ISSN 0233–7657. Biopolymers and Cell. 2012. Vol. 28. N 3. P. 163–170

UDC 577.218

Огляд методів моделювання мереж генної регуляції: булеві і баєсові мережі

А. О. Фролова

Інституту молекулярної біології і генетики НАН України
Вул. Академіка Заболотного, 150, Київ, Україна, 03680

fshodan@gmail.com

Однією з проблем сучасної системної біології є моделювання мереж генної регуляції, які у найповнішому вигляді відтворюють регуляторні взаємодії між генами всього організму. Надзвичайна обчислювальна складність цієї задачі та відсутність ґрунтовних оглядів методів реконструкції генних мереж є значною перешкодою для подальшого розвитку цього напрямку системної біології. У даній статті розглянуто два найпоширеніших методи моделювання мереж генної регуляції: булеві і баєсові мережі, та наведено математичний опис кожного з них, а також розкрито декілька алгоритмічних підходів до моделювання генних мереж за допомогою цих методів, вказано на складність алгоритмів та зазначено проблеми, що виникають при їхньому застосуванні.

Ключові слова: реконструкція мереж генної регуляції, булеві мережі, баєсові мережі.

Вступ. Мережа генної регуляції – це сукупність опосередковано пов'язаних між собою модульних елементів ДНК (генів), які приймають множинні вхідні сигнали у вигляді РНК і білків, обробляють сигнали і зумовлюють темп, за якого гени мережі транскрибуються в РНК і транслюються у білки. Архітектура мережі віддзеркалює взаємодію її різних елементів і дає найповніше уявлення щодо регуляції функціонування клітини на відміну від традиційного дослідження поодиноких генів, тому реконструкція генних мереж є важливим предметом вивчення системної біології.

Наразі використовують близько 10 підходів до моделювання генних мереж, серед них машинне навчання, баєсові мережі, булеві мережі, диференційні рівняння, теорія інформації, мережі Петрі, нейронні мережі, генетичні алгоритми [1–3]. Кожен з цих підходів, звісно, має свої переваги та недоліки, визначення яких ускладнене нечисельністю ґрунтовних оглядів у науковій літературі. Іншою проблемою є те, що згадані підходи застосовують для реконструкції невеликих мереж, що нараховують лише 10–20 генів. Зі збільшенням кількості генів обчислювальна складність зростає експоненційно: для 30 генів вже маємо $2{,}71{\cdot}10^{158}$ можливих варіантів мереж у разі використання баєсових мереж [4], хоча для теорії інформації існує значно менша оцінка складності [5]. Однак задача побудови генних мереж все одно є NP-повною [1], тому важливою складовою оглядів повинна бути оцінка часової складності алгоритму реконструкції та аналіз алгоритмів, що дозволить виявити можливість розпаралелення для застосування їх на розподілених обчислювальних системах, кластерах.

У даній статті розглянуто два методи реконструкції – булеві та баєсові мережі, розкрито декілька алгоритмічних підходів і зроблено їхню оцінку.

Різні методи представлення генних мереж. Одну й ту саму мережу генної регуляції можна репрезентувати по-різному (рис. 1). Найпростішим способом є орієнтований або неорієнтований граф.

Орієнтований граф G – це пара $<V, E>$, де V – множина вершин, а E – множин ребер. Вершини

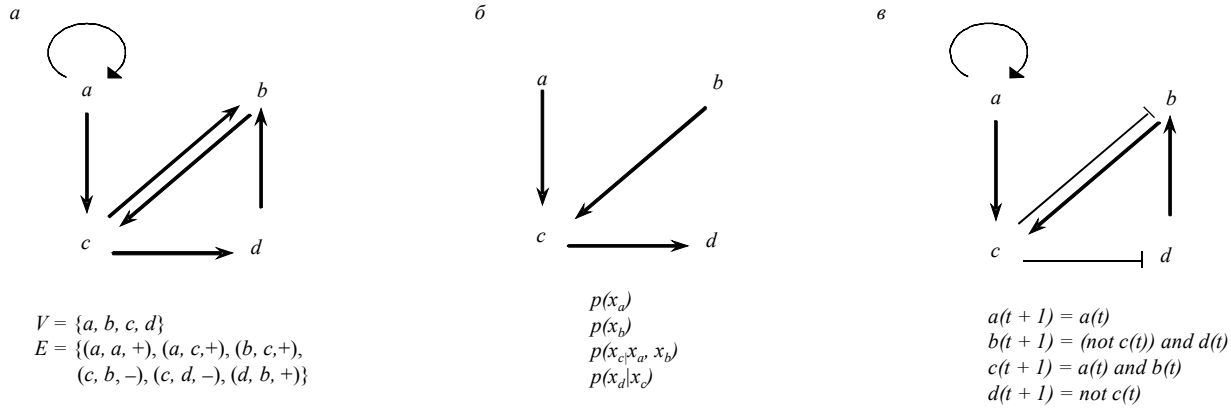

Рис. 1. Різні види представлення мережі генної регуляції з чотирьох генів *a*–*d* [6]: *а* – мережа у вигляді орієнтованого графа (напрямок взаємодії вказано знаками «+» і «–», тобто активація і інгібування); *б* – відповідна модель у вигляді баєсової мережі (варто звернути увагу на те, що деякими взаємодіями знехтувано, зокрема, інгібуванням гена *b* геном *c* та активацією гена *b* геном *d* для того, щоб отримати мережу без циклів); *в* – булева мережа

відповідають генам (або іншим компонентам системи), а ребра, що позначаються як пара вершин <*i*, *j*>, відповідають регуляторним взаємодіям компонентів. Граф буде орієнтований, якщо *i* та *j* будуть присвоєні голові і хвосту ребра відповідно. Визначення вершин і ребер можуть бути розширені для зберігання додаткової інформації про гени та їхні взаємодії. Наприклад, можна визначити ребро як трійку <*i*, *j*, *properties*>. Слот *properties* може показувати, інгібує (–) чи активує (+) один ген інший (див. рис. 1, *а*). Також *properties* може бути списком регуляторів і їхнього впливу на дане ребро, наприклад, <*i*, *j* ((*k*, *активатор*), (*l*, *інгібітор як гомодимерний білок*))> [6].

Булеві мережі. У булевих синхронних мережах рівень експресії генів визначається бінарною змінною, яка приймає значення 0 або 1, тобто ген є або вимкненим, або експресуючим. Стан генів змінюється одночасно з кожним дискретним кроком часу, тому мережі називаються синхронними. Новий стан гена може залежати від попереднього стану цього гена та від попередніх станів інших генів. *N* вузлів булевої мережі – це *N* генів регуляторної мережі, *k* входів кожного з вузлів (під *k* потрібно розуміти максимальну кількість входів для кожного з вузлів) – *k* взаємодій, що регулюють експресію певного гена. *k* входів у конкретний вузол обумовлюють бінарний рівень експресії відповідного гена. Оскільки кожна вершина перебуває лише у двох станах, мережа з *N* генів може мати 2^N різних станів. *N*-вимірний вектор змінних може описувати стан у час *t*. Значення кожної змінної у час $t + 1$ залежить від вхідних даних, які можуть бути обчислені за допомогою булевих функцій. Для вершини з *k* входами кількість можливих булевих функцій дорівнює 2^{2^k}.

Для прикладу наведемо правила, що застосовують для мережі на рис. 1, *в*:

$$
\begin{aligned}
a(t+1) &= f_a(a(t)) = a(t), && k = 1;\\
b(t+1) &= f_b(c(t), d(t)) = (\neg c(t)) \wedge d(t), && k = 2;\\
c(t+1) &= f_c(a(t), b(t)) = a(t) \wedge b(t), && k = 2;\\
d(t+1) &= f_d(c(t)) = (\neg c(t)), && k = 1.
\end{aligned}
$$

За цими правилами можна побудувати таблицю переходів з одного стану в інший, з якої видно, що ця мережа має два типи стаціонарної поведінки. Якщо початковий стан *a* дорівнює 0, то система набуває стабільного стану 0101, що означає: гени *a*, *c* є вимкненими, а гени *b*, *d* – ввімкненими. Якщо ж початковий стан *a* дорівнює 1, то система зациклюється, постійно проходячи наступний ланцюг станів: $1000 \rightarrow$ $\rightarrow 1001 \rightarrow 1101 \rightarrow 1111 \rightarrow 1010 \rightarrow 1000$ [6].

Послідовність станів, утворених булевими перетвореннями, являє собою траєкторію системи. Оскільки кількість станів є скінченною, набір можливих переходів теж скінченний. Тому кожна траєкторія веде або до стаціонарного стану, або до циклічного. Такі стани називають атракторами. Всі стани, що ведуть до однакового атрактору, утворюють басейн атракції.

Булеві мережі використовують для дослідження загальних властивостей великих генних мереж. Роз-

Стани булевої мережі

$abcd \to a'b'c'd'$	
Інтерпретація станів при $a = 0$	Інтерпретація станів при $a = 1$
$0000 \to 0001$	$1000 \to 1001$
$0001 \to 0101$	$1001 \to 1101$
$0010 \to 0000$	$1010 \to 1000$
$0011 \to 0000$	$1011 \to 1000$
$0100 \to 0001$	$1100 \to 1011$
$0101 \to 0101$	$1101 \to 1111$
$0110 \to 0000$	$1110 \to 1010$
$0111 \to 0000$	$1111 \to 1010$

глядаючи випадкові булеві мережі (кількість входів k на один ген та відповідні булеві функції обираються випадково), Кауфман [7, 8] знайшов, що така система показує досить впорядковану динаміку при маленьких k та конкретні вибори правил. Середня очікувана кількість атракторів є $\sqrt{N}$, і середня довжина атракторів обмежена значенням, пропорційним $\sqrt{N}$. Кауфман зробив припущення щодо інтерпретації числа можливих атракторів як числа клітин різних типів. Ця цифра значною мірою корелює з відомими наразі знаннями про типи клітин [9].

Для реконструкції булевих мереж за даними мікромасив-експериментів можна використовувати алгоритм, описаний у [10, 11]. Цей алгоритм визначає, чи пояснює набір вершин $\nu_1, \nu_2, ..., \nu_k$, $k \le N$ експресію певної вершини ν_i. Булеву функцію активатор–інгібітор, що приписується вершині ν_i, можна визначити методом перебору, вона має вигляд

$$\nu(t) = (\nu_1(t) \vee \nu_2(t) \vee ...) \wedge \neg(\nu_j(t) \vee \nu_{j+1}(t) \vee ...),$$

де перша дужка – це вершини-активатори, а друга – вершини-інгібітори.

Очевидно, що при невеликих k складність алгоритму буде поліноміальна, але це може значно вплинути на якість отриманої мережі. Як уже зазначалося вище, кількість усіх можливих булевих функцій дорівнює 2^{2^k}, тому при збільшенні значення k отримуємо експоненційну складність.

Більш загальний та ґрунтовний підхід до вирішення проблеми знаходимо у [12–14], автори яких використали обмежені булеві мережі. У цьому разі регуляторні відносини представлені матрицею $A_{n \times n}$, де $a_{ij} = 1$ за позитивної регуляції гена x_i геном x_j; $a_{ij} = -1$ при негативній регуляції гена x_i геном x_j та $a_{ij} = 0$ в інших випадках.

Отже, булева функція f_i визначається відповідно до матриці A та значень генів $x_j, j = 1, ..., n$ у час t:

$$x_i(t+1) = \begin{Bmatrix} 1, if \sum_j a_{ij} x_j(t) > 0; \\ 0, if \sum_j a_{ij} x_j(t) < 0; \\ x_i(t), if \sum_j a_{ij} x_j(t) = 0. \end{Bmatrix}$$

Сума $\sum_j a_{ij} x_j(t) > 0$ – вхід гена x_i у час t. Оскільки не всі булеві функції можуть бути задані при такому представленні, то булеву мережу називають обмеженою. Реконструкція генних мереж зводиться до вирішення проблеми відповідності обмежувальним умовам (Constraint satisfaction problem, далі CSP).

CSP визначається набором змінних $X = \{x_1, x_2, ..., x_n\}$; кортежів $D = \{D_1, D_2, ..., D_n\}$, де D_i – доменний кортеж для x_i; обмежень $C = \{C_1, C_2, ..., C_m\}$, які обмежують значення, які змінні можуть приймати одночасно, де кожен набір C_i містить обмеження підмножини змінних та визначає допустиму комбінацію значень для цих змінних. Рішенням CSP є присвоєння кожній змінній x_i значення з її домену D_i таким чином, щоб задовольнити усі обмеження в C [15].

CSP, визначені на скінченних доменах, зазвичай розв'язуються пошуковими алгоритмами, а саме – покроковим присвоєнням можливих значень змінним та перевіркою виконуваності усіх обмежень. Відомими алгоритмами є бектрекінг, розширення обмежень, локальний пошук [12]. Процеси вибору змінних і присвоєння значень цим змінним чутливі до порядку, в якому здійснюється вибір, тому існує чимало евристик для вирішення CSP [15], що, звичайно, впливає на точність реконструкції.

Окрім зазначеної проблеми, можна виділити низку загальних вад реконструкції булевих мереж з реальних даних [6]:

1. Бінаризація – складний процес, який значно впливає на результат. З даних експресії не завжди

легко визначити, який саме рівень бінаризації повинен бути.

2. Стани неповні. На практиці більшість переходів між станами втрачається після бінаризації.

3. Наявність великої кількості часових точок є критичною. Щоб відсіяти правильні стани від неправильних, необхідно здійснити багато переходів між станами для отримання стабільного результату.

4. Часові точки не повинні бути дуже близькими одна до одної. Це хитрий баланс між отриманням якомога більшої кількості переходів станів і одержанням false-positive станів. Якщо дві часові точки занадто близькі, то перехід між ними не виявляє змін, оскільки бінаризація – це дуже грубий поріг, який не розрізняє незначних варіацій концентрації. Це призводить до появи великої кількості false-positive циклів у відповідному графі.

Варто зауважити, що булеві мережі не є суто методом реконструкції, радше методом представлення, тому для моделювання використовують багато різних підходів [16].

Так, булеву генну мережу можна реконструювати за допомогою теорії інформації, як це зроблено у відомому алгоритмі REVEAL [17]. Однак теорія інформації є окремою сукупністю методів реконструкції і потребує окремого детального огляду, тому обмежимося наразі лише згадуванням [18–21]. Як бачимо, класифікація методів реконструкції не є такою вже суворою, і часто для вирішення задачі моделювання генних мереж застосовують комплекс підходів.

Баєсові мережі відображають регуляторні генні мережі як орієнтований ациклічний граф $G = <V, E>$. Як і у визначенні для звичайного графа, вершини $i \in V$ відповідають генам, а ребра – регуляторним взаємодіям. Змінні x_i належать вершинам i і позначають регуляторні властивості, наприклад, рівень експресії гена або кількість активних білків. Умовний ймовірнісний розподіл $p(x_i \mid L(x_i))$ визначено для кожного x_i, де $L(x_i)$ – змінна, що належить до прямих регуляторів i.

Орієнтований граф G та умовний розподіл разом описують об'єднаний ймовірнісний розподіл $p(x)$, що визначає баєсову мережу. Його можна розкласти наступним чином:

$$p(x) = \prod_i p(x_i \mid L(x_i)).$$

Орієнтований граф виражає залежності ймовірностей: рівень експресії гена, представленого дочірньою вершиною, залежить від рівня експресії генів-батьків. Звідси граф також має умовні незалежності $i(x_i; y \mid z)$, що означає: x_i не залежить від y за умови, що є z. Два графи, які відображають баєсову мережу, є еквівалентними, якщо множини їхніх незалежностей однакові. Але в такому разі їх можна вважати лише однаковими неорієнтованими графами. Повністю еквівалентні графи неможливо виявити тільки з досліджень змінної x [22].

Для мережі на рис. 1, *б*, умовні незалежні відношення мають вигляд [6]:

$$i(x_a; x_b) \cdot i(x_d; x_a, x_b \mid x_c),$$

а об'єднаний ймовірнісний розподіл мережі [6] –

$$p(x_a, x_b, x_c, x_d) = \\ = p(x_a) \cdot p(x_b) \cdot p(x_c \mid x_a; x_b) \cdot p(x_d \mid x_c).$$

При реконструкції мереж генної регуляції з даних експресії за допомогою баєсових мереж метою є знаходження мережі або класу еквівалентних мереж, які найкраще пояснюють дані експерименту. Проблема полягає у визначенні початкового ймовірнісного розподілу. Однак доцільнішим є використання динамічних баєсових мереж, які можна розглядати як розширення звичайних баєсових мереж і які здатні відобразити динаміку генних мереж. Нехай змінна серійних даних мікромасив-експерименту $x \in R^{n \times p}$, x_{ti}, де n – кількість часових точок, а p – кількість генів, позначає спостереження гена i у час t, тоді вектор спостереження у час t можна представити у вигляді $x_{(t)} = [x_{t1}, ..., x_{tp}]^T$ а i-й ген у всіх часових точках – $x_{(i)} = [x_{1i}, ..., x]^T$.

Динамічні баєсові мережі припускають залежність від часу, в якій направлені ребра повинні «рухатися» вперед із часом [23].

Зазвичай припускають, що такі мережі – це моделі Маркова першого порядку, в яких на кожен ген безпосередньо впливають лише попередні гени [24]. Оскільки ці моделі залежні від часу, то легко побудувати мережу з циклами зворотного зв'язку. На

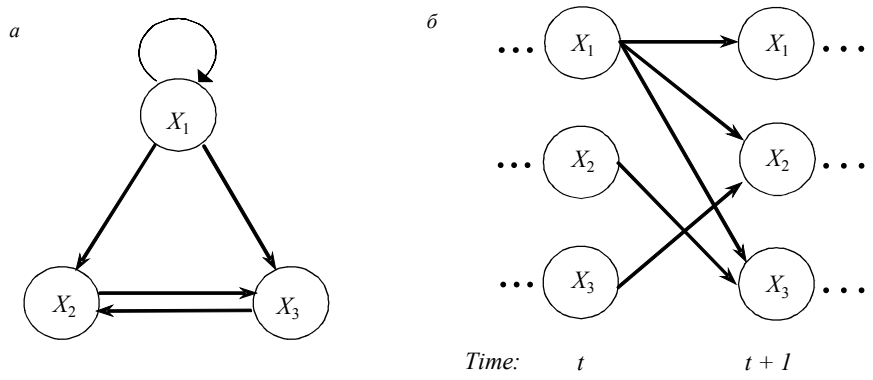

Рис. 2. Перетворення простої мережі на динамічну баєсову мережу [23]: *а* – проста генна мережа, де X_2 та X_3 формують цикл, а X_1 авторегулюється; *б* – еквівалентна динамічна баєсова мережа без циклів

рис. 2 зображено, як циклічну мережу з трьох генів легко перетворити на ациклічну динамічну баєсову мережу [23].

Сумісний ймовірнісний розподіл для динамічної баєсової мережі може бути обчислений як

$$P(x_{11},\ldots,x_{np}) = \prod_{i=1}^{p}\prod_{t=1}^{n} P(x)_{ti} \mid L(x_{ti}).$$

Вище зазначалося, що головною метою реконструкції генних мереж є побудова таких мереж, які б найкраще пояснювали експериментальні дані. Для цього потрібно знайти структуру і параметри динамічної баєсової мережі з даних. Цю задачу можна сформулювати наступним чином.

Маємо набір даних з різних часових точок, коли $D = \{x_1, \ldots, x_n\}$, потрібно знайти модель $M = (G, \theta)$, що найкраще відповідає D, де M визначається структурою динамічної баєсової мережі G та відповідним параметром θ із сімейства умовного ймовірнісного розподілу.

За правилом Баєса, апостеріорний розподіл моделі M

$$P(M \mid D) = \frac{P(M)P(D \mid M)}{P(D)},$$

де знаменник $P(D) = \sum P(D \mid M)P(M)$ – фактор нормалізації, що не залежить від M, тому, взявши логарифм, можна обчислити оціночну функцію для M [25]:

$$S(M) = \log P(D \mid M) + \log P(M),$$

де параметр $P(M)$ – апріорний для моделі, а $P(D \mid M)$ – гранична ймовірність для даних D за умови, що модель – це M.

$P(D \mid M) = \int P(D \mid \theta, G)P(\theta \mid G)d\theta$, де $P(\theta \mid G)$ – апріорний розподіл для параметрів. Вибір найоптимальнішої моделі M зводиться до максимізації граничної ймовірності.

Щоб обчислити інтеграл, можна використати розподіл Діріхле для дискретних поліноміальних розподілів і розподіл Вішарта – для неперервних Гаусових розподілів [23].

Навіть маючи функцію оцінки, знаходження оптимальної динамічної баєсової мережі для моделювання генних мереж є дуже складною задачею. По-перше, набір батьківських вершин для кожної вершини становить 2^N, де N – загальна кількість вузлів. Отже, оптимізаційна задача із знаходження моделі з найбільшою функцією оцінки є експоненційною [26].

По-друге, пошуковий алгоритм не завжди знаходить найкращу модель, зазвичай досягається лише локальний максимум, тому єдина обрана модель з максимальною функцію оцінки не завжди є найкращою.

Існує декілька традиційних підходів для вирішення описаної задачі. Один з них – жадібний пошук вгору з випадковими рестартами [27]. При кожному рестарті обирається випадкова модель мережі. Мутація такої базової структури здійснюється додаванням або відніманням одного ребра. Алгоритм визначає усі можливі мутації базової структури та обирає одну з найбільшою оцінкою, після чого вона стає базовою. Ця процедура повторюється, поки не буде досягнутий локальний максимум, потім модель зберігається, і здійснюється рестарт. У результаті отримуємо набір моделей, кількість яких дорівнює кількості рестартів. Даний алгоритм зна-

ходимо у роботі [23], і формулюється він у псевдокоді таким чином:

Greedy Hill-Climbing Search with Restarts for DBN
Input: D (*тренувальні дані часових точок*)
N_{res} (*кількість рестартів*)
Output: M_{out} (*набір моделей з найвищими оцінками*)

for $i = 1$ *to* N_{res} **do**
produce random structure M_0
repeat
$M_{best} \leftarrow M_0$
foreach *pair of nodes in DBN* **do**
if *edge* = 0 (*відсутній зв'язок між двома вершинами*)
then
$M' \leftarrow addEdge(M_0)$
else
$M' \leftarrow removeEdge(M_0)$
end
if $score(M') > Score(M_{best})$ **then**
$M_{best} \leftarrow M'$
end
end
until $M_{best} = M_0$ (*досягається локальний максимум*)
return $M_{out} \leftarrow M_{best}$
end.

Інший клас евристичних алгоритмів, що застосовується для вирішення задачі знаходження найкращої моделі, – ланцюг Маркова Монте Карло метод (Markov Chain Monte Carlo (MCMC) method) [28], який використовує багатовимірний складний розподіл. Механізм цього методу – побудова ланцюга Маркова, в якому нова модель $\widetilde{M}$ генерується тільки на основі попередньої M. Врешті-решт отримується ланцюг моделей, що збігається з шуканим розподілом. Достатньою умовою для такого збігу є рівняння балансу для всіх моделей [23]:

$$P(M_i|M_k)P(M_k|D) = P(M_k|M_i)P(M_i|D),$$

де $P(M_i|M_k)$ – перехідна ймовірність від $P(M_k)$ до $P(M_i)$.

Одним із важливих алгоритмів MCMC є алгоритм Метрополіса-Гастінгса, заснований на алгоритмі семплювання – «вибірка з відхиленням» (acceptance-rejection sampling algorithm) [29]. При кожному запуску алгоритм генерує нову модель – кандидат з розподілу $Q(\widetilde{M}, M)$, що є ймовірністю повернення нової моделі $\widetilde{M}$ при даній моделі M. Маючи модель-кандидат $\widetilde{M}$, можна обчислити ймовірність її прийняття

$$\alpha(\widetilde{M}, M) = \min\left\{1, \frac{P(\widetilde{M}|D)Q(M|\widetilde{M})}{P(M|D)Q(\widetilde{M}|M)}\right\},$$

якщо ймовірність задовольняє даним умовам, то ланцюг Маркова обирає поточну модель-кандидат. Для ілюстрації алгоритму у псевдокоді знову звернемося до роботи [23]:

Metropolis-Hastings sampling algorithm for DBN
Input: D (*тренувальні дані часових точок*)
N_{sam} (*кількість «семплів»*)
Output: M_{out} (*ланцюг моделей*)

Produce initial model M_0
for $i = 1$ *to* N_{sam} **do**
sample a new model $\widetilde{M}$ from $Q(\widetilde{M}, M)$
compute
$\alpha(\widetilde{M}, M) = \min\left\{1, \frac{P(\widetilde{M}|D)Q(M_i|\widetilde{M})}{P(M_i|D)Q(\widetilde{M}|M_i)}\right\}$,
sample u from $U_{(0,1)}$ (*рівномірний розподіл на* (0,1))
if $\alpha(\widetilde{M}, M_i) > u$ **then**
$M_i + 1 \leftarrow \widetilde{M}$
else
$M_i + 1 \leftarrow M_i$
end
return $M_{out} \leftarrow M_{i+1}$
end.

Застосування MCMC до знаходження оптимальної баєсової мережі є також обчислювально затратною задачею. Із збільшенням вузлів у мережі складність алгоритму зростає експоненційно [30]. При порівнянні MCMC із жадібним пошуком перший в цілому показує кращі результати та є швидшим [23].

Прикладом програмної реалізації баєсових мереж є програмне застосування і платформа для вивчення структури статичних та динамічних баєсових мереж Banjo (http://www.cs.duke.edu/~amink/software/banjo/). Автори використали жадібний пошук вгору з випадковими рестартами, імітацію відпалу та генетичні алгоритми, які показали приблизно одна-

кові результати під час тестування за умови тривалого виконання. Проте кожен метод потребує різної кількості часу для знаходження найкращої мережі (20 генів на 2000 точок): на Dell PC, 2.26 GHz CPU, 1 GB RAM жадібний пошук був найшвидшим (хвилини), імітація відпалу на другому місці (десятки хвилин), генетичний алгоритм виявився найповільнішим (години) [31].

Баєсові мережі можна ефективно поєднувати з іншими методами. Так, у [32] жадібний пошук вгору, взятий з реалізації Banjo, використано разом з методом LASSO (least absolute shrinkage and selection operator) і селектором Данціга із сімейства регресивних методів.

Висновки. Аналізуючи два різних підходи до вирішення проблеми реконструкції генних мереж, стає очевидним, що зі збільшенням кількості генів у мережі ми постаємо перед експоненційно складною обчислювальною задачею.

У разі булевих мереж бінаризація значення експресії генів (активний чи пригнічений) надає змогу розглядати більші мережі, досліджуючи їхні загальні властивості. Крім того, ми маємо змогу обмежувати кількість регуляторів окремого гена, тим самим спрощуючи алгоритм для виграшу в часі. Однак згадані спрощення, звичайно, впливають на якість мережі.

У випадку ж баєсових мереж наведені алгоритми демонструють, що існує великий ризик потрапити в локальний максимум, оскільки через експоненційну складність доводиться використовувати евристичні алгоритми.

На думку автора, якіснішого результату в обох випадках можна досягти, якщо розподілити обчислювальне навантаження, використавши кластер комп'ютерів. Для цього не обов'язково застосовувати паралелізм за алгоритмом, простіший спосіб – це розподіл даних між вузлами кластеру. Крім того, доцільним є використання ансамбль-методів, тобто поєднань декількох підходів, що, безперечно, підвищить точність реконструкції.

Отже, можна зробити висновок стосовно того, що алгоритми реконструкції генних мереж на основі булевих і баєсових мереж потребують не тільки детального математичного підґрунтя, яке б давало змогу реконструювати модель мережі, що найкраще відповідає експериментальним даним, але й сучасних обчислювальних підходів у галузі інформаційних технологій, оскільки розглянуті методи не позбавляють нас від проблеми експоненційного пошуку.

Автор висловлює подяку д-ру біол. наук, проф. М. Ю. Оболенській та канд. біол. наук Б. Т. Токовенку (Інститут молекулярної біології і генетики НАН України) за цінні поради і слушні зауваження.

A. O. Frolova

Overview of methods of reverse engineering of gene regulatory networks: Boolean and Bayesian networks

Institute of Molecular Biology and Genetics, NAS of Ukraine
150, Akademika Zabolotnoho Str., Kyiv, Ukraine, 03680

Summary

Reverse engineering of gene regulatory networks is an intensively studied topic in Systems Biology as it reconstructs regulatory interactions between all genes in the genome in the most complete form. The extreme computational complexity of this problem and lack of thorough reviews on reconstruction methods of gene regulatory network is a significant obstacle to further development of this area. In this article the two most common methods for modeling gene regulatory networks are surveyed: Boolean and Bayesian networks. The mathematical description of each method is given, as well as several algorithmic approaches to modeling gene networks using these methods; the complexity of algorithms and the problems that arise during its implementation are also noted.

Keywords: reconstruction of gene regulatory networks, Boolean networks, Bayesian networks.

А. О. Фролова

Обзор методов моделирования сетей генной регуляции: булевы и баесовы сети

Резюме

Одна из проблем современной системной биологии – моделирование сетей генной регуляции, в наиболее полной мере отображающих регуляторные взаимодействия между генами всего организма. Большая вычислительная сложность такой задачи и отсутствие основательных обзоров методов реконструкции генных сетей являются значительной преградой для дальнейшего развития этого направления системной биологии. В данной статье рассмотрены два наиболее распространенных метода моделирования сетей генной регуляции: булевые и баесовые сети, а также дано их математическое описание, а также раскрыто несколько алгоритмических подходов к моделированию генных сетей с помощью этих методов, указаны сложность алгоритмов и проблемы, которые возникают при их использовании.

Ключевые слова: реконструкция сетей генной регуляции, булевые сети, баесовые сети.

REFERENCES

1. *Lee W.-P., Tzou W.-S.* Computational methods for discovering gene networks from expression data // Brief. Bioinform.–2009.–**10**, N 4.–P. 408–423.

2. *Hecker M., Lambeck S., Toepfer S., van Someren E., Guthke R*. Gene regulatory network inference: data integration in dynamic models – a review // Biosystems.–2009.–**96**, N 1.–P. 86–103.
3. *Karlebach G., Shamir R*. Modelling and analysis of gene regulatory networks // Nat. Rev. Mol. Cell Biol.–2008.–**9**.–P. 770–780.
4. *Ott S., Imoto S., Miyano S*. Finding optimal models for small gene networks // Pac. Symp. Biocomp.–2004.–**9**.–P. 557–567.
5. *Margolin A. A., Nemenman I., Basso K., Wiggins C., Stolovitzky G., Favera R. D., Califano A*. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context // BMC Bioinformatics.–2006.–**7**, Suppl. 1.–S 7.
6. *Klipp E*. Systems biology in practice: concepts, implementation and application.–New York: Wiley-VCH, 2005.–465 p.
7. *Kauffman S*. Antichaos and adaptation // Sci. Am.–1991.–**265**, N 2.–P. 78–84.
8. *Kauffman S*. The Origins of Order.–Oxford: Univ. press, 1993.–709 p.
9. *Kauffman S*. Investigations.–Oxford: Univ. press, 2002.–308 p.
10. *Akutsu T., Miyano S., Kuhara S*. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model // Pac. Symp. Biocomp.–1999.–**4**.–P. 17–28.
11. *Martin S., Zhang Z., Martino A., Faulon J*. L. Boolean dynamics of genetic regulatory networks inferred from microarray time series data // Bioinformatics.–2007.–**23**, N 7.–P. 866.
12. *Higa C., Louzada V., Andrade T., Hashimoto R*. Constraint-based analysis of gene interactions using restricted boolean networks and time-series data // BMC Proceedings.–2011.–**5**, Suppl. 2.–S 5.
13. *Lau K., Ganguli S., Tang C*. Function constrains network architecture and dynamics: a case study on the yeast cell cycle Boolean network // Phys. Rev. E. Stat. Nonlin. Soft Matter Phys.–2006.–**75**, N 5, pt 1.–051907.
14. *Xia Q., Liu L., Ye W., Hu G*. Inference of gene regulatory networks with the strong-inhibition Boolean model // New J. Phys.–2011.–**13**, N 8.–083002.
15. *Tsang E. P. K*. Foundations of constraint satisfaction.–London; San Diego: Acad. press, 1993.–405 p.
16. *Lee W.-P., Tzou W.-S*. Computational methods for discovering gene networks from expression data // Brief Bioinform.–2009.–**10**, N 4.–P. 408–423.
17. *Liang S., Fuhrman S., Somogyi R*. Reveal, a general reverse engineering algorithm for inference of genetic network architectures // Pac. Symp. Biocomp.–1998.–**3**.–P. 22.
18. *Zola J., Aluru M., Aluru S*. Parallel information theory based construction of gene regulatory networks // Hipc.–2008.–**5374**.–P. 336–349.
19. *Margolin A. A., Nemenman I., Basso K., Wiggins C., Stolovitzky G., Favera R. D., Califano A*. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context // BMC Bioinformatics.–2006.–**7**, suppl. 1.–S 7.
20. *Zola J., Aluru M., Sarje A., Aluru S*. Parallel information-theory-based construction of genome-wide gene regulatory networks // IEEE Transactions on Parallel and Distributed Systems.–2010.–**21**, N 12.–P. 1721–1733.
21. *Daub C. O., Steuer R., Selbig J., Kloska S*. Estimating mutual information using B-spline functions-an improved similarity measure for analyzing gene expression data // BMC Bioinformatics.–2004.–**5**–P. 118.
22. *Friedman N., Linial M., Nachman I., Pe'er D*. Using Bayesian networks to analyze expression data // J. Comp. Biol.–2000.–**7**, N 3–4.–P. 601–620.
23. *Wu H., Liu X*. Dynamic bayesian networks modeling for inferring genetic regulatory networks by search strategy: Comparison between greedy hill climbing and mcmc methods // Proc. World Acad. Sci., Engin. Technol.–2008.–**34**.–P. 224–234.
24. *Sima C., Hua J., S. Jung S*. Inference of gene regulatory networks using time-series data: A survey // Curr. Genomics.–2009.–**10**, N 6.–P. 416–429.
25. *Yu J., Smith V. A., Wang P. P., Hartemink A. J., Jarvis E. D*. Using Bayesian network inference algorithms to recover molecular genetic regulatory networks // 3rd Int. Conf. Syst. Biol. (ICSB02).–Stockholm, 2002.
26. *Chickering D., Heckerman D., Meek C*. Large-sample learning of Bayesian networks is NP-hard // J. Mach. Learn. Res.–2004.–**5**.–P. 1287–1330.
27. *De Campos L., Fernandez-Luna J., Puerta J*. An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests // Int. J. Intellig. Syst.–2003.–**18**, N 2.–P. 221–235.
28. *Scollnik D*. An introduction to Markov Chain Monte Carlo methods and their actuarial applications // Proc. Casualty Actuarial Soc.–1996.–**83**.–P . 114–165.
29. *Chib S., Greenberg E*. Understanding the Metropolis-Hastings algorithm // Am. Statistic.–1995.–**49**, N 4.–P. 327–335.
30. *Friedman N., Koller D*. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks // Machine Learning.–2003.–**50**, N 1.–P. 95–125.
31. *Yu J., Smith V., Wang P., Hartemink,A., Jarvis E*. Advances to Bayesian network inference for generating causal networks from observational biological data // Bioinformatics.–2004.–**20**, N 18.–P. 3594–3603.
32. *Vignes M., Vandel J., Allouche D., Ramadan-Alban N., Cierco-Ayrolles C., Schiexet T., Mangin B., de Givry B*. Gene regulatory network reconstruction using Bayesian networks, the dantzig selector, the lasso and their meta-analysis // PLoS ONE.–2011.–**6**, N 12.–e29165.

Received 11.11.11