3. Яковлева Г.В., Стехина А.А. Основные подходы к оценке свойств нанообъектов. // Методологические проблемы изучения и оценки био- и нанотехнологий (нановолны, частицы, структуры, процессы, биообъекты) в экологии человека и гигиене окружающей среды». М., 2007. – С.64-67.
4. de Barros A.B., Tsourkas A., Saboury B., Cardoso V.N., Alavi A. Emerging role of radiolabeled nanoparticles as an effective diagnostic technique.// EJNMMI Res. - 2012 Vol.2, №1. – P.39-54.
5. Grabtchak S., Callaghan K.B., Whelan W.M.Tagging photons with gold nanoparticles as localized absorbers in optical measurements in turbid media. //Biomed. Opt. Express. -2013.- Vol.4, №12. – P.2989-3006.
6. Фатхутдинова Л.М., Халиуллин Т.О., Залялов Р.Р. Токсичность искусственных наночастиц // Казанский медицинский журнал. – 2009. - Том 90. - № 4. – P.578-584.
7. Каплун А.П., Безруков Д.А., Швец В.И., Кубатиев А.А. Наноразмерные лекарственные формы: взаимосвязь структура-свойства // Нанотехнологии и охрана здоровья. – 2011. - №2 (7). – С.10-15.
8. Липатова В.А., Ботчей В.М., Писцова Т.В., Федосеев В.А., Клочкова А.В., Беляков В.К., Сухенко Е.П., Суслов В.Б., Василенко И.А., Румянцев С.А., Эттингер А.П., Куцев С.И. Ультраструктурные признаки кардиотоксичности наночастиц золота в эксперименте in vivo. // Вестник РГМУ. 2013. №3. – С.59-63.
9. Hutter E., Maysynger D. Gold Nanoparticles and Quantum Dots for Bioimaging. // Microscopy research and technique. - 2011.- Vol.74, №7 – P.592-604.
10. Сухенко Е.П., Лисицын Н.Б., Ночка А.Д., Соловьев А.А., Кудрявцева Н.Б., Шикалова О.А. Способ определения биологической активности веществ, содержащихся в жидких средах (в том числе наночастиц) // Патент РФ № 2426794 от 30.03.2010
11. Интерфазное ядро как биосенсорная система в клиническом мониторинге / Беляков В.К., Билева Дж.С., Василенко И.А. [и др.] – М.: Форте принт, 2014. 252 с.

УДК 519.23

1.4. ЭФФЕКТИВНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ СРЕДНЕГО ДЕРЕВА В ДВОЙНОМ ВЛОЖЕНИИ
EFFECTIVE ALGORITHM FOR CONSTRUCTING A MEDIUM TREE IN DOUBLE NUTRITION

Горбунов К.Ю.
Gorbunov K.Yu.

РГУ им. А.Н. Косыгина (Технологии. Дизайн. Искусство), Москва
The Kosygin State University of Russia

ПОСТАНОВКА ЗАДАЧИ
Деревья и поддеревья

Назовем *корневым бинарным деревом* граф, в котором степень каждой вершины не превышает 3, есть единственная вершина степени 2 (*корень* дерева)

и каждая вершина степени 1 (*лист* дерева) помечена некоторым символом, эти пометки в листьях не повторяются. Множество всех пометок в листьях дерева T обозначаем как $L(T)$. Все вершины, кроме листьев, называем *внутренними*. Будем говорить, что из двух вершин, лежащих на одном пути от корня, ближайшая к корню расположена *выше*, а дальняя от корня – *ниже*; в этом случае первая является *предком* второй. Та же терминология применяется и к рёбрам дерева. Нижний конец ребра e обозначим e_+. Две вершины – концы одного ребра, – назовем, соответственно, *родителем и сыном*; та же терминология применяется к инцидентным им рёбрам. Нам удобно снабдить дерево *корневым ребром*, т.е. ребром, входящим в корень сверху, будем называть верхний конец корневого ребра *суперкорнем*. Если g – вершина в G, назовём *поддеревом* T_g с корнем g дерево, состоящее из g и всех вершин и рёбер ниже g, а также корневого ребра, входящего сверху в g. Поддерево с корневым ребром e обозначим G_e. *Кладой* в дереве назовём множество пометок листьев некоторого его поддерева; будем говорить также: *клада ребра* (корневого ребра этого поддерева). *Ближайшим общим предком* множества M вершин называется самая нижняя вершина, являющаяся предком каждой вершины из M.

Вложение деревьев

Пусть даны два дерева T_1 и T_2, причем каждому листу l из T_1 соответствует лист $s(l)$ из T_2 (разным листьям из T_1 может соответствовать один и тот же лист из T_2). Тогда *каноническим отображением* (иногда называемым вложением "альфа") называется продолжение отображения s на внутренние вершины T_1, где образ каждой вершины v – ближайший общий предок в T_2 образов всех листьев из T_v. Нам удобно рассматривать модифицированное каноническое отображение, отличающееся от обычного тем, что если хотя бы один из сыновей вершины v отображается (обычным отображением) в ту же вершину $s(v)$, что и v, то образ вершины v при модифицированном отображении – ребро, входящее сверху в $s(v)$. Кроме того, определим, что суперкорень отображается в корневое ребро. Далее рассматриваем лишь модифицированное отображение и обозначаем его как s'. Подробное изложение его свойств, а также примеры его наглядного представления можно найти в [1]. Исходя из этого представления, мы будем называть рёбра дерева T_2 *трубами*. Скажем, что ребро e из T_1 *входит сверху в трубу* t, если его начало отображается в начало трубы t или выше, а конец – в t или ниже. Аналогично, ребро e *выходит из трубы* t, если его конец отображается в конец трубы t или ниже, а начало – в t или выше. Отметим очевидное свойство вложения: в любую некорневую трубу t входят сверху ровно те рёбра, клады которых содержатся в кладе A трубы t и являются максимальными по включению с этим свойством. Множество этих рёбер обозначим $E(A)$. Будем выделять в нём подмножество $E'(A)$ тех рёбер, верхний конец которых отображается в верхний конец трубы t.

Дупликацией назовём вершину в T_1 (кроме суперкорня), отображающуюся в трубу, остальные вершины назовём *дивергенциями*. *Потерей* назовём пару, первый член которой – ребро (v,u) в T_1, а второй –вершина w в T_2 такая, что w лежит строго между $s'(v)$ и $s'(u)$. Потерю назовём *неявной*, если одно из поддеревьев с корневой трубой, выходящей из w, не содержит в листьях пометок из

$s(L(T_1))$ (т.е. образов листьев из T_1); в противном случае назовём ее *явной*. *Местом дупликации или дивергенции* считаем её образ, местом потери – вершину w. Отметим: эти понятия проистекают из филогенетики, где считаются основными эволюционными событиями. Каждому из них сопоставляем цену – неотрицательное рациональное число. Цену дивергенции, как правило, полагают равной нулю, цены остальных событий – строго положительными. *Ценой отображения* s' назовём суммарную цену всех событий в нём.

Мы рассматриваем следующую комбинаторно-оптимизационную задачу, поставленную в [2]. Даны два дерева: P – дерево белков и S – дерево видов, а также отображения s_1 из $L(P)$ в некоторое множество M и s_2 – из M в $L(S)$. Требуется построить дерево генов G с листьями, взаимно однозначно помеченными элементами из M так, чтобы сумма цен вложений s'_1 из P в G и s'_2 из G в S была минимальной. В [2] доказано, что эта задача является NP-трудной. Там описан эвристический квадратичный по времени алгоритм её решения при условии инъективности отображения s_1. Мы предлагаем точный кубический по времени алгоритм решения этой задачи при другом условии. А именно, рассмотрим следующее ограничение: G содержит лишь клады, из заранее фиксированного списка B непустых подмножеств множества M, включая само M. Такое ограничение естественно для филогенетики: клады филогенетического дерева как правило являются множествами родственных друг другу видов или генов. Также, мы обобщаем постановку задачи для произвольных цен событий дупликации, явной и неявной потери (в [2] цены не рассматривались).

Обзор литературы

Практическая значимость рассматриваемой задачи поясняется в [2], там же приведены ссылки, связанные с историей задачи. Ограничение на клады использовалось автором в [1], где описан эффективный алгоритм построения супердерева. Понятие канонического вложения (известного в литературе как "отображение альфа") известно ещё с середины девяностых годов. Его первые определения [3–4] носили абстрактно алгебраический характер, как отображения из множества вершин одного дерева в множество вершин другого. В [1,5–6] дано более наглядное описание этого отображения, при котором вершины дерева генов могут отображаться как в вершины дерева видов, так и в его рёбра, называемые трубами.

Известны алгоритмы построения канонического вложения за линейное время [7–8], которые строят его "от листьев к корню". В [5] приводится алгоритм его построения "от корня к листьям", который работает квадратичное время, но даёт более наглядную конструкцию, удобную для дальнейшего обобщения понятия вложения на случай, когда допускаются события горизонтальных переносов и возникновений генов, которые при каноническом вложении не учитываются. Такое обобщение описано в [5–6].

АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

Описание алгоритма

Обозначим через c_{dp} и c_{dm} соответственно цены дупликаций при отображениях s_1 и s_2; c_{lp} и c_{lm} – цены явных потерь, c'_{lp} и c'_{lm} – цены неявных потерь.

Перебираем пары: множество A из B, труба t из S, где клада трубы t содержит образ $s_2(A)$ множества A. Пары перебираются в порядке возрастания $|A|$, а при каждом фиксированном A трубы t перебираем в порядке от листьев к корню. Для каждой пары $<A,t>$ строим дерево $T(A,t)$, оптимальное в следующем смысле: сумма цен вложений в $T(A,t)$ всех поддеревьев из P с корневыми ребрами из $E(A)$ и вложения самого $T(A,t)$ в поддерево S_t минимальна (будем называть эту сумму *ценой сценария* и обозначать как $C(A,t)$). Деревья $T(A,t)$ строим по индукции в порядке перебора пар $<A,t>$. Базис: для листовых t дерево состоит из единственного листа, $C(A,t)=c_{dm}(|A|-1)$.

Опишем шаг индукции. Обозначим t_1 и t_2 – трубы, на которые разветвляется t, K_1 и K_2 – их клады, t' – корневое ребро искомого дерева. Перебираем всевозможные разбиения множества A на непересекающиеся множества A_1 и A_2 из B. Для каждого из них имеет место ровно один из следующих случаев (первый случай зависит только от A).

1) Образ $s_2(A)$ клады A содержится в K_1 или в K_2 (рассмотрим случай K_1, другой случай симметричен). В этом случае t' выходит из t, "сворачивая" в трубу t_1 на развилке в S, т.е. $s'_2(t'_+)<t_+$, $s'_2(t'_+)\le t_1$. Тогда $T(A,t)=T(A,t_1)$, $C(A,t)=C(A,t_1)+l$, где $l=c_{lm}$, если K_2 содержит элемент из $s_2(M)$ и $l=c'_{lm}$ иначе.

2) Множество $s_2(A_1)$ содержится в K_1, множество $s_2(A_2)$ содержится в K_2 (или симметричный случай). В этом случае t' разветвляется на развилке в t_+ на t'_1 и t'_2 (т.е. $s'_2(t'_+)=t_+$), при этом t'_1 входит сверху в t_1, t'_2 входит сверху в t_2. Пусть k – число таких вершин v из P, что одно выходящее из v ребро лежит в $E'(A_1)$, а другое – в $E'(A_2)$ (т.е. $s'_1(v)=t'_+$). Тогда для цены C рассматриваемого варианта и соответствующего дерева T имеем:

$C=C(A_1,t_1)+C(A_2,t_2)+c_{dp}(|E(A_1)|+|E(A_2)|-|E(A)|-k)+c_{lp}(|E(A_1)|+|E(A_2)|-2k)$, если $s_1(L(P))$ пересекается с A_1 и с A_2 и $C=C(A_1,t_1)+C(A_2,t_2)+c'_{lp}|E(A)|$ иначе; $T=T(A_1,t_1)\wedge T(A_2,t_2)$, что означает объединение деревьев $T(A_1,t)$ и $T(A_2,t)$ под общий корень.

3) Не имеют места случаи 1 и 2. Тогда t' разветвляется в t на t'_1 и t'_2 (т.е. $s'_2(t'_+)=t$), где клада трубы t'_1 равна A_1, клада трубы t'_2 равна A_2. Пусть k – число таких вершин v из P, что $s'_1(v)=t'_+$. Тогда

$C=C(A_1,t)+C(A_2,t)+c_{dm}+c_{dp}(|E(A_1)|+|E(A_2)|-|E(A)|-k)+c_{lp}(|E(A_1)|+|E(A_2)|-2k)$,
если $s_1(L(P))$ пересекается с A_1 и с A_2 и $C=C(A_1,t)+C(A_2,t)+c_{dm}+c'_{lp}|E(A)|$ иначе; $T=T(A_1,t)\wedge T(A_2,t)$.

В качестве $T(A,t)$ берётся дерево T с максимальным $C=C(A,t)$. Заметим, что $T(A,t)$ не зависит от t (хотя $C(A,t)$ зависит), так что для его определения достаточно рассмотреть в качестве t ближайший общий предок множества $s_2(A)$. В качестве итогового дерева берётся $T(M,r)$, где r – корневая труба в S.

Точность и время работы алгоритма

Доказательство оптимальности дерева $T(A,t)$ проходит по той же индуктивной схеме, что и его построение. Базис индукции очевиден. Корректность вычисления C в первом случае также ясна. Для доказательства корректности вычисления C в других двух случаях заметим, что в трубы t'_1 и t'_2 входят сверху, соответственно, рёбра из $E(A_1)$ и $E(A_2)$. Между рёбрами из $E(A)$ и рёбрами из

$E(A_1)$ и $E(A_2)$ в дереве P находятся $|E(A_1)|+|E(A_2)|-|E(A)|$ вершин, из которых ровно k не являются дупликациями. Из $|E(A_1)|+|E(A_2)|$ рёбер, входящих сверху в t'_1 и t'_2, ровно $2k$ не порождают потерю в t'_+. Потери в t'_+ (как и в любой фиксированной вершине) либо все явные, либо все неявные, в последнем случае $k=0$. Таким образом, корректность вычисления C обоснована, из чего следует точность алгоритма.

Оценим время работы алгоритма. Перед построением деревьев $T(A,t)$ проводятся следующие «заготовки». Строятся множества $E(A)$ для всех A из B, что требует времени порядка $|B||P|$. Строятся отношения включения и пересечения множеств из B – время порядка $|B|^2|M|$. Строятся отношения включения множеств из B во множества клад дерева S – время порядка $|B||S||M|$. Строятся все разбиения каждого множества A из B на два множества A_1 и A_2 из B – время порядка $|B|^3$ (для каждой тройки множеств A, A_1, A_2 проверяется, что A_1 и A_2 не пересекаются и $|A_1|+|A_2|=|A|$). Для каждого A из B и каждого его разбиения на A_1 и A_2 вычисляется значение k, используемое в алгоритме, т.е. число таких вершин v в P, что одно выходящее из v ребро принадлежит $E(A_1)$, а второе – $E(A_2)$. Число разбиений множества A не превосходит $|B|$, поэтому время вычисления всех k имеет порядок $|B|^2|P|$.

При построении деревьев $T(A,t)$ на каждом индуктивном шаге для каждого множества из B и трубы t перебирается не более $|B|$ вариантов его разбиения, а для каждого варианта вычисление цены сценария требует константного времени, так что общее время построения всех $T(A,t)$ имеет порядок $|B|^2|S|$. Отсюда следует общая оценка времени алгоритма порядка $|B|(|B||M|+|S||M|+|B|^2+|B||P|+|B||S|)$.

Построение набора возможных клад

В качестве «обязательного минимума» в набор B естественно взять образы (относительно s_1) всех клад в P и прообразы (относительно s_2) всех клад в S. Набор, состоящий лишь из этих множеств, назовем *стандартным*. Однако, при нём алгоритм часто выдаёт результаты, далёкие от оптимальных (при безусловной постановке задачи). Тестирование показало, что результаты, близкие к оптимальным, алгоритм выдаёт для следующего *расширенного* (относительно стандартного) набора. Построим каноническое вложение дерева P в S, взяв в качестве отображения s композицию s_1 и s_2. Для каждой трубы из S рассмотрим множество O выходящих из неё рёбер. Каждому его подмножеству O' соответствует объединение $K(O')$ клад этих рёбер. Добавляем в B множество $s_1(K(O'))$ образов этого объединения. Поскольку в реальных примерах множество O, как правило, невелико (паралогов в одном виде не слишком много), использование расширенного набора не сильно увеличивает время работы алгоритма при существенном увеличении качества результата.

Пример работы алгоритма

Продемонстрируем работу алгоритма на примере из [2]. Пусть $P=(((a,f),c),(e,(b,d)))$, $S=(((a,b),c),(f,(d,e)))$, $M=\{a,b,c,d,e,f\}$, s_1 и s_2 отображают каждый элемент в одноимённый, цены всех явных потерь равны 2, неявных – 1,

цены всех дупликаций равны 3. Стандартный набор B: $\{a,f\}$, $\{a,f,c\}$, $\{b,d\}$, $\{e,b,d\}$, $\{a,b\}$, $\{a,b,c\}$, $\{d,e\}$, $\{f,d,e\}$, все одноэлементные множества и M. Построим каноническое вложение P в S, рис. 1.

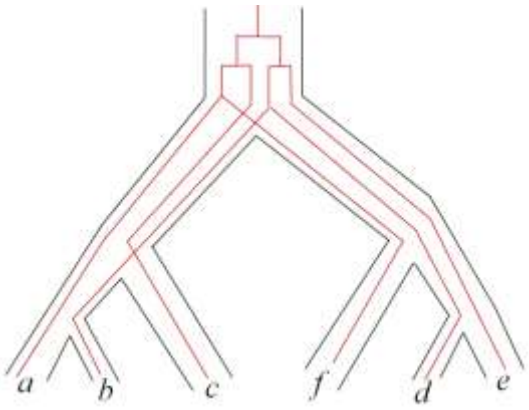


Рис. 1. Каноническое вложение дерева P в дерево S

Корневая труба добавляет в расширенный набор множества $\{a,b,f,d\}$, $\{a,f,e\}$, $\{b,c,d\}$, $\{c,e\}$, $\{a,b,c,f,d\}$, $\{a,c,f,e\}$, $\{a,b,d,e\}$, $\{b,c,d,e\}$. Левая труба добавляет туда множества $\{a,c\}$ и $\{b,c\}$, правая – $\{f,d\}$ и $\{f,e\}$.

Строим деревья для всех множеств расширенного набора. Для одноэлементных и двухэлементных множеств деревья тривиальны. Рассмотрим трёхэлементные множества. Для множества $\{a,f,c\}$ оба дерева $((a,f),c)$ и $((a,c),f)$ имеют одну и ту же цену. Для $\{e,b,d\}$ лучший вариант: $(e,(b,d))$. Для $\{a,b,c\}$ и $\{f,d,e\}$ лучшие варианты $((a,b),c)$ и $(f,(d,e))$. Для $\{a,f,e\}$ и $\{b,c,d\}$ лучшие варианты $(a,(f,e))$ и $((b,c),d)$.

Рассмотрим четырёхэлементные множества. Для множества $\{a,b,f,d\}$ лучший вариант даёт его разбиение на $\{a,b\}$ и $\{f,d\}$. Для $\{a,c,f,e\}$ лучший вариант – разбиение на $\{a,c\}$ и $\{f,e\}$. Для $\{a,c,f,e\}$ лучший вариант – разбиение на $\{a,c\}$ и $\{f,e\}$. Для $\{a,b,d,e\}$ и $\{b,c,d,e\}$ лучшие варианты разбиений, соответственно, на $\{a,b\}$ и $\{d,e\}$ и на $\{b,c\}$ и $\{d,e\}$.

Для пятиэлементного множества $\{a,b,c,f,d\}$ лучший вариант – разбиение на $\{a,b,f,d\}$ и $\{c\}$. Наконец, для всего M получаем лучший вариант разбиения – на $\{a,b,f,d\}$ и $\{c,e\}$. Таким образом, итоговое дерево G: $(((a,b),(f,d)),(c,e))$, а полученное двойное вложение изображено на рис. 2. Оно содержит 2 дупликации и 10 потерь.

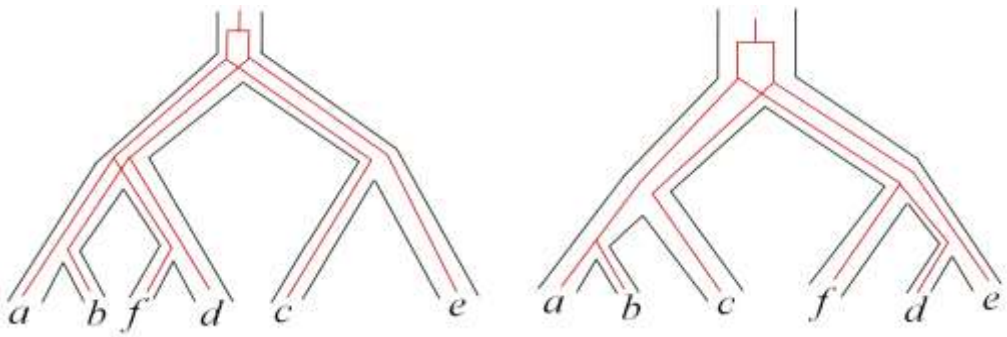


Рис. 2. Двойное вложение $P \to G \to S$, где дерево G построено нашим алгоритмом при расширенном наборе B

Алгоритм из [2] построил на том же примере дерево G=(((*c*,*a*),*f*),(*b*,(*d*,*e*))), соответствующее двойное вложение изображено на рис. 3. В нём три дупликации и 11 потерь.

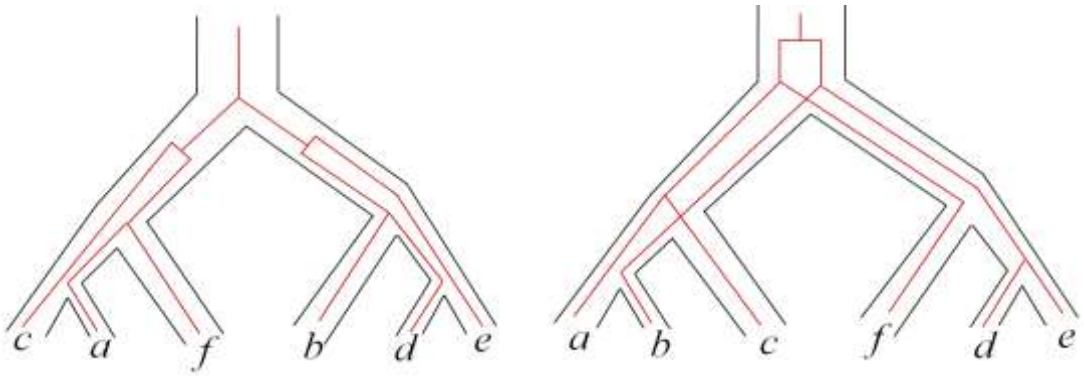


Рис. 3. Двойное вложение $P \to G \to S$, где дерево *G* построено алгоритмом из [2]

Впрочем, если ограничиться стандартным набором *B*, наш алгоритм тоже строит не лучшее дерево. В соответствующем двойном вложении также 3 дупликации и 11 потерь, рис. 4.

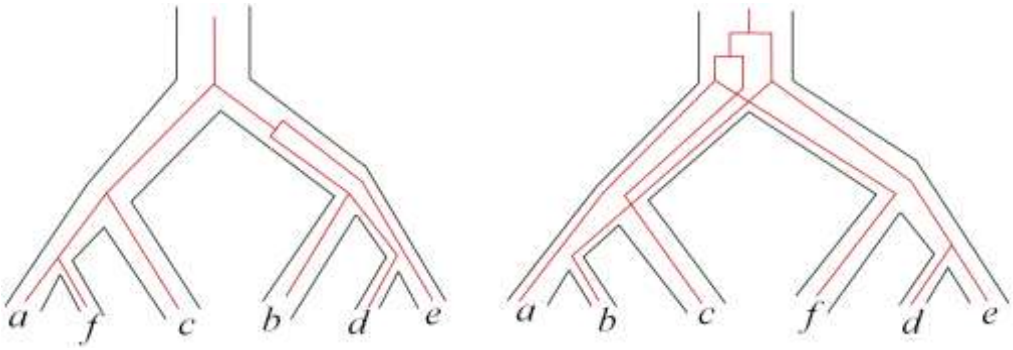


Рис. 4. Двойное вложение $P \to G \to S$, где дерево *G* построено нашим алгоритмом при стандартном наборе *B*

ЗАКЛЮЧЕНИЕ

Мы описали алгоритм построения двойного вложения, учитывающий события дупликаций и потерь. Его легко обобщить для учёта событий горизонтальных переносов при вложении *G* в *S*. Напомним [9, 6], что в постановке этой задачи трубы данного дерева *S* должны быть разбиты на временнЫе слои, что часто приводит появлению вершин лишь с одним сыном. Горизонтальным переносом называется перескок ребра дерева *G* из одной трубы дерева S в другую, лежащую в том же временном слое. Он может быть без сохранении копии в источнике (ребро переносится без развилки) или с сохранением (в трубе происходит развилка, и одно из "сыновьих" рёбер сразу переносится). К описанным в алгоритме трём случаям следует добавить ещё три, описанные в [9, 6]. Это – простой спуск (переход ребра *e* на следующий временной слой, если труба *t* имеет одного сына), разветвление ребра *e* в *t* с переносом одной из «сыновьих» труб в другую трубу того же временного слоя и перенос ребра *e* в другую трубу с последующим событием простого спуска, дупликации, дивергенции или переноса с сохранением (4 подслучая). Кроме того, во всех случаях исклю-

чается требование, чтобы клада трубы t содержала образ клады ребра, которое в ней расположено. Теперь дерево $T(A,t)$ может зависеть от трубы t.

Отметим также возможности уточнения списка B возможных клад. В случае, когда расширенный список слишком большой и замедляет работу алгоритма, можно вместо канонического вложения P в S взять вложение с горизонтальными переносами, где, как правило, паралогов меньше (эффективный алгоритм построения такого вложения описан в [9, 6]). Если же, наоборот, список недостаточно большой (не обеспечивает качественного результата), можно в качестве вложения P в S взять вложение, обеспечивающее минимум дупликационных кластеров (эффективный алгоритм построения такого вложения описан в [10]). Здесь кластером называется множество дупликаций, которые, нестрого говоря, изображаются на одном уровне (строгое определение приведено в [10]); например, на рис. 1 две нижних дупликации образуют кластер. Такое вложение часто отличается от канонического тем, что дупликации происходят «раньше времени», что приводит к росту числа паралогов.

Дальнейшие исследования могут быть направлены на учёт горизонтальных переносов также и при вложении P в G, уточнение способа построения набора B и поиск других вариантов постановки задачи, допускающих эффективный алгоритм решения. Представляет интерес также случай, когда дерево P небинарное, тогда в задачу добавляется поиск его оптимальной бинаризации. В [6] описан некоторый способ бинаризации для задачи построения супердерева, который легко адаптируется для нашей задачи. Он, однако, не гарантирует оптимальной бинаризации.

Благодарность. Работа поддержана грантом РФФИ № 18-29-13037.

Л И Т Е Р А Т У Р А

1. Горбунов К.Ю., Любецкий В.А. Дерево, ближайшее в среднем к данному набору деревьев // Проблемы передачи информации. – 2011. – Т.47(3). – С.274-288.
2. Kuitche E., Lafond M., Ouangraoua A. Reconstructing protein and gene phylogenies using reconciliation and soft-clustering // Journal of Bioinformatics and Computational Biology. – 2017. – V.15(6): 1740007.
3. Page R.D.M. Maps between trees and cladistic analysis of historical associations among genes, organisms and areas // Systematic Biology. – 1994. – V.43. – P.58-77.
4. Guigo R., Muchnik I., Smith T. Reconstruction of ancient molecular phylogeny // Mol. Phylogenet. Evol. – 1996. – V.6(2). – P.189-213.
5. Горбунов К.Ю., Любецкий В.А. Реконструкция эволюции генов вдоль дерева видов // Молекулярная биология. – 2009. – Т.43(5). – С.946-958.
6. Rusin L.Yu., Lyubetskaya E.V., Gorbunov K.Yu., Lyubetsky V.A. Reconciliation of Gene and Species Trees // BioMed Research International. – 2014. – V.2014. – Art.642089.
7. Zhang L. On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies // Journal of Computational Biology. – 1997. – V.4(2). – P.177-187.
8. Ma B., Li L., Zhang L. From gene trees to species trees // SIAM Journal of Computing. – 2000. – V.30(3). – P.729-752.

9. Doyon J.-Ph., Scornavacca C., Gorbunov K.Yu., Szollosi G.J., Ranwez V., Berry V. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers // In: Comparative Genomics, Lecture Notes in Bioinformatics, Springer-Verlag Berlin Heidelberg. – 2010. – V.6398. – P.93-108.
10. Iersel van L., Janssen R., Jones M., Murakami Y., Zeh N. Polynomial-Time Algorithms for Phylogenetic Inference Problems Involving Duplication and Reticulation // IEEE/ACM Transactions on Computational Biology and Bioinformatics. – 2020. – V.17(1).– P.14-26.

УДК 004.4, 620.186.1

1.5. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 3D ВИЗУАЛИЗАЦИИ И ДЕНСИТОМЕТРИЧЕСКОЙ ОЦЕНКИ МИКРО И НАНООБЪЕКТОВ
DEVELOPMENT OF SOFTWARE FOR 3D VISUALIZATION AND DENSITOMETRIC EVALUATION OF MICRO AND NANO OBJECTS

Кожухарь И.И.
Kozhukhar I.I.

РГУ им. А.Н. Косыгина (Технологии. Дизайн. Искусство), Москва
The Kosygin State University of Russia

Актуальность темы определяется повышенным интересом к неинвазивным исследованиям различных фазовых объектов со стороны различных отраслей промышленности, физики, биологии и медицины. Важной задачей при исследовании любых фазовых микрообъектов, будь то прозрачные структуры, плазма или биологические объекты (клетки, бактерии и т.д.), является не только их визуализация, но и получение количественных данных об этих объектах с максимально высокой точностью.

Любой фазовый объект характеризуется пространственным распределением показателя преломления, зная которое, возможно рассчитать ряд других физических параметров, описывающих состояние объекта, таких как плотность, температура, концентрация каких-либо веществ внутри объекта и т.п. Для получения информации о распределении показателя преломления, главным образом, применяется метод оптической интерферометрии.

Современная вычислительная техника и полупроводниковые технологии, расширив возможности существующих оптических интерференционных систем, позволили рассчитывать необходимые физические параметры слабоконтрастных микрообъектов по их фазовым изображениям, в каждой точке которых известна оптическая разность хода. Однако, в большинстве случаев, подобные оптические системы работают в когерентном лазерном свете, что, с одной стороны, позволяет с лёгкостью регистрировать интерферограммы, но с