

## 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:

<u>Format</u>	<u>Description</u>	<u>Object</u>
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

More information 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://hmmmer.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

evaluate

The evaluate 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

**eval**

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 environment. 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