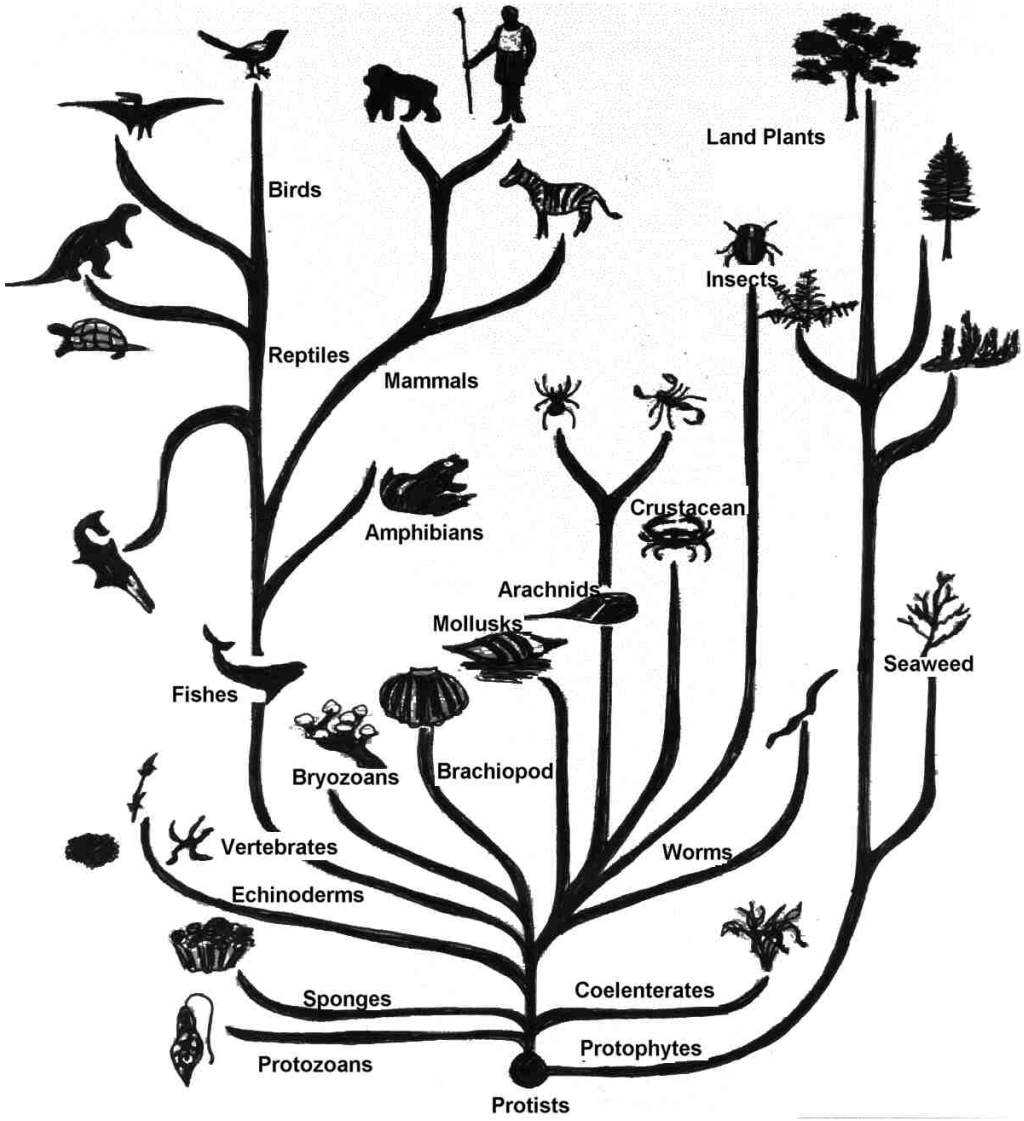
Реконструкция эволюции
Филогенетические деревья

С.А.Спирин
НИИ физико-химической биологии им. А.Н.Белозерского МГУ

Древо жизни



В наше время выяснять детали происхождения видов помогают последовательности ДНК и белков.

Гены и белки

Геном
3·109 букв у человека,
~ 106 букв у бактерий

содержит

Гены
~1,5% генома у человека,
~ 90% у бактерий

кодируют

Белки
~ 25 000 у человека,
600 – 6000 у бактерий

Генетический код

	T(U)	C	A	G
T(U)	TTT Phe	TCT Ser	TAT Tyr	TGT Cys
	TTC Phe	TCC Ser	TAC Tyr	TGC Cys
	TTA Leu	TCA Ser	TAA Stop	TGA Stop
	TTG Leu	TCG Ser	TAG Stop	TGG Trp
C	CTT Leu	CCT Pro	CAT His	CGT Arg
	CTC Leu	CCC Pro	CAC His	CGC Arg
	CTA Leu	CCA Pro	CAA Gln	CGA Arg
	CTG Leu	CCG Pro	CAG Gln	CGG Arg
A	ATT Ile	ACT Thr	AAT Asn	AGT Ser
	ATC Ile	ACC Thr	AAC Asn	AGC Ser
	ATA Ile	ACA Thr	AAA Lys	AGA Arg
	ATG Met	ACG Thr	AAG Lys	AGG Arg
G	GTT Val	GCT Ala	GAT Asp	GGT Gly
	GTC Val	GCC Ala	GAC Asp	GGC Gly
	GTA Val	GCA Ala	GAA Glu	GGA Gly
	GTG Val	GCG Ala	GAG Glu	GGG Gly

Аминокислоты

A Ala Alanine Аланин
R Arg Arginine Аргинин
N Asn Asparagine Аспарагин
D Asp Aspartic Acid Аспарагиновая кислота
C Cys Cysteine Цистеин
Q Gln Glutamine Глютамин
E Glu Glutamic Acid Глутаминовая кислота
G Gly Glycine Глицин
H His Histidine Гистидин
I Ile Isoleucine Изолейцин
L Leu Leucine Лейцин
K Lys Lysine Лизин
M Met Methionine Метионин
F Phe Phenylalanine Фенилаланин
P Pro Proline Пролин
S Ser Serine Серин
T Thr Threonine Треонин
W Trp Thryptophan Триптофан
Y Tyr Tyrosine Тирозин
V Val Valine Валин
"Stop" в таблице кода означает стоп-кодон — сигнал окончания трансляции.

Мутации

gatcaacactacttgacttcaa**g**acttaccataaagaaaac

gatcaacactacttgacttcaa**a**acttaccataaagaaaac

точечная замена

---

gatcaacactacttgacttcaa**ga**cttaccataaagaaaac

gatcaacactacttgacttcaacttaccataaagaaaac

делеция

---

gatcaacactacttgacttcaagacttaccataaagaaaac

gatcaacactacttgacttcaaga**ta**cttaccataaagaaaac

инсерция
(вставка)

Мутации (точечные замены) в гене

… AATCCGTCAAGTCTA…

… **Asn Pro Ser Ser Leu** …

1) “молчащая”(синонимическая)мутация

… AATCCGTCGAGTCTA…

… **Asn Pro Ser Ser Leu** …

2) замена остатка на близкий по свойствам

… AATCCGACAAGTCTA…

… **Asn Pro Thr Ser Leu** …

3) замена остатка на остаток с иными свойствами

… AATCCGTCAAGACTA…

… **Asn Pro Ser Arg Leu** …

Эволюция белков

Мутации возникают случайно.

Конкретная мутация может быть:

- летальной;
- вредной;
- слабовредной;
- нейтральной;
- полезной.

Мутация порождает **полиморфизм** данного белка в популяции.

Доля каждого варианта подвержена случайным изменением (модель: «случайное блуждание с поглощением»).

За исторически короткое время один из вариантов (старый или новый) исчезает. Во втором случае говорят, что мутация **закрепилась.**

Мы видим лишь закрепившиеся мутации

А шанс закрепиться есть лишь у безвредных мутаций…

```
CYB5_CHICK          1 MVGSSEAGGEAWRGRYYRLEEVQKHNNSQSTWIIVHHRIYDITKFLDEHP     50
                      .:|...||  .:||.|||:||||:|:|||:|:||::||:||||:|||
CYB5_HUMAN          1 ---MAEQSDEA--VKYYTLEEIQKHNHSKSTWLILHHKVYDLTKFLEEHP     45

CYB5_CHICK         51 GGEEVLREQAGGDATENFEDVGHSTDARALSETFIIGELHPDDRPKLQKP    100
                      ||||||||||||||||||||||||||.:|:|||||||||||||||.||
CYB5_HUMAN         46 GGEEVLREQAGGDATENFEDVGHSTDAREMSKTFIIGELHPDDRPKLNKP     95

CYB5_CHICK        101 AETLITTVQSNSSSWSNWVIPAIAAIIVALMYRSYMSE-    138
                      .||||||:.|:||.|:||||||||:|:.||||||.||:|
CYB5_HUMAN         96 PETLITTIDSSSSWWTNWVIPAISAVAVALMYRLYMAED    134
```

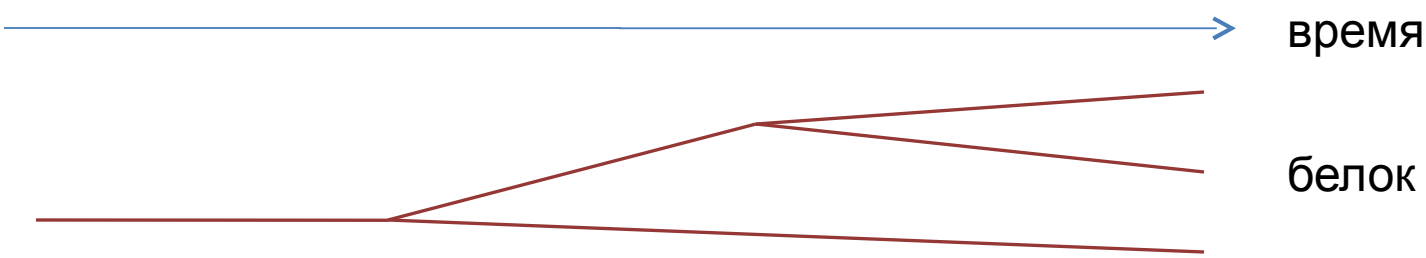
История белка

Приближённая картина: один белок – это конкретный белок в конкретный момент времени у конкретного вида живых организмов.

Можно (теоретически) проследить историю данного белка во времени. С течением времени последовательность белка меняется. Это и называется **эволюцией** белка.

При разделении вида на два все белки этих видов начинают эволюционировать независимо

Кроме того, нередко случается дупликация гена в геноме; после дупликации соответствующие белки также эволюционируют независимо



Эволюция видов и эволюция белков

Когда виды разделяются, то разделяются пути эволюции всех их белков…

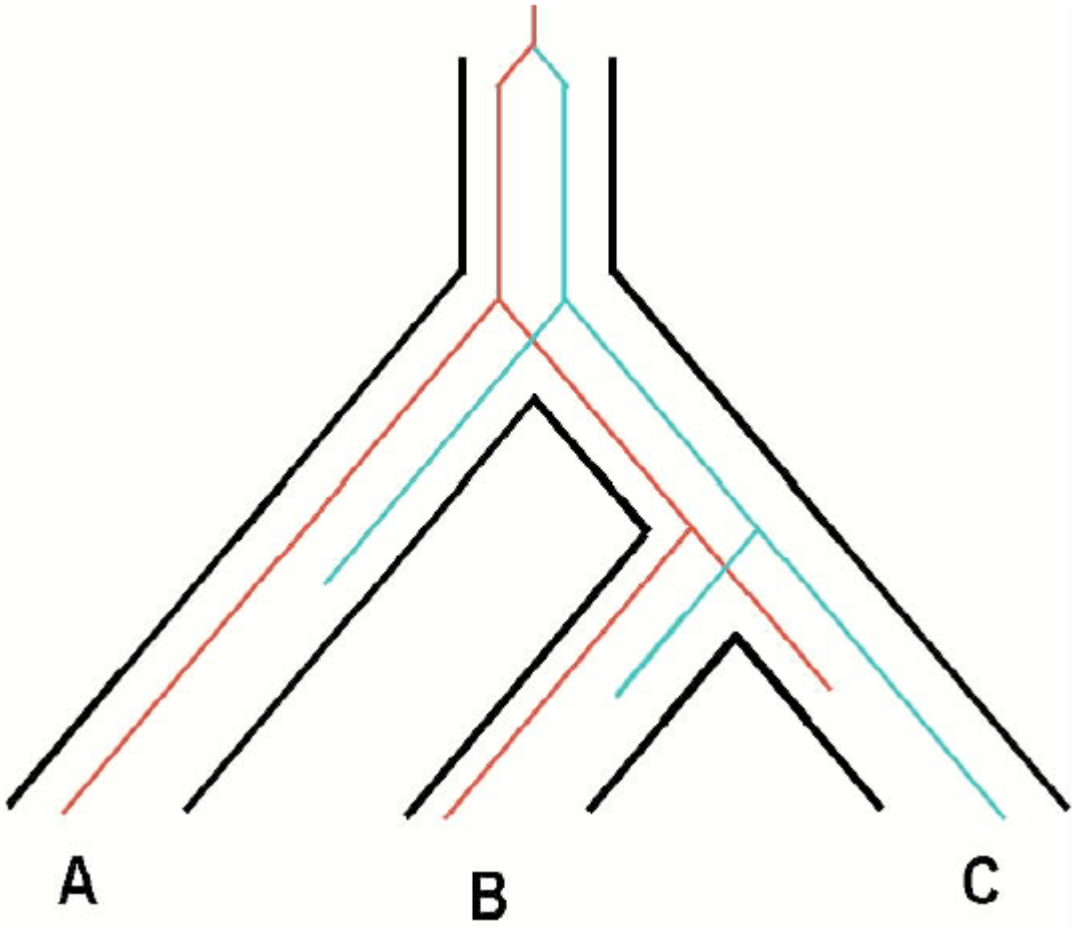
В результате большинству белков одного вида соответствует **ортолог** в другом виде.

Но:

1. Бывают дупликации белков без разделения видов.
   Два родственных белка существуют в одном геноме и эволюционируют (почти) независимо – такие белки называются **паралогами**.
2. Бывают потери генов.
   Если в двух видах потерялись по одному белку из пары паралогов, то получается, что общий предок белков, которые выглядят как ортологи, «жил» существенно раньше, чем общий предок видов.
3. Бывает, что два белка объединяются в один многодоменный, и наоборот.
   Поэтому правильнее говорить об эволюции белковых доменов.

Дерево видов и дерево белков

(пример ситуации, возникающей в результате дупликации генов и потерей паралогов)

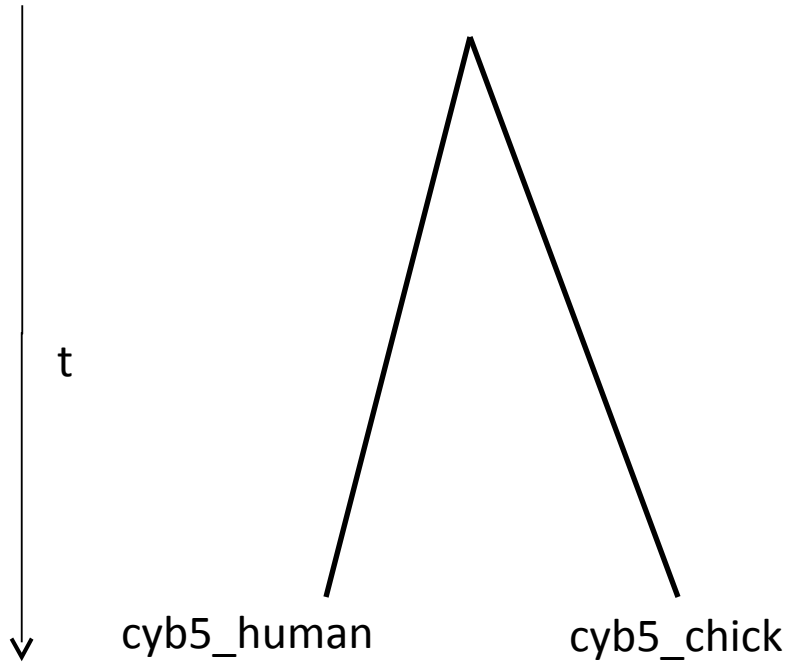


Попытка реконструкции эволюции организмов по данному белку приведёт к ошибке: данный белок из организма B имеет ближайшего родственника в организме A, хотя B ближе к C, чем к A.

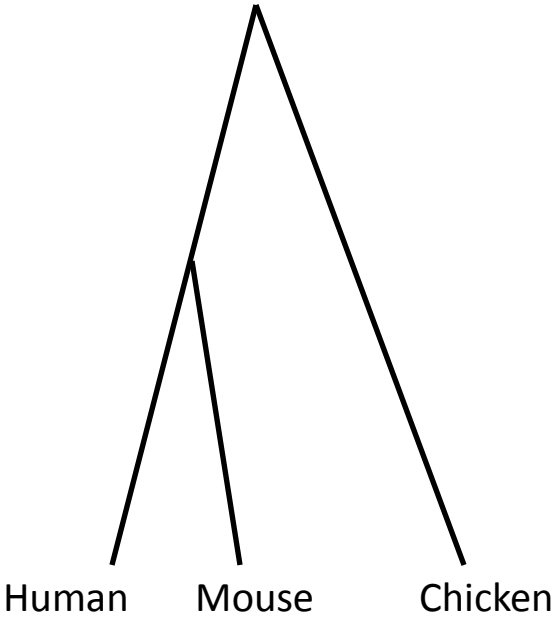
Путь эволюции

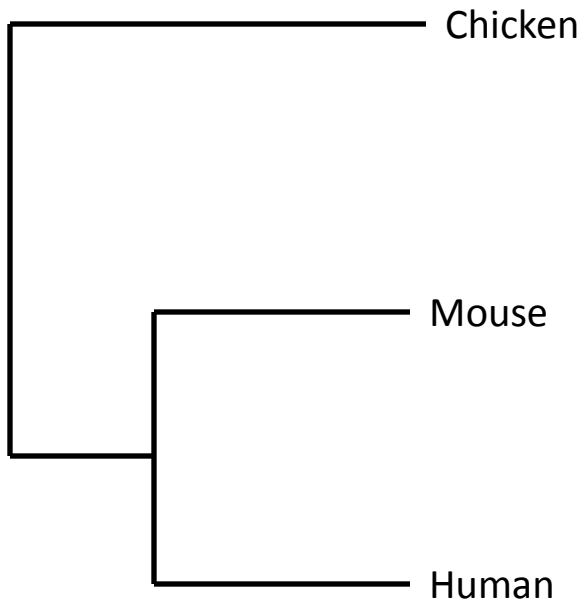
cyb5_human ——— cyb5_chick

или

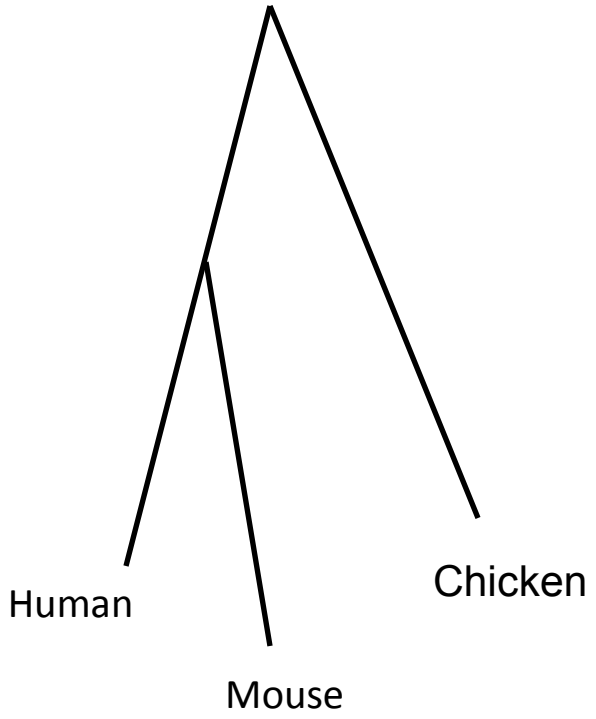


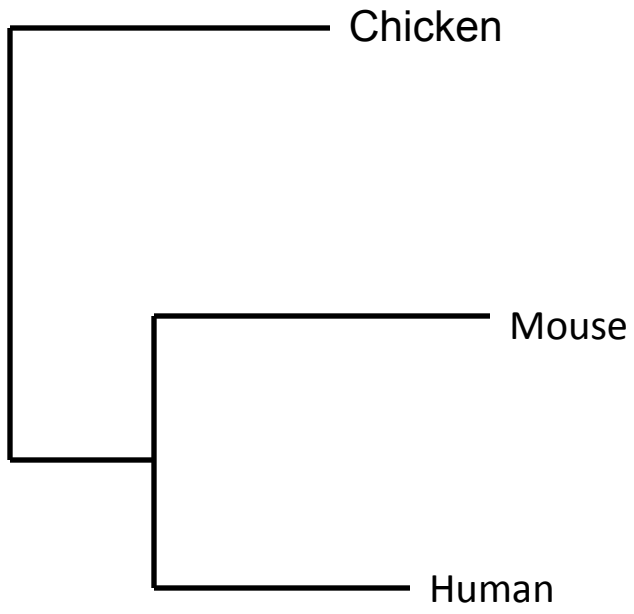
Филогенетическое дерево





«Молекулярные часы»: всегда идут, но иногда неточно

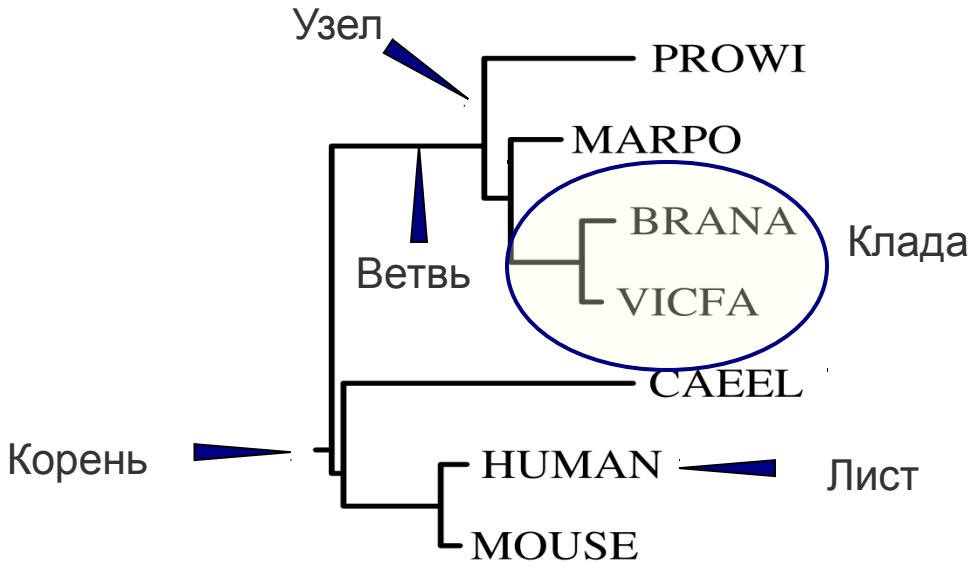




Когда хотят отразить разное число мутаций, произошедших на пути от общего предка, получается что-то вроде такого.

Филогенетическое дерево (терминология)

- **Узел (node)** — точка разделения предковой последовательности. Соответствует внутренней вершине графа, изображающего эволюцию.
- **Лист (leaf)** — реальный (современный) объект; внешняя вершина графа.
- **Ветвь (branch)** — связь между узлами или между узлом и листом; ребро графа.
- **Корень (root)** — гипотетический общий предок всех рассматриваемых объектов.
- **Кла́да (clade)** — группа всех потомков некоторого ранее существовавшего объекта.



Длины ветвей дерева

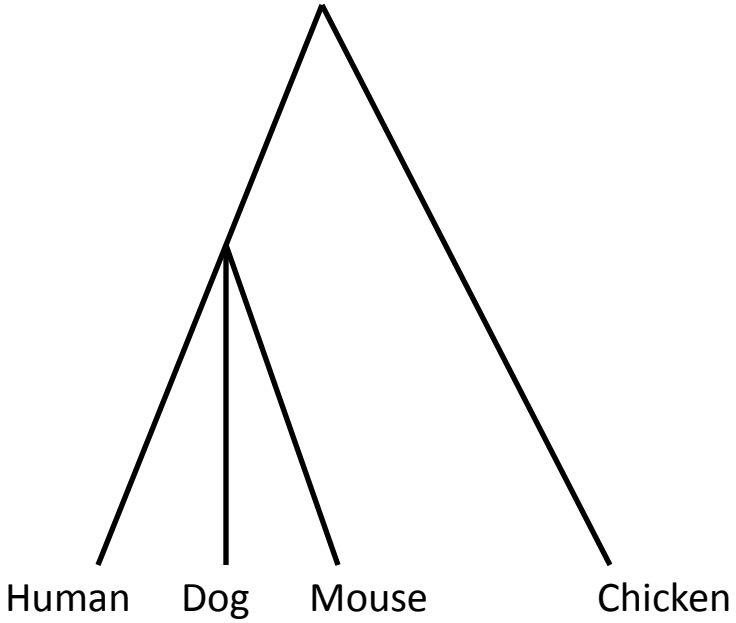
Каждая точка дерева – некоторая последовательность, существовавшая в некоторый момент времени (в прошлом, если эта точка – не лист).

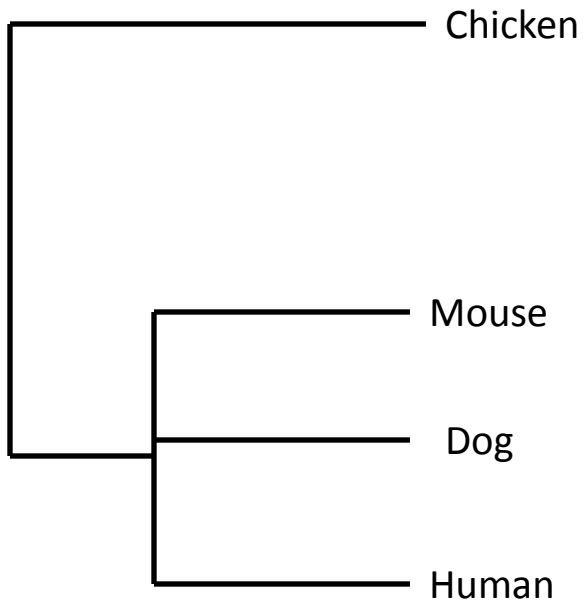
Длины ветвей могут иметь двоякий смысл:

1) интервал времени между моментами существования двух последовательностей;

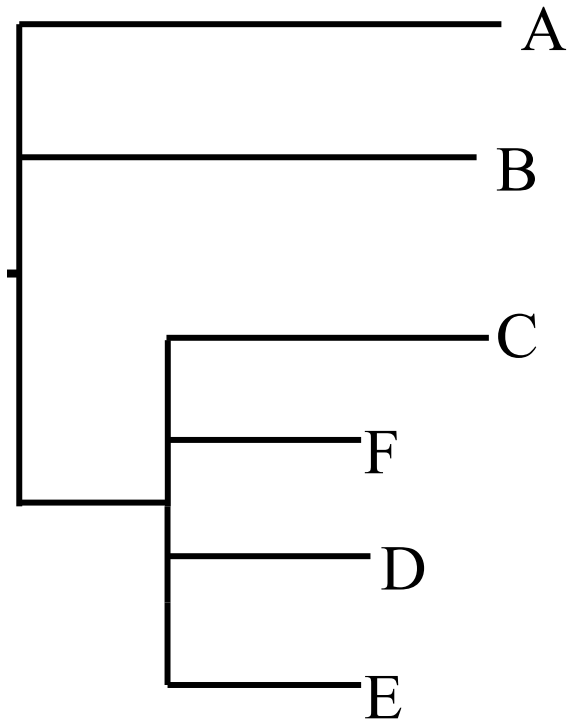
2) число мутаций, случившихся на пути от одной последовательности до другой.

Небинарное дерево

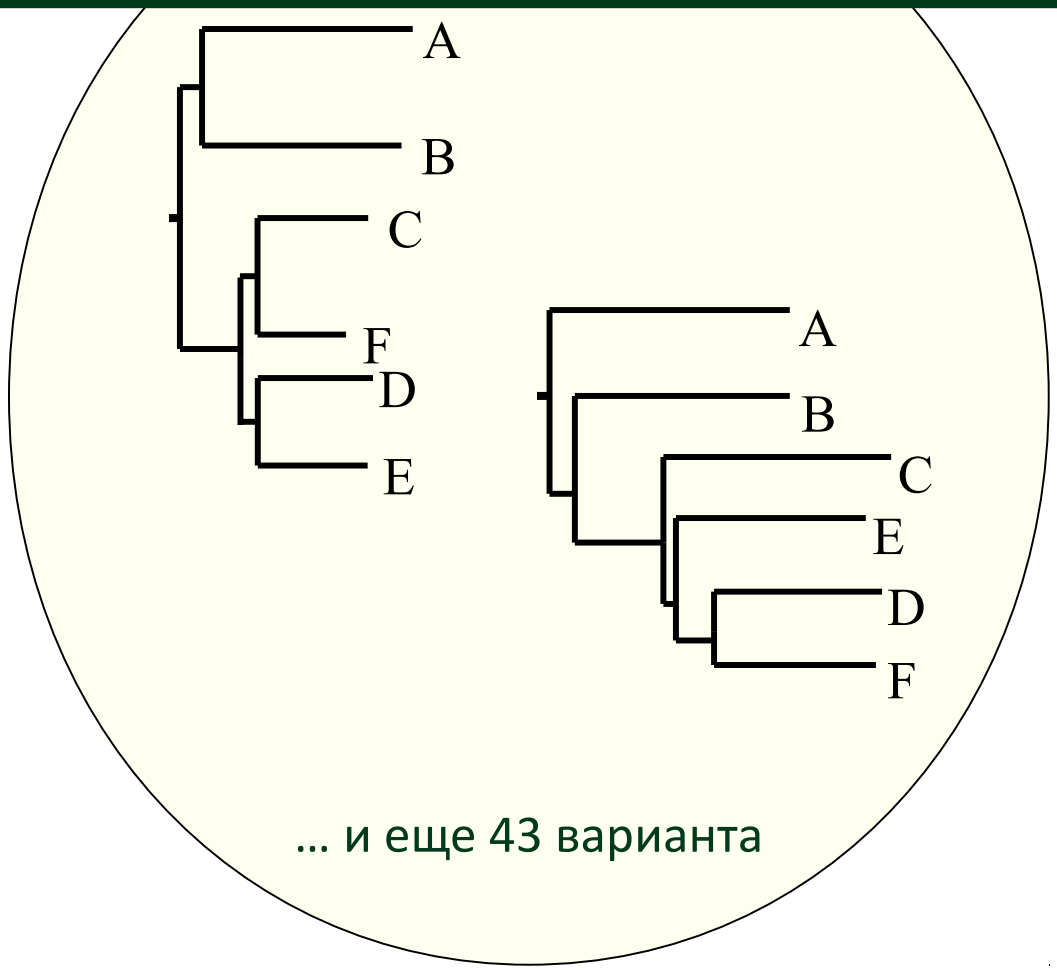




Небинарное дерево следует понимать как множество возможных «разрешений»

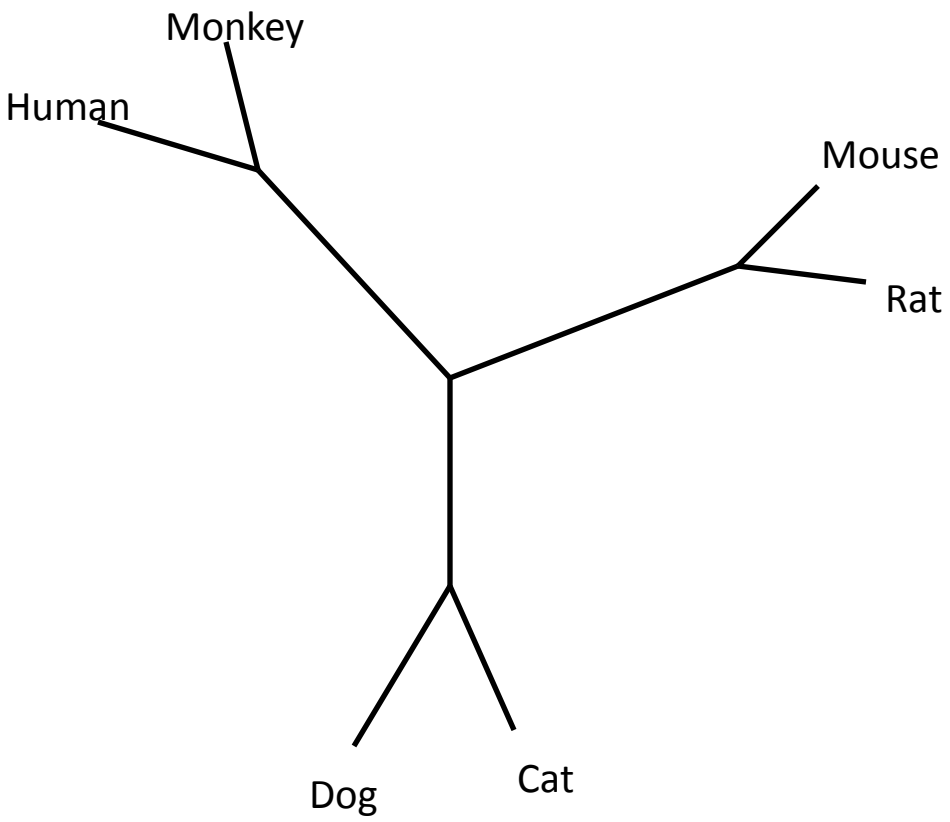


=

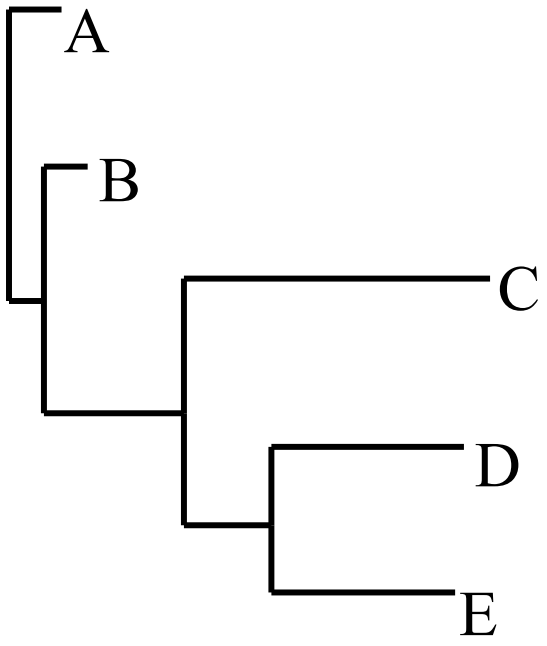


Термины «небинарное» и «неразрешённое» будем считать синонимами.

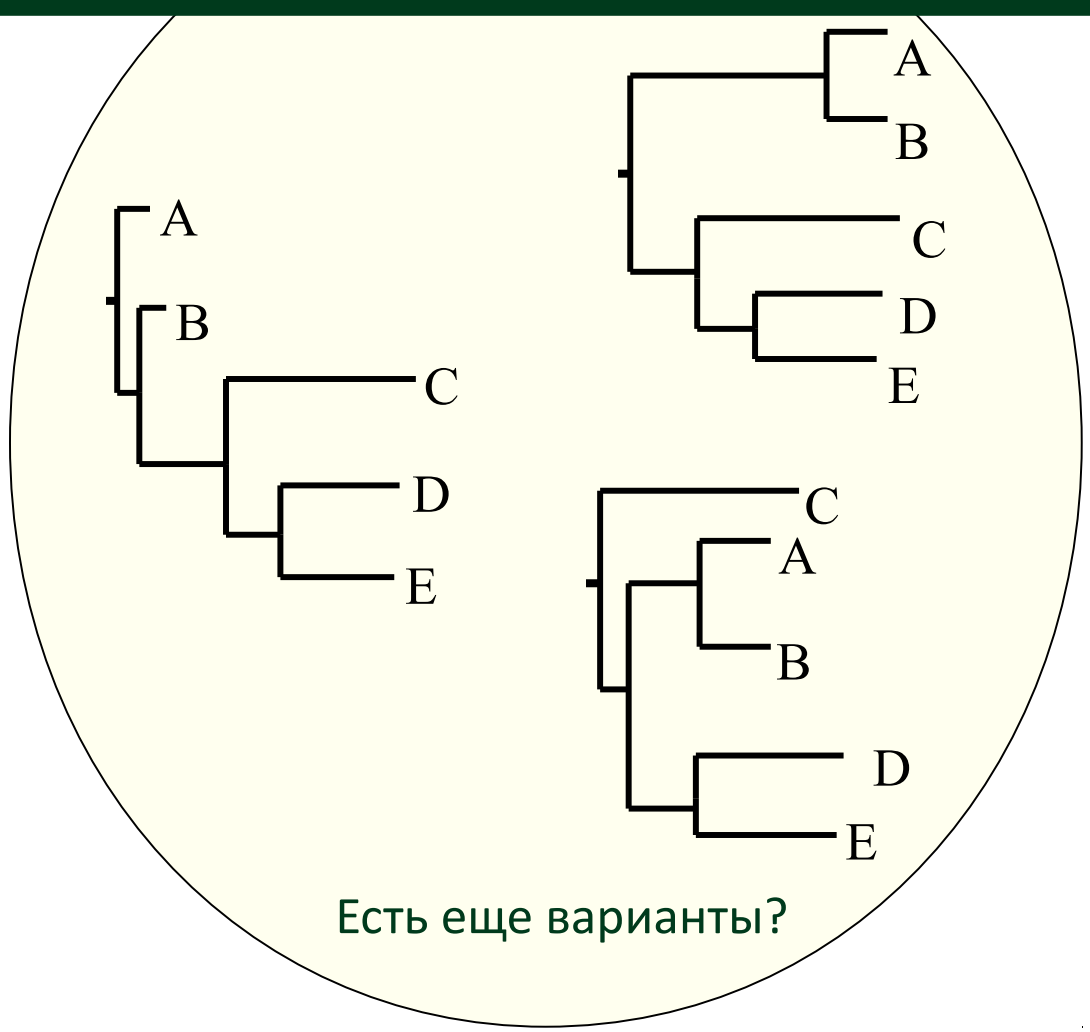
Неукоренённое дерево



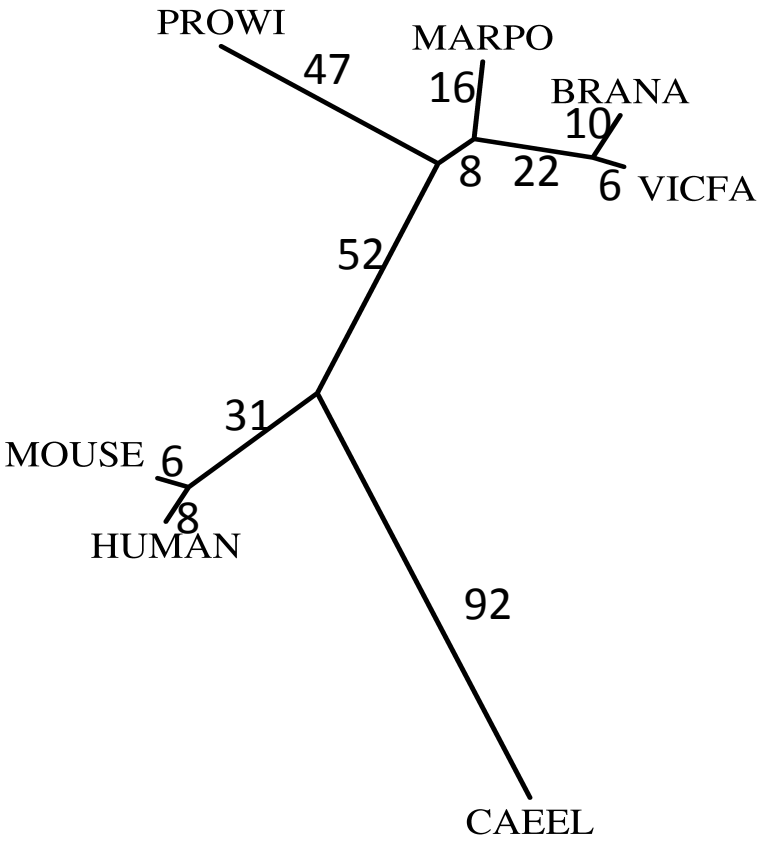
Неукоренённое дерево следует понимать как множество возможных укоренений



=

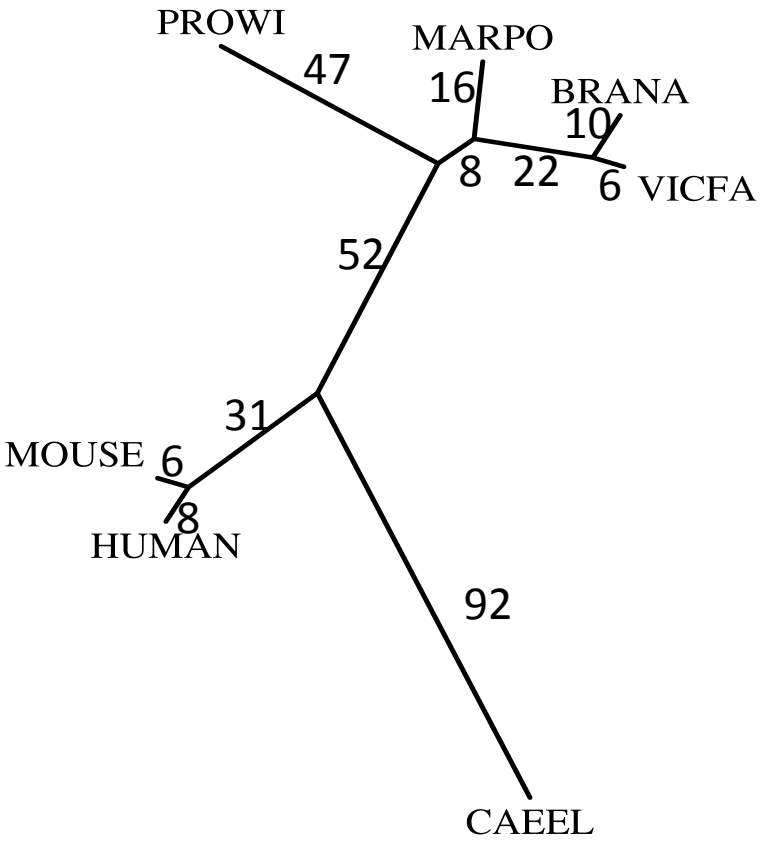


Длины ветвей и расстояния по дереву между листьями



D(MOUSE, CAEEL) = 6+31+92 = 129

Длины ветвей и расстояния по дереву между листьями



D(MOUSE, CAEEL) = 6+31+92 = 129

Дерево с заданными длинами ветвей порождает метрическое пространство, элементами которого являются листья

Ультраметрические деревья

Дерево называется ультраметрическим, если на нём есть точка, расстояния от которой до всех листьев одинаковы.

В этом случае множество листьев является ультраметрическим пространством: для любых трёх листьев a,b,c верно $d(a,b) \leq \max\big(d(a,c), d(b,c)\big)$.

Если все листья представляют **современные** последовательности, а длины ветвей имеют смысл **времени**, то дерево ультраметрическое.

Молекулярные часы

Гипотеза молекулярных часов: за одинаковое время происходит в среднем одинаковое число мутаций

Если гипотеза верна, то можно оценивать **эволюционное время** между современными последовательностями и на основании этих оценок строить **укоренённое ультраметрическое дерево.**

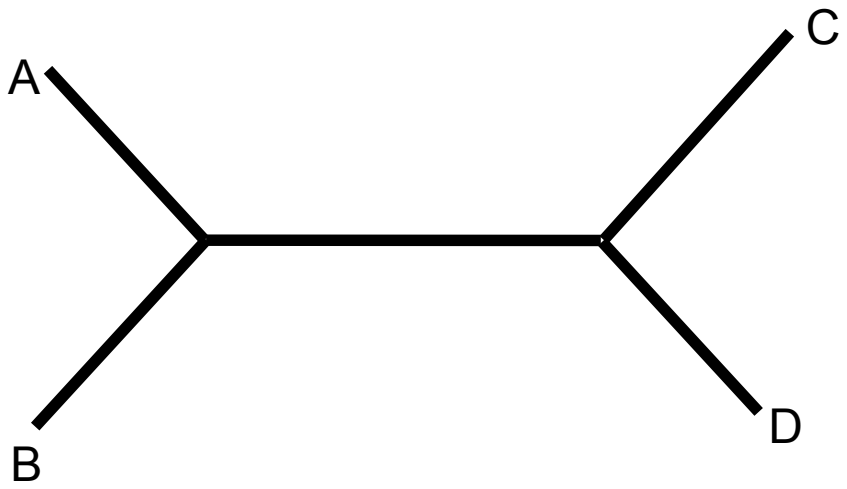
Но гипотеза МЧ часто не выполняется.

Расстояние как число мутаций

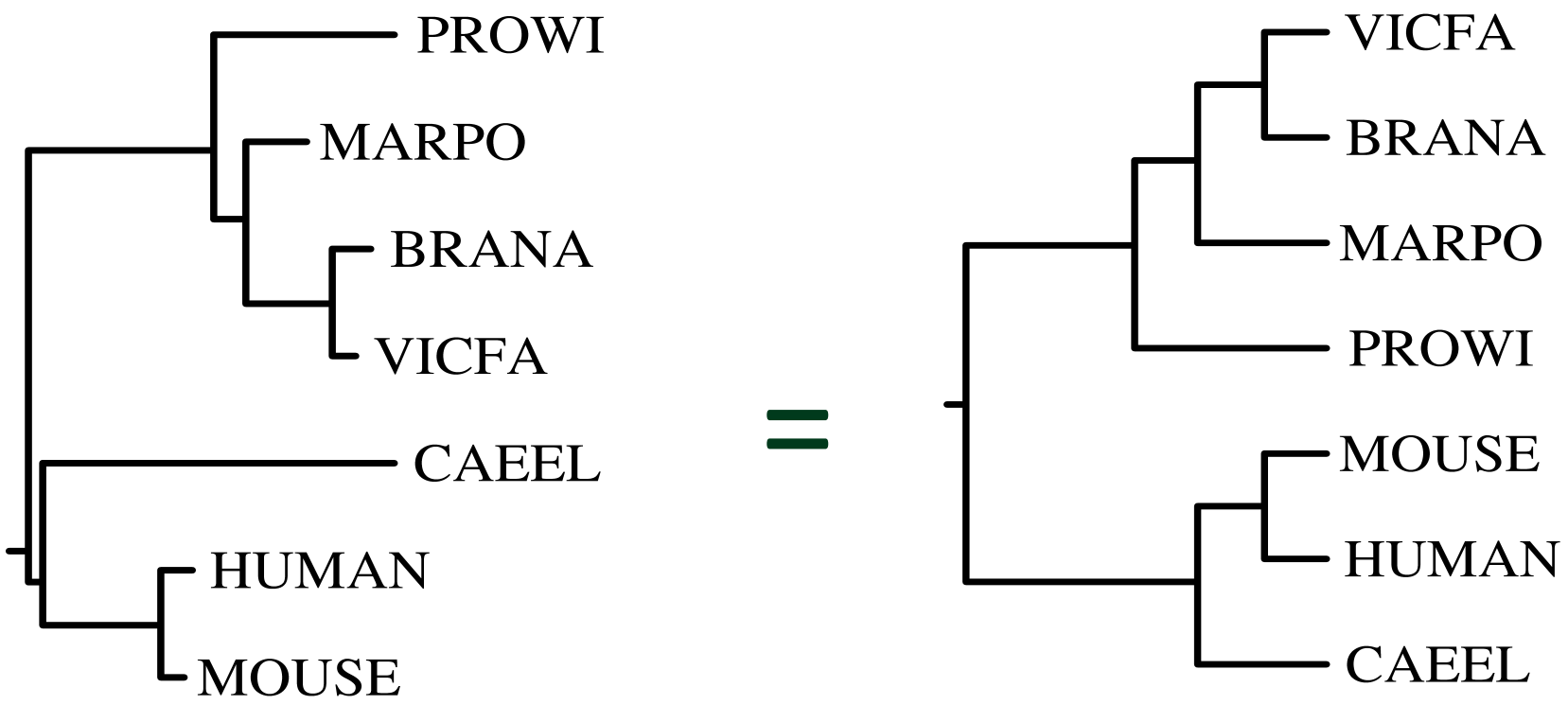
Расстояние между последовательностями ультраметрично, если его понимать как эволюционное время...

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

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.



Топология дерева



Топология дерева

Каждая ветвь разбивает множество листьев на два.

В каждом дереве есть **тривиальные** ветви (отделяющие один лист от всех остальных), они не зависят от топологии.

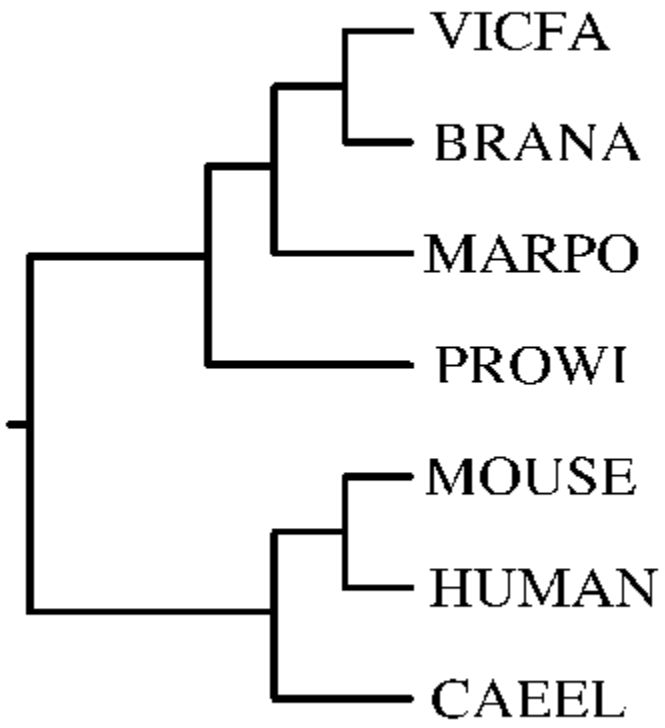
Топологию (неукоренённого) дерева можно однозначно записать набором нетривиальных разбиений. Например:

{HUMAN, MOUSE} vs {CAEEL,PROWI,MARPO,BRANA,VICFA}

{HUMAN, MOUSE,CAEEL} vs {PROWI,MARPO,BRANA,VICFA}

{HUMAN,MOUSE, CAEEL,PROWI} vs {MARPO,BRANA,VICFA}

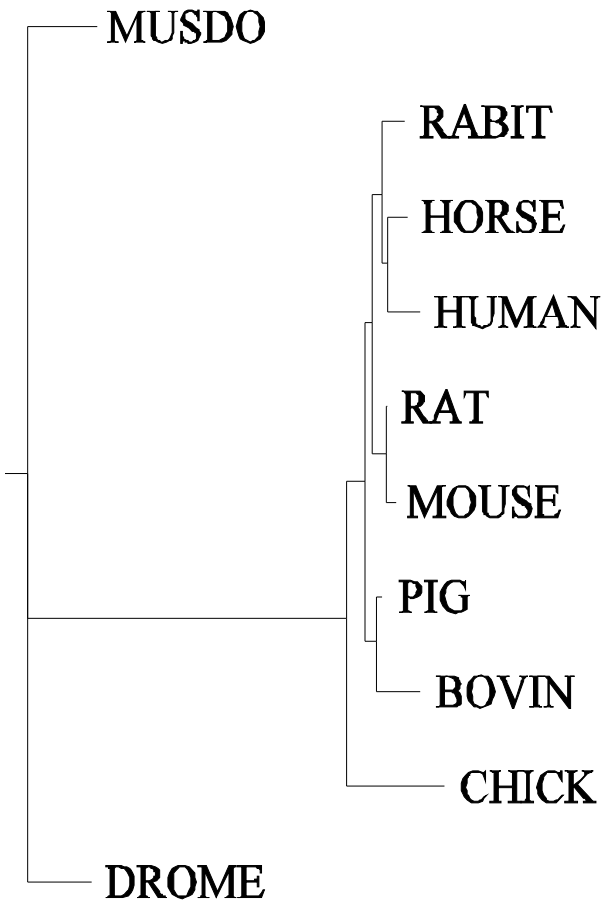
{HUMAN,MOUSE,CAEEL,PROWI,MARPO} vs {BRANA,VICFA}

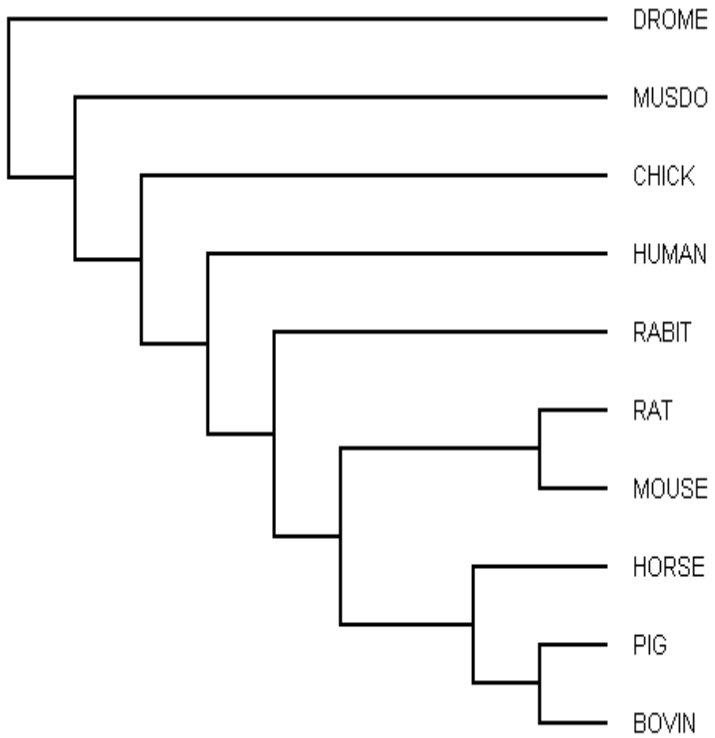


HUMAN	MOUSE	CAEEL	VICFA	BRANA	MARPO	PROWI
+	+	-	-	-	-	-
+	+	+	-	-	-	-
+	+	+	-	-	-	+
+	+	+	-	-	+	+

Представление топологии дерева разбиениями позволяет отождествлять ветви разных деревьев с одним и тем же множеством листьев.

В частности, если имеются две реконструкции эволюции по одним и тем же данным, то можно сказать, в каких ветвях они согласуются, а в каких – расходятся.





Найдите три ветви, общие для этих двух деревьев!

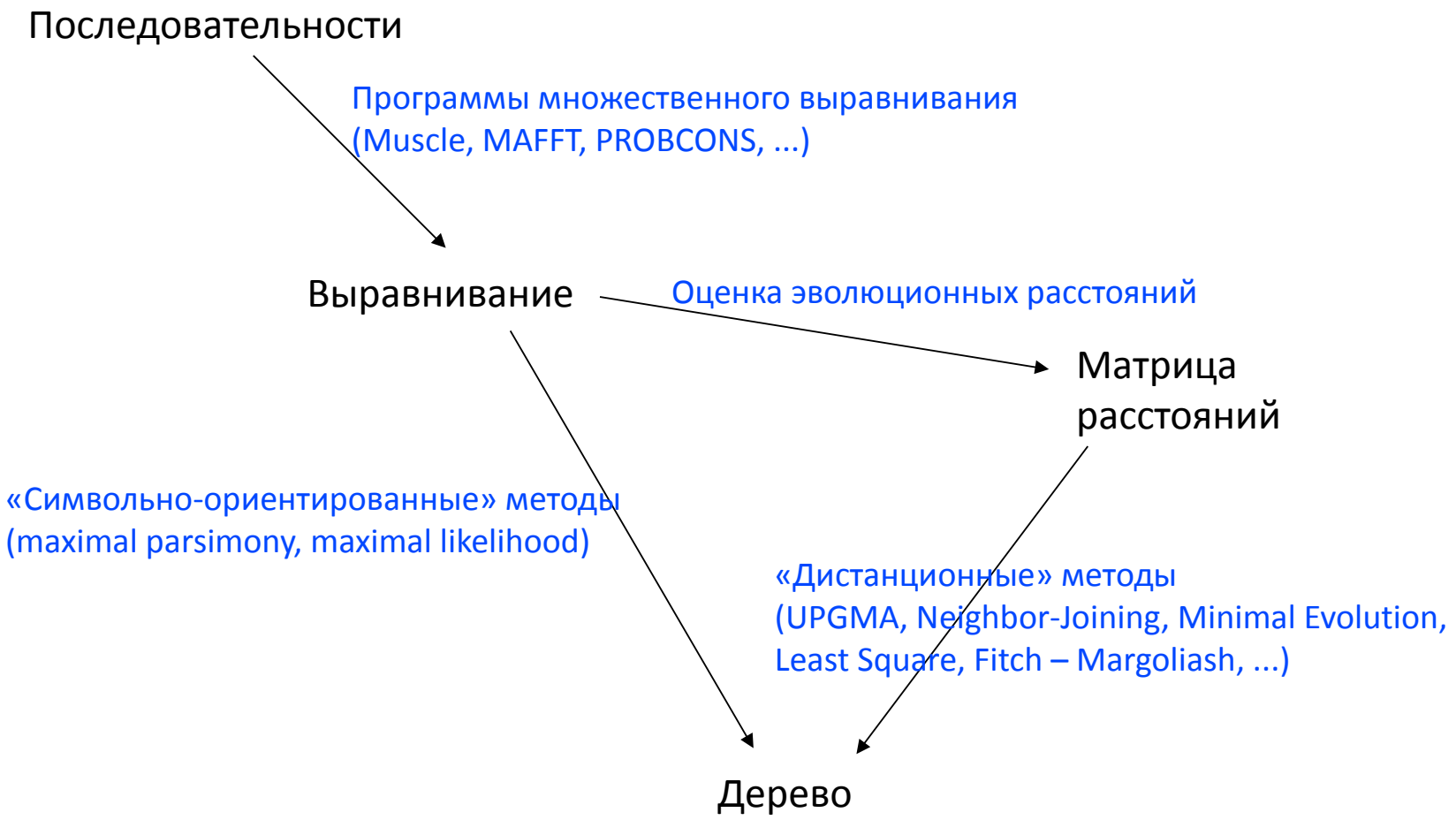
Исходный материал для реконструкции дерева: множественное выравнивание последовательностей белков

```
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 (maximal parsimony)
  - * наибольшего правдоподобия
- использующие матрицу расстояний
  - * кластерные (UPGMA и др.)
  - * Neighbor-joining
  - * минимальной эволюции
  - * наименьших квадратов
  - * Фитча – Марголиаша
  - * …

Схема реконструкции филогении по последовательностям



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

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

Но в реальности нам даны последовательности и требуется оценить число произошедших мутаций. Это не так просто, поскольку мутации могут происходить в одной и той же позиции.

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.
Кроме того, учитывается, что разные замены аминокислотных остатков имеют в среднем разные шансы закрепиться.

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

Классификация методов

Название метода	Переборный / эвристический	Предполагает молекулярные часы	Символьно ориентированный
UPGMA	Эвристический	Да	Нет
Neighbor-Joining	Эвристический	Нет	Нет
Наименьших квадратов	Переборный	Может	Нет
Фитча – Марголиаша	Переборный	Может	Нет
Минимальной эволюции	Переборный	Нет	Нет
Максимальной экономии	Переборный	Нет	Да
Наибольшего правдоподобия	Переборный	Может	Да

Методы, предполагающие молекулярные часы, строят укоренённые ультраметрические деревья.

Методы, не предполагающие молекулярные часы строят, как правило, 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 square)

Пусть дана матрица расстояний и топология дерева;
i, j — две последовательности, тогда мы имеем расстояние $d(i,j)$ из матрицы, и, приписав ветвям длину, будем иметь расстояние $d'(i,j)$ «по дереву».

Подберём длины ветвей так, чтобы сумма $(d(i,j) - d'(i,j))^2$ (по всем парам i,j) была наименьшей — это наименьшее значение и будет критерием качества (та топология считается лучшей, для которой это значение получится меньшим).

Поиск «лучшего» дерева

Имеется единственная топология (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

Все деревья перебрать (как правило) нельзя!

Число различных деревьев с N листьями равно:

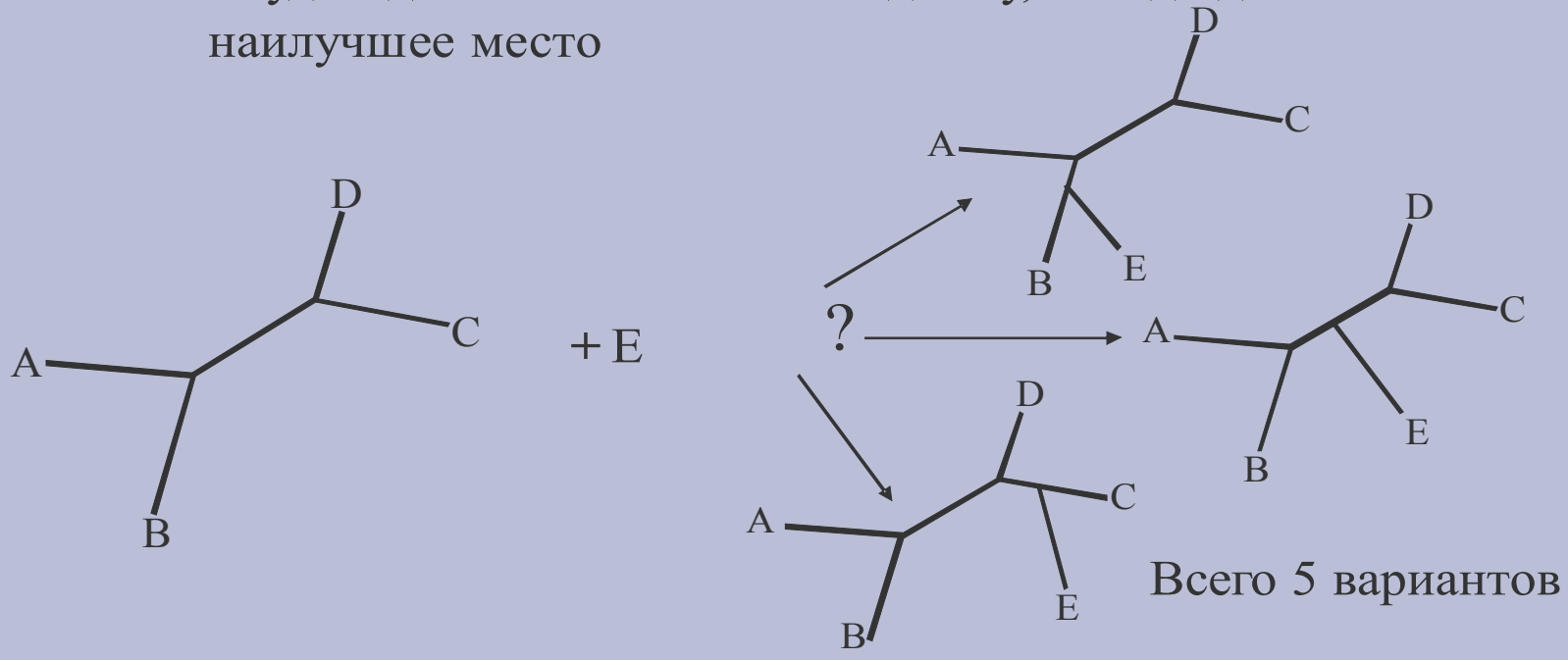
$$(2N - 5)!! = 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2N - 5)$$

Это число очень быстро растёт!

Полный перебор возможен, если число последовательностей не превышает 10–12

Поиск лучшего дерева: «выращивание»

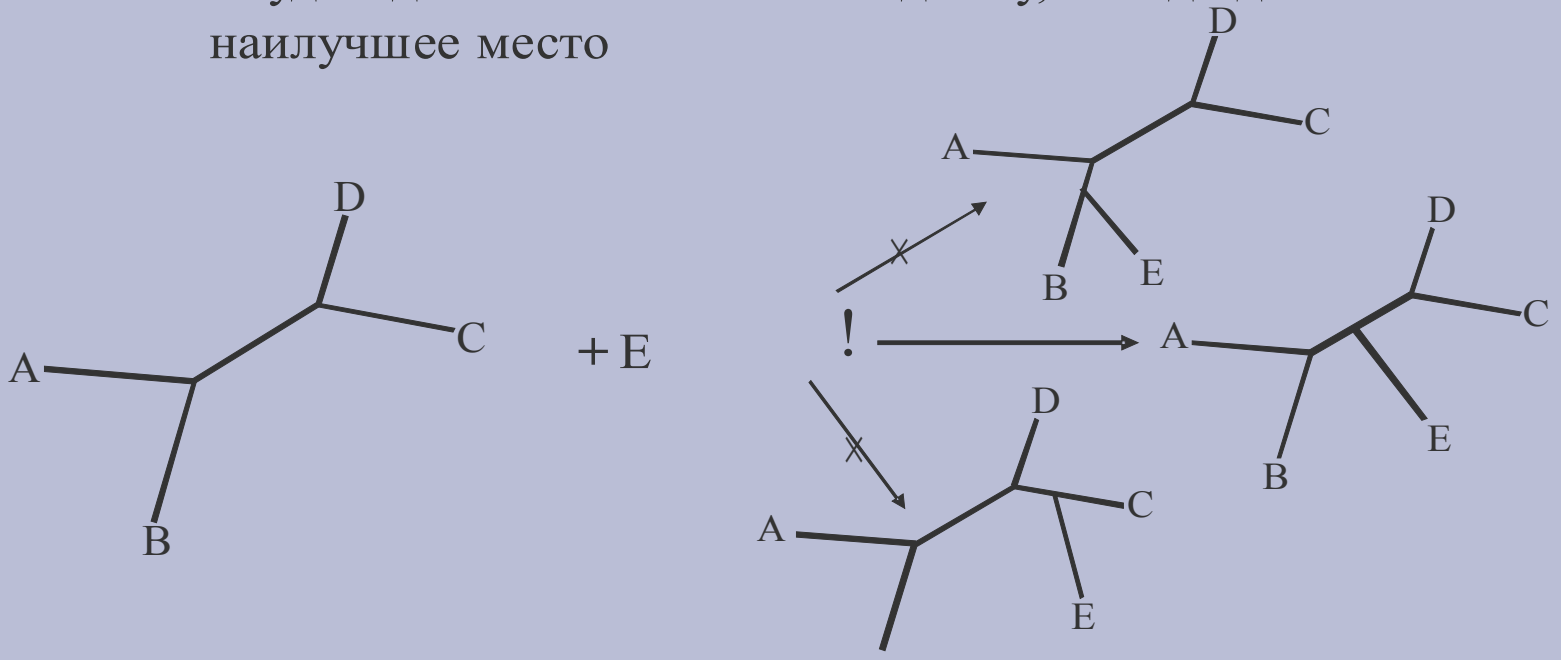
- Найдем лучшее дерево для части последовательностей
- Будем добавлять листья по одному, находя для них наилучшее место



Всего 5 вариантов

Поиск лучшего дерева: «выращивание»

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



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

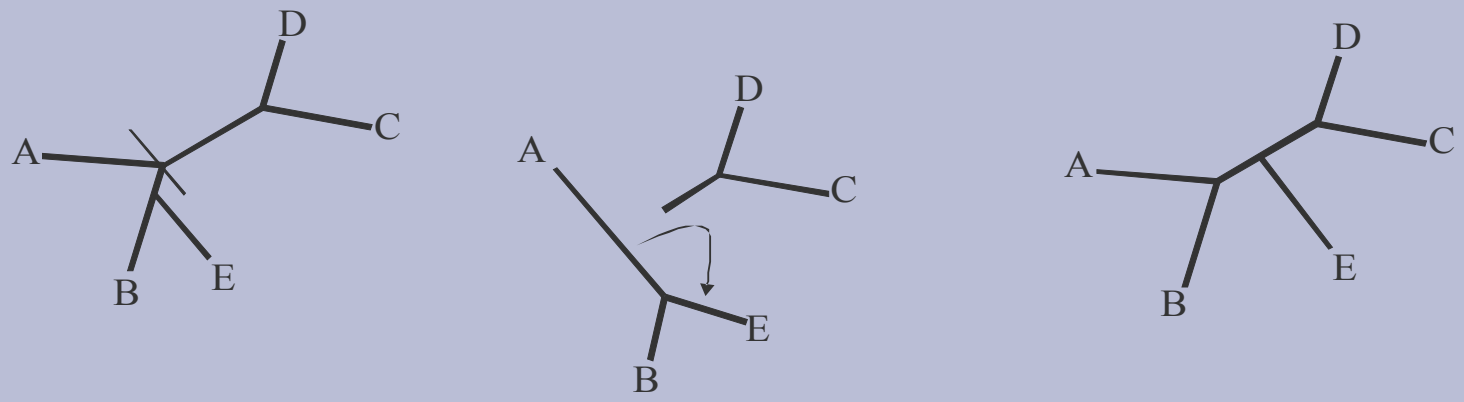
Что такое «соседние» деревья

- 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

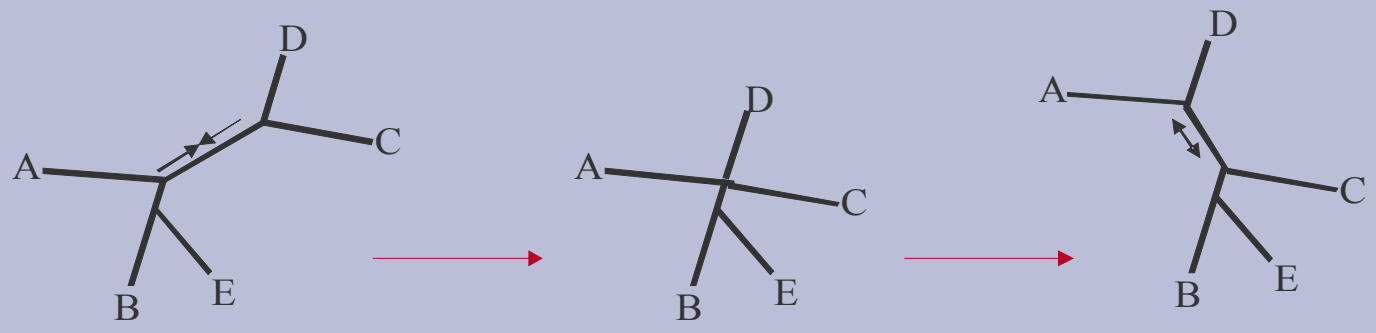


В пакете PHYLIP это называется “Global rearrangement”

Поиск лучшего дерева: просмотр соседних деревьев

Что такое «соседние» деревья

- Можно «схлопнуть» одну ветвь и заменить её другой



В пакете PHYLIP это называется “Local rearrangement”

Поиск лучшего дерева

Алгоритм поиска

- Строим черновое дерево
  (два варианта: эвристический метод или «выращивание» с использованием критерия качества).
- Анализируем соседние деревья;
  если находим среди соседей лучшее, берём за основу его.
- Повторяем предыдущий пункт, пока текущее дерево не окажется лучше всех своих соседей.

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 (на практике сводится к поиску «достаточно качественного» дерева).

Как правило, название метода совпадает с названием критерия.

Переборные методы

- Максимальной экономии (или «бережливости», maximum parsimony, MP)
- Наибольшего правдоподобия (maximal likelihood, ML)
- Наименьших квадратов (least squares, LS)
- Фитча – Марголиаша (Fitch – Margoliash, FM)

Все методы, кроме MP, допускают предположение о молекулярных часах (но чаще используются без этого предположения!).

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

Эвристические 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

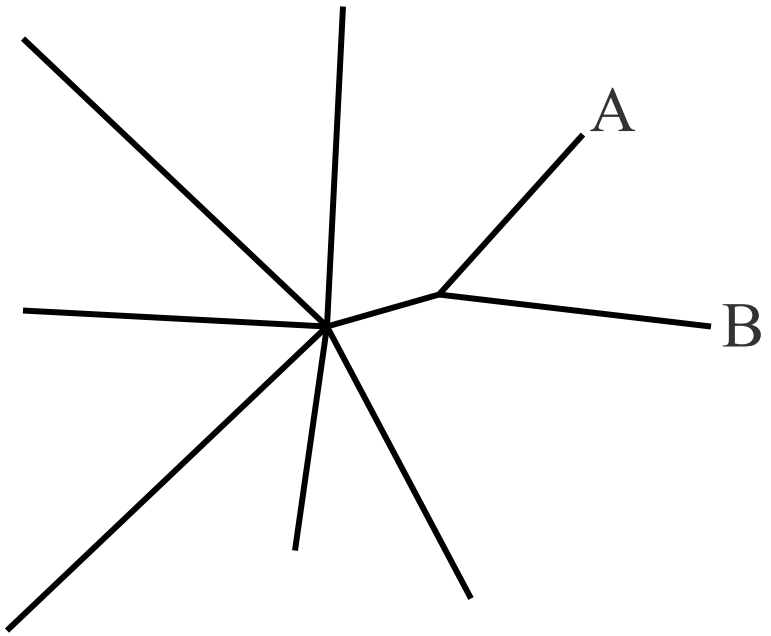
Укоренённое дерево строится «снизу вверх»

- 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.

Идея алгоритма neighbor-joining



A и B — такая пара последовательностей, для которых минимальна величина $(n - 2)d(A, B) - M(A,B)$, где n – число последовательностей, d — расстояние из матрицы, а $M(A,B) = \sum_C \big(d(A, C) + d(B, C)\big)$ (сумма по всем C, включая A и B).

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

Свойства алгоритма NJ

Время работы: $O(n^3)$, где n – число листьев.
Предложен очень близкий алгоритм с временем работы $O(n^2)$.

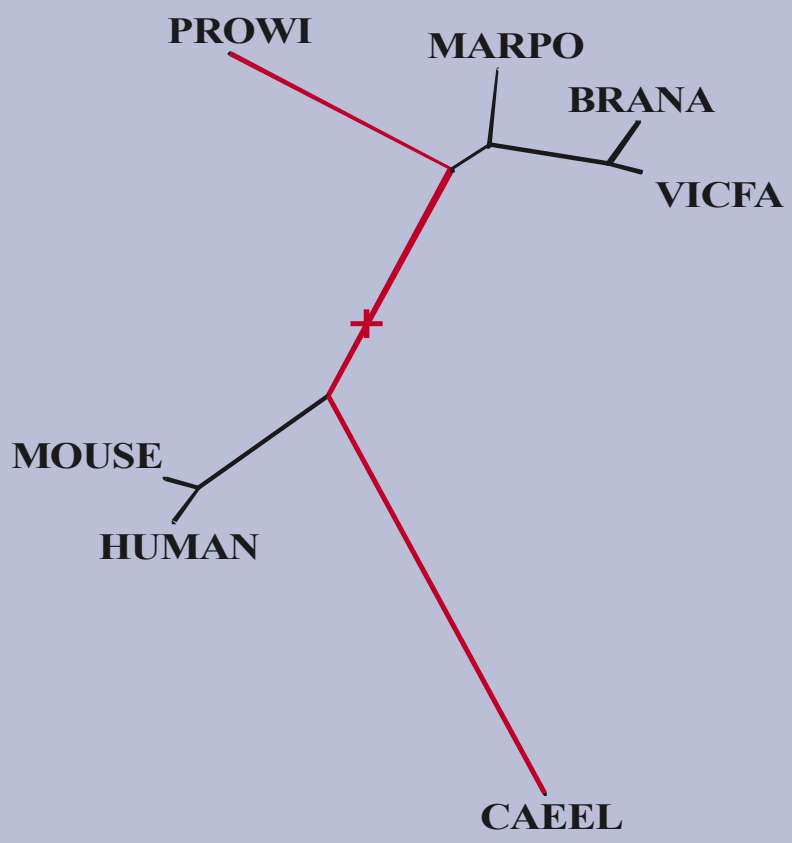
Если исходная матрица расстояний состояла из расстояний, вычисленных по некоторому дереву (то есть удовлетворяющих свойству аддитивности), то это дерево будет восстановлено.

Показывает хорошие результаты на практике.

Укоренение

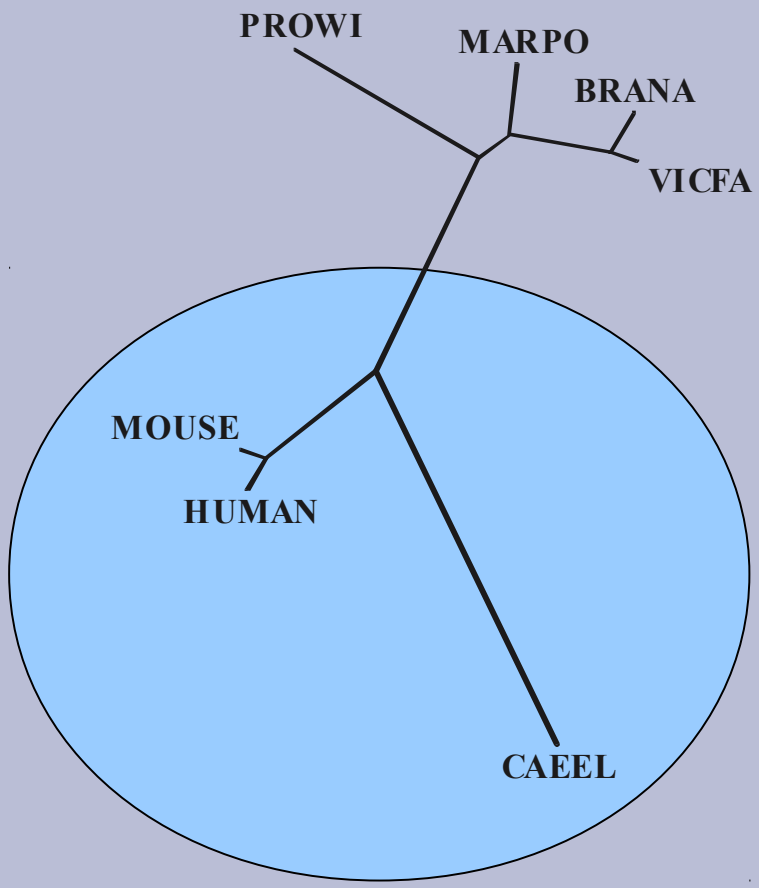
- В среднюю точку:

Находим на дереве самый длинный путь от листа к листу и за корень принимаем середину этого пути



Укоренение

- Используя внешнюю группу (outgroup):



В данном случае укоренено дерево четырёх растений, для чего пришлось построить дерево с участием внешней группы — трёх животных (в синем круге)

Сравнение деревьев

- Консенсусное (небинарное) дерево
- Максимальное общее поддерево
- Дерево из ветвей, поддержанных большинством (majority-rule tree)
- Меры сходства деревьев ("расстояние")

  i. Доля общих ветвей
  ii. Расстояние в "пространстве ветвей"
  iii. Доля общих четверок
  iv. Длина пути в пространстве деревьев

Бутстрэп-анализ

- создаём из входного выравнивания 100 «бутстрэп-реплик»;
- для каждой из реплик строим по дереву;
- из 100 деревьев строим дерево по методу большинства («majority-rule tree»).

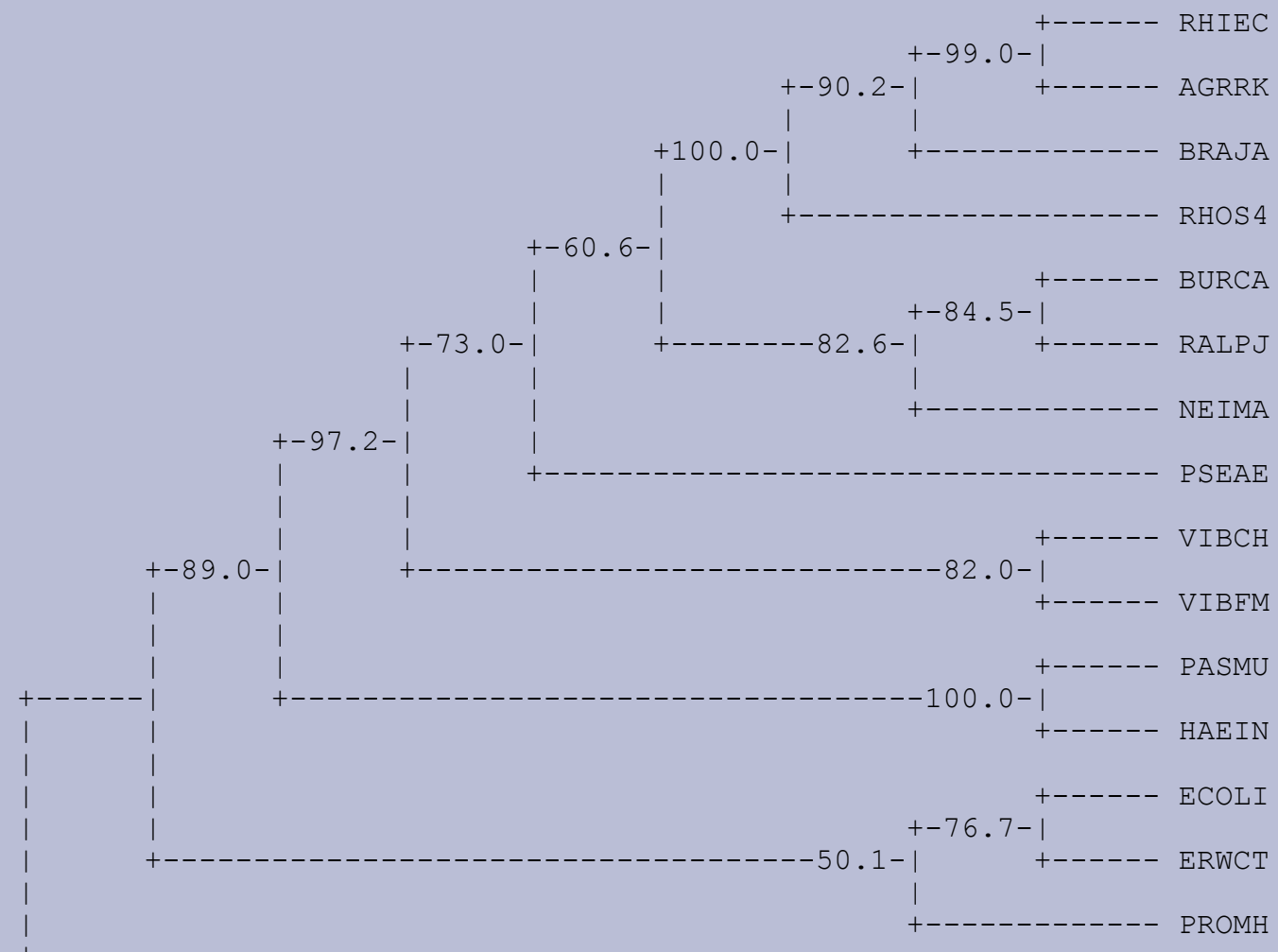
Помимо того, что (как правило) возрастает качество реконструкции, есть возможность оценить достоверность каждой ветви по т.н. «бутстрэп-поддержке», то есть проценту деревьев, в которых встретилась данная ветвь.

Бутстрэп-анализ

Каждая бутстрэп-реплика получается в результате случайного удаления половины столбцов из выравнивания с заменой их копиями других (тоже случайно выбранных) столбцов.

Смысл в том, чтобы построить дерево по половине данных и затем сравнить результаты от по разному выбранных половин.

Бутстрэп-анализ
(пример результата)



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).
  Имеется удобный веб-интерфейс:
  http://bioweb.pasteur.fr/phylogeny/intro-en.html
- 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)