[compile/install]

```
cd fasta-36.3.5e/
ls -l
ls –l bin
less README
cd src
make -f ../make/Makefile.linux_sse2
all
cd ../bin/
ls -l
ssearch
cd
ssearch
```

```
/usr/bin/ld: cannot find -lz
collect2: ld returned 1 exit status
make: *** [fasta36] Error 1
```

Trouble-shooting:

../make/Makefile.linux_sse2 called ../make/Makefile.common

vi ../make/Makefile36m.common to modify:
Change LIB_M= -lm -lz
LIB_M= -lm

```
make -f ../make/Makefile.linux_sse2 all
```

```
yyin@asp:~/tools/fasta-36.3.5e/src$ ls ../bin/
fasta36  fastm36  fastx36  ggsearch36  lalign36  lav2svg  ssearch36  tfastm36  tfastx36
fastf36  fasts36  fasty36  glsearch36  lav2ps   map_db   tfastf36   tfasts36  tfasty36
```

[edit PATH variable]

```
cd [go to your home]
vi .bashrc
[add the following line to the end of this file]
export PATH=“$PATH:absolute path to fasta bin folder”;

EXAMPLE for me:
export PATH=“$PATH:/home/yyin/tools/fasta-36.3.5e/bin/”;

. .bashrc [execute the script]
ssearch
```

[add alias of a command ll]

```
vi .bashrc
alias ll='ls -l‘
alias lt='ls -lt'
```

Environment variable

An environment variable is a named object that contains data used by one or more applications. The value of an environmental variable can for example be the location of all executable files in the file system, the default editor that should be used, or the system locale settings. Users new to Linux may often find this way of managing settings a bit unmanageable. However, environment variables provides a simple way to share configuration settings between multiple applications and processes in Linux.

`env` to list all built-in environment variable

PATH is a very important environment variable. This sets the path that the shell would be looking at when it has to execute any program. It would search in all the directories that are defined in the variable. Remember that entries are separated by a ' : ' . You can add any number of directories to this list.

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

Install BLAST using the common way

```
lftp ftp.ncbi.nih.gov:/blast/executables/LATEST> get ncbi-blast-2.2.27+-ia32-linux.tar.gz
```

```
tar -zxf ncbi-blast-2.2.27+-ia32-linux.tar.gz
ll
cd ncbi-blast-2.2.27+/bin
ll
./blastp -h
```

Download ncbi-blast-2.2.27+-x64-linux.tar.gz if your machine is 64 bit, to find out

```
uname -a
```

[edit path variable]

```
vi .bashrc
export PATH="$PATH:absolute path to blast bin folder";
. .bashrc
blastp
```

Install HMMER

```
sudo apt-get install hmmer
```

Hard way:

http://hmmer.janelia.org/software

bioperl

http://www.bioperl.org/wiki/Installing_BioPerl_on_Ubuntu_Server

```
sudo apt-get install bioperl
```

The hard way to install bioperl

```
wget -q http://bioperl.org/DIST/current_core_unstable.tar.bz2
tar -xjvf current_core_unstable.tar.bz2
cd bioperl-*
perl Build.PL   # choose the defaults
./Build test
./Build install
```

http://www.bioperl.org/wiki/Installing_BioPerl_on_Ubuntu_Server

Install MAFFT the hard way

```
wget -q http://mafft.cbrc.jp/alignment/software/mafft-7.029-with-extensions-src.tgz

tar xzf mafft-7.029-with-extensions-src.tgz

cd mafft-7.029-with-extensions/core/

sudo make
sudo make install
unset MAFFT_BINARIES   # change environmental variable

mafft # test if installed properly
```

http://mafft.cbrc.jp/alignment/software/source.html

Also edit .bashrc in your home to add the path to the executables to the PATH environmental variables

Install Galaxy

http://wiki.galaxyproject.org/Admin/Get%20Galaxy

```
sudo apt-get install mercurial
hg clone https://bitbucket.org/galaxy/galaxy-dist/
hg update stable
cd galaxy-dist
sh run.sh
```

http://localhost:8080

Edit universe_wsgi.ini file to allow access from other computers

Setup admin user:
http://wiki.galaxyproject.org/Admin/Interface
edit universe_wsgi.ini file

Run BLAST and HMMER in command line

Yanbin Yin

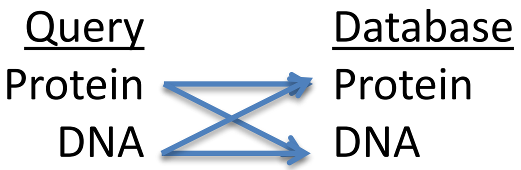
Spring 2013

BLAST

```
blastall - | less
-p # specify blastp, blastn, blastx, tblastn, tblastx

More commands in blast package

formatdb (format database)
megablast (faster version of blastn)
rpsblast (protein seq vs. CDD PSSMs)
impala (PSSM vs protein seq)
bl2seq (two sequence blast)
blastclust (given a fasta seq file, cluster them based on sequence similarity)
blastpgp (psi-blast, iterative distant homolog search)
```



Query
Protein
DNA

Database
Protein
DNA

http://www.ncbi.nlm.nih.gov/books/NBK1763/pdf/ch4.pdf

blastall options

-p program name

-d database file name (text fasta sequence file)

-i query file name

-e e-value cutoff (show hits less than the cutoff)

-m output format

-o output file name (you can also use >)

-F filter low-complexity regions in query

-v number of one-line description to be shown

-b number of alignment to be shown

-a number of processers to be used

New version blast, e.g. blastp

```
blastp -help | less
```

-query	query file name
-db	database file name
-out	output file name
-evalue	e-value cutoff
-outfmt	output format
-num_descriptions	
-num_alignments	

do the following in /media/DATAPART1/z1576493/class/mar19/

```
formatdb -i ecoli-all.faa
formatdb - # see the options, for nt db, also use -p F
less ecoli-all.faa # select the 3rd protein sequence(YP_488309.1)
vi test-query.fa # create a file to store this protein seq

[now blast, which is in your path alreay]
blastall -p blastp -i test-query.fa -d ecoli-all.faa
blastall -p blastp -i test-query.fa -d ecoli-all.faa > test-qery.fa.out

[-m 9, the tabular format output without alignment, easy to parse]
blastall -p blastp -i test-query.fa -d ecoli-all.faa -m 9
blastall -p blastp -i test-query.fa -d ecoli-all.faa -m 9 > test-qery.fa.out.m9

[-e 1e-2, showing only hits with evalue < 1e-2]
blastall -p blastp -i test-query.fa -d ecoli-all.faa -m 9 -e 1e-2

[Now try something big (and slow)]
blastall -p blastp -i test-query.fa -d /home/yyin/work/class/metagenemark_predictions.faa -m 9 -e 1e-2 > test-qery.fa.cowrumen.out.m9 &

[Do some parsing]
less test-query.fa.cowrument.out.m9 | cut -f1,2,3,7- | less
less test-query.fa.cowrument.out.m9 | cut -f1,2,3,7- | grep -v '^#' | cut -f2 | sort -u | head
```

If a program (e.g. BLAST) runs so long on a remote Linux machine that it won’t finish before you leave for home …

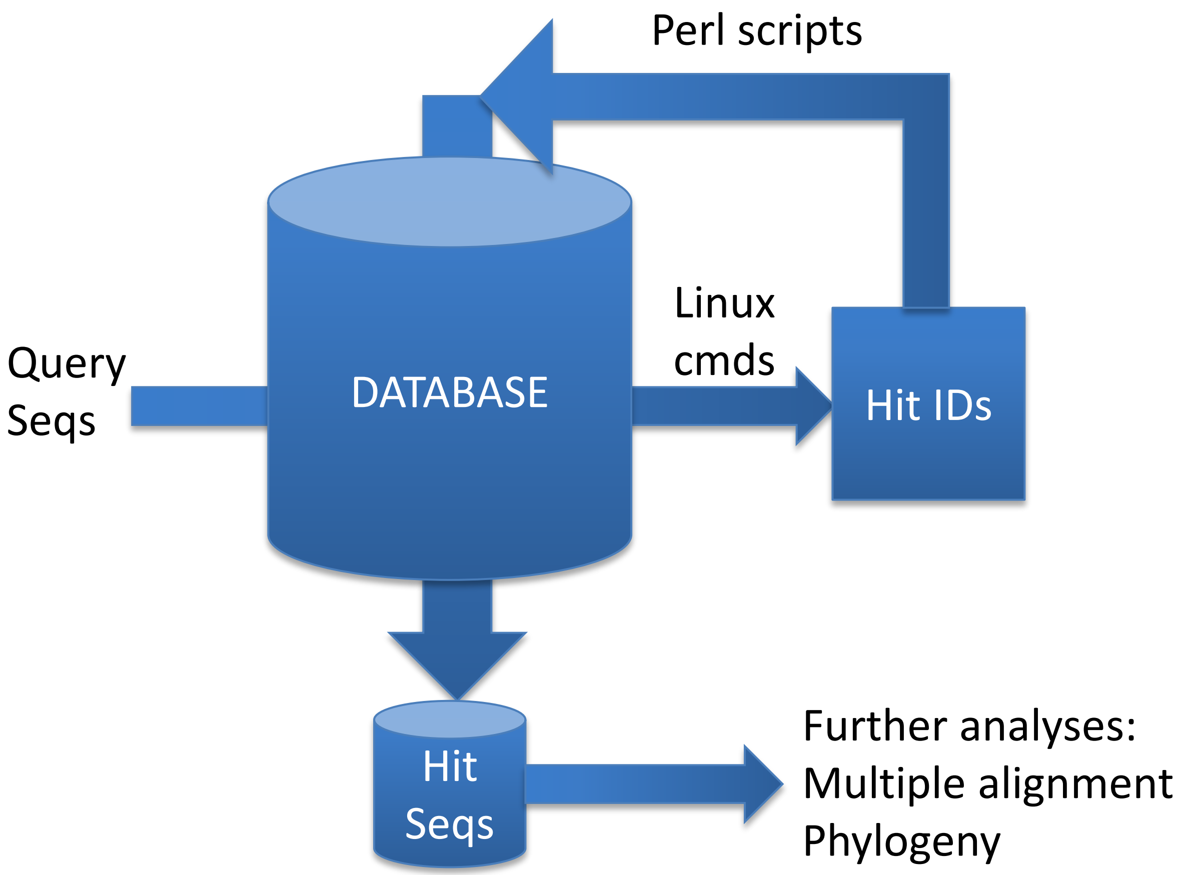
Or if you somehow want to restart your laptop/desktop where you have a Putty session is running (Windows) or a shell terminal is running (Ubuntu) …

In any case, you have to close the terminal session (or have it be automatically terminated by the server). If this happens, your program will be terminated without finishing. If you expect your program will run for a very long time, e.g. longer than 10 hours, you may put “nohup” before your command; this ensures that even if you close the terminal, the program will still run in the background until it is finished and you can log in again the next day to check the output. For example:

```
nohup blastp -query yeast.aa -db yeast.aa -out yeast.aa.ava.out
-outfmt 6 &
```

You will get an additional file nohup.out in the working folder and this file will be empty if nothing wrong happened.

How do you extract the sequences of the blast hits?



Multiple sequence alignment: run mafft using command line

```
mafft -h
```

```
/usr/local/bin/mafft: Cannot open --help.
------------------------------------------------------------------------------
  MAFFT v6.955b (2012/11/20)
  http://mafft.cbrc.jp/alignment/software/
  NAR 30:3059-3066 (2002), Briefings in Bioinformatics 9:286-298 (2008)
------------------------------------------------------------------------------
High speed:
  % mafft in > out
  % mafft --retree 1 in > out (fast)

High accuracy (for <~200 sequences x <~2,000 aa/nt):
  % mafft --maxiterate 1000 --localpair  in > out (% linsi in > out is also ok)
  % mafft --maxiterate 1000 --genafpair  in > out (% einsi in > out)
  % mafft --maxiterate 1000 --globalpair in > out (% ginsi in > out)

If unsure which option to use:
  % mafft --auto in > out

--op # :         Gap opening penalty, default: 1.53
--ep # :         Offset (works like gap extension penalty), default: 0.0
--maxiterate # : Maximum number of iterative refinement, default: 0
--clustalout :   Output: clustal format, default: fasta
--reorder :      Outorder: aligned, default: input order
--quiet :        Do not report progress
--thread # :     Number of threads (if unsure, --thread -1)
```

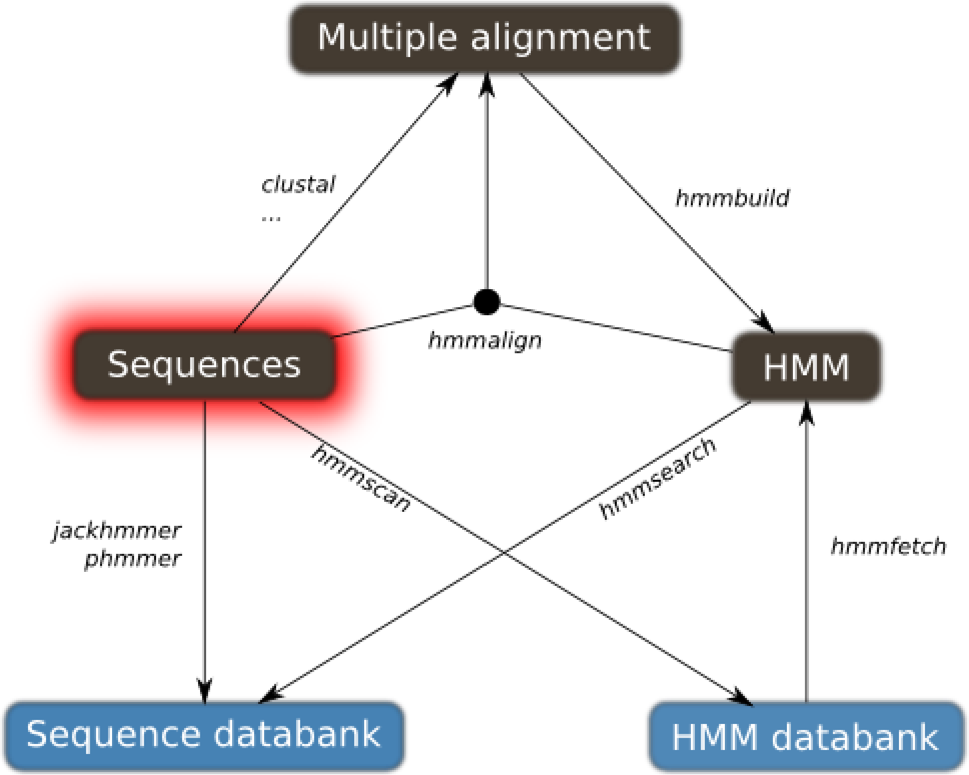
```
cp /home/yyin/work/class/test-query.fa.cowrument.out.m9.head10.fa .
```

```
mafft --auto  test-query.fa.cowrument.out.m9.head10.fa > test-query.fa.cowrument.out.m9.head10.fa.l
```

HMMER: http://hmmer.janelia.org/

What is HMMER? ftp://selab.janelia.org/pub/software/hmmer3/3.0/Userguide.pdf
HMMER is a software package that is used for searching sequence databases for homologs, making protein sequence alignments, and making **profile hidden Markov models** (profile HMMs)**.** It implements methods using probabilistic models called **profile hidden Markov models**, mathematically representing multiple sequence alignments.

Compared to BLAST, FASTA, and other sequence alignment and database search tools based on older scoring methodology, HMMER aims to be significantly *more* accurate and *more* able to detect remote homologs because of the strength of its underlying mathematical models. In the past, this strength came at significant computational expense, but in the new HMMER3 project, HMMER is now essentially **as fast as** BLAST



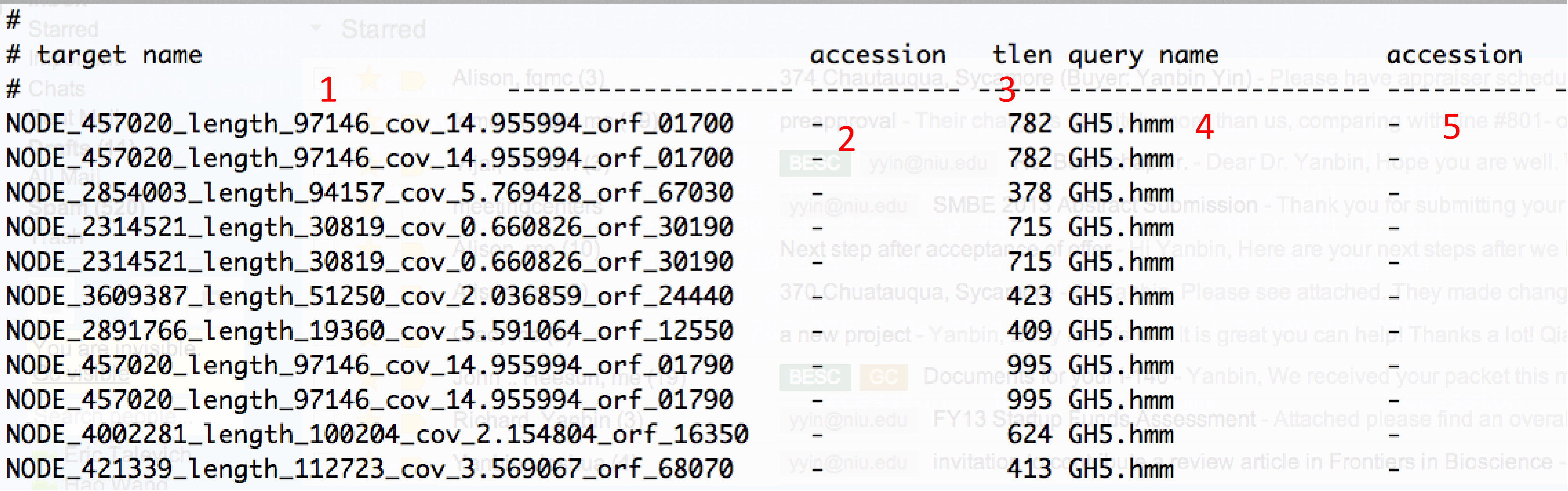
http://drmotifs.genouest.org/2010/10/sequence-hammering/

Go to http://cys.bios.niu.edu/dbCAN/family.php?ID=GH5 and download

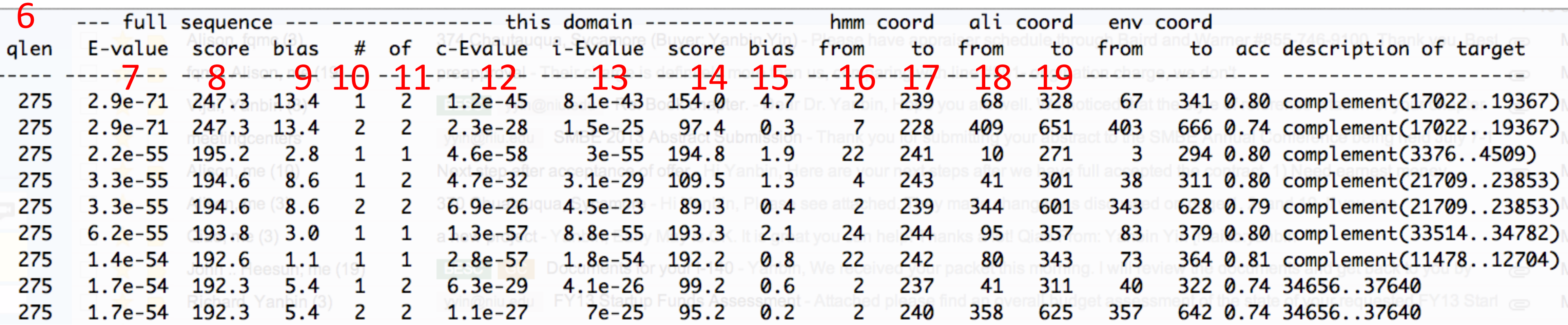
```
wget -q http://cys.bios.niu.edu/dbCAN/data/aln/cazy-family/aln/GH5.aln
less GH5.aln

hmmbuild # list options
hmmbuild -h  # list complete options
hmmbuild --informat afa GH5.hmm GH5.aln # build model, afa: aligned fasta format, see User Guide page 16 footnote
less GH5.hmm # profile HMM file is a text file

hmmsearch
hmmsearch –h
hmmsearch --domtblout GH5.hmm.cowrumen.dm GH5.hmm metagenemark_predictions.faa > GH5.hmm.cowrumen.out & # save easy-to-parse table of per-domain hits to file in addition to the regular output (with alignment)
```



```
#
# target name                                accession  tlen query name            accession
#                   ------------------- ---------- ----- -------------------- ----------
NODE_457020_length_97146_cov_14.955994_orf_01700  -     782 GH5.hmm              -
NODE_457020_length_97146_cov_14.955994_orf_01700  -     782 GH5.hmm              -
NODE_2854003_length_94157_cov_5.769428_orf_67030  -     378 GH5.hmm              -
NODE_2314521_length_30819_cov_0.660826_orf_30190  -     715 GH5.hmm              -
NODE_2314521_length_30819_cov_0.660826_orf_30190  -     715 GH5.hmm              -
NODE_3609387_length_51250_cov_2.036859_orf_24440  -     423 GH5.hmm              -
NODE_2891766_length_19360_cov_5.591064_orf_12550  -     409 GH5.hmm              -
NODE_457020_length_97146_cov_14.955994_orf_01790  -     995 GH5.hmm              -
NODE_457020_length_97146_cov_14.955994_orf_01790  -     995 GH5.hmm              -
NODE_4002281_length_100204_cov_2.154804_orf_16350 -     624 GH5.hmm              -
NODE_421339_length_112723_cov_3.569067_orf_68070  -     413 GH5.hmm              -
```



```
      --- full sequence --- -------------- this domain -------------   hmm coord   ali coord   env coord
qlen  E-value  score  bias   #  of  c-Evalue  i-Evalue  score  bias  from    to  from    to  from    to  acc description of target
----- --------- ------ ----- --- --- --------- --------- ------ ----- ----- ----- ----- ----- ----- ----- ---- ---------------------
 275  2.9e-71  247.3  13.4   1   2  1.2e-45  8.1e-43  154.0   4.7     2   239    68   328    67   341 0.80 complement(17022..19367)
 275  2.9e-71  247.3  13.4   2   2  2.3e-28  1.5e-25   97.4   0.3     7   228   409   651   403   666 0.74 complement(17022..19367)
 275  2.2e-55  195.2   2.8   1   1  4.6e-58    3e-55  194.8   1.9    22   241    10   271     3   294 0.80 complement(3376..4509)
 275  3.3e-55  194.6   8.6   1   2  4.7e-32  3.1e-29  109.5   1.3     4   243    41   301    38   311 0.80 complement(21709..23853)
 275  3.3e-55  194.6   8.6   2   2  6.9e-26  4.5e-23   89.3   0.4     2   239   344   601   343   628 0.79 complement(21709..23853)
 275  6.2e-55  193.8   3.0   1   1  1.3e-57  8.8e-55  193.3   2.1    24   244    95   357    83   379 0.80 complement(33514..34782)
 275  1.4e-54  192.6   1.1   1   1  2.8e-57  1.8e-54  192.2   0.8    22   242    80   343    73   364 0.81 complement(11478..12704)
 275  1.7e-54  192.3   5.4   1   2  6.3e-29  4.1e-26   99.2   0.6     2   237    41   311    40   322 0.74 34656..37640
 275  1.7e-54  192.3   5.4   2   2  1.1e-27    7e-25   95.2   0.2     2   240   358   625   357   642 0.74 34656..37640
```

[a little parsing, alignment in GH5.hmm.cowrumen.out]

```
less GH5.hmm.cowrumen.dm | grep -v '^#' | awk '{print
$1,$3,$6,$7,$12,$13,$16,$17,$18,$19}' | less
less GH5.hmm.cowrumen.dm | grep -v '^#' | awk '{print
$1,$3,$6,$7,$12,$13,$16,$17,$18,$19}' | awk '$6<1e-2&&($8-$7)/$3>.8' |
sed ‘s/ /\t/g’ | less
```

Extracting domain regions is easy if using perl and bioperl

emboss

```
seqret -help
```
http://emboss.sourceforge.net/apps/release/6.1/emboss/apps/seqret.html

```
seqret -sequence test-query.fa.cowrument.out.m9.head10.fa.l -outseq test-query.fa.cowrument.out.m9.head10.fa.l.aln -sformat fasta -osformat aln
```

```
infoseq -help
```
http://emboss.sourceforge.net/apps/release/6.2/emboss/apps/infoseq.html

```
infoseq -sequence test-query.fa.cowrument.out.m9.head10.fa -name -only -length
```

More command examples:

```
needle -help
water -help
fuzznuc -help
pepstats -help
pepinfo -help
```

```
plotorf  -help
transeq  -help
garnier -help
prettyseq -help
est2genome -help
```