

CAP 5510: Introduction to Bioinformatics
CGS 5166: Bioinformatics Tools

Giri Narasimhan

ECS 254; Phone: x3748

giri@cis.fiu.edu

www.cis.fiu.edu/~giri/teach/BioinfF18.html

BioPerl



Example 1: Convert SwissProt to fasta format

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

use strict;
use Bio::SeqIO;
my $in = Bio::SeqIO->newFh ( -file => '<seqs.html',
                             -format => 'swiss' );
my $out = Bio::SeqIO->newFh ( -file => '>seqs.fasta',
                              -format => 'fasta' );

print $out $_ while <$in>;

exit; #bioperl1.pl
```

Example 2 : Load sequence from remote server

```
#!/usr/bin/perl -w
use Bio::DB::SwissProt;

$database = new Bio::DB::SwissProt;

$seq = $database->get_Seq_by_id('MALK_ECOLI');

my $out = Bio::SeqIO->newFh(-fh => STDOUT,
    -format => 'fasta');

print $out $seq;

exit;
```

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

use Bio::DB::GenBank;

my $gb =
    new Bio::DB::GenBank(
        -retrievaltype=>'tempfile',
        -format=>'Fasta');

my ($seq) = $seq =
    $gb->get_Seq_by_id("5802612");
print $seq->id, "\n";
print $seq->desc(), "Sequence: \n";
print $seq->seq(), "\n";

exit;
```

Sequence Formats in BioPerl

```
#!/local/bin/perl -w
use strict;
use Bio::SeqIO;
my $in = Bio::SeqIO->new ( -file => 'seqs.html', -format => 'swiss' );
my $out = Bio::SeqIO->new ( -file => 'seqs.fas', -format => 'fasta' );

while ($seq = $in->next_seq()) {
    $accNum = $seq->accession_number();
    print "Accession# = $accNum\n";
    $out->write_seq($seq);
}

exit; #bioperl2.pl
```

BioPerl

```
#!/usr/bin/perl -w
# define a DNA sequence object with given sequence
$seq = Bio::Seq->new('-seq'=>'actgtggcgtcaact',
    '-desc'=>'Sample Bio::Seq object',
    '-display_id' => 'somethingxxx',
    '-accession_number' => 'accnumxxx',
    '-alphabet' => 'dna' );
$gb = new Bio::DB::GenBank();

$seq = $gb->get_Seq_by_id('MUSIGHBA1'); #returns Seq object
$seq = $gb->get_Seq_by_acc('AF303112'); #returns Seq object
# this returns a SeqIO object :
$seqio = $gb->get_Stream_by_batch([ qw(J00522 AF303112)]));
exit; #bioperl3.pl
```

Sequence Manipulations

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

use Bio::DB::GenBank;

$gb = new Bio::DB::GenBank();

$seq1 = $gb->get_Seq_by_acc('AF303112');
$seq2=$seq1->trunc(1,90);
$seq2 = $seq2->revcom();

print $seq2->seq(), "\n";
$seq3=$seq2->translate;
print $seq3->seq(), "\n";
exit; #bioperl4.pl
```

BioPython



Key Links for BioPython

- ❑ Tutorial: <http://biopython.org/DIST/docs/tutorial/Tutorial.html>
- ❑ Doc: https://biopython.org/wiki/Category%3AWiki_Documentation
- ❑ Seq object: <https://biopython.org/wiki/Seq>
- ❑ SeqIO object: <https://biopython.org/wiki/SeqIO>
- ❑ AlignIO: <https://biopython.org/wiki/AlignIO>

BioPerl vs BioPython

- ❑ Comparable features
- ❑ Cleaner Syntax in Python
- ❑ Similar string/regex operations
 - Perl has built-in functions
 - Python has 3rd party support
- ❑ Python has more OO/thread support
- ❑ Perl may be more efficient with regex functions
- ❑ Python has more support for
 - BigData
 - Machine Learning
 - NLP
- ❑ DropBox (Python) vs Amazon/Craigslist/BBC (Perl)

Python Notebook

□ bioPy1.ipynb

□ **Web version:** bioPy1.ipynb