Title

Semyon Grigorev, Polina Lunina
Saint Petersburg State University
7/9 Universitetskaya nab., St. Petersburg, 199034, Russia
semen.grigorev@jetbrains.com, lunina_polina@mail.ru

Algorithms that can efficiently perform sequences classification and subsequences detection have recently become a focus in bioinformatics and many of them utilize the idea about considering these sequences secondary structure [1, 2, 3, 4]. One of the classical ways of describing secondary structure is formal grammars.

In the work [5] we proposed an approach for biological sequences processing by combination of formal grammars and neural networks. We use context-free grammar to describe only the main secondary structure features (opposed to the classical way when the purpose of grammar is to model secondary structure of the full sequence). Then we extract these features by parsing algorithm and perform some sort of classification on the parsing-provided data by using neural network.

In this work, we provide the further development of the proposed approach usage on several tRNA sequences analysis problems: classification of tRNA into 2 classes (eukaryotes and prokaryotes) and 4 classes (archaea, bacteria, plants, and fungi). We use the parsing-provided data as input to neural network, therefore we need to transform it into some readable format. The result of matrix-based parsing algorithm for an input string and fixed nonterminal is an upper triangular boolean matrix. Presently, we came up with two possible ways of these matrices representation. The first one is to drop out the bottom left triangle, vectorize the rest of matrix row by row and transform it to the byte vector. It requires the equal length of the input sequences, therefore we propose to either cut sequences or add some special symbol till the definite length. The second way is to represent the matrix as

an image: the false bits of matrix as white pixels and the true bits as black ones. This approach makes it possible to process sequences with different length since the images could be easily resized to the same constant size. Besides, each approach requires certain network architecture: for vectorized data we use dense layers because data locality is broken during vectorization and dropout layers with batch normalization to stabilize training. For image data we use a small number of convolutional layers, then linearization and then go to the same architecture as for vectorized data.

The most time-consuming operation here is parsing. The possible solution is to add an extra step to the approach describes above. Let us say, we have a trained neural network (vector or image based) that takes parsing-provided data as an input and performs classification of some kind, so, in order to improve its accuracy or test it on a huge amount of data, we can create another neural network that takes the initial nucleotide sequence as an input and converts it to the input for the trained classifier. The evaluation of both image- and vector-based classifiers extensions is presented in the table 1.

		Base model accuracy	Extended model accuracy
Eukaryotik/prokaryotik classifier	Vector-based	94.1%	97.5%
	Image-based	96.2%	97.8%
Archaeal/bacterial/fungi/plant classifier	Vector-based	86.7%	96.2%
	Image-based	93.3%	95.7%

Table 1: Test results

References

[1] Rivas E, Eddy S.R. *The language of RNA: a formal grammar that includes pseudoknots* // Bioinformatics. — 2000.

[2] Knudsen Bjarne, Hein Jotun. *RNA secondary structure prediction using stochastic context-free grammars and evolutionary history.* //Bioinformatics (Oxford, England).— 1999.— Vol. 15, no. 6.— P. 446–454.

[3] Yuan C. et al. *Reconstructing 16S rRNA genes in metagenomic data* //Bioinformatics. – 2015. – №. 12. – P. 135-143.

[4] Dowell R. D., Eddy S. R. *Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction* //BMC bioinformatics.– 2004.– №. 1.– P. 71.

[5] Grigorev S., Lunina P. *The Composition of Dense Neural Networks and Formal Grammars for Secondary Structure Analysis*