УДК 519.217.2

И.В. СЕРГИЕНКО, Б.А. БЕЛЕЦКИЙ, А.М. ГУПАЛ

ЖИВАЯ КЛЕТКА КАК КОМПЬЮТЕР ОБЩЕГО НАЗНАЧЕНИЯ

Ключевые слова: *вычислительные схемы, универсальная вычислительная схема, компьютер общего назначения, молекулярные вычисления, внутриклеточные процессы жизнедеятельности.*

ВВЕДЕНИЕ

В настоящее время механизм реализации белковой функции остается неясным, несмотря на всеобщее убеждение, что функция белка определяется его структурой.

Стремительное развитие вычислительной техники во второй половине прошлого века совпало во времени с фундаментальными открытиями в молекулярной биологии, позволившими по-новому взглянуть на процессы жизнедеятельности, протекающие в живой клетке. Выяснилось, что на молекулярном уровне она устроена чрезвычайно сложно, а эффективность организации протекающих в ней процессов, позволяет сравнивать живую клетку с молекулярным компьютером.

На пути развития вычислительной техники возникали различные проблемы, решение которых требовало все более эффективной организации вычислительных процессов. Накопленные экспериментальные данные подтверждают, что живые организмы сталкивались с подобными проблемами в процессе эволюции, а аналогичные решения были найдены миллиарды лет тому назад, задолго до появления на Земле человека.

Огромные усилия направлены на объяснение принципов функционирования живой клетки, при этом процессы жизнедеятельности изучаются на молекулярном уровне. Используемые для этого физические модели, как правило, характеризуются высокой сложностью, являясь слишком низкоуровневыми для описания процессов жизнедеятельности на клеточном уровне. Аналогично законы электродинамики описывают принципы работы компьютера и не дают полного представления о протекающих в нем вычислительных процессах. Для описания вычислений используются языки программирования. Предполагается, что в живой природе существует некий аналог языка компьютерного программирования, позволяющий описывать клеточные процессы жизнедеятельности.

ВЫЧИСЛИТЕЛЬНЫЕ СХЕМЫ

Важным событием первой половины прошлого века, предшествовавшим бурному развитию вычислительной техники, стало появление вычислительных схем. Они использовались для формализации понятия алгоритма и позволяли описывать сложные вычисления в виде последовательности элементарных шагов. Предлагалось множество вычислительных схем, например рекурсивные функции Геделя–Клини, исчисление Черча, машина Тьюринга, машина Поста и др. Впоследствии выяснилось, что хотя принципы работы, лежащие в основе предложенных вычислительных схем, различны, они эквивалентны. Этот неожиданный факт позволил сформулировать интуитивное представление о вычислимости в виде тезиса Черча–Тьюринга, гласящего, что для любой интуитивно вычислимой функции существует вычисляющая ее значения машина Тьюринга [1].

Рассмотрим устройство и принцип работы наиболее популярной вычислительной схемы. Машина Тьюринга состоит из бесконечной в обе стороны ленты, разбитой на ячейки, где хранятся значения из конечного алфавита A, а также управляющего устройства, находящегося в одном из конечного множества состояний Q, которое способно перемещаться вдоль ленты, считывать и изменять

значения в ячейках. Программа машины Тьюринга имеет вид конечного множества правил перехода вида $t:(q,a)\to(q',a',d)$, где $q\in Q$ — исходное состояние управляющего устройства, $a\in A$ — символ, считываемый им с ленты, $q'\in Q$ — новое состояние управляющего устройства, $a'\in A$ — новый символ, записываемый им на ленту, $d\in\{\leftarrow,\to\}$ — направление, в котором управляющее устройство сдвигается на одну ячейку вдоль ленты.

При выполнении вычислений исходные данные подаются на вход соответствующей машине Тьюринга в виде конфигурации символов на ленте, а результат вычислений появляется на ленте по достижению ее терминального состояния.

Любую конфигурацию символов на ленте можно однозначно закодировать в виде натурального числа, тогда любая машина Тьюринга M будет представлена в виде функции $f_M:\mathbf{N}\mapsto\mathbf{N}$. Аналогично программу машины Тьюринга M можно закодировать, поставив ей в соответствие ее код $c(M)\in\mathbf{N}$, по которому всегда однозначно восстанавливается описание исходной машины M. Множество всех машин Тьюринга не боле чем счетно, тогда как множество $\mathbf{N}^{\mathbf{N}}$ всех функций, действующих из множества натуральных чисел в себя, имеет мощность континуума. Таким образом, подавляющее большинство функций такого вида невычислимы.

УНИВЕРСАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СХЕМЫ И АРХИТЕКТУРА ФОН НЕЙМАНА

Возможность закодировать произвольную машину Тьюринга в виде натурального числа позволяет ввести понятие универсальной машины Тьюринга U, с помощью которой можно имитировать работу любой машины Тьюринга M на произвольном входе. Универсальная машина Тьюринга U представляется в виде функции $f_U:\mathbf{N}\times\mathbf{N}\mapsto\mathbf{N}$, принимающей на вход пару натуральных чисел: код $c(M)$ самой машины M, работу которой требуется имитировать, а также код $k\in\mathbf{N}$ исходной конфигурации ленты. При этом для произвольной начальной конфигурации с кодом k имеет место равенство $f_U(c(M),k)=f_M(k)$.

Вычислительная схема, позволяющая определять результат работы любой машины Тьюринга на произвольном входе, называется универсальной. Идея универсальности использовалась Тьюрингом для доказательства неразрешимости проблемы остановки в общем случае. Кроме теоретической ценности, эта идея имела огромное практическое значение: появившаяся возможность хранить программы в памяти вместе с данными стала началом эры программируемых компьютеров.

В 1945 г. фон Нейман, используя идею универсальности, предложил архитектуру компьютера общего назначения (рис. 1), которая лежит в основе большинства современных вычислительных устройств: программы хранились в памяти вместе с данными, что делало компьютер программируемым и позволяло его использовать для выполнения любых программ, которые можно описать.

Компьютер общего назначения имел четыре составляющих:

- постоянное запоминающее устройство (ПЗУ) или энергонезависимая память, позволяющая хранить информацию после выключения компьютера, надежность хранения информации компенсируется относительно медленной работой, поскольку для доступа к случайной ячейке ПЗУ, как правило, требуется перемещение считывающего устройства относительно накопителя данных (в современных компьютерах ПЗУ соответствует жесткому диску);
- оперативное запоминающее устройство (ОЗУ) или энергозависимая память, позволяющая хранить информацию только во время работы компьютера, для которой свойственно

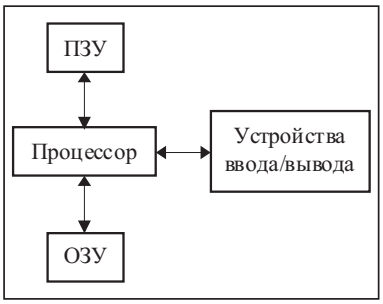

Рис. 1. Архитектура фон Неймана компьютера общего назначения

высокое быстродействие, поскольку в любой момент времени имеется доступ к каждой ячейке памяти (в современных компьютерах ОЗУ соответствует оперативной памяти);

• устройства ввода/вывода, позволяющие осуществлять обмен данными с компьютером в процессе вычислений (в современных компьютерах такими устройствами являются клавиатура, монитор или сетевая карта);

• процессор, изменяющий состояние ОЗУ и ПЗУ в ходе вычислений, а также координирующий обмен данными между всеми взаимодействующими устройствами.

ЖИВАЯ КЛЕТКА КАК РЕАЛИЗАЦИЯ УНИВЕРСАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СХЕМЫ

В 1944 г. была опубликована книга Е. Шредингера [2], в которой анализировались достижения классической генетики, а также выдвигались предположения, положившие начало новой эры молекулярной генетики. Ко времени выхода книги было сформулировано понятие гена как единицы наследственности, а также установлено, что гены локализованы в хромосомах, где они организованы линейно, что позволило составить первые генетические карты, описывавшие порядок следования известных генов в хромосомах. Структура ДНК и белков, из которых состоят хромосомы, еще не была известна, поэтому роль этих молекул в процессах передачи и реализации наследственной информации оставалась неясной.

Шредингер также отмечал важную роль информации в процессах жизнедеятельности. Оценив размеры гена, он пришел к выводу, что хранение и реализация генетической информации осуществляются отдельными молекулами, а не их ансамблями, и эти процессы иногда имеют скорее детерминированный, чем статистический характер. Роль носителя генетической информации Шредингер отводил некоторой достаточно большой молекуле, названной им апериодическим кристаллом, которая хранит информацию в структуре собственных химических связей. Описав физические свойства этой молекулы, он определил направление для ее поиска.

Удивительную способность организма развиваться из единственной клетки Шредингер объяснял тем, что в хромосомах содержатся его описание и инструменты его построения согласно этому описанию. Во время деления клетки количество хромосом удваивается, при этом описание организма копируется вместе с инструментами его построения и каждой дочерней клетке достается по одной копии. В результате она содержит все необходимое для дальнейшего деления.

В 1948 г. фон Нейман сделал уточнение [3], предположив, что описания организма и инструментов его построения хранятся в хромосомах аналогично тому, как данные вместе с программами хранятся в памяти универсальной машины Тьюринга. Фон Нейман предложил архитектуру самовоспроизводящегося автомата, флуктуирующего в среде с деталями, который состоял из трех компонентов: управляющего модуля, координирующего взаимодействие всех составляющих компонентов; универсального конструктора, собирающего новый автомат из деталей, флуктуирующих в окружающей среде; инструкций, описывающих процесс сборки этого нового автомата. Следуя входящим в его состав инструкциям, автомат мог создавать свою идентичную копию. При этом универсальный конструктор под управлением управляющего модуля сначала собирал по инструкциям новую копию автомата. После этого аналогично создавалась новая копия инструкций и подавалась на вход нового автомата. Универсальность конструктора заключалась в его способности собирать широкий спектр структур, не ограничиваясь только используемыми при построении новой копии автомата.

В 1953 г. открыли структуру молекулы ДНК, и стала очевидна ее ключевая роль в передаче наследственной информации. Молекула ДНК имеет вид двух комплементарных цепей, состоящих из нуклеотидов четырех типов: A, T, C, G. Цепи соединены таким образом, что нуклеотид типа A одной цепи всегда соеди-

няется с нуклеотидом типа T комплементарной цепи, аналогичное свойство выполняется для пары нуклеотидов C и G. Каждая цепь однозначно определяет нуклеотидную последовательность комплементарной цепи, что делает молекулу ДНК удобной для хранения и копирования генетической информации. В процессе клеточного деления белковая машина (реплисома) собирает копию молекулы ДНК из нуклеотидов, флуктуирующих в окружающей среде. При этом белки самой реплисомы закодированы в копируемой ею молекуле ДНК.

Предположения, выдвинутые фон Нейманом, окончательно подтвердились к 1970 г., когда был описан механизм реализации генетической информации. Последняя реализуется в клетке за счет синтеза белков — высокоактивных органических молекул, выполняющих разнообразные функции в клетке, среди которых формирование структуры клетки, копирование и реализация генетической информации, координация процессов обмена веществ, транспортной функции и др. Белки — крупные молекулы, составляющие приблизительно треть сухой массы клетки, являются спутанными в клубки цепочками, состоящими из аминокислот 20 типов, остальные молекулы можно сравнить с инертным фоном, на котором действуют белки. Функция белка зависит от пространственной структуры его клубка, определяющейся в свою очередь последовательностью аминокислот в белковой цепи.

В основе работы механизма реализации генетической информации лежит рибосома — белковый комплекс, выполняющий роль универсального конструктора в живой клетке. Рибосома собирает новые белки по инструкциям, хранящимся в структуре ДНК. С помощью рибосомы собираются белки, которые входят в ее состав, инструкции по сборке рибосомных белков хранятся в ДНК вместе с остальными белками. Универсальность рибосомы заключается в ее способности производить огромное количество различных белков, которое значительно превосходит множество белков используемых любой живой клеткой.

Живая клетка — высокоорганизованная автономная система, состоящая из дискретных взаимодействующих компонентов, подобных молекулярной машине. Для изучения этих взаимодействий на системном уровне используются различные формальные методы описания биологических процессов, параллельно протекающих в клетках. Один из подходов изложен в работе Л. Карделли [4], где дана классификация взаимодействующих компонентов и выделено три основных их вида: генные машины, отвечающие за регуляцию клеточных реакций; белковые машины, отвечающие за такие процессы, как обмен веществ, распространение и обработка сигналов; мембранные машины, отвечающие за локализацию, хранение и транспорт органических веществ.

ЖИВАЯ КЛЕТКА КАК КОМПЬЮТЕР ОБЩЕГО НАЗНАЧЕНИЯ

Рассмотрим строение клеток наиболее простых организмов прокариотов и выделим структуры, необходимые для поддержания процессов жизнедеятельности. Прокариоты, к которым относятся и бактерии, были первыми живыми организмами на Земле (появились приблизительно 4 млрд лет тому назад и оставались единственной формой жизни на протяжении последующих 2 млрд лет).

Прокариотические клетки имеют линейные размеры от 1 до 15 мкм. Внутриклеточное пространство клетки отделено от окружающей среды клеточной мембраной, состоящей из белков и особых органических соединений — липидов. Мембрана имеет избирательную проницаемость и важна в осуществлении обмена веществ с окружающей средой. Внутриклеточные процессы протекают в ограниченной мембраной среде цитоплазме, где в водном растворе имеются молекулы, участвующие во внутриклеточных реакциях. В цитоплазме прокариотической клетки находится молекула ДНК, хранящая генетическую информацию в виде инструкций по сборке белков. Молекула ДНК постоянно взаимодействует со специальными белками, формируя хромосому. Кроме этого, в цитоплазме постоянно находятся белки, отвечающие за процесс экспрессии генов, в результате

которого новые белки собираются из флуктуирующих в цитоплазме аминокислот согласно хранящимся в ДНК инструкциям.

Таким образом, для поддержания жизни необходимо наличие мембраны, цитоплазмы, механизма экспрессии генов и хромосомы. Устройство живой клетки представлено в виде схемы (рис. 2), напоминающей архитектуру фон-неймановского компьютера общего назначения.

Хромосому можно сравнить с ПЗУ или с жестким диском современного компьютера, на котором хранятся программы. Аналогично ПЗУ компьютера хромосома способна хранить информацию независимо от процессов клеточного метаболизма, например, она может покидать пределы клетки в виде вирусной частицы, не имеющей собственного метаболизма. После проникновения вирусной частицы в новую клетку хранящаяся в ней генетическая информация успешно реализуется с помощью механизма экспрессии генов зараженной клетки. ДНК отличается высокой надежностью хранения информации, поэтому манипуляции с этой молекулой требуют большой точности и протекают относительно медленно.

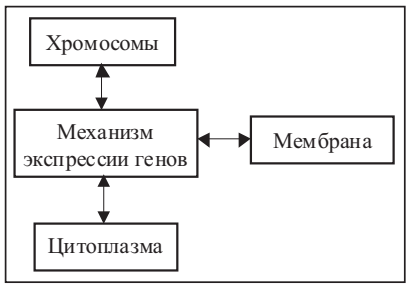

Рис. 2. Схема устройства живой клетки

Цитоплазму клетки можно сравнить с ОЗУ или оперативной памятью современного компьютера. Находящиеся в цитоплазме молекулы соответствуют данным в ячейках оперативной памяти, которые изменяются в результате работы программы. Целостность информации, хранящейся в структуре молекул цитоплазмы, зависит от процессов клеточного метаболизма, поддерживающих специальные условия в цитоплазме.

Механизм экспрессии генов соответствует процессору, который выполняет хранящиеся на жестком диске компьютерные программы. В живой клетке программы работают за счет сборки из флуктуирующих в цитоплазме аминокислот функциональных белков согласно инструкциям, хранящимся в генах хромосомы. Этот процесс называется экспрессией генов и осуществляется сложными белковыми комплексами, инструкции по сборке которых также находятся в ДНК.

В мембране, кроме липидов, локализованы белковые комплексы, образующие мембранные каналы и осуществляющие избирательный транспорт молекул сквозь мембрану. Последняя соответствует устройствам ввода/вывода в архитектуре компьютера общего назначения.

БЕЛКИ — ИСПОЛНЯЕМЫЕ КЛЕТОЧНЫЕ ПРОГРАММЫ

Выполнение компьютерных программ, как правило, связано с последовательным изменением состояний ячеек оперативной памяти. Если сравнивать клетку с молекулярным компьютером, а процессы клеточного метаболизма — с вычислениями, то белки, изменяющие молекулярный состав цитоплазмы, являются исполняемыми программами в живой клетке. Белки флуктуируют в цитоплазме, сканируя окружающие их молекулы, и, оказавшись в благоприятных условиях, вступают во взаимодействия.

Компьютерные программы на этапе исполнения имеют вид машинного кода с императивной структурой, состоящего из последовательности процессорных инструкций. Это обусловлено устройством процессора, который поочередно выполняет элементарные инструкции из конечного множества.

Аминокислотную последовательность белка можно сравнить с машинным кодом программы, а сами аминокислоты — с элементарными процессорными инструкциями. В отличие от исходного кода компьютерных программ аминокислотная последовательность белка неким образом определяет последовательность взаимодействий, в которые вступает белок, блуждая по внутриклеточному про-

странству. Аминокислотные последовательности белков скорее напоминают декларативный стиль программирования: замена отдельных аминокислот влияет на работу всего белка как целого. В то время, когда одни аминокислотные замены никак не влияют на белковую функцию, другие приводят к полной ее потере. Как правило, в результате аминокислотных замен происходит изменение вероятности выполнения белком своей функции.

Активность таких крупных молекул, как аминокислоты, описана в [5, 6] с помощью случайных полей с локальным взаимодействием, для которых введены операции композиции, соответствующие химическим связям. В результате таких операций порождаются сложные поля с новыми свойствами. Подобным образом белки строятся из аминокислот и имеют сложные функции, не присущие исходным аминокислотам по отдельности.

Жизненный цикл компьютерной программы можно представить тремя этапами:

• запуск — процессорные инструкции и исходные данные программы копируются из ПЗУ, где программы хранятся в неактивном виде, в ОЗУ, где происходит выполнение программ;

• выполнение — процессор осуществляет серию последовательных изменений данных в оперативной памяти в соответствии с инструкциями, которые вместе с данными хранятся в оперативной памяти;

• завершение — выделенная для программы оперативная память освобождается, а связанные с ней данные и процессорные инструкции, находящиеся в ОЗУ, теряются.

В жизненном цикле белка можно выделить аналогичные этапы:

• запуск — процесс экспрессии гена, при этом клеточная программа копируется из ДНК, где хранится в неактивном состоянии в виде нуклеотидных цепей, в цитоплазму, где принимает вид функционального белка;

• выполнение — функциональный белок реализует свою функцию в клетке, изменяя молекулярный состав цитоплазмы;

• завершение — белки распадаются на отдельные аминокислоты, которые используются впоследствии для сборки новых белков.

Белковая молекула, как и любая термодинамическая система, со временем теряет свою структуру, а вместе с ней и функцию в клетке. Распад белков сопровождается образованием коротких аминокислотных цепей, которые могут формировать устойчивые клубки с непредсказуемыми функциями. В связи с этим в клетке существует механизм выявления и утилизации изношенных или поврежденных белков.

ЯЗЫК ПРОГРАММИРОВАНИЯ

Механизм получения новых белков в процессе эволюции неясен. Основная догма молекулярной биологии гласит, что не существует клеточного механизма, позволяющего распутать белковый клубок и построить нуклеотидную последовательность, отвечающую его аминокислотной последовательности. Поэтому единственным способом получения новых белков является построение соответствующих белок-кодирующих генов путем манипуляций с их генетическим материалом.

Задача получения нового гена с полезной функцией размером 200 аминокислот кажется NP-сложной с точки зрения клетки. Существует относительно простой способ проверить, обладает ли ген необходимой функцией: для этого достаточно синтезировать его с помощью клеточного механизма экспрессии генов. Но чтобы отыскать таким образом нужный ген, потребовалось бы проверить порядка 4^{600} нуклеотидных последовательностей. При этом в геномах простейших бактерий содержится порядка 1000 разных генов.

Аналогично можно рассуждать о сложности написания новых компьютерных программ. В настоящее время существует огромное разнообразие компьютерных программ и скорость их разработки постоянно растет, что достигается за счет использования специфического представления программ, которое называется исходным кодом. На этапе выполнения программы удобен машинный код, а на эта-

пе разработки и поддержки программы целесообразно представление в виде исходного кода. Это представление предполагает наличие некоторого языка программирования, грамматика которого позволяет исключать из рассмотрения множество некорректных программ, а также определять преобразования — из существующих программ получать новые работающие программы. Кроме этого, языки программирования предлагают различные механизмы повторного применения исходного кода, среди них: композиция, наследование, использование процедур и функций. Появление новых компьютерных программ, как правило, связано с глобальными манипуляциями с целыми блоками исходного кода, а не с одиночными заменами символов.

Подобным образом образование новых генов связано с крупномасштабными геномными перестройками, при которых генетический материал может приноситься вирусами, копироваться или перемещается из других участков хромосомы. Выполнение таких перестроек возможно за счет хранения генетического материала в виде молекулы ДНК, чья структура и химические свойства больше подходят для осуществления подобных манипуляций, чем аминокислотные последовательности белков.

Экспериментальные данные подтверждают существование в клетке специальных белковых комплексов, в чьи функции входит манипуляция с генетическим материалом для получения новых генов. Примером такого механизма является иммунная система млекопитающих [7], производящая белки-антитела, которые специфически связываются с чужеродными телами и участвуют в их нейтрализации.

Иммунная система производит специфические антитела для огромного множества различных соединений, включая искусственно синтезированные, которые ранее не встречались на Земле. Эти факты исключают возможность хранения всех вырабатываемых иммунной системой антител в виде отдельных генов и свидетельствуют о том, что гены антител создаются не в процессе эволюции, а в режиме реального времени при появлении нового типа антигенов. Антитела производятся в специальных клетках иммунной системы, геномы которых целенаправленно подвергаются широкомасштабным мутациям. В результате этих манипуляций гены антител собираются из более коротких последовательностей — заготовок.

Статистические свойства геномов различных живых организмов исследованы в [8], где нуклеотидные последовательности ДНК описаны с помощью моделей цепей Маркова и приведены аргументы в пользу эффективности применения этих моделей для описания объектов с зависимыми признаками при решении задач распознавания, а также сделан вывод о том, что новые гены получаются в результате работы индуктивных процедур, которые используют имеющийся генетический материал в качестве обучающих выборок для эффективного построения новых генов.

ПАРАЛЛЕЛЬНЫЕ И РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ

Необходимость выполнения одновременно нескольких программ и решения сразу нескольких задач возникла и перед создателями вычислительной техники. Для того чтобы добиться эффекта параллельности на одном процессоре, была предложена модель потоков. Согласно этой модели исполняемые программы ассоциировались с так называемыми потоками выполнения, которые выстраивались в циклическую очередь, где каждому из них периодически выдавалась квота процессорного времени для выполнения своей задачи. Такой подход позволял имитировать параллельность, но имел несколько существенных ограничений, возникающих в результате конкуренции параллельных потоков за доступ к оперативной памяти, или за процессорное время.

Следует учесть проблему конкуренции за использование оперативной памяти: огромное количество накопленных программ, которые создавались для работы в однопоточной среде, могли приводить к коллизиям и непредсказуемым ре-

зультатам при параллельном выполнении. Для решения этой проблемы предлагалась модель процессов, основанная на логическом разделении оперативной памяти. Процесс представляет собой совокупность потоков, а также область оперативной памяти, доступную для потоков данного процесса и не доступную для потоков других процессов. Программа ассоциировалась с отдельным процессом, а их создание и логическое разделение оперативной памяти между ними выполняла специальная программная прослойка — операционная система. Модель процессов позволила повторно использовать в параллельной среде программы, разработанные для выполнения в однопоточном режиме.

В условиях параллельности время выполнения потока становится практически непредсказуемым, поскольку оно зависит не только от самого потока, но и от параллельно выполняемых потоков, которые могут запускаться и завершаться динамически. Проблема конкуренции за использование процессорного времени не была тупиковой, пока выполнялся закон Мура, гарантировавший экспоненциальный рост производительности процессоров. В случаях, когда мощности отдельного вычислительного устройства становилось недостаточно, несколько вычислительных устройств объединялись для совместного решения сверхсложных задач. При этом развивались различные схемы организации распределенных вычислений в условиях реальной параллельности, такие как модель акторов или сервис-ориентированная архитектура. Интерес к подобным схемам организации распределенных вычислений повысился, когда закон Мура перестал выполняться вследствие достижения в 2008 г. технологических пределов производительности процессоров. После этого вычислительная мощность компьютеров возрастала за счет увеличения количества процессоров.

Молекулярные вычисления в клетке также имеют параллельный характер, поскольку в цитоплазме одновременно находятся множество функциональных белков, отвечающих за различные функции. Параллельное функционирование белков в клетке можно сравнить с выполнением компьютерных потоков. По мере усложнения живых организмов, становились сложнее процессы жизнедеятельности, протекающие в их клетках. Для координации этих процессов требовалось одновременное присутствие в цитоплазме различных функциональных белков. Это приводило к конкуренции внутриклеточных процессов за доступ к молекулам цитоплазмы и к механизму экспрессии генов.

В процессе эволюции прокариотических клеток появилась проблема, аналогичная конкуренции параллельных потоков за использование оперативной памяти. Функциональные белки, одновременно находящихся в цитоплазме, могут вступать во взаимодействия с молекулами, задействованными в других внутриклеточных процессах, в результате чего естественное течение этих процессов нарушается. Эту проблему нельзя было решить в условиях простой структуры прокариотических клеток до тех пор, пока не появился на Земле новый тип живых организмов — эукариоты (примерно 2 млрд лет тому назад, к ним относятся все высшие организмы, включая человека).

Хотя изначально эукариоты были одноклеточными организмами, строение их клеток имело ряд революционных нововведений, например компартментизацию. Цитоплазма эукариотической клетки разбита внутренними мембранами на множество компартментов, в которых локализованы специфические химические реакции. Протекающие в различныхных компартментах процессы взаимно не связаны, что позволило изолировать их несовместимость в различных компартментах.

Разделение цитоплазмы на компартменты подобно логическому разделению оперативной памяти в модели процессов. Клеточный компартмент, ограничивающий некоторую область цитоплазмы и содержащий функциональные белки, можно сравнить с процессом, состоящим из выделенной области оперативной памяти и содержащим работающие с ней потоки.

Для выполнения некоторого внутриклеточного процесса необходимо поддерживать определенный уровень концентрации соответствующих белков в ци-

топлазме. Поскольку внутриклеточные процессы реализуются за счет активности аминокислот, из которых состоят белки, и их количество в клетке ограничено, интенсивность любого внутриклеточного процесса становится зависимой от наличия параллельных процессов. Параллельные клеточные процессы конкурируют за доступ к механизму экспрессии генов, который распределяет свободные аминокислоты между белками, связанными с различными процессами.

Пока размеры клеток были относительно невелики, количество параллельных процессов легко наращивалось за счет увеличения объема клетки. Поскольку объем клетки определяется кубически от линейных размеров, а площадь ее мембраны — квадратом, то при достижении некоторого критического размера объем клетки увеличивается настолько, что площади цитоплазмы и ее пропускной способности становится недостаточно для выполнения обмена веществ с внешней средой.

Решение этой проблемы было связано с возникновением многоклеточных организмов примерно 1 млрд лет тому назад. Процессы жизнедеятельности в многоклеточном организме распределены среди различных групп клеток, формирующих органы. Клетки многоклеточного организма содержат копию генома и имеют собственный механизм экспрессии генов. Несмотря на то, что в клетках многоклеточного организма содержится одинаковый геном, кодирующий все белки, работающие во всех органах организма, клетки разных органов специализируются на выполнении определенных процессов. Дифференциация клеток происходит за счет включения специальных генов, связанных с работой конкретного органа, и выключения остальных.

ЗАКЛЮЧЕНИЕ

В настоящей статье предпринята попытка связать протекающие в живой клетке процессы жизнедеятельности с исполняемыми компьютерными программами. Для этого клеточные органеллы проецируются на элементы архитектуры фон-неймановского компьютера общего назначения. При этом активные белки сравниваются с исполняемыми компьютерными программами, гены, кодирующие эти белки, отождествляются с исходным кодом компьютерных программ, а цитоплазма клетки — с оперативным запоминающим устройством, в котором выполняются компьютерные программы. Установленные аналоги позволяют проследить за эволюционными изменениями в живой клетке, повлекшими более эффективную организацию процессов клеточной жизнедеятельности. Рассмотрена возможность существования языка программирования, используемого для описания генетических программ.

СПИСОК ЛИТЕРАТУРЫ

1. Turing A.M. On computable numbers, with an application to the Entscheidungsproblem // Proc. of the London Math. Soc. — 1937. — Ser. 2, **42**. — P. 230–265.
2. Schrödinger E. What is life?: The physical aspect of the living cell. — Cambridge: The Univ. press, 1944. — 91 p.
3. Jeffress L.A. Cerebral mechanisms in behavior. The hixon symposium. — New York; London: Hafner Publ. Co., 1967. — 330 p.
4. Cardelli L. Abstract machines of systems biology // Transact. Comput. Syst. Biology III, Lect. Notes in Comput. Sci. — 2005. — **3737**. — P. 145–168.
5. Белецкий Б.А., Гупал А.М. Моделирование внутриклеточных процессов с помощью активных частиц // Кибернетика и системный анализ. — 2012. — № 4. — С. 65–72
6. Белецкий Б.А. Моделирование распределенных внутриклеточных процессов при помощи активных заряженных частиц // Проблемы управления и информатики. — 2012. — № 3. — С. 84–96.
7. Стил Э., Линдли Р., Бландэн Р. Что, если Ламарк прав? Иммуногенетика и эволюция. — М.: Мир, 2002. — 237 с.
8. Гупал А.М., Сергиенко И.В. Оптимальные процедуры распознавания. — Киев: Наук. думка, 2008. — 232 с.

Поступила 29.01.2013