BioPerl I:
An Introduction

Jason Stajich
University of California, Berkeley

Topics to cover

- Introduction to BioPerl
- Using Sequence & Feature modules
- Using the modules for BLAST parser
- Accessing sequence databases
- Using the GFF processing modules
  - GFF Database
- Evolutionary data
  - Trees and Population data

Overview of Toolkit

- Bioperl is...
  - A Set of Perl modules for manipulating genomic and other biological data
  - An Open Source Toolkit with many contributors
  - A flexible and extensible system for doing bioinformatics data manipulation

Some things you can do

- Read in sequence data from a file in standard formats (FASTA, GenBank, EMBL, SwissProt,...)
- Manipulate sequences, reverse complement, translate coding DNA sequence to protein.
- Parse a BLAST report, get access to every bit of data in the report

Major Areas covered in Bioperl

- Sequences, Features, Annotations,
- Pairwise alignment reports
- Multiple Sequence Alignments
- Bibliographic data
- Graphical Rendering of sequence tracks
- Database for features and sequences

Additional things

- Gene prediction parsers
- Trees, Parsing Phylogenetic and Molecular Evolution software output
- Population Genetic data and summary statistics
- Taxonomy
- Protein Structure

Practical Examples

- Manipulate a DNA or Protein Sequence
- Read and write different Sequence formats
- Extract sequence annotations and features
- Parse a BLAST report

How the code is organized

- http://cvs.open-bio.org
- bioperl-live - Core packages
- bioperl-run - for running applications
- bioperl-ext - C language extension
- bioperl-db - bioperl BioSQL implementation
- bioperl-pedigree, bioperl-microarray are side-projects

Within bioperl-live (core)

- Bio/ top-level
- Bio::SeqIO - sequence input/output
- Bio::PrimarySeq.pm - basic sequence obj
- Bio::SearchIO - parsers for BLAST, FASTA
- Bio::AlignIO - multiple sequence alignments
- Bio::Tools - misc collection of parsers for different programs

Website

- http://bioperl.org or http://bio.perl.org/
  - Wiki based documentation
  - Project Tracking
- HOWTOs
- Frequently Asked Questions (FAQ)
- News
- Links to online Documentation
- http://bugzilla.open-bio.org - bug tracking

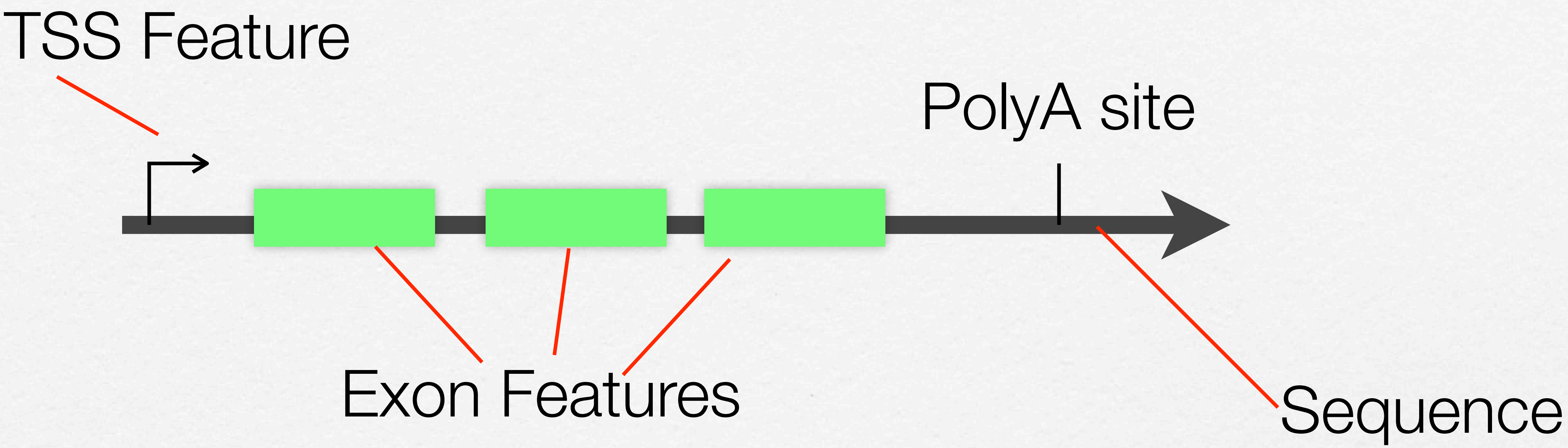
Anatomy of a Bioperl Module

- perldoc Module -- perldoc Bio::SeqIO
- SYNOPSIS -- runnable code
- DESCRIPTION -- summary about the module
- Each module will have methods that are documented.
- Don’t be afraid to look at the raw source of a module - try:
    - perldoc -m Bio::SeqIO::fasta | less

A Tour: Core Objects

- Bioperl Sequences, Features, Locations, Annotations
- Sequence searching & pairwise alignments
- Multiple Sequence Alignments

Sequences and Features



Genomic Sequence with
3 exons
1 Transcript Start Site (TSS)
1 Poly-A Site

Sequence File Formats

- Simple formats - without features
  - FASTA (Pearson), Raw, GCG
- Rich Formats - with features and annotations
  - GenBank, EMBL
  - Swissprot, GenPept
  - XML - BSML, CHAOS, GAME, TIGRXML, CHADO

```
>ID Description(Free text)
AGTGATGATAGTGAGTAGGA

>gi|number|emb|ACCESSION
AGATAGTAGGGATAGAG

>gi|number|sp|BOSS_7LES
MTMFWQQNVDHQSDEQDKQAKGAAPTKRLN
```

Rich Formats

- Combine
  - Sequence data
  - Bibliographic references
  - Taxonomic information
  - Features
  - Annotations

GenBank Format

```
[illegible]
DEFINITION  [illegible]
            sequence, clone p427/428 right end.
ACCESSION   U63596
NID         g2393749
KEYWORDS    .
SOURCE      Dictyostelium discoideum.
  ORGANISM  Dictyostelium discoideum
            Eukaryota; Dictyosteliida; Dictyostelium.
REFERENCE   1  (bases 1 to 310)
  AUTHORS   Wells,D.J.
  TITLE     Tdd-4, a DNA transposon of Dictyostelium that encodes proteins
            similar to LTR retroelement integrases
  JOURNAL   Nucleic Acids Res. 27 (11), 2408-2415 (1999)
FEATURES             Location/Qualifiers
     source          1..310
                     /organism="Dictyostelium discoideum"
                     /strain="AX4"
                     /db_xref="taxon:44689"
                     /clone="p427/428"
     misc_feature    5.12
                     /note="Fuzzy location"
     misc_feature    join(J00194:(100..202),1..245,256..258)
                     /note="Location partly in another entry"
BASE COUNT      118 a     46 c     67 g     79 t
ORIGIN
        1 gtgacagttg gctgtcagac atacaatgat tgtttagaag aggagaagat
tgatccggag
       61 taccgtgata gtattttaaa aactatgaaa gcgggaatac ttaatggtaa
actagttaga
```

EMBL Format

```
ID   U63596     standard; genomic DNA; INV; 310 BP.
XX
AC   U63596;
XX
SV   U63596.1
XX
DT   20-SEP-1997 (Rel. 52, Created)
DT   17-MAY-1999 (Rel. 59, Last updated, Version 5)
XX
DE   Dictyostelium discoideum Tdd-4 transposable element flanking sequence,
DE   clone p427/428 right end.
XX
KW   .
XX
OS   Dictyostelium discoideum
OC   Eukaryota; Mycetozoa; Dictyosteliida; Dictyostelium.
XX
RN   [1]
RP   1-310
RX   MEDLINE; 99263047.
RX   PUBMED; 10325432.
RA   Wells D.J.;
RT   "Tdd-4, a DNA transposon of Dictyostelium that encodes proteins similar to
RT   LTR retroelement integrases";
RL   Nucleic Acids Res. 27(11):2408-2415(1999).
XX
```

```
FH   Key             Location/Qualifiers
FH
FT   source          1..310
FT                   /db_xref="taxon:44689"
FT                   /mol_type="genomic DNA"
FT                   /organism="Dictyostelium discoideum"
FT                   /strain="AX4"
FT                   /clone="p427/428"
XX
SQ   Sequence 310 BP; 118 A; 46 C; 67 G; 79 T; 0 other;
     gtgacagttg gctgtcagac atacaatgat tgtttagaag aggagaagat tgatccggag        60
     taccgtgata gtattttaaa aactatgaaa gcgggaatac ttaatggtaa actagttaga       120
     ttatgtgacg tgccaagggg tgtagatgta gaaattgaaa caactggtct aaccgattca       180
     gaaggagaaa gtgaatcaaa agaagaagag tgatgatgaa tagccaccat tactgcatac       240
     tgtagccctt acccttgtcg caccattagc cattaataaa aataaaaaat tatataaaaa       300
     ttacacccat                                                              310
//
```

Swissprot Format

```
ID   7LES_DROME     STANDARD;      PRT;  2554 AA.
AC   P13368; Q9TYI0; Q9U5V7; Q9VZ36;
DT   01-JAN-1990 (Rel. 13, Created)
DT   16-OCT-2001 (Rel. 40, Last sequence update)
DT   15-JUN-2004 (Rel. 44, Last annotation update)
DE   Sevenless protein (EC 2.7.1.112).
GN   SEV OR HD-265 OR CG18085.
OS   Drosophila melanogaster (Fruit fly).
OC   Eukaryota; Metazoa; Arthropoda; Hexapoda; Insecta; Pterygota;
OC   Neoptera; Endopterygota; Diptera; Brachycera; Muscomorpha;
OC   Ephydroidea; Drosophilidae; Drosophila.
OX   NCBI_TaxID=7227;
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=Canton-S;
RX   MEDLINE=88282538; PubMed=2840202;
RA   Basler K., Hafen E.;
RT   "Control of photoreceptor cell fate by the sevenless protein requires
RT   a functional tyrosine kinase domain.";
RL   Cell 54:299-311(1988).
RN   [2]
RP   SEQUENCE FROM N.A.
RC   STRAIN=Oregon-R;
RX   MEDLINE=88329706; PubMed=3138161;
RA   Bowtell D.L.L., Simon M.A., Rubin G.M.;
RT   "Nucleotide sequence and structure of the sevenless gene of
RT   Drosophila melanogaster.";
RL   Genes Dev. 2:620-634(1988).
```

```
CC         instruct a cell to differentiate into an R7 photoreceptor. The
CC         ligand for sev is the boss (bride of sevenless) protein on the
CC         surface of the neighboring R8 cell.
CC     -!- CATALYTIC ACTIVITY: ATP + a protein tyrosine = ADP + protein
CC         tyrosine phosphate.
CC     -!- SUBUNIT: May form a complex with drk and Sos.
CC     -!- DOMAIN: It is unclear whether the potential membrane spanning
CC         region near the N-terminus is present as a transmembrane domain in
CC         the native protein or serves as a cleaved signal sequence.
CC     -!- SIMILARITY: Belongs to the Tyr family of protein kinases. Insulin
CC         receptor subfamily.
CC     -!- SIMILARITY: Contains 7 fibronectin type III domains.
CC
DR     EMBL; J03158; AAA28882.1; -.
DR     EMBL; X13666; CAA31960.1; ALT_INIT.
DR     EMBL; X13666; CAB55310.1; -.
DR     EMBL; AE003484; AAF47992.2; -.
DR     EMBL; AJ002917; CAA05752.1; -.
DR     PIR; A28912; TVFF7L.
DR     HSSP; P08069; 1JQH.
DR     FlyBase; FBgn0003366; sev.
DR     GO; GO:0005886; C:plasma membrane; IDA.
DR     GO; GO:0004713; F:protein-tyrosine kinase activity; IDA.
DR     GO; GO:0045467; P:R7 development; NAS.
DR     GO; GO:0008293; P:torso signaling pathway; NAS.
DR     InterPro; IPR003961; FN_III.
DR     InterPro; IPR008957; FN_III-like.
DR     InterPro; IPR000033; Ldl_receptor_rep.
DR   InterPro; IPR002011; ReceptyrkinsII.
DR   InterPro; IPR001245; Tyr_pkinase.
DR   InterPro; IPR008266; Tyr_pkinase_AS.
DR   Pfam; PF00041; fn3; 6.
DR   Pfam; PF00069; Pkinase; 1.
DR   PRINTS; PR00109; TYRKINASE.
DR   ProDom; PD000001; Prot_kinase; 1.
DR   SMART; SM00060; FN3; 6.
DR   SMART; SM00135; LY; 2.
DR   SMART; SM00219; TyrKc; 1.
DR   PROSITE; PS50853; FN3; 7.
DR   PROSITE; PS00107; PROTEIN_KINASE_ATP; 1.
DR   PROSITE; PS50011; PROTEIN_KINASE_DOM; 1.
DR   PROSITE; PS00109; PROTEIN_KINASE_TYR; 1.
DR   PROSITE; PS00239; RECEPTOR_TYR_KIN_II; 1.
KW   Transferase; Tyrosine-protein kinase; Receptor; Vision; Transmembrane;
KW   Glycoprotein; ATP-binding; Phosphorylation; Repeat.
FT   DOMAIN        1   2123       Extracellular (Potential).
FT   TRANSMEM   2124   2147       Potential.
FT   DOMAIN     2148   2554       Cytoplasmic (Potential).
FT   DOMAIN      311    431       Fibronectin type-III 1.
FT   DOMAIN      436    528       Fibronectin type-III 2.
FT   DOMAIN      822    921       Fibronectin type-III 3.
FT   DOMAIN     1298   1392       Fibronectin type-III 4.
FT   DOMAIN     1680   1794       Fibronectin type-III 5.
FT   DOMAIN     1797   1897       Fibronectin type-III 6.
FT   DOMAIN     1898   1988       Fibronectin type-III 7.
FT   DOMAIN     2038   2046       Poly-Arg.
FT   CONFLICT   2271   2271       C -> R (in Ref. 1).
SQ   SEQUENCE   2554 AA;  287022 MW;  09E238A0F27684F8 CRC64;
     MTMFWQQNVD HQSDEQDKQA KGAAPTKRLN ISFNVKIAVN VNTKMTTTHI NQQAPGTSSS
     SSNSQNASPS KIVVRQQSSS FDLRQQLARL GRQLASGQDG HGGISTILII NLLLLILLSI
     CCDVCRSHNY TVHQSPEPVS KDQMRLLRPK LDSDVVEKVA IWHKHAAAAP PSIVEGIAIS
     SRPQSTMAHH PDDRDRDRDP SEEQHGVDER MVLERVTRDC VQRCIVEEDL FLDEFGIQCE
     KADNGEKCYK TRCTKGCAQW YRALKELESC QEACLSLQFY PYDMPCIGAC EMAQRDYWHL
     QRLAISHLVE RTQPQLERAP RADGQSTPLT IRWAMHFPEH YLASRPFNIQ YQFVDHHGEE
     LDLEQEDQDA SGETGSSAWF NLADYDCDEY YVCEILEALI PYTQYRFRFE LPFGENRDEV
     LYSPATPAYQ TPPEGAPISA PVIEHLMGLD DSHLAVHWHP GRFTNGPIEG YRLRLSSSEG
```

Sequences, Features, Annotations

- Sequence - DNA, RNA, AA
  - Feature container
- Feature - Information with a Sequence Location
- Annotation - Information without explicit Sequence location

Parsing Sequences

- Bio::SeqIO
  - multiple drivers: genbank, embl, fasta,...
- Sequence objects
  - Bio::PrimarySeq
  - Bio::Seq
  - Bio::Seq::RichSeq

Investigate the Sequence object

- Common (Bio::PrimarySeq) methods
  - seq() - get the sequence as a string
  - length() - get the sequence length
  - subseq($s,$e) - get a subseqeunce
  - translate(...) - translate to protein [DNA]
  - revcom() - reverse complement [DNA]
  - display_id() - identifier string
  - description() - description string

Using a Sequence

```
use Bio::PrimarySeq;
my $str = “ATGAATGATGAA”;
my $seq = Bio::PrimarySeq->new(-seq => $str,
                    -display_id=>”example”);

print “id is “, $seq->display_id,”\n”;
print $seq->seq, “\n”;
my $revcom = $seq->revcom;
print $revcom->seq, “\n”;
print “frame1=”,$seq->translate->seq,“\n”;

id is example
ATGAATGATGAA
TTCATCATTCAT
trans frame1=MNDE
```

Sequence Features

- Bio::SeqFeatureI - interface - GFF derived
  - start(), end(), strand() for location information
  - location() - Bio::LocationI object (to represent complex locations)
  - score,frame,primary_tag, source_tag - feature information
  - spliced_seq() - for attached sequence, get the sequence spliced.

The GFF format

- “Generic Feature Format”
- tab delimited
- sequence_id, source, type, start, stop, score, strand, frame, description
- Different versions of GFF: GFF1, GFF2 (GTF), GFF3
  - Variation is in how the description column is formatted

GFF3

- http://song.sourceforge.net/gff3.shtml
- ‘type’ column values must be in the **sequence ontology**
- description col must have ID or Parent field to describe relationships to other features
- gene feature
    - mRNA feature
        - CDS feature

Sequence Feature (cont.)

- Bio::SeqFeature::Generic
  - add_tag_value($tag,$value) - add a tag/value pair
  - get_tag_value($tag) - get all the values for this tag
  - has_tag($tag) - test if a tag exists
  - get_all_tags() - get all the tags

Using a SeqFeature

```
#!/usr/bin/perl -w
use strict;
use Bio::SeqFeature::Generic;

my $f = Bio::SeqFeature::Generic->new
(-start => 10,-end    => 20,-strand => 1, -seq_id=> ‘hs.1’,
-primary_tag => 'CDS',
-source_tag  => 'genscan',
-score => 30,
-tag => { 'Parent' => 'Gene1' });

printf "start=%d end=%d strand=%d primary_tag=%s source_tag=%s\n",
$f->start,$f->end, $f->strand,
$f->primary_tag,
$f->source_tag;
for my $tag ($f->get_all_tags ) {
 print "Tag=$tag: ";
 for my $val ($f->get_tag_values($tag) ) {
  print "$val ";
 }
 print "\n";
}
```

```
start=10 end=20 strand=1 primary_tag=CDS source_tag=genscan
Tag=Parent: Gene1
```

Read and Writing SeqFeatures

```
#!/usr/bin/perl -w
use strict;
use Bio::SeqFeature::Generic;
use Bio::Tools::GFF;

my $f = Bio::SeqFeature::Generic->new
(-start => 10,
 -end    => 20,
 -strand => 1,
 -seq_id=> ‘hs.1’,
 -primary_tag => 'CDS',
 -source_tag  => 'genscan',
 -score => 30,
 -tag => { 'Parent' => 'Gene1' });

my $out = Bio::Tools::GFF->new(-gff_version => 3,
                               -file        => “>output.gff”);
$out->write_feature($f);
```

GFF writing results

```
hs.1 genscan CDS 10 20 30 + . Parent=Gene1
```

Sequences with Features

- Bio::Seq objects have the methods
  - add_SeqFeature($feature) - attach feature(s)
  - get_SeqFeatures() - get all the attached features.
  - species() - a Bio::Species object
  - annotation() - Bio::Annotation::Collection

Reading in a Sequence

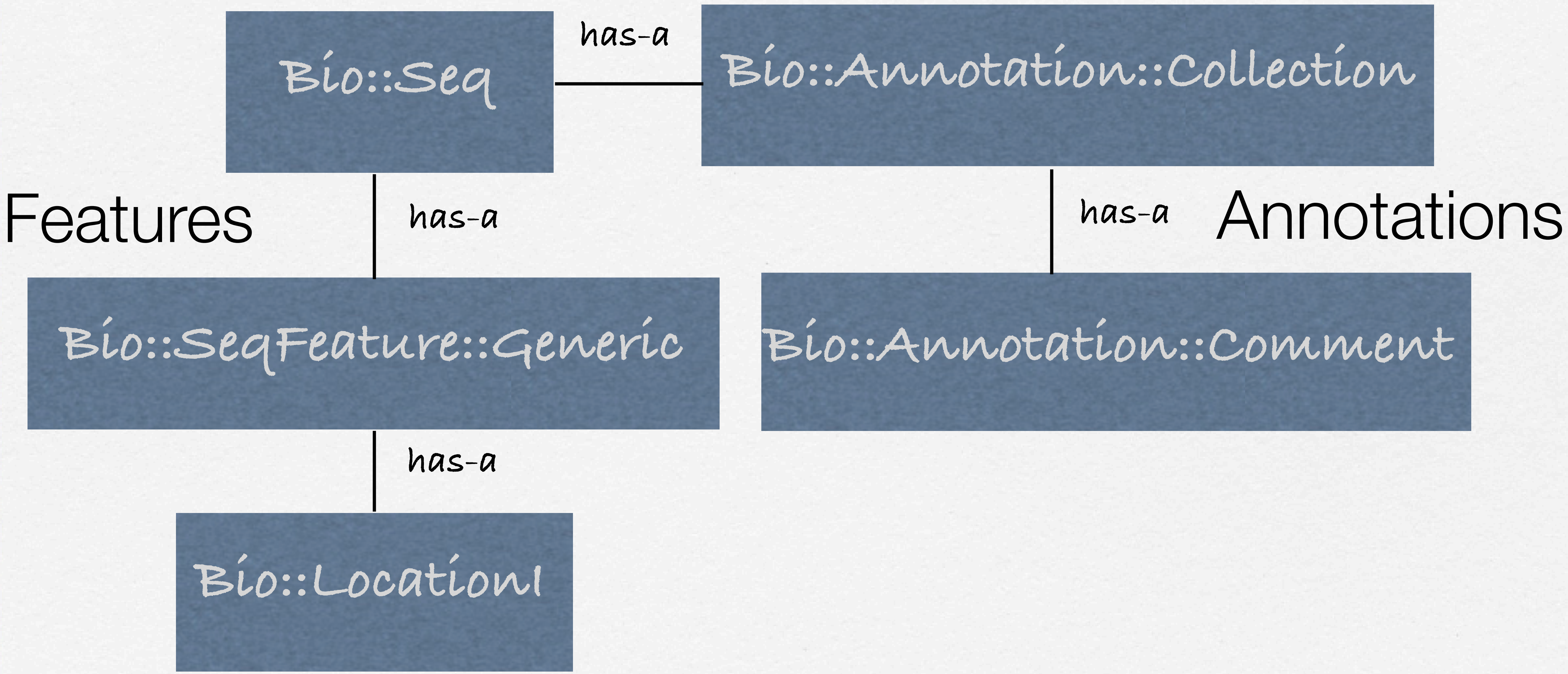
```
use Bio::SeqIO;
my $in = Bio::SeqIO->new(-format => ‘genbank’,
                         -file   => ‘file.gb’);
while( my $seq = $in->next_seq ) {

 print “sequence name is “, $seq->display_id,
       “ length is ”,$seq->length,”\n”;
 print “there are “,(scalar $seq->get_SeqFeatures),
       “ features attached to this sequence and “,
 scalar $seq->annotation->get_Annotations(’reference’),
       “ reference annotations\n”;
}
```

Annotations

- Each Bio::Seq has a Bio::Annotation::Collection via $seq->annotation()
- Annotations are stored with keys like ‘comment’ and ‘reference’
- `@com=$annotation->get_Annotations(’comment’)`
- `$annotation->add_Annotation(’comment’, $an)`

Annotations

- Annotation::Comment
  - comment field
- Annotation::Reference
  - author,journal,title, etc
- Annotation::DBLink
  - database, primary_id, optional_id, comment
- Annotation::SimpleValue



Bio::Seq

has-a

Bio::Annotation::Collection

Features

has-a

has-a

Annotations

Bio::SeqFeature::Generic

Bio::Annotation::Comment

has-a

Bio::LocationI

Reading and Writing Sequences

- Bio::SeqIO
  - fasta, genbank, embl, swissprot,...
- Takes care of writing out associated features and annotations
- Two functions
  - next_seq (reading sequences)
  - write_seq (writing sequences)

Writing a Sequence

```
use Bio::SeqIO;
# Let’s convert swissprot to fasta format
my $in  = Bio::SeqIO->new(-format => ‘swiss’,
                          -file   => ‘file.sp’);
my $out = Bio::SeqIO->new(-format => ‘fasta’,
                          -file   => ‘>file.fa’);`
while( my $seq = $in->next_seq ) {
 $out->write_seq($seq);
}
```

Sequence Database Searching

```
        opt      E()
< 20   323     0:==
  22     0     0:           one = represents 184 library sequences
  24     2     0:=
  26    12     2:*
  28    61    26:*
  30   211   157:*=
  32   664   607:===*
  34  1779  1645:========*=
  36  3558  3379:==================*=
  38  5908  5584:===============================*==
  40  8049  7790:===========================================*=
  42 10001  9522:====================================================*===
  44 10660 10503:==========================================================*
  46 10987 10698:===========================================================*=
  48 10332 10242:========================================================*=
  50  9053  9346:===================================================*
  52  7736  8217:============================================ *
  54  6828  7018:=======================================*
  56  5448  5863:============================== *
  58  4484  4813:========================= *
  60  3818  3899:=====================*
  62  2942  3126:================*
  64  2407  2486:=============*
  66  1866  1965:==========*
  68  1495  1545:========*
  70  1169  1211:======*
  72   886   946:=====*
  74   708   738:====*
  76   542   574:===*
  78   451   446:==*
  80   355   347:=*
  82   271   265:=*
  84   211   210:=*
  86   151   163:*
  88   104   126:*           inset = represents 3 library sequences
  90   101    97:*
  92    78    75:*          :========================*=
  94    56    58:*          :===================*
  96    38    45:*          :============= *
  98    26    35:*          :=========  *
 100    26    27:*          :========*
 102    20    21:*          :======*
 104    13    16:*          :=====*
 106    22    12:*          :===*====
 108    10    10:*          :===*
 110     5     7:*          :==*
 112     4     6:*          :=*
 114     4     4:*          :=*
 116     3     3:*          :*
 118     9     3:*          :*==
>120   110     2:*          :*====================================
```

A Detailed look at BLAST parsing

- 3 Components
  - Result: Bio::Search::Result::ResultI
  - Hit: Bio::Search::Hit::HitI
  - HSP: Bio::Search::HSP::HSPI

```
Copyright (C) 1996-2000 Washington University, Saint Louis, Missouri USA.
All Rights Reserved.

Reference:  Gish, W. (1996-2000) http://blast.wustl.edu

Query=  BOSS_DROME Bride of sevenless protein precursor.
        (896 letters)

Database:  wormpep87
           20,881 sequences; 9,238,759 total letters.
Searching....10....20....30....40....50....60....70....80....90....100% done

                                                                      Smallest
                                                                        Sum
                                                                 High  Probability
Sequences producing High-scoring Segment Pairs:                  Score  P(N)      N

F35H10.10 CE24945    status:Partially_confirmed TR:Q20073...   182  4.9e-11   1
M02H5.2 CE25951    status:Predicted TR:Q966H5 protein_id:...   86  0.15       1
ZC506.4 CE01682  locus:mgl-1 metatrophic glutamate recept...   91  0.18       1
F23D12.2 CE05700    status:Partially_confirmed TR:Q19761 ...    73  0.45      3

>F35H10.10 CE24945    status:Partially_confirmed TR:Q20073
           protein_id:AAA81683.2
        Length = 1404

 Score = 182 (69.1 bits), Expect = 4.9e-11, P = 4.9e-11
 Identities = 75/315 (23%), Positives = 149/315 (47%)

Query:   511 YPFLFDGESVMFWRIKMDTWVATGLTAAILGLIATLAILVFIVVRISLGDVFEGNPTTSI 570
             Y  +F+  +   WR     +V   L   ++  +  +A+LV ++V++ L  V +GN +  I
Sbjct:  1006 YQSVFEHITTGHWRDHPHNYVLLALITVLV--VVAIAVLVLVLVKLYLR-VVKGNQSLGI 1062
```

```
use Bio::SearchIO;
my $cutoff = ’0.001’;
my $file = ‘BOSS_Ce.BLASTP’,
my $in = new Bio::SearchIO(-format => ‘blast’,
                           -file    => $file);
while( my $r = $in->next_result ) {
  print "Query is: ", $r->query_name, " ",
  $r->query_description," ",$r->query_length," aa\n";
  print " Matrix was ", $r->get_parameter(’matrix’), "\n";
  while( my $h = $r->next_hit ) {
    last if $h->significance > $cutoff;
    print "Hit is ", $h->name, "\n";
    while( my $hsp = $h->next_hsp ) {
     print " HSP Len is ", $hsp->length(’total’), " ",
           " E-value is ", $hsp->evalue, " Bit score ",
           $hsp->score, " \n",
           " Query loc: ",$hsp->query->start, " ",
           $hsp->query->end," ",
           " Sbject loc: ",$hsp->hit->start, " ",
           $hsp->hit->end,"\n";
    }
  }
}
```

BLAST Script Results

```
Query is: BOSS_DROME Bride of sevenless protein
precursor. 896 aa
Matrix was BLOSUM62
Hit is F35H10.10
HSP Len is 315 E-value is 4.9e-11 Bit score 182
Query loc: 511 813 Sbject loc: 1006 1298
HSP Len is 28 E-value is 1.4e-09 Bit score 39
Query loc: 508 535 Sbject loc: 427 454
```

FASTA Report

```
 Version 3.4t20 Sep 26, 2001
Please cite:
 W.R. Pearson & D.J. Lipman PNAS (1988) 85:2444-2448

Query library BOSS_DROME.aa vs /blast/wormpep87 library
searching /blast/wormpep87 library

  1>>>BOSS_DROME Bride of sevenless protein precursor. - 896 aa
 vs  /blast/wormpep87 library
9238759 residues in 20881 sequences
  Expectation_n fit: rho(ln(x))= 5.6098+/-0.000519; mu= 12.8177+/- 0.030
 mean_var=107.8223+/-22.869, 0's: 0 Z-trim: 2  B-trim: 0 in 0/62
 Lambda= 0.123515
 Kolmogorov-Smirnov  statistic: 0.0333 (N=29) at  48

FASTA (3.45 Mar 2002) function [optimized, BL50 matrix (15:-5)] ktup: 2
 join: 38, opt: 26, gap-pen: -12/-2, width:  16
 Scan time:  9.680
The best scores are:                                      opt bits E(20881)
F35H10.10 CE24945    status:Partially_confirmed T  (1404)  207  48.5 6.8e-05
T06E4.11 CE06377  locus:pqn-63  status:Predicted   ( 275)  122  32.6      0.8
C33B4.3 CE01508   ankyrin and proline rich domain  (1110)  124  33.6      1.6
Y48C3A.8 CE22141    status:Predicted TR:Q9NAG3 pr  ( 291)  110  30.5      3.7
Y34D9A.2 CE30217    status:Partially_confirmed TR  ( 326)  108  30.2      5.1
K06H7.3 CE26941   Isopentenyl-diphosphate delta i  ( 618)  107  30.3      8.9
F44B9.8 CE29044   ARPA status:Partially_confirmed  ( 388)  104  29.5      9.4

>>F35H10.10 CE24945    status:Partially_confirmed TR:Q20  (1404 aa)
 initn:  94 init1:  94 opt: 207  Z-score: 197.9  bits: 48.5 E(): 6.8e-05
Smith-Waterman score: 275;  22.527% identity (27.152% ungapped) in 728 aa
overlap (207-847:640-1330)

        180       190       200       210       220          230
BOSS_D RAISIDNASLAENLLIQEVQFLQQCTTYSMGIFVDWELYKQLESVIKD---LEYNIWPIP
```

FASTA Parsing Script

```
use Bio::SearchIO;
my $cutoff = ’0.001;
my $file = ‘BOSS_Ce.FASTP’,
my $in = new Bio::SearchIO(-format => ‘fasta’,
                           -file    => $file);
while( my $r = $in->next_result ) {
  print "Query is: ", $r->query_name, " ",
  $r->query_description," ",$r->query_length," aa\n";
  print " Matrix was ", $r->get_parameter(’matrix’), "\n";
  while( my $h = $r->next_hit ) {
    last if $h->significance > $cutoff;
    print "Hit is ", $h->name, "\n";
    while( my $hsp = $h->next_hsp ) {
     print " HSP Len is ", $hsp->length(’total’), " ",
           " E-value is ", $hsp->evalue, " Bit score ",
           $hsp->score, " \n",
           " Query loc: ",$hsp->query->start, " ",
           $hsp->query->end," ",
           " Sbject loc: ",$hsp->hit->start, " ",
           $hsp->hit->end,"\n";
    }
  }
}
```

FASTA Script Results

```
Query is: BOSS_DROME Bride of sevenless protein
precursor. 896 aa
Matrix was BL50
Hit is F35H10.10
HSP Len is 728 E-value is 6.8e-05 Bit score 197.9
Query loc: 207 847 Sbject loc: 640 1330
```

Using the Search::Result object

```
use Bio::SearchIO;
use strict;
my $parser = new Bio::SearchIO(-format => ‘blast’,
                               -file => ‘file.bls’);
while( my $result = $parser->next_result ){
  print “query name=“, $result->query_name, “ desc=”,
        $result->query_description, “, len=”,$result-
>query_length,“\n”;
  print “algorithm=“, $result->algorithm, “\n”;
  print “db name=”, $result->database_name, “ #lets=”,
  $result->database_letters, “ #seqs=”,$result->database_entries,
“\n”;
  print “available params  “, join(’,’,
        $result->available_parameters),”\n”;
  print “available stats “, join(’,’,
        $result->available_statistics), “\n”;
  print “num of hits “, $result->num_hits, “\n”;
}
```

Using the Search::Hit Object

```
use Bio::SearchIO;
use strict;
my $parser = new Bio::SearchIO(-format => ‘blast’,
                               -file => ‘file.bls’);
while( my $result = $parser->next_result ){
  while( my $hit = $result->next_hit ) {
   print “hit name=”,$hit->name, “ desc=”, $hit->description,
         “\n len=”, $hit->length, “ acc=”, $hit->accession,
”\n”;
   print “raw score “, $hit->raw_score, “ bits “, $hit->bits,
         “ significance/evalue=“, $hit->evalue, “\n”;
  }
}
```

Cool Hit Methods

- start(), end() - get overall alignment start and end for all HSPs
- strand() - get best overall alignment strand
- matches() - get total number of matches across entire set of HSPs (can specify only exact ‘id’ or conservative ‘cons’)

Using the Search::HSP Object

```
use Bio::SearchIO;
use strict;
my $parser = new Bio::SearchIO(-format => ‘blast’, -file => ‘file.bls’);
while( my $result = $parser->next_result ){
  while( my $hit = $result->next_hit ) {
     while( my $hsp = $hit->next_hsp ) {
       print “hsp evalue=“, $hsp->evalue, “ score=“ $hsp->score, “\n”;
       print “total length=“, $hsp->hsp_length, “ qlen=”,
             $hsp->query->length, “ hlen=”,$hsp->hit->length, “\n”;
       print “qstart=”,$hsp->query->start, “ qend=”,$hsp->query->end,
             “ qstrand=“, $hsp->query->strand, “\n”;
       print “hstart=”,$hsp->hit->start, “ hend=”,$hsp->hit->end,
             “ hstrand=“, $hsp->hit->strand, “\n”;
       print “percent identical “, $hsp->percent_identity,
             “ frac conserved “, $hsp->frac_conserved(), “\n”;
       print “num query gaps “, $hsp->gaps(’query’), “\n”;
       print “hit str =”, $hsp->hit_string, “\n”;
       print “query str =”, $hsp->query_string, “\n”;
       print “homolog str=”, $hsp->homology_string, “\n”;
  }
}
```

Cool HSP methods

- rank() - order in the alignment (which you could have requested, by score, size)
- matches - overall number of matches
- seq_inds - get a list of numbers representing residue positions which are
  - conserved, identical, mismatches, gaps

SearchIO system

- BLAST (WU-BLAST, NCBI, XML, PSIBLAST, BL2SEQ, MEGABLAST, TABULAR (-m8/m9))
- FASTA (m9 and m0)
- HMMER (hmmpfam, hmmsearch)
- UCSC formats (WABA, AXT, PSL)
- Gene based alignments
  - Exonerate, SIM4, {Gene,Genome}wise

SearchIO reformatting

- Supports output of Search reports as well
- Bio::SearchIO::Writer
  - “BLAST flavor” HTML, Text
  - Tabular Report Format

Bioperl Reformatted HTML of BLASTP Search Report for gi|6319512|ref|NP_009594.1|

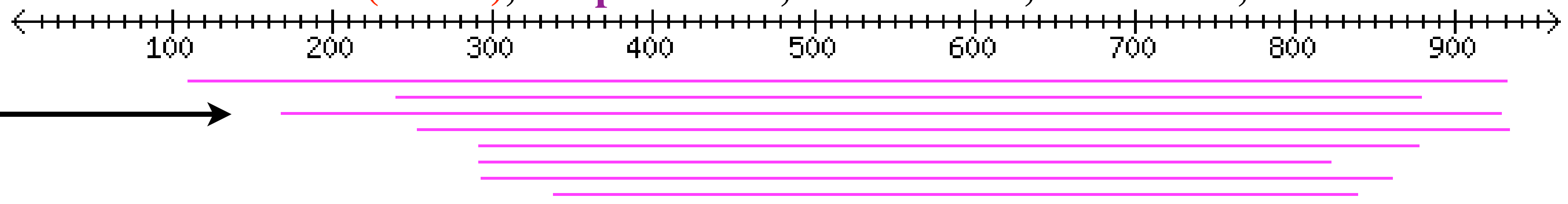
BLASTP 2.0MP-WashU [04-Feb-2003] [linux24-i686-ILP32F64 2003-02-04T19:05:09]

Copyright (C) 1996-2000 Washington University, Saint Louis, Missouri USA.
All Rights Reserved.

Reference: Gish, W. (1996-2000) http://blast.wustl.edu

Hit Overview

Score: Red= (>=200), Purple 200-80, Green 80-50, Blue 50-40, Black <40



Build custom overview (Bio::Graphics)

Query= gi|6319512|ref|NP_009594.1| chitin synthase 2; Chs2p [Saccharomyces cerevisiae]
(963 letters)

Database: cneoA_WI.aa
9,645 sequences; 2,832,832 total letters

Hyperlink to external resources

Sequences producing significant alignments:	**Score (bits)**	**E value**
cneo_WIH99_157.Gene2 Start=295 End=4301 Strand=1 Length=912 ExonCt=24	1650	1.6e-173
cneo_WIH99_63.Gene181 Start=154896 End=151527 Strand=-1 Length=876 ExonCt=13	1441	3.9e-149
cneo_WIH99_133.Gene1 Start=15489 End=19943 Strand=1 Length=1017 ExonCt=23	1357	3e-142
cneo_WIH99_45.Gene2 Start=84 End=3840 Strand=1 Length=839 ExonCt=25	1311	1.5e-138
cneo_WIH99_112.Gene165 Start=122440 End=118921 Strand=-1 Length=1036 ExonCt=9	198	1.2e-15
cneo_WIH99_11.Gene7 Start=39355 End=42071 Strand=1 Length=761 ExonCt=9	172	6.4e-13
cneo_WIH99_60.Gene9 Start=36153 End=32819 Strand=-1 Length=1020 ExonCt=5	166	1.2e-12
cneo_WIH99_106.Gene88 Start=242538 End=238790 Strand=-1 Length=1224 ExonCt=3	157	6.3e-09

Hyperlink to alignment part of report

Turning BLAST into HTML

```
use Bio::SearchIO;
use Bio::SearchIO::Writer::HTMLResultWriter;

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

  my $writer = new
Bio::SearchIO::Writer::HTMLResultWriter();
  my $out = new Bio::SearchIO(-writer => $writer
                              -file    => “>file.html”);
  $out->write_result($in->next_result);
```

Turning BLAST into HTML

```
# to filter your output
  my $MinLength = 100; # need a variable with scope outside the method
  sub hsp_filter {
      my $hsp = shift;
      return 1 if $hsp->length('total') > $MinLength;
  }
  sub result_filter {
      my $result = shift;
      return $hsp->num_hits > 0;
  }

  my $writer = new Bio::SearchIO::Writer::HTMLResultWriter
                     (-filters => { 'HSP' => \&hsp_filter} );
  my $out = new Bio::SearchIO(-writer => $writer);
  $out->write_result($in->next_result);

  # can also set the filter via the writer object
  $writer->filter('RESULT', \&result_filter);
```

Summary

- Lots of modules to do lots of things
- How to find out what exists?
- Read HOWTOs, bptutorial, Browse the docs website - http://doc.bioperl.org/
- Ask on-list bioperl-l@bioperl.org