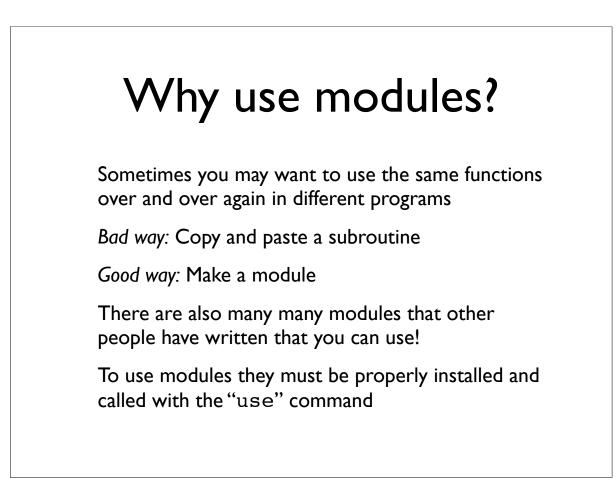
# Using Modules

### v2 2012



## File::Basename

### basename

Input = long UNIX path name i.e. '/home/dave/dna.fa'

Output = file name i.e. 'dna.fa'

dirname

Input = long UNIX path name i.e. '/home/dave/dna.fa'

Output = directory '/home/dave/'

### Let's try it.

# File:Basename. #!/usr/bin/perl # file: basename.pl we strict; use strict; use File::Basename; my \$path = '/home/dave/dna.fa'; my \$base = basename(\$path); my \$dir = dirname(\$path); print "The base is \$base and the directory is \$dir.\n"; Output: The base is dna.fa and the directory is /home/dave. Common Indefined subroutine &main::basename called at basename.pl error: Line 8.

# Env

This standard module imports a set of scalar variables that describe your environment

\$HOME \$PATH \$USER

### Env

#!/usr/bin/perl # file env.pl

use strict; use Env;

print "My home is \$HOME\n"; print "My path is \$PATH\n"; print "My username is \$USER\n";

My home is /home/dave Output: My path is /home/dave/pfb2012 My username is dave

# Which modules are installed?

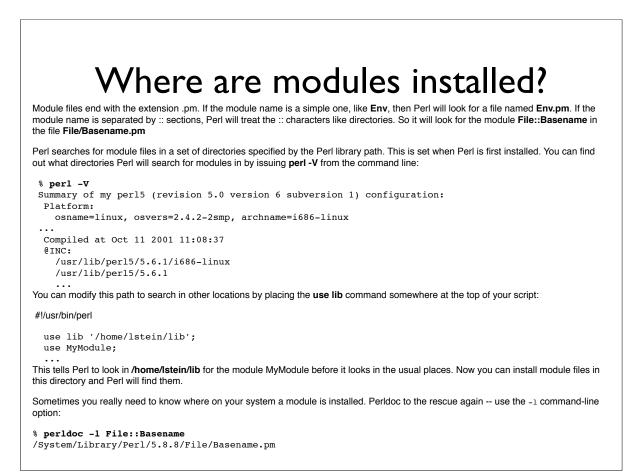
dave\$ perldoc perlmodlib Which modules are installed with basic perl installation? <u>http://perldoc.perl.org/perlmodlib.html</u>

dave\$ perldoc perllocal

Which modules are installed on your machine?

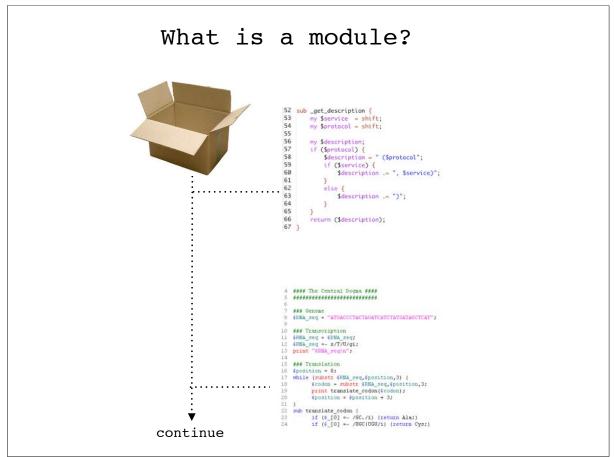
### Installing modules manually % tar zxvf bioperl-1.6.1.tar.gz bioperl-1.6.1/ bioperl-1.6.1/Bio/ . . . % perl Makefile.PL Generated sub tests. go make show tests to see available subtests Writing Makefile for Bio % make cp Bio/Tools/Genscan.pm blib/lib/Bio/Tools/Genscan.pm Manifying blib/man3/Bio::Location::CoordinatePolicyI.3 Manifying blib/man3/Bio::SeqFeature::Similarity.3 % make test PERL\_DL\_NONLAZY=1 /net/bin/perl -Iblib/arch -Iblib/lib -I/net/lib/perl5/5.6.1/i686-linux -I/net/lib/perl5/5.6.1 -e 'use Test::Harness qw(&runtests \$verbose); \$verbose=0; runtests @ARGV;' t/\*.t t/AAChange.....ok All tests successful, 95 subtests skipped. Files=60, Tests=1011, 35 wallclock secs (25.47 cusr + 1.60 csys = 27.07 CPU) % make install Installing /net/lib/perl5/site\_perl/5.6.1/bioback.pod Installing /net/lib/perl5/site\_perl/5.6.1/biostart.pod . . .

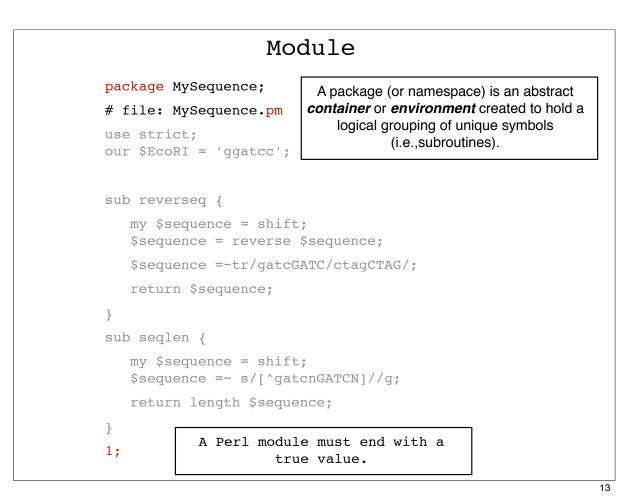
Installing Modules Using the CPAN Shell
المعالم
- cpan shell CPAN exploration and modules installation (v1.59_54) ReadLine support enabled
cpan> From this shell, there are commands for searching for modules, downloading them, and installing them.
[The first time you run the CPAN shell, it will ask you a lot of configuration questions. Generally, you can just hit return to accept the defaults. The only trick comes when it asks you to select CPAN mirrors to download from. Choose any ones that are in your general area on the Internet and it will work fine.]
To search for a module:
<pre>cpan&gt; i /Wrap/ Going to read /bush_home/bush1/lstein/.cpan/sources/authors/01mailrc.txt.gz CPAN: Compress::Zlib loaded ok Going to read /bush_home/bush1/lstein/.cpan/sources/modules/02packages.details.txt.gz Database was generated on Tue, 16 Oct 2001 22:32:59 GMT </pre>
<pre>Module Text::Wrap (M/MU/MUIR/modules/Text-Tabs+Wrap-2001.0929.tar.gz) 41 items found</pre>
<pre>cpan&gt; install Text::Wrap Running install for module Text::Wrap quit</pre>



# Making modules

Programming for Biology





```
script
#!/usr/bin/perl
# file: sequence.pl
use strict;
use warnings;
use MySequence;
my $sequence ='gattccggatttccaaagggttcccaattggg';
my $complement = MySequence::reverseq($sequence);
print "original = $sequence\n";
print "complement = $complement\n";
Must explicitly qualify each MySequence function by
using the notation
MySequence::function_name
```

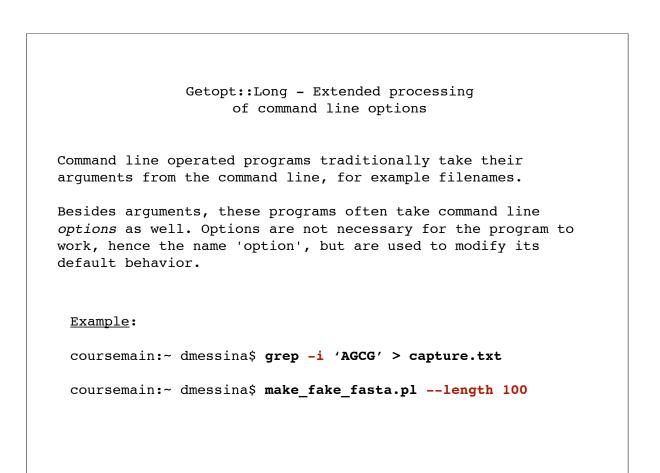
## Module using Exporter

```
package MySequence;
# file: MySequence.pm
use strict;
use base 'Exporter';
our @EXPORT = qw(reverseq seqlen);
our @EXPORT_OK = qw($EcoRI);
our $EcoRI = 'ggatcc';
sub reverseq {
  my $sequence = shift;
   $sequence = reverse $sequence;
   $sequence =~ tr/gatcGATC/ctagCTAG/;
   return $sequence;
}
sub seqlen {
  my $sequence = shift;
   $sequence =~ s/[^gatcnGATCN]//g;
   return length $sequence;
}
1;
```

Script using Exporter
#!/usr/bin/perl
# file: sequence.pl
use strict;
use warnings;
use MySequence;
my \$sequence ='gattccggatttccaaagggttcccaatttggg';
my \$complement = reverseq(\$sequence);
print "original = \$sequence\n";
print "complement = \$complement\n";

\*

Exporter - Implements default import method for modules ..... use base 'Exporter'; our @EXPORT = qw(reverseq seqlen); our @EXPORT\_OK = qw(\$EcoRI); ; use base 'Exporter' Tells Perl that this module is a type of "Exporter" module our @EXPORT = qw(reverseq seqlen) line tells Perl to export the functions reverseq and seqlen automatically. Also, can export qw(afunc \$scalar @array \$hash); our @EXPORT OK = qw(\$EcoRI) tells Perl that it is OK for the user to import the \$EcoRI variable, but not to export it automatically.



```
Script using Getopt::long
#!/usr/bin/env perl
use strict;
use warnings;
use Getopt::Long;
my $length = 30;
my $number = 10;
my $help;
my $usage = "make_fake_fasta.pl - generate random DNA seqs
-r <number> the number of sequences to make (default: 10)
-1 <length> the length of each sequence (default: 30)
";
Options:
die $usage if $help;
my @nucs = qw(A C T G);
for (my $i = 1; $i <= $number; $i++) {</pre>
    my $seq;
    for (my $j = 1; $j <= $length; $j++) {</pre>
    my $index = int(rand (4));
    my $nuc = $nucs[$index];
    $seq .= $nuc;
    }
    print ">fake$i\n";
    print $seq, "\n";
}
```

```
*
19
```