Геном как функциональная программа

С.В. Козырев, МИАН

Формулировка подхода *"геном как программа"* в рамках функционального программирования, то есть геном является функциональной программой (λ-термом), а проблема эволюции генома примет вид проблемы обучения для функционального программирования.

Функциональное программирование — высокая параллельность, простая система состояний позволяет легче модифицировать программы (контроль за ошибками).

Сравнить с биологией — высокая параллельность процессов в клетке, случайные модификации генетической программы в ходе эволюции обычно не приводят к немедленной поломке программы.

Гиббсовское распределение и скейлинг: геномика, закон Ципфа

Скейлинг размеров семейств паралогичных генов, скейлинг в метаболических сетях и сетях взаимодействующих генов (scale free graphs).

Е.В.Кунин: геном есть *"газ взаимодействующих генов"*, скейлинг должен быть следствием гиббсовского распределения для такой гипотетической модели.

Ю.И.Манин: модель статистической механики для описания степенного закона Ципфа распределения частот слов в корпусах текстов, в которой в качестве гамильтониана использовалась колмогоровская сложность (*"сложность как энергия"*).

С.К.: Эти два подхода можно объединить, если рассмотреть эволюцию как модель температурного обучения, с регуляризацией в виде оценки для колмогоровской сложности.

E. V. Koonin, The Logic of Chance: The Nature and Origin of Biological Evolution, FT Press, 2012.

Y. I. Manin, Complexity vs energy: theory of computation and theoretical physics, Journal of Physics: Conference Series 532 (2014) 012018. arXiv:1302.6695

S.V. Kozyrev, Biology as a constructive physics, p-Adic Numbers, Ultrametric Analysis and Applications, 10:4 (2018), 305–311. arXiv:1804.10518

S.V. Kozyrev, Learning problem for functional programming and model of biological evolution, p-Adic Numbers, Ultrametric Analysis and Applications, 12:2 (2020), 112–122.

S.V. Kozyrev, Genome as a functional program, arXiv:2006.09980

Биология:

Молекулы суть линейные полимеры (белки и нуклеиновые кислоты) — строки символов, состояние системы — набор строк с кратностью (мультистрока).

Химические реакции — преобразования мультистрок, локальные по подпоследовательностям (склейки, разрезания, замены и т.д.). Физические преобразования (перенос молекул) — меняют кратности строк в мультистроке. Обобщение контекстно свободной грамматики по Хомскому на мультистроки.

Геном — набор генов, ген определяет преобразование мультистрок (многозначное отображение). При этом ген сам представляется строкой, а геном мультистрокой.

Гены как преобразования действуют параллельно.

Эволюция — преобразование геномов как мультистрок.

Геном как программа

Системы функционального программирования Дж.Бэкуса.
Функции (не везде определённые), действующие на множестве объектов (отображающие объекты в объекты).
Функциональные формы (например, композиция функций $\circ$) позволяют создавать новые функции из существующих.

J. Backus, Can Programming Be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs, Comm. ACM 21 (8), 613–641 (1978).

Геном — функциональная программа, определяемая рекурсивно через список функций $G = [g_1, \ldots, g_n]$

$$\widetilde{G} = \widetilde{G} \circ G = [\widetilde{G} \circ g_1, \ldots, \widetilde{G} \circ g_n]. \qquad (1)$$

Здесь пространство объектов есть множество S мультистрок.
Программа задаётся списком функций $G = [g_1, \ldots, g_n]$, каждая из функций g_k есть преобразование мультистрок $S \to S$.
g_k – гены, G – геном, $\widetilde{G}$ – геном как программа.

Для гена g_k область определения не обязана совпадать со всем S и отображение является многозначным (применение функции g_k к объекту v представляется лямбда–термом, редукцию в котором можно сделать неоднозначным образом). Например, g_k может осуществлять операцию разрезания строки на две в области подстроки

$$u'uvv' \mapsto u'u + vv',$$

таких подстрок в строке может быть несколько, также g_k может подействовать на разные строки из набора.

Список $G = [g_1, \ldots, g_n]$ есть многозначная функция $S \to S$: к объекту $v \in S$ может быть применена любая из функций g_k из списка (в свою очередь, многозначная).

Множество S мультистрок есть пространство наборов биологических последовательностей (молекул), g_k есть ген, кодирующий белок, осуществляющий преобразование биологических последовательностей (химическую реакцию). Также имеются операции переноса молекул, меняющие кратность некоторых строк в мультистроке (мы рассматриваем операции переноса как некоторые "гены" g_j в G).

Набор $G = [g_1, \ldots, g_n]$ есть геном (список генов), определяющий программу $\widetilde{G}$ (многозначное отображение $S \to S$) рекурсивно (программа $\widetilde{G}$ есть неподвижная точка генома G как лямбда–терм).

При этом каждый ген g_k (рассматриваемый выше как отображение $S \to S$) также кодируется биологической последовательностью, то есть гены g_k и геном G как список генов представляются объектами из S.

Метаболическая сеть как редукционный граф

Пусть $v_0 \in S$ есть мультистрока. Определим граф $\Gamma_{\widetilde{G}}(v_0)$ для программы $\widetilde{G}$ (в интерпретации через лямбда-исчисление это редукционный граф программы в "ленивой" стратегии):

Шаг 0) Стартуем с вершины v_0.

Шаг 1) Применим G к v_0, при этом может быть применён к v_0 любой из $g_k \in G$ (неоднозначным образом). Включим в граф все вершины, полученные из v_0 при помощи многозначного отображения G (отождествляя вершины, совпадающие как мультистроки) и соединим полученные вершины рёбрами с v_0.

Шаг 2 и т. д.) По рекурсивному определению снова применим к полученным на шаге 1 вершинам многозначное отображение G (действуем только на вершины, полученные на предыдущем шаге). Включим в граф $\Gamma_{\widetilde{G}}(v_0)$ все получаемые таким образом вершины и рёбра. Итерируем процесс и получим граф $\Gamma_{\widetilde{G}}(v_0)$.

Некоторые гены g_k в программе G отвечают операциям переноса, меняющим кратность некоторых строк в мультистроке. Такие преобразования позволяют замкнуть метаболические циклы в графе. Физические преобразования позволяют объединить некоторые графы $\Gamma_{\widetilde{G}}(v_0)$ с разными v_0 в единый граф. Такой "большой" граф мы будем обозначать $\Gamma_{\widetilde{G}}$, он объединяет графы $\Gamma_{\widetilde{G}}(v_0)$ с разными "разумными" v_0 и соответствует метаболическому графу для генома G.

Сопоставим каждому гену g в геноме G пару неотрицательных чисел $r_+(g)$, $r_-(g)$ — скоростей перехода для соответствующей реакции. Такие скорости определяют марковское случайное блуждание на графе $\Gamma_{\widetilde{G}}$ со скоростями переходов $r_+(g)$, $r_-(g)$ вдоль ребра и в обратном направлении (рёбра отвечают действию генов).

Рассмотрим некоторый линейный функционал $A(f)$ на распределениях $f(v)$ на вершинах графа. Например, поток через ребро v_1v_2 со скоростями переноса r_+ и r_- вдоль и против направления ребра (от v_1 к v_2 и наоборот) равен $r_+f(v_1) - r_-f(v_2)$. Разные рёбра могут отвечать одному и тому же химическому преобразованию, где пары вершин v_1v_2 отвечают разному количеству реагентов. Тогда для вычисления полного потока следует просуммировать величину $r_+f(v_1) - r_-f(v_2)$ по всем таким парам вершин v_1v_2.

Будем считать, что для рассматриваемой системы кинетических уравнений существует единственное стационарное состояние $f_{\widetilde{G}}$, к которому решение системы сходится, и будем рассматривать функционалы (потоки) $A(f_{\widetilde{G}})$ в стационарном состоянии.

Программа $\widetilde{G}$ имеет высокий параллелизм. Корректность работы метаболических сетей связана со свойством Чёрча–Россера для лямбда–исчисления (при разной последовательности применения генов мы можем получить нужный результат).

Программа $\widetilde{G}$ для генома зацикливается — это отвечает метаболическим циклам в метаболической сети $\Gamma_{\widetilde{G}}$.

Набор чисел $r_+(g_k)$, $r_-(g_k)$ — скоростей перехода вдоль рёбер графа $\Gamma_{\widetilde{G}}$ (скоростей реакций и скоростей переноса) и соответствующее стационарное состояние $f_{\widetilde{G}}$ — образуют состояние для генома как программы $\widetilde{G}$.

Генная регуляция — изменяя величины $r_+(g)$, $r_-(g)$, мы будем менять стационарное состояние $f_{\widetilde{G}}$ и вклады в функционал $A(f_{\widetilde{G}})$ от различных метаболических путей.

Генная регуляция осуществляется при помощи репрессоров и промоторов (в частности, для lac operon) — задание приоритета при экспрессии генов. Аналог в функциональном программировании — монады.

Другой механизм генной регуляции: эпигенетика — метилирование генома, гистоновый код, пространственный код (укладка хроматина).

Биологическая эволюция — действие "эволюционной программы" $\widetilde{E}$ с "генами эволюции" $E = [e_1, \ldots, e_m]$ (операциями редактирования генома), определённой рекурсивно

$$\widetilde{E} = \widetilde{E} \circ E = [\widetilde{E} \circ e_1, \ldots, \widetilde{E} \circ e_m]. \qquad (2)$$

Эволюция преобразует геномы как наборы слов в геномы, преобразует скорости перехода $r_+(g)$, $r_-(g)$ для генов, стационарное состояние $f_{\widetilde{G}}$ и функционал $A(f_{\widetilde{G}})$.

Температурное обучение

Задача машинного обучения — минимизация по пространству параметров суммы функционала потерь и регуляризатора

$$H(s) = R(s) + Reg(s) \to \min .$$

Регуляризация нужна для борьбы с переобучением (снижения энтропии пространства параметров s, см. VC–теория).

Обучение при ненулевой температуре — вместо минимизации рассматривается статсумма, $\beta > 0$ есть обратная температура

$$Z = \sum_s e^{-\beta H(s)}.$$

В пределе нулевой температуры $\beta \to \infty$ задача вычисления Z переходит в задачу минимизации H (температурное обучение переходит в обычное).

Дарвиновская эволюция — машинное обучение (генерация программы по данным). Машинное обучение предложено А.Тьюрингом, им же отмечена аналогия с дарвиновской эволюцией.

A. M. Turing, Can machines think? Computing Machinery and Intelligence. Mind 49: 433–460 (1950).

Идеи машинного обучения в эволюции — регуляризация в эволюции оценкой для колмогоровской сложности для борьбы с переобучением.
Телеология — развитие, объясняемое конечными целями.
Телеология в эволюции есть разрешимость задачи машинного обучения для эволюции.
Механизм — сочетание функции и низкой сложности.
Минимизация функционала потерь с регуляризацией в виде сложности — биологические системы суть механизмы для выполнения биологической функции.

Температурная дарвиновская эволюция — статсумма вместо оптимизации. Температурные модели в эволюции (эффективный размер популяции есть обратная температура эволюции). Скейлинг в геномике как проявление гиббсовского распределения взаимодействующего газа генов (Кунин), скейлинг для закона Ципфа — сложность как энергия (Манин).

— Скейлинг объясняется вкладом от регуляризации в функционал обучения — универсальная регуляризация (в виде оценки для колмогоровской сложности) объясняет универсальность скейлинга.

Температурное обучение для функциональных программ.
Рассмотрим эволюционную программу $\widetilde{E}$ вида (2) с редукционным графом $\Gamma_{\widetilde{E}}(G_0)$ для генома–предка G_0. Сопоставим действию эволюционной операции e_k вес (положительное число) $K(e_k)$, а ориентированному пути p из u в v в графе $\Gamma_{\widetilde{E}}(G_0)$ сопоставим функционал действия — сумму весов рёбер в пути

$$K_{\widetilde{E}}(p) = \sum_k K(e_{i_k}). \tag{3}$$

Такой функционал действия рассматривается как стоимость вычисления вдоль пути p, или взвешенная оценка колмогоровской сложности порождения v из u.

Определим дарвиновскую эволюцию как температурную задачу обучения при обратной "температуре эволюции" β' со статсуммой

$$Z[\widetilde{E}, G_0] = \sum_{G \in \Gamma_{\widetilde{E}}(G_0)} A(f_{\widetilde{G}}) \sum_{p \in \mathrm{Path}\left(\Gamma_{\widetilde{E}}(G_0)\right): G_0 \to G} e^{-\beta' K_{\widetilde{E}}(p)}. \quad (4)$$

Суммирование идёт по путям p из предкового генома G_0 в геном–потомок G, затем суммируем по потомкам G.
Эта статистическая сумма сосредоточена на геномах с большим функционалом $A(f_{\widetilde{G}})$ (например, отбор идёт по высоким значениям функционала потока).
Суммирование по путям описывает параллелизм метаболических путей в клетке и параллелизм в эволюции (вычисление типичного функционала $A(f_{\widetilde{G}})$ включает суммирование по путям, что описывает параллелизм в метаболизме). Гиббсовский фактор от функционала действия ограничивает сложность дающих вклад в статсумму операций для эволюционной программы и служит регуляризацией для задачи обучения функциональной программы (регуляризация при помощи оценки колмогоровской сложности).

Недетерминированный алгоритм задаётся недетерминированной машиной Тьюринга (НМТ), которая на некоторых шагах может дуплицироваться и выполнять две (или более) ветки вычислений, многократная дупликация позволяет организовать перебор с одновременным выполнением всех возникающих веток вычислений.

Функционирование клетки и эволюция — программы (1), (2) суть программы для НМТ — G и E суть многозначные отображения, рекурсия многозначных отображений порождает много веток вычислений.

Параллелизм в биологии — в клетке химические реакции проходят параллельно и эволюция есть сложный параллельный процесс. Предлагается рассматривать такой параллелизм как проявление свойств НМТ. Геном есть функциональный недетерминированный алгоритм, дарвиновская эволюция есть функциональная задача обучения НМТ при ненулевой температуре.

Выводы

Предложена модель генома как функциональной программы (1), определённой рекурсивно набором генов, *"жизнь есть неподвижная точка генома"*.

Работа такой программы описывается распределением $f_{\widetilde{G}}$ для *"взаимодействующего газа генов"*.

Эволюция описывается функциональной программой (2), *"дарвиновская эволюция путём отбора есть задача температурного обучения"* (4) для программы эволюции. Статсумма задачи температурного обучения функциональной программы эволюции содержит сумму по путям редукции от гиббсовских факторов от функционала действия (3) (цены вычисления вдоль пути редукции, *"сложности как энергии"*) и отвечает степенным законам в геномике.

Параллелизм в работе клетки и в эволюции описывается недетерминированными алгоритмами.