

Разработка веб-приложения для анализа последовательностей РНК



АВТОРЫ: МАЛЛАБАЕВ АЗАМАТ, ЛУНИНА ПОЛИНА,
СУСАНИНА ЮЛИЯ, ВАСЕНИНА АННА

243 ГРУППА

РУКОВОДИТЕЛЬ: к. ф.-м. н. ГРИГОРЬЕВ С.В.

МАТЕМАТИКО-МЕХАНИЧЕСКИЙ ФАКУЛЬТЕТ
САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

26.12.2016

Цели и задачи приложения



Цель работы:

поиск подпоследовательностей РНК, являющихся генами, в метагеномных сборках

Задачи:

- Поиск всех подцепочек РНК указанной длины и соответствующих некоторой (выбранной по умолчанию или заданной пользователем) грамматике
- Выделение полученных подпоследовательностей на входном графе (для небольших входных данных)

Задачи участников команды



- Создание документации к веб-приложению
- Выполнение задач приложения на сервере и интеграция с пользовательским интерфейсом
- Реализация пользовательского интерфейса
- Визуализация графа и выделение ребер

Сценарии использования

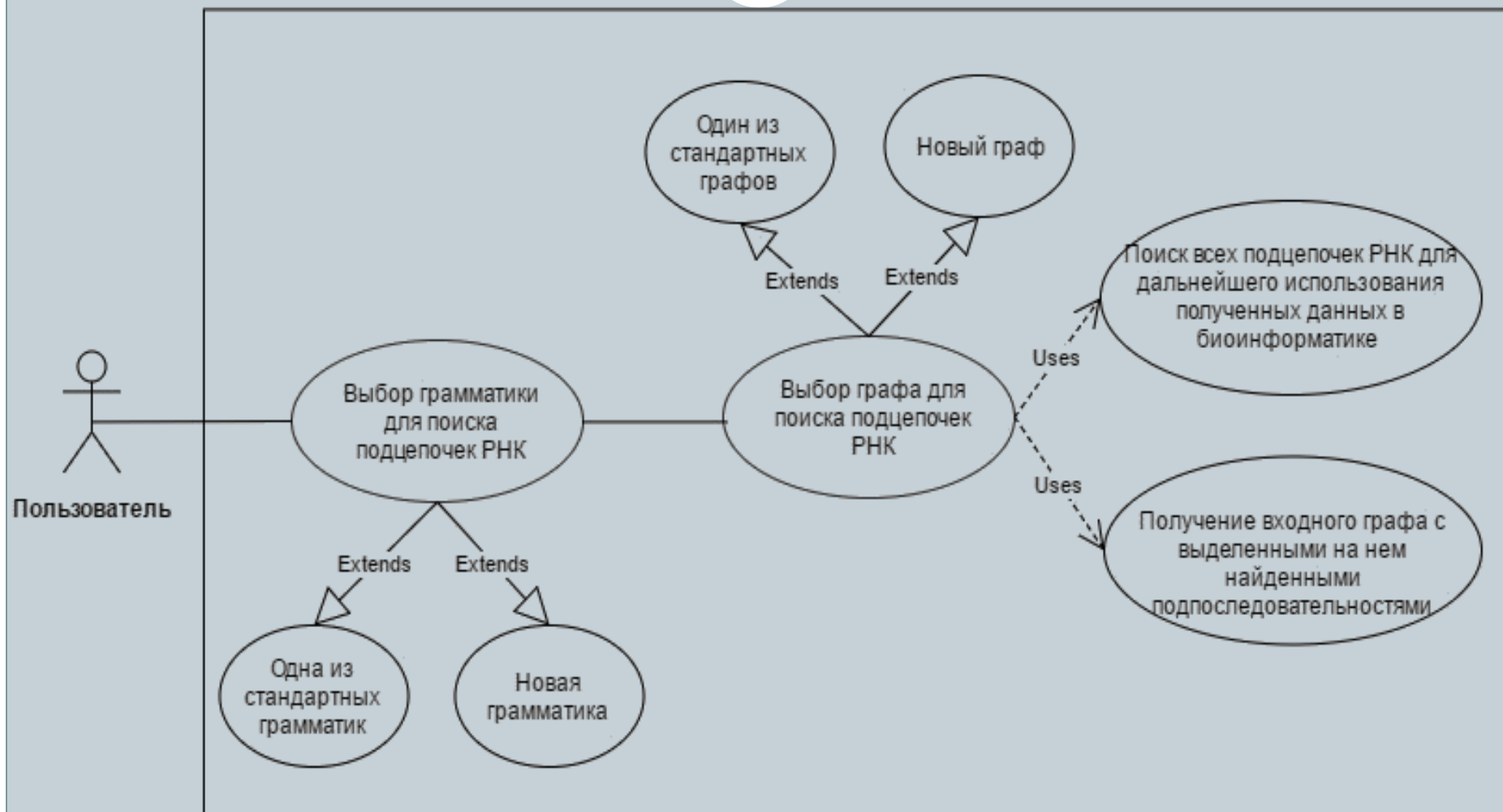
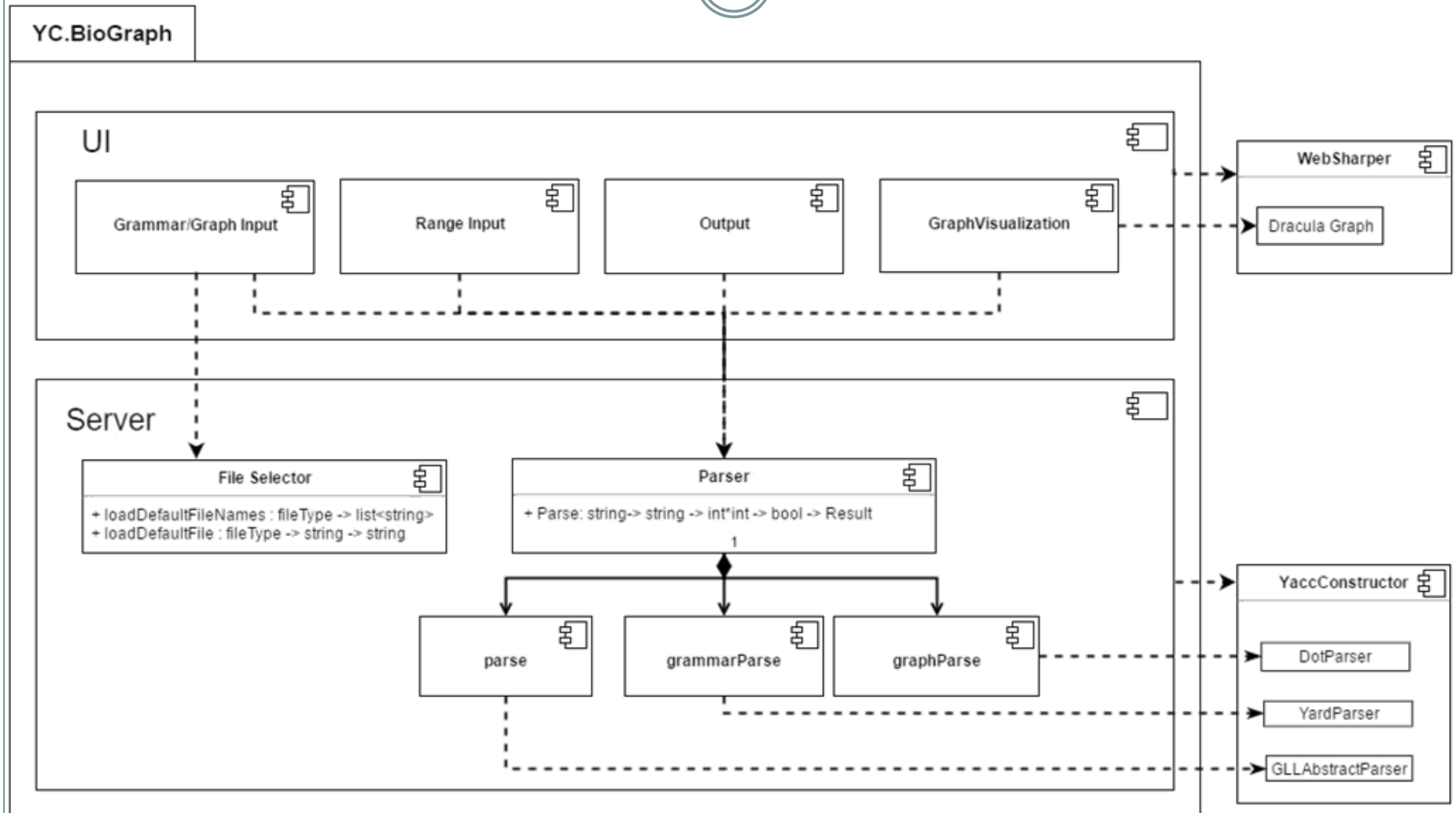


Диаграмма модулей



Сервер



File selector



LoadDefaultFileNames: FileType -> list<string>
LoadDefaultFiles: FileType -> string -> string

Parser



Parse: grammar:string -> graph:string -> range:(int * int) -> isDrawGraph:bool -> Result

File Selector

Grammar

Graph

Result

Error

message: string

Success

graph: option<Graph>

seqs: list<string>

Парсер



Parser



```
graphParse: string -> ParserInputGraph
grmParse: string -> GLLParserSource
parse: GLLParserSource -> ParserInputGrammar -> YCResult

tree2extGraph: ASTGLL.Tree -> ExtensionGraph

extGraph2iLazyTree: ExtensionGraph -> LazyTree
lazyTree2seqs: LazyTree -> list<string>
lazyTree2guardedSeqs: range:(int*int) -> LazyTree -> list<string>

extGraph2edges: ExtensionGraph -> list<int * int>
markGraph: ParserInputGraph -> markers:list<int * int> -> Graph
```

ExtensionGraph



```
start: ExtensionEdge
terminals: list<int64<extension>>
ands: Map<ExtensionEdge, ExtensionEdge * ExtensionEdge>
ors: Map<ExtensionEdge, list<ExtensionEdge>>
```

«YC» DotParser



«YC» YardParser



«YC» GLLAbstractParser



LazyRegTree

Or

seq<LazyRegTree>

And

Nuc * (unit -> LazyRegTree)

Empty

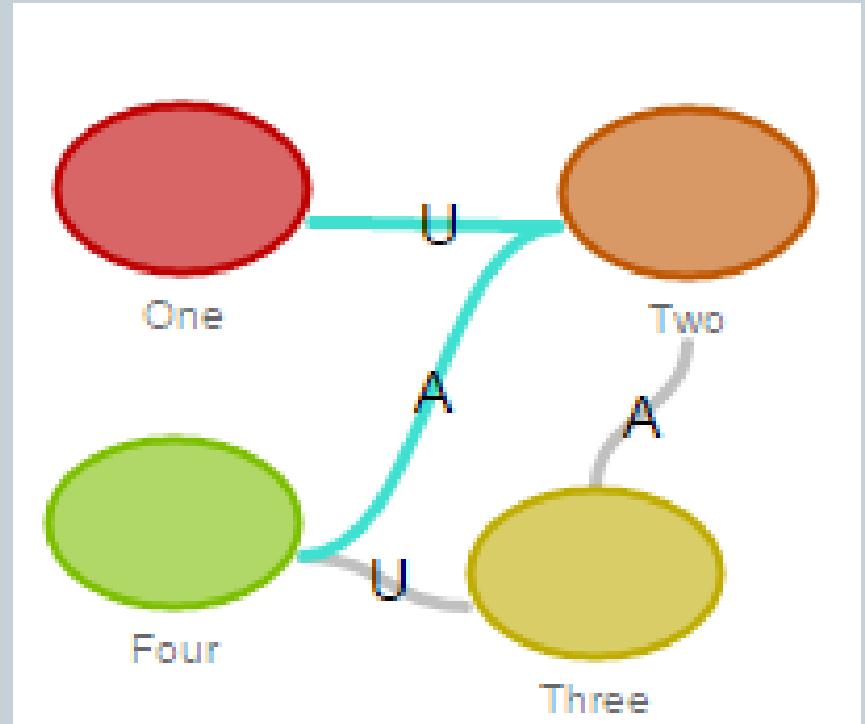
Дизайн интерфейса

YC.BioGraph		About								
<div data-bbox="175 347 355 386"><u>Grammar</u></div> <div data-bbox="175 401 850 598"></div> <div data-bbox="175 598 761 686"><table border="1"><tr><td>Select file</td><td></td></tr><tr><td>Choose default</td><td><input type="text"/> ▾</td></tr></table></div> <div data-bbox="162 706 289 753"><u>Graph</u></div> <div data-bbox="162 762 844 959"></div> <div data-bbox="162 959 753 1048"><table border="1"><tr><td>Select file</td><td></td></tr><tr><td>Choose default</td><td><input type="text"/> ▾</td></tr></table></div> <div data-bbox="170 1059 299 1106"><u>Range</u></div> <div data-bbox="181 1106 738 1155">from <input type="text"/> to <input type="text"/></div> <div data-bbox="168 1153 421 1192"><input type="checkbox"/> DRAW GRAPH</div> <div data-bbox="214 1230 305 1276">GO</div> <div data-bbox="923 364 1773 686"><div data-bbox="1217 508 1476 545">Graph Visualization</div></div> <div data-bbox="923 726 1083 776"><u>Output</u></div> <div data-bbox="923 791 1765 1192"></div> <div data-bbox="925 1210 1049 1249"><input type="checkbox"/> wrap</div>			Select file		Choose default	<input type="text"/> ▾	Select file		Choose default	<input type="text"/> ▾
Select file										
Choose default	<input type="text"/> ▾									
Select file										
Choose default	<input type="text"/> ▾									

Визуализация графа

Технология: DraculaGraph

- создание функции визуализации графа
- создание функции раскрашивания необходимых ребер графа
- предоставление пользователю возможности изменения отображения графа



Справка



YC.BioGraph

[github page](#)

Tutorial

Getting started

To start working with YC.BioGraph web application you need to fill in grammar, graph and range input forms.

You can use one of default grammars/graphs

Choose default

lite

choose file from your device

Выберите файл

Файл не выбран

or write them in the input textbox as you can see here.

```
[<Start>]
s: a b | b c | d
a: A
b: C
c: G
d: U
```

```
digraph {
  0 -> 1 [label = U]
  1 -> 2 [label = C]
}
```



YC.BIOGRAPH

[Home page](#)

[Source Code on GitHub](#)

[License](#)

[Release Notes](#)

GETTING STARTED

[Sample tutorial](#)

DOCUMENTATION

[API Reference](#)

Пример работы веб-приложения



YC.BioGraph

About

Grammar

```
[<Start>]
s: a b | b c | d
a: A
b: C
```

Выберите файл Файл не выбран

Choose default lite

Graph

```
digraph {
  0 -> 1 [label = U]
  1 -> 2 [label = C]
}
```

Выберите файл Файл не выбран

Choose default lite

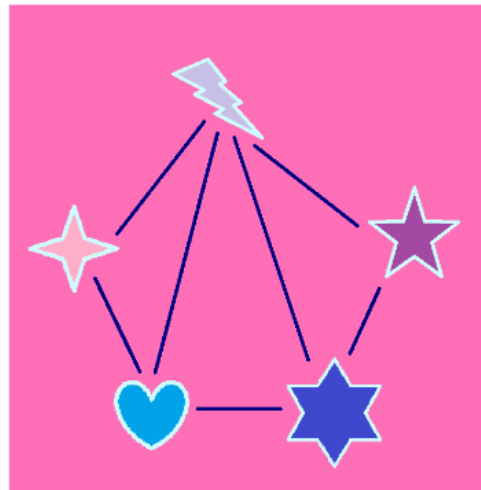
String range

from 0 to 1

☐ DRAW GRAPH

GO

Graph visualisation



Output

U

☐ wrap

Результаты



- Реализация пользовательского интерфейса
- Выполнение задач приложения на сервере и интеграция с пользовательским интерфейсом
- Визуализация графа и выделение ребер
- Создание документации к веб-приложению