ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

RNAsynth: constraints learning for RNA inverse folding.

Fabrizio Costa, Parastou Kohvaei, Robert Kleinkauf

Institut für Informatik, Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 106
D-79110 Freiburg, Germany

Abstract.

RNA polymers are an important class of molecules: not only they are involved in a variety of biological functions, from coding to decoding, from regulation to expression of genes, but crucially, they are nowadays easily synthesizable, opening interesting application scenarios in biotechnological and biomedical domains. Here we propose a constructive machine learning framework to aid in the rational design of such polymers. Using a graph kernel approach in a supervised setting we define an *importance* notion over molecular parts. We then convert the set of most important parts into specific sequence and structure constraints. Finally an inverse folding algorithm uses these constraints to compute the desired RNA sequence.

1 Introduction

RNA macro molecules undergo *cis*-interactions and fold into stable structures. It is the chemo-physical properties of the unbound portion of the RNA polymer together with their 3D configuration that determines the function of the molecule, i.e. the type of interactions that are feasible with other biological entities. Different types of functions are currently classified in $\sim$2400 different groups or *families* [2]. Crucially, different sequences can fold into identical structures, a fact that makes the prediction of the function of a given RNA sequence a hard task. Within biotechnological and biomedical application development, e.g. RNA aptamers [4, 5], RNA nanoparticles [9], or CRISPR/cas9 systems [13], the design of RNA sequences with a desired structure, and by proxy, with a desired function, is becoming of great interest.

In the following we will consider the case of the hammerhead ribozyme (Figure 1). This is a RNA molecule with therapeutic applications that catalyzes reversible cleavage and joining reactions at a specific site within an RNA molecule.

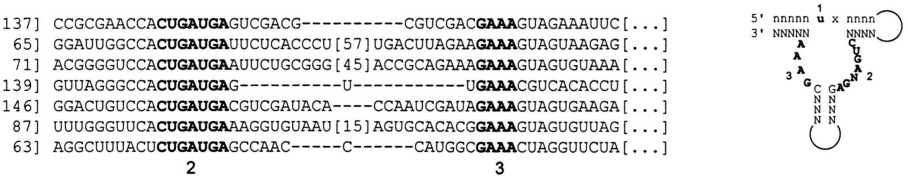

Fig. 1: Hammerhead ribozyme aligment and motif.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

The structural-sequence alignment of the rybozyme sequences identifies a conserved motif in an unpaired region that is enclosed by three stems (paired regions) with a variable nucleotide composition. The development of a computational method capable to reliably construct a large number of sequences sharing the same function of the hammerhead ribozyme can be a key step towards engineering artificial molecules with regulatory capabilities. Note that the approach that we introduce is completely generic and can be used to sample from any given set of functionally related RNAs.

2 Constraint learning framework

In the proposed framework, that we call RNASynth, we will assume that a set of RNA sequences sharing the same function is made available, without assuming any other a priori knowledge. The task is then to jointly identify the necessary constraints that define the class membership and produce novel sequences belonging with high probability to the desired class. To this end we use a graph kernel approach in a supervised setting to define an importance notion over parts of the structures. We then convert the set of most important parts into specific label and structure constraints. Finally these constraints are given in input to inverse folding algorithm (antaRNA [11] in our case) to output the desired samples.

2.1 Graph kernel

An essential component of the proposed framework is a linear discriminant estimator: this allows to explicitly access the discriminative importance of each feature which will be later used to define the notion of part importance. In order to work with graphs in input we make use of the graph kernel developed in [3], called Neighborhood Subgraph Pairwise Distance Kernel (NSPDK). NSPDK is an instance of a decomposition kernel [10], where the *"parts"* in which the graph is decomposed are pairs of small near neighborhood subgraphs of increasing radii $r < r_{max}$. All pairs of such subgraphs whose roots are at a distance not greater than $d < d_{max}$ are considered as individual features. The similarity notion is finally given as the fraction of features in common between two graphs. Formally the relation is defined in terms of neighborhood subgraphs (i.e. subgraphs induced by the nodes at bounded distance from a root vertex) as: $R_{r,d} = \{(N_r^v(G), N_r^u(G), G) : d(u,v) = d\}$ that is, a relation $R_{r,d}$ that identifies pairs of neighborhoods N of radius r whose roots u, v are exactly at distance d. The kernel is then defined as:

$$\kappa_{r,d}(G, G') = \sum_{\substack{A,B \in R_{r,d}^{-1}(G) \\ A',B' \in R_{r,d}^{-1}(G')}} \mathbf{1}_{A \cong A'} \cdot \mathbf{1}_{B \cong B'} \quad (1)$$

where $R_{r,d}^{-1}(G)$ is the inverse of the R relation and indicates the multi-set of all pairs of neighborhoods of radius r with roots at distance d that exist in G, and where $\mathbf{1}$ denotes the indicator function and $\cong$ the isomorphism between graphs. The final kernel is then the sum of $\kappa_{r,d}$ kernels. The optimal bounds for the values of r and d are typically obtained via cross validation.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

2.2 Importance notion

Kernels are generally used in their implicit form, i.e. to compute the dot product between two instances in the corresponding Reproducing Kernel Hilbert Space. Since graph kernels yield in general descriptions with an exponential amount of features this is the preferred practice. For NSPDK instead we use the hashing technique introduced in [8] to obtain the explicit feature encoding of the graphs as sparse vectors in a very high dimensional space: $\Phi(G) \in \mathbb{R}^m$. This is possible since we can efficiently enumerate all features of the NSPDK representation in quasi linear time. In this way learning can happen in the *primal* form and we can employ a stochastic gradient descent (SGD) algorithm [1] to efficiently induce a linear estimator over binary classification problems.

We define the importance score S of a node $u \in G$ as the distance of the explicit NSPDK representation of the neighborhood graph rooted in u with radius $R = r_{max} + d_{max}$, $\Phi(N_R^u(G))$ from a given hyperplane h with norm w and bias b :

$$S_R^{w,b}(u, G) = \Phi(N_R^u(G))' \cdot w + b.$$

The hyperplane h is derived from the SVM solution to the binary classification problem between sequences of interest vs. randomly permuted sequences. Given the norm w' and bias b' of the solution hyperplane h', we let $w = w'$ and $b = \frac{b'}{|V(G)|}$.

Intuitively, the neighborhood graph of radius R of a vertex u comprises all nodes that can form NSPDK features with u and hence $\Phi(N_R^u(G))$ forms a partition over the features $\Phi(G)$. In this way the SVM score associated to a graph can be decomposed as the sum of importance scores S over the nodes of the graph:

$$\Phi(G)' \cdot w' + b' = \sum_{u \in V(G)} S_R^{w,b}(u, G).$$

Finally the importance score of any subgraph $H \subset G$ is defined as the sum of importance scores for each node:

$$S_R^{w,b}(H, G) = \sum_{u \in V(H)} S_R^{w,b}(u, G),$$

where $V(H)$ denotes the vertex set of a subgraph H.

3 Inverse folding

Ant Colony Optimization (ACO) [6] is an optimization algorithm for problems that can be reduced to finding good paths through graphs. The algorithm is inspired by the principle of ant foraging: ants explore a terrain and evaluate foraging paths according to their length, the presence of obstacles and the amount and quality of food sources at the path's end. The terrain is marked with a signal (pheromone) that tracks paths. As time passes, more ants perceive and get influenced by the local pheromone information. This information is then modified by the notion of *evaporation* whereby low quality paths are forgotten while high quality ones are progressively reinforced.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

```
1. GACUGCUUGGCGCAAUGGUAGCGCGUUCGACUCCAGAUCGAAAGGUUGGGCGUUCGAUCCGCUCAGUGGUCA
2. ((((((...((((.......))))...........((((....))))(((((.......)))))).)))))).

3. ---------GCGC-------GCGC---------------GAAA-----------------------------
4. ---------((((.......))))-----------((((----))))-------------------------
```

Fig. 2: Sequence information (1) and corresponding minimum free energy structure (2). Constraints are position specific and can involve the type of nucleotides (3) or the presence (parenthesis) or absence (dots) of base pairing (4). Unconstrained elements are encoded with the dash symbol.

antaRNA [11] uses ACO in order to produce RNA sequences that satisfy user specified constraints and fold in the desired structure. The virtual ants modify the sequence construction graph (the graph representing the superposition of all possible sequences of 4 nucleotides) according to the sequence quality (i.e. number and degree of violation of the constraints). antaRNA supports several types of constraints, among which: partial secondary structure specification, partial sequence specification and Guanine-Cytosine (GC) ratio specification. Secondary structure constraints are specified in *dot-bracket* notation (i.e. with the explicit identification of pairing bases). Sequence constraints are specified as position specific nucleotide types. Finally the GC ratio is the proportion of G and C nucleotides present in a given sequence, a constraint which allows to tailor the satisfying sequences to specific organisms.

3.1 Constraints learning

We derive structural and sequence constraints from important subgraphs identified in each sequence. We start by computing the optimal minimum free energy (MFE) structure of each sequence. We then score each node with the importance notion detailed in Section 2.2 and remove nodes that have an importance score below a user defined threshold. Thresholding allows us to identify the corresponding connected components. Since AntaRNA accepts only discrete constraints we define the sequence constrains as being the nucleotides in an unpaired region that belong to components with a size larger than a second user specified threshold. Analogously for structural constraints we extract all base pairs between important nodes in components of user defined minimal size. In the two cases the user can specify different thresholds. As for the desired GC ratio, this can be computed directly from the input sequence. As an example of the generated constraints consider Figure 2.

4 Experimental results and conclusions

In order to perform an in-silico evaluation of the quality of the constructed sequences we perform a learning curve analysis: we compare the predictive performance of a classifier trained over the original sequences against a classifier trained over a dataset that additionally includes the sequences sampled by the proposed approach; if the samples

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

belong to the same functional class and in addition are also non redundant, then we expect a higher predictive performance for the latter case over a held out set, since the training set has virtually increased.

We retrieve 42 seed sequences for family RF02276 (hammerhead rybozyme) from the RFam database [2] and use a stochastic gradient descent SVM classifier over the minimum free energy graphs computed using the RNAfold program [12]. Negative sequences are obtained by di-nucleotide random shuffling of the seed sequences. The code is available at `https://github.com/fabriziocosta/RNAsynth`.

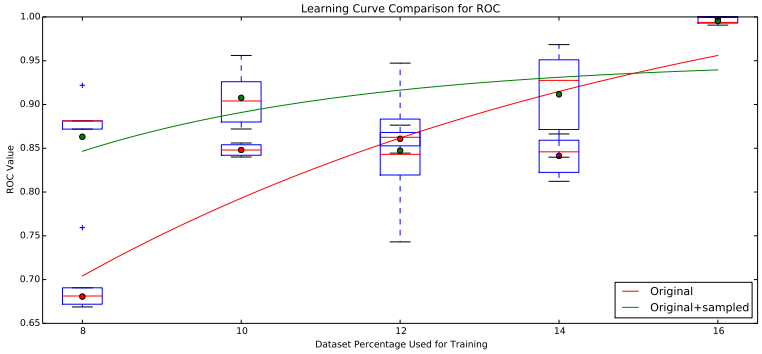

Fig. 3: Area under the ROC curve predictive performance of a SGD SVM as a function of the fraction of rybozyme sequences. Dots mark the mean ROC value, the boxplots mark the median value and the quartiles. The continuous line is the optimal fit of a saturating function.

From these experiments we see a significant effect when additional sequences are added to an increasingly large training set in a ratio of 2 to 1. This encouraging results seem to indicate that it is possible to frame the construction of non coding RNA sequences using the proposed approach where a generic data driven module identifies constraints suitable for a domain specific inverse folding algorithm.

In another set of experiments (on the RF00005 family) we have used RNASynth to *improve* on atypical sequences. We used a state of the art context free grammar approach known as covariance model (CM) [7] to compute the RNA family membership probability of each sequence. Using the sequences with the smallest score as seeds and filtering the results according to the SVM predictor we were able to obtain sequences with significantly higher membership probability in more than 13% of the cases.

Note that CMs can be also run in generative mode to emit sequences that belong with highly probability to a modeled family and can therefore be used in a similar fashion as RNASynth. This approach however has two major shortcomings: 1) it is parametric and 2) it can represent only a single (consensus) configuration. As a consequence of the first limitation, CMs cannot deal with structures that exhibit complex interactions known as pseudo-knots. The second limitation is of interest when a set of RNAs that share the same function possess two or more structural configurations that differ sig-

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

nificantly. For these reasons we need a more flexible approach, like RNASynth, that is non-parametric and that can model an arbitrary number of structural configurations. In future work we will demonstrate the advantage of these properties using the more recent AntaRNA version to include pseudo-knots as constraints.

References

[1] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[2] Sarah W. Burge, Jennifer Daub, Ruth Eberhardt, John Tate, Lars Barquist, Eric P. Nawrocki, Sean R. Eddy, Paul P. Gardner, and Alex Bateman. Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res*, 41(Database issue):D226–32, 2013.

[3] F. Costa and K. De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262, 2010.

[4] Camille J. Delebecque, Ariel B. Lindner, Pamela A. Silver, and Faisal A. Aldaye. Organization of intracellular reactions with rationally designed RNA assemblies. *Science*, 333(6041):470–4, 2011.

[5] Camille J. Delebecque, Pamela A. Silver, and Ariel B. Lindner. Designing and using RNA scaffolds to assemble proteins in vivo. *Nat Protoc*, 7(10):1797–807, 2012.

[6] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. The MIT press, One Rogers Street, Cambridge, MA, USA, 2004.

[7] S R Eddy and R Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Res.*, 22(11):2079–2088, 11 June 1994.

[8] Paolo Frasconi, Fabrizio Costa, Luc De Raedt, and Kurt De Grave. klog: A language for logical and relational learning with kernels. *Artificial Intelligence*, 217:117–143, 2014.

[9] Peixuan Guo. The emerging field of RNA nanotechnology. *Nature Nanotechnology*, 5:833–842, 2010.

[10] D. Haussler. Convolution kernels on discrete structures. Technical Report 99-10, UCSC-CRL, 1999.

[11] R. Kleinkauf, M. Mann, and R. Backofen. antaRNA – ant colony based RNA sequence design. *Bioinformatics*, 2015. Published Online.

[12] Ronny Lorenz, Stephan H. Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA Package 2.0. *Algorithms Mol Biol*, 6:26, 2011.

[13] Rebecca M. Terns and Michael P. Terns. CRISPR-based technologies: prokaryotic defense weapons repurposed. *Trends in Genetics*, 30(3):111–118, 2014.