MCB 371/372

vi, perl,
Sequence alignment,
PHYLIP
4/6/05

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

vi

There are several good 5 minute tutorials that introduce vi.
If you don't know vi already, do one ASAP. **This** one should be entirely appropriate.

`vi filename` : opens filename in vi
Cursor movement : usually the arrow keys work, if not, the h j k l keys always work to move the curser
`x` : deletes text under the cursor
`i` or `a` open the edit mode either as insert under the cursor or append to the right of the cursor
`<ESCAPE>` to leave the edit mode, press the escape key
`ZZ` writes the file and quits
`:`starts a line for vi commands, e.g., `:wq` write the file and quits, `:q!` quits without writing changes
`/string <enter>` searches for string

As usual, you can type `man vi` at the command line to get more information

the dreaded end of line symbol

solution a: use vi on a unix system

solution b: figure out a way to translate files that works for you

In case you transfered a file from a PC:
sed s/.$// PCfile.txt > unixfile.txt
(this remove all character from the end of each line except the new line character.)

In case you created the file in the "mac" environment, you could use
tr '\r' '\n' < macfile.txt > unixfile.txt

(see http://kinemage.biochem.duke.edu/software/softdocs/ftptrouble.html#linefeeds for more info)
mac \r
PC \n\r
unix \n

assignment:

On Monday we used the perl script extract_lines.pl .

Modify the script so that it prints out an additional column that for each blasthit it alse gives the bitscore of the alignment divided by the alignment lengths.

Hints:
`chomp ($line);` removes the \n character at the end of $line
`$parts[i]` contains the (i+1)s entry of the line.
If $a[11] and $a[3] are variables in an array that contain numbers you can assign a new variable to the ratio using the command
`$ratio_of_values=$a[11]/$a[3];`
To print things in a line in the output separated by tabs and a new line symbol at the end you could say for example:
`print OUT "$line\t$parts[12]\t$ratio\n";`

Go over mod.pl in jpgogarten/blasttemp2/

new assignment

Assume that you have the following non-aligned multiple sequence files in a directory:

A.fa : vacuolar/archaeal ATPase catalytic subunits ;
B.fa : vacuolar/archaeal ATPase non-catalytic subunits;
alpha.fa : F-ATPases non-catalytic subunits,
beta.fa : F-ATPases catalytic subunits,
F.fa : ATPase involved in the assembly of the bacterial flagella.

Write a perl script that executes muscle or clustalw and
1) aligns the sequences within each file
2) successively calculates profile alignments between all aligned sequences.

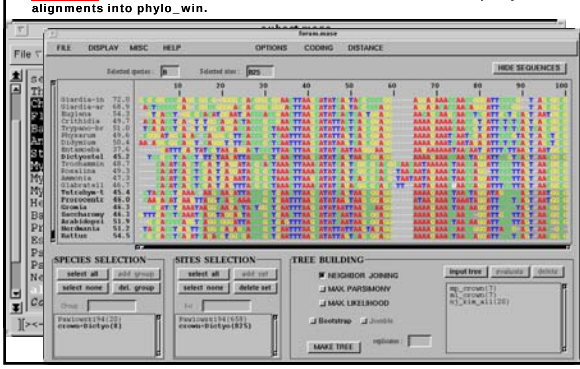
Hints:
`system (command)` executes "command"as if you had typed `command` in the command line

Example script

```
#!/usr/bin/perl
# tells shell that the interpreter to use is in /usr/bin/perl
# which perl at the command prompt tell you where perl is
# on your system.
#
# the following command checks for files that with names that
# fit *.fa, and for each executes the command in {}
#
while(defined($file=glob("*.fa"))){
#
# the next command executes a system command, in this case a
# blast search.  Note that perl substitutes the file name for
# the variable $file before it sends the "line" to the system.
# Note that this occurs twice once for –i and once for –o
system("blastall -i $file -d /matrix/db/13genomes.faa -p blastp -o
$file.blast -e 0.0000000000000000001 -I T -K 10 -m 8");
}
exit;
```

seaview – phylo_win

Another useful multiple alignment editor is **seaview, the companion sequence editor to phylo_win. It runs on PC and most unix flavors, and is the easiest way to get alignments into phylo_win.**



more alignment programs: statalign

statalign from Jeff Thorne deserves more attention than it receives. Especially for divergent sequences the initial pairwise alignment usually determines the ultimate result of the phylogenetic reconstruction.

Statalign solves this problem by not calculating a multiple sequence alignment, rather it spends a lot of computational power to calculate pairwise alignments and it extract distances (and their potential error) from these pairwise alignments and then uses these in a distance pased reconstruction. The errors from the individual distances are used to generate bootstrap samples for the distance matrices.

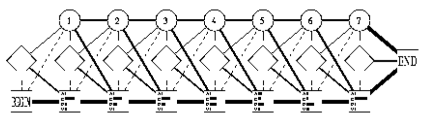
More at *Thorne JL, Kishino H (1992) Freeing phylogenies from artifacts of alignment. Mol Bio Evol 9:1148-1162*

statalign is available in several software archives (e.g. here), the readme file has plenty of information.

more alignment programs: SAM

SAM (sequence alignment and modeling system) by Richard Hughey, Anders Krogh, Christian Barrett, & Leslie Grate at UCSC.
http://www.cse.ucsc.edu/research/compbio/sam.html

The input consists of a multiple sequence file (aligned or not aligned) in FASTA format. The program uses secondary structure predictions, neighboring sites, etc. to place gaps. The program can be accessed through the www and run at UCSC



A linear hidden Markov model is a sequence of nodes, each corresponding to a column in a multiple alignment. In our HMMs, each node has a match state (square), insert state (diamond) and delete state (circle). Each sequence uses a series of these states to traverse the model from start to end. Using a match state indicates that the sequence has a character in that column, while using a delete state indicates that the sequence does not. Insert states allow sequences to have additional characters *between* columns. In many ways, these models correspond to profiles.

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

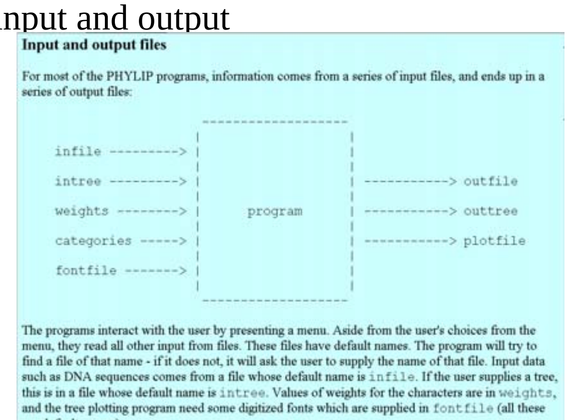
PHYLIP (the *PHYL*ogeny *I*nference *P*ackage) is a package of programs for inferring phylogenies (evolutionary trees).

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

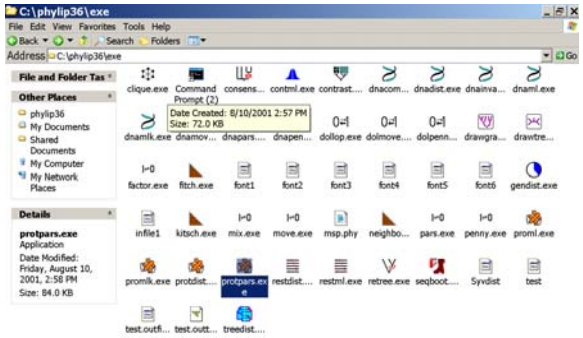
The easiest way to run PHYLIP programs is via a command line menu (similar to clustalw). The program is invoked through clicking on an icon, or by typing the program name at the command line.

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

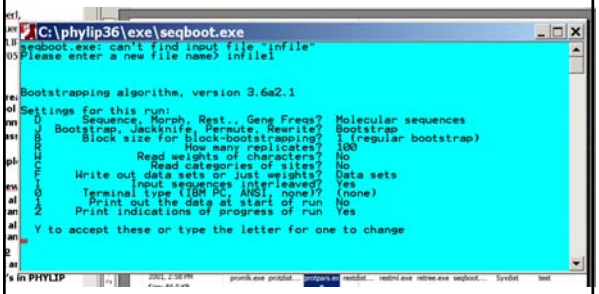
If there is no file called infile the program responds with:

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

program folder



menu interface



example: seqboot and protpars on infile1