БИОИНФОРМАТИКА
BIOINFORMATICS

УДК 004.64, 577.21
https://doi.org/10.37661/1816-0301-2022-19-1-59-71

Оригинальная статья
Original Paper

Разработка базы данных мотивов регуляции транскрипции у бактерий

В. В. Скакун✉, Е. А. Николайчик

Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь
✉*E-mail: skakun@bsu.by*

Аннотация

Цели. Объемы данных, генерируемые современными методами высокопроизводительного секвенирования, таковы, что их анализ выполняется преимущественно в автоматическом режиме. В частности, использование вновь расшифрованных геномных последовательностей возможно только после аннотации функциональных элементов генома, которая, как правило, выполняется автоматическими конвейерами. Такие конвейеры аннотации успешно справляются с идентификацией генов, но ни один из них не аннотирует регуляторные элементы, без которых нельзя понять, когда и как гены могут экспрессироваться. Информация о регуляторных элементах бактерий собрана в нескольких специализированных базах данных (RegulonDB, CollecTF, Prodoric2 и др.), однако только часть этой информации можно использовать для аннотации регуляторных элементов и только у очень ограниченного круга бактерий. Ранее авторами был предложен четкий формальный критерий для применения регуляторной информации к любым бактериальным геномам. Таким критерием стал CR-тег – последовательность аминокислотных остатков транскрипционного регулятора, специфически контактирующих с азотистыми основаниями регуляторного элемента в геномной ДНК. Связанная с CR-тегом математическая модель регуляторного элемента (мотив) может быть корректно применена для аннотации подобных элементов в любых геномах, кодирующих транскрипционный регулятор с идентичным CR-тегом. Накопление связанных с CR-тегами мотивов поставило вопрос об их упорядоченном хранении для удобства последующего применения при аннотации геномных последовательностей. Поскольку ни одна из известных баз данных не использует концепцию CR-тегов, потребовалась разработка новой базы данных. Таким образом, целью работы является создание базы данных с информацией о бактериальных транскрипционных факторах и распознаваемых ими последовательностях ДНК, пригодной для аннотации регуляторных последовательностей в бактериальных геномах.

Методы. Инфологическое моделирование предметной области производилось с помощью методологии IDEF1X. Разработка базы данных выполнялась посредством СУБД Microsoft SQL Server. Кроссплатформенное приложение по импорту данных в базу данных написано на языке C++ с использованием технологии Qt.

Результаты. В результате проведенного исследования предметной области была разработана и реализована в СУБД Microsoft SQL Server реляционная модель данных, позволяющая целостное хранение информации о накопленных мотивах регуляции транскрипции у бактерий, включая и информацию о публикациях, подтверждающих корректность этих мотивов. Для автоматизации процесса ввода накопленных данных разработано кроссплатформенное приложение для импорта структурированных данных о транскрипционных факторах.

Заключение. Основным отличием разработанной базы данных является использование концепции CR-тега. Записи математических моделей регуляторных элементов (мотивов) в базе данных связаны с CR-тегом и поэтому могут быть корректно применены для аннотации подобных элементов в любых геномах, кодирующих транскрипционный регулятор с идентичным CR-тегом. Разработанная база данных обеспечит структурированное и целостное хранение данных, а также их быстрый поиск при использовании в конвейере автоматической аннотации регуляторных элементов в бактериальных геномных последовательностях.

Ключевые слова: регуляция транскрипции, регуляторные мотивы, последовательности ДНК, CR-тег, программа SigmoID, базы данных

Благодарности. Работа выполнялась в рамках задания 1.10.5 ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства» (2021–2025).

Для цитирования. Скакун, В. В. Разработка базы данных мотивов регуляции транскрипции у бактерий / В. В. Скакун, Е. А. Николайчик // Информатика. – 2022. – Т. 19, № 1. – С. 59–71.
https://doi.org/10.37661/1816-0301-2022-19-1-59-71

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 01.11.2021
Подписана в печать | Accepted 19.01.2022
Опубликована | Published 29.03.2022

Development of a bacterial regulatory motif database

Victor V. Skakun✉, Yevgeny A. Nikolaichik

Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus
✉*E-mail: skakun@bsu.by*

Abstract

Objectives. The amount of data generated by modern methods of high-throughput sequencing is such that their analysis is performed mainly in automatic mode. In particular, the use of newly decoded genomic sequences is possible only after the annotation of functional elements of the genome, which, as a rule, is performed by automatic pipelines. Such annotation pipelines do a good job to identify the genes, but none of them annotate regulatory elements. Without these elements it is not possible to understand when and how genes can be expressed. Information on the regulatory elements of bacteria is collected in several specialized databases (RegulonDB, CollecTF, Prodoric2, etc.), however, only a part of this information can be used for annotation of regulatory elements, and only for a very limited range of bacteria. Previously, we proposed a clear formal criterion for applying regulatory information to any bacterial genome. Such a criterion is the CR tag, a sequence of amino acid residues of a transcriptional regulator that specifically contacts the nitrogenous bases of regulatory element in genomic DNA. The mathematical model of a regulatory element (motif) associated with a CR tag can be correctly applied to annotate similar elements in any genomes encoding a transcriptional regulator with an identical CR tag. The accumulation of motifs associated with CR tags raised the question of their ordered storage for the convenience of subsequent use in the annotation of genomic sequences. Since no one of well-known databases uses the concept of CR tags, a new database ought to be developed. Thus, the goal of this work is to create a database with information about bacterial transcription factors and DNA sequences recognized by them, suitable for annotation of regulatory sequences in bacterial genomes.

Methods. Infological modeling of the subject area was carried out using the IDEF1X methodology. The database was developed using the Microsoft SQL Server DBMS. A cross-platform application for importing data into a database is written in C++ using Qt technology.

Results. As a result of the study of the subject area, a relational data model was developed and implemented in the Microsoft SQL Server DBMS, which allows holistic storage of information about accumulated transcription regulation motifs in bacteria, including information about the publications confirming their correctness. To automate the process of entering accumulated data, a cross-platform application was developed for importing structured data on transcription factors.

Conclusion. The main difference of the developed database is the use of CR-tag concept. Records of mathematical models of regulatory elements (motifs) in the database are associated with a CR tag and, therefore, can be correctly used to annotate similar elements in any genomes encoding a transcriptional regulator with an identical CR tag. The developed database will provide structured and holistic data storage, as well as their quick search when used in the pipeline for automatic annotation of regulatory elements in bacterial genomic sequences.

Keywords: regulation of transcription, regulatory motifs, DNA sequences, CR tag, SigmoID program, databases

Acknowledgements. The work was carried out within the task 1.10.5 of the State Scientific Research Program "Digital and Space Technologies, Human, Society and State Security" (2021–2025).

For citation. Skakun V. V., Nikolaichik Y. A. *Development of a bacterial regulatory motif database*. Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 59−71 (In Russ.). https://doi.org/10.37661/1816-0301-2022-19-1-59-71

Conflict of interest. The authors declare no conflict of interest.

Введение. Идентификация регуляторных элементов геномов является одной из наиболее актуальных задач современной геномики. Особую важность этой задаче придает то, что в подавляющем большинстве геномных последовательностей, депонированных в нуклеотидных базах данных (БД), регуляторные элементы вообще не аннотированы. Одной из причин сложившейся ситуации является сложность статистически достоверной идентификации большинства регуляторных элементов в геномных масштабах. Тем не менее решение этой задачи возможно для прокариот, поскольку их геномы компактны, а регуляторные элементы расположены преимущественно в межгенных участках, занимающих порядка 10 % генома. При общем сходстве структур генома две группы прокариот, бактерии и археи, существенно отличаются по механизмам транскрипционной регуляции, поэтому дальнейшее обсуждение касается только бактерий.

Можно выделить три основных типа регуляторных элементов, контролирующих экспрессию генов у бактерий: промоторы, операторы и терминаторы [1]. В отличие от терминаторов промоторы и операторы являются в значительной степени геноспецифическими и очень вариабельными из-за распознавания их большим числом (несколькими сотнями) различных транскрипционных факторов. Бактериальные транскрипционные факторы в подавляющем большинстве случаев являются гомоолигомерами (чаще всего ди- или тетрамерами), в связи с этим типичный регуляторный элемент распознается двумя идентичными ДНК-связывающими доменами и имеет четко выраженную симметрию, что облегчает его идентификацию [1, 2]. Размеры сайтов связывания транскрипционных факторов (операторов) обычно варьируют в пределах 15–25 пар нуклеотидов [1, 2], поэтому с учетом суммарной длины всех регуляторных последовательностей прокариотического генома порядка нескольких сот тысяч пар нуклеотидов статистический анализ позволяет отличать регуляторные элементы от всех прочих геномных последовательностей. Действительно, простейшая математическая модель (весовая матрица) регуляторного элемента, созданная на основе нескольких десятков экспериментально охарактеризованных операторов для конкретного транскрипционного фактора, может быть успешно использована для идентификации операторов в родственных геномах [3, 4].

Вместе с тем проблемой является значительно более высокая скорость эволюции (и, соответственно, вариабельность) регуляторных элементов и генов транскрипционных факторов в сравнении с другими функционально значимыми участками геномов [5–7], из-за чего модель регуляторного элемента, справедливую для одного генома, нельзя применять к другому без доказательства идентичности контактов между белком-регулятором и оператором. Еще одной проблемой является то, что для построения надежной статистической модели регуляторного элемента требуется достаточное число (не менее 10) известных последовательностей, тогда как большинство транскрипционных факторов контролируют небольшие регулоны (например, у бактерий *E. coli* более половины транскрипционных факторов контролируют всего один-три оперона [8]), поэтому для них известно меньшее число мишеней. Сложилась парадоксальная ситуация: опубликовано много экспериментальных работ, характеризующих отдельные транскрипционные факторы, выявлены соответствующие операторные последовательности, но в большинстве случаев эту информацию нельзя непосредственно использовать для поиска со-

ответствующих операторов в других геномных последовательностях из-за отсутствия моделей регуляторных элементов и четких критериев их применения к конкретным геномам.

Существует несколько специализированных БД, обобщающих опубликованную информацию о регуляторных элементах и предлагающих их математические модели. Самой известной БД, собирающей всю доступную информацию об одном организме, является RegulonDB – специализированная БД для *Escherichia coli* [8]. Это наиболее полная БД для отдельно взятого организма, которая содержит информацию о промоторных последовательностях для всех семи сигма-факторов данной бактерии, а также об операторах для 221 из ~300 транскрипционных факторов. Однако только для малого количества представленных в RegulonDB транскрипционных факторов есть пригодные к непосредственному использованию модели операторных последовательностей, что обусловлено двумя основными причинами: для многих транскрипционных факторов число охарактеризованных операторов слишком мало; при достаточном числе операторов, охарактеризованных в разных публикациях, они часто имеют разную ширину или просто не выровнены из-за различий в их описании в разных экспериментальных работах. Такие БД, как CollecTF [9], Prodoric [10], CoryneRegNet [11], собирают экспериментально полученную регуляторную информацию уже для многих видов. Каждая из этих БД имеет определенную специализацию и свои достоинства, но вместе с тем и существенный недостаток – малое число операторных моделей, пригодных для непосредственного использования.

Среди БД с регуляторной информацией самой обширной является RegPrecise [12]. Версия 4.0 этой БД содержит информацию об операторных мотивах для 11 520 транскрипционных факторов, причем в большинстве случаев с моделями, пригодными для непосредственного использования, однако операторные последовательности в RegPrecise в подавляющем числе случаев выявлены методами сравнительной геномики и не имеют экспериментального подтверждения.

Для аннотации известных операторов и промоторов путем применения регуляторной информации из БД RegPrecise, RegulonDB и CollecTF к неохарактеризованным геномам была разработана программа SigmoID [13]. Полученный авторами опыт использования перечисленных выше БД для исследования транскрипционной регуляции у немодельных видов бактерий не имел большого успеха. Первая версия программы SigmoID хотя и позволяла применять регуляторную информацию из RegPrecise, RegulonDB и CollecTF к неохарактеризованным геномам, но не обладала четкими критериями корректности такого переноса регуляторной информации. Фактически пользователю предлагалось самостоятельно установить наличие у исследуемого организма ортологов известных транскрипционных факторов и принять решение о достаточности уровня гомологии между транскрипционными факторами для переноса регуляторной информации. В версии 2 программы SigmoID [14, 15] в качестве критерия возможности применения имеющейся операторной модели к исследуемой геномной последовательности взята высказанная ранее идея [16] о строгом соответствии оператора так называемому CR-тегу – последовательности аминокислотных остатков ДНК-связывающего домена транскрипционного фактора, непосредственно контактирующих с азотистыми основаниями оператора. CR-тег является уникальным идентификатором пары «транскрипционный фактор – операторный мотив», поэтому в версии 2 SigmoID все операторные мотивы связаны с CR-тегами своих транскрипционных факторов и применяются для аннотации операторов только в тех геномах, которые кодируют транскрипционный фактор с идентичным CR-тегом. Благодаря использованию концепции CR-тегов программа SigmoID версии 2.0 способна корректно аннотировать операторные последовательности любых бактериальных геномов в полностью автоматическом режиме с применением имеющейся коллекции калиброванных профилей операторных мотивов.

Результатом процесса анализа регуляторной информации с помощью SigmoID является набор папок (по одной для каждого транскрипционного фактора), содержащих пять текстовых файлов с общим описанием транскрипционного фактора и с данными о CR-теге, найденных операторах, регуляторном мотиве, описываемом профильной скрытой марковской моделью (hidden markov model, HMM) и позиционной весовой матрицей (position weight matrix, PWM), а также о параметрах поиска с использованием этих HMM и PWM. Дополнительно формируются два файла для хранения данных по доказательной базе, подтверждающей справедливость

мотива: публикации, результаты экспериментального подтверждения и сведения о лице, курирующем проведенные исследования. По мере накопления связанных с CR-тегами операторных моделей встает вопрос о хранении этой информации. Ни одна из общеизвестных на сегодня БД не использует концепцию CR-тегов, поэтому потребовалась разработка принципиально новой БД. Ее дизайн предполагает два ключевых отличия от имеющихся решений:

1) каждый транскрипционный фактор имеет математическую модель оператора (скрытую марковскую модель), непосредственно применимую для идентификации операторов в геномных последовательностях;

2) все операторные модели ассоциированы с CR-тегами.

Кроме того, несмотря на использование сравнительной геномики при конструировании операторных моделей, каждая такая модель обязательно должна иметь экспериментальное подтверждение корректности операторного мотива.

Для структурированного и целостного хранения набора взаимосвязанных данных наилучшим решением является применение технологии БД и, в частности, реляционной модели данных. БД может служить ядром системы обработки и анализа геномных данных с целью предсказания и верификации мотивов регуляции транскрипции у бактерий. Хранение в БД множества структурированных и взаимосвязанных данных предоставляет возможность дополнительного статистического анализа, что позволяет повысить детализацию результатов анализа и улучшить качество интерпретации данных. Таким образом, целью настоящей работы является разработка БД мотивов регуляции транскрипции у бактерий. Выполнению этой цели предшествует решение следующих задач: моделирование предметной области для создания схемы БД, создание БД с помощью некоторой системы управления БД СУБД и импорт в БД уже накопленного массива данных, что, в свою очередь, предполагает написание специальной программы, автоматизирующей процесс импорта данных из набора текстовых файлов.

Разработка базы данных. Начальным этапом разработки БД является этап инфологического моделирования, заключающийся в анализе предметной области и создании концептуальной модели данных, цель которой – максимально отразить семантику предметной области в терминах модели данных [17]. Стандартом де-факто здесь выступает технология IDEF1X (URL: https://www.idef.com/idef1-information_modeling_method/). Моделирование предметной области и разработка схемы БД выполнялись с помощью CASE (Computer-aided Software Engineering) системы DBDesigner Fork (URL : https://sourceforge.net/projects/dbdesigner-fork). Моделирование проводилось согласно технологии IDEF1X с отображением связей по нотации “Crow’s foot” («воронья ножка») [17].

При анализе предметной области выделены следующие высокоуровневые сущности: CRTags (CR-теги), TFs (транскрипционные факторы), TF_families (семейства транскрипционных факторов), Motifs (мотивы) и Operators (операторы). Транскрипционные факторы описываются посредством следующих атрибутов: названия, CR-тега, идентификатора в некоторой референсной БД и самой последовательности транскрипционного фактора. Для однозначного определения семейства транскрипционного фактора кроме его названия добавлен и идентификатор Accession (код доступа) соответствующей референсной БД (PFAM или SMART [18, 19]). Регуляторные мотивы представляются посредством профильной HMM и (или) PWM. Сайты связывания транскрипционных факторов (операторы) описываются идентификатором и собственно самой последовательностью сайта связывания. Между сущностями TFs и Motifs установлено отношение «один ко многим», так как для одного транскрипционного фактора может быть описано несколько регуляторных мотивов. Для хранения параметров анализа введена сущность Settings (настройки). Задание параметров в виде пары {Name (название параметра), Value (значение параметра)} позволяет хранить произвольное количество любых параметров. Для хранения детализированной информации, обосновывающей найденные мотивы и операторы транскрипционных факторов, введены сущности Publications (публикации), Curators (кураторы), Evidence_types (типы подтверждения), а также связующие сущности Motifs_curators и Motif_references, которые реализуют отношения

«многие-ко-многим», существующие между сущностями Motifs, Curators и Publications, а также Motifs_references и Evidence_types соответственно. Перечисленные свойства обеспечивают модели данных универсальность и инвариантность к различным видам анализа.

Разработанная схема БД представлена на рис. 1, где пиктограмма ключа – это первичный ключ, красный ромбик слева от названия поля означает, что для данного поля задано ограничение ссылочной целостности. Индексы перечислены внизу каждой таблицы. Схема содержит 12 сущностей и может быть транслирована в реляционную модель, предоставляя уровень нормализации не ниже третьей нормальной формы [17]. В ней учтены необходимые валидаторы значений (поле Email) и ограничения ссылочной целостности. По внешним ключам, осуществляющим связь с сущностями, для которых предполагается много экземпляров, созданы индексы в целях повышения скорости выполнения многотабличных запросов, поиска и фильтрации данных. С помощью индексов реализовано требование уникальности значений полей Accession, CRTag, ProteinID, Email. Прочие поля, для которых предполагается проведение поиска, например Name, Date и др., также проиндексированы.

В распоряжении авторов имеется доступ к серверу под управлением ОС Microsoft Windows Server 2012 с установленной на ней системой управления БД Microsoft SQL Server 2017 (URL: https://www.microsoft.com/en-us/sql-server/sql-server-2017). Данная СУБД относится к разряду промышленных высокопроизводительных и надежных решений и способна эффективно решать задачи хранения и обработки больших наборов данных, включая данные по регуляции транскрипции у бактерий. Соответственно, на следующем этапе разработанная схема БД была транслирована в реляционную модель с учетом требований вышеуказанной СУБД и развернута на сервере. Для создания БД использовалась среда разработки Microsoft SQL Server Management Studio (URL: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms).

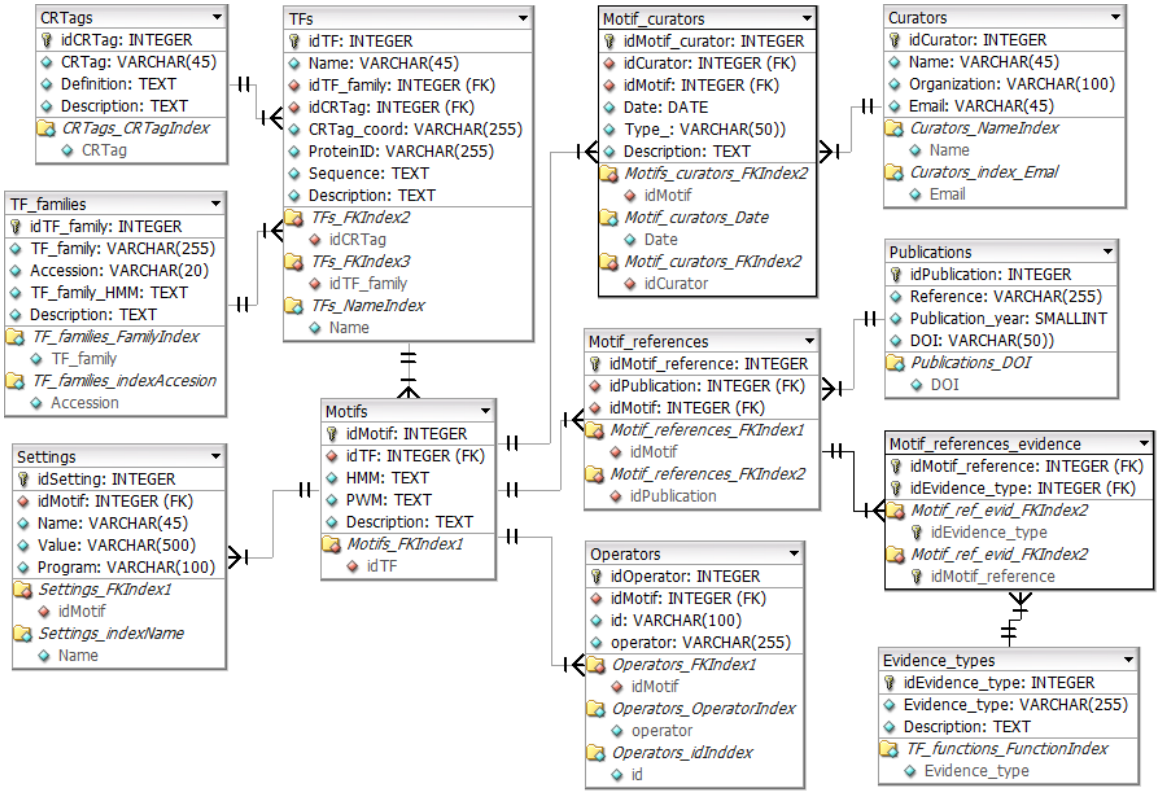

Рис. 1. Схема базы данных мотивов регуляции транскрипции у бактерий в формате IDEF1X

Fig. 1. Schematic of the database of bacterial transcription regulation motifs in the IDEF1X format

Типы и размеры некоторых полей были изменены. Благодаря тому что SQL Server [20] позволяет определять текстовые поля с максимальным размером в 8000 Б, тип полей Definition (отношение CRTags) и Sequence (отношение TFs) был заменен с TEXT на VARCHAR(8000). Такая замена позволяет строить эффективные индексы по вышеуказанным полям в целях ускорения поиска и доступа к данным и дополнительно определять уникальность значений в этих полях.

Учитывая то, что SQL Server не позволяет создавать индексы с требованием уникальности значений для полей с допуском пустых значений, ряд индексов был сформирован с добавлением предиката IS NULL, например:

```
CREATE UNIQUE NONCLUSTERED INDEX idx_TFs_ProteinID_notnull ON TFs(ProteinID) WHERE Pro-
teinID IS NOT NULL;
```

В целях упрощения и стандартизации доступа к данным был разработан набор представлений. Представления по своей сути являются виртуальными таблицами, получающимися в результате выполнения определенного запроса к БД, т. е. представляют собой именованный запрос [17]. Они определяют логическую независимость от данных и интерфейс пользователя для доступа к ним. Приведем SQL-код [21] основных представлений.

Представление TFs_TF_familiesView предназначено для просмотра объединенной информации по транскрипционным факторам и их семействам (основано на таблицах TFs, CRTags и TF_families с правым внешним соединением [21] таблиц TFs и TF_families):

```
CREATE VIEW TFs_TF_familiesView AS
SELECT dbo.TF_families.TF_family, dbo.TF_families.Accession,
dbo.TF_families.TF_family_HMM, dbo.TF_families.Description AS TF_family_description,
dbo.TFs.idTF, dbo.TFs.Name AS TF_name, dbo.CRTags.CRTag, dbo.TFs.CRTag_coord,
dbo.TFs.ProteinID, dbo.TFs.Sequence,  dbo.TFs.Description AS TF_Description FROM
dbo.CRTags INNER JOIN dbo.TFs ON dbo.CRTags.idCRTag = dbo.TFs.idCRTag RIGHT OUTER JOIN
dbo.TF_families ON dbo.TFs.idTF_family = dbo.TF_families.idTF_family;
```

В коде представления dbo есть название схемы, принадлежащей владельцу БД (database owner). Внешнее соединение таблиц позволяет вывести все семейства независимо от того, есть ли в БД хотя бы один транскрипционный фактор данного семейства. Пример вызова представления TFs_TF_familiesView:

```
SELECT TF_family, Accession, TF_family_HMM, TF_family_description, idTF, TF_name, CRTag,
CRTag_coord, ProteinID, Sequence, TF_Description FROM TFs_TF_familiesView;
```

Представление MotifsView предназначено для просмотра объединенной информации по мотивам (основано на таблицах Motifs, TFs, CRTags и TF_families):

```
CREATE VIEW MotifsView AS
SELECT dbo.Motifs.idMotif, dbo.TFs.Name, dbo.CRTags.CRTag, dbo.TF_families.TF_family,
dbo.TF_families.Accession AS TF_family_accession, dbo.TF_families.Description AS
TF_family_description, dbo.TF_families.TF_family_HMM, dbo.Motifs.HMM, dbo.Motifs.PWM,
dbo.TFs.CRTag_coord, dbo.TFs.ProteinID, dbo.TFs.Sequence, dbo.TFs.Description AS
TF_description, dbo.Motifs.Description AS Motif_description FROM dbo.Motifs INNER JOIN
dbo.TFs ON dbo.Motifs.idTF = dbo.TFs.idTF INNER JOIN dbo.CRTags ON dbo.TFs.idCRTag =
dbo.CRTags.idCRTag INNER JOIN dbo.TF_families ON dbo.TFs.idTF_family =
dbo.TF_families.idTF_family;
```

Пример вызова представления MotifsView:

```
SELECT idMotif, Name, CRTag, TF_family, TF_family_accession, TF_family_description,
TF_family_HMM, HMM, PWM, CRTag_coord, ProteinID, Sequence, TF_description, Mo-
tif_description FROM MotifsView;
```

Представление OperatorsView предназначено для просмотра списка операторов определенного мотива (основано на таблицах Motifs, TFs, CRTags, TF_families и Operators):

```
CREATE VIEW OperatorsView AS
SELECT dbo.Operators.idOperator, dbo.Operators.ID, dbo.Operators.Operator, dbo.TFs.Name
AS TF_name, dbo.TF_families.TF_family, dbo.CRTags.CRTag, dbo.Motifs.idMotif,
dbo.TFs.idTF FROM dbo.Motifs INNER JOIN dbo.Operators ON dbo.Motifs.idMotif =
dbo.Operators.idMotif INNER JOIN dbo.TFs ON dbo.Motifs.idTF = dbo.TFs.idTF INNER JOIN
dbo.CRTags ON dbo.TFs.idCRTag = dbo.CRTags.idCRTag INNER JOIN dbo.TF_families ON
dbo.TFs.idTF_family = dbo.TF_families.idTF_family;
```

Пример вызова представления OperatorsView с выборкой для мотива с идентификатором 2:

```
SELECT idOperator, ID, Operator, TF_name, TF_family, CRTag, idMotif, idTF FROM Opera-
torsView WHERE idMotif = 2;
```

Представление MotifSettingsView предназначено для просмотра параметров определенного мотива (основано на таблицах Motifs, TFs, CRTags, TF_families и Settings):

```
CREATE VIEW MotifSettingsView AS
SELECT dbo.Settings.idSetting, dbo.Settings.Name, dbo.Settings.Value,
dbo.Settings.Program, dbo.TFs.Name AS TF_name, dbo.CRTags.CRTag,
dbo.TF_families.TF_family, dbo.Motifs.idMotif, dbo.TFs.idTF FROM dbo.Settings INNER JOIN
dbo.Motifs ON dbo.Settings.idMotif = dbo.Motifs.idMotif INNER JOIN dbo.TFs ON
dbo.Motifs.idTF = dbo.TFs.idTF INNER JOIN dbo.CRTags ON dbo.TFs.idCRTag =
dbo.CRTags.idCRTag INNER JOIN dbo.TF_families ON dbo.TFs.idTF_family =
dbo.TF_families.idTF_family;
```

Пример вызова представления MotifSettingsView с выборкой для транскрипционного фактора с идентификатором 7:

```
SELECT idSetting, Name, Value, Program, TF_name, CRTag, TF_family, idMotif, idTF FROM
MotifSettingsView WHERE idTF = 7;
```

Представление ReferencesView предназначено для просмотра списка публикаций, подтверждающих определенный мотив (основано на таблицах Motifs, TFs, Publications и Motif_references):

```
CREATE VIEW ReferencesView AS
SELECT dbo.TFs.Name AS TF_name, dbo.Publications.Reference,
dbo.Publications.Publication_year, dbo.Publications.DOI, idTF, idMotif FROM
dbo.Motif_references INNER JOIN dbo.Motifs ON dbo.Motif_references.idMotif =
dbo.Motifs.idMotif INNER JOIN dbo.Publications ON dbo.Motif_references.idPublication =
dbo.Publications.idPublication INNER JOIN dbo.TFs ON dbo.Motifs.idTF = dbo.TFs.idTF;
```

Пример вызова представления ReferencesView с выборкой для транскрипционного фактора GVRTDVTRR_WalR:

```
SELECT TF_name, Reference, Publication_year, DOI, idTF, idMotif FROM ReferencesView
WHERE TF_name = 'GVRTDVTRR_WalR';
```

Разработка программы импорта данных. Результатом работы программы SigmoID является набор текстовых файлов (от пяти до семи). Два файла формируются в результате деятельности сторонних программ из пакета MEME Suite (https://meme-suite.org/meme/) [22], используемых SigmoID в своей работе. Остальные файлы экспортируются непосредственно самой SigmoID.

Для выполнения импорта данных разработан алгоритм, основанный на создании промежуточного контейнера данных в виде словаря (map), хранящего параметры анализа и другие данные в виде множества пар «ключ – значение» (URL: https://doc.qt.io/qt-5/qmap.html#details). Удобством словаря является наличие быстрого (индексированного) поиска значений по ключу. Вначале файлы поочередно считываются в оперативную память компьютера и происходит заполнение словаря. Пустые строки и строки с комментариями игнорируются. Как только встречаются требуемые к импорту данные, происходит вставка записи в словарь. Ключ формируется исходя из названия поля в БД, куда впоследствии планируется запись соответствующего значения, или же исходя из контекста

импортируемых данных. На следующем этапе ведущим становится уже БД, вернее ее структура. Поскольку перечень требуемых к импорту данных точно известен, импорт данных в БД происходит согласно этому перечню. Если параметр или другие данные находятся в словаре, происходит вставка записи в БД. Если поиск в словаре завершается возвратом пустого значения, ничего не вставляется. Такой подход позволяет избавиться от множества рутинных проверок во время импорта и гарантирует надежность всего процесса импорта. Недостатками предложенного алгоритма являются более высокие требования к объему оперативной памяти и наличие повторного поиска данных в словаре. Учитывая то, что весь объем импортируемых данных, как правило, не превышает 50 кБ, а поиск по ключу в словаре очень быстрый, указанные недостатки не оказывают существенного влияния на эффективность процесса импорта.

Поскольку программа SigmoID является кроссплатформенной, доступна под лицензией GPL 3.0 и скомпилирована для трех основных десктопных ОС (GNU/Linux, macOS и Windows), для реализации алгоритма импорта данных была выбрана технология Qt (URL: https://www.qt.io/). Языком программирования в Qt является высокоуровневый язык C++ (а также Python). Технология Qt позволяет писать единый исходный код, который, будучи скомпилированным средствами данной технологии, доступными для соответствующей ОС, приводит к получению программного продукта, выполняющегося в нативном режиме в этой ОС. Наличие полноценной графической библиотеки, а также ряда других библиотек (например, библиотеки доступа к БД) делает технологию Qt очень привлекательной для разработки кроссплатформенных графических приложений. Для доступа к данным, хранящимся на сервере под управлением СУБД Microsoft SQL Server, использовались модуль QtSQL и драйвер QODBC3, осуществляющий соединение с SQL Server по технологии ODBC (URL: https://docs.microsoft.com/en-us/sql/odbc).

Для создания словаря применялся класс QMap. Особенностью технологии Qt является эффективная реализация собственной библиотеки стандартных компонентов, в состав которой и входит класс QMap. Импорт данных из текстовых файлов выполняется в потоковом виде с помощью классов QFile, QIODevice и QTextStream. Соединение с БД производилось с помощью класса QSqlDatabase и драйвера QODBC3, осуществляющего соединение с СУБД SQL Server по технологии ODBC. Для вставки записей в БД использовался класс QSqlQuery, позволяющий выполнять запросы к СУБД, написанные на языке SQL [21].

С целью упрощения процесса вставки записей в базовые отношения CRTags, TF_families, Curators, Publications и EvidenceTypes были написаны сохраненные процедуры на языке TransactSQL [18], являющемся языком программирования серверной логики СУБД SQL Server. Сохраненная процедура хранится и осуществляется на сервере, что предоставляет намного более эффективный способ взаимодействия с СУБД, если планируется выполнение нескольких инструкций SQL с проверкой результатов и условий их запуска. Чтобы не дублировать вставку одного и того же CR-тега, семейства транскрипционного фактора, публикации или куратора, производится вначале поиск записи, содержащей соответствующий CR-тег, семейство и т. д. Если запись найдена, возвращается первичный ключ данной записи, в противном случае вставляется запись с последующим возвратом ее первичного ключа. Приведем код сохраненной процедуры для вставки CR-тега:

```
CREATE PROCEDURE [dbo].[pr_addCRTag] @ID int = null OUTPUT, @CRTag varchar(255) AS
 if NOT EXISTS(SELECT * FROM CRTags WHERE CRTag = @CRTag)
   begin
      INSERT INTO CRTags(CRTag) VALUES(@CRTag)
      SET @ID = SCOPE_IDENTITY()
   end
 else
     SELECT @ID = idCRTag FROM CRTags WHERE CRTag = @CRTag
```

Остальные процедуры имеют аналогичный синтаксис. Возврат значения первичного ключа производится через первый параметр процедуры, описанный с ключевым словом OUTPUT.

Интерфейс разработанного приложения SigmoID data importer по импорту данных в БД (скомпилированный для ОС Windows) изображен на рис. 2.

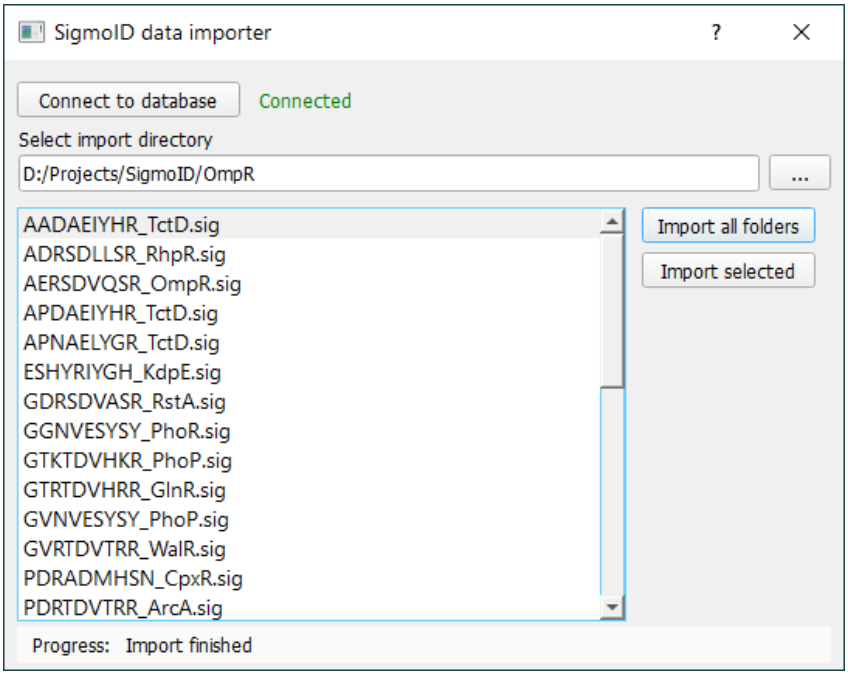

Рис. 2. Внешний вид приложения по импорту данных в БД

Fig. 2. Appearance of the database import application

Заключение. В результате моделирования предметной области была получена инфологическая модель БД мотивов регуляции транскрипции у бактерий. Модель транслирована в реляционную модель и развернута на сервере под управлением СУБД SQL Server. Разработанная БД протестирована путем ввода данных и их обновления с помощью графического клиентского приложения Microsoft SQL Management Studio. Удобство и логичность выполнения действий по вводу данных, полученных из дистрибутива программы SigmoID, доказали соответствие разработанной модели предметной области. Объявленные ограничения на вводимые значения, требования уникальности ввода и правила ссылочной целостности позволяют обеспечить целостность и согласованность данных при их вводе и в процессе дальнейшей работы.

Для облегчения и автоматизации процесса ввода данных в БД разработан и реализован в кроссплатформенном приложении SigmoID data importer алгоритм импорта наборов текстовых файлов, являющихся результатом работы программы SigmoID. Алгоритм основан на промежуточном чтении данных из файлов в индексируемый контейнер типа словаря (map), что позволяет обеспечивать корректное чтение данных из набора файлов, имеющих несколько версий своих структур. Для текущей версии БД разработан набор представлений, предоставляющих удобный интерфейс доступа к хранящимся данным.

В дальнейшем авторы планируют разработать веб-интерфейс к БД и создать полноценную интегрированную систему по обработке и хранению данных мотивов регуляции транскрипции у бактерий.

Вклад авторов. *В. В. Скакун* разработал базу данных и программу импорта данных. *Е. А. Николайчик* сформулировал задачу, обосновал актуальность работы и адаптировал формат файлов программы SigmoID для удобства импорта в базу данных. Инфологическое моделирование предметной области, разработка алгоритма импорта данных и подготовка текста статьи выполнялись совместно обоими авторами.

Список использованных источников

1. Van Hijum, S. A. F. T. Mechanisms and evolution of control logic in prokaryotic transcriptional regulation / S. A. F. T. van Hijum, M. H. Medema, O. P. Kuipers // Microbiology and Molecular Biology Reviews. – 2009. – Vol. 73, no. 3. – P. 481–509. https://doi.org/10.1128/MMBR.00037-08

2. Browning, D. F. Local and global regulation of transcription initiation in bacteria / D. F. Browning, S. J. W. Busby // Nature Reviews Microbiology. – 2016. – Vol. 14, no. 10. – P. 638–650. https://doi.org/10.1038/nrmicro.2016.103

3. Stormo, G. D. DNA binding sites: representation and discovery / G. D. Stormo // Bioinformatics. – 2000. – Vol. 16, no. 1. – P. 16–23. https://doi.org/10.1093/bioinformatics/16.1.16

4. Rodionov, D. A. Comparative genomic reconstruction of transcriptional regulatory networks in bacteria / D. A. Rodionov // Chemical Reviews. – 2007. – Vol. 107, no. 8. – P. 3467–3497. https://doi.org/10.1021/cr068309+

5. Gelfand, M. S. Evolution of transcriptional regulatory networks in microbial genomes / M. S. Gelfand // Current Opinion in Structural Biology. – 2006 – Vol. 16, no. 3. – P. 420–429. https://doi.org/10.1016/j.sbi.2006.04.001

6. Lozada-Chavez, I. Bacterial regulatory networks are extremely flexible in evolution / I. Lozada-Chavez // Nucleic Acids Research. – 2006. – Vol. 34, no. 12. – P. 3434–3445. https://doi.org/10.1093/nar/gkl423

7. Perez, J. C. Evolution of transcriptional regulatory circuits in bacteria / J. C. Perez, E. A. Groisman // Cell. – 2009. – Vol. 138, no. 2. – P. 233–244. https://doi.org/10.1016/j.cell.2009.07.002

8. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12 / A. Santos-Zavaleta [et al.] // Nucleic Acids Research. – 2019. – Vol. 47, no. D1. – P. D212–D220. https://doi.org/10.1093/nar/gky1077

9. CollecTF: a database of experimentally validated transcription factor-binding sites in Bacteria / S. Kılıç [et al.] // Nucleic Acids Research. – 2014. – Vol. 42, iss. D1. – P. D156–D160. https://doi.org/10.1093/nar/gkt1123

10. PRODORIC (release 2009): a database and tool platform for the analysis of gene regulation in prokaryotes / A. Grote [et al.] // Nucleic Acids Research. – 2009. – Vol. 37, iss. suppl_1. – P. D61–D65. https://doi.org/10.1093/nar/gkn837

11. CoryneRegNet 7, the reference database and analysis platform for corynebacterial gene regulatory networks / M. T. D. Parise [et al.] // Scientific Data. – 2020. – Vol. 7, no. 1. – P. 142. https://doi.org/10.1038/s41597-020-0484-9

12. RegPrecise 3.0 – A resource for genome-scale exploration of transcriptional regulation in bacteria / P. S. Novichkov [et al.] // BMC Genomics. – 2013. – Vol. 14. – P. 745. https://doi.org/10.1186/1471-2164-14-745

13. Nikolaichik, Y. SigmoID: a user-friendly tool for improving bacterial genome annotation through analysis of transcription control signals / Y. Nikolaichik, A. U. Damienikan // PeerJ. – 2016. – Vol. 4. – P. e2056. https://doi.org/10.7717/peerj.2056

14. Nikolaichik, Y. Genome-wide inference of bacterial transcription factor binding sites: new method and its applications / Y. Nikolaichik, P. Vychik // BMC Bioinformatics. – 2020. – Vol. 21, no. S20. – P. O2. https://doi.org/10.1186/s12859-020-03838-2

15. Nikolaichik, Y. New approach to genome-wide automated inference of bacterial transcription factor binding sites / Y. Nikolaichik, P. Vychik // Abstracts of the XII Intern. Multiconf. "Bioinformatics of Genome Regulation and Structure/Systems Biology". – Novosibirsk, 2020. – P. 75–76. https://doi.org/10.18699/BGRS/SB-2020-046

16. Sahota, G. Novel sequence-based method for identifying transcription factor binding sites in prokaryotic genomes / G. Sahota, G. D. Stormo // Bioinformatics. – 2010. – Vol. 26, no. 21. – P. 2672–2677. https://doi.org/10.1093/bioinformatics/btq501

17. Скакун, В. В. Системы управления базами данных : пособие / В. В. Скакун. – Минск : БГУ, 2020. – 159 с.

18. The Pfam protein families database: towards a more sustainable future / R. D. Finn [et al.] // Nucleic Acids Research. – 2016. – Vol. 44, no. D1. – P. D279–D285. https://doi.org/10.1093/nar/gkv1344

19. Letunic, I. 20 years of the SMART protein domain annotation resource / I. Letunic, P. Bork // Nucleic Acids Research. – 2018. – Vol. 46, no. D1. – P. D493–D496. https://doi.org/10.1093/nar/gkx922

20. Нильсен, П. SQL Server 2005. Библия пользователя : пер с англ. / П. Нильсен. – М. : Вильямс, 2008. – 1232 с.

21. Грофф, Д. П. SQL. Полное руководство : пер. с англ. / Д. П. Грофф, П. Н. Вайнберг, Э. Д. Оппель. – 3-е изд. – М. : Вильямс, 2016. – 960 с.

22. The MEME Suite / T. L. Bailey [et al.] // Nucleic Acids Research. – 2015. – Vol. 43, no. W1. – P. W39–W49. https://doi.org/10.1093/nar/gkv416

References

1. Van Hijum S. A. F. T., Medema M. H., Kuipers O. P. Mechanisms and evolution of control logic in prokaryotic transcriptional regulation. *Microbiology and Molecular Biology Reviews*, 2009, vol. 73, no. 3, pp. 481–509. https://doi.org/10.1128/MMBR.00037-08

2. Browning D. F., Busby S. J. W. Local and global regulation of transcription initiation in bacteria. *Nature Reviews Microbiology*, 2016, vol. 14, no. 10, pp. 638–650. https://doi.org/10.1038/nrmicro.2016.103

3. Stormo G. D. DNA binding sites: representation and discovery. *Bioinformatics*, 2000, vol. 16, no.1, pp. 16–23. https://doi.org/10.1093/bioinformatics/16.1.16

4. Rodionov D. A. Comparative genomic reconstruction of transcriptional regulatory networks in bacteria. *Chemical Reviews*, 2007, vol. 107, no. 8, pp. 3467–3497. https://doi.org/10.1021/cr068309+

5. Gelfand M. S. Evolution of transcriptional regulatory networks in microbial genomes. *Current Opinion in Structural Biology*, 2006, vol. 16, no. 3, pp. 420–429. https://doi.org/10.1016/j.sbi.2006.04.001

6. Lozada-Chavez I. Bacterial regulatory networks are extremely flexible in evolution. *Nucleic Acids Research*, 2006, vol. 34, no. 12, pp. 3434–3445. https://doi.org/10.1093/nar/gkl423

7. Perez J. C., Groisman E. A. Evolution of transcriptional regulatory circuits in bacteria. *Cell*, 2009, vol. 138, no. 2, pp. 233–244. https://doi.org/10.1016/j.cell.2009.07.002

8. Santos-Zavaleta A., Salgado H., Gama-Castro S., Sánchez-Pérez M., Gómez-Romero L., …, Collado-Vides J. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. *Nucleic Acids Research*, 2019, vol. 47, no. D1, pp. D212–D220. https://doi.org/10.1093/nar/gky1077

9. Kılıç S., White E. R., Sagitova D. M., Cornish J. P., Erill I. CollecTF: a database of experimentally validated transcription factor-binding sites in Bacteria. *Nucleic Acids Research*, 2014, vol. 42, iss. D1, pp. D156–D160. https://doi.org/10.1093/nar/gkt1123

10. Grote A., Klein J., Retter I., Haddad I., Behling S., …, Münch R. PRODORIC (release 2009): a database and tool platform for the analysis of gene regulation in prokaryotes. *Nucleic Acids Research*, 2009, vol. 37, iss. suppl_1, pp. D61–D65. https://doi.org/10.1093/nar/gkn837

11. Parise M. T. D., Parise D., Kato R. B., Pauling J. K., Tauch A., …, Baumbach J. CoryneRegNet 7, the reference database and analysis platform for corynebacterial gene regulatory networks. *Scientific Data*, 2020, vol. 7, no. 1, p. 142. https://doi.org/10.1038/s41597-020-0484-9

12. Novichkov P. S., Kazakov A. E., Ravcheev D. A., Leyn S. A., Kovaleva G. Y., …, Rodionov D. A. RegPrecise 3.0 – A resource for genome-scale exploration of transcriptional regulation in bacteria. *BMC Genomics*, 2013, vol. 14, p. 745. https://doi.org/10.1186/1471-2164-14-745

13. Nikolaichik Y., Damienikan A. U. SigmoID: a user-friendly tool for improving bacterial genome annotation through analysis of transcription control signals. *PeerJ*, 2016, vol. 4, p. e2056. https://doi.org/10.7717/peerj.2056

14. Nikolaichik Y., Vychik P. Genome-wide inference of bacterial transcription factor binding sites: new method and its applications. *BMC Bioinformatics*, 2020, vol. 21, no. S20, p. O2. https://doi.org/10.1186/ s12859-020-03838-2

15. Nikolaichik Y., Vychik P. New approach to genome-wide automated inference of bacterial transcription factor binding sites. *Abstracts of the XII International Multiconference "Bioinformatics of Genome Regulation and Structure/ Systems Biology"*. Novosibirsk, 2020, pp. 75–76. https://doi.org/10.18699/BGRS/SB-2020-046

16. Sahota G., Stormo G. D. Novel sequence-based method for identifying transcription factor binding sites in prokaryotic genomes. *Bioinformatics*, 2010, vol. 26, no. 21, pp. 2672–2677. https://doi.org/10.1093/bioinformatics/btq501

17. Skakun V. V. Sistemy upravlenija bazami dannyh. *Database Managements Systems*. Minsk, Belorusskij gosudarstvennyj universitet, 2020, 159 p. (In Russ.).

18. Finn R. D., Coggill P., Eberhardt R. Y., Eddy S. R., Mistry J., …, Bateman A. The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 2016, vol. 44, no. D1, pp. D279–D285. https://doi.org/10.1093/nar/gkv1344

19. Letunic I., Bork P. 20 years of the SMART protein domain annotation resource. *Nucleic Acids Research*, 2018, vol. 46, no. D1, pp. D493–D496. https://doi.org/10.1093/nar/gkx922

20. Nielsen P. *Microsoft SQL Server 2005 Bible*. 1st ed. Wiley, 2006, 1344 p.

21. Groff J. R., Weinberg P. N., Oppel A. J. *SQL: The Complete Reference*, 3rd ed. McGraw Hill, 2009, 912 p.

22. Bailey T. L., Johnson J., Grant C. E., Noble W. S. The MEME Suite. *Nucleic Acids Research*, 2015, vol. 43, no. W1, pp. W39–W49. https://doi.org/10.1093/nar/gkv416

Информация об авторах

Скакун Виктор Васильевич, кандидат физико-математических наук, доцент, заведующий кафедрой, Белорусский государственный университет.
https://orcid.org/0000-0003-0880-4188
E-mail: skakun@bsu.by

Николайчик Евгений Артурович, кандидат биологических наук, доцент, Белорусский государственный университет.
https://orcid.org/0000-0002-6718-9309
E-mail: nikolaichik@bsu.by

Information about the authors

Victor V. Skakun, Ph. D. (Phys.-Math.), Associate Professor, Head of Department, Belarusian State University.
https://orcid.org/0000-0003-0880-4188
E-mail: skakun@bsu.by

Yevgeny A. Nikolaichik, Ph. D. (Biol.), Associate Professor, Belarusian State University.
https://orcid.org/0000-0002-6718-9309
E-mail: nikolaichik@bsu.by