Assignments from last week

Geneplot

In a perfect world you do not want to plot gi numbers but positions in a genome. The script addnumnuc.pl adds the nucleotide position of the ORF (the central one) to the beginning of the annotation line.
.ptt files

Available on the ftp server at NCBI or each chromosome. E.g.

Fervidobacterium nodosum Rt17-B1, complete genome - 1..1948941
1750 proteins

<table>
<thead>
<tr>
<th>Location</th>
<th>Strand</th>
<th>Length</th>
<th>PID</th>
<th>Gene</th>
<th>Synonym</th>
<th>Code</th>
<th>COG</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>43..1377</td>
<td>+</td>
<td>444</td>
<td>154248706</td>
<td>-</td>
<td>Fnod_0001</td>
<td>-</td>
<td>-</td>
<td>chromosomal replication initiator protein DnaA</td>
</tr>
<tr>
<td>1453..1635</td>
<td>+</td>
<td>60</td>
<td>154248707</td>
<td>-</td>
<td>Fnod_0002</td>
<td>-</td>
<td>-</td>
<td>4Fe-4S ferredoxin iron-sulfur binding domain protein</td>
</tr>
<tr>
<td>1976..3829</td>
<td>+</td>
<td>617</td>
<td>154248708</td>
<td>-</td>
<td>Fnod_0003</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
<tr>
<td>3826..4926</td>
<td>+</td>
<td>366</td>
<td>154248709</td>
<td>-</td>
<td>Fnod_0004</td>
<td>-</td>
<td>-</td>
<td>ABC transporter related translocator</td>
</tr>
<tr>
<td>5136..6701</td>
<td>+</td>
<td>521</td>
<td>154248710</td>
<td>-</td>
<td>Fnod_0005</td>
<td>-</td>
<td>-</td>
<td>inner-membrane translocator</td>
</tr>
<tr>
<td>6698..7732</td>
<td>+</td>
<td>344</td>
<td>154248711</td>
<td>-</td>
<td>Fnod_0006</td>
<td>-</td>
<td>-</td>
<td>inner-membrane translocator</td>
</tr>
<tr>
<td>7729..8688</td>
<td>+</td>
<td>319</td>
<td>154248712</td>
<td>-</td>
<td>Fnod_0007</td>
<td>-</td>
<td>-</td>
<td>ABC transporter related translocator</td>
</tr>
<tr>
<td>8734..9132</td>
<td>+</td>
<td>132</td>
<td>154248713</td>
<td>-</td>
<td>Fnod_0008</td>
<td>-</td>
<td>-</td>
<td>protein of unknown function</td>
</tr>
<tr>
<td>9261..9617</td>
<td>+</td>
<td>118</td>
<td>154248714</td>
<td>-</td>
<td>Fnod_0009</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
<tr>
<td>9745..10020</td>
<td>+</td>
<td>91</td>
<td>154248715</td>
<td>-</td>
<td>Fnod_0010</td>
<td>-</td>
<td>-</td>
<td>histone family protein</td>
</tr>
<tr>
<td>10098..11342</td>
<td>-</td>
<td>414</td>
<td>154248716</td>
<td>-</td>
<td>Fnod_0011</td>
<td>-</td>
<td>-</td>
<td>metal dependent phosphohydrolase</td>
</tr>
<tr>
<td>11361..13514</td>
<td>-</td>
<td>717</td>
<td>154248717</td>
<td>-</td>
<td>Fnod_0012</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
<tr>
<td>13511..14161</td>
<td>-</td>
<td>216</td>
<td>154248718</td>
<td>-</td>
<td>Fnod_0013</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
<tr>
<td>14158..15102</td>
<td>-</td>
<td>314</td>
<td>154248719</td>
<td>-</td>
<td>Fnod_0014</td>
<td>-</td>
<td>-</td>
<td>putative membrane protein</td>
</tr>
<tr>
<td>15115..15969</td>
<td>-</td>
<td>284</td>
<td>154248720</td>
<td>-</td>
<td>Fnod_0015</td>
<td>-</td>
<td>-</td>
<td>putative adenylate/guanulate cyclase</td>
</tr>
<tr>
<td>16022..17008</td>
<td>-</td>
<td>328</td>
<td>154248721</td>
<td>-</td>
<td>Fnod_0016</td>
<td>-</td>
<td>-</td>
<td>putative ClpP family protein</td>
</tr>
<tr>
<td>17156..17566</td>
<td>+</td>
<td>136</td>
<td>154248722</td>
<td>-</td>
<td>Fnod_0017</td>
<td>-</td>
<td>-</td>
<td>protein of unknown function</td>
</tr>
<tr>
<td>17594..19282</td>
<td>+</td>
<td>562</td>
<td>154248723</td>
<td>-</td>
<td>Fnod_0018</td>
<td>-</td>
<td>-</td>
<td>sigma54 specific transcriptional regulator</td>
</tr>
<tr>
<td>19623..19859</td>
<td>+</td>
<td>78</td>
<td>154248724</td>
<td>-</td>
<td>Fnod_0019</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
<tr>
<td>19856..20074</td>
<td>+</td>
<td>72</td>
<td>154248725</td>
<td>-</td>
<td>Fnod_0020</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
<tr>
<td>20095..20289</td>
<td>+</td>
<td>64</td>
<td>154248726</td>
<td>-</td>
<td>Fnod_0021</td>
<td>-</td>
<td>-</td>
<td>hypothetical protein</td>
</tr>
</tbody>
</table>
#!/usr/bin/perl -w

decided to have input file entered in command line
call program followed by genome name.
the program assumes that a file with the extensions ptt and faa exist in the same directory.
#####INPUT Name of multiple seq file containing ORF of genome, open file and assign IN filehandle
unless(@ARGV==1) {die "please provide genome name in command line 
file should contain multiple sequences in fasta format \n a file with the ptt table should be in the same directory\n\n";}
#$num=0;
$filename=$ARGV[0];
@nameparts=split(\./., $filename);
# print $parts[0];
$orf="$nameparts[0]"."\.faa";
$ptt="$nameparts[0]"."\.ptt";

open(IN, "< $ptt") or die "cannot open $ptt:$!";
$line=<IN>; # read fist line
  if ($line=~/complete genome/) { #look for header
    print "$line\n";};
$line=<IN>; # read 2nd line
print "$line\n";
$line=<IN>; # read 3rd line
if ($line=~/Location Strand/) { #look for beginning of table
  while (defined ($line=<IN>)) { # read through rest of table line by line
    @parts=split(\t,$line;
    @fromto=split(\./., $parts[0];
    $middle = (($fromto[1]+$fromto[0])/2);
    print "$fromto[1]\t$fromto[0]\t$middle\t$parts[3]\t";
    $gi_hash{$parts[3]}=$middle;
    print "\n";
  }
@gi_names = sort(keys($gi_hash));
$total=scalar(@gi_names);

print "total number of GIs= $total\n";
foreach (@gi_names) {
  print "gi number $_ is located at $gi_hash{$_}\n";
}
close(IN);
Addnumnuc.pl part 2

# read in and process faa file

open(IN, "< $orfs") or die "cannot open $filename:$!");
$outfilename = "$nameparts[0]"."\num\faa";
open(OUT, "> $outfilename") || die "cannot open: ";

while (defined ($line<IN>)) { # read through file line by line

  if ($line=~/\>/) { # look for beginning of line starting with > (^ is an anchor for the beginning of the line)
    $line =~ m/gi\/(\d+)/; # match gi\number capture number in $1
    $num=$gi_hash{$1};
    $line =~ s/^//;
    # print "$1 $num \n"
    $line = "">".$num\t" $line"
  }

  print "$line"; #print to screen
  print OUT "$line"; #print to OUT


close(IN);
close(OUT);

Results in a multiple fasta file where each annotation line starts with the nucleotide position in the chromosome:
>385.5 gi|15642776|ref|NP_227817.1| hypothetical protein TM0001 [Thermotoga maritima MSB8]
MryGKEGYGRSKNILLSECVCGIISELNGFQYFLRGMETL

>545.5 gi|15642777|ref|NP_227818.1| hypothetical protein TM0002 [Thermotoga maritima MSB8]
MYPEDWKRLICFHTSKEVLKQTLDDAQQNISDSVSIPLRKY

>1828 gi|15642778|ref|NP_227819.1| hypothetical protein TM0003 [Thermotoga maritima MSB8]
METVKAYEVEDIPAIGFNNSLEVWKLFPASSRSTSSSFQ

>1974.5 gi|15642779|ref|NP_227820.1| hypothetical protein TM0004 [Thermotoga maritima MSB8]
MKDLYERFNSLEVWKLVELFSGTSIRIHLFQ

>4131 gi|15642780|ref|NP_227821.1| DNA helicase, putative [Thermotoga maritima MSB8]
MTVOQFIIKLVLWELERNAEINAMLDEMKRLSVEEREKGRAVGLTGFIGEEELGYFLRFGRRKKID
TEIGVGDVLISKGNPLKSDYTGTVEKEKGERFITVAVRDLPSWLKNVRIDLFAIDITFRRQIEHLMTLS
SEGKKALEFIIGLGRKPEESFEEFPTFDEGNESQREAVSLARLSSDFLHGPFGTGTKRTLVEYIRQ	
VARGK KlLVT ASEN LAVDLVERLGVSRGHPSRVSISHLKELSTLAHQIESSEAYEKKMKEELAK
LIKRRDSFTKPSQWRGLSDKKILEYAEKNWSARGVSEKEKiEKEMAWEIKLSQIQDIRDLIERKEELIA
SRIVREAQVULSTNSAAELGIVFDVVVDASQATIPSILIPISKGKFKVLAGDHKQLPPTISED
AKDLSRTLFEELITRYPEKSSLLDTQYRMNELLMEFPSEEFYDGKIAEKVRNITLFDLGVEIPNGKF
WDVVLSDKPVNLVFIDTNRSDFRKRDSPSRENPLEAQIVKEVEKLLSMGVKEDWIGIITPYDDQVN
LIRELIEAKVEVHSVDGFQGREKEVIISFVRSNKNGEIGFLEDLRLNVSLTRAKRKLATGDSSTLSV
HPTYRFFVEFVKKGTYVIF

.............
Geneplot using EXCEL part 1

Format databank using Tpet.num.faa
>formatdb -i Tpet.num.faa -p T -o T

Search databank using Tmar.num.faa using blastall with -m8
> blastall -p blastp -d Tlet.num.faa -i Tmar.num.faa -o Tlet_Tmar.tab -F F -m 8 -W 2 -a 2 -e 0.001
You could use different E values

Load output (in this case Tlet_Tmar.tab) into Excel;
(note the script addnumnuc added an extra tab)
Geneplot using EXCEL  part 2

Plotting column B against A  ->
To only plot the top scoring hits use extract_lines.pl -->
Geneplot using EXCEL part 4

Plotting Tpet_Tmar.tab.top
PSIBlast to find transposase homologs

• Download transposase sequence transposase.fa
• Download genome as nucleotide sequence
• Format genome

- formatdb -i Tpet.fna -p F -o T
- blastpgp -i transposase.fa -d nr -l T -h 0.00001 -j 6 -C transposase.chk -a2
- blastall -i transposase.fa -d Tpet.fna -p psitblastn -R transposase.chk -o transposase_Tpet.tab -a2 -m8 -F F

transposase_Tpet.tab:
OLD ASSIGNMENTS
Write a script that reads in a sequence and prints out the reverse complement. Modify your script to that it can handle a sequence that goes over several lines.

• Background: `$comp =~ tr/ATGC/TACG/;` #translates every A in $comp into a T; every T into an A; every G into a C and every C into a G

• Read P 14 on hashes, write the program suggested in the chapter.
• Write a script reads in a sequence and prints out the reverse complement.
• Modify your script to that it can handle a sequence that goes over several lines?

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

#input sequence; chomp every line, and concatenate into one big scalar called $seq
unless (@ARGV == 1) {die "please provide name of the file in the command line!!!\n";}
$filename = $ARGV[0];
open(IN, "< $filename") or die "cannot open $filename:$!";

$seq = "";
while (defined($line=<IN>)) {
    chomp($line);
    $seq .= $line;
}

#Calculate reverse complement

$rev = reverse ($seq);
$rev_comp = $rev;
$rev_comp =~ tr/atgcATGC/TACGTACG/;

print "\n\n\nthe reverse complement of \n$f filename : \n$rev_comp\n"; #print output
```

Go through class4Answers.pl
Go through sort_example1.pl and sort_example2.pl
Do the following statements evaluate to true or false? (Check P5)

- 1
- 0 && 1
- 0 || 1
- 45
- 45 - 45
- 45 / 45
- 45 == 45
- 45 <= 45
- 45 <= 50
- 55 >= 50
- 50 <= 70
- 45 != 45
- 45 != 50

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equal to</td>
<td>if ($x == $y)</td>
</tr>
<tr>
<td>!=</td>
<td>not equal to</td>
<td>if ($x != $y)</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>if ($x &gt; $y)</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>if ($x &lt; $y)</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
<td>if ($x &gt;= $y)</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
<td>if ($x &lt;= $y)</td>
</tr>
<tr>
<td>&lt;=&gt;</td>
<td>comparison</td>
<td>if ($x &lt;=&gt; $y)</td>
</tr>
</tbody>
</table>

from [http://korflab.ucdavis.edu/Unix_and_Perl/unix_and_perl_v2.3.3.pdf](http://korflab.ucdavis.edu/Unix_and_Perl/unix_and_perl_v2.3.3.pdf)
True or False?

```perl
#!/usr/bin/perl -w
my @array = qw (1 0&1 0|1 105 45-45 45/45 45==45 $a=45 45<>45 45<=50 55>=50 50<>70 45!=45 45!=50);
# last line reads in all the expressions to be tested into an array
foreach (@array) {
    # this loop tests each of the expressions
    # eval($_) causes the execution/evaluation of the string stored in $_
    if(eval($_)){
        print "\$_ is true \n"
    } else {
        print "\$_ is false \n"
    }
}
```
NEW ASSIGNMENTS

Read through **P20. Functions** (subroutines)

Turn your script that calculates the reverse complement of a sequence into a subroutine

Write a script that takes all files with the extension .fa (containing a single fasta formatted sequence) and writes their contents in a single multiple sequence file.

Read through class5.pl
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 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
- Muscle homepage is at [http://www.drive5.com/muscle/docs.htm](http://www.drive5.com/muscle/docs.htm)
- Sequence files are also in script folder.
Alignments can be global or local. BLAST calculates **local** alignments, for databank searches and to find pairwise similarities local alignments are preferred.

Example using [bl2seq](#) with GIs: 137464 *versus* 6319974 and 137464 *versus* 254565713

However, for multiple sequences to be used in phylogenetic reconstruction, global alignments are the usual choice.

We will use two programs: MUSCLE and CLUSTALW

**Note:** Multiple alignments are more accurate than pairwise alignments! (see Fig 12.2. in Higgs and Atwood). The more sequences one includes, the more reliable the result. Same for phylogenetic reconstruction (taxon sampling).
dotlet

The Swiss Institute for Bioinformatics provides a JAVA applet that performs interactive dot plots. It is called Dotlet. The main use of dot plots is to detect domains, duplications, insertions, deletions, and, if you work at the DNA level, inversions (excellent illustrations of the use of dot plots are given on the examples page).

One application of this program is to find internal duplications and to locate exons.

Example: this sequence against itself (if time do in bl2seq as well) genomic sequence against Protein

As similar result can be obtained using blastx against a protein databank
The Needlemann Wunsch Algorithm

a step by step illustration is here

a) fill in scoring matrix
b) calculate max. possible score for each field
c) trace back alignment through matrix

NOTE that clustalw and other multiple sequence alignment programs do NOT necessarily find an alignment that is optimal by any given criterion.

Even if an alignment is optimal (like in the Needleman-Wunsch algorithm), it usually is not UNIQUE. It often is a good idea to take different extreme pathways through the alignment matrix, or to use a program like tcoffee that uses many different alignment programs.
**clustalw** runs on all possible platforms (unix, mac, pc), and it is part of most multiprogram packages, and it is also available via different web interfaces. Examples: [here](#), and [here](#).

Clustalw uses a very simple menu driven command-line interface, and you also can run it from the command line only (i.e., it is easy to incorporate into scripts for repeated analyses – to get info on the commandline options type clustalw –options and clustalw -help.)

**Clustalx** uses the same algorithms as clustalw. However, it has a much nicer interface, it displays information on the level of similarity, and it uses color in the alignment. Especially for amino acids the use of color greatly enhances the ability to recognize conservative replacements. Clustalx is available for different platforms at the [ebi's ftp](#) site (follow your platform, clustalx is stored in the clustalw folders)

Clustal reads and writes most formats used by different programs. The easiest format is the FASTA format:

---


To align sequences clustal performs the following steps:

1) Pairwise distance calculation
2) Clustering analysis of the sequences
3) Iterated alignment of two most similar sequences or groups of sequences.

It is important to realize that the second step is the most important. The relationships found here will create a serious bias in the final alignment. The better your guide tree, the better your final alignment.

You can load a guide tree into clustal. This tree will then be used instead of the neighbor joining tree calculated by clustalw as a default. (The guide tree needs to be in normal parenthesis notation WITH branch lengths).

Sample input file  Sample output file
clustal

Sample input

> Acetabularia acetabulum gi|1303673|gn1|PID|d1009732 adenosine triphosphatase A subunit
MSKAKEGDYGSIKKVSGPVPVVDAMGCGSMYELVRVGTGLIEREGMTATIQYVEETSGLTVGDGV
LRKQPLSVDLGPILGNIFDGIQPRKPIAIADVSDGFVFIPRGNVPSDLQTWKFEPFRPSAFLKVGDRVTTG
DIICINSPLLDSHKVMLPQAKGTVTYYIAAPGNTINEKIEIVEFQGAQKYEYSMKQSWPVSPRPVVEK
LLADTPPLTQGRVLDLSFPGVGRGCTAIPAGFAGCGKTIVSQALSKYSNSDGIVYVGCGERGNEMAEVLMD
FPQLTMTMPGREGESIMKRTTLVANTSVPNMPVAARESIYGTILSEYFRDMGYFAMADSTSRWAUNLR
EISGRLAEMPADSGPAYLGARLASFYERSGVRACIGSPEREQGSTVTQAGSFPGGDSDFPVTSATLGV
QVFWGDLKHAQKRHFPSSVNWLISYSKYNLANEPFYEKFDSDVFVLRTQVAREVLQKEDELNEIVQLVGKD
ALAEASDKIELATFLKEDYLQNSSTKYDPIFYKSVGMMRNIVTFHRLATQAERTAAGNVGQKIT
FNIIKAKALCDLLYKVSNQFEDPSDGECGVVTADNLNEELKEKFALEDEYR

>Drosophila melanogaster gi|1373433 vacuolar ATPase subunit A
MSNLKRFDEERESKYCRGFVAVSVPVVTAEAMSAGSAMYELVRVGGYELGEIIRLECDMATIQYVEETS
VTVGDPVLRRTGKLPSVELPGCPGQG

>Saccharomyces cerevisiae gi|137464|sp|P17255|VATA_YEAST VACUOLAR ATP SYNTHASE CATALYTIC SUBUNIT
MAGAIENARKEIKRISLEDHAESEGYAIYSGSVPVVIAEMICCAMYELVKGHDNLVGEVIRIDGKAT
IQYVEETAGLTVDPVLRTGKPLSVELPGPMLETIYDGIQRPLKAIKESESQSIYPRGIDTPALRDTIKW
QFTPGKQFGDHISGDDIYGSVFENLSSLKILLPRRSRTGRTTIWAPIAGEYTLDEKILEVFDGKKSDF
TLHWTWPVVRPRPVTEKLSADYPLLTTGQRVLDALFCPCVQQGETCIPAGFAGCGKTIVSQALSKYSNSDAII
YVGCFAKGTNVLMADESIECENIEVNGKVMKGDPRVEIKLPRGRETMSYVQKISQHRAIKSDSSREV
PELLKFTCNATHELVRPRTPSRRLSRTIKGVEYEFITFEEMGQKKAPGRIVELKVEKSYSPISGPE
RANELVESYRKASNKNAYFEWTEARDLSLGLSSHVRKATYQTYAPILYENHDFFDYMQKSKFLHTIGPKV
LAYLGLWIGDGLSRAFSVDSDSRTSMERYTEAYEKLNLCAEDYKDRKPEPQAVTYNLYSVKVRNgIR
NNLNENTLPWAIAVLGFLPKDGKNIFSPLSTDNITFELAGLDSDGYVTDHGHIATIKTIHTSRSV
DGLVSLARSLGLVVSVNAEPAKUVDNMGTHKHSAYIYMSGDVDLLNVLCSACGSKKFRPAPAAPAFARECR
GFYFELQELKEDDYGYITLSDSDHQP LLANQVQYVHCNGERGNEMAEVLMEFPELYTEMSTGKEPMIMKRT
TLVANTSVPNMPVAARESIYGTILAIYFRDQCGKNMADSSRSWAELREISGRLEMPADQGFPAYLG
AKLASFYERAGKAVALGSPPDTRGSVSIIVAASPAGDFSPVTTATLGICTQVFNWGLDKCLAQRRKLFPSIN
TSSYSYKYNVLNSKFDNSYEPFPVRDLRMRKELSNAAELEQVQLVGKSALSDDSKTLDDLVATLIKEDF
LQQNGYSTYDAFCIPIKFTFDMRFAFISYHDEAQKAVANGANWSKLDSTGDVKHAVSSSKFEPSPRGKE
VHGEFEKLLSTMQERFAESTD
### Sample output file

**CLUSTAL X (1.8) multiple sequence alignment**

<table>
<thead>
<tr>
<th>Clustal</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sulfolobus</strong></td>
<td>------------------------------- MVSEGRVVVRNGPLVIAADMREAOFMQMEFVVFVYLMVKMDLVGE</td>
</tr>
<tr>
<td><strong>Thermococcus</strong></td>
<td>------------------------------- MGRIRVTGELVADGCMGKAMYETEVGEM KlVGGE</td>
</tr>
<tr>
<td><strong>Acetabularia</strong></td>
<td>------------- M------------- SKAKEglySikkVSggSCVVAADNMGSSACMYELVQRGTGCLE</td>
</tr>
<tr>
<td><strong>Daucus</strong></td>
<td>------------------------------- MPSVYGDRLTTEF -----SEKESEGYVVRKVSPVGPPVADGCMGAMAYELVQRHDLNIGE</td>
</tr>
<tr>
<td><strong>Trypanosoma</strong></td>
<td>------------------------------- MTSKDK--- ----- PKETKQRMAGAVKASGVPVIAENMGSSACMYELVQRFSLVGE</td>
</tr>
<tr>
<td><strong>Drosophila</strong></td>
<td>------------------------------- MSNLRKF------ DEERESQGRVFAVSFGFVTAAAESGMYELVQYYELIGE</td>
</tr>
<tr>
<td><strong>Candida</strong></td>
<td>MAGALENARKEIKRSLDITNSESQYGQIYVSYSGPVSINGACMYELVQRHDLNIGE</td>
</tr>
<tr>
<td><strong>Neurospora</strong></td>
<td>MAPQQNGA-------- EVDG-IHTGKlYSVSGPVVAEDMIGVAMYELVQRHDLQVE</td>
</tr>
<tr>
<td><strong>Saccharomyces cerevisiae</strong></td>
<td>MAGALENARKEIKRSLDITAESYGAISYSGPVSINGACMYELVQRHDLNIGE</td>
</tr>
<tr>
<td><strong>Borrelia</strong></td>
<td>------------------------------------ MNEVLFVKTGARNLKAE</td>
</tr>
</tbody>
</table>

**Sulfolobus**

| **Thermococcus** | IIRLEGDKAVIQQYYETTAGIRPGEPVEGTSSGGLTSMYFDIGQRPLVDLRQLS |
| **Acetabularia** | IIRLEGDTATIQQYYETSGLTVGDGPLRTQPLSVDLGPILGNIIFDQIRPPLKAIADV |
| **Daucus** | IIRLEGDSAITIQQYYETAGLMVNDPLVRTKHPILSGPLGIGNIFDQIRPPLKIIAKS |
| **Trypanosoma** | IIRLEGDTIQQYYETGGLTVGDGPLYCTGKPLSLLGPGMSIFDQIRPPLDIYRMOV |
| **Drosophila** | VIRINGDKATIQQYYETAVGDPPLRTGPQPLSVDLGPILGNIIFDQIRPPLKAIDES |
| **Candida** | VIRINGDKATIQQYYETAVGDPPLRTGPQPLSVDLGPILGNIIFDQIRPPLKAIDES |
| **Neurospora** | VIRINGDKATIQQYYETAVGDPPLRTGPQPLSVDLGPILGNIIFDQIRPPLKAIDES |
| **Saccharomyces cerevisiae** | VIRINGDKATIQQYYETAVGDPPLRTGPQPLSVDLGPILGNIIFDQIRPPLKAIDES |
| **Borrelia** | VIRINGKNEVAQQYFELTKGSISSVDLVEFDTDLVLTTGLRQVYDQGLNPLLEAIQC |

| **Sulfolobus** | NSPFVARUVSVIPALDRQTQKHVFV---KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Thermococcus** | G-DPIARGTMATPPLRDKWKHFPT---KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Acetabularia** | GDFIPPRGVNPLSDLQTKWEFRPSAFKVDGTVGDDGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Daucus** | GDVFIPPRGVNPLSDLQTKWEFRPSAFKVDGTVGDDGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Trypanosoma** | ENVIFPRGVQVKSNLQKDQWDFPK---CLKVDGLSVDGDLGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Drosophila** | ESIYIPKNGPVNLSSGAVSNFENPLNLVSVDGDLGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Candida** | CSIYIPRIGTIDPVPSRTTVQYDFTPQPLGBKHDITGDDIFGSIDYENLELDHDHLLIPPRAR |
| **Neurospora** | NSIYIPRIGTIDPVPSRTTVQYDFTPQPLGBKHDITGDDIFGSIDYENLELDHDHLLIPPRAR |
| **Saccharomyces cerevisiae** | QSIYIPRIGTIDPVPSRTTVQYDFTPQPLGBKHDITGDDIFGSIDYENLELDHDHLLIPPRAR |
| **Borrelia** | G-FPLERGVYLRPLNKDKKWNFKK---TKVDGDIVAAGDFGLGFIEGTVHHLIMIPFKPD |

| **Sulfolobus** | NSPFAVGRUVSVIPALDRQTQKHVFV---KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Thermococcus** | G-DPIARGTMATPPLRDKWKHFPT---KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Acetabularia** | GDFIPPRGVNPLSDLQTKWEFRPSAFKVDGTVGDDGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Daucus** | GDVFIPPRGVNPLSDLQTKWEFRPSAFKVDGTVGDDGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Trypanosoma** | ENVIFPRGVQVKSNLQKDQWDFPK---CLKVDGLSVDGDLGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Drosophila** | ESIYIPKNGPVNLSSGAVSNFENPLNLVSVDGDLGIGVPPNLSSGLH-KVKSDKGKVGPDIIGGYVQETDLIE---HLIRIPNVH |
| **Candida** | CSIYIPRIGTIDPVPSRTTVQYDFTPQPLGBKHDITGDDIFGSIDYENLELDHDHLLIPPRAR |
| **Neurospora** | NSIYIPRIGTIDPVPSRTTVQYDFTPQPLGBKHDITGDDIFGSIDYENLELDHDHLLIPPRAR |
| **Saccharomyces cerevisiae** | QSIYIPRIGTIDPVPSRTTVQYDFTPQPLGBKHDITGDDIFGSIDYENLELDHDHLLIPPRAR |
| **Borrelia** | G-FPLERGVYLRPLNKDKKWNFKK---TKVDGDIVAAGDFGLGFIEGTVHHLIMIPFKPD |
Clustal also reads aligned sequences. If you input aligned sequences you can go directly to the tree section.

!! Be careful if you make a mistake, and the sequences are not aligned, your tree will look strange!!

!!! ALWAYS CHECK YOUR ALIGNMENT!!!

Also be careful when using the ignore positions with gaps option – there might not be many positions left.

Clustal is much better than its reputation. It is doing a great job in handling gaps, especially terminal gaps, and it makes good use of different substitution matrices, and the empirical correction for multiple substitutions is better than many other programs.
tcoffee

TCOFFEE extracts reliably aligned positions from several multiple or pairwise sequence alignments. It requires more thought and attention from the user than clustalw, but it helps to focus further analyses on those sites that are reliably aligned. A web interface is here.
muscle

If you have very large datasets muscle is the way to go. It is fast, takes fasta formatted sequences as input file, and has a refinement option, that does an excellent job cleaning up around gaps.

The muscle home page is [here](#), the manual is [here](#) Muscle allows also allows profile alignments.

```
muscle -in VatpA.fa -out VatpA.afa
muscle -in VatpA.afa -out VatpA.rafa –refine
muscle -in beta.fa -out beta.afa
muscle -in beta.afa -out beta.rafa -refine
muscle -profile -in1 beta.rafa -in2 VatpA.rafa -out Abeta.afa
muscle -refine -in Abeta.afa -out Abeta.rafa
```
muscle alignment
muscle vs clustal

more on alignment programs (statalign, pileup, SAM) here
the same region using tcoffee with default settings

more on alignment programs (statalign, pileup, SAM) here
Sequence editors and viewers

Jalview [Homepage](#), [Description](#)

Jalview is easy to install and run.
Test file is [here](#) (ATPase subunits)
(Intro to ATPases: 1bmf in spdbv)
/gif of rotation [here](#)
(Load all.txt into Jalview,
   colour options,
   mouse use,
   PID tree,
   Principle component analysis -> sequence space)
More on sequence space [here](#)
Another useful multiple alignment editor is **seaview**, it runs on most platforms, uses either **clustal** or **muscle** alignments, and has simple parsimony, distance and ml programs.
Clustalw (and other progressive alignment programs):

Good alignment programs, alignments match regions that have same structures.

Not very useful for phylogenetic reconstruction: Alignment is strongly biased towards guide tree. Also, quality of alignment presumably depends on guide tree.

Solutions:
• Use different guide trees, especially if you want to test different phylogenetic hypotheses
• Use an alignment program that creates less bias (muscle)
• Use a program that optimizes tree and alignment simultaneously.
SATé
Simultaneous Alignment and Tree Estimation

http://phylo.bio.ku.edu/software/sate/sate.html

GUI works well on iMacs, but uses only local processors.
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!
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.
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.


**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](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.
challenge:

Often one wants to build families of homologous proteins extracted from genomes. One way to do so is to find reciprocal best hits.

Tools:
The script `blastall.pl` takes the genomes indicted in the first line and calculates all possible genome against genome searches.

This script `simple_rbh_pairs.pl` takes two blastall searches (genome A versus genome B) in -m8 format and listing only the top scoring blast hit for each query) and writes the GI numbers of reciprocal best hits into a table.

The script `run_pairs.pl` runs all possible pairwise extractions of RBHs

Task: write a script that combines the pairwise tables keeping only those families that have a strict reciprocal best blast hit relationship in all genomes.