Дополнительные материалы к статье К. Гобунова, В. Любецкого
«Быстрый алгоритм построения супердерева видов
по набору белковых деревьев»

1. Описание подготовительного этапа к основному алгоритму. При решении задачи Б нам потребуется для каждого дерева генов G_i и всех множеств V из P построить множество $Ed(V,G_i)$. При данном множестве V и дереве генов G_i множество $Ed(V,G_i)$ можно построить с помощью следующего тривиального *подготовительного этапа.*

Подготовительный этап (при заданном наборе P) включает: (1) построение для каждой вершины каждого дерева генов вектора из нулей и единиц длины $|V_0|$, представляющего кладу, заданную этой вершиной; таким образом, каждое множество из P представляется вектором из нулей и единиц длины $|V_0|$; здесь же для каждого множества из P определяется число элементов в нем; (2) построение для каждой пары множеств (т.е. пары векторов) из P пометок, указывающих включено ли первое из них во второе, а также – пересекаются ли они; (3) построение множеств $Ed(V,G_i)$ для всех множеств $V \in P$ и деревьев G_i и определение их мощности, $i = 1, .., n$; (4) для каждого множества из P построение списка всех его разбиений на два множеств из P.

А именно, (1) построение векторов выполняется индукцией от корней деревьев к листьям; напомним: мы считаем, что среднее число листьев в дереве генов имеет порядок $|V_0|$, поэтому общее количество вершин, которые нужно просмотреть, порядка $|V_0|\cdot n$, время обработки одной вершины порядка $|V_0|$; этот шаг требует времени порядка $|V_0|^2\cdot n$. (2) Отношения включения и пересечения всех пар множеств из P строятся за время порядка $|P|^2\cdot|V_0|$. (3) Построение множеств $Ed(V,G_i)$ при данном множестве V для всех деревьев генов G_i выполняется обходом ребер каждого G_i в глубину. Если для очередного ребра e выполняется $M_e \subseteq V$, то зачисляем e в $Ed(V,G_i)$ и переходим к первому (в смысле этого обхода) ребру e_1, для которого e_1 следует после e при этом обходе и e_1 несравнимо с e в смысле порядка $<$. Такой обход требует времени порядка $|V_0|\cdot n$, так как информация о включении $M_e \subseteq V$ уже получена на шаге (2). Тем самым, общее время построения всех множеств $Ed(V,G_i)$ для всех множеств V из P порядка $|P|\cdot|V_0|\cdot n$. (4) Разбиения всех множеств V из P на два множества из P можно построить за время порядка $|P|^3$, что для стандартного набора не превышает $Cn^3\cdot|V_0|^3$. А именно, для каждой тройки множеств V_1, V_2, V из P нужно проверить, что V_1 и V_2 не пересекаются, V_1, V_2 – подмножества V и $|V_1|+|V_2| =$

$|V|$. Для хранения всех разбиений всех множеств из P нужна память порядка $|P|^2$, так как для одного множества имеется не более $|P|$ разбиений.

Таким образом для выполнения подготовительного этапа требуется время порядка $|V_0|^2 \cdot n + |P|^2 \cdot |V_0| + |P| \cdot |V_0| \cdot n + |P|^3$, что для стандартного набора не превышает $Cn^3 \cdot |V_0|^3$. Память, необходимая для хранения всех векторов, отношений и разбиений, множеств $Ed(V,G_i)$ имеет порядок $|P| \cdot |V_0| + |P|^2$, что для стандартного набора не превышает $Cn^2 \cdot |V_0|^2$.

2. Схема алгоритма построения базисных деревьев (основного алгоритма). Подробно алгоритм изложен, как и его обоснование, в [6]. Здесь приведем его общий план.

На подготовительном этапе для каждого дерева генов G_i и всех множеств V из P и каждого множества V из P строится множество $Ed(V,G_i)$. Затем для каждого множества V вычисляются все его разбиения в P. Затем, как указано ниже, методом динамического программирования выделяются базисные множества V из P и для каждого базисного множества V по индукции определяется его *цена* $c(V)$ и минимальное разбиение V на два множества из набора P. Это разбиение для каждого базисного множества V задает дерево $S(V)$, которое мы будем называть *базисным* деревом множества V. Вершины дерева $S(V)$ соответствуют множествам из набора P, корень соответствует множеству V. Множество видов в листьях базисного дерева $S(V)$ совпадает с множеством V и все клады $S(V)$ принадлежат набору P. Если множество V_0 не является базисным, то задача Б, очевидно, не имеет решения. В противном случае дерево $S(V_0)$ является решением этой задачи. Одновременно строится сценарий для всех деревьев генов G_i и дерева видов $S(V_0)$.

Будем называть поддерево T дерева генов G *согласованным* с множеством видов V, если (1) множество видов в листьях T – подмножество в V и (2) никакое над-дерево дерева T в дереве генов G не обладает свойством (1). Говоря неформально, суперкорни всех под-деревьев деревьев G_i, согласованных с V, при искомом вложении будут отображаться в корневую трубу $S(V)$, а $c(V)$ - суммарная цена этих вложений. Одновременно с вычислением цены будут вычисляться дополнительные данные, которые позволят восстановить сами вложения (см. пример ниже).

Итак, опишем индукцию, по которой обрабатываются множества из набора P.

Начальный шаг – одноэлементные множества V из P. В этом случае, по определению, $c(V)=0$ и соответствующее *базисное* дерево $S(V)$ состоит из одного листа.

Пусть V – множество из P, причем все множества меньшего размера уже обработаны. Опишем, в чем состоит обработка пары (V_1,V_2), которая образует разбиение множества V (если ни одной такой пары не найдено, то множество V помечается, как не базисное и его обработка заканчивается). По индукционному предположению, уже известны цены $c(V_1)$, $c(V_2)$ и деревья $S(V_1)$, $S(V_2)$.

В произвольном порядке переберем все деревья G_i. В текущем G_i переберем все вершины и определим число $k(V,V_1,V_2,G_i)$ вершин в G_i, для которых ребро одного сына принадлежит $Ed(V_1,G_i)$, а ребро другого сына принадлежит $Ed(V_2,G_i)$. Положим

$$l(V,V_1,V_2,G_i)= Ed(V_1,G_i)|+|Ed(V_2,G_i)|–2\ k(V,V_1,V_2,G_i)$$

и

$$d(V,V_1,V_2,G_i)=Ed(V_1,G_i)|+|Ed(V_2,G_i)|–|Ed(V,G_i)|–k(V,V_1,V_2,G_i).$$

Напомним: мощности всех множеств $Ed(V,G_i)$ определены на подготовительном этапе.

Найдем разбиение V на V_1^* и V_2^*, для которого величина

$$c(V,V_1,V_2) = \sum_i [c_l \cdot l(V,V_1,V_2,G_i) + c_d \cdot d(V,V_1,V_2,G_i)] + c(V_1) + c(V_2) \qquad (2)$$

достигает минимума по всем разбиениям V на различные базисные множества V_1 и V_2. Тогда *по определению* $c(V)$ – значение величины (2) на *минимальном разбиении* $<V_1^*,V_2^*>$. После этого *базисное* дерево $S(V)$ получается добавлением корня к базисным деревьям $S(V_1^*)$ и $S(V_2^*)$; корень соответствует V, а его сыновья соответствуют V_1^* и V_2^*. Конец описания основного алгоритма. Строгое и довольно сложное обоснование формулы (2) приведено в [6]. Читатель может обратиться к этой работе или рассматривать основной алгоритм как эвристический.

Неформально поясним эту оценку сложности. Для каждого множества из P перебирается не более $|P|$ вариантов его разбиения, а для каждого варианта просматриваются все вершины во всех деревьях генов, что соответствует времени порядка $|P|^2\cdot|V_0|\cdot n$.

Построение отношений включения и пересечения множеств из P требует времени порядка $|P|^2\cdot|V_0|$, построение множеств $Ed(V,G_i)$ для всех V из P – времени порядка $|V_0|^2\cdot n+|P|^2\cdot|V_0|+|P|\cdot|V_0|\cdot n$; построение всех разбиений множеств из P на две части из P – порядка $|P|^3$. Поэтому оценка времени работы основного алгоритма вместе с подготовительным этапом порядка $|V_0|^2\cdot n+|P|^2\cdot|V_0|+|P|\cdot|V_0|\cdot n+|P|^3+|P|^2\cdot|V_0|\cdot n$, что для стандартного набора не превышает $Cn^3\cdot|V_0|^3$. Память, необходимая для хранения всех векторов, отношений и разбиений, множеств $Ed(V,G_i)$ имеет порядок $|P|\cdot|V_0|+|P|^2$, что для стандартного набора не превышает $Cn^2\cdot|V_0|^2$.

В заключение этого пункта приведем интуитивные соображения, которые лежат в основе формулы (2) и формализованы в [6]. Пусть фиксированы вложения f_i каждого G_i в некоторое дерево видов S с множеством листьев V_0 и в S имеется поддерево S' с множеством листьев V. Местом дупликации g из G_i или видообразования g из G_i называется $f_i(g)$, если соответственно $f_i(g)$ труба или вершина дерева видов. Местом потери $<e,s>$ назовем вершину s. Из Леммы 2а следует, что суммарная по всем деревьям генов стоимость $c(V,S')$ дупликаций и потерь, имеющих место в поддереве S', начиная с корневой трубы d поддерева, зависит только от S'. Действительно, $c(V,S')$ определяется множеством $\mathsf{U}_i(\mathrm{Ed}(V,G_i))$ и топологией поддерева S'. Корневая развилка S' задает разбиение множества V на множества V_1 и V_2. Поскольку в алгоритме перебираются все возможные разбиения множества V на два множества из P, то достаточно показать, что формула (2) правильно подсчитывает минимальную стоимость $c(V,S')$ по всем таким S', что все клады в S' принадлежат P и в корневой развилке дерева S' происходит разбиение V на V_1 и V_2. По индукции второе и третье слагаемые уже равны минимальным стоимостям для поддеревьев с множествами листьев V_1 и соответственно V_2, начиная с выходящей из корневой развилки трубы d_1 и соответственно d_2. Остается показать, что первое слагаемое равно суммарной стоимости дупликаций в корневой трубе d и потерь в корневой развилке r поддерева S'. Пусть в d суммарно по всем деревьям генов входят n ребер, а в трубу d_1 – входят n_1 ребер и в трубу d_2 – входят n_2 ребер. По Лемме 2 эти числа равны мощностям множеств $\mathsf{U}_i(\mathrm{Ed}(V,G_i))$, $\mathsf{U}_i(\mathrm{Ed}(V_1,G_i))$, $\mathsf{U}_i(\mathrm{Ed}(V_2,G_i))$. Число ребер в трубе d не может уменьшиться (так как потеря не может происходить в трубе), но их число может увеличиться за счет дупликаций. Все ребра из G_i, дошедшие до конца трубы d, проходят развилку r с разветвлением на два ребра из G_i (видообразование) или с сворачивая в одну из труб d_1 или d_2 с потерей копии на развилке (фактически копия теряется в смежной трубе). Таким образом, $n \le n_1+n_2$ (см. также лемму 2b). Рассмотрим множество M тех вершины деревьев генов, которые отображаются при вложении в d или в r. Это те вершины, которые заключены между ребрами, входящими в d, и ребрами, входящими в d_1 или d_2. Множество M распадается на непересекающиеся части, где каждая часть соответствует своему ребру, входящему в d, и образует дерево с листьями, соответствующими некоторым ребрам, входящим в d_1 или d_2. Число внутренних вершин произвольного дерева равно уменьшенному на 1 числу листьев. Поэтому, в M ровно $k= n_1+n_2-n$ вершин. Каждая из них – дупликация в трубе d или видообразование на развилке r. Поэтому сумма чисел дупликаций и видообразований равна k. Число видообразований найдем просмотром всех вершин во всех деревьях генов и подсчетом числа вершин, у которых одно сыновнее ребро входит в d_1, а другое в d_2: такие вершины – это в точности

видообразования в r согласно описанию минимального вложения в лемме 1. Вычтя из k это число, получим число дупликаций в трубе d. Наконец, число потерь в r равно n_1+n_2-2k, поскольку каждое из n_1+n_2 входящих в d_1 или d_2 ребер составляло видообразование в паре с другим ребром или имело в r потерю. Можно также сказать, что каждое видообразование «охватывает» два ребра: одно входящее в d_1, другое – в d_2, а все «неохваченные» ребра соответствуют потерям. В соответствии с описанием алгоритма первое слагаемое в (2) действительно равно суммарной стоимости дупликаций в трубе d и потерь в корневой развилке r.

3. Пример работы основного алгоритма. Иллюстрируем работу основного алгоритма на искусственном примере, в котором даны десять деревьев генов G_i, показанных на рис. 3. Эти деревья подобраны так, чтобы было легко найти их супердерево S*, которое показано на том же рисунке. Здесь $V_0 = \{a,b,c,d,e\}$. Пусть P – стандартный набор множеств видов, т.е. набор всех клад во всех показанных деревьях генов. Пусть цена потери равна 2, а цена дупликации равна 3. По формуле (2) рекурсивно вычислим цены всех множеств, входящих в набор P. Для одноэлементных множеств все цены по определению равны 0. Вычислим цены всех десяти двухэлементных множеств V, они входят в P. Результат для множества $V = \{x,y\}$, вычисленный по его единственному разбиению на $V_1 = \{x\}$ и $V_2 = \{y\}$, приведен в таблице 1. Поясним ход вычислений. Рассмотрим два случая: текущее дерево G_i не содержит множество V как кладу или это не так. В обоих случаях имеем $|Ed(V_1,G_i)| = |Ed(V_2,G_i)| = 1$, но в первом случае $|Ed(V,G_i)| = 2$, а во втором $|Ed(V,G_i)| = 1$; это проверяется по определению множества $Ed(V,G_i)$ или по шагам алгоритма, приведенного выше для его вычисления. В первом случае в дереве G_i нет вершин, у которых одно сыновнее ребро принадлежит $Ed(V_1,G_i)$, а второе – $Ed(V_2,G_i)$, а во втором случае одна такая вершина имеется. Поэтому $l(V,V_1,V_2,G_i)$ и $d(V,V_1,V_2,G_i)$ в первом случае равны 2 и 0, а во втором случае равны 0 и 0. Эти $l(V,V_1,V_2,G_i)$ и $d(V,V_1,V_2,G_i)$ подставляем в формулу (2). В столбце t таблицы 1 для каждого множества V указано количество деревьев генов, которые не содержат кладу V. Отсюда получаем значение $c(V)$, указанное в этой таблице. В результате видим, что множества $\{a,b\}$ и $\{c,d\}$ имеют минимальную цену, для них $c(V) = 24$.

Таблица 1. В первом и четвертом столбцах после первой строки указаны значения x и y.

$V=\{x,y\}$	t	$c(V)$		$V=\{x,y\}$	t	$c(V)$
a,b	6	24		b,d	8	32
c,d	6	24		a,e	9	36
a,c	8	32		b,e	9	36
a,d	8	32		c,e	9	36
b,c	8	32		d,e	9	36

Разбиение множества V, совпадающее с разбиением в S^*, назовем *стандартным*, а остальные разбиения – *нестандартными*. Конечно, наш алгоритм не использует дерево S^*, а перебирает все разбиения.

После двухэлементных множеств из P нужно рассмотреть все трехэлементные множества из P. Сделаем это на примере множества $V = \{c,d,e\}$. Для него стандартным является разбиение на $V_1 = \{c,d\}$ и $V_2 = \{e\}$. Вычислим на этом разбиении значение $c(V,V_1,V_2)$ функционала (2). Для каждого дерева генов G_i имеет место один из четырех случаев: 1) $\{c,d\}$ клада в G_i, а $\{c,d,e\}$ не клада в G_i, таких деревьев генов два; 2) $\{c,d,e\}$ клада в G_i, а $\{c,d\}$ не клада в G_i, таких деревьев также два; 3) $\{c,d\}$ и $\{c,d,e\}$ не клады в G_i, таких деревьев четыре; 4) $\{c,d\}$ и $\{c,d,e\}$ клады в G_i, таких деревьев два.

В первом и четвертом случаях имеем $|Ed(V_1,G_i)| = |Ed(V_2,G_i)| =1$, а во втором и третьем – $|Ed(V_1,G_i)| =2$, $|Ed(V_2,G_i)| =1$. В первом случае $|Ed(V,G_i)| =2$, во втором и четвертом – $|Ed(V,G_i)| =1$, в третьем $|Ed(V,G_i)| =3$. В первом и третьем случаях в дереве G_i не имеется вершин, у которых одно сыновнее ребро принадлежит $Ed(V_1,G_i)$, а второе – $Ed(V_2,G_i)$), а во втором и четвертом случаях имеется одна такая вершина. Поэтому $l(V,V_1,V_2,G_i)$ и $d(V,V_1,V_2,G_i)$ равны в первом случае 2 и 0, во втором случае 1 и 1, в третьем случае 3 и 0, а в четвертом случае 0 и 0. По формуле (2) для стандартного разбиения получаем: $c(V,V_1,V_2) = 8 + 10 + 24 + 0 + c(\{c,d\}) + c(\{e\}) = 42 + 24 + 0 = 66$, так как по индукции $c(\{c,d\}) = 24$ и $c(\{e\}) = 0$.

Теперь рассмотрим в качестве примера нестандартное разбиение того же множества $V=\{c,d,e\}$ на $V_1 = \{c,e\}$ и $V_2 = \{d\}$, одно из двух симметричных разбиений. Вычислим значение $c(V,V_1,V_2)$. Снова для каждого дерева генов G_i имеет место один из четырех случаев: 1) $\{c,d\}$ клада в G_i, а $\{c,d,e\}$ не клада в G_i, таких деревьев два; 2) $\{c,d,e\}$ клада в G_i, а $\{c,e\}$ не клада в G_i, таких деревьев три; 3) никакое из трех множеств $\{c,d\}$, $\{c,e\}$, $\{d,e\}$ не клада в G_i, таких деревьев четыре; 4) $\{c,e\}$ и $\{c,d,e\}$ клады в G_i, таких деревьев одно.

В первом, втором и третьем случаях имеем $|Ed(V_1,G_i)| = 2$, $|Ed(V_2,G_i)| = 1$, а в четвертом $|Ed(V_1,G_i)| = |Ed(V_2,G_i)| = 1$. В первом случае $|Ed(V,G_i)| = 2$, во втором и четвертом $|Ed(V,G_i)| = 1$, в третьем $|Ed(V,G_i)| = 3$. В первом, втором и четвертом случаях в дереве G_i имеется одна вершина, у которой одно сыновнее ребро принадлежит $Ed(V_1,G_i)$, а второе – $Ed(V_2,G_i)$), а в третьем случае таких вершин нет. Поэтому $l(V,V_1,V_2,G_i)$ и $d(V,V_1,V_2,G_i)$ равны в первом случае 1 и 0, во втором случае 1 и 1, в третьем случае 3 и 0 и в четвертом случае 0 и 0.

По формуле (2) для этого нестандартного разбиения получаем: $c(V,V_1,V_2) = 4 + 15 + 24 + 0 + c(\{c, e\}) + c(\{d\}) = 43 + 36 + 0 = 79$, так как по индукции имели $c(\{c, e\}) = 36$ и $c(\{d\}) = 0$. Второе нестандартное разбиение имеет ту же цену, поэтому заключаем, что разобраны все случаи разбиения V на две части из P и выбираем разбиение с наименьшим значением (равным 66) функционала (2), в данном случае это – стандартное разбиение. Поэтому дерево $S(\{c, d, e\})$ совпадает с поддеревом в S^*.

Аналогично, вычисляется значение $c(V_0,V_1,V_2)$ для множества $V_0 = \{a,b,c,d,e\}$ всех видов и его стандартного разбиения на $V_1 = \{a,b\}$ и $V_2 = \{c,d,e\}$ (оно равно 128) и для его нестандартных разбиений (среди них наименьшее значение равно 143). В итоге алгоритм выдает дерево $S(V_0)$ с ценой 128, которое совпадет с деревом S^*.

4. Вспомогательный алгоритм. Для задачи А2 наш алгоритм предлагает лишь эвристическое решение. А именно, в качестве супердерева он выдает *результат S' применения к набору* $\{S(V)$: V – базисное множество$\}$ описанного чуть ниже вспомогательного алгоритма. Его роль – согласовать набор $S(V)$ в единое дерево S'. Неверно, что в S' все клады принадлежат P, поэтому S' не обязательно является решением задачи Б, также мы не можем доказать, что S' является решением задачи А2. Компьютерный анализ показал, что возможны оба случая $S' = S(V_0)$ и $S' \neq S(V_0)$, во втором случае биологически предпочтительным представляется дерево S'.

Вспомогательный алгоритм: склейка базисных деревьев $S(V)$ в единое дерево S'. Известны разные эвристические алгоритмы, основанные на идее склейки заданного набора деревьев в единое дерево, например, алгоритм из [12] (там же приведены дальнейшие ссылки). В [12] отсутствуют какие-либо оценки сложности алгоритма, но не вызывает сомнения, что он работает экспоненциально долго. Для эффективности алгоритмов склейки важно, чтобы склеиваемые деревья были согласованы между собой, например, в смысле указанном в Теореме 1b, что имеет место для набора базисных деревьев. Итак, вспомогательный алгоритм состоит в

следующем. Определим цену произвольного дерева видов S с множеством V листьев относительно произвольного дерева генов G как цену вложения в S дерева генов G', полученного из G обрезанием всех листьев, соответствующих видам, не принадлежащим V; точнее, обрезаются поддеревья, все листья которых не принадлежат V. *Цену* произвольного дерева видов S определим как сумму цен дерева S относительно всех деревьев из $\{S(V)$: V – базисное множество$\}$.

Начальный шаг в алгоритме склейки базисных деревьев. Перебором найдем дерево с тремя видами, у которого цена минимальная. Это дерево возьмем в качестве исходного S.

Индуктивный шаг в алгоритме склейки базисных деревьев. Перебираем всевозможные пары $<s,e>$, где s – вид, не присутствующий в уже построенном дереве S, а e – ребро в S, включая корневое. Для каждой пары $<s,e>$ присоединяем к S новое ребро, соединяющее середину ребра e с видом s в качестве листа. Вычисляем цену полученного дерева $S(<s,e>)$ и выбираем пару $<s',e'>$, для которой она минимальная. После этого расширяем дерево S до дерева $S(<s',e'>)$, т.е. новое S становится равным $S(<s',e'>)$. И так далее до исчерпания всех видов, в результате получаем дерево S'. Легко видеть, что сложность этого алгоритма кубическая, а, как показало тестирование, в большинстве случаев даже квадратичная. Конец вспомогательного алгоритма склейки базисных деревьев.

Заметим, что могут быть несколько минимальных разбиений. Их число тривиально оценивается сверху величиной $|P|$. Основной алгоритм, описанный выше, выдает результат на каком-то одном минимальном разбиении для каждого множества V из P, поэтому он выдает, вообще говоря, не все базисные деревья $S(V)$. Чтобы получить множество всех базисных деревьев $S(V)$ нужно рассмотреть вариант этого алгоритма, который на каждом шаге индукции перебирает все минимальные разбиения и для каждого из них образует $S(V)$. Такой вариант алгоритма может быть экспоненциальным, так как его тривиальная оценка сложности сверху включает $|P|^{|V0|}$. Однако в нашем тестировании встречалось одно или редко до трех минимальных разбиений, и этот вариант алгоритма работал столько же времени, как и основной алгоритм.

Чтобы охарактеризовать деревья из набора $\{S(V)$: V – базисное множество$\}$, т.е. в сущности дать аксиоматическое определение базисного дерева, нужно расширить задачу Б, заменив в ней функционал (1*) на его обобщение, в котором суммирование по i дополнено суммированием по всем их поддеревьям G', у которых корневыми являются ребра из $Ed(V,G_i)$, а V_0 заменено на V. Получаем функционал

$$C(V,S) = \Sigma_i \Sigma_{G'} [c_l \cdot l(f_{G'},G',S) + c_d \cdot d(f_{G'},G',S)], \qquad (3)$$

где V одно из базисных множеств в P.

Это расширение задачи Б *назовем* задачей В. Если $V = V_0$, то все G' из G_i совпадают с самим G_i и, тем самым, функционал (3) совпадает с функционалом (1), а задача В – с задачей Б. Для любых данных G' и S вместо переменной $f_{G'}$ можно подставить в формулу (3) единственный (по Лемме 1) сценарий $h(G',S)$ для этих G' и S.

Теорема 1А. Пусть P – набор клад.

Для любого базисного множества V из P найденное алгоритмом базисное дерево $S(V)$ – решение задачи В. И наоборот, любое решение задачи В имеет вид $S(V)$, где V – базисное множество из P, при соответствующем выборе последовательности минимальных разбиений.

Доказательство теоремы 1А приведено в работе [6].

5. Компьютерная программа построения супердерева.

Содержится на сайте http://lab6.iitp.ru/ru/super3gl/.

6. Примеры построения супердерева на биологических данных.

Содержится на сайте http://lab6.iitp.ru/ru/super3gl/.