Практическая работа 1.

Форматы данных используемых для филогенетического (и не только) анализа

Наиболее важный ньюанс при работе с любыми форматами данных касается выбора наименований для заголовков последовательностей и таксонов.

Общие советы следующие:

1. Название таксона (заголовка) должно быть:
   - максимально коротким (лучше – до 10 символов)
   - не содержать пробелов (они могут быть заменены на символ нижнего _ подчеркивания)
   - не содержать редкие символы (# , . : ; % ^ & * ? | ! () [] и т.д.)
2. С самого начала анализа создайте таблицу (например в excel), в которой в одной колонке вы будете хранить настоящие, расшифрованные названия таксонов и последовательностей, а в другой – коды, использованные в рабочих файлах. Например:

   ```
   Eriogonum fasciculatum, er_fasci

   Eriogonum umbellatum, er_umbel
   ```

   Перед публикацией и созданием рисунков к статье всегда можно будет заменить коды на реальные названия, но вы сэкономите много времени на поиск и разбор ошибок возникающих при неудачных наименованиях таксонов и заголовков.
3. Обратите внимание, что все форматы описанные ниже – текстовые, и следовательно они могут иметь любое арбитрарное расширение (например – файл фаста – *.fas, *.fa, *.fasta, файлип - *.phy, *.ph, *.phylip, нексус - *.nex, *.nexus). Эти арбитрарные расширения никакой роли не играют, важно лишь внутреннее форматирование данных и любое расширение будет понятно для программ. Остановимся подробнее на внутреннем устройстве каждого из форматов.

Fasta (фаста)

Формат фаста является наиболее распространенным форматом хранения нуклеотидных последовательностей. Фаста может содержать как выровненные (aligned), так и невыровненные (raw, unaligned) последовательности. Тем не менее, в большинстве случае в формате фаста хранят невыровненные исходные последовательности. Файл фаста может содержать сколько угодно большое число последовательностей различной длины.

Форма фаста прост и состоит из двух блоков:

(1) строки-заголовка, который должен начинаться символом >;

(2) следующей за ним строки или строк с нуклеотидными или аминокислотными последовательностями:

```
>Homo sapiens, cytochrome B
AACAGTTAGACGTGACTTAGGATAG
>Gorilla gorilla, cytochrome B (partial)
AAAACCGTTAAGGATTTATTTATTATACCCAG
```

Кроме того, данные в фаста-файле (и в некоторых других форматах – см. phylip ниже) могут быть представлены в двух видах:

- Sequential (последовательный)

```
>gi|161085638|dbj|AB305033.1|
ATATGCCTGAAAGTGGCGGACGGGTGAGTAACACGTGGGTGACCTGCCTCGGAGTGGGGGATAACCATGGGAAACTGCGG
CCAACGAGTAAAGCTTTAGTGCTTC...
 >gi|161085638|dbj|AB305644.1|
ATATGCCTGAAAGTGGCGGACGGGTGAGTAACACGTGGGTGACCTGCCTCGGAGTGGGGGATAACCATGGGGCTAATACC
GCATGGGCTTGTTGGCTTTGGCGGC...
```

- Interleaved (послойные)

```
>gi|161085638|dbj|AB305033.1|
ATATGCCTGAAAGTGGCGGACGGGTGAGTAACACGTGGGTGACCTGCCTCGGAGTGGGG
GAAACTGTGGCTAATACCGCATGGGCTTGTTGGCTTTGGCGGCCAACGAGTAAAGCTTT
AGGGGCCTGCGTCCGATTAGGTAGTTGGTGAGGTAATGGCTCACCAAGCCGATGATCGG
>gi|161085638|dbj|AB305644.1|
ATATGCCTGAAAGTGGCGGACGGGTGAGTAACACGTGGGTGACCTGCCTCGGAGTGGGG
GAAACTGTGGCTAATACCGCATGGGCTTGTTGGCTTTGGCGGCCAACGAGTAAAGCTTT
AGGGGCCTGCGTCCGATTAGGTAGTTGGTGAGGTAATGGCTCACCAAGCCGATGATCGG
```

Разница между ними в том, что в случае interleaved фаста последовательности превышающие некоторую фиксированную длину (например, 60 символов как в примере выше) разбиваются символом каретки (перенос на новую строку), а в случае sequential – записываются в одну длинную строку.

Phylip (файлип)

Формат phylip унаследовал свое название от исходной программы, для которой он был разработан (phylip, PHYLogeny Inference Package, автор Joseph Felsenstein). Этот формат используется для хранения выровненных нуклеотидных или аминокислотных последовательностей и имеет намного более строгие требования к форматированию входных данных чем фаста.
Ниже представлен пример фомата файлип:

```
      6     13
Archaeop  CGATGCTTAC CGCCGATGCT CGCCGAT
Hesperor  CGTTACTCGT TGTCGATGCT TGTCGAT
Baluchit  TAATGTTAAT TGTCGATGCT TGTCGAT
Bvirgini  TAATGTTCGT TGTCGATGCT TGTCGAT
Brontosa  CAAAACCCAT CATCGATGCT CATCGAT
Bsubtils  GGCAGCCAAT CACCGATGCT CACCGAT
```

Первая строка всегда начинается либо с символа пробела, либо с символа табуляции, за которым идет число, обзначающее количество нуклеотидных последовательностей в файле. Затем еще один пробел или табуляция, и длина нуклеотидной или аминокислотной последовательности. На следующей строке первым идет название последовательности длинной до 10 символов. Если название короче, в конце добавляются пробелы, чтобы общая длина строки до начала последовательности составляла 10 символов. На 11ой позиции начинается нуклеотидная последовательность. Она преставлена в блоках по 10 символов. Нужно добавить, что также как и фаста формат phylip может хранится либо в виде interleaved, либо в виде sequential, но в отличие от фаста файлов, конвертация между этим двумя подтипами нетривиальна.

В последнее время, в связи с широким использованием данного формата в филогенетических реконструкциях (например, программы phylip, phyml, paml, raxml) он был несколько упрощен (длина названий последовательностей может быть увеличена, число пробелов между видами и последовательностями также), но к сожалению, предсказать заранее, какая программа может без ошибок «читать» более упрощенный формат, а какая нет, сложно, поэтому основная рекомендация – стараться придерживаться строгой нотации форматов.

Nexus (нексус)

Это один из наиболее популярных форматов в филогенетике на данный момент. Он используется в таких программах как paup, macclade, mesquite, MrBayes, BEAST и R. Он удобен в силу своей легкой расширяемости под конкретные задачи. Особенность данного формата в том, что блоки информации определяются ключевыми словами-директивами, интерпретация которых реализована в конкретных программных пакетах. В самой базовой конфигурации файл в формате нексус выглядит следующим образом:

```
#NEXUS
begin data;
dimensions ntax=3 nchar=10;
format type=nucleotide missing=? gap=-;
matrix
Home_sapiens AACAGTTAGC
Gorilla_gorilla AATAGTTTTC
Pan_sp AACAGTTATC
;
end;
```

Сам файл начинается с директивы `#NEXUS`, означающей начало нексус-файла. Затем идет блок данных обозначенный директивой `begin data;` и заканчивающийся директивой `end;`. Обратите внимание на знак `;` в конце строк с директивами. На самом деле любой блок кода может быть запрограммирован и выделен ключевыми словами `begin <something>; … end;`, внутри данного блока будут описаны действия относящиеся к `<something>`. После директивы `begin data;` на новой строке в обязательно порядке указываются т.н. размерности данных – число последовательностей (`ntax`) и их длина (`nchar`), а также (опять же с новой строки) формат данны – например, нуклеотидная или аминокислотная последовательность, как именно кодируются пропущенные данные, как кодируются гэпы. Затем на следующей строке идет ключевое слово matrix и затем блок матрицы данных.

Обратите внимание, что если названия последовательностей содержат пробелы, то всё название нужно заключать в кавычки (например, `'Home sapiens'`). Важно, тем не менее, помнить, что некоторые программы все равно будут «ругаться» на такие пробелы и лучше избегать пробелов и необычных символов и в nexus файлах. Последовательности могут быть как interleaved так и

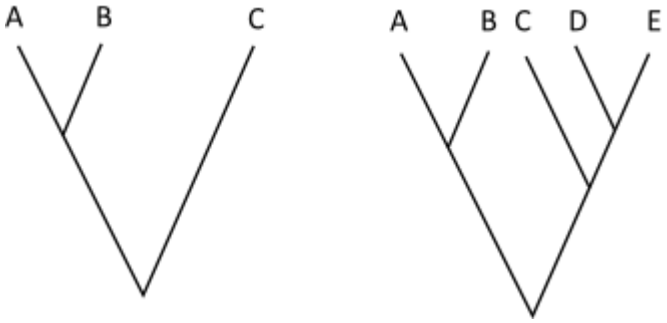
sequential. Если формат данных interleaved в строке format необходимо указать также `interleave=yes`.

Файл nexus может содержать не только нуклеотидные матрицы, но и филогенетические деревья, и блоки кода, и информацию о реконструкции биогеографических событий. Со всем этим мы будем знакомиться в течение практических занятий.

```
#NEXUS
begin trees;
   translate
           0 F_graminea,
           1 N_crassa,
           2 A_nidulans,
           3 S_cerevisiae,
           4 S_mikatae,
           5 S_paradoxus,
           6 S_bayanus,
           7 S_castellii,
           8 S_kluyveri,
           9 C_albicans,
           10 S_pombe,
           11 C_neoformans,
           12 M_grisea,
           13 C_elegans,
           14 D_melanogaster;
           tree No_001 =
((((((((((3:0.0130691766,5:0.0151973184):0.0022974291,4:0.0267966019):0.0
114844024,6:0.0296853757):0.0204440681,7:0.0465196055):0.0120969941,8:0.0
442292708):0.0209112159,9:0.1115276110):0.2651100835,10:0.1513436882):0.1
205270384,(((1:0.0662049379,0:0.0606875577):0.0155381574,12:0.1123581170)
:0.0360606937,2:0.0968434947):0.0236107824):0.0350977591,11:0.1712569743)
:0.1442959071,(14:0.1088280199,13:0.2266176629):0.1442959071);
           tree No_002 =
((((((((((3:0.0155205592,5:0.0127990548):0.0080254949,4:0.0233694460):0.0
097639217,6:0.0308958262):0.0277049990,7:0.0648142009):0.0179206608,8:0.0
428995908):0.0264746423,9:0.1098601223):0.2408391724,10:0.1199050581):0.1
222602939,(((1:0.0598463471,0:0.0709403415):0.0129094643,12:0.1216388258)
:0.0436879034,2:0.0893288737):0.0277157583):0.0379354529,11:0.1931459172)
:0.1114094552,(14:0.1720042840,13:0.2580361822):0.1114094552);
end;
```

Форматы хранения филогенетических деревьев

Иерархическая структура хранения деревьев наиболее часто используемая в филогенетических программах называет newick (ньюик) форматом. Разобраться с ней довольно просто взглянуть на картинки ниже.



В дереве слева, newick формат в первую очередь сгруппирует два наиболее родственных вида (A,B), а внешняя группа будет добавлена за скобоками ((A,B),C). На более сложном дереве справа, виды будут группироваться следующим образом: ((A,B), (C,(D, E))). Обратите внимание, что переставление порядка клад никоем образом не меняет топологию дерева и ((A,B), (C,(D, E))) = ((C,(D, E)) , (A,B)). Деревья с политомиями описываются в формате Newick просто как (A,B,C). Длина ветвей дерева добавляется после наименования таксона через двоеточие: ((A:0.5, B:0.5):0.3,C:0.8) . Дерево с корнем указывается имеет всего один наиболее внешний таксон, а дерево без корня – три корневых (внешних) таксона - (((A;B);C);D;E). В таких программах как phylip, paml, phyml хранение деревьев очень просто – в начале файла просто указывается число терминальных таксонов и число деревьев.

Например:

5 1
(((A,B),C),(D,E));

Практические задания:

1. Откройте файл ctenotus.in – в каком формате находятся данные? Это Interleaved или sequential формат? Сколько нуклеотидных последовательностей находится в файле?

2. Откройте файл ctenotus.phylip – сколько видов в данном файле? Какова длина последовательностей? Это Interleaved или sequential формат?

3. Откройте файл ctenotus.nexus – что находится в этом файле? Какую информацию мы можем получить о данных?