Superawesome Graph Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract

[1]Nulla pulvinar ante rutrum efficitur pellentesque. Donec augue tortor, dapibus vitae quam at, ultricies bibendum mauris. Etiam est nisi, ultricies vitae est euismod, rutrum eleifend ligula. Donec dictum orci ullamcorper ipsum vulputate gravida. Phasellus dignissim commodo feugiat. Proin bibendum interdum malesuada. Phasellus nibh dolor, pulvinar vel hendrerit vitae, egestas eu mi. Aenean ornare mauris purus, porta euismod turpis tristique eget. Maecenas magna velit, tempus aliquet nunc quis, vulputate faucibus dui. Cras egestas viverra libero faucibus vehicula. Donec molestie lectus in mattis convallis.

1 Introduction

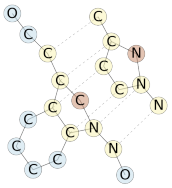

Figure 1: Sample figure caption.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tempus nunc tempus, pretium leo sed, blandit ante. Nulla sit amet consectetur nunc. Curabitur dictum, risus in eleifend tincidunt, ipsum libero dapibus magna, eu lacinia purus ligula ut lorem. Sed vel eleifend massa, eget tincidunt erat. Nulla vel semper ipsum, quis tincidunt dolor. Duis convallis diam at auctor laoreet. Maecenas at lorem egestas, mollis felis sit amet, sagittis est. Sed eget lorem ut sapien porttitor malesuada. Fusce vehicula, nunc sit amet auctor ultrices, odio ligula gravida enim, sed bibendum risus quam sit amet justo. Donec eget mauris at tortor iaculis finibus. Suspendisse sollicitudin tellus quam, eget aliquam ipsum mollis vitae. Morbi et hendrerit eros, ut consequat justo. Ut blandit lobortis sem. Donec sem felis, scelerisque tincidunt nibh sit amet, eleifend consectetur magna. Nunc ultricies gravida commodo. Morbi vestibulum tellus nec turpis imperdiet tempor

2 Method

Previously Costa [2] introduced a method on the construction of novel graphs according to a distribution given by example graphs. Graphs were vectorized via a *decomposition kernel* to train a machine learning model, e.g. an SVM. Fragments of the graphs would be collected in a *grammar* (resembling a string grammar) to alter the set of graphs incrementally. Changes to a graph are evaluated with the model. We present a method to increase the flexibility of the graph grammar.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

Modification to the grammar. We work with different CIPs that consider a contracted version of the original graphs. We obtain a contracted graph G' by contracting edges in G. After a contraction, the set of contracted vertices of the created vertex is accessible with the $contracted$ function. We extract $C^v_R(G')$ and $I^v_{R,T}(G')$ as usual, but from G'. The core graph $C^v_R(G', G)$ is induced by the nodes $\bigcup_{u \in C^v_R(G')} contracted(u)$. The new interface graph $I^v_{R,B}(G', G)$ is then obtained by the nodes $\{w | d(w, v) \leq B \wedge v \in C^v_R(G', G) \wedge w \in G \wedge w \notin C^v_R(G', G)\}$. B is the thickness of the base graph. At this point we can construct a CIP from $C^v_R(G', G)$ and $I^v_{R,B}(G', G)$. To find a congruent CIP, we previously only compared the hashed $I^v_{R,T}(G)$ graphs. By hashing the hashes of $I^v_{R,T}(G')$ and $I^v_{R,B}(G, G')$ we increase the specificity of this comparison. The vertices in $I^v_{R,B}(G, G')$ might have been relabeled to represent a concept of its $contracted$ set. In our test, we will contract according to the secondary RNA structure and label the resulting vertex accordingly e.g. 'Hairpin loop'. This way we encode far reaching and abstract information about the surrounding vertices in our CIP interface matching.

An improvement to the notion of congruency. For CIPs to be congruent, isomorphism is required. We expand this requirement to incorporate the distance to nodes in the core graph when determining if graphs are congruent. $\forall u \in I^v_{R,T}(G) : \min_{z \in C^v_R(G)} d(u, z) = \min_{z' \in C^{v'}_R(G')} d(\phi(u), z')$ i.e. the distance to the closest core node is equal for every u and $\phi(u)$.

Extending what we can contract. Edge contraction is one way to obtain a contraction graph. One might also contract nodes that are not connected by an edge. We did so with multi-loop nodes except stem nucleotides. Although the underlaying EDeN kernel does not support directed graphs, the grammar may contain directed graphs. This is implemented by extracting CIPs as if undirected and creating the grammar as usual. A constraint to the contraction is that we need to be able to generate is algorithmicaly. In the sampling phase we recalculate the contraction after changes to the underlaying graph.

3 Evaluation

RNA sequences are sequences over the four nucleotides A,G,U and C. They have a 5' and a 3' end, meaning that the direction matters. Nucleotides in the sequences bind to other nucleotides so the sequence forms a *structure*. RNAs are grouped into functional families whose classification is a problem of biology. *Infernal*[3] can classify and create members of these families using covariance models. Infernal gives us a domain specific way to evaluate our Method for the generation of new graphs.

We chose our RNA sequences from the seed sequences of each family that Infernal is using[1]. Sequences were chosen from RNA families with a) many members, because some families contain 10 instances which is very few for a classification task. b) interesting structure, many sequences fold into structures that exhibit a large number of unpaired bases which would result in a very simple contracted graph c) similar length, because classification should no be trivial.

We evaluated our results on secondary RNA structure graphs, where each nucleotide is a vertex. The contracted graphs are obtained by contracting vertices that belong to the same structural element, e.g. adjacend stem vertices are contracted into a single vertex. Secondary structures are obtained by *muscle*-aligning [4] with the four closest neighbors by sequence and folding with *RNAalifold* [5]. For RNA sequences, the direction in which a sequence is read is important. Working with directed graphs in the grammar will enable us to extract the sequence along the backbone easily as well as adapt the model to the directed biological reality.

We evaluate our generated sequences against the Infernal model. The Infernal model is human optimized and thus highly reliable. **INFERNALEVAL** shows the average bit-score of generated graphs and the size of the training set. Each family has its own quality theshold. Sequences above that threshold (marked in black) can reliably be considered part of the family. We observe that **y out of x** generated graphs.

Costa [2] defines *the constructive learning problem for finite samples* as mimicking the distribution of given graphs while producing new instances. We can estimate this distribution with a machine learning model and compare two distributions by evaluating a set of test instances on the models. We can also

meassure how different generated sequences are by comparing the edit distance. **SOMEGRAPHZ** shows the learning curves for the original set of graphs and the generated set. Notice, that they are very similar. We can also see the average edit distance for the generated graphs to their closest match in the original set.

The method presented here is adapting an older method to new problems. **SOMEGRAPHZ** is a repetition of the first experiment with the old algorithm. We observe that its not producing reliable graphs.

4 Discussion

We have compared the algorithm to its predecessor and shown that its result hold up in a state of the art, domain speciffic evaluation. Its drawback is the reliance on the ability to generate a contraction. In some domains this contraction is obvious as in chemical compounds cycles or chargemaps are good candidates. For the general case the contraction can be learned.

5 todo

ok so there is nthis new experiment idea which plays nice with the formula that fabrizio wrote for the problem definition: same distribution but different instances- should be learned. use the learning curves to show that original and sample are similar, then use average edit distance to shot that i accieved that with different instances :)

References

[1] M. Marshall S. Griffiths-Jones, A. Bateman. Rfam: an rna family database. *Nucleic Acids Res.*, 31:439–441, 2003.

[2] F. Costa and D. Sorescu. The constructive learning problem: an efficient approach for hypergraphs. *Workshop on Constructive Machine Learning (CML) NIPS, Lake Tahoe, Nevada, USA, 10 December*, 2013.

[3] S. R. Eddy E. P. Nawrocki. Infernal 1.1: 100-fold faster rna homology searches. *Bioinformatics*, 29:2933–2935, 2013.

[4] Robert C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, 32:1792–1797, 2004.

[5] Ivo L. Hofacker Stephan H. Bernhart. Rnaalifold: improved consensus structure prediction for rna alignments. *BMC Bioinformatics*, 9:474, 2008.