \leq i \leq l$) is a data point in an input space $\mathcal{X}$, $z_i \in \mathbb{R}$ is its target label. For siRNA sequences, $\mathcal{X}$ is a space of strings, and $z_i \in [0, 1]$ is the silencing efficacy. For siRNA design rules, $\mathcal{X}$ is an Euclidean space, where the coordinates are generated by the descriptors of the rules. We want to learn a regression function

$$f(x) = \omega^T \phi(x) + \beta, \quad (1)$$

that can best predict the label of an unseen data point x , where ω is a weight in the kernel feature space, $\phi(x)$, the kernel feature map of x , and β , a threshold constant. f can be solved through the following dual optimization problem [16].

$$\begin{aligned}
& \max_{\alpha^+, \alpha^-} -\frac{1}{2}(\alpha^+ - \alpha^-)^T K (\alpha^+ - \alpha^-) - \varepsilon \sum_{i=1}^l (\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^l z_i (\alpha_i^+ - \alpha_i^-) \\
& \text{s.t.} \quad \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) = 0 \\
& \quad \alpha_i^+ \in [0, C] \\
& \quad \alpha_i^- \in [0, C] \quad i=1, \dots, l
\end{aligned} \tag{2}$$

where $\alpha^+ = \{\alpha_1^+, \dots, \alpha_l^+\}^T$ and $\alpha^- = \{\alpha_1^-, \dots, \alpha_l^-\}^T$ are dual variables, and $K \in \mathbb{R}^{l \times l}$ is a kernel matrix evaluated from a kernel function $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ with $K_{ij} = k(x_i, x_j)$ [22]. Parameter C adjusts the tradeoff between the regression error and the regularization on f . Solving α^+ , α^- , and β using KKT (Karush-Kuhn-Tucker) conditions applied to (2), the regression function of (1) becomes

$$f(x) = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) k(x, x_i) + \beta, \tag{3}$$

where $f(x)$ depends only on training examples having nonzero coefficients (support vectors) through the representation of the kernel function k . In the regression function in (3), k is usually a single kernel function. Since a combination of kernels with nonnegative coefficients yields a legitimate kernel, we can combine them into one kernel.

Suppose we have a set $\mathcal{K}$ of kernel matrices, each of which is a linear combination of g kernel matrices generated from different kernels or the same type of kernel with different parameters. We want to learn the best combination, in addition to learning the coefficients α^+ , α^- , and β in (3). Since kernel matrices of strong diagonal dominance overfit data and decrease generalization capability [23], we impose a constraint on the trace of the kernel matrices. Therefore,

$$\mathcal{K} = \left\{ K \mid K = \sum_{j=1}^g \mu_j K_j, K_j \succcurlyeq 0, \mu_j \geq 0, \text{trace}(K) = d \right\}, \tag{4}$$

where $\succcurlyeq$ denotes positive semidefiniteness, and d is some positive constant. Thus, the multiple kernel regression can be written as,

$$f(x) = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) \sum_{j=1}^g \mu_j k_j(x_i, x) + \beta. \tag{5}$$

2.2 A QCQP formulation

We define $\gamma = \alpha^+ - \alpha^- \in \mathbb{R}^l$ and $\eta = \alpha^+ + \alpha^- \in \mathbb{R}^l$. Let e denote the vector of all ones, and $z = \{z_1, \dots, z_l\}$, the target labels. To derive a formulation for the multiple kernel SV regression, we employ a min-max approach. According to the dual problem of SV regression in (2), the maximum is searched and achieved in terms of the α variables of the single kernel regression. We add one more set of variables, the μ variables in (4), to determine the kernel

combination coefficients. They are searched by minimizing the objective of the maximization step. Following a similar treatment to that of Lanckriet *et al.* in dealing with classification [20], the problem of kernel selection can be written as

$$\begin{aligned}
 \min_K \quad & \max_{\gamma, \eta} \quad 2z^T \gamma - 2\epsilon e^T \eta - \gamma^T K \gamma \\
 \text{s.t.} \quad & e^T \gamma = 0 \\
 & -C \leq \gamma \leq C \\
 & 0 \leq \eta \leq 2C \\
 & \text{trace}(K) = \delta \\
 & K = \sum_{j=1}^g \mu_j K_j \\
 & K_j \succeq 0 \\
 & \mu_j \geq 0, \quad 1 \leq j \leq g.
 \end{aligned} \tag{6}$$

Since $\max_{\gamma, \eta} 2z^T \gamma - 2\epsilon e^T \eta - \gamma^T K \gamma$ is convex in K , and the above constraints are convex, (6) is a convex optimization problem. However, since $K_j \succeq 0$ is not a linear constraint, the above optimization problem is not a quadratic programming problem.

By introducing intermediate variables and rearranging the objective and constraints in (6), the above multiple kernel support vector regression can be formulated into the following standard form [24].

$$\begin{aligned}
 \min_{\gamma, \eta, \theta_j, v, s, t_j} \quad & v - 2z^T \gamma + 2\epsilon e^T \eta \\
 \text{s.t.} \quad & u = 1 \\
 & at_1 - s = 0 \\
 & e^T \gamma = 0 \\
 & -C \leq \gamma \leq C \\
 & 0 \leq \eta \leq 2C \\
 & t_i - t_1 = 0, \quad 2 \leq i \leq g \\
 & 2vu \geq s^2 \\
 & H_j^T \gamma - \theta_j = 0, \quad j = 1, \dots, g \\
 & t_j \geq \sqrt{\theta_j^T \theta_j}, \quad j = 1, \dots, g
 \end{aligned} \tag{7}$$

where $a = \sqrt{2d/\text{trace}(K_j)}$ when each K_j is normalized and has trace 1, and $H_j H_j^T = K_j$ [24]. $2vu - s^2$ is a rotated conic constraint, and the last two constraints yields $t_j^2 \geq \gamma^T K_j \gamma$, a conic constraint, resulting in a QCQP problem. Using an interior point solver, μ_j can be derived from the dual variable solution of the conic constraints. Additionally, a component kernel corresponding to a larger μ_j is more significant in characterizing the data.

The above QCQP formulation has achieved improvements in prediction accuracy. However, solving (7) requires $O(g^3)$ time and the training process can sometimes be slow, especially when the kernel matrices are dense. It also requires a commercial solver package. We next develop simple heuristics for speedup and ease of use.

3 Heuristics for multiple kernel learning

Since the best prediction accuracy is achieved by a best kernel combination, there must be some fitness between a kernel and its accuracy. We quantify this fitness by exploiting the relationships between a kernel matrix and the target labels to develop our heuristics.

The kernel alignment $F(K_j, z)$ between a kernel matrix K_j and a label set z is defined as [21]

$$F(K_j, z) = \frac{\langle K_j, zz^T \rangle_F}{\sqrt{\langle K_j, K_j \rangle_F \langle zz^T, zz^T \rangle_F}}, \quad (8)$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product, i.e. $\langle A, B \rangle_F = \sum_{i=1}^l \sum_{j=1}^l a_{ij} b_{ij}$ for matrices $A = (a_{ij}) \in \mathbb{R}^{l \times l}$ and $B = (b_{ij}) \in \mathbb{R}^{l \times l}$. Conceptually, $F(K_j, z)$ evaluates a similarity between two matrices in a similar fashion to a normalized inner product between two vectors. If a kernel matrix K_j has a larger alignment to z , it should contribute a larger proportion to the combined kernel. This intuition leads to the following heuristic, which we call the F -heuristic for simplicity

$$\mu_j = \frac{F(K_j, z)}{\sum_{j=1}^g F(K_j, z)}. \quad (9)$$

Additionally, a kernel generating a larger Pearson correlation coefficient R between the predicted labels and the true labels should contribute more to the combined kernel. Let R_j be the R generated by the regressor using kernel matrix K_j . Thus this indirect measure based on R leads to a heuristic, which we call the R -heuristic

$$\mu_j = \frac{R_j}{\sum_{j=1}^g R_j}. \quad (10)$$

Furthermore, a kernel producing a smaller MSE error should contribute more to the combined kernel. Let M_j be the MSE error generated by the regressor using kernel matrix K_j . Our heuristic based on MSE (the M -heuristic) is

$$\mu_j = \frac{\sum_{i=1}^g M_i - M_j}{\sum_{j=1}^g \left(\sum_{i=1}^g M_i - M_j \right)} = \frac{\sum_{i=1}^g M_i - M_j}{(g-1) \sum_{i=1}^g M_i}, \quad (11)$$

where $\sum_{i=1}^g M_i$ is the total MSE error of all kernels, and $\sum_{i=1}^g M_i - M_j$ represents the contribution of K_j . For comparison of the kernel contributions, we normalize, i.e.,

$\sum_{j=1}^g \mu_j = 1$. These heuristics may not always yield global optimal solutions, but they improve prediction accuracy and easy to compute.

4 String kernels and numerical kernels

Previously, it was shown that the RNA and the randomized string kernels performed best [12]. We therefore combine these string kernels with numerical kernels. We first describe these kernels briefly.

4.1 The randomized string kernel

The randomized string kernel (RSK) is constructed based on the locality-sensitive hash [12], [25]. Let s be a point in $\mathcal{A}^p$, the space of strings of length p , where $\mathcal{A} = \{A, C, G, U\}$ is the RNA alphabet. $q < p$ characters from s were picked and concatenated to form a substring of length q . Thus q leads to an index set pointing to q characters of the original string. V index sets were obtained by V repetitions. At the i^{th} repetition, the index set can be denoted as

$$I_q^i = \{j_1^i, j_2^i, \dots, j_q^i\}, \quad (12)$$

where $j_t^i, 1 \leq t \leq q$, are uniformly and uniquely drawn at random from $\{1, 2, \dots, p\}$. The feature map of a string s at the i^{th} repetition $\phi_{q,i}: \mathcal{A}^p \rightarrow \mathcal{A}^q$ is defined as the subsequence formed by concatenating the randomly chosen q characters,

$$\phi_{q,i}(s) = \{s[j_1^i], s[j_2^i], \dots, s[j_q^i]\}. \quad (13)$$

For a collection of index sets $I_q = \{I_q^1, I_q^2, \dots, I_q^V\}$ generated by V repetitions, the total feature map is,

$$\Phi_q(s) = \{\phi_{q,1}(s), \phi_{q,2}(s), \dots, \phi_{q,V}(s)\}. \quad (14)$$

And RSK is defined as

$$k_q^{rsk}(s_1, s_2) = \exp \left\{ -\gamma \sum_{i=1}^V d_H(\phi_{q,i}(s_1), \phi_{q,i}(s_2)) \right\} \quad (15)$$

where $\gamma > 0$ is a width parameter, and $d_H(., .)$ is the Hamming distance. Due to γ , the exponent can also be viewed as the average Hamming distance.

4.2 The RNA string kernel

The RNA string kernel (RNK) simulates RNA hybridization and is defined through the notion of similarity neighborhood [19]. For $s \in \mathcal{A}^p$, its mismatch neighborhood, $N_{m,r}^{mis}(s)$, is defined as all length- p strings δ that differ from s by at most m contiguous mismatches starting at position r in s . Only contiguous mismatches are implemented (with G-U wobbles and bulges ignored), since they are frequently observed in biological experiments [26], [27],

[28]. The mismatch feature map of s is defined as $\Phi_{m,r}^{mis}(s) = \{\phi_\delta(s)\}_{\delta \in \mathcal{A}^p}$, where $\phi_\delta(s) = 1$ if $\delta \in N_{m,r}^{mis}(s)$, and $\phi_\delta(s) = 0$, otherwise.

For an m nt long contiguous mismatch starting at position r , the mismatched indices are $I_{m,r} = \{r, r+1, \dots, r+m-1\}$, whose characters are not compared by the kernel. The substring indices that the kernel compares are in $J_{m,r} = \{1, 2, \dots, p\} \setminus I_{m,r}$. Using $J_{m,r}[i]$ to denote the i^{th} element in $J_{m,r}$, $1 \leq i \leq |J_{m,r}|$, for each mismatch position r , s is mapped into a space of $\mathcal{A}^{p-m}$ by the following feature map,

$$\phi_{J_{m,r}}(s) = \{s[J_{m,r}[1]], s[J_{m,r}[2]], \dots, s[J_{m,r}[p-m]]\}.$$

Then the feature space induced by the mismatch kernel is spanned by all mismatch positions, $1 \leq r \leq p-m+1$,

$$\Phi_m(s) = \{\phi_{J_{m,1}}(s), \phi_{J_{m,2}}(s), \dots, \phi_{J_{m,p-m+1}}(s)\}.$$

The kernel function between two input strings s_1 and s_2 is evaluated as

$$\begin{aligned} k_m^{rnk}(s_1, s_2) &= \langle \Phi_m(s_1), \Phi_m(s_2) \rangle \\ &= \sum_{r=1}^{p-m+1} \langle \phi_{J_{m,r}}(s_1), \phi_{J_{m,r}}(s_2) \rangle_c \\ &= \sum_{r=1}^{p-m+1} \langle \tilde{s}_1, \tilde{s}_2 \rangle, \end{aligned}$$

where $\tilde{s}_1 = \phi_{J_{m,r}}(s_1)$ and $\tilde{s}_2 = \phi_{J_{m,r}}(s_2)$ are the maps of s_1 and s_2 . $\langle \cdot, \cdot \rangle_c$ is calculated number of common characters, $\langle \tilde{s}_1, \tilde{s}_2 \rangle_c = \sum_{i=1}^{p-m} I(\tilde{s}_1[i] = \tilde{s}_2[i])$, where $I(\cdot)$ is an indicator function, i.e.,

$$I(\tilde{s}_1[i] = \tilde{s}_2[i]) = \begin{cases} 1 & \text{if } \tilde{s}_1[i] = \tilde{s}_2[i] \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Equivalently, the kernel function is computed as

$$k_m^{rnk}(s_1, s_2) = \sum_{r=1}^{p-m+1} \sum_{i=1}^{p-m} I(\tilde{s}_1[i] = \tilde{s}_2[i]), \quad (17)$$

which is normalized as

$$\hat{k}_m^{rnk}(s_1, s_2) = \frac{k_m^{rnk}(s_1, s_2)}{\sqrt{k_m^{rnk}(s_1, s_1) k_m^{rnk}(s_2, s_2)}}.$$

4.3 Numerical kernels from siRNA design rules

We found that on all data sets with all the rules, the Gaussian kernel performed the best. We therefore only need to construct the vector spaces using the siRNA design rules.

Ui-Tei's rule—Ui-Tei et al. investigated the relationship between siRNA sequence and RNAi effectiveness in human, Chinese hamster, mouse, and *Drosophila* cells, and presented four criteria describing functional siRNAs capable of reducing target mRNA by 70% or higher [7]. These criteria are: (i) an A/U residue at the 5' end of the antisense strand; (ii) G/C at the 5' end of the sense strand; (iii) at least five A/U residues in the 5' terminal one-third of the antisense strand; and (iv) the absence of any GC stretch of more than nine nt in length.

Amarzguioui's rule—Amarzguioui and Prydz studied siRNA silencing efficacy in human and mouse cells and proposed properties to characterize functional and non-functional siRNA sequences [29]. They defined functional siRNAs as those capable of performing 70% or higher target reduction. Their criteria are: (1) $\Delta T_3 = T_3 - T_5$, the difference between the number of A/U residues in the three terminal positions at the 3' end and at the 5' ends (relative to the sense strand of the siRNA). $\Delta T_3 > 1$ is positively correlated (correlated with functional siRNA); (2) S_1 – a G or C residue at position 1, positively correlated; (3) U_1 , a U residue at position 1, negatively correlated; (4) A_6 , an A residue at position 6, positively correlated; (5) W_{19} , A or U at position 19, positively correlated; (6) G_{19} , G at position 19, negatively correlated. They also found that a G-C content of 31.6–57.9% was preferable.

Reynolds' rule—Reynolds et al. examined siRNAs targeting the firefly luciferase and human cyclophilin genes and discovered eight conditions for functional siRNA sequences, where being functional is defined as being able to reduce targets by 80% [30]. These conditions are summarized in Table 1. The third condition, absence of internal repeats, is imposed to eliminate the possibility of forming a short hairpin structure on the siRNA sequence. This condition is equivalent to requiring the sequence to have a low melting temperature (T_m).

Jagla's rules—Jagla et al. uncovered four rule sets using decision tree algorithms on 600 chemically modified 21-mer duplexes including 2 nt overhang targeting human genes [8]. Functional siRNAs are those having 60% knockdown efficacy or higher. These rules are listed in Table 2. We only use the criteria in Jagla's first rule.

Huesken's motifs—Huesken et al. trained a neural network using more than 2400 siRNAs targeting human and rodent genes, and discovered a collection of important motifs characterizing the most active and most inactive siRNA sequences [9]. They consider single nucleotide motifs of statistical significance (p -value < 0.05) as strong contributions to siRNA efficacy. The p -values were computed by comparing the background frequencies with the overrepresented frequencies using a binomial distribution. Table 3 lists the motifs represented as IUPAC codes in the first ten positions on antisense strand.

We use sparse encoding to map each siRNA sequence into a vector space. Let s denote an siRNA sequence of 19 nt long. Consider the first condition of Ui-Tei's rule, which requires the existence of an A/U residue at the 5' end of the antisense strand, i.e., an A/U base at

position 19 of the sense strand. We use two binary attributes to encode each criterion. If position 19 of s is an A or U base, we generate a pair of (1, 0) for this criterion. If position 19 of s is either a G or a C base, then we generate a (0, 1) to encode this condition. Other conditions can be similarly encoded. Since Ui-Tei's rule has four conditions, we have eight such binary descriptors. This process of sparse encoding maps each sequence into an eight dimensional vector space. On the JA data set, which has 600 sequence samples, we have a data matrix of 600×8 . Let us denote the vector space as $\mathcal{X}$. Then, $\mathcal{X}$ is a subspace of an eight dimensional Euclidean space, $\mathcal{X} \subseteq \mathbb{R}^8$.

Applying the same sparse encoding to Reynolds' rule, $\mathcal{X} \subseteq \mathbb{R}^{16}$. For Jagla's first group of rules, $\mathcal{X} \subseteq \mathbb{R}^8$. For Huesken's motifs, $\mathcal{X} \subseteq \mathbb{R}^{76}$. Finally, we encode Amarzguoui's rules using a 9 dimensional space, the three extra descriptors being T_3 , T_5 , and G-C content.

5 Empirical evaluations

We test the performance of the heuristics on biological data sets.

5.1 Data sets and kernel combinations

We first summarize the data sets. The 73 siRNA sequences targeting firefly luciferase gene and human cyclophilin B mRNA were originally used to develop Reynolds' rule [6]. We use 70 samples from this data source and call it the KR data set. Sætrom *et al.* collected 560 data points for training their SVM classifiers [11]. A small number of this data set may overlap with the KR data set. We call this data set SA data set. Jagla *et al.* tested their decision tree algorithm on 600 data points [8]. We call this data set JA data set. Huesken *et al.* studied more than 2400 siRNAs targeting 34 human and rodent mRNAs [9]. We call this 2400 siRNAs HU data set. Each data point in the KR, SA, and JA data sets contain a 19-mer siRNA sequence and its corresponding efficacy score, whereas the HU data set has samples of 21 nt long.

To use the heuristics for multiple kernel regression, we first test accuracies of single kernel regressions using numerical kernels according to the efficacy rules, the RNA and the RSK string kernels. We next combine the numerical kernels with the string kernels using the weights generated by our F -heuristic, R -heuristic, and M -heuristic introduced in Section 3. The multiple kernel approach is illustrated in Figure 2. We report correlation coefficients (R) and MSE error rates between predicted and target labels of 10 fold cross validations on the data sets.

5.2 Prediction accuracies

We found that on each data set, using each heuristics, combining numerical kernels from design rules and string kernels achieved higher accuracy. We show improvements in R in Table 4 and MSE in Table 5. The overall accuracies of the F -heuristic is shown in Figure 3 and its accuracies on the HU data set is displayed in Table 6, where accuracies of numerical kernels of the design rules and numerical kernel combined with the RNA and the RSK kernels are shown. On average, combined kernels generated an average R of 0.616, an

increase of R by 24.4%, and an MSE of 0.0251, a decrease by 16.7%, over single kernel results.

Figure 3 shows accuracies of multiple kernel regression using the F -heuristic averaged over all data sets and all rules. Performance enhancements using the R -heuristic and the M -heuristic are similar to those shown in Figure 3, thus not shown separately. The average R using numerical kernels was 0.469. In contrast, the average R using numerical kernels combined with the RNA kernels was 0.538, representing a 14.7% increase over single-numerical kernels. On the other hand, the average R using numerical kernels combined with the RSK kernels was 0.535, representing a 14.1% increase. These increases were statistically significant, with p-values of 0.003 and 0.007, respectively, as displayed in Table 4.

We also noticed that the standard deviations decreased when combined kernels were used. This increase in R demonstrates that combining the numerical kernel generated from an efficacy rule and an RNA string kernel or an RSK kernel using kernel alignment as weight, significantly improved prediction accuracy over using the numerical kernel alone. Meanwhile using combined kernels also yielded better accuracies than using the RSK kernel alone, although RSK performed the best among the string kernels.

Using the R -heuristic, the average R using numerical kernels combined with the RNA kernels was 0.537, a 14.7% increase over single-numerical kernels as shown in Table 4. Additionally, the average R on all data sets using numerical kernels combined with the RSK kernels was 0.533, a 13.9% increase. These increases were statistically significant, with p-values of 0.003 and 0.01, respectively (Table 4). Meanwhile using combined kernels also yielded better accuracies than using the RSK kernel alone.

Using the M -heuristic, the average R using numerical kernels combined with RNK was 0.538, achieving a 14.7% increase over single kernel regressions (Table 4). Also, the average R using numerical kernels combined with RSK was 0.530, a 13.0% increase. These increases were statistically significant, with p-values of 0.002 and 0.013 respectively. Furthermore, using combined kernels also yielded better accuracies than using the RSK alone.

Using combined kernels reduced MSE error rates as well. For example, the average MSE on all data sets using the numerical kernels was 0.0671, whereas it was 0.0614, 0.0623, and 0.0613 using numerical kernels combined with the RNA string kernels, using the F , R and M heuristics, representing decreases of 8.5%, 7.2%, and 8.6%, respectively (Table 5). It seemed that the F -heuristic and the M -heuristic performed better than the R -heuristic, when MSE reduction is concerned. Since the models were tuned based on correlation coefficients, improvements on MSE were less significant than those on R . However, if MSE is more critical than R , as possibly required by a particular design, we can re-train the models to generate more reductions on MSE .

We noticed that the average R generated by multiple kernel on the HU data set was around 0.62 (highest 0.63), as displayed in Tables 6, whereas it was about 0.54 on all data sets, as shown in Figure 3. This lower average on all data sets is because performance of all methods on the smaller data sets, especially on KR, was rather poor. Another reason is that we did

not perform data preprocessing, such as outlier removal and subdivision, as is frequently done in the literature [9]. Therefore, when the average accuracy on all data sets are concerned, we should focus on relative improvements, which have been shown to be significant (Tables 4 and 5). However, higher accuracies were achieved on individual data sets, especially on larger data sets, as shown in Tables 6 on the HU data set, where R was increased by 24.4% and MSE decreased by 16.7%. Performances of the heuristics on the SA and JA data sets were similar, but not shown separately.

5.3 Which design rule is more informative?

When performing multiple kernel regression using the heuristics, μ_1 represents the contributions of the numerical kernels corresponding to the design rules, and μ_2 represents the contributions of the string kernels, as shown in Figure 2. With a little abuse of notation, this can be represented as $k = \mu_1 k(\text{numerical}) + \mu_2 k(\text{string})$.

We rank the rules according to $\overline{\mu_1}$, average of μ_1 (over heuristics), which is the average of μ_1 over the data sets, as shown in Table 7. Although the values of $\overline{\mu_1}$ generated by different design rules appeared to be close to each other, their differences can be statistically significant due to small variances. A design rule with a larger $\overline{\mu_1}$ is more informative in describing the siRNA sequences and predicting the efficacy. This ranking suggests that Huesken's motif is most significant, followed by Amarzguioui's rule, Jagla's first rule, Ui-Tei's rule, and Reynolds' rule. This ranking helps select appropriate design rules for gene knockdown.

5.4 Computational performance

We show how the heuristics improved computational performance over the QCQP formulation. Table 8 shows running times for solving the QCQP optimization and computing the F -heuristics to combine a numerical kernel based on Amarzguioui's rule and the RNA string kernel for 10 fold cross validation.

To solve the QCQP, we called the MOSEK solver from Matlab. To avoid heavy computational burden on Matlab, we computed the kernel matrices in advance. The SV regression using the heuristics are solved by LIBSVM, a free SVM solver [31]. As shown, the heuristics is 241 times faster than the QCQP. On the largest HU data set, for example, QCQP used 2.46 hours, whereas the heuristics only used 33 seconds.

6 Summary and discussion

The siRNA molecules are intrinsically sequences, which are natively represented as strings. The design rules, on the other hand, extract numerical descriptors from the sequences. Previously, we demonstrated that string kernels achieved better prediction accuracies than design rules. To further improve prediction performance, we proposed to use multiple kernel support vector regression by combining the numerical kernels and the string kernels. Our QCQP formulation achieved optimal solutions and achieved better accuracies. But it is slow in training. For computational efficiency, we introduced three heuristics. In a typical case,

the heuristics were more than 200 times faster than solving the QCQP. We experimented our heuristic framework on the siRNA efficacy data sets.

We found combining numerical kernels computed from efficacy rules and string kernels significantly increased prediction performances over using the numerical kernels alone. Using the heuristics, improvements on the correlation coefficients (R) were around 24% and improvements on MSE were around 17%, on the larger data sets, where accuracies are stable. Overall on all data sets, the improvements on the correlation coefficients were around 14% and improvements on MSE were around 8%. We found that the heuristics based on kernel alignment and on correlation coefficient gave rise to best combining weights, when enhancement on correlation coefficient is concerned. On the other hand, if reduction on MSE is crucial, the heuristics based on alignment and on MSE error are better choices.

In addition to accuracy improvement, we found that using multiple kernels reduced the number of support vectors required in the regression model (details not shown). Since a complicated data set usually requires more support vectors [16], [32], SV reduction by multiple kernel regression equivalently transforms a difficult problem into an easier one that needs fewer SVs. Therefore multiple kernel regression has a better ability to model complex data and are adaptive to more difficult problems.

According to the contribution to the combined kernel made by a numerical kernel generated by an efficacy design rule, we found that the best two rules are Huesken's motifs and Amarzguioui's rule. Huesken's motifs are based on statistically significant single nucleotide motifs corresponding to efficient and inefficient siRNA sequences, and cover a wide range of important occurrences. Amarzguioui's rule also covers many descriptions for siRNA sequences. This rule comparison can help RNAi designers select the appropriate rules. While more descriptors in a rule provide more insights into the efficacy, as is true with Huesken's motifs, the number of criteria is not always the dominant factor. For example, Jagla's first rule, which has only four conditions, performed better than Reynolds' rule of eight criteria. It is also possible to combine subsets of conditions from multiple rules to derive a hybrid rule which will be able to perform better than any single rule. This composition of conditions can be done, for example, by methods of feature selection.

Prediction of siRNA silencing efficacy can help select and design efficient RNAi initiator sequences. Since the problem is hard, overall prediction accuracies are not very high. This is true both in our work and in the literature in general. Nonetheless, our methods have significantly improved the accuracy. Previously, we performed both cross validation and independent validation as required by more sophisticated learning designs, where we used a major portion of the data for model tuning using cross validation and held out a relatively smaller part for independent model validation [12], [17]. We found that performance of independent validation was very close to that of cross validation, which was especially true on the larger data sets. Therefore, our cross validation results in this work are consistent and reliable, although we only conducted cross validation without independent validation.

Although combined kernels improve prediction accuracy over single kernels, the improvements depend on which kernels are combined. Rules that performed well, such as

Huesken's motifs and Amarzguioui's rule, already captured crucial properties in the sequences. Combining their numerical kernels with string kernels did not improve performance as drastically as combining less well performed rules. For example, a Gaussian kernel using Amarzguioui's rule yielded an average (over the four data sets) R of 0.497. An average R of 0.537 was obtained by combining it with RNK using the F -heuristic, and an average R of 0.518, when combined with RSK. These two improvements only increased the correlation coefficients by 8% and 4%, respectively. In contrast, Reynolds' rule, which did not perform so well alone, improved performance dramatically when combined with a string kernel. For instance, a Gaussian kernel on Reynolds' rule yielded an average R of 0.354. And combining it with RNA and RSK yielded average R of 0.529 and 0.526, increasing R by 47% and 46%, respectively, much higher than those gained by combining Amarzguioui's rule. It seems that more accuracy improvement can be obtained when diverse and disparate kernels are combined.

Since support vector regression employs a sparse representation of the training samples, each support vector covers a local region through a kernel function. The shape of the kernel controls the coverage of the local region and the contributions of the support vectors nearby. Thus distinctive kernels control localities in different ways. Although cross validation finds the best single kernel parameter, such as the width parameter (σ) of a Gaussian kernel, this single kernel with its best parameter might not be able to perfectly control the locality of the data. As shown in Figure 4, an exponential kernel might cover too wide a neighborhood around a support vector and might not emphasize its closest neighbors enough due to its flat tail and low shoulder. A Gaussian kernel, however, might just do the opposite. And a rectangle kernel, on the other hand, might only conserve a small neighborhood and exclude too much [33]. In practice, a data set is noisy and might not be generated by a pure process, making it hard for a single kernel function to optimally describe its locality. Using a combination of kernels increases the chance of better locality control, which in turn results in more accurate prediction. Although the shape of a combined kernel may not look perfect, it can nonetheless better handle the contributions of the support vectors. This stronger capability of locality description and control is consistent with support vector reduction, which has been observed in multiple kernel regression. This insight of locality control is also consistent with the observation that combining kernels with different properties leads to better accuracy improvement.

We investigated support vector regression models which have unique optimal solution due to convex optimization and have better generalization capabilities, whereas related work used artificial neural networks whose solutions depend on initial values and do not guarantee global optimality [9]. Historically, we first used support vector regression on the efficacy prediction problem. We performed 10 fold cross validation which generates reliable results and is widely accepted in the community. In contrast, some related work used fixed test set to evaluate accuracy of their methods, increasing the chance of overfitting and over estimate [9]. We tested our methods on a large collection of samples from diverse sources without any preprocessing, whereas most related work only tested on one data set after certain preprocessing such as outlier removal and subdivision of the data into smaller groups so that better performance can be obtained on the groups. We not only reported values of R , but

also the *MSE* rates. We made this choice since we noticed that some packages only print the absolute value of *R* and ignore its sign, generating the possibility of unreliable results. As a comparison, the values of *R* were between 0.47 to 0.66 with a mean of 0.61 and a standard deviation of 0.061 in the largest all-all group reported by Huesken et al. [9], whereas our numerical combined with RNA kernel using the *M*-heuristic generated the *R* between 0.61 to 0.63 with a mean of 0.62 and a standard deviation of 0.007. Even though our highest *R* is lower than theirs, our average *R* is higher and our standard deviation is smaller. Furthermore, our results are based on 10 fold cross validation without any preprocessing or subdivision.

Although we have demonstrated that the heuristics are much faster than the QCQP formulation, we have not compared their prediction accuracies. It is conceivable that the heuristics cannot outperform their true optimal counterparts. In the future, we will investigate the extent to which the heuristics would lose their accuracies by applying the two methods on the same data sets. This analysis would provide us with guidelines and tradeoffs between accuracy and computational performance.

Although our heuristics are developed for support vector regression, they can be extended to classification. The *F*-heuristic and the *R*-heuristic are applicable to a classification scenario if continuous labels are replaced with binary labels and signs of the measures are handled properly. Additionally, classification error rates can be used to construct a heuristic. Let E_j be the classification error rate yielded by the *j*th kernel, then the *E*-heuristic for classification can be built as,

$$\mu_j = \frac{\sum_{i=1}^g E_i - E_j}{\sum_{i=1}^g \left(\sum_{i=1}^g E_i - E_j \right)} = \frac{\sum_{i=1}^g E_i - E_j}{(g-1) \sum_{i=1}^g E_i}. \quad (18)$$

We believe the multiple kernel framework is applicable to other applications where data have heterogeneous components. One example is disease diagnosis, where microarray, clinical, and genetic data are available yet heterogeneous. The kernel combination may possibly increase accuracy and give diagnostic insights. Additionally, combining sequence similarity and similarities of other numerical aspects of the individual samples, it is possible to increase the accuracy of clustering in the kernel feature space, which might be useful in unsupervised data description and visualization.

Acknowledgement

This work was supported by NIH grant 1P20RR18754 from the Institutional Development Award (IDeA) Program of NCRR. Dr. Lane's work was also partially supported by NIMH grant 1R01MH076282-01 of the NSF/NIH Collaborative Research in Computational Neuroscience Program.

Biography

Shibin Qiu Shibin Qiu is Senior Scientist at Pathwork Diagnostics Inc, Sunnyvale, California, USA. He received his MS degree from Purdue University, Indianapolis, IN, 2001. He received his Ph.D. degree in 2006 from The University of New Mexico, Albuquerque, NM, where he studied computer engineering, machine learning, statistics, and mathematics. His research interests include machine learning, data mining, bioinformatics, and high performance computing. He is also interested in developing and implementing algorithms for commercial products.

Terran Lane Terran Lane is an assistant professor at the University of New Mexico Department of Computer Science. Dr. Lane received his Ph.D. in 2000 from Purdue University, where he studied machine learning algorithms for computer security applications. Subsequently, he spent two years at the MIT AI LAB, where he worked on planning and learning techniques for stochastic control domains representable as Markov decision processes. His current interests include machine learning and artificial intelligence in general, but specifically applications to bioinformatics; neuroinformatics; reinforcement learning and control; time series analysis; user modeling; and security applications.

REFERENCES

- [1]. Hannon GJ. RNA interference. *Nature*. Jul.2002 418:244–251. [PubMed: 12110901]
- [2]. Check E. Hopes rise for RNA therapy as mouse study hits target. *Nature*. 2004; 432:136. [PubMed: 15538330]
- [3]. Brummelkamp TR, Bernards R, Agami R. A system for stable expression of short interfering RNAs in mammalian cells. *Science*. 2002; 296:550–553. [PubMed: 11910072]
- [4]. Pei Y, Tuschl T. On the art of identifying effective and specific siRNA. *Nature Methods*. 2006; 3(no. 9):670–676. [PubMed: 16929310]
- [5]. Amarzguioui M, Prydz H. An algorithm for selection of functional siRNA sequences. *B.B.R.C.* 2004; 316:1050–1058.
- [6]. Reynolds A, Leake D, Boese Q, Scaringe S, Marshall WS, Khovorova A. Rational siRNA design for RNA interference. *Nature Biotechnology*. 2004; 22:326–330.
- [7]. Ui-Tei K, Naito Y, Takahashi F, Haraguchi T, Ohki-Hamazaki H, Juni A, Ueda R, Saigo K. Guidelines for the selection of highly effective siRNA sequences for mammalian and chick RNA interference. *Nucleic Acids Research*. 2004; 32:936–948. [PubMed: 14769950]
- [8]. Jagla B, Aulner N, Kelly P, Song D, Volchuk A, Zatorski A, Shum D, Mayer T, Angelis DD, Ouerfelli O, Rutishauser U, Rothman J. Sequence characteristics of functional siRNAs. *RNA*. Jun.2005 11:864–872. [PubMed: 15923373]
- [9]. Huesken D, Lange J, Mickanin C, Weiler J, Asselbergs F, Warner J, Meloon B, Engel S, Rosenberg A, Cohen D, Labow M, Reinhardt M, Natt F, Hall J. Design of a genome-wide siRNA library using an artificial neural network. *Nature Biotechnology*. 2005; 23(no. 8):995–1001.

- [10]. Ge G, Wong G, Luo B. Prediction of siRNA knockdown efficacy using artificial neural network models. *Biochem Biophys. Res. Comm.* 2005; 336:723–728. [PubMed: 16153609]
- [11]. Sætrom P, Snøve O Jr. A comparison of siRNA efficacy predictors. *Biochemical and Biophysical Research Communications.* 2004; 321:247–253. [PubMed: 15358242]
- [12]. Qiu, S.; Lane, T.; Buturovic, L. A randomized string kernel and its applications to RNA interference. *Proc. 22 AAAI Conference on Artificial Intelligence*; Vancouver, BC, Canada. July 2007; AAAI Press; p. 627–632.
- [13]. Teramoto R, Aoki M, Kimura T, Kanaoka M. Prediction of siRNA functionality using generalized string kernel and support vector machine. *FEBS Lett.* 2005; 579:2878–2882. [PubMed: 15878553]
- [14]. Jia P, Shi T, Cai Y, Li Y. Demonstration of two novel methods for predicting functional siRNA efficiency. *BMC Bioinformatics.* 2006; 7:271. [PubMed: 16729898]
- [15]. Vert J-P, Foveau N, Lajaunie C, Vandenbrouck Y. An accurate and interpretable model for siRNA efficacy prediction. *MBC Bioinformatics.* 2006; 7:520.
- [16]. Vapnik, VN. *Statistical Learning Theory.* John Wiley and Sons; 1998.
- [17]. Qiu, S.; Lane, T. The RNA string kernel for siRNA efficacy prediction. *Proc. 7th IEEE Int'l Conf. on Bioinformatics and Bioengineering (BIBE2007)*; Boston, MA. October 2007; p. 307–314.
- [18]. Qiu S, Adema C, Lane T. A computational study of off-target effects of RNA interference. *Nucleic Acids Research.* 2005; 33:1834–1847. [PubMed: 15800213]
- [19]. Qiu S, Lane T. RNA string kernels for RNAi off-target evaluation. *Int. J. Bioinformatics Research and Applications (IJBRA).* 2006; 2(no. 2):132–146.
- [20]. Lanckriet GRG, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI. Learning the kernel matrix with semidefinite programming. *J. Machine Learning Research.* 2004; 5:27–72.
- [21]. Cristianini, N.; Shawe-Taylor, J.; Elissee, A.; Kandola, J. On kernel-target alignment. In: Dietterich, T.; Becker, S.; Ghahramani, Z., editors. *Advances in Neural Information Processing Systems.* Vol. 14. MIT Press; 2002.
- [22]. Smola A, Schölkopf B. A tutorial on support vector regression. *NeuroCOLT2.* 1998 Tech. Rep. NC2-TR-1998-030.
- [23]. Weston, J.; Schölkopf, B.; Eskin, E.; Leslie, C.; Noble, WS. A kernel approach for learning from almost orthogonal patterns. *Proc. 6th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'02)*; Helsinki, Finland. Aug. 2002; Springer; LNCS 243
- [24]. Qiu, S.; Lane, T. Computer Science Department. The University of New Mexico; Albuquerque, NM, USA: 2005. Multiple kernel learning for support vector regression. Tech. Rep. TR-CS-2005-42
- [25]. Indyk, P.; Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality; *Proc. 30th Ann. ACM Symp. on Theory of Computing*; Dallas, TX. 1998. p. 604–613.
- [26]. Elbashir SM, Martinez J, Patkaniowska A, Lendeckel W, Tuschl T. Functional anatomy of siRNA for mediating efficient RNAi in *Drosophila melanogaster* embryo lysate. *The EMBO Journal.* 2001; 20(no. 23):6877–6888. [PubMed: 11726523]
- [27]. Saxena S, Jonsson ZO, Dutta A. Small RNAs with imperfect match to endogenous mRNA repress translation. *Journal of Biological Chemistry.* 2003; 278(no. 45):44 312–44 319.
- [28]. Jackson A, Bartz S, Schelter J, Kobayashi S, Burchard J, Mao M, Li B, Cavet G, Linsley PS. Expression profiling reveals off-target gene regulation by RNAi. *Nature Biotechnology.* 2003; 21(no. 6):635–637.
- [29]. Amarzguioui M, Prydz H. An algorithm for selection of functional siRNA sequences. *Biochemical and Biophysical Research Communications.* 2004; 316:1050–1058. [PubMed: 15044091]
- [30]. Reynolds A, Leake D, Boese Q, Scaringe S, Marshall WS, Khovorova A. Rational siRNA design for RNA interference. *Nature Biotechnology.* 2004; 22:326–330.
- [31]. Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. 2001 software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

- [32]. Ma J, Theiler J, Perkins S. Accurate online support vector regression. *Neural Computation*. 2003; 15:2683–2703. [PubMed: 14577858]
- [33]. Qiu, S.; Lane, T. Parallel computation of RBF kernels for support vector classifiers. *Proc. 5th SIAM International Conference on Data Mining (SDM05)*; Newport Beach, CA. April 2005; SIAM; p. 334-345.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Fig. 1.

The RNAi mechanism. (1) A dsRNA is introduced to initiate RNAi; (2) Dicer cleaves the dsRNA into siRNAs; (3) RISC takes up the guide strand; (4) RISC-siRNA complex recognizes the target; (5) The target mRNA is degraded by RISC.

Fig. 2. Illustration of multiple kernel regression on siRNA efficacy data set. The multiple kernel learning unit is in the box at the bottom right corner.

Fig. 3.

Comparison of average R over the data sets of single kernel (blank bar) and multiple kernel (filled bar) regression using the F -heuristic. Vertical line represents standard deviation; “A, C”, R of numerical kernel of rules only; “E, G”, RSK kernel; “B, F”, numerical kernel combined with RNA kernel; “D, H”, numerical kernel combined with RSK kernel.

Fig. 4.
Shapes of three kernel functions and their combined kernel.

TABLE 1

Reynolds' rules for functional siRNA sequences.

	Criterion
1	30%-50% G/C content
2	At least 3 'A/U' bases at position 15-19
3	Absence of internal repeats ($T_m < 20^\circ$)
4	An 'A' base at position 19
5	An 'A' base at position 3
6	A 'U' base at position 10
7	A base other than 'G' or 'C' at 19
8	A base other than 'G' at position 13

TABLE 2

Jagla’s rules for functional siRNA sequences.

Group	Condition
1 st	1. position 19=(A/U), 2. position 13-19 >3(A/U) 3. position 1=(G/C), 4. position 10=(A/U)
2 nd	1. position 19=(A/U), 2. position 13-19 >3(A/U) 3. position 1=(G/C), 4. position 10=(G/C)
3 rd	1. position 19=(G/C), 2. position 1=(G/C) 3. position 5-19 >6(A/U), 4. position 11=(G/C)
4 th	1. position 19=(A/U), 2. position 13-19 >3(A/U) 3. position 1=(A/U)

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE 3

Huesken’s motifs. “Pos” denotes antisense strand position.

Active motifs			Inactive motifs		
Pos	Code	Significance	Pos	Code	Significance
1	A	4.5E-06	1	B	9.7E-06
1	D	8.8E-07	1	C	1.5E-05
1	H	8.9E-16	1	G	1.6E-10
1	T	4.9E-11	1	R	1.3E-02
1	W	2.9E-27	1	S	4.3E-25
1	Y	1.9E-02	1	V	2.7E-16
2	H	5.3E-03	2	C	9.0E-05
2	T	8.0E-04	2	M	2.2E-02
2	W	5.2E-05	2	S	7.4E-06
2	T	8.0E-04	2	M	2.2E-02
3	A	4.9E-04	2	V	1.3E-04
3	D	4.6E-02	4	G	1.0E-02
3	H	2.8E-02	4	S	1.9E-04
3	W	2.9E-04	4	V	1.4E-02
5	T	1.5E-03	5	M	4.8E-02
5	W	1.0E-02	6	A	3.8E-02
6	W	1.6E-02	7	B	1.0E-02
7	D	1.1E-03	7	C	5.7E-05
7	H	3.7E-02	7	S	1.1E-05
7	T	1.7E-03	7	V	9.9E-03
7	W	4.4E-06	10	B	3.6E-03
10	A	9.1E-05	10	K	2.7E-02
10	R	4.3E-02	10	T	4.8E-02
10	W	3.4E-03			

TABLE 4

Increase of R by multiple kernel regression over all data sets. “A–H” indicate types of kernel as explained in Figure 3. “p” denotes t -test p -value when significant.

Single vs. multiple kernel	F -heuristic	R -heuristic	M -heuristic
Rule-only(A) vs. Rule+RNA(B)	14.7% (p: 0.003)	14.7% (p: 0.003)	14.7% (p: 0.002)
Rule-only(C) vs. Rule+RSK(D)	14.1% (p: 0.007)	13.9% (p: 0.01)	13.0% (p: 0.013)
RSK(E) vs. Rule+RNA(F)	4.7%	4.5%	4.7%
RSK(G) vs. Rule+RSK(H)	4.1%	3.7%	3.1%
Average	9.4%	9.2%	8.9%

TABLE 5

Improvement on MSE of multiple kernel regression over numerical kernels (average MSE : 0.0671).

Heuristic	Rule+RNA	ΔMSE	Rule+RSK	ΔMSE
F	0.0614	8.5%	0.0622	7.3%
R	0.0623	7.2%	0.0635	5.4%
M	0.0613	8.6%	0.0630	6.1%
Average	0.0617	8.1%	0.0629	6.3%

TABLE 6

Accuracy of F -heuristic on HU data set by combining numerical kernel, denoted as “Rule-only”, with string kernels. “Rule+RNA” and “Rule+RSK” denote numerical kernel from the rule combined with RNA and RSK string kernel, respectively.

R			
Rule	Rule-only	Rule+RNA	Rule+RSK
Amarzguioui	0.579	0.622	0.624
Huesken	0.601	0.610	0.613
Jagla	0.483	0.612	0.613
Reynolds	0.332	0.612	0.607
Ui-Tei	0.478	0.622	0.620
Average	0.495	0.616	0.615
Stdev	± 0.1064	± 0.0059	± 0.0066

MSE			
Rule	Rule-only	Rule+RNA	Rule+RSK
Amarzguioui	0.0268	0.0247	0.0246
Huesken	0.0257	0.0253	0.0251
Jagla	0.0309	0.0252	0.0252
Reynolds	0.0359	0.0252	0.0256
Ui-Tei	0.0311	0.0248	0.0248
Average	0.0301	0.0250	0.0251
Stdev	± 0.00405	± 0.00027	± 0.00038

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE 7

Comparison of design rule contributions. $\overline{\mu_1}(F)$, $\overline{\mu_1}(R)$, and $\overline{\mu_1}(M)$ denote averages of μ_1 of the F -heuristic, the R -heuristic, and the M -heuristic, respectively.

Rule	$\overline{\mu_1}(F)$	$\overline{\mu_1}(R)$	$\overline{\mu_1}(M)$	$\overline{\mu_1}$	Rank
Amarzguioui	0.4970	0.4834	0.5163	0.4988	2
Huesken	0.5001	0.5075	0.5124	0.5067	1
Jagla	0.4984	0.4733	0.4914	0.4877	3
Reynolds	0.4994	0.4003	0.4591	0.4529	5
Ui-Tei	0.4909	0.4684	0.4855	0.4816	4

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE 8

Running times (in seconds) used by QCQP and the F -heuristic.

	KR	SA	JA	HU	Total	Comparison
QCQP	12	120	180	8850	9162	241
Heuristic	1	2	2	33	38	1