УДК 574:004.2

А.А. Фефелов, В.И. Литвиненко, М.А. Таиф, М.А. Вороненко

Объектно-ориентированная архитектура информационной системы реконструкции генных регуляторных сетей

Описана архитектура информационной системы, предназначенной для решения задач реконструкции генных регуляторных сетей, в основу которой положен объектно-ориентированный подход, определяющий ее открытость, универсальность и расширяемость. Предложен сценарий реконструкции, в котором осуществляется оптимизация пространства поиска значений параметров *S*-системы.

Ключевые слова: генные регуляторные сети, реверсная инженерия, экспрессия генов, *S*-система, алгоритм клонального отбора, информационная система, структурно-параметрическая идентификация.

Описано архітектуру інформаційної системи, призначеної для вирішення завдань реконструкції генних регуляторних мереж, в основу якої закладено об'єктно-орієнтований підхід, який визначає її відкритість, універсальність і розширюваність. Запропоновано сценарій реконструкції, в якому проводиться оптимізація простору пошуку значень параметрів *S*-системи.

Ключові слова: генні регуляторні мережі, реверсна інженерія, експресія генів, *S*-система, алгоритм клонального відбору, інформаційна система, структурно-параметрична ідентифікація.

Введение. Генные регуляторные сети (ГРС) – это сложные биологические системы, состоящие из множества взаимосвязанных компонентов и обладающие нелинейной динамикой. Недостаточный уровень понимания природы регуляции и механизмов функционирования ГРС не позволяет строить их математические модели, базируемые на фундаментальных законах взаимодействия компонентов. Однако современные исследования в области молекулярной биологии совместно с новейшими техническими достижениями, такими как ДНК-микрочипы, создали необходимые условия, при которых можно одновременно измерять уровни экспрессии множества генов, получая внушительные объемы данных, ранее недоступных для исследования [1]. Рост объема экспериментальной информации обусловил научный интерес к проблеме создания новых методов идентификации, позволяющих использовать данные экспрессии для реконструкции архитектуры и поведения ГРС.

Цель реконструкции ГРС – воспроизведение регуляторных взаимодействий и механизмов, функционирующих на уровне генов. На данный момент разработано множество различных моделей и методов реконструкции ГРС (от булевых сетей до систем обыкновенных дифференциальных уравнений), обладающих достоинствами и недостатками [2–5]. При выборе описательной модели необходимо учитывать, что математические модели, как правило, обладают собственной структурой и рядом параметров, которые необходимо настраивать (идентифицировать). Для структурно-параметрической идентификации моделей разработано большое количество вычислительных методов, многие из которых обладают повышенной устойчивостью к шумам и неопределенности, содержащимся в исходных данных. Данное свойство при выборе вычислительного метода актуально и для профилей экспрессии генов, несмотря на то, что, как правило, данные перед использованием подвергаются предобработке.

Концептуальная модель информационной системы реконструкции генных регуляторных сетей

Разработка архитектуры информационной системы (ИС) осуществлялась с учетом требований открытости, универсальности и взаимозаменяемости компонентов. Существенные различия в математическом аппарате и алгоритмах функционирования моделей и методов их настройки, требующих создания средств сопряжения между элементами системы, определили структуру ИС (рис. 1).

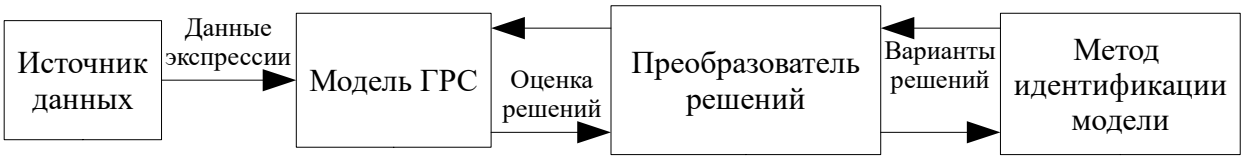

Рис. 1. Концептуальная модель ИС реконструкции ГРС

Центральный компонент ИС – модель ГРС и средства ее оценивания. В данной статье в качестве модели ГРС выбрана система обыкновенных дифференциальных уравнений, выраженная в форме *S*-системы. Метод идентификации модели представлен алгоритмом клонального отбора, в котором реализованы два способа кодирования индивидуумов: бинарное и вещественное. Блок идентификации – это тот компонент системы, в котором выполняется поиск оптимальной структуры и параметров модели ГРС в соответствии с данными экспрессии, поступающими в систему из источников данных. Из блока идентификации генерируемые варианты решений поступают в блок модели, где осуществляется их оценивание, которое в свою очередь влияет на алгоритмы идентификации. Для сопряжения модели и метода ее настройки введен блок преобразования решений. Здесь проводится кодирование и декодирование решений для преобразования их из формы, используемой в методе идентификации, в форму структуры и параметров конкретной модели ГРС и обратно. Благодаря наличию этого блока модель и метод независимы друг от друга по данным, что позволяет достаточно легко заменять соответствующие компоненты ИС, а также расширять ее архитектуру. Далее рассматривается функционирование каждого из выделенных блоков.

Модель генных регуляторных сетей

Системы обыкновенных дифференциальных уравнений (ОДУ) – наиболее точные модели, позволяющие максимально близко к реальности воспроизводить динамику ГРС. Чаще всего систему ОДУ представляют в форме *S*-системы [6], которая, с одной стороны, является нелинейной моделью и поэтому достаточно точно описывает генную сеть, а с другой, – обладая характерной структурой, дает возможность легко реконструировать топологию ГРС. В общем виде *S*-система выглядит так:

$$\frac{dx_i}{dt} = \alpha_i \prod_{J=1}^{N} x_j^{g_{ij}} - \beta_i \prod_{J=1}^{N} x_j^{h_{ij}}, \qquad (1)$$

где $x_i(t)$ – переменная состояния, выражающая концентрацию продукта экспрессии *i*-го гена в момент времени *t*; *N* – число компонентов (генов) в сети; параметры g_{ij} и h_{ij} определяют характер и степень воздействия гена x_j на ген x_i; α_i, β_i – неотрицательные коэффициенты.

Поскольку данная система ОДУ не имеет аналитического решения, ее решают одним из методов численного интегрирования, например методом Рунге–Кутты. Принимая во внимание сказанное, получена следующая структура классов блока модели ГРС (рис. 2).

Класс *SSystemDomain* представляет собой архетип модели генной сети. Он инкапсулирует объект класса *SSystemModel*, в котором выполняется вычисление правой части системы (1). Класс *SSystemModel* реализует интерфейс *RealValuedModel*, используемый объектами *OdeSystemEvaluator* в качестве подынтегральной функции. В классе *OdeSystemEvaluator* осуществляется решение системы ОДУ одним из выбранных методов (*Methods*). *OdeSystemEvaluator* получает данные о начальных условиях из источника данных (класс *RealDataCacher*). Для оценивания текущего решения используется интерфейс *ErrorMeasure*, реализуемый классом *StdErrorMeasure*, где вычисляется ошибка модели на временных рядах данных экспрессии генов. В настоящей статье для расчета ошибки применяется следующее выражение [7]:

$$f = \sum_{i=1}^{N} \sum_{j=1}^{T} \left(\frac{x_i^M(t_0 + j\Delta t) - x_i(t_0 + j\Delta t)}{x_i(t_0 + j\Delta t)} \right)^2, \qquad (2)$$

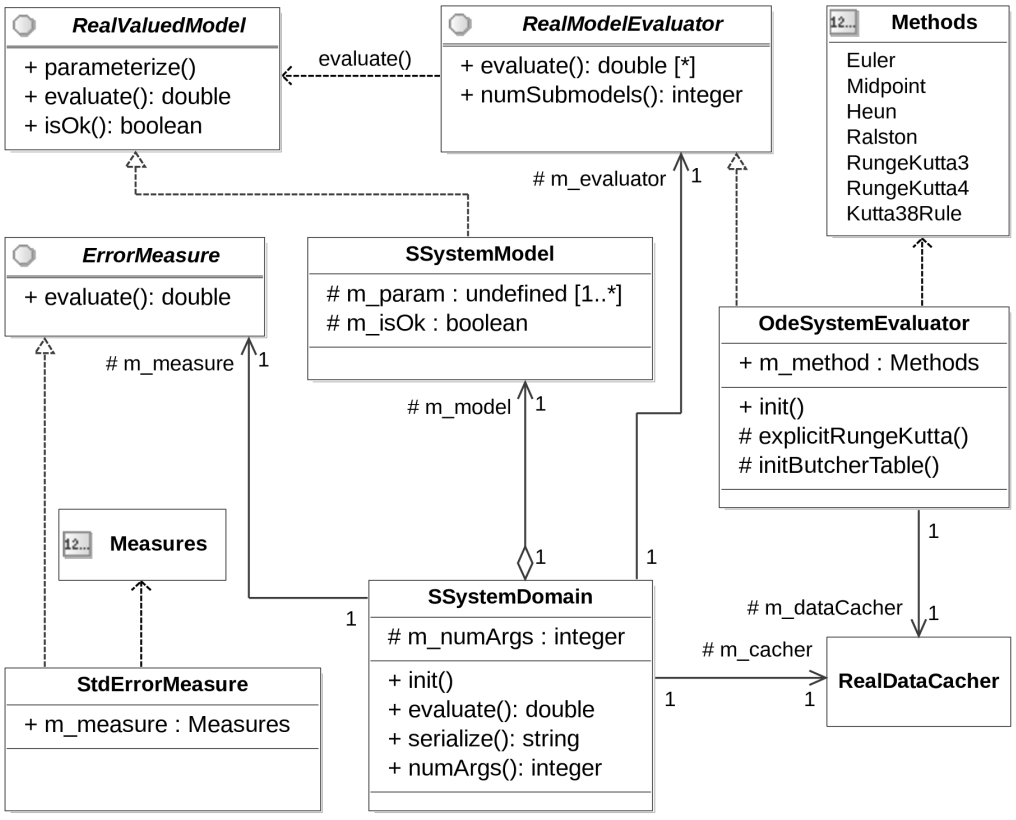

Рис. 2. Диаграмма классов блока модели ГРС

где t_0 – начальное время; Δt – временной шаг; T – количество данных временного ряда экспрессии; $x_i^M\left(t_0 + j\Delta t\right)$ – значения концентрации, полученные решением системы ОДУ (1); $x_i\left(t_0 + j\Delta t\right)$ – наблюдаемые значения концентрации из временного ряда экспрессии (класс *RealDataCacher*).

Вместо класса *SSystemDomain* можно использовать другие типы вычислительных моделей, например, нейронную сеть [8], взвешенную сумму [9] или генетическую программу [10]. В этом случае разработанная структура классов позволяет либо заменить соответствующий компонент, либо расширить архитектуру, добавив новый архетип и ассоциировав его с теми же объектами, что и у класса *SSystemDomain*.

Метод идентификации модели

Основная трудность идентификации моделей в форме *S*-системы – это высокая размерность задачи. Количество параметров, которые необходимо найти, определяется выражением $2N(N+1)$. В решении подобных задач оптимальные результаты показывают искусственные иммунные системы (ИИС) [11] – производные модели, в основу которых положены результаты исследований теоретической иммунологии. Согласно этим исследованиям естественные иммунные механизмы высших существ обладают признаками, свойственными системам распознавания образов. Специфические свойства иммунной системы как распределенной, децентрализованной и неоднородной структуры положены в основу различных моделей, описывающих процессы поиска, обнаружения, распознавания и защиты организма от чужеродных агентов – вирусов и бактерий. Одна из таких моделей – клональный отбор [12], функция которого заключается в последовательной адаптации популяции антител – вариантов решения задачи к входящему антигену – целевой функции. Во время адаптации популяция антител подвергается ряду циклических воздействий, таких как: селекция, клонирование, мутация и повторная селекция. Одним из параметров работы каждой из фаз

является мера аффинности, выражающая степень близости индивидуума к оптимальному решению.

В данной статье алгоритм клональной селекции выбран в качестве метода идентификации модели ГРС. Кодирование решений и псевдокод алгоритма клонального отбора показаны на рис. 3 и 4.

Объектная декомпозиция модуля ИС, осуществляющего структурно-параметрическую идентификацию модели генной сети, показана на рис. 5.

α_1	β_1	g_{11}	...	g_{1N}	h_{11}	...	h_{1N}	...	α_i	β_i	g_{i1}	...	g_{iN}	h_{i1}	...	h_{iN}	...	α_N	β_N	g_{N1}	...	g_{NN}	h_{N1}	...	h_{NN}

Рис. 3. Структура антитела, кодирующего *S*-систему

Поколение = 0
Создать начальную популяцию антител (Ab^0)
цикл пока условие_останова *e* = false
    Поколение = *Поколение* + 1
    Вычислить аффинность антител к антигену (*f*)
    Выбрать антитела с наибольшей аффинностью
    Создать клоны выбранных антител (в количестве *n*~*f*)
    Провести мутацию клонов (с вероятностью $p_{мут}$!~*f*)
    Вычислить аффинность клонов к антигену (f_c)
    Выбрать клоны с наибольшей аффинностью
    Перенести клоны в основную популяцию
    Заменить *d* антител с наименьшей аффинностью
конец цикла пока

Рис. 4. Псевдокод алгоритма клонального отбора

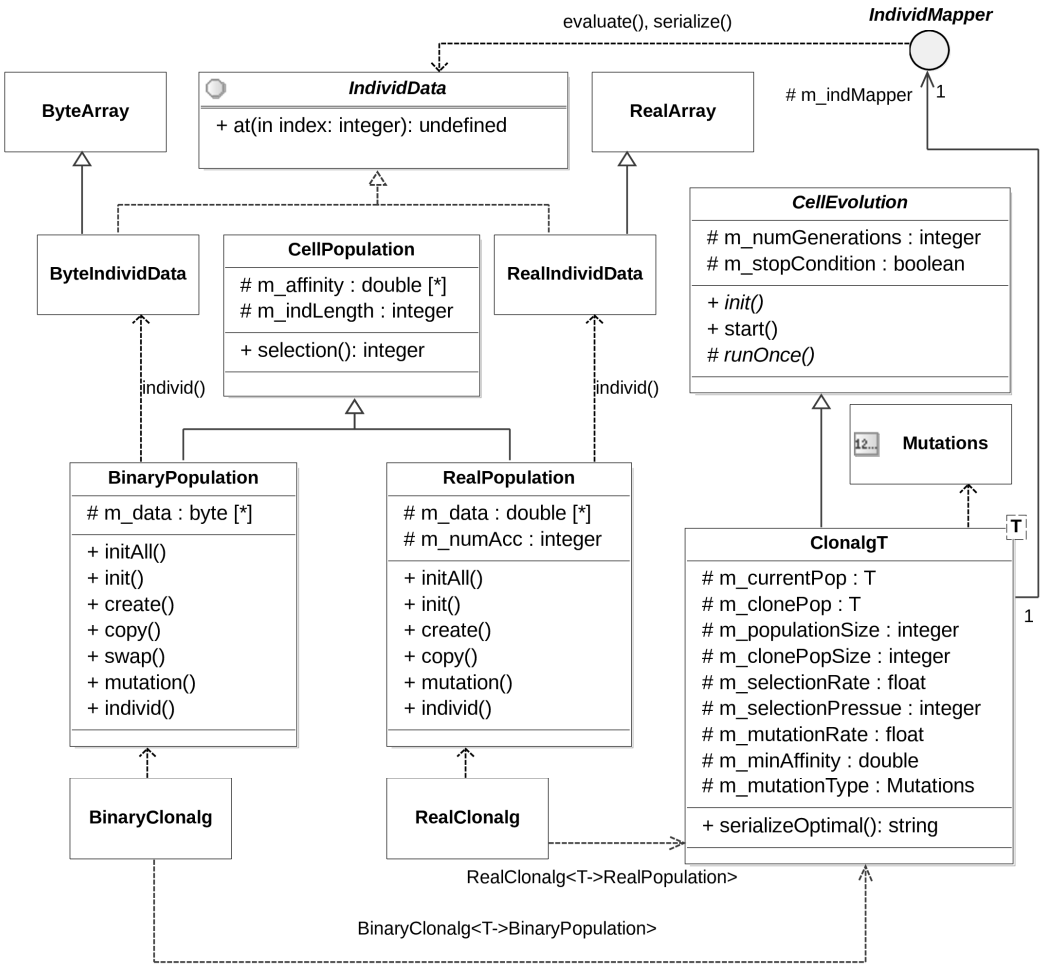

Рис. 5. Диаграмма классов блока идентификации модели

В ИИС, в зависимости от типа задачи, можно использовать различные способы представления решений. Наиболее часто применяются бинарное и вещественное представления, при которых в первом случае антитело формируется как непрерывная строка бит, а во втором – как вектор вещественных чисел. Оба варианта представления поддерживаются разработанной ИС. Класс *ByteIndividData* соответствует антителу с бинарным кодированием. Класс *RealIndividData* представляет индивидуумы с вещественным кодированием. Оба класса являются наследниками векторной модели данных и реализуют один интерфейс *IndividData*. Единый интерфейс удобен для доступа к данным антитела из других модулей ИС. Для каждого представления созданы отдельные классы популяции: *BinaryPopulation* и *RealPopulation*. В этих классах реализованы основные методы воздействия на индивидуумы, такие как селекция, клонирование, мутация. Клональный алгоритм наследует абстрактный класс *CellEvolution*, разработанный для создания возможности расширения архитектуры другими эволюционными методами, например, генетическими алгоритмами. Класс клонального алгоритма *ClonalgT* реализован в виде шаблона, позволяющего абстрагироваться от способов представления индивидуумов. Его наследуют классы *BinaryClonalg* и *RealClonalg*, работающие с конкретными типами данных.

Преобразователь решений и источник данных

Два блока ИС выполняют функции, обеспечивающие нормальное функционирование остальных конструкций (рис. 6, 7).

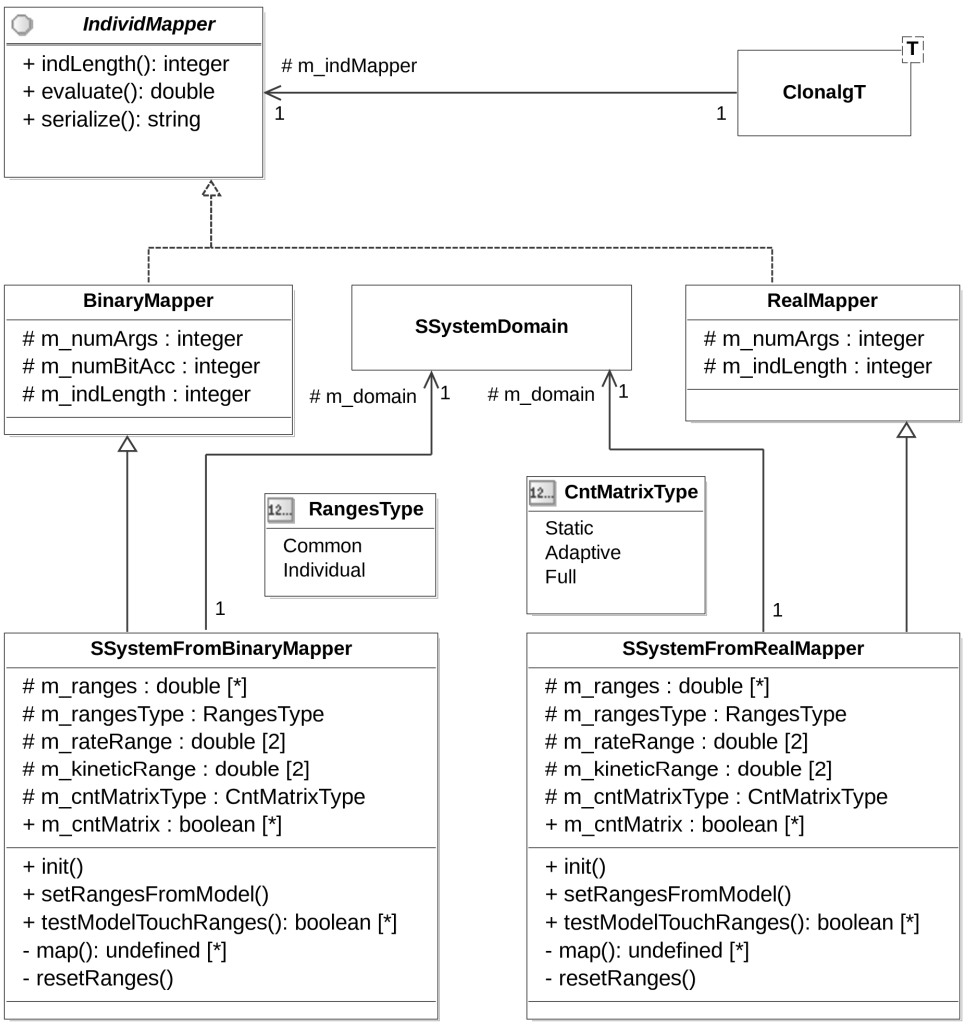

Рис. 6. Диаграмма классов блока преобразования решений

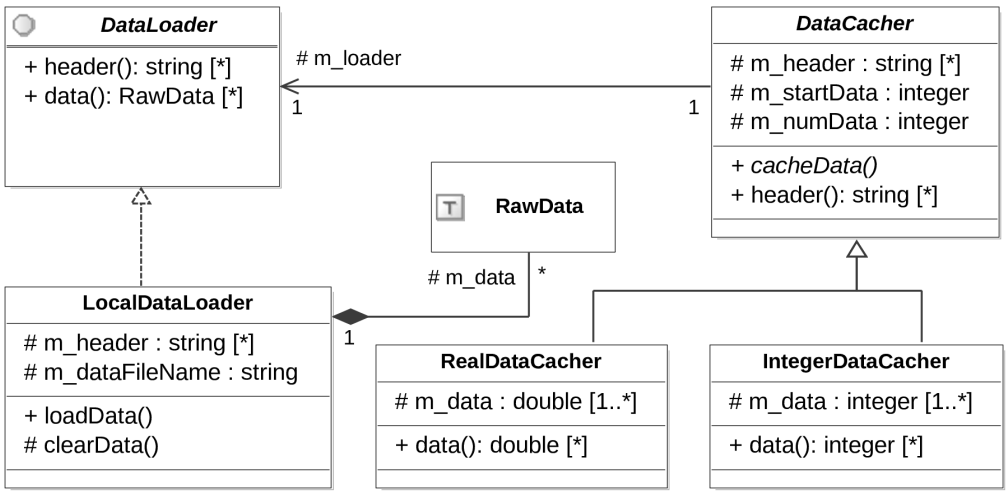

Рис. 7. Диаграмма классов блока источника данных

Как упоминалось ранее, преобразователь решений служит связующим звеном между моделью и методом, обеспечивая им независимое функционирование. В диаграмме (см. рис. 6) модель представлена архетипом *SSystemDomain*, а метод – шаблоном *ClonalgT*. Метод ассоциирован с моделью через интерфейс *IndividMapper*, в котором декларируется операция *evaluate*(), обеспечивающая передачу данных варианта решения в блок модели и возвращающая значение оценки этого варианта решения. Интерфейс *IndividMapper* реализован классами *SSystemFromBinaryMapper* и *SSystemFromRealMapper* в соответствии с бинарным и вещественным представлением решений. Преобразователь решений включает в себя ряд существенных свойств, основными из которых являются верхняя и нижняя границы интервалов представления значений (*m_ranges*, *m_rateRange*, *m_kineticRange*). Данные интервалы определяют границы гиперкуба, внутри которого алгоритм идентификации ведет поиск оптимального решения. Границы интервалов можно сделать вариативными, что создаст дополнительные преимущества при решении задач идентификации.

Блок источника данных организован так, чтобы, с одной стороны, обеспечивать загрузку и хранение данных разных типов, а с другой – давать возможность быстрого доступа к ним при поддержке производительности ИС на высоком уровне. Класс *LocalDataLoader* служит контейнером для объектов типа *RawData*, который хранит в себе запись с произвольным количеством полей, содержащих данные любого типа. В данной статье *LocalDataLoader* обеспечивает загрузку данных с локального носителя (диска). Однако наличие интерфейса *DataLoader* позволяет расширять ассортимент загрузчиков и возможность загрузки по сети или с внешнего устройства (рис. 7).

Прямой доступ к данным в формате *RawData* слишком медленный, что негативно сказывается на производительности ИС. Обращение к данным осуществляется на протяжении всего времени работы системы через блок модели, где они сравниваются с данными модельного эксперимента для каждого варианта решения. Для ускорения доступа к данным авторами введен абстрактный класс *DataCacher*, имеющий конкретные реализации: *RealDataCacher* и *IntegerDataCacher*. Данные кешируются строго по типам (что видно по названиям классов) и ассоциируются с теми объектами системы, которые могут обратиться к ним во время выполнения программы.

Взаимодействие объектов в системе

Объекты ИС взаимодействуют посредством ассоциативных связей, процесс установки которых подобен построению блочной конструкции, где каждый отдельный блок выполняет какое-либо элементарное действие, а все блоки в совокупности решают общую задачу. Такой подход дает определенную свободу раз-

работчикам и даже пользователям, позволяя вносить коррективы в структуру системы, оптимизирующие ее работу. Пример диаграммы взаимодействия объектов ИС с расширенной структурой показан на рис. 8. Как видно из рисунка, в системе приведены два блока идентификации, использующих разные способы представления решений (*algBin* и *algReal*) и, соответственно, два преобразователя решений (*mapperBin* и *mapperReal*). При этом оба алгоритма идентификации могут функционировать в параллельном режиме. Между ними может быть организовано взаимодействие по аналогии с меметическим алгоритмом.

Работа ИС демонстрируется на рис. 9, где показана диаграмма деятельности, реализующая один из сценариев решения задачи реконструкции ГРС. В сценарии используются объекты класса *SSystemFromBinaryMapper* для оптимизации границ интервалов значений, которые могут принимать параметры *S*-системы в процессе решения задачи структурно-параметрической идентификации. В каждой итерации сценария выполняется однократный запуск алгоритма идентификации, и в зависимости от полученного результата осуществляется коррекция интервалов согласно следующим соотношениям, предложенным авторами:

$$R_i = \left(\upsilon_{i-1}^{opt} - \frac{s_i |R_0|}{2}, \upsilon_{i-1}^{opt} + \frac{s_i |R_0|}{2} \right), \quad (3)$$

$$s_i = \begin{cases} s_{i-1} k_{g,\ \text{если}} \left| \upsilon_{i-1}^{opt} - r_{i-1}^{l} \right| \le \varepsilon \text{ или } \left| \upsilon_{i-1}^{opt} - r_{i-1}^{r} \right| \le \varepsilon \\ s_{i-1} k_{s^r,\ \text{в противном случае}} \end{cases},$$

где R_i – интервал значений, которые может принимать параметр *S*-системы в *i*-й итерации сценария; υ_{i-1}^{opt} – оптимальное значение параметра *S*-системы, полученное в $(i-1)$-й итерации сценария; R_0 – начальный интервал значений параметра *S*-системы; $k_g \left(k_g > 1\right)$ – коэффициент расширения интервала; $k_s \left(0 < k_s < 1\right)$ – коэффициент сжатия интервала; $R_{i-1}\left(r_{i-1}^{l}, r_{i-1}^{r}\right)$ – интервал значений, которые может принимать параметр *S*-системы в $(i-1)$-й итерации сценария; ε – пороговое значение, фиксирующее факт достижения параметром υ_{i-1}^{opt} левой или правой границы интервала R_{i-1}. Применительно к *S*-системе обозначение υ подразумевает один из параметров α, β, *g* или *h*.

По сути данный сценарий выполняетт поочередную оптимизацию решений внутри фиксированного пространства поиска и оптимизацию собственно пространства относительно текущего лучшего решения.

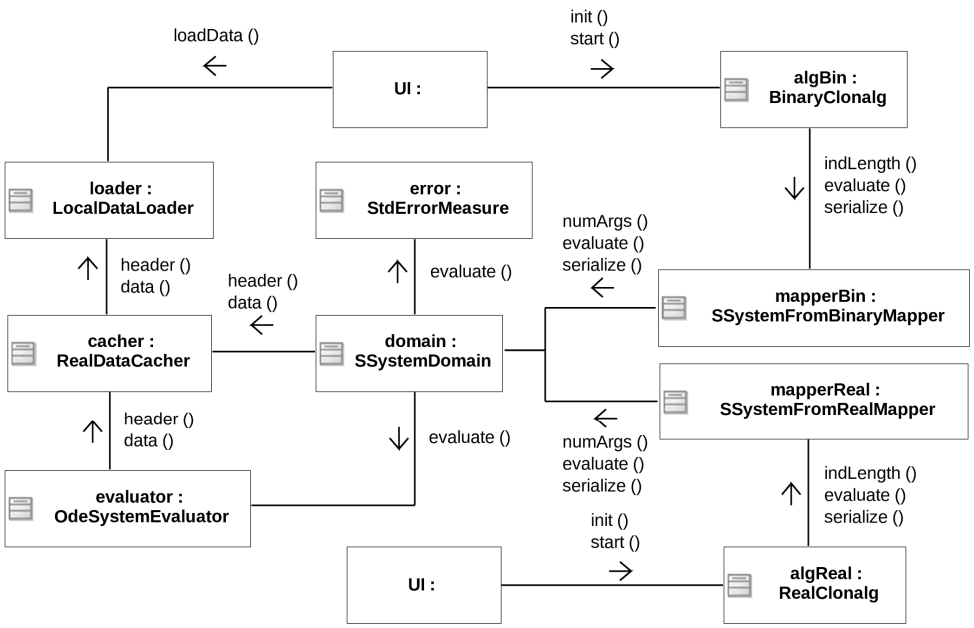

Рис. 8. Диаграмма взаимодействия объектов системы

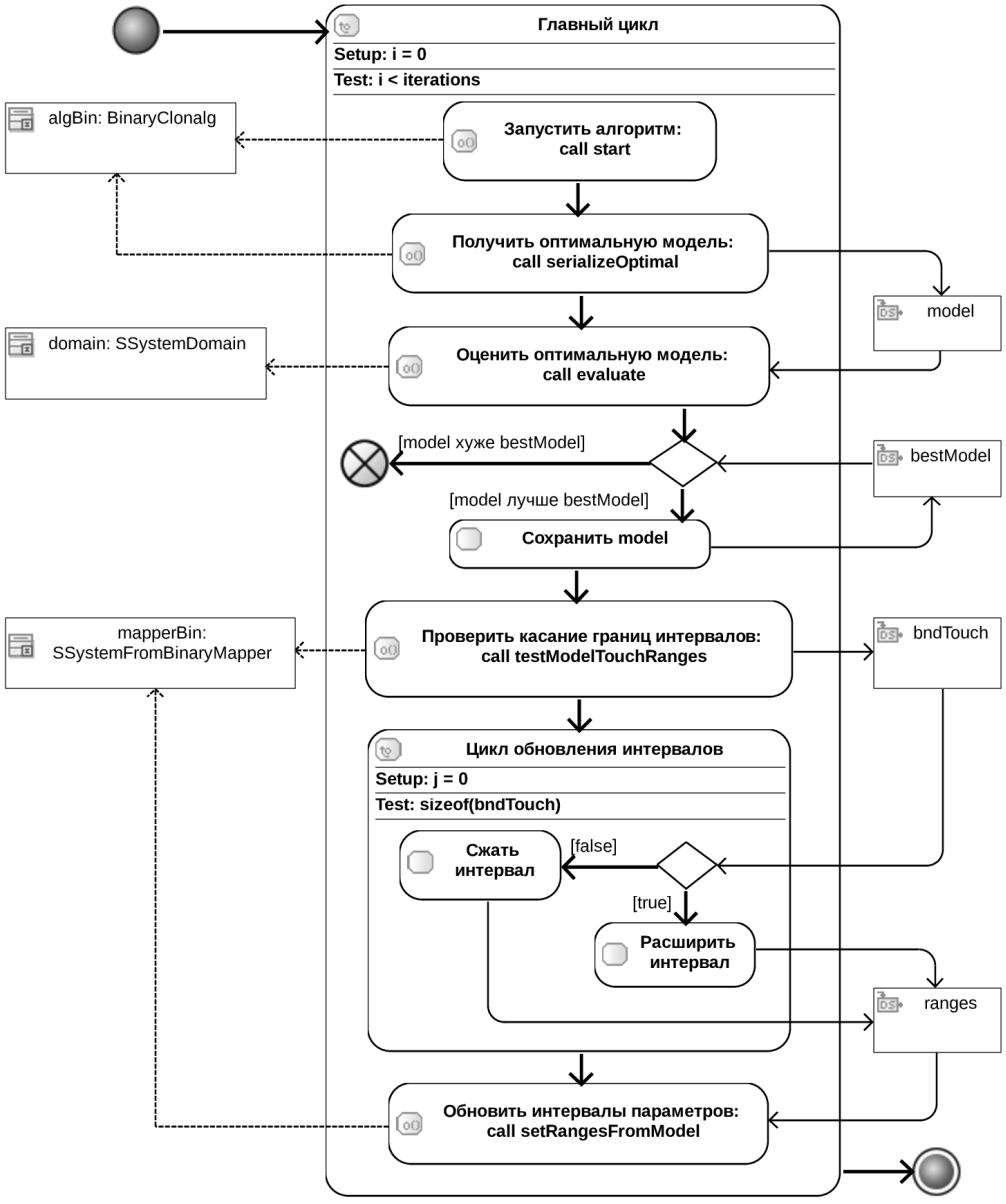

Рис. 9. Диаграмма деятельности сценария решения задачи реконструкции ГРС

Заключение. Архитектура информационной системы реконструкции генных регуляторных сетей, основана на использовании объектно-ориентированного подхода. Архитектура открыта, имеет возможность расширения, добавления или замещения отдельных компонентов. В минимальном исполнении система содержит четыре компонента: источник данных, модель, преобразователь решений и метод идентификации. В качестве вычислительной модели выбрана *S*-система, которая подвергается структурно-параметрической идентификации методом клонального отбора. На основе разработанной архитектуры предложен сценарий решения задачи реконструкции генной сети, в котором реализован итерационный алгоритм оптимизации пространства поиска значений параметров вычислительной модели. Дальнейшие исследования предполагают расширение ассортимента используемых моделей (добавление в ИС модели радиально-базисной и вейвлет-нейронной сети, а также системы программирования экспрессии генов) и новых эволюционных алгоритмов, усовершенствова-

ние работы эволюционных операторов и разработку новых сценариев для решения задач реконструкции генных сетей.

1. *DeRisi J.L.*, *Lyer V.R.*, *Brown P.O.* Exploring the metabolic and genetic control of gene expression on a genomic scale // Science. – 1997. – **278**. – P. 680–686.
2. *Akutsu T.*, *Miyano S.*, *Kuhara S.* Identification of genetic networks from a small number of gene expression patterns under the boolean network model // Pacific Symposium on Biocomputing (PSB'99), Jan. 1999: proc. – Maui, Hawaii, USA, 1999. – P. 17–28.
3. *Bower J.M.*, *Bolouri H.* Computational Modeling of Genetic and Biochemical Networks. – The MIT Press, 2001. – 336 p.
4. *Development* of a system for the inference of large scale genetic networks / Y. Maki, D. Tominaga, M. Okamoto et al. // Pacific Symposium on Biocomputing (PSB'01), Jan. 2001: proc. – Maui, Hawaii, USA, 2001. – P. 446–458.
5. *Gene* networks inference using dynamic bayesian networks / B.-E. Perrin, L. Ralaivola, A. Mazurie et al. // Bioinformatics. – 2003. – **19**(2). – P. 138–148.
6. *Savageau M.A*. Introduction to S-systems and the underlying power-law formalism // Mathematical and Computer Modelling. – 1988. – **11**. – P. 546–551.
7. *Tominaga D.*, *Koga N.*, *Okamoto M.* Efficient numerical optimization algorithm based on genetic algorithm for inverse problem // Genetic and Evolutionary Computation Conference (GECCO'00), July 2000: proc. – Las Vegas, Nevada, USA, 2000. – **251**. – P. 251–258.
8. *Hybrid* Approach for Gene Regulatory Networks Reconstruction / A.A. Fefelov, V.I. Lytvynenko, M.A. Taif et al. // Upr. Sist. Mas., 2017. – № 3. – P. 63–72.
9. *Weaver D.C.*, *Workman C.T.*, *Stormo G.D.* Modeling regulatory networks with weight matrices // Pacific Symposium on Biocomputing 4 (PSB'99), 4–9 Jan. 1999: proc. – Maui, Hawaii, USA, 1999. – P. 112–123.
10. *Sakamoto E.*, *Iba H.* Inferring a system of differential equations for a gene regulatory network by using genetic programming // Congress on Evolutionary Computation, 27–30 May 2001: proc. – COEX, Seoul, Korea, 2001. – P. 720–726.
11. *De Castro L.N.*, *Timmis J.* Artificial Immune Systems: A New Computational Intelligence Approach. – Heidelberg: Springer, 2002. – 357 p.
12. *De Castro L.N.*, *Von Zuben F.J.* Learning and optimization using the clonal selection principle // IEEE Transactions on Evolutionary Computation. – 2002. – **6**(3). – P. 239–251.

Поступила 15.08.2017

E-mail: *fao*1976@*ukr.net*, *immun*56@*gmail.com*, *taifmohamedali@gmail.com*, *mary_voronenko@i.ua*

A.A. Fefelov[1], V.I. Lytvynenko[2], M.A. Taif[3], M.A. Voronenko[4]

[1] PhD in Techn. Sciences, Associate Professor, Department of Design of Kherson National Technical University, Bereslavskoe Shosse, 24, Kherson, 73008, Ukraine, fao1976@ukr.net

[2] Doctor of Technical Sciences, Professor, Head of the Department of Informatics and Computer Science of Kherson National Technical University, Bereslavskoe Shosse, 24, Kherson, 73008, Ukraine, immun56@gmail.com

[3] Graduate student of the Department of Informatics and Computer Science, Kherson National Technical University, Bereslavskoe Shosse, 24, Kherson, 73008, Ukraine, taifmohamedali@gmail.com

[4] PhD in Techn. Sciences, Associate Professor, the Department of Informatics and Computer Science of Kherson National Technical University, Beryslavske highway, 24, Kherson, 73008, Ukraine, mary_voronenko@i.ua

Object-Oriented Architecture of the Information System for the Reconstruction of the Gene Regulatory Networks

Keywords: gene regulatory networks, reverse engineering, gene expression, *S*-system, clonal selection algorithm, information system, structural-parametric identification.

Introduction. Insufficient level of understanding of the nature of regulation and functional mechanisms of gene regulatory networks does not allow to build their mathematical models, based on the fundamental laws of component's interaction. Now, many different models and methods of gene regulatory reconstruction are developed, which have the advantages and disadvantages. At the choice of descriptive model it is necessary to consider the fact, that mathematical models, as a rule, have their own structure and a number of parameters, which need to be identified. A large number of computational methods are developed, for structural-parametric model's identification. The majority of them have increased resistance to noise and uncertainty contained in the initial data. The presence of this property is real for the selection of a computational method, used for solving the reconstructing problem of the gene regulatory network based on the gene expression data.

Purpose. The purpose of this work is the development of the information system architecture for the gene regulatory network reconstruction, based on the object-oriented approach.

Method. The authors used the method of object-oriented design for the developing of this information system.

Окончание на стр. 82