ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В данном разделе размещается введение.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Биоинформатика

Биоинформатика — наука, объединяющая в себе методы прикладной математики, статистики, информатики для создания новых методов и алгоритмов для анализа разного рода биологических данных.

Биоинформатика занимается биохимией, биофизикой, экологией и многими другими областями биологии. Однако в данной работе фокус направлен на геномную биоинформатику и на конкретную ее задачу — анализ экспрессии генов.

1.1.1. Анализ экспрессии генов

Экспрессия генов — процесс преобразования наследственной информации от гена (в виде последовательности нуклеотидов ДНК) в функциональный продукт (РНК или белок).

Анализ экспрессии генов позволяет выяснить как ведет себя отдельный ген в разных условиях, тканях или организмах. Так, например, можно исследовать экспрессию вирусных белков или экспрессию онкогенов.

1.1.2. Используемые методы

Как было сказано ранее, биоинформатика использует в себе математику, информатику и статистику. Соответственно, задача анализа экспрессии генов сводится к исследованию путем статистических методов и алгоритмов числовой двумерной матрицы, как, например, небольшой срез матрицы в таблице 1.

Таблица 1 – Срез матрицы GSE14308. Строки матрицы соответствуют генам, столбцы — образцам.

	GSM357839	GSM357841	GSM357842	GSM357843	GSM357844
Rps29	16.32	16.30	16.25	16.32	16.30
Rpl13a	16.27	16.23	16.32	16.30	16.27
Rps3a1	16.23	16.19	16.30	16.25	16.25
Rpl38	16.21	16.25	16.27	16.27	16.21
Tmsb4x	16.30	16.32	16.23	16.21	16.32

На рисунке можно увидеть визуализацию матрицы экспрессии в виде тепловой карты.

Также к основным методам анализа относятся:

— Иерархическая и вероятностная кластеризации;
— Дифференциальная экспрессия;
— Метод главных компонент и визуализация его результатов.

1.2. Существующие решения для анализа экспрессии генов

1.2.1. R/Bioconductor

R - язык программирования для статистического анализа данных и работы с графикой [1].

Bioconductor - библиотека, содержащая в себе множество реализаций биоинформатических алгоритмов и методов обработки биологических данных на R. Она постоянно обновляется, пополняется новыми библиотеками, модерируется сообществом [2]. R и Bioconductor очень популярны в биоинформатической среде ввиду предоставляемых возможностей.

Однако для качественного и полноценного анализа с помощью этих инструментов, нужно иметь навыки программирования на R, что весьма неудобно для исследователей биологических специальностей.

1.2.2. GENE-E

GENE-E - Платформа для анализа данных и визуального исследования данных, созданная на Java и R [3]. Содержит в себе множество полезных для исследования инструментов: тепловые карты, кластеризацию, фильтрацию, построение графиков и т.д. Позволяет исследовать любые данные в виде матрицы. К тому же, содержит дополнительные инструменты для геномных данных.

Недостатки:

— Чтобы использовать, необходимо устанавливать на свой компьютер;
— Поддержка данного приложения прекратилась в связи с созданием morpheus.js [4];
— Не имеет открытого исходного кода, а только API для взаимодействия и создания новых приложений на его основе.

1.2.3. morpheus.js

Morpheus.js - веб-приложение для визуализации и анализа матриц от создателя GENE-E [4]. Создано уже на JavaScript и с открытым исходным кодом. Удобно для использования исследователями без навыков программирования и так же, как и GENE-E, применимо к любым матрицам.

Недостатки:

— Ограниченный набор функций, которых недостаточно для полноценного анализа;
— Для расширения биоинформатическими алгоритмами, не прибегая к дополнительным инструментам, требуется реализовывать их заново на JavaScript.

1.3. Инструменты, которые могут быть применены

1.3.1. Язык R и библиотека Bioconductor

Как было сказано ранее, Bioconductor полон актуальными и широко используемыми биоинформатическими алгоритмами, в том числе и для анализа экспрессии генов. Соответственно, реализовывать их заново обычно нет необходимости и можно использовать их для достижения целей этой работы.

1.3.2. JavaScript и Node.js

JavaScript — мультипарадигменный скриптов язык программирования, широко используемый для создания веб-приложений.

Node.js — [5]

1.3.3. R shiny

1.3.4. OpenCPU

1.3.4.1. Необходимые определения

HTTP HTTP API RPC Веб-сервер

OpenCPU — система для встроенных научных вычислений и воспроизводимых исследований, предоставляющая HTTP API для взаимодействия с R-серверами [6].

Имеется также библиотека *opencpu.js* для интеграции JavaScript и R.

1.3.5. Gene Expression Omnibus

Gene Expression Omnibus (GEO) — международный публичный репозиторий, агрегирующий и распространяющий различные формы геномных данных от исследовательского сообщества [7].

В библиотеке Bioconductor есть R-пакет *GEOquery* для удобной загрузки данных из GEO [8].

1.3.6. Docker

Docker — программное обеспечение для автоматизации запуска и внедрения приложений внутри контейнеров [9].

Для дальнейшего описания данного инструмента введем несколько определений.

Образ — отдельный исполняемый пакет, включающий себя все необходимое для запуска единицы программного обеспечения, в том числе исходный код, библиотеки, переменные окружения, конфигурационные файлы. Зачастую образ построен на основе другого образа с дополнительной конфигураций. Образ компилируется по *Dockerfile*, каждая команда в котором соответствует новому слою. При перекомпиляции обновляются только те слои, которые изменились.

Контейнер — запущенный экземпляр образа. Контейнер обычно исполняется изолированно от окружения, имея доступ к файлам или портам хост-системы только при наличии соответствующей конфигурации.

В отличие от виртуальных машин, которые запускают гостевую операционную систему в каждом экземпляре, контейнеры могут разделять общее ядро, и вся информация, которая должна быть в контейнере, это исполняемый процесс и его зависимости. Исполняемые процессы из контейнеров работают как нативные процессы, и могут управляться по отдельности.

Для контроля версий и хранения образов в открытом доступе используется Docker Hub [10]. В этом хранилище можно как добавлять репозитории, управляемые вручную, так и поддерживать автоматические сборки (*Automated Build*), которые привязаны к репозиториям на в популярных системах контроля версий: GitHub [11] и Bitbucket [12].

1.3.7. JSON

1.3.8. Protocol Buffers

Protocol Buffers (Protobuf) — гибкий, универсальный и автоматизированный механизм для сериализации структурированных данных [13].

Структура информации задается с помощью `*.proto` файлов в форме сообщений (Message).

1.3.9. Apache

1.3.10. HTML

1.4. Постановка задачи

Рассмотрев существующие решения для анализа экспрессии генов и инструментов, которые могли бы пригодиться для будущих решений, можно сформулировать цель и основные задачи данной работы

1.4.1. Цель работы

Создать веб-приложение, интегрирующее существующие возможности веб-приложения morpheus.js и методы анализа, реализованные в Bioconductor.

1.4.2. Основные задачи

а) Разработать способ взаимодействия между js-клиентом и R и встроить его в morpheus.js, чтобы избежать реализации с нуля уже существующих алгоритмов;
б) Реализовать графический интерфейс в js-клиенте и серверную реализацию в R-пакете;
в) Соединить все составляющие в одном веб-приложении phantasus;
г) Запустить веб-приложение в открытый доступ для исследователей.

1.4.3. Требования к веб-приложению phantasus

1.4.3.1. Доступность

Необходимо, чтобы веб-приложение phantasus было доступно для исследователей независимо от их местоположения и времени суток. Варианты действий:

а) Сделать его доступным по определенному веб-адресу, и тогда пользователь сможет продолжать исследования из любой точки, где есть подключение к интернету;

б) Предоставить возможность запускать приложение локально, например, с помощью Docker или внутри R.

1.4.3.2. Возможность дальнейшего расширения функционала

Как уже было сказано выше, библиотека Bioconductor постоянно обновляется и пополняется новыми алгоритмами, а исследователи находят новые методы для анализа экспресси генов, так что необходимо не только реализовать дополнительные методы, но и отладить и описать алгоритм действий для добабления новых.

Выводы по главе 1

ГЛАВА 2. АРХИТЕКТУРА ПРОЕКТА PHANTASUS

В этой главе будут подробно рассмотрены составляющие проекта:

— morpheus.js;
— R-пакет phantasus;
— OpenCPU.

Также будут описаны взаимосвязи между компонентами, сопутствующие инструменты и их предназначение в системе и ключевые для архитектуры выдержки из исходного кода.

2.1. morpheus.js

Как уже было рассказано в обзоре, morpheus.js — веб-приложение, полностью созданное на JavaScript, для визуализации и анализа матриц.

В этом разделе будут описаны основные классы и функции, реализованные в исходном коде morpheus.js, которые будут в дальнейшем необходимы для расширения функционала.

2.1.1. Чтение данных

В morpheus.js данные могут быть загружены из файла, полученного одним из следующих путей:

— Из компьютера;
— По URL-ссылке;
— Из Dropbox.

Допустимые форматы загружаемых файлов:

— txt-файл с tab-разделителями;
— Excel-таблица;
— MAF [14];
— GCT [15];
— GMT [16].

Для каждого формата файла в исходном коде morpheus.js присутствует соответствующий обработчик данных.

Также, morpheus.js предлагает набор предзагруженных данных из базы TCGA [17].

2.1.2. Класс Dataset

Одним из ключевых классов всего веб-приложения является класс Dataset. В каждом экземпляре этого класса хранится вся необходимая информация о данных, в которую входят:

- Числовая матрица, характеризующая уровень экспрессии всех генов во всех образцах;
- Количество строк и столбцов в матрице;
- Аннотация к образцам, например:
  - пол, возраст, контактную информацию испытуемых, если образцы были взяты с людей;
  - есть или нет инфекция в данном образце;
  - способ лечения;
  - контакты ответственного за взятие данного образца и пр.;
- Аннотация к генам, например:
  - Идентификатор гена в том или ином стандарте;
  - Числовые характеристики гена (средний уровень экспрессии по образцам, номер кластера) и пр.

Аннотация реализована в классе MetadataModel, который представляет собой не что иное, как набор именованных векторов с характеристиками. В каждом векторе хранятся:

- Название;
- Формат (строка, число);
- Массив значений.

Для векторов так же предусмотрены утилиты для визуализации. Так, например, есть возможность показать аннотацию в виде текста и/или цветом, что удобно для категориальных характеристик.

2.1.3. Класс SlicedDatasetView

Чаще всего во время работы программы экземпляры класса Dataset становятся обернуты в оболочку из SlicedDatasetView. Этот дополнительный класс дает возможность не пересоздавать каждый раз Dataset, а просто добавляет к данным информацию о том, какие индексы строк и столбцов выбраны и используются в данный момент.

2.1.4. Класс HeatMap

Данный класс предназначен для визуализации данных, обернутых в класс Dataset или SlicedDatasetView. Он дает возможность выбирать, какая аннотация будет представлена на экране, цветовой код, выбирать строки и столбцы, с которыми будут работать те или иные инструменты.

2.1.5. Реализованные методы

В morpheus.js имеются реализации следующих методов:

- Adjust — инструмент для корректировки данных:
  - $\log_2$;
  - $\log_2^{-1}$;
  - Квантиль-нормализация;
  - Z-тест;
  - Устойчивый Z-тест;
- Collapse — инструмент, позволяющий агрегировать строки или столбцы с одинаковыми значениями с помощью функции: min, max, $mean$, $median$, sum, максимум 25-го и 75-го перцентилей;
- Создать вычисленную аннотацию для строк или столбцов;
- Similarity Matrix;
- Transpose;
- t-SNE;
- Построение графиков.

Также присутствуют фильтрация и сортировка.

2.2. phantasus.js

В этом разделе будет рассмотрен модифицированный вариант morpheus.js и потребовавшиеся для его расширения компоненты.

2.2.1. Клиентская сторона OpenCPU — opencpu.js

В обзоре было рассказано об OpenCPU и его необходимости. В данной работе он нужен для связи JavaScript-клиента и R-сервера.

Opencpu.js реализует RPC-вызовы по принципу Asynchronous JavaScript and XML (Ajax [18]), позволяя тем самым пользоваться HTTP API: отправлять и получать HTTP-сообщения в фоновом режиме, тем самым не замедляя работу графического интерфейса и вычислений, осуществляемых на стороне клиента.

В данной библиотеке реализован класс Session, хранящий в себе ключ сессии, адреса на ссылки, файлы и переменные, содержащиеся внутри сессии.

Для подключения к R-пакету на R-сервере удобно использовать код, представленный на листинге 1. Для успешного подключения R-пакет должен быть предварительно установлен на host-машину, на которой располагается сервер.

```
1  ocpu.seturl("//hostname/ocpu/library/phantasus/R");
```

Листинг 1 – Подключение к R-пакету

После этого можно вызывать и запускать функции, содержащиеся в данном R-пакете, например, как в листинге 2.

```
1  var req = ocpu.rpc("function.name", arguments, callback(session) {
2    \\ Handling result
3  });
```

Листинг 2 – Пример вызова R-функции из JavaScript

2.2.2. Поддержка Protocol Buffers — protobuf.js

Чаще всего размеры обрабатываемых матриц 10000-40000 строк на 12-40 столбцов. Соответственно, пересылать их между клиентом и сервером в JSON-формате слишком долго.

Как было сказано в обзоре, Protocol Buffers позволяют лучше сериализовать данные, чтобы уменьшить размер пересылаемого пакета.

К сожалению, Google Developers официально поддерживают только Java, Python, C++, Go, Objective-C, Ruby, JavaNano и C#. Для JavaScript сообщество создает поддержку самостоятельно. После анализа существующих решений, было решено выбрать библиотеку ProtoBuf.js [19].

С помощью класса Builder, обрабатывающего файлы с протоколом структуры (`*.proto`), можно закодировать соответствующий JSON объект в Uint8Array, чтобы после пересылать его на сервер.

2.2.3. Интерактивные графики — plotly.js

Для отображения интерактивных графиков используется библиотека plotly.js [20], которая предоставляет удобное API, в котором опи-

сание графика строится в JSON-формате. Соответственно, вся графическая работа лежит на клиенте.

2.3. R-пакет phantasus

Весь реализованный функционал должен иметь клиентскую часть в виде графического интерфейса и серверную в виде R-функции. Прежде чем рассматривать созданные функции, будут представлены имеющиеся необходимые элементы.

2.3.1. Biobase и ExpressionSet

Необходимый минимум функций для работы с геномными данными содержится в R-пакете Biobase [21].

Класс ExpressionSet [22] так же содержится в Biobase. Он помогает представлять данные об экспрессии генов в удобном формате:

- assayData — описание матрицы:
  - features — количество генов;
  - samples — количество образцов;
  - exprs — числовая матрица экспрессии;
- phenoData — аннотация к образцам:
  - sampleNames — идентификаторы образцов;
  - varLabels — названия характеристик;
  - varMetadata — описание характеристик;
  - pData — матрица характеристик;
- featureData — аннотация к генам:
  - featureNames — идентификаторы генов;
  - fvarLabels — названия характеристик;
  - fvarMetadata — описание характеристик;
  - fData — матрица характеристик.

Для доступа к каждому из элементов есть одноименная функция, что позволяет удобно взаимодействовать с экземплярами класса. Также многие из функций обработки данных в Bioconductor и в Biobase в частности завязаны на использование ExpressionSet.

Все реализованные в R-пакете phantasus функции принимают на вход в качестве одного из аргументов экземпляр класса ExpressionSet.

2.3.2. Создание ExpressionSet из данных

В начале работы с phantasus необходимо загрузить данные. Если данные загружены из файла, то они будут сначала обработаны на клиенте, а после пересланы на сервер для создания ExpressionSet из них с помощью кода на листинге 3

```
1  createES <- function(data, pData, varLabels, fData, fvarLabels) {
2  exprs <- t(data)
3  phenoData <- AnnotatedDataFrame(data.frame(pData))
4  varLabels(phenoData) <- varLabels
5
6  featureData <- AnnotatedDataFrame(data.frame(fData))
7  varLabels(featureData) <- fvarLabels
8
9  es <- ExpressionSet(assayData = exprs, phenoData=phenoData, featureData =
       featureData)
10 assign("es", es, envir = parent.frame())
11 es
12 }
```

Листинг 3 – Функция создания ExpressionSet из исходных данных

По завершении функция отправляет `es` в глобальные переменные, чтобы ExpressionSet был доступен по адресу: /ocpu/tmp/session-key/R/es. Таким образом, получив ключ данной сессии, можно иметь доступ и к ExpressionSet, находящемуся в ней.

2.3.3. Загрузка данных из GEO — GEOquery

2.3.4. Дифференциальная экспрессия — limma

2.3.5. Статистические функции — stats

2.3.6. Поддержка Protocol Buffers — protolite

2.4. Связь через OpenCPU API

2.5. Неотсортировано

Выводы по главе 2

ГЛАВА 3. РЕАЛИЗАЦИЯ

Выводы по главе 3

ЗАКЛЮЧЕНИЕ

В данном разделе размещается заключение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Gentleman R.*, *Ihaka R.* R project. — URL: https://www.r-project.org/. [Электронный ресурс].

2 Bioconductor / A. Sonali [и др.]. — URL: https://www.bioconductor.org/. [Электронный ресурс].

3 *Gould J.* GENE-E. — URL: http://www.broadinstitute.org/cancer/software/GENE-E/. [Электронный ресурс].

4 *Gould J.* morpheus.js. — URL: https://clue.io/morpheus.js/. [Электронный ресурс].

5 *Foundation N.* Node.js. — URL: https://nodejs.org/. [Электронный ресурс].

6 *Jeroen O.* OpenCPU. — URL: https://www.opencpu.org/. [Электронный ресурс].

7 *Biotechnology Information N. C. for*. Gene Expression Omnibus. — URL: https://www.ncbi.nlm.nih.gov/geo/. [Электронный ресурс].

8 *Davis S.*

9 *Hykes S.* Docker. — URL: https://www.docker.com. [Электронный ресурс].

10 *Docker I.* Docker Hub. — URL: https://hub.docker.com/. [Электронный ресурс].

11 *GitHub*. GitHub. — URL: https://github.com/. [Электронный ресурс].

12 *Atlassian*. Bitbucket. — URL: https://bitbucket.org/. [Электронный ресурс].

13 *Developers G.* Protocol Buffers. — URL: https://developers.google.com/protocol-buffers/. [Электронный ресурс].

14 *Institute N. C.* Mutation Annotation Format. — URL: https://wiki.nci.nih.gov/display/TCGA/Mutation+Annotation+Format+%28MAF%29+Specification/. [Электронный ресурс].

15 GCT. — URL: http://software.broadinstitute.org/cancer/software/genepattern/file-formats-guide#GCT/. [Электронный ресурс].

16 Gene Matrix Transposed file format. — URL: http://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats#GMT:_Gene_Matrix_Transposed_file_format_.28.2A.gmt.29/. [Электронный ресурс].

17 *Insitute N. C.* The Cancer Genome Atlas. — URL: https://cancergenome.nih.gov/. [Электронный ресурс].

18 Asynchronous JavaScript and XML. — URL: http://api.jquery.com/jquery.ajax/. [Электронный ресурс].

19 *dcode*. Protocol Buffers for JavaScript (& TypeScript). — URL: https://github.com/dcodeIO/ProtoBuf.js/. [Электронный ресурс].

20 *Plotly*. plotly. — URL: https://plot.ly/company/team/. [Электронный ресурс].

21 Orchestrating high-throughput genomic analysis with Bioconductor / Huber [и др.] // Nature Methods. — 2015. — Т. 12, № 2. — С. 115–121. — URL: http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

22 *Falcon S.*, *Morgan M.*, *Gentleman R.* An Introduction to Bioconductor's ExpressionSet Class. — 2006. — URL: https://www.bioconductor.org/packages/devel/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf. [Электронный ресурс].