

Introducing Bioperl

Toward the Bioinformatics Perl programmer's
nirvana

What Is Bioperl?

`http://www.bioperl.org`

Bioperl's Relationship To Project Ensembl

<http://www.ensembl.org>

Installing Bioperl

```
$ tar -zxvf current_core_stable.tar.gz  
$ tar -zxvf current_run_stable.tar.gz
```

```
$ perl Makefile.PL  
$ make  
$ make test  
$ su  
$ make install  
$ <Ctrl-D>
```

```
$ su  
$ perl -MCPAN -e "shell"  
$ cpan> install Bundle::BioPerl  
$ cpan> quit  
$ <Ctrl-D>
```

Using Bioperl: Fetching Sequences

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

# simple_get_sequence.pl - a simple Bioperl example program, which
# downloads SWISS-PROT sequences.

use strict;

use Bio::Perl;

my $ID = shift;

my $Sequence = get_sequence( 'swiss', $ID );

write_sequence( ">./seqs/$ID.swp", 'swiss', $Sequence );
write_sequence( ">./seqs/$ID.fsa", 'fasta', $Sequence );
```

Results of Fetching Sequences ...

```
$ mkdir seqs  
  
$ chmod +x simple_get_sequence.pl  
  
$ ./simple_get_sequence.pl P00392  
  
$ ./simple_get_sequence.pl P00392  
  
$ ./simple_get_sequence.pl MERA_PSEAE  
  
$ diff seqs/P00392.fsa seqs/MERA_PSEAE.fsa  
  
$ diff seqs/P00392.swp seqs/MERA_PSEAE.swp  
  
$ perldoc Bio/Perl.pm
```

The Multi_Seq_Get.pl program

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

# Multi_Seq_Get.pl - when provided with a list of IDs, download them.

use strict;

use Bio::Perl;

while ( my $ID = <> )
{
    chomp( $ID );

    print "Fetching Sequence: $ID.\n";

    my $Sequence = get_sequence( 'swiss', $ID );

    write_sequence( ">./seqs/$ID.swp", 'swiss', $Sequence );
    write_sequence( ">./seqs/$ID.fsa", 'fasta', $Sequence );
}
```

Fetching multiple sequences

```
$ ./Multi_Seq_Get.pl Merp_IDs.lst
```

```
$ ./Multi_Seq_Get.pl Mera_IDs.lst
```


Maxim 20.1

Don't reinvent the wheel. Use Bioperl whenever possible.

Remote BLAST Searches

```
$blast_result = blast_sequence( $seq );  
write_blast( $filename, $blast_result );
```

Maxim 20.2

Combine Bioperl with other tools to get your work done

A quick aside: the blastcl3 NetBlast client

```
$ blastcl3 -i ./seqs/P00392.fsa -d swissprot -p blastp \  
> Swiss-Prot.NetBlast_P00392.res
```

The Blast_parse.pl program

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

# Blast_parse.pl - post-process/parse BLAST output.

use strict;

use Bio::SearchIO;

my $bls_report = shift;

my $in = new Bio::SearchIO( -format => 'blast', -file => $bls_report );

while ( my $result = $in->next_result )
{
    while( my $hit = $result->next_hit )
    {
        print "Hit = ", $hit->name, "\n";
    }
}
```

Parsing BLAST outputs

```
$ ./Blast_parse.pl Swiss-Prot.NetBlast_P00392.res
```

Results from Blast_parse.pl ...

```
Hit = sp | P00392 | MERA_PSEAE
Hit = sp | P94702 | MERA_ENTAG
Hit = sp | Q52109 | MERA_ACICA
Hit = sp | P94188 | MERA_ALCSP
Hit = sp | P08332 | MERA_SHIFL
Hit = sp | Q51772 | MERA_PSEFL
Hit = sp | P17239 | MERA_THIFE
Hit = sp | Q54465 | MERA_SHEPU
Hit = sp | | P08662_1
Hit = sp | P30341 | MERA_STRLI
```

Where To From Here