# Филогенетические деревья

(продолжение)

Филогенетические деревья и таксономия организмов

Алгоритмы реконструкции филогении

С.А.Спирин

17 февраля 2023

ФББ МГУ

# Филогенетические деревья и таксономия организмов

Любая ветвь дерева делит множество листьев на два.

Если листья соответствуют ортологичным белкам разных организмов, то одно из получившихся множеств может соответствовать какой-нибудь таксономической группе.

Стандарт таксономии для биоинформатики – банк “NCBI taxonomy database”

**`https://www.ncbi.nlm.nih.gov/taxonomy`**

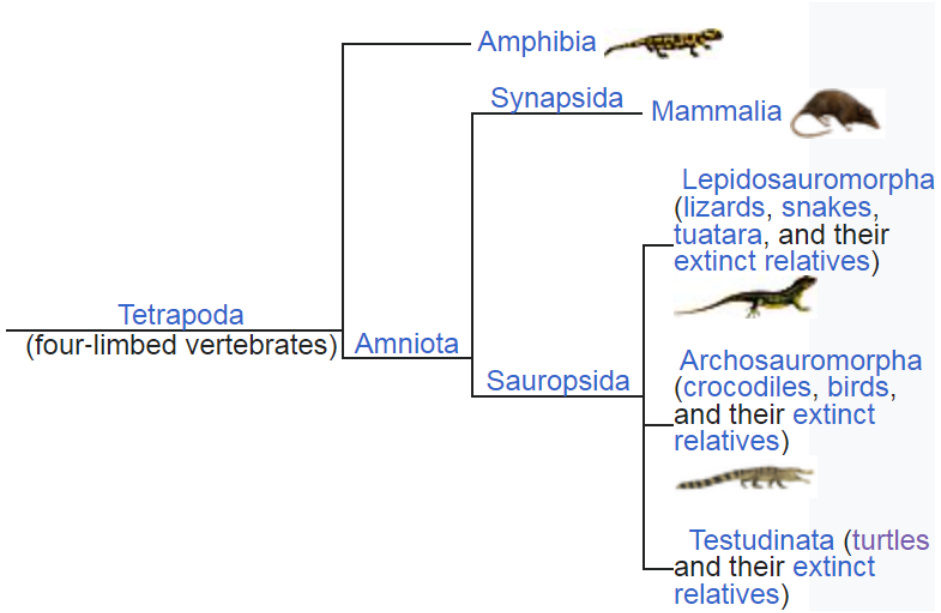
Тамошняя таксономия сознательно приближена к филогении (парафилетические таксоны, такие как Pongidae, старательно вычищаются).

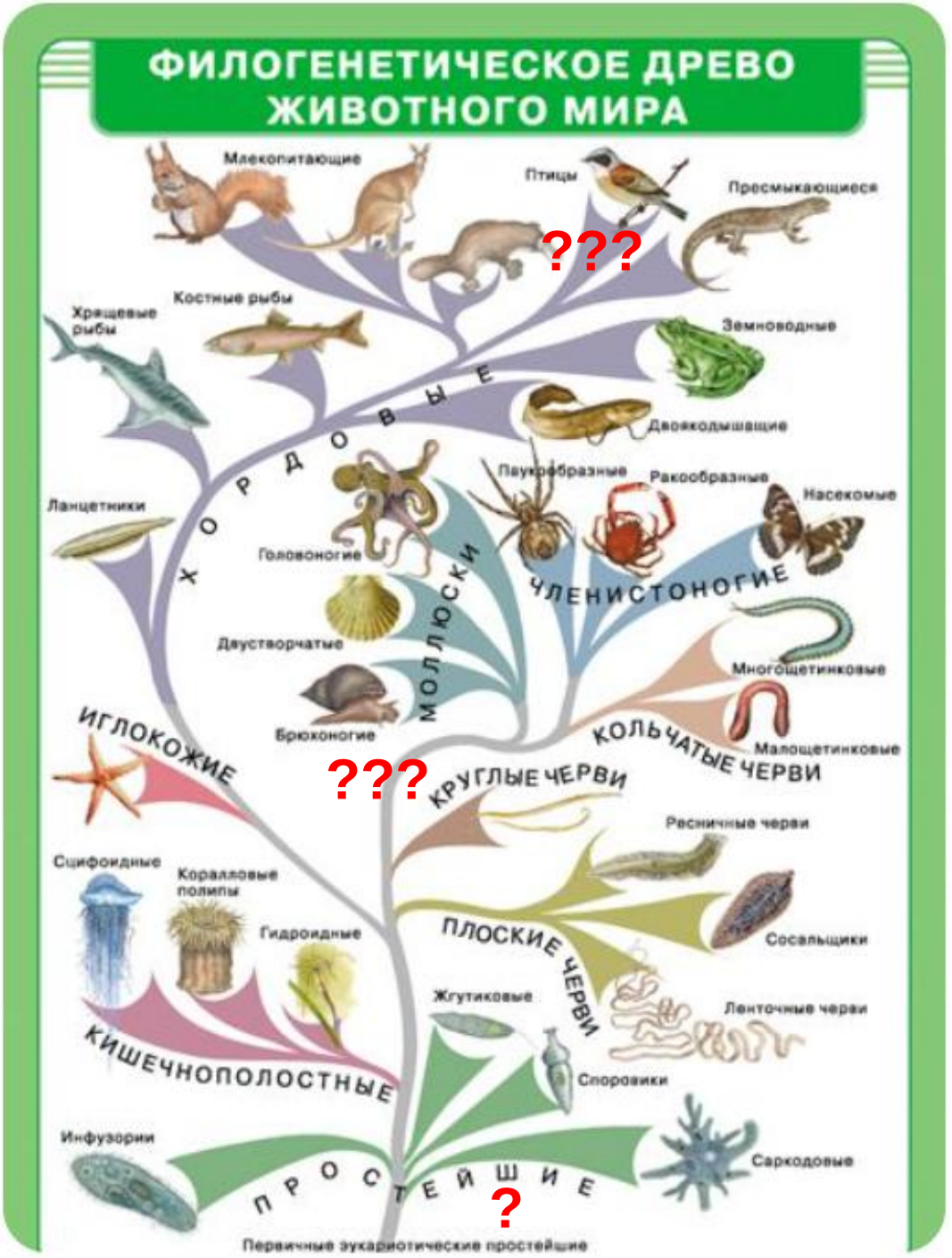
Это вряд ли разумно с точки зрения общей биологии, но удобно, так как позволяет, по сути, свести классификацию к филогении.

# Парафилетические таксоны

**Монофилетическая** группа: включает общего предка и ВСЕХ его потомков. Например: Хордовые, Птицы, Круглые черви

**Парафилетическая** группа: получается изъятием из монофилетической одной или нескольких терминальных групп. Например: Беспозвоночные, Рептилии, Черви, Рыбы, Простейшие

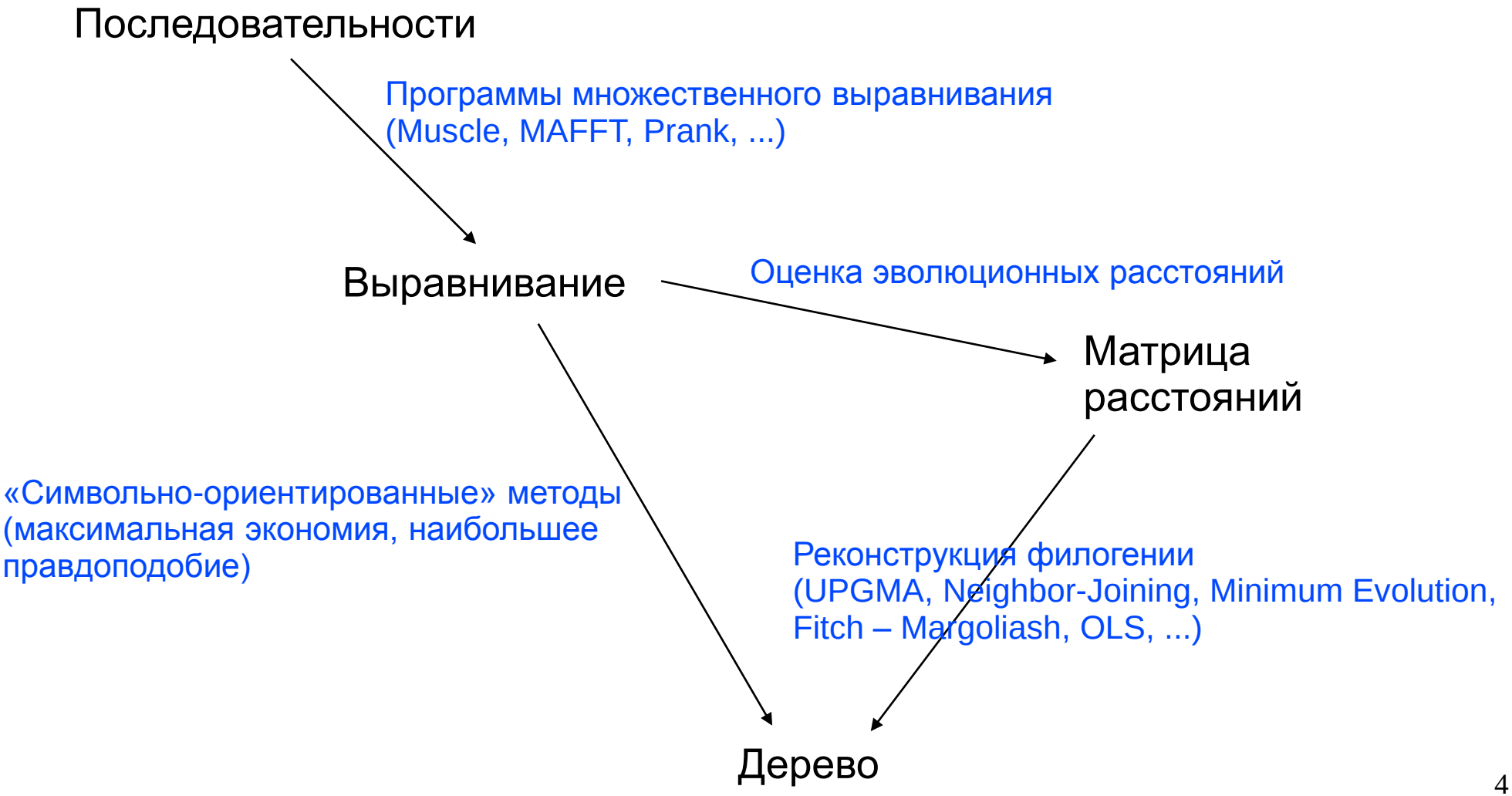





**Источник**: https://ru.pinterest.com/pin/850124867143613166/ и https://en.wikipedia.org/wiki/Chordate

# Схема реконструкции филогении по последовательностям



# Матрица расстояний

| | MUSDO | CHICK | BOVIN | HUMAN |
|---|---|---|---|---|
| MUSDO | 0 | 9.5 | 8.9 | 9.2 |
| CHICK | 9.5 | 0 | 3.4 | 2.8 |
| BOVIN | 8.9 | 3.4 | 0 | 1.7 |
| HUMAN | 9.2 | 2.8 | 1.7 | 0 |

Множество объектов (последовательностей) превращается в **метрическое пространство**

# Аксиомы метрического пространства:

1) $d$ (A,A) = 0

2) $d$ (A,B) > 0, если A $\neq$ B

3) $d$ (A,B) = $d$ (B,A)

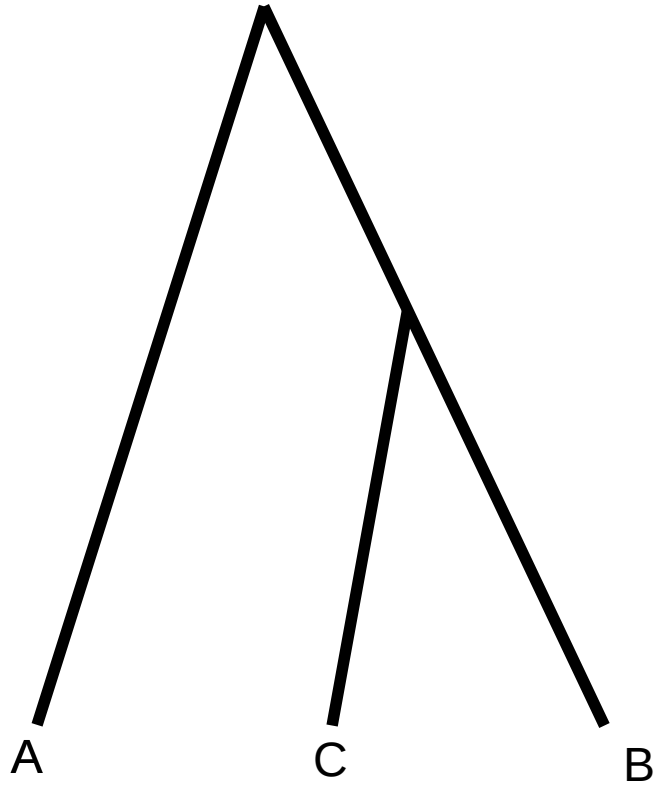
4) $d$ (A,B) $\leq$ $d$ (A,C) + $d$ (B,C)

Если расстояния пропорциональны эволюционному времени, то эти аксиомы выполняются.

Но для таких расстояний верно и нечто большее:

4') $d$ (A,B) $\leq \max(d\,(A,C), d\,(B,C))$
(«ультраметрическое пространство»)

# Ультраметрическое расстояние




Если $d$(A,B) > $d$ (B,C), то $d$ (A,C) = $d$ (A,B)

Или: из трёх расстояний между тремя объектами два всегда равны между собой и не меньше третьего
(это равносильно аксиоме ультраметричности).

# Расстояние как число мутаций

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

Но если неверно предположение о «молекулярных часах», то больше информации несёт понимание расстояния как числа произошедших мутаций.

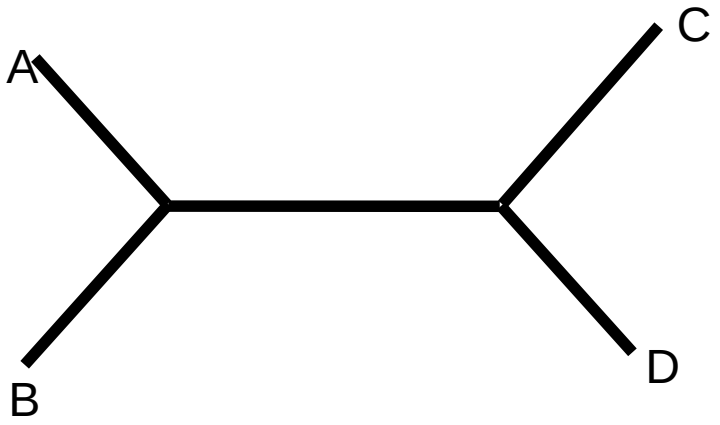
**Такое расстояние не обязательно ультраметрично.**

**Аддитивность:**

если есть четыре последовательности A,B,C,D, то из трёх сумм:

$d(A,B) + d(C,D)$ $d(A,C) + d(B,D)$ $d(A,D) + d(B,C)$

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.

# 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, FastME, 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 популярны модели JTT (по первым буквам фамилий её авторов: Jones, Taylor, Thornton, 1992) и LG (Le, Gascuel, 2008)

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.

# 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

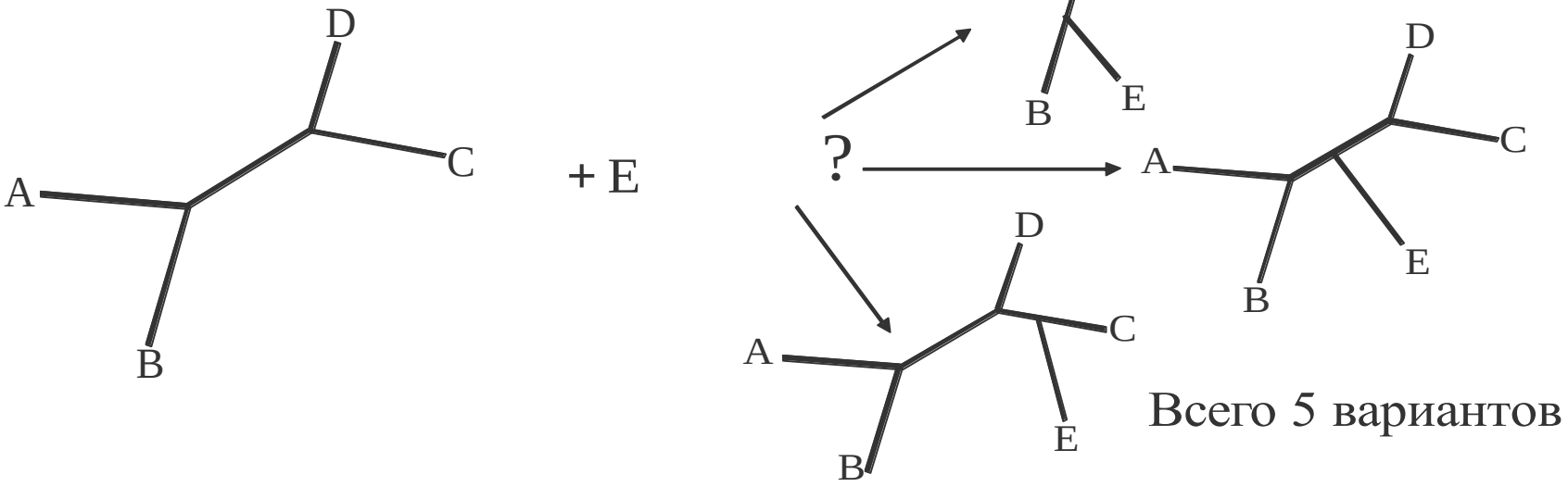
Имеется единственная  топология (бинарного и неукоренённого) дерева с тремя листьями, три разных топологии деревьев с четырьмя листьями,
15 топологий деревьев с пятью листьями,
... ... ... ...
~ 2 млн. топологий деревьев с десятью листьями,
... ... ... ...
~ 8 трлн. топологий деревьев с 15 листьями,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

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.

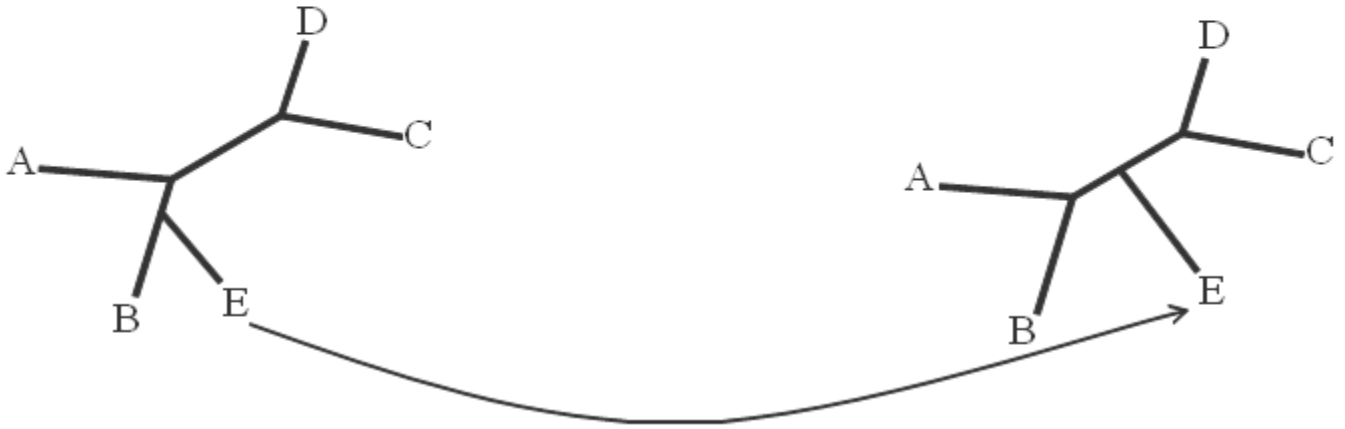
Черновое дерево можно построить одним из эвристических методов или «вырастить».

Улучшать будем, просматривая «соседние» деревья.

# Поиск лучшего дерева: просмотр соседних деревьев

## Что такое «соседние деревья»

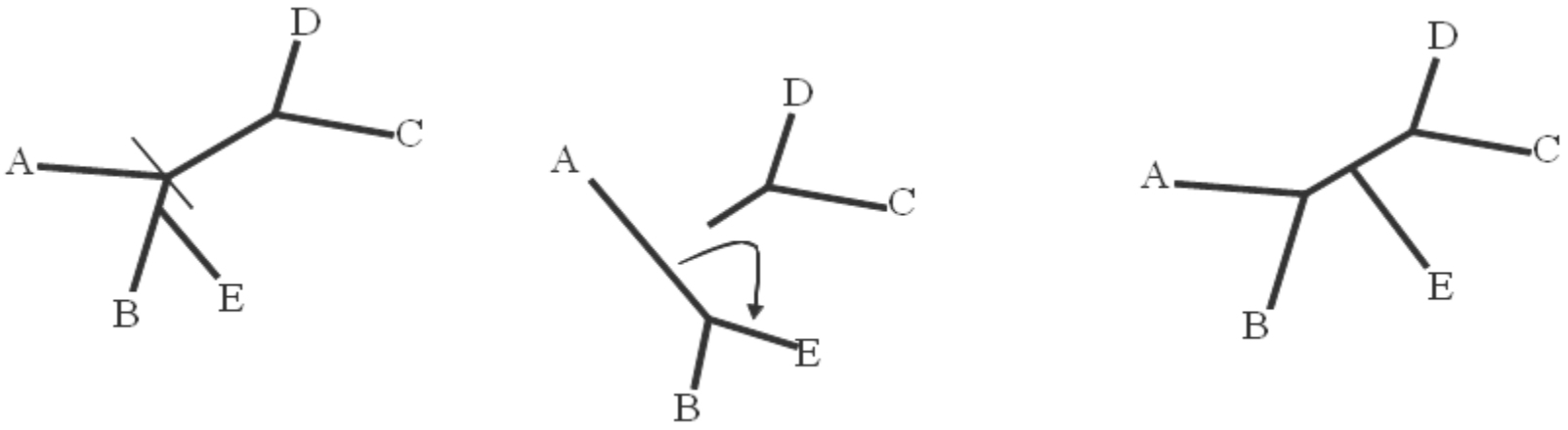
- Оторвём один лист и «привьём» его на другую ветвь



# Поиск лучшего дерева: просмотр соседних деревьев

## Что такое «соседние деревья»

- Можно проделать аналогичную операцию с целой кладой

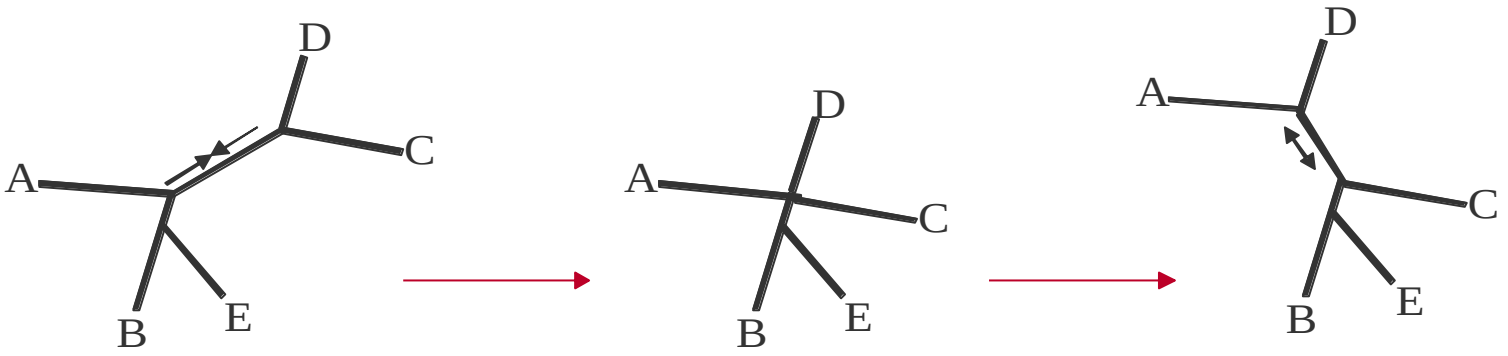



Такая операция обычно называется “SPR” : Subtree Pruning and Regrafting
В пакете PHYLIP она называется “Global rearrangement”.

# 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

- 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 (**least squares**, LS)
- 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, LS, FM, ME и многие другие принимают на вход матрицу расстояний.

# Прямые методы
## (они же эвристические)

- 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.

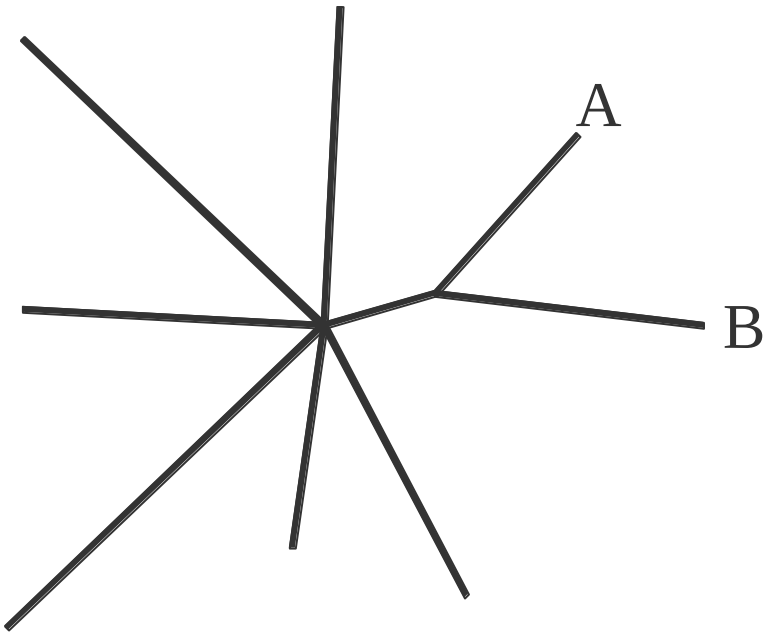
К этому прибавляется способ вычисления длин ветвей.

Результат — укоренённое ультраметрическое дерево с длинами ветвей.

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

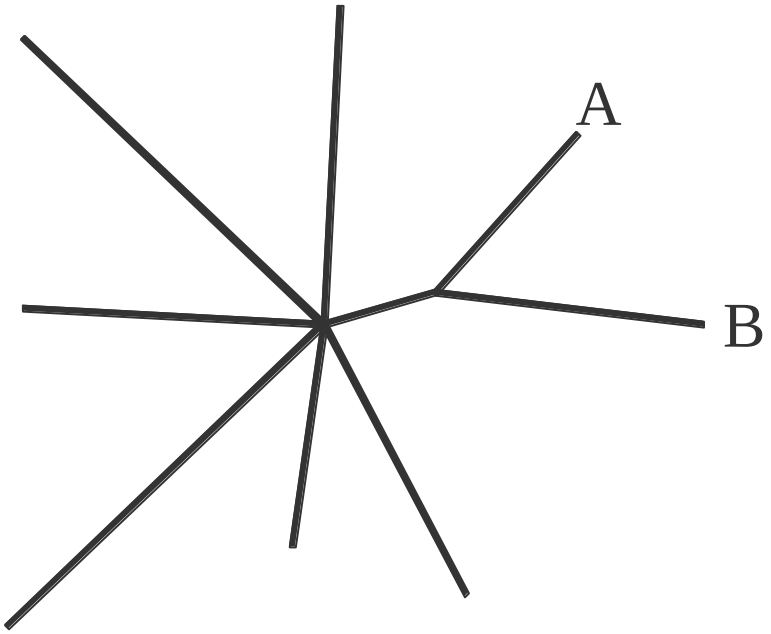
# Neighbor-joining

Выбираем пару последовательностей $A,B$ , для которых наименьшее значение имеет величина $(n–2)d(A,B) – s(A) – s(B)$,
где $d$ — расстояние из входной матрицы, $n$ — число последовательностей, а $s(A)$ — сумма расстояний от $A$ до всех остальных последовательностей.
Объединяем пару в кластер, с которым далее обращаемся как с одной последовательностью.



# Neighbor-joining (продолжение)

Повторяем объединение, пока не останется три кластера.




В отличие от UPGMA, даже при ультраметрической матрице «соседями» не обязательно будут две самые близкие последовательности!

**Полученное методом Neighbor-joining дерево — неукоренённое!**

Метод «по ходу дела» оценивает длины ветвей

(хотя эти длины иногда получаются отрицательными! :( )

# ПРОГРАММЫ РЕКОНСТРУКЦИИ ФИЛОГЕНИИ

# JalView

Основное назначение – работа с выравниваниями.

Включает реализацию двух примитивных способов оценки эволюционных расстояний (по числу совпадений и по весу сравнения с использованием BLOSUM62) и двух эвристических алгоритмов: UPGMA (переименованного в “Average Distance”) и Neighbor-Joining.

Очень часто этого вполне достаточно!

# Пакет PHYLIP

- Реализация методов UPGMA и Neighbor-Joining (программа *neighbor*), наименьших квадратов и Фитча – Марголиаша (*fitch* и *kitsch*), максимальной экономии (*dnapars* и *protpars*), наибольшего правдоподобия (*dnaml*, *dnamlk*, *proml*, *promlk*)
- Оценка эволюционных расстояний: программы *dnadist* и *protdist*
- Сравнение деревьев: *consense, treedist, treedistpair*
- Редактура (включая укоренение в среднюю точку): *retree*
- Бутстрэп: *seqboot*
- Визуализация: *drawtree, drawgram*

# Пакет PHYLIP

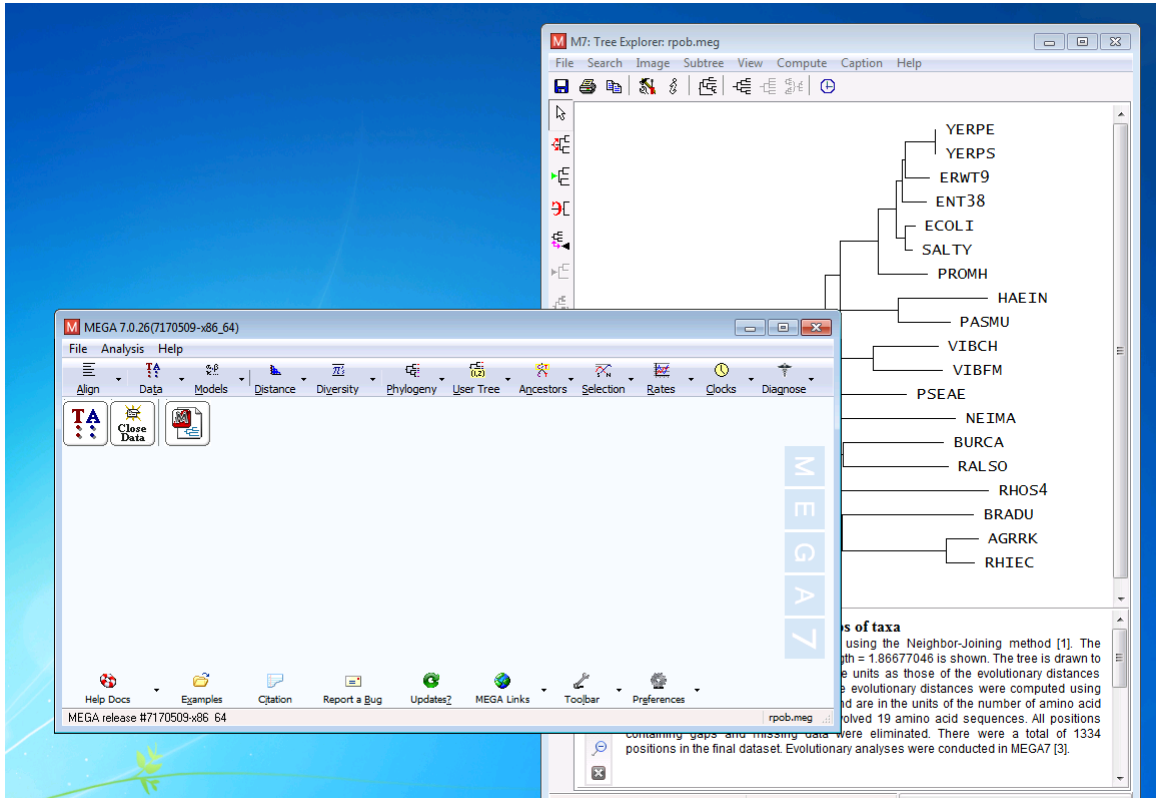
- Свободно распространяется, имеются версии для всех основных операционных систем. Доступен для скачивания на сайте http://evolution.genetics.washington.edu/phylip.html
- Своеобразный интерактивный (не оконный) интерфейс
- В пакет EMBOSS в качестве дополнения включены варианты всех программ пакета PHYLIP, снабженные интерфейсом в стиле EMBOSS (отличаются буквой *f* в начале, например *fprotpars* вместо *protpars*)

# MEGA

http://www.megasoftware.net/

Molecular Evolutionary Genetics Analysis

Помимо хорошей визуализации деревьев, включает реализацию ряда алгоритмов: вычисление расстояний, UPGMA, NJ, ML, MP, один из вариантов ME (не лучший).



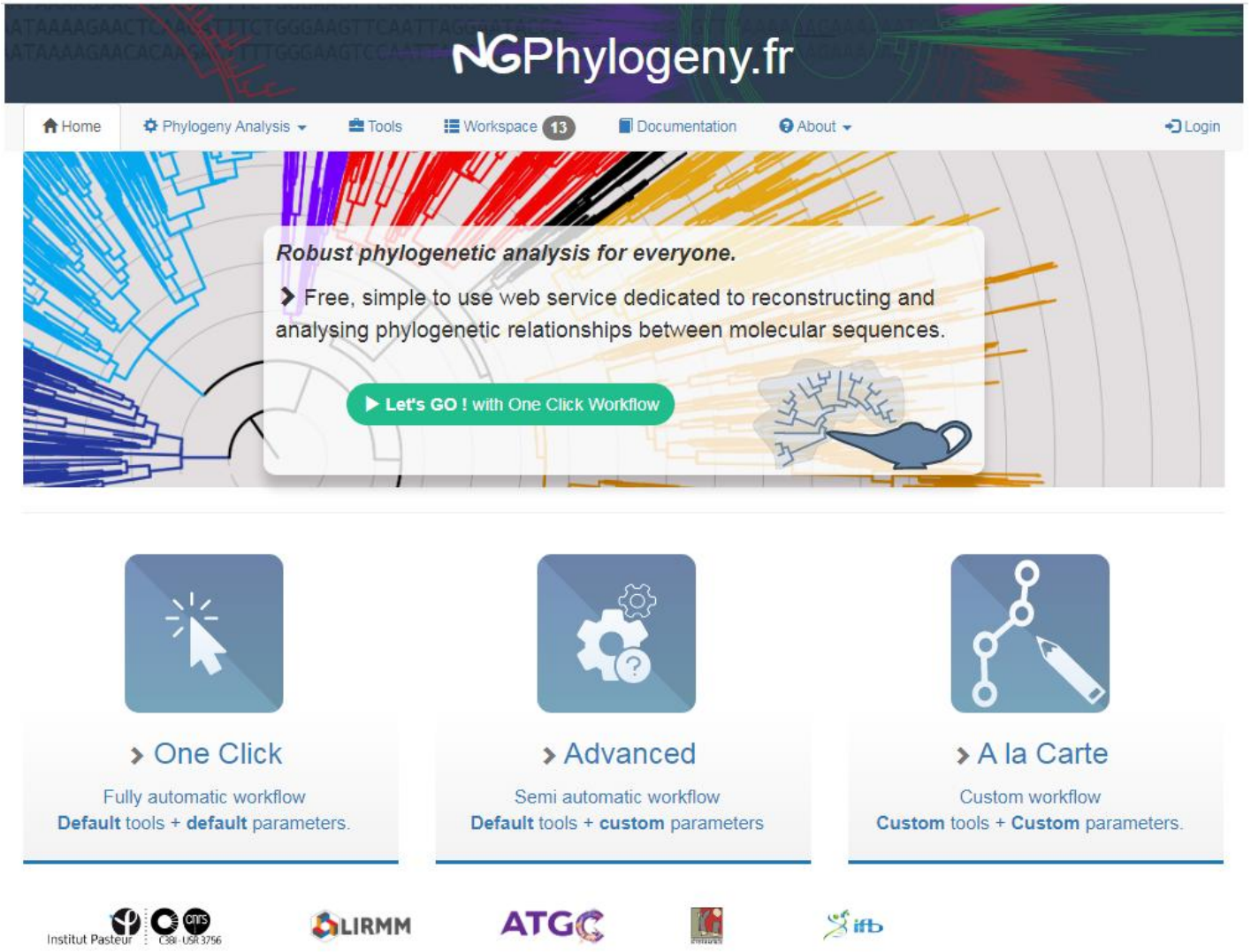
# Другие бесплатные программы

В отличие от MEGA, без оконного интерфейса

- FastME – очень хороший вариант минимальной эволюции
- TNT – максимальная экономия
- PhyML и RAxML – два пакета, реализующих наибольшее правдоподобие, с большим количеством моделей
- MrBayes – т.н. байесов метод (вариант ML, очень медленный, но часто даёт лучшие результаты, чем обычный ML).
- PhyloBayes – ещё одна популярная реализация байесова метода

См. также http://evolution.genetics.washington.edu/phylip/software.html

# NGPhylogeny.fr



NGPhylogeny.fr

Home | Phylogeny Analysis | Tools | Workspace 13 | Documentation | About | Login

## A La Carte Using the workflow maker, customize your workflow by selecting the right tools

### Name

Workflow name..

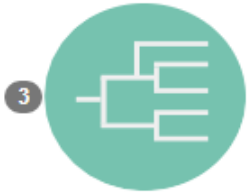
### Tools


**Multiple Alignment**

- MAFFT
- MUSCLE
- Clustal Omega

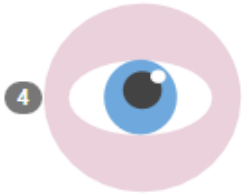

**Alignment Curation**

- BMGE
- Gblocks
- Noisy
- trimAl




**Tree Inference**

- FastME
- TNT
- PhyML+SMS
- PhyML
- FastTree
- MrBayes




**Tree Rendering**

- Newick Display