16s rRNA Detection by Using Neural Networks

Neural networks for secondary structure information processing

Semyon Grigorev[1], Polina Lunina[1]

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

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

Motivation

Algorithms that can efficiently and accurately identify and classify bacterial taxonomic hierarchy have become a focus in computational genetics. The idea that secondary structure of genomic sequences is sufficient for solving the detection and classification problems lies at the heart of many tools [?, ?, ?, ?]. The secondary structure can be specified in terms of formal grammars. The sequences obtained from the real bacteria usually contain a huge number of mutations and "noise" which renders precise methods impractical. Probabilistic grammars and covariance models (CMs) are a way to take the noise into account [?]. For example, CMs are successfully used in the Infernal tool.Neural networks is another way to deal with "noisy" data. The works [?, ?] utilize neural networks for 16s rRNA processing and demonstrate promising results.

Results

- We propose the graph parsing algorithms based on different parsing techniques [?, ?, ?].
- We solve some problems of existing approaches (such as cycles processing problem, [?]).
- Our solution provides an ability to use GPGPU and multi-core systems for graph parsing which can be useful for large biological data analysis.

Performance comparison of context-free querying algorithms

Graph	#edges	#results	GLL(ms)	GPGPU(ms)
g_1	8688	141072	1926	82
g_2	14712	532576	6246	185
g_3	15840	449560	7014	127

Context-free path querying

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.

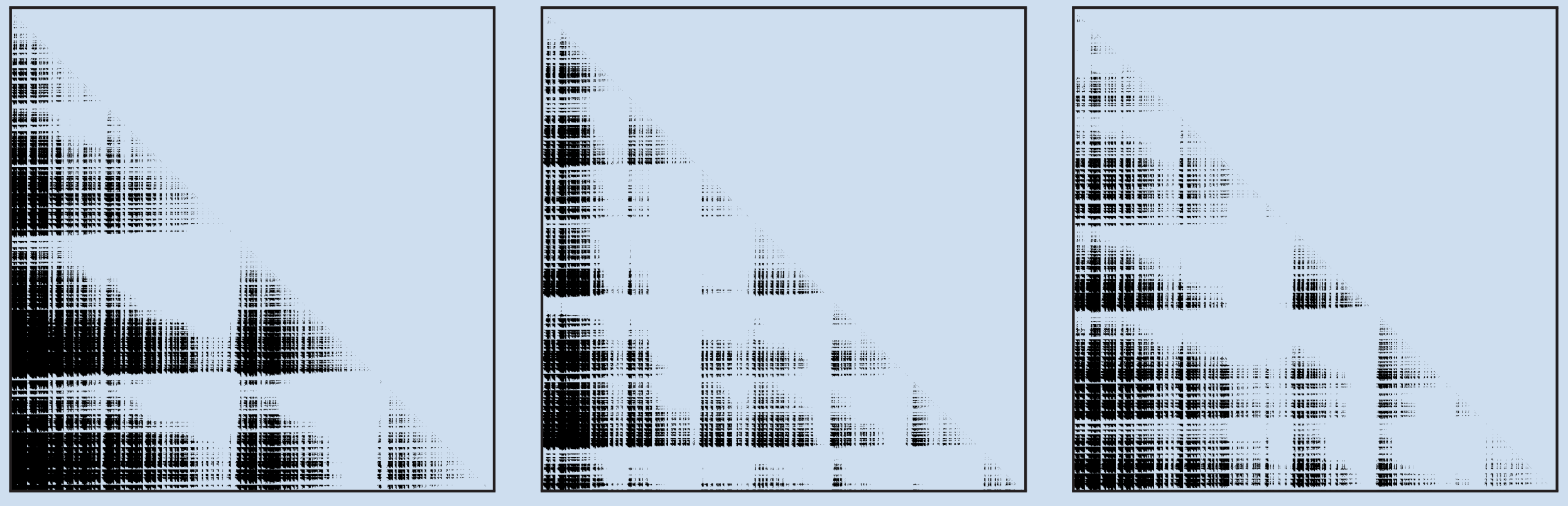
Sequences

Genom parts of fixed lengs. Current length is 512. Length is variable parameter and can be changed in order to increase quality of solution.

Parser

Parser extracts features of secondary strcture. Parsing algrithm is based on Okhotin [?] algorithm, so the grammar can be extended with cinjunctive rules for pseudocnots description. Implementation utilizes GPGPU.

Matrices



Parsing result is boolean (0-1) matrix which represents secondary structure features for seqence ω: cell $[i,j]$ contains 1 iff $\omega.[j,i]$ is derivale from `s1` and 0 in other case.

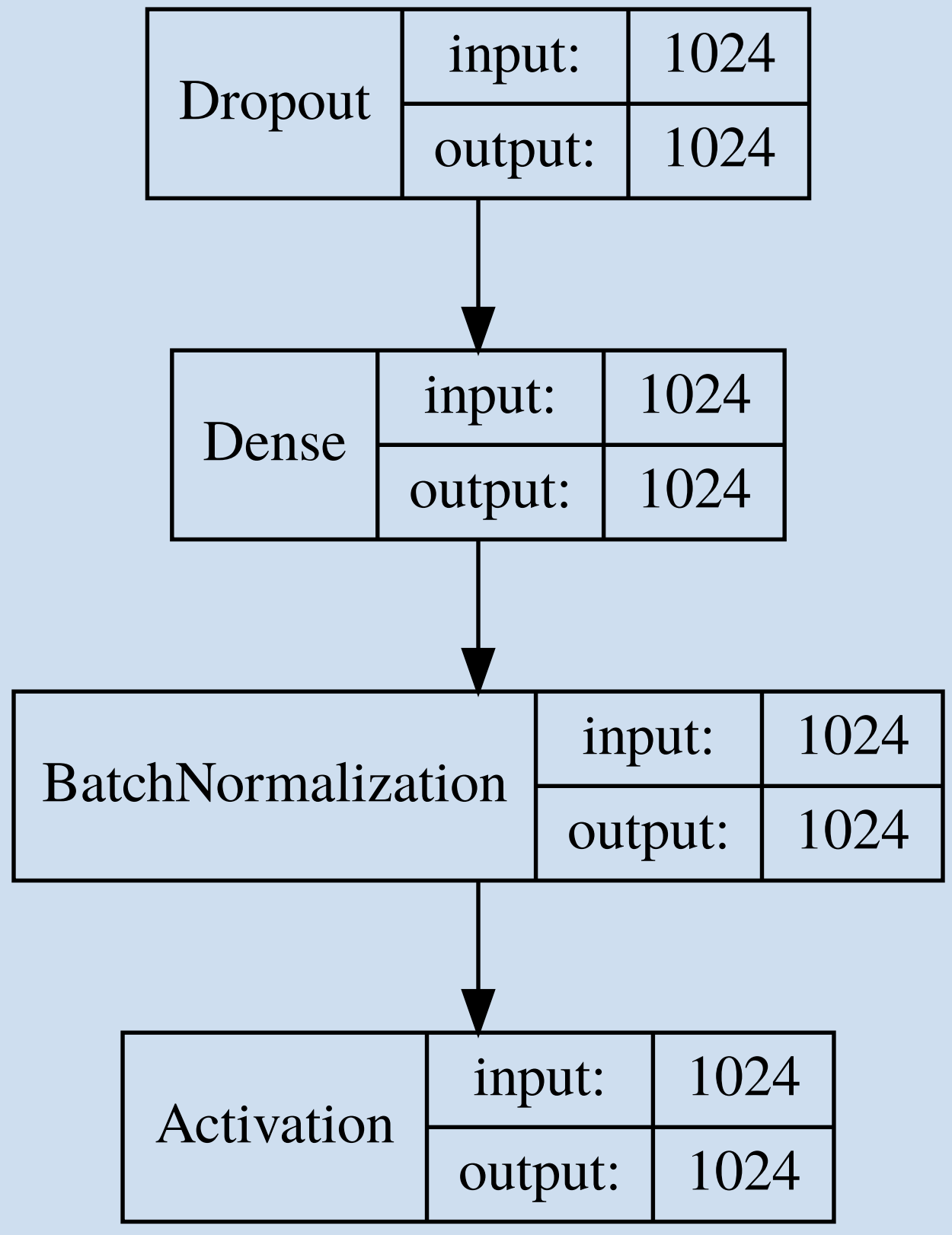
Vectors

Line-by-line compressed matrix representation: sequence of 32 cells is compressed to unsigned integer. Top right triangle of matrix is always empty, so can be ignored.

DNN

Dense newral network with 10 dense layers. 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:



Result of classification

Currently we implement just binary classifier that separates 16s and non-16s sequences.

Database querying

One of the examples of database querying is an analysis of graphs where vertices correspond to entities and concepts such as gene or phenotype while edges represent the known relationships such as "codes for", "interacts with", etc.

Example of graph structured data [?] is presented below.

Querying paths with special constraints may shed light upon unknown before links between vertices, forming the basis for new hypotheses.

References

Metagenomic assemblies analysis

Metagenomic assemblies can be presented as graph structured data. Some sequences have specific secondary structure, which can be described in terms of a context-free grammar, and this grammar can be used for searching and classification.

Acknowledgments

This work is supported by grant from JetBrains Research.

Information

All materials available on GitHub: https://github.com/YaccConstructor