LINUX Command Line Reference for Bioinformatics

DIRECTORIES

CREATE/DELETE DIRECTORIES

mkdir SeqDir
Make the directory SeqDir

rmdir SeqDir
Remove the empty directory SeqDir.

rm -rf SeqDir/
If the directory is not empty, you can can delete the dir and all subdirs and files using the rm command with the r and f options.

NAVIGATION

cd SeqDir
Change your current working directory to the SeqDir. Using cd without any options 'cd' will take you to your home directory.

cd .. Change to the parent directory.

cd /home/username/Dir/SubDir
Change dir using the full directory path.

DIR INFORMATION

pwd List the full path of your current directory.

ls List the files in the current directory.

ls -alh
List all files and show file size in a human readable format.

ls -l | wc -l
Count the number of files in a directory.

DIR COMPRESSION

tar -cvf SeqDir.tar SeqDir/ *or*
tar -cvfz SeqDir.tar.gz SeqDir/
Use the tar (Tape Archive) program to archive the directory named SeqDir. Use of the z option will zip the archive. Use

tar -xvf SeqDir.tar
Use x to extract the tar archive.

DIRECTORIES

UNCOMPRESS TREE TO DIR

The commands sequence of commands below allow you to uncompress an entire directory tree to a single directory. This is useful if you have downloaded sequence trace files from genbank and you would like all of the data in a single directory.

```
tar -C output -zxf archive.tar.gz
cd output
find . -type f -exec mv -i {} . \;
find * -type d -prune -exec rm -rf {} \;
```

FILES

More information on dealing with large text files is listed under the FASTA FILES heading.

COPY, RENAME & MOVE FILES

cp MySeqs.fasta MyCopy.fasta
Copy the file to MyCopy.fasta

cp *.fasta SeqDir/
Copy all files with the fasta extension to the destination folder.

mv MySeqs.fasta New.fasta
Rename the MySeqs.fasta file to New.fasta

mv *.fasta SeqDir/
Move all files with the fasta extension to the SeqDir directory.

FILE PERMISSIONS

chmod ### MyFile.txt *ie:*
chmod 755 MyProgram.pl
Change the permissions associated with files and directories. (ie make a PERL programs executable). The ### refer to file permission numeric code for

FILE COMPRESSION

gzip MyFile.txt
This will gzip the file MyFile.txt

gunzip MyFile.txt.gz
This will unzip the file MyFile.txt

bzip2 MyFile.txt
Compress the file with bzip (better).

bunzip2 MyFile.txt.bz
Unzip the bzipped file.

FIND FILES

locate MyFile.txt
The locate command can be used to find the location of files on your hard drive.

GENERAL PROGRAMS

`man progname`
The man command displays the manual for Linux command line programs.

`nohup`
The nohup command allows you to close your terminal connection to the Linux machine but keep your program running.

`clear`
Clear the screen.

`passwd`
Set your user password.

SPECIAL CHARACTERS

`|` **Pipe Output**
You can pass output from one program to another program using the pipe character |. Examples are:
`ls | less` *or* **`head myfile | less`**

`>` **Write to File**
The > character can be used to send program output to a text file. Examples:
`ls > File.txt` or
`locate perl > Perl.txt`

`>>` **Append to File**
Results are appended to the outfile.

`*` **Wildcard Character**
The asterix is often used as the wildcard character. It will match a set of characters for any length. Example use: ls *.fasta

`?` **Wildcard Character**
The question mark is a wildcard for a single character.

RESOURCE USAGE

It is important to keep track of your resource usage in multiuser environments. The following commands help you keep track of your storage, and processor use on your Linux machine.

DISK USAGE

`quota`
See what your disk usage quota is on the current machine. You may have no quota.

`df -h`
Look at the amount of disk space used by you and everyone else on the server.

`du -h --max-depth=1`
Display your disk use in the current dir. This is a good way to check which files or directories are using up disk space.

PROCESSES

`top`
Display the top CPU processes on the local machine. This will show the processes as well as their memory and processor usage.

`ps -ef`
Show all processes currently running

`ps -ef | grep username`
Show only your processes. If a process is running that you want to stop, use the 'kill' command.

`kill PID`
Kill the process identified by the process id (PID). The PID can be determined using the ps utility. WARNING: 'kill -9' is the nuclear option and will kill the hell out of your runaway process; it may however trash your database, files etc.

USERS

`who`
Show who else is logged on.

`finger username`
Get information about the user including real name, home dir etc.

FASTA FILES

For a FASTA file named MySeqs.fasta:

FILE OVERVIEW

`ls *.fasta`
Show all fasta files in the directory.

`grep -c '>' MySeqs.fasta`
Count the number of sequence records.

`wc -l MySeqs.fasta`
Count the number of lines in the file.

`wc -c MySeq.fasta`
Count the total number of characters.

VIEW FILE

`less MySeqs.fasta`
View the entire fasta file.

`head -n 50 MySeqs.fasta` *or*
`head -n 300 MySeqs.fasta | less`
Look at the beginning of a fasta file. Use (-n) to select the number of lines. For large n pipe the output to the `less` utility.

`tail -n 50 MySeqs.fasta` *or*
`tail -n 300 MySeqs.fasta | less`
Look at the end of the fasta file.

MERGE AND SPLIT FILES

`cat *.fasta > AllSeqs.fasta`
Combine all fasta files in the current directory into a single file.

`csplit AllSeqs.fasta '/>/' {*}` *or*
`csplit -f Seq -n 8 AllSeqs.fasta '/>/' {*}`
Split the fasta file into a separate fasta file for each record. The following options are available csplit:
- **`-f`** Prefix in output names
- **`-n`** Num digits long for output names
- **`-b`** Suffix for output names

BASIC PERL

For a PERL program named MyPerl.pl:

MODIFY AND RUN PROGRAMS

`emacs MyPerl.pl`
Use the emacs text editor to edit the program.

`chmod 755 MyPerl.pl`
Make the program executable by you and other people in your group and anyone else on the server but other people do not have wrtie access to the program.

`./MyPerl.pl`
Run the perl program 'MyPerl.pl' in the current directory.

LOOPS

`for ( $i=0; $i<=$MaxNum; $i++) {}`
Loop variable $i from zero to MaxNum

FREQUENTLY USED PERL MODULES

`DBI`
Database interface for connection to database servers (MySQL).

`Getopt::Std`
Accept command line arguments

`Term::ANSIColor`
Print in color. Useful for drawing attention to error messages, table headers etc. example:
print color 'bold red';
print "WARNING\n";
print color 'reset';

`Text::Wrap`
I use this for printing strings of sequence residues that are tabbed over.

EMACS TEXT EDITOR

Emacs is a powerful text editor available on many linux distributions. Emacs makes heavy use of the Clt, Meta (or ALT) and Shift keys. These are indicated below as C, M and S. To launch emacs from the command line simply type:

`emacs MyProgram.pl`
This will open the file MyProgram.pl for editing in emacs. If the file does not already exist, a new file will be created.

`C-h`	Online help
`C-g`	Stop current operation.

FILES

`C-x C-s`	Save the current file.
`C-x C-W`	Save the file to a new name.
`C-x C-c`	Close the current file.
`C-x d`	Open the directory.
`C-x i`	Insert another file.

EDIT

`Backspace`	Delete previous character.
`C-k`	Kill to end of the current line.
`C-y`	Paste.
`C-S-_`	Undo.
`C-w`	Cut.

SEARCH

`M-S->`	Go to end of the buffer.
`C-s`	Search forward
`C-r`	Search backward

CURSOR MOVEMENT

`C-l`	Recenter, refresh screen (lc L)
`C-a`	Move to beginning of current line
`C-e`	Move to end of the current line
`M-f`	Move forward one work
`M-b`	Move backward one word
`C-v`	Move forward one screen
`M-v`	Move back one screen
`M-S-<`	Go the beginning of the buffer
`M-x goto-line`	Goto line number

NCBI BLAST

The NCBI Standalone BLAST program is available for download from NCBI:
http://www.ncbi.nih.gov/BLAST/download.shtml.

`formatdb -p F -i MySeqs.fasta -t Seq -n Seq`
Format the fasta file named MySeqs.fasta. For more variables available type 'formatdb –help'. The title {-t} and name {-n} of the database will both be set to Seq.

`blastall --help`
Display the NCBI BLAST help.

`blastall -p program -i infile -d DB -o outfile`

- *`program`* is one of:

	Query	Database
blastn	Nucleotide	Nucleotide
blastp	Protein	Protein
blastx	Trans. Nucl.	Protein
tblastn	Protein	Trans. Nucl.
tblastx	Trans. Nucl.	Trans. Nucl.

- *`infile`* is a fasta formatted text file
- *`DB`* is a blast database created using formadb
- *`outfile`* is the path of the output file. I like to give the outfile the *.blo extension to represent this as a blast output.
- A number of other command line options are available for blastall. These include:
  - `-a` Number of processors to use
  - `-e` E-value cutoff
  - `-U` Mask out lowercase letters
  - `-G` Cost to open a gap
  - `-E` Cost to extend a gap
  - `-W` Default word size

SFTP

SFTP is a secure file transfer program that comes installed by default with most Linux distributions. This is the most secure way to transfer files from the command line.

CONNECTING

`sftp ftp.here.edu`
: Connect to the ftp server at the address specified. You will be prompted for a valid user name and password.

`exit` Quit the SFTP session. Also: **`quit`**

`help` Display SFTP help. Also: **`?`**

`!` Escape to the local shell

`! cmd` Run command 'cmd' in the local shell

DIRECTORY NAVIGATION

`mkdir MyDir`
: Create a directory on the ftp server

`lmkdir MyDir`
: Create a directory on the local machine

`pwd` Display the remote working dir.

`lpwd` Display the local working dir.

`cd` Change dir on the ftp server.

`lcd` Change dir on the local machine

`ls` List files on the server dir

`lls` List files in the local dir

TRANSFERRING FILES

`get myfile.fasta`
: Download a file from the server.

`get *.fasta`
: Download multiple files.

`put myfile.fasta`
: Upload file from the localmachine to the ftp server.

`put *.fasta`
: Upload multiple files.

MySQL

MySQL is a great freeDatabase commands. Remember that all MySQL commands must end with ;or \g .

GETTING STARTED

OUTPUT

using

using the \g

CREATE A DATABASE

```
CREATE DATABASE dbName;
```
Create a new database named 'dbName'.
```
SHOW DATABASES;
```
Show all databases in mysql.
```
USE dbName;
```
Use the Selected database.

CREATING TABLES

```
CREATE TABLE tblName
  (ColOne integer,
   ColTwo char(10),
   ColThree integer);
```
Creates a table with three columns named ColOne, ColTwo and Col Three.
```
SHOW DATABASES;
```
Show all databases in mysql.
```
USE dbName;
```
Use the Selected database.

WORKING WITH TABLES

For a table name tblName;

```
SELECT COUNT(*) FROM tblName;
```
Count the number of records in the table;
```
SELECT * FROM tblName;
SELECT * FROM tblName\g
```
Print all of the records from the table. Use of \g
```
SHOW COLUMNS FROM tblName;
```
Show the names of the table columns.
```
USE dbName;
```
Use the Selected database

BioPERL Reference Card
Reference for Bioinformatics

BIOPERL BLAST PARSING

OVERVIEW

```
use Bio::SearchIO;
$in = new::Bio::SearchIO(
      format => 'blast',
      file => 'FilePath')
while ($result = $in->next_result)
while ($hit = $result->next_hit)
while ($hsp = $hit->next_hsp)
```

RESULT

algorithm
The algorithm used (ie. blastn)
algorithm_version
algorithm version (ie. 2.2.12)
query_name
Name of the query sequence
query_accession
Accession number of query sequence
query_length
Length of the query sequence
query_description
Description of query sequence
database_name
Name of the database use for query
database_letters
Number of residues in the query
database_entries
Number of records in the database
available_statistics
Stats use for the BLAST search
available_parameters
Parameters used for the BLASTsearch
num_hits
The total number of hits for the query.
hits
Returns all the hits for the query sequence

BIOPERL BLAST PARSING

HIT

name Name of the matching sequence.
length
Total length of the hit sequence
accession
Accession number of the hit seq.
description
Description of hit seq.
algorithm
Blast algorithm use (ie. blastn)
raw_score
Raw score of the match.
significance
Significance of the match
bits Bit score of the match
num_hsps
Total number of hsps
locus Locus name of the hit
accession_number
Accession number
hsps Returns all hsps for hit

HSP

algorithm
BLAST algorithm used. (ie blastn)
evalue
E Value of HSP
frac_identical
Fraction of residues identical.
frac_conserved
Fraction of residues conserved (proteins)
gaps Number of gaps in alignment.
query_string
Query sequence from alignment
hit_string
Hit sequence from alignment
homology_string
Homology string from alignment
length('total')
Length of hsp including gaps

BIOPERL BLAST PARSING

HSP (CONT'D)

length('hit')
Length of aligned hit minus gaps
length('query')
Length of aligned query minus gaps
num_conserved
Number of conserved residues
num_identical
Number of identical residues
rank Rank of the HSP
score Score
bits HSP score in bits
range('query')
Start and end of qry as an array
range('hit')
Start and end of hit has an array
percent_identity
Percent identical in HSP alignment
strand('hit' or 'query')
Strand of the hit or query.
start('query' or 'hit')
Start position of the hit or query
end('query' or ''hit')
End position of the hit or query.

new::Bio::SearchIO

file Path to input file
format
Format of the IO (ie. blast)
-report_type
-inclusion_threshold
signif E value cutoff
score Blast Score value cutoff
bits Bit value cutoff
hit_filter
overlap

More information available at:
http://bioperl.org/wiki/HOWTO:SearchIO

BIOPERL SEQ OBJECT

Information that can be fetched from the BioPERL Seq Object

OVERVIEW

```
use Bio::Seq;
$seq_in  = Bio::SeqIO->new (
       '-format' => 'fasta',
       '-file' => '<$infile' );
$seq_out = Bio::SeqIO-> new (
       '-format' => 'fasta',
       '-file' => '>$outfile' );

while(
 ( my $seqobj = $seq_in->next_seq() ) )
 { DoSomething with $seqobj }
```

SEQUENCE FORMATS

Sequence format can be one of the following:

Format	Description	Object
abi	abi tracefile	
ace	ace format	PrimarySeq
chadoxml	chado xml	
embl	EMBL	Seq::RichSeq
fasta	fasta format	Seq
fastq	quality info	
game	game xml	
genbank	genbank *.gb	Seq::RichSeq
qual	Phred	
scf	Standard chrom	
swiss	SwissProt	Seq::RichSeq
strider	DNA Strider	
tigr	TIGR XML	
tinyseq	NCBI TinySeq	
ztr	ZTR Tracefile	

BIOPERL SEQ OBJECT

Bio::Seq

seq() **$**
Sequence string

subseq(i,j) **$**
Substring of sequence from position i to j

accession_number()$
Accession number of the sequence

alphabet() **$**
Residues identified as dna, rna or protein

seq_version() **$**
Sequence version when available

keywords() **$**
Keywords line when available

length() **$**
Length of the sequence string

desc() **$**
Description of the sequence

primary_id() **$**
Primary id for the sequence

display_id() **$**
Display id for the sequence

revcom **$**
Reverse complement of the sequence

translate **$**
Translate sequence

species() **Bio::Species**
Species object

annotation() **Bio::Annotation::Reference**
Bio::Annotation::Comment
Annotation object

get_SeqFeatures **SeqFeatureI**
Top level sequence features

get_all_SeqFeatures
All sequence features (ie. exons etc.)

Information at:
http://doc.bioperl.org/releases/bioperl-current/bioperl-live/Bio/Seq.html

Bio::Seq::RichSeq
Moreinformation available at
http://bioperl.org/wiki/HOWTO:SeqIO

BIOPERL HMMER PARSING

HMMER is a program that uses profile hidden Markov models to identify protein families.
http://hmmer.janelia.org/

OVERVIEW

```
use Bio::Tools::HMMER::Results;
$HmmRes = new::Bio::Tools::HMMER::Results (
       -type => 'hmmsearch',
       -file => $FilePath);
foreach $seq ( $HmmRes->each_Set)
foreach $domain ( $seq->each_Domain)
```

-type can be hmmsearch, hmmpfam
-type
hmmsearch or hmmpfam

SEQ (usage: ie. $seq->bits)

accession
Accession number of the qry sequence

bits
The bit score for the set of hits

description
Description of the qry sequence

evalue
The evalue of the set of hits

name
The name of the query sequence

BIOPERL HMMER PARSING

DOMAIN (usage: ie. $seq->bits)

bits
: Bit score of the domain match

evalue
: Eval of the domain match

get_nse
: Return the name start end

hmmacc
: Accession for -type=>hmmpfam

hmmname
: Name of the domain match

seqbit
: Bits for the sequence (eq $seq->bits)

seq_id
: Name of the sequence (eq $seq->name)

start
: Start of the match in the end sequence

end
: End of the match in the end sequence

hstart
: Start of the match in the hit sequence

hend
: End of the match in the hit sequence

WINDOWS SOFTWARE

The following sources of software for windows are useful for connecting to a Linux box from MS Windows or working with programs and files generated on the Linux side.

Context Text Editor
http://www.context.cx/
: A useful program for programming on the MS windows machine. It can convert between UNIX, Windows, and MAC text file formats.

CygWinX
http://xfree86.cygwin.com/

Putty
http://www.chiark.greenend.org.uk/~sgtatham/putty/
: Open source SSH client for windows.

Unix Utilities For Windows
http://unxutils.sourceforge.net/
: A number of Linux/Unix programs that run in the native windows envrionment. Programs include gzip, bzip, grep, tar and less. Just install these in the directory: C:/Windows/System32 and you will be able to use them from the windows command line.

XwinLogin
http://www.calcmaster.net/visual-c++/xwinlogon/

James C. Estill
jestill@sourceforge.net
Sept 19, 2006