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

Sequences Analysis

- Sequences classification
- Subsequences seqrch
- ...

Sequences Analysis

- Sequences classification
- Subsequences seqrch
- ...

Problem: high variability of data

- Mutattions
- Different kinds of noice
- ...

Sequences Analysis

- Sequences classification
- Subsequences seqrch
- ...

Problem: high variability of data

- Mutattions
- Different kinds of noice
- ...

We need probabilistic approaches

Sequences Analysis

- Sequences classification
- Subsequences seqrch
- ...

Problem: high variability of data

- Mutattions
- Different kinds of noice
- ...

We need probabilistic approaches

Level of abstraction

- Plain text
- Secondary structure
- 3D structure

Sequences Analysis

- Sequences classification
- Subsequences seqrch
- ...

Problem: high variability of data

- Mutattions
- Different kinds of noice
- ...

We need probabilistic approaches

Level of abstraction

- Plain text
- Secondary structure
- 3D structure

We should handle secondary structure

Probabilistic approaches

- Hidden Markov's Models (HMM's)
- Probabilistic grammars
- Covariation Models (CM's)
- Artificial neural networks

Our receip: Parsing + DNN

- Idea: not secondary structure modelling, but features extraction!

Our receip: Parsing + DNN

- Idea description. Figure

Grammar

```
s1: stem<s0>
any_str : any_smb*[2..10]
any_smb: A | T | C | G
s0: any_str | any_str stem<s0> s0
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> >
```

Grammar

```
s1: stem<s0>
any_str : any_smb*[2..10]
any_smb: A | T | C | G
s0: any_str | any_str stem<s0> s0
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> >
```

Grammar

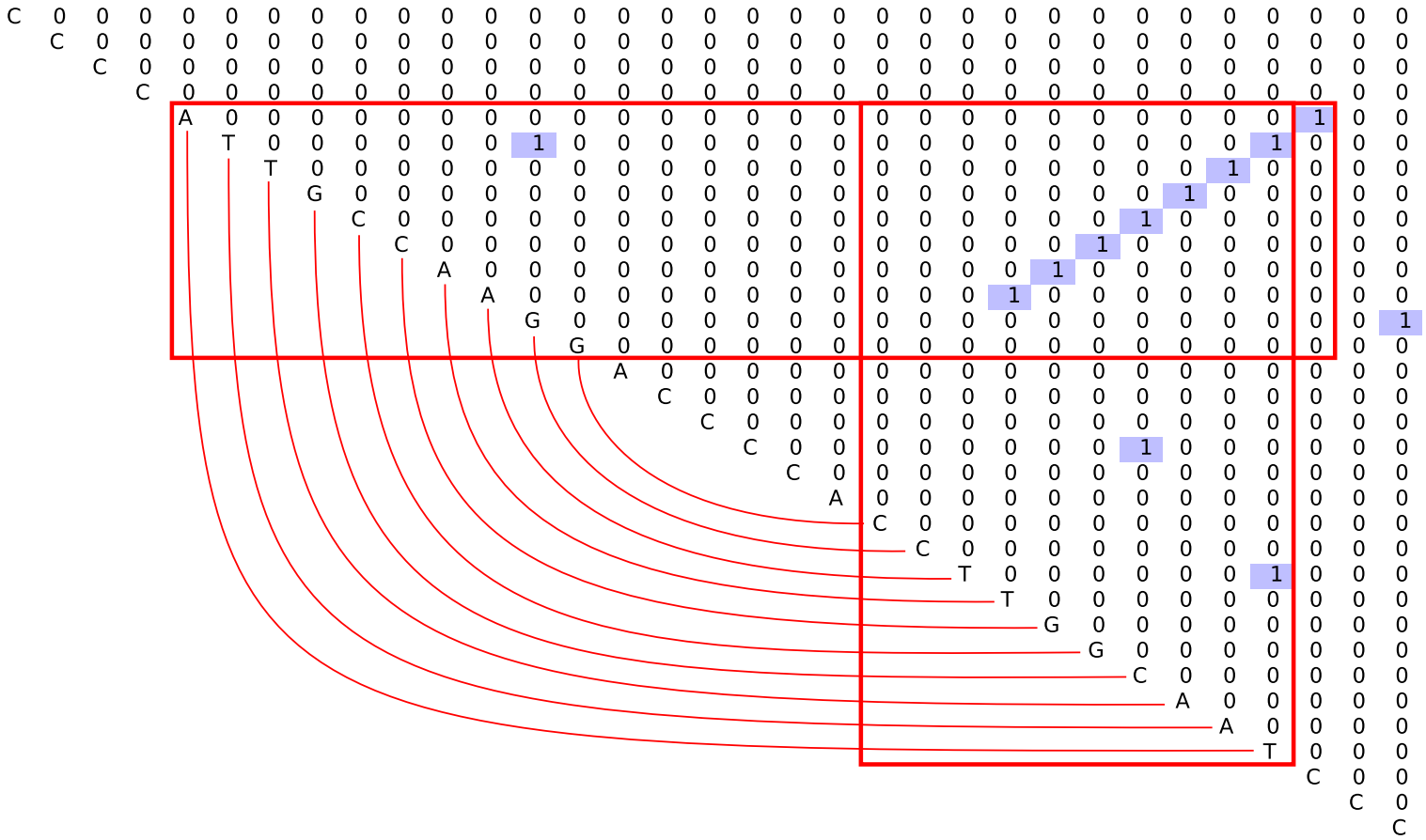
```
s1: stem<s0>
any_str : any_smb*[2..10]
any_smb: A | T | C | G
s0: any_str | any_str stem<s0> s0
stem1<s>:                    \\ stem of height exactly 1
      A s T | G s C | T s A | C s G
stem2<s>:                    \\ stem of height exactly 2
      stem1< stem1<s> >
stem<s>:                  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

Grammar

```
s1: stem<s0>
any_str : any_smb*[2..10]
any_smb: A | T | C | G
s0: any_str | any_str stem<s0> s0
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> >
```

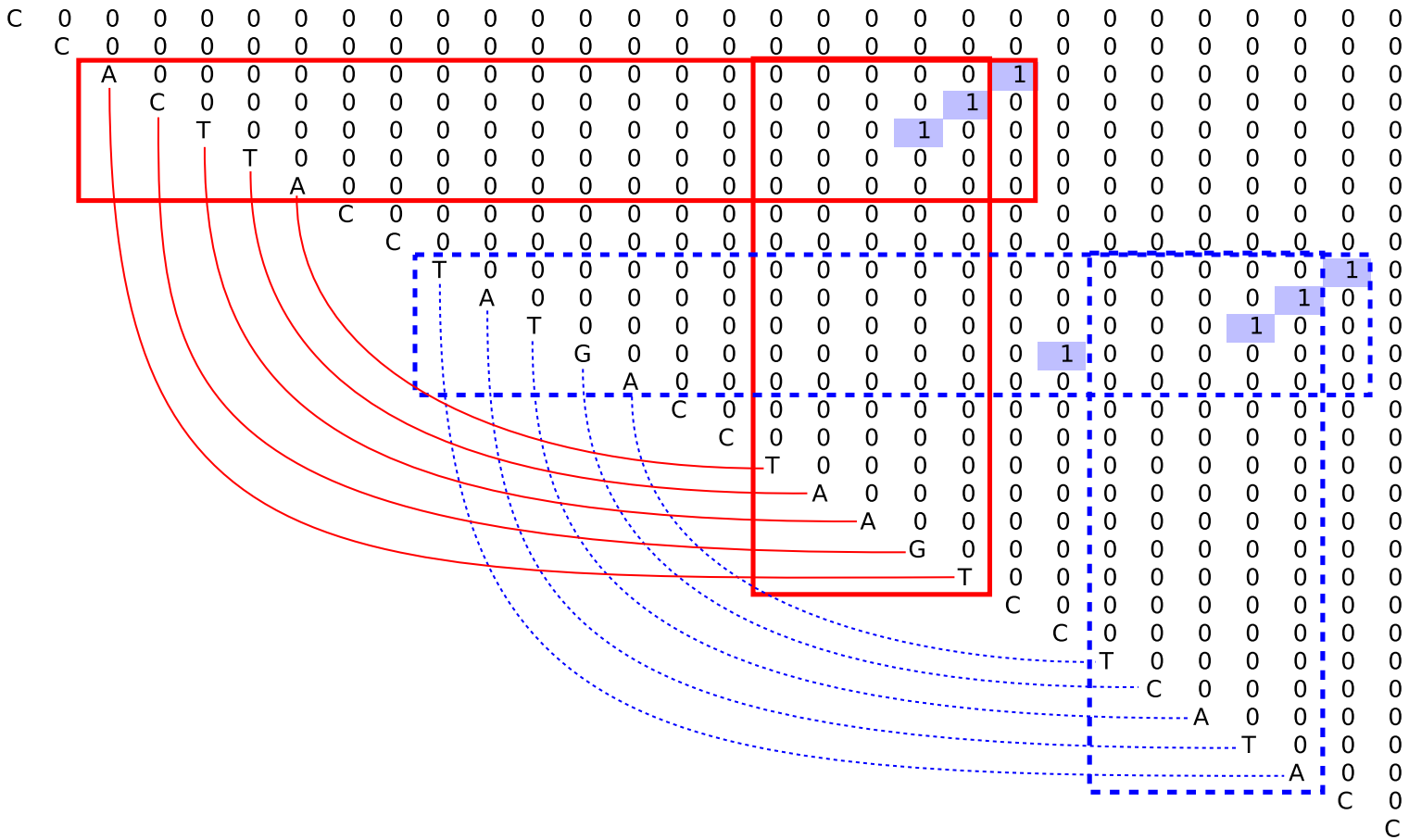
Example 1: Stem

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



Example 2: Pseudoknot

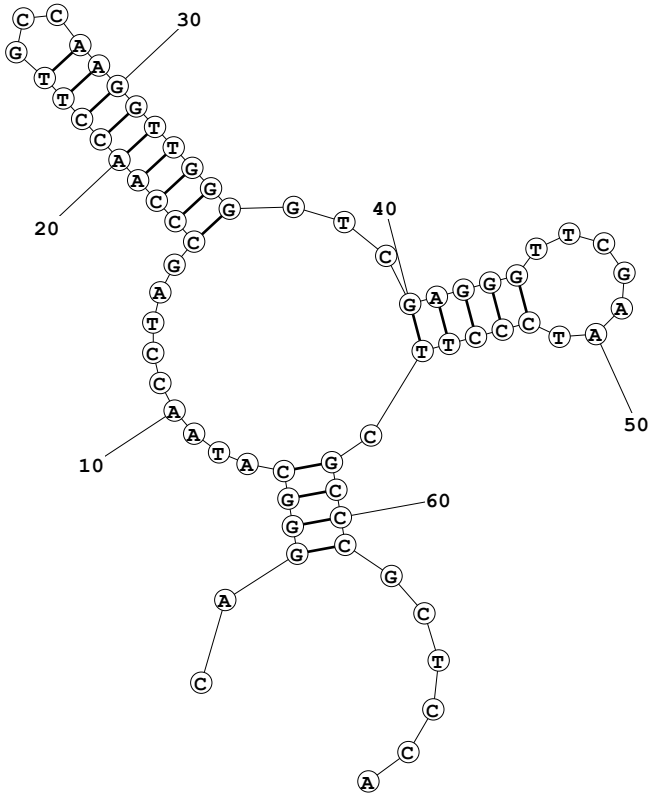
$$\omega_2 = \texttt{CCACTTACCTATGACCTAAGTCCTCATACC}$$

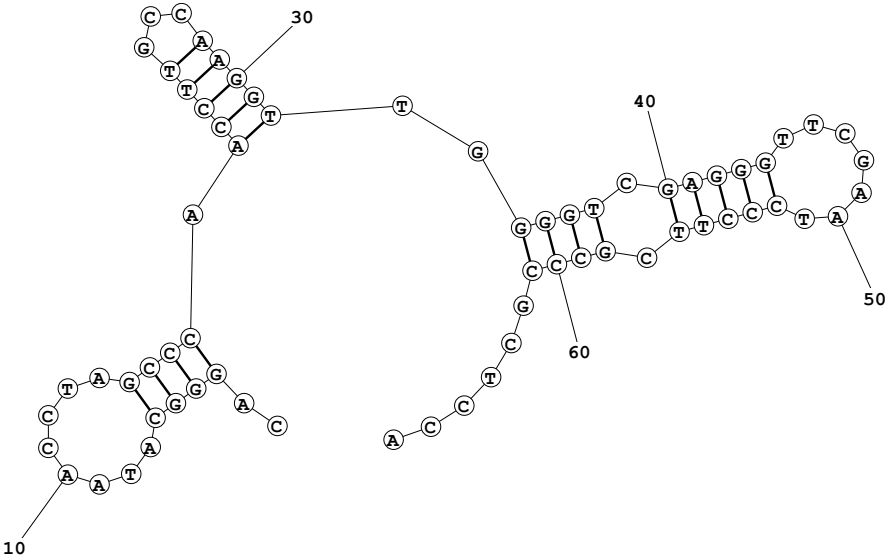


Example 3: real tRNA

$$\omega_3 = \texttt{CAGGGCATAACCTAGCCCAACCTTGCCAAGG}$$
$$\texttt{TTGGGGTCGAGGGTTCGAATCCCTTCGCCCGCTCCA}^1$$

Predicted secondary structures[2]

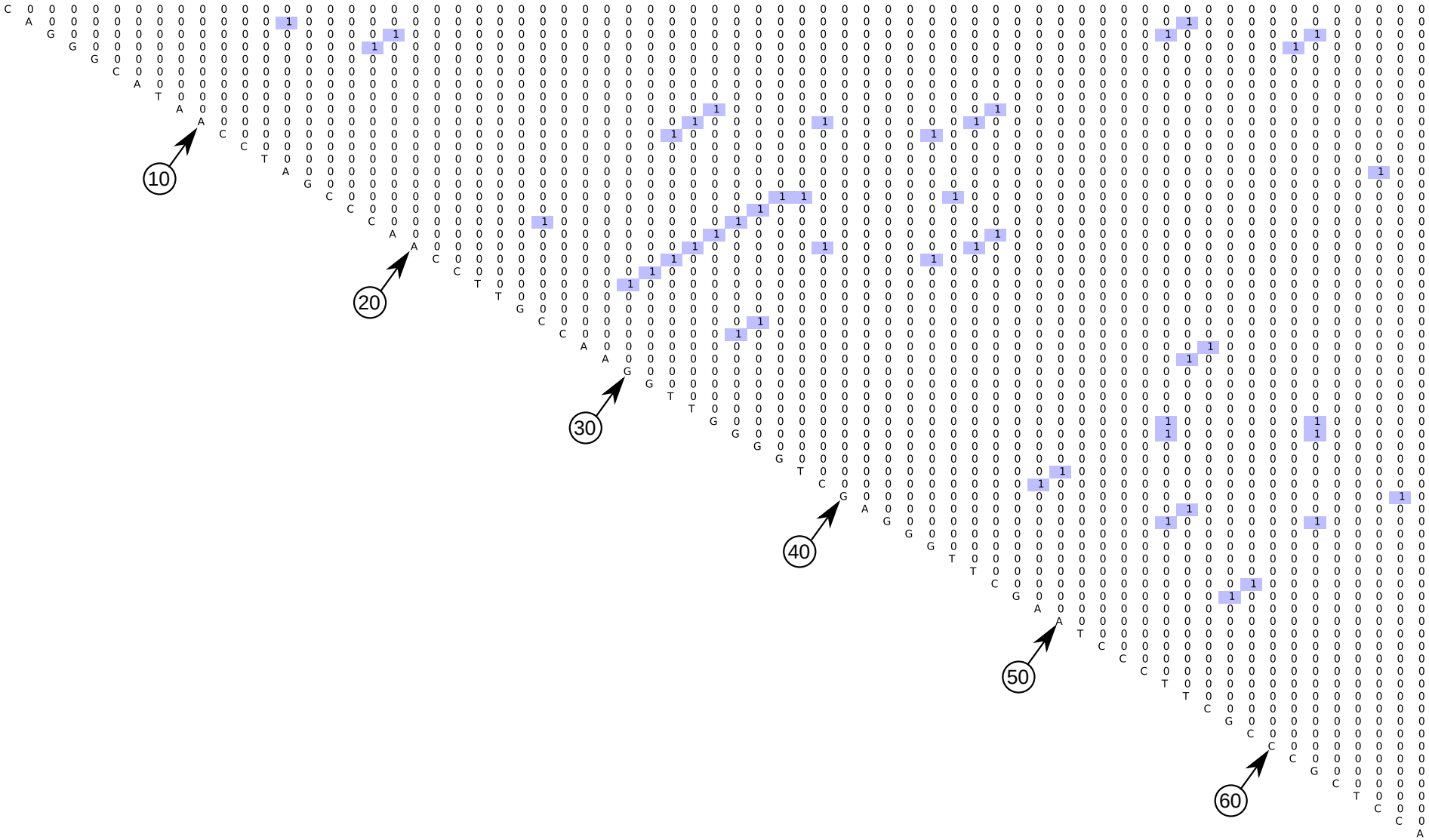




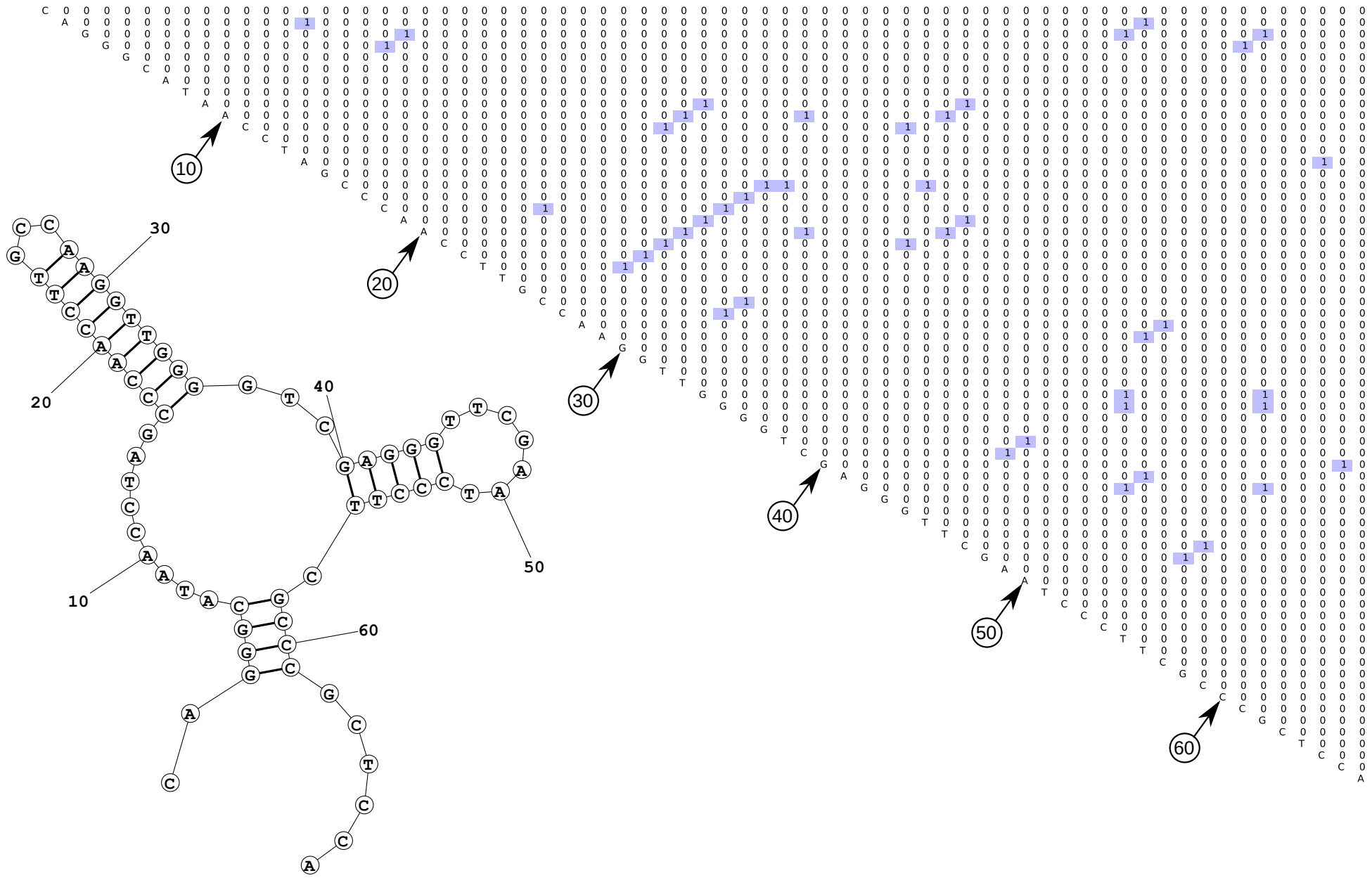
[1] Novosphingobium aromaticivorans from GtRNAdb.

[2] Results are given by using the Fold Web Server with default settings.

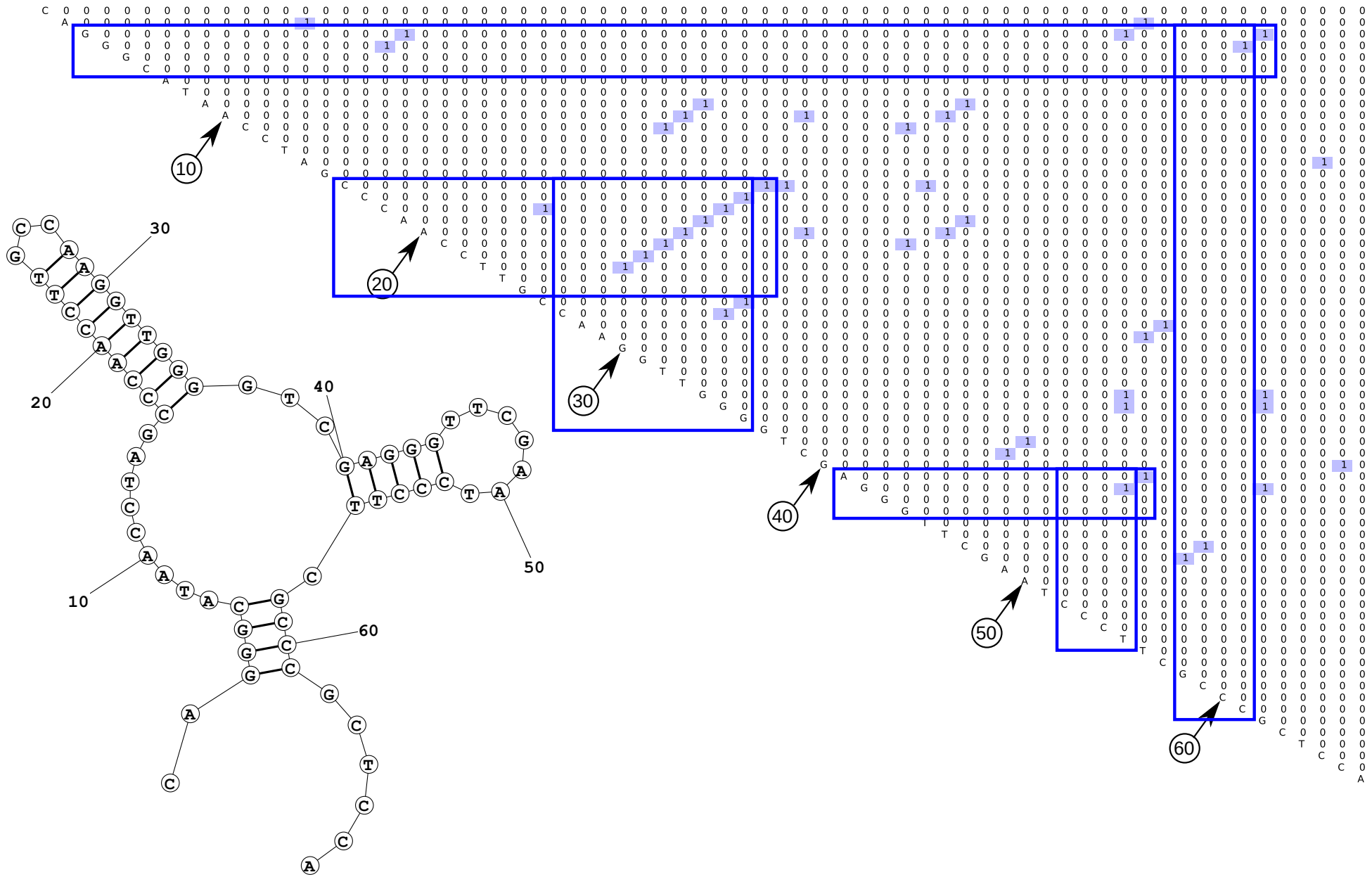
Example 3: real tRNA



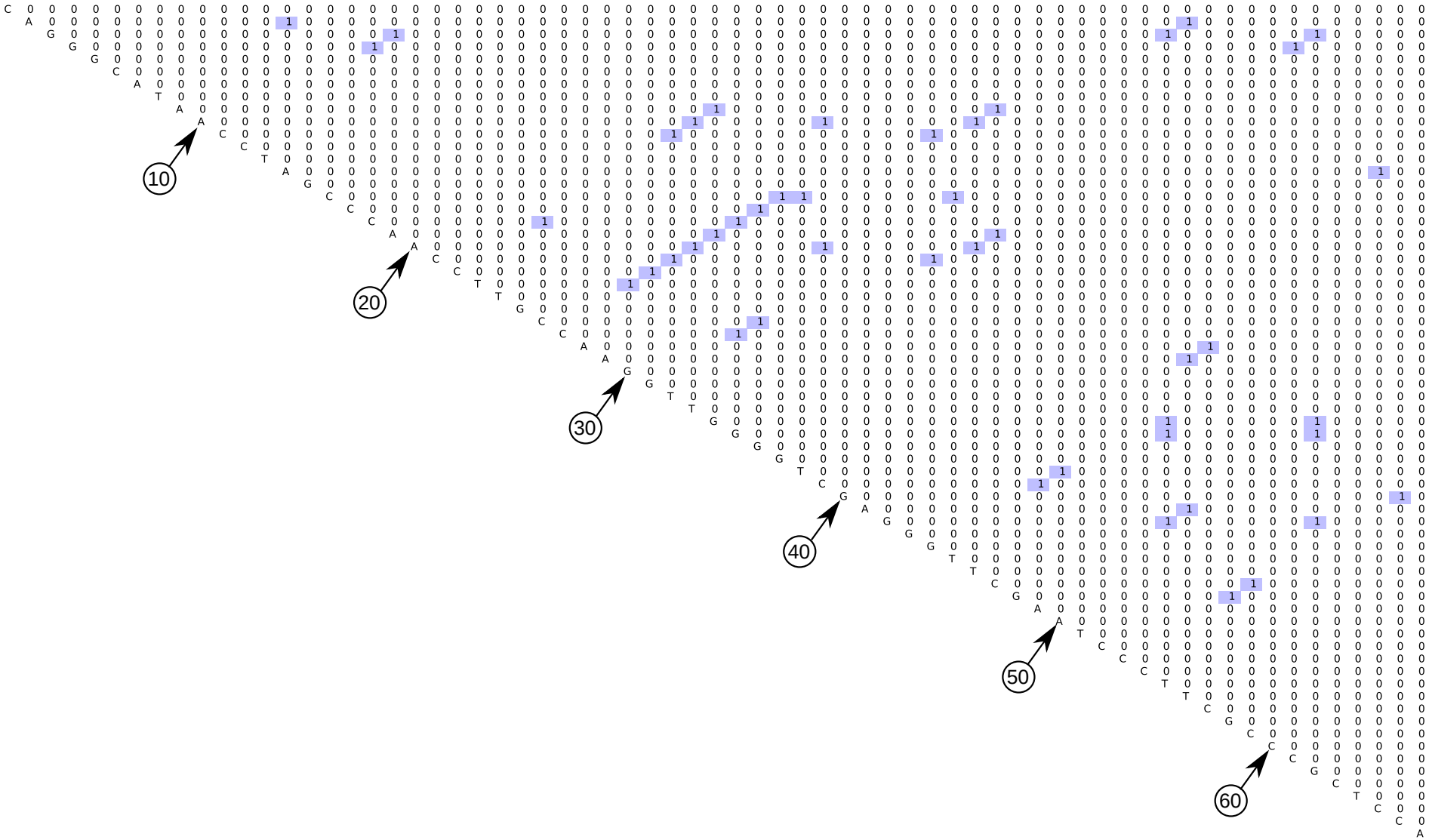
Example 3: real tRNA



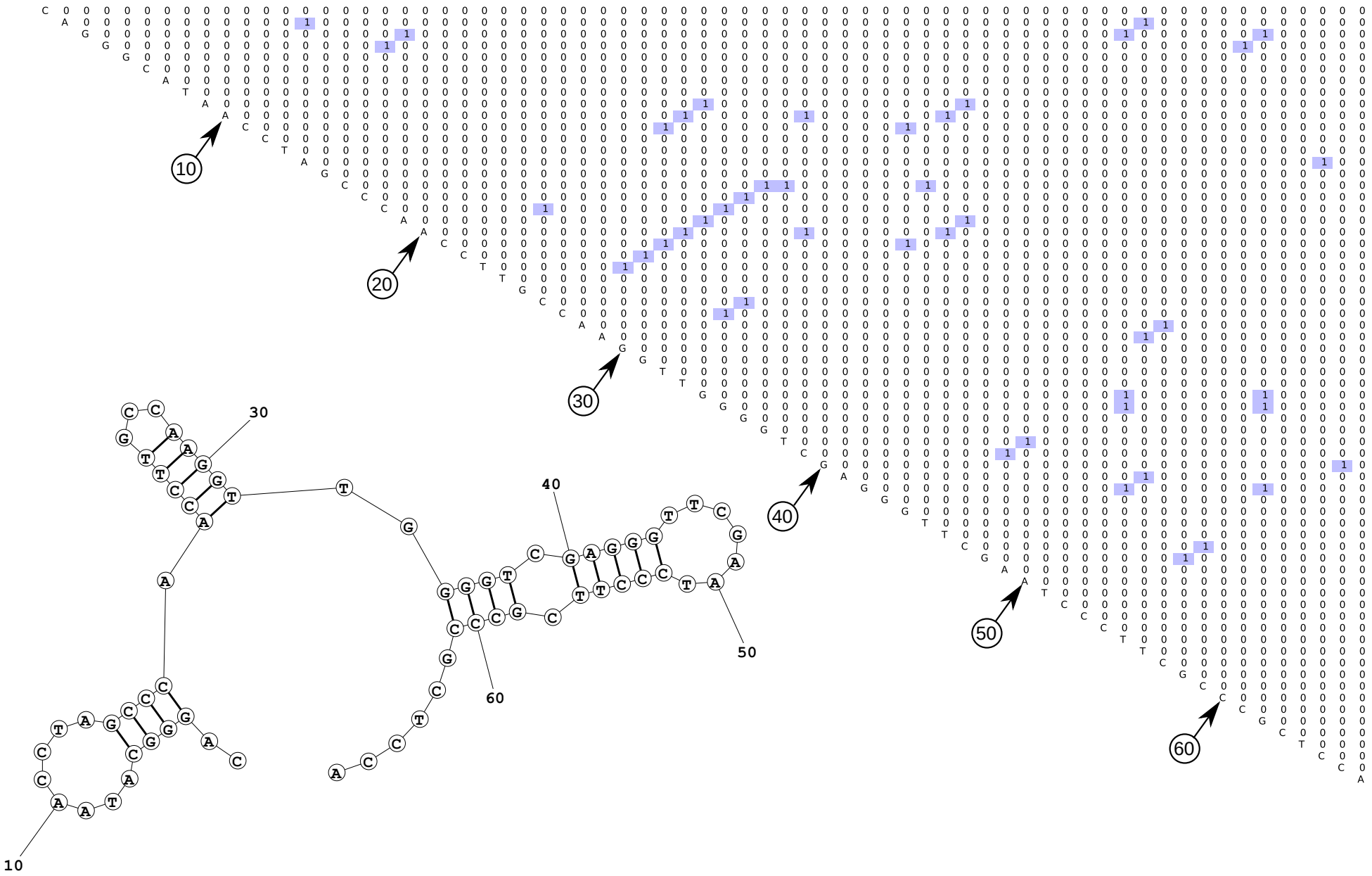
Example 3: real tRNA



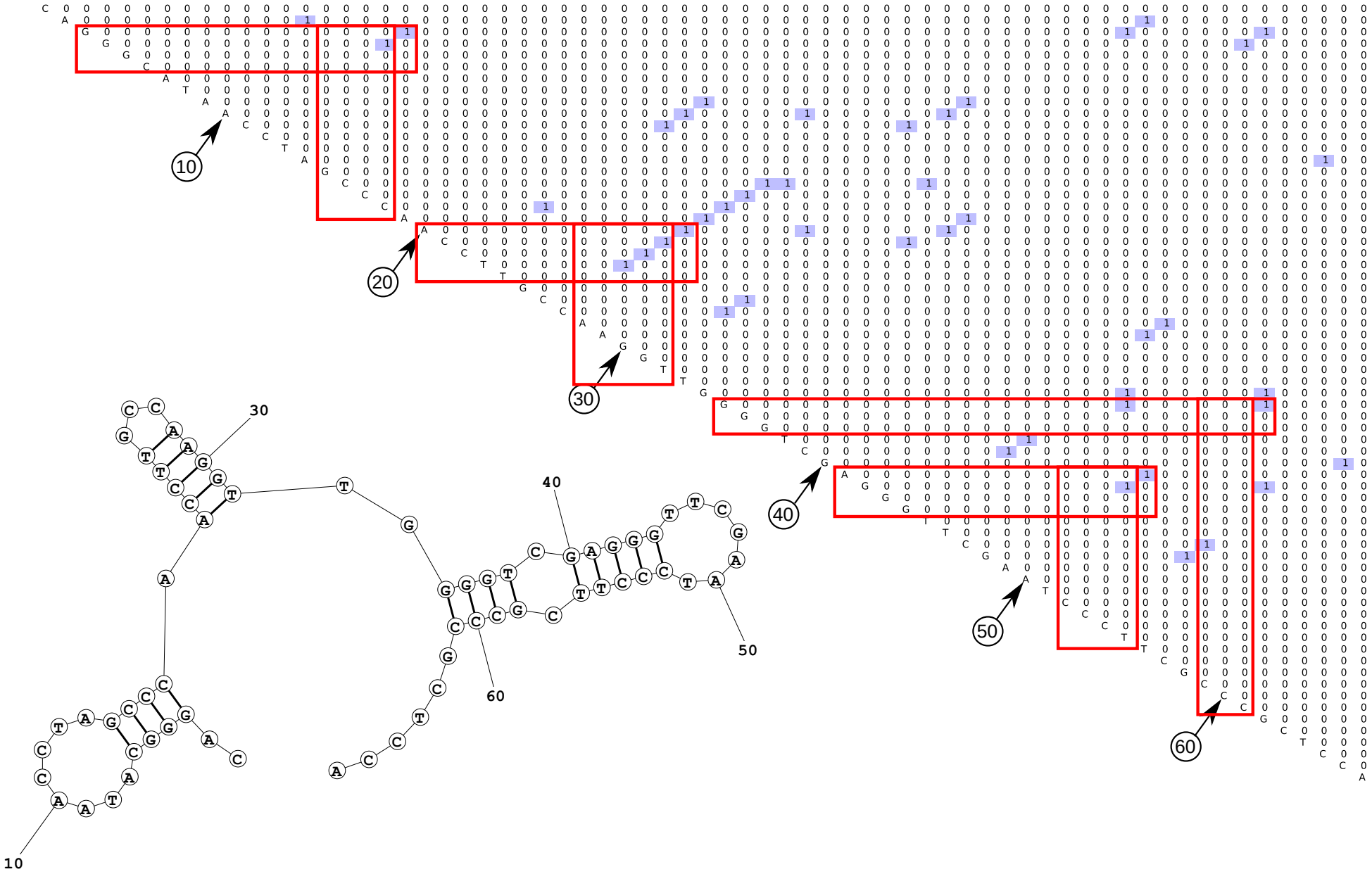
Example 3: real tRNA



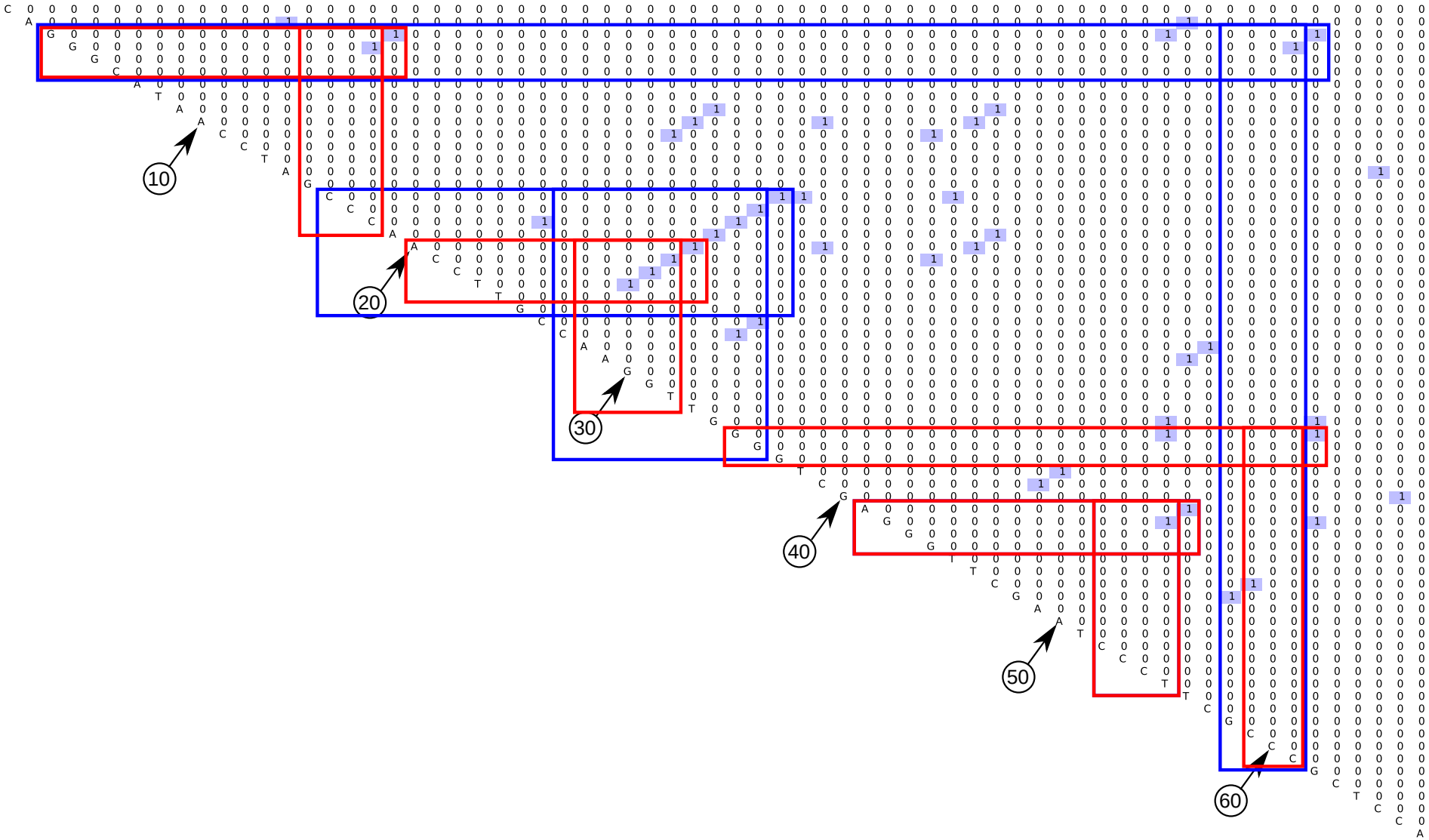
Example 3: real tRNA



Example 3: real tRNA



Example 3: real tRNA



Summary

- Parser is a features extructor
- All possible foldings
- Grammar is a varaible parameter
- It is possible to detect features which is not expressable in language class which in use

Artificial Neural Networks

We use dence neural network

- Problem: fixed size of input. Special simbol
- agressive dropout and batch normalization
- !!!

Evaluation

- 16s rRNA detection
  - Type provider is a **function which constructs type**

Evaluation

- 16s rRNA detection
  - Type provider is a **function which constructs type**
- tRNA classification
  - Type provider is a **function which constructs type**

Future work

- DNN without parsing
- Other types of neural networks
- More !!! Evaluation
- Comprison with other tools

Contact Information

- Semyon Grigorev:
  - s.v.grigoriev@spbu.ru
  - Semen.Grigorev@jetbrains.com
- Polind Lunina:
  - lunina_polina@mail.ru

- Trained models:

Contact Information

- Semyon Grigorev:
  - s.v.grigoriev@spbu.ru
  - Semen.Grigorev@jetbrains.com
- Polind Lunina:
  - lunina_polina@mail.ru

- Trained models:

Thanks!