Вычислительная филогенетика

Реконструкция филогении

С.А.Спирин

18 сентября 2019

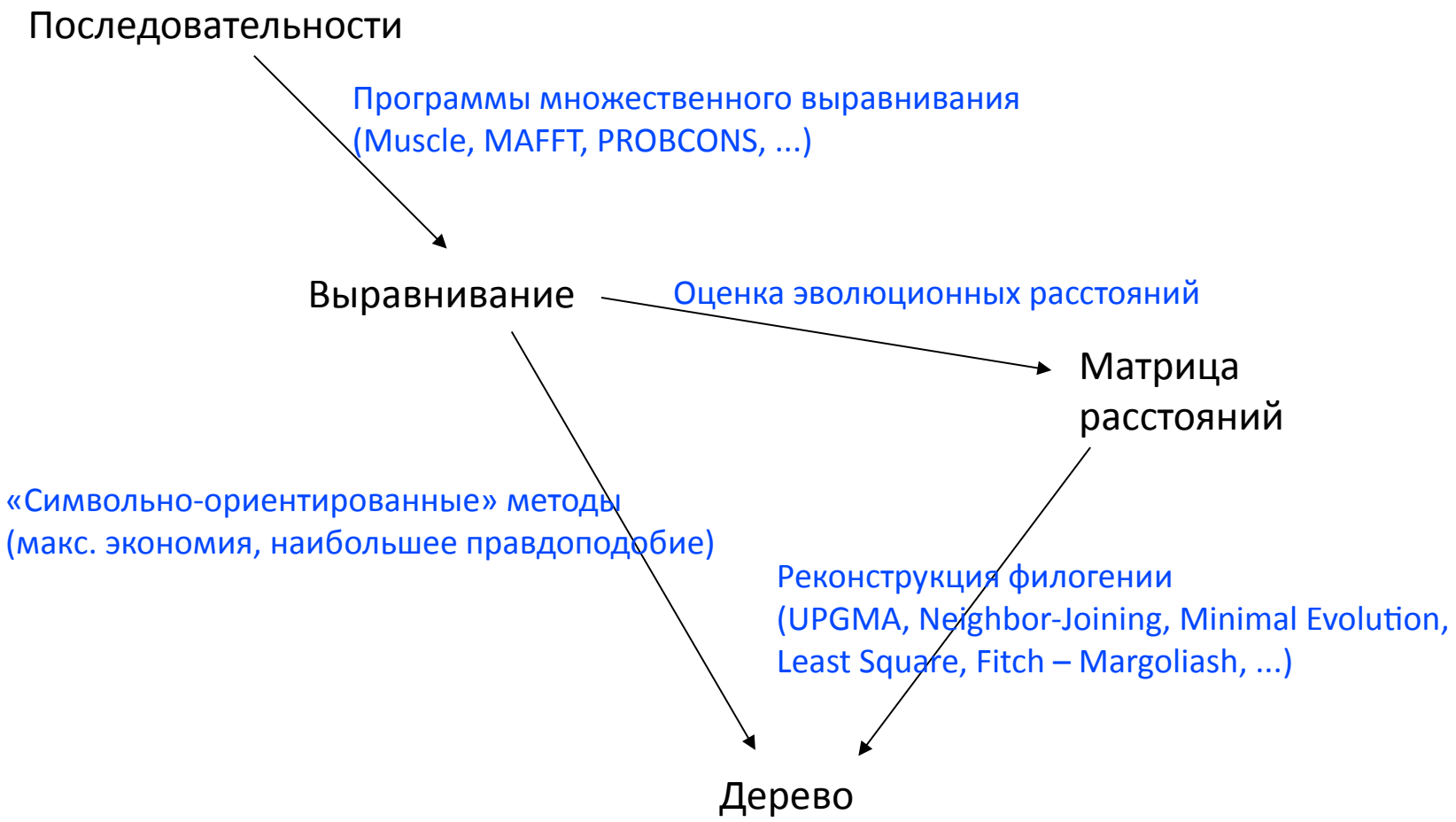
Исходный материал для реконструкции дерева: множественное выравнивание последовательностей белков

```
CYB5_CHICK      MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHPGGEEVLREQA
CYB5_HUMAN      ---MAEQSDEAV--KYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHPGGEEVLREQA
CYB5_HORSE      ---MAEQSDKAV--KYYTLEEIKKHNHSKSTWLILHHKVYDLTKFLEDHPGGEEVLREQA
CYB5_MUSDO      ------MSSEDV--KYFTRAEVAKNNTKDKNWFIIHNNVYDVTAFLNEHPGGEEVLIEQA
CYB5_DROME      ------MSSEET--KTFTRAEVAKHNTNKDTWLLIHNNIYDVTAFLNEHPGGEEVLIEQA
```

Методы реконструкции филогенетических деревьев:

- символьно ориентированные:
  - * максимальной экономии (maximum parsimony)
  - * наибольшего правдоподобия (maximum likelihood)
- использующие матрицу расстояний
  - * кластерные (UPGMA и др.)
  - * объединения соседей (neighbor-joining)
  - * минимальной эволюции (minimum evolution)
  - * наименьших квадратов (least squares)
  - * Фитча – Марголиаша (Fitch – Margoliash)
  - * …

Схема реконструкции филогении по последовательностям



Матрица расстояний
(в формате PHYLIP)

```
   21
CETZ_THEKO  0.000000  2.149903  1.941138  2.043970  2.137562  2.102083
  2.039360  2.020450  2.007926  2.039499  2.041967  2.055799  2.047074
  2.062495  2.066431  2.034956  2.073500  2.066273  2.040881  2.140810
  2.094438
TBB1_HUMAN  2.149903  0.000000  0.428319  0.434481  0.342623  0.379223
  0.342809  0.355026  0.359189  0.378471  0.363068  0.359523  0.365018
  0.360190  0.369711  0.340047  0.314438  0.247306  0.235824  0.322147
  0.292847
TBB_CANAX   1.941138  0.428319  0.000000  0.167005  0.321116  0.365730
  0.214509  0.223333  0.225870  0.221884  0.217145  0.220554  0.224902
  0.227771  0.226902  0.335031  0.290127  0.287522  0.294223  0.352636
  0.324296
TBB_YEAST   2.043970  0.434481  0.167005  0.000000  0.308844  0.350996
  0.215497  0.244587  0.246533  0.258107  0.247217  0.245066  0.249728
  0.258710  0.258576  0.329194  0.317476  0.305437  0.292579  0.348126
  0.334176
TBB_ENCCU   2.137562  0.342623  0.321116  0.308844  0.000000  0.116055
  0.240984  0.272431  0.264568  0.249911  0.217915  0.225991  0.225850
  0.225799  0.233869  0.271022  0.223482  0.210274  0.226772  0.278814
  0.276614
TBB_9MICR   2.102083  0.379223  0.365730  0.350996  0.116055  0.000000
  0.284960  0.310827  0.310250  0.300236  0.285545  0.284959  0.285917
  0.290830  0.296625  0.290482  0.261017  0.261408  0.270484  0.318179
  0.308960
TBB_PNECA   2.039360  0.342809  0.214509  0.215497  0.240984  0.284960
  0.000000  0.139244  0.138274  0.140744  0.127720  0.112638  0.122743
  0.128883  0.131674  0.225152  0.238892  0.169932  0.189354  0.251747
  0.225255
```

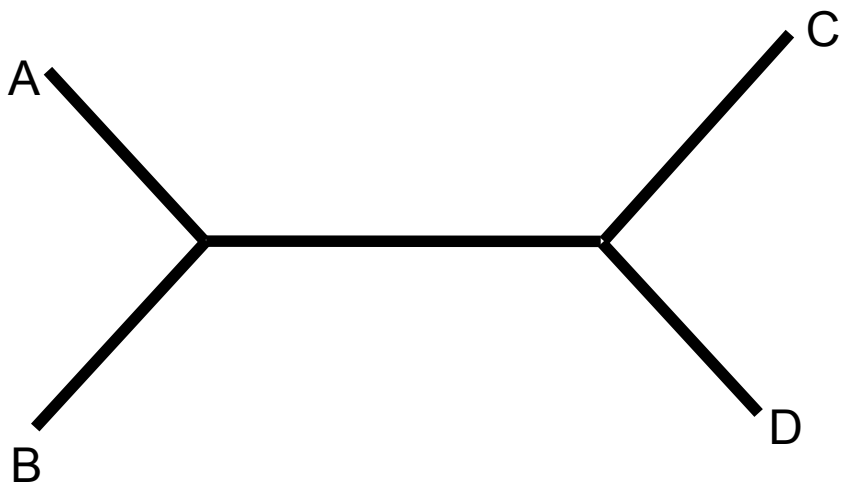
Расстояние как число мутаций

Расстояние между последовательностями ультраметрично, если его понимать как эволюционное время.

Но если неверно предположение о «молекулярных часах», то удобнее понимать расстояние как числа произошедших мутаций. **Такое расстояние не обязательно ультраметрично.**

«Аддитивность» расстояний по дереву

Для расстояний по дереву выполняется свойство, названное **«аддитивность»:** для любых четырёх листьев A,B,C,D из трёх сумм
1) $d(A,B) + d(C,D)$ 2) $d(A,C) + d(B,D)$ 3) $d(A,D) + d(B,C)$
две равны между собой и больше третьей.



Как оценить расстояние между последовательностями

По аддитивному набору расстояний дерево (с длинами ветвей) восстанавливается однозначно!

Но в реальности нам даны последовательности и требуется подсчитать расстояния, то есть оценить число произошедших мутаций.

Это не так просто, поскольку мутации могут происходить в одной и той же позиции.

Расстояния можно оценивать разными способами

На входе практически всегда – множественное выравнивание последовательностей.

Самый простой способ – расстояние равно проценту различных букв.
Наиболее популярный ныне способ основан на принципе наибольшего правдоподобия (используется некоторая вероятностная модель точечных замен).

Как оценить расстояние между последовательностями

Всё же простейшая оценка расстояния есть число различий, делённое на длину последовательности.

Более изощрённые методы учитывают тот факт, что чем больше наблюдаемое различие между последовательностями, тем больше можно ожидать повторных и возвратных мутаций в одинаковых позициях.

Программы Mega и protdist пакета Phylip оценивают расстояния по методу **наибольшего правдоподобия**.

То, что получается, как правило, не обладает свойством аддитивности в точности!

Принцип наибольшего правдоподобия

Оцениваем причины по последствиям.

Принимаем как наиболее обоснованную гипотезу тот вариант причины, при котором вероятность наблюдаемых последствий наибольшая.

В нашем случае «причина» – это эволюционное расстояние, а «последствия» – наблюдаемые замены букв. Эволюционная модель (вероятности замен для всех пар букв) предполагается фиксированной.

(моделей много, наиболее популярная называется JTT по первым буквам фамилий её авторов: Jones, Taylor, Thornton).

Для каждого расстояния (= общего числа мутаций) считаем вероятность получить из первой последовательности вторую. За оценку расстояния принимаем то, при котором эта вероятность максимальна.

Классификация методов

Название метода	Переборный / прямой	Использует молекулярные часы	Символьный/ дистанционный	Реконструирует длины ветвей
UPGMA	Прямой	Да	Дистанционный	Да
Neighbor-Joining	Прямой	Нет	Дистанционный	Да
Наименьших квадратов	Переборный	Может	Дистанционный	Да
Фитча – Марголиаша	Переборный	Может	Дистанционный	Да
Минимальной эволюции	Переборный	Может	Дистанционный	Да
Максимальной экономии	Переборный	Нет	Символьный	Нет
Наибольшего правдоподобия	Переборный	Может	Символьный	Да

Методы, предполагающие молекулярные часы, строят укоренённые ультраметрические деревья.

Методы, не предполагающие молекулярные часы, строят неукоренённые деревья.

Прямые методы

- UPGMA = «Unweighted pair group method with arithmetic mean»
  Строит укоренённое ультраметрическое дерево
  Видимо, реально лучший из методов, предполагающих молекулярные часы.

- Neighbor-Joining
  Строит неукоренённое дерево. Если и уступает некоторым переборным алгоритмам, то не сильно.

Оба метода принимают на вход матрицу расстояний.

UPGMA – схема алгоритма

Укоренённое дерево строится «снизу вверх»

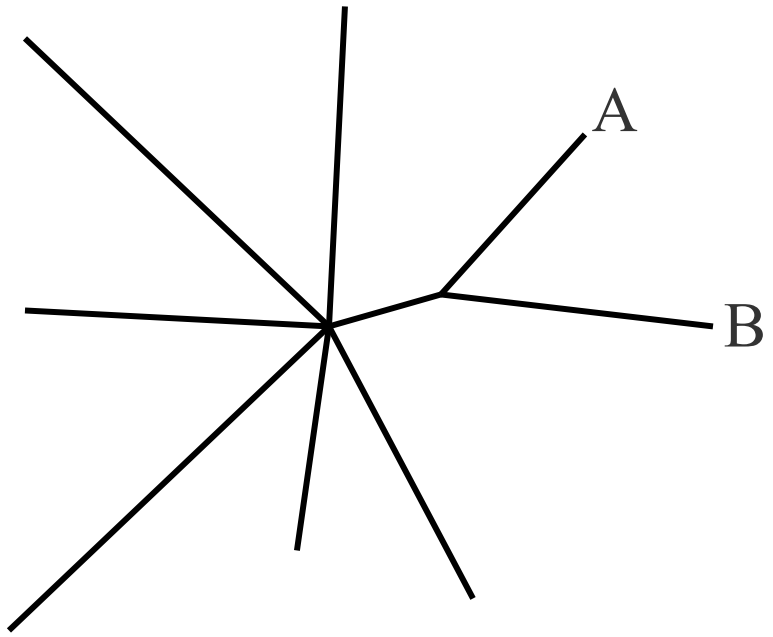
- Найдём в матрице расстояний наименьший элемент.
- Объединим два ближайших листа в кластер (это – узел дерева, соединённый ветвями с листьями, образовавшими его).
- Пересчитаем матрицу расстояний, рассматривая кластер как новый лист. Расстоянием до кластера будем считать **среднее арифметическое** расстояний до его элементов (отсюда название метода).
- Повторяем с начала, пока не останется всего два кластера.

К этому прибавляется способ вычисления длин ветвей. Результат — укоренённое ультраметрическое дерево с длинами ветвей.

В программе Jalview этот метод реализован под названием «Average distance»

Идея алгоритма neighbor-joining

(Saitou and Nei, 1987)



A и B — такая пара последовательностей, для которых минимальна величина

$$(n - 2)\rho(A, B) - M(A, B),$$

где n – число последовательностей, ρ — расстояние из матрицы, а

$$M(A, B) = \sum_C \left(\rho(A, C) + \rho(B, C)\right)$$

Такие «соседи» дальше рассматриваются как один лист. «Объединение соседей» продолжается, пока не останутся только три «листа».

Свойства алгоритма NJ

Время работы: $O(n^3)$
Предложен очень близкий алгоритм с временем работы чуть хуже $O(n^2)$ (J.F. Li, 2015)

Если исходная матрица расстояний состояла из расстояний, вычисленных по некоторому дереву, то это дерево будет восстановлено.

Показывает хорошие результаты на практике.

Переборные методы

Алгоритм, реализующий переборный метод, должен включать:

а) критерий сравнения деревьев (какая из двух топологий лучше соответствует исходным данным?)

б) алгоритм поиска лучшего по критерию дерева.

Пример критерия
(метод наименьших квадратов, OLS — ordinary least squares)

Пусть дана матрица расстояний и топология дерева;
i, j — две последовательности, тогда мы имеем расстояние $d(i,j)$ из матрицы.

Приписав ветвям дерева длину, будем иметь расстояние $d'(i,j)$ «по дереву».

Подберём длины ветвей так, чтобы сумма величин $(d(i,j) - d'(i,j))^2$

(по всем парам листьев i,j) была наименьшей.

Это наименьшее значение и будет критерием качества: будем считать ту топологию лучшей, для которой это значение получится меньшим.

Поиск лучшего дерева

Имеется единственная топология (бинарного и неукоренённого) дерева с тремя листьями, три разных топологии деревьев с четырьмя листьями,
15 топологий деревьев с пятью листьями,
... ...
~ 2 млн. топологий деревьев с десятью листьями,
... ...
~ 8 трлн. топологий деревьев с 15 листьями,
... ...

Триллионы проверок компьютер будет делать слишком долго.
А ведь приходится строить деревья и с сотней листьев…

Поиск лучшего дерева

Имеется единственная топология (бинарного и неукоренённого) дерева с тремя листьями, три разных топологии деревьев с четырьмя листьями,

15 топологий деревьев с пятью листьями,

... ...

~ 2 млн. топологий деревьев с десятью листьями,

... ...

~ 8 трлн. топологий деревьев с 15 листьями,

... ...

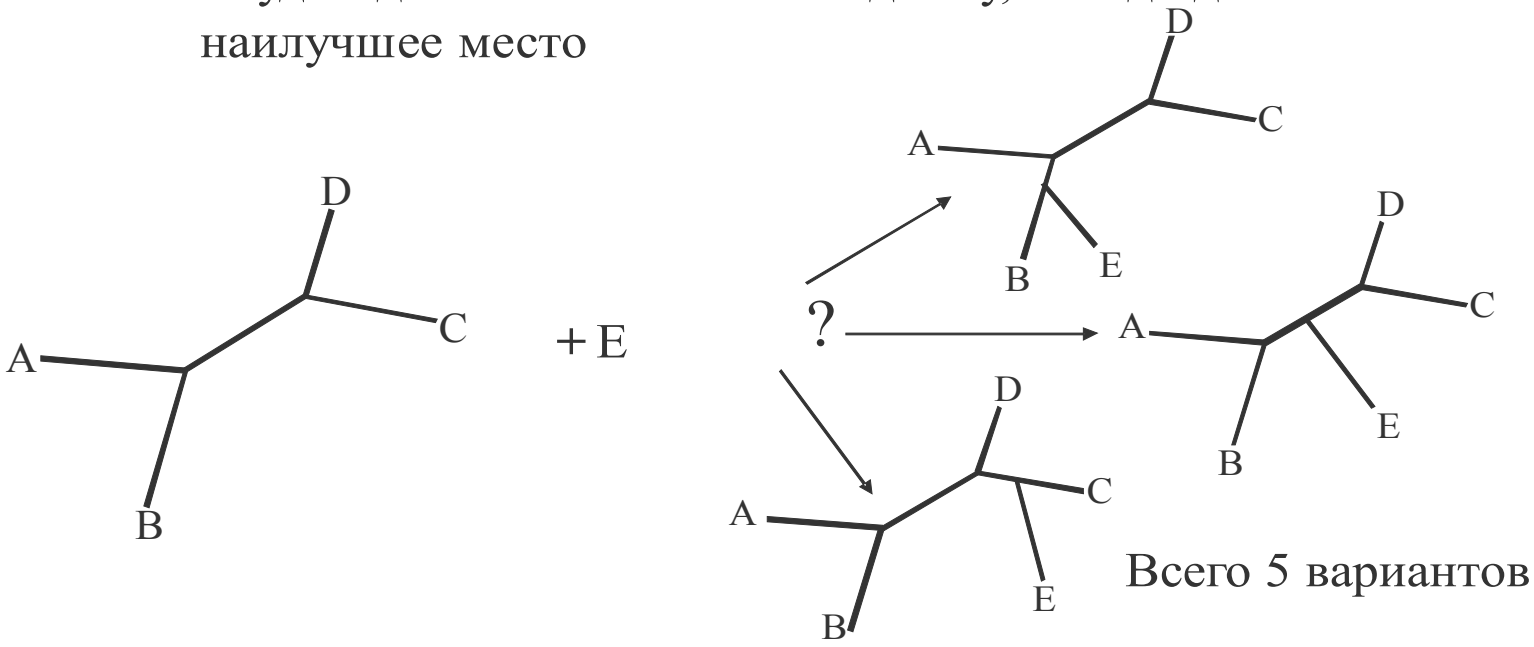
Триллионы проверок компьютер будет делать слишком долго.

А ведь приходится строить деревья и с сотней листьев…

Поэтому программы, реализующие переборные методы, практически никогда не включают **полный** перебор всех возможных деревьев

Поиск лучшего дерева: «выращивание»

- Найдем лучшее дерево для части последовательностей
- Будем добавлять листья по одному, находя для них наилучшее место



Всего 5 вариантов

Поиск лучшего дерева: «выращивание»

Дерево с N листьями всегда имеет $2N–3$ ветви.
Поэтому, чтобы “вырастить” дерево с N листьями, надо проанализировать
$3 + 5 +…+ (2N – 5) = (N – 3)(N – 1)$ деревьев.
Уже для N=10 это число меньше числа всех возможных деревьев в 32175 раз!

Выращивание не гарантирует нахождение “лучшего” дерева, но при хороших данных не должно приводить к большим ошибкам.

Поиск лучшего дерева: просмотр соседних деревьев

Построим сначала «черновое» дерево, а затем попробуем его улучшить.

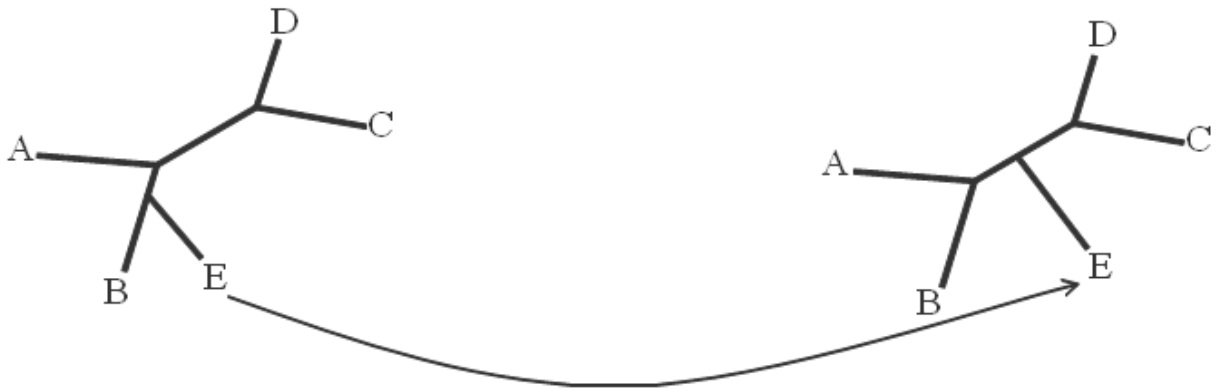
Черновое дерево можно построить одним из эвристических методов или «вырастить».

Улучшать будем, просматривая «соседние» деревья.

Поиск лучшего дерева: просмотр соседних деревьев

Что такое «соседние деревья»?

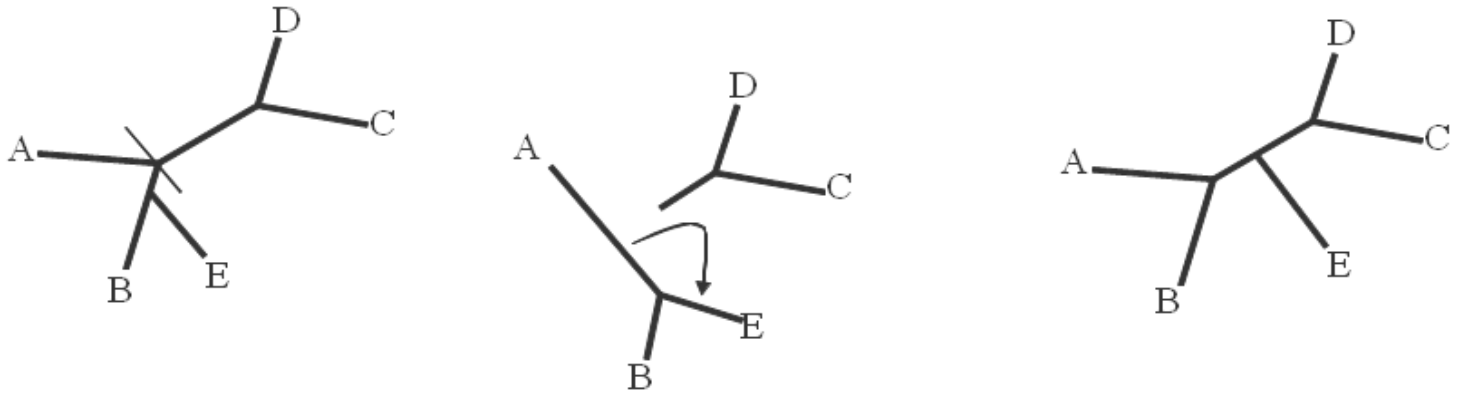
- Оторвём один лист и «привьём» его на другую ветвь



Поиск лучшего дерева: просмотр соседних деревьев

Что такое «соседние деревья»?

- Можно проделать аналогичную операцию с целой кладой

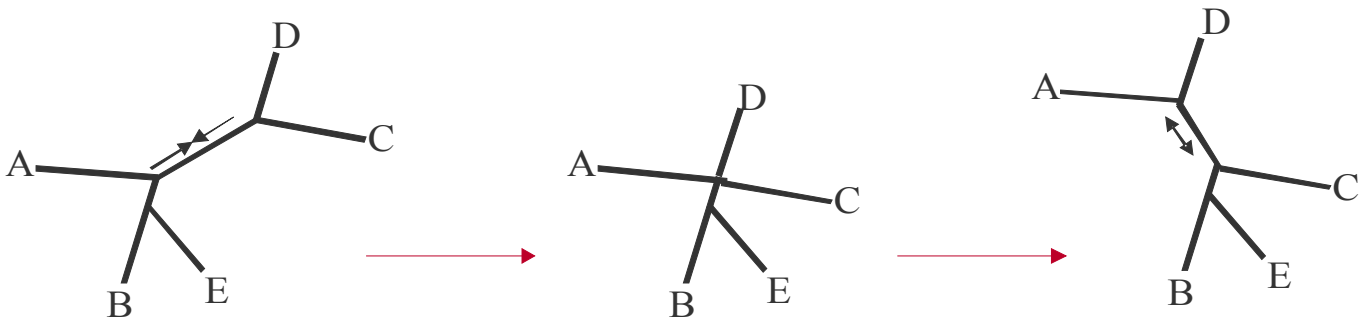


Такая операция обычно называется “SPR” : Subtree Pruning and Regrafting
В пакете PHYLIP она называется “Global rearrangement”.

Поиск лучшего дерева: просмотр соседних деревьев

Что такое «соседние деревья»?

- Можно «схлопнуть» одну ветвь и заменить её другой



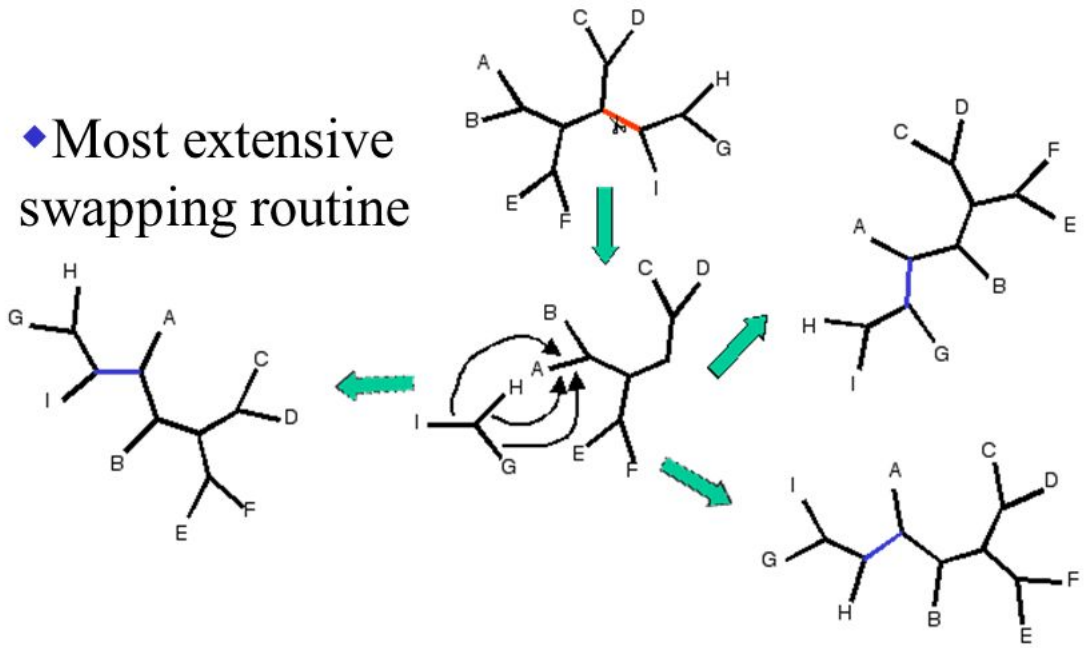
Такая операция обычно называется “NNI” : Nearest Neighbor Interchange.
В пакете PHYLIP она называется “Local rearrangement”.

Поиск лучшего дерева: просмотр соседних деревьев

Что такое «соседние деревья»?

Tree Bisection and Reconnection

Another Branch Swapping Algorithm



https://slideplayer.com/slide/7408141/

Поиск лучшего дерева

- Строим черновое дерево
  - прямым методом *или*
  - выращиванием с использованием того же критерия качества *или*
  - выращиванием с использованием другого критерия (вычисляемого быстрее, например максимальной экономии при основном критерии наибольшего правдоподобия)
- Анализируем соседние деревья (NNI или SPR)
  если находим среди соседей лучшее дерево, берём за основу его
- Повторяем предыдущий пункт, пока текущее дерево не окажется лучше всех своих соседей

Переборные методы

Название метода всегда совпадает с названием критерия качества

- Максимальной экономии
  (или «бережливости», **maximum parsimony**, MP)
- Наибольшего правдоподобия (**maximum likelihood**, ML)
- Наименьших квадратов (**ordinary least squares**, OLS)
- Фитча – Марголиаша (**Fitch – Margoliash**, FM)
- Минимальной эволюции (**minimum evolution**, ME)

Все методы, кроме максимальной экономии, допускают предположение о молекулярных часах (но чаще используются без этого предположения!) и оценивают длины ветвей.

Методы MP и ML — символьно-ориентированные, OLS, FM, ME и многие другие принимают на вход матрицу расстояний.

Исходный материал для реконструкции дерева: множественное выравнивание последовательностей белков

```
CYB5_CHICK      MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHPGGEEVLREQA
CYB5_HUMAN      ---MAEQSDEAV--KYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHPGGEEVLREQA
CYB5_HORSE      ---MAEQSDKAV--KYYTLEEIKKHNHSKSTWLILHHKVYDLTKFLEDHPGGEEVLREQA
CYB5_MUSDO      ------MSSEDV--KYFTRAEVAKNNTKDKNWFIIHNNVYDVTAFLNEHPGGEEVLIEQA
CYB5_DROME      ------MSSEET--KTFTRAEVAKHNTNKDTWLLIHNNIYDVTAFLNEHPGGEEVLIEQA
```

Методы реконструкции филогенетических деревьев:

- символьно ориентированные:
  - * максимальной экономии (maximum parsimony)
  - * наибольшего правдоподобия (maximum likelihood)
- использующие матрицу расстояний
  - * кластерные (UPGMA и др.)
  - * объединения соседей (neighbor-joining)
  - * минимальной эволюции (minimum evolution)
  - * наименьших квадратов (least squares)
  - * Фитча – Марголиаша (Fitch – Margoliash)
  - * …

Программы реконструкции филогении

- **FastME**
метод сбалансированной минимальной эволюции (balanced minimum evolution)
- **RAxML**
методы наибольшего правдоподобия (maximum likelihood) и максимальной экономии (maximum parsimony)
- **PhyML**
другая реализация метода наибольшего правдоподобия
- **MrBayes**
байесова оценка филогении (Bayesian inference using Markov chain Monte Carlo, MCMC)
- **PhyloBayes**
другая популярная реализация байесова подхода

Программы работы с деревьями

- MEGA

https://www.megasoftware.net/
Визуализация и построение деревьев (оконный интерфейс)

- Пакет PHYLIP

http://evolution.genetics.washington.edu/phylip.html
Построение и визуализация деревьев (интерактивный интерфейс с запуском из командной строки).

- FigTree

http://tree.bio.ed.ac.uk/software/figtree/
Визуализация деревьев

См. также
https://en.wikipedia.org/wiki/List_of_phylogenetic_tree_visualization_software

Программы работы с деревьями онлайн

- https://ngphylogeny.fr/

Филогенетическая реконструкция различными методами и визуализация полученных деревьев

- iTol https://itol.embl.de/

Визуализация деревьев

Что важно помнить при реконструкции филогении по последовательностям

1. Последовательности должны быть гомологичны по всей длине.

2. Если последовательности нуклеотидные, надо убедиться, что часть из них не представлена комплементарными вариантами.

3. Последовательности необходимо выровнять.
Кстати, по виду выравнивания можно оценить, действительно ли последовательности такие, как надо: должно быть много консервативных колонок и мало хаотично расположенных гэпов.
Помните: программа выравнивания выдаст результат даже для совершенно неродственных последовательностей, но смысла в этом результате не будет!

4. Большинство программ реконструируют неукоренённое дерево (даже если оно выглядит как укоренённое). Определение положения корня – отдельная задача.

5. Результат реконструкции – не абсолютная истина. Достоверность той или иной ветви можно оценить путём сравнения результатов разных программ и/или бутстрепа.
Как правило, чем короче выравнивание, тем хуже качество реконструкции.