A Novel Framework to Improve siRNA Efficacy Prediction

Bui Thang Ngoc

Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi City, Ishikawa, 923-1211 Japan
thangbn@jaist.ac.jp

Abstract. Short interfering RNA sequences (siRNAs) can knockdown target genes and thus have an immense impact on biology and pharmacy research. The key question of which siRNAs have high knockdown ability in siRNA research remains challenging as current known results are still far from expectation. This work aims to develop a generic framework to enhance siRNA knockdown efficacy prediction. The key idea is first to enrich siRNA sequences by incorporating them with rules found for designing effective siRNAs and representing them as transformed matrices, then to employ the bilinear tensor regression to do prediction on those matrices. Experiments show that the proposed method achieves results better than existing models in most cases.

1 Introduction

In 2006, Fire and Mello received their Nobel Prize for their contributions to research on RNA interference (RNAi) that is the biological process in which RNA molecules inhibit gene expression, typically by causing the destruction of specific mRNA molecules. Their work and that of others on discovery of RNAi have had an immense impact on biomedical research and will most likely lead to novel medical applications. On RNAi research, designing of siRNAs (short interfering RNAs) with high efficacy is one of the most crucial RNAi issues. Highly effective siRNAs can be used to design drugs for viral-mediated diseases such as Influenza A virus, HIV, Hepatitis B virus, RSV viruses, cancer disease and so on. As a result, siRNA silencing is considered one of the most promising techniques in future therapy. Finding highly effective siRNAs among thousands of potential siRNAs for an mRNA remains a great challenge.

Various siRNA design rules have been found by empirical processes since 1998. The first rational siRNA design rule was detected by Elibalshir *et al.* [2]. They suggested that siRNAs having 19–21 nt (nucleotide) in length with 2 nt overhangs at 3' end can efficiently silence mRNAs. Scherer *et al.* reported that the thermodynamic properties (G/C content of siRNA) to target specific mRNAs are important characteristics [11]. Soon after these works, many rational design rules for effective siRNAs have been found, typically those in [10], [15], [1], [4], [7], [14]. For example, Reynolds *et al.* [10] analyzed 180 siRNAs and found eight criteria for improving siRNA selection: (1) G/C content 30−52%, (2) at least 3

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 400–412, 2014.

As or Us at positions from 15 to 19, (3) absence of internal repeats, (4) an A at position 19, (5) an A at position 3, (6) a U at position 10, (7) a base other than G or C at position 19, (8) a base other than G at position 13.

However, most of siRNA design tools using the above-mentioned design rules have low accuracy, because about 65% of the siRNAs predicted as high effective was failed when tested experimentally as they were 90% in inhibition and near 20% of them were found to be inactive [9]. One reason is the previous empirical analyses only based on small datasets and focused on specific genes. Therefore, each of these rules certainly is poor to individually design effective siRNAs.

Since nearly a decade, machine learning techniques have alternatively been applied to predict knockdown efficacy of siRNAs. The first predictive model was proposed by Huesken *et al.* in which motifs for effective and ineffective siRNA sequences were detected basing on the significance of nucleotides by using a neural network to train 2,182 scoring siRNAs (scores are real numbers in [0, 1], the high score the higher knockdown efficacy) and test on 249 siRNAs [5]. This data set was consequently used to build other predictive models [6], [13], [16]. Recently, Qui *et al.* used multiple support vector regression with RNA string kernel for siRNA efficacy prediction [8], and Sciabola *et al.* applied three dimension structural information of siRNA to increase predictability of the regression model [12]. However, most of those methods suffer from some drawbacks. Their correlations between predicted values and experimental values of dependent variable ranging from 0.60 to 0.68 were considerably decreased when testing on independent data sets. It may be caused by the fact that the Huesken dataset may not be representative of the siRNA population having about 4^{19} siRNAs and the sample size is small. Besides the scoring siRNA dataset, the labelled siRNA datasets, e.g. siRecord database [9] with labels such as 'very high", 'high', 'medium', 'low' for the knockdown ability were also exploited by classification methods.

Our work aims to develop a novel framework for better prediction of the siRNA knockdown ability. The key idea is not only focusing on learning algorithms but also exploiting results of the empirical process to enrich the data. To this end, we first learn transformation matrices by incorporating existing siRNA design rules with labelled siRNAs in siRecord database. We then use the transformation matrices to enrich scoring siRNAs as transformed matrices and do prediction with them by bilinear tensor regression where the Frobenius norm is appropriately replaced by L_2 regularization norm for an effective computation. Experiments show that the proposed method achieves results better than most existing models. The contributions of this work are summarized as follows

1. A novel generic framework to predict siRNA efficacy by enriching siRNA sequences with domain knowledge and appropriately using bilinear tensor regression.
2. An optimization method to enrich siRNAs using siRNA design rules found by empirical works.
3. The use of L_2 norm instead of Frobenius norm in bilinear tensor regression that allows effectively learning the set of model parameters.

2 The Framework to Improve siRNA Efficacy Prediction

The problem of siRNA knockdown efficacy prediction using siRNA design rules is formulated as follows:

- **Given:** Two sets of labelled siRNA and scoring siRNA sequences of length n, and a set of K siRNA design rules.
- **Find:** A function that assigns a right score to a given siRNA.

The proposed framework consists of four steps in two phases. The first phase is to encode siRNAs and learn transformation matrices. The second phase is to use transformation matrices to enrich siRNAs as transformed matrices and learn model parameters of the bilinear tensor regression to predict the score of siRNAs using transformed matrices. The steps of the framework are summarized in Table 1.

Table 1. Framework for siRNA knockdown efficacy prediction

1. To encode each siRNA sequence as an encoding matrix X representing the nucleotides A, C, G, and U at n positions in the sequence. Thus, siRNA sequences are represented as $n \times 4$ encoding matrices.
2. To learn transformation matrices $T_k, k = 1, ..., K$, each characterizes the knockdown ability of nucleotides A, C, G, and U at n positions in the siRNA sequence regarding the kth design rule. Each T_k is learned from the set of labelled siRNAs and the kth design rule. This incorporation of each design rule with siRNAs leads to solve a newly formulated optimization problem.
3. To transform siRNA (encoding matrices) to transformed matrices by K transformation matrices. The transformed matrices of size $K \times n$ are considered as second order tensor representations of the siRNA sequences.
4. To build a bilinear tensor regression model that uses transformed matrices of scoring siRNAs to predict the knockdown ability of new siRNAs.

2.1 Encoding siRNA and Transformation Matrix Learning

Step 1 of the framework can be easily done where each siRNA sequence with n nucleotides in length is encoded as a binary encoding matrix of size $n \times 4$. In fact, four nucleotides A, C, G, or U are encoded by encoding vectors (1,0,0,0), (0,1,0,0), (0,0,1,0) and (0,0,0,1), respectively. If a nucleotide from A, C, G, and U appears at the jth position in a siRNA sequence, $j = 1, ..., n$, its encoding vector will be used to encode the jth row of the encoding matrix.

Step 2 is to learn transformation matrices T_k regarding the kth design rule, $k = 1, ..., K$. T_k has size of $4 \times n$ where the rows correspond to nucleotides A, C, G, and U and the columns correspond to n positions on sequences. T_k are learned one by one from the set of siRNAs and the kth design rule, thus we use T instead of T_k for simplification. Each cell $T[i, j], i = 1, ..., 4, j = 1, ..., n$, represents the knockdown ability of nucleotide i at position j regarding the kth

Sequence	Encoding matrix X	Transformation matrix T	Transformed data vector $x = T \circ X$
AUGCU	1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1	$\overline{0.5}$ 0.7 0.32 0.2 0.5 0.3 0.1 0.6 $\overline{0.6}$ 0.3 0.1 0.1 $\overline{0.08}$ 0.1 0.1 0.1 $\overline{0.1}$ 0 0.1 $\overline{0.1}$	(0.5, 0.1, 0.08, 0.6, 0.1)

Position	Knockdown ability	Nucleo-tides	Mapping to T	Constraints on T
19	Effective	A,U	$T[1,19]$, $T[4,19]$	$T[3,19] - T[1,19] < 0$ $T[3,19] - T[4,19] < 0$
	Ineffective	C	$T[2,19]$	$T[2,19] - T[1,19] < 0$ $T[2,19] - T[3,19] < 0$ $T[2,19] - T[4,19] < 0$

Fig. 1. The left table shows an example of encoding matrix, transformation matrix, and transformed vector (the values $\overline{0.5}, \overline{0.1}$ etc. are taken to the transformed vector). The right table is an example of incorporating the condition of a design rule at position 19 to a transformation matrix T by designing constraints.

design rule. Each cell $T[i,j]$ to be learned have to satisfy a number of constraints. First, they are basic and normalization constraints on elements of T

$$T[i,j] \geq 0, \qquad i = 1, ..., 4; \;\; j = 1, 2, \ldots, n \tag{1}$$

$$\textstyle\sum_{i=1}^{4} T[i,j] = 1, \;\; j = 1, \ldots, n \tag{2}$$

The second kind of constraints related to design rules. Each design rule propositionally describes the occurrence or absence of nucleotides at different positions of effective siRNA sequences. Therefore, if a design rule shows the occurrence (absence) of some nucleotides on jth position, then their corresponding values in the matrix T would be greater (smaller) than other values at column j. For example, the design rule in the right table in Figure 1 illustrates that at position 19, nucleotides A/U are effective and nucleotide C is ineffective. It means that knockdown ability of nucleotides A/U are bigger than that of nucleotides G/C and knockdown ability of nucleotide C is smaller than that of the other nucleotides. Thus, values $T[1,19], T[2,19], T[3,19]$ and $T[4,19]$ show the knockdown ability of nucleotides A, C, G and U at position 19, respectively. Therefore, five constraints at column 19 of T are formed. Generally, we denote the set of R trick inequality constraints on T by the design rule under consideration by

$$\{g_r(T) < 0\}_{r=1}^{R} \tag{3}$$

The third kind of constraints relating to preservation of the siRNA classes after being transformed by using transformation matrices T_k, it means that siRNAs belonging to the same class should be more similar to each other than siRNAs belonging to the other class.

Let vector x_l of size $1 \times n$ denote the transformed vector of the lth siRNA sequence using the transformation matrix T. The jth element of x_l is the element of T at column j and the row corresponds to the jth nucleotide in the siRNA sequence. To compute x_l, new column-wise inner product is defined as follows

$$x_l = T \circ X_l = (\langle X_l[1,.], T[.,1] \rangle, \langle X_l[2,.], T[.,2] \rangle, \ldots, \langle X_l[n,.], T[.,n] \rangle) \tag{4}$$

where $X_l[j,.]$ and $T[.,j]$ are the jth row vector and the jth column of the matrix X_l and T, respectively, and $\langle x, y\rangle$ denotes the inner product of vectors x and y.

The left table in Figure 1 shows an example of encoding matrix X, transformation matrix T and transformed vector x of the given sequence AUGCU. The rows of X represent encoding vectors of nucleotides in the sequence. Given transformation matrix T of size 4 × 5. The sequence AUGCU is represented by the vector $x = (T[1,1], T[4,1], T[3,3], T[2,4], T[4,5]) =$ (0.5, 0.1, 0.08, 0.6, 0.1). Therefore, transformed data can be computed by the column-wise inner product $x = T \circ X$.

The problem of transformation matrix learning is now formulated as finding T under constraints (1), (2) and (3) so that the similarity of transformed vectors x_l in the same class is minimum and the dissimilarity of x_l in different classes is maximum. The learning problem then leads to solve the optimization problem with the following objective function

$$Min \sum_{p,q\in N_1} d^2(x_p,x_q) + \sum_{p,q\in N_2} d^2(x_p,x_q) - \sum_{\substack{p\in N_1\\ q\in N_2}} d^2(x_p,x_q) \tag{5}$$

$$\text{Subject to}$$
$$T[i,j] \geq 0,\ \sum_{i=1}^{4} T[i,j] = 1, g_r(T) < 0,\ i = 1,...,4; j = 1,...,n; r = 1,..,R.$$

In the objective function, the two first components are the sum of similarity of sequence pairs belonging to the same class and the last one is similarity of sequence pairs belonging to two different classes; $d(x,y)$ is the similarity measure between x and y (in this work we use Euclidean distance and L_2 norm); N_1 and N_2 are the two index sets of high and low efficacy siRNAs, respectively. Constraints $g_i(T)$ can also help to avoid the trivial solution of the objective function.

This optimization problem is solved by the following Lagrangian form

$$\begin{aligned}
E &= \sum_{p,q\in N_1} d^2(x_p,x_q) + \sum_{p,q\in N_2} d^2(x_p,x_q) - \sum_{\substack{p\in N_1\\ q\in N_2}} d^2(x_p,x_q) + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{4} T[i,j] - 1\right) + \sum_{r=1}^{R} \mu_r g_r(T)\\
&= \sum_{\substack{p\in N_1\\ q\in N_1}} \parallel x_p - x_q \parallel_2^2 + \sum_{\substack{p\in N_2\\ q\in N_2}} \parallel x_p - x_q \parallel_2^2 - \sum_{\substack{p\in N_1\\ q\in N_2}} \parallel x_p - x_q \parallel_2^2 + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{4} T[i,j] - 1\right) + \sum_{r=1}^{R} \mu_r g_r(T)\\
&= \sum_{p,q\in N_1} \sum_{j=1}^{n} (\langle X_p[j,.], T[.,j]\rangle - \langle X_q[j,.], T[.,j]\rangle)^2 + \sum_{p,q\in N_2} \sum_{j=1}^{n} (\langle X_p[j,.], T[.,j]\rangle - \langle X_q[j,.], T[.,j]\rangle)^2\\
&\quad + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{4} T[i,j] - 1\right) + \sum_{r=1}^{R} \mu_r g_r(T) - \sum_{\substack{p\in N_1\\ q\in N_2}} \sum_{j=1}^{n} (\langle X_p[j,.], T[.,j]\rangle - \langle X_q[j,.], T[.,j]\rangle)^2
\end{aligned}$$

where $\mu_r, r = 1,...,R$ and $\lambda_j, j = 1,\ldots,n$ are Lagrangian multipliers. To solve the minimization problem, an iterative method is applied. For each pair of (i,j), $T[i,j]$ is solved while keeping the other elements of T. The Karush-Kuhn-Tucker conditions are

- Stationarity: $\frac{\partial E}{\partial T[i,j]} = 0, i = 1,...,4$ and $j = 1,\ldots,n$.
- Primal feasibility: $T[i,j] \geq 0, \sum_{i=1}^{4} T[i,j] = 1, g_r(T) < 0, i = 1,...,4;$
  $j = 1,...,n; r = 1,...,R.$

- Dual feasibility: $\mu_r \geq 0, r = 1, \ldots, R$.
- Complementary slackness: $\mu_r g_r(T) = 0, r = 1, \ldots, R$.

From the last three conditions, we have $\mu_r = 0, r = 1, \ldots, R$. Therefore, the stationarity condition can be derived as follows

$$\begin{aligned}\frac{\partial E}{\partial T[i,j]} &= 2\sum_{p,q\in N_1}(\langle X_p[j,.],T[.,j]\rangle - \langle X_q[j,.],T[.,j]\rangle)(X_p[j,i]-X_q[j,i])\\ &+2\sum_{p,q\in N_2}(\langle X_p[j,.],T[.,j]\rangle - \langle X_q[j,.],T[.,j]\rangle)(X_p[j,i]-X_q[j,i])\\ &-2\sum_{p\in N_1,q\in N_2}(\langle X_p[j,.],T[.,j]\rangle - \langle X_q[j,.],T[.,j]\rangle)(X_p[j,i]-X_q[j,i]) + \lambda_j = 0\end{aligned}$$

Set $Z_{p,q} = (X_p - X_q)^T$ and A_{ij} is the vector resulting from the column j of matrix A by removing the element $A[i,j]$. Therefore, the above formulation is derived as follows

$$\begin{aligned}\frac{\partial E}{\partial T[i,j]} &= 2(\sum_{p,q\in N_1}\langle (Z_{p,q})_{ij}, T_{ij}\rangle Z_{p,q}[i,j] + \sum_{p,q\in N_2}\langle (Z_{p,q})_{ij}, T_{ij}\rangle Z_{p,q}[i,j]\\ &-\sum_{p\in N_1,q\in N_2}\langle (Z_{p,q})_{ij}, T_{ij}\rangle Z_{p,q}[i,j])\\ &+2T[i,j]\left(\sum_{p,q\in N_1} Z^2_{p,q}[i,j] + \sum_{p,q\in N_2} Z^2_{p,q}[i,j] - \sum_{p\in N_1,q\in N_2} Z^2_{p,q}[i,j]\right) + \lambda_j = 0\end{aligned}$$

We define the following equations

$$S(i,j) = \sum_{p,q\in N_1} Z^2_{p,q}[i,j] + \sum_{p,q\in N_2} Z^2_{p,q}[i,j] - \sum_{p\in N_1,q\in N_2} Z^2_{p,q}[i,j] \quad (6)$$

$$\begin{aligned}B(i,j) = &\sum_{p,q\in N_1}\langle (Z_{p,q})_{ij}, T_{ij}\rangle Z_{p,q}[i,j] + \sum_{p,q\in N_2}\langle (Z_{p,q})_{ij}, T_{ij}\rangle Z_{p,q}[i,j]\\ &-\sum_{p\in N_1,q\in N_2}\langle (Z_{p,q})_{ij}, T_{ij}\rangle Z_{p,q}[i,j].\end{aligned} \quad (7)$$

Substitute (6) and (7) to $\frac{\partial E}{\partial T[i,j]}$, we have

$$T[i,j] = \frac{\frac{-\lambda_j}{2} - B(i,j)}{S(i,j)} \quad (8)$$

At a column j, T has to satisfy

$$\textstyle\sum_{i_1=1}^4 T(i_1,j) = 1 \Leftrightarrow \displaystyle\sum_{i_1=1}^4 \frac{\frac{-\lambda_j}{2} - B(i_1,j)}{S(i_1,j)} = 1 \Rightarrow \frac{-\lambda_j}{2} = \frac{1 + \sum_{i_1=1}^4 \frac{B(i_1,j)}{S(i_1,j)}}{\sum_{i_1=1}^4 \frac{1}{S(i_1,j)}} \quad (9)$$

Substitute (9) to (8), equation (8) can be derived as

$$T[i,j] = \frac{\frac{1+\sum_{i_1=1}^4 \frac{B(i_1,j)}{S(i_1,j)}}{\sum_{i_1=1}^4 \frac{1}{S(i_1,j)}} - B(i,j)}{S(i,j)} = \frac{1+\sum_{i_1\neq i}\frac{B(i_1,j)-B(i,j)}{S(i_1,j)}}{\sum_{i_1=1}^4 \frac{S(i,j)}{S(i_1,j)}} \quad (10)$$

In this task, K design rules are used to learn K transformation matrices. The main steps are summarized in Algorithm 1. For each siRNA design rule, the algorithm will update each element of the transformation matrix according to equation (10). In each iterative step, the transformation matrix without trick inequality constraints is updated to reach the global optimal solution. If updated elements in a column satisfy the trick inequality constraints characterizing the condition at the corresponding position of the rule, that column will be updated to the target solution. The transformation matrix is updated until meeting the convergence criteria. $\| \, . \, \|_{Fro}$ is the Frobenious norm of a matrix.

Algorithm 1. Transformation matrices learning

```
Input: A data set $S = \{(s_l, y_l)\}_1^N$ where $s_l$ are siRNA sequences and $y_l$ are their labels, a set $DR$ of $K$ design rules, the length $n$ of siRNA sequences.
Output: $K$ transformation matrices $T_1, T_2, \ldots, T_K$.
Encoding siRNA sequences in $S$.
for $rule_k$ in $DR$ do
  Form the set of constraints $C_k$ based on $rule_k$
  Initialize the transformation matrix $T_k$ satisfying $C_k$.
  $t = 0$ { Iterative step}
  repeat
    $t \leftarrow t + 1$
    for $j = 1$ to $n$ do
      $v = T_k^{(t-1)}[., j]$ { A temporary vector}
      for $i = 1$ to 4 do
        Compute $v[i]$ using equation (10)
      end for
      if ($v$ satisfies the constraints at the position $j$ in $C_k$) then
        $T_k^{(t)}[., j] \leftarrow v$
      end if
    end for
  until $(\frac{\|T_k^{(t)} - T_k^{(t-1)}\|_{Fro}}{\|T_k^{(t-1)}\|_{Fro}} \le \epsilon)$ or $(t > t_{Max})$
end for
```

2.2 Tensor Regression Model Learning

Given a siRNA data set $D = \{(s_l, y_l)\}_1^N$ where s_l is the lth siRNA sequence of size n and $y_l \in \mathbb{R}$ is the knockdown efficacy score of s_l. Let X_l denotes the encoding matrix of s_l. Each encoding matrix X is transformed to K representations by K transformation matrices, $(T_1 \circ X, T_2 \circ X, \ldots, T_K \circ X)$. $R(X) = (T_1 \circ X, T_2 \circ X, \ldots, T_K \circ X)^T$ denotes the second order tensor of size $K \times n$.

The regression model can be defined as the following bilinear form

$$f(x) = \alpha R(X) \beta \tag{11}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ is a weight vector of the K representations of X and $\beta = (\beta_1, \beta_2, \ldots, \beta_n)^T$ is a parameter vector of the model, and $\alpha R(X)$ component is the linear combination of representations $T_1 \circ X, T_2 \circ X, \ldots, T_K \circ X$. It also shows the relationship among elements on each column of the second order tensor or each dimension of $T_k \circ X, k = 1, 2, \ldots, K$. Equation (11) can be derived as follows

$$f(X) = \alpha R(X)\beta = \left(\beta \otimes \alpha^T\right)^T vec(R(X)) = \left(\beta^T \otimes \alpha\right) vec(R(X)) \tag{12}$$

where $A \otimes B$ is the Kronecker product of two matrices A and B, and $vec(A)$ is the vectorization of matrix A. The weight vector α and the parameter vector β are learned by minimizing the following regularized risk function

$$L(\alpha, \beta) = \sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)^2 + \lambda \parallel \beta^T \otimes \alpha \parallel_{Fro}^2 \tag{13}$$

where λ is the turning parameter to tradeoff between bias and variance, and $\parallel \beta^T \otimes \alpha \parallel_{Fro}$ is the Frobenius norm of the first order tensor $\beta^T \otimes \alpha$. $L(\alpha, \beta)$ can be derived as follows

$$\begin{aligned} L(\alpha, \beta) &= \sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)^2 + \lambda \sum_{k=1}^{K}\sum_{j=1}^{n} (\alpha_k \beta_j)^2 = \sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)^2 + \lambda \sum_{k=1}^{K} \alpha_k^2 \sum_{j=1}^{n} \beta_j^2 \\ &= \sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)^2 + \lambda \sum_{k=1}^{K} \alpha_k^2 \parallel \beta \parallel_2^2 = \sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)^2 + \lambda \parallel \alpha \parallel_2^2 \parallel \beta \parallel_2^2 \quad (14) \end{aligned}$$

The risk function with Frobenius norm is converted to equation (14) with L_2 norm. In order to solve this optimization problem, an alternative iteration method is used. At each iteration, the parameter vector β is effectively solved by keeping the weight vector α and vice versa.

$$\begin{aligned} \tfrac{\partial L(\alpha,\beta)}{\partial \alpha} &= -2\sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)\left(R(X_l)\beta\right)^T + 2\lambda\alpha \parallel \beta \parallel_2^2 = 0 \\ \Leftrightarrow \quad & \sum_{l=1}^{N} \alpha \left(R(X_l)\beta\right)\left(R(X_l)\beta\right)^T - \sum_{l=1}^{N} y_l \left(R(X_l)\beta\right)^T + \lambda\alpha \parallel \beta \parallel_2^2 = 0 \\ \Rightarrow \quad \alpha = & \sum_{l=1}^{N} y_l \left(R(X_l)\beta\right)^T \left(\sum_{l=1}^{N} \left(R(X_l)\beta\right)\left(R(X_l)\beta\right)^T + \lambda \parallel \beta \parallel_2^2 I\right)^{-1} \quad (15) \\ \tfrac{\partial L(\alpha,\beta)}{\partial \beta} &= -2\sum_{l=1}^{N} (y_l - \alpha R(X_l)\beta)\left(\alpha R(X_l)\right)^T + 2\lambda\beta \parallel \alpha \parallel_2^2 = 0 \\ \Leftrightarrow \quad & \sum_{l=1}^{N} \alpha R(X_l)\beta \left(\alpha R(X_l)\right)^T - \sum_{l=1}^{N} y_l \left(\alpha R(X_l)\right)^T + \lambda\beta \parallel \alpha \parallel_2^2 = 0 \\ \Leftrightarrow \quad & \sum_{l=1}^{N} \left(\left(\alpha R(X_l)\right)^T \otimes \left(\alpha R(X_l)\right)\right)\beta - \sum_{l=1}^{N} y_l \left(\alpha R(X_l)\right)^T + \lambda\beta \parallel \alpha \parallel_2^2 = 0 \\ \Rightarrow \quad \beta = & \left(\sum_{l=1}^{N} \left(\left(\alpha R(X_l)\right)^T \otimes \left(\alpha R(X_l)\right)\right) + \lambda \parallel \alpha \parallel_2^2 I\right)^{-1} \sum_{l=1}^{N} y_l \left(\alpha R(X_l)\right)^T (16) \end{aligned}$$

Our proposed tensor regression model learning is summarized in Algorithm 2. In this algorithm, siRNA sequences are firstly represented as encoding matrices. The encoding matrices are then transformed to tensors by using K transformation matrices. After that, the weight vector α and the coefficient vector β are updated until meeting the convergence criteria, where t_{Max} denotes the maximum iterative step to update α and β, and ϵ_1 and ϵ_2 are thresholds for vectors α and β.

Algorithm 2. Tensor Regression Model Learning

```
Input: A data set $S = \{(s_i, y_i)\}_1^N$ where $s_i$ are scoring siRNA sequences and $y_i \in \mathbb{R}$.
$K$ transformation matrices $R_1, R_2, \ldots, R_k$, and the length $n$ of siRNA sequence.
Output: Weight vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ and parameter vector
$\beta = (\beta_1, \beta_2, \ldots, \beta_n)$ that minimize the regularized risk function
    – Represent siRNA sequences in $S$ as enconding matrices.
    – Transform encoding matrices to tensors using $K$ transformation matrices.
    – Initialize $\alpha$ and $\beta$ randomly.
    – $t = 0$ { Iterative step}
repeat
    $t \leftarrow t + 1$
    Compute $\alpha^{(t)}$ using equation (15)
    Compute $\beta^{(t)}$ using equation (16)
until $((\frac{\|\alpha^{(t)} - \alpha^{(t-1)}\|_2}{\|\alpha^{(t-1)}\|_2} \leq \epsilon_1)$ and $(\frac{\|\beta^{(t)} - \beta^{(t-1)}\|_2}{\|\beta^{(t-1)}\|_2} \leq \epsilon_2))$ or $(t > t_{Max})$
```

3 Experimental Evaluation

This section presents experimental evaluation in comparing the proposed method TRM (stands for 'tensor regression model') with the most recent reported methods for siRNA knockdown efficacy prediction on commonly used datasets. Discussion on the framework and methods will follow the experiment report.

Comparative Evaluation. The comparison is carried out using four data sets

- The Huesken dataset of 2431 siRNA sequences targeting 34 human and rodent mRNAs, commonly divided into the training set HU_train of 2182 siRNAs and the testing set HU_test of 249 siRNAs [5].
- The Reynolds dataset of 240 siRNAs [10].
- The Vicker dataset of 76 siRNA sequences targeting two genes [17].
- The Harborth dataset of 44 siRNA sequences targeting one gene [3].

TRM is compared to most state-of-the-art methods for siRNA knockdown efficacy prediction recently reported in the literature. As experiments in those methods cannot be repeated directly, we employed the results reported in the literature and carried out experiments on TRM in the same conditions of the other works. Concretely, the comparative evaluation is done as follows

1. Comparison of TRM with Multiple Kernel Support Vector Machine proposed by Qui *et al.* [8]. The author of [8] reported their Pearson correlation coefficient (R) of 0.62 obtained by 10-fold cross validation on the whole Huesken

Table 2. The R values of 18 models and TRM on three independent data sets

Algorithm	$R^{Reynolds}$	R^{Vicker}	$R^{Harborth}$	Algorithm	$R^{Reynolds}$	R^{Vicker}	$R^{Harborth}$
	(244si/7g)	(76si/2g)	(44si/1g)		(244si/7g)	(76si/2g)	(44si/1g)
GPboot	0.55	0.35	0.43	Stockholm 1	0.05	0.18	0.28
Uitei	0.47	0.58	0.31	Stockholm 2	0.00	0.15	0.41
Amarzguioui	0.45	0.47	0.34	Tree	0.11	0.43	0.06
Hsieh	0.03	0.15	0.17	Luo	0.33	0.27	0.40
Takasaki	0.03	0.25	0.01	i-score	0.54	0.58	0.43
Reynolds 1	0.35	0.47	0.23	Biopredsi	0.53	0.57	0.51
Reynolds 2	0.37	0.44	0.23	DSIR	0.54	0.49	0.51
Schawarz	0.29	0.35	0.01	Katoh	0.40	0.43	0.44
Khvorova	0.15	0.19	0.11	SVM	0.54	0.52	0.54
				TRM	**0.60**	**0.58**	**0.55**

dataset. The Pearson correlation coefficient (R) is carefully evaluated by TRM by 10 times of 10-fold cross validation with the average value of 0.64.

2. Comparison of TRM with four state-of-the-art methods of BIOPREDsi [5], DSIR [16], Thermocomposition21 [13], SVM [12] by HU_train and HU_test. The Pearson correlation coefficients of the four models BIOPREDsi, DSIR, Thermocomposition21 and SVM are 0.66, 0.67, 0.66 and 0.80, respectively. The performance of TRM estimated on HU_test is 0.68 that is slightly higher than that of the first three models but lower than that of the last model.

3. Comparison of TRM with 18 methods including BIOPREDsi, DSIR, Thermocomposition21, SVM when training on HU_train and testing on three independent datasets of Reynolds, Vicker and Harborth as reported in the recent article [12]. As shown in Table 2 (taken from [12] with the added last row of the TRM result), TRM considerably achieved results higher than all of 18 methods on the all three independent testing datasets.

In running Algorithm 2, the thresholds for the weight vector α and the coefficient vector β are set up as 0.001 and the maximum iterative step is 1000. The turning parameter λ is chosen by minimizing the risk function when testing on validation dataset. Particularly, we do 10–fold cross validation on the training set for each λ belonging to $[0, \log 50]$ and compute the risk function

$$R(\lambda) = \frac{1}{F}\sum_{i=1}^{F}\left(\frac{1}{\| fold_i \|}\sum_{x_j \in fold_i}(y_j - f(x_j))^2\right)$$

where $fold_i$ is validation set, $f(x)$ is a tensor predictor learnt from training set except validation set $fold_i$. F is the number of folds to do cross validation on training set. In our work, we do F-fold cross validation thus F equals to 10.

In the transformation matrices learning task, we use the labelled dataset collected from siRecord database [9]. This data set has 2470 siRNA sequences in 'very high' class and 2514 siRNA sequences in 'low' and 'medium' classes. Each siRNA sequence has 19 nucleotides. Seven design rules used to learn matrices are Reynolds rule, Uitei Rule, Amarzguioui rule, Jalag rule, Hsieh rule, Takasaki

Table 3. The learnt transformation matrix containing characteristics of Reynolds rule

	1	2	3	4		10	11	12		19
A	0.29704	0.217977	0.423469	0.266597	...	0.363636	0.246021	0.224727	...	0.393939
C	0.231159	0.235744	0.255102	0.226922	...	0	0.252513	0.267744	...	0.0757576
G	0.155341	0.211418	0.0459184	0.237968	...	0.229437	0.221336	0.260756	...	0.161616
U	0.31646	0.33486	0.27551	0.268513	...	0.406926	0.28013	0.246773	...	0.368687

rule and Huesken rule. The convergence criteria in Algorithm 1 are set up as following: threshold ϵ for transformation matrices is $2.5E^{-8}$ and the maximum iterative step is 5000.

Discussion. As reported in the experimental comparative evaluation, the proposed TRM achieved higher results than most other methods for prediction of siRNA knockdown efficacy. There are some reasons of that. First, it is expensive and hard to analyze the knockdown efficacy of siRNAs, and thus most available datasets are of relatively small size leading to limited results. Second, TRM has its advantages by incorporating domain knowledge (siRNA design rules) found from different datasets in experiments. Third, TRM is generic and can be easily exploited when new design rules are discovered or more analyzed siRNAs be obtained. Four, one drawback of TRM is its transformation matrices are learned using positional features of available design rules, and thus they lack some characteristics effecting to knockdown efficacy of siRNA sequences such as GC content, thermodynamic properties, GC stretch, etc. It may be one of reasons that at this moment TRM cannot get higher performance when testing on HU_test set than the best current model SVM [12].

Table 3 shows the learned transformation matrix capturing positional characteristics of Reynolds rule. One of characteristics is described as "An nucleotide 'A' at position 19". That characteristic means that at column 19, the cell (1,19) has to be the maximum value. In the matrix, the value at this cell is 0.393939 and is the highest value of this column. In this column, we also know knockdown efficacy of each nucleotide at position 19. Therefore, nucleotides can be arranged by the decreasing order of their efficacy: A,U, G, and C. In the order, nucleotide U has efficacy of 0.368687 that also can be used to design effective siRNAs. In addition, if a position on siRNAs is not described in characteristics of the design rules, values at the column corresponding to this position is learned to satisfy classification assumption and property to get knockdown efficacy of each nucleotide such as values at columns 1, 2, 4 and so on.

4 Conclusion

In this paper, we have proposed a novel framework to predict knockdown efficacies of siRNA sequences by successfully enriching the siRNA sequences into transformed matrices incorporating the effective siRNA design rules and predicting the siRNA knockdown efficacy by bilinear tensor regression. The experimental comparative evaluation on commonly used datasets with standard evaluation procedure in different contexts shows that the proposed framework and corresponding methods achieved better results than most existing methods for doing the same task. One significant feature of the proposed framework is it can be easily extended when new design rules are discovered as well as more siRNAs are analyzed by empirical works.

References

1. Amarzguioui, M., Prydz, H.: An algorithm for selection of functional siRNA sequences. Biochem. Biophys. Res. Commun. 316(4), 1050–1058 (2004)
2. Elbashir, S.M., Lendeckel, W., Tuschl, T.: RNA interference is mediated by 21– and 22–nucleotide RNAs. Genes Dev. 15, 188–200 (2001)
3. Harborth, J., Elbashir, S.M., Vandenburgh, K., Manninga, H., Scaringe, S.A., Weber, K., Tuschl, T.: Sequence, chemical, and structural variation of small interfering RNAs and short hairpin RNAs and the effect on mammalian gene silencing. Antisense Nucleic Acid Drug Dev. 13, 83–105 (2003)
4. Hsieh, A.C., Bo, R., Manola, J., Vazquez, F., Bare, O., Khvorova, A., Scaringe, S., Sellers, W.R.: A library of siRNA duplexes targeting the phosphoinositide 3-kinase pathway: Determinants of gene silencing for use in cell-based screens. Nucleic Acids Res. 32(3), 893–901 (2004)
5. Huesken, D., Lange, J., Mickanin, C., Weiler, J., Asselbergs, F., Warner, J., Mellon, 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 23(8), 955–1001 (2005)
6. Ichihara, M., Murakumo, Y., Masuda, A., Matsuura, T., Asai, N., Jijiwa, M., Ishida, M., Shinmi, J., Yatsuya, H., Qiao, S., et al.: Thermodynamic instability of siRNA duplex is a prerequisite for dependable prediction of siRNA activities. Nucleic Acids Res. 35(8), e123 (2007)
7. Jagla, B., Aulner, N., Kelly, P.D., Song, D., Volchuk, A., Zatorski, A., Shum, D., Mayer, T., De Angelis, D.A., Ouerfelli, O., Rutishauser, U., Rothman, J.E.: Sequence characteristics of functional siRNAs. RNA 11(6), 864–872 (2005)
8. Qiu, S., Lane, T.: A Framework for Multiple Kernel Support Vector Regression and Its Applications to siRNA Efficacy Prediction. IEEE/ACM Trans. Comput. Biology Bioinform. 6(2), 190–199 (2009)
9. Ren, Y., Gong, W., Xu, Q., Zheng, X., Lin, D., et al.: siRecords: An extensive database of mammalian siRNAs with efficacy ratings. Bioinformatics 22, 1027–1028 (2006)
10. Reynolds, A., Leake, D., Boese, Q., Scaringe, S., Marshall, W.S., Khvorova, A.: Rational siRNA design for RNA interference. Nat. Biotechnol. 22(3), 326–330 (2004)
11. Scherer, L.J., Rossi, J.J.: Approaches for the sequence-specific knockdown of mRNA. Nat. Biotechnol. 21, 1457–1465 (2003)
12. Sciabola, S., Cao, Q., Orozco, M., Faustino, I., Stanton, R.V.: Improved nucleic acid descriptors for siRNA efficacy prediction. Nucl. Acids Res. 41(3), 1383–1394 (2013)

13. Shabalina, S.A., Spiridonov, A.N., Ogurtsov, A.Y.: Computational models with thermodynamic and composition features improve siRNA design. BMC Bioinformatics 7, 65 (2006)
14. Takasaki, S.: Methods for Selecting Effective siRNA Target Sequences Using a Variety of Statistical and Analytical Techniques. Methods Mol. Biol. 942, 17–55 (2013)
15. 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 Res. 32, 936–948 (2004)
16. Vert, J.P., Foveau, N., Lajaunie, C., Vandenbrouck, Y.: An accurate and interpretable model for siRNA efficacy prediction. BMC Bioinformatics 7, 520 (2006)
17. Vickers, T.A., Koo, S., Bennett, C.F., Crooke, S.T., Dean, N.M., Baker, B.F.: Efficient reduction of target RNAs by small interfering RNA and RNase H-dependent antisense agents. A Comparative Analysis. J. Biol. Chem. 278, 7108–7118 (2003)