***Доклады Международной конференции "Математическая биология и биоинформатика"**. Под ред. В.Д. Лахно. Том 7. Пущино: ИМПБ РАН, 2018. Статья № e50. doi: 10.17537/icmbb18.104*

Создание Web-сервера для получения сборки генома *de novo* на основе объединения результатов, полученных различными программами-сборщиками

Романенков К.В.[1], Тюльбашева Г.Э.[2], Устинин М.Н.[2], Назипова Н.Н.[2]

[1]*Сколковский институт науки и технологий*
[2]*ИМПБ РАН – филиал ИПМ им. М.В. Келдыша РАН*

В настоящее время существуют различные технологические платформы для секвенирования геномов. У каждой из этих технологий есть свои ограничения по длине и количеству прочтений, по цене, по доступности программного обеспечения и другим параметрам. На выходе у всех современных машин для секвенирования получаются наборы коротких прочтений, которые многократно покрывают секвенируемый геном. Реконструкция геномной последовательности путем анализа взаимопересечений огромного количества прочтений называется сборкой генома. Поскольку эффективного решения задача сборки генома не имеет, ассемблеры используют различного рода алгоритмы аппроксимации для снижения избыточности ребер в графах, способы исправления ошибок секвенирования, эвристические подходы к снижению сложности структуры с целью удлинения простых путей и другие способы упрощения графов. Из-за многообразия алгоритмов сборки, каждый из которых имеет свои сильные и слабые стороны, но ни один не выдает единственный и правильный ответ, задача сборки генома *de novo* остается очень сложной. В работе описывается новый подход к сборке *de novo* целевого генома, основанный на использовании результатов черновых сборок генома разными ассемблерами. Разработанный ресурс встроен в информационно-вычислительную среду Mathcell (http://www.mathcell.ru/model8.php), которая легко позволяет интегрировать различные расчетные модели, предоставляя web-интерфейс для задания параметров алгоритма, проведения расчетов с использованием мощностей выделенного узла вычислительного кластера ИМПБ РАН, а также отложенного по времени получения, представления и интерпретации результатов. Сервер GAR, предоставляющий пользователям возможность объединения результатов работы минимально трех и максимально десяти сборщиков геномных последовательностей, позволяет эффективно использовать преимущества и исправлять ошибки отдельных алгоритмов для получения адекватной безреференсной сборки целевого генома. Исходный код программы GAR находится в свободном доступе: https://bitbucket.org/kromanenkov/gar.

Ключевые слова: *секвенирование геномов, NGS, сборка генома de novo, контиги, выравнивание контигов; граф контигов, кластеризация.*

Creation of a web server for *de novo* genome assembly on the basis of combining the results obtained by various assemblers

Romanenkov K.V.[1], Tyulbasheva G.E.[2], Ustinin M.N.[2], Nazipova N.N.[2]

[1]*Skolkovo Institute of Science and Technology*
[2]*IMPB RAS – Branch of KIAM RAS*

A number of highly effective technologies for genome sequencing exist nowadays. Each of them has its own limitations on the length and number of reads, price, software availability and other parameters. Modern sequencers produce hundreds of billions short reads which repeatedly cover the sequenced genome. The process of the genomic sequence reconstruction through the analysis of reads intersection is called genome assembly. Since the problem of genome assembly does not have an effective solution, assemblers use various approximation algorithms to reduce the redundancy of edges in graphs, special methods for correcting sequencing errors, heuristic approaches for reducing the structure complexity for simple paths extension, and other graph simplification methods. Due to the variety of genome assemblers, each having its own strengths and weaknesses but none yielding a single and correct answer, the task of the *de novo* genome assembly remains rather complicated. The

paper describes a new GAR (Genome Assembly Refinement) approach for merging two or more assemblies without relying on a reference genome. The developed resource is incorporated in the infrastructure of Mathcell Portal (http://www.mathcell.ru/model8.php). The flexible structure of Mathcell allows easy integration for various calculation models, providing a web interface for specifying the algorithm's parameters, launching computations on the dedicated node of the IMPB RAS cluster, receiving asynchronous response from the server as well as presentation and interpretation of results. The GAR server allows users to merge from three to ten genome assemblies and by taking the best parts from individual assemblies provides the adequate *de novo* target genome assembly. The source code for the GAR program is available at: https://bitbucket.org/kromanenkov/gar.

Key words: genome sequencing, NGS, de novo genome assembly, contig, alignment of contigs; contig graph, clustering.

Новые технологии секвенирования (NGS), появившиеся в конце первого десятилетия 21-го века, более чем на четыре порядка снизили стоимость прочтения одного генома по сравнению с методикой Сэнгера, применявшейся в конце 20-го века. С появлением высокотехнологичных методов секвенирования, позволяющих за несколько дней получить сотни миллиардов нуклеотидов, из которых состоит любой геном, появился целый ряд прикладных областей биомедицины, которые занимаются исследованием индивидуальных геномов.

Машины для NGS (секвенаторы) позволяют параллельно обрабатывать миллионы фрагментов генома, повторяя сотни раз однотипные операции, каждая из которых приводит к получению больших массивов данных для анализа. Для их обработки каждый секвенатор снабжается мощным серверным оборудованием, которое помогает решать задачи собственно прочтения до сотен миллиардов нуклеотидов в час, в результате чего биологи получают последовательности ДНК (так называемые прочтения, риды, reads) определенной длины. Своим появлением технологии NGS обязаны открытию полимеразной цепной реакции (ПЦР, PCR) и автоматизации основных этапов чтения ДНК и основываются на распараллеливании процесса чтения ДНК.

Существуют различные технологические платформы и различные производители машин для секвенирования [1]. Основными платформами NGS второго поколения, которые распространены в последнее время, являются разработки Illumina Inc., Thermo Fisher Scientific и Roche. У каждой из этих технологий есть свои ограничения по длине и количеству прочтений, по цене, по доступности программного обеспечения и другим параметрам. Современные секвенаторы относительно дешево обеспечивают исследователей короткими прочтениями (от 100 до 400 н.п. и короче). Но это сильно осложняет задачу сборки целой хромосомы, потому что короткие фрагменты несут меньше информации. Для хорошей сборки необходимо иметь достаточную глубину покрытия *n* (*n*-fold coverage) исследуемого образца прочтениями, т.е. среднее количество прочтений, в которые входит каждый нуклеотид целевого генома, должно быть достаточно большим. Однако, чем глубже покрытие, тем сложнее вычислительная задача оперирования этими большими данными в процессе сборки. Качество получаемой сборки зависит также от GC-состава целевой ДНК, обогащенности простыми последовательностями, от факта, что все платформы в большей или меньшей степени имеют снижение точности определения нуклеотидов в районе 3'-концов ридов. Кроме того, у каждой платформы есть свои характерные технологические ошибки и разные вероятности ошибок (от 0.1 % до 10 %).

Сборка – это иерархическая структура данных, которая наносит данные секвенирования на предполагаемую реконструкцию целевой последовательности. Она группирует прочтения в контиги (contigs, сокращение от contiguous), а контиги – в скэффолды (scaffolds). Контигом называется непрерывный участок, являющийся консенсусной последовательностью набора ридов, которые взаимно-пересекаются на заданную минимальную величину пересечения. Скэффолд, называемый еще суперконтигом или метаконтигом, определяет участок целевой последовательности, про который известны порядок и ориентация расположения на нем контигов и расстояния между ними. Кроме контигов и скэффолдов в сборке присутствуют отдельные прочтения. Сборки оцениваются размерами контигов и скэффолдов, учитываются максимальная длина, средняя длина, общая длина, а также параметры N50, N90, NG50, NG90. N50 (N90) – это наибольшая длина контига такая, что в контигах не меньшей длины содержится 50 (90) процентов суммарной длины контигов. NG50 (NG90) – наибольшая длина контига такая, что в контигах не меньшей длины содержится 50 (90) процентов суммарной длины генома. Аналогично для скэффолдов.

Различают два типа сборки геномов – референсная и безреференсная. Первая – это сборка генома, для которого существует секвенированный геном особи того же или родственного вида. Алгоритм сборки основан на попарных выравниваниях ридов и референсного генома. Ключевой этап сборки генома *de novo* в отсутствие референсной ДНК – это склеивание прочитанных

фрагментов генома путем совмещения их перекрывающихся участков.

В настоящее время в основе большинства ассемблеров используются метод перекрытий (OLC, Overlap-Layout-Consensus Assembly) [12] и метод построения графа де Брёйна (DBG, De Bruijn Graph Assembly) [13]. Обе платформы решают задачу построения пути в графе, оптимальное решение для которой является NP-трудной задачей. Поскольку эффективного решения задача сборки генома не имеет, ассемблеры используют различного рода алгоритмы аппроксимации для снижения избыточности ребер в графах, способы исправления ошибок секвенирования, эвристические подходы к снижению сложности структуры с целью удлинения простых путей и другие способы упрощения графов. Из-за многообразия алгоритмов сборки, каждый из которых имеет свои сильные и слабые стороны, но ни один не выдает единственный и правильный ответ, задача сборки генома *de novo* остается очень сложной. В условиях, когда существует более двух десятков сборщиков для одной только платформы Illumina, выбор программы для сборки конкретного генома для экспериментаторов является трудной задачей. Показано [2], что все сборщики показывают наилучшие результаты на специфических исходных данных, оцениваемые разными метриками (для каждого алгоритма особенная).

В последнее время появились исследования, направленные на независимую оценку различных сборщиков. Проект The Assemblathon, стартовавший в 2011 году, [3] был первой попыткой сравнения производительности ассемблеров на искусственно сгенерированной геномной последовательности, конечным результатом которого является большая таблица, в которой показано, что все ассемблеры эффективны по различным показателям, поэтому сравнение их между собой затруднено.

Более практичным кажется проект GAGE [4], который позволяет оценивать работу наиболее популярных сборщиков *de novo* (ABySS [5], ALLPATHS-LG [6], Bambus2 [7], CABOG [8], MSR-CA, SGA [9], SOAPdenovo [10], Velvet [11]) на реальных геномных последовательностях, просеквенированных с глубоким покрытием (бактериальные геномы *S. aureus* (однохромосомный) и *R. sphaeroides* (двухромосомный), а также эукариотическая 14-я хромосома *H. sapiens*).

Наиболее перспективной стратегией получения безреференсной сборки генома является синтез результатов работы нескольких сборщиков на одних и тех же исходных данных.

В последнее время были созданы программные средства, реализующие стратегию согласования для пары результатов работы геномных сборщиков. Обычно один набор контигов считается «ведущим», а второй «ведомым», задача согласования заключается в объединении контигов из пары наборов, при этом необходимо обнаружить и изолировать проблемные регионы. Цель подобной стратегии – уменьшение фрагментированности наборов контигов и увеличение средней длины последовательностей в объединении. Основные проблемы, возникающие при согласовании результатов различных геномных сборщиков, это ошибочное склеивание контигов, соответствующих различным фрагментам референсной последовательности и появление дубликатов в сборке. Вторая проблема возникает из-за того, что большая часть собираемого генома дублируется в наборах контигов, поэтому необходимо отфильтровывать последовательности, которые полностью покрываются либо непосредственно контигами, либо их объединениями.

Одной из первых программ для объединения набора контигов от разных сборщиков стала Reconciliator [15]. В ней производится поиск участков, являющихся уникальными как для ведомой, так и для ведущей последовательности. На следующем этапе закрываются пропуски в контигах из первого набора с использованием последовательностей из второго набора. В случае наличия нескольких вариантов выбирается тот, который отвечает лучшему статистическому критерию. Похожий подход использует GAA [16], строящая граф соответствия между наборами контигов, используемый для объединения сборок, и ZORRO [17], которая предваряет шаг объединения этапом фильтрования контигов, содержащих ошибки. Программа GAM-NGS [18] ищет в контигах блоки соответствия, которые зависят от количества ридов, проецирующихся на сравниваемые последовательности. Она не проводит процедуру выравнивания каждого контига с каждым. Исходя из информации, полученной на стадии картирования ридов, строится граф сборок, анализируя который, можно установить участки несоответствия между наборами контигов.

Вышеперечисленные программы объединения результатов работы геномных сборщиков предназначены для сопоставления только двух наборов контигов. Также к списку их недостатков можно отнести значительное потребление ими оперативной памяти (на одном из входных наборов данных больше 100 Гб [15]), программную реализацию, выполненную на интерпретируемых языках программирования, отсутствие явных результатов улучшения качества при объединении сборок.

Существуют программные решения MIX [19] и CISA [20], позволяющие объединять произвольное число наборов контигов. Эти программы совмещают различные геномные сборки, используя симметричный подход, не выделяя ведущую сборку. Данным методам для работы не требуется референсная последовательность.

Основные проблемы при использовании программы MIX связаны с обработкой повторяющихся участков в геноме. Поскольку данный метод никак не учитывает наличие повторов в геноме, возможна ситуация, когда граф контигов

будет содержать ложные ребра, то есть ребра, которые соединяют контиги, не следующие друг за другом в референсном геноме, но имеющие перекрытие из-за ошибок, допущенных на этапе сборки, или из-за наличия у них одинаковых повторяющихся участков. Также из-за повторов в геноме граф контигов может содержать ложное ребро между вершинами, которые относятся к разным хромосомам. До этого момента во всех рассматриваемых примерах считалось, что референс – это одна непрерывная последовательность, однако геномы многих организмов состоят из нескольких хромосом, что существенно усложняет как задачу сборки, так и задачу объединения полученных сборок.

В отличие от MIX, алгоритм CISA учитывает наличие повторяющихся участков в геноме и не объединяет контиги, длина перекрытия которых превосходит самый протяженный повтор в контиге. Это позволяет уменьшить число ошибок в наборе объединенных контигов. Однако разбиение и удаление контигов, выполняемое алгоритмом, может привести к тому, что качество итоговой сборки может оказаться хуже, чем у отдельно взятой сборки, участвующей в объединении. Не совсем понятен выбор констант, используемых на различных этапах работы алгоритма. Не исключено, что эти значения констант обеспечивают хорошую работу метода только на определенных наборах данных.

В силу недостатков вышеперечисленных программ был предложен новый метод (GAR – Genome Assembly Refinement) [21, 22] для объединения наборов контигов, полученных от разных сборщиков. Проблема выявления ложных ребер в графе контигов может быть неразрешима без информации о референсе, поэтому предлагаемый метод не ставит своей целью исключить все такие ребра, однако объединение контигов выполняется более консервативно, чем это делает MIX. Работа метода разбита на несколько этапов:

— попарное выравнивание контигов;

— построение неориентированного графа контигов и его кластеризация;

— объединение контигов в найденных кластерах;

— корректирование полученного множества, а именно: удаление перекрывающихся концевых участков в объединенных контигах и исключение из результирующего множества вложенных последовательностей.

Для решения проблемы появления дубликатов в итоговой сборке на этапе объединения контигов используется следующий подход: в кластере выбираются контиги, не являющиеся подпоследовательностями других контигов (в том числе и из других кластеров), называемые затравками. Затравки расширяются в левую и правую стороны за счет контигов, находящихся в том же кластере. Для каждого кластера подсчитывается суммарная длина контигов, вложенных в последовательности из этого кластера, называемая числом вложенности. Последовательность кластеров упорядочивается по убыванию числа вложенности, при этом при присоединении очередного контига к затравке дерево вложенных в него контигов рекурсивно помечается как уже использованное. Использованные контиги больше не могут участвовать в объединении. Используемая концепция затравок в кластерах позволяет уменьшить показатель дублирования в объединенном наборе сборок, сохранив при этом высокий показатель покрытия генома.

Для определения того, покрывается ли контиг другим контигом или комбинацией других контигов, используются два параметра алгоритма: порог подстроки и порог покрытия контига. Порог подстроки – это доля подряд идущих символов в последовательности контига, содержащихся в другом контиге, при превышении которой первый контиг считается подстрокой второго. Значение по умолчанию – 0.9. Порог покрытия контига – это доля символов в последовательности контига, содержащихся во множестве других контигов, при превышении которой контиг исключается из итоговой сборки. Значение по умолчанию – 0.999.

На рисунке 1 показан пример работы предложенного метода объединения результатов работ геномных сборок для трех наборов контигов. В данном примере в результате кластеризации контиги будут разбиты на три группы. При этом в контиг 1.2 вложены контиги 2.2 и 2.3. Это значит, что число вложенности для кластера 1 равно двум, а для кластеров 2 и 3 оно равно нулю. Таким образом, объединение контигов будет начато с кластера 1. Контиг 1.2 входит в объединение контигов в данном кластере (более того, он является затравкой, как самый длинный), поэтому контиги 2.2 и 2.3, которые он покрывает, будут помечены как использованные. Поскольку в кластере 2 не осталось неиспользованных контигов, объединение последовательностей в нем не производится. Контиг 3.4, являющийся левым концом объединения контигов в кластере 3, перекрывается с контигом 1.2, поэтому участок перекрытия между ними удаляется из контига 3.4.

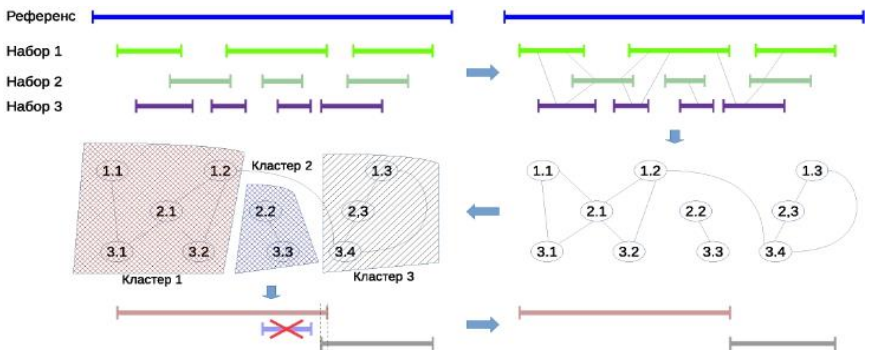

Рис. 1. Иллюстрация работы метода объединения контигов из трех наборов контигов.

Разработанный ресурс был встроен в информационно-вычислительную среду Mathcell [23, 24], которая легко позволяет интегрировать различные расчетные модели, предоставляя web-интерфейс для задания параметров модели, запуска их на счет с использованием мощностей выделенного узла вычислительного кластера ИМПБ РАН, представления и интерпретации результатов расчетов. Созданный сервер GAR, предоставляющий пользователям возможность объединения результатов работы нескольких сборщиков геномных последовательностей, снабжен подробным описанием метода на русском и английском языках с объяснением всех параметров модели и их значениями, задаваемыми по умолчанию [25]. Для качественной сборки на вход сервера надо подать от трех до десяти наборов контигов в FASTA-формате. Размер каждого из наборов не должен превышать 10 Мб.

Создание нового сервера в рамках портала Mathcell позволяет пользователям решать задачу объединения результатов работы различных ассемблеров для улучшения черновых версий сборок генома *de novo*, особенно в ситуациях, когда целевой геном неизвестен. На рисунке 2 показан пример страницы пользовательского интерфейса, на которой задаются параметры работы алгоритма GAR. Запуск модели на счет с использованием вычислительных возможностей портала Mathcell избавляет пользователя от необходимости установки специализированных библиотек, необходимых для работы программной реализации алгоритма, не требует знания операционной системы Linux и обеспечивает удаленный доступ к результатам расчетов в удобное для пользователя время. Результаты расчетов хранятся во временном хранилище данных Mathcell в течение 14 дней.

GAR позволяет пользователю получить сборку генома, которая будет интегрировать результаты работы разных программ-сборщиков генома. Для качественной сборки нужно подать на вход сервера не меньше трех наборов контигов в FASTA-формате. Если наборов контигов меньше трех, программа не работает. Запуск сборщиков пользователь должен осуществить самостоятельно.

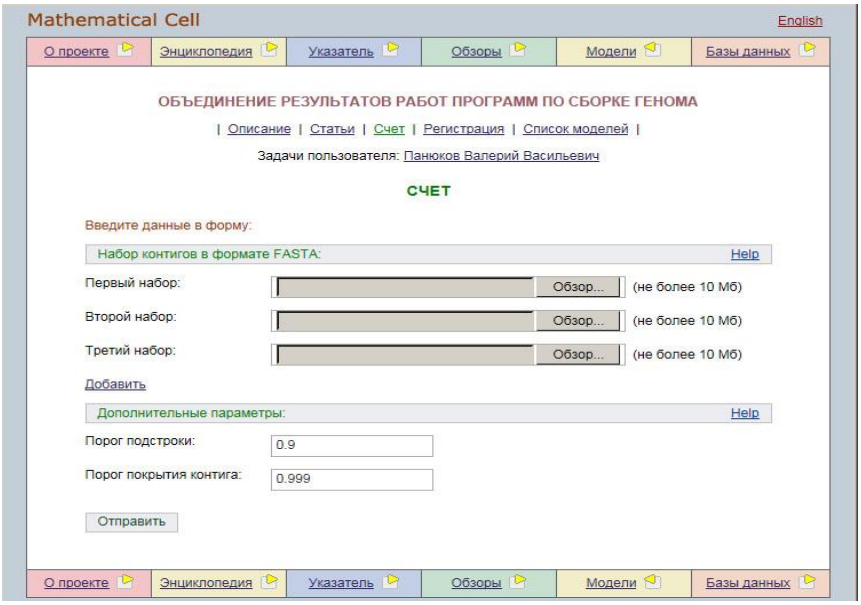

Рис. 2. Скриншот web-формы MathCell для запуска GAR.

Список литературы

1. Buermans H.P.J., den Dunnen J.T. Next generation sequencing technology. Advances and applications. *BBA – Molecular Basis of Disease*. 2014. V. 1842. №. 10. P. 1932–1941.
2. Miller J.R., Koren S., Sutton G. Assembly algorithms for next-generation sequencing data. *Genomics*. 2010. V. 95. P. 315–327.
3. Earl D.A., Bradnam K., St John J., Darling A., Lin D., Faas J., Yu H.O., Vince B., Zerbino D.R., Diekhans M., et al. Assemblathon 1: A competitive assessment of *de novo* short read assembly methods. *Genome Res*. 2011. V. 21. P. 2224–2241.
4. Salzberg S.L., Phillippy A.M., Zimin A., Puiu D., Magoc T., Koren S., Treangen T.J., Schatz M.C., Delcher A.L., Roberts M., Marcxais G., Pop M., Yorke J.A. GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res*. 2012. V. 22. P. 557–567.
5. Simpson J.T., Wong K., Jackman S.D., Schein J.E., Jones S.J., Birol I. ABySS: a parallel assembler for short read sequence data. *Genome Res.* 2009. V. 19. P. 1117–1123.
6. Gnerre S., Maccallum I., Przybylski D., Ribeiro F.J., Burton J.N., Walker B.J., Sharpe T., Hall G., Shea T.P., Sykes S. et al. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci*. 2011. V. 108. P. 1513–1518.
7. Koren S., Treangen T.J., Pop M. Bambus 2: Scaffolding metagenomes. *Bioinformatics.* 2011. V. 27. P. 2964–2971.
8. Miller J.R., Delcher A.L., Koren S., Venter E., Walenz B.P., Brownley A., Johnson J., Li K., Mobarry C., Sutton G. Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics*. 2008. V. 24. P. 2818–2824.
9. Simpson J.T., Durbin R. Efficient *de novo* assembly of large genomes using compressed data structures. *Genome Res*. 2012. V. 22. № 3. P. 549–556.
10. Li R., Zhu H., Ruan J., Qian W., Fang X., Shi Z., Li Y., Li S., Shan G., Kristiansen K., et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.* 2010. V. 20. P. 265–272.
11. Zerbino D.R., Birney E. Velvet: Algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome Res*. 2008. V. 18. P. 821–829.
12. Myers E.W. Toward simplifying and accurately formulating fragment assembly. *J. Comput. Biol*. 1995. V. 2. P. 275–290.
13. Idury R.M., Waterman M.S. A new algorithm for DNA sequence assembly. *J. Comput. Biol*. 1995. V. 2. P. 291–306.
14. Nagarajan N., Pop M. Parametric complexity of sequence assembly: theory and applications to next generation sequencing. *J. Comput. Biol*. 2009. V. 16. P. 897–908.
15. Zimin V.A., Smith D.R., Sutton G. Assembly reconciliation. *Bioinformatics*. 2008. V. 24. P. 42–45.
16. Yao G., Ye L., Gao H. Graph accordance of next-generation sequence assemblies. *Bioinformatics*. 2012. V. 28. P. 13–16.
17. *Zorro – The masked assembler*. URL: http://lge.ibi.unicamp.br/zorro/ (дата обращения: 19.09.2018).
18. Vicedomini R., Vezzi F., Scalabrin S., Arvestad L., Policriti A. GAM-NGS: genomic assemblies merger for next generation sequencing. *BMC Bioinformatics*. 2013. V. 14. Article No. S6.
19. Soueidan H., Maurier F., Groppi A., Sirand-Pugnet P., Tardy F., Citti C., Dupuy V., Nikolski M., et al. Finishing bacterial genome assemblies with Mix. *BMC Bioinformatics*. 2013. V. 14. Article No. S16.
20. Lin S.H., Liao Y.C. CISA: Contig Integrator for Sequence Assembly of Bacterial Genomes. *PLoS ONE*. 2013. V. 8. Article No. e60843.
21. Романенков К.В., Сальников А.Н., Алексеевский А.В. Параллельный метод объединения результатов работы программ по сборке генома *Вестник ЮУрГУ, Серия "Вычислительная математика и информатика"*. 2016. Т. 5. № 1. С. 24–34.
22. Romanenkov K.V. Parallel merging method to integrate different genome assemblies. In: *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2015. P. 1461–1464.
23. Лахно В., Назипова Н., Ким В., Филиппов С., Фиалко Н., Устинин Д., Теплухин А., Тюльбашева Г., Зайцев А., Устинин М. Информационно-вычислительная среда Mathcell для моделирования живой клетки. *Математическая биология и биоинформатика*. 2007. Т. 2. № 2. С. 361–376.
24. Nazipova N., Tyulbasheva G., Zaitsev A., Teplukhina E., Panyukov V., Ustinin M., Lakhno V.D., Ozoline O. Modeling Living Cell Base Processes Using Mathematical Cell Models Collection. In: *International Conference on Mathematical Modeling and Computational Physics (MMCP 2015)*: Book of Abstracts. Košice: FEE&I TU, 2015. P. 70-71.
25. Романенков К.В. *Объединение результатов работ программ по сборке генома.* URL: http://www.mathcell.ru/model8.php (дата обращения: 19.09.2018).