Improved Architecture of Artificial Neural Network for Secondary Structure Analysis

Semyon Grigorev[1], Polina Lunina[1]

[1] *Saint Petersburg State University, JetBrains Reserach, St. Petersburg, Russia*

E-mail: *semyon.grigorev@jetbrains.com*

Motivation

An approach for biological sequences processing by combination of formal grammars and neural networks is proposed in the work [**?**]. While classical way is to model secondary structure of the full sequence by using grammar, the proposed approach utilizes it only for primitive secondary structure features description. These features can be extracted by parsing algorithm and processed by using artificial neural network. It is shown that this approach is applicable for real-world data processing. Our goal is to improve this approach.

Questions

Is it possible to move parsing to network training step? Parsing is the most time-consuming operation of the proposed solution, so whis way we can improve performance of final solution.

Is it possible to use convolutional neural networks for parsing result processing? The result of parsing algorithm is a set of buulean matrices. The original solution uses vectorized representation of these matrices, and DNNs for these vectors processing. Matrces can be treated as bitmaps. This way we can simplify data normalization: we can resize figures if input sequences have different lengths.

Results

We use the proposed improvements to create neural networks for tRNA sequences analysis problems: classification of tRNA into 2 classes: eukaryotes and prokaryotes (EP) and 4 classes: archaea, bacteria, plants and fungi (ABFP). We train networks on sequences from databases [**?**, **?**]. Results for both image- and vector-based classifiers are presented in the table, where base model means network which handles parsing result and extended model handles sequences and is based on the corresponding base model.

Classifier	EP		ABFP	
Approach	Vectors	Images	Vectors	Images
Base model accuracy	94.1%	96.2%	86.7%	93.3%
Extended model accuracy	97.5%	97.8%	96.2%	95.7%
Total samples (train:valid:test)	20000:5000:10000		8000:1000:3000	

Solution Overview

Grammar

```
s1: stem<s0> any
a_0_7 : any*[2..10]
s0: a_0_7 | a_0_7 stem<s0> s0
any: A | U | C | G
stem1<s>: A s U | G s C | U s A | C s G
stem2<s>: stem1< stem1<s> >
stem<s>:
    A stem<s> U
    | U stem<s> A
    | C stem<s> G
    | G stem<s> C
    | stem1< stem2<s> >
```

Fixed context-free grammar describes features of secondary structure and can be tuned in order to increase result quality.

Sequences

Genoms parts of fixed length. Current length is 512. Length is variable parameter and can be changed in order to increase quality of solution. Each sequence is treated as a text in $\{A, U, G, C\}$ alphabet.

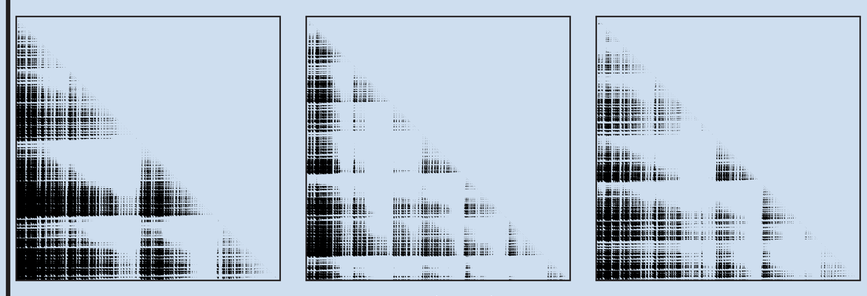
Result of classification

Currently we implement only simple binary classifier that separates 16s and non-16s sequences. One of interesting question is may this classifier be used for chimeric sequences filtering. We hope that it is possible because "global" secondary structure of chimeric sequences should be broken.

Parser

Parser is features extractor: it extracts features of given sequence secondary structure. Parsing algorithm is based on the algorithm [**?**], so the grammar can be extended with conjunctive rules [**?**] for pseudoknots description.Parser performance is a bottleneck of our solution. Current implementation utilizes GPGPU and should be optimized in future.

Matrices

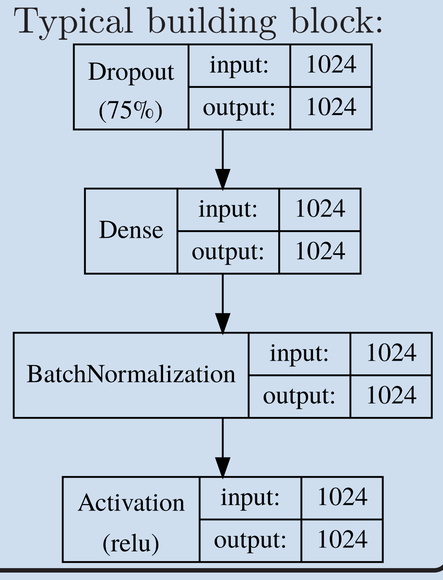


Parsing result is boolean (0-1) matrix which represents secondary structure features for sequence ω: cell $[i, j]$ contains 1 iff $\omega.[j, i]$ is derivable from `s1` and 0 in other case. These matrices can be handled as images or binary vectors by using respective types of neural networks (convolutional or binary) and one of topics for future research is to select better approach. Currently we compress each matrix to integer vector and use dense neural network.

Neural Network

We use dense neural network with 14 dense layers. Almost all of them are wrapped with dropout (up to 75%) and batch normalization layers for learning stabilization. Our network is trained on up to 310000 sequences of length 512: positive (16s rRNA) from NCBI database, negative (non-16s) from Green Genes database. Current accuracy for validation set (up to 81000 sequences) is 90%.

Typical building block:



Vectors

We use line-by-line compressed matrix representation: sequence of 32 cells (bits) is compressed to unsigned integer. Top right triangle of matrix is always empty, so can be ignored. We hope that compression to 16 bit integer or byte may decrease complexity of neural network and improve result quality, but it requires significantly more memory on GPGPU which can be serious technical problem.

Future Research

- Chimeric sequences filtration
- Secondary structure prediction
- Proteins functions prediction

References

Acknowledgments

The research was supported by the

Information

All materials available on GitHub: