BioPerl: Developing Open Source Software

Jason Stajich
Duke University

BOSC 2005

Why Talk About Bioperl?

- Successful open-source project
- Bioinformatics is a difficult field to straddle
- Toolkit still has (many) flaws
- An evolving project

How Did We Get Here?

- A Brief History
  - Development Stats
  - People
- Bioperl in Action

Bioperl Development Goals

- Provide a useable Perl toolkit for life science data
- Solve problems that developers need
- Avoid the 1-off “everyone write a BLAST|GenBank|FASTA parser”
- Continuity of solutions

A Brief History

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

1996

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

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 paper
Gbrowse paper

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 released

100+ Paper references
bioperl-db to have official release
BOSC2005

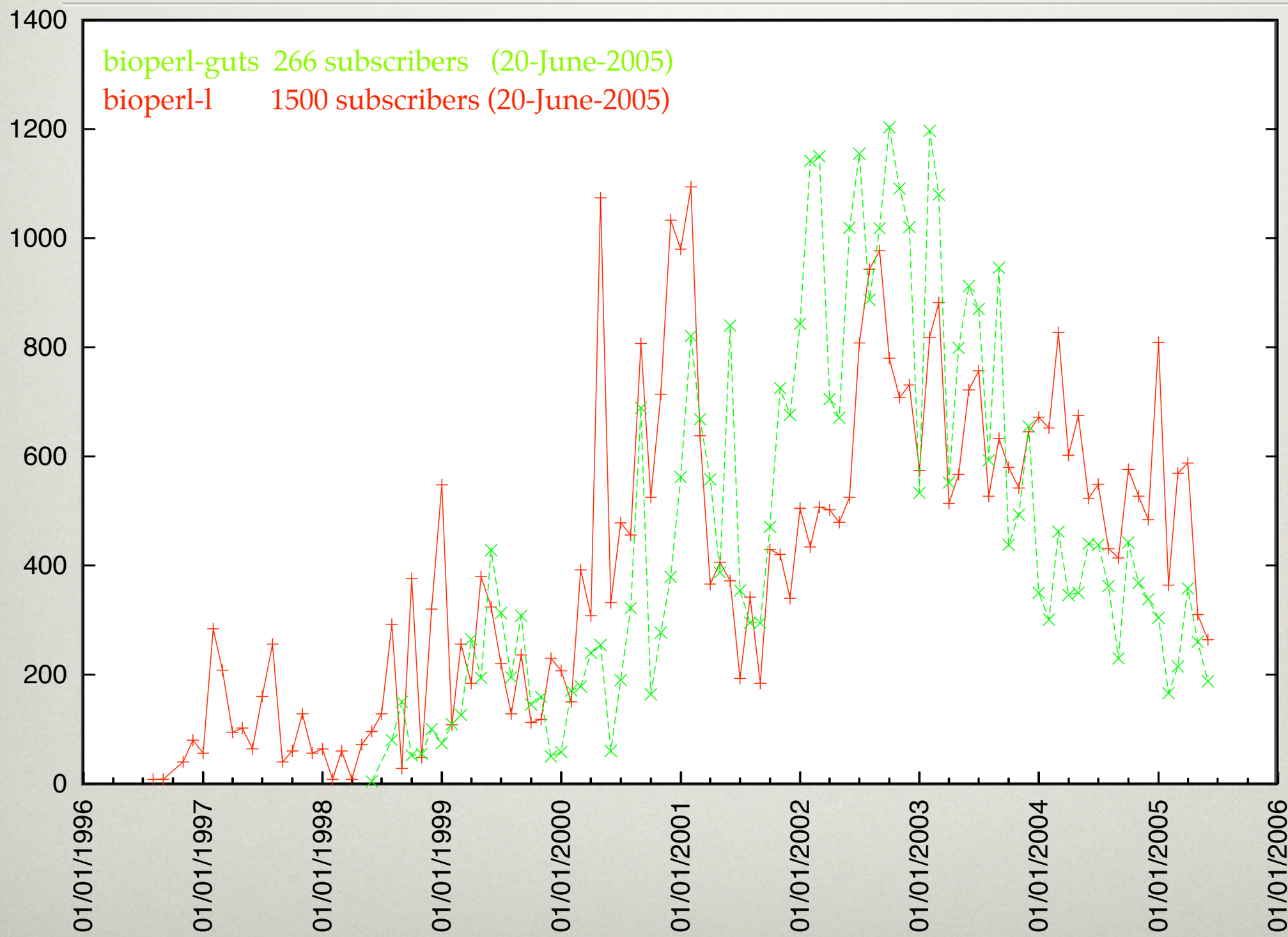
Defining Points

- Vision of module(s) set by code owner
- She/He who writes it, "wins" argument
- Project leader/core developers
  - can remove code before a release to insure a working release
  - Enforce code and module continuity
- Preserve API as much as possible

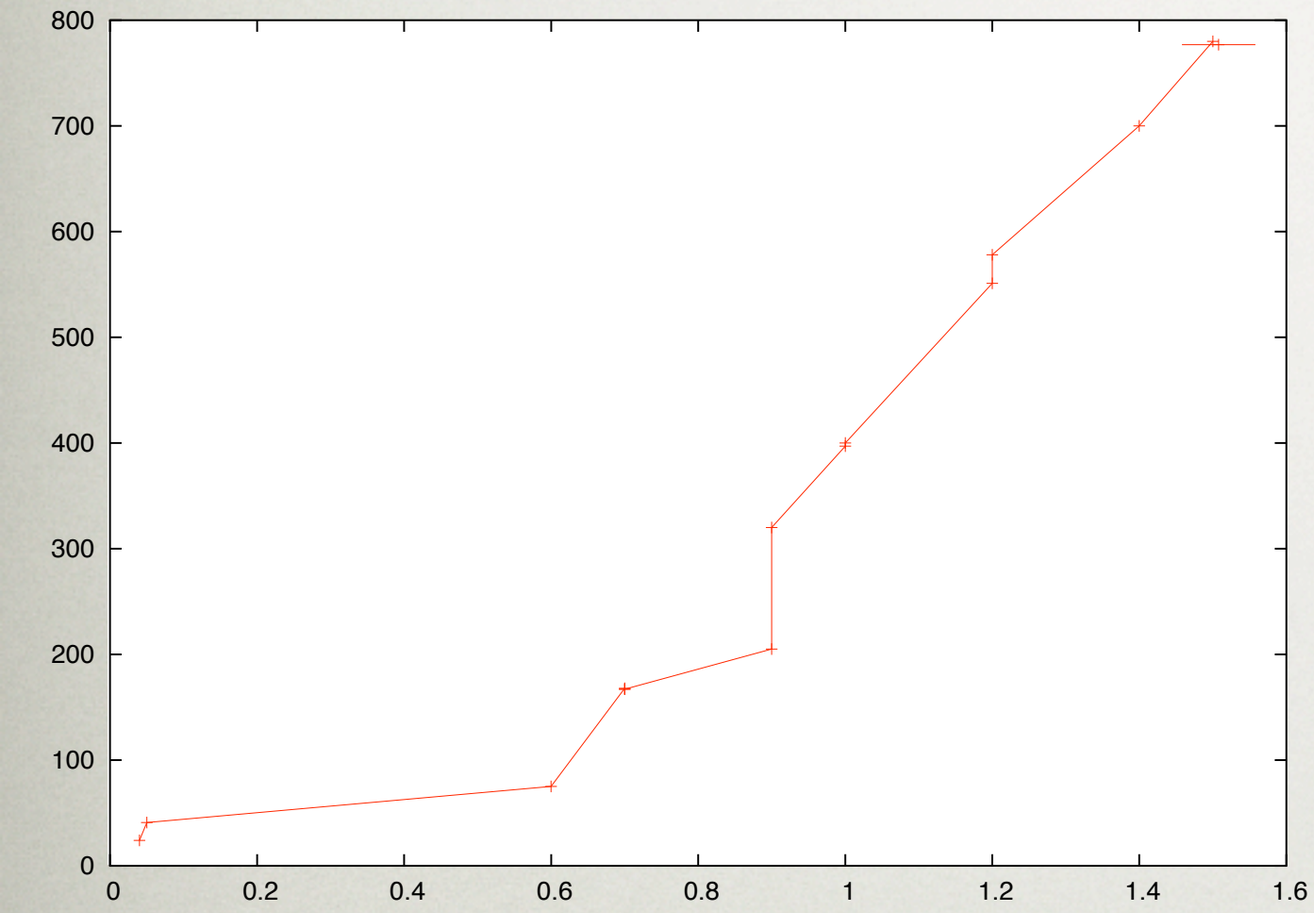
Project Mechanics

- Code is on a shared server (O|B|F)
- Publicly accessible
- Write-access to repository granted by main developers
- Discussion of code, ideas on-list

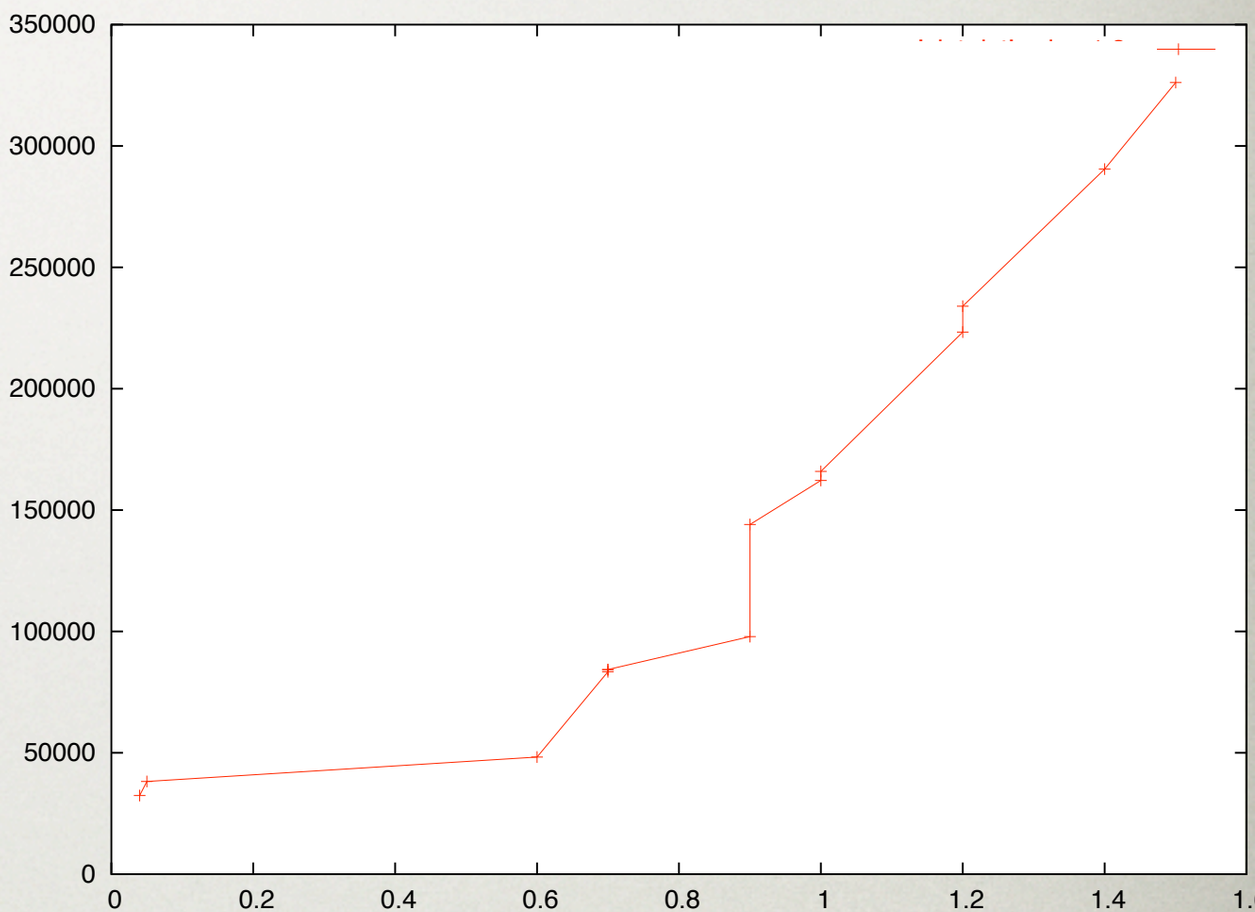
Bioperl Mailing List Traffic



Code Growth

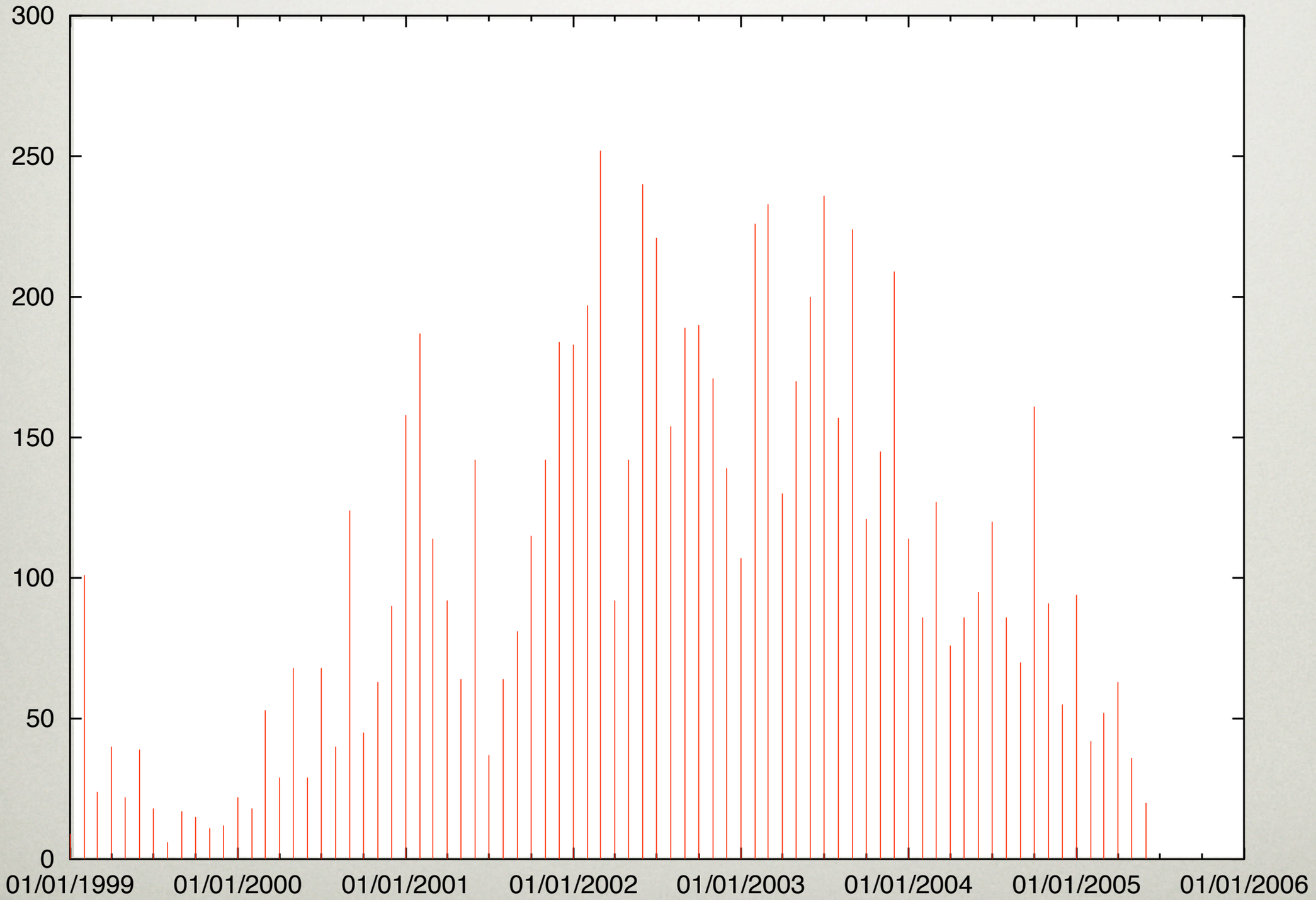


Modules per release



Lines per release

CVS Commits


300
250
200
150
100
50
0
01/01/1999
01/01/2000
01/01/2001
01/01/2002
01/01/2003
01/01/2004
01/01/2005
01/01/2006

Bioperl in Action

```
# Convert sequence formats
use Bio::SeqIO;
my $in = Bio::SeqIO->new(-format => 'genbank', -file=> 'file.gbk');
my $out = Bio::SeqIO->new(-format =>'fasta',-file =>'>file.fas');

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

```
use Bio::SearchIO;
my $in = Bio::SearchIO->new(-format => 'blast',
                            -file=> 'result.blastp');
while( my $r = $in->next_result ) {
  while( my $h = $r->next_hit ) {
    while( my $hsp = $h->next_hsp ) {
         print $hsp->query->start, "..",$hsp->query->end,"\n";
    }
  }
}
```

Bioperl in Action

```
# Convert sequence formats
use Bio::SeqIO;
my $in = Bio::SeqIO->new(-format => 'embl', -file=> 'file.embl');
my $out = Bio::SeqIO->new(-format =>'gcg',-file =>'>file.gcg');

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

```
use Bio::SearchIO;
my $in = Bio::SearchIO->new(-format => 'fasta',
                            -file   => 'result.fastp');
while( my $r = $in->next_result ) {
  while( my $h = $r->next_hit ) {
    while( my $hsp = $h->next_hsp ) {
         print $hsp->query->start, "..",$hsp->query->end,"\n";
    }
  }
}
```

Measuring sucess

- 2002 Paper has 100+ 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

Knowledge of Bioperl is A Marketable Skill

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://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://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://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

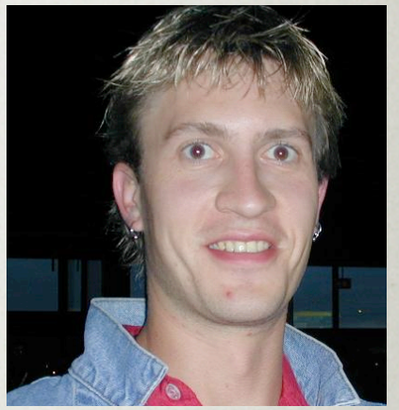
Why has the Project Worked?

- A critical mass of individuals who wanted to solve a common problem
- Infrastructure that works
- Sense of community
- Sharing information, tutorials

Some Key Developers

Not Pictured

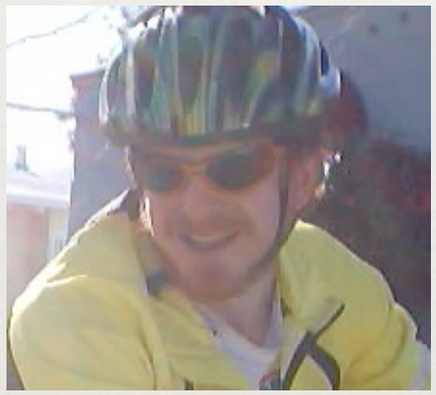
Richard Adams
Scott Cain
Sean Davis
Stefan Kirov
Nathan Haigh
Marc Logghe



Chad

Heikki

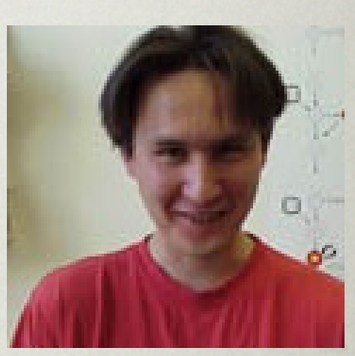
Aaron



Steve

Jason

Shawn



Allen

Ewan

Chris M

Hilmar

Lincoln

Brian

Chris D

Infrastructure

- Easily accessible CVS, web service
- Reliable
- Machines that we own
  - Additional services can be added
- Moving to faster machines this summer

Collaborative Nature of the Project

- Folks working on similar problems
- Pooled development resources
- Friendly & interactive way to do science
- Shared user support load...

Open Source is Good!

- Fix things faster
  - (If you really care about the fix)
- Everyone can see solution, audit code
- Contribution can be modular
- “Give a little, get a lot”
- More developers for less resources

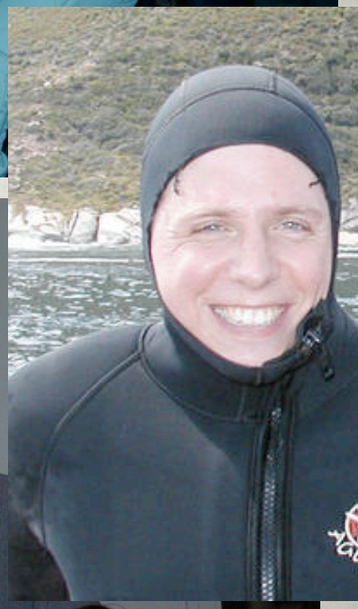
Open Source Doesn't Solve Everything

- Best laid plans
  - Decision by consensus ...
  - ... Free-for-all committing
- Things don't "just happen"
  - Need strong leadership, vision, and commitment
- Documentation tends to lag
- Interesting, needed code trumps boring

Great Power == Great Responsibility

- Making releases
  - Point releases maintain API compatibility between stable releases
- Many people depend on the code
- Additional modules should maintain continuity

We Know how to have fun!



Bioperl: Present Day

- Parsers for
  - Sequence & alignment parsing
  - Gene Prediction output
  - Phylogenetic trees
  - Weight Matricies (TFBS)
  - Ontologies
  - Structure data
  - Various CompBio, MolEvol apps

Bioperl: Present Day

- Tools for
  - Sequence manipulation (I/O, manip)
  - Graphical sequence rendering
  - Sequence & alignment stats, popgen tests, molevo tests
  - Access to Local & Remote Sequence Databases

Present Day Stats

- 780+ modules
- 336,000 lines of code
- ~9,000 tests in unit test ‘t’ dir
- 57 utility scripts, 60 example scripts
- 11 HOWTO docs
- 44 Q&A in FAQ

Bioperl Magic

- Using interfaces to hide specifics
  - Bio::DB::GenBank & Bio::Index::GenBank
- Objects can masquerade like other objects
  - Bio::SeqFeature::Generic, Bio::Location, DB::GFF::RelSegment, DB::GFF::Feature

Bioperl Magic

```
use Bio::DB::GenBank;
use Bio::DB::Fasta;

my $db = Bio::DB::GenBank->new;
my $idx = Bio::DB::Fasta->new($fafile_dir);
```

```
my $seq1 = $db->get_Seq_by_acc($acc);
my $seq2 = $idx->get_Seq_by_acc($acc);
```

```
use Bio::DB::FileCache;
my $cachedb = Bio::DB::FileCache->new($db);
my $acc = $cachedb->get_Seq_by_acc($acc);
```

...The Problems

- Interfaces system is cluttered
  - Learning curve is steep
  - Nonsensical to some people
- OO Code is too slow
- Developer community needs growth

Why is Bioperl Slow (in some places)?

1. Object-Oriented Perl is a Hack!
    1. $class->SUPER::new(@_);
    2. $class->SUPER::_rearrange(@_);
    3. Object overhead can be high.

Research with Bioperl

- Genome annotation pipeline
- Genome Browsing
- Comparative genomics
- Phylogenomics

Existing Genome Annotation

New Annotation

Hemiascomycota

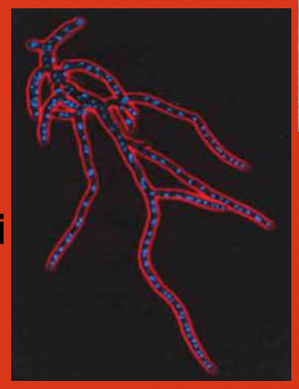
Euascomycota

Archiascomycota

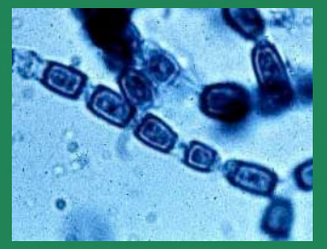
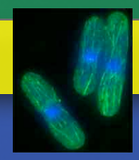
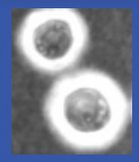
Basidiomycota

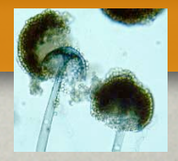
Zygomycota

S.cerevisiae
C.glabrata
K.lactis
K.waltii
A.gossypii
D.hansenii
C.guilliermondii
C.lusitaniae
Y.lipolytica
P.anserina
C.globsum
N.crassa
M.grisea
F.graminearum
A.fumigatus
A.nidulans
H.capsulatum
C.immitis
S.pombe
C.cinereus
P.chrysosporium
C.neoformans
U.maydis
R.oryzae
H.sapiens
A.thaliana



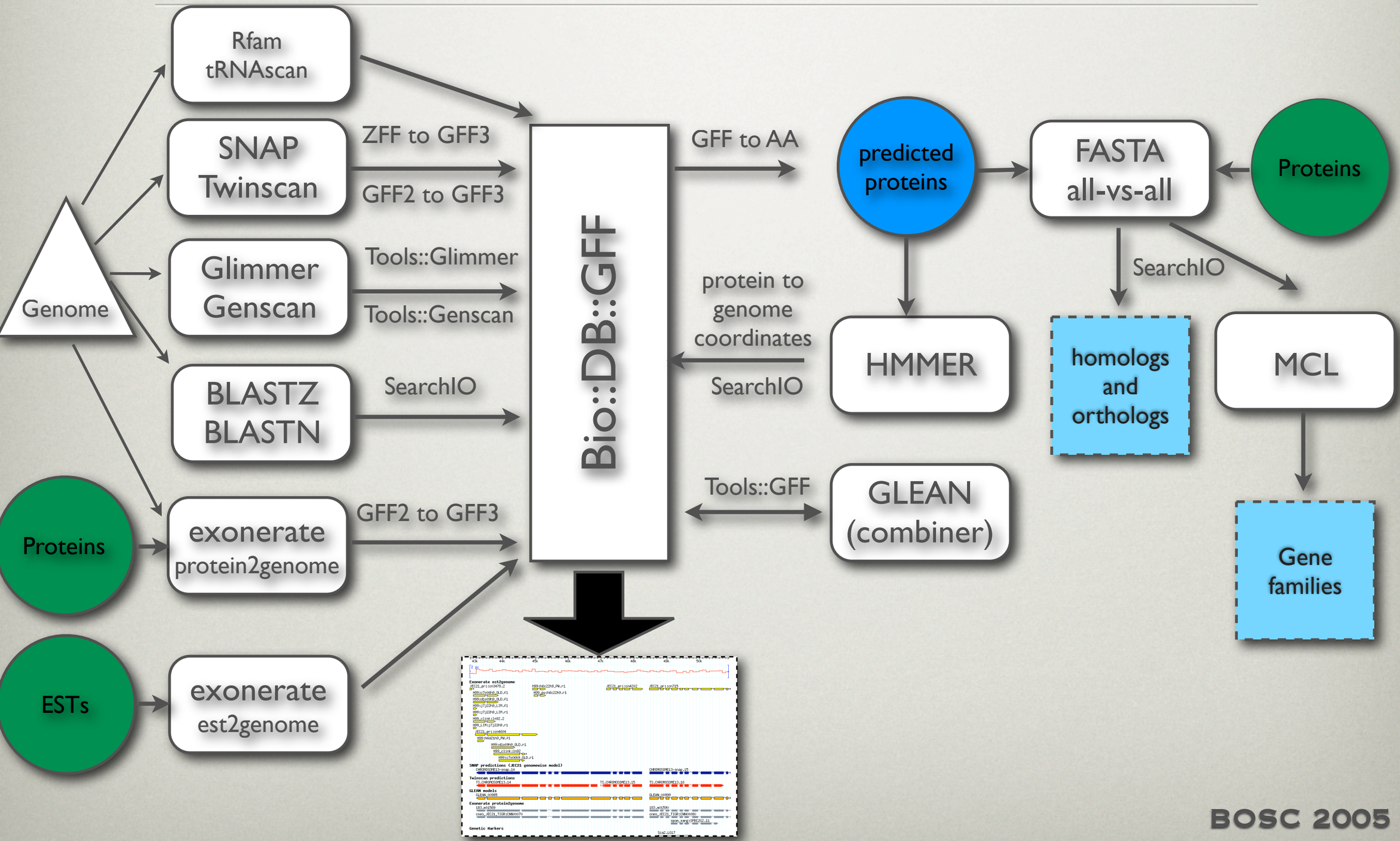






10

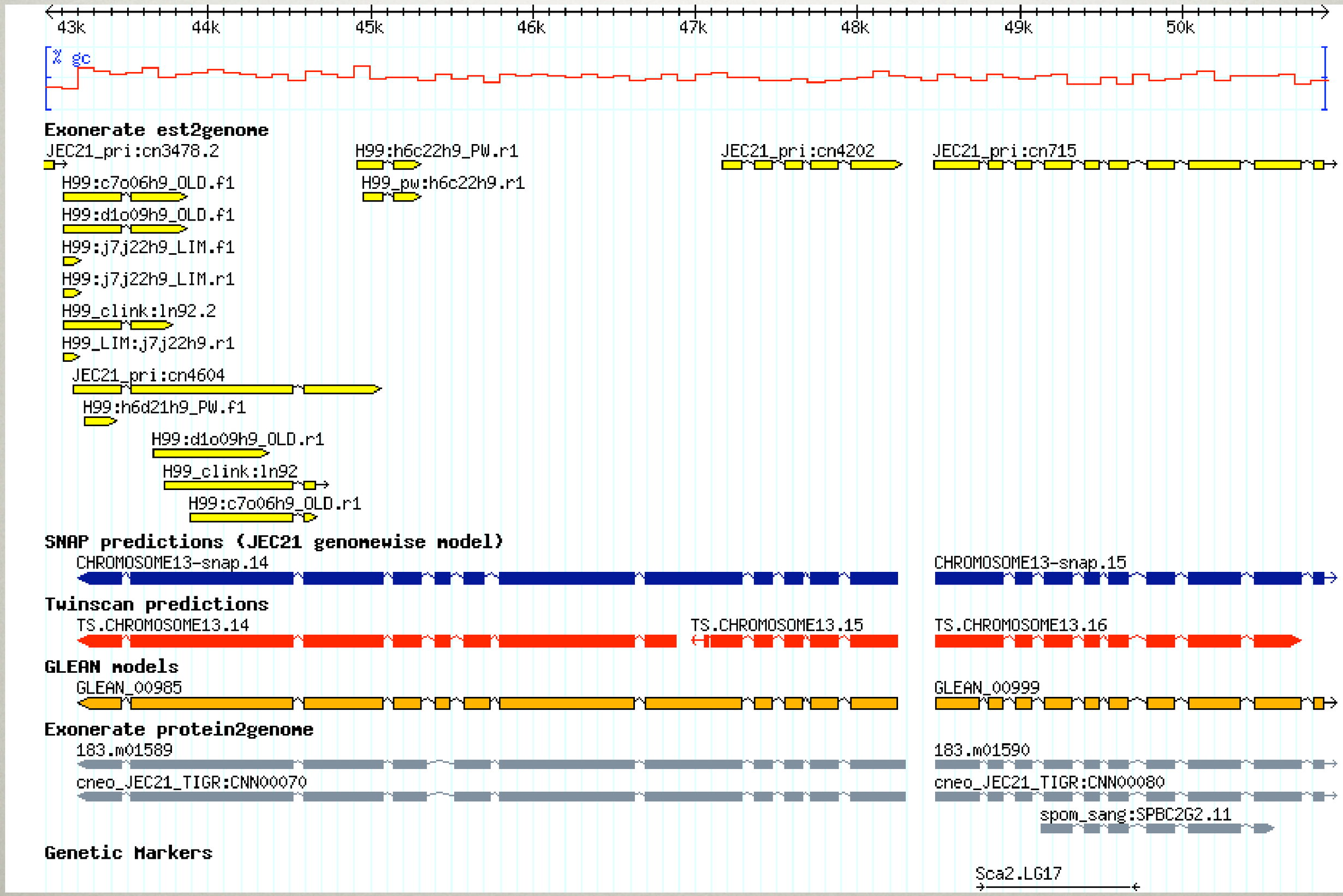
Protein ML tree based on 100 random orthologs

Genome Annotation Pipeline



BOSC 2005

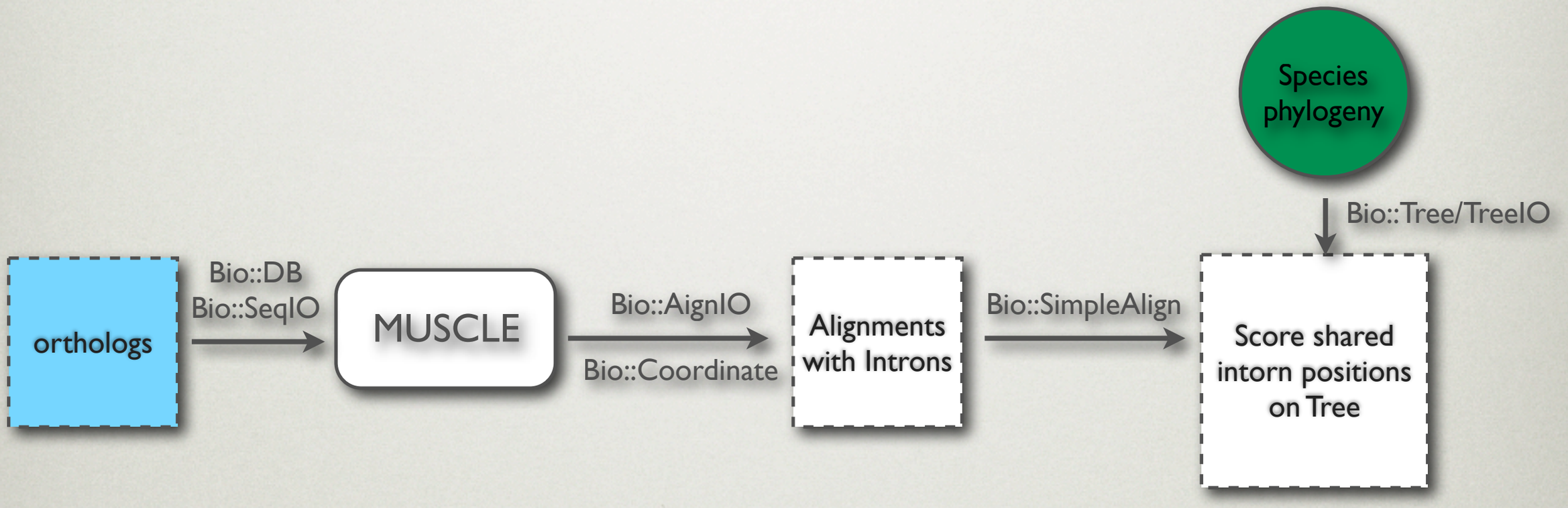
Genome Browsing



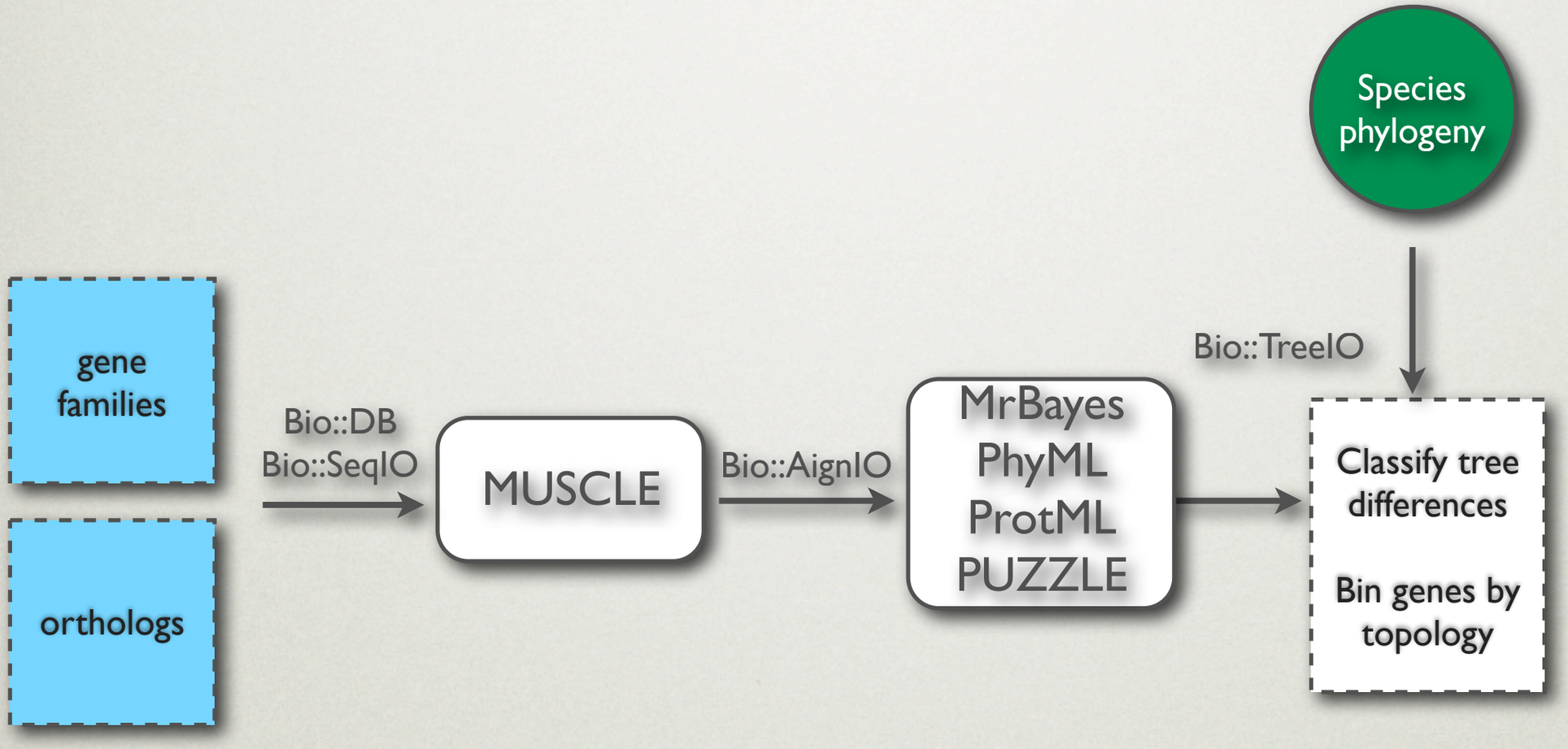
Comparative Genomics

- Find orthologs, align, build trees
  - Assess gene structure among orthologs
- Find gene families
  - Assess gene structure
  - Copy number among genomes

Comparative Genomics: Gene Structure



Phylogenomics



Bioperl - The Future

Where Should Bioperl be Headed?

- Concise set of functionality that just works?
- Supporting leading edge of research?
- Beginner level tools?
- Advanced tools?

Challenges

- Logistics
  - Building and maintaining large projects takes a lot of time
  - Project leaders with vision (and time)
  - Face-to-face time invaluable for coordination
  - Need more dedicated developers

Starting Over?

- Years of organic development don't make a wholly consistent toolkit
- Keeping POD up to date vs API maintenance
  - Autodoc-ing?
- Perl6 rewrite?
- Do we start a Bioperl 2, from scratch?

Supporting the Toolkit

- Consider applying for funding to support a developer?
- Finding companies to provide more dedicated user support?
- Gatherings (hackathons)
  - DocFests to increase documentation
- Better incentives for bug fixes

My Priority Queue

- Simplified API, hidden interfaces
- Overview docs point functionality to modules
- Speed
- Extensibility

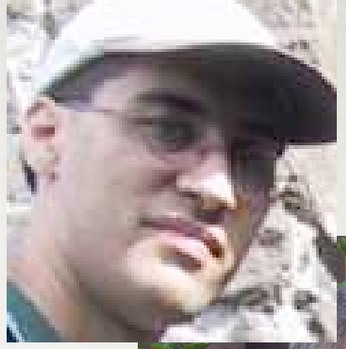
A Plan

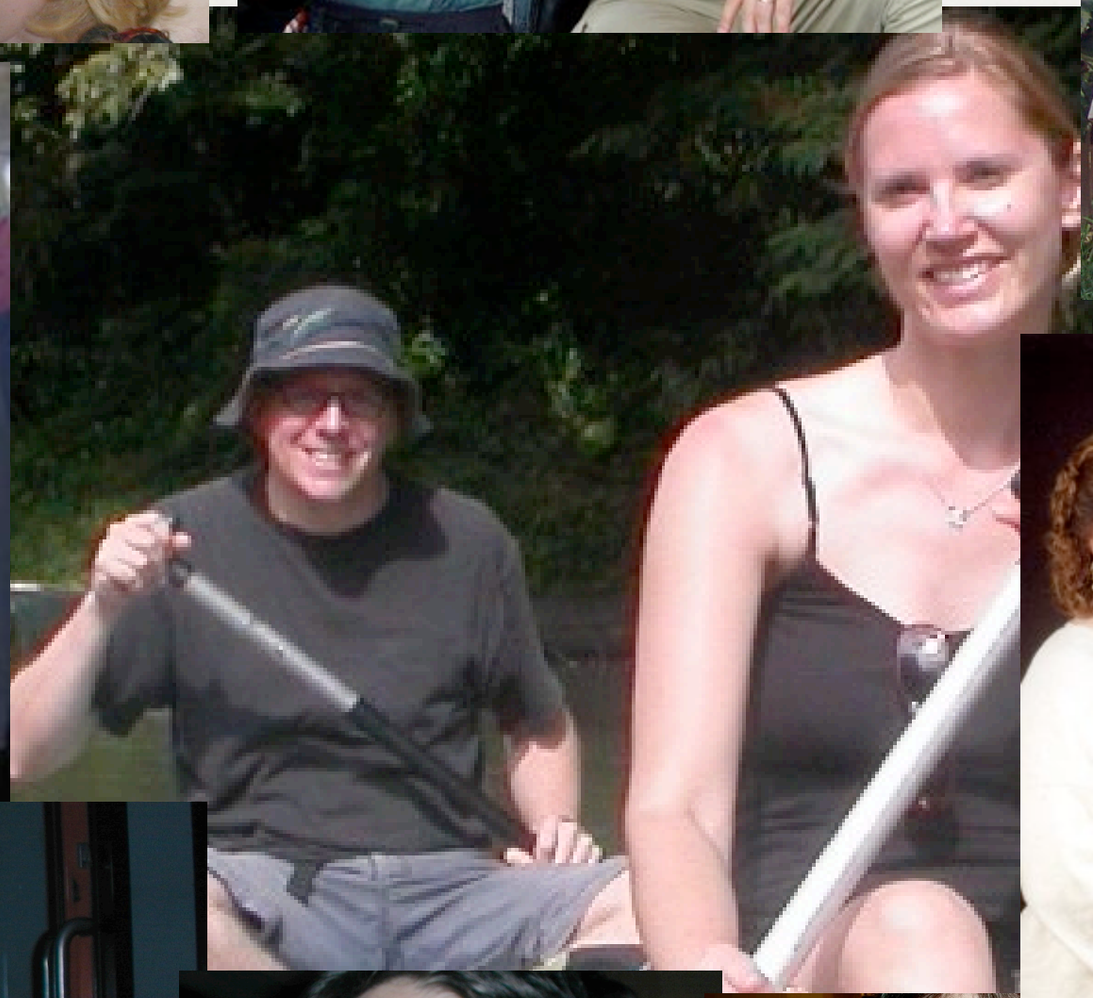
- Continue Bioperl 1.x
  - Try and fix interface overload
  - Work out speed issues (maybe)
- Encourage dabbling into a 2.0 codebase
  - Outline some key principals in new codebase.
  - Consider prototypes & Perl6

We've Grown Up

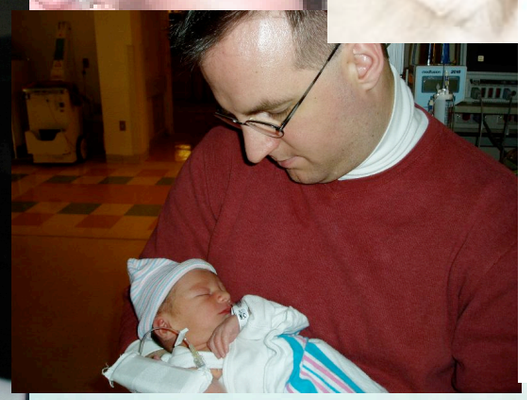
- Over the course of the project we have
  - Gotten married
  - Had kids
  - Changed continents
  - Changed jobs








2005

Thanks

- Developers - past and present
  - Ewan Birney
  - Lincoln Stein
  - Chris Dagdigian
  - Brian Osborne