BioPerl: The evolution of a bioinformatics toolkit

Jason Stajich
Duke University

BioPerl

- Perl modules for life sciences programming
  - Parsers and Objects
  - Some analysis tools
  - Bioinformatics Duct tape
- International collaboration including biologists, computational sciences

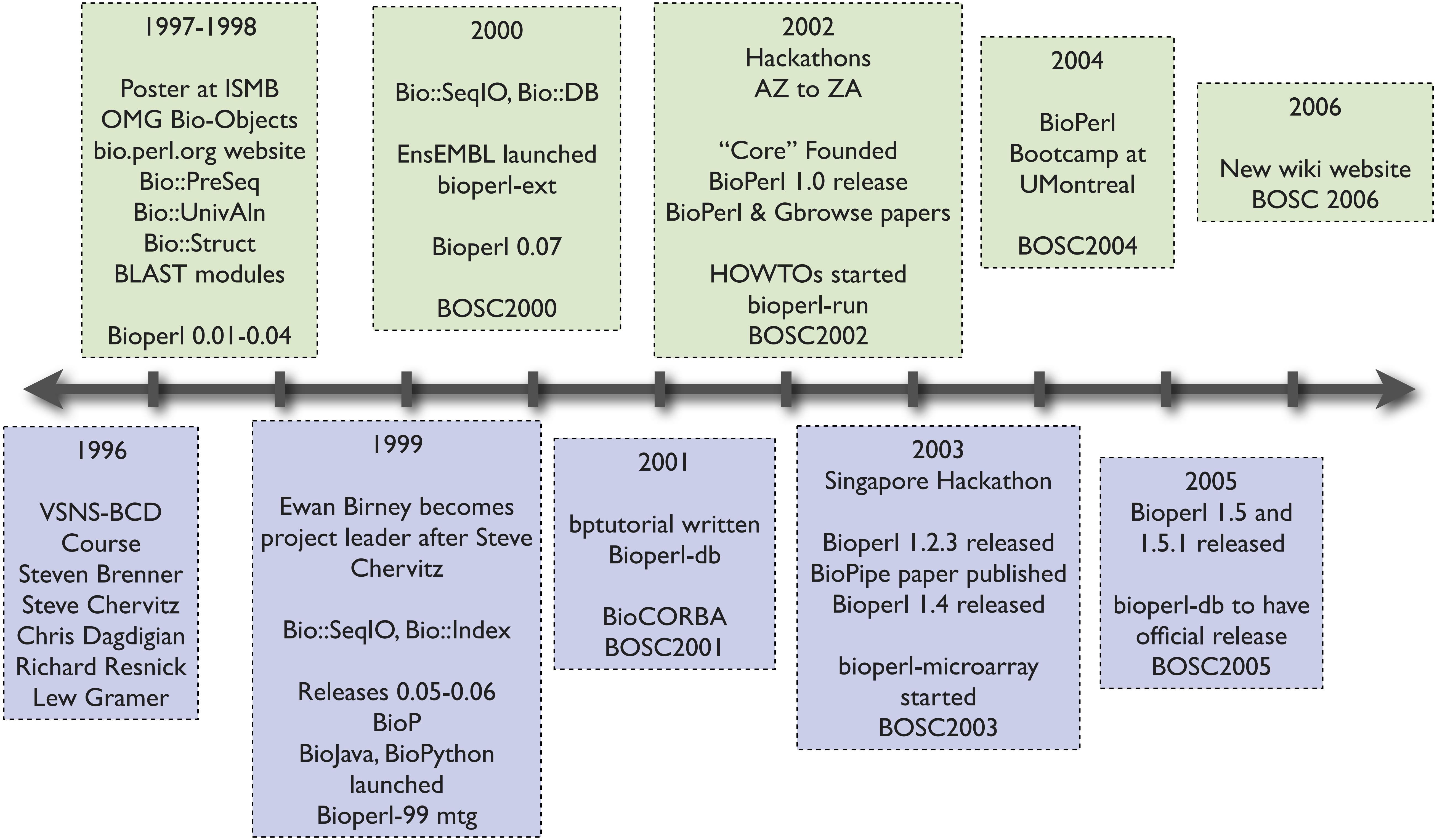
Example code

```
use Bio::SeqIO;
my $in = Bio::SeqIO->new(-format => ‘genbank’,
                         -file   => ‘input.gbk’);
my $out = Bio::SeqIO->new(-format => ‘fasta’,
                          -file   => ‘>output.fas’);
while( my $seq = $in->next_seq ) {
 next unless $seq->length > 50;
 $out->write_seq($seq);
}
```

Who uses it?

- GMOD - Gbrowse, BLAST wrappers
- Many academic labs of all sizes
- Sequencing centers
- Early basis of Ensembl
- Some commercial bioinformatics software

Timeline



1996

VSNS-BCD
Course
Steven Brenner
Steve Chervitz
Chris Dagdigian
Richard Resnick
Lew Gramer

1997-1998

Poster at ISMB
OMG Bio-Objects
bio.perl.org website
Bio::PreSeq
Bio::UnivAln
Bio::Struct
BLAST modules

Bioperl 0.01-0.04

1999

Ewan Birney becomes project leader after Steve Chervitz

Bio::SeqIO, Bio::Index

Releases 0.05-0.06
BioP
BioJava, BioPython launched
Bioperl-99 mtg

2000

Bio::SeqIO, Bio::DB

EnsEMBL launched
bioperl-ext

Bioperl 0.07

BOSC2000

2001

bptutorial written
Bioperl-db

BioCORBA
BOSC2001

2002
Hackathons
AZ to ZA

“Core” Founded
BioPerl 1.0 release
BioPerl & Gbrowse papers

HOWTOs started
bioperl-run
BOSC2002

2003
Singapore Hackathon

Bioperl 1.2.3 released
BioPipe paper published
Bioperl 1.4 released

bioperl-microarray
started
BOSC2003

2004

BioPerl
Bootcamp at
UMontreal

BOSC2004

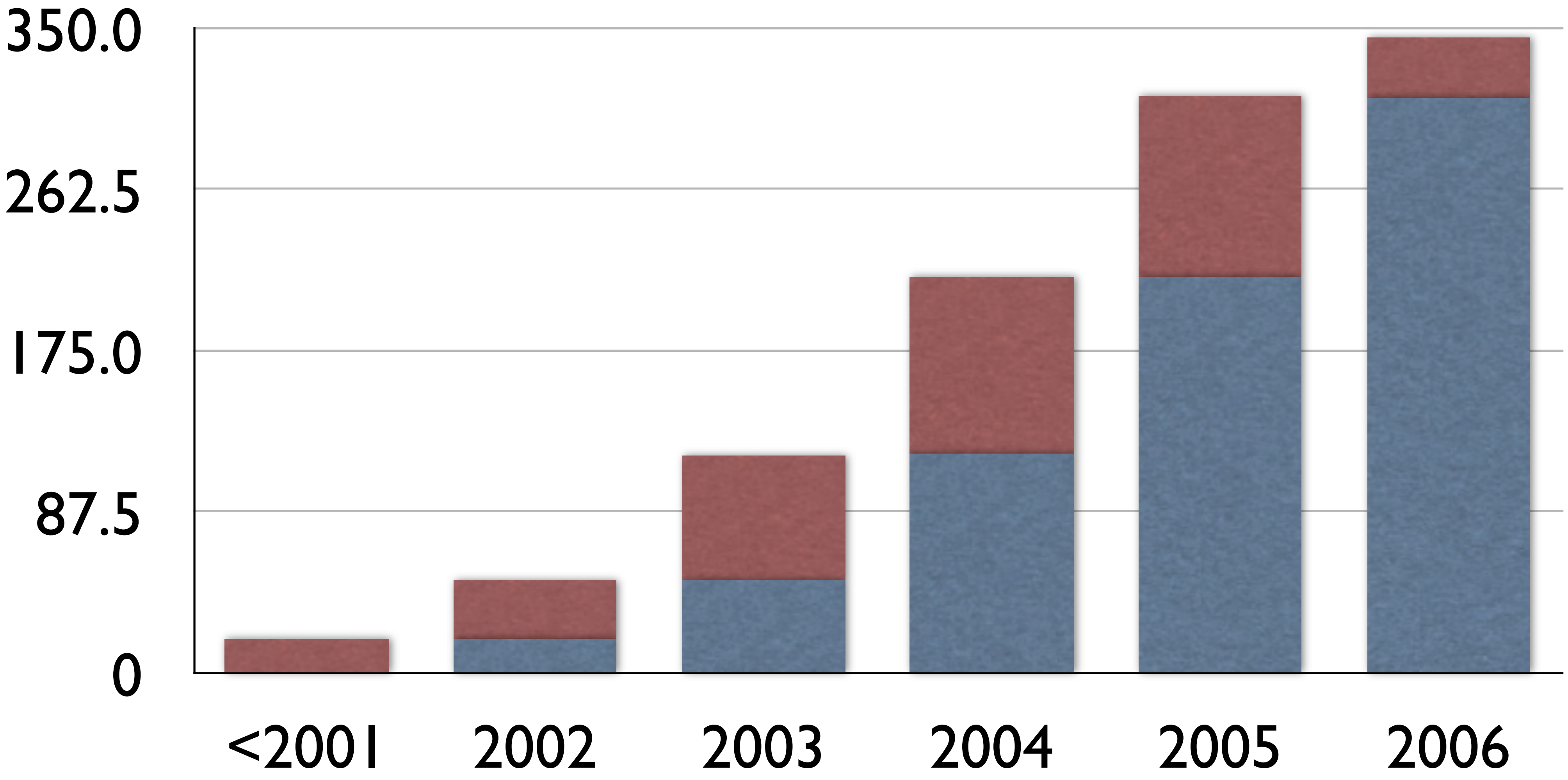
2005
Bioperl 1.5 and
1.5.1 released

bioperl-db to have
official release
BOSC2005

2006

New wiki website
BOSC 2006

Total citations



Measuring sucess

- 2002 Paper has 300+ Citations
- Bits and pieces used in many informatics tools
  - Bio::SearchIO (Rfam, In-Paranoid)
  - Generic Genome Browser (Stein et al, 2002)
  - Comparative Genomics Library (Yandell, Mungall, et al)
  - [EnsEMBL]

Bioperl Taught At Tutorials and Courses

Bootcamp 2004
(Montreal)

CSHL Bioinformatics
Software Courses

Various mini courses
MIT, Duke, EBI, Pasteur

BioPerl is a desired skillset

http://www.ce.com/education/Bioinformatics-Certificate-Program-10082109.htm
Track 1: Suggested electives for computer scientists and IT professionals
- * Advanced Sequence Analysis in Bioinformatics (2 units)
- * Gene Expression and Pathways (2 units)
- * Protein Structure Analysis in Bioinformatics (2 units)
- * Design and Implementation of Bioinformatics Infrastructures (3 units)
- * DNA Microarrays - Principles, Applications and Data Analysis (2 Units)
- * Parallel and Distributed Computing for Bioinformatics (2 units)

Track 2: Suggested electives for molecular biologists and other scientists
- * Introduction to Programming for Bioinformatics II*** (3 units)
- * Introduction to Programming for Bioinformatics III (3 units)
- * **BioPerl for Bioinformatics** (2 units)
- * Design and Implementation of Bioinformatics Infrastructures (3 units)
- * Gene Expression and Pathways (2 units)
- * Protein Structure Analysis in Bioinformatics (2 units)
- * DNA Microarrays - Principles, Applications and Data Analysis (2 units)

http://bioag.byu.edu/botany/homepage/botweb/jobs_Ph.D.htm
Academic Facilities Coordinator II (Facilities)
Bioinformatics Position at the UCR Genomics Institute
UC, Riverside

The Center for Plant Cell Biology (CEPCEB) in the Genomics Institute of the University of California, Riverside, invites applicants for an Academic Facilities Coordinator II position, an academic-track 11-month appointment. Salary for the position is commensurate with education and experience.

The successful applicant will be expected to organize a small bioinformatics team to provide support to The Center for Plant Cell Biology. This team will implement currently available bioinformatics tools including relational database support and will develop user-specific data-mining tools. The appointee will be expected to develop research collaborations with the faculty and teach or organize short courses that will inform that will inform the local community about the available bioinformatics resources. Applicants must have a Ph.D. in the Biological Sciences (Plant Biology is preferred). Applicants with experience in leading a bioinformatics group will be given preference. Additionally, the applicant must be proficient in one or more programming languages (PERL, PYTHON, JAVA, C++) and have a good understanding of database design and implementation. In addition, applicants should have experience using **one of the open source bioinformatics frameworks such as BIOJAVA or BIOPERL**. The applicant should have experience with software collaboration tools such as CVS. The applicant will be expected to oversee the purchase, installation, and management of the necessary computer hardware and software required to provide bioinformatics support to several users. A good understanding of the UNIX operating system and systems administration is also an important qualification.

http://www.foothill.fhda.edu/bio/programs/bioinfo/curric.shtml

CAREER CERTIFICATE REQUIREMENTS (49 units)*

Biotechnology Core Courses (14 units)
BTEC 51A Cell Biology for Biotechnology (3 units)
BTEC 52A Molecular Biology for Biotechnology (3 units)
BTEC 65 DNA Electrophoretic Systems (1 unit)
BTEC 68 Polymerase Chain Reaction (1 unit)
BTEC 71 DNA Sequencing & Bioinformatics (1 unit)
BTEC 76 Introduction to Microarray Data Analysis (2 unit)
BTEC 64 Protein Electrophoretic Systems (1 unit)
BTEC 66 HPLC (2 units)

Computer Science Core Courses (30 units)
CIS 52A Introduction to Data Management Systems (5 units)
CIS 52B2 Introduction to Oracle SQL (5 units)
CIS 68A Introduction to UNIX (5 units)
CIS 68E Introduction to PERL (5 units)
CIS 68H **Introduction to BioPerl** (5 units)
COIN81 Bioinformatics Tools & Databases (5 units)

http://careers.psgs.com/CareerOppLocation2.asp?location=BETHESDA
CF-046: Bioinformatics Specialist

The NIAID Office of Technology and Information Systems (OTIS)/Bioinformatics and Scientific IT Program (BSIP) is seek-ing a bioinformatics specialist. The position includes managing our bioinformatics services and applications, training NIH scientists, creating and modifying simple bioinformatics software, and collaborating with NIH scientists on specific projects. [snip]

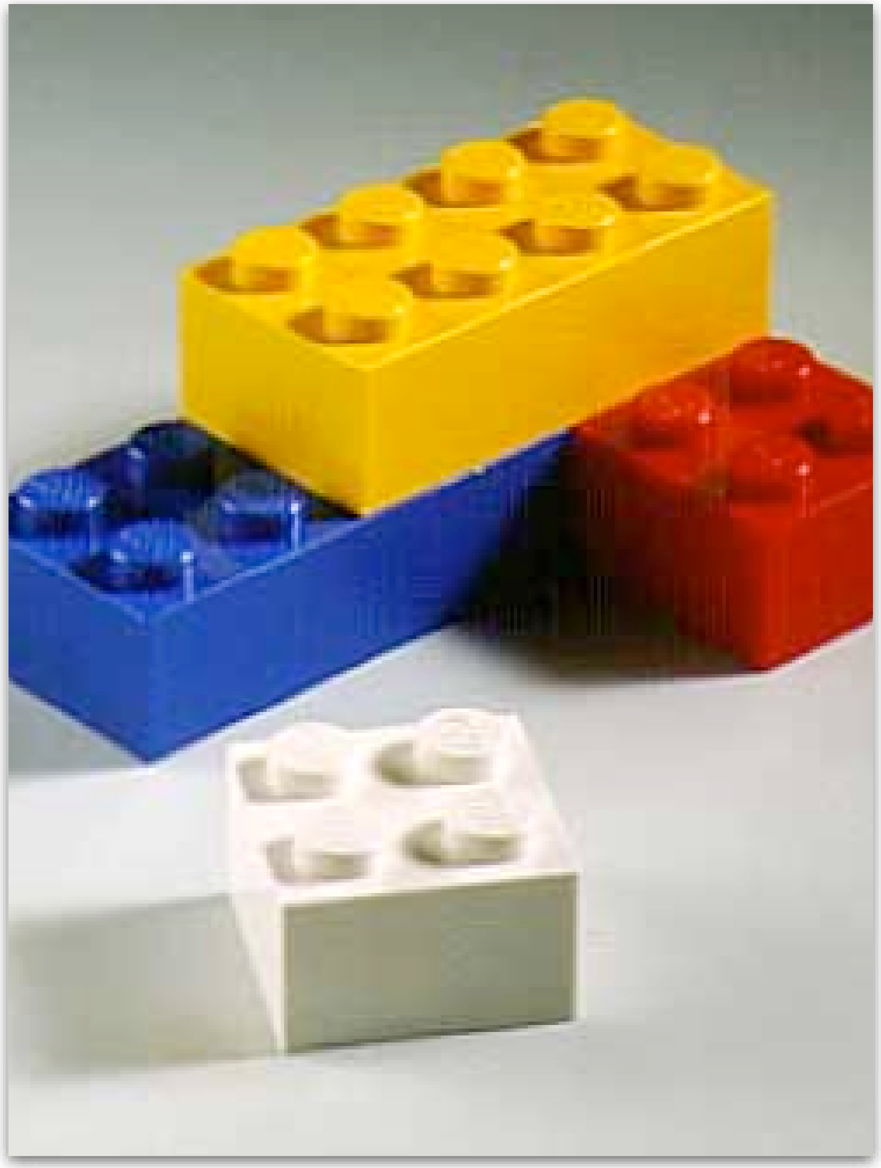
The qualified candidate must hold a Master's Degree (or equivalent) and three years of experience or a Ph. D in life science or computer science. The candidate must have strong interpersonal, written and oral communication skills and be a lateral thinker. Must be able to communicate current bioinformatics technology in a clear and precise manner and to discuss projects with scientists and advise what relevant tools may be used or implemented, have the ability to locate relevant data/information and put this into the context of projects they work on. Candidate must have expert knowledge of UNIX (Mac OS X Darwin a plus), with the ability to install and configure command-line-based applications and services. These include: UNIX windowing systems (X11, FreeX86), UNIX-based open source scientific applications, **Perl and BioPerl scripts**, HTML, XML. Must have experience with one or more of the following: · Web services (WSDL, SOAP) · Genomics and proteomics · Protein structure prediction/visualization. · Sequence analysis, alignment and database searches. · Regular expressions and relational database queries. · Data mining, text mining · Biostatistics

Users and developers

- Toolkit development different from standalone application development
    - Reusable parts in a library
    - Users are often also developers
- New users
    - Need simplified interfaces needed to hide complexity
    - Overwhelmed by number of modules

Development success

- Open source = everyone can help
    - Many contributors
- Modular project
    - Contributors don’t need an expertise in everything
- Designed to have interchangeable parts



http://www.w3.org/2005/Talks/0914-semweb-em/

Open source problems

- Developers have to spend time doing user support
    - Could be done better
    - Documentation comes (too) late
- Open Source = everyone can help
    - Everyone is a leader - Too many cooks spoil broth?
    - No one leads - Don't stick your neck out? or Someone else will do it..

http://www.pixieland.co.uk/product.asp?id=504

Development process

- “Core” developers
- Project vision, release goals, & functionality
- Design and implementation
- Bug fixing
- Releases

Responding to users' needs and requests

- Coordination via mailing list
- She who volunteers to write it
- Usually has to scratch an itch or be sufficiently annoying to a developer to go in and fix it
- Balancing new development and priorities with supporting existing code

Writing yourself into a corner

- Baggage from built up API and need to preserve backwards compatability
- Either change API or have lots of re-direct methods
- Heavy object structure not as efficient as we hoped
- Confusing to novices

http://www.charlise.com/

What can the toolkit do?

- Sequence and Feature Objects & Parsers
  - Graphics
- Alignment reports
  - Pairwise (BLAST) and MSA (Clustalw)
- Weight matricies
- Parsers for many different report formats
- Analysis Tools
- Phylogenetic Trees
- Population genetics summary statistics

Practical use of toolkit

- Automate extraction from sequence files
- BLAST/FASTA/HMMER post-processing
- Small scale pipelines

Sequence file manipulations

```
use Bio::SeqIO;
use strict;
my $in = Bio::SeqIO->new(-format => ‘genbank’,-file=>’in.gbk’);
my $out = Bio::SeqIO->new(-format => ‘fasta’);
while( my $seq = $in->next_seq ) { # read each sequence
 my $ct = 1;
 for my $feat ($seq->get_SeqFeatures ) { # get each feature
   if( $feat->primary_tag eq ‘CDS’ ) { # process each CDS feature
     my $genename = $seq->display_id;
     if( $feat->has_tag(‘gene’) ) {
       ($genename) = $feat->get_tag_values(‘gene’);
     }
     my $cdsseq = $feat->spliced_seq; # get the spliced feature
     my $pepseq = $cdsseq->translate;
     $pepseq->display_id($genename.”.P$ct”);
     $out->write_seq($pepseq);
     $ct++;
   }
 }
}
```

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

Post-processing search results

- Summarize a BLAST/FASTA/HMMER report
- Extract local alignment regions
- Reformat back into HTML or Text
    - Filter out by another criteria

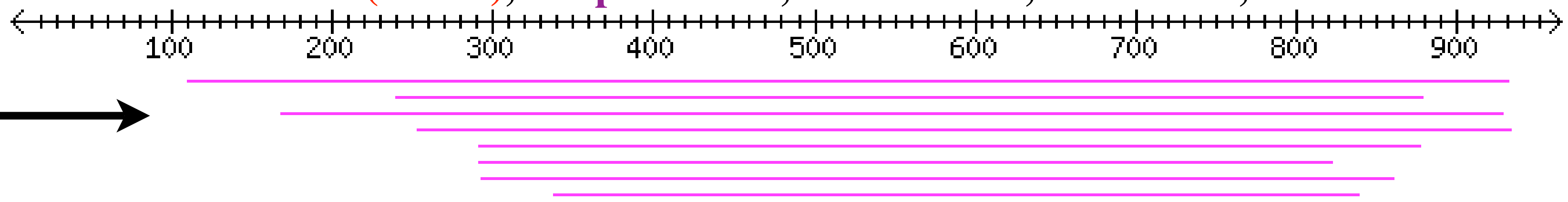
Bioperl Reformatted HTML of BLASTP Search Report for gi|6319512|ref|NP_009594.1|

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

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)

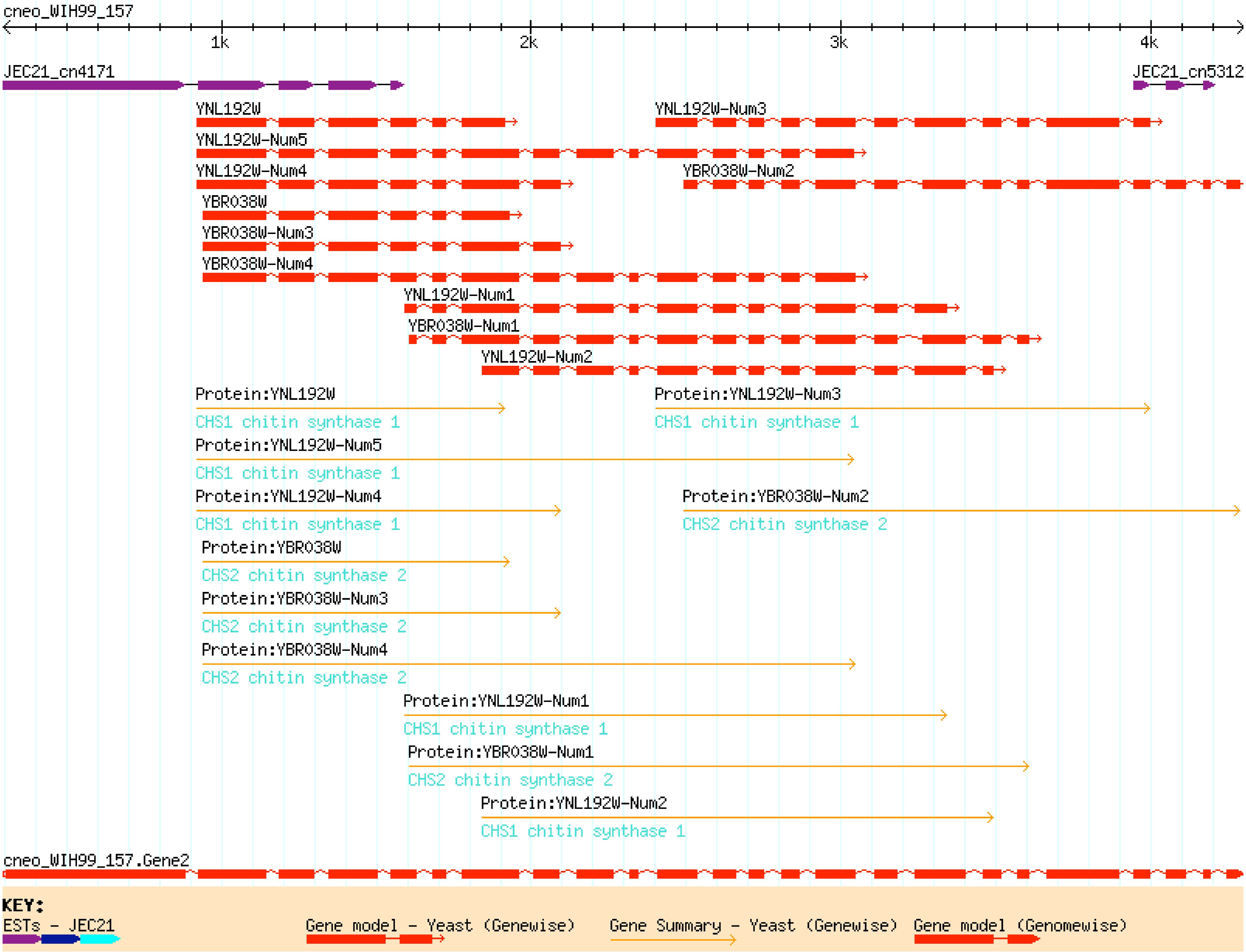
Query= gi|6319512|ref|NP_009594.1| chitin synthase 2; Chs2p [Saccharomyces cerevisiae]
(963 letters)

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


cneo_WIH99_157
1k
2k
3k
4k
JEC21_cn4171
JEC21_cn5312
YNL192W
YNL192W-Num3
YNL192W-Num5
YNL192W-Num4
YBR038W-Num2
YBR038W
YBR038W-Num3
YBR038W-Num4
YNL192W-Num1
YBR038W-Num1
YNL192W-Num2
Protein:YNL192W
CHS1 chitin synthase 1
Protein:YNL192W-Num3
CHS1 chitin synthase 1
Protein:YNL192W-Num5
CHS1 chitin synthase 1
Protein:YNL192W-Num4
CHS1 chitin synthase 1
Protein:YBR038W-Num2
CHS2 chitin synthase 2
Protein:YBR038W
CHS2 chitin synthase 2
Protein:YBR038W-Num3
CHS2 chitin synthase 2
Protein:YBR038W-Num4
CHS2 chitin synthase 2
Protein:YNL192W-Num1
CHS1 chitin synthase 1
Protein:YBR038W-Num1
CHS2 chitin synthase 2
Protein:YNL192W-Num2
CHS1 chitin synthase 1
cneo_WIH99_157.Gene2
KEY:
ESTs - JEC21
Gene model - Yeast (Genewise)
Gene Summary - Yeast (Genewise)
Gene model (Genomewise)

Turning BLAST into HTML

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

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

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

Turning BLAST into HTML

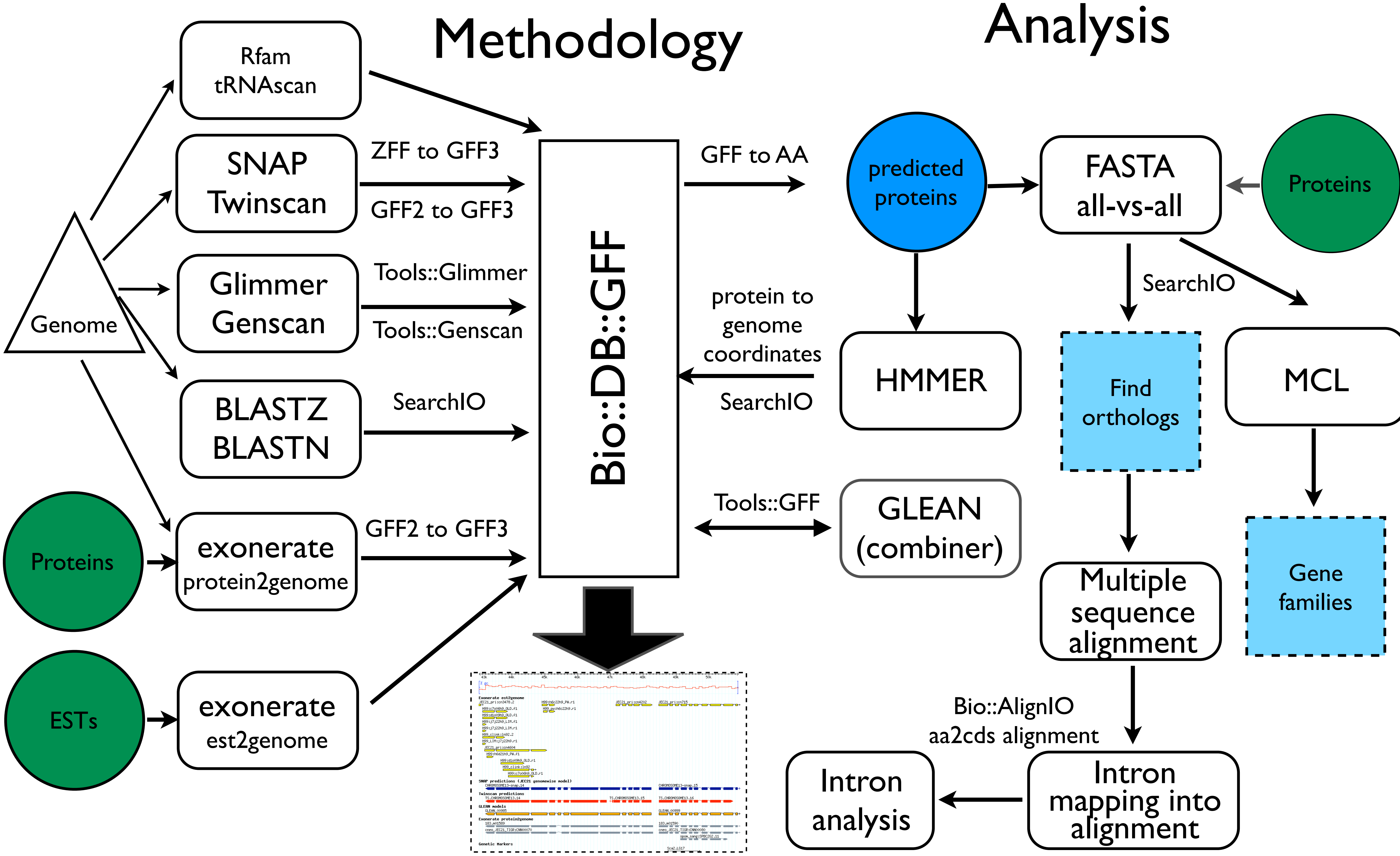
```
# to filter your output
  my $MinLength = 100; # need a variable with scope outside the method
  sub hsp_filter {
      my $hsp = shift;
      return 1 if $hsp->length('total') > $MinLength;
  }
  sub result_filter {
      my $result = shift;
      return $hsp->num_hits > 0;
  }

  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);
```

Pipelines with BioPerl

- Annotation pipeline for fungal genomes
- Glue connecting different pieces
- Highly customized for our needs & compute resources

Methodology

Analysis



Genome

Rfam
tRNAscan

SNAP
Twinscan

ZFF to GFF3

GFF2 to GFF3

Glimmer
Genscan

Tools::Glimmer

Tools::Genscan

BLASTZ
BLASTN

SearchIO

Proteins

exonerate
protein2genome

GFF2 to GFF3

ESTs

exonerate
est2genome

Bio::DB::GFF

GFF to AA

protein to genome coordinates

SearchIO

Tools::GFF

predicted proteins

HMMER

GLEAN (combiner)

FASTA all-vs-all

Proteins

SearchIO

Find orthologs

MCL

Gene families

Multiple sequence alignment

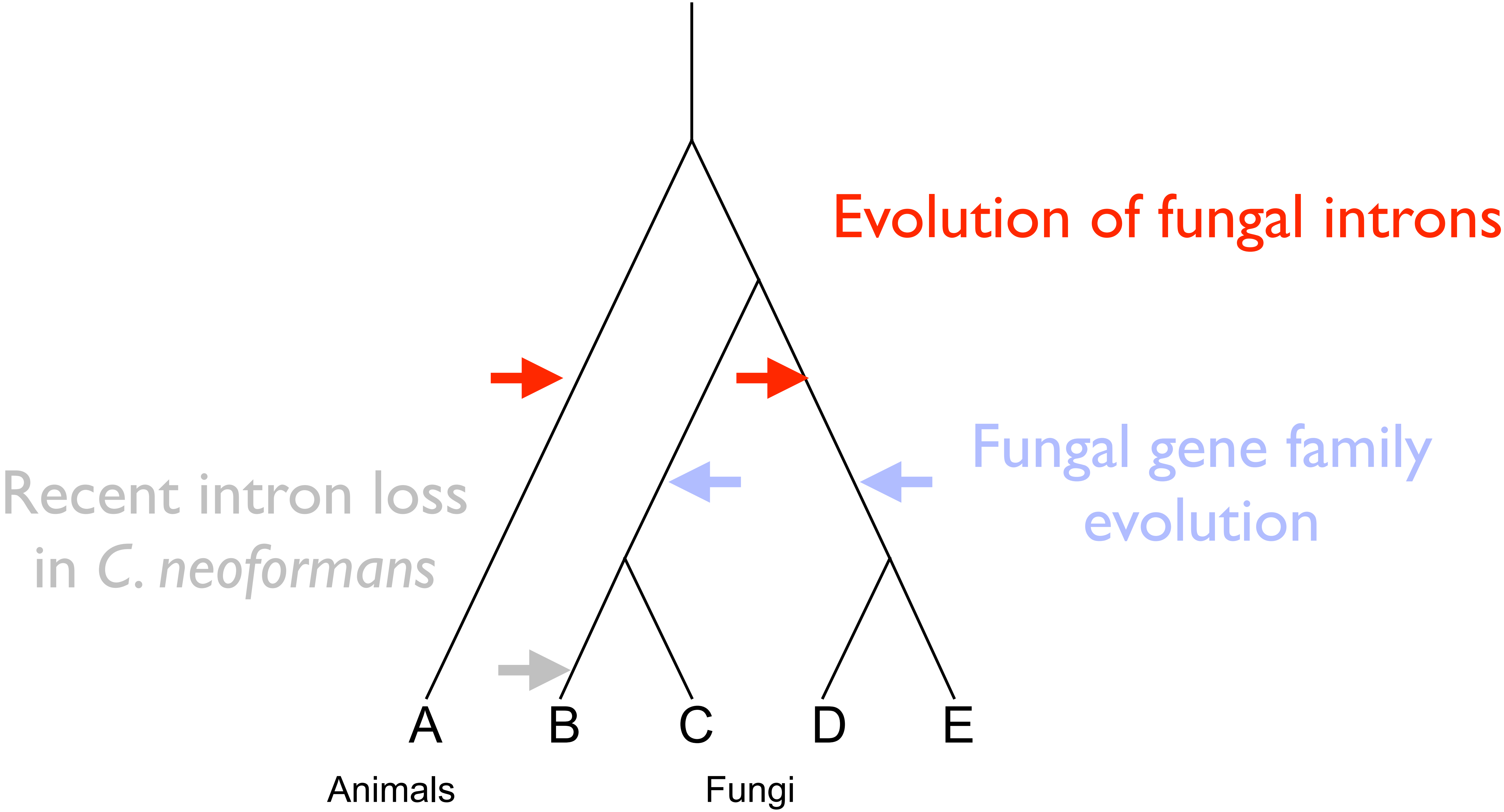
Bio::AlignIO
aa2cds alignment

Intron mapping into alignment

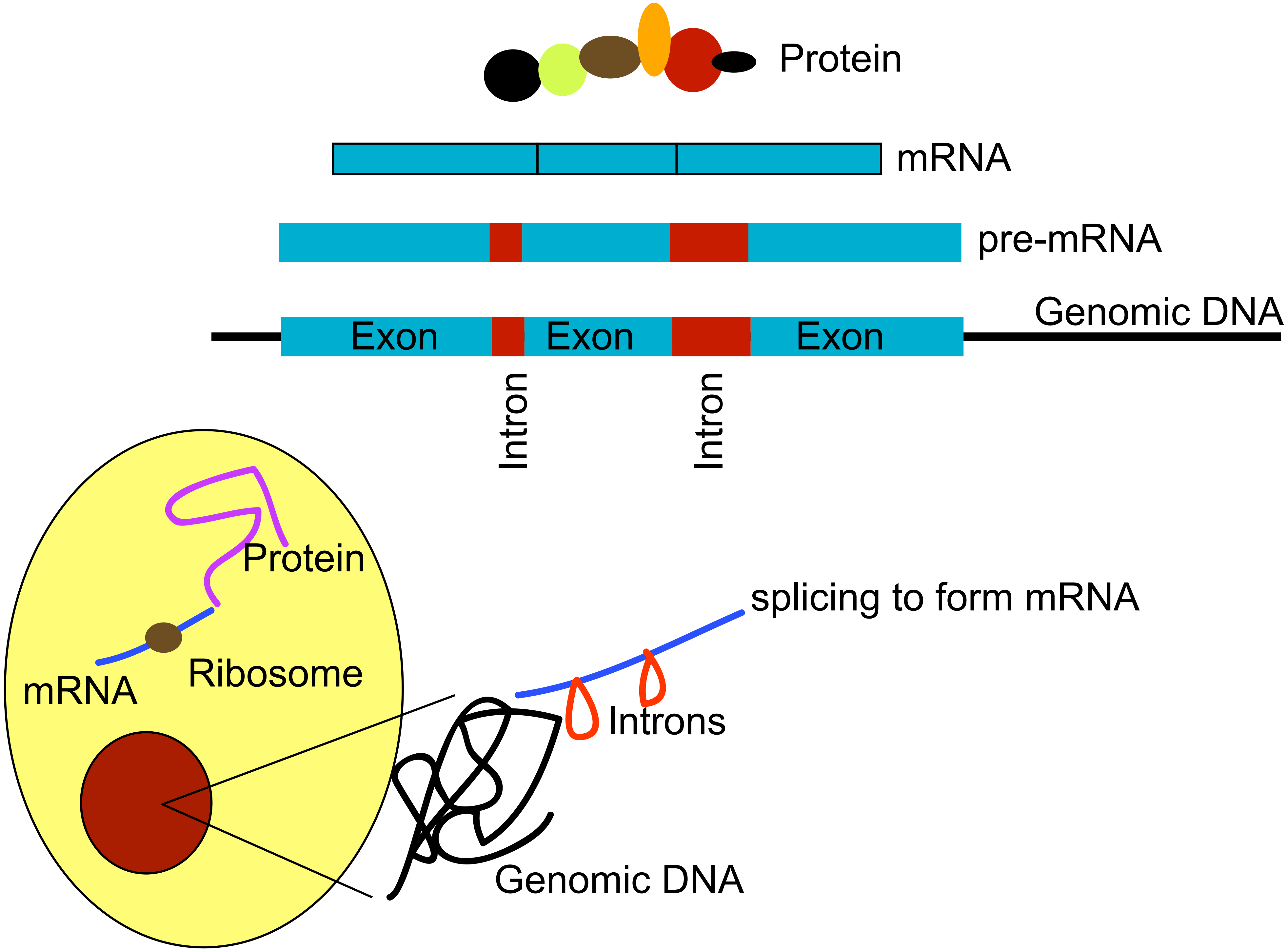
Intron analysis

http://fungal.genome.duke.edu

Fungal comparative genomics



Central dogma of eukaryotic biology

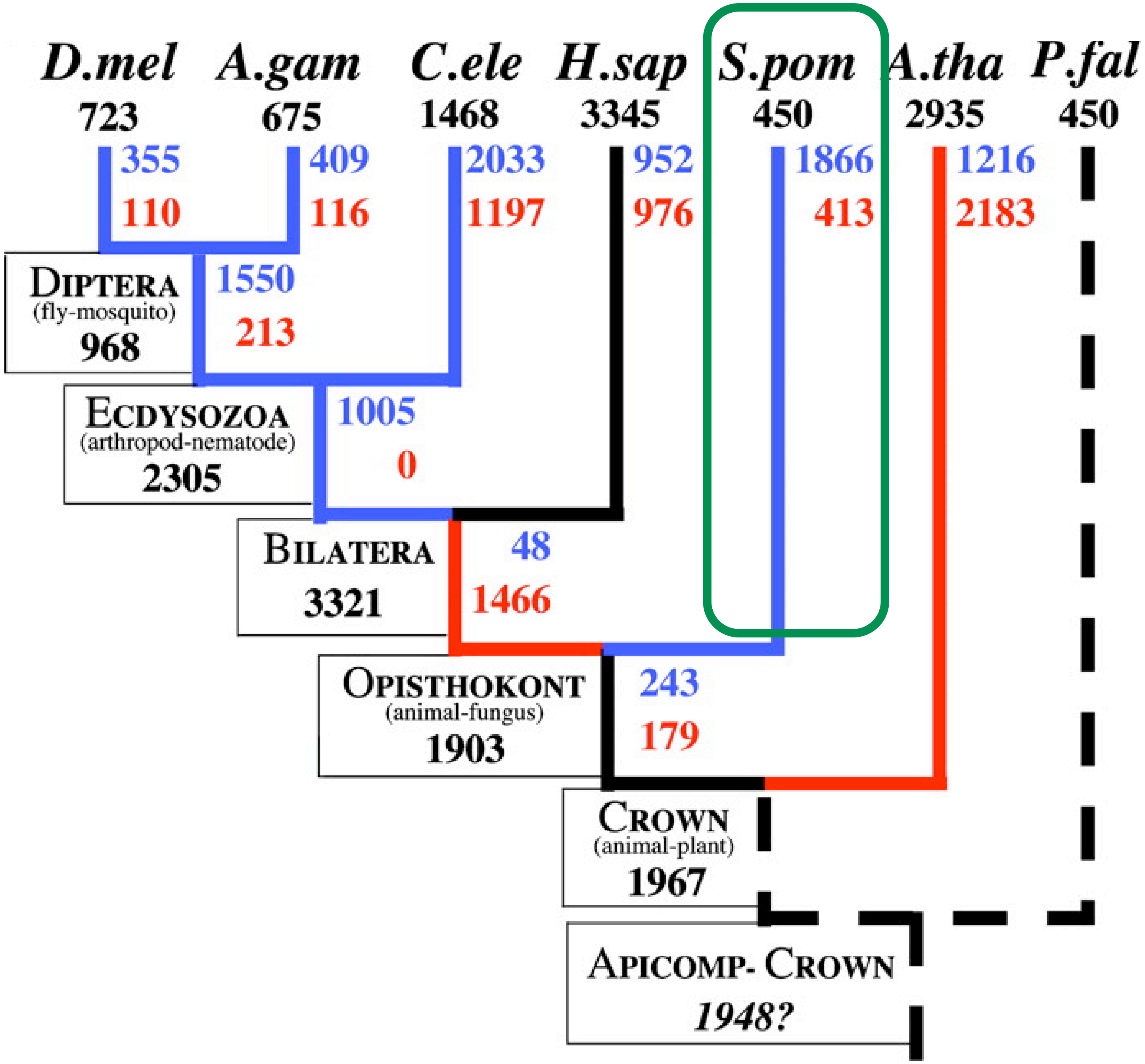


Evolution of gene structure

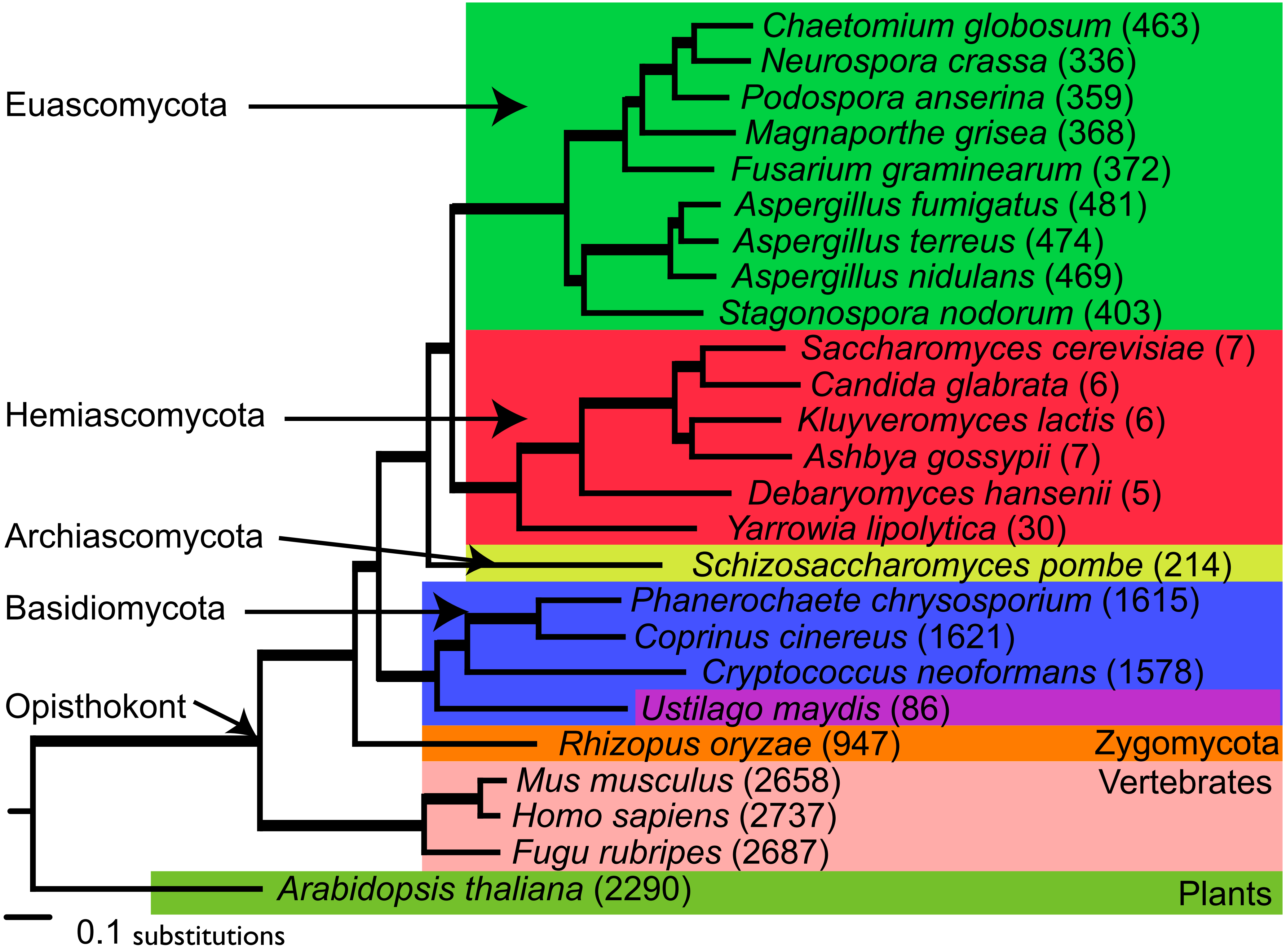
- Present day introns
  - Recent insertions?
    - Introns late hypothesis
  - Present in eukaryotic ancestor?
    - Introns early hypothesis / exon theory of genes
  - Mixture of two?

Previous work on intron evolution

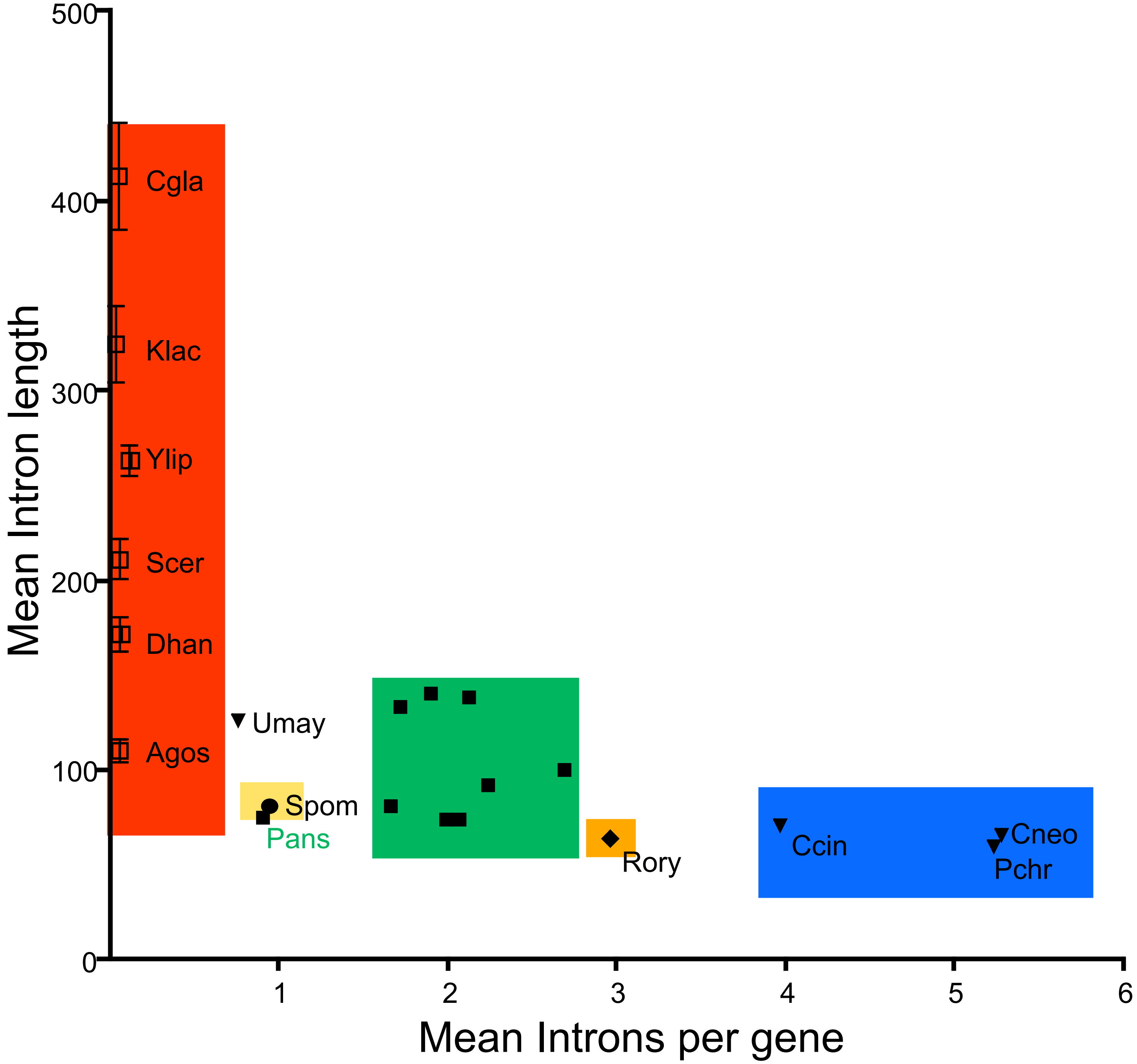
- Rogozin et al. 2003
  - 7 genomes
  - 684 genes, 7236 positions
  - Parsimony analysis
- Analysis methods
  - Roy and Gilbert. 2005
  - Csũrös. 2005
  - Nguyen et al. 2006



Calculating intron densities across a phylogeny


Euascomycota
Hemiascomycota
Archiascomycota
Basidiomycota
Opisthokont
Chaetomium globosum (463)
Neurospora crassa (336)
Podospora anserina (359)
Magnaporthe grisea (368)
Fusarium graminearum (372)
Aspergillus fumigatus (481)
Aspergillus terreus (474)
Aspergillus nidulans (469)
Stagonospora nodorum (403)
Saccharomyces cerevisiae (7)
Candida glabrata (6)
Kluyveromyces lactis (6)
Ashbya gossypii (7)
Debaryomyces hansenii (5)
Yarrowia lipolytica (30)
Schizosaccharomyces pombe (214)
Phanerochaete chrysosporium (1615)
Coprinus cinereus (1621)
Cryptococcus neoformans (1578)
Ustilago maydis (86)
Rhizopus oryzae (947)
Zygomycota
Mus musculus (2658)
Vertebrates
Homo sapiens (2737)
Fugu rubripes (2687)
Arabidopsis thaliana (2290)
Plants
0.1 substitutions

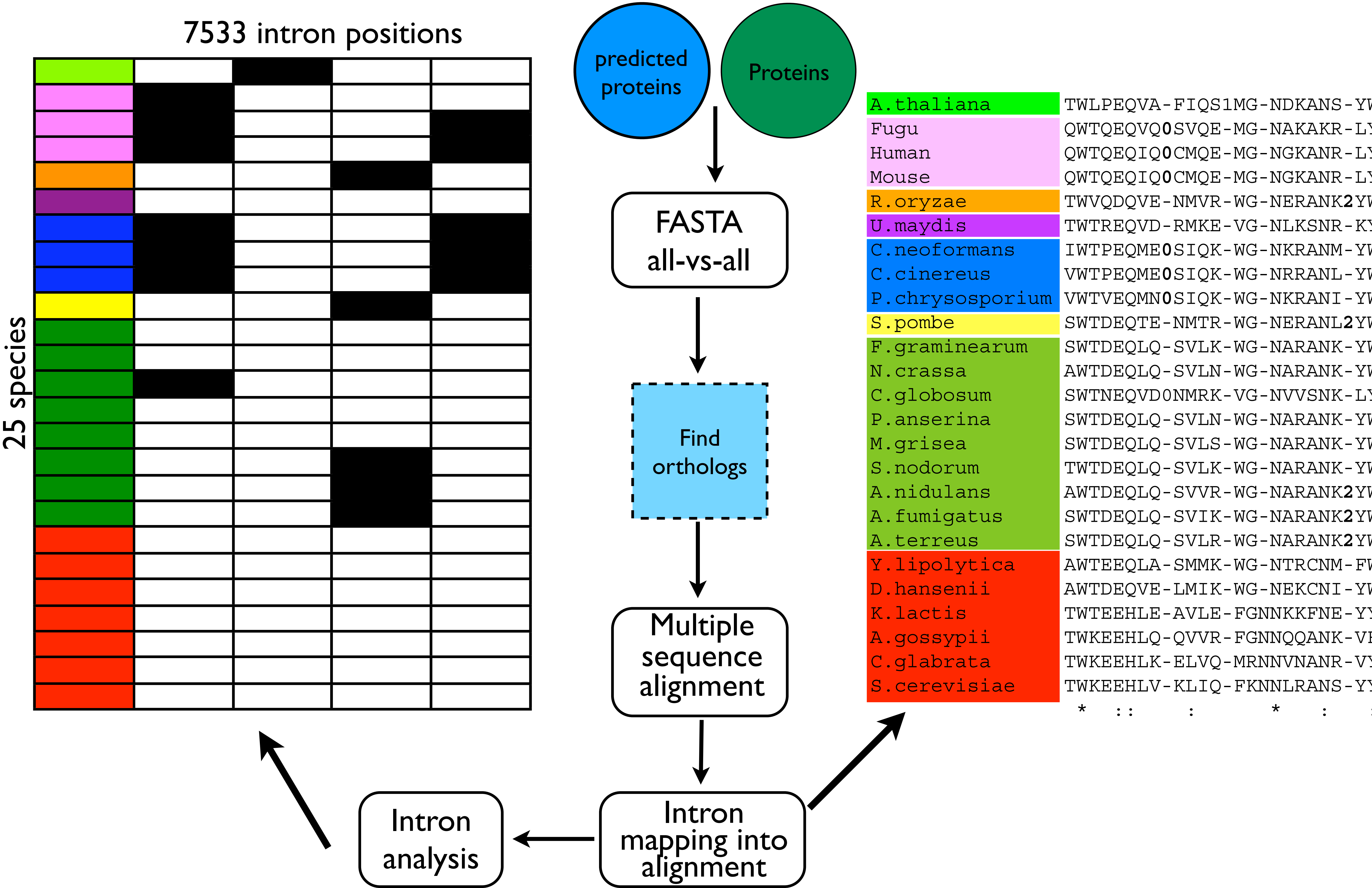
Intron frequency varies among the fungi



Analysis of whole genomes

- 25 entire genomes
  - 21 fungi, 3 vertebrates, 1 plant
- Largest dataset ever assembled for intron analysis
- 1160 orthologous genes
- 7533 intron positions
- 4.15 Mb conserved coding sequence per genome

Analysis methods



7533 intron positions

25 species

predicted proteins

Proteins

FASTA all-vs-all

Find orthologs

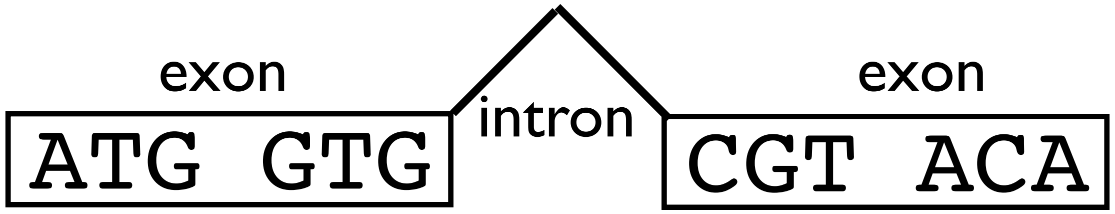
Multiple sequence alignment

Intron mapping into alignment

Intron analysis

```
A.thaliana         TWLPEQVA-FIQS1MG-NDKANS-YW
Fugu               QWTQEQVQ0SVQE-MG-NAKAKR-LY
Human              QWTQEQIQ0CMQE-MG-NGKANR-LY
Mouse              QWTQEQIQ0CMQE-MG-NGKANR-LY
R.oryzae           TWVQDQVE-NMVR-WG-NERANK2YW
U.maydis           TWTREQVD-RMKE-VG-NLKSNR-KY
C.neoformans       IWTPEQME0SIQK-WG-NKRANM-YW
C.cinereus         VWTPEQME0SIQK-WG-NRRANL-YW
P.chrysosporium    VWTVEQMN0SIQK-WG-NKRANI-YW
S.pombe            SWTDEQTE-NMTR-WG-NERANL2YW
F.graminearum      SWTDEQLQ-SVLK-WG-NARANK-YW
N.crassa           AWTDEQLQ-SVLN-WG-NARANK-YW
C.globosum         SWTNEQVD0NMRK-VG-NVVSNK-LY
P.anserina         SWTDEQLQ-SVLN-WG-NARANK-YW
M.grisea           SWTDEQLQ-SVLS-WG-NARANK-YW
S.nodorum          TWTDEQLQ-SVLK-WG-NARANK-YW
A.nidulans         AWTDEQLQ-SVVR-WG-NARANK2YW
A.fumigatus        SWTDEQLQ-SVIK-WG-NARANK2YW
A.terreus          SWTDEQLQ-SVLR-WG-NARANK2YW
Y.lipolytica       AWTEEQLA-SMMK-WG-NTRCNM-FW
D.hansenii         AWTDEQVE-LMIK-WG-NEKCNI-YW
K.lactis           TWTEEHLE-AVLE-FGNNKKFNE-YY
A.gossypii         TWKEEHLQ-QVVR-FGNNQQANK-VF
C.glabrata         TWKEEHLK-ELVQ-MRNNVNANR-VY
S.cerevisiae       TWKEEHLV-KLIQ-FKNNLRANS-YY
                    *  ::    :      *   :   :
```

Intron phase



exon

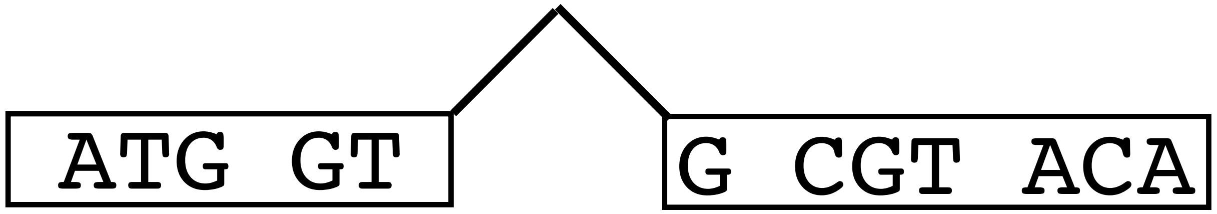
intron

exon

ATG GTG

CGT ACA

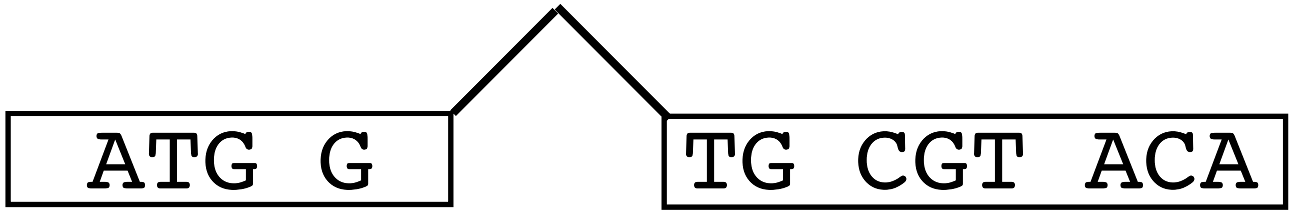
phase 0



ATG GT

G CGT ACA

phase 1

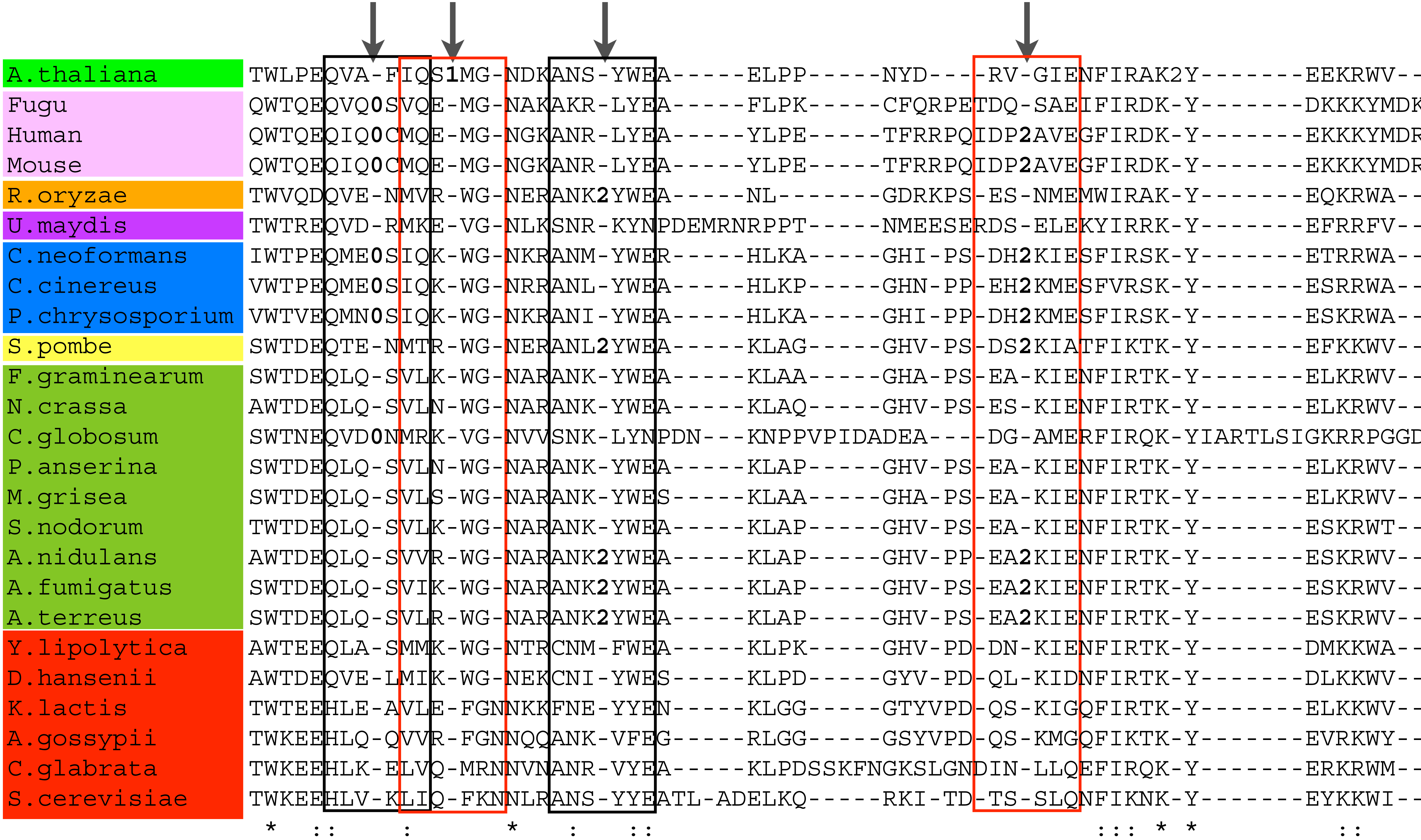


ATG G

TG CGT ACA

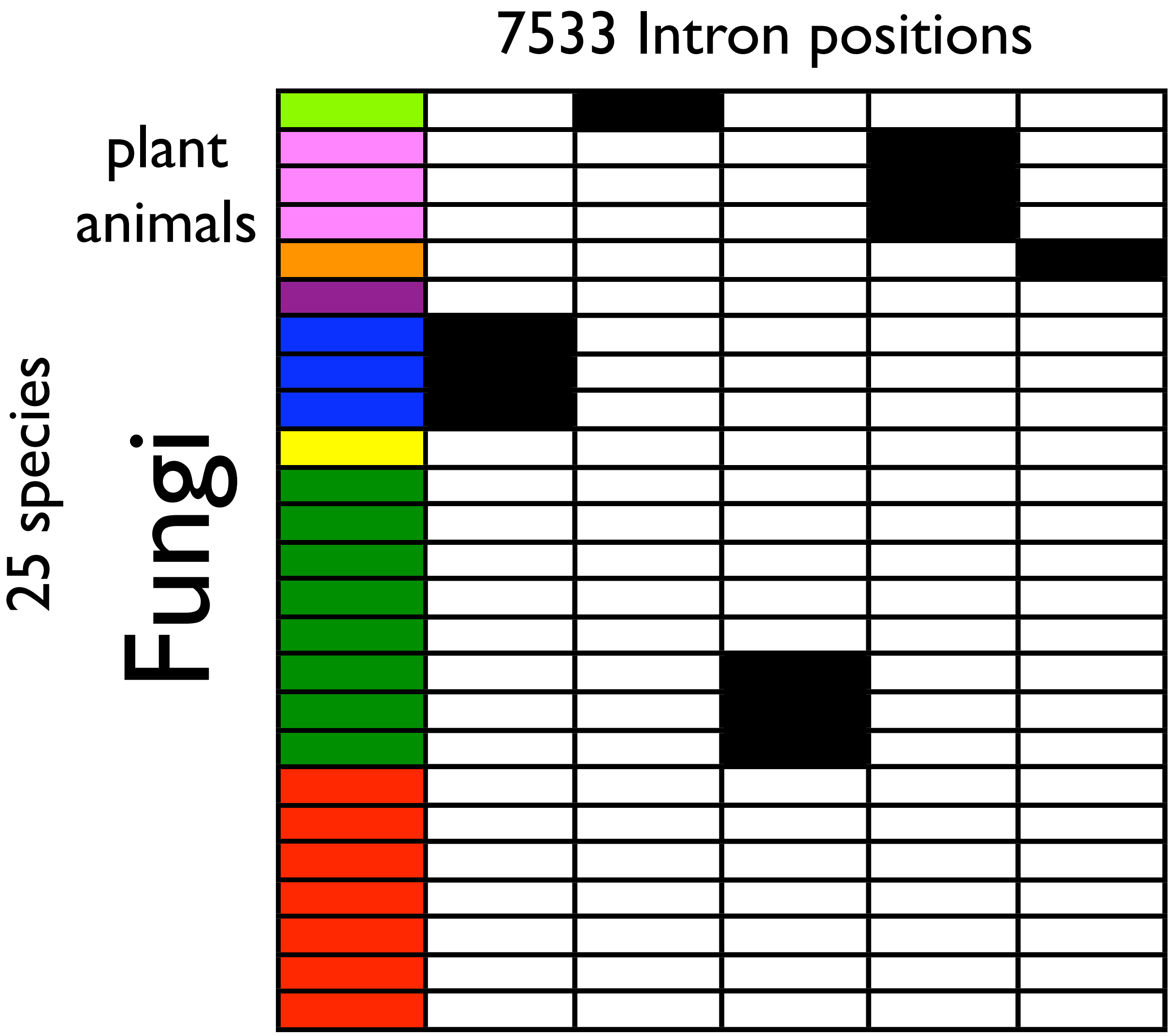
phase 2

Conserved intron positions

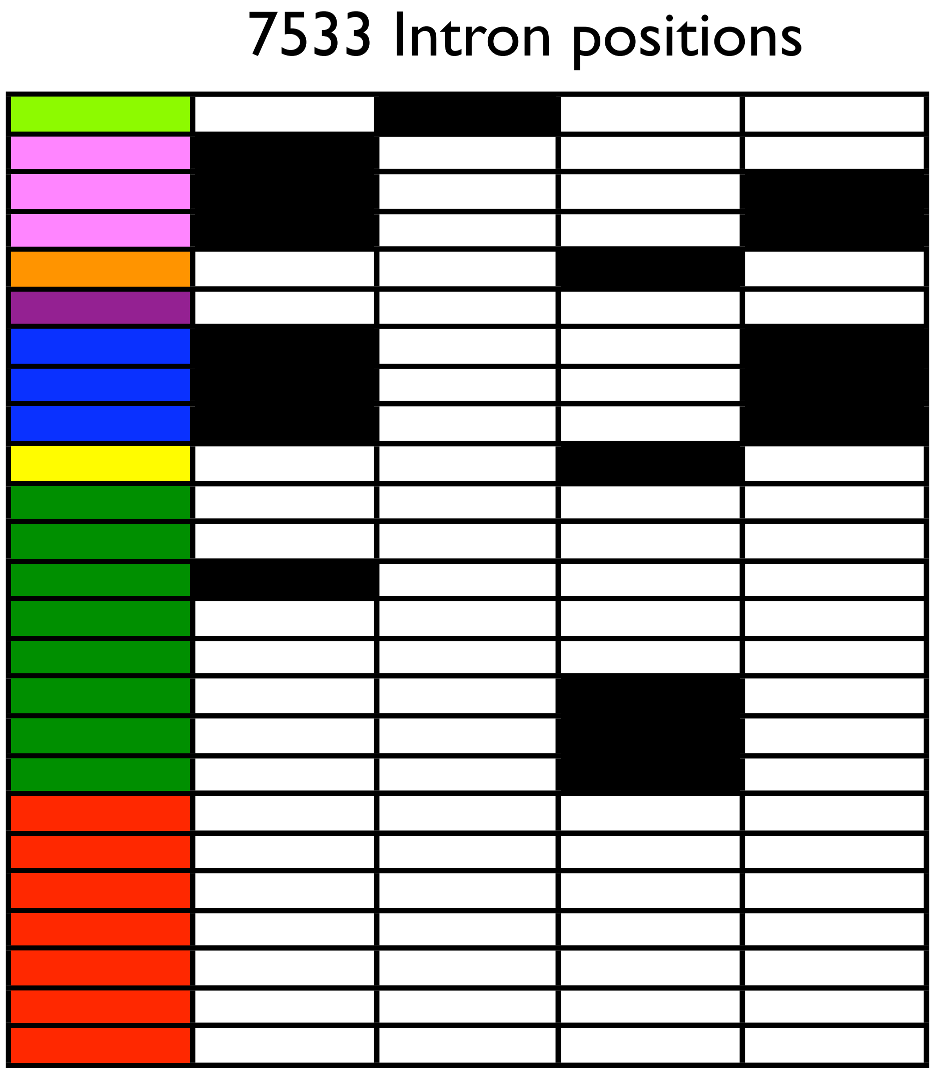


```
A.thaliana         TWLPEQVA-FIQS1MG-NDKANS-YWEA-----ELPP-----NYD----RV-GIENFIRAK2Y-------EEKRWV--
Fugu               QWTQEQVQ0SVQE-MG-NAKAKR-LYEA-----FLPK-----CFQRPETDQ-SAEIFIRDK-Y-------DKKKYMDK
Human              QWTQEQIQ0CMQE-MG-NGKANR-LYEA-----YLPE-----TFRRPQIDP2AVEGFIRDK-Y-------EKKKYMDR
Mouse              QWTQEQIQ0CMQE-MG-NGKANR-LYEA-----YLPE-----TFRRPQIDP2AVEGFIRDK-Y-------EKKKYMDR
R.oryzae           TWVQDQVE-NMVR-WG-NERANK2YWEA-----NL-------GDRKPS-ES-NMEMWIRAK-Y-------EQKRWA--
U.maydis           TWTREQVD-RMKE-VG-NLKSNR-KYNPDEMRNRPPT-----NMEESERDS-ELEKYIRRK-Y-------EFRRFV--
C.neoformans       IWTPEQME0SIQK-WG-NKRANM-YWER-----HLKA-----GHI-PS-DH2KIESFIRSK-Y-------ETRRWA--
C.cinereus         VWTPEQME0SIQK-WG-NRRANL-YWEA-----HLKP-----GHN-PP-EH2KMESFVRSK-Y-------ESRRWA--
P.chrysosporium    VWTVEQMN0SIQK-WG-NKRANI-YWEA-----HLKA-----GHI-PP-DH2KMESFIRSK-Y-------ESKRWA--
S.pombe            SWTDEQTE-NMTR-WG-NERANL2YWEA-----KLAG-----GHV-PS-DS2KIATFIKTK-Y-------EFKKWV--
F.graminearum      SWTDEQLQ-SVLK-WG-NARANK-YWEA-----KLAA-----GHA-PS-EA-KIENFIRTK-Y-------ELKRWV--
N.crassa           AWTDEQLQ-SVLN-WG-NARANK-YWEA-----KLAQ-----GHV-PS-ES-KIENFIRTK-Y-------ELKRWV--
C.globosum         SWTNEQVD0NMRK-VG-NVVSNK-LYNPDN---KNPPVPIDADEA----DG-AMERFIRQK-YIARTLSIGKRRPGGD
P.anserina         SWTDEQLQ-SVLN-WG-NARANK-YWEA-----KLAP-----GHV-PS-EA-KIENFIRTK-Y-------ELKRWV--
M.grisea           SWTDEQLQ-SVLS-WG-NARANK-YWES-----KLAA-----GHA-PS-EA-KIENFIRTK-Y-------ELKRWV--
S.nodorum          TWTDEQLQ-SVLK-WG-NARANK-YWEA-----KLAP-----GHV-PS-EA-KIENFIRTK-Y-------ESKRWT--
A.nidulans         AWTDEQLQ-SVVR-WG-NARANK2YWEA-----KLAP-----GHV-PP-EA2KIENFIRTK-Y-------ESKRWV--
A.fumigatus        SWTDEQLQ-SVIK-WG-NARANK2YWEA-----KLAP-----GHV-PS-EA2KIENFIRTK-Y-------ESKRWV--
A.terreus          SWTDEQLQ-SVLR-WG-NARANK2YWEA-----KLAP-----GHV-PS-EA2KIENFIRTK-Y-------ESKRWV--
Y.lipolytica       AWTEEQLA-SMMK-WG-NTRCNM-FWEA-----KLPK-----GHV-PD-DN-KIENFIRTK-Y-------DMKKWA--
D.hansenii         AWTDEQVE-LMIK-WG-NEKCNI-YWES-----KLPD-----GYV-PD-QL-KIDNFIRTK-Y-------DLKKWV--
K.lactis           TWTEEHLE-AVLE-FGNNKKFNE-YYEN-----KLGG-----GTYVPD-QS-KIGQFIRTK-Y-------ELKKWV--
A.gossypii         TWKEEHLQ-QVVR-FGNNQQANK-VFEG-----RLGG-----GSYVPD-QS-KMGQFIKTK-Y-------EVRKWY--
C.glabrata         TWKEEHLK-ELVQ-MRNNVNANR-VYEA-----KLPDSSKFNGKSLGNDIN-LLQEFIRQK-Y-------ERKRWM--
S.cerevisiae       TWKEEHLV-KLIQ-FKNNLRANS-YYEATL-ADELKQ-----RKI-TD-TS-SLQNFIKNK-Y-------EYKKWI--
                    *  ::    :      *   :   ::                             ::: * *         ::
```

Patterns of conservation

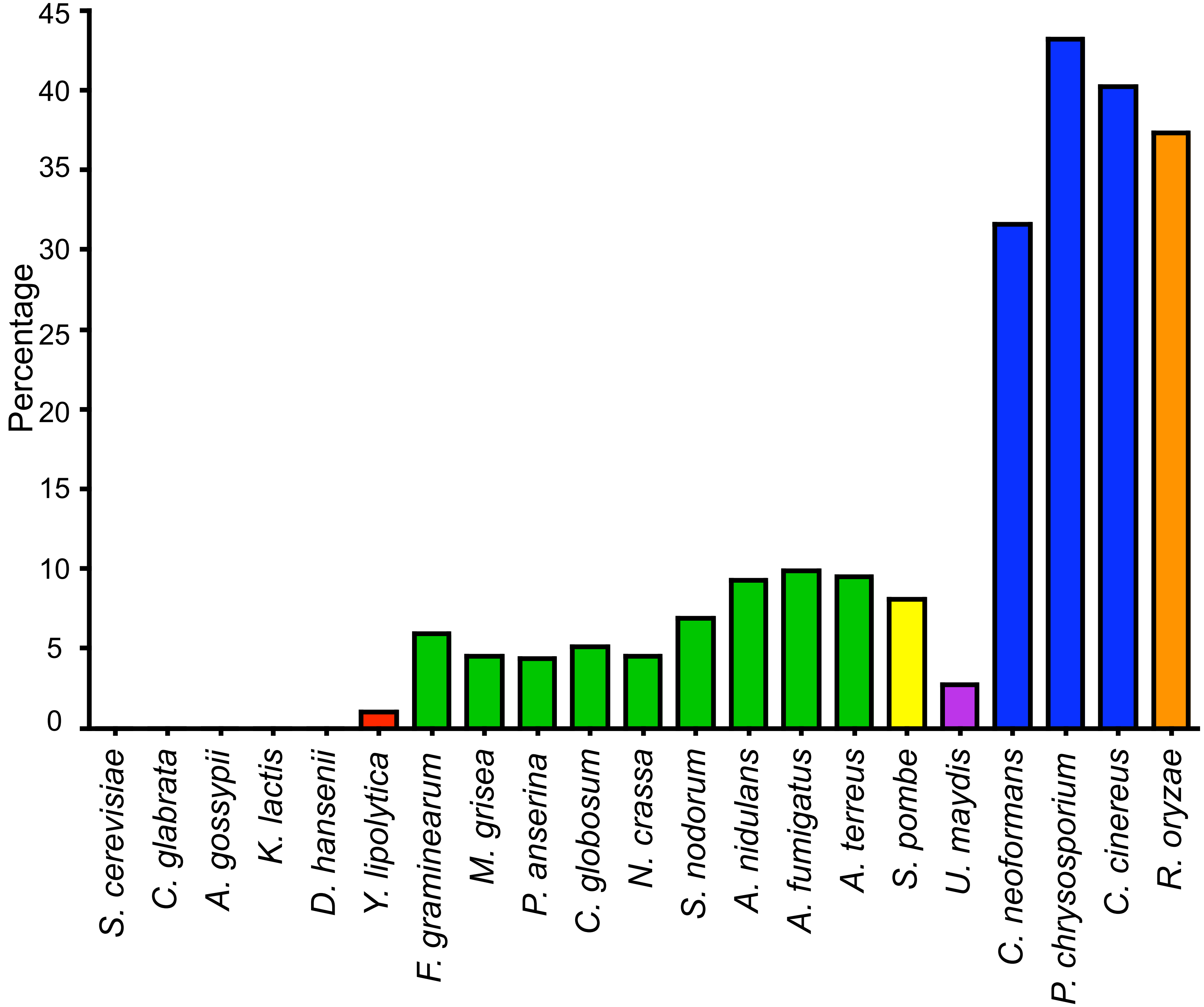


Introns late



Introns early(ier)

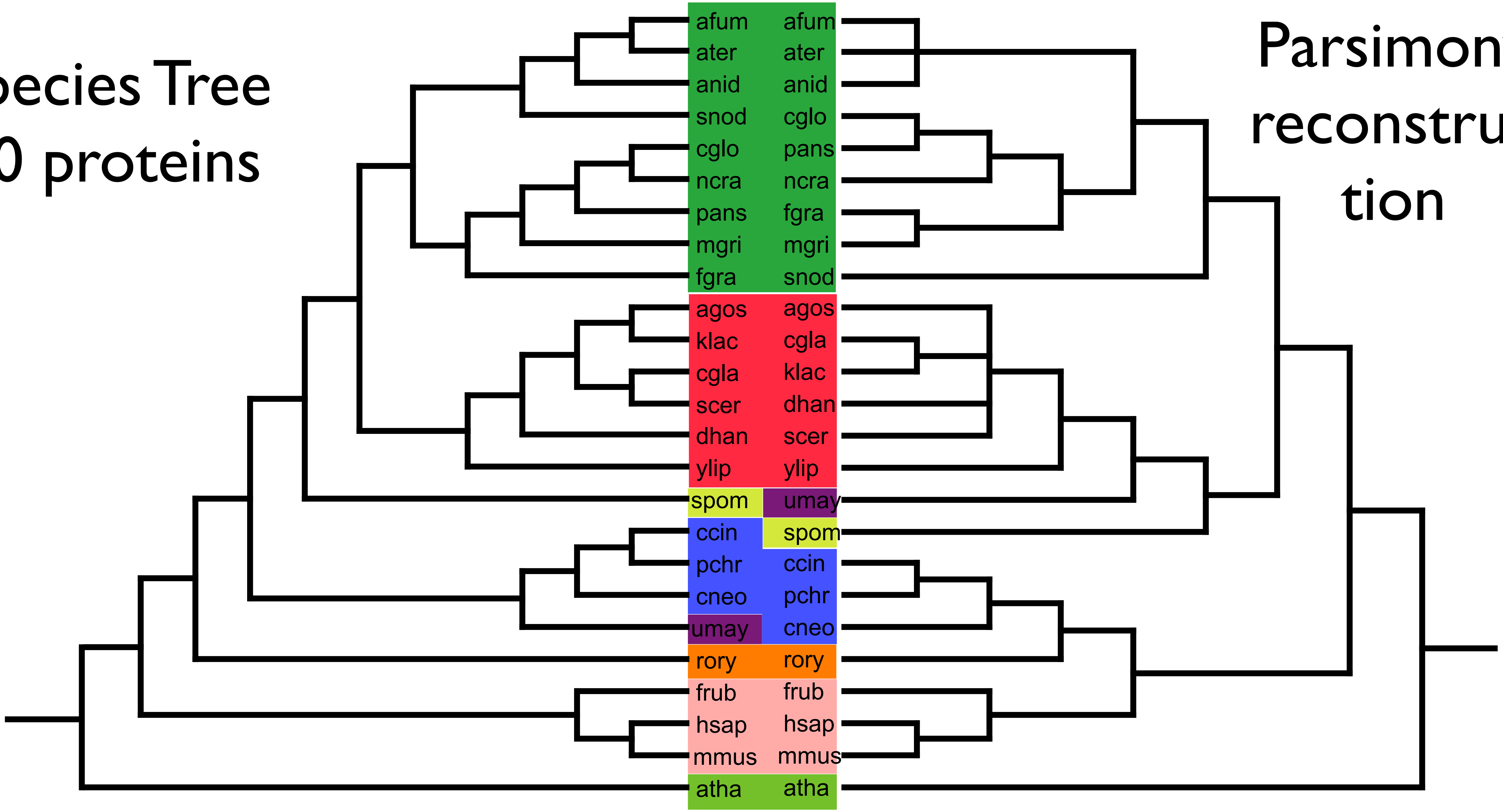
Intron positions shared with animals or plants



Phylogenetic signal in intron positions

Species Tree
30 proteins

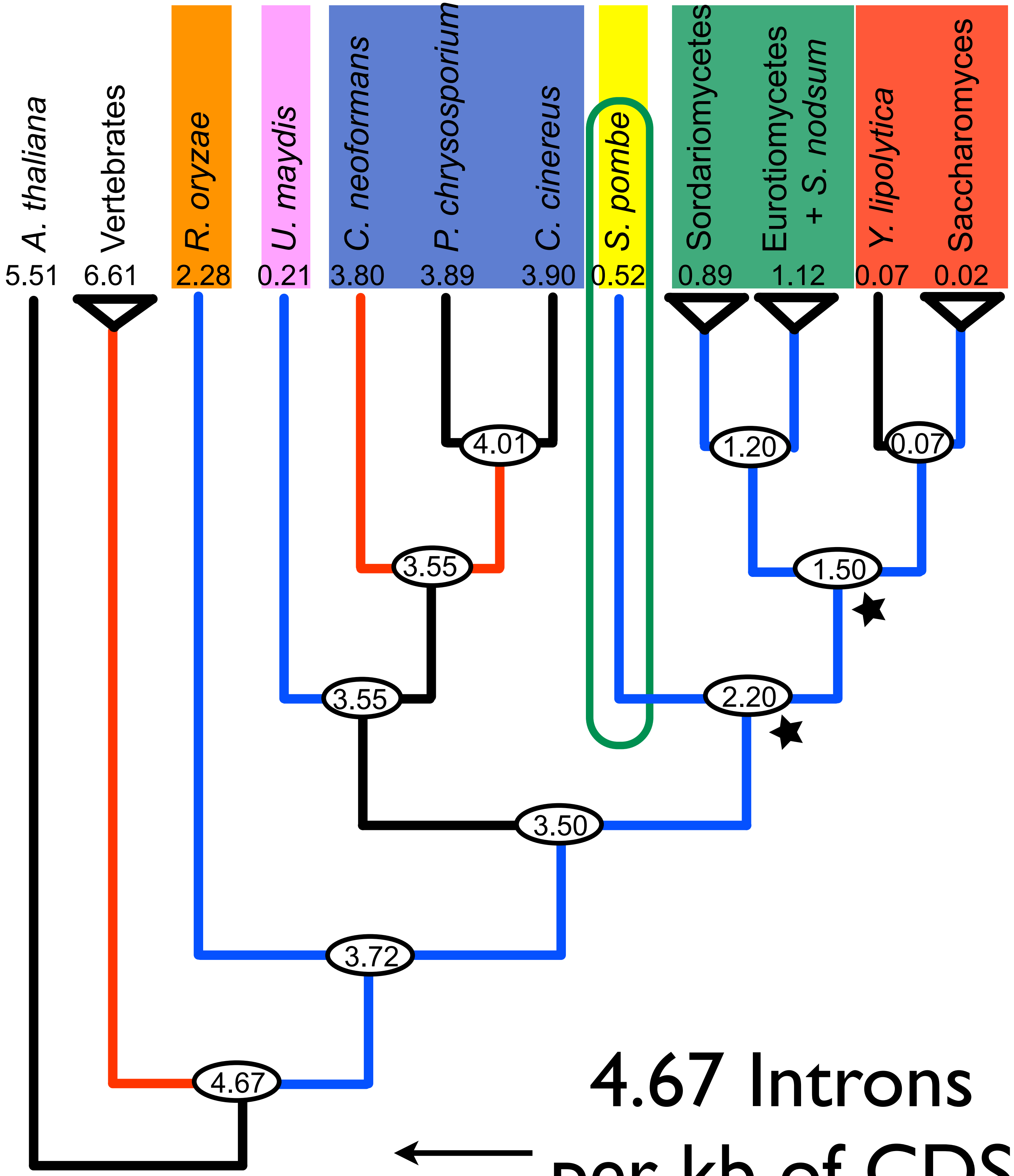
Parsimony
reconstruc
tion



Intron position reconstruction

- 3 Methods
  - Roy and Gilbert. 2005
  - Csũrös. 2005
  - Nguyen et al. 2006
- Methods agree for all but 2 nodes in tree

Reconstruction of ancestral intron densities



← 4.67 Introns per kb of CDS

Exon length of 214 bp

Conclusions

- Early eukaryotic crown genes were complex!
  - Ancestor had 70% of the introns in vertebrates - many more than previously reported
- Intron loss has dominated among the fungi
  - Hemiascomycota experienced loss
- No significant evidence for intron sliding or double insertions
- Sampling can bias interpretations - all fungi are not equal.

How can I...

- Contribute a new module?
  - Discuss on list - have a developer commit it for you or get a CVS account
- A patch for a new feature or bug fix?
  - Bugzilla - http://bugzilla.open-bio.org
- Contribute documentation?
  - Wiki pages are open to all

Future directions

- Better documentation
- Improve speed
- Perl6?

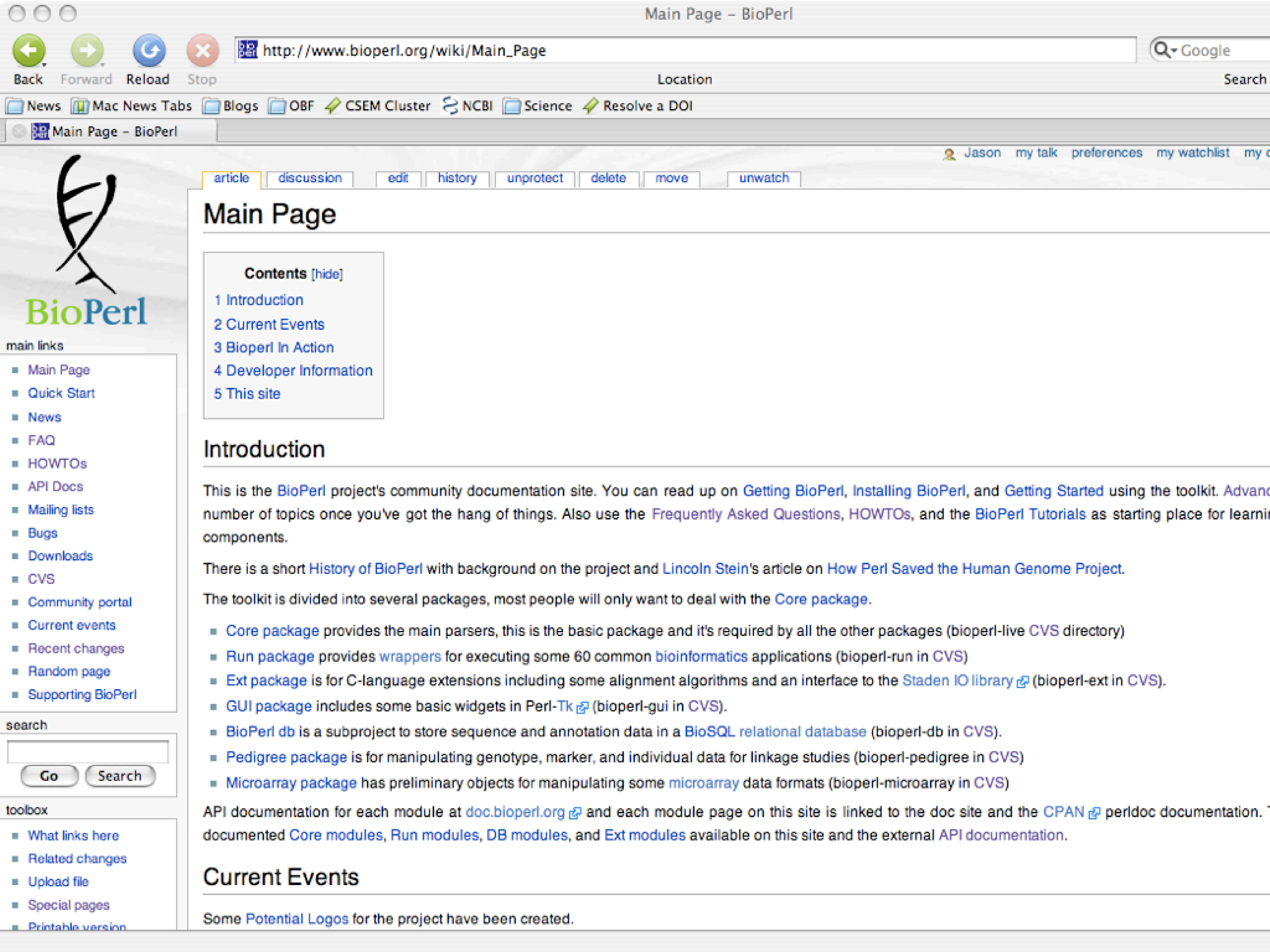
Improving speed

- Objects are heavy to create
  - Use lightweight arrays or hashes on demand instead of objects
- Simplify inheritance

http://www.wickedjester.com/heavy.gif

Better Documentation

- Wiki site - www.bioperl.org
  - Few hundred web pages of prose
  - Community maintainable
- HOWTOs -narrative descriptions
- Pdoc - doc.bioperl.org
  - Autogenerated API docs from POD
- “DeObfuscator”



Main Page

Contents [hide]

1 Introduction
2 Current Events
3 Bioperl In Action
4 Developer Information
5 This site

Introduction

This is the BioPerl project's community documentation site. You can read up on Getting BioPerl, Installing BioPerl, and Getting Started using the toolkit. Advanc number of topics once you've got the hang of things. Also use the Frequently Asked Questions, HOWTOs, and the BioPerl Tutorials as starting place for learnin components.

There is a short History of BioPerl with background on the project and Lincoln Stein's article on How Perl Saved the Human Genome Project.

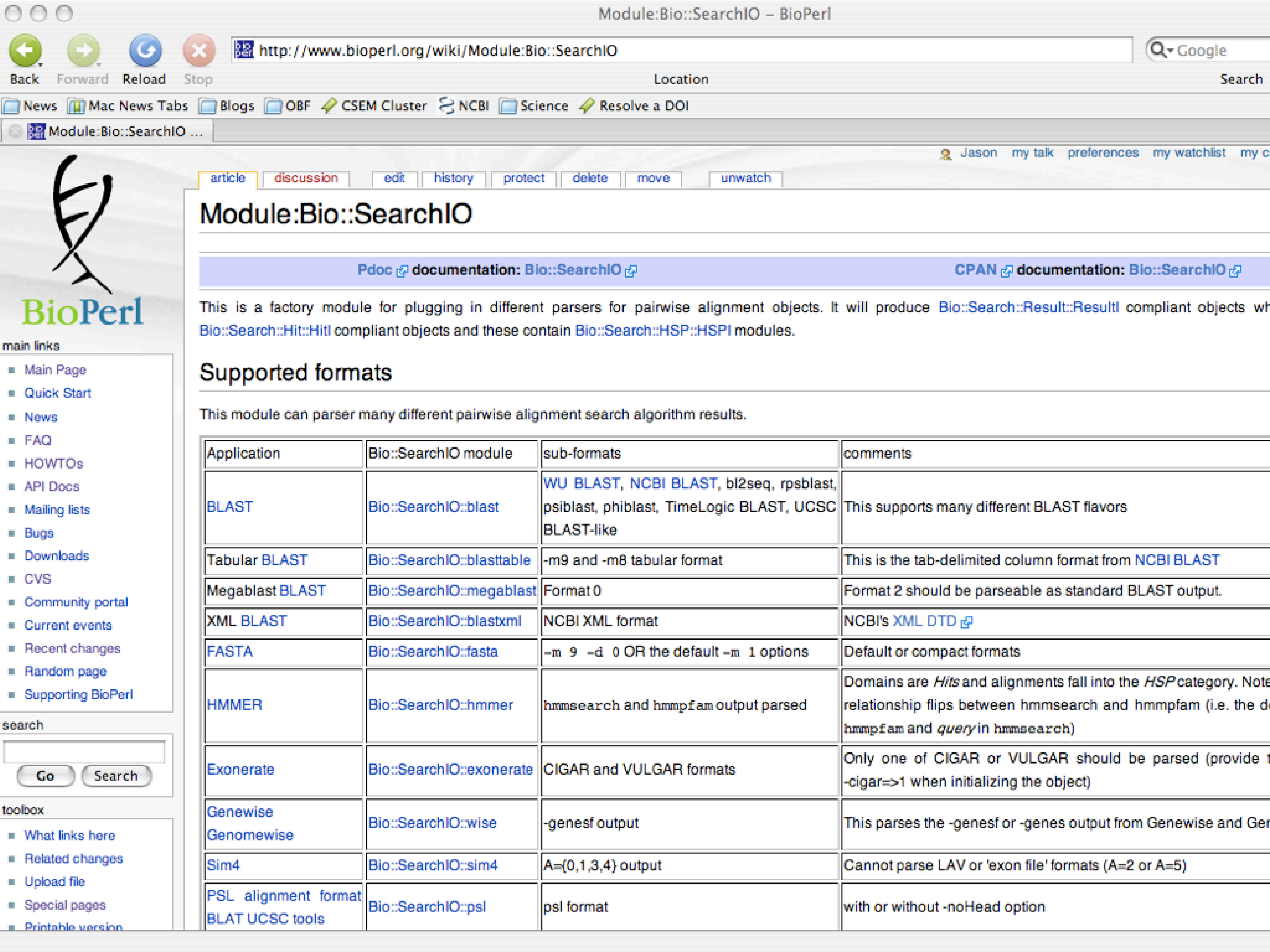
The toolkit is divided into several packages, most people will only want to deal with the Core package.

- Core package provides the main parsers, this is the basic package and it's required by all the other packages (bioperl-live CVS directory)
- Run package provides wrappers for executing some 60 common bioinformatics applications (bioperl-run in CVS)
- Ext package is for C-language extensions including some alignment algorithms and an interface to the Staden IO library (bioperl-ext in CVS).
- GUI package includes some basic widgets in Perl-Tk (bioperl-gui in CVS).
- BioPerl db is a subproject to store sequence and annotation data in a BioSQL relational database (bioperl-db in CVS).
- Pedigree package is for manipulating genotype, marker, and individual data for linkage studies (bioperl-pedigree in CVS)
- Microarray package has preliminary objects for manipulating some microarray data formats (bioperl-microarray in CVS)

API documentation for each module at doc.bioperl.org and each module page on this site is linked to the doc site and the CPAN perldoc documentation. T documented Core modules, Run modules, DB modules, and Ext modules available on this site and the external API documentation.

Current Events

Some Potential Logos for the project have been created.



Module:Bio::SearchIO

Pdoc documentation: Bio::SearchIO **CPAN documentation: Bio::SearchIO**

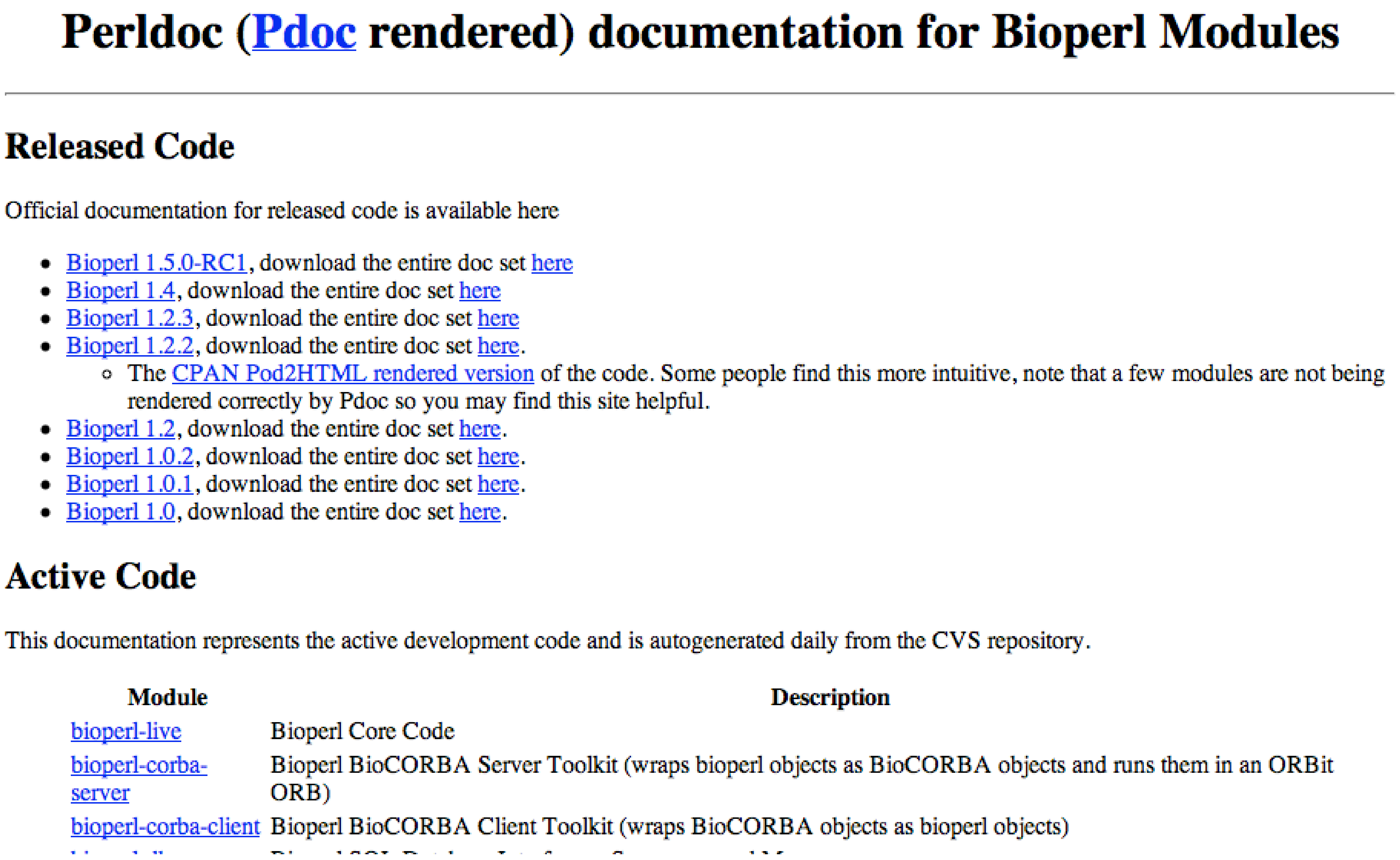
This is a factory module for plugging in different parsers for pairwise alignment objects. It will produce Bio::Search::Result::ResultI compliant objects wh Bio::Search::Hit::HitI compliant objects and these contain Bio::Search::HSP::HSPI modules.

Supported formats

This module can parser many different pairwise alignment search algorithm results.

Application	Bio::SearchIO module	sub-formats	comments
BLAST	Bio::SearchIO::blast	WU BLAST, NCBI BLAST, bl2seq, rpsblast, psiblast, phiblast, TimeLogic BLAST, UCSC BLAST-like	This supports many different BLAST flavors
Tabular BLAST	Bio::SearchIO::blasttable	-m9 and -m8 tabular format	This is the tab-delimited column format from NCBI BLAST
Megablast BLAST	Bio::SearchIO::megablast	Format 0	Format 2 should be parseable as standard BLAST output.
XML BLAST	Bio::SearchIO::blastxml	NCBI XML format	NCBI's XML DTD
FASTA	Bio::SearchIO::fasta	`-m 9 -d 0` OR the default `-m 1` options	Default or compact formats
HMMER	Bio::SearchIO::hmmer	`hmmsearch` and `hmmpfam` output parsed	Domains are *Hits* and alignments fall into the *HSP* category. Note relationship flips between hmmsearch and hmmpfam (i.e. the d `hmmpfam` and *query* in `hmmsearch`)
Exonerate	Bio::SearchIO::exonerate	CIGAR and VULGAR formats	Only one of CIGAR or VULGAR should be parsed (provide -cigar=>1 when initializing the object)
Genewise Genomewise	Bio::SearchIO::wise	-genesf output	This parses the -genesf or -genes output from Genewise and Ger
Sim4	Bio::SearchIO::sim4	A={0,1,3,4} output	Cannot parse LAV or 'exon file' formats (A=2 or A=5)
PSL alignment format BLAT UCSC tools	Bio::SearchIO::psl	psl format	with or without -noHead option

Pdoc - HTML Perldoc



Perldoc (Pdoc rendered) documentation for Bioperl Modules

Released Code

Official documentation for released code is available here

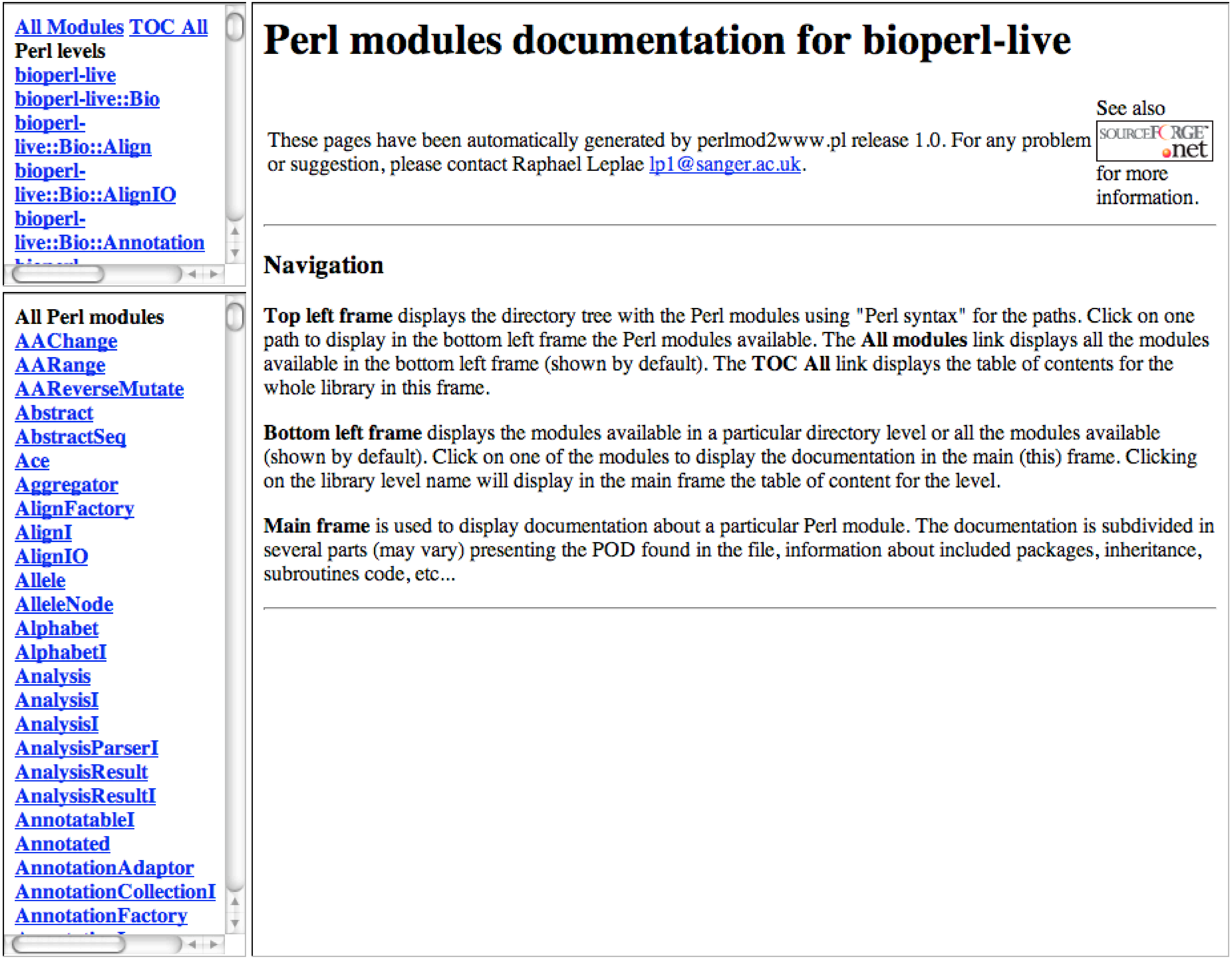
- Bioperl 1.5.0-RC1, download the entire doc set here
- Bioperl 1.4, download the entire doc set here
- Bioperl 1.2.3, download the entire doc set here
- Bioperl 1.2.2, download the entire doc set here.
  - The CPAN Pod2HTML rendered version of the code. Some people find this more intuitive, note that a few modules are not being rendered correctly by Pdoc so you may find this site helpful.
- Bioperl 1.2, download the entire doc set here.
- Bioperl 1.0.2, download the entire doc set here.
- Bioperl 1.0.1, download the entire doc set here.
- Bioperl 1.0, download the entire doc set here.

Active Code

This documentation represents the active development code and is autogenerated daily from the CVS repository.

Module	Description
bioperl-live	Bioperl Core Code
bioperl-corba-server	Bioperl BioCORBA Server Toolkit (wraps bioperl objects as BioCORBA objects and runs them in an ORBit ORB)
bioperl-corba-client	Bioperl BioCORBA Client Toolkit (wraps BioCORBA objects as bioperl objects)

http://doc.bioperl.org



All Modules TOC All

Perl levels

bioperl-live

bioperl-live::Bio

bioperl-live::Bio::Align

bioperl-live::Bio::AlignIO

bioperl-live::Bio::Annotation

All Perl modules

AAChange

AARange

AAReverseMutate

Abstract

AbstractSeq

Ace

Aggregator

AlignFactory

AlignI

AlignIO

Allele

AlleleNode

Alphabet

AlphabetI

Analysis

AnalysisI

AnalysisI

AnalysisParserI

AnalysisResult

AnalysisResultI

AnnotatableI

Annotated

AnnotationAdaptor

AnnotationCollectionI

AnnotationFactory

Perl modules documentation for bioperl-live

These pages have been automatically generated by perlmod2www.pl release 1.0. For any problem or suggestion, please contact Raphael Leplae lp1@sanger.ac.uk.

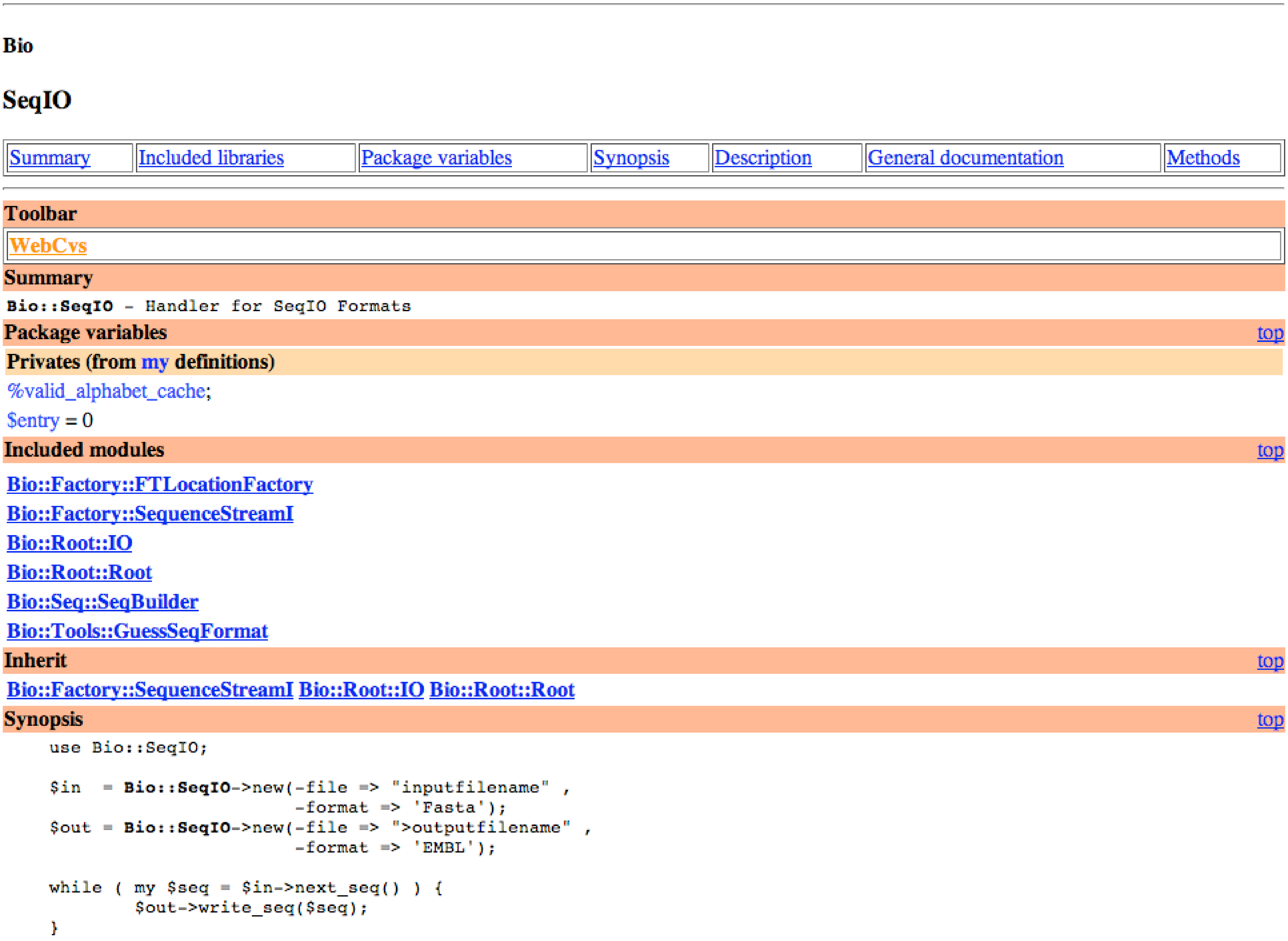
See also SOURCEFORGE.net for more information.

Navigation

Top left frame displays the directory tree with the Perl modules using "Perl syntax" for the paths. Click on one path to display in the bottom left frame the Perl modules available. The **All modules** link displays all the modules available in the bottom left frame (shown by default). The **TOC All** link displays the table of contents for the whole library in this frame.

Bottom left frame displays the modules available in a particular directory level or all the modules available (shown by default). Click on one of the modules to display the documentation in the main (this) frame. Clicking on the library level name will display in the main frame the table of content for the level.

Main frame is used to display documentation about a particular Perl module. The documentation is subdivided in several parts (may vary) presenting the POD found in the file, information about included packages, inheritance, subroutines code, etc...



Bio

SeqIO

Summary	Included libraries	Package variables	Synopsis	Description	General documentation	Methods

Toolbar

WebCvs

Summary

Bio::SeqIO - Handler for SeqIO Formats

Package variables

top

Privates (from my definitions)

%valid_alphabet_cache;

$entry = 0

Included modules

top

Bio::Factory::FTLocationFactory
Bio::Factory::SequenceStreamI
Bio::Root::IO
Bio::Root::Root
Bio::Seq::SeqBuilder
Bio::Tools::GuessSeqFormat

Inherit

top

Bio::Factory::SequenceStreamI Bio::Root::IO Bio::Root::Root

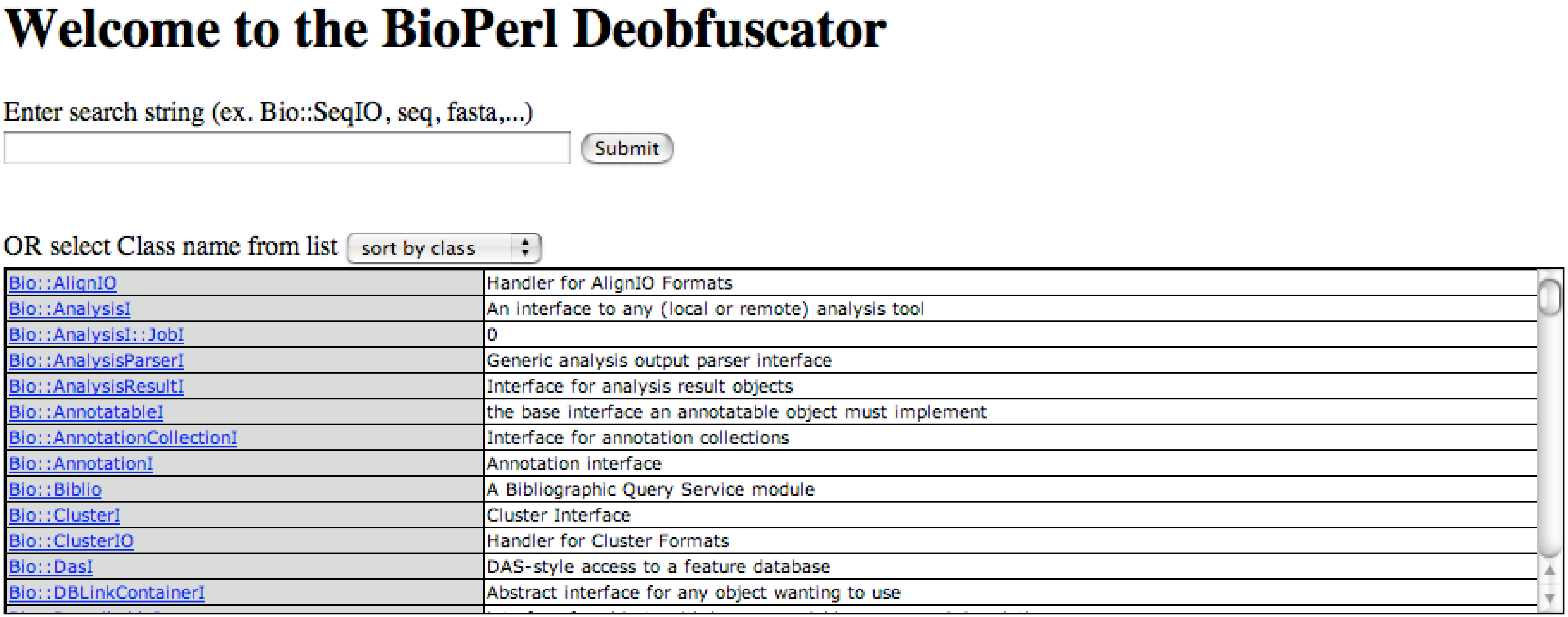
Synopsis

top

```
    use Bio::SeqIO;

    $in  = Bio::SeqIO->new(-file => "inputfilename" ,
                           -format => 'Fasta');
    $out = Bio::SeqIO->new(-file => ">outputfilename" ,
                           -format => 'EMBL');

    while ( my $seq = $in->next_seq() ) {
            $out->write_seq($seq);
    }
```

http://doc.bioperl.org/bioperl-live/Bio/SeqIO.html

DeObfuscator



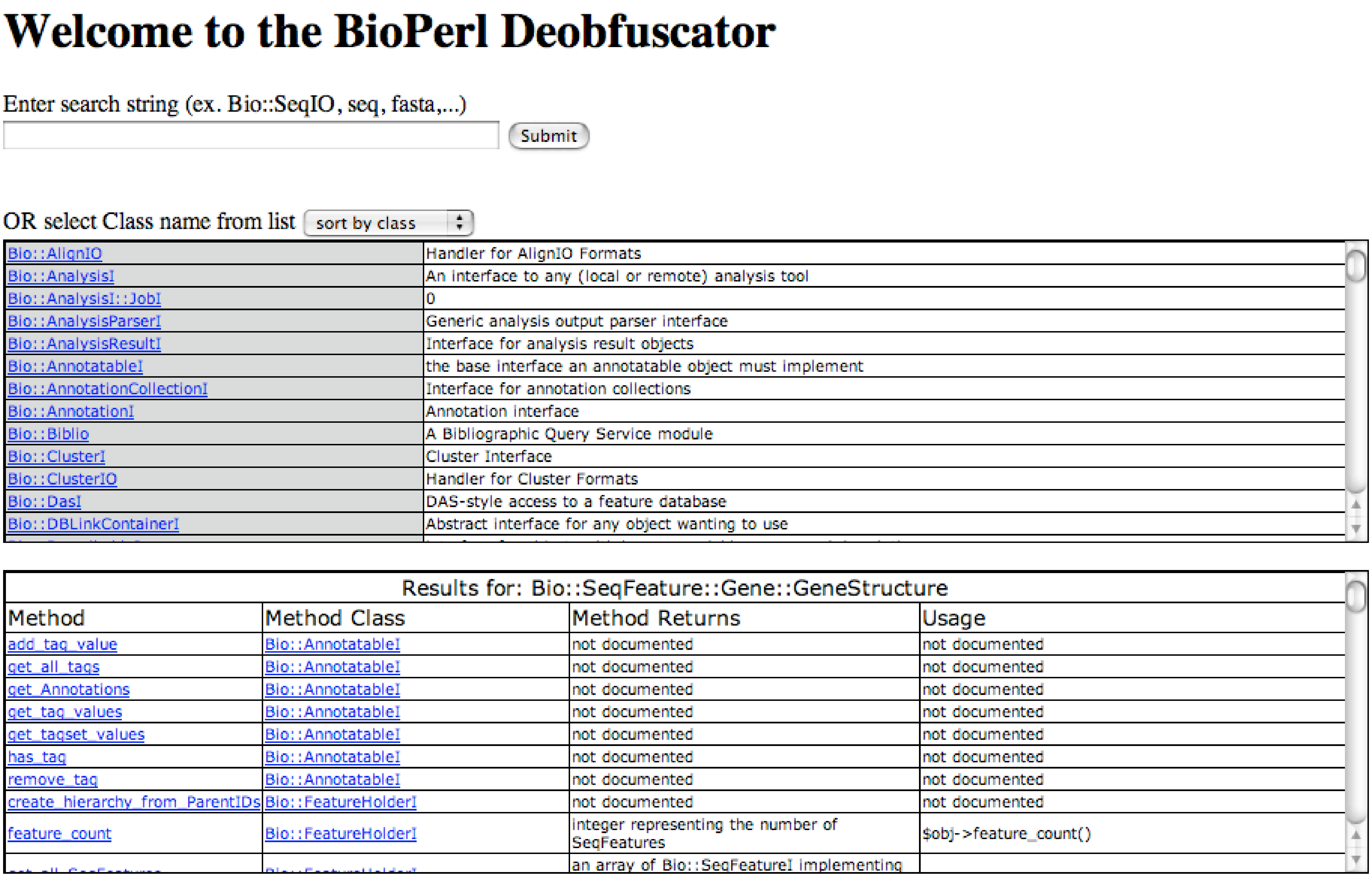
Welcome to the BioPerl Deobfuscator

Enter search string (ex. Bio::SeqIO, seq, fasta,...)

OR select Class name from list

Bio::AlignIO	Handler for AlignIO Formats
Bio::AnalysisI	An interface to any (local or remote) analysis tool
Bio::AnalysisI::JobI	0
Bio::AnalysisParserI	Generic analysis output parser interface
Bio::AnalysisResultI	Interface for analysis result objects
Bio::AnnotatableI	the base interface an annotatable object must implement
Bio::AnnotationCollectionI	Interface for annotation collections
Bio::AnnotationI	Annotation interface
Bio::Biblio	A Bibliographic Query Service module
Bio::ClusterI	Cluster Interface
Bio::ClusterIO	Handler for Cluster Formats
Bio::DasI	DAS-style access to a feature database
Bio::DBLinkContainerI	Abstract interface for any object wanting to use

http://www.davemessina.com/deobfuscator/CGI_interface.pl



Welcome to the BioPerl Deobfuscator

Enter search string (ex. Bio::SeqIO, seq, fasta,...)

OR select Class name from list

Bio::AlignIO	Handler for AlignIO Formats
Bio::AnalysisI	An interface to any (local or remote) analysis tool
Bio::AnalysisI::JobI	0
Bio::AnalysisParserI	Generic analysis output parser interface
Bio::AnalysisResultI	Interface for analysis result objects
Bio::AnnotatableI	the base interface an annotatable object must implement
Bio::AnnotationCollectionI	Interface for annotation collections
Bio::AnnotationI	Annotation interface
Bio::Biblio	A Bibliographic Query Service module
Bio::ClusterI	Cluster Interface
Bio::ClusterIO	Handler for Cluster Formats
Bio::DasI	DAS-style access to a feature database
Bio::DBLinkContainerI	Abstract interface for any object wanting to use

Results for: Bio::SeqFeature::Gene::GeneStructure

Method	Method Class	Method Returns	Usage
add_tag_value	Bio::AnnotatableI	not documented	not documented
get_all_tags	Bio::AnnotatableI	not documented	not documented
get_Annotations	Bio::AnnotatableI	not documented	not documented
get_tag_values	Bio::AnnotatableI	not documented	not documented
get_tagset_values	Bio::AnnotatableI	not documented	not documented
has_tag	Bio::AnnotatableI	not documented	not documented
remove_tag	Bio::AnnotatableI	not documented	not documented
create_hierarchy_from_ParentIDs	Bio::FeatureHolderI	not documented	not documented
feature_count	Bio::FeatureHolderI	integer representing the number of SeqFeatures	$obj->feature_count()
get_all_SeqFeatures	Bio::FeatureHolderI	an array of Bio::SeqFeatureI implementing	

http://www.davemessina.com/deobfuscator/CGI_interface.pl?Search=Search&module=Bio::SeqFeature::Gene::GeneStructure&search_string=&words=sort+by+class

How do we deal with Perl6?

- LONG way off in our estimation
  - We target for minimum Perl versions supported by linux/BSD distros
- Could be good excuse for re-write
- Perl6 should be a better for Object-Oriented programming

Open Bioinformatics Foundation

- Not-for-profit foundation to support the Bio* projects
- BioPerl, BioJava, BioPython, BioMOBY, BioDAS, BioRuby
- Support for EMBOSS (mailing list and FTP)
- Infrastructure (3-4 servers for CVS, Mail, and Web)
- Run BOSC Conference

O|B|F Board

- President - Jason Stajich
- Secretary - Andrew Dalke
- Treasurer - Chris Dagdigian
- Parlimentarian - Hilmar Lapp
- At-Large members - Ewan Birney and Steven Brenner
- BOSC Chairman - Darin London

http://www.open-bio.org/

O|B|F Mission

- Supporting bioinformatics infrastructure toolkits
- Getting developers together
    - BOSC
    - Hackathons
- Promoting open source bioinformatics
Getting more developers involved

How can you get involved?

- Join mailing list of a project
- Contribute code
- Help the Board or BOSC
  - Website dev
  - Logistics for conferences, hackathons

Acknowledgements

BioPerl Core

Hilmar Lapp
Jason Stajich
Ewan Birney
Lincoln Stein
Brian Osborne
Heikki Lehväslaiho

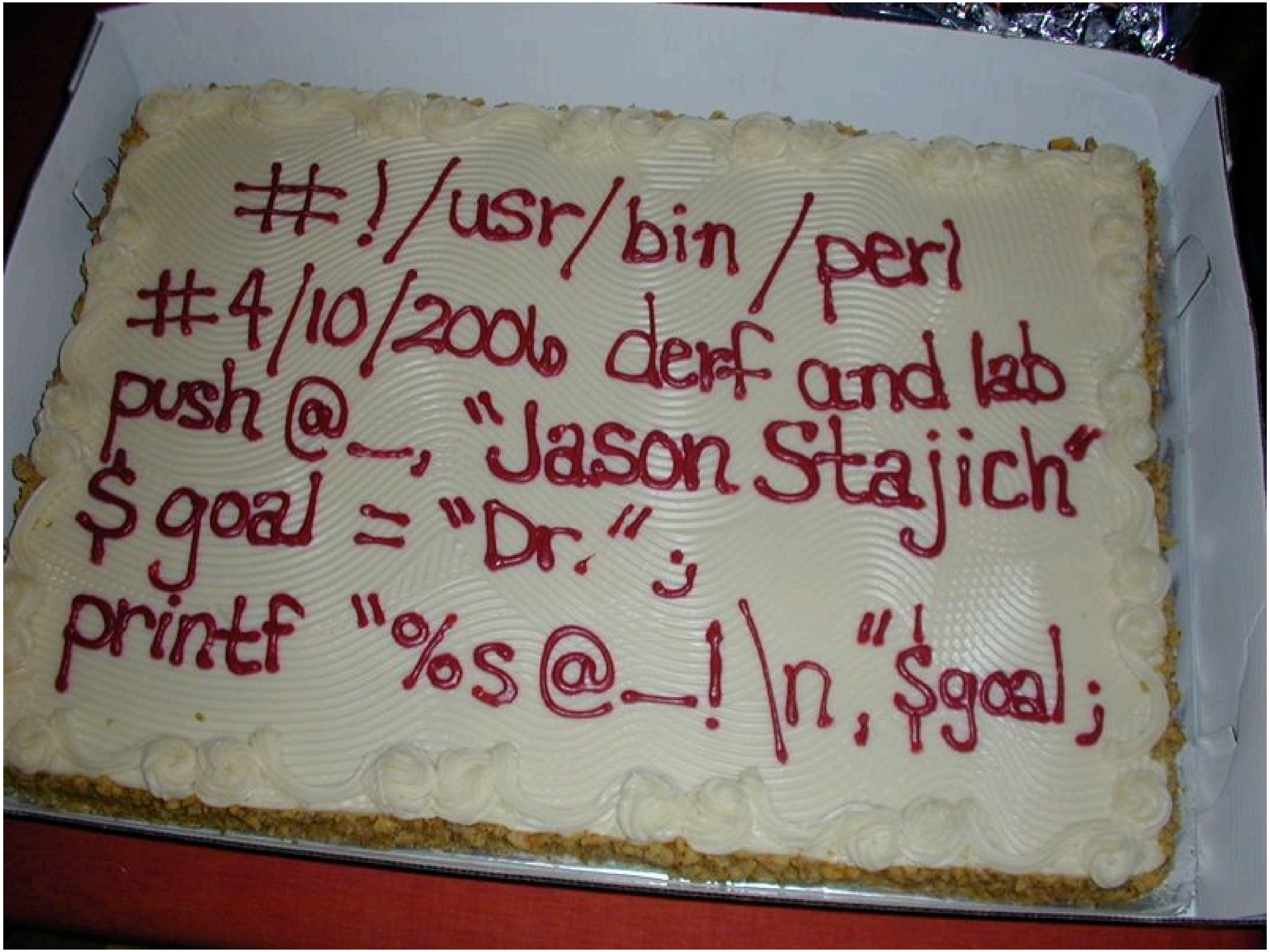
BioPerl

Aaron Mackey
Mauricio Herrrera Cuadra
Chris Fields
Marc Logghe
Todd Harris
Chris Mungall
Torsten Seemann
Steve Chervitz

Open Bioinformatics Foundation

Chris Dagdigian - Treasurer & SysOp

O|B|F



```
#!/usr/bin/perl
#4/10/2006 derf and lab
push @_, "Jason Stajich"
$goal = "Dr.";
printf "%s @_!\n,"$goal;
```