

Компьютерное моделирование процесса возникновения генетического кода

В.А. Дементьев

Институт геохимии и аналитической химии РАН

dementiev@geokhi.ru

Выполнены компьютерные эксперименты, предназначенные для логической проверки внутренней согласованности всех химических и биологических представлений, заложенных в предложенную Э.М. Галимовым теорию возникновения генетического кода. Создана компьютерная модель химического мира, в который включены только аминокислоты, нуклеиновые основания и АТФ. Наложены известные ограничения на реакционные способности этих молекул. Показано, что в условиях хаотического теплового движения модель самопроизвольно устремляется к упорядочению. Последнее проявляется в накоплении полипептидов и полинуклеотидов, представленных символически в виде линейных текстов. С течением времени длины текстов растут, и появляются повторяющиеся тексты, несмотря на постоянное разрушающее действие теплового движения. Такое явление можно считать моделью процесса возникновения предбиологического генетического кода.

1. Постановка проблемы

Раскрытие природы генетического кода – фундаментальная проблема естествознания. Построение предлагаемой компьютерной модели основано на принципах, изложенных в монографии (*Галимов*). Они состоят в следующем:

Жизнь с момента ее зарождения представляет собой эволюционирующую систему ограничений, налагаемых на свободу взаимодействий в системе. Совокупность ограничений, характеризующих систему, определяет меру ее упорядочения.

Ограничения проявляются в селективности, установлении определенных или предпочтительных путей взаимодействия, т.е. в тех свойствах, которыми наделены катализаторы, в том числе биологические – ферменты.

В мире органических соединений нет более эффективного инструмента упорядочения, чем пептиды. В силу разнообразия возможных конфигураций и образования вторичных и третичных структур цепочки аминокислот могут служить катализаторами, обладающими исключительной селективностью. Высокоупорядоченное состояние современных организмов обусловлено тем, что практически все химические реакции в организме управляются белковыми ферментами.

Однако упорядочение, основанное на эволюции только пептидов, не имеет перспективы, поскольку пептиды не способны к самокопированию. С другой стороны нуклеотидные цепочки, например РНК, могут реплицироваться, но обладают слабыми по сравнению с пептидами селективными (каталитическими) свойствами. Эволюционируя по пути возрастающего упорядочения, жизнь нашла путь к разрешению этой коллизии.

Копирование пептидов происходит через посредство нуклеотидов. Для этого каждая аминокислота должна найти свое отражение в виде комбинации нуклеотидов. Тогда последовательность аминокислот может воспроизводиться, используя репродуктивные свойства последовательности нуклеотидов.

Настоящая работа представляет собой попытку компьютерной имитации этого процесса. Задача состояла в том, чтобы, ограничив реакционные способности частиц небольшим числом простых правил и предоставив им возможность взаимодействовать, уяснить, будет ли при этом происходить эволюционирующее упорядочение. Достаточными признаками последнего предложено считать прогрессирующее удлинение полимерных цепей и

увеличение числа копий одноименных полимеров (содержащих одинаковые линейные «тексты», составленные из аминокислот, либо из кодонов). При этом не вводятся дарвиновские правила естественного отбора.

Для того, чтобы этот механизм начал действовать, прежде всего, необходим посредник, который мог бы ставить в соответствие аминокислоту и группу нуклеотидов (фактически три нуклеотида – кодон). Такой посредник в современных организмах существует – это транспортная рибонуклеиновая кислота (t-РНК). Сложившийся в ходе эволюции механизм трансляции в организмах весьма сложен. Он осуществляется на рибосоме, образованной несколькими РНК и десятками белков. Синтез белка происходит на матричной РНК (m-РНК), к которой t-РНК прикрепляется участком, несущим три нуклеотида (антикодон), комплиментарных кодону матричной РНК. Распознавание аминокислоты осуществляется при помощи фермента *аминоацил-t-РНК-синтетазы*. Образование комплекса синтетазы с аминокислотой активируется АТФ. В добиологическом мире, на ранней Земле, такого сложного механизма не было. Но в примитивной форме соответствующие функции могли проявиться очень рано и их можно имитировать на компьютере, вводя в модель химического мира частицы типа полипептидов и полинуклеотидов определенной структуры и приписывая им необходимые свойства.

Наиболее простые, легко синтезируемые абиогенно, аминокислоты, которые, очевидно, были наиболее распространены в предбиологическом мире, кодируются, как известно, сочетанием двух нуклеотидов. Третий может быть любым из четырех типов нуклеотидов, составляющих РНК: глицин – GG[A/C/G/U], аланин – GC[A/C/G/U], валин – GU[A/C/G/U], серин – UC[A/C/G/U], лейцин CU[A/C/G/U], треонин – AC[A/C/G/U]. В квадратных скобках указан третий нуклеотид в соответствующем кодоне. Исключение составляют две относительно простые и распространенные аминокислоты, которые, тем не менее, кодируются тремя значимыми нуклеотидами: аспарагиновая кислота – GA[U/C] и глутаминовая кислота – GA[A/G]. Однако, возможно, в примитивном мире эти аминокислоты не различались (структурно они близки) и кодировались одинаково: GA[A/C/G/U].

Более сложные аминокислоты кодируются триплетом нуклеотидов. Большинство из них также имеет вырождение, а именно, два сходных триплета, отличающиеся только третьим нуклеотидом в кодоне, кодируют одну и ту же аминокислоту. К ним относятся: гистидин CA[U/C], глутамин – CA[A/G], аспарагин AA[U/C], лизин – AA[A/G], фенилаланин – UU[U/C], тирозин – UA[U/C], цистеин – UG[U/C]. Лишь метеонин и триптофан кодируются триплетами однозначно: AUG и UGG – соответственно.

Подобная закономерность наводит на мысль, что первоначально кодирование осуществлялось дублетом, может быть даже синглетом нуклеотидов. Однако, дублетное соответствие нуклеотидов и аминокислот не могло эволюционно преобразоваться в триплетное. Если бы даже дублетный код возник, он не эволюционировал бы в триплетный, а скорее, был бы вытеснен последним, допускающим большую возможность упорядочения. Поэтому мы полагаем, что в получившем развитие варианте жизни изначально возникли, в силу, возможно, случайных обстоятельств, условия, ведущие к триплетному кодированию. В то же время распознавание α_i -синтетазой полинуклеотида, несущего специфический кодон, могло быть достаточным, если распознавались лишь только два первых нуклеотида в кодоне.

Компьютерное исследование всех указанных процессов было осуществлено в рамках двух моделей. Модель I была ориентирована на проведение поисковых исследований и позволяла наглядно представлять себе происходящие процессы на экране монитора персонального компьютера. В связи с таким требованием модель была ограничена в объеме представляемых данных. Модель II была ориентирована на проведение более масштабных исследований с накоплением большой статистики. Это потребовало написания нового варианта программы, чтобы вывести расчеты на суперкомпьютер МВС-

1000. В работе приводятся результаты двух серий компьютерных экспериментов, выполненных с двумя вариантами моделей.

2. Описание модели I

Указанные выше принципы можно формализовать в модели следующим образом:

- Пусть имеется множество аминокислот (α_i), представленных 20 аминокислотами, встречающимися в белковых ферментах живых организмов.
- Пусть имеется множество нуклеотидов A_i представляющих 4 нуклеотида, встречающиеся в РНК (A, U, G, C). Для удобства последующих операций, положим, что нуклеотиды изначально представлены в виде триплетов, так что A_i – одна из 64 возможных комбинаций из четырех по три.
- Пусть имеется возможность соединения аминокислот между собой и нуклеотидов между собой. Считаем, что реакция присоединения возможна только с участием АТФ (*Galimov*). Поэтому ее, как участника, в модель не вводим, имея в виду, что любая реакция типа $\alpha + \beta = \alpha\beta$ успешна только, если $\alpha + \text{АТФ} = \alpha\text{АТФ}$; $\beta + \alpha\text{АТФ} = \alpha\beta + \text{АДФ} + P_i$.
- Имеются частицы Not, моделирующие присущий природе процесс дезинтеграции и упрощения в результате столкновения частиц, участвующих в тепловом движении.
- Условие стационарности предполагает, что компоненты, израсходованные в ходе образования нового продукта, восполняются.

- Упорядочение выражается в росте размера полимеров $\prod_i^n \alpha_i$ и в их возрастающей способности влиять на ход процессов в системе по мере роста размера.

- Полимеры типа $\prod_i^n \alpha_i$ не способны к самовоспроизведению.

- Полимеры типа $\prod_i^k A_i$ способны к самовоспроизведению, но не могут влиять на какие-либо процессы в системе.

Поскольку модель имеет дело с символами и с линейными текстами из этих символов, а не с реальными химическими свойствами соединений, моделирование химических особенностей соединений будем производить условно, назначая число символов в цепи или определенное сочетание α_i в $\prod_i^n \alpha_i$ и A_i в $\prod_i^k A_i$.

Введем следующие правила:

1) Полипептид $\prod_i^n \alpha_i$ начиная с $n \geq 4$ (условно) способен катализировать соединение двух

полинуклеотидных фрагментов. Возникновение комплекса $\prod_{i=1}^n \alpha_i$ и $\prod_{i=1}^k A_i$ разрешает

присоединение к $\prod_{i=1}^k A_i$ другого полинуклеотида, содержащего текст nA_i . Фактически

возникает расширение полинуклеотида до $\prod_{i=1}^{k+n} A_i$. Так формируется пул некодированных

РНК.

2) Полипептид $\prod_i^n \alpha_i$ начиная с $n \geq 4$ (условно), если он содержит две дважды

повторяющиеся аминокислоты подряд (условно), т.е. $\alpha\beta\gamma\gamma$ играет роль полимеразы, т.е.

образование комплекса $\prod_{i=1}^n A_i$ с таким полипептидом разрешает возникновение копии данного полинуклеотида.

- 3) Полипептид $\prod_i^n \alpha_i$, начиная с $n \geq 5$, если он содержит на обоих концах аминокислоту α_1 , образует комплекс с ней, т.е. играет роль аминоацил-t-РНК-синтетазы.
- 4) Если в цепи некоторого полинуклеотида возникли два триплета, являющиеся кодоном и антикодоном, то этот полинуклеотид может быть распознан α_1 -синтеазой, которая образует с ним комплекс, несущий α_1 -аминокислоту. При этом на ранних стадиях эволюции двух первых нуклеотидов в кодоне было достаточно для распознавания. С другой стороны, антикодон этого полинуклеотида может образовать связь с комплиментарным ему триплетом свободной РНК. Полинуклеотид, приобретший эти свойства, выполняет функцию t-РНК.
- 5) Полинуклеотид, к которому прикрепилась t-РНК служит матрицей (m-РНК) для построения полипептида так, что, если к соседнему кодону присоединится новая t-РНК, то предшествующая высвобождается, возникает пептидная связь между соседними α_i , и так до завершения всего построения. Если соответствующая $R\alpha_i$ еще отсутствует в пуле, то полинуклеотид функцию матрицы выполнить не может и уничтожается при столкновении с частицей Not. Если все R_{α_i} необходимые для построения полипептида на данной матрице, имеются в пуле, то для экономии времени не будем ожидать последовательного построения, а получим соответствующий полипептид сразу.
- 6) Если полипептид выстроен на матрице полинуклеотида по правилу (5), то полинуклеотид и полипептид разделяются. Возникший полипептид может участвовать в процессах, включенных в правила (1), (2), (3).
- 7) Полученный по правилу (5) полипептид, в отличие от возникших путем случайного соединения α_i , является кодированным, так как ему соответствует его полинуклеотидный «имидж». Придадим этому полипептиду свойство быть эффективной *полимеразой* по отношению к соответствующему полинуклеотиду, например, пусть условно одна встреча кодированного полипептида с кодирующим его полинуклеотидом приводит к появлению n -числа копий последнего.
- Таким образом, правила (1), (2), (3), (7) реализуют свойства полипептидов производить упорядочивание, правила (2) и (7) реализует способность полинуклеотидов к самокопированию, а правила (3), (4) и (5) представляют механизм, посредством которого эти свойства объединяются в единый процесс эволюционирующего упорядочения. Следует подчеркнуть, что предлагаемая модель принципиально не включает каких-либо критериев отбора. Отсутствие отбора – подчеркивается постоянным введением в систему исходных частиц по мере их вовлечения в усложняющиеся структуры. Таким образом, предлагаемая модель имитирует поведение открытой неравновесной системы, взаимодействующей с окружающей средой, откуда в моделируемую систему поступают необходимые строительные материалы.

3. Описание программы I

Программа, названная gem, построена по принципу клеточного автомата. Имитация химического мира выполняется на квадратном поле размером 200x200 клеток. Клетки содержат частицы тех типов, которые были описаны выше. Пустые клетки имитируют фон. Программа случайным образом выбирает экземпляры A_i и α_i из рабочих словарей и частицы Not и распределяет их по полю.

Рабочие словари создаются программой при ее инициализации. Программа имитирует самопроизвольное возникновение частиц, которые могли бы распознавать структурное соответствие при встречах. Имеется в виду, что молекулы распознают друг друга в химической реакции только при определенном геометрическом соответствии их эквипотенциальных поверхностей. Такие поверхности в программе кодируются изображающими шестнадцатиричными числами. Изображающее число частицы X

обозначено как $\#X$. Шестнадцатиричная запись числа отмечена буквой H . В программе изначально задана информация о том, что в природе нашлись четыре попарно комплементарных нуклеотида: A, C, G, U. Этим частицам приписаны четырехзначные изображающие числа: $\#A = H5555$, $\#C = H3333$, $\#G = H0F0F$, $\#U = H00ff$, которые введены в программу в качестве инициализирующих данных.

Программа строит словарь кодонов в виде всех комбинаций из четырех нуклеотидов по три. Каждый кодон в словаре определяется своим порядковым номером и двенадцатизначным изображающим числом, в котором объединены все разряды изображающих чисел всходящих в кодон нуклеотидов. Например, один из кодонов лейцина – CUG, изображается числом $\#CUG = H333300FF0F0F$. Программа строит словарь из всех 64 возможных кодонов, которые в качестве A_i вводятся в рабочий словарь.

Далее программа строит словарь частиц, которые структурно соответствуют кодонам. Для упрощения полагаем, что для распознавания α_i -аминоацил-*t*-РНК-синтетазой соответствующей $t - R_{\alpha_i}$ существенным является вид кодона-антикодона, который последняя несет. В современных системах структурное соответствие может быть более сложным. Каждую частицу, претендующую представлять комплекс α_i -аминоацил-*t*-РНК-синтетазы, изображаем также двенадцатизначным числом и требуем, чтобы структурное соответствие комплекса соответствующему кодону распознавалось с помощью условия: логическое произведение изображающих чисел комплекса и кодона должно быть равно нулю. Это означает, что выступу на поверхности одной молекулы (двоичная единица) может соответствовать только впадина (двоичный ноль) на поверхности другой взаимодействующей молекулы. Совпадение двух впадин не мешает взаимодействию, а лишь немного его ослабляет.

Программа генерирует последовательность случайных изображающих чисел. Каждое новое число испытывается на соответствие какому-либо из кодонов. Отбираются те числа, которые соответствуют одному или нескольким кодонам. Преимущество отдается числам, соответствующим нескольким кодонам. Для упрощения алгоритма и ускорения работы программы мы приписываем найденное изображающее число комплекса и самой аминокислоте. Например, найдется случайное число HCCCCFF008000. Оно даст ноль при логическом умножении на $\#CUG$. То же самое будет получаться и при испытании чисел $\#CUA$, $\#CUC$, $\#CUU$. Так обнаружится вырожденный код комплекса с лейцином. Заодно находится условное изображающее число самого лейцина, его порядковый номер в словаре и ссылки на номера тех кодонов, которые кодируют лейцин.

Эксперименты с данной программой показали, что получение каждого нужного изображающего числа в результате генерации сразу двенадцатизначного шестнадцатиричного числа требует необозримого машинного времени. Если же поручить программе подобрать сначала первую четверку разрядов (соответствие первому нуклеотиду в кодоне), закрепить успех, а затем таким же образом подбирать вторую и, наконец, третью четверку разрядов, то на всю инициализацию уходят секунды.

Нумерация частиц позволяет программе работать с числами, а не с символьными изображениями частиц, участвующих в реакциях. При выдаче окончательных результатов прохождения реакций могут быть восстановлены символьные изображения получающихся полипептидов и полинуклеотидов.

Таблица аминокислот и их кодонов

A = &H5555 ~A = &HAAAA
 C = &H3333 ~C = &HCCCC
 G = &H0F0F ~G = &HFF0F
 U = &H00FF ~U = &HFF00

N	Name	K1	K2	K3	k1	k2	k3
1	Gly	G	G	A C G U	&HFF0F	&HFF0F	&H8000

2	Ala	G	C	A C G U	&HF0F0	&HCCCC	&H8000
3	Val	G	U	A C G U	&HF0F0	&HFF00	&H8000
4	Leu	C	U	A C G U	&HCCCC	&HFF00	&H8000
21	Leu	U	U	A G	&HFF00	&HFF00	&HA0A0
4	Leu	C U	U	A C G U	&HCC00	&HFF00	&H8000
5	Ile	A	U	A C U	&HAAAA	&HFF00	&H8800
6	Phe	U	U	C U	&HFF00	&HFF00	&HCC00
7	Pro	C	C	A C G U	&HCCCC	&HCCCC	&H8000
8	Trp	U	G	G	&HFF00	&HF0F0	&HF0F0
9	Ser	U	C	A C G U	&HFF00	&HCCCC	&H8000
22	Ser	A	G	C U	&HAAAA	&HF0F0	&HCC00
9	Ser	A U	CG	A C G U	&HAA00	&HC0C0	&H8000
10	Thr	A	C	A C G U	&HAAAA	&HCCCC	&H8000
11	Met	A	U	G	&HAAAA	&HFF00	&HF0F0
12	Asn	A	A	C U	&HAAAA	&HAAAA	&HCC00
13	Gln	C	A	A G	&HCCCC	&HAAAA	&HA0A0
14	Cys	U	G	C U	&HFF00	&HF0F0	&HCC00
15	Asp	G	A	C U	&HF0F0	&HAAAA	&HCC00
16	Glu	G	A	A G	&HF0F0	&HAAAA	&HA0A0
17	Tyr	U	A	C U	&HFF00	&HAAAA	&HCC00
18	His	C	A	C U	&HCCCC	&HAAAA	&HCC00
19	Lys	A	A	A G	&HAAAA	&HAAAA	&HA0A0
20	Arg	C	G	A C G U	&HCCCC	&HF0F0	&H8000
23	Arg	A	G	A G	&HAAAA	&HF0F0	&HA0A0
20	Arg	A C	G	A C G U	&H8888	&HF0F0	&H8000
24	Trm	U	G	A	&HFF00	&HF0F0	&HAAAA
25	Trm	U	A	A G	&HFF00	&HAAAA	&HA0A0
21	Trm	U	A G	A G	&HFF00	&HA0A0	&HA0A0

Примечания:

&Nx - восьмеричная запись числа x

&H5 = 0101

&H3 = 0011

&HF = 1111

&HC = 1100

&HA = 1010

&H8 = 1000

&HA = 1010

~&HA = 0101 = &H5

| = логическое OR

~ = логическое NOT

После инициализации работа программы состоит в выполнении шагов, имитирующих течение исторического времени. На каждом шаге программа случайным образом находит в поле две частицы и проверяет, способны ли они взаимодействовать друг с другом. Шаг считается состоявшимся, если нашлась подходящая пара частиц, и реакция проведена. Реакции проводятся по описанным выше правилам с учетом структурных соответствий, найденных при инициализации. Если короткие цепи полипептидов или полинуклеотидов включаются в более сложные комплексы, то программа добавляет в поле случайные аминокислоты или кодоны из словарей, поддерживая начальное количество частиц этих

типов. Так имитируется открытость моделируемой химической системы - ее стационарное состояние.

4. Эксперименты с программой I

В результате компьютерного моделирования были установлены следующие закономерности.

Правило, допускающее удлинение (катализируемое) цепи нуклеотидов, приводит к появлению структур R_{α_i} (рис. 1). Рост длины увеличивает вероятность одновременного появления в составе полинуклеотида комплиментарных триплетов – кодона и антикодона, т.е. появления R_{α_i} в соответствии с правилом (4). Какой конкретно аминокислоте α_i будет соответствовать R_{α_i} в природе определяется согласованием свойств R_{α_i} и *аминоацил-t-RНК- α_i -синтазы*. В программе это соответствие устанавливается на основе рабочего словаря.

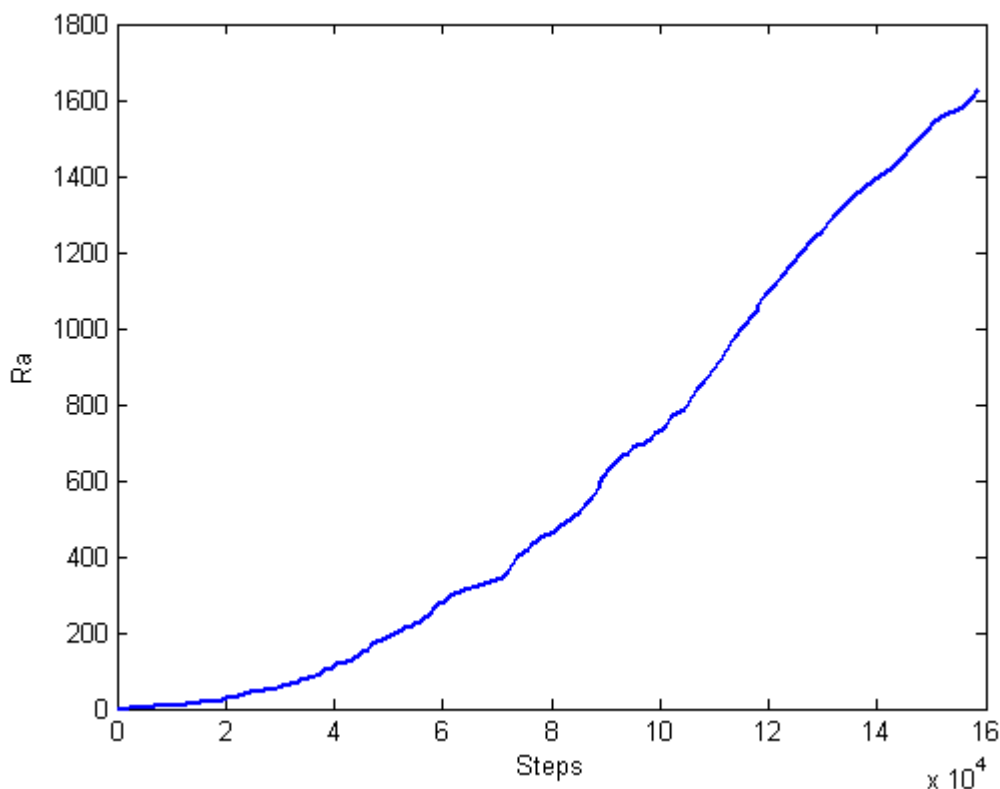


Рис. 1. Динамика накопления комплексов Ra

Отметим, что соответствие аминокислот и триплетов нуклеотидов, отвечающее существующему генетическому коду, не навязано модели, а генерируется программой. Поскольку требования к каталитическим свойствам полипептида, сшивающего полинуклеотидные фрагменты Π_A (правило 2) достаточно мягкие ($n \geq 4$, однократно встречающееся удвоение α_i), в системе формируется значительный пул промежуточных комплексов $\Pi_A \Pi_\alpha \Pi_A$ (рис. 2). Поэтому значительная часть материала оказывается вовлеченной в это промежуточное состояние. Если сделать требования более жесткими, например, установить, что полипептид, катализирующий процесс (сшивания) должен обладать более редким сочетанием или большим числом мономеров α_i , то размер пула уменьшается, но при этом существенно замедлится динамика накопления пула R_{α_i} .

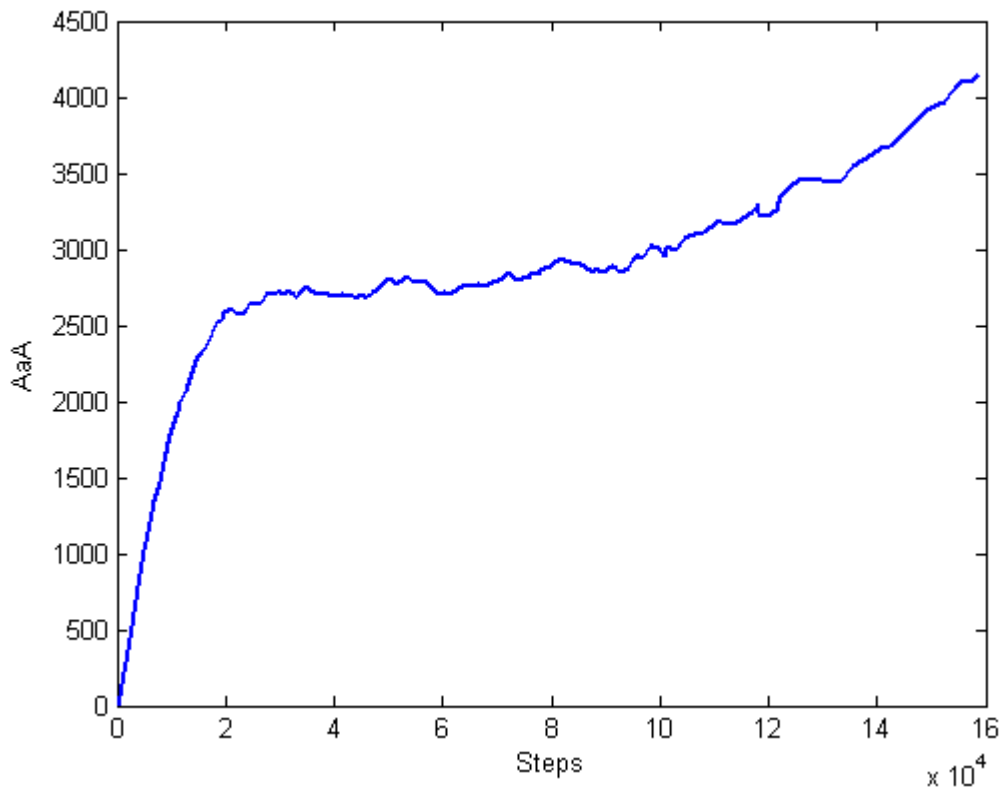


Рис. 2. Динамика накопления комплексов АаА

На приведенных ниже рисунках показаны результаты одного из экспериментов. В начальном состоянии в модели насчитывалось 400 кодонов, 400 аминокислот и 50 горячих частиц. Предварительные эксперименты показали, что такое начальное состояние обеспечивает наилучшую динамику самоорганизации мира. Более холодный мир (число горячих частиц заметно меньше 50) не развивается, поскольку возникающие сложные комплексы цепей не разделяются ударами горячих частиц на индивидуальные длинные цепи. Более горячий мир (число горячих частиц заметно больше 50) развивается очень медленно, поскольку возникающие длинные цепи интенсивно разрушаются ударами горячих частиц. Эти наблюдаемые в компьютерном эксперименте закономерности согласуются с тем, что биологические процессы осуществляются в сравнительно узком интервале температур.

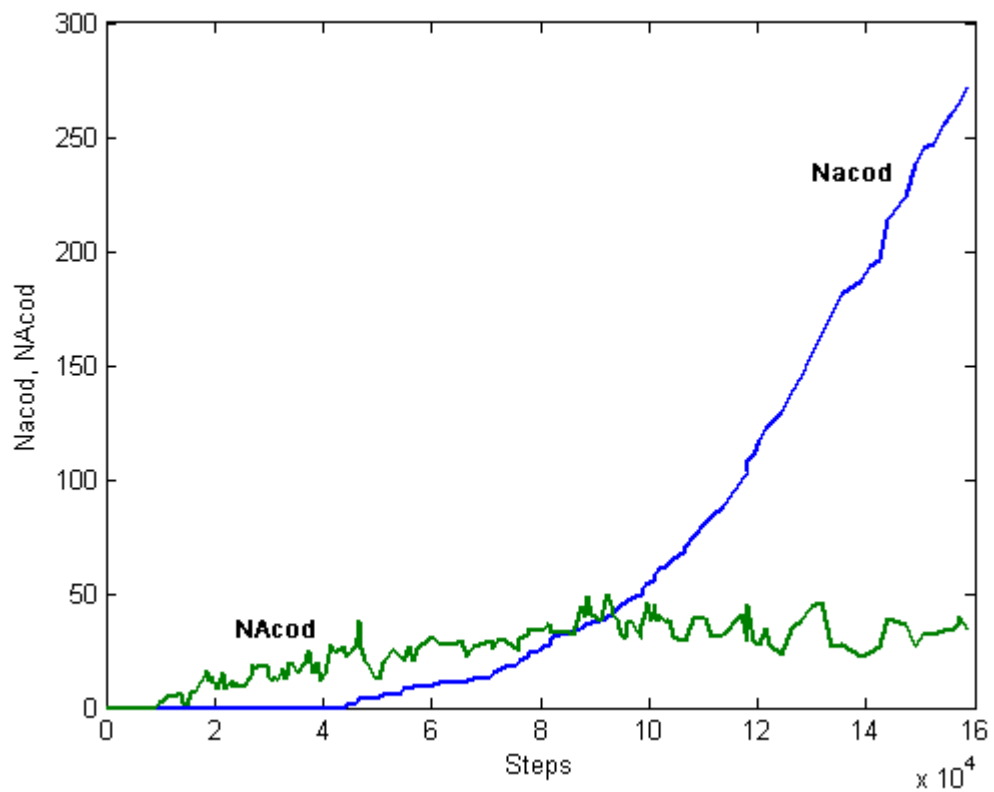


Рис. 3. Динамика накопления кодированных цепей плинуклеотидов Acod и полипептидов acod

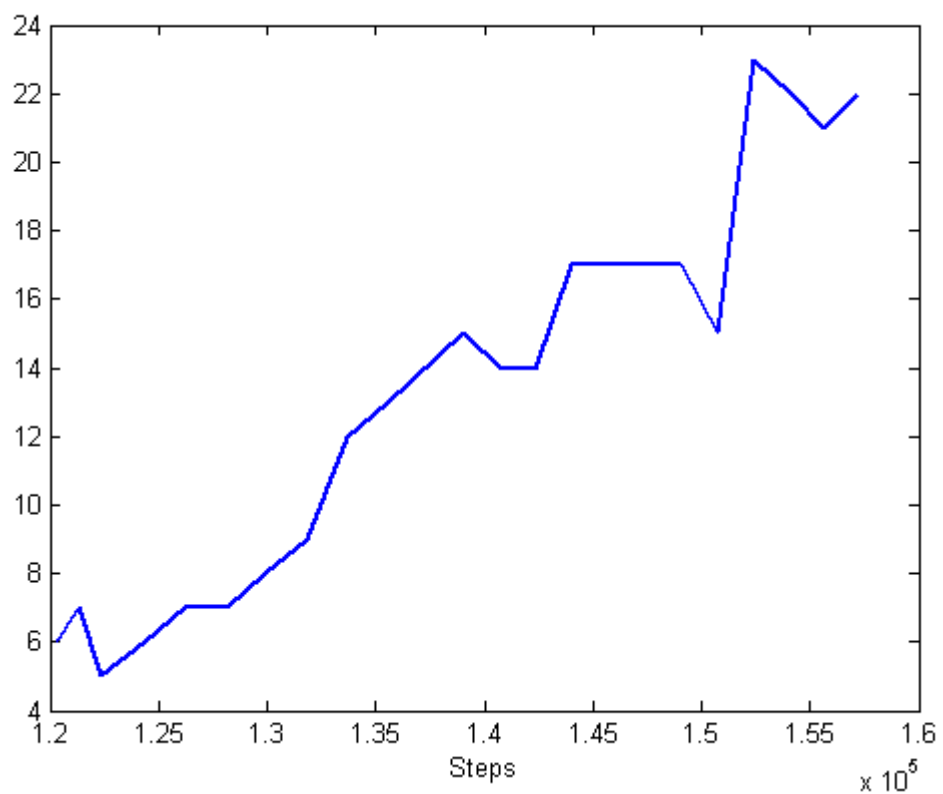


Рис. 4. Динамика накопления совпадающих текстов в кодированных цепях полипептидов

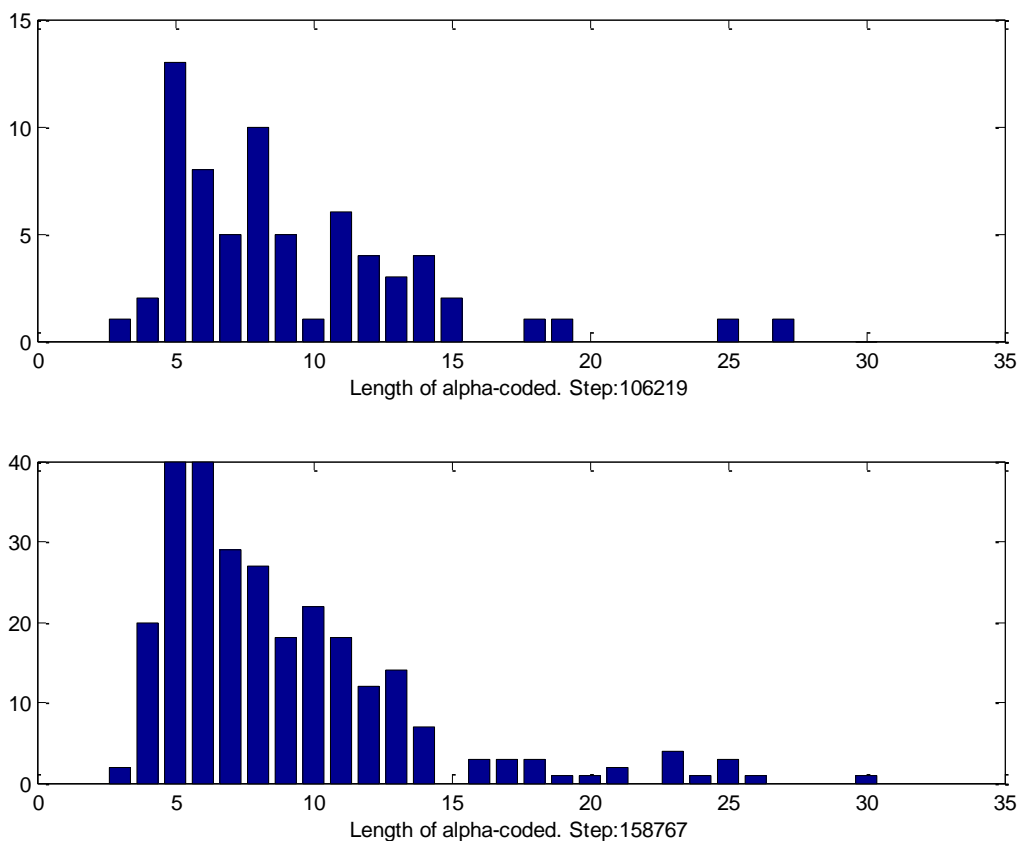


Рис. 5. Гистограмма распределения длин кодированных «полипептидов» с увеличением числа шагов программы а) шаг 106219, б) шаг 158767.

Процесс трансляции последовательности нуклеотидов в полипептидную последовательность, т.е. возникновение кодированных полипептидов, начинается с задержкой, обусловленной действием правила (5). Только после того, как возникло множество R_{α_i} , охватывающее большинство участвующих в процессе видов α_i , процесс накопления кодированных Π_{α_i} начинает стремительно нарастать (рис. 2).

Постепенно нарастает длина и число кодированных полипептидов (рис. 3), т.е. происходит эволюционирующее упорядочение.

Происходит увеличение числа одноименных (т.е. имеющих одну и ту же последовательность аминокислот) полипептидов (рис. 4), т.е. осуществляется опосредованное производство копий полипептидов. Ниже приведена распечатка анализа совпадений текстов в длинных цепях полипептидов. Для первого найденного случая совпадений даны подробные пояснения обозначений, а далее приведены результаты без пояснений.

Шаг 157147, совпадений текстов 22

Текст полипептида с номером $i = 7$ и длиной $l_i = 17$

Совпал с текстом полипептида с номером $j = 15$ и длиной $l_j = 17$

Вот текст полипептида с номером $j = 15$

SerPheLeuLeuProSerGlnAlaArgArgThrProValLeuValGlyIle,

а текст полипептида с номером $i = 7$ не приводится. Он такой же.

$i =$ 8 $l_i =$ 11 $j =$ 113 $l_j =$ 11

SerArgHisHisCysIleSerGluTyrAlaSer
i = 11 li = 5 j = 190 lj = 5
TrpHisProGluTyr
i = 23 li = 7 j = 222 lj = 7
ArgAlaGlyValGlyLeuSer
i = 30 li = 6 j = 367 lj = 6
AsnArgProValSerArg
i = 41 li = 9 j = 239 lj = 9
ValLeuAsnValProIleThrSerMet
i = 65 li = 8 j = 169 lj = 11
GlnThrAlaArgGlyArgLysSer
i = 67 li = 6 j = 72 lj = 7
AsnCysPheGlyLeuGly
i = 76 li = 13 j = 255 lj = 13
SerGlnAlaSerGlySerLysHisLeuAlaThrArgCys
i = 85 li = 7 j = 256 lj = 7
ProArgAlaArgPheThrHis
i = 100 li = 5 j = 195 lj = 7
ArgGlnThrProGln
i = 104 li = 14 j = 302 lj = 14
ThrAlaAsnCysPheProGluValAsnAspSerAlaThrAsp
i = 104 li = 14 j = 306 lj = 14
ThrAlaAsnCysPheProGluValAsnAspSerAlaThrAsp
i = 105 li = 9 j = 191 lj = 10
IleSerGlyMetAspLysThrGlnSer
i = 107 li = 25 j = 288 lj = 25
AsnTyrLeuHisLeuGlyTyrSerLeuHisPheGlnLeuSerLysGlyIlePheArgLeuThrArgValPheLeu
i = 107 li = 25 j = 385 lj = 26
AsnTyrLeuHisLeuGlyTyrSerLeuHisPheGlnLeuSerLysGlyIlePheArgLeuThrArgValPheLeu
i = 210 li = 6 j = 330 lj = 6
LeuPheSerThrIleLeu
i = 234 li = 6 j = 363 lj = 10
ProTyrSerThrGluLys
i = 268 li = 8 j = 381 lj = 8
ValArgAlaGlyTyrThrMetLeu
i = 302 li = 14 j = 306 lj = 14
ThrAlaAsnCysPheProGluValAsnAspSerAlaThrAsp
i = 341 li = 11 j = 372 lj = 11
ArgAsnThrLysIleThrValIleTyrProSer

Шаг 157277, совпадений текстов 18

Это очень близкий шаг. Мало что изменилось. Пропали некоторые самые длинные цепи, но много коротких совпадений сохранилось. Например, первое совпадение в первой и в этой серии. Номер $j = 30$ здесь изменился, но это ничего не значит. Программа перетасовывает имеющиеся частицы случайным образом. Так что это может быть старая частица, или вновь созданная. Определить трудно, не прослеживая результаты каждой реакции.

i = 8 li = 11 j = 30 lj = 11
SerArgHisHisCysIleSerGluTyrAlaSer
i = 10 li = 5 j = 289 lj = 5

TrpHisProGluTyr
 i = 16 li = 5 j = 364 lj = 6
 GlyIleGlySerPro
 i = 55 li = 8 j = 199 lj = 8
 ValArgAlaGlyTyrThrMetLeu
 i = 77 li = 14 j = 156 lj = 14
 ThrAlaAsnCysPheProGluValAsnAspSerAlaThrAsp
 i = 77 li = 14 j = 302 lj = 14
 ThrAlaAsnCysPheProGluValAsnAspSerAlaThrAsp
 i = 90 li = 7 j = 96 lj = 7
 ArgAlaGlyValGlyLeuSer
 i = 92 li = 19 j = 343 lj = 21
 GlnAsnGlnAspThrAlaLeuProSerIleGlnAsnLeuGlyGluSerLeuAlaSer
 i = 94 li = 13 j = 389 lj = 13
 SerGlnAlaSerGlySerLysHisLeuAlaThrArgCys
 i = 123 li = 17 j = 172 lj = 17
 SerPheLeuLeuProSerGlnAlaArgArgThrProValLeuValGlyIle
 i = 156 li = 14 j = 302 lj = 14
 ThrAlaAsnCysPheProGluValAsnAspSerAlaThrAsp
 i = 159 li = 11 j = 298 lj = 11
 ArgAsnThrLysIleThrValIleTyrProSer
 i = 168 li = 9 j = 354 lj = 9
 ValLeuAsnValProIleThrSerMet
 i = 185 li = 7 j = 211 lj = 7
 ProArgAlaArgPheThrHis
 i = 220 li = 7 j = 233 lj = 8
 LeuThrGlyLeuLeuAsnAla
 i = 225 li = 6 j = 274 lj = 6
 LeuPheSerThrIleLeu
 i = 293 li = 25 j = 356 lj = 25
 AsnTyrLeuHisLeuGlyTyrSerLeuHisPheGlnLeuSerLysGlyIlePheArgLeuThrArgValPheLeu
 i = 320 li = 6 j = 378 lj = 6
 AsnArgProValSerArg

Полученные результаты показывают, что представленная модель удовлетворяет исходному замыслу. Сравнительно небольшое число введенных ограничений и соответствий, которые адекватны возможным ограничениям, обусловленным спецификой химических свойств, приводят к фундаментальному результату – увеличению числа копий и длины кодированных полипептидных цепей с одновременным существованием соответствующих кодирующих полинуклеотидных цепей. Введение ограничения на размеры ресурсов исходных частиц, в конечном счете, привело бы к проявлению селективного выживания структур в зависимости от приписываемой им эффективности или устойчивости, т.е. к дарвиновскому отбору. Следовательно, естественный отбор модифицирует систему, но он не является причиной ее эволюционирующего упорядочения.

Из приведенных выше рисунков видно, что в имитационном эксперименте наблюдаются и другие закономерности, согласующиеся с биологической реальностью и с фактами ранней эволюции биосферы. Например, кодированные полипептидные цепи возникают, лишь спустя длинный инкубационный период, после чего начинается бурный рост, как их числа, так и длин. Анализ текстов кодированных полипептидов показывает, что с течением исторического времени всё большее число совпадающих аминокислотных последовательностей начинает встречаться не только в цепях одинаковой длины, но более

длинные цепи начинают включать в себя более короткие, ранее образовавшиеся последовательности.

Технически возможно объединить два описанных этапа в один. Тогда нам не пришлось бы готовить заранее и вводить в программу словарь Кодон-Аминокислота. В такой обобщенной программе понадобится ввести только шестнадцатиричную кодировку четырех избранных нуклеотидов и задать аналогичные начальные условия. При добавлении частицы типа аминокислоты программа должна была бы генерировать случайное четырехзначное число до выполнения условия соответствия первому нуклеотиду любого кодона из их полного набора. После аналогичного подбора второй и третьей части изображающего числа программа самостоятельно узнавала бы, какая подходящая аминокислота случайно сгенерирована. Дальше программа работала бы по сценарию второго этапа, и результаты были бы такими же. Мы разделили процесс моделирования на два этапа ради упрощения алгоритмов и сокращения времени работы. В результате эксперимент потребовал скромных вычислительных ресурсов, и был проведен на персональном компьютере в системе Visual Basic 6.

Таким образом, средствами компьютерного моделирования показана достаточность приведенных в монографии (*Галимов*) правил взаимодействия кодонов и аминокислот для самопроизвольного возникновения генетического кода, причем словарь Кодон-Аминокислота можно не навязывать модели, а поручить его генерацию программе. Это соответствует положению, что в природе лишь некоторые аминокислоты самостоятельно нашли соответствие определенным кодонам и по похожему сценарию закрепили свойство так кооперироваться с полинуклеотидами, чтобы передавать эту важную информацию в будущее. Остальные возможные аминокислоты не смогли войти в эти процессы и остались за бортом жизни.

5. Модель II и второй вариант программы моделирования самоорганизации химического мира

В процессе работы возникло впечатление, что можно упростить модель, в частности, уменьшить количество промежуточных комплексов. Тем самым, можно снизить количество вносимой в программу априорной информации о процессах самоорганизации химического мира.

Кроме того, первая модель не давала должной статистики набираемых в процессе работы программы данных. Малая статистика проявляется в больших флуктуациях динамики популяций различных частиц. Поэтому новый вариант модели и программы был сразу ориентирован на Фортран, способный оперировать с большими массивами информации. Было также ясно, что достаточную статистику невозможно получить без выхода модели на суперкомпьютер, а для этого было необходимо реализовать описанные и новые искомые алгоритмы на Фортране. В связи с этим пришлось отказаться от наглядного представления исследуемых процессов на экране монитора, поскольку суперкомпьютер мало приспособлен для выдачи графической информации. Это обстоятельство также привело к упрощению программного представления данных о структуре исследуемого химического мира. В результате удалось упростить те структуры данных, которые были характерными для языка сверхвысокого уровня, каким является Visual Basic 6, и ввести вместо них более простые структуры типа связанных списков, представимые в языке Фортран-90, реализованном на суперкомпьютере МВС-1000.

Модель II и соответствующий ей вариант программы, названный gem2, наследуют многое из модели I, поскольку мы исследуем процессы в рамках одной и той же теории самопроизвольного возникновения предбиологического генетического кода (*Галимов*).

Ниже приведены только характерные отличия новой модели и нового алгоритма.

В мире gem2 имеется только два типа частиц – α и A.

Частицы α это мономеры, олигомеры и полипептидные цепи.

Частицы A это кодоны (считаются мономерами), их олигомеры и полинуклеотидные цепи.

Частицы $A\alpha$ и AA это всевозможные комплексы, а не самостоятельный тип. Являются промежуточными частицами. Через них формируются соответствующие олигомеры и полимеры.

Фоновых частиц, в том числе, специальных горячих не вводим. Вместо этого приписываем каждой частице некую фиксированную среднюю энергию хаотического теплового движения $E \sim kT$. Пока вводим три градации E : низкая энергия cool, далее обозначаемая символом (c); средняя энергия warm (w); высокая энергия hot (h). Энергия, усредненная по всем частицам мира gem2, дает термодинамическую внутреннюю энергию мира, что и определяет температуру среды. Пока мы не приписываем энергиям частиц численных значений. Поэтому температура среды определяется символически формулой $T \sim (N_c + N_w + N_h) / N$.

Мир gem2 инициализируется вспомогательной программой gem2init, где задаются количества мономеров N_α и N_A , а также относительные количества частиц с разными энергиями N_c/N , N_w/N , N_h/N . Больше никаких исходных данных не требуется.

Ограничения на возможные события в мире gem2 определяются следующими основными положениями из монографии (Галимов).

1. Частицы α способны формировать цепи, обладающие хорошими каталитическими свойствами. Отсюда вводится требование, чтобы все события проходили с неизменным участием частиц α . Следствия: полимеризация частиц α происходит автокаталитически, а полимеризация частиц A происходит только под воздействием сравнительно длинных цепей α . Частицы α неспособны к саморепликации.
2. Частицы A неспособны к полимеризации при случайных столкновениях, а только в присутствии цепи α . Частицы A способны к распознаванию структурного соответствия с частицами как типа A, так и типа α . Это обуславливает принципиальную возможность как саморепликации цепей A, так и транскрипции цепи α с матричной цепи A. Возможность взаимного распознавания частиц технически задается словарями Кодон-Антикодон и Кодон-Аминокислота. Ранее было показано, что эти словари могут быть сформулированы программой автоматически путем отбора из многообразия частиц, изображаемых случайными последовательностями чисел (или символов).

Работа второй части программы, gem2go, как и раньше, сводится к случайному поиску пары частиц, между которыми может произойти реакция, если выполняются указанные выше ограничения, а также некоторые дополнительные условия, о которых сказано ниже. Если реакция, промежуточная или конечная, произошла, считается, что в мире gem2 случился один исторический шаг. Мы считаем, что любой такой шаг связан с понижением энергии взаимодействующих частиц ради образования химической или водородной связи. Выделившаяся энергия удаляется за пределы нашей открытой неравновесной системы. Но система находится в термостате, откуда после ряда шагов поступает порция энергии, не позволяющая системе понизить свою температуру. Кроме потока тепла в систему вводится поток вещества. Дело в том, что в реакциях присоединения или образования промежуточных комплексов затрачивается строительный материал в виде мономеров A и α . Программа следит, чтобы после ряда шагов в систему поступало некоторое количество мономеров и тем самым поддерживает количество строительного материала на заданном уровне. Этот уровень определяется исходными данными при инициализации мира gem2. Если исторический шаг не случился, то программа заставляет частицы обмениваться энергиями. Это обеспечивает правдоподобную имитацию теплового движения частиц в системе.

На каждом историческом шаге может произойти одно из следующих событий.

$\alpha_i + \alpha_j \rightarrow \alpha_c$. Здесь индексы обозначают энергию частицы. Сочетание ij может быть любой из комбинаций ws, sw, ww . Считаем, что для образования химической связи между двумя α_i надо преодолеть довольно высокий энергетический барьер, поэтому хотя бы одна из сталкивающихся частиц должна обладать энергией w . Более короткая цепь присоединяется к концу более длинной и возникает цепь α_c с энергией c , поскольку образуется новая настоящая химическая связь. Программа следит, чтобы не образовывались сразу слишком длинные цепи α_c . Это сравнительно легкое для мира gem2 событие, поэтому исходный строительный материал может интенсивно уходить на автокаталитическое построение длинных цепей. Пока не возникли цепи A , программа не допускает создания цепи α_c длиннее 5 мономеров. С появлением цепи A длиной более 5 мономеров программа следит, чтобы новые цепи α_c были не длиннее этой цепи. Тем самым мы понуждаем программу генерировать длинные цепи α_c под воздействием информации о цепях A . Тогда у нас появляется больше шансов, что длинные полипептиды произошли в процессе репликации цепей A и последующей транскрипции полипептидов с этих матриц. Косвенным признаком таких событий, а не случайных синтезов, послужит наличие совпадений текстов в длинных полипептидах.

$A_i + A_j \rightarrow AA_c$. Это промежуточный комплекс, в котором две частицы только подготовились к соединению. В таком состоянии они будут ждать столкновения со случайной цепочкой α подходящей длины, которая послужит катализатором. Только после этого события будем считать, что две частицы соединились и готовы к дальнейшим преобразованиям. Для распознавания такой готовности припишем комплексу признак «катализовано». Некатализованный комплекс AA_c может получиться в двух вариантах. Если $ij = ss$, то энергии недостаточно для образования химической связи, а только для образования водородной связи. Значит, две частицы смогут соединиться только по боковым поверхностям, а не образовать в будущем цепь. Смысл существования этого комплекса в будущей репликации цепи кодонов в форме цепи антикодонов. Поэтому проверяется соответствие частиц в обеих цепях. Более короткая цепь мысленно пристраивается боковой поверхностью к более длинной. Начала цепей совпадают. Мы пометим такой комплекс признаками «не цепь», «не катализовано». Если ij одна из комбинаций ws, sw, ww , то энергии достаточно для образования настоящей химической связи. Поэтому мы пометим такой комплекс признаками «цепь», «не катализовано». Тексты в двух цепях могут быть любыми, поэтому структурное соответствие не проверяется

$A_i + \alpha_j \rightarrow A\alpha_c$. Это промежуточный комплекс, в котором две частицы только подготовились к соединению. Здесь ij одна из комбинаций ws, sw, ww , поскольку требуется энергия на преодоление барьера при образовании будущей цепи α . Смысл существования этого комплекса в будущей транскрипции цепи с матрицы A . Поэтому проверяется соответствие частиц в обеих цепях. Более короткая цепь α мысленно пристраивается боковой поверхностью к более длинной цепи A . Начала цепей совпадают. Мы пометим такой комплекс признаками «не цепь», «не катализовано». Здесь может присутствовать цепь α , а не мономер или короткий олигомер. Поэтому не будем сразу помечать комплекс как «катализовано». Будем ждать, пока к этому комплексу не присоединятся частицы α до полного построения полной транскрипции. Тогда пометим его как «катализовано».

$AA_i + A_j \rightarrow AA_c$. Здесь недостроенный комплекс AA_i должен обладать признаками «не цепь», «не катализовано». Тогда цепь A_j достраивает более короткую боковую цепь комплекса до более полной длины при условии структурного соответствия. Здесь ij одна

из комбинаций wc , sw , ww , поскольку требуется энергия на преодоление барьера при образовании будущей полной реплики A . Признаки «не цепь», «не катализовано» не изменяются.

$A\alpha_i + \alpha_j \rightarrow A\alpha_c$. Здесь недостроенный комплекс $A\alpha_i$ должен обладать признаками «не цепь», «не катализовано». Тогда цепь α_j достраивает боковую α -цепь комплекса до более полной длины при условии структурного соответствия. Здесь ij одна из комбинаций wc , sw , ww , поскольку требуется энергия на преодоление барьера при образовании будущей полной цепи α . Признаки «не цепь», «не катализовано» не изменяются.

$AA_i + \alpha_j \rightarrow AA_c + \alpha_c$. Здесь ij одна из комбинаций wc , sw , ww , поскольку требуется активация воздействия катализатора на достроенный комплекс AA_i . Если AA_i обладает признаками «не цепь», «не катализовано», то AA_c приобретает признаки «не цепь», «катализовано». Комплекс AA_c будет ждать столкновения с горячей частицей, которая позволит расщепиться кодоновой и полной антикодоновой цепям. Если AA_i обладает признаками «цепь», «не катализовано», то AA_c приобретает признаки «цепь», «катализовано». В этом случае комплекс AA_c будет ждать столкновения с горячей частицей, которая позволит соединиться двум заготовкам длинной цепи в одно целое.

$A\alpha_i + \alpha_j \rightarrow A\alpha_c + \alpha_c$. Здесь ij одна из комбинаций wc , sw , ww , поскольку требуется активация воздействия катализатора на достроенный комплекс $A\alpha_i$. Комплекс $A\alpha_i$ обладает признаками «не цепь», «не катализовано», а комплекс $A\alpha_c$ приобретает признаки «не цепь», «катализовано». Комплекс $A\alpha_c$ будет ждать столкновения с горячей частицей, которая позволит отсоединиться транскрибированной полной цепи α_j от своей матрицы.

$X_i + X_j \rightarrow X_c + X_c$. Здесь X_i , X_j – любые частицы, а ij одна из комбинаций wh , ch , hh . Горячие частицы способны вызвать преобразования в обеих сталкивающихся исходных частицах. Символом X_c обозначен результат такого возможного преобразования. Это может быть не одна частица, если распадается полный катализованный комплекс. Так, комплекс AA «не цепь» дает цепь кодонов и цепь отвечающих им антикодонов. Комплекс AA «цепь» дает удлинившуюся полинуклеотид, где более короткая исходная цепь присоединена к концу более длинной. Комплекс $A\alpha$ дает транскрибированную полную цепь α и ее полинуклеотидную матрицу. Если в данном событии участвует длинная цепь α или A , то от цепи отщепляется последний мономер. Так что это событие может приводить как к усложнению мира $gem2$, так к его упрощению. Усложнение проявляется в появлении более длинных цепей, так и в появлении их копий как результата репликации или транскрипции.

При первом прогоне программы, когда начальное число частиц было задано небольшим ради простоты осмотра результатов, были получены следующие графики, отражающие динамику развития модели.

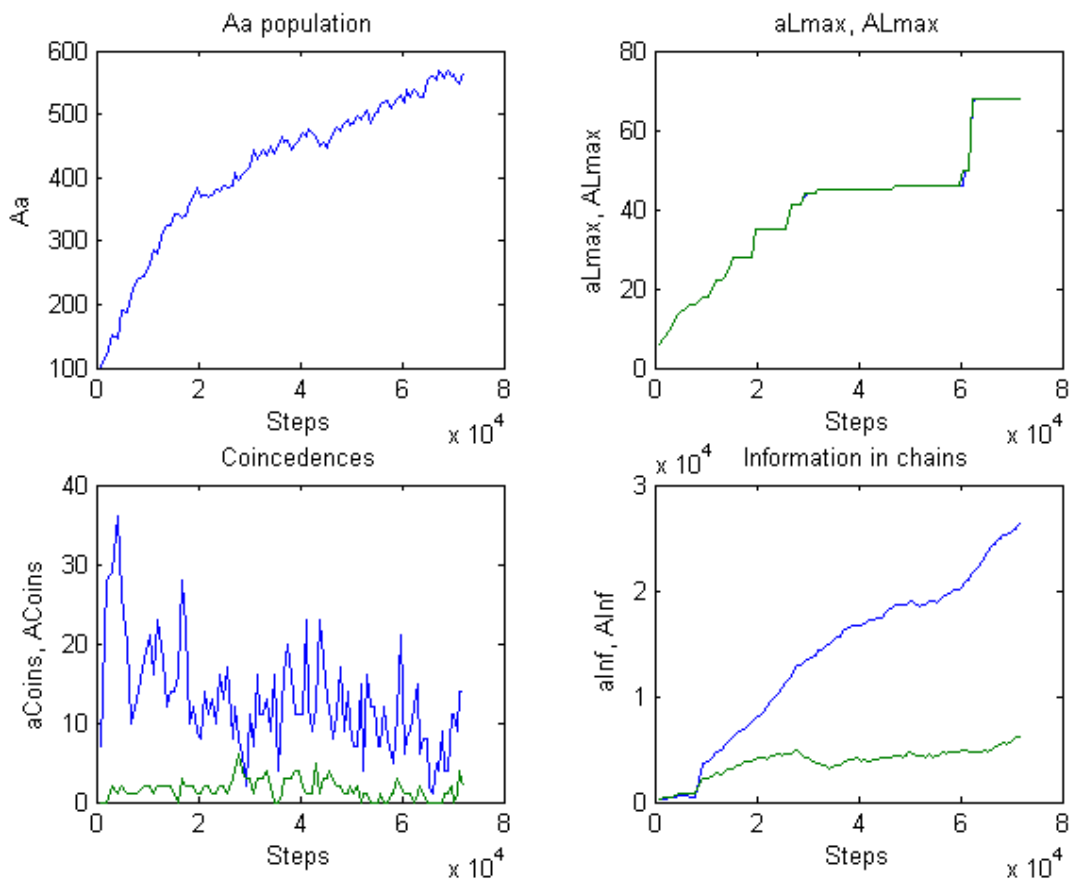


Рис. 6. Динамика изменения параметров модели II при малом начальном числе частиц.

На графике *Aa population* рисунка 6 показана динамика накопления промежуточных комплексов *Aa*. Программа подсчитывает количество всех комплексов – достроенных, недостроенных, катализированных и некатализированных. Учтем, что при инициализации было задано по 100 мономеров того и другого типов. Программа поддерживает число этих мономеров почти неизменным, периодически восполняя их уход. На последнем шаге процесса изменения мира нашлось всего частиц α 240 (из них цепочек 140), *A* 193 (из них цепочек 93), *Aa* 567. Видно, что в процессе эволюции мира строительный материал интенсивно накапливается в пуле *Aa*. Но скорость этого накопления и относительный объем пула куда меньше, чем в модели I, где мы требовали наличия большего разнообразия промежуточных комплексов и частиц t-РНК. Так что мы получили значительное упрощение модели II по сравнению с моделью I. Тем не менее, упрощенная модель также обладает способностью к саморазвитию. Показано, что требование минимизировать подсказки программе реалистично и осуществимо – мир эволюционирует в нужном направлении. И модель справилась с опасностью забыть всё фазовое пространство промежуточными частицами, где почти бесполезно гибнет первичный строительный материал.

На графике *aLmax, ALmax* две кривые практически совпадают. Дело в том, что мы искусственно сдерживаем рост цепей α . Здесь отражается только момент рождения цепи, более длинной по сравнению с ранее рожденными. Многие из них затем укорачиваются. При построении графика *Coincedences* количество одинаковых текстов программа подсчитывает только для сравнительно длинных цепей (> 5 мономеров). Цепь сравнивается с равной по длине или с более длинной цепью. В последнем случае выясняется, совпадает ли текст более короткой цепи с текстом в начале более длинной

цепи. Это позволяет уловить факт утилизации уже сформированных текстов в процессе усложнения мира. Отраженные на графике данные не позволяют пока уловить какие-то закономерности. Тут могут быть и случайные совпадения. Ради поисковых работ здесь мы ввели в компьютер очень упрощенную модель. Кодонов только 4, есть парное соответствие Кодон-Антикодон. Мономеров α только 2. Каждый кодируется двумя кодонами. Так что тексты полимеров получаются примитивными, и случайные повторения текстов в цепях α просто неизбежны. Но в четырехбуквенных цепях A такие случайности уже менее вероятны, и их получается меньше. Ниже показаны эти явления на более сложном и обширном материале.

Для графика *Information in chains* использована стандартная формула Шеннона. Количество информации подсчитывается в битах только для свободных цепей. Информация в промежуточных комплексах программе не подсчитывается.

6. Модель II и компьютерные эксперименты по моделированию самоорганизации химического мира в условиях большой статистики

6.1. Организация параллельных вычислений в процессе моделирования химической самоорганизации.

Рассматривались два пути увеличения статистики в рамках разработанной модели мира полипептидов и полинуклеотидов.

1. Увеличение размерности модели за счет использования однопроцессорного компьютера с высокой частотой работы процессора и с большим объемом оперативной памяти
2. Постановка задачи на многопроцессорный компьютер, работающий в режиме параллельных вычислений

В первом случае не требовалось изменять разработанный ранее алгоритм. Необходимо было только предусмотреть в программе динамическое распределение памяти для описания всех участвующих в модели структур. Это позволяло резко повысить производительность труда при построении и при отладке нового поколения программ. Была также получена большая свобода в выборе размерности модели при проведении компьютерных экспериментов. Нарастивание размерности модели (числа частиц в модели) позволило проследить, как меняется динамика самоорганизации модельного мира и не опасаться при этом переполнения доступной компьютерной памяти.

Однако такой подход обладает одним существенным недостатком. За один рабочий цикл программы (один исторический шаг) она способна выполнить только одну пробную реакцию, ведущую к изменению состояния модельного мира. Это дает малую скорость работы программы даже при высокой частоте работы процессора. Кроме того, это неадекватно событиям, развивающимся в реальном органическом мире, где в органической среде одновременно протекает множество независимых реакций.

Во втором случае за каждый цикл работы программы каждый процессор суперкомпьютера МВС-1000 также выполняет только одну попытку провести реакцию с доступными ему частицами. Но зато одновременно работают несколько процессоров. Это ускоряет эволюцию модельного мира и адекватно реальному поведению химической системы.

Однако реализация нашего алгоритма в режиме параллельных вычислений потребовала значительных усилий, поскольку простой перенос программы с персонального компьютера на суперкомпьютер невозможен без существенной переработки этой программы. Организация данных также требует переработки ради приспособления к особенностям операционной системы суперкомпьютера.

Все эти трудности были успешно преодолены. Более того, в связи с необходимостью переработки программы и данных было достигнуто усовершенствованию самого алгоритма развития сценария.

Будем считать, что один процессор суперкомпьютера осуществляет моделирование событий в одном модельном резервуаре с эволюционирующей химической системой. В Природе все такие резервуары более или менее связаны и потому могут обмениваться генетической информацией. (Это похоже на идею панспермии). Можно промоделировать связь между резервуарами, если воспользоваться свойствами суперкомпьютера в процессе работы вести обмен данными между всеми процессорами, участвующими в данной задаче. Однако реализация такого свойства в программе ведет к очень большим накладным расходам и, потому, к замедлению работы программы и к ее алгоритмическому усложнению. Последнее создает угрозу целостности ранее отлаженного алгоритма и задержки в исполнении данного проекта.

Было реализовано простое решение указанной проблемы. Решение состоит в отказе от организации обменов между процессорами во время работы программы. Такой обмен был осуществлен во время программной организации так называемых контрольных точек. Выглядит это следующим образом.

Задается определенное число исторических шагов, которые должен выполнить каждый процессор суперкомпьютера. Перед остановкой программы создается контрольная точка, то есть во внешнюю память отправляются все структуры частиц. Тем самым запоминаются все состояния резервуаров. При следующем запуске задачи каждый процессор читает из внешней памяти информацию о структурах всех частиц в резервуаре и снова выполняет заданное количество исторических шагов. Количество запусков программы ничем не ограничено и определяется только желанием исследователя. Таким образом можно очень далеко проследить эволюцию модельных миров в каждом резервуаре, не опасаясь, что ценные данные погибнут при сбоях процессоров в процессе длительного счета.

Описанный процесс запусков программы и создания контрольных точек соответствует независимым, изолированным резервуарам. Это не адекватно Природе. Поэтому в схему этого процесса было внесено существенное изменение. В момент создания контрольной точки процессор большую часть своих данных записывает в свою внешнюю память, а меньшую часть – во внешнюю память соседнего процессора. В результате у каждого процессора в контрольной точке возникает две порции данных во внешней памяти – своя и чужая. При следующем запуске программы каждый процессор читает из внешней памяти эти две порции данных. Тем самым моделируется случайное смешивание материала в различных резервуарах.

6.2. Полученные результаты.

С помощью описанной выше программы проведены компьютерные эксперименты, направленные на набор большой статистики, касающейся эволюции мира полипептидов и полинуклеотидов. Выход на большие объемы модели сразу дал новые результаты, недоступные для получения на персональном компьютере стандартной мощности. Это не только количественно новые данные, но и качественно новые закономерности, которые ускользали ранее из-за больших флуктуаций, проявляющихся в системах с малым числом частиц. При большом числе простых исходных частиц (аминокислот и кодонов) получены динамические кривые накопления длинных цепей, почти не искаженные флуктуациями. Это позволило подобрать эмпирические формулы, описывающие кривые накопления

длинных цепей и количества совпадений линейных текстов, представляемых этими цепями. Показано, что характер полученных аналитических зависимостей не меняется при изменении начальных условий. С начальными условиями связаны только значения коэффициентов в найденных зависимостях. Это важный результат. Он говорит о том, что исследуемый сценарий самоорганизации химического мира работает, причем работает устойчиво.

На следующих рисунках приведены результаты экспериментов при последовательном увеличении числа частиц в исходном состоянии модели.

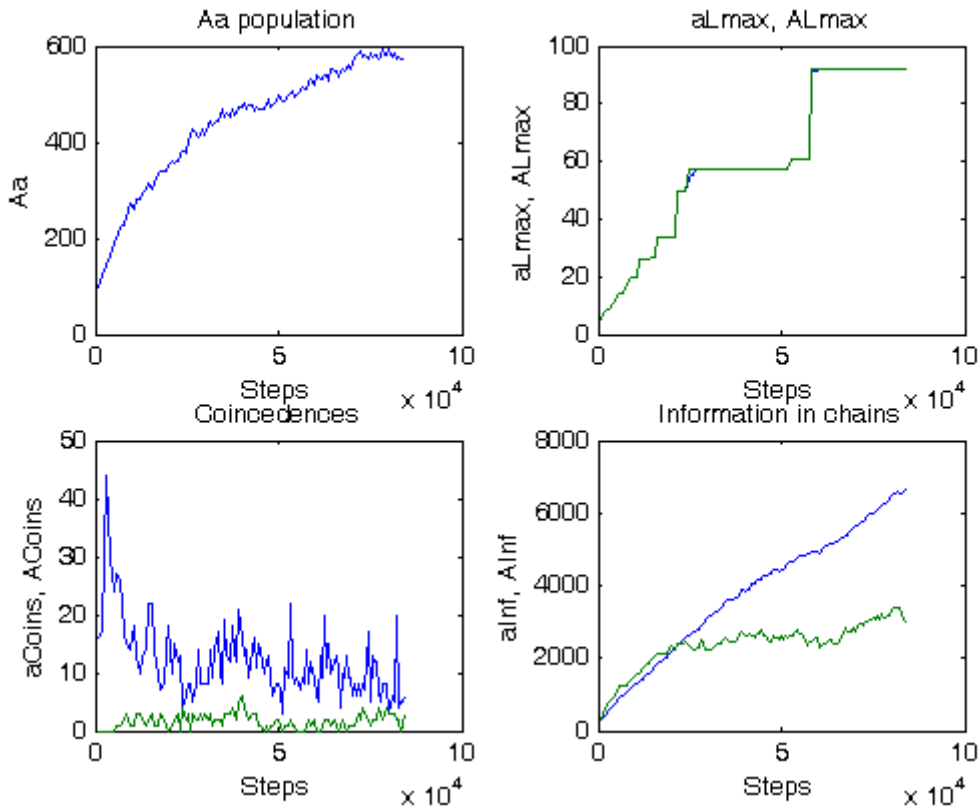


Рис. 7 (Повторение рисунка 6). Динамика изменения параметров модели II при постоянно поддерживаемом объеме исходного строительного материала $N_{acids} = 100$, $N_{codons} = 100$

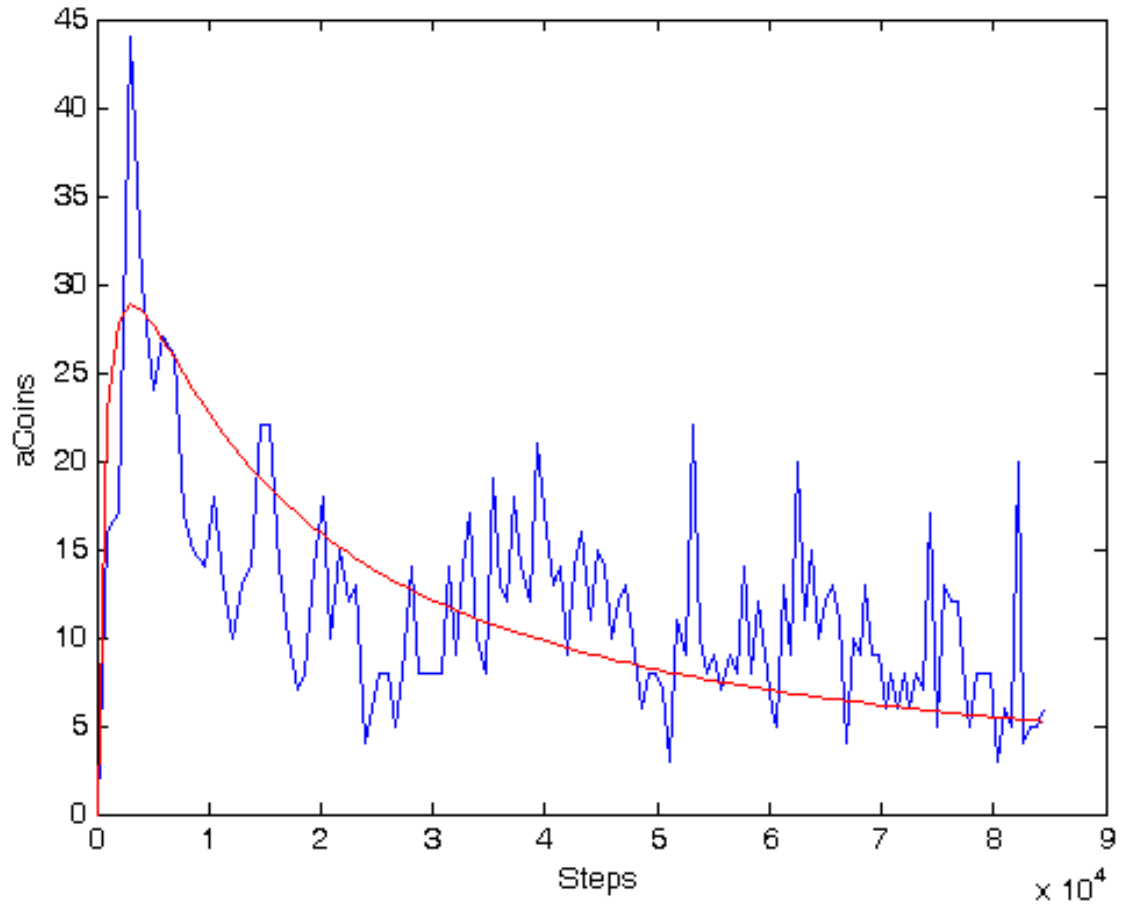


Рис. 8. Аппроксимация графика aCoins аналитическим выражением $aCoins\text{-}approx = steps / (a * steps^2 + b * steps + c)$ при $a = 2 * 10^{-6}$; $b = 2.2 * 10^{-2}$; $c = 20$

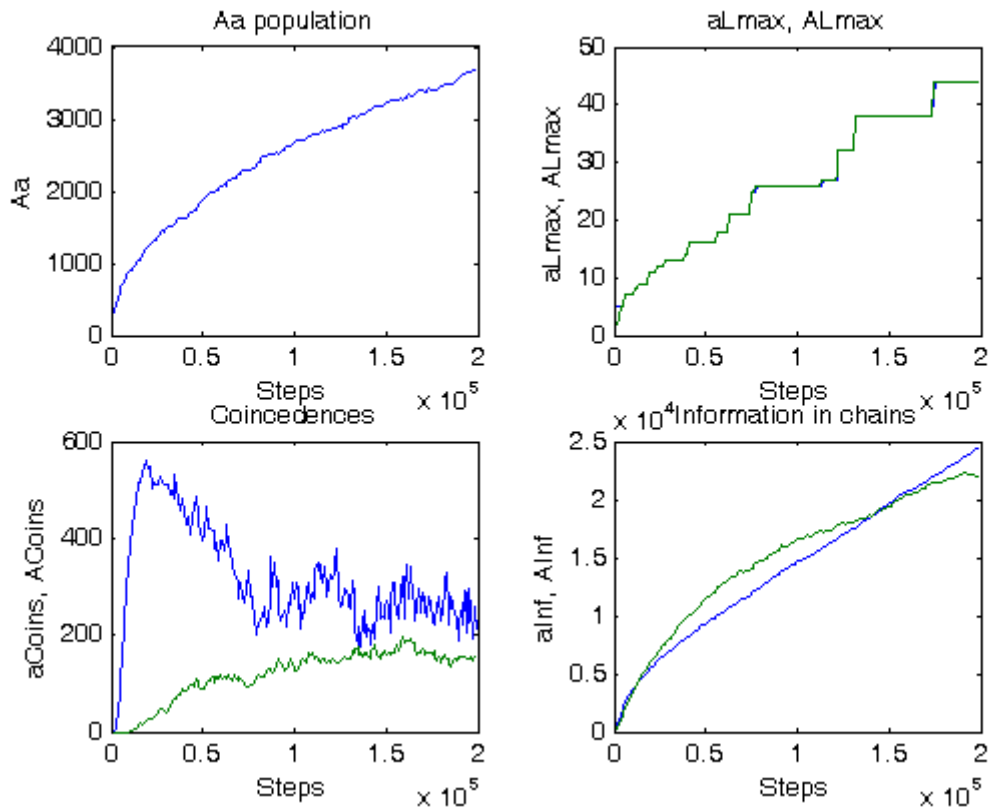


Рис. 9. Динамика изменения параметров модели II при постоянно поддерживаемом объеме исходного строительного материала $N_{acids} = 1000$, $N_{codons} = 1000$

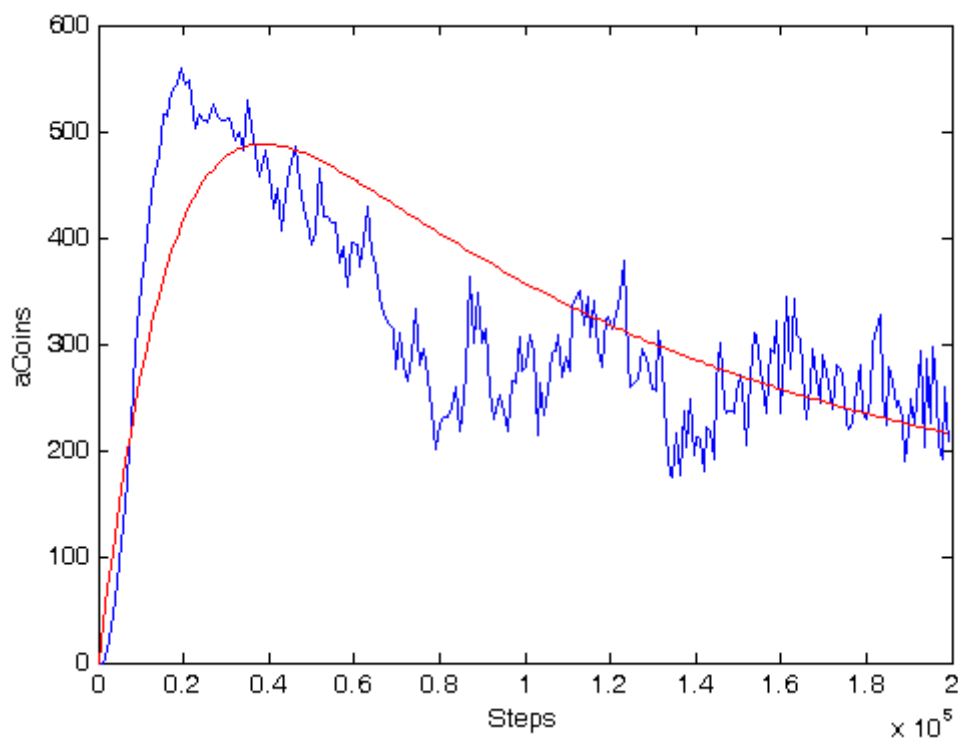


Рис. 10. Аппроксимация графика aCoins аналитическим выражением $aCoins_{approx} = \frac{steps}{a \cdot steps^2 + b \cdot steps + c}$ при $a = 2 \cdot 10^{-8}$; $b = 5 \cdot 10^{-4}$; $c = 30$

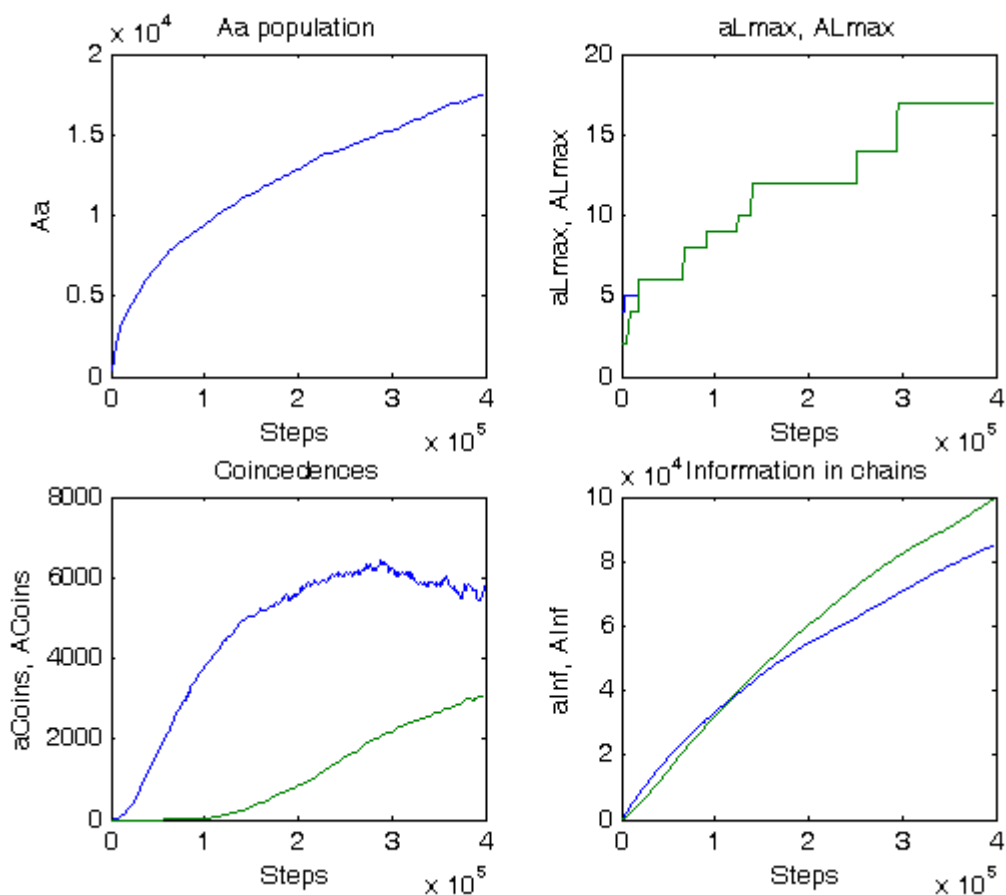


Рис. 11. Динамика изменения параметров модели II при постоянно поддерживаемом объеме исходного строительного материала $N_{acids} = 10000$, $N_{codons} = 10000$

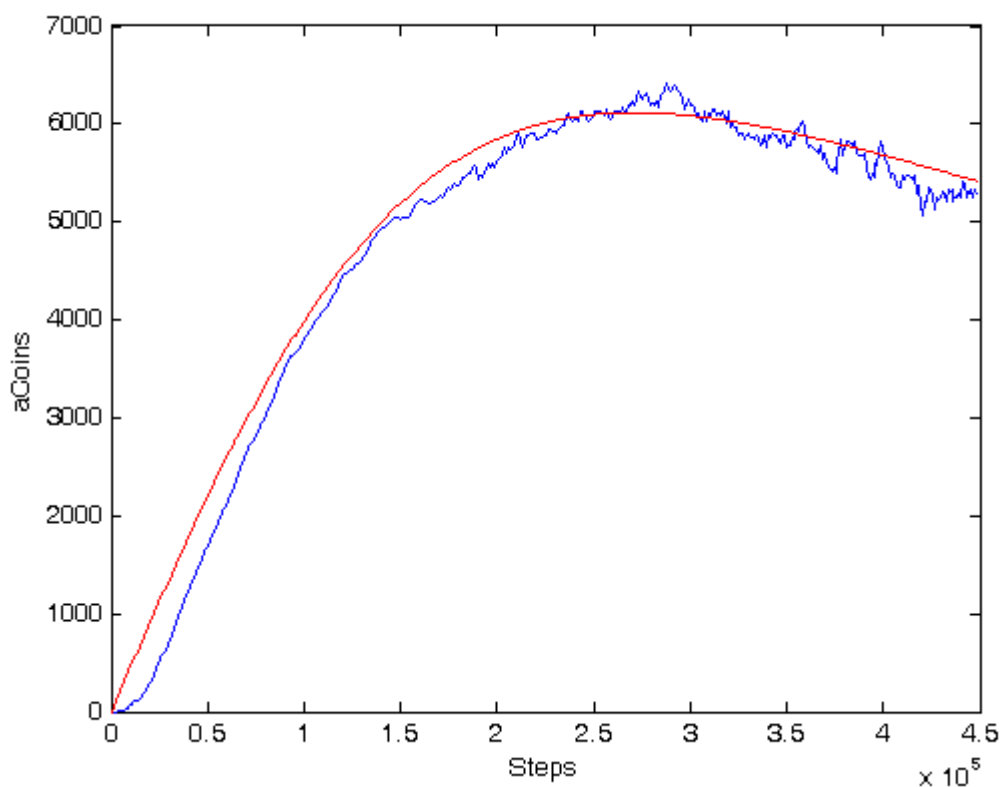


Рис. 12. Аппроксимация графика aCoins аналитическим выражением $aCoins\text{-}approx = steps / (a * steps^2 + b * steps + c)$ при $a = 3 * 10^{-10}$; $b = 1 * 10^{-6}$; $c = 22$

7. Выводы

Полученные в данной работе результаты компьютерных экспериментов подтверждают работоспособность предложенного в книге (Галимов) механизма самопроизвольного упорядочения органического мира, включающего полипептиды и полинуклеотиды, вплоть до возникновения предбиологического генетического кода. Подтверждаются также другие положения, выдвинутые в книге (Галимов) на основе анализа современных знаний, относящихся к феномену жизни, в частности, следующие положения.

Природа, создав электроны и ядра, заложила в них способность будущего химического мира к самоорганизации. Это ведет мир к возникновению генетического кода как к основе феномена жизни. Таким образом, структура современного генетического кода инкорпорирована в механические и электрические свойства микрочастиц и в значения мировых констант, управляющих физикой поведения вещества во Вселенной. Более того, сейчас невозможно найти среди известных классов химических соединений такие два класса, чьи представители могли бы так же эффективно найти и использовать ради своей эволюции друг друга, как это сделали полипептиды и полинуклеотиды.

Исследования проведены при финансовой программы Президиума РАН № 18.

Литература

Галимов Э.М. Феномен жизни. Между равновесием и нелинейностью. Происхождение и принципы эволюции. Москва, УРСС, 2001б 254 с.

Galimov E.M. (2003) Phenomenon of Life: between equilibrium and non-linearity. – *Origin of Life and Evolution of the Biosphere* (in Press).