CLASSIFICATION OF NON-CODING RNA USING GRAPH REPRESENTATIONS OF SECONDARY STRUCTURE

YAN KARKLIN

Department of Computer Science
Carnegie Melon University
Pittsburgh, PA USA
E-mail: yan+@cs.cmu.edu

RICHARD F. MERAZ* AND STEPHEN R. HOLBROOK

Physical Biosciences Division
Lawrence Berkeley National Laboratory
Berkeley, CA USA
E-mail: rfmeraz@gmail.com, srholbrook@lbl.gov

Some genes produce transcripts that function directly in regulatory, catalytic, or structural roles in the cell. These non-coding RNAs are prevalent in all living organisms, and methods that aid the understanding of their functional roles are essential. RNA secondary structure, the pattern of base-pairing, contains the critical information for determining the three dimensional structure and function of the molecule. In this work we examine whether the basic geometric and topological properties of secondary structure are sufficient to distinguish between RNA families in a learning framework. First, we develop a *labeled dual graph* representation of RNA secondary structure by adding biologically meaningful labels to the dual graphs proposed by Gan *et al* [1]. Next, we define a similarity measure directly on the labeled dual graphs using the recently developed marginalized kernels [2]. Using this similarity measure, we were able to train Support Vector Machine classifiers to distinguish RNAs of known families from random RNAs with similar statistics. For 22 of the 25 families tested, the classifier achieved better than 70% accuracy, with much higher accuracy rates for some families. Training a set of classifiers to automatically assign family labels to RNAs using a *one vs. all* multi-class scheme also yielded encouraging results. From these initial learning experiments, we suggest that the *labeled dual graph* representation, together with kernel machine methods, has potential for use in automated analysis and classification of uncharacterized RNA molecules or efficient genome-wide screens for RNA molecules from existing families.

*to whom correspondence should be addressed.

1. Introduction

Non-coding RNA (ncRNA) molecules are those RNAs that do not encode proteins, but instead serve some other function in the cell [3]. They play a variety of critical roles and are ubiquitous in all kingdoms of life [4]. The function of non-coding RNAs is uniquely determined by the three dimensional structure of the molecule. To reach its functional form, a single stranded RNA molecule undergoes folding – driven by GC/AU/GU base-pairing and stacking interactions – to form short helices and various single stranded loop regions that define its secondary structure [5]. Some RNAs require metals or proteins to chaperone the folding process, but for the most part, the final three dimensional structure, and hence the functional role, is fully determined by the secondary structure [6]. This suggests that development of computational tools based on RNA secondary structure is essential for discovery of new non-coding RNAs and classification of their functional roles.

A variety of computational methods have used the secondary structure of RNA molecules to search and categorize ncRNAs, but many of these methods are limited in their use of secondary structure. Regular-expression-like pattern matching algorithms have been used to scan genome sequences for regions that fold into the canonical structures of specific families [7]. However, they are designed to match stringent configurations of secondary structure elements, and therefore perform poorly on families with variations in folding. Pair Stochastic Context Free Grammars (P-SCFG) look for evidence of secondary structure conservation by modeling covariance of mutations from related genomes [8] – but determining an appropriate grammar is a non-trivial problem [9]. Some discriminative classifiers use secondary structure stability as an input feature to distinguish non-coding RNAs from intergenic sequence [10], but they ignore important topological information. On the other hand, methods that use computable representations of secondary structure, such as trees and graphs, have been restricted to categorization and enumeration of gross topological features [11, 1].

Here we present a kernel-based machine learning method for classifying RNA families that avoids some of these limitations by learning directly from a graphical representation of secondary structure. This discriminative method does not require the estimation of any parameters or training of cumbersome generative models, yet it captures some of the topological relationships of RNA secondary structures. First, we define an appropriate representation of RNA secondary structure by extending the RNA dual graph

representation [1] with a biologically relevant labeling scheme. Second, we define a similarity measure between RNA secondary structures by applying the recently developed marginalized kernel [2] to compare RNA molecules represented as labeled dual graphs. We tested the ability of this method to learn non-coding RNA structure by training Support Vector Machine [12] classifiers to distinguish known ncRNAs from random RNA sequences with similar nucleotide statistics. We also tested whether this approach can pick up on and generalize from structural features that distinguish non-coding RNA families.

2. An Algorithm for Classification Based on Secondary Structure Topology

Classification of RNA secondary structures with Support Vector Machines (SVMs) requires both a representation that captures the secondary structure and a kernel function that provides a reasonable similarity measure for the chosen representation. Below we present a graph representation of RNA secondary structure, the *labeled dual graph*, and show how it captures the basic structural features of the molecule. We then describe a method for applying kernel functions to the labeled dual graphs.

2.1. *Labeled Dual Graphs*

Given a secondary structure of an RNA molecule (see Figure 1A for examples), we want to construct a graph that captures essential properties of the structure. The dual graph [1] is a concise representation that captures basic topological properties of the folded RNA molecule, such as the number and relative position of the helical regions. In this representation, helical regions of the RNA are represented as vertices of a graph, while single RNA strands that connect the helical regions are edges. Thus, internal loops, bulges, and multi-loops become edges that connect vertices (helices adjacent to the loops), and external loops become edges from a vertex to itself. The result is a multigraph – up to two edges may connect a pair of vertices when a bulge or an internal loop separates two helical regions – that excludes the free 5' and 3' ends and ignores the directionality of the molecule, but captures its basic topology.

We augment the graph representation by adding labels that correspond to the length and type of secondary structure elements. The resulting *labeled dual graphs* (LDGs) are comprised of vertices labeled according to the number of nucleotide-pairs in the helical region they represent, and edges

labeled according to the length (in number of nucleotides) and type (internal/external) of the loop they represent. See Figure 1B for an illustration of labeled dual graphs.

2.2. *Marginalized Kernels for Labeled Dual Graphs*

In order to use an SVM classifier on graph objects, we need a kernel function to define a similarity between two labeled dual graphs. Several kernels for graph objects have been proposed [13, 2]; here we use the recently developed *marginalized kernel* for labeled graphs [2] because it is relatively simple to implement, computationally efficient, and yielded promising results. Intuitively, this kernel function computes a similarity measure between two arbitrary labeled graphs by comparing the label sequences produced by taking random walks on each of the two graphs; the more similar the sets of label sequences, the higher the similarity score for the pair of graphs.

The computation of the kernel function between two graphs G and G' proceeds as follows. First, generate a random walk h on graph G and a walk h' on graph G', according to some defined probability of transitioning from vertex to vertex. Each walk produces a sequence of vertex and edge labels, $z = \{v_1, e_{12}, v_2, e_{23}, v_3, \ldots\}$ and $z' = \{v'_1, e'_{12}, v'_2, e'_{23}, v'_3, \ldots\}$ (see Figure 1C for an example). Next, define the label sequence kernel $K_z(z, z')$ as the product of the vertex label kernels $K_v(v, v')$ and the edge label kernels $K_e(e, e')$ over the sequence of labels,

$$K_z(z, z') = K_v(v_1, v'_1)K_e(e_{12}, e'_{12})K_v(v_2, v'_2)\ldots . \tag{1}$$

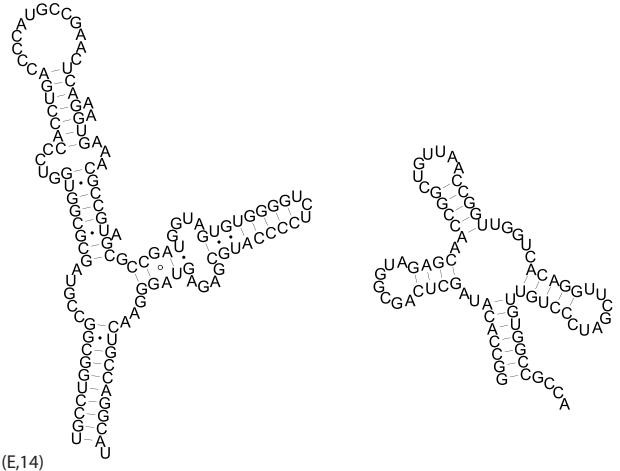
If the two walks are of different lengths, we define the label sequence kernel to be 0. Now that a similarity measure $K_z(z, z')$ is defined for each pair of walks, the value of the full graph kernel $K(G, G')$ is computed as the expected value of $K_z(z, z')$ over all possible walks h and h', weighted by the probability of generating the walks,

$$K(G, G') = \langle K_z(z, z')\rangle_{h,h'} \ . \tag{2}$$

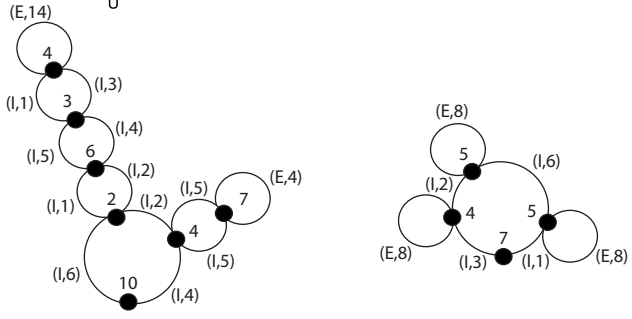
The probability of taking a random walk on a graph, $p(h, h')$ depends on the probability of starting at a particular vertex and transitioning to subsequent vertices. We assumed a uniform starting probability over all vertices, a uniform probability of transitioning from a vertex to one of its neighbors, and a constant probability (0.1) of terminating the walk after any step.

Finally, we need to specify the edge and the vertex kernel functions, $K_e(.,.)$, $K_v(.,.)$. These should reflect the similarities in RNA structural motifs – similar helices should produce high similarity scores, as should

A



B



C

$L=1$	$\{4\},\{3\},\{6\},\{2\},\{10\},\{4\},\{7\}$	$\{5\},\{4\},\{7\},\{5\}$
$L=2$	$\{4,(\mathrm{E},14),4\},\{4,(\mathrm{I},1),3\},\ldots$	$\{7,(\mathrm{I},3),4\},\{5,(\mathrm{I},6),5\},\ldots$
$L=3$	$\{4,(\mathrm{I},5),7,(\mathrm{E},4),7\},\ldots$	$\{7,(\mathrm{I},3),4,(\mathrm{E},8),4\},\ldots$
$L=4$	$\{7,(\mathrm{I},1),5,(\mathrm{E},7),5,(\mathrm{I},4),5\},\ldots$	$\{7,(\mathrm{I},5),7,(\mathrm{E},55),7,(\mathrm{I},7),8\},\ldots$
$\vdots$		$\vdots$

D

$$K_z = K_v(4,7)K_e(5,3)K_v(7,4)K_e(4,8)K_v(7,4)$$
$$= exp(-(\log\frac{4}{7})^2 - (\log\frac{5}{3})^2 - (\log\frac{7}{4})^2 - (\log\frac{4}{8})^2 - (\log\frac{7}{4})^2)$$

Figure 1. (*A*) Secondary structure diagram and (*B*) *labeled dual graph* (LDG) representation of 5S rRNA (left) and tRNA (right) molecules. In the LDG, the numbers and ordered pairs are the vertex (helix) and edge (loop) labels, respectively. The labels E and I are used to distinguish external from internal loops. (*C*) A subset of label sequences generated by taking random walks on the two graphs. Here L refers to the length of the path. (*D*) An example of the label sequence kernel and its output, as it is applied to the highlighted pair of paths in (*C*). The full kernel between the two graphs is computed as the expected value of the path kernels over all possible pairs of walks.

comparable loops. The choice of biophysical parameters that can serve as the basis for similarity comparisons is large – base composition, sequence or structural alignment, feature lengths, among others. As a first step we chose edge and vertex kernels that reflect the most basic structural parameters: the number of nucleotides that comprise a secondary structure motif. The vertices and edges of the dual graphs are labeled with these distances, and the vertex and edge kernels are defined as the Gaussian distance on the log-ratio of the two labels (lengths). This choice of kernel means that, compared to a particular structural element, elements twice its length or half its length score similarly, and that the similarity measure drops off smoothly as the ratio of the lengths deviates from 1. The vertex kernel is thus defined as

$$K_v(v_i, v_j) = exp(-\lambda_{ij}^2)\,, \tag{3}$$

where $\lambda_{ij} = log(v_i/v_j)$. For two edges of the same type of loop (internal or external), the edge kernel is similarly defined,

$$K_e(e_{ij}, e_{kl}) = exp(-\lambda_{ij,kl}^2)\,, \tag{4}$$

and for edges of different types, the edge kernel is 0. See Figure 1D for an illustrative example.

Effectively, two labeled dual graphs are considered similar when the two sets of all possible walks on each graph are similar. The similarity between individual walks is calculated as a product of simple functions defined on their constituent labels. Thus, if all the vertex and edge labels in the two walks match up, the output of the kernel function on the two walks will be high; and if many of the walks on the two graphs are similar, the kernel function will return a high value (with $\max K(G, G') = 1$) for the two graph objects. Hence this computation captures some topological relationships between structural elements of RNA secondary structure.

3. Methods

We performed two sets of experiments to test the ability of the classifier to learn RNA secondary structure and predict RNA family labels. First we trained SVM classifiers to distinguish non-coding RNAs from random RNAs with similar di-nucleotide composition. We also trained a system of multi-class SVMs to determine the family labels of RNA sequences.

Single family classification was tested on a number of RNA families from the RFAM database [14] (see the Results section for the list of tested RFAM families). When possible, we trained and tested the classifier on 500

RNA sequences, randomly selected from all RNAs in the family. However, some RFAM families contained fewer sequences, in which case all were used for classification. The negative data set was constructed by shuffling the nucleotide sequences of the positive data set while preserving the di-nucleotide frequencies (see [15] for methods), which destroys characteristic secondary structure but produces random RNAs with sequence statistics similar to real RNA.

RNA sequences were converted to secondary structures with the Vienna RNA [16] folding prediction package, then converted to labeled dual graphs as described above. We implemented the kernel computation using an iterative method described in [2]; one thousand kernel computations took between 2 and 40 seconds on a desktop machine (2GHz Athlon), depending on the average complexity of the secondary structure. SVM classification was performed with 10 fold cross validation, with the precision parameter set to 10000. We assessed classifier performance with sensitivity and specificity measures and by computing the area under the Receiver Operating Characteristic (ROC) curve, a general measure of the discrimination ability [17].

We also trained a multiclass classifier on nine large RFAM families using the *one vs. all* method, a simple and frequently used approach to multiclass classification [18]. In this method, a separate classifier is trained to distinguish each class from the remaining ones. During classification, a test sample (in this case an RNA sequence) is tested against each of the trained classifiers, and a label assigned according to the classifier that produced the highest decision value.

In this experiment, we grouped together several related RNA families in order to have a sufficient number of sequences in each class for training and testing (see Results for details). We assessed performance with the generalized *class sensitivity* and *class specificity* measures [19]. For each classifier, the class sensitivity (Q^D) represents the percentage of samples correctly predicted relative to the total number of samples in that family, while the class specificity (Q^M) captures the number of samples correctly predicted relative to the total number of samples predicted to be in that family.

4. Results

4.1. *Single Family SVM*

Figure 2 shows the results of SVM classifiers trained to identify individual RFAM families. For sufficient training data we used only families with 50 or more sequences. The generation of negative training data is described in the previous section. The classifiers showed good performance for a large number of families, with $A_{ROC} > 0.7$ for 22 of 25 families tested. This suggests that the learning method is useful for learning a variety of secondary structure topologies. A notable result is the good classifier performance on several riboswitch and microRNA families, two particularly exciting non-coding RNA classes that have recently been shown to be involved in novel mechanisms for regulating gene expression.

4.2. *Multi-class SVM*

Table 1 shows the cross validation results of the *one vs. all* multi-class SVM trained on nine RFAM families. The MICRO and RNASE groups represent aggregates of functionally related individual RFAM families (see the caption for details). Again, classifier sensitivity and specificity were good over a range of families, although specificity clearly degraded for RNA families with larger molecules and possibly more complicated secondary structures. In these instances, it is possible that shorter walks pick up spurious similarities.

5. Discussion

The method presented here was able to learn to distinguish a number of non-coding RNA families; however, it is worth highlighting a few factors that may have adversely impacted its performance. First and most important is the reliance of the algorithm on accurate secondary structure prediction. Because the classifier uses solely secondary structure as input, it is sensitive to incorrectly predicted structures. As an example, training and testing a classifier on tRNAs for which correct folding was manually verified increased the accuracy from 89% to 98% (A_{ROC}). Nevertheless, because the kernel computation considers local paths over the entire structures, parts of the molecule that are correctly folded will still contribute to the correct computation of the kernel, even if some parts of the molecule are mis-folded. More accurate folding algorithms will likely improve the performance of this classifier. Alternatively, we can incorporate the con-

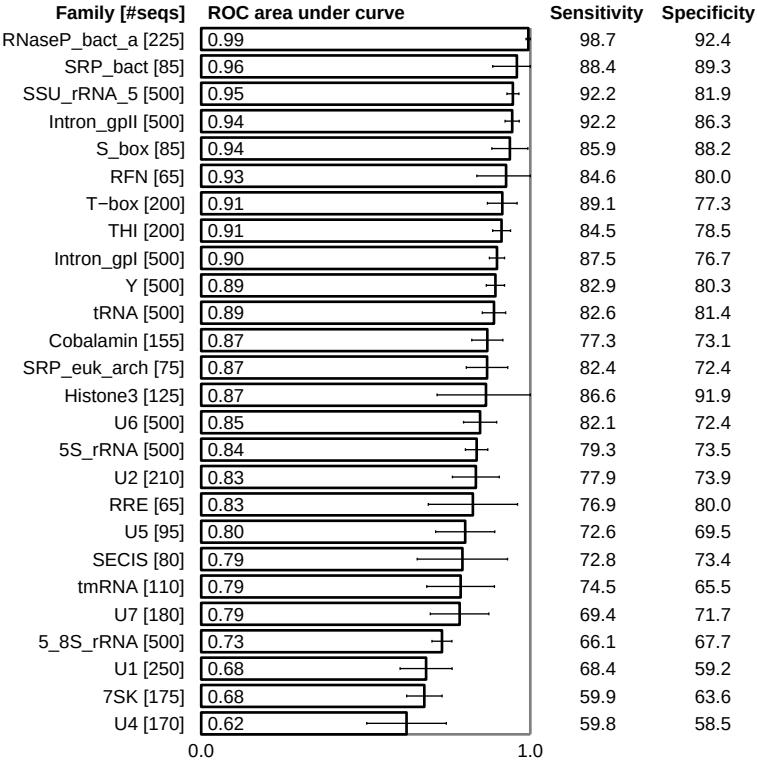


Figure 2. Performance of SVM classifiers trained on single RFAM families vs. shuffled sequences with the same di-nucleotide composition. Area under the ROC curve (A_{ROC}) is computed as the mean of the areas for each ROC curve of the 10 cross validation trials; error bars are standard deviation of A_{ROC}.

fidence in the secondary structure prediction into the learning algorithm, or even use the set of predicted suboptimal structures provided by folding algorithms as input to the classifier.

As a representation of RNA secondary structure, the labeled dual graph captures the basic features of the molecule: the number and length of helical regions and their relative position. However, some of the structural information is not represented. For example, the natural 5'-3' directionality of the molecule, the lengths of free 5' and 3' strands, as well as more complex topological features such as chirality. Much of this information could be included with natural extensions to the labeling scheme.

The computation of similarity between graphs (implemented with the

Table 1. Contingency table showing results for 10 fold cross validation of *one vs. all* multi-class SVM. For each RNA family (table row), the number of RNAs classified as a certain family appears in the respective column. Q^D and Q^M refer to generalized sensitivity and specificity, respectively. If z_{ij} is an element in the contingency table, then $Q_i^D = \frac{z_{ii}}{\sum_j z_{ij}}$ and $Q_j^M = \frac{z_{jj}}{\sum_i z_{ij}}$. Several functionally related small RFAM families were grouped together to form aggregate families, **MICRO**: *let-7, lin-4, mir-1, mir-10, mir-101, mir-103, mir-124, mir-130, mir-135, mir-148, mir-156, mir-16, mir-160, mir-166, mir-17, mir-181, mir-19, mir-192, mir-194, mir-196, mir-199, mir-2, mir-218, mir-219, mir-24, mir-26, mir-29, mir-30, mir-46, mir-6, mir-7, mir-8, mir-9*; and **RNASE**: *RNaseP_bact_a, RNaseP_bact_b, RNaseP_nuc, RNase_MRP*, and these were trained and tested as single classes.

	Histone3	Intron_gpI	Intron_gpII	MICRO	RNASE	SSU_rRNA_5	tRNA	U6	Y	$\mathbf{Q^D}$
Histone3	123	0	0	0	0	0	1	3	0	**.97**
Intron_gpI	0	355	82	1	18	33	1	5	5	**0.71**
Intron_gpII	0	27	443	0	5	7	6	9	3	**0.89**
MICRO	0	3	2	165	0	0	1	1	8	**0.92**
RNASE	0	26	5	1	251	52	0	3	3	**0.74**
SSU_rRNA	0	17	0	0	5	474	0	1	3	**0.95**
tRNA	0	16	8	4	8	27	370	33	26	**0.75**
U6	0	17	4	0	4	25	10	409	31	**0.82**
Y	0	32	3	4	11	31	14	30	375	**0.75**
$\mathbf{Q^M}$	**1.0**	**0.72**	**0.81**	**0.94**	**0.83**	**0.73**	**0.92**	**0.83**	**0.83**	

marginalized kernels) is also an imperfect measure. It does not account for relative position of helical regions, it is sensitive to bulges in helical regions, and it ignores global features such as the number of helices and the size of the molecule. Some of these might not be critical for discriminating ncRNAs – we tried a variant of LDGs that ignores bulges and observed no improvement in performance – but others should be incorporated into the representation and the kernel computation. Finally, the parameters used for computing the marginalized kernels also have an impact on the kernel output. For larger walks the random walk transition probabilities affect the relative contributions of local or global structural features to the similarity measure. Instead of adapting these parameters for optimal performance, we simply chose a set of sensible values, and it is possible that performance can be improved by adjusting these parameters. In order to address these concerns, it will be essential to look at exactly what aspects of the representation and the kernel allow the algorithm to learn to distinguish ncRNAs and to generalize to new structures. This will help us understand where and why it succeeds, and which aspects require improvement, and

would also suggest areas of application for which this method is particularly suited.

6. Conclusion

We have presented a novel, simple, and computationally efficient approach for learning RNA secondary structures that requires no tuning of parameters and can be applied to a wide range of learning problems. It uses graph representations of folded RNA structures and kernels defined on graph objects to train SVM classifiers. Applied to non-coding RNAs from the RFAM database, the method gave promising results. It could distinguish many families from random RNA sequences with identical di-nucleotide composition, and showed some ability to differentiate one family from another. Because this conceptually simple approach produced relatively accurate classifiers, and because no other automated discriminative method for classification or discovery of ncRNA families exists, we believe there is great potential for extending this method or combining it with other techniques. Specific applications could include automated class-discovery of uncharacterized RNA molecules and computationally efficient heuristic filters in conjunction with other methods for RNA family prediction.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by grant 5RO1H6002665-02 from the NHGRI. Yan Karklin is supported by a Department of Energy Computational Science Graduate Fellowship (DOE-CSGF).

References

1. H. H. Gan, S. Pasquali, and T. Schlick. Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. *Nucleic Acids Res*, 31(11):2926–43, 2003.
2. H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *International Conference on Machine Learning*, volume 20, pages 321–328. AAAI Press, 2003.
3. S. R. Eddy. Noncoding RNA genes. *Curr Opin Genet Dev*, 9(6):695–9, 1999.
4. G. Storz. An expanding universe of noncoding RNAs. *Science*, 296(5571):1260–3, 2002.
5. M. Zuker. Calculating nucleic acid secondary structure. *Curr Opin Struct Biol*, 10(3):303–10, 2000.
6. I. Tinoco and C. Bustamante. How RNA folds. *J. Mol Biol*, 293(2):271–281, 1999.

7. V. Tsui, T. Macke, and D. A. Case. A novel method for finding tRNA genes. *RNA*, 9(5):507–17, 2003.
8. E. Rivas, R. J. Klein, T. A. Jones, and S. R. Eddy. Computational identification of noncoding RNAs in E. coli by comparative genomics. *Curr Biol*, 11(17):1369–73, 2001.
9. R. D. Dowell and S. R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5(1):71, 2004.
10. R. J. Carter, I. Dubchak, and S. R. Holbrook. A computational approach to identify genes for functional RNAs in genomic sequences. *Nucleic Acids Res*, 29(19):3928–38, 2001.
11. G. Benedetti and S. Morosetti. A graph-topological approach to recognition of pattern and similarity in RNA secondary structures. *Biophys Chem*, 59(1-2):179–84, 1996.
12. C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
13. T. Gartner. A survey of kernels for structured data. *ACM Special Interest Group on Knowledge Discovery Explorations*, 5(1):49–58, 2003.
14. S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy. Rfam: an RNA family database. *Nucleic Acids Res*, 31(1):439–41, 2003.
15. C. Workman and A. Krogh. No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucleic Acids Res*, 27(24):4816–22, 1999.
16. S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–65, 1999.
17. T. Fawcett. ROC graphs: Notes and practical considerations for data mining researchers. Technical report, HP Laboratories, 1/17/2003 2003.
18. R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
19. P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–24, 2000.