

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Научный руководитель,  
Доцент департамента больших данных и  
информационного поиска, заведующая  
лабораторией «Научно-учебная лаборатория  
биоинформатики» факультета  
компьютерных наук

\_\_\_\_\_ М. С. Попцова  
«\_\_» \_\_\_\_\_ 2020 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия», канд.  
техн. наук, профессор департамента  
программной инженерии факультета  
компьютерных наук

\_\_\_\_\_ В. В. Шилов  
«\_\_» \_\_\_\_\_ 2020г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

**ПРОГРАММА ДЛЯ ИССЛЕДОВАНИЯ ПРИМЕНИМОСТИ МЕТОДОВ  
КЛАСТЕРИЗАЦИИ И СНИЖЕНИЯ РАЗМЕРНОСТИ ДЛЯ АВТОМАТИЧЕСКОГО  
СРАВНЕНИЯ ЭКСПЕРИМЕНТОВ ОДНОКЛЕТОЧНОГО СЕКВЕНИРОВАНИЯ  
КЛИЕНТ**

Текст программы

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.13-01 12 01-1-ЛУ**

Исполнитель  
Студент группы БПИ 182  
\_\_\_\_\_/И. С. Егоров. /  
«\_\_» \_\_\_\_\_ 2020 г.

**Москва 2020**

УТВЕРЖДЕН  
RU.17701729.04.13-01 12 01-1-ЛУ

**ПРОГРАММА ДЛЯ ИССЛЕДОВАНИЯ ПРИМЕНИМОСТИ МЕТОДОВ  
КЛАСТЕРИЗАЦИИ И СНИЖЕНИЯ РАЗМЕРНОСТИ ДЛЯ АВТОМАТИЧЕСКОГО  
СРАВНЕНИЯ ЭКСПЕРИМЕНТОВ ОДНОКЛЕТОЧНОГО СЕКВЕНИРОВАНИЯ  
КЛИЕНТ**

**Текст программы**

**RU.17701729.04.13-01 12 01-1**

**Листов 9**

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

**Москва 2020**

## Содержание

<b>1. Текст программы .....</b>	<b>4</b>
<b>1.1 logic.js .....</b>	<b>4</b>
<b>1.2 index.html .....</b>	<b>7</b>
<b>1.3 main.css .....</b>	<b>8</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 1. Текст программы

Текст программы доступен к просмотру по ссылке: <https://github.com/Igor-SeVeR/ReductionMethodsForComparingUnicellularSequencingExperiments-Coursework2/tree/master/djagnoBioInformaticsProject>

### 1.1 logic.js

```
'use strict';

/**
 * Ajax с телом FormData
 * @param {string} route - адресс
 * @param {string} method - метод запроса
 * @param {FormData} formData - данные
 * @param {function} callback - функция, которая будет вызвана после запроса
 */
async function ajaxForm(route, method, formData, callback) {

  const reqBody = {
    method: method,
    mode: 'cors',
    credentials: 'include',
  };

  if (method !== 'GET' && method !== 'HEAD') {
    reqBody['body'] = formData;
  }

  const req = new Request(route, reqBody);
  let responseJson = null;
  try {
    const response = await fetch(req);
    if (response.ok) {
      responseJson = await response.json();
      callback(responseJson);
    } else {
      throw new Error('Response not ok');
    }
  } catch (exception) {
    let test = document.querySelectorAll('.textProgress');
    for( let i = 0; i < test.length; i++ )
      { test[i].outerHTML = ""; }
    let textField = creteTextField("Input data is in incorrect format!.");
  }
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

let groupAdd = document.getElementsByClassName("choseFile")[0]
groupAdd.appendChild(textField);
}
}

function createImageField(textAddition,imageSrc) {
  let itemDiv=document.createElement("div")
  itemDiv.className="item"

  itemDiv.innerHTML=`<figure>
    
    <figcaption> <b class="textAddition"> ${textAddition} </b> </figcaption>
  </figure>`
  return itemDiv
}

function appendImages(images) {
  let test = document.querySelectorAll('.containerImage');
  for( let i = 0; i < test.length; i++ )
    { test[i].innerHTML = ""; }
  let containerImage=document.createElement("div")
  containerImage.className="containerImage"
  for (var textAddition in images) {
    let itemDiv=createImageField(textAddition, images[textAddition])
    containerImage.appendChild(itemDiv)
  }
  app.appendChild(containerImage)
}

function creteTextField(text) {
  let itemDiv=document.createElement("div");
  itemDiv.className = "textProgress"
  itemDiv.textContent = text;
  return itemDiv;
}

function createLinkToDownload(url, text) {
  var link = document.createElement('a');
  link.setAttribute('href',url);
  link.download = url;
  link.text = text;
  return link
}

function appendText(images) {
  let firstUrl = images['Predicted cells by clusters:']

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

let secondUrl = images['Interactive pyLDAvis graph:']
console.log(firstUrl, secondUrl)
delete images['Predicted cells by clusters:']
delete images['Interactive pyLDAvis graph:']
let elem = createLinkToDownload(firstUrl, 'Predicted cells by clusters');
let elem2 = createLinkToDownload(secondUrl, 'Interactive pyLDAvis graph');
let groupAdd = document.getElementsByClassName("choseFile")[0]
elem.className = "infoForDownload";
elem2.className = "infoForDownload";
groupAdd.appendChild(elem);
groupAdd.appendChild(elem2);
}

function deleteAllInfo() {
  let elems = document.querySelectorAll('.textProgress');
  for( let i = 0; i < elems.length; i++ )
    { elems[i].outerHTML = ""; }
  elems = document.querySelectorAll('.containerImage');
  for( let i = 0; i < elems.length; i++ )
    { elems[i].outerHTML = ""; }
  elems = document.querySelectorAll('.infoForDownload');
  for( let i = 0; i < elems.length; i++ )
    { elems[i].outerHTML = ""; }
}

function usingExistingModel(e) {
  e.preventDefault();
  deleteAllInfo();
  let textField = creteTextField("Build in progress.");
  let groupAdd = document.getElementsByClassName("choseFile")[0]
  groupAdd.appendChild(textField);
  var formData = new FormData()
  let file = document.getElementById("inp").files.item(0)
  formData.append('photo', file)
  let images = []
  ajaxForm("http://localhost:8000/api/get_mtx_data", "POST", formData, (r) => {
    images = r
    console.log(images)
    appendText(images)
    appendImages(images)
    let elems = document.querySelectorAll('.textProgress');
    for( let i = 0; i < elems.length; i++ )
      { elems[i].outerHTML = ""; }
    let textField = creteTextField("Build finished successfully.");
    let groupAdd = document.getElementsByClassName("choseFile")[0]
    groupAdd.appendChild(textField);
  })
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

    console.log(file)
  }

function usingJustBuildModel(e) {
  e.preventDefault();
  deleteAllInfo();
  let textField = creteTextField("Build in progress.");
  let groupAdd = document.getElementsByClassName("choseFile")[0]
  groupAdd.appendChild(textField);
  var formData = new FormData()
  let file = document.getElementById("inp").files.item(0)
  formData.append('photo', file)
  let images = []
  ajaxForm("http://localhost:8000/api/build_model_and_get_mtx_data", "POST", formData, (r)
=> {
    images = r
    console.log(images)
    appendText(images)
    appendImages(images)
    let elems = document.querySelectorAll('.textProgress');
    for( let i = 0; i < elems.length; i++ )
      { elems[i].innerHTML = ""; }
    let textField = creteTextField("Build finished successfully.");
    let groupAdd = document.getElementsByClassName("choseFile")[0]
    groupAdd.appendChild(textField);
  })

  console.log(file)
}

let app = document.getElementById("app")
let form = document.getElementById("form")
let usingModelEvent = document.getElementById("usingBuildModelBtn")
usingModelEvent.onclick = usingExistingModel
let buildNewModelBtn = document.getElementById("buildNewModelBtn")
buildNewModelBtn.onclick = usingJustBuildModel

```

## 1.2 index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

<title>BioInformatics</title>

</head>
<body>
  <link rel="stylesheet" href="main.css">

  <div class="choseFile">
    <form enctype="multipart/form-data" id="form">
      <input type="file" id="inp">
      <input type="submit" id="buildNewModelBtn" value="Train your own model.">
      <input type="submit" id="usingBuildModelBtn" value="Using learned model.">
    </form>
    <select name="select1" required>
      <option> Blood model </option>
    >
  </select>
  <p hidden="hidden">Process completed</p>
</div>

<div id="app"></div>

<script src="igor.js"></script>
</body>
</html>

```

### 1.3 main.css

```

.containerImage {
  display: flex;
  flex-wrap: wrap;
  flex-direction: column;
  justify-content: space-between;
  align-items: center;
  align-content: space-around
}

.choseFile{
  display: flex;
  flex-wrap: wrap;
  flex-direction: column;
  justify-content: space-between;
  align-content: center;
  margin-bottom: 100px;
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



```
.item {
  border-style: dashed;
  flex: 0 0 auto;
  margin: 40px;
}
```

```
.readyImages{

  width: 500px;
  height: 500px;
  clear: both;
}
```

```
.textAddition{
  font-size: xx-large;
  clear: both;
}
```

```
.textProgress {

}
```

```
.infoForDownload {

}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.507140-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата