Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Факультет информационных технологий и программирования
Кафедра компьютерных технологий

Реализация эффективного взаимодействия между платформой для анализа экспрессии генов Morpheus и библиотекой вычислительных методов R/Bioconductor

Зенкова Д.М.
Научный руководитель: Сергушичев А. А.

Санкт-Петербург
2017

ОГЛАВЛЕНИЕ

Стр.

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, в таблице **??**.

Таблица 1.1 — 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 экспресии в виде тепловой карты.

Также к основным методам анализа относятся:

– Иехархическая и вероятностная кластеризация;
– Дифференциальная экспрессия;
– Метод главных компонент и визуализация его результатов.

1.2. Существующие решения для анализа экспрессии генов

1.2.1. GENE-E

GENE-E - Платформа для анализа данных и визуального исследования данных, созданная на Java и R [**?**]. 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 [**?**];
– Не имеет открытого исходного кода, а только API для взаимодействия и создания новых приложений на его основе.

1.2.2. morpheus.js

Morpheus.js - веб-приложение для визуализации и анализа матриц от создателя GENE-E. 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 требуется реализовывать их заново на JavaScript.

1.2.3. R/Bioconductor

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

Bioconductor - библиотека, содержащая в себе множество реализаций биоинформатических алгоритмов и методов обработки биологических данных на R. Она постоянно обновляется, пополняется новыми библиотеками, модерируется сообществом [**?**]. 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.

1.3. 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

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

Node.js — [**?**]

1.3.3. R shiny

1.3.4. OpenCPU

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 [**?**]. 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 [**?**].

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

1.3.6. Docker

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

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 [**?**]. 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 [**?**] и Bitbucket [**?**].

1.3.7. JSON

1.3.8. Protocol Buffers

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

Структура информации задается с помощью *.proto-* файлов в форме сообщений (Message). Пример такого файла можно увидеть на листинге 1.1

```
1
2  message Person {
3    required string name = 1;
4    required int32 id = 2;
5    optional string email = 3;
6
7    enum PhoneType {
8      MOBILE = 0;
9      HOME = 1;
10     WORK = 2;
11   }
12
13   message PhoneNumber {
14     required string number = 1;
15     optional PhoneType type = 2 [default = HOME];
16   }
17
18   repeated PhoneNumber phone = 4;
19 }
```

Листинг 1.1 — Сообщение, структурирующее информацию о человеке

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

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.

Возможность дальнейшего расширения функционала веб-приложения

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

В данной главе была кратко описана предметная область и необходимые биоинформатические определения, рассмотрены существующие решения и инструменты, которые могли бы быть применены для разработки новых решений. Исходя из обзора, была сформулирована цель работы и требования к результату:

а) доступность;

б) возможность дальнейшего расширения функционала.

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 [**?**];
– GCT [**?**];
– GMT [**?**].

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

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

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 исследования и обработки данных

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, maximum of 25 and 75 percentiles;
- Создать вычисленную аннотацию для строк или столбцов;
- Similarity Matrix;
- Transpose;
- t-SNE;
- Построение графиков.

Также присутствуют фильтрация и сортировка.

2.2. R-пакет Phantasus

2.3. Связь через OpenCPU API

2.4. Рассортировать по секциям

2.4.1. Реализация Dataset в morpheus.js

В исходном коде morpheus.js имеется класс для работы с данными, позволяющий рассматривать сечения, работать с аннотациями, в целом, осуществлять любое взаимодействие с имеющимися данными. Код представлен на листинге 2.1.

Листинг 2.1 — класс Dataset

2.4.2. Стандартный класс ExpressionSet

2.4.3. opencpu.js

2.4.4. Protocol Buffers

На стороне клиента

На стороне сервера

2.4.5. Схема взаимодействия клиент-сервер

2.4.6. Загрузка и разбор данных из GEO

2.4.7. Загрузка данных в phantasus

Из файла

а) My computer;
б) URL;
в) Dropbox.

Из GEO

2.4.8. Аннотации строк и столбцов матрицы

2.4.9. Git-репозиторий phantasus

2.4.10. Запуск phantasus локально и на сервере

Выводы к главе 2

ГЛАВА 3. **РЕАЛИЗАЦИЯ**

Выводы к главе 3

ЗАКЛЮЧЕНИЕ