BIOINFORMATICS
2019

The Composition of Dense Neural Networks and Formal Grammars for Secondary Structure Analysis

Polina Lunina, **Semyon Grigorev**

JetBrains Research, Programming Languages and Tools Lab
Saint Petersburg University

Febrary !!!, 2019

Problem Statement

- Sequence classification
  - Lexer and parser
  - Translator
  - Types mapping
  - Headers files processing
  - ...
- Unify kernels on client side
  - Currently native Brahma.FSharp's kernel and kernel loaded by type provider are different types
- Improve mechanism of kernels composition

Secondary structure!!!

- Key is secondary structure
- Problem: hight variability.
- Solutions: PCFG, CM

Our receip: Parsing + DNN

- Idea: not seci=ondary structure modelling, but features extraction!

Our receip: Parsing + DNN

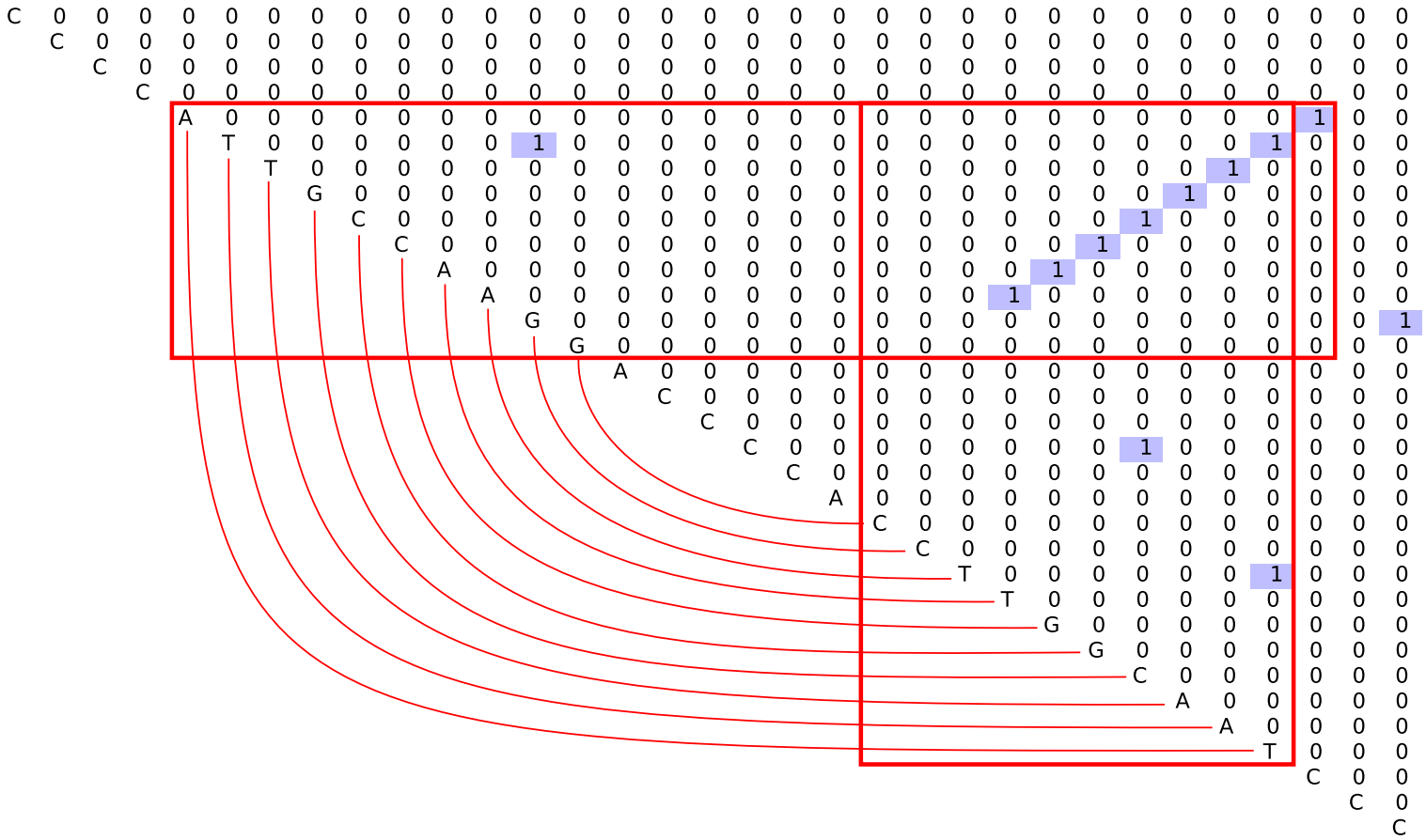
- Idea description. Figure

Grammar

```
s1: stem<s0>
any_str : any_smb*[2..10]
s0: any_str | any_str stem<s0> s0
any_smb: A | T | C | G
stem1<s>:                    \\ stem of height exactly 1
      A s T | G s C | T s A | C s G
stem2<s>:                    \\ stem of height exactly 2
      stem1< stem1<s> >
stem<s>:                     \\ stem of height 3 or more
      A stem<s> T
    | T stem<s> A
    | C stem<s> G
    | G stem<s> C
    | stem1< stem2<s> >
```

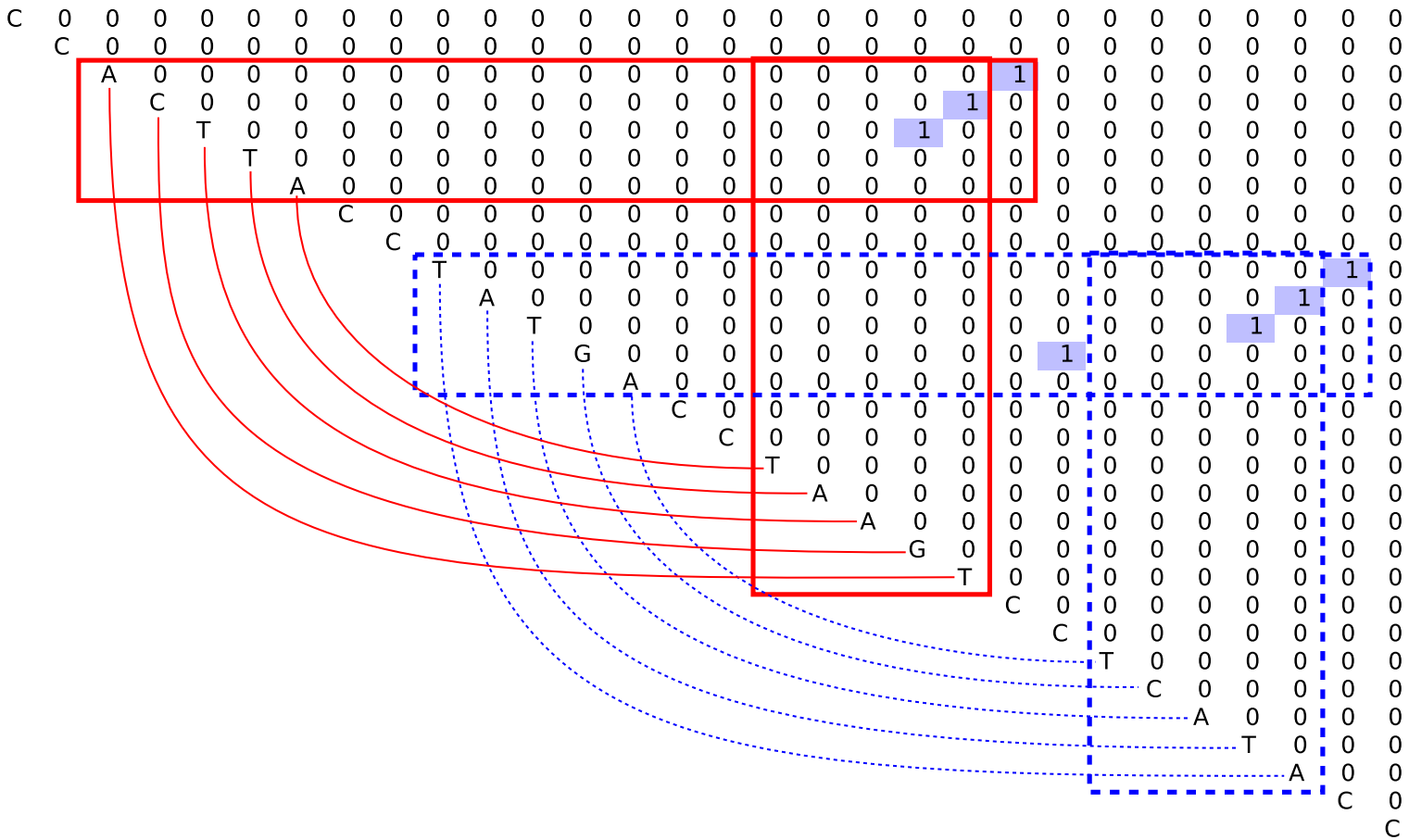
Example 1: Stem

$$\omega_1 = \texttt{CCCCATTGCCAAGGACCCCACCTTGGCAATCCC}$$



Example 2: Pseudoknot

$$\omega_2 = \texttt{CCACTTACCTATGACCTAAGTCCTCATACC}$$

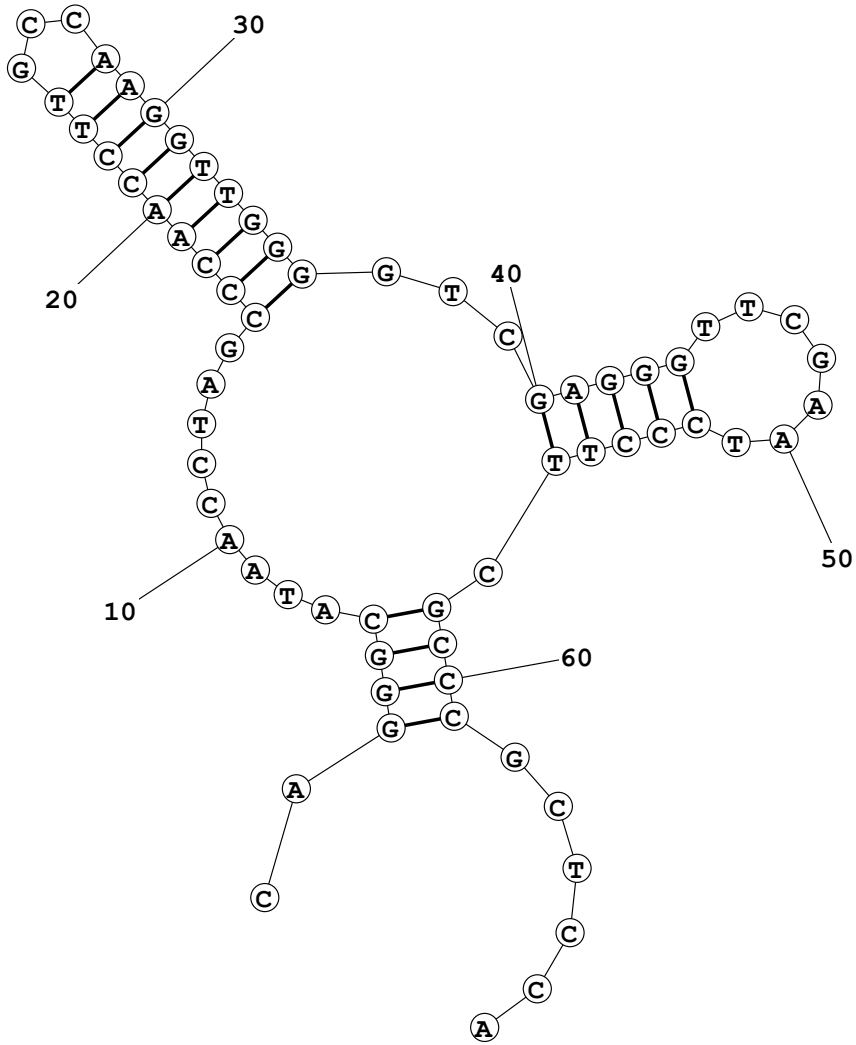


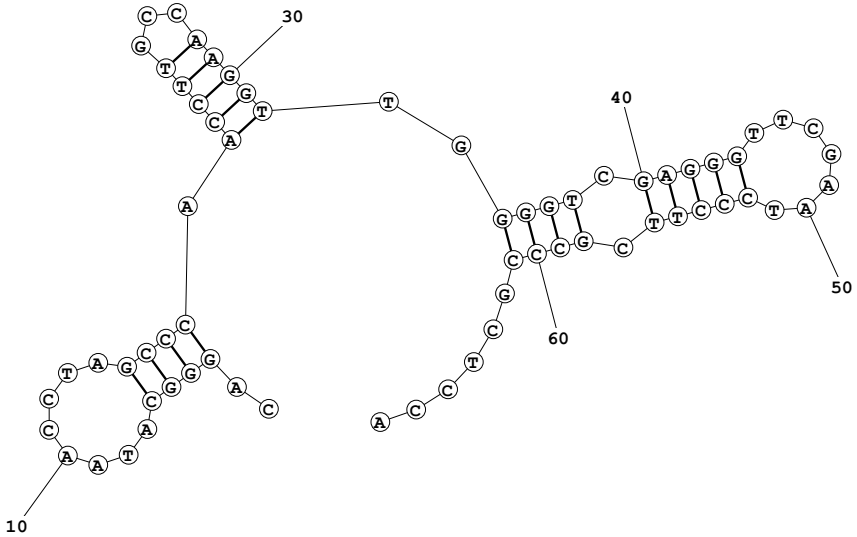
Example 2: Pseudoknot

$$\omega_3 =$$

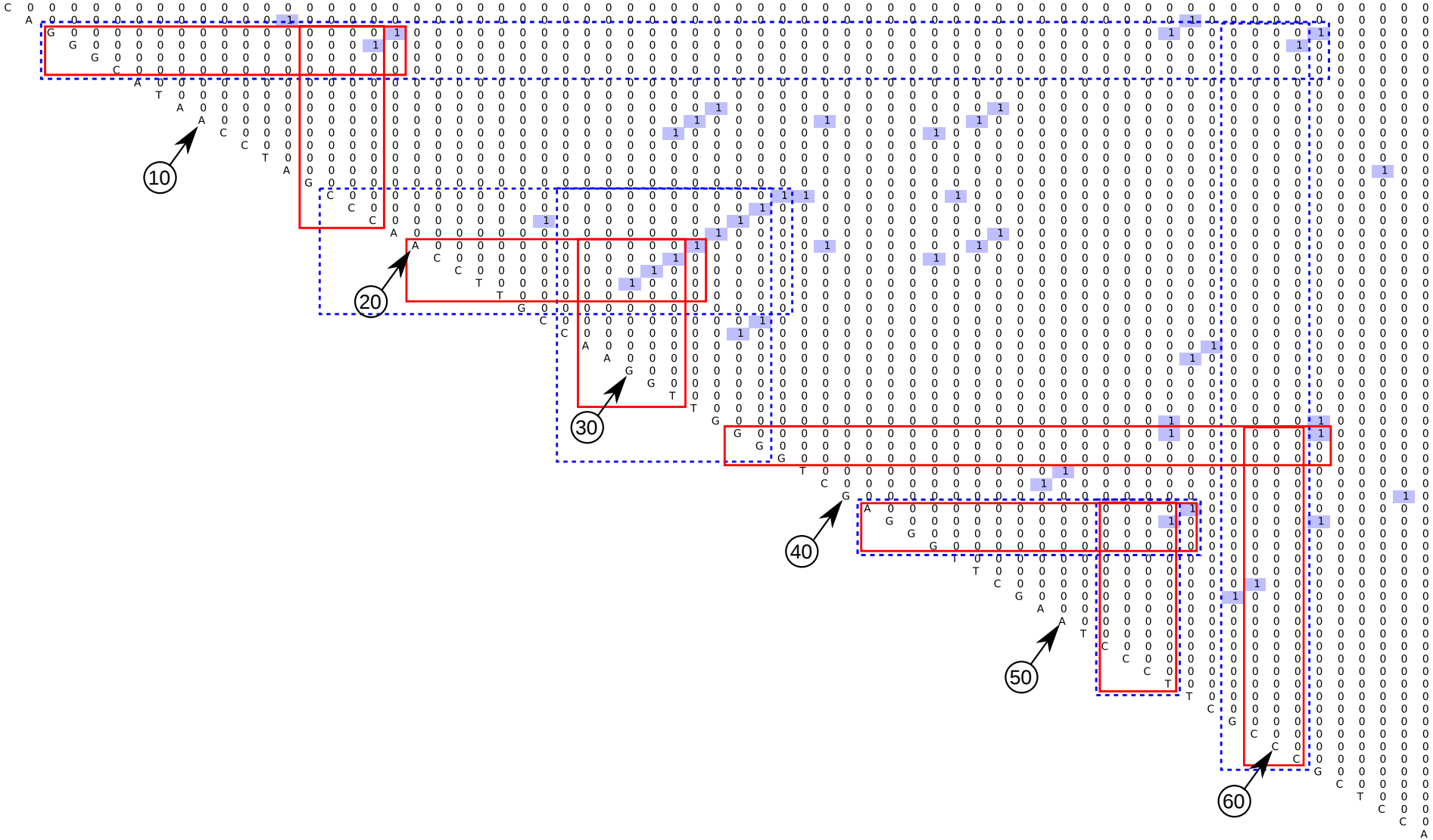
CAGGGCATAACCTAGCCCAACCTTGCCAAGG

TTGGGGTCGAGGGTTCGAATCCCTTCGCCCGCTCCA





Example 2: Pseudoknot



Evaluation setup

- Compile-time metaprogramming for types creation
  - Type provider is a **function which constructs type**

Evaluation results

- Compile-time metaprogramming for types creation
  - Type provider is a **function which constructs type**

Future work

- Compile-time metaprogramming for types creation
  - Type provider is a **function which constructs type**

Summary

- F# OpenCL C type provider
  - Type-safe integration of existing OpenCL C code in F# applications
  - Proof of concept

- Source code on GitHub:
  `https://github.com/YaccConstructor/Brahma.FSharp`
- Package on NuGet:
  `https://www.nuget.org/packages/Brahma.FSharp/`

Contact Information

- Semyon Grigorev: s.v.grigoriev@spbu.ru
- Kirill Smirenko: k.smirenko@gmail.com

- Brahma.FSharp:
  https://github.com/YaccConstructor/Brahma.FSharp

Thanks!