федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский национальный
исследовательский университет информационных технологий, механики и
оптики»

На правах рукописи

Сергушичев Алексей Александрович

Методы вычислительного анализа метаболических моделей для интерпретации транскриптомных и метаболомных данных

05.13.18 — Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
PhD, профессор-исследователь
Университета ИТМО,
Артемов М.

Санкт-Петербург — 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Общая характеристика работы

Актуальность проблемы. С развитием технологий все большую роль в фундаментальных задачах биологии и медицины играет сбор больших объемов экспериментальных данных и последующий их анализ и интерпретация. Из-за больших объемов анализ и интерпретация вручную становятся практически невозможными, что влечет за собой необходимость разработки соответствующих вычислительных методов.

Одной из актуальных и быстро развивающихся областей биологии, требующих разработки новых методов для интерпретации данных, является изучение регуляции метаболизма (набора биохимических реакций, необходимых для жизнедеятельности клетки). Во-первых, стала ясна большая роль, которую метаболизм играет в биологических процессах, особенно в иммунной системе и раковых клетках. Во-вторых, появилась возможность широкого изучения метаболических процессов из-за удешевления технологий получения данных транскриптомного и метаболомного профилирования, отражающих активность ферментов и изменения в концентрациях веществ в клетке, соответственно.

Важным понятием, с точки зрения интерпретации данных, является метаболический путь – набор последовательных реакций, связанных одной функцией. Именно в терминах метаболических путей удобно интерпретировать данные, и поэтому можно рассматривать задачу интерпретации как задачу идентификации регулируемых метаболических путей и их взаимосвязей.

Исследовать метаболические пути удобно с помощью анализа метаболических моделей. В них обычно содержится информация о том, какие реакции могут происходить в клетке, как реакции связаны друг с другом, активность каких генов может регулировать реакции, какие из реакций относятся к стандартным метаболическим путям и т. д. С помощью таких моделей, а также экспериментальных данных, можно проанализировать, какие реакции явля-

ются наиболее важными, как они объединяются в метаболические пути и как эти пути взаимодействуют между собой.

Отметим, что хотя существуют наработки в области анализа метаболических путей или, в более общем виде, молекулярных путей, вопрос создания качественных эффективных вычислительных методов является актуальным. Даже для задачи, в которой требуется идентифицировать регулируемые молекулярные пути среди набора заданных путей, имеется несколько конкурирующих подходов. Поиск же регулируемых путей без привязки к заранее заданному набору путей является еще более сложным. Для этого есть как минимум две причины. Во-первых, такие методы сильно зависят от рассматриваемой области и требуют доработки для каждой области отдельно. Во-вторых, многие такие задачи сводятся к оптимизационным задачам на графах, являющимся NP-трудными, что требует разработки специальных методов для нахождения оптимальных или субоптимальных решений.

Таким образом, рассматривая тема является актуальной, как с точки зрения развития вычислительных методов анализа метаболических моделей, так и с точки зрения практической применимости для интерпретации экспериментальных данных в области изучения регуляции метаболизма.

В соответствии с паспортом специальности 05.13.18 – «Математическое моделирование, численные методы и комплексы программ» диссертация относится к трем областям исследований: «3. Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий»; «6. Разработка новых математических методов и алгоритмов проверки адекватности математических моделей объектов на основе данных натурного эксперимента»; «7. Разработка новых математических методов и алгоритмов интерпретации натурного эксперимента на основе его математической модели».

Целью работы является разработка и программная реализация набора эффективных вычислительных методов анализа метаболических моделей для

идентификации регулируемых метаболических путей и их взаимосвязей по транскриптомным и метаболомным экспериментальным данным.

Основные задачи диссертационной работы состоят в следующем:

1. Разработка и реализация эффективного метода идентификации регулируемых путей в метаболических моделях на основе анализа представленности, без использования информации о связях между реакциями.
2. Разработка и реализация эффективного метода идентификации регулируемых путей и их взаимосвязей в метаболических моделях на основе подхода поиска активного модуля.
3. Разработка и реализация эффективного метода идентификации регулируемых путей и их взаимосвязей в метаболических моделях на основе подхода поиска активного модуля с использованием информации об атомной структуре метаболитов.

Научная новизна. В работе получены следующие новые научные результаты:

1. Разработан метод *FGSEA* (*Fast Gene Set Enrichment Analysis*) для проведения эффективного взвешенного анализа представленности функциональных наборов генов. Он позволяет идентифицировать регулируемые метаболические пути, используя только информацию из модели о множестве возможных реакций, их регуляции генами и их участии в метаболических путях. Метод является развитием существующего метода анализа представленности *GSEA* (*Gene Set Enrichment Analysis*). За счет использования разработанного алгоритма кумулятивного вычисления *GSEA*-статистики представленности, он позволяет достичь ускорения в сотни раз.
2. Разработан метод *GAM* (*Genes And Metabolites*) для выделения активных метаболических модулей с помощью анализа сети метаболических реакций. Он позволяет, используя информацию о связях реакций в метаболической модели, идентифицировать регулируемые ме-

таболические пути и их взаимосвязи. По сравнению с существующими методами в нем существует возможность использования нескольких вариантов представления сети реакций в виде графа в зависимости от входных данных. Также для возникающей в общем случае в методе обобщенной задачи поиска связного подграфа максимального веса (*GMWCS*, *Generalized Maximum Weight Connected Subgraph*), являющейся *NP*-трудной, разработан точный решатель.

3. Разработан метод *GATOM* (от *GAM* и *atom*) для выделения активных метаболических модулей с помощью анализа графа атомных переходов. Он позволяет идентифицировать регулируемые метаболические пути и их взаимосвязи, используя информацию о связях реакций в метаболической модели и о внутренней атомной структуре метаболитов. По сравнению с существующими методами в этом методе используется представление сети реакций в виде графа атомных переходов. Для учета структуры этого графа был сформулирован сигнальный вариант задачи *GMWCS* (*SGMWCS*, *Signal GMWCS*). Для задачи *SGMWCS*, также являющейся *NP*-трудной, разработан точный решатель.

Разработанные решатели являются точными в том смысле, что могут найти доказуемо оптимальное решение при неограниченных вычислительных ресурсах.

Методы исследований. В работе используются методы дискретной математики, теории вероятности и математической статистики.

На защиту выносятся:

1. Метод *FGSEA* для проведения эффективного взвешенного анализа представленности функциональных наборов генов.
2. Метод *GAM* для выделения активных метаболических модулей с помощью анализа сети метаболических реакций.
3. Метод *GATOM* для выделения активных метаболических модулей с помощью анализа графа атомных переходов.

Отличия этих методов от известных указаны в разделе *Научная новизна*.

Достоверность научных положений, выводов и практических рекомендаций, полученных в диссертации, подтверждается корректным обоснованием постановок задач, точной формулировкой критериев, результатами экспериментов по использованию предложенных в диссертации методов и их анализом.

Теоретическое значение работы состоит в разработанном алгоритме кумулятивного подсчета *GSEA*-статистики представленности и асимптотической оценке времени его работы, формулировке задачи *SGMWCS*, формулировке задачи поиска активного модуля в виде задач *GMWCS* и *SGMWCS*, разработанных методах решения *NP*-трудных задач *GMWCS* и *SGMWCS*.

Практическое значение работы состоит в том, что разработанные методы могут и уже используются в настоящее время для изучения регуляции метаболизма, играющей особенно важную роль в работе иммунной системы млекопитающих и в развитии раковых опухолей.

Внедрение результатов работы. Программный пакет для анализа представленности, реализующий метод *FGSEA*, принят в библиотеку *R/Bioconductor* (http://bioconductor.org/packages/fgsea). Метод также внедрен в рабочие процессы компании *Immuneering* (Кембридж, США, http://immuneering.com/). Метод *GAM* для анализа сетей реакций используется в компании *Elucidata* (Кембридж, США, http://www.elucidata.io/).

Апробация результатов работы. Основные результаты докладывались на следующих конференциях: 16th Workshop on Algorithms in Bioinformatics (WABI 2016), Оорхус, Дания; Всероссийской научной конференции по проблемам информатики «СПИСОК 2016», СПбГУ, Матмех; Moscow Conference on Computational Molecular Biology (MCCMB'15), 2015, Москва; Cold Spring Harbor Laboratory meeting on Systems Biology: Networks, 2015, Колд-Спринг-Харбор, США; IV международной научно-практической конференции «Постгеномные методы анализа в биологии, лабо-

раторной и клинической медицине», 2014, Казань; Metabolism and Immunity: A Rediscovered Frontier, 2014, Дублин, Ирландия.

Личный вклад автора. Решение задач диссертации, разработанные методы *FGSEA*, *GAM* и *GATOM* принадлежат лично автору, а разработка решателей для задач *GMWCS* и *SGMWCS* была выполнена в соавторстве с А. А. Лободой.

Публикации. Основные результаты по теме диссертации изложены в девяти публикациях [93–101], в том числе одна из них в российском журнале из списка рекомендованных ВАК [93], и шесть, входящих в базы *Scopus* и *Web of Science* [94–99].

Регистрация программ. Автором по теме диссертации было получено свидетельство о регистрации программы для ЭВМ: Сергушичев А. А. Программа для быстрого анализа представленности метаболических путей по упорядоченному списку генов с весами // Свидетельство №2016 660664 от 20.09.2016.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и одного приложения. Объем диссертации – 126 страниц с 40 рисунками и двумя таблицами. Список литературы содержит 101 наименование.

ГЛАВА 1. МЕТАБОЛИЧЕСКИЕ МОДЕЛИ. ОСНОВНЫЕ ПОНЯТИЯ И ПОДХОДЫ К АНАЛИЗУ

В настоящей главе вводятся основные понятия предметной области, рассматриваются метаболические модели и способы их анализа.

1.1. Регуляция метаболизма

В классическом подходе считалось, что регуляция метаболизма необходима клетке в основном для расщепления питательных веществ и получения энергии [1]. В последнее же время стало ясно, что метаболизм тесно связан и с регуляторными функциями [7]. Некоторые метаболиты участвуют в посттрансляционной модификации белков, эпигенетической регуляции и т. д. В частности, относительно недавно стала ясна значительная роль, которую регуляция метаболизма играет в развитии раковых опухолей [8] и в иммунной системе [9].

1.1.1. Основные понятия

Ключевым понятием метаболизма является биохимическая реакция. Реакция состоит в том, что набор веществ, являющихся субстратами реакции, превращается в набор других веществ, являющихся продуктами реакции. Низкомолекулярные вещества, участвующие в биохимических реакциях в клетке, называются метаболитами.

Большинство реакций, важных для работы клетки, проходят с участием катализатора: белка, называемого ферментом, или комплекса белков. Например, реакция $ADP + Phosphoenolpyruvate \rightarrow ATP + Pyruvate$ протекает в мышечных тканях при участии белка PKM (*Pyruvate Kinase, Muscle*). Субстратами этой реакции являются ADP и *Phosphoenolpyruvate*, а продуктами – ATP и *Pyruvate*.

Скоростью прохождения реакции (или потоком через реакцию) называется количество элементарных превращений субстратов в продукты в единицу времени. Она может специфичным для каждой реакции образом зависеть

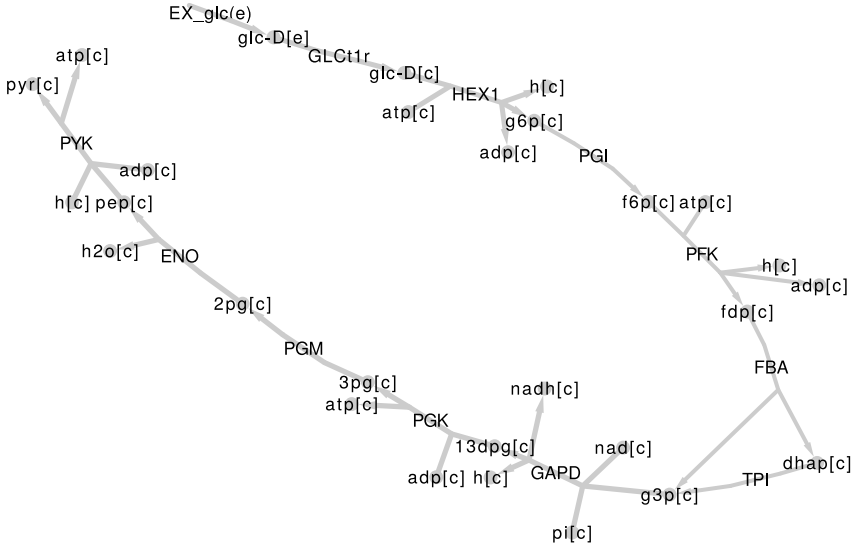


Рисунок 1 – Метаболический путь гликолиза расщепления глюкозы (*glc-D[e]*)

от концентраций субстратов, продуктов и ферментов, а также от других факторов.

Концентрация фермента, в свою очередь, зависит от концентрации матричной РНК (мРНК) транскрипта гена, кодирующего этот фермент. Количество фермента в клетке называется экспрессией фермента, а количество мРНК – экспрессией гена.

Таким образом, регуляция отдельной реакции может происходить за счет изменения концентраций участвующих в ней веществ и за счет регуляции экспрессии соответствующих генов.

Другим важным понятием является метаболический путь. Метаболический путь – это связный набор биохимических реакций, происходящих в клетке. Примером метаболического пути является гликолиз, расщепляющий глюкозу с выделением энергии в виде аденозинтрифосфата (АТФ) (рисунок 1).

Регуляция метаболических путей осуществляется через сложное взаимодействие большого числа факторов [10]. Но, в итоге, клетка может поддерживать гомеостаз в стабильной ситуации и правильным образом реагировать на изменения во внешней среде.

1.1.2. Нецелевое профилирование

Изучению регуляции метаболизма способствует развитие технологий так называемого нецелевого профилирования. В таких технологиях измеряется одновременно большое число параметров (профиль). Нецелевыми они называются, потому что набор измеряемых параметров является большим и практически не зависит от конкретного эксперимента. В контексте регуляции метаболизма обычно рассматривают два типа такого профилирования: транскриптомное и метаболомное.

Транскриптомное профилирование позволяет практически для всех генов оценить их экспрессию. В настоящее время для этого применяются две технологии: микрочипы [2, 11] и РНК-секвенирование [12]. Первая технология появилась раньше и позволяет определять экспрессию для большого, но конечного числа генов. Вторая – позволяет измерить экспрессию всех значительно транскрибируемых генов. В целом, даже первая технология хорошо покрывает пространство ферментов.

Метаболомное профилирование позволяет, в свою очередь, оценить концентрации большого числа метаболитов [13]. Из-за использования в своей основе методов масс-спектрометрии у этой технологии есть две особенности. Во-первых, с помощью нее сложно различить метаболиты, имеющие одинаковую массу. Во-вторых, сложно получить сигнал для некоторых метаболитов, в частности, у которых слишком большая или слишком маленькая масса.

Отметим, что оба метода не позволяют напрямую узнать реальные концентрации активных ферментов и метаболитов. Это накладывает ограничения на методы анализа и интерпретации этих данных.

1.1.3. Анализ дифференциальной экспрессии

Базовым инструментом для работы с данными нецелевого профилирования является анализ дифференциальной экспрессии [14, 15]. При таком анализе рассматривается два состояния клеток: например, контрольное и по-

сле какого-либо воздействия. Профилирование производится для нескольких образцов, относящихся к первому и второму состоянию.

Целью этого анализа является выделить сигналы, уровни которых значимо отличаются от одного состояния к другому. Этот анализ может применяться как для транскриптомных, так и для метаболомных данных. Результатом анализа являются P-значения тестов дифференциальной экспрессии для каждого гена или метаболита при нулевой гипотезе, состоящей в отсутствии регуляции.

1.2. Полногеномные метаболические модели

1.2.1. Структура метаболических моделей

В общем случае полногеномные метаболические модели пытаются обобщить в себе имеющиеся представления о метаболизме и его регуляции в клетке организма [16]. Упор делается на наиболее широком описании, не сконцентрированном на отдельных процессах (отсюда «полногеномные» в названии). Такие модели могут включать в себя:

1. Список всех возможных реакций. При этом могут включаться не только обычные биохимические реакции, но и, например, образование комплекса ферментов. Кроме этого, могут различаться реакции, происходящие в разных отделах клетки: например, в митохондрии или во внутриклеточной жидкости.
2. Набор правил регуляции отдельных реакций. Такие правила могут быть разной сложности: от простых правил, что для протекания реакции необходим комплекс ферментов, до правил с более сложными связями как, например, снижение эффективности фермента при наличии большой концентрации вещества-ингибитора.
3. Наличие термодинамических ограничений, задающих возможные направления реакций.

4. Параметры связи с окружающей средой, например, наличие веществ во внешней среде и возможность их транспорта внутрь клетки.

Для записи таких моделей может использоваться, например, язык *SBML* [17]. Модель на языке *SBML* может включать в себя:

1. Отделы. Отделом может являться любой «резервуар» конечного размера.
2. Вещества. Любые вещества, которые могут участвовать в реакциях.
3. Реакции. Утверждения о возможности превращения, транспортировки или связывания веществ. Реакции может быть сопоставлен кинетический закон.
4. Параметры. Поддерживаются как глобальные параметры, общие для всей модели, так и параметры отдельных реакций. Например, могут быть заданы ограничения на направление и предельную скорость реакции.
5. Единицы измерения. Могут описаны, единицы, использующиеся по умолчанию, аббревиатуры для их комбинаций и т. д.
6. Правила. Модель может быть дополнена количественными правилами, которые нельзя выразить в описании реакций.

Также в модель могут быть включены дополнительные элементы, например:

1. Описание генов, необходимых для прохождения реакции. Такое описание может быть представленно, например, в виде булевой формулы.
2. Разбиение реакций на подсистемы и метаболические пути.

1.2.2. Метаболические базы данных

Метаболические модели могут браться из разных источников. В некоторых они доступны в явном виде в виде *SBML*-файлов. В других – требуется некоторые дополнительные шаги для их получения.

База *KEGG* [18, 19] содержит в себе большое количество информации о метаболических реакций. Она включает в себя несколько отдельных баз данных, например, таких, как:

— *KEGG REACTION* – с информацией о биохимических реакциях;

— *KEGG COMPOUND* и *KEGG GLYCAN* – о метаболитах;

— *KEGG ENZYME* – о ферментах;

— *KEGG GENES* – о генах;

— *KEGG PATHWAY* и *KEGG MODULES* – о метаболических путях.

Всего в этой базе представлено около 4000 организмов, включая 123 животных. База является курируемой и требует платной лицензии для полного доступа. Некоторая информация доступна бесплатно с помощью веб-интерфейса [86] и программного интерфейса.

База данных *BioCyc* [20] является аналогичной базе *KEGG*. Она содержит около 7600 организм-специфичных баз. Семь из них являются полностью курируемыми (проверяемыми людьми), включая базу для человека. Еще 43, включая мышь – частично курируемые. С 2016 г. для базы *BioCyc* осуществляется переход на платную подписку.

В явном виде метаболические модели хранятся в базе *EBI BioModels Database* [21, 22]. В ней доступно около 2500 полногеномных метаболических моделей в формате *SBML*, но все они сгенерированы автоматически из баз *KEGG* и *MetaCyc*. Доступ к базе свободный.

База данных *Reactome* [23] является еще одной бесплатной базой с информацией о метаболических и молекулярных путях. База является полностью курируемой, но сконцентрирована в основном на биологии человека. Реакции в базу не всегда добавляются быстро. Данные из базы могут выгружены как в формате *SBML*, так и в других форматах. Доступ к базе свободный.

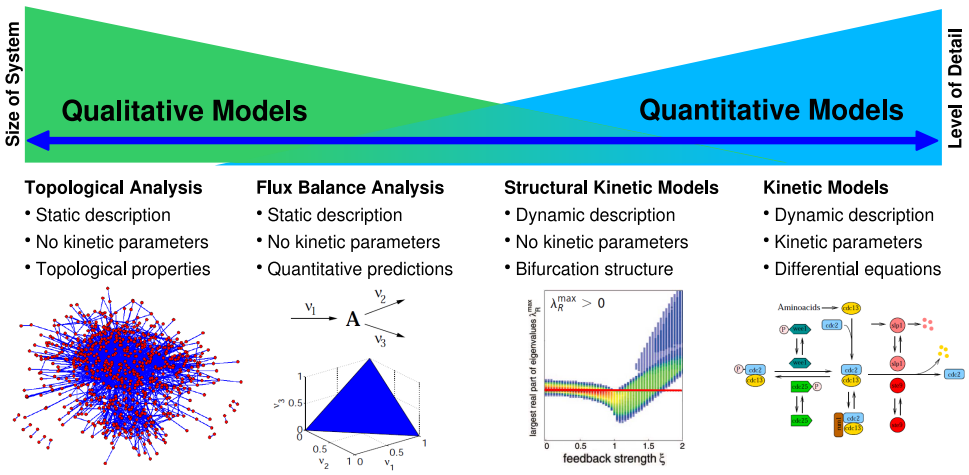


Рисунок 2 – Методы анализа метаболических моделей, упорядоченные по степени детализации моделей. Рисунок заимствован из [24]

1.2.3. Методы анализа метаболических моделей

Для анализа метаболических моделей существует множество методов [24]. Одной из их основных характеристик является степень детализации моделей, с которыми они работают (рисунок 2). Некоторые из методов работают с наиболее детализированными моделями, включающими кинетические параметры реакций, и используют аппарат дифференциальных уравнений. Это обеспечивает возможность на их основе делать количественные предсказания. Другие методы, наоборот, работают только с информацией о связях между реакциями. С одной стороны, эти методы позволяют делать только качественные предсказания, а с другой — позволяют работать с большими моделями.

Анализ детализированных кинетических моделей является исторически первым подходом [24]. С помощью описания модели на уровне дифференциальных уравнений он обеспечивает точное описание возможных состояний системы и ее динамику [3]. Это дает возможность понимания регуляции отдельных метаболических путей. К сожалению, такие методы плохо масштабируются на большие модели. Это обусловлено недостатком точных измерений кинетических параметров реакций, а также вычислительной сложностью подобного анализа.

Для анализа полногеномных метаболических моделей применяется метод баланса потоков (*flux balance analysis*, *FBA*) и его разновидности [4, 25, 26]. В этих методах вводится предположение стационарности: поддержания концентраций веществ на постоянном уровне. Математически, это выражается в виде:

$$Sv = 0,$$

где v – вектор величин потоков через реакции, а S – матрица стехиометрических коэффициентов реакций, в которой строчки соответствуют метаболитам, а столбцы – реакциям. Методы баланса потоков позволяют делать численные предсказания, анализируя пространство возможных потоков в той или иной ситуации. Эти методы хорошо зарекомендовали себя для организмов уровня бактерий и дрожжей, в которых можно составить достаточно точную структурную модель организма [27]. Одним из недостатков этих методов является необходимость наличия достаточно хорошей модели, содержащей множество ограничений [28]. В противном случае эти методы не могут делать нетривиальные предсказания. В настоящее время метаболические модели уровня млекопитающих не настолько проработаны, как, например, некоторые бактериальные модели. Тем не менее, попытки использования этих методов на больших моделях ведутся [29]. Другим важным недостатком этих методов является сложность использования в них метаболомных данных. Это нетривиально, так как по концентрациям веществ в стационарном состоянии сложно сказать что-либо про потоки без знания кинетических параметров реакций.

Менее требовательным к детализации метаболических моделей являются методы топологического сетевого анализа. В самом простом случае может быть выполнен анализ соседних элементов в сети из ферментов, реакций и метаболитов [30]. Такие методы уже хорошо применимы для анализа метаболических сетей растений [31] и человека [32]. В более сложном случае может

ставится задача поиска активного метаболического модуля. В этом случае целью является выделение фрагмента заданной сети реакций, гены и метаболиты которых имеют регулярное поведение в экспериментальных данных. Одним из первых такой подход в контексте метаболических моделей предложили использовать К. Патил и Дж. Нилсен в [33]. Более подробно такие методы изложены в разделе 1.4. Важной особенностью такого подхода является возможность не только находить регулируемые известные метаболических путей по отдельности, но и выявлять новые пути, а также их взаимосвязи.

Другим подходом к анализу метаболических моделей является выполнение анализа представленности на метаболических путях [34]. Этот анализ состоит в том, что из набора метаболических путей выделяются те, гены и/или метаболиты которых хорошо представлены среди индивидуально регулируемых генов и метаболитов. Такие методы достаточно легко применяются как к транскриптомным, так и метаболомным данным [35–37]. Более подробный обзор методов представленности приведен в разделе 1.3.

1.2.4. Использование информации об атомной структуре метаболитов

Отдельно выделим методы, которые используют атомную структуру метаболитов. В них рассматриваются как отдельные атомы одних метаболитов переходят в атомы других метаболитов в биохимических реакциях (рисунок 3).

Наибольшее распространение эти методы получили в контексте анализа метаболомных данных экспериментов, в которых использовались вещества меченые атомами углерода-13 [38, 39]. Эти методы направлены на определение абсолютных и относительных потоков через реакции с помощью сопоставления количеств одного и того же метаболита с разной долей атомов углерода-12 и углерода-13. Такие методы в основном применяются для ана-

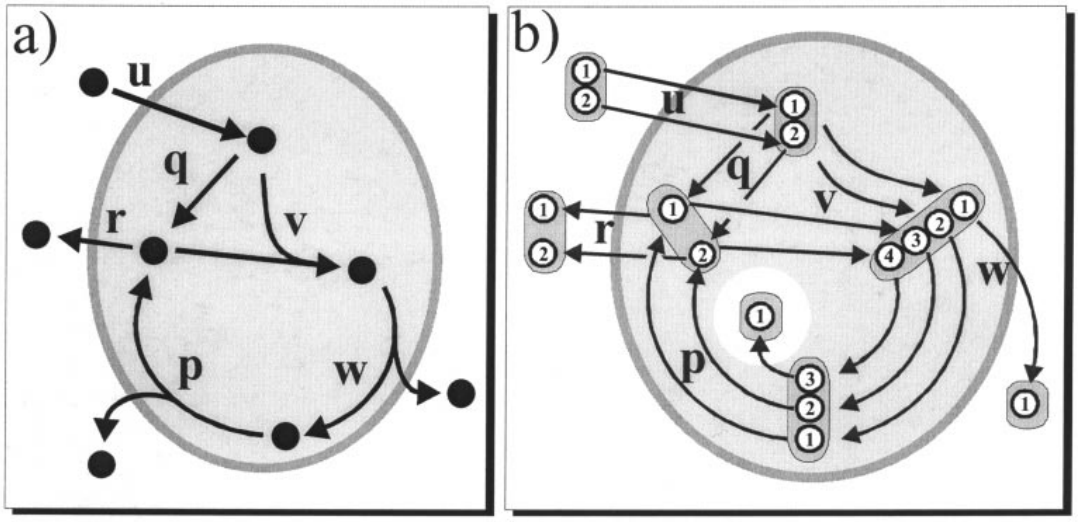


Рисунок 3 – Пример одного и того же метаболического пути, рассматриваемого на разных уровнях: на уровне метаболитов (a) и на уровне атомов углерода (b). Рисунок заимствован из [38]

лиза отдельных метаболических путей, но существуют примеры применения и на бактериальных геномах [40].

Другие методы используют информацию об атомной структуре для поиска потенциальных метаболические путей [41, 42]. Они основаны на том, что во всех метаболических путях происходит перенос атомов углерода от исходных веществ к конечным веществам путей. Таким образом, можно выделить метаболические пути, выполняя поиск путей в графе атомных переходов.

1.3. Анализ представленности

На настоящий момент существует множество методов и программ для выполнения представленности наборов генов [43–45]. Общая суть этих методов состоит в том, чтобы рассмотреть большое число функционально связанных наборов генов и выделить из них те, которые являются наиболее важными в рассматриваемом эксперименте: те, которые наиболее представлены среди сильно регулируемых индивидуальных генов. Развитие этих методов связано с появлением технологий высокопроизводительного профилирования, которые позволили получать информацию о большом числе генов одновременно.

Следуя [43], можно выделить три класса методов для анализа представленности:

— *простой анализ представленности* принимает на вход список «интересных» генов;

— *беспороговый анализ представленности* работает с информацией обо всех генах;

— *модульный анализ передставленности* учитывают информацию о связях между наборами генов.

1.3.1. Простой анализ представленности

Традиционным классом методов для анализа представленности является простой анализ представленности (*singular enrichment analysis*) [43]. Такие методы обычно основаны на точном тесте Фишера. Из-за своей простоты они нашли широкое применение для интерпретации данных в разных направлениях биоинформатики.

Общая их идея состоит в следующем. Пусть задан набор всех генов G. Из них с помощью анализа экспериментальных данных выбирается набор «интересных» генов q. Например, это могут быть все значимо регулируемые гены или только самые регулируемые. Кроме этого, дан список наборов генов $P = \{p_1, p_2, \ldots, p_m\}$, где $p_i \subset G$. Для каждого p_i можно рассмотреть размер пересечения $q \cap p_i$. Если этот размер большой, то можно предположить, что функция, связанная с набором p_i, играет важную роль в рассматриваемом эксперименте. Для того, чтобы оценить статистическую значимость этого утверждения, можно вычислить P-значение – вероятность получения настолько большого пересечения для случайного набора генов. Эта вероятность может быть вычислена из гипергеометрического распределения.

Необходимым дополнительным шагом в этих методах является поправка на множественное сравнение. Эта необходимость возникает из-за того, что часто одновременно тестируется большое число наборов генов, и существу-

ет вероятность получить значимые P-значения случайно. Для этой поправки обычно применяется метод Бенджамина-Хохберга [46], позволяющий контролировать уровень *FDR* (*False Discovery Rate*, доля ложных отклонений нулевой гипотезы среди всех отклонений).

1.3.2. Беспороговый анализ представленности

Беспороговые методы, в отличие от простых методов представленности, используют не просто один набор генов q, а информацию о всех генах G. Это позволяет избавиться от необходимости выбора порога, по которому выбирается набор генов и от которого могут сильно зависеть результаты анализа. С другой стороны, это усложняет процедуру.

Одним из первых предложенных методов для беспорогового анализа представленности был метод *GSEA* (*Gene Set Enrichment Analysis*) [34, 47]. Этот же метод является и наиболее распространенным в своем классе: по данным *Scopus* на эту статью ссылаются почти 7500 раз. В этом методе предлагается сопоставить каждому гену i вещественное число S_i, отражающее степень и направление индивидуальной регуляции. Так для генов, экспрессия которых сильно идет вверх при воздействии, S_i является большим положительным значением, для тех, экспрессия которых идет вниз, – большим отрицательным. Гены упорядочиваются так, что $S_i > S_j$ для $i < j$. Затем для набора генов p вычисляется значение специальной *GSEA*-статистики, являющейся взвешенным расширением статистики Колмогорова-Смирнова. Более формально, для каждого гена вычисляется индивидуальное значение представленности ES_i по формуле:

$$\mathrm{ES}_i = \begin{cases} 0 & \text{если } i = 0, \\ \mathrm{ES}_{i-1} + \frac{1}{\mathrm{NS}}|S_i| & \text{если } 1 \leqslant i \leqslant N \text{ и } i \in p, \\ \mathrm{ES}_{i-1} - \frac{1}{N-k} & \text{если } 1 \leqslant i \leqslant N \text{ и } i \notin p, \end{cases}$$

где $\mathrm{NS} = \sum_{i \in p} |S_i|$, $N = |S|$ и $k = |p|$. Итоговое значение статистики $s_r(p)$ соответствует наибольшему по модулю индивидуальному значению представленности ES:

$$s_r(p) = \mathrm{ES}_{i^*}, \text{ где } i^* = \arg\max_i |\mathrm{ES}_i|.$$

Для оценки статистической значимости выполняется сравнение с распределением значений *GSEA*-статистики для случайных наборов генов такого же размера $|p|$, либо для случайных перестановок меток образцов, при достаточном их числе. С одной стороны, использование перестановки образцов является предпочтительным, так как позволяет учесть корреляцию между экспрессией генов, не зависящую от экспериментальных состояний. С другой стороны, в большинстве экспериментов используется достаточно мало образцов, что затрудняет такой анализ. Поэтому получил распространение взвешенный вариант метода *GSEA* – *pre-ranked GSEA*, который принимает на вход не таблицу экспрессии, а только вектор весов генов S, а сравнение проводится только со случайными наборами генов.

Недостатком метода *GSEA* является необходимость проведение сэмплирования для вычисления фонового распределения *GSEA*-статистики, что является вычислительно затратным. Так, типичный анализ с помощью референсной реализации с вычислением фонового распределения по 1000 случайных наборов занимает несколько минут на персональном компьютере. Это потребовало от авторов разработки специального алгоритма поправки на множественное сравнение, основанном на методе Бенджамина-Хохберга, но не обладающего его свойствами контроля уровня ошибок. Несмотря на это, метод широко используется. Кроме референсной реализации [87], существует еще несколько [48, 88], в некоторых из них используется поправка Беджамини-Хохберга [49]. Кроме этого, недавно был опубликован метод, позволяющий ускорить метод *GSEA* с помощью использования вычислений на видеокартах [50].

Кроме метода *GSEA* существуют и другие беспороговые подходы. Некоторые в основе используют метод *GSEA*, но используют приближенные методы вычисления *P*-значений [51]. Другим достаточно перспективным методом является метод *CAMERA* [52], в котором предлагается подход для учета межгенных корреляций без выполнения перестановок образцов.

1.3.3. Модульный анализ представленности

Методы модульного анализа представленности обычно в качестве основы используют простые методы представленности, но дополняют их информацией о связях между наборами генов [53, 54].

Например, в методе *topGO* [53] рассматривается база молекулярных путей *GO* [55]. В этой базе наборы генов расположены иерархически, что учитывается в анализе. Основной идеей метода является рассмотрение наборов в порядке от наиболее специфичных и малых по размеру к менее специфичным и большим. При этом, если два набора генов являются одинаково значимыми в смысле простого анализа представленности, а первый набор больше и стоит выше в иерархии, большая значимость назначается второму набору, так как он более специфичный.

Таким образом, методы модульного анализа являются надстройкой на другие методы представленности, упрощающей интерпретацию анализа большого числа сложно-связанных наборов генов.

1.4. Задача поиска активного модуля

С момента выхода первых статей про задачу поиска активного модуля было разработано несколько методов, отличающихся в математической формулировке этой задачи и методах ее решения. Рассмотрим некоторые из них.

1.4.1. Исходная формулировка поиска активного модуля

Одними из первых предложили выделять активные регуляторные модули Т. Айдекер и соавторы [56]. Они предложили рассматривать профили

экспрессии генов совместно с регуляторными сетями, в которых узлами являлись гены, а ребрами – их возможные взаимодействия.

Авторами был предложен подход, позволяющий оценить важность подсети. Для этого они рассматривали P-значения p_i анализа дифференциальной экспрессии для каждого гена i. Затем, каждому гену присваивался вес $z_i = F^{-1}(1 - p_i)$, где F^{-1} – обратная функции распределения стандартного нормального распределения. Для подсети A из k генов вес z_A назначался по форумле:

$$z_A = \frac{1}{\sqrt{k}} \sum_{i \in A} z_i.$$

При таком методе оценки выполняется важное свойство: если все z_i для $i \in A$ выбраны из стандартного нормального распределения, то и z_A будет распределено стандартно нормально, вне зависимости от размера k. Таким образом, большие значения z_i будут соответствовать генам с маленькими P-значениями, а подсети с высоким z_A будут соответствовать биологически важным активным модулям.

Для того, чтобы лучше отразить важность связности, строится фоновое распределение веса z_A для случайных наборов генов, без учета связности. Эта процедура выполняется с помощью сэмплирования случайных наборов генов и вычисления для них веса z_A. Затем для каждого размера k вычисляется среднее значение веса μ_k и стандартное отклонение σ_k. С использованием этих параметров вводится скорректированный вес подсети s_A:

$$s_A = \frac{z_A - \mu_k}{\sigma_k}.$$

Задача поиска связной подсети с максимальным весом является NP-трудной, в связи с чем авторы использовали для поиска таких подсетей метод отжига [57]. Кроме этого авторами было разработано несколько эвристик. Метод доступен в виде плагина *jActiveModules* для программы *Cytoscape* [58].

В [33] этот же подход было предложено применять и для метаболических моделей. Главным отличием было использование сети ферментных взаимо-

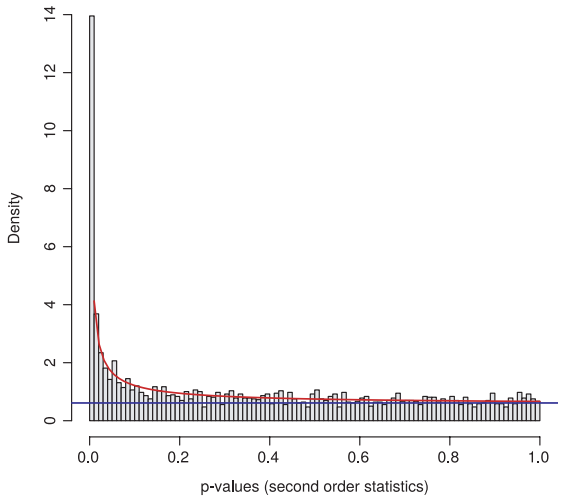


Рисунок 4 – Пример гистограммы P-значений и плотности соответствующего бета-равномерного распределения

действий через реакции, где ребрам соответствовало наличие у ферментов общего субстрата или продукта. Поиск активных модулей так же выполнялся с помощью метода отжига. Этот же метод применялся и в нескольких работах для метаболической моделей клеток человека [59, 60].

1.4.2. Формулировка через сведение к задаче поиска связного подграфа максимального веса

М. Диттрих с соавторами в [61, 62] предложили другую математическую формулировку задачи поиска активного модуля. В этой работе рассматривается модель максимального правдоподобия, где для каждого гена определяется вероятность принадлежности его активному модулю из данных дифференциальной экспрессии. В этом случае задача максимизации правдоподобия становится эквивалентной поиску связного подграфа максимального веса (*Maximum Weight Connected Subgraph*, *MWCS*).

Следуя [63], авторы рассматривают распределение P-значения как смесь бета-распределения $B(\alpha, 1)$ и равномерного распределения $\mathcal{U}(0, 1)$, что хорошо соответствует наблюдаемым данным (рисунок 4). Предполагается, что бета-распределение соответствует сигналу, а равномерное – шуму.

По аналогии с тестом отношения правдоподобия вес одного гена для P-значения p_i определяется по формуле:

$$S(p_i) = \log\left(\frac{B(\alpha,1)(p_i)}{\mathcal{U}(0,1)(p_i)}\right) = \log\alpha + (\alpha-1)\log p_i.$$

Из-за использования модели смеси распределений становится возможной оценка уровня FDR. Вводится порог на P-значения $\tau(FDR)$, позволяющий контролировать этот уровень, а также делается соответствующая поправка на вес гена:

$$S^{FDR}(p_i) = (\alpha-1)\Big(\log x - \log\big(\tau(FDR)\big)\Big).$$

Вес подграфа определяется как сумма весов отдельных генов. Поиск активного модуля – модуля с максимальным весом – соответствует задаче $MWCS$.

Важным преимуществом этого метода по сравнению с методами, описанными выше, является наличие практического решателя, который позволяет за приемлемое время находить хорошие или даже доказуемо оптимальные решения задачи $MWCS$. Изначально авторы для решения использовали сведение к другой NP-трудной задаче – варианту задачи Штейнера – задаче $PCST$ (*Prize Collecting Steinter Tree*). Экземпляр $PCST$ затем решался с помощью существующего точного решателя [64].

В конце своей статьи авторы проводят сравнение с методом *jActiveModules* и показывают, что сведение к задаче $MWCS$ позволяет значительно лучше и стабильнее находить активные модули на модельных данных. В более поздней работе [65] тот же коллектив показал, что такая формулировка является устойчивой к шуму.

Этот же подход в [66] был применен для метаболической модели организма тихоходки – типа микроскопических беспозвоночных. В этой работе авторы рассматривали граф, в котором вершинам соответствовали метаболиты, а ребрам – реакции. И метаболитам, и реакциям назначался вес, исходя из данных метаболомного и транскриптомного профилирования.

1.4.3. Другие подходы к постановке задачи активного модуля

Кроме перечисленных выше, существуют и другие подходы к определению и поиску активного модуля.

В [67] предлагается метод *NetWeAvers*. В нем также рассматривается список индивидуальных P-значений для всех генов и регуляторная сеть межгенных взаимодействий. В этой сети с помощью алгоритма *Walktrap* [68], использующего модель случайных блужданий, выполняется поиск сильносвязных модулей. Поиск происходит без учета P-значений. После выделения модулей выполняется их оценка с использованием среднего или медианного P-значений в модуле. Статистическая значимость оценивается с помощью перестановочного теста. Концептуально, можно сказать, что в этом методе сначала независимо от экспериментальных данных из сети выделяются потенциальные функциональные наборы генов, которые затем анализируются по типу анализа представленности.

Другой подход предложен в методе *KeyPathwayMiner* [69]. В нем формулируется задача поиска наибольшего модуля, содержащего не больше заданного числа слабо регулируемых, «исключительных», генов. Так же, как *MWCS*, эта задача является NP-трудной. Авторами было предложено несколько алгоритмов для ее решения, два неточных: жадный и муравьиный алгоритмы, и один точный алгоритм на основе метода ветвей и границ. Отметим, что точный алгоритм работал за приемлемое время только для небольших значений числа «исключительных» генов. В [70] тем же коллективом метод был расширен для поддержки нескольких типов данных. В этом случае, на модуль ставилось одновременно несколько ограничений по числу исключительных узлов по разным типам данных. К серьезным недостаткам этого метода можно отнести сложность выбора параметров алгоритма.

1.5. Подходы к решению задачи поиска связного подграфа максимального веса

В этом разделе рассмотрим варианты задачи *MWCS* и подходы к ее решению с помощью сведения к задаче целочисленного программирования.

1.5.1. Варианты задачи подграфа максимального веса

Во всех вариантах задачи *MWCS* требуется найти в данном графе максимальный по весу связный подграф. Отличия формулировок заключаются в том, какие веса могут быть у вершин и ребер графа, и в возможных дополнительных ограничениях на подграф. Во всех вариантах можно рассматривать компоненты связности независимо, поэтому для упрощения будем предполагать, что исходный граф связен, если явно не указано обратное.

Наиболее часто под задачей *MWCS* понимают задачу, в которой веса есть только у вершин [71, 72]. Будем называть этот вариант простым вариантом – *SMWCS* (*Simple MWCS*). Пусть дан связный неориентированный граф $G = (V, E)$ и весовая функция $\omega : V \to \mathbb{R}$, тогда задача *SMWCS* состоит в поиске связного подграфа $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ с максимальным суммарным весом:

$$\Omega(\widetilde{G}) = \sum_{v \in \widetilde{V}} \omega(v) \to max.$$

Важным свойством задачи *SMWCS* является то, что ее оптимальное решение всегда является ациклическим графом. Это, в частности, позволяет свести ее к задаче *PCST*, как было сделано в [61].

Также можно ввести обобщенный вариант этой задачи – *GWMCS* (*Generalized MWCS*), в котором могут быть взвешены и вершины, и ребра, без ограничений на знак весов. Такой вариант, например, рассматривается в [73]. Пусть дан связный неориентированный граф $G = (V, E)$ и весовая функция $\omega : (V \cup E) \to \mathbb{R}$, задача *GMWCS* состоит в поиске связного подграфа $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ с максимальным суммарным весом:

$$\Omega(\widetilde{G}) = \sum_{v \in \widetilde{V}} \omega(v) + \sum_{e \in \widetilde{E}} \omega(e) \to max.$$

Задача *SMWCS* тривиальным образом может быть сведена к задаче *GMWCS*, в которой веса всех ребер равны нулю. В то же время, тривиальное обратное сведение невозможно из-за того, что в общем случае решением задачи *GMWCS* может быть подграф с циклами.

Кроме этого можно выделить ациклический вариант – *AMWCS* (*Acyclic MWCS*). В нем, как и в *GMWCS*, могут быть взвешены и вершин, и ребра без ограничений на знак, но рассматриваются только ациклические решения. Этот вариант задачи используется в [66], где задача поиска активного модуля применяется к метаболическим сетям. В [65] тем же коллективом показывается, что эта задача так же, как и *SMWCS*, может быть сведена к задаче *PCST*.

Еще могут быть определены корневые варианты задач *SMWCS* и *GMWCS*: *R-SMWCS* (*Rooted SMWCS*) и *R-GWMCS* (*Rooted GMWCS*), используемые при решении соответствующих некорневых вариантов. В этих задачах рассматриваются не все подграфы, а только те, которые содержат некоторую заранее заданную вершину $r \in V$, называемую корнем.

Все приведенные варианты задачи *MWCS* являются *NP*-трудными.

1.5.2. Сведение задачи *SMWCS* к задаче целочисленного линейного программирования

По результатам соревнования *DIMACS11* [89] одним из лучших алгоритмов для решения задачи *SMWCS* является решатель *Heinz2* [72]. Решатель состоит из процедур предобработки, упрощающих граф, метода декомпозиции, разбивающего экземпляр на более мелкие подзадачи, и сведения к задаче целочисленного линейного программирования.

Основные правила предобработки достаточно просты и состоят в объединении групп соседних неотрицательных вершин, объединении цепочек от-

рицательных ребер и некоторых других. Кроме этого, вводится несколько дополнительных правил.

В решателе *Heinz2* декомпозиция использует разбиение графа на двусвязные компоненты. Для этого последовательно рассматриваются компоненты двусвязности с только одной точкой сочленения. Для каждой компоненты решается некорневая и корневая задачи, а корнем является единственная точка сочленения. После этого в графе эта компонента заменяется на одну вершину с весом, равным весу оптимального решения корневой задачи.

Ядром решателя является сведение к задаче целочисленного программирования. При этом для каждой вершины v вводится бинарная переменная x_v, принимающая единичное значение в случае, если вершина v принадлежит подграфу, и нулевое – в противном случае.

Затем можно определить целевую функцию:

$$\sum_{v \in V} \omega(v) x_v \to max.$$

Для описания ограничений на связность в подграфе выделяется корневая вершина и вводятся дополнительные бинарные переменные r_v, такие, что $r_v = 1$ только для вершины v, являющейся корнем:

$$\sum_{v \in V} r_v = 1;$$

$$r_v \leqslant x_v, \forall v \in V.$$

Используется незамкнутая формулировка, включающая экспоненциально много ограничений:

$$x_v \leqslant \sum_{u \in \delta(S)} x_u + \sum_{u \in S} r_u, \forall v \in V, \{v\} \subset S \subset V.$$

Эти ограничения требуют, чтобы для всех множеств вершин $S \subset V$, содержащих вершину подграфа v, выполнялось одно из двух условий:

— либо внутри S есть корневая вершина,

— либо в решение взята какая-то из вершин $\delta(S) = \{v \in V \setminus S \mid \exists u \in S : (u, v) \in E\}$ – множества вершин, «прилегающих» к S.

Так как этих ограничений экспоненциально много, они добавляются по мере решения задачи [74]. Для этого в некоторые моменты решения с помощью поиска минимального разреза выполняется поиск самого нарушенного ограничения, которое и добавляется.

1.5.3. Сведение задачи *GWMCS* к задаче целочисленного линейного программирования

Возможное сведение к задаче целочисленного программирования задачи *GMWCS* приведено в [73]. Отметим, что в этой работе не разрабатывается практический решатель для этой задачи.

Для представления подграфа используются два набора бинарных переменных y_v и w_e, определяемых присутствием соответствующих вершин v и ребер e в подграфе. Корректность подграфа гарантируется следующими ограничениями:

$$w_e \leqslant y_v, \qquad \forall v \in V, e \in \delta_v. \tag{1.1}$$

Целевая функция задается как:

$$\sum_{e \in E} \omega(e) w_e + \sum_{v \in V} \omega(v) y_v \to max. \tag{1.2}$$

Затем формулируются ограничения на связность. Эти ограничения основаны на идее, что для любого связного подграфа для каждой вершины существуют его обход, начинающийся в этой вершине. Так же, как в сведении для *SMWCS*, вводятся бинарные переменные r_v, в которых хранится начальная вершина обхода. Соответственно, вводится ограничение на уникальность такой вершины:

$$\sum_{v \in V} r_v = 1. \tag{1.3}$$

Кроме этого вместо графа $G = (V, E)$ рассматривается ориентированный граф $S = (V, A)$, полученный из G заменой каждого ребра $e = (v, u)$ на две направленные дуги $a = (v, u)$ и $b = (u, v)$. Описание обхода графа проводится в терминах графа S с помощью бинарных переменных x_a, значение которых определяется вхождением дуги a в дерево обхода. Корректность обхода гарантируется с помощью определения переменных d_v, в которых хранятся расстояния от стартовой вершины, и введения соответствующих ограничений:

$$1 \leqslant d_v \leqslant |V|, \qquad \forall v \in V; \tag{1.4}$$

$$\sum_{(u,v)\in A} x_{uv} + r_v = y_v, \qquad \forall v \in V; \tag{1.5}$$

$$x_{vu} + x_{uv} \leqslant w_e, \qquad \forall e = (v, u) \in E; \tag{1.6}$$

$$d_v r_v = r_v, \qquad \forall v \in V; \tag{1.7}$$

$$d_u x_{vu} = (d_v + 1) x_{vu}, \qquad \forall (v, u) \in A. \tag{1.8}$$

М. Хаоуари с соавторами в [73] показали, что система (1.1)–(1.8) корректно описывает задачу *GWMCS*.

Из-за нелинейности ограничений (1.7) и (1.8) эта система не является задачей целочисленного линейного программирования. Поэтому авторы воспользовались техникой линеаризации, предложенной в статьях [75, 76]. В формулировку были добавлены переменные t_{uv} для каждой дуги $a = (u, v)$ из A, а также переменные z_v и u_v для каждой вершины v. Переменные d_v для расстояний были исключены из формулировки. Сведение приняло следующий вид:

$$\sum_{e\in E} \omega(e) w_e + \sum_{v\in V} \omega(v) y_v \to max;$$

$$\sum_{v\in V} r_v = 1;$$

$$\sum_{(u,v)\in A} x_{uv} + r_v = y_v, \qquad \forall v \in V;$$

$$\sum_{(u,v)\in A} t_{uv} + u_v + \sum_{(u,v)\in A} x_{uv} = z_v, \qquad \forall v \in V;$$

$$2y_v - r_v \leqslant z_v, \qquad \forall v \in V;$$

$$z_v \leqslant ny_v - (|V|-1)r_v, \qquad \forall v \in V;$$

$$x_{uv} \leqslant t_{uv}, \qquad \forall (u,v) \in A;$$

$$t_{uv} \leqslant (|V|-1)x_{uv}, \qquad \forall (u,v) \in A;$$

$$x_{uv} + x_{vu} \leqslant w_e, \qquad \forall e = (v,u) \in E;$$

$$z_v - y_v + x_{vu} \geqslant t_{vu} + t_{uv}, \qquad \forall (u,v) \in A, v \in V;$$

$$z_v - ny_v + (|V|-1)x_{uv} + nx_{vu} \leqslant t_{uv} + t_{vu}, \qquad \forall (u,v) \in A, v \in V;$$

$$w_e \leqslant y_v, \qquad \forall v \in V, e \in \delta_v;$$

$$z_v \geqslant 0, \qquad \forall v \in V;$$

$$t_{uv} \geqslant 0, \qquad \forall (u,v) \in A, u \in V.$$

1.6. Задачи, решаемые в диссертационной работе

Из проведенного обзора следует, что в настоящее время не существует хорошо проработанных методов для работы с большими метаболическими моделями уровня клетки человека и использования их для интерпретации транскриптомных и метаболомных данных.

Даже для методов, основанных на анализе представленности, существуют разные подходы со своими достоинствами и недостатками. В методах простого анализа представленности остро стоит проблема выбора порога отсечки, а применение наиболее распространенного метода для беспорогового анализа *GSEA* [34] требует достаточно больших вычислительных затрат.

Для более современных методов поиска активного модуля существует только два основных подхода, имеющих хорошее статистическое обоснование: они предложены в [56] и [61]. Оба эти подхода сводятся в итоге к NP-трудным задачам и только для второго существует решатель, способный за разумные время решать до оптимальности реальные экземпляры задач. Применение

этого подхода к метаболическим моделям ограничивается одной работой [66], в которой рассматривается только относительно небольшая модель организма микроскопического беспозвоночного.

Таким образом, не существует *метода*, позволяющего одновременно работать с метаболическими моделями уровня клетки человека, имеющего обоснованную статистическую модель и способного работать с транскриптомными и метаболомными данными. Кроме этого, ни в одном из подобных подходов не используется информация об атомной структуре графа, которая может быть полезна для определения метаболических путей, как было показано, например, в работе [41].

Отметим также существование весьма ограниченного набора решателей для вариантов задач *MWCS*. Это требует либо использования только сведений к задачам *SMWCS* или *AMWCS*, либо разработки новых специализированных решателей.

На основании приведенного обзора сформулируем **цель** диссертационной работы: разработка и программная реализация набора эффективных вычислительных методов анализа метаболических моделей для идентификации регулируемых метаболических путей и их взаимосвязей по транскриптомным и метаболомным экспериментальным данным.

Задачами диссертационной работы являются:

1. Разработка и реализация эффективного метода идентификации регулируемых путей в метаболических моделях на основе анализа представленности, без использования информации о связях между реакциями.
2. Разработка и реализация эффективного метода идентификации регулируемых путей и их взаимосвязей в метаболических моделях на основе подхода поиска активного модуля.
3. Разработка и реализация эффективного метода идентификации регулируемых путей и их взаимосвязей в метаболических моделях на осно-

ве подхода поиска активного модуля с использованием информации об атомной структуре метаболитов.

Выводы по главе 1

1. Проведен обзор работ по темам «Полногеномные метаболические модели», «Анализа представленности», «Задача поиска активного модуля», «Подходы к решению задачи поиска связного подграфа максимального веса».
2. Для анализа метаболических моделей существуют несколько классов методов. С точки зрения анализа больших моделей наиболее актуальными являются методы, основанные на анализе предстваленности, и методы, основанные на задаче поиска активного модуля.
3. Для анализа представленности наиболее актуальными являются методы простого анализа и методы беспорогового анализа. Методы простого анализа обладают неотъемлимым недостатком необходимости выбора порога для выбора набора генов. С другой стороны, наиболее распространенный метод беспорогового анализа является медленным.
4. Для задачи поиска активного модуля существует два основных подхода. Оба этих подхода сводятся к NP-трудным задачам и только для одного из них, использующего сведение к задаче $MWCS$, есть практический решатель для соответствующей задачи.
5. Для задачи $MWCS$ существуют несколько формулировок. Для двух более узких формулировок, $SMWCS$ и $AMWCS$, существуют практические решатели, основанные либо на сведении к задаче $PCST$, либо на сведении к задаче целочисленного линейеного программирования. Для более общего варианта, $GMWCS$, отсутствует практический решатель, но существуют теоретические наработки.
6. На основе обзора сформулированы цель и задачи диссертационной работы.

ГЛАВА 2. МЕТОД БЫСТРОГО ВЗВЕШЕННОГО АНАЛИЗА ПРЕДСТАВЛЕННОСТИ НАБОРОВ ГЕНОВ

Настоящая глава посвящена задаче разработки и реализации эффективного метода идентификации регулируемых путей в метаболических моделях на основе анализа представленности, без использования информации о связях между реакциями.

Для решения этой задачи был разработан метод *FGSEA* (*Fast Gene Set Enrichment Analysis*) для беспорогового анализа представленности, основанный на идеях взвешенного варианта метода *GSEA* (раздел 1.3.2). В отличие от исходного метода, метод *FGSEA* позволяет выполнять тот же анализ на два порядка быстрее даже с использованием одного потока выполнения.

Разработанный метод заключается в том, чтобы вместо независимого подсчета фонового распределения для каждого данного размера набора генов, эти распределения строятся одновременно. Сначала идея метода описывается для статистики представленности простого вида, равной среднему значению весов генов в наборе. Затем метод применяется к *GSEA*-статистике, для которой был разработан специальный алгоритм кумулятивного вычисления. Глава завершается экспериментальной проверкой, показывающей, что результаты работы метода эквивалентны результатам исходного метода, при значительном ускорении.

2.1. Быстрый взвешенный анализ представленности для статистики среднего

Для начала рассмотрим идею быстрого взвешенного анализа представленности для статистики представленности s_m среднего веса генов в наборе p, определяемой по формуле:

$$s_m(p) = \frac{1}{|p|}\Sigma_{i \in p} S_i,$$

где S_i – вес i-го гена.

Идея алгоритма состоит в том, чтобы переиспользовать результаты сэмплирования для разных входных наборов генов. Такой подход является корректным, так как для оценки фонового распределения случайные наборы должны быть независимы только для каждого набора в отдельности. Таким образом, возможно иметь случайные наборы, зависимые *между* разными входными наборами.

Следовательно, вместо генерации nm независимых случайных наборов возможно генерировать только n случайных наборов размера K, где n – размер выборки для оценки фонового распределения, m – число рассматриваемых наборов генов, а K – максимальный размер набора генов. Из одного i-го случайного набора π_i размера K можно получить случайные наборы для всех размеров $k \leqslant K$, взяв его префиксы: $\pi_{i,k} = \pi_i[1..k]$.

Следующим шагом является вычисление значений статистики представленности для всех наборов $\pi_{i,j}$. Вместо того, чтобы вычислять статистику представленности независимо для всех наборов, будем вычислять значения одновременно для всех $\pi_{i,j}$ для фиксированного i. С помощью простой процедуры это можно сделать эффективно за время $O(K)$. Для этого достаточно поэлементно поделить частичные суммы рангов генов на длины соответствующих префиксов:

$$s_m(\pi_i[1..k]) = \frac{1}{k}\Sigma_{i \in \pi_i[1..k]} S_i.$$

Дополнительный проход за время $O(m)$ позволяет выбрать значения статистики для наборов из запроса.

Таким образом, с помощью представленного алгоритма возможно вычислить P-значения для всех наборов генов из запроса за время $O(n(K+m))$. Это соответствует ускорению примерно в $\min(K, m)$ раз по сравнению с простой реализаций независимым сэмплированием для такого же уровня точности.

2.2. Кумулятивное вычисление *GSEA*-статистики представленности

Идею использования кумулятивного вычисления статистики для ускорения вычисления P-значений можно применить и для *GSEA*-статистики. В этом разделе так же будут генерироваться случайнеы наборы генов размера K и для всех префиксов будет вычисляться значение статистики. Однако, для случая *GSEA*-статистики кумулятивное вычисления требует разработки отдельного алгоритма.

Для простоты, будем рассматривать только положительную моду *GSEA*-статистики представленности s_r^+. Она может быть вычислена как $s_r^+(p) = \mathrm{ES}_{i^+}$, где $i^+ = \arg\max_i \mathrm{ES}_i$. Значения ES_i определяются так же, как и в исходном методе *GSEA*:

$$\mathrm{ES}_i = \begin{cases} 0, & \text{если } i = 0, \\ \mathrm{ES}_{i-1} + \frac{1}{\mathrm{NS}}|S_i|, & \text{если } 1 \leqslant i \leqslant N \text{ и } i \in p, \\ \mathrm{ES}_{i-1} - \frac{1}{N-k}, & \text{если } 1 \leqslant i \leqslant N \text{ и } i \notin p, \end{cases}$$

где $\mathrm{NS} = \sum_{i \in p} |S_i|$, $N = |S|$. Вычисление отрицательной моды $s_r^-(p) = \mathrm{ES}_{i^-}$, где $i^- = \arg\min_i \mathrm{ES}_i$, может быть выполнено аналогичным образом. Из двух этих значений легко получить итоговое значение статистики $s_r(p)$, которое равно $s_r^+(p)$, если $|s_r^+(p)| > |s_r^-(p)|$ или $s_r^-(p)$ в противном случае.

2.2.1. Геометрическая интерпретация *GSEA*-статистики

Удобно рассмотреть *GSEA*-статистику с геометрической точки зрения. Рассмотрим $N + 1$ точку (рисунок 5) с координатами (x_i, y_i) для $0 \leqslant i \leqslant N$, определяемыми как:

$$(x_0, y_0) = (0, 0), \tag{2.1}$$

$$x_i = x_{i-1} + [i \notin p], \qquad \forall i \in 1..N, \tag{2.2}$$

$$y_i = y_{i-1} + [i \in P] \cdot |S_i|, \qquad \forall i \in 1..N. \tag{2.3}$$

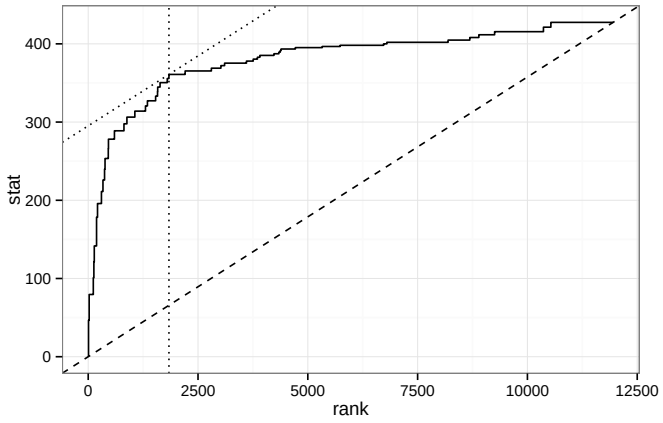


Рисунок 5 – График, соответствующий геометрическому представлению *GSEA*-статистики для некоторого набора генов. Значение статистики представленности соответствует точке на графике (пересечение пунктирных линий), наиболее удаленной от диагонали (линия с крупным пунктиром)

Вычисление s_r^+ соответствует поиску среди (x_i, y_i) точки, наиболее удаленной вверх (далее будем говорить просто наиболее удаленной) от диагонали $((x_0, y_0), (x_N, y_N))$. Действительно, заметим, что $x_N = N - |p| = N - k$ и $y_N = \Sigma_{j \in p}|S_j| = \mathrm{NS}$, а значения отдельных элементов ES_i может быть вычислено как $\mathrm{ES_i} = \frac{1}{NS} y_i - \frac{1}{N-k} x_i$. Их этой формулы следует, что значение $\mathrm{ES_i}$ прямо пропорционально направленному расстоянию от прямой, проходящей через (x_0, y_0) и (x_N, y_N), до точки (x_i, y_i).

Зафиксируем набор генов π размера K. Одним из возможных подходов к эффективному кумулятивному вычислению значений $s_r^+(\pi[1..k])$ для $k \leqslant K$ является разработка эффективного алгоритма обновления самой удаленной от диагонали точки при добавлении нового гена в набор. В таком случае возможно последовательно добавлять гены из π и вычислять значение $s_r^+(\pi[1..k])$ через соответствующее наибольшее расстояние.

Так как значения статистики вычисляются для $\pi[1..k]$ только для $k \leqslant K$, воспользуемся знанием, что в набор будет добавляться только K генов из π. Это позволяет рассматривать только $K + 1$ точку вместо $N + 1$ для каждой отдельной операции сэмплирования. Пусть массив o содержит порядок генов в π, то есть $\pi_{o_1} < \pi_{o_2} < \ldots < \pi_{o_K}$. Тогда координаты точек на итерации после

добавления k генов (x^k, y^k) могут быть вычислены по формуле:

$$(x_0^k, y_0^k) = (0, 0), \tag{2.4}$$

$$x_i^k = x_{i-1}^k + \pi_{o_i} - \pi_{o_{i-1}} - [o_i \leqslant k], \qquad \forall i \in 1..K, \tag{2.5}$$

$$y_i^k = y_{i-1}^k + [o_i \leqslant k] \cdot |S_i|, \qquad \forall i \in 1..K, \tag{2.6}$$

где для удобства примем значение π_{o_0} равным нулю.

Можно показать, что поиск точки самой удаленной точки среди задаваемых системой (2.4)–(2.6) эквивалентно поиску самой удаленной точкой среди задаваемых системой (2.1)–(2.3), при этом координаты точки (x_i^k, y_i^k) равны $(x_{\pi_{o_i}}, y_{\pi_{o_i}})$, вычисленным для $p = \pi[1..k]$.

Рассмотрим разность $x_{\pi_{o_i}} - x_{\pi_{o_{i-1}}}$. По определению x она равна:

$$x_{\pi_{o_i}} - x_{\pi_{o_{i-1}}} = \sum_{i=1}^{\pi_{o_i}} [i \notin \pi[1..k]] - \sum_{i=1}^{\pi_{o_{i-1}}} [i \notin \pi[1..k]] = \sum_{i=\pi_{o_{i-1}}+1}^{\pi_{o_i}} [i \notin \pi[1..k]] =$$

$$= \pi_{o_i} - \pi_{o_{i-1}} - \sum_{i=\pi_{o_{i-1}}+1}^{\pi_{o_i}} [i \in \pi[1..k]].$$

По определению o, в интервале $[\pi_{o_{i-1}} + 1, \pi_{o_i} - 1]$ нет генов из π и, следовательно, и из $\pi[1..k]$. Поэтому можно в сумме оставить только ее последний член:

$$x_{\pi_{o_i}} - x_{\pi_{o_{i-1}}} = \pi_{o_i} - \pi_{o_{i-1}} - [\pi_{o_i} \in \pi[1..k]] = \pi_{o_i} - \pi_{o_{i-1}} - [o_i \leqslant k].$$

Разница получилась такой же, как и в равенстве (2.5).

Теперь рассмотрим разницу $y_{\pi_{o_i}} - y_{\pi_{o_{i-1}}}$. По определению y она равна:

$$y_{\pi_{o_i}} - y_{\pi_{o_{i-1}}} = \sum_{i=1}^{\pi_{o_i}} [i \in \pi[1..k]] \cdot |S_i| - \sum_{i=1}^{\pi_{o_{i-1}}} [i \in \pi[1..k]] \cdot |S_i| = \sum_{i=\pi_{o_{i-1}}+1}^{\pi_{o_i}} [i \in \pi[1..k]] \cdot |S_i|.$$

Так же, как и раньше, в интервале $[\pi_{o_{i-1}} + 1..\pi_{o_i} - 1]$ нет генов из $\pi[1..k]$. Следовательно, сумму можно заменить на ее последнее слагаемое:

$$y_{\pi_{o_i}} - y_{\pi_{o_{i-1}}} = [\pi_{o_i} \in \pi[1..k]] \cdot |S_i| = [o_i \leqslant k] \cdot |S_i|.$$

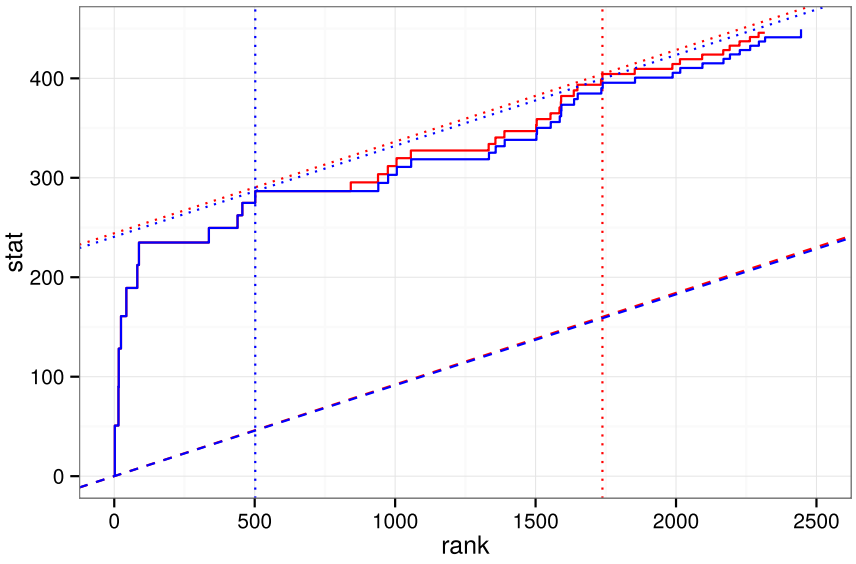


Рисунок 6 – Пример график, соответствующего *GSEA*-статистике до и после добавления гена $\pi_k \approx 800$. На рисунке показан только фрагмент. Синий график соответствует набору генов $\pi[1..k-1]$, красный – $\pi[1..k]$. Часть графика, расположенная слева от $x = x_{r_k}$, не изменяется, в то время, как часть справа от $x = x_{r_k}$ равномерно сдвигается вверх и влево. Диагональ $((x_0, y_0), (x_N, y_N))$ поворачивается против часовой стрелки

Получили разницу такую же, как и в равенстве (2.6).

Другие точки рассматривать не требуется, так как точки $o_{i-1}..o_i - 1$ имеют одну и ту же y координату и точка o_{i-1} является самой левой среди них. Следовательно, при добавлении хотя бы одного гена диагональ $((x_0, y_0), (x_N, y_N))$ будет не горизонтальной и точка o_{i-1} будет являться самой дальней среди точек $o_{i-1}..o_i - 1$.

2.2.2. Применение корневой оптимизации

Рассмотрим, что происходит, когда ген π_k добавляется к текущему набору генов $\pi[1..k-1]$ (рисунок 6). Пусть r_k – ранг гена π_k среди генов π. Тогда координаты точек с номерами $i < r_k$ не изменяются, а координаты всех точек $i \geqslant r_k$ сдвигаются на одинаковый вектор $(\Delta_x, \Delta_y) = (-1, |S_{\pi_k}|)$.

Для быстрого инкрементального обновления применим корневую оптимизацию [90] и разобьем задачу на несколько задач меньшего размера. Для

простоты будем считать, что $K+1$ является точным квадратом некоторого целого числа b. Разобьем $K+1$ точку на b последовательных блоков размера b: $\{(x_0^k, y_0^k), ..., (x_{b-1}^k, y_{b-1}^k)\}$, $\{(x_b^k, y_b^k), ..., (x_{2b-1}^k, y_{2b-1}^k)\}$ и т. д.

Для каждого из b блоков будем поддерживать индекс самой удаленной от диагонали точки. Если для каждого блока известна самая удаленная точка внутри блока, то глобально самую удаленную точку можно найти простым линейным проходом за время $O(b)$.

Теперь покажем, как можно обновлять индексы наиболее удаленных точек в блоках за амортизированное время $O(b)$. Такое обновление при совмещении с проходом за время $O(b)$ дает алгоритм для обновления глобально самой удаленной точки за амортизированное время $O(b)$.

Обозначим индекс блока, котором принадлежит гена π_k как $c = \lfloor r_k/b \rfloor$.

Во-первых, рассмотрим процедуру, которая позволяет эффективно обновлять координаты точек. Для хранения x-координат точек будем использовать два массива: B размера b и D размера $K+1$, чтобы значение x-координаты i-й точки могло быть вычислено как $x_i = B_{i/b} + D_i$. При добавлении гена π_k все координаты x_i для $i \geqslant r_k$ уменьшаются на единицу. Чтобы отразить эти изменения, уменьшим на единицу значения B_j для всех $j > c$ и значения D_I для всех i, $r_k \leqslant i < cb$. В этой процедуре затрагиваются только $O(b)$ элементов и, таким образом, время обновления координат тоже составляет $O(b)$. После такого обновления процедура получения значения x_i занимает время $O(1)$. Аналогичная процедура может быть применена и для значения y-координат, только с увеличением значений на $|S_{\pi_k}|$.

Во-вторых, для каждого блока будем поддерживать верхнюю часть выпуклой оболочки его точек. Знание выпуклой оболочки удобно, потому что самая удаленная точка всегда лежит на ней. Во всех блоках кроме блока c координаты точек либо не изменяются, либо сдвигаются на одинаковый вектор. Это означает, что для этих блоков множество и последовательность точек на их выпуклых оболочках не изменяются. Для блока c можно построить вы-

пуклую оболочку заново, используя алгоритм Грэхема [5]. Так как точки уже отсортированы по x-координате, шаг сортировки можно опустить, и построить оболочку за время $O(b)$. Итого, обновление выпуклых оболочек требует времени $O(b)$.

В-третьих, индексы самых удаленных точек в блоках можно быстро обновлять пользуясь знанием выпуклых оболочек. Сначала рассмотрим блок, в котором выпуклая оболочка не изменилась: любой блок, кроме, возможно, блока c. Так как диагональ, от которой рассчитывается расстояние, может вращаться только в направлении против часовой стрелки, то при переходе от итерации $k-1$ к итерации k самая удаленная точка либо не изменяется, либо сдвигается влево по выпуклой оболочке. Следовательно, для каждого блока можно последовательно сдвигать индекс влево по выпуклой оболочке, до тех пор, пока это увеличивает расстояние от диагонали. Для блока c, если его выпуклая оболочка изменилась, самую удаленную его точку можно найти с помощью отдельного прохода по всем точкам обновленной выпуклой оболочки.

Воспользовавшись методом потенциалов, покажем, что амортизированное время на обновление самой удаленной точки составляет $O(b)$. Пусть потенциал после добавления k-го гена Φ_k будет равен сумме относительных индексов самых удаленных точек в каждом блоке. Так как имеется b блоков размера b, то сумма всех относительных индексов будет лежать от 0 до b^2. Следовательно, $\Phi_k = O(b^2)$. Для обновления индексов самых удаленных точек во всех $b-1$ блоках, кроме c, алгоритм на k-й итерации суммарно совершает $t_k = b - 1 + z$ операций, где z – суммарное число совершенных сдвигов. Обновление блока c требует $O(b)$ операций. В худшем случае суммарное время работы может составить $\Theta(b^2)$, если, например, в каждом блоке индексы изменится на $b/2$. Однако, можно заметить, что изменение потенциала $\Phi_k - \Phi_{k-1}$ составляет $-z + O(b)$: в $b-1$ блоках сумма относительных индексов уменьшается на z, а в блоке c относитель-

ный индекс может увеличиться, но не больше чем на b. В итоге получается, что амортизированная оценка на время добавления k-го гена составляет $a_k = t_k + \Phi_k - \Phi_{k-1} = b - 1 + z - z + O(b) = O(b)$. Общее реальное время работы K итераций составляет $\sum_{k=1}^{K} a_k + \Phi_0 - \Phi_K = O(Kb) + O(b^2) = O(Kb)$.

Суммарно, предложенный алгоритм позволяет кумулятивно вычислить значения *GSEA*-статистики представленности $s_r(\pi[1..k])$ за время $O(Kb) = O(K\sqrt{K})$. Простая же реализация с независимым вычислением статистики требует $O(K^2 \log K)$ операций. Таким образом, производительность увеличивается в $O(K\frac{\log K}{\sqrt{K}})$ раз.

2.2.3. Оптимизации

Для оптимизации алгоритм может быть немного изменен: вместо вычисления выпуклой оболочки c с самого начала, можно обновлять только измененные точки. Это изменение не влияет на асимптотику алгоритма, но позволяет увеличить производительность на практике.

Во-первых, будем начинать обновление оболочки в блоке c не с первой точки, а с точки r_k. Для того, чтобы иметь такую возможность, будем хранить массив `prev`, который для каждого гена $g \in \pi$ хранит предыдущую точку на выпуклой оболочке, если бы ген g был бы последним в блоке. Эквивалентно, этот массив можно рассматривать, как содержащий для каждого гена g вершину стека из алгоритма Грэхема в момент перед добавлением g в стек, что соответствует сохранению состояния прохода Грэхема. Так как относительные координаты всех точек слева от g не изменяются, то для любой такой точки h значение $\texttt{prev}_h$ так же не изменяется и не требует обновления.

Во-вторых, будем заканчивать обновление выпуклой оболочки, когда проход Грэхема достигает точки, лежащей на выпуклой оболочке на предыдущей итерации. Это можно делать, так как все точки слева от g вращаются против часовой стрелки относительно любой точки справа от g. Это означает,

что начиная с первой точки справа от g, лежащей на выпуклой оболочке на $(k-1)$-й итерации, выпуклая оболочка не изменяется на k-й итерации.

2.2.4. Детали реализации

Метод был реализован на языках *R* и *C++* и скомпонован виде *R*-пакета *fgsea*. На *C++* выполнена реализация алгоритм кумулятивного вычисления *GSEA*-статистики. Пакет был принят в библиотеку *R/Bioconductor* и доступен по адресу http://bioconductor.org/packages/fgsea. Пакет распространяется под свободной лицензией *MIT*.

Отметим небольшое отличие от референсной реализации метода *GSEA* в способе вычисления *P*-значений. Вместо использования формулы

$$\frac{\sum \pi[s_r(\pi) \geqslant s_r(p)]}{\sum \pi[s_r(\pi) > 0]}$$

для наборов p с положительным значением статистики $s_r(p) > 0$, применяется формула

$$\frac{\sum \pi[s_r(\pi) \geqslant s_r(p)] + 1}{\sum \pi[s_r(\pi) > 0] + 1}.$$

Аналогично для наборов p с отрицательным значением статистики. Такой подход позволяет лучше оценить настоящие *P*-значения [77].

Для того, чтобы можно было делать большое число итераций сэмплирования, полная таблица значений $s_r(\pi[1..k])$ не хранится. Вместо этого после очередной итерации сэмплирования выполняется обновление значений $\sum_\pi[s_r(\pi) \geqslant s_r(p_i)]$, $\sum_\pi[s_r(\pi) > 0]$, $\sum_\pi[s_r(\pi) \leqslant s_r(p_i)]$, $\sum_\pi[s_r(\pi) < 0]$, а также еще несколько подобных значений для вычисления значений *GSEA*-статистики, нормализованных на среднее значение среди случайных наборов. Эти значения могут быть быстро обновлены и из них легко вычисляются соответствующие *P*-значения.

2.3. Экспериментальное исследование

Проведенное экспериментальное исследование состояло из двух частей. Во-первых, была проанализирована производительность реализации алгорит-

ма кумулятивного вычисления *GSEA*-статистики. Во-вторых, было проведено сравнение результатов работы предложенного метода и референсной реализации [34].

Веса генов в экспериментальном исследовании были получены с использованием набора данных *GSE14308* [78] из базы *GEO Omnibus*. Для этого был проведен анализ дифференциальной экспрессии для состояний *Naive* и *Th1* процесса дифференциации мышиных Т-клеток. Анализ выполнялся с помощью программного пакета *limma* [15]. В качестве веса гена использовалось финальное значение t-статистики (после выполнения Байесовской поправки).

Эксперименты проводились на компьютере с шестиядерным процессором *AMD Phenom II X6 1090T* с тактовой частотой 3,2 ГГц.

2.3.1. Анализ производительности кумулятивного вычисления *GSEA*-статистики

В этом разделе была проведена экспериментальная проверка, что производительность практической реализации алгоритма кумулятивного вычисления *GSEA*-статистики соответствует теоретической оценке. Напомним, что ранее было показано, что разработанный алгоритм работает за время $O(K\sqrt{K})$.

Сначала была проанализирована зависимость от K. Вычисление запускалось для значений K от 25 до 2000. Из 20770 измеренных в соответствующем эксперименте генов рассматривались только 12000 с самой высокой средней экспрессией. В качестве наборов генов выбирались первые K генов из некоторой случайной перестановки всех рассматриваемых генов. Так как время одного кумулятивного вычисления достаточно маленькое, рассматривалось время, необходимое для выполнения 1000 запусков. Для того, чтобы уменьшить влияние сложно предсказуемых факторов, как сборка мусора, запуски разбивались на 10 групп по 100 и выполнялись в случайном порядке для всех рассматриваемых значений K.

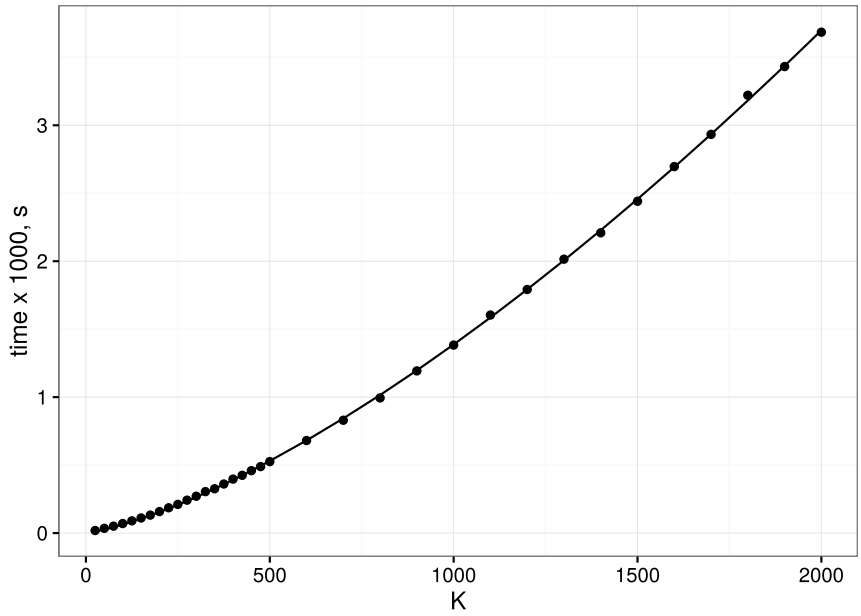


Рисунок 7 – Зависимость времени кумулятивного вычисления *GSEA*-статистики от размера набора

Результаты этого анализа представлены на рисунке 7. На рисунке каждая точка соответствует времени, затраченному на 1000 операций кумулятивного вычисления статистики для заданного значения K. Время работы хорошо согласуется с теоретической оценкой $O(K\sqrt{K})$. На графике приведена кривая, полученная линейной регрессией зависимости времени работы от K и $K\sqrt{K}$, коэффициент детерминации для которой равен 99,995%.

Также было проверено, что время работы практически не зависит от общего числа N рассматриваемых генов и от формы распределения весов (таблица 1). Были выполнены запуски для значений N, равных 5000, 10000, 15000 и 20000. Программа запускалась как для исходного распределения весов генов (`param` $= 1$), так и для весов, равных единице (`param` $= 0$), но при сохранении порядка генов. Запуск выполнялся для значения $K = 500$, измерялось время 1000 запусков.

2.3.2. Сравнение с референсной реализацией

Было проведено сравнение разработанного метода с рефернсной реализацией (версия 2.2.2.) [87]. Далее в этом разделе реализация алгоритма, пред-

Таблица 1 – Зависимость времени кумулятивного вычисления *GSEA*-статистики от числа рассматриваемых генов (N) и от формы распределения весов ($param$). Время указано для 1000 запусков для наборов генов размера $K = 500$

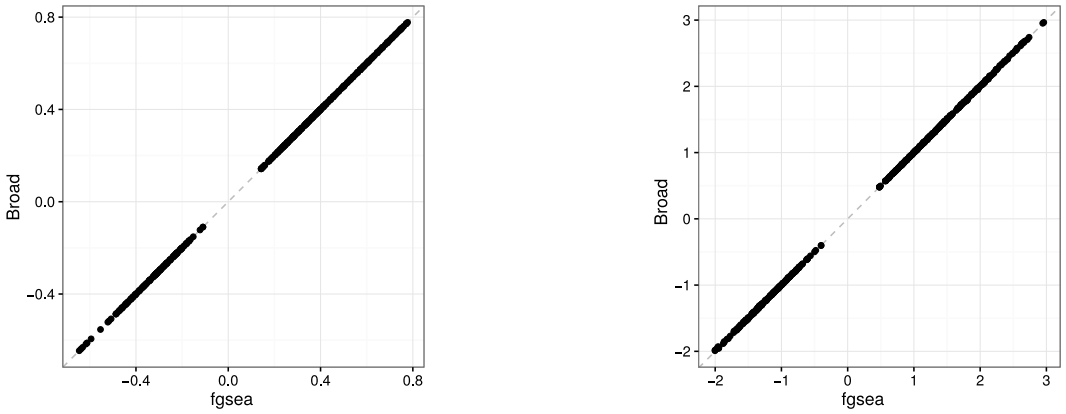
N	param	Время $\times 1000$, с
5000	0	0.505
5000	1	0.509
10000	0	0.513
10000	1	0.527
15000	0	0.516
15000	1	0.529
20000	0	0.515
20000	1	0.518

ложенного в настоящей работе, будет обозначаться как *fgsea*, а референсная реализация – как *Broad*.

В качестве входных данных так же, как и раньше, рассматривались гены и соответствующие значения t-статистики из набора данных *GSE14308*. Использовались только 12000 генов с максимальной средней экспрессией.

Для получения набора мышиных молекулярных путей использовалась база данных *Reactome* [79]. В соответствии со значениями по умолчанию в методе *GSEA*, рассматривались только наборы генов размера от 15 до 500. Таких наборов было 586.

Обе реализации были запущены с генерацией 1000 и 10000 случайных наборов для оценки фонового распределения. Время работы измерялось с учетом времени считывания входных данных из файлов. Время работы реализации *Broad* составило 96 с для 1000 наборов и 1048 с для 10000 наборов. Для реализации *fgsea* время работы составило 0,8 с для 1000 наборов и 5,5 с – для 10000. Таким образом, для 1000 наборов ускорение составило 120 раз, а для 10000 – 190 раз.



(а) (б)

Рисунок 8 – Значения ES (а)и NES (б), вычисленные двумя методами. Одна точка соответствует одному молекулярному пути

Для проверки корректности было выполнено сравнение вычисленных значение статистик *ES* и их нормализованных версий *NES* для обеих реализаций (рисунок 8). Это сравнение показало, что как прямые значение статистики *ES*, так и нормализованные значения *NES*, хорошо согласуются друг с другом. Максимальная разница в значениях *ES* составила примерно $10^{-4}\%$. Эта небольшая ошибка предположительно возникает из-за неточностей операций с вещественными числами. Различие в значениях *NES* не превышает 2%. Эта ошибка больше, чем для значений *ES*, так как в отличие от них, вычисление значение *NES* является недетерминированным и зависит от сгенерированных наборов.

Также было выполнено сравнение номинальных P-значений, без поправки на множественное сравнение, полученных для 10000 перестановок (рисунок 9). В результатах реализации *Broad* нулевые значения заменялись на 10^{-4}. Полученные P-значения достаточно хорошо согласуются для обоих методов (коэффициент корреляции 99,98%). При этом увеличение относительного разброса при уменьшении значений следует из свойств биномиального распределения, которое возникает из того, что эмпирическое P-значение является случайной величиной, имеющей биномиальное распределение.

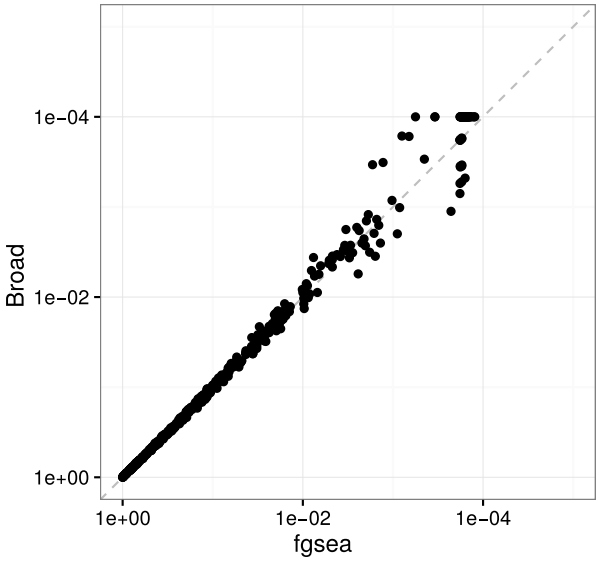


Рисунок 9 – Номинальные P-значения, вычисленные методами *fgsea* и *Broad*

Важным моментом является возможность использования совместно с разработанным методом стандартных методов поправки на множественное сравнение: например, Бенджамини-Хохберга [46] или Бонферрони [80]. При генерации только 1000 случайных наборов применение этих методов является непрактичным из-за низкой точности вычисления P-значений. Например, для рассматриваемого примера после поправки методом Бенджамини-Хохберга в этом случае не остается ни одного набора генов с поправленным P-значением, меньшим одного процента. В то же время при генерации 10000 наборов было найдено 78 таких входных наборов. Из этих путей три – регулируются вниз, в то время как при использовании нестандартного метода поправки, включенного в реализацию *Broad*, таких значимых регулируемых вниз наборов не находится.

Кроме этого, были выполнены запуски разработанной программы с генерацией ста тысяч, одного миллиона и десяти миллионов случайных наборов. В отличие от предыдущих запусков, разрешалось использовать шесть потоков выполнения. Запуски заняли, соответственно, 13, 98 и 941 секунд. При ста тысячах перестановок по сравнению с десятью тысячами из-за более точных P-значений для одного входного набора генов номинальное P-значение

увеличилось с $1{,}3 \cdot 10^{-3}$ до $2{,}0 \cdot 10^{-3}$, и он перестал подходить под заданный порог значимости. Для другого набора P-значение уменьшилось с $1{,}4 \cdot 10^{-3}$ до $0{,}9 \cdot 10^{-3}$, и он начал подходить под порог значимости. Для еще одного набора в этих трех запусках P-значение колеблется в интервале от $1{,}3 \cdot 10^{-3}$ до $1{,}4 \cdot 10^{-3}$, а скорректированное значение колеблется около порога 0,01. В остальном результаты качественно не изменяются.

2.4. Пример применения метода на данных активации Т-клеток

Рассмотрим применение метода *FGSEA* для анализа процесса дифференциации «наивных» Т-клеток (*Th0*) в *Th1*-клетки, участвующие в развитии клеточного иммунного ответа. Целью анализа является выявление того, какие из стандартных метаболических путей регулируются в этом процессе. Соответствующие исходные транскриптомные данные были взяты из [78]. Из них с помощью анализа дифференциальной экспрессии (раздел 1.1.3) была получена таблица с индивидуальной регуляцией каждого гена. Описание стандартных метаболических путей взято из базы *KEGG MODULE*, таких путей было 92.

Для применения метода каждому гену сопоставим вес, отражающий степень и направление его *индивидуальной* регуляции. Большой положительный вес означает, что экспрессия гена значительно увеличивается при дифференциации, а большой по модулю отрицательный вес означает значительное уменьшение экспрессии. В качестве такого веса берется t-статистика из теста на дифференциальную экспрессию.

После этого метод вычисляет для каждого набора генов, соответствующего стандартному метаболическому пути, значение *GSEA*-статистики. Оно отражает степень и направление *групповой* регуляции набора генов. Так же, как и для индивидуальных весов, большое положительное значение означает значительное увеличение экспрессии большинства генов в наборе, большое отрицательное – значительное уменьшение.

Таблица 2 – Пример результата работы метода *FGSEA* на данных дифференциации Т-клеток. Представлены только результаты с P-значением после поправки на множественное сравнение, меньшим 0,01. Поправка выполнена методом Бенджамини-Хохберга

Метаболический путь	P-значение	поправленное P-значение	значение *GSEA*-статистики
Glycolysis (Embden-Meyerhof pathway), glucose => pyruvate	0,0000177	0,0004113	0,8127154
Gluconeogenesis, oxaloacetate => fructose-6P	0,0000183	0,0004113	0,8186035
Glycolysis, core module involving three-carbon compounds	0,0000375	0,0005627	0,8404853
Adenine ribonucleotide biosynthesis, IMP => ADP,ATP	0,0002925	0,0030387	0,7522286
C5 isoprenoid biosynthesis, mevalonate pathway	0,0003376	0,0030387	0,8121245

Для оценки значимости полученных значений *GSEA*-статистики, метод выполняет сравнение со значениями этой статистики, вычисленными для большого числа случайных наборов генов. В рассматриваемом примере было сгенерировано сто тысяч случайных наборов. По этим данным выполняется вычисление P-значений: вероятности случайного набора иметь такое большое положительное (или отрицательное) значение *GSEA*-статистики.

Результаты работы метода представлены в таблице 2. В ней присутствуют только значимо регулируемые метаболические пути. Их пять:

— расщепление глюкозы (*Glycolysis*);

— синтез глюкозы (*Gluconeogenesis*);

— ядро метаболического пути расщепления глюкозы, включающее только вещества с тремя атомами углерода (*Glycolysis, core module*);
— синтез аденозинтрифосфата (АТФ, *Adenine ribonucleotide biosynthesis*);
— синтез мевалоната, являющийся частью процесса синтеза холестерина (*C5 isoprenoid biosynthesis*).

Таким образом, с точки зрения интерпретации, с помощью метода *FGSEA* можно выделить набор важных метаболических процессов. Это позволяет биологу вместо работы с отдельными генами и реакциями работать с метаболическими процессами. Это является более высокоуровневым представлением процессов, происходящих в клетке. После этого биолог может строить гипотезы о том, зачем эти процессы регулируются, можно ли как-то через них повлиять на клетку и т. д.

Выводы по главе 2

1. Предложен метод *FGSEA* для беспорогового анализа представленности, основанный на идеях взвешенного варианта метода *GSEA*.
2. Разработан эффективный алгоритм кумулятивного вычисления *GSEA*-статистики.
3. Проведены экспериментальные исследования, подтверждающие высокую скорость работы по сравнению с методом *GSEA* при сохранении точности.
4. Разработанный метод реализован в виде программного пакета для языка *R* и доступен в библиотеке *R/Bioconductor* под свободной лицензией (http://bioconductor.org/packages/fgsea).

ГЛАВА 3. МЕТОД ПОИСКА АКТИВНОГО МЕТАБОЛИЧЕСКОГО МОДУЛЯ С ПОМОЩЬЮ АНАЛИЗА СЕТИ МЕТАБОЛИЧЕСКИХ РЕАКЦИЙ

Настоящая глава посвящена задаче разработки и реализации эффективного метода идентификации регулируемых путей и их взаимосвязей в метаболических моделях на основе подхода поиска активного модуля.

Для решения этой задачи был разработан метод *GAM* (*Genes And Metabolites*) для выделения активного модуля в сети метаболических реакций с помощью сведения к задаче *GMWCS*. Метод позволяет работать как с транскриптомными, так и метаболомными данными. Кроме этого, был разработан решатель для задачи *GMWCS*, позволяющий для большинства реальных экземпляров находить точное оптимальное решение за разумное время. Был разработан веб-сервис *Shiny GAM* (http://genome.ifmo.ru/shiny/gam) с возможностью воспользоваться разработанным методом без необходимости сложной настройки.

3.1. Общая схема предлагаемого метода

Схема предлагаемого метода представлена на рисунке 10. Сначала из базы данных *KEGG* выделяются реакции, потенциально возможные в выбранном организме (рисунок $10A$). Затем, при наличии транскриптомных данных, удаляются реакции без экспрессированных генов (рисунок $10B$). Далее набор реакций представляется в виде графа (рисунок $10C$). На основе результатов анализа дифференциальной экспрессии для генов и метаболитов вершинам и ребрам графа приписываются веса: положительные важным элемента, отрицательные – не важным (рисунок $10D$). В полученном графе выделяется активный модуль с помощью решения задачи *GMWCS* (рисунок $10E$). Дополнительные процедуры постобработки приводят к финальному модулю (рисунок $10F$).

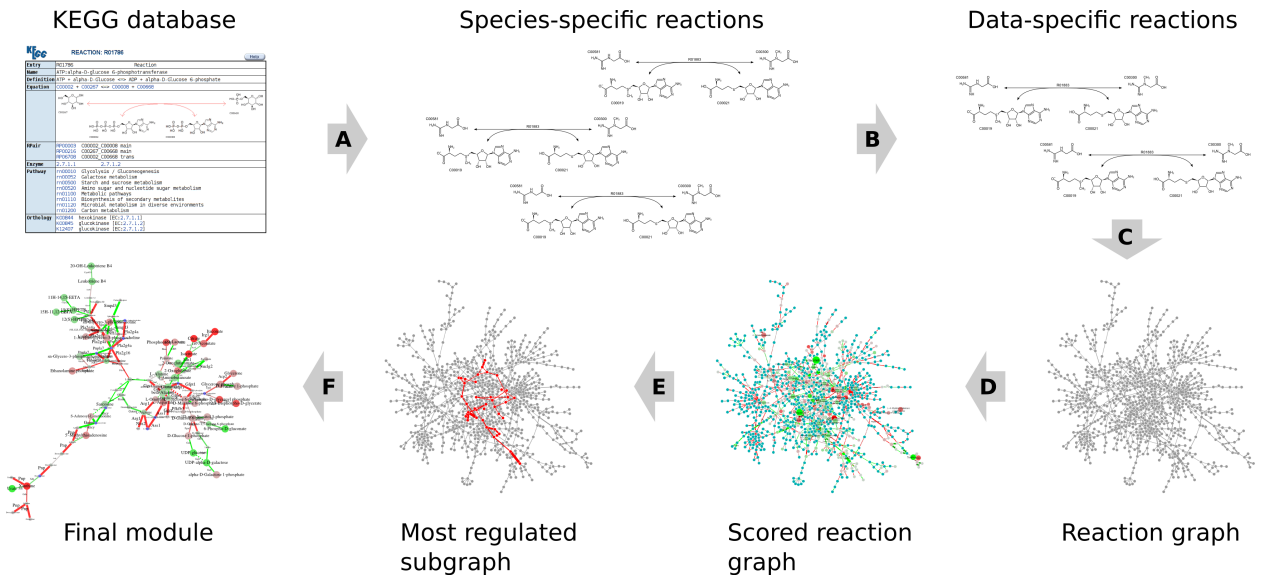


Рисунок 10 – Схема предлагаемого метода сетевого интегративного анализа

3.2. Сведение задачи поиска активного модуля к задаче *GWMCS*

Рассмотрим предлагаемый способ формулировки задачи поиска активного модуля в виде экземпляра задачи *GMWCS*.

3.2.1. Входные данные

На вход предлагаемый метод принимает таблицы с результатами дифференциальной экспрессии для транскриптомных и метаболомных данных. Возможно использование как одновременно обоих типов данных, так и только одного из них. Необходимыми в таблице являются три поля:

— идентификатор гена или метаболита;

— P-значение теста дифференциальной экспрессии;

— направление и величина изменения экспрессии.

Предполагается, что P-значения внутри одного набора данных подчиняются бета-равномерному распределению (раздел 3.2.4). Также предполагается, что транскриптомные данные представлены для всех генов, активных в данном эксперименте (раздел 3.2.2). Для некоторых метаболитов данные могут отсутствовать.

При наличии транскриптомных данных, создается таблица результатов дифференциальной экспрессии для реакций. С каждой реакцией может быть

ассоциировано несколько генов: например, катализатором реакций может быть белковый комплекс, состоящий из нескольких генов или несколько разных генов могут обладать одной и той же ферментативной функцией. Из нескольких таких генов реакции ставится в соответствие только один, с минимальным P-значением. Это соответствует тому, что реакция может регулироваться даже за счет изменения только одного гена. Дальше будем говорить, что этот ген *регулирует* эту реакцию.

3.2.2. Построение сети реакций по входным данным

Для построения сети реакций используется база данных *KEGG* [18].

В анализе участвуют только те реакции, которые потенциально могут проходить в рассматриваемом типе клеток. Во-первых, выбираются только те реакции, для которых существуют ферменты в заданном организме. Во-вторых, при наличии транскриптомных данных, предполагаются, что они содержат все экспрессирующиеся гены, соответственно, все реакции без экспрессированных ферментов удаляются. Рекомендуется подавать на вход 10–15 тысяч генов самых экспрессированных генов в эксперименте. Типично, после этой процедуры для организма человека или мыши получается набор из 1000–1500 реакций.

Кроме этого, уменьшатся дублирование информации путем удаления отдельных шагов многоступенчатых реакций. Например, реакция *R00267* превращения изоцитрата в оксоглутарат представлена двумя одноступенчатыми реакциями: *R01899* – превращения изоцитрата в оксалосукцинат, и *R00268* – превращения оксалосукцината в оксоглутарат. Из этих трех реакций остается только *R00267*. Определение того, что реакция является шагом последовательной реакции выполняется с помощью поиска регулярного выражения `"of \w*-step"` в записи реакции в базе *KEGG*.

Также, для улучшения качества анализа, удаляются неспецифичные метаболиты, создающие чрезмерную связность, и некоторые группы метабо-

литов объединяются. К неспецифичным метаболитам были отнесены: простые неорганические вещества (вода, аммиак и т. д.), моно-, ди- и трифосфаты (АТФ, УМФ и т. д.), некоторые метаболиты общего вида (*ROH*, *Acceptor* и т. д.). Объединялись метаболиты, являющиеся аномерами, например, альфа- и бета- *D*-глюкоза. Определение таких метаболитов выполнялось по наличию префикса `alpha-` и `beta-` в их названии.

3.2.3. Представление сети в виде графа

Из-за существования мультимолекулярных реакций, в которых участвует более одного метаболита с одной из сторон, невозможно тривиальным образом представить набор возможных реакций в виде простого графа. В диссертации рассматриваются несколько вариантов возможного представления, принципиально разделяемых на два класса: в одном, *вершинном*, и реакции, и метаболиты отображаются на вершины, в другом, *вершинно-реберном*, метаболиты отображаются на вершины, а реакции на ребра (рисунок 11). Бимолекулярная реакция *R01883* (рисунок 11*A*) может быть представлена как четыре вершины-метаболита, соединенные четырьмя попарными ребрами (рисунок 11*B*) или только двумя «основными» (рисунок 11*C*). Основными считаются те ребра, которые соединяют пару метаболитов со значением поля `RPTYPE` в базе *KEGG*, равным `main`. При вершинном представлении вершина, соответствующая реакции, будет соединена со всеми метаболитами (рисунок 11*D*).

Кроме этого, группы расположенных рядом вершин для реакций, регулируемых одним и тем же геном, могут быть объединены в одну вершину (рисунки 11*E* и 11*F*). Это позволяет уменьшить систематические ошибки из-за представленности одного и того же гена несколько раз в небольшой окрестности. Для того, чтобы провести объединение, строится вспомогательный граф, в котором вершинами являются реакции, а ребро соединяет две реакции, если у них есть общий метаболит и с обеими реакциями ассоцииро-

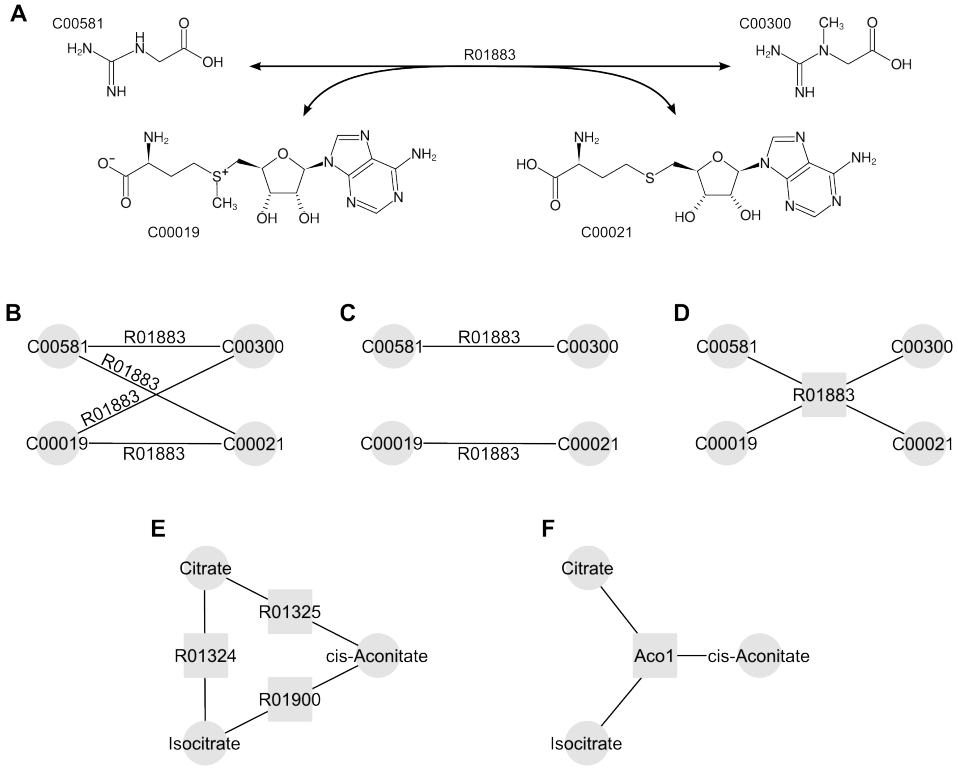


Рисунок 11 – Способы представить реакцию в виде фрагмента графа

ван один и тот же ген. Затем в этом графе выделяются компоненты связности, соответствующие тому, какие реакции надо объединить.

При использовании вершинно-реберного представления из нескольких параллельных ребер в графе оставляется только одно, с минимальным P-значением.

Разные варианты представления сети реакций оказываются предпочтительны в разных ситуациях в зависимости от присутствия данных по генам или по метаболитам. В случае наличия только транскриптомных данных рекомендуется выбирать вершинное представление реакций. При наличии же метаболомных данных рекомендуется выбирать вершинно-реберное представление.

3.2.4. Назначение весов

Схема назначения весов была адаптирована из [61]. Она она основана на предположении, что P-значения подчиняются смеси бета-распределения $B(\alpha, 1)$ и равномерного распределения $\mathcal{U}(0, 1)$ (раздел 1.4.2). С помощью, на-

пример, алгоритма из [61] можно определить параметры этой смеси: параметр α бета-распределения и параметр λ, определяющий в каком соотношении в смеси участвуют эти два распределения. Параметры определяются независимо для присутствующих в сети генов и метаболитов. Далее по выбранному пользователем желаемого порога на уровень FDR вычисляется вес генов (раздел 1.4.2).

Вес метаболитов определяется по той же процедуре независимо от назначения весов генов. Отличием является то, что для некоторых метаболитов могут отсутствовать в исходных метаболомных данные. Вес, назначаемый таким метаболитам, является свободным параметром. По умолчанию им назначается вес, соответствующий единичному P-значению.

В случае полного отсутствия одного типа данных, соответствующим элементам присваивается нулевой вес.

Когда несколько вершин соответствуют одному гену (или метаболиту) и имеют одинаковый положительный вес, то вес каждой из этих вершин делится на их число. Эта процедура позволяет избежать ситуации, когда один и тот же ген катализирует несколько достаточно близких реакций, но, когда даже после процедуры объединения вершин (раздел 3.2.3), он все еще представлен в графе несколько раз.

После назначения весов возникает экземпляр обобщенной задачи поиска связного подграфа максимального веса – $GMWCS$ (раздел 1.5.1). В случае, когда используется вершинное представление сети реакций, либо когда используется вершинно-реберное, но при отсутствии транскриптомных данных, веса ребер в графе равны нулю, и можно рассматривать простой вариант задачи поиска связного подграфа максимального веса – $SMWCS$ (раздел 1.5.1).

3.2.5. Постобработка

Были разработаны несколько процедур постобработки, упрощающей восприятие результатов человеком.

Во-первых, при использовании вершинного представления и отсутствии одного типа данных убираются необязательные вершины с нулевым весом. Для этого решается еще один экземпляр задачи *GMWCS* для графа, полученного из найденного модуля путем замены нулевых весов вершин на небольшое отрицательное число. В результате получается модуль, обладающий таким же весом, но минимальным числом вершин.

Во-вторых, так же при использовании вершинного представления, имеется возможность добавить метаболиты, являющиеся общими для хотя бы двух реакций, присутствующих в модуле. Хотя такие метаболиты не являются обязательными для связности модуля и не имеют положительного веса, их отображение может помочь в анализе внутренних взаимосвязей в модуле.

В-третьих, при вершинно-реберном представлении с использованием только основных пар субстрат-продукт, можно добавить ребра, соединяющие вершины модуля, но имеющие другой тип. Как и в случае с добавлением вершин, это помогает упросить понимание внутренних взаимосвязей.

3.3. Решатель обобщенной задачи поиска связного подграфа максимального веса

Для задачи *GMWCS* был разработан точный решатель с помощью сведения к задаче целочисленного линейного программирования и последующего решения с помощью библиотеки *IBM ILOG CPLEX* [91]. Решатель включает в себя несколько компонент:

— правила предобработки, позволяющие упростить задачу;
— метод декомпозиции, позволяющий разбить задачу на более мелкие, из решений которых можно восстановить решение исходной задачи;
— формулировка ограничений на связность подграфа в виде набора линейных ограничений.

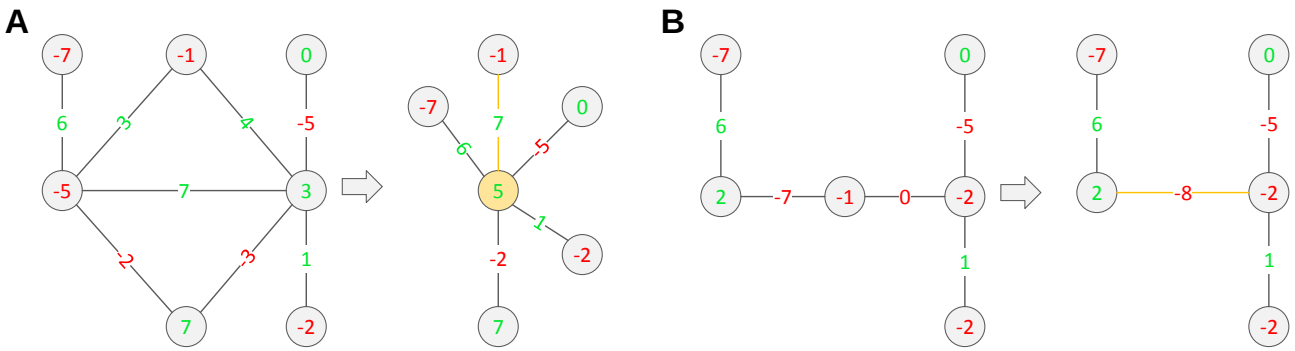


Рисунок 12 – Правила предобработки экземпляра задачи *GMWCS*

Решатель был реализован на языке *Java*. Исходный код доступен в открытом доступе под лицензией *MIT* по адресу https://github.com/ctlab/gmwcs-solver/.

3.3.1. Правила предобработки

Два правила предобработки были адаптированы из [72] для использования с реберно-взвешенными графами.

Первое правило объединяет группы вершин, которые одновременно либо присутствуют, либо отсутствуют в решении (рисунок 12*A*). Пусть $e = (u, v)$ – ребро с весом $\omega(e) \geqslant 0$, и одновременно $\omega(e) + \omega(v) \geqslant 0$ и $\omega(e) + \omega(u) \geqslant 0$. В этом случае, если хотя бы одна из вершин присутствует в решении, то ребро и другая вершина могут быть добавлены в решение без уменьшения суммарного веса. Поэтому, можно стянуть ребро e в одну вершину w с весом $\omega(w) = \omega(e) + \omega(u) + \omega(v)$. После данной процедуры могут возникнуть параллельные ребра. Пусть они возникли между вершинами w и t. В этом случае все неотрицательные ребра объединяются в одно с весом, равным сумме весов этих ребер. Затем все ребра между w и t, кроме одного с максимальным весом, удаляются. Правило применяется до тех пор, пока граф изменяется.

Второе правило похоже на первое, но объединяет цепочки ребер (рисунок 12*B*). Пусть v – вершина с ровно двумя инцидентными ей ребрами $e_1 = (u, v)$ и $e_2 = (v, w)$. Если все три веса $\omega(v)$, $\omega(e_1)$ и $\omega(e_2)$ неположительны, тогда v, e_1 и e_2 могут быть заменены на одно ребро $e = (u, w)$ с

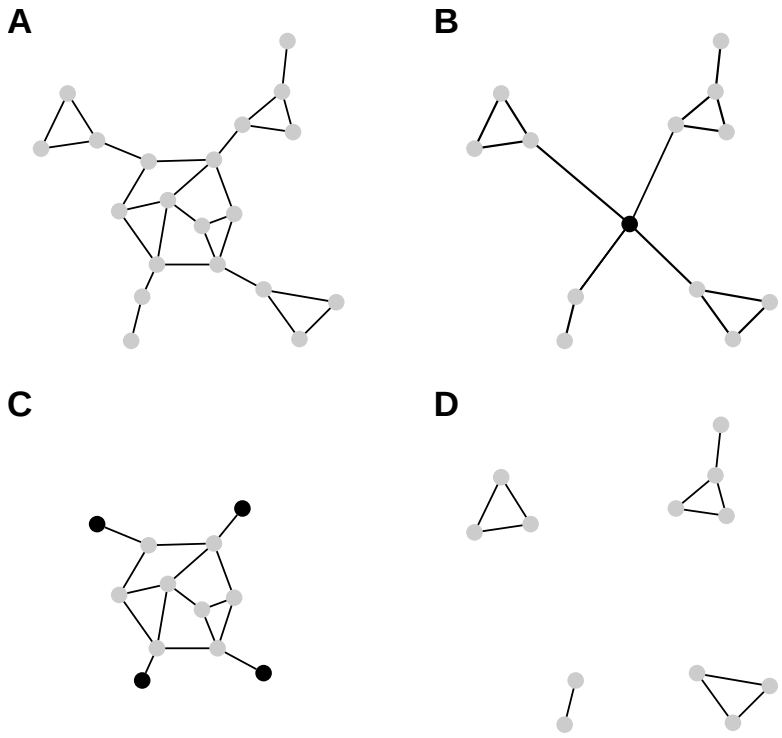


Рисунок 13 – Применения метода декомпозиции. Исходный граф (A) и экземпляры, порожденные декомпозицией (B, C и D)

весом $\omega(e) = \omega(v) + \omega(e_1) + \omega(e_2)$. Объединение всех таких неположительных цепей может быть совершено в один проход последовательным применением правила ко всем вершинам.

3.3.2. Метод декомпозиции по точкам сочленения

Разработанный метод декомпозиции по точкам сочленения основан на том, что при решении задачи *MWCS* двусвязные компоненты могут быть рассмотрены отдельно [72].

Идея метода состоит в следующем. Рассмотрим экземпляр задачи *GMWCS* (рисунок 13A). Сначала решим корневой вариант задачи *GMWCS*, *R-GMWCS*, для графа, полученного коллапсированием наибольшей двусвязной компоненты в одну вершину с нулевым весом, являющуюся корнем (рисунок 13B). Затем заменим каждую из частей графа, отходящих от наибольшей двусвязной компоненты на одну вершину с весом, равным весу соответствующего подграфа в решении корневой задачи с предыдущего шага (рисунок 13C). Из решения последней задачи, возможно получить ответ исходной,

если хотя бы одна вершина ответа лежит в наибольшей двусвязной компоненте. Для того, чтобы найти ответ в противном случае, достаточно решить задачу *GMWCS* для графа, полученного из исходного удалением наибольшей двусвязной компоненты (рисунок $13D$).

Формально, пусть B – двусвязная компонента графа G с максимальным числом вершин. Пусть C – набор точек сочленения графа G, содержащихся в B. Пусть B_c – компонента связности в графе $G \setminus (B \setminus C)$, содержащая вершину $c \in C$.

Лемма 1. *Пусть подграф $\widetilde{G}$ графа G – оптимальное решение задачи GMWCS для графа G, а $\widetilde{G}_c$ – оптимальные решения для графов B_c с корнями в вершинах c для всех $c \in C$. В этом случае, если $\widetilde{G}$ содержит вершину $c \in C$, тогда можно сконструировать оптимальное решение $\widetilde{G}'$, такое, что:*

1. $\widetilde{G}' \cap B = \widetilde{G} \cap B$ *и*
2. $\widetilde{G}' \cap B_c = \widetilde{G}_c$.

Доказательство. Обозначим часть решения $\widetilde{G}$, лежащую в подграфе B_c как $\widetilde{B}_c = \widetilde{G} \cap B_c$. Покажем, что эта часть может быть заменена на $\widetilde{G}_c$ без потери связности и оптимальности.

Во-первых, граф $\widetilde{B}_c$ должен быть связен. Докажем от противного. Пусть граф $\widetilde{B}_c$ несвязен. Тогда не существует пути между c и какой-то вершиной $v \in \widetilde{B}_c$. Поскольку $\widetilde{G}$ связен, то существует простой путь $v \rightsquigarrow c$ в $\widetilde{G}$. Однако, по определению точки сочленения, путь $v \rightsquigarrow c$ не может содержать вершин из $G \setminus B_c$ и, следовательно, он полностью лежит в B_c. Так как путь $v \rightsquigarrow c$ полностью лежит в $\widetilde{G}$ и в B_c, то он лежит и в их пересечении $\widetilde{B}_c$. Пришли к противоречию, что означает, что $\widetilde{B}_c$ связен. Поскольку $\widetilde{B}_c$ связен и содержит c, то он не может иметь вес больший, чем $\widetilde{G}_c$ по построению $\widetilde{G}_c$. Следовательно, замена $\widetilde{B}_c$ на $\widetilde{G}_c$ не может привести к уменьшению веса решения.

Во-вторых, связность решения также сохраняется. Это можно показать, повторив рассуждения из предыдущего шага доказательства и получив, что $\widetilde{G} \cap B$ должен быть связен. Таким образом, $\widetilde{G}_c$ связен, $\widetilde{G} \cap B$ связен и оба этих подграфа содержат вершину c. Поэтому и $\widetilde{G}'$ тоже связен. □

Эта лемма позволяет рассматривать только оптимальные решения, которые либо включают хотя бы одну вершину из B и во всех подграфах B_c для c из вошедших в решение точек сочленения идентичны решению соответствующего экземпляра корневой задачи, либо полностью лежат в одном из подграфов B_c.

Таким образом, на первом шаге декомпозиции необходимо для каждой вершины $c \in C$ найти оптимальное решение задачи *GMWCS* для графа B_c, содержащее вершину c. Это в точности соответствует экземпляру корневой задачи. При практической реализации, чтобы уменьшить накладные расходы, лучше на это шаге порождать один экземпляр задачи вместо $|C|$ отдельных экземпляров. Для этого рассмотрим граф $G^* = \bigcup_{c \in C} B_c$ и объединим в нем все вершины из C в одну вершину r с весом $\omega(r) = 0$. Для полученного графа задачу *R-GMWCS*. Пусть S – это решение такого экземпляра. Для того, чтобы получить решение для графа B_c можно заменить обратно r на c в S и удалить все вершины, которые не содержатся в B_c.

На втором шаге найдем подграф максимального веса, который не лежит полностью ни в одном из подграфов B_c. Пусть $\widetilde{G}_c$ – решения корневых задач для графов B_c с корнями c для всех $c \in C$, полученные на предыдущем шаге. Для того, чтобы получить новый экземпляр задачи *GMWCS*, рассмотрим граф B^*, полученный из компоненты B добавлением для каждой вершины $c \in C$ вершины v_c с весом $\omega(v_c) = \Omega(\widetilde{G}_c)$. Для того, чтобы восстановить решение исходной задачи из решения для B^*, заменим все присутствующие в решении для B^* вершины v_c на решения $\widetilde{G}_c$.

Последний, третий, шаг необходим, чтобы найти решения, которые целиком лежат в одной из компонент B_c для $c \in C$. Для этого достаточно решить экземпляр задачи *GMWCS* для графа $G^* = \bigcup_{c \in C} B_c$. Если решение задачи для графа G лежит полностью в одной из компонент B_c, то оно найдется на этом шаге работы алгоритма.

3.3.3. Сведение к задаче целочисленного линейного программирования

В этом разделе предлагается сведение задачи *GMWCS* к задаче смешанного целочисленного программирования. Сведение состоит из двух компонент: функции веса подграфа, которую необходимо оптимизировать, и ограничений на связность подграфа. Функция веса является линейной и записывается прямым образом. Для записи ограничений на связность были взяты за основу нелинейные ограничения, предложенные в работе [73] (раздел 1.5.3), для которых была разработана новая линеаризация. Кроме этого, вводятся дополнительные ограничения нарушения симметрии, уменьшающие пространство поиска.

3.3.3.1. Линеаризация

Разработанная линеаризация состоит в замене нелинейных равенств (1.7) и (1.8) на линейные неравенства следующего вида:

$$d_v + nr_v \leqslant n, \qquad \forall v \in V; \tag{3.1}$$

$$n + d_u - d_v \geqslant (n+1)x_{vu}, \qquad \forall (v,u) \in A; \tag{3.2}$$

$$n + d_v - d_u \geqslant (n-1)x_{vu}, \qquad \forall (v,u) \in A. \tag{3.3}$$

Лемма 2. *Любое возможное решение, удовлетворяющее ограничениям* (1.1)–(1.8)*, также удовлетворяет ограничениям* (1.1)–(1.6)*,* (3.1)–(3.3)*, и наоборот.*

Доказательство. Будем говорить, что две системы ограничений эквивалентны, если они ограничивают одни и те же наборы решений. В данном случае система из одного неравенства отождествляется самому неравенству.

Сначала покажем, что неравенство (3.1) эквивалентно (1.7). Поскольку r_v – бинарная переменная, то можно рассмотреть два случая. Сначала предположим, что $r_v = 1$, тогда ограничение (1.7) примет вид $d_v = 1$, а (3.1) примет вид $d_v \leqslant 1$, что вместе с ограничением на пределы d_v (1.4) дает так же $d_v = 1$. Теперь предположим, что $r_v = 0$. Неравенство (1.7) примет вид $0 = 0$, что является тождеством и эквивалентно отсутствию этого ограничения. Неравенство же (3.1) примет форму $d_v \leqslant n$, что так же эквивалентно отсутствию ограничения, так как система уже содержит более строгое ограничение (1.7). Для всех возможных значений r_v ограничение (3.1) эквивалентно (1.7), значит они эквивалентны и в целом.

Теперь покажем, что ограничение (1.8) может быть представлено в виде системы линейных неравенств (3.2) и (3.3). Во второй части доказательства будем использовать тот же подход, что и на предыдущем шаге.

Во-первых, рассмотрим случай $x_{vu} = 1$. После подстановки в (1.8) имеем $d_u = d_v + 1$. После подстановки $x_{vu} = 1$ в (3.2) и (3.3) получим:

$$n + d_u - d_v \geqslant n + 1;$$
$$n + d_v - d_u \geqslant n - 1;$$

или

$$d_u \geqslant d_v + 1;$$
$$d_v + 1 \geqslant d_u;$$

или $d_u = d_v + 1$.

Во-вторых, рассмотрим случай $x_{vu} = 0$. Нелинейное равенство (1.8) принимает вид $0 = 0$. Покажем, что (3.2) и (3.3) тоже не добавляют дополнительных ограничений на переменные. После подстановки неравенства принимают

вид:

$$n + d_u - d_v \geqslant 0;$$
$$n + d_v - d_u \geqslant 0;$$

или $|d_v - d_u| \leqslant n$. Это неравенство является более слабым вариантом неравенства (1.4), поэтому дополнительных ограничений не добавляется. □

3.3.3.2. Ограничения нарушения симметрии

С практической точки зрения важным является уменьшить число возможных решений задачи смешанного целочисленного программирования посредством ограничения числа различных, но логически эквивалентных, решений. Такие решения называются симметричными.

В используемой в данной работе формулировке ограничения (1.1)–(1.6), (3.1)–(3.3) разрешают любому дереву обхода доказывать связность графа. Рассмотрим, как уменьшить число таких возможных деревьев обхода, и, таким образом, уменьшить пространство поиска.

Прежде всего, введем правило, единственным образом определяющее возможный корень обхода для подграфа. Для этого для некорневой обобщенной задачи введем порядок $\prec$ на вершинах графа такой, что в начале идут вершины с наименьшим весом: $v \prec u$, если $\omega(v) < \omega(u)$ или если $\omega(v) = \omega(u)$ и номер v меньше номера u. Правило выбора корня заключается в том, что корнем должна быть вершина в подграфе, максимальная по введенному порядку:

$$\sum_{v \prec u} r_v \leqslant 1 - y_u, \qquad \forall u \in V. \tag{3.4}$$

Для корневой задачи корень дерева обхода устанавливается таким же, как и корень экземпляра корневой задачи.

Более того, даже из одной вершины связный подграф может быть обойден различными способами. Можно ограничить допустимые обходы только

до тех, которые являются обходами в ширину [5]. Действуя схожим с работами [6, 81] образом, введем следующие ограничения:

$$d_v - d_u \leqslant n - (n-1)w_e, \qquad \forall e = (v,u) \in E; \tag{3.5}$$

$$d_u - d_v \leqslant n - (n-1)w_e, \qquad \forall e = (v,u) \in E. \tag{3.6}$$

Эти ограничения соответствуют правилу, что если подграф содержит ребро e, то расстояния от корня до двух инцидентных вершин отличаются не более, чем на единицу.

Лемма 3. *Для любого связного подграфа G_s графа G существует решение $(\overline{r}, \overline{y}, \overline{w}, \overline{x}, \overline{d})$, которое кодирует подграф G_s и удовлетворяет:* (1.1)–(1.6), (3.1)–(3.3) *и* (3.4)–(3.6).

Доказательство. Для любого подграфа G_s можем выбрать любую его вершину как корень, в том числе и максимальную по введенному ранее порядку. Выполним из этой вершины обход в ширину. Далее, для любого связного подграфа G_s и любого его дерева обхода существует соответствующее кодирование $(\overline{r}, \overline{y}, \overline{w}, \overline{x}, \overline{d})$, которое удовлетворяет ограничениям (1.1)–(1.6) и (3.1)–(3.3). Способ выбора вершины, являющейся корнем дерева обхода, прямым образом влечет выполнение ограничения (3.4). Ограничения (3.5)–(3.6) следуют из использования обхода в ширину. □

Эта лемма позволяет добавить ограничения нарушения симметрий, не сужая набор описываемых системой подграфов.

3.4. Веб-сервис для сетевого анализа метаболомных и транскриптомных данных

Разработанный метод *GAM* был реализован в виде веб-сервиса, доступного по адресу http://genome.ifmo.ru/shiny/gam. В этом веб-сервисе включена поддержка метаболических моделей для человека и нескольких модельных организмов: мыши, резуховидки и дрожжей.

С точки зрения пользователя анализ состоит из двух шагов:

Рисунок 14 – Экран веб-сервиса после загрузки исходных данных

1. Создание графа реакций, соответствующего входным данным.
2. Поиск активного модуля в этой сети.

На первом шаге от пользователя требуется выбрать используемый организм, загрузить результаты дифференциальной экспрессии и выбрать параметры отображения сети в виде графа. Наборы данных по генам и метаболитам загружаются в виде текстовых файлов. После того, как файлы загружены, отображается информация о них, и выбираются рекомендованные параметры представления сети реакций в виде графа (рисунок 14). Нажатие на кнопку *Step 1: make network* позволяет создать необходимый граф.

После создания графа может быть выполнен поиск активного модуля. Это требует выбора значения *FDR*-порога и выбора требуется ли поиск точного решения соответствующей задачи *GMWCS*. Кнопка *Autogenerate FDRs* позволяет автоматически подобрать значения *FDR*-порога, так чтобы размер результирующего модуля был около 100–150 реакций. По умолчанию, поиск активного модуля ограничен по времени до 30 с. Такое ограничение позволяет сервису быть более интерактивным, при этом за это время часто возможно

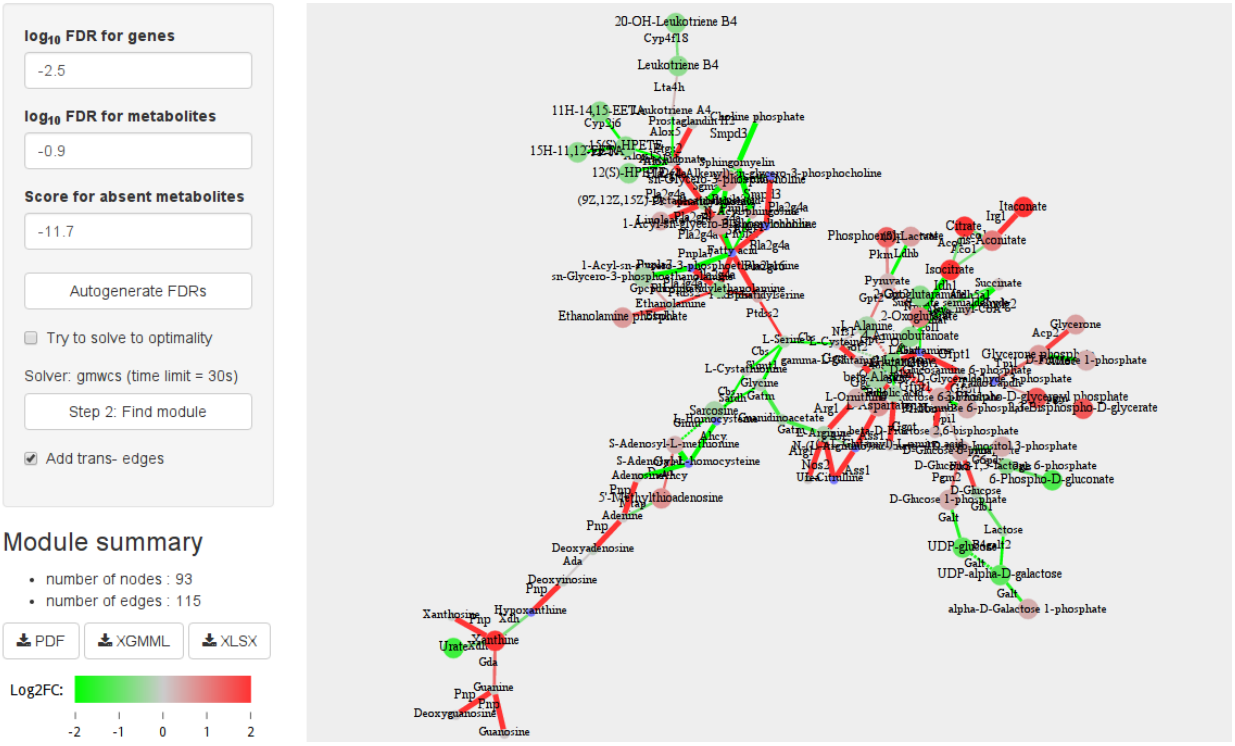


Рисунок 15 – Экран веб-сервиса после поиска модуля

найти оптимальное решение соответствующей задачи *GMWCS*, либо найти достаточно хорошее решение. Для получения финальной версии модуля рекомендуется выбирать решение задачи до оптимальности. Кнопка *Find module* запускает решение соответствующей задачи *GMWCS* и отображает найденный активный модуль (рисунок 15). Для удобства пользователя реакции и метаболиты содержат ссылки на соответствующие записи в базе *KEGG*.

Кнопка *Run step 1, autogenerate FDRs and run step 2* позволяет выполнить полный анализ в один клик с использованием рекомендованных параметров.

Результирующий модуль может быть сохранен в форматах: *PDF*, *XLSX* или *XGMML*. Формат *XGMML* позволяет загрузить модуль в программу *Cytoscape* [58] для последующего анализа.

Веб-сервис реализовав на языках *R* и *JavaScript*. Исходный код доступен под свободной лицензией по адресу https://github.com/ctlab/shinygam.

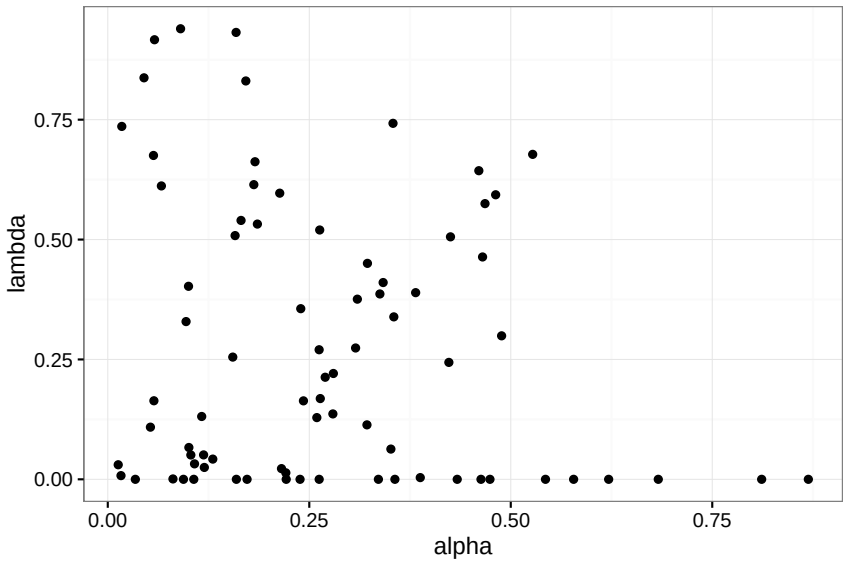


Рисунок 16 – Значения параметров λ и α для собранных реальных наборов транскриптомных данных. Каждая точка соответствует одному набору

3.5. Экспериментальное исследование

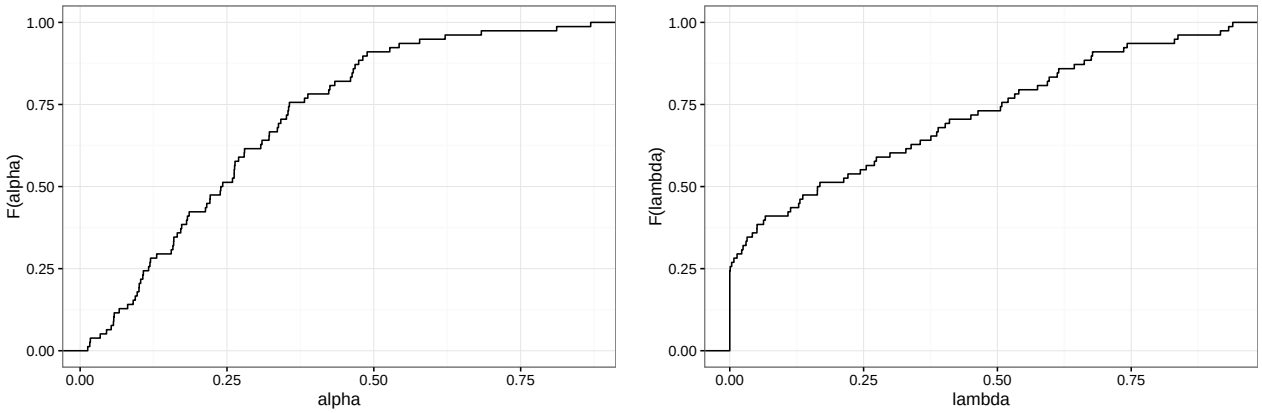
Для анализа разработанного метода *GAM* были проведены несколько вычислительных экспериментов, подтверждающих его эффективность.

3.5.1. Описание рассматриваемых наборов данных

Сначала приведем описание данных, полученных в ходе тестирования разработанного веб-сервиса (раздел 3.4). Эти реальные данные, загруженные пользователями, затем были использованы в ходе экспериментального исследования.

Во-первых, было собрано 78 таблиц с результатами анализа дифференциальной экспрессии генов. Это число исключает из себя похожие файлы, файлы со слишком малым числом генов и т. д.

Было проанализировано распределение параметров смеси бета-равномерного распределения в этих данных (рисунок 16). Параметр α определяет форму распределения: чем меньше α тем больше сильнозначимых P-значений в выборке. Параметр λ определяет соотношение распределений в смеси: чем больше λ тем больше шума в данных. Можно заметить, что при значениях $\alpha \gtrsim 0{,}5$ значение λ почти всегда равно нулю. Это



(а) (б)

Рисунок 17 – Эмпирические функции распределения параметров α (а) и λ (б)

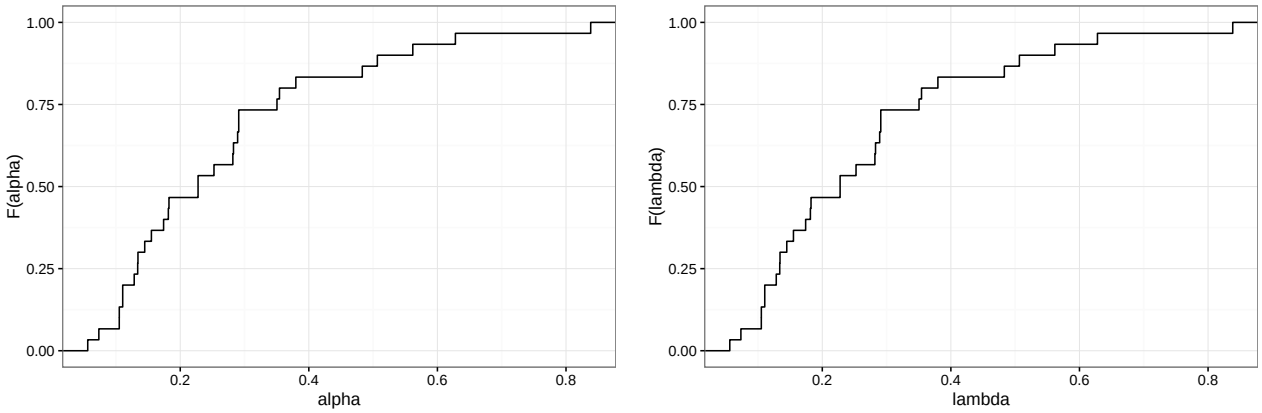
объясняется тем, что при таких больших значениях α бета-распределение становится сложно отличить от равномерного.

Маргинальные распределения параметров α и λ представлены на рисунке 17. Отметим, что в области значений $(0, 0{,}5)$ значения α хорошо подчиняются равномерному распределению. Аналогична ситуация для λ для значений в интервале $(0, 0{,}75)$.

Также было рассмотрено 30 наборов данных, содержащих одновременно и метаболомные, и транскриптомные данные. В этих данных распределение параметров α и λ было аналогично рассмотрено для генов, а также метаболитов. Для генов в распределении не было заметных изменений по сравнению со всем наборов. Маргинальные распределения параметров для метаболитов представлены на рисунке 18. В целом, можно заключить, что метаболомные данные ведут себя похожим на транскриптомные данные образом, за исключением отсутствия наборов данных с $\lambda = 0$.

3.5.2. Исследование точности метода на искусственных данных дифференциальной экспрессии генов

Сначала исследовалась работа метода GAM на искусственных результатах дифференциальной экспрессии. Использование таких данных позволяет оценить точность метода, так как для них известен правильный модуль.



(а) (б)

Рисунок 18 – Эмпирические функции распределения параметров α (а) и λ (б) для метаболомных данных

Был разработан следующий протокол. Во-первых, за основу берется граф, получившийся в результате анализа данных реального эксперимента. Во-вторых, генерируется «правильный» модуль, состоящий из нескольких метаболических путей. Затем для генов случайным образом генерируются P-значения: из бета-распределения для генов из правильного модуля и из равномерного распределения для всех остальных. На таких сгенерированных данных запускается анализ, результатом которого является список реакций, входящих в модуль, и соответствующий список генов. Этот список генов можно сравнить с генами правильного модуля и оценить точность работы метода как на уровне отдельных генов, так и на уровне метаболических путей.

В качестве основы для эксперимента использовался граф, полученный для результатов дифференциальной экспрессии эксперимента *GSE59228* [82], в котором сравнивалось контрольное состояние клетки и состояние после добавления 2-деокси-глюкозы. Так как использовались только данные по экспрессии генов, был выбран вершинный вариант представления сети реакций в виде графа. Получившийся граф состоял из 2882 вершин и 3345 ребер, 1048 вершин соответствовало реакциям, 1834 – метаболитам. С реакциями в этом графе было ассоциировано 628 генов, которые и рассматриваются в дальнейшем. Обозначим множество этих генов как U.

Для конструирования искусственного активного модуля был отобран набор M метаболических путей из базы *KEGG*. Были отфильтрованы модули, размер пересечения генов которых со множество U был меньше двух. Кроме этого, был построен граф связности этих путей, где ребро соответствовало наличию общего метаболита, и были оставлены только пути, входящие в наибольшую компоненту связности. После этих операций осталось 43 пути.

Искусственный активный модуль генерировался путем объединения случайных связанных метаболических путей из набора M. Будем обозначать множество генов в полученном модуле как A. Для модуля случайным образом выбирался ожидаемый размер множества A из равномерного распределения от 20 до 100. Затем, итеративно, в активный модуль по одному добавлялись случайные пути из M, пока размер не достигал необходимого. При этом, пути добавлялись таким образом, чтобы они образовывали связный граф на каждой итерации. Для этого первым выбирался равномерно случайный некоторый путь из M, а последующие – выбирались из путей M, имеющих хотя бы один общий метаболит с хотя бы одним из уже выбранных путей. Медианный размер сгенерированного активного модуля составил 16 метаболических путей.

После выбора активного модуля выполнялась генерация P-значений для всех рассматриваемых генов из множества U. Для множества генов A, входящих в активный модуль, P-значения генерировались из бета-распределения $B(\alpha, 1)$, в соответствии с рассматриваемой моделью. Значение параметра α выбиралось из равномерного распределения $\mathcal{U}(0{,}01, 0{,}05)$. Это распределение примерно соответствует распределению α в реальных данных (раздел 3.5.1). Для всех остальных генов P-значения выбирались из распределения $\mathcal{U}(0, 1)$. Всего было сгенерировано 200 наборов данных.

Для определения точности разработанного метода на широком диапазоне значений параметра FDR-порога на P-значения, поиск модуля выполнялся до 30 раз с разными значениями этого параметра. Выбор значения по-

рога производился так, чтобы число генов, имеющих положительный вес, варьировалось от одного до $|U|$. Запуск для значений FDR-порога, больших 0,1, не проводился.

Затем анализировалась точность нахождения отдельных генов, если рассматривать ее как задачу классификации генов на гены, принадлежащие активному модулю, и не принадлежащие. Для этого было решено исследовать зависимость точности (*precision*) от полноты (*recall*).[1] Эта зависимость сравнивалась с таковой для базового алгоритма, выводящего просто список всех генов, P-значения которых меньше заданного порога. Построение кривой точность-полнота требует иметь на входе ранжированный список генов. Для разработанного метода гены ранжировались в зависимости от числа порогов, для которых этот ген входил в соответствующий найденный модуль. Для базового алгоритма, ранжирование естественным образом выполнялось по P-значениям.

Возможны несколько вариантов соотношений графиков точность-полнота для разработанного метода и для базового метода (рисунок 19). Во-первых, в общем случае, эти графики могут быть несравнимыми, пример такой ситуации показан на рисунке 19*а*. Этот вариант возникает, когда метод GAM неправильно включает какой-то ген в результирующий модуль, даже при строгих порогах (поэтому точность быстро падает), но за счет связности позволяет выявить больше правильных генов при слабых порогах. Во-вторых, метод GAM может во всех точках быть лучше базового метода (рисунок 19*б*, график для базового метода лежит полностью ниже графика для GAM). В-третьих, наоборот, базовый метод может во всех точках быть лучше метода GAM (рисунок 19*в*).

[1] Значение точности и полноты вычисляется по формулам `TP/(TP+FP)` и `TP/(TP + FN)`, где `TP` – число истинно-положительных результатов, `FP` – число ложно-положительных, `FN` – число ложно-отрицательных.

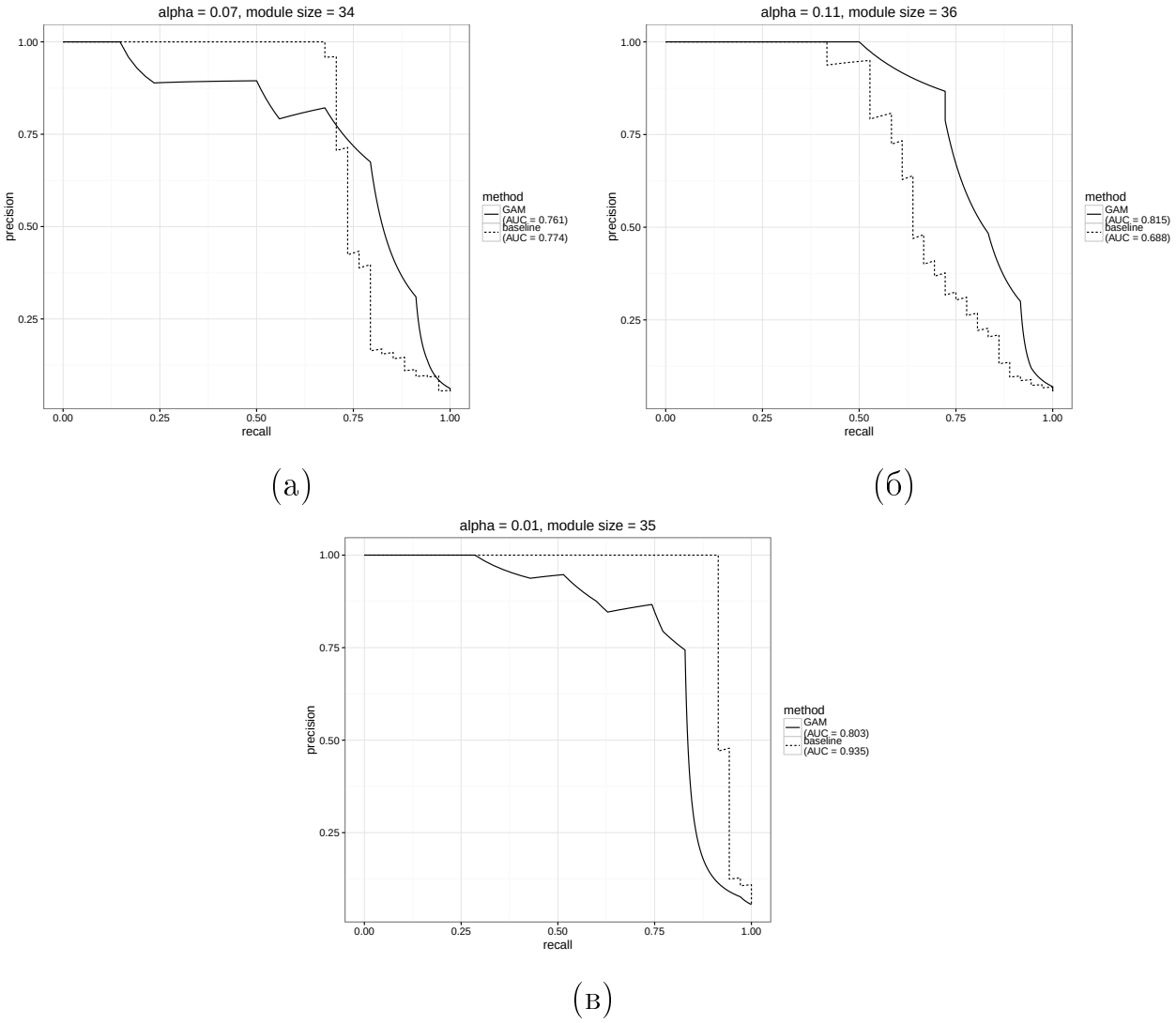
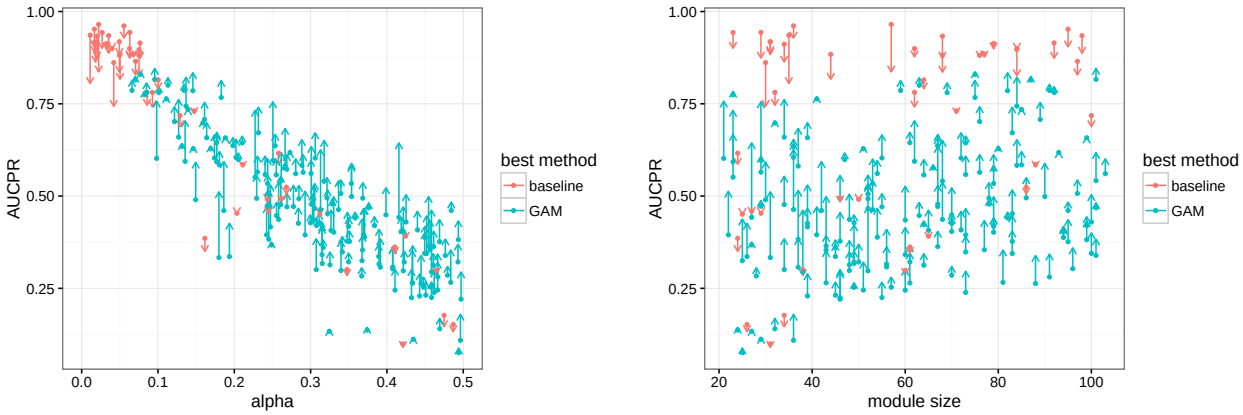


Рисунок 19 – Примеры графиков точность-полнота для предложенного метода (*GAM*, сплошная линия) и базового методоа (*baseline*, пунктирная линия) при разных значений параметра α и размера модуля. (а) Пример, когда площади под кривой примерно совпадают. (б) Пример, когда метод *GAM* лучше во всех точках. (в) Пример, когда метод *GAM* хуже во всех точках



(а) (б)

Рисунок 20 – Зависимость площади под кривой точность-полнота от значения параметра α (а) и размера модуля (б). Каждому тесту соответствует одна стрелка. Ордината начала стрелки соответствует значению площади под кривой для базового метода, ордината конца – для метода *GAM*. Красным цветом обозначены тесты, в которых точнее классифицирует гены базовый метод, бирюзовым – метод *GAM*

Для сравнения большого числа таких экспериментов часто рассматривается площадь под кривой точность-полнота. Зависимость этой площади параметра α и от размера модуля представлена на рисунке 20. Во-первых, можно заметить, что площади под кривой точность-полнота для обоих методов падает с ростом значения α (рисунок 20*а*). Это объясняется тем, что при росте α распределение $B(\alpha, 1)$ становится все более «плоским» и генерируется все больше P-значений, которые сложно отличить от равномерного распределения $\mathcal{U}(0, 1)$. Таким образом, становится сложно отличить сигнал от шума и общая точность снижается. В то же время, площадь под кривой при росте α для предложенного метода *GAM* падает медленнее, чем для базового метода. В сложных ситуациях при значении $\alpha \gtrsim 0.1$ метод *GAM* начинает выигрывать по этому параметру. Зависимости площади под кривой от размера модуля не наблюдается (рисунок 20*б*).

Следовательно, если рассматривать разработанный метод *GAM* как метод определения генов, принадлежащих активному модулю, то в зависимости от параметра α, общая точность метода может быть как хуже базового мето-

да ранжированием по P-значению, так и лучше. При этом, для большинства реальных наборов, α принимает значения в области, где метод GAM работает лучше.

Следующим шагом был выполнен анализ точности на уровне метаболических путей. Сравнивалось два метода определения активных метаболических путей. Во-первых, был рассмотрен метод, выполняющий анализ представленности с помощью точного теста Фишера для генов модуля, полученного методом GAM. Пусть G – гены, в выведенном GAM активном модуле, а $m \in M$ – некоторый метаболический путь из $KEGG$. Тогда можно с помощью точного теста Фишера вычислить вероятность получить хотя бы $|m \cap G|$ генов из модуля m, если выбрать случайные $|G|$ генов из U. Кроме это использовался метод, в котором брались $|G|$ генов из U с минимальными P-значениями, после этого так же выполнялся тест Фишера. Для того, чтобы тесты на активность метаболических путей были достаточно независимыми, тестировались только те метаболические пути, которые либо использовались в построении искусственного активного модуля, либо пересекались с таковыми не больше чем на один ген. Оба метода запускались один раз на каждом тесте. Для анализа из модулей, полученных для разных значений FDR-порога, выбирался один случайный, имеющий размер от 10 до 100. Во всех методах выполнялась поправка на множественное сравнение методом Бенджамини-Хохберга.

Для начала была рассмотрено, насколько в такой процедуре определения активных метаболических путей контролируется ошибка первого рода (уровень ложноположительных срабатываний), когда неактивные метаболические пути принимаются за активные. Для это было проанализировано распределение получающихся номинальных P-значений для неактивных путей. Было проведено сравнение этого распределения с равномерным распределением $\mathcal{U}(0, 1)$ (рисунок 21). В идеале эти два распределения должны совпадать, что на графике отражалось бы в виде диагональной линии с единичным уклоном. Можно заметить, что как для метода GAM, так и для базово-

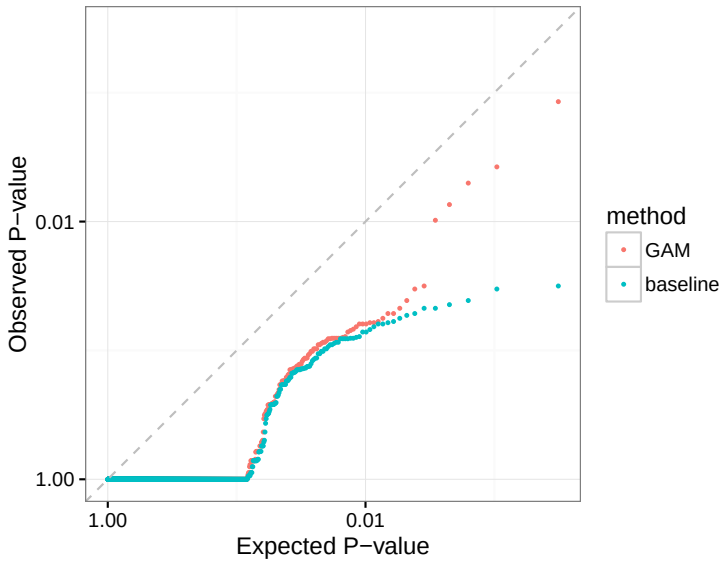


Рисунок 21 – Графики квантиль-квантиль для распределений P-значений при верной нулевой гипотезе и равномерного распределения $\mathcal{U}(0,1)$ для метода GAM (красным) и базового метода (бирюзовым)

го метода, графики лежат ниже диагонали. С одной стороны, это означает, что ошибка первого рода контролируется и число ложно-положительных срабатываний не будет превышать номинального P-значения. С другой стороны, оба теста являются консервативными, что влечет к повышенному числу ложно-отрицательных срабатываний. Консервативность теста Фишера является известной его особенностью и происходит из-за дискретности данных.

В соответствии с тем, что оба теста контролируют уровень ошибок первого рода и используется процедура коррекции на множественное сравнение методом Бенджамини-Хохберга, уровень FDR также контролируется (рисунок 22).

Затем был проведен ROC-анализ (*Receiver Operating Characteristic*), в котором исследуется зависимость числа истинно-положительных значений от числа ложно-положительных (рисунок 23). В таком анализе исследуются насколько хорошо метод ранжирует результаты вне зависимости от того, контролируется ли уровень ложно-положительных результатов или нет. Результаты показывают, что оба метода достаточно хорошо работают в области

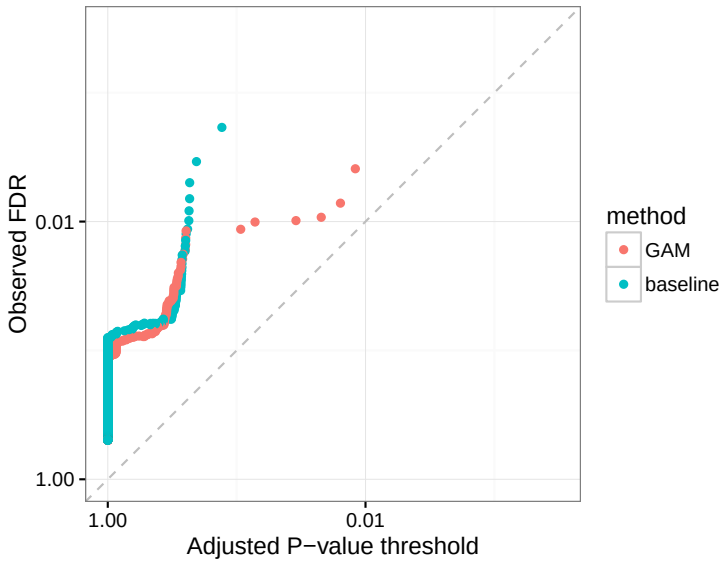


Рисунок 22 – График зависимости наблюдаемого уровня *FDR* от порога на скорректированное *P*-значение

значений уровня *FPR* (`FPR=FP/(FP+TN)`) близких к нулю. При этом при увеличении *FPR* базовый метод начинает выигрывать.

Также была исследована зависимость числа истинно-положительных результатов от заданного порога на скорректированное *P*-значение (рисунок 24). Несмотря на то, что базовый метод несколько лучше ранжирует тестируемые метаболические пути, важной является возможность выводить результаты с заранее заданным уровнем *FDR*. Анализ показал, что при выборе фиксированного порога меньшего стандартного значения 0,05, разработанный метод *GAM* выдает больше результатов по сравнению с базовым методом, при том, что в обоих методах контролируется *FDR*. При этом, например, при значении порога 0,01 числа найденных методом *GAM* путей превышает на 40% число таковых для базового метода.

3.5.3. Исследование точности метода на искусственных данных совместно для генов и метаболитов

Протокол исследования работы метода при совместном запуске на сгенерированных транскриптомных и метаболомных данных в целом повторяет протокол из предыдущего раздела. Отличием является выбранный исходный

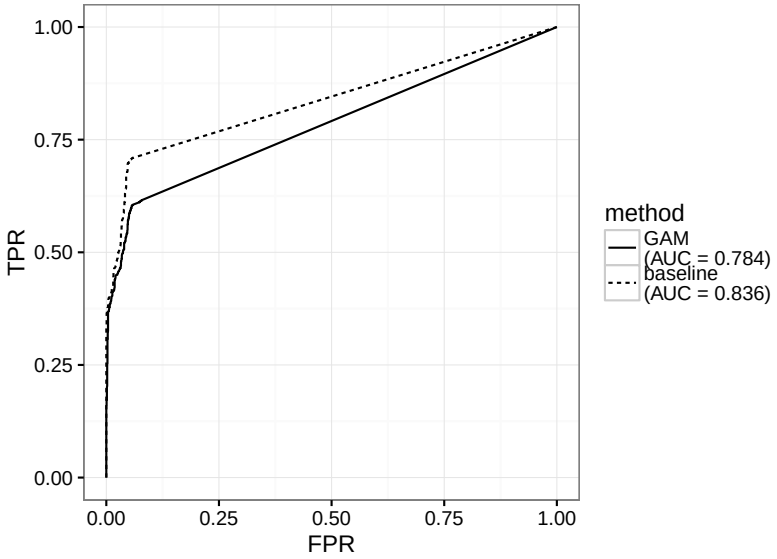


Рисунок 23 – Кривая ошибок определения активных метаболических путей

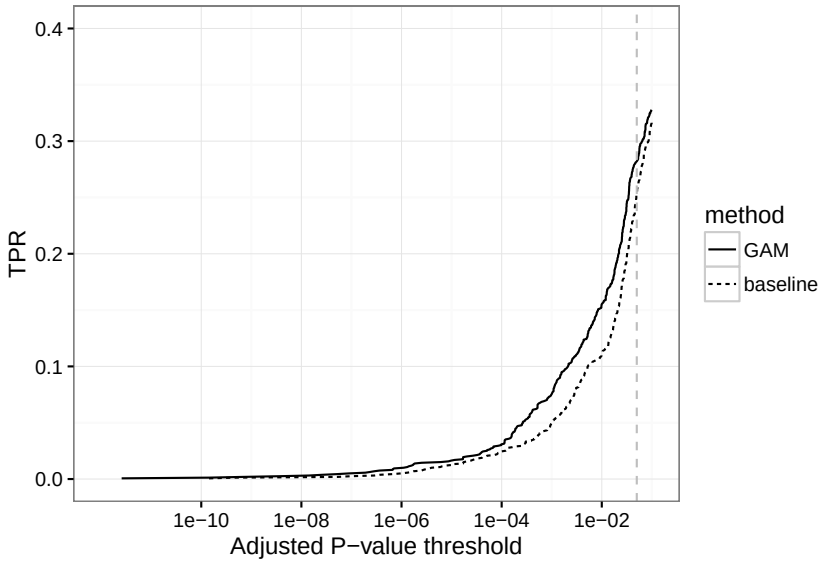
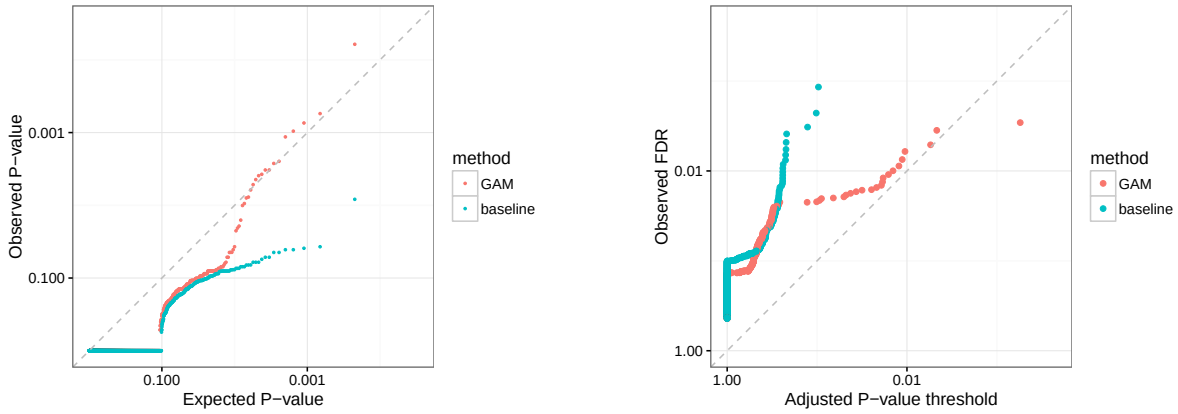


Рисунок 24 – Зависимость TPR от порога на скорректированное P-значение. Вертикальной пунктирной линией отмечено значение порога 0,05



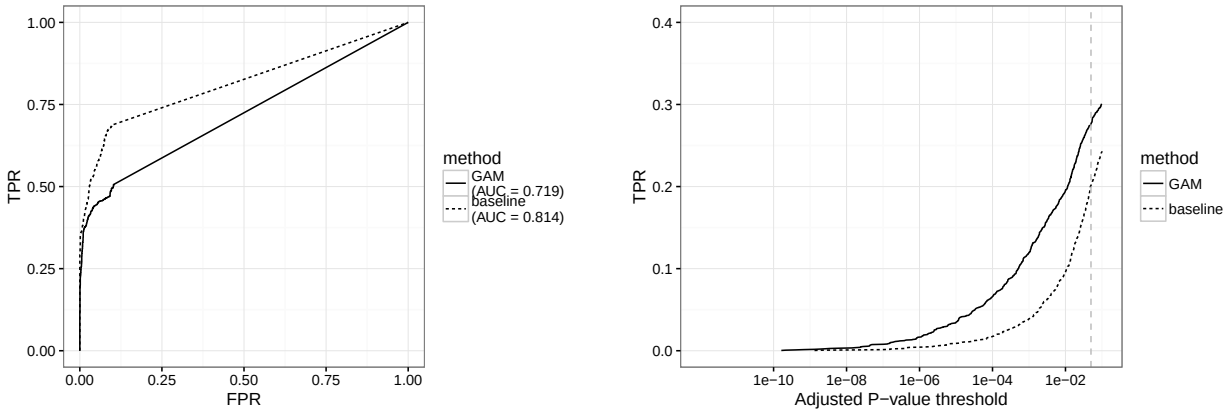
(а) (б)

Рисунок 25 – а) Графики квантиль-квантиль для распределений P-значений при верной нулевой гипотезе и равномерного распределения $\mathcal{U}(0,1)$. б) График зависимости наблюдаемого уровня FDR от порога на скорректированное P-значение. Кривые для метода GAM обозначены красным, для базового метода – бирюзовым

граф: он был получен из тех же данных, но при использовании вершинно-реберного представления. Получившийся граф состоял из 1562 вершин и 1493 ребер, с реакциями было ассоциировано 534 гена.

Генерация P-значений для метаболитов выполнялась похожим образом на генерацию для генов. Для учета особенности метаболомных данных, что не для всех метаболитов может быть получен соответствующий сигнал, использовалось только множество метаболитов, измеренных в работе [99]. Кроме этого, генерация выполнялась на уровне химических формул, так как обычно с помощью масс-спектрометрии сложно отличить два разных вещества, имеющих общую химическую формулу. Таких измеренных и представленных в сети формул было 216. Всего было сгенерировано 300 наборов данных.

Был проведен анализ точности нахождения активных метаболических путей таким же, как и в предыдущем разделе, методом. По сравнению с запуском только на транскриптомных данных, тест стал менее консервативным (рисунок 25). Имеется небольшой выход за диагональную линию, но, в целом, метод почти хорошо контролирует уровень ошибки первого рода и становится неконсервативным.



(а) (б)

Рисунок 26 – (а) Кривая ошибок и (б) зависимость TPR от порога на скорректированное P-значение для определения активных метаболических путей при одновременном анализе сгенерированных транскриптомных и метаболомных данных. Кривые для метода GAM обозначены сплошной линией, для базового метода – пунктиром

Затем были проанализированы положительные результаты (рисунок 26). Хотя метод GAM уступает базовому методу по зависимости числа истинно-положительных результатов от числа ложно-положительных, метод GAM выигрывает, если рассматривать зависимость числа истинно-положительных результатов от заданного порога на FDR.

3.5.4. Исследование работы метода на реальных данных

Для исследования метода на реальных данных анализировалось число найденных активных метаболических путей. Такой метод был выбран из-за невозможности напрямую оценить точность метода из-за неизвестности правильного ответа.

Во-первых, по процедуре, аналогичной описанной в разделе 3.5.2, был проведен анализ только на транскриптомных данных. Для каждого входного набора выполнялось до 30 запусков разработанного метода GAM для разных значений FDR-порога. Из полученных модулей выполнялся один, с числом генов от 10 до 100 и FDR-порогом меньшим 0,1. Затем выполнялся точный тест Фишера на множестве независимых метаболических путей,

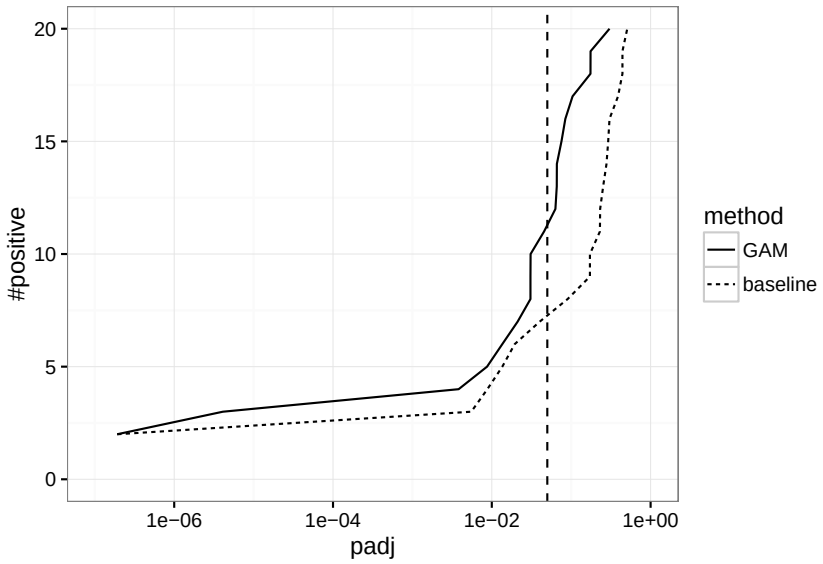


Рисунок 27 – Зависимость числа положительных результатов анализа представленности в зависимости от порога на скорректированное P-значение для реальных транскриптомных данных

имеющих пересечение между собой не больше единичного размера. Для того, чтобы уменьшить эффект множественного сравнения, рассматривались только метаболических пути размера не меньше пяти.

Анализ показал, что число положительных результатов, найденных с помощью метода GAM больше соответствующего числа для базового метода (рисунок 27). В качестве базового метода, как и раньше, производился выбор такого же по размеру набора генов с минимальными P-значениями. Из 78 наборов для 69 существовал модуль подходящего размера. При выборе значения порога на скорректированное P-значение равным 0,05 методом GAM обнаруживается десять метаболических путей для восьми входных наборов, а базовым – шести путей для четырех наборов.

Во-вторых, были выполнен запуск метода GAM на 30 наборах с одновременно транскриптомными и метаболомными данными. Сравнение числа найденных метаболических путей с базовым методом представлено на рисунке 28. Из 30 наборов методом GAM было найдено 14 путей для 11 наборов, базовым методом – два пути для одного набора.

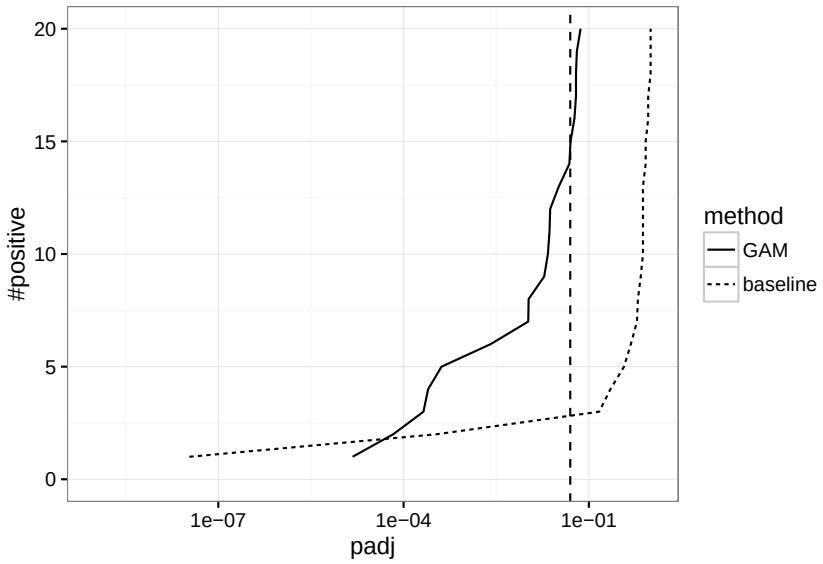


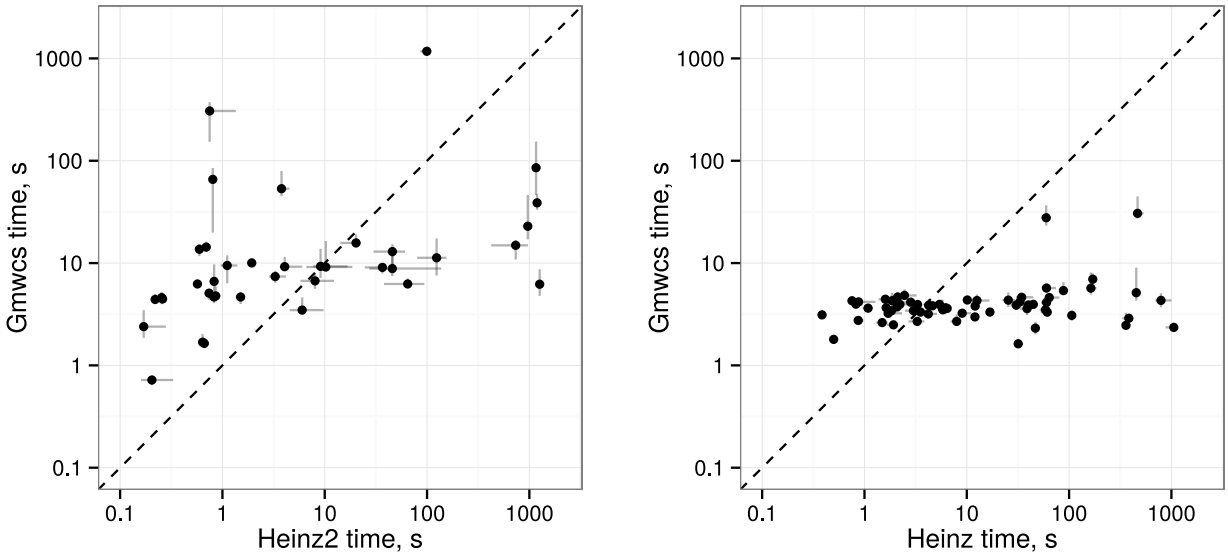
Рисунок 28 – Зависимость числа положительных результатов анализа представленности в зависимости от порога на скорректированное P-значение для реальных транскриптомных и метаболомных данных

3.5.5. Анализ времени работы решателя

Дополнительно была проанализирована эффективность решателя на экземплярах задач *SMWCS* и *GMWCS*, возникших при работе разработанного веб-сервиса. Всего было рассмотрен 101 экземпляр, из них 38 экземпляров *SMWCS* и 63 – *GMWCS*.

Сравнение проводилось с двумя другими решателями: *Heinz* версии 1.68 [61] и *Heinz2* версии 2.1 [72] (раздел 1.5.2). Каждый решатель был запущен на каждом соответствующем экземпляре задачи десять раз с ограничением по времени, равным 1000 с. Запуск выполнялся с возможностью использования до четырех потоков исполнения. Тестирование проводилось на компьютере с процессором *AMD Opteron 6380 2.5ГГц*.

Для задачи *SMWCS* результаты сравнения времени работы разработанного решателя *GMWCS* и решателя *Heinz2* представлены на рисунке 29*a*. В целом, времена работы решателей достаточно похожи. Для 24 экземпляров (63%) решение было найдено быстрее с помощью *Heinz2*. С другой стороны, для 32 экземпляров (84%) решатель *GMWCS* отработал быстрее чем



(а) (б)

Рисунок 29 – Сравнение времени работы: (а) решателей *GMWCS* и *Heinz2* на экземплярах задачи *SMWCS*; (б) решателей *GMWCS* и *Heinz* на экземплярах задачи *GMWCS*. Точки представляют медианное значение времени работы решателей при десяти запусках. Серые линии обозначают вторые по максимальности и минимальности времена работы

за 30 с, в то время как аналогичное значение для *Heinz2* составило 27 экземпляров (71%). Кроме того, только один экземпляр был не решен решателем *GMWCS* за отведенные 1000 с, в то время, как *Heinz2* не успел решить четыре экземпляра.

Для экземпляров задачи *GMWCS* выполнялось сравнение с решателем *Heinz*, который может работать с весами ребер, но ищет только подграфы без циклов (рисунок 29*б*). На этих экземплярах разработанный решатель показал себя значительно лучше, чем *Heinz*. Для всех кроме двух экземпляров было найдено оптимальное решение меньше чем за десять секунд. В то же время решение с помощью *Heinz* заняло больше десяти секунд на 30 экземплярах (48%). Более того, только 35 экземпляров (56%) имели ациклические оптимальные решения, поэтому 28 экземпляров не были им решены до оптимальности в терминах задачи *GMWCS*.

3.6. Пример применения метода на данных активации мышиных макрофагов

Применение метода GAM рассматривается на одном из типов клеток иммунной системы – клетках макрофагов. Биологической задачей является определение того, какие метаболические процессы важны при происходящем в процессе воспаления переходе макрофагов из пассивного состояния ($M0$) в классически активированное ($M1$). Соответствующие транскриптомные и метаболомные данные взяты из [99]. В них содержится информация о значимости и направлении регуляции отдельных генов и веществ.

Сначала метод строит граф, в котором вершинам соответствуют вещества, а ребрам – все известные теоретически возможные биохимические реакции, в которых эти вещества участвуют. Получающийся граф состоит из 942 вершин и 1177 ребер. По построению, в этом графе ребра инцидентны, если у соответствующих двух реакций есть общее вещество, причем часто оно является продуктом одной реакции и субстратом другой.

Потом, исходя из предположения, что все важные метаболические процессы должны быть связаны между собой, метод ищет в этом графе связный подграф. Из всех возможных подграфов выбирается один, максимальный по критерию, отражающему наличие в подграфе большого числа сильно регулируемых реакций и малого числа слабо регулируемых. Поиск такого подграфа выполняется путем решения экземпляра задачи $GWMCS$ с помощью разработанного решателя.

Получившийся в рассматриваемом примере подграф представлен на рисунке 30. В нем размер вершин и ребер соответствует значимости индивидуальной регуляции, а цвет – направлению регуляции. Красным обозначены гены и вещества, концентрация которых увеличивается при активации, зеленым – которых уменьшается, а синим обозначены вещества, для которых данные отсутствуют.

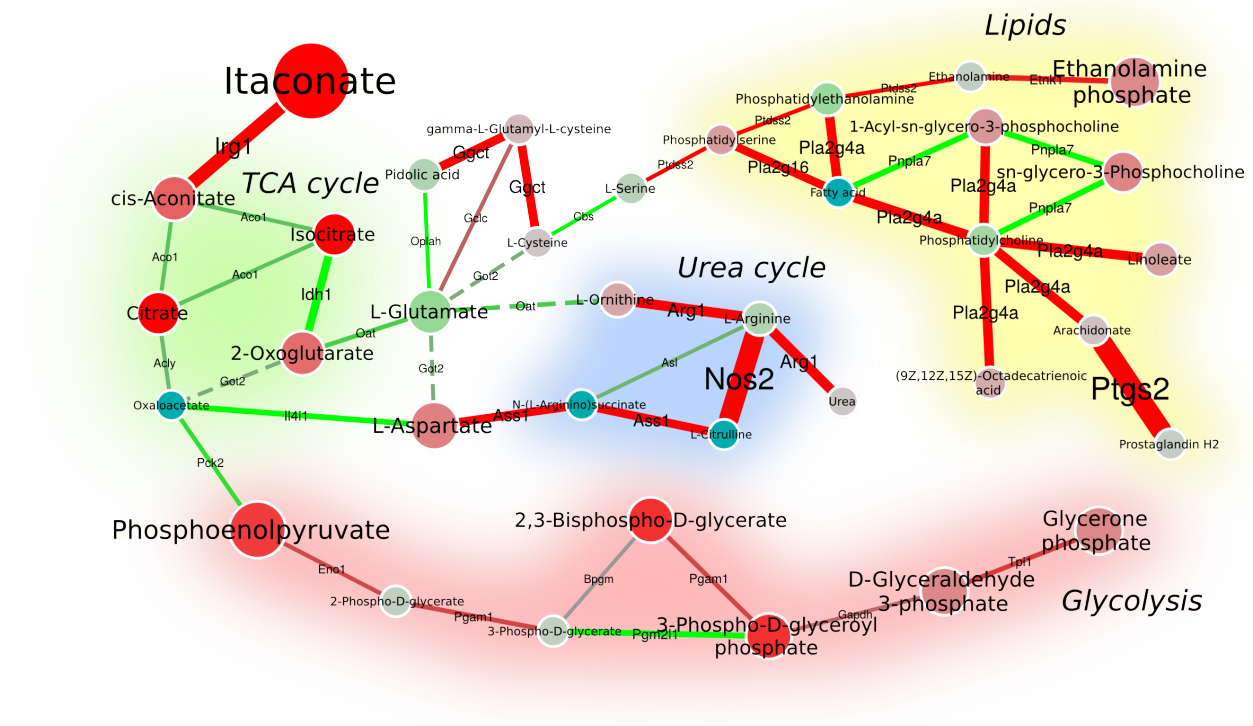


Рисунок 30 – Модуль, найденный с помощью метода *GAM*, при сравнении неактивированных и классически активированных макрофагов. Красным цветом обозначены метаболиты и гены, экспрессия которых идет вверх при активации, зеленым – для которых экспрессия идет вниз

Рассмотрим, как этот подграф можно использовать для интерпретации экспериментальных данных. Во-первых, его можно проанализировать на наличие стандартных метаболических процессов (метаболических путей). В этом подграфе биолог может выделить четыре таких процесса:

— процесс расщепления глюкозы (*Glycolysis* на рисунке);

— одну из ключевых частей процесса дыхания – цикл Кребса (*TCA Cycle*);

— распад азотсодержащих веществ через цикл мочевины (*Urea cycle*);

— распад и синтез липидов (*Lipids*).

Важность регуляции этих процессов при классической активации макрофагов подтверждается литературными источниками [83].

Во-вторых, в модуле присутствуют реакции, не относящиеся к таким стандартным метаболическим процессам. Они либо могут обеспечивать связь перечисленных выше процессов, либо могут быть важны сами по себе. Так,

например, вещество *Itaconate* – продукт реакции *Irg1* – является одним из регуляторов иммунного ответа [97].

Таким образом, предлагаемый метод позволяет по метаболической модели построить граф всех возможных реакций и выделить в нем подграф, содержащий наиболее важные метаболические процессы.

Выводы по главе 3

1. Разработан метод *GAM* для выделения активного модуля в сети метаболических реакций с помощью сведения к задаче *GMWCS*.
2. Для задачи *GMWCS* разработан практический точный решатель, выложенный в открытый доступ по адресу https://github.com/ctlab/gmwcs-solver/.
3. Разработан веб-сервис *Shiny GAM*, реализующий предложенный метод, доступный по адресу http://genome.ifmo.ru/shiny/gam/.
4. Проведено экспериментальное исследование подтверждающее эффективность метода на модельных и экспериментальных данных.
5. Приведен пример, демонстрирующий применение метода к реальным экспериментальным данным для их интерпретации специалистом по биологии.

ГЛАВА 4. МЕТОД ПОИСКА АКТИВНОГО МЕТАБОЛИЧЕСКОГО МОДУЛЯ С ПОМОЩЬЮ АНАЛИЗА ГРАФА АТОМНЫХ ПЕРЕХОДОВ

Настоящая глава посвящена задаче разработки и реализации эффективного метода идентификации регулируемых путей и их взаимосвязей в метаболических моделях на основе поиска активного модуля с использованием информации об атомной структуре метаболитов.

Для решения этой задачи был разработан метод *GATOM* (от *GAM* и *atom*) для выделения активного метаболического модуля с помощью анализа графа атомных переходов. Для этого вводится сигнальный вариант задачи *GMWCS – SGMWCS* (*Signal GMWCS*), к которому выполняется сведение. Для задачи *SGMWCS* разработан решатель, который способен для многих экземпляров, возникающих на практике, найти точное решение за приемлемое время.

Разработанный метод встроен в описанный ранее (раздел 3.4) веб-сервис *Shiny GAM*.

4.1. Использование графа атомных переходов

Метод *GATOM* так же, как и метод *GAM*, состоит из двух частей. Сначала строится большой граф возможных реакций, в котором, в отличие от метода *GAM*, вершинами являются отдельные атомы углерода. Затем в нем выделяется связный подграф, содержащий наиболее важные реакции.

4.1.1. Сравнение графа атомных переходов с графом метаболических реакций

Рассмотрим реакцию *R00900* (*L-Cysteine* + *Glutathione* + $NADP^+$ <=> *S-Glutathionyl-L-cysteine* + *NADPH*, рисунок 31), которая позволяет в методе *GAM* последовательно соединить *L-Cysteine*, *S-Glutathionyl-L-cysteine* и *Glutathione* даже при представлении реакций в виде ребер и использовании

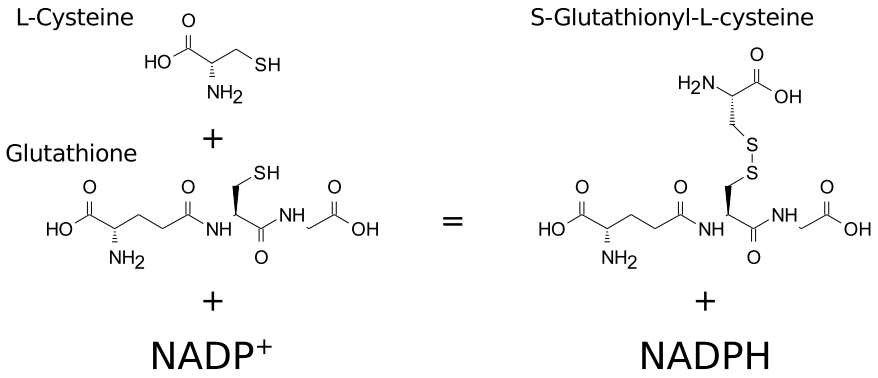


Рисунок 31 – Пример реакции *R00900*, которая позволяет соединить два субстрата реакции (*L-Cysteine* и *Glutathione*) через общий продукт (*S-Glutathionyl-L-cysteine*)

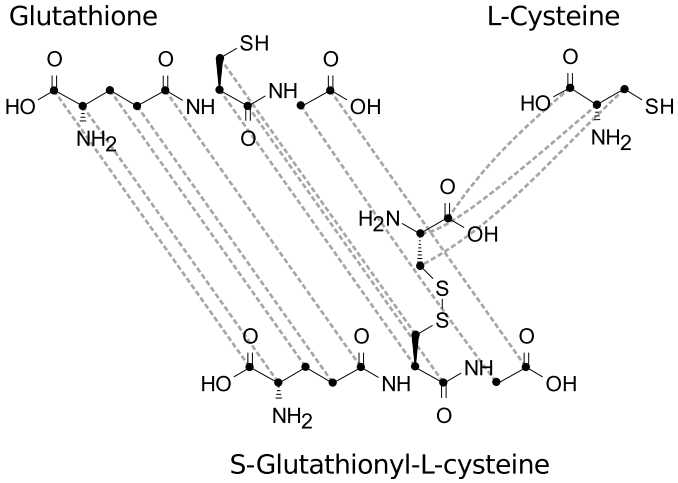


Рисунок 32 – Пример фрагмента графа атомных переходов для реакции *R00900*. Серые пунктирные линии соответствуют ребрам, соединяющим отдельные атомы углерода

только «основных» пар субстрат-продукт. Такая тройка метаболитов с точки зрения интерпретации сильно отличается от последовательного превращения одного метаболита в другой через промежуточный метаболит.

Для того, чтобы учитывать этот эффект, можно рассмотреть граф атомных переходов. В нем каждому метаболиту соответствует по одной вершине на каждую молекулу углерода. Ребро соединяет две вершины, если существует реакция, при которой эти атомы переходят друг в друга. В таком графе прямое соединение *L-Cysteine* – *S-Glutathionyl-L-cysteine* – *Glutathione* отсутствует (рисунок 32).

4.1.2. Построение графа атомных переходов

Граф атомных переходов строится на основе базы *KEGG RPAIR* [18]. В этой базе для большого числа реакций (примерно 8000) указаны переходы атомов субстратов в атомы продуктов. Хотя метод применим для любых атомов одного типа, на практике наиболее осмысленными является рассмотрение только атомов углерода.

К сожалению, нумерация атомов для одного и того же метаболита может отличаться от реакции к реакции. При построении же графа атомных переходов, важно понимать возможные превращения конкретного атома метаболита. В связи с этим возникает задача нормализации нумерации – приведения ее в одинаковый вид для всех вхождений одного метаболита в разные реакции.

Поставим формально эту задачу следующим образом. Даны два графа S и T, соответствующие химической структуре некоторого метаболита C. В этих графах вершинами являются атомы метаболита с пометками, обозначающими тип атома (например, водород, углерод и т. д.), а ребрами являются связи между атомами. Ребрам сопоставлены пометки, соответствующие типу связи (например, одинарная, двойная и т. д.). Вершины в каждом из графов пронумерованы от единицы до числа вершин. Рассмотрим все такие перестановки π вершин графа T, что получающийся перенумерацией вершин граф $T[\pi]$ полностью совпадает с S. Совпадением графов при этом называется ситуация, когда одновременно:

— в обоих графах одинаковое число вершин n и ребер m;

— для каждого номера вершины i совпадают типы атомов в i-х вершинах обоих графов;

— и для каждой пары номеров вершин i и j в обоих графах либо одновременно отсутствуют, либо одновременно присутствует ребро между i-й и j-й вершинами, а пометки на соответствующих ребрах совпадают.

Задачей является, во-первых, проверка того, что существует хотя бы одна такая перенумерация. Во-вторых, если перенумерация существуют, то необходимо определить все вершины, соответствующие атомам углерода, для которых перенумерация уникальна. Таким образом, требуется найти такие атомы углерода, для которых существует однозначное соответствие в разных нумерациях.

Для решения этой задачи был разработан следующий алгоритм. Сначала проверяется совпадение числа вершин и ребер отдельных типов. Затем рекурсивным образом находятся все возможные перенумерации. Рекурсивная функция принимает частично определенную перенумерацию, и выбирает для следующей вершины из T возможные кандидаты из S, постановка в соответствие которых совместима с уже определенной перенумерацией. Алгоритм прерывается, либо когда были найдены все возможные перенумерации, либо когда число различных перенумераций атомов углерода превысило заданный порог.

4.1.3. Систематические ошибки при сведении к обобщенной задаче поиска подграфа максимального веса

Особенностью графа атомных переходов является его сложная повторяющаяся структура, возникающая из-за «расслоения» реакций и метаболитов. Получается, что каждый ген и каждый метаболит представлены в графе несколько раз и распределены по всему графу.

Такая структура графа усложняет оценку веса модуля. Например, рассмотрим модуль, получающийся простым расширением метода GAM на такой граф (рисунок 33). Видно, что этот модуль состоит из повторяющихся фрагментов. При этом повторы одного фрагмента могут быть расположены как близко в топологическом смысле, так и далеко. Это влечет систематические ошибки при сведении к задаче $GMWCS$: становится «выгодным» добавлять в модуль сильно регулируемые реакции, которые много раз и достаточ-

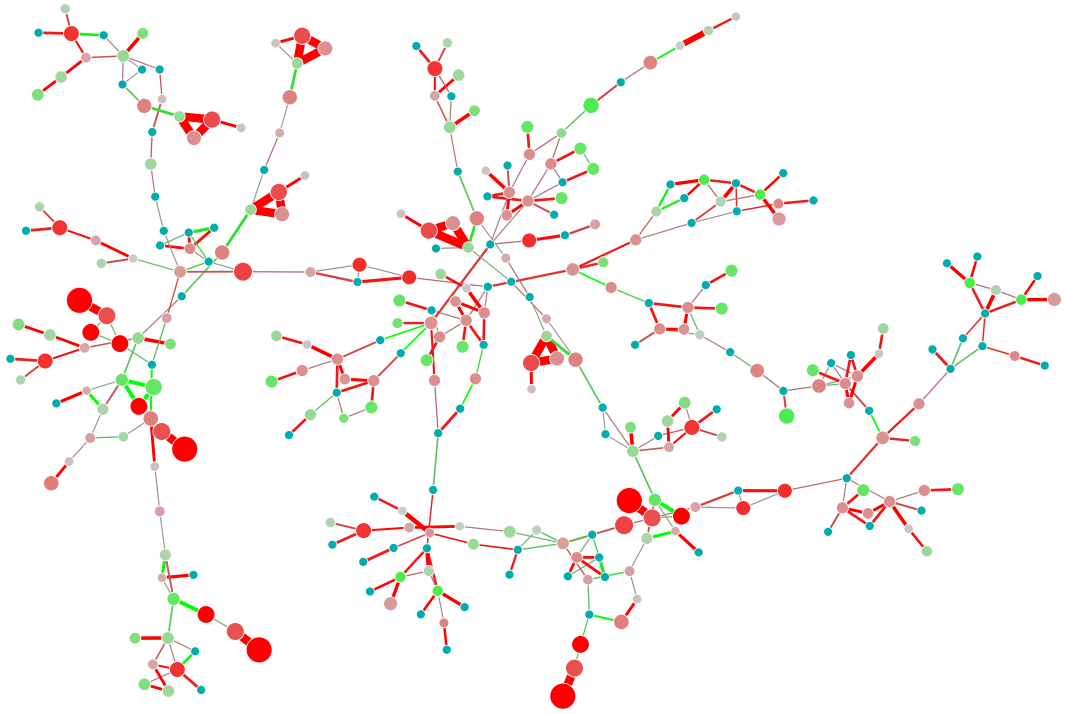


Рисунок 33 – Фрагмент графа атомных переходов с повторяющейся структурой. Цвет и размер вершин и ребер соответствует степени и направлению изменения экспрессии соответствующих веществ или генов

но плотно представлены в графе, причем реакции встречающиеся столько же раз, но менее плотно, брать уже не так выгодно.

4.1.4. Сведение к сигнальному варианту задачи поиска подграфа максимального веса

Для того, чтобы учесть структуру графа введем сигнальный вариант задачи *GWMCS* – *SGMWC*. В нем вместо обычных весов каждой вершине и ребру ставится в соответствие сигнал. Каждому сигналу присваивается вес, причем сигналу с отрицательным весом может соответствовать либо одна вершина, либо одно ребро. Соответственно, для каждого гена (метаболита), для которого в методе *GAM* (раздел 3.2.4) назначался бы положительный вес, вводится один сигнал на все его вхождения. Для генов (метаболитов) с отрицательным весом вводится по отдельному сигналу для каждого вхождения. Вес модуля определяется как сумма всех весов его сигналов без учета повторов – каждый сигнал учитывается в весе не больше одного раза. При та-

кой схеме становится невыгодным добавлять одно и то же вещество несколько раз, что исключает преимущество веществ, для которых есть короткий путь, соединяющий их атомы.

Введем формальное определение этой задачи. Пусть дан связный граф $G = (V, E)$, множество сигналов S, некоторое разбиение вершин и ребер на сигналы $\sigma : (V \cup E) \to S$ и весовая функция на сигналах: $\omega : S \to \mathbb{R}$. Кроме этого, вводится ограничение, что сигналу с отрицательным весом может соответствовать только одна вершина или ребро: $\omega(s) < 0 \implies |\sigma^{-1}(s)| = 1$. Задача *SGMWCS* состоит в поиске связного подграфа $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ с максимальным весом:

$$\Omega(\widetilde{G}) = \sum_{s \in \sigma(\widetilde{V} \cup \widetilde{E})} \omega(s) \to max,$$

где $\sigma(\widetilde{V} \cup \widetilde{E}) = \bigcup_{x \in (\widetilde{V} \cup \widetilde{E})} \sigma(x)$.

При этом отметим, что задача *GMWCS* является частным случаем задачи *SGMWCS*, когда каждому сигналу соответствует ровно одна вершина или одно ребро. Из этого следует, что задача *SGMWCS* так же является *NP*-трудной.

4.2. Решатель для сигнального варианта задачи поиска подграфа максимального веса

Для задачи *SGMWCS* разработан решатель, основанный на решателе для *GMWCS* (раздел 3.3).

Решатель реализован на языке *Java*. Исходный код доступен в открытом доступе под лицензией *MIT* по адресу https://github.com/ctlab/sgmwcs-solver/.

4.2.1. Правила предобработки

Так же, как и в решателе для *GMWCS*, вводятся два правила предобработки, объединяющие некоторые вершины и ребра.

Сначала, для удобства, расширим определение задачи *SGMWCS*, разрешив сопоставление одновременно нескольких сигналов одной вершине или ребру. В дальнейшем будем считать, что функция σ действует из $(V \cup E)$ в 2^S – множество всех подмножеств S. Определение веса подграфа $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ при этом не изменяется, вес так же определяется через сумму весов всех сигналов, присутствующих в подграфе: $\sigma(\widetilde{V} \cup \widetilde{E}) = \bigcup_{x \in (\widetilde{V} \cup \widetilde{E})} \sigma(x)$.

Первое правило предобработки объединяет группы вершин, которые одновременно либо присутствуют, либо отсутствуют в решении. В отличие от решателя для *GMWCS* применяется более строгое правило. Пусть $e = (u, v)$ – такое ребро, что веса всех сигналов в $s = \sigma(e) \cup \sigma(u) \cup \sigma(v)$ неотрицательны. Это условие гарантирует, что если взята хотя бы одна из вершин u и v, то можно взять и другую, не уменьшив веса. Соответственно, эти вершины заменяются на одну вершину w с множеством сигналов $\sigma(w) = s$.

Второе правило, объединяющее цепочки ребер, практически не изменяется. Пусть v – вершина с ровно двумя инцидентными ей ребрами $e_1 = (u, v)$ и $e_2 = (v, w)$. Если веса всех сигналов в $s = \sigma(v) \cup \sigma(e_1) \cup \sigma(e_2)$ неположительны, то эта тройка заменяется на одно ребро $e = (u, w)$ с сигналами $\sigma(e) = s$.

Отметим, что при таких правилах всегда выполняется инвариант, что сигналы для одной вершины или ребра либо все имеют неотрицательный вес, либо все – неположительный.

4.2.2. Метод декомпозиции

Так как из-за наличия сигналов вклад одной части подграфа в общий вес зависит от других его частей, применение декомпозиции из решателя для *GMWCS* становится невозможным. Действительно, возможность такой декомпозиции в решателя для *GMWCS* была обусловлена тем, что можно было заменить компоненту графа, связанную с оставшейся частью графа через точку сочленения, на одну вершину с фиксированным весом. В сигналь-

ном варианте такое невозможно, так как вклад подграфа зависит от других вершин. Поэтому может быть выгодно использовать разные решения внутри компоненты в зависимости от сигналов, присутствующих в других частях графа.

Так же, как и в решателе *GMWCS*, для декомпозиции вводится понятие корневой вариант задачи SGMWCS – *R-SGMWCS* (*Rooted SGMWCS*). В нем множество рассматриваемых подграфов ограничивается теми, которые содержат некоторую фиксированную вершину $r \in V$, называемую корнем.

Декомпозиция заключается в том, что решением задачи *SGMWCS* для графа G является либо решение *R-GMWCS* для графа G с некоторым корнем r, либо решение обычного варианта задачи для одной из связных компонент графа *G'*, получающихся после удаления вершины r из графа G. Действительно, для любой вершины r верно, что либо оптимальное решение ее включает, и тогда оно является решением *R-GMWCS* с r в качестве корня, либо не включает вершину r, и тогда оно полностью содержится в одной из оставшихся компонент связности.

Процедура декомпозиции состоит в последовательном разбиении экземпляра *SGMWCS* на один экземпляр задачи *R-SGMWCS* и несколько экземпляров *SGMWCS* меньшего размера. При этом в качестве корня для *R-GMWCS* рассматриваются только точки сочленения. Из этих точек выбирается такая, что максимальный размер получающихся при разбиении компонент яаляется минимальным. Декомпозиция прекращается в случае, когда граф становится двусвязным и в нем отсутствуют точки сочленение, либо когда размер графа становится меньше некоторого заданного значения, являющегося параметром алгоритма.

4.2.3. Сведение к задаче целочисленного линейного программирования

Так же, как и для *GMWCS*, сведение задачи *SGMWCS* к задаче целочисленного линейного программирования состоит в записи целевой функции и ограничений на связность подграфа. При этом ограничения на связность для *GMWCS* естественным образом подходят и к *SGMWCS*.

Для записи целевой функции вводятся дополнительные бинарные переменные z_s, соответствующие наличию сигнала $s \in S$ в подграфе. При этом вводятся разные ограничения в зависимости от множества элементов, соответствующих данному сигналу $X_s = \sigma^{-1}(s) = \{x \in (V \cup E) \mid s \in \sigma(x)\}$:

$$\begin{aligned} &z_s = y_v, && \text{если } X_s = \{v\} \text{ для } v \in V; \\ &z_s = w_e, && \text{если } X_s = \{e\} \text{ для } e \in E; \\ &z_s \leqslant \sum_{v \in X_s \cap V} y_v + \sum_{e \in X_s \cap E} w_e && \text{в противном случае.} \end{aligned}$$

Целевая функция при этом записывается следующим образом:

$$\sum_{s \in S} \omega(s) z_s \to max.$$

Кроме этого, из-за возникновения большого числа корневых экземпляров было реализовано добавление дополнительных ограничений «на лету». Эти ограничения были адаптированы из [71] для возможности применения к реберно-взвешенной задаче. В результате добавлялись ограничения вида:

$$y_v \leqslant \sum_{e \in C} w_e, \qquad \forall v \in V, C \in C_v^*,$$

где C_v^* — это набор всех r-v разрезов графа. Для поиска нарушенных ограничений выполняется поиск такого r-v разреза C, что $y_v > \sum_{e \in C} w_e$. Поскольку слева стоит константа, то лучшим кандидатом нарушенного ограничения будет то, что соответствует минимальному разрезу. Для поиска минимального разреза используется алгоритм Эдмондса-Карпа.

4.2.4. Использование нескольких потоков выполнения

Для более эффективного использования вычислительных ресурсов был разработан способ распараллеливания разработанного решателя.

Во-первых, естественным образом, получающиеся в результате декомпозиции экземпляры задач *R-SGMWCS* и *SGMWCS* могут решаться в разных потоках. В программе существует пул потоков заданного пользователем размера, в которых запускается решение экземпляров из очереди по мере освобождения потоков.

Во-вторых, была введена общая переменная, в которой поддерживается значение веса наиболее оптимального из найденных решений. Использование этой информации позволяет не обрабатывать ветви дерева решений, верхняя граница на вес в которых меньше уже найденного решения. Такое отсечение позволяет быстрее завершить решение отдельных экземпляров и перейти к следующим.

4.2.5. Поиск реберно-минимального решения

В общем случае у задачи *SGMWCS* существует множество решений, имеющих одинаковый вес. Такая ситуация возникает из-за того, что вес подграфа определяется весом присутствующих в нем сигналов, а один и тот же набор сигналов может соответствовать нескольким подграфам.

В таких ситуациях обычно удобно рассматривать решение с минимальным числом ребер. Для того, чтобы находить такое решение, у решателя есть опция, позволяющая добавить небольшой штраф за каждое дополнительное ребро в решении. При использовании этой опции правила предобработки не используются.

4.3. Экспериментальное исследование

Методика экспериментального исследования метода *GATOM* повторяет методику экспериментального исследования метода *GAM* (раздел 3.5).

Сначала на искусственных данных рассматривается точность метода с точки зрения идентификации отдельных генов и метаболических путей. Затем анализируется работа на реальных данных.

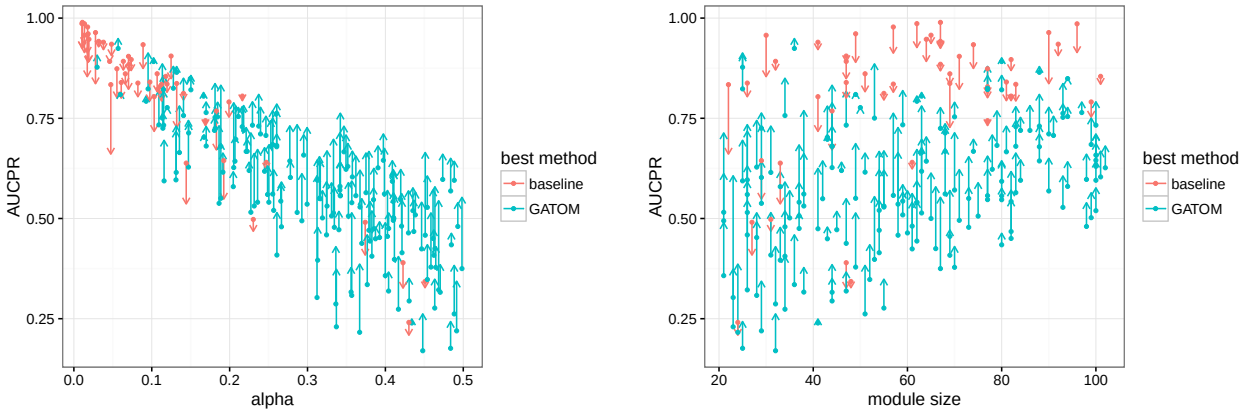
4.3.1. Исследование точности метода на искусственных данных дифференциальной экспрессии генов

Метод генерации искусственных данных повторяет использованный для метода *GAM*. Кратко, сначала берется сеть реакций, возникающая в реальном эксперименте. Затем в этой сети комбинированием метаболических путей из базы *KEGG* конструируется искусственный активный модуль. В конце, исходя из равномерного и бета- распределений, для генов в сети генерируются P-значения.

Так же, как и в предыдущей главе, для генерации искусственных данных была взята сеть для эксперимента *GSE59228* [82]. Так как не для всех реакций известна информация про переходы атомов, в этой сети участвует меньшее число генов и метаболитов по сравнению с методом *GAM*: 358 генов и 645 метаболитов. При этом в сети было 4205 вершин и 4584 ребра. Как и раньше, множество генов в этой сети обозначим как U.

Метод тестировался на 200 сгенерированных наборах данных. Для каждого набора выполнялось до 30 запусков на разных значениях *FDR*-порога.

Сначала была рассмотрена точность нахождения отдельных генов из активного модуля. Были построены зависимости площади под кривой точность-полнота от параметра α и размера сгенерированного модуля (рисунок 34). Так же, как и для метода *GAM*, есть четкая зависимость точности от параметра α (рисунок 34*а*). При увеличении α точность уменьшается, но для метода *GATOM* она уменьшается медленнее и, так же, как и метод *GAM*, начиная со значения около 0,1, он начинает превосходить базовый метод. Также наблюдается небольшая зависимость от размера модуля: метод работает более точно для больших модулей (рисунок 34*б*).



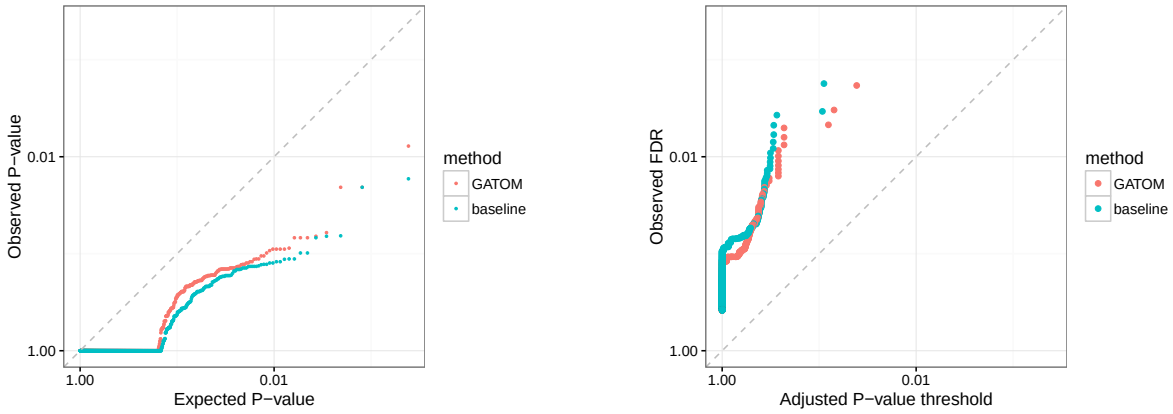
(а) (б)

Рисунок 34 – Зависимость площади под кривой точность-полнота от значения параметра α (а) и размера модуля (б). Каждому тесту соответствует одна стрелка. Ордината начала стрелки соответствует значению площади под кривой для базового метода, ордината конца – для метода *GATOM*. Красным цветом обозначены тесты, в которых точнее классифицирует гены базовый метод, бирюзовым – метод *GATOM*

Затем был проведен анализ точности определения активных метаболических путей с помощью комбинации метода *GATOM* и метода Фишера. Для анализа рассматривались только запуски, для которых в модуле было от 10 до 100 генов и *FDR*-порог был меньше 0,1.

Была проведена проверка, что при таком методе контролируется ошибка первого рода. Для этого было проанализировано распределение P-значений для неактивных метаболических путей. На рисунке 35*а* приведено сравнение этого распределения с равномерным распределением $\mathcal{U}(0,1)$. Из этого сравнения видно, что для метода *GATOM* ошибка первого рода контролируется: все точки лежат ниже диагонали. Процедура идентификации активных метаболических путей для метода *GATOM* с поправкой методом Бенджамина-Хохберга также контролирует и уровень *FDR* (рисунок 35*б*).

Был проведен анализ точности определения активных метаболических путей методом *GATOM*. Кроме сравнения с базовым методом, также выполнялось сравнение с методом *GAM*. Для этого метод *GAM* запускался на том же наборе генов U, что и метод *GATOM*, а модуль выбирался с наи-



(а) (б)

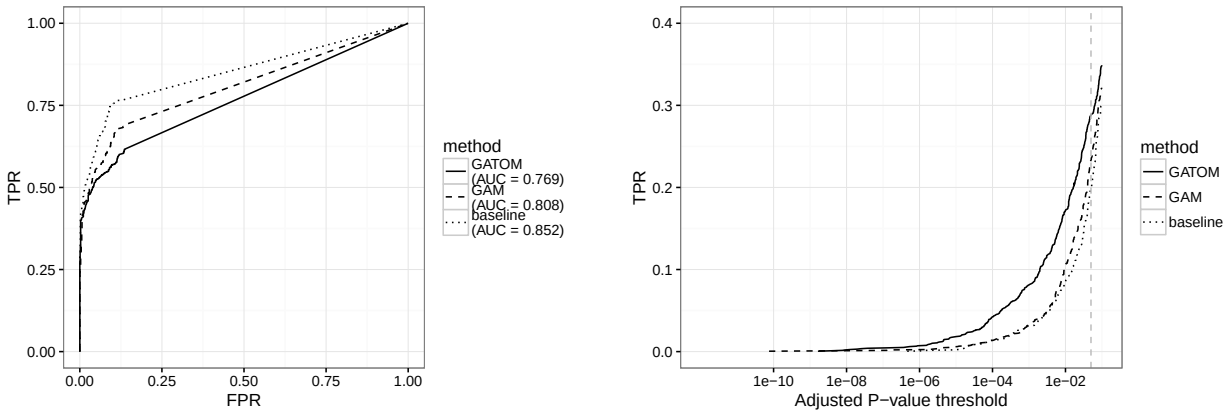
Рисунок 35 – (а) Графики квантиль-квантиль для распределений P-значений при верной нулевой гипотезе и равномерного распределения $\mathcal{U}(0,1)$. (б) График зависимости наблюдаемого уровня FDR от порога на скорректированное P-значение. Кривые для метода $GATOM$ обозначены красным, для базового метода – бирюзовым

более близким размером к модулю $GATOM$. Хотя кривая ошибок определения активных метаболических путей для метода $GATOM$ расположена ниже таковых для метода GAM и для базового метода (рисунок 36$а$), по числу истинно-положительных результатов в зависимости от заданного порога на скорректированное P-значение метода $GATOM$ выигрывает оба других метода (рисунок 36$б$). Для метода $GATOM$ число таких результатов примерно на 50% превышает аналогичное значение для базового метода для значений порога 0,05 и в два раза – для порога 0,01.

4.3.2. Исследование точности работы метода на искусственных данных совместно для генов и метаболитов

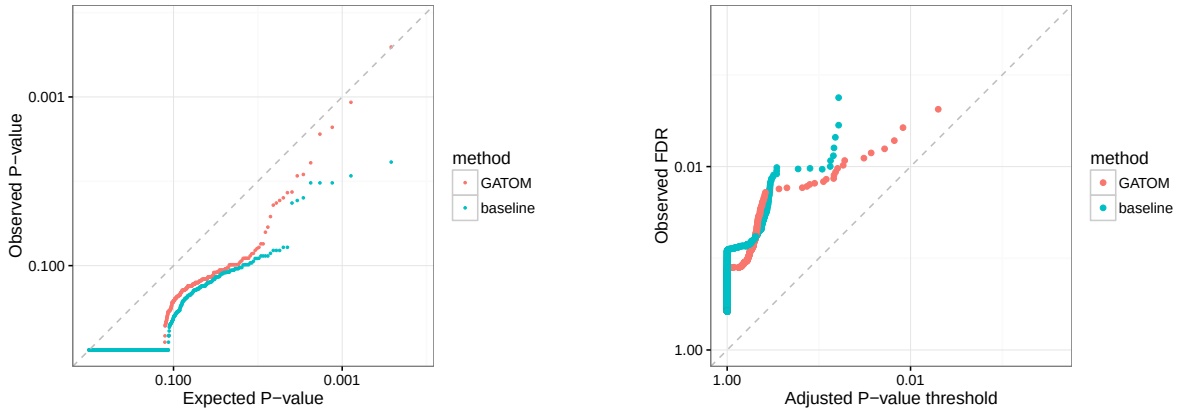
Был исследована точность метода и для совместного анализа сгенерированных транскриптомных и метаболомных данных. Было подтверждено, что комбинация метода $GATOM$ и метода Фишера контролирует уровень ошибки первого рода (рисунок 37).

Результаты анализа числа положительных результатов представлены на рисунке (рисунок 38). Как для площади под кривой ошибок так и для числа положительных результатов в зависимости от порога на скорректированное



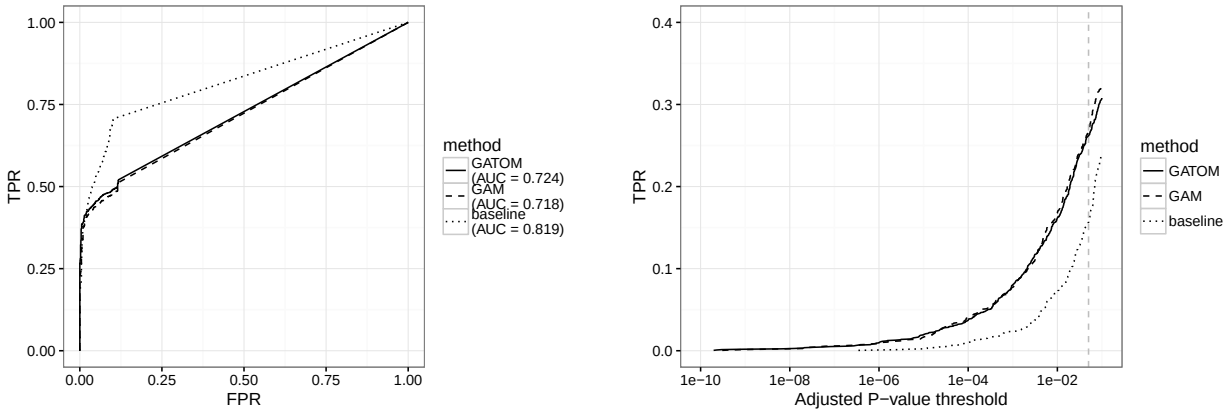
(а) (б)

Рисунок 36 – Кривая ошибок (а) и зависимость TPR от порога на скорректированное P-значение (б) для определения активных метаболических путей по сгенерированным транскриптомным данным. Кривые для метода $GATOM$ обозначены сплошной линией, для метода GAM – крупным пунктиром, для базового метода – мелким пунктиром, вертикальной пунктирной линией отмечено значение порога 0,05



(а) (б)

Рисунок 37 – (а) Графики квантиль-квантиль для распределений P-значений при верной нулевой гипотезе и равномерного распределения $\mathcal{U}(0,1)$. (б) График зависимости наблюдаемого уровня FDR от порога на скорректированное P-значение. Оба графика выполнены для запусков одновременно на транскриптомных и метаболомных данных. Кривые для метода $GATOM$ обозначены красным, для базового метода – бирюзовым



(а) (б)

Рисунок 38 – Кривая ошибок (а) и зависимость TPR от порога на скорректированное P-значение (б) для определения активных метаболических путей при одновременном анализе сгенерированных транскриптомных и метаболомных данных. Кривые для метода $GATOM$ обозначены сплошной линией, для базового метода – пунктиром

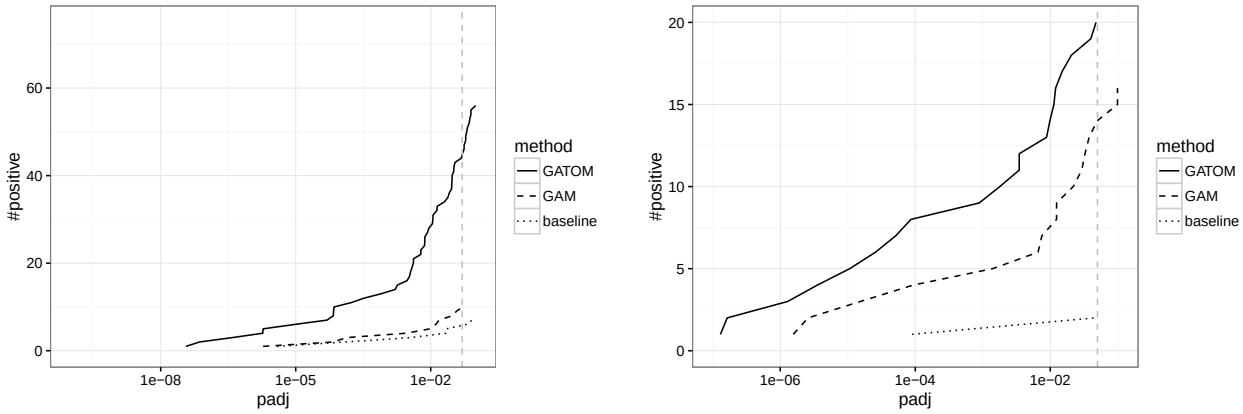
P-значения результаты для методов $GATOM$ и GAM практически не отличаются. Так же, как и раньше, метод $GATOM$ не так хорошо ранжирует метаболические пути, как метод Фишера, но показывает более высокие результаты по второму показателю.

4.3.3. Исследование работы метода на реальных данных

Исследование на реальных данных так же, как и для метода GAM, проводилось на 78 наборах транскриптомных данных и 30 наборах с одновременно транскриптомными и метаболомными данными (раздел 3.5.1).

По аналогичной процедуре был проведен поиск активных метаболических путей. Для метода $GATOM$ число найденных путей в несколько раз превышало число таковых для базового метода и для метода GAM как в случае с использованием только транскриптомных данных (рисунок 39$а$), так и обоих типов данных вместе (рисунок 39$б$).

Из 78 наборов с только транскриптомными данными для 64 существовал модуль подходящего размера. При выборе значения порога на скорректированное P-значение равным 0,05 методом $GATOM$ обнаруживается 44 регулируемых метаболических путей для 33 входных наборов, методом GAM –



(а) (б)

Рисунок 39 – Зависимость числа положительных результатов анализа представленности в зависимости от порога на скорректированное P-значение при запуске только для реальных транскриптомных данных (а) и при одновременном запуске на транскриптомных и метаболомных данных (б). Кривые для метода *GATOM* обозначены сплошной линией, для метода *GAM* – крупным пунктиром, для базового метода – мелким пунктиром, вертикальной пунктирной линией отмечено значение порога 0,05

девять путей для восьми наборов, базовым методом – пять путей для четырех наборов.

При запуске на 30 наборах с одновременно транскриптомными и метаболомными данными, из которых для 23 был найден модуль подходящего размера. Для *FDR*-порога, равного 0,05 методом *GATOM* обнаруживается 20 регулируемых метаболических путей для 13 входных наборов, методом *GAM* – 14 путей для 9 наборов, базовым методом – два пути для двух наборов.

4.4. Пример применения метода для анализа метаболической регуляции в глиоме

Рассмотрим применение метода *GATOM* для анализа влияния мутации в гене *TP53* на метаболизм опухолевых клеток глиомы – рака головного мозга. Ген *TP53* является одним из наиболее часто мутирующих в этом типе рака. В качестве исходных данных взят набор с экспрессией генов *Lower Grade Glioma* из базы данных *TCGA* (*The Cancer Genome Atlas*) [84, 92]. Мета-

боломные данные не использовались. Для получения уровней значимости и направления регуляции отдельных генов был выполнен анализ дифференциальной экспрессии между образцами без мутации в гене *TP53* и с мутацией.

Сначала метод строит граф атомных переходов, в котором вершинам соответствуют атомы углерода отдельных веществ, а ребрам – теоретически возможные реакции. Получающийся граф состоит из 4698 вершин и 5211 ребер. Из-за использования графа атомных переходов практически всегда инцидентность двух ребер означает возможность последовательного выполнения соответствующих реакций.

Как и в методе *GAM*, предполагается, что важные реакции должны быть связаны, и в построенном графе выделяется связный подграф. Из возможных подграфов выбирается один, так, чтобы многие из соответствующих генов обладали значимой индивидуальной регуляцией. Поиск подграфа выполняется с помощью разработанного решателя для задачи *SGMWCS*.

Получившийся подграф представлен на рисунке 40. Так же, как и в методе *GAM*, в этом подграфе можно выделить некоторые стандартные метаболические пути как, например, расщепление глюкозы, цикл Кребса и другие. Используя эти результаты, можно строить гипотезы о роли этих путей в развитии опухолевых клеток.

Важным отличием от метода *GAM* являются более «естественные» соединения реакций в получившемся подграфе. Косвенно это подтверждается присутствием в нем метаболического пути биосинтеза мевалоната: *ACAT2*, *HMGCS1*, *HMGCR*, *MVK*, *PMVK* и *MVD* (выделен на рисунке 40). Во-первых, регуляция этого пути подтверждается статьей Ч. Лаецца и соавторов [85], в которой показывается, что именно мутация в гене *TP53* влечет такие изменения. Во-вторых, этот метаболический путь появляется в подграфе из-за сильного изменения в экспрессии фермента *TRIT1*, субстрат которого в млекопитающих может получаться только через этот метаболический путь. В то же время при использовании обычного графа метаболических ре-

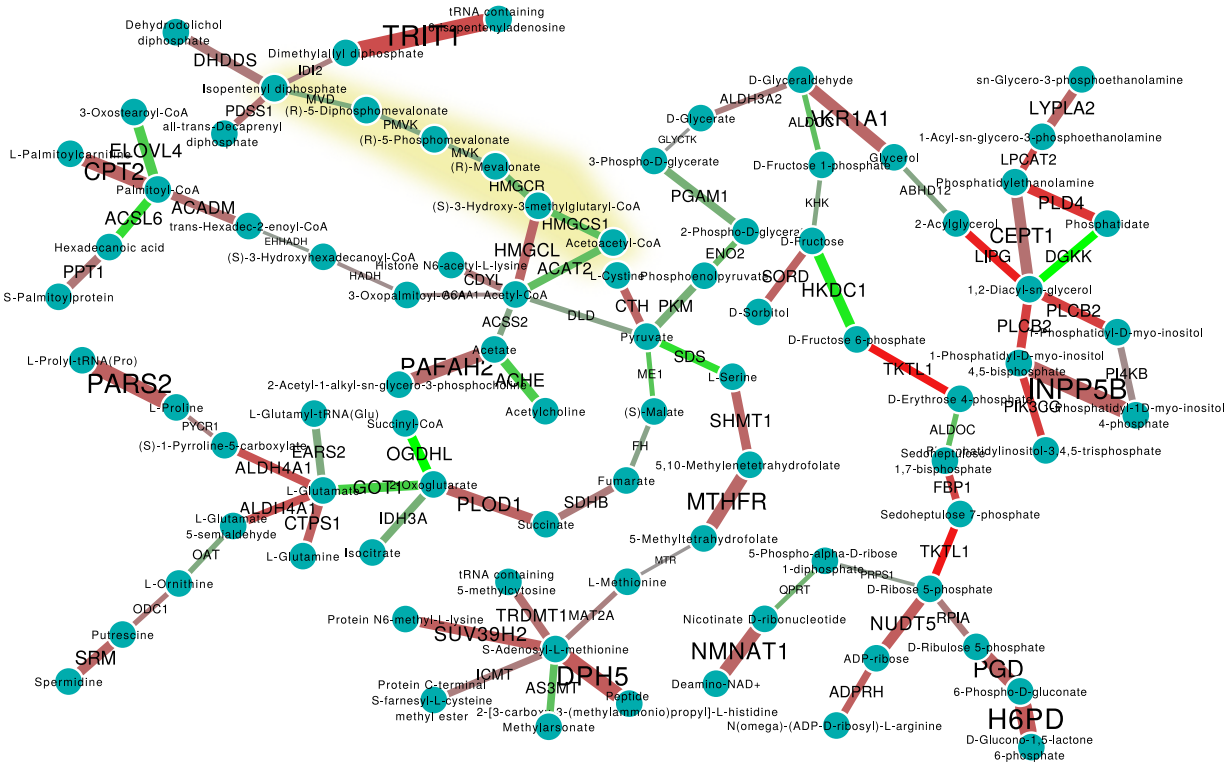


Рисунок 40 – Активный метаболический модуль, найденный с помощью метода *GATOM* при сравнении образцов глиомы с мутацией в гене *TP53* и без нее. В модуле выделен метаболический путь биосинтеза мевалоната

акций находятся другие, опосредованные, способы соединить этот фермент с другими компонентами модуля, не включающие метаболический путь биосинтеза мевалоната. Соответственно, при применении метода *GAM* этот путь не обнаруживается.

Таким образом, предлагаемый метод, как и метод *GAM* позволяет выделить наиболее важные метаболические процессы и их взаимосвязи. При этом использование графа атомных переходов позволяет обнаруживать более тонкие эффекты.

Выводы по главе 4

1. Разработан метод *GATOM* для выделения активного метаболического модуля с помощью анализа графа атомных переходов.
2. Сформулирована задача *SGMWCS*, позволяющая учитывать структуру графа атомных переходов.

3. Для задачи *SGMWCS* разработан практический точный решатель, выложенный в открытый доступ по адресу https://github.com/ctlab/sgmwcs-solver/.
4. Разработанный метод включен в веб-сервис *Shiny GAM*, доступный по адресу http://genome.ifmo.ru/shiny/gam/.
5. Проведено экспериментальное исследование подтверждающее эффективность метода на модельных и экспериментальных данных.
6. Приведен пример, демонстрирующий применение метода к реальным экспериментальным данным.

ЗАКЛЮЧЕНИЕ

В результате диссертационного исследования получены следующие результаты:

1. Разработан метод быстрого взвешенного анализа представленности наборов генов. Он основан на алгоритме для кумулятивного подсчета значения статистики представленности. Метод реализован в виде *R*-пакета *fgsea* и доступен в открытый библиотеке *Bioconductor* (http://bioconductor.org/packages/fgsea).
2. Разработан метод поиска активного метаболического модуля с помощью анализа сети метаболических реакций. Он состоит в представлении сети реакций и исходных данных в виде взвешенного графа и последующего решения задачи поиска связного подграфа максимального веса. Метод доступен для использования в виде веб-сервиса *Shiny GAM* (http://genome.ifmo.ru/shiny/gam/). Для решения обобщенного варианта задачи, возникающего при одновременном наличии данных транскриптомного и метаболомного профилирования, был разработан открытый точный решатель (https://github.com/ctlab/gmwcs-solver/).
3. Разработан метод поиска активного метаболического модуля с помощью анализа графа атомных переходов. Использование графа атомных переходов позволяет избежать некоторых типов соединений и лучше соответствует структуре метаболических путей. Метод также доступен в веб-сервисе *Shiny GAM*. Для учета повторяющейся структуры графа атомных переходов была разработан сигнальный вариант обобщенной задачи поиска связного подграфа максимального веса. Для этого варианта также был разработан открытый точный решатель (https://github.com/ctlab/sgmwcs-solver/).

Поясним результаты, полученные в диссертационной работе. На вход все разработанные методы принимают транскриптомные и метаболомные данные, в которых содержится информация об *индивидуальной* регуляции фер-

ментов (генов) и метаболитов, участвующих в биохимических реакциях. Целью всех трех методов является упрощение *интерпретации* биологом этих данных. Методы направлены на выделение важных в рассматриваемом процессе *метаболических путей* и их взаимосвязей. Это обосновывается тем, что описание метаболических процессов в клетке в терминах метаболических путей является более *высокоуровневым* по сравнению с описанием в терминах отдельных реакций. Методы используют разные уровни описания *метаболических моделей*.

Первый из них (*FGSEA*) работает на уровне описания метаболических путей как множества реакций. В нем рассматриваются стандартные метаболические пути (или пути, указанные в конкретной модели), такие как, например, гликолиз – процесс расщепления глюкозы. Метод позволяет выделить метаболические пути, в которых происходят неслучайные изменения в экспрессии генов. Это может означать, что такой метаболический путь важен в рассматриваемом клеточном процессе. Метод позволяет выделять такие метаболические пути более эффективно по сравнению с существующими методами. При этом его недостатками являются невозможность напрямую понять, как регулируемые пути связаны между собой, и невозможность определения новых метаболических путей.

Второй метод (*GAM*) устраняет указанные выше недостатки за счет использования информации о связях между отдельными реакциями. Он выделяет связный набор реакций так, чтобы соответствующие гены обладали индивидуальной регуляцией. В результате, оказывается, что в этом связном наборе реакций, во-первых, хорошо представлены метаболический пути. Во-вторых, видно, как эти метаболические пути связаны между собой. Отметим также, что метод не использует информацию о стандартных метаболических путях, и поэтому его можно использовать для поиска новых путей. Таким образом, этот метод позволяет интерпретировать данные на высокоуровневом языке метаболических путей и их взаимодействий друг с другом. В качестве

недостатка метода можно выделить, что не все соединения в получающемся наборе реакций могут быть легко интерпретируемыми.

Третий метод ($GATOM$) является развитием предыдущего и позволяет уменьшить число сложноинтерпретируемых соединений. Это обеспечивается за счет использования графа переходов атомов углерода в реакциях. В результате, практически все последовательные соединения реакций в получающемся наборе реакций могут соответствовать их последовательному прохождению в клетке.

СПИСОК ИСТОЧНИКОВ

Печатные издания на русском языке

1. *Ньюсхолм Э.*, *Старт К.* Регуляция метаболизма. — Мир, 1977. — С. 407.

2. *Самсонова М. Г.*, *Суркова С. Ю.*, *Козлов. К. Н.*, *Писарев А. С.* Что и как изучает биоинформатика // Труды Санкт-Петербургского политехнического университета Петра Великого. — 2009. — № 511. — С. 169–190.

3. *Колчанов Н.*, *Игнатьева Е.*, *Подколодная О.*, *Лихошвай В.* [и др.] Генные сети // Вавиловский журнал генетики и селекции. — 2013. — Т. 17, № 4. — С. 833–850.

4. *Назипова Н. Н.*, *Елькин Ю. Е.*, *Панюков В. В.*, *Дроздов-Тихомиров Л. Н.* Расчёт скоростей метаболических реакций в живой растущей клетке методом баланса стационарных метаболических потоков (метод БСМП) // Матем. биология и биоинформ. — 2007. — Т. 2, № 1. — С. 98–119.

5. *Кормен Т. Х.*, *Лейзерсон Ч. И.*, *Ривест Р. Л.*, *Штайн К.* Алгоритмы: построение и анализ. 2-е изд. — М.: Вильямс, 2005. — С. 1296.

6. *Ульянцев В. И.* Генерация конечных автоматов с использованием программных средств решения задач выполнимости и удовлетворения ограничений. Диссертация на соискание ученой степени кандидата технических наук. Университет ИТМО. — 2015. — URL: http://is.ifmo.ru/disser/ulyantsev-dissertation.pdf.

Печатные издания на английском языке

7. *Chandel N.* Navigating Metabolism. — Cold Spring Harbor Laboratory Press, 2015. — P. 264.

8. *Cairns R. A.*, *Harris I. S.*, *Mak T. W.* Regulation of cancer cell metabolism. // Nat Rev Cancer. — 2011. — Vol. 11, no. 2. — Pp. 85–95.

9. *Mathis D.*, *Shoelson S. E.* Immunometabolism: an emerging frontier. // Nat Rev Immunol. — 2011. — Vol. 11, no. 2. — P. 81.

10. *Metallo C. M.*, *Vander Heiden M. G.* Understanding metabolic regulation and its influence on cell physiology // Mol. Cell. — 2013. — Vol. 49, no. 3. — Pp. 388–398.

11. *Li C.*, *Wong W. H.* Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection // Proc. Natl. Acad. Sci. U.S.A. — 2001. — Vol. 98, no. 1. — Pp. 31–36.

12. *Wang Z.*, *Gerstein M.*, *Snyder M.* RNA-Seq: a revolutionary tool for transcriptomics // Nat. Rev. Genet. — 2009. — Vol. 10, no. 1. — Pp. 57–63.

13. *Fuhrer T.*, *Zamboni N.* High-throughput discovery metabolomics // Curr. Opin. Biotechnol. — 2015. — Vol. 31. — Pp. 73–78.

14. *Love M. I.*, *Huber W.*, *Anders S.* Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. // Genome Biol. — 2014. — Vol. 15, no. 12. — P. 550.

15. *Ritchie M. E.*, *Phipson B.*, *Wu D.*, *Hu Y.*, [et al.] limma powers differential expression analyses for RNA-sequencing and microarray studies // Nucleic Acids Research. — 2015. — gkv007.

16. *Thiele I.*, *Palsson B. O.* A protocol for generating a high-quality genome-scale metabolic reconstruction // Nat Protoc. — 2010. — Vol. 5, no. 1. — Pp. 93–121.

17. *Hucka M.*, *Finney A.*, *Sauro H. M.*, *Bolouri H.*, [et al.] The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models // Bioinformatics. — 2003. — Vol. 19, no. 4. — Pp. 524–531.

18. *Kanehisa M.* KEGG: Kyoto Encyclopedia of Genes and Genomes // Nucleic Acids Research. — 2000. — Vol. 28, no. 1. — Pp. 27–30.

19. *Kanehisa M.*, *Goto S.*, *Sato Y.*, *Furumichi M.*, [et al.] KEGG for integration and interpretation of large-scale molecular data sets. // Nucleic Acids Res. — 2012. — Vol. 40, Database issue. — Pp. D109–D114.

20. *Caspi R.*, *Altman T.*, *Dale J. M.*, *Dreher K.*, [et al.] The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases // Nucleic Acids Res. — 2010. — Vol. 38, Database issue. — Pp. D473–D479.

21. *Le Novere N.*, *Bornstein B.*, *Broicher A.*, *Courtot M.*, [et al.] BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems // Nucleic Acids Res. — 2006. — Vol. 34, Database issue. — Pp. D689–691.

22. *Chelliah V.*, *Juty N.*, *Ajmera I.*, *Ali R.*, [et al.] BioModels: ten-year anniversary // Nucleic Acids Res. — 2015. — Vol. 43, Database issue. — Pp. D542–548.

23. *Croft D.*, *O'Kelly G.*, *Wu G.*, *Haw R.*, [et al.] Reactome: a database of reactions, pathways and biological processes // Nucleic Acids Res. — 2011. — Vol. 39, Database issue. — Pp. D691–D697.

24. *Steuer R.*, *Junker B. H.* Computational models of metabolism: stability and regulation in metabolic networks // Advances in chemical physics. — 2009. — Vol. 142. — P. 105.

25. *Orth J. D.*, *Thiele I.*, *Palsson B. Ø.* What is flux balance analysis? // Nature Biotechnology. — 2010. — Vol. 28, no. 3. — Pp. 245–248.

26. *Blazier A. S.*, *Papin J. A.* Integration of expression data in genome-scale metabolic network reconstructions // Front Physiol. — 2012. — Vol. 3. — P. 299.

27. *Lee S. Y.*, *Lee D.-Y.*, *Kim T. Y.* Systems biotechnology for strain improvement // Trends in biotechnology. — 2005. — Vol. 23, no. 7. — Pp. 349–358.

28. *Ghaffari P.*, *Mardinoglu A.*, *Nielsen J.* Cancer metabolism: a modeling perspective // Front Physiol. — 2015. — Vol. 6. — P. 382.

29. *Bordbar A.*, *Mo M. L.*, *Nakayasu E. S.*, *Schrimpe-Rutledge A. C.*, [et al.] Model-driven multi-omic data analysis elucidates metabolic immunomodulators of macrophage activation. // Molecular systems biology. — 2012. — Vol. 8, no. 1. — P. 558.

30. *Karnovsky A.*, *Weymouth T.*, *Hull T.*, *Tarcea V. G.*, [et al.] Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data // Bioinformatics. — 2012. — Vol. 28, no. 3. — Pp. 373–380.

31. *Landesfeind M.*, *Kaever A.*, *Feussner K.*, *Thurow C.*, [et al.] Integrative study of Arabidopsis thaliana metabolomic and transcriptomic data with the interactive MarVis-Graph software // PeerJ. — 2014. — Vol. 2. — e239.

32. *Evans C. R.*, *Karnovsky A.*, *Kovach M. A.*, *Standiford T. J.*, [et al.] Untargeted LC-MS metabolomics of bronchoalveolar lavage fluid differentiates acute respiratory distress syndrome from health // J. Proteome Res. — 2014. — Vol. 13, no. 2. — Pp. 640–649.

33. *Patil K. R.*, *Nielsen J.* Uncovering transcriptional regulation of metabolism by using metabolic network topology // Proceedings of the National Academy of Sciences. — 2005. — Vol. 102, no. 8. — Pp. 2685–2689.

34. *Subramanian A.*, *Tamayo P.*, *Mootha V. K.*, *Mukherjee S.*, [et al.] Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. // Proceedings of the National Academy of Sciences of the United States of America. — 2005. — Vol. 102, no. 43. — Pp. 15545–15550.

35. *Sun H.*, *Wang H.*, *Zhu R.*, *Tang K.*, [et al.] iPEAP: integrating multiple omics and genetic data for pathway enrichment analysis // Bioinformatics. — 2014. — Vol. 30, no. 5. — Pp. 737–739.

36. *Xia J.*, *Psychogios N.*, *Young N.*, *Wishart D. S.* MetaboAnalyst: a web server for metabolomic data analysis and interpretation // Nucleic Acids Res. — 2009. — Vol. 37, Web Server issue. — W652–W660.

37. *Xia J.*, *Sinelnikov I. V.*, *Han B.*, *Wishart D. S.* MetaboAnalyst 3.0–making metabolomics more meaningful // Nucleic Acids Res. — 2015. — Vol. 43, W1. — W251–W257.

38. *Wiechert W.* 13C metabolic flux analysis // Metab. Eng. — 2001. — Vol. 3, no. 3. — Pp. 195–206.

39. *Zamboni N.* 13C metabolic flux analysis in complex systems // Curr. Opin. Biotechnol. — 2011. — Vol. 22, no. 1. — Pp. 103–108.

40. *Suthers P. F.*, *Burgard A. P.*, *Dasika M. S.*, *Nowroozi F.*, [et al.] Metabolic flux elucidation for large-scale models using 13C labeled isotopes // Metab. Eng. — 2007. — Vol. 9, 5-6. — Pp. 387–405.

41. *Pitkanen E.*, *Jouhten P.*, *Rousu J.* Inferring branching pathways in genome-scale metabolic networks // BMC Syst Biol. — 2009. — Vol. 3. — P. 103.

42. *Heath A. P.*, *Bennett G. N.*, *Kavraki L. E.* Finding metabolic pathways using atom tracking // Bioinformatics. — 2010. — Vol. 26, no. 12. — Pp. 1548–1555.

43. *Huang D. W.*, *Sherman B. T.*, *Lempicki R. A.* Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. // Nucleic acids research. — 2009. — Vol. 37, no. 1. — Pp. 1–13.

44. *Maciejewski H.* Gene set analysis methods: statistical models and methodological differences // Brief. Bioinformatics. — 2014. — Vol. 15, no. 4. — Pp. 504–518.

45. *Tarca A. L.*, *Bhatti G.*, *Romero R.* A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity // PLoS ONE. — 2013. — Vol. 8, no. 11. — e79217.

46. *Yoav Benjamini Y. H.* Controlling the false discovery rate: a practical and powerful approach to multiple testing // Journal of the Royal Statistical Society. Series B (Methodological). — 1995. — Vol. 57, no. 1. — Pp. 289–300.

47. *Mootha V. K.*, *Lindgren C. M.*, *Eriksson K. F.*, *Subramanian A.*, [et al.] PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes // Nat. Genet. — 2003. — Vol. 34, no. 3. — Pp. 267–273.

48. *Varemo L.*, *Nielsen J.*, *Nookaew I.* Enriching the gene set analysis of genome-wide data by incorporating directionality of gene expression and combining statistical hypotheses and methods // Nucleic Acids Res. — 2013. — Vol. 41, no. 8. — Pp. 4378–4391.

49. *Yu G.*, *Wang L. G.*, *Yan G. R.*, *He Q. Y.* DOSE: an R/Bioconductor package for disease ontology semantic and enrichment analysis // Bioinformatics. — 2015. — Vol. 31, no. 4. — Pp. 608–609.

50. *Hundt C.*, *Hildebrandt A.*, *Schmidt B.* rapidGSEA: Speeding up gene set enrichment analysis on multi-core CPUs and CUDA-enabled GPUs // BMC Bioinformatics. — 2016. — Vol. 17, no. 1. — P. 394.

51. *Larson J. L.*, *Owen A. B.* Moment based gene set tests // BMC Bioinformatics. — 2015. — Vol. 16. — P. 132.

52. *Wu D.*, *Smyth G. K.* Camera: a competitive gene set test accounting for inter-gene correlation // Nucleic Acids Res. — 2012. — Vol. 40, no. 17. — e133.

53. *Alexa A.*, *Rahnenfuhrer J.*, *Lengauer T.* Improved scoring of functional groups from gene expression data by decorrelating GO graph structure // Bioinformatics. — 2006. — Vol. 22, no. 13. — Pp. 1600–1607.

54. *Huang D. W.*, *Sherman B. T.*, *Lempicki R. A.* Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources // Nat Protoc. — 2008. — Vol. 4, no. 1. — Pp. 44–57.

55. *Blake J. A.*, *Christie K. R.*, *Dolan M. E.*, *Drabkin H. J.*, [et al.] Gene Ontology consortium: going forward // Nucleic Acids Res. — 2015. — Vol. 43, Database issue. — Pp. D1049–D1056.

56. *Ideker T.*, *Ozier O.*, *Schwikowski B.*, *Siegel A. F.* Discovering regulatory and signalling circuits in molecular interaction networks. // Bioinformatics (Oxford, England). — 2002. — Vol. 18 Suppl 1. — S233–S240.

57. *Kirkpatrick S.*, *Gelatt C. D.*, *Vecchi M. P.* Optimization by simulated annealing // Science. — 1983. — Vol. 220, no. 4598. — Pp. 671–680.

58. *Shannon P.*, *Markiel A.*, *Ozier O.*, *Baliga N. S.*, [et al.] Cytoscape: a software environment for integrated models of biomolecular interaction networks // Genome Res. — 2003. — Vol. 13, no. 11. — Pp. 2498–2504.

59. *Mardinoglu A.*, *Gatto F.*, *Nielsen J.* Genome-scale modeling of human metabolism – a systems biology approach // Biotechnology Journal. — 2013. — Vol. 8, no. 9. — Pp. 985–996.

60. *Mardinoglu A.*, *Nielsen J.* New paradigms for metabolic modeling of human cells // Curr. Opin. Biotechnol. — 2015. — Vol. 34. — Pp. 91–97.

61. *Dittrich M. T.*, *Klau G. W.*, *Rosenwald A.*, *Dandekar T.*, [et al.] Identifying functional modules in protein-protein interaction networks: an integrated exact approach. // Bioinformatics. — 2008. — Vol. 24, no. 13. — Pp. i223–i231.

62. *Beisser D.*, *Klau G. W.*, *Dandekar T.*, *Muller T.*, [et al.] BioNet: an R-package for the functional analysis of biological networks // Bioinformatics. — 2010. — Vol. 26, no. 8. — Pp. 1129–1130.

63. *Pounds S.*, *Morris S. W.* Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values // Bioinformatics. — 2003. — Vol. 19, no. 10. — Pp. 1236–1242.

64. *Ljubić I.*, *Weiskircher R.*, *Pferschy U.*, *Klau W. G.*, [et al.] An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem // Mathematical Programming. — 2006. — Vol. 105, no. 2. — Pp. 427–449.

65. *Beisser D.*, *Brunkhorst S.*, *Dandekar T.*, *Klau G. W.*, [et al.] Robustness and accuracy of functional modules in integrated network analysis. // Bioinformatics (Oxford, England). — 2012. — Vol. 28, no. 14. — Pp. 1887–1894.

66. *Beisser D.*, *Grohme M. A.*, *Kopka J.*, *Frohme M.*, [et al.] Integrated pathway modules using time-course metabolic profiles and EST data from Milnesium tardigradum // BMC Syst Biol. — 2012. — Vol. 6. — P. 72.

67. *McClellan E. A.*, *Moerland P. D.*, *Spek P. J. van der*, *Stubbs A. P.* NetWeAvers: an R package for integrative biological network analysis with mass spectrometry data // Bioinformatics. — 2013. — Vol. 29, no. 22. — Pp. 2946–2947.

68. *Pons P.*, *Latapy M.* Computing communities in large networks using random walks // International Symposium on Computer and Information Sciences. — Springer. 2005. — Pp. 284–293.

69. *Alcaraz N.*, *Friedrich T.*, *Kotzing T.*, *Krohmer A.*, [et al.] Efficient key pathway mining: combining networks and OMICS data // Integr Biol (Camb). — 2012. — Vol. 4, no. 7. — Pp. 756–764.

70. *Alcaraz N.*, *Pauling J.*, *Batra R.*, *Barbosa E.*, [et al.] KeyPathwayMiner 4.0: condition-specific pathway analysis by combining multiple omics studies and networks with Cytoscape. // BMC systems biology. — 2014. — Vol. 8, no. 1. — P. 99.

71. *Álvarez-Miranda E.*, *Ljubić I.*, *Mutzel P.* The maximum weight connected subgraph problem // Facets of Combinatorial Optimization. — Springer, 2013. — Pp. 245–270.

72. *El-Kebir M.*, *Klau G. W.* Solving the maximum-weight connected subgraph problem to optimality. — eprint: 1409.5308. — URL: arXiv:1409.5308.

73. *Haouari M.*, *Maculan N.*, *Mrad M.* Enhanced compact models for the connected subgraph problem and for the shortest path problem in digraphs with negative cycles // Computers & Operations Research. — 2013. — Vol. 40, no. 10. — Pp. 2485–2492.

74. *Gomory R. E.* Solving linear programming problems in integers // Combinatorial Analysis. — 1960. — Vol. 10. — Pp. 211–215.

75. *Sherali H. D.*, *Adams W. P.* A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems // SIAM Journal on Discrete Mathematics. — 1990. — Vol. 3, no. 3. — Pp. 411–430.

76. *Sherali H. D.*, *Adams W. P.* A hierarchy of relaxations and convex hull characterizations for mixed-integer zeroone programming problems // Discrete Applied Mathematics. — 1994. — Vol. 52, no. 1. — Pp. 83–106.

77. *Phipson B.*, *Smyth G. K.* Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn // Stat Appl Genet Mol Biol. — 2010. — Vol. 9. — P. 39.

78. *Wei G.*, *Wei L.*, *Zhu J.*, *Zang C.*, [et al.] Global mapping of H3K4me3 and H3K27me3 reveals specificity and plasticity in lineage fate determination of differentiating CD4+ T cells. // Immunity. — 2009. — Vol. 30, no. 1. — Pp. 155–167.

79. *Joshi-Tope G.*, *Gillespie M.*, *Vastrik I.*, *D'Eustachio P.*, [et al.] Reactome: a knowledgebase of biological pathways. // Nucleic acids research. — 2005. — Vol. 33, Database issue. — Pp. D428–D432.

80. *Dunnett C. W.* A multiple comparison procedure for comparing several treatments with a control // Journal of the American Statistical Association. — 1955. — Vol. 50, no. 272. — Pp. 1096–1121.

81. *Ulyantsev V.*, *Zakirzyanov I.*, *Shalyto A.* BFS-based symmetry breaking predicates for DFA identification // LATA. — Springer, 2015. — Pp. 611–622.

82. *Enzo E.*, *Santinon G.*, *Pocaterra A.*, *Aragona M.*, [et al.] Aerobic glycolysis tunes YAP/TAZ transcriptional activity // EMBO J. — 2015. — Vol. 34, no. 10. — Pp. 1349–1370.

83. *Biswas S. K.*, *Mantovani A.* Orchestration of metabolism by macrophages // Cell Metab. — 2012. — Vol. 15, no. 4. — Pp. 432–437.

84. *Brat D. J.*, *Verhaak R. G.*, *Aldape K. D.*, *Yung W. K.*, [et al.] Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas // N. Engl. J. Med. — 2015. — Vol. 372, no. 26. — Pp. 2481–2498.

85. *Laezza C.*, *D'Alessandro A.*, *Di Croce L.*, *Picardi P.*, [et al.] p53 regulates the mevalonate pathway in human glioblastoma multiforme // Cell Death Dis. — 2015. — Vol. 6. — e1909.

Ресурсы сети Интернет

86. KEGG: Kyoto Encyclopedia of Genes and Genomes. — URL: http://www.genome.jp/kegg/.

87. GSEA — Downloads. — URL: http://software.broadinstitute.org/gsea/downloads.jsp.

88. gseapy 0.6.2: Gene Set Enrichment Analysis in Python. — URL: https://pypi.python.org/pypi/gseapy.

89. 11th DIMACS Implementation Challenge. — URL: http://dimacs11.zib.de/.

90. *Иванов М.* Sqrt-декомпозиция. — URL: http://e-maxx.ru/algo/sqrt_decomposition.

91. IBM CPLEX Optimizer. — URL: https://www.ibm.com/software/commerce/optimization/cplex-optimizer/.

92. Lower Grade Glioma — TCGA. — URL: http://cancergenome.nih.gov/cancersselected/lowergradeglioma.

ПУБЛИКАЦИИ АВТОРА ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в журналах из перечня ВАК

93. ***Сергушичев А. А.*** Алгоритм кумулятивного вычисления статистики представленности набора генов // Научно-технический вестник информационных технологий, механики и оптики. — 2016. — Т. 16, № 5. — С. 956–959.

Публикации в рецензируемых изданиях, индексируемых Web of Science или Scopus

94. *Loboda A. A.*, *Artyomov M. N.*, ***Sergushichev A. A.*** Solving generalized maximum-weight connected subgraph problem for network enrichment analysis // Algorithms in Bioinformatics: 16th International Workshop, WABI 2016, Proceedings. — Springer Int. Pub., 2016. — Pp. 210–221.

95. *Izreig S.*, *Samborska B.*, *Johnson R. M.*, ***Sergushichev A.***, [et al.] The miR-17-92 microRNA cluster is a global regulator of tumor metabolism // Cell Rep. — 2016. — Vol. 16, no. 7. — Pp. 1915–1928.

96. ***Sergushichev A. A.***, *Loboda A. A.*, *Jha A. K.*, *Vincent E. E.*, [et al.] GAM: a web-service for integrated transcriptional and metabolic network analysis // Nucleic Acids Research. — 2016. — Vol. 44, W1. — W194–W200.

97. *Lampropoulou V.*, ***Sergushichev A.***, *Bambouskova M.*, *Nair S.*, [et al.] Itaconate links inhibition of succinate dehydrogenase with macrophage metabolic remodeling and regulation of inflammation // Cell Metab. — 2016. — Vol. 24, no. 1. — Pp. 158–166.

98. *Vincent E. E.*, ***Sergushichev A. A.***, *Griss T.*, *Gingras M.*, [et al.] Mitochondrial phosphoenolpyruvate carboxykinase regulates metabolic adaptation and enables glucose-independent tumor growth // Molecular Cell. — 2015. — Vol. 60, no. 2. — Pp. 195–207.

99. *Jha A. K.*, *Huang S. C.*, ***Sergushichev* A. A.**, *Lampropoulou V.*, [et al.] Network integration of parallel metabolic and transcriptional data reveals metabolic modules that regulate macrophage polarization // Immunity. — 2015. — Vol. 42, no. 3. — Pp. 419–430.

Другие публикации

100. ***Sergushichev A.*** An algorithm for fast preranked gene set enrichment analysis using cumulative statistic calculation // bioRxiv. — 2016. — DOI: 10.1101/060012. — URL: http://biorxiv.org/content/early/2016/06/20/060012.

101. ***Сергушичев А. А.*** Алгоритм для быстрого анализа перепредставленности генов // Всероссийская научная конференция по проблемам информатики *СПИСОК*. — СПб. : ВВМ. СПбГУ, 2016. — С. 517–524.