MCB 372

Phylogenetic reconstruction

PHYLIP

Peter Gogarten
Office: *BSP 404*
phone: *860 486-4061,*
Email: gogarten@uconn.edu

Terminology - reminder

Related terms:

autapomorphy = a derived character that is only present in one group; an autapomorphic character does not tell us anything about the relationship of the group that has this character ot other groups.

homoplasy = a derived character that was derived twice independently (convergent evolution). ***Note that the characters in question might still be homologous (e.g. a position in a sequence alignment, frontlimbs turned into wings in birds and bats).***

paraphyletic = a taxonomic group that is defined by a common ancestor, however, the common ancestor of this group also has decendants that do not belong to this taxonomic group. Many systematists despise paraphyletic groups (and consider them to be polyphyletic). Examples for paraphyletic groups are reptiles and protists. Many consider the archaea to be paraphyletic as well.

holophyletic = same as above, but the common ancestor gave rise only to members of the group.

Terminology- reminder

•Branches, splits, bipartitions
•In a rooted tree: clades
•Mono-, Para-, polyphyletic groups, cladists and a natural taxonomy

The term cladogram refers to a strictly bifurcating diagram, where each clade is defined by a common ancestor that only gives rise to members of this clade. I.e., a clade is monophyletic (derived from one ancestor) as opposed to polyphyletic (derived from many ancestors). (note you need to know where the root is!)

A clade is recognized and defined by shared derived characters (= synapomorphies). Shared primitive characters (= sympleisiomorphies , aternativie spelling is symplesiomorphies) do not define a clade. (see in class example drawing ala Hennig).

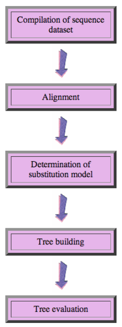
To use these terms you need to have polarized characters; for most molecular characters you don't know which state is primitive and which is derived (exceptions:....).

Steps of the phylogenetic analysis

Phylogenetic analysis is an inference of evolutionary relationships between organisms. Phylogenetics tries to answer the question "How did groups of organisms come into existence?"

Those relationships are usually represented by tree-like diagrams.

Note: the assumption of a tree-like process of evolution is controversial!


Compilation of sequence dataset
Alignment
Determination of substitution model
Tree building
Tree evaluation

Phylogenetic reconstruction - How

Distance analyses

calculate pairwise distances
(different distance measures, correction for multiple hits, correction for codon bias)

make distance matrix (table of pairwise corrected distances)

calculate tree from distance matrix

i) using optimality criterion
(e.g.: smallest error between distance matrix and distances in tree, or use
ii) algorithmic approaches (UPGMA or neighbor joining) B)

Phylogenetic reconstruction - How

Parsimony analyses

find that tree that explains sequence data with minimum number of substitutions
(tree includes hypothesis of sequence at each of the nodes)

Maximum Likelihood analyses

given a model for sequence evolution, find the tree that has the highest probability under this model.
This approach can also be used to successively refine the model.

Bayesian statistics use ML analyses to calculate posterior probabilities for trees, clades and evolutionary parameters. Especially MCMC approaches have become very popular in the last year, because they allow to estimate evolutionary parameters (e.g., which site in a virus protein is under positive selection), without assuming that one actually knows the "true" phylogeny.

Else:
spectral analyses, like evolutionary parsimony, look only at patterns of substitutions,

Another way to categorize methods of phylogenetic reconstruction is to ask if they are using

an optimality criterion (e.g.: smallest error between distance matrix and distances in tree, least number of steps, highest probability), or

algorithmic approaches (UPGMA or neighbor joining)

Packages and programs available: PHYLIP, phyml, MrBayes, Tree-Puzzle, PAUP*, clustalw, raxml, PhyloGenie, PyPhy

Phylip

written and distributed by Joe Felsenstein and collaborators (some of the following is copied from the PHYLIP homepage)

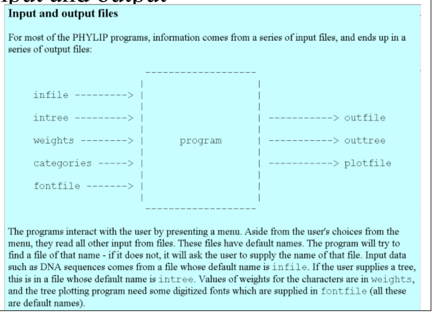
PHYLIP (the *PHYL*ogeny *I*nference *P*ackage) is a package of programs for inferring phylogenies (evolutionary trees).

PHYLIP is the most widely-distributed phylogeny package, and competes with PAUP* to be the one responsible for the largest number of published trees. PHYLIP has been in distribution since 1980, and has over 15,000 registered users.

Output is written onto special files with names like "outfile" and "outtree". Trees written onto "outtree" are in the Newick format, an informal standard agreed to in 1986 by authors of a number of major phylogeny packages.

Input is either provided via a file called "infile" or in response to a prompt.

input and output



Input and output files

For most of the PHYLIP programs, information comes from a series of input files, and ends up in a series of output files:

```
                    -------------------
                   |                   |
    infile --------->|                   |
                   |                   |
    intree --------->|                   | -----------> outfile
                   |                   |
    weights -------->|      program      | -----------> outtree
                   |                   |
    categories ----->|                   | -----------> plotfile
                   |                   |
    fontfile ------->|                   |
                   |                   |
                    -------------------
```

The programs interact with the user by presenting a menu. Aside from the user's choices from the menu, they read all other input from files. These files have default names. The program will try to find a file of that name - if it does not, it will ask the user to supply the name of that file. Input data such as DNA sequences comes from a file whose default name is `infile`. If the user supplies a tree, this is in a file whose default name is `intree`. Values of weights for the characters are in `weights`, and the tree plotting program need some digitized fonts which are supplied in `fontfile` (all these are default names).

What's in PHYLIP

Programs in PHYLIP allow to do parsimony, distance matrix, and likelihood methods, including bootstrapping and consensus trees. Data types that can be handled include molecular sequences, gene frequencies, restriction sites and fragments, distance matrices, and discrete characters.

Phylip works well with protein and nucleotide sequences
Many other programs mimic the style of PHYLIP programs.
(e.g. TREEPUZZLE, phyml, protml)

Many other packages use PHYIP programs in their inner workings (e.g., PHYLO_WIN)

PHYLIP runs under all operating systems

Web interfaces are available

Programs in PHYLIP are Modular

For example:

SEQBOOT take one set of aligned sequences and writes out a file containing bootstrap samples.

PROTDIST takes a aligned sequences (one or many sets) and calculates distance matices (one or many)

FITCH (or NEIGHBOR) calculate best fitting or neighbor joining trees from one or many distance matrices

CONSENSE takes many trees and returns a consensus tree

…. modules are available to draw trees as well, but often people use treeview or njplot

The Phylip Manual **is an excellent source of information.**

Brief one line descriptions of the programs are here

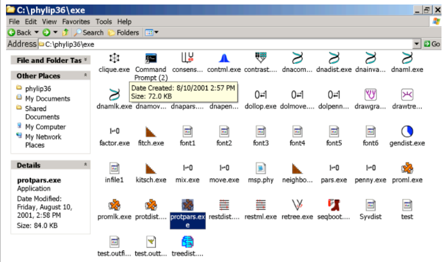
The easiest way to run PHYLIP programs is via a command line menu (similar to clustalw). The program is invoked through clicking on an icon, or by typing the program name at the command line.

```
> seqboot
> protpars
> fitch
```

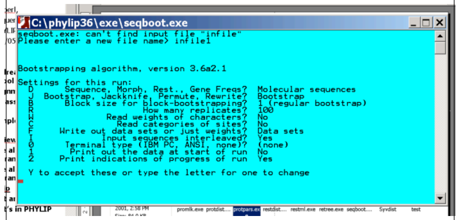
If there is no file called infile the program responds with:

```
[gogarten@carrot gogarten]$ seqboot
seqboot: can't find input file "infile"
Please enter a new file name>
```

program folder

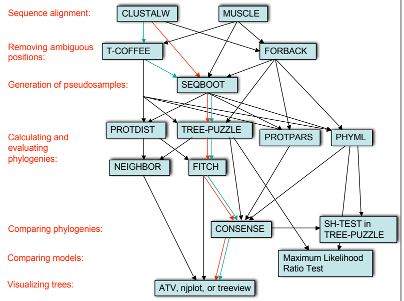


menu interface



```
seqboot.exe: can't find input file "infile"
Please enter a new file name> infile1


Bootstrapping algorithm, version 3.6a2.1

Settings for this run:
  D       Sequence, Morph, Rest., Gene Freqs?  Molecular sequences
  J  Bootstrap, Jackknife, Permute, Rewrite?  Bootstrap
  B       Block size for block-bootstrapping?  1 (regular bootstrap)
  R                     How many replicates?  100
  W                Read weights of characters?  No
  C                 Read categories of sites?  No
  F     Write out data sets or just weights?  Data sets
  I               Input sequences interleaved?  Yes
  0      Terminal type (IBM PC, ANSI, none)?  (none)
  1       Print out the data at start of run  No
  2     Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
```

example: seqboot and protpars on infile1



Phylip programs can be combined in many different ways with one another and with programs that use the same file formats.

Example 1 Protpars

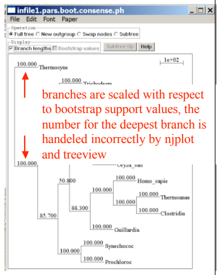
example: seqboot, protpars, consense on infile1

NOTE the bootstrap majority consensus tree does not necessarily have the same topology as the "best tree" from the original data!

threshold parsimony,
gap symbols - versus ?
(in vi you could use :`%s/-/?/g` to replace all – ?)
outfile
outtree compare to distance matrix analysis

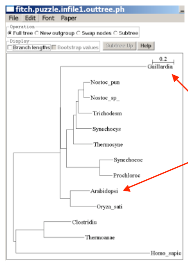
protpars (versus distance/FM)

```
Extended majority rule consensus tree

CONSENSUS TREE:
the numbers on the branches indicate the number
of times the partition of the species into the two sets
which are separated by that branch occurred
among the trees, out of 100.00 trees

                                            +------Prochloroc
                +---------------------100.-|
                |                           +------Synechococ
                |
                |            +--------------------Guillardia
         +-85.7-|            |
         |      |      +-88.3-|             +------Clostridiu
         |      |      |     |      +-100.-|
         |      |      |     +-100.-|       +------Thermoanae
         |      +-50.8-|            |
         |             |            +-------------Homo sapie
  +------|             |
  |      |             |                    +------Oryza sati
  |      |             +---------------100.0-|
  |      |                                  +------Arabidopsi
  |      |
  |      |               +--------------------Synechocys
  |      |               |
  |      +--------------53.0-|              +------Nostoc pun
  |                      |      +-99.5-|
  |                      +-38.5-|       +------Nostoc sp
  |                             |
  |                             +-------------Trichodesm
  |
  +---------------------------------------------Thermosyne


  remember: this is an unrooted tree!
```



branches are scaled with respect to bootstrap support values, the number for the deepest branch is handeled incorrectly by njplot and treeview

(protpars versus) distance/FM



Tree is scaled with respect to the estimated number of substitutions.

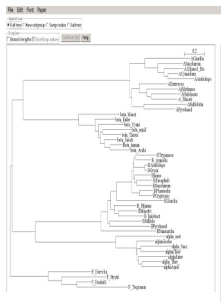
what might be the explanation for the red algae not grouping with the plants?

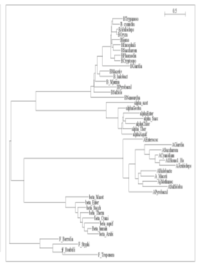
If time: demo of njplot

protdist

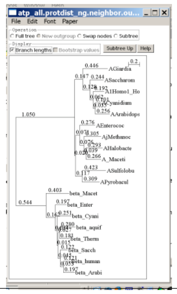
```
PROTdist
Settings for this run:
  P  Use JTT, PMB, PAM, Kimura, categories model?  Jones-Taylor-Thornton matrix
  G  Gamma distribution of rates among positions?  No
  C           One category of substitution rates?  Yes
  W                    Use weights for positions?  No
  M                   Analyze multiple data sets?  No
  I                  Input sequences interleaved?  Yes
  0                 Terminal type (IBM PC, ANSI)?  ANSI
  1            Print out the data at start of run  No
  2          Print indications of progress of run  Yes
```

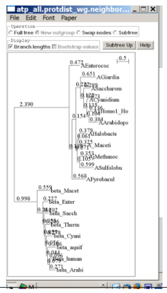
without and with correction for ASRV



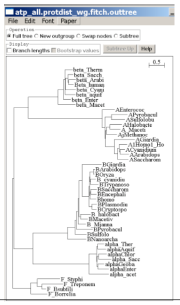


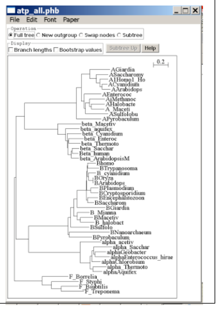
subtree with branch lengths
without and with correction for ASRV



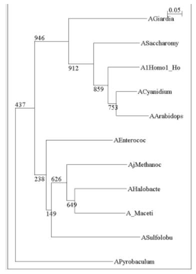


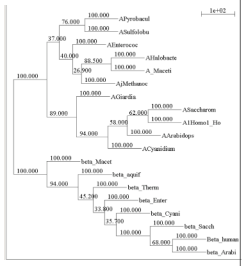
compare to trees with FITCH and clustalw – same dataset





bootstrap support ala clustal protpars (gaps as ?)





phyml

PHYML - A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood

An online interface is here ;
there is a command line version that is described here (not as straight forward as in clustalw);
a phylip like interface is automatically invoked, if you type “phyml” – the manual is here.

Phyml is installed on bbcxsrv1.

Do example on atp_all.phy
Note data type, bootstrap option within program, models for ASRV (pinvar and gamma), by default the starting tree is calculated via neighbor joining.

phyml - comments

Under some circumstances the consensus tree calculated by phyml is wrong. It is recommended to save all the individual trees and to also evaluate them with *consense* from the phylip package.
Note: phyml allows longer names, but consense allows only 10 characters!

phyml is fast enough to analyze dataset with hundreds of sequences (in 1990, a maximum likelihood analyses with 12 sequences (no ASRV) took several days).

For moderately sized datasets you can estimate branch support through a bootstrap analysis (it still might run several hours, but compared to protml or PAUP, this is extremely fast).

The paper describing phyml is here,
a brief interview with the authors is here

TreePuzzle ne PUZZLE

TREE-PUZZLE is a very versatile maximum likelihood program that is particularly useful to analyze protein sequences. The program was developed by Korbian Strimmer and Arnd von Haseler (then at the Univ. of Munich) and is maintained by von Haseler, Heiko A. Schmidt, and Martin Vingron

(contacts see http://www.tree-puzzle.de/).

TREE-PUZZLE

- allows fast and accurate estimation of ASRV (through estimating the shape parameter alpha) for both nucleotide and amino acid sequences,
- It has a “fast” algorithm to calculate trees through quartet puzzling (calculating ml trees for quartets of species and building the multispecies tree from the quartets).
- The program provides confidence numbers (puzzle support values), which tend to be smaller than bootstrap values (i.e. provide a more conservative estimate),
- the program calculates branch lengths and likelihood for user defined trees, which is great if you want to compare different tree topologies, or different models using the **maximum likelihood ratio test**.
- Branches which are not significantly supported are collapsed.
- TREE-PUZZLE runs on "all" platforms
- TREE-PUZZLE reads PHYLIP format, and communicates with the user in a way similar to the PHYLIP programs.

Maximum likelihood ratio test

If you want to compare two models of evolution (this includes the tree) given a data set, you can utilize the so-called maximum likelihood ratio test.
If L_1 and L_2 are the likelihoods of the two models, $d = 2(\log L_1 - \log L_2)$ approximately follows a Chi square distribution with n degrees of freedom. Usually n is the difference in model parameters. I.e., how many parameters are used to describe the substitution process and the tree. In particular n can be the difference in branches between two trees (one tree is more resolved than the other).
In principle, this test can only be applied if on model is a more refined version of the other. In the particular case, when you compare two trees, one calculated without assuming a clock, the other assuming a clock, the degrees of freedom are the number of OTUs – 2 (as all sequences end up in the present at the same level, their branches cannot be freely chosen) .

To calculate the probability you can use the CHISQUARE calculator for windows available from Paul Lewis.

TREE-PUZZLE allows (cont)

- TREEPUZZLE calculates distance matrices using the ml specified model. These can be used in FITCH or Neighbor.
PUZZLEBOOT automates this approach to do bootstrap analyses – WARNING: this is a distance matrix analyses!
The official script for PUZZLEBOOT is here – you need to create a command file (puzzle.cmds), and puzzle needs to be envocable through the command puzzle.
Your input file needs to be the renamed outfile from **seqboot**
A slightly modified working version of puzzleboot_mod.sh is here, and here is an example for puzzle.cmds . Read the instructions before you run this!
- Maximum likelihood mapping is an excellent way to assess the phylogenetic information contained in a dataset.
- ML mapping can be used to calculate the support around one branch.

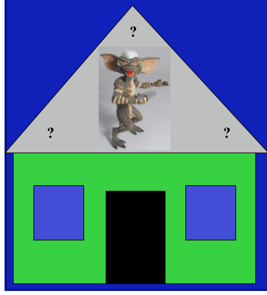
@@@ Puzzle is cool, don't leave home without it! @@@

TREE-PUZZLE – PROBLEMS/DRAWBACKS

- The more species you add the lower the support for individual branches. While this is true for all algorithms, in TREE-PUZZLE this can lead to completely unresolved trees with only a few handful of sequences.

- Trees calculated via quartet puzzling are usually not completely resolved, and they do not correspond to the ML-tree: The determined multi-species tree is not the tree with the highest likelihood, rather it is the tree whose topology is supported through ml-quartets, and the lengths of the resolved branches is determined through maximum likelihood.

Elliot Sober's Gremlins



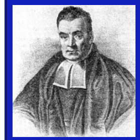
Observation: Loud noise in the attic

Hypothesis: *gremlins in the attic playing bowling*

Likelihood = *P(noise|gremlins in the attic)*

P(gremlins in the attic|noise)

Bayes' Theorem


Reverend Thomas Bayes (1702-1761)

$$\underbrace{P(\text{model}|\text{data}, I)}_{\text{Posterior Probability}} = \underbrace{P(\text{model}, I)}_{\text{Prior Probability}} \frac{\overbrace{P(\text{data}|\text{model}, I)}^{\text{Likelihood}}}{\underbrace{P(\text{data}, I)}_{\text{Normalizing constant}}}$$

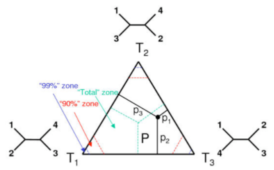
Likelihood: describes how well the model predicts the data

Posterior Probability: represents the degree to which we believe a given **model** accurately describes the situation given the available **data** and all of our prior information **I**

Prior Probability: describes the degree to which we believe the model accurately describes reality based on all of our prior information.

Normalizing constant

ml mapping



From: Olga Zhaxybayeva and J Peter Gogarten *BMC Genomics* 2002, 3:4

ml mapping

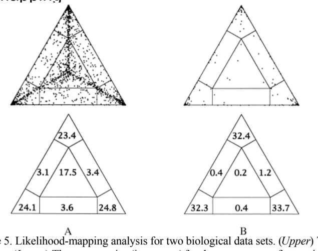
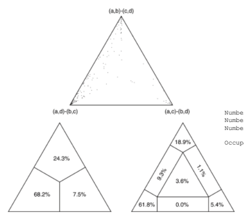


Figure 5. Likelihood-mapping analysis for two biological data sets. (*Upper*) The distribution patterns. (*Lower*) The occupancies (in percent) for the seven areas of attraction. (*A*) Cytochrome-*b* data from ref. 14. (*B*) Ribosomal DNA of major arthropod groups (15).

From: Korbinian Strimmer and Arndt von Haeseler Proc. Natl. Acad. Sci. USA Vol. 94, pp. 6815-6819, June 1997



```
Cluster a: 14 sequences
outgroup (prokaryotes)

Cluster b: 20 sequences
other Eukaryotes

Cluster c: 1 sequences
Plasmodium

Cluster d: 1 sequences
Giardia
```

```
              (a,b)-(c,d)
                  /\
                 /  \
                /    \
               /   1  \
              / \    / \
             /   \  /   \
            /     \/     \
           /  3    :   2  \
          /        :       \
         /__________________\
 (a,d)-(b,c)                  (a,c)-(b,d)

Number of quartets in region 1: 68 (= 24.3%)
Number of quartets in region 2: 21 (= 7.5%)
Number of quartets in region 3: 191 (= 68.2%)

Occupancies of the seven areas 1, 2, 3, 4, 5, 6, 7:

              (a,b)-(c,d)
                  /\
                 /  \
                / 1  \
               / \  / \
              /   /\   \
             / 6 /  \ 4 \
            /   / 7  \   \
           / \ /______\ / \
          / 3  :   5  :  2 \
         /__________________\
 (a,d)-(b,c)                  (a,c)-(b,d)

Number of quartets in region 1: 53 (= 18.9%)
Number of quartets in region 2: 15 (= 5.4%)
Number of quartets in region 3: 173 (= 61.8%)
Number of quartets in region 4: 3 (= 1.1%)
Number of quartets in region 5: 0 (= 0.0%)
Number of quartets in region 6: 26 (= 9.3%)
Number of quartets in region 7: 10 (= 3.6%)
```

Alternative Approaches to Estimate Posterior Probabilities

Bayesian Posterior Probability Mapping with MrBayes
(Huelsenbeck and Ronquist, 2001)

Problem:

Strimmer's formula $p_i = \frac{L_i}{L_1+L_2+L_3}$ only considers 3 trees (those that maximize the likelihood for the three topologies)

Solution:

Exploration of the tree space by sampling trees using a biased random walk (Implemented in MrBayes program)

Trees with higher likelihoods will be sampled more often

$p_i \approx \frac{N_i}{N_{total}}$,where N_i - number of sampled trees of topology *i*, *i*=1,2,3
N_{total} – total number of sampled trees (has to be large)

Illustration of a biased random walk

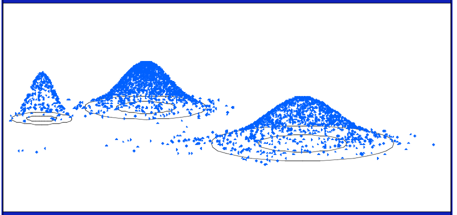
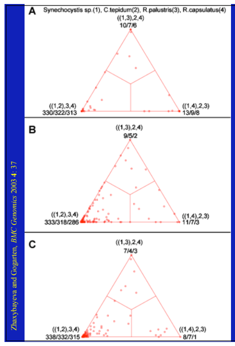


Figure generated using MCRobot program (Paul Lewis, 2001)



Zhaxybayeva and Gogarten, *BMC Genomics* 2003 4: 37

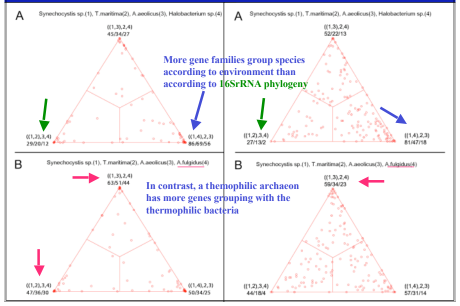
COMPARISON OF DIFFERENT SUPPORT MEASURES

A: mapping of posterior probabilities according to Strimmer and von Haeseler

B: mapping of bootstrap support values

C: mapping of bootstrap support values from extended datasets

ml-mapping *versus* bootstrap values from extended datasets



More gene families group species according to environment than according to 16SrRNA phylogeny

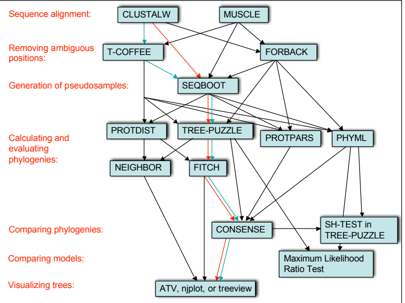
In contrast, a themophilic archaeon has more genes grouping with the thermophilic bacteria

puzzle example

archaea_euk.phy in puzzle_temp

usertree

check outfile



Phylip programs can be combined in many different ways with one another and with programs that use the same file formats.

Perl assignment

Write a script that takes all phylip formated aligned multiple sequence files present in a directory, and performes a bootstrap analyses using maximum parsimony.

Files you might want to use are A.fa, B.fa, alpha.fa, beta.fa, and atp_all.phy. BUT you first have to convert them to phylip format AND you should replace some or all gaps with ?
(In the end you would be able to answer the question “does the resolution increase if a more related subgroup is analyzed independent from an outgroup?)

hints

Rather than typing commands at the menu, you can write the responses that you would need to give via the keyboard into a file (e.g. your_input.txt)

You could start and execute the program protpars by typing

protpars < your_input.txt

your input.txt might contain the following lines:

```
infile1.txt
r
t
10
y
r
r
```

in the script you could use the line

```
system (“protpars < your_input.txt”);
```

The main problem are the owerwrite commands if the oufile and outtree files are already existing. You can either create these beforehand, or erase them by moving (mv) their contents somewhere else.

create *.phy files

the easiest (probably) is to run clustalw with the phylip option:
For example (here):

```
#!/usr/bin/perl -w
print "# This program aligns all multiple sequence files with names *.fa \n
# found in its directory using clustalw,  and saves them in phyip format.\n“;
while(defined($file=glob("*.fa"))){
        @parts=split(/\./,$file);
        $file=$parts[0];
        system("clustalw -infile=$file.fa -align -output=PHYLIP");
        };
# cleanup:
system ("rm *.dnd");
exit;
```

Alternatively, you could use a web version of readseq – this one worked great for me ☺

Alternative for entering the commands for the menu:

```
#!/usr/bin/perl -w
        system ("cp A.phy infile");
        system ("echo -e 'y\n9\n'|seqboot");
exit;
```

echo returns the string in ' ', i.e., y\n9\n.
The -e options allows the use of \n
The | symbol pipes the output from echo to seqboot

Old Assignments:

- **Read chapters 5 and 6**
- **Write a script that determines the number of elements in a %ash.**
- **Write a script (or subroutine) that prints out a hash sorted on the keys in alphabetical order.**
- **How can you remove an entry in a hash (key and value)?**
- **Write a program that it uses hashes to calculates mono-, di-, tri-, and quartet-nucleotide frequencies.**

exercises:

Write a script that determines the number of elements in %ash.

```
@keys = keys(%ash); #assigns keys to an array
$number =@keys; # determines number of different keys (uses array in scalar context).
print "$number \n";
```

Write a script that prints out a hash sorted on the keys in alphabetical order.

```
@gi_names = sort(keys(%gi_hash)); # sorts key and assigns keys to an array
foreach (@gi_names){
   print "$_ occurred $gi_hash{$_} times\n";
   }
```

Remove an entry in a hash (key and value):

```
delete $gi_hash{$varaible_denoting_some_key};
```

From Perl in a Nutshell:

`sort`

`sort [code] list`

Sorts a list and returns the sorted list value. By default (without a code argument), it sorts in standard string comparison order (undefined values sorting before defined null strings, which sort before everything else). code, if given, may be the name of a subroutine or a code block (anonymous subroutine) that defines its own comparison mechanism for sorting elements of list. The routine must return to the sort function an integer less than, equal to, or greater than 0, depending on how the elements of the list are to be ordered.
(The handy <=> and cmp operators can be used to perform three-way numeric and string comparisons.)

The normal calling code for subroutines is bypassed, with the following effects: the subroutine may not be a recursive subroutine, and the two elements to be compared are passed into the subroutine as $a and $b, not via @_. The variables $a and $b are passed by reference, so don't modify them in the subroutine. ...

examples to sort a hash by value (followed by key)

```
  @sorted_by_value = sort { $gi_hash{$a} <=> $gi_hash{$b}} keys (%gi_hash);
```

or

```
  @sorted_by_value = sort by_value keys %gi_hash;
    sub by_value { $gi_hash{$a} <=> $gi_hash{$b}};
   # defines the order smaller befor larger (a before b)
```

or

```
  @sorted_by_value = sort by_value (keys (%gi_hash));
    sub by_value {
    $gi_hash{$a} <=> $gi_hash{$b}
    or
    $a <=> $b
#if the values are the same,
#then sort ascibethically (cmp) or numerically (<=>) on the keys
    } # defines the order smaller befor larger (a before b)
```

Sort Example

```
#!/usr/bin/perl -w

@oligos = qw/ AGTCC AGT GTAC AGGAGGAT AGAGG GAGCCCCA CCICC GA /;
@sorted = sort byLength @oligos;

sub byLength {
return (length($a) <=> length($b)); #needs to be $a and $b
}

print join("\n", @sorted),"\n";
```

Count Oligos; program is here: oligos.pl

```
#!/usr/bin/perl -w
##########INPUT Sequence, concatenated into a single string##########
#skip annotation lines in case of fasta. if multiple annotation lines, concatenate these too.
#
unless(@ARGV==1) {die "please provide name of the file in the command line!!\n";}
my$filename=$ARGV[0]; #takes filenname from input line
open(IN, "< $filename") or die "cannot open $filename:$!"; #assigns filehandle IN to filename or dies

my$seq=''; #assigns empty string
my$line='';
my$name='';
my@bases=(); #assigns empty list

while(defined($line=<IN>)){

        chomp($line);
        if ($line=~/^>/) {  #look for beginning of line starting with > (^ is an anchor for the beginning
                $name .= $line;
                }
        else {
                $seq .= $line ;
                }
}
#  clean up sequence
#  check for all CAPS, report non ATGCs, remove white spaces
#
$seq =~ tr/atgc/ATGC/; #translates all ATGC to upper case
$seq =~ s/\s//g;# substitutes all white spaces \s with nothing globally in $seq
```

```
#####################sequence to array
@bases=split(//,$seq);  #splits string into separate elements (bases)

$num_bases=@bases; #length of array

####################calculate GC content
$n=5;#if you want different n-mers change this number
%nlet=(); #reset things
$nmer='';

for ($i=0; $i<$num_bases+1-$n; $i++)  #go through @bases and form nlets of consecutive nucleotides
        {
        for ($k=0;$k<$n;$k+=1){ #joins $n consecutive nucleotides

                $nmer .= $bases[$i+$k];#form nlet
                }
        $nlet{$nmer} += 1; #increase nlet counter for one particular nlet by one
        $nmer='';

}
#print hash
@nlet_present = sort {$nlet{$b} <=> $nlet{$a} or $a cmp $b } (keys(%nlet));
foreach (@nlet_present){
        print "$_  occurred $nlet{$_} times\n";
};
```

ml mapping can asses the topology surrounding an individual branch :

E.g.: If we want to know if *Giardia lamblia* forms the deepest branch within the known eukaryotes, we can use ML mapping to address this problem.
To apply ml mapping we choose the "higher" eukaryotes as cluster a, another deep branching eukaryote (the one that competes against Giardia) as cluster b, Giardia as cluster c, and the outgroup as cluster d. For an example output see this sample ml-map.

An analysis of the carbamoyl phosphate synthetase domains with respect to the root of the tree of life is here.

ml mapping can assess the not necessarily treelike histories of genome

Application of ML mapping to comparative Genome analyses

see here for a comparison of different probability measures.
Fig. 3: outline of approach
Fig. 4: Example and comparison of different measures

see here for an approach that solves the problem of poor taxon sampling that is usually considered inherent with quartet analyses.
Fig. 2: The principle of "analyzing extended datasets to obtain embedded quartets"
Example next slides: