УДК 573.22; 57.052; 57.087
DOI 10.25205/1818-7900-2019-17-4-5-74-86

Разработка программного комплекса WebMCOT для поиска совместно встречаемых ДНК-мотивов сайтов связывания транскрипционных факторов

А. М. Мухин, **В. Г. Левицкий**, **С. А. Лашин**

Новосибирский государственный университет
Новосибирск, Россия

Институт цитологии и генетики СО РАН
Новосибирск, Россия

Аннотация

Регуляция биологических процессов транскрипции генов осуществляется с помощью специфичных белков, а именно транскрипционных факторов. Транскрипционные факторы связываются с определенными участками геномной ДНК (сайтами связывания), которые называются мотивами. Совместное действие двух и более транскрипционных факторов является очень широко распространенным механизмом их действия. Поэтому термином «композиционный элемент» называют устойчивое сочетание двух близко расположенных и часто встречаемых мотивов. Композиционные элементы можно классифицировать по наличию перекрывания двух мотивов или их расположению со спейсером. В настоящее время развитие технологий массового секвенирования (ChIP-seq) сделало доступными геномные профили сайтов связывания для многих «якорных» транскрипционных факторов, поэтому задача поиска комбинаций «якорного» и других («партнерских») мотивов стала очень актуальной. Однако у существующих подходов к поиску пар таких мотивов есть ограничения: либо решение не способно предсказывать пересечение мотивов, либо оно может это делать, но тогда ему нужны дополнительные данные экспериментов ChIP-seq по партнерским мотивам, получение которых является дорогостоящим. В Институте цитологии и генетики СО РАН была реализована программа MCOT, сочетающая поиск мотивов с учетом их пересечения и анализ одного набора данных ChIP-seq. Для данной программы отсутствовал графический интерфейс для того, чтобы пользователи могли вводить файлы в удобной форме и получать результаты в графическом и табличном виде. В данной работе представлен разработанный программный комплекс WebMCOT, предназначенный для поиска совместно встречаемых ДНК-мотивов на основе результатов эксперимента ChIP-seq. Программный комплекс состоит из трех частей с использованием различных языков программирования. В статье описываются список используемых программных инструментов с аргументацией их выбора, архитектура программного комплекса и интерфейс веб-сайта.

Ключевые слова

веб-сервис, транскрипционные факторы, сайты связывания, мотивы

Благодарности

Работа была поддержана частично поддержана РФФИ № 18-29-13040 «Мета-анализ полногеномных данных на основе выявления и аннотации различных структурно-функциональных типов сайтов связывания транскрипционных факторов в составе композиционных элементов» и бюджетным проектом № 0259-2019-0008.

Для цитирования

Мухин А. М., *Левицкий В. Г.*, *Лашин С. А.* Разработка программного комплекса WebMCOT для поиска совместно встречаемых ДНК-мотивов сайтов связывания транскрипционных факторов // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 4. С. 74–86. DOI 10.25205/1818-7900-2019-17-4-74-86

© А. М. Мухин, В. Г. Левицкий, С. А. Лашин, 2019

Developing of WebMCOT Web-Service for Finding Cooperative Site-Binding TF DNA-Motifs

A. M. Mukhin, V. G. Levitsky, S. A. Lashin

Novosibirsk State University
Novosibirsk, Russian Federation

Institute of Cytology and Genetics SB RAS
Novosibirsk, Russian Federation

Annotation

Regulation of eukaryotic gene transcription is controlled by specific proteins transcription factors. Transcription factors bind certain regions of genomic DNA (binding sites or motives). Common action of two or more transcription factors is widespread mechanism of transcription factor action. Hence, the term 'composite element' implied two closely located and frequently occurred in genomic DNA motives. Composite elements are partitioned onto those with two overlapped motifs, or with these two motifs separated with a spacer. Currently, the chromatin immunoprecipitation high throughput approach ChIP-seq is used to locate binding sites for a certain "anchor" transcription factor in vivo in genomic scale. Thus, the search of composite elements with the help of ChIP-seq whole-genome transcription factor binding profiles is the actual bioinformatics issue. But existing approaches for prediction of composite elements on the basis of ChIP-seq data either omit an overlap of motifs (but require only a single ChIP-seq dataset) or consider an overlap of motifs (but require additional ChIP-seq data for a partner motif). But, ChIP-seq experiments are very expensive. In the Institute of Cytology and Genetics, MCOT program has been recently developed. It performs search of motifs taking into account their overlaps based on a single ChIP-seq dataset. MCOT is a console application and does not have many user friendly functions like data preparation and report generation. This work presents a web service WebMCOT for prediction of co-occurred DNA motifs in ChIP-seq data. WebMCOT consists of three parts: client, server, and worker. Software tools list, architecture and web interface are presented.

Keywords

Web-service, transcription factors, site binding, motifs

Acknowledgements

The work is partially supported by RFBR, project no. 18-29-13040 "Meta-analysis of genome-wide data based on discovery and annotation of various structural-functional transcription factor binding sites participating in composite elements" and by Budget Project no. 0259-2019-0008

For citation

Mukhin A. M., Levitsky V. G., Lashin S. A. Developing of WebMCOT Web-Service for Finding Cooperative Site-Binding TF DNA-Motifs. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 4, p. 74–86. (in Russ.) DOI 10.25205/1818-7900-2019-17-4-74-86

Введение

Несколько последних десятилетий в биологии ознаменовались открытием и внедрением множества новых экспериментальных методов. Массовое применение этих методов для решения прикладных и фундаментальных задач привело к накоплению гигантских объёмов экспериментальных данных (на 2019 год – петабайты информации [1]). Обработка столь больших объёмов данных невозможна в «ручном» режиме и требует привлечения методов компьютерного анализа и моделирования. В настоящее время разработкой и программной реализацией методов анализа биологических данных занимается такой раздел науки как биоинформатика. Одним из больших разделов биоинформатики является компьютерная геномика – анализ полногеномных данных с целью выявления в них различного рода закономерностей и функциональных сигналов.

ДНК является последовательностью нуклеотидов, записанной с помощью алфавита из четырех букв, а именно А, Г, Ц, Т (английские названия: A, G, C, T, соответственно). В геномной ДНК выделяют *гены* – фрагменты, по которым синтезируются функциональные продукты (белки или РНК). Все организмы живой природы состоят из белков, без них организм не сможет существовать. Белки, в свою очередь, разделяются по функциям: существуют сигнальные белки (гормоны), переносчики необходимых молекул (например, гемоглобин крови

переносит кислород в разные ткани организма), защитные белки (например, гликопротеины) и многие другие.

Важным этапом экспрессии (т. е. «работы») генов является *транскрипция* – синтез с помощью РНК-полимераз копии гена в виде матричной одноцепочечной молекулы – рибонуклеиновой кислоты (мРНК). Но для того, чтобы РНК-полимераза смогла инициировать транскрипцию гена, необходимо формирование специального комплекса инициации транскрипции гена, так как ДНК в ядре компактизована и представлена в виде хроматина в сложном комплексе с белками. Белки, которые способствуют регуляции транскрипции генов, называются *транскрипционными факторами* (*ТФ*), а их сайты посадки (длиной около 15–30 нуклеотидов) называются *сайтами связывания транскрипционных факторов* (*ССТФ*). Наиболее консервативная часть ССТФ представляется мотивом ДНК. *Мотив* ДНК в наиболее широкоиспользуемом виде – это вероятностный паттерн, который показывает, с какой вероятностью на определенной позиции может быть каждый из нуклеотидов (A, G, T и C) [2]. Механизм действия ТФ, как правило, кооперативный [3], поэтому часто рассматривается набор двух ССТФ в составе *композиционного элемента* (*КЭ*) [4]. КЭ – специфичная комбинация двух ССТФ, совместное действие которых оказывает эффект, существенно отличный от воздействия каждого из них в отдельности. Мотивы в составе КЭ могут перекрываться или быть разделены промежутком – *спейсером* – небольшой длины (до 30–50 нуклеотидов между двумя мотивами).

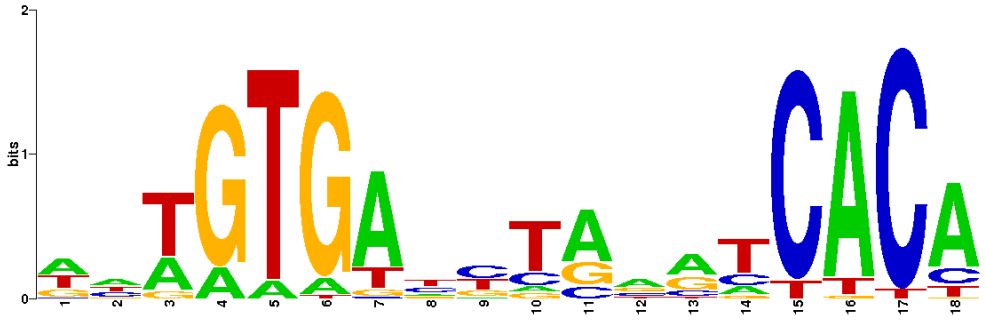

Рис. 1. Иллюстрация мотива [1]
Fig. 1. Motif illustration

Для того чтобы экспериментально найти мотивы ССТФ для конкретного ТФ, в полном геноме используется экспериментальный метод ChIP-seq. В основе этого метода – использование белков иммунного ответа (антител) на определенный ТФ с выделением нужного фрагмента ДНК вместе с белком. У этого метода есть два ограничения:

- стандартно данная экспериментальная методика позволяет работать только с одним белком; выявлять КЭ вместе с их мотивами, используя этот метод, невозможно;
- эксперимент ChIP-seq требует больших материальных затрат.

Участок геномной ДНК, выявленный в эксперименте ChIP-seq, называют *пиком*.

Вопросом для биологов в ходе исследования ТФ является взаимодействие двух и более ТФ между собой (исследование КЭ). Дело в том, что для запуска транскрипции одного ТФ недостаточно, поэтому стоит задача нахождения второго (*партнерского*) мотива по первому (*якорному*), с антителами для которого выполнялся ChIP-Seq-эксперимент.

Соответственно, перед биоинформатиками стоит задача предсказания мотивов других ТФ, которые совместно встречаются с якорным ТФ. С точки зрения молекулярных биологов, два

[1] Источник: http://genomeinformatics.uni-due.de/research/motif-discovery/.

ТФ связаны тогда, когда они находятся на ДНК на расстоянии друг от друга до 50–100 п.о., так как при таком расстоянии возможны физические взаимодействия между белками.

Существующие программы для предсказания мотивов партнерских ТФ работают недостаточно хорошо ввиду следующих ограничений:

- допускаются перекрытия между мотивами ТФ, но тогда нужны данные ChIP-Seq для потенциального партнерского ТФ [5; 6];
- запрещаются перекрытия мотивов в рамках одного пика, но тогда для анализа достаточны данные одного эксперимента ChIP-seq [7; 8].

Таблица 1

Существующие программные решения для предсказания КЭ

Table 1

Existing programs for composition elements prediction

Имя	Один эксперимент	Предсказание перекрывающихся мотивов	WWW	Ссылка
Gem	Нет	Да	http://groups.csail.mit.edu/cgs/gem	[5]
Taco	Нет	Да	http://bioputer.mimuw.edu.pl/taco/	[6]
iTFs	Да	Нет	http://veda.cs.uiuc.edu/iTFs/	[7]
SpaMo	Да	Нет	http://meme-suite.org/tools/spamo	[8]
MCOT	Да	Да	https://gitlab.sysbio.cytogen.ru/academiq/mcot-kernel	[9]

В Институте цитологии и генетики СО РАН в работе [9] был представлен алгоритм предсказания КЭ, лишенный этих недостатков. В основе лежит перестановочный тест, который позволяет сгенерировать случайное распределение любого из мотивов в пиках с учетом частоты его встреч в каждом пике и кластеризации хитов в пределах пика [10].

Таким образом, для любых якорного и партнерского мотивов рассчитывается ожидаемое число совместно встречающихся мотивов (число КЭ), которое и сравнивается с наблюдаемым с помощью точного критерия Фишера. Иллюстрация работы данной программы [2] изображена на рис. 2.

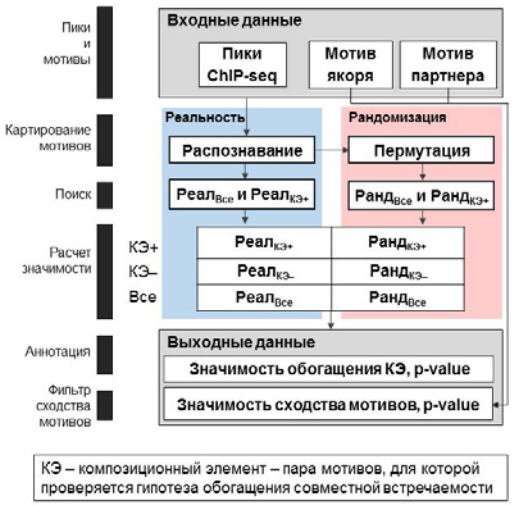

Рис. 2. Иллюстрация алгоритма программы MCOT

Fig. 2. MCOT algorithm illustration

[2] Свидетельство о госрегистрации № 2019613677, зарег. 13.03.2019.

Данный набор программ реализован на языке C++ и представляет собой два проекта на каждый режим работы: «один партнер» и «несколько партнеров». Основную долю вычислений составляет перестановочный тест, поэтому нераспараллеленный алгоритм на данный момент работает долго. Большая часть программного кода, реализующего основные алгоритмы, является идентичным для этих проектов.

На данный момент программный комплекс работает со следующими организмами:

- *Arabidopsis thaliana* (Резуховидка таля);
- *Homo sapience* (Человек разумный);
- *Mus musculus* (Мышь домовая).

Программы принимают на вход как минимум следующие параметры и файлы.

- Fasta-файл с пиками. Этот файл содержит последовательности ДНК, которые были получены из экспериментов. Представляет собой текстовый файл, в котором на одну последовательность выделяется две строки. Первая – описание последовательности (описание может быть различным: в нем могут быть идентификатор последовательности, GC-состав, наименование вида, для которого проводился эксперимент, и др.). Данная строка всегда должна начинаться с символа >. Вторая строка – сама последовательность.
- Якорный мотив – текстовый файл с описанием якорного мотива (для которого проводился эксперимент ChIP-seq). На первой строке, начиная с символа >, идет описание мотива. Начиная со второй строки представляется частотная матрица встречаемости нуклеотидов, где столбцами обозначены типы нуклеотидов (порядок столбцов – A, C, G, T), строками обозначаются координаты мотива, а в ячейке записана частота встречаемости нуклеотида определенного типа на определенной позиции.
- Минимальная и максимальная длина спейсера – рассматриваемый интервал между мотивами. Числа должны быть целыми неотрицательными. Если мотивы окажутся вне этого интервала, то такой комплекс мотивов не будет считаться композиционным элементом.
- Путь до файла с описанием полногеномной выборки промоторов белок-кодирующих генов для конкретного вида организма. В папке должен лежать файл ups2kb.plain, в котором на каждой строке находятся последовательности промоторов.

Как уже было сказано, данный комплекс программ работает в двух режимах.

- «Один мотив». Данная программа принимает дополнительно партнерский мотив, и в таком случае проводится поиск композиционных элементов только для этих мотивов.
- «Несколько мотивов». Данная программа принимает на вход название библиотеки партнерских мотивов и проводится поиск композиционных элементов среди мотивов из библиотеки. Возможные названия баз данных (количество мотивов, целевой организм):

- hs_core (396, *Homo sapience*);
- mm_core (353, *Mus musculus*);
- hs_full (747, *Homo sapience*);
- mm_full (509, *Mus musculus*);
- dapseq (514, *Arabidopsis thaliana*).

На выходе данная программа выводит основные два файла:

- таблица значимости положений двух мотивов. Мотивы могут перекрываться (полностью или частично) или располагаться отдельно друг от друга на некотором расстоянии. Столбцами обозначаются классы местоположений, строками обозначаются идентификатор партнерского мотива, а значениями являются уровни значимости статистической гипотезы;
- числовая гистограмма, в которой для каждого партнерского мотива показана доля пиков, содержащих композиционный элемент заданной структуры. Рассматриваются четыре возможные ориентации, а также все возможные перекрывания и спейсеры для пары мотивов «якорь» – «партнер».

Данная программа работает в консольном режиме, что не очень удобно для биологов-экспериментаторов, привыкших к работе с программами с графическим интерфейсом. Более

того, для обеспечения работы программного комплекса требуется обеспечить распределение задач по нескольким узлам, а данная программа работает только в консольном режиме.

Целью данной работы является разработка программного комплекса WebMCOT для поиска и аннотации совместно встречающихся ДНК-мотивов сайтов связывания транскрипционных факторов. Для достижения данной цели были поставлены следующие задачи:

1) анализ предметной области;
2) анализ требований к системе;
3) построение архитектуры веб-сервиса и системы распределения вычислительных задач на высокопроизводительные платформы;
4) программная реализация веб-сервиса, графического интерфейса и системы распределения вычислительных задач на высокопроизводительные платформы.

Обзор технологии для реализации серверной части

Для быстрой разработки программного обеспечения в части сервера требуется легкий инструмент, таким инструментом является Python [11]. Он является динамическим, объектно-ориентированным и функциональным языком программирования (ЯП).

У данного ЯП существуют большой набор преимуществ для разработки помимо вышеуказанных. Так как данный язык является простым для освоения и быстрым для разработки, то под него были разработаны большое количество библиотек, в частности, для математических расчетов и графиков (numpy, scipy, matplotlib), машинного обучения (scikit-learn для запуска базовых алгоритмов машинного обучения, PyTorch и Tensorflow для глубокого обучения) и разработки веб-приложений (Django, Flask, Tornado).

Для данного проекта для разработки веб-приложения была выбрана библиотека Flask. Данное решение является простым и быстрым для реализации проектов любого размера. Основной его конкурент – Django – MVC-фреймворк для разработки веб-приложений. Однако Django является сложным инструментом, пригодным для реализации более комплексных решений, таких как онлайн-магазин, бухгалтерский учет, управление контента и т. д. Запросы он обрабатывает медленнее, чем Flask, и занимает больше памяти и процессорного времени, что не является приемлемым для данного проекта. Более того, в целом для биоинформатических решений наблюдается тренд на реализацию микросервисной архитектуры веб-приложения [12].

Однако у данного ЯП имеется и ряд недостатков в архитектуре, и один из них – GIL (Global Interpreter Lock) – способ глобальной синхронизации данных между потоками. В языке Python существуют средства для реализации многопоточных приложений, однако по факту работает один поток в один момент времени. Эта особенность негативно влияет на производительность программ. Поэтому желательно, чтобы веб-приложение обрабатывало запрос быстро, соответственно, должно быть написано минимум кода в самом веб-приложении, а «тяжелые» вычисления следует выполнять в отдельных процессах. Более того, чтобы приложением могли пользоваться несколько клиентов одновременно, необходимо иметь возможность запускать несколько процессов одновременно. Такую задачу выполняет веб-сервер uWSGI. Также для асинхронного выполнения сложных задач была выбрана платформа Celery в паре с базой данных Redis. Было выбрано именно это решение, так как оно предоставляет больше функциональности для контроля выполнения задач, в отличии от Dask и DisPy. Однако для последнего существуют эксперименты, в котором Dask работает быстрее чем Celery (http://matthewrocklin.com/blog/work/2016/09/13/dask-and-celery).

Обзор технологий для реализации клиентской части

Для реализации веб-приложения, с которым будет работать клиент, был выбран язык программирования JavaScript [13], так как альтернативы для данной задачи не существует, а реа-

лизация WebAssembly, с помощью которого можно писать клиентские веб-приложения на C++, пока что находится в бета-режиме. Данный язык программирования обладает огромными преимуществами в области динамической работы с DOM (Document Object Model – объектная модель представления XML\HTML-документов) и работы по сети. Для сетевого взаимодействия между клиентской и серверной частями программного комплекса используется библиотека Fetch.

Язык программирования JavaScript является мультипарадигменным: объектно-ориентированным, функциональным, событийно-ориентированным, императивным. Данный язык программирования является интерпретируемым и динамическим языком. Приложения на данном языке работают параллельно с отображением веб-сайта, отображенного в браузере пользователя (Firefox, Chrome, IE, Microsoft Edge, Opera). В основе работ каждого из браузеров лежит важная часть для исполнения программ на данном языке программирования – так называемый «движок». Также движок решает следующий пул задач:

- оптимизация выполнения программ на языке JS с использованием методов Just-in-Time компиляции;
- сборка неиспользуемых объектов в области динамической памяти (куче) программы (сборщик мусора). Под неиспользуемыми объектами подразумевается объект, на который не ссылается ни один указатель;
- обеспечение безопасности выполнение программ: запускаемая JS программа может нести опасность системе и\или целостности, конфиденциальности данных пользователя.

Основными движками являются SpiderMonkey, который используется в браузере Firefox от Mozilla, и V8, который используется в Google Chrome и Chromium. Основная проблема в работе с этими движками на данный момент состоит в том, что они запускают JS-программы в однопоточном режиме, а это очень сильно ограничивает производительности. Более того, существует ограничение по производительности ввиду того, что изменения в DOM являются тяжеловесными, даже если изменился один объект.

В работе с компьютерной сетью, например, программист должен использовать парадигму событийно-ориентированного программирования: для одной функции (назовем ее, например, firstFunc) в качестве параметров требуется как минимум одна функция (например, firstHandle), которая будет функцией-обработчиком. Сначала запускается функция firstFunc, и после успешного выполнения данной функции будет выполняться функция-обработчик в своем контексте. Например, с помощью функции fetch JS-программа сначала запрашивает у сервера нужные данные, а затем – в функции-обработчике – полученные данные присваиваются некоторой переменной таким образом, чтобы эти данные сразу же отобразились на веб-сайте.

Хотя «чистый» JavaScript позволяет выполнять манипуляции над DOM, данная методика не является простой для сопровождения программного обеспечения, существуют сложности синхронизации между данными и отображением. Библиотека Vue.JS решает проблему синхронизации и предоставляет удобный программный интерфейс для манипулирования DOM.

Архитектура программного комплекса

Для данного проекта была реализована следующая архитектура программного комплекса (рис. 3), состоящего из трех частей:

1) веб-приложение WebMCOT, который состоит из веб-сайта и сервиса, реализованного с помощью Flask;
2) система для запуска асинхронных задач на основе Celery;
3) ядро программы, реализованное на C++.

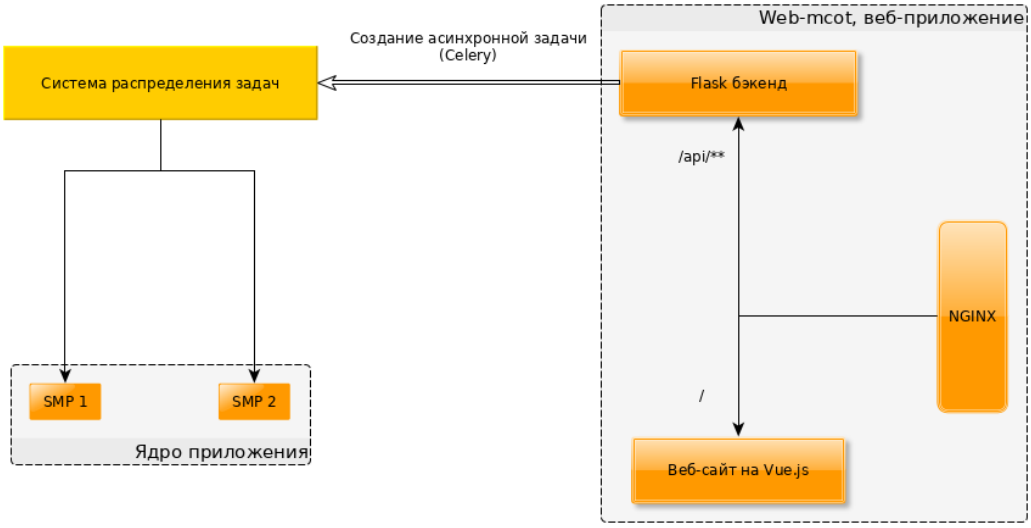

Рис. 3. Архитектура программного комплекса WebMCOT
Fig. 3. WebMCOT architecture

Поток данных организован следующим образом: клиент с помощью браузера отправляет HTTP GET-запрос на получение SPA-приложения, далее клиент отправляет HTTP-POST-запрос с JSON данными на URL /api/send_one (или /api/send_many) для запуска алгоритма поиска ССТФ. Далее запрос и файлы на поиск анализируются, и оформляется задача в терминах Celery для выполнения анализа на кластерах. Для передачи необходимых параметров и данных кластеру, сначала проводится этап сериализации (упаковка данных для удобной передачи другой стороне), и далее данные отправляются в базу данных Redis, которая является в данной архитектуре брокером сообщений. Брокер извещает рабочий процесс, что пришло новое задание для анализа, и рабочий процесс десериализует входные данные и начинает работу ядра программного комплекса.

После загрузки кластера новой задачи клиент каждые 5 секунд отправляет запрос веб-сервис о статусе работы. Если задача еще не до конца выполнена, клиенту отправляется JSON форма со значением “WAITING”.

После успешного поиска мотивов все данные сохраняются на жесткий диск кластера и рабочий процесс сообщает брокеру об успешной работе. Когда клиент, обращаясь к сервису о получении статуса задачи, получает успешный статус, он обращается к URL для получения результатов, которые может скачать в виде набора файлов. Эта схема визуализирована с использованием диаграммы последовательностей UML на рис. 4.

Для веб-сервиса были выбраны следующие языки программирования и технологии:

1) Python – язык программирования, на котором реализована логика сервера;
2) Matplotlib – визуализация диаграмм;
3) Flask – для построения RESTful API, uWSGI – протокол для производительной работы с nginx;
4) JavaScript – язык программирования для веб-сайта;
5) Vue.JS – построение компонент;
6) Fetch – HTTP клиент.

Для распределения задач используется библиотека Celery для соединения с кластерами и выполнения кода по протоколу SSH. В качестве базы данных для передачи параметров и результатов используется Redis.

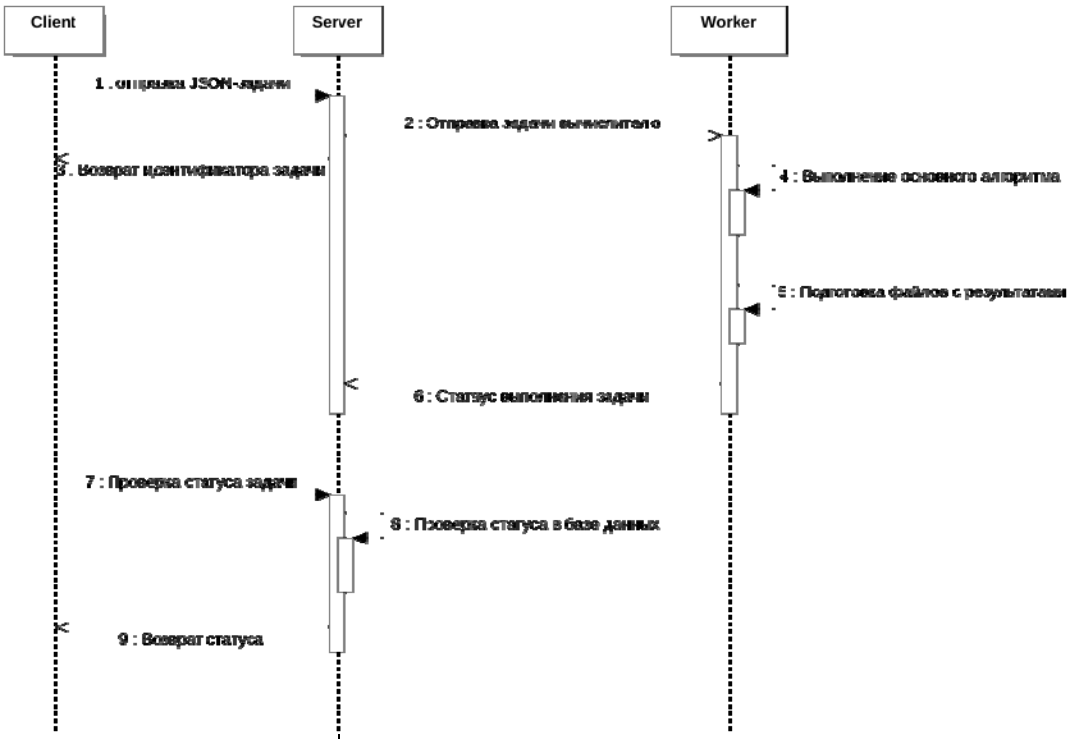

Рис. 4. Диаграмма последовательностей UML взаимодействий между клиентом, сервером и рабочим процессом
Fig. 4. Sequence UML diagram interaction between client, server and worker process

API разработанного комплекса

На данном этапе был реализован следующий API для работы с задачами:

- POST /api/send_one – клиент посылает файлы, создаётся задача в режиме «один мотив», задача ставится в обработку, клиенту возвращается UUID задачи, по которой клиент может получить результаты и статус;
- POST /api/send_many – клиент посылает файлы, создаётся задача в режиме «несколько мотивов», задача ставится в обработку, клиенту возвращается UUID задачи, по которой клиент может получить результаты и статус;
- GET /api/one/get/<task_id>/status – вывод статуса задачи, которую посылали в режиме «один мотив»;
- GET /api/many/get/<task_id>/status – вывод статуса задачи, которую посылали в режиме «несколько мотивов»;
- GET /api/get/<task_id>/hist/count – количество вычисленных гистограмм для задачи task_id, нумерация начинается с 0;
- GET /api/get/<task_id>/hist/<number> – пользователь получает гистограмму уровней значимости типов перекрытий между якорным и партнерским мотивами по типу перекрывания формата SVG с номером number для задачи task_id, если вычисления закончились, иначе 0;
- GET /api/get/<task_id>/pval – табличные значения уровней значимостей различных типов перекрываний, если вычисления закончились, иначе 0.

Описание веб-интерфейса программного комплекса

Интерфейс программы на данный момент реализован с использованием библиотек Vue.JS и языка программирования JavaScript. Сайт реализован по принципу Single Page Application, так как в рамках данной работы реализован RESTful API сервис, и для сайта требуется динамическое управление контентом.

Сайт состоит из трех частей.

1. Стартовая страница. Эта страница кратко описывает проект и алгоритм анализа. Также на этой странице располагаются кнопки перехода на другие страницы, о которых будет написано далее.
2. Страница с помощью (Help). На данной странице размещена исчерпывающая информация об использовании сайта и метода, алгоритм использования сайта для получения результатов, описание параметров и результатов.
3. Страница с описанием веб-сервиса (API). На этом сайте описан RESTful API для использования продвинутыми пользователями. На странице расположена таблица со следующими полями [URL, метод, параметры, описание функции].

Под контентом подразумевается как статический, так и динамический контент, который создается во время работы алгоритма. Статическим контентом являются:

- текст;
- скриншот сайта;
- таблицы с описанием.

Динамический контент создается с помощью ядра программного комплекса. Программный комплекс проводит обработку выходных файлов, которые были описаны во введении, и отправляет их в табличном виде и в виде диаграммы.

На рис. 5 изображен скриншот сайта, в котором пользователь вносит свои данные для анализа. Сначала пользователь указывает FASTA-файл с пиками, мотив, относительно которого проводился ChIP-Seq эксперимент, максимальный размер вероятного спейсера и вид организма. Далее пользователь выбирает один из двух режимов и в зависимости от выбора должен выполнить следующие действия:

«Один партнер» – пользователь отправляет текстовый файл партнерского мотива;

«Несколько мотивов» – пользователь выбирает нужную ему базу данных из предложенных.

Рис. 5. Окно для ввода данных
Fig. 5. Input window

Вывод анализа приведен на рис. 6 и 7. На рис. 6 продемонстрирован скриншот таблицы с уровнями значимостей расположений мотивов друг относительно друга, где числовыми столбцами обозначены различные гипотезы о расположениях якорного и партнерского мотивов (Spacer – со спейсером, Overlap – перекрывание мотивов, Partial – частичное перекрывание, Full – один мотив располагается внутри другого, Any – любое положение мотивов, которые находятся в КЭ). На рис. 7 продемонстрирована гистограмма, показывающая долю пиков (в %), содержащих КЭ каждой из стратегий расположений мотивов между собой (S<число> – мотивы разделены спейсером длины <число>, P<число> – мотивы пересекаются на <число> пар нуклеотидов, F<число> – полная вложенность одного мотива в другой, <число> означает расстояние между ближайшими краями мотивов).

Partner Num	Partner Nam	Any Pv	Full Pv	Partial Pv	Overlap Pv	Spacer Pv
0	Anchor_Homer	1e-300	4.6758e-9	1e-300	1e-300	0.656615
1	AHR_MOUSE.H11MO.0.B	0.732118	0.395738	1	1	0.357528
2	AIRE_MOUSE.H11MO.0.C	0.0413342	0.256436	0.156425	0.157293	0.0147089
3	ALX1_MOUSE.H11MO.0.B	0.150892	1	0.991018	0.995638	0.0203723
4	ANDR_MOUSE.H11MO.0.A	0.0177451	1	0.00000661314	0.000229472	0.161975

Рис. 6. Таблица с уровнями значимостей расположений мотивов друг относительно друга (Spacer – со спейсером, Overlap – перекрывание мотивов, Partial – частичное перекрывание, Full – один мотив располагается внутри другого, Any – любое положение мотивов, которые находятся в КЭ)
Fig. 6. Table with motif location p-values

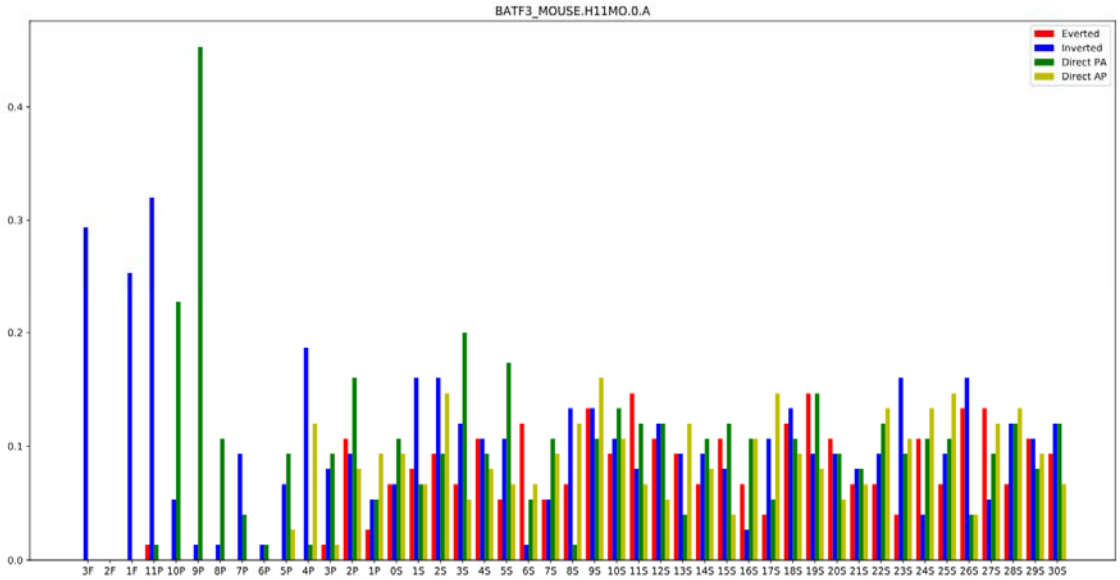

Рис. 7. Гистограмма, показывающая долю пиков (в %, ось Y), содержащих КЭ каждого из четырех возможных ориентаций (разные цвета и подпись в правом верхнем углу) и расположений мотивов, обозначенному на оси X (слева направо: F<число> – полная вложенность одного мотива в другой, <число> – расстояние между ближайшими краями мотивов в п.о., P<число> – мотивы пересекаются на <число> п. о., S<число> – мотивы разделены спейсером длины <число> п. о.)

Fig. 7. Peak Histogram which shows count (in %) composite elements for each location strategy

Заключение

В результате выполнения работы были выполнены следующие задачи:

- реализована архитектура программного комплекса WebMCOT;
- реализованы интерфейс и RESTful API сервер программного комплекса WebMCOT;
- реализована программа на языке Python, которая принимает запросы на запуск вычислительного ядра MCOT и выдачи результатов пользователю.

В дальнейшем будет проводиться работа по оптимизации вычислительного ядра MCOT для ускорения работы программы.

Список литературы / References

1. **Greene C. S. et al.** Big Data Bioinformatics. *J. Cell. Physiol.*, 2014, vol. 229, no. 12, p. 1896–1900. DOI 10.1002/jcp.24662
2. **Morgunova E., Taipale J.** Structural perspective of cooperative transcription factor binding. *Curr. Opin. Struct. Biol. Elsevier Current Trends*, 2017, vol. 47, p. 1–8. DOI 10.1016/J.SBI.2017.03.006
3. **Wasserman W. W., Sandelin A.** Applied bioinformatics for the identification of regulatory elements. *Nat. Rev. Genet.*, 2004, vol. 5, no. 4, p. 276–287. DOI 10.1038/nrg1315
4. **Kel O. V et al.** A compilation of composite regulatory elements affecting gene transcription in vertebrates. *Nucleic Acids Res.*, 1995, vol. 23, no. 20, p. 4097. DOI 10.1093/NAR/23.20.4097
5. **Guo Y., Mahony S., Gifford D. K.** High Resolution Genome Wide Binding Event Finding and Motif Discovery Reveals Transcription Factor Spatial Binding Constraints. *PLoS Comput. Biol.*, 2012, vol. 8, no. 8, p. e1002638. DOI 10.1371/journal.pcbi.1002638
6. **Jankowski A., Prabhakar S., Tiuryn J.** TACO: a general-purpose tool for predicting cell-type-specific transcription factor dimers. BMC Genomics, 2014, vol. 15, no. 1, p. 208. DOI 10.1186/1471-2164-15-208
7. **Kazemian M. et al.** Widespread evidence of cooperative DNA binding by transcription factors in Drosophila development. *Nucleic Acids Res.*, 2013, vol. 41, no. 17, p. 8237–8252. DOI 10.1093/nar/gkt598
8. **Whitington T. et al.** Inferring transcription factor complexes from ChIP-seq data. *Nucleic Acids Res.*, 2011, vol. 39, no. 15, p. e98–e98. DOI 10.1093/nar/gkr341
9. **Levitsky V. et al.** The overlapped motifs co-occurrence in ChIP-seq data. In: Proc. of XI Int. Conf., 2018, p. 58–58. DOI 10.18699/bgrssb-2018-035
10. **Boldyreva L. V et al.** Protein and Genetic Composition of Four Chromatin Types in Drosophila melanogaster Cell Lines. *Curr. Genomics*, 2017, vol. 18, no. 2, p. 214–226. DOI 10.2174/1389202917666160512164913
11. **Srinath K. R.** Python – The Fastest Growing Programming Language. *Int. Reasearch J. Eng. Technol.*, 2018, vol. 4, no. 12, p. 354–357. DOI 10.1016/j.jsams.2016.11.018
12. **Williams C. L. et al.** The growing need for microservices in bioinformatics. *J. Pathol. Inform.*, 2016, vol. 7, p. 45. DOI 10.4103/2153-3539.194835
13. **Theisen K. J.** Programming languages in chemistry: A review of HTML5/JavaScript. *J. Cheminform.*, 2019, vol. 11, no. 1, p. 1–19. DOI 10.1186/s13321-019-0331-1

Материал поступил в редколлегию
Received
21.05.2019

Сведения об авторах

Мухин Алексей Максимович, магистрант 2 курса факультета информационных технологий Новосибирского национального исследовательского государственного университета (Новосибирск, Россия); инженер-программист Института цитологии и генетики СО РАН (Новосибирск, Россия)
mukhin@bionet.nsc.ru

Левицкий Виктор Георгиевич, кандидат биологических наук, старший научный сотрудник Института цитологии и генетики СО РАН (Новосибирск, Россия)
levitsky@bionet.nsc.ru
ResearcherID R-5696-2016

Лашин Сергей Александрович, кандидат биологических наук, ведущий научный сотрудник Института цитологии и генетики СО РАН (Новосибирск, Россия)
lashin@bionet.nsc.ru
ResearcherID D-8142-2012
ORCID 0000-0003-3138-381X

Information about the Authors

Aleksey M. Mukhin, Master Student 2yr, Faculty of Information Technologies, Novosibirsk State University (Novosibirsk, Russian Federation), programmer, Institute Cytology and Genetic SB RAS (Novosibirsk, Russian Federation)
mukhin@bionet.nsc.ru

Victor G. Levitsky, PhD of Biology, Senior Research Scientist, ICG SB RAS (Novosibirsk, Russian Federation)
levitsky@bionet.nsc.ru
ResearcherID R-5696-2016

Sergey A. Lashin, PhD of Biology, Leading Research Scientist, ICG SB RAS (Novosibirsk, Russian Federation)
lashin@bionet.nsc.ru
ResearcherID D-8142-2012
ORCID 0000-0003-3138-381X