BioRuby and distributed development

Ruby project update

Pjotr Prins, Ben Woodcroft, Francesco Strozzi, Hiroyuki Mishima, Toshiaki Katayama, Naohisa Goto, Raoul Bonnal, Joachim Baran

Bioinformatics Open Source FEST (BOSF) 2014
I think I'd rather manage a large software development project.

The daydreams of cat herders
Language wars

Python has won
I am a Commodore guy

Commodore won!
And now for something... 

Ruby is different
Why Ruby?

Ruby is my favorite programming language because

- Dynamically typed *(Ruby LISP Python Perl R JS)*
- Fully OOP *(Ruby Python Perl)*
- Pretty good functional programming *(LISP Ruby Python Perl6 JS ,Coffee)*
- Great meta programming *(LISP Ruby Python Perl6)*
- Ruby allows concise but *readable* code
- Ruby attracts the right crowd
Who are Rubyistas?

Ruby has leading initiatives in

- Web frameworks (Rails, Sinatra, Padrino etc.)
- Testing frameworks (Cucumber, Travis CI)
- Web solutions (github, twitter, Hulu, Heroku etc.)
- Software deployment (Homebrew, Chef, Puppet, Vagrant)
- Notably meta: innovation at the process level
Norwegian reminder

SO, what has this to do with BioRuby?
We leveraged Rubygems to provide BioRuby plugins (meta)

Easy! Run:
biogem -with-bin -with-ffi -cucumber name

100 new projects in 4 years (2/month)

6 new contributors in last year

_Biogem: an effective tool based approach for scaling up open source software development in bioinformatics_, Bonnal _et al._, Bioinformatics 2012
doi:10.1093/bioinformatics/bts080
<table>
<thead>
<tr>
<th>Bio</th>
<th>Description</th>
<th>Author</th>
<th>Version</th>
<th>Age</th>
<th>Rating</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>lazyblatxml</td>
<td>A plugin that allows you to parse large file formats</td>
<td>robayne</td>
<td>0.4.0</td>
<td>3 years</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>pileup</td>
<td>Iterate through a samtools pileup file</td>
<td>Ben J. Woodcroft</td>
<td>0.0.4</td>
<td>19 months</td>
<td>2 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>cd hit</td>
<td>Read and manipulate cd hit clusters</td>
<td>George Githinji</td>
<td>0.1.0</td>
<td>14 months</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>kb illumina</td>
<td>Gene expression analysis &amp; annotation</td>
<td>Raoul J.P. Bonnal</td>
<td>0.1.0</td>
<td>3 years</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>plasmoap</td>
<td>Implementation of the plasmoap algorithm</td>
<td>Ben J Woodcroft</td>
<td>0.0.2</td>
<td>2 years</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>tabix</td>
<td>Ruby binding for samtools tabix file</td>
<td>throwern</td>
<td>1.0.1</td>
<td>9 months</td>
<td>3 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>gff3</td>
<td>Ruby wrapper for the gff3 file</td>
<td>Marjan Pavolni</td>
<td>0.3.0</td>
<td>23 months</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>chembl</td>
<td>Chembli rest api client, parser and container</td>
<td>Mitsuteru Nakao</td>
<td>0.1.3</td>
<td>23 months</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>ucsf util</td>
<td>Ruby binding to the ucsf annotation utilities</td>
<td>throwern</td>
<td>0.1.2</td>
<td>21 months</td>
<td>2 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>gmqm</td>
<td>Next-generation mapping of mutations</td>
<td>Dan MacLean</td>
<td>0.2.1</td>
<td>19 months</td>
<td>2 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>rdf</td>
<td>Bioinformatics triple-store support</td>
<td>Pjotr Prins</td>
<td>0.0.2</td>
<td>2 years</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>affy</td>
<td>Parse otu metaxa data files</td>
<td>Pjotr Prins</td>
<td>0.5.1</td>
<td>2 years</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>basespace</td>
<td>Basespace Ruby sdk is a Ruby based software</td>
<td>Joachim Barna, Raoul Bonnal, Eri Kubukawa et al.</td>
<td>0.1.7</td>
<td>11 months</td>
<td>2 ✭</td>
<td>biogems.info</td>
</tr>
<tr>
<td>core</td>
<td>BioRuby core plugins, most and stable plugins</td>
<td>Raoul J.P. Bonnal</td>
<td>0.0.1</td>
<td>2 years</td>
<td>1 ✭</td>
<td>biogems.info</td>
</tr>
</tbody>
</table>
RDF OOP

BioInterchange: conversion of standardized life sciences data formats into RDF (GFF3, GVF, Newick, Pubmet JSON, PDFx XML, etc.)

- standardisation & ontologies
- efficient (batch processing, no in-memory data model)
- extensible via clear use of OOP design patterns
- CLI

```
biointerchange -input phylotastic.newick -rdf rdf.phylotastic.newick
```

**RDF JSON**

- rdf2json: translating RDF into JSON-LD or minimized JSON alternative
  - goodness of NoSQL (Elasticsearch, MongoDB, RethinkDB, etc.)
  - CLI

```
rdf2json [options] -input filename.nt -output filename.json
```

Joachim Baran & Michel Dumontier [https://github.com/joejimbo/rdf2json](https://github.com/joejimbo/rdf2json)
RDF Functional

- Bio-rdf - introduce macros for SPARQL and RDF using erb
- Bio-vcf - convert VCF to RDF/HTML/LaTeX
- Bio-table - convert tabular data to RDF/HTML/LaTeX
- More on that in ‘small tools’

http://biogems.info/
SciRuby

Small tools manifesto

https://github.com/pjotrp/bioinformatics/blob/master/README.md

Branching out beyond life sciences

Cross over to SciRuby

Attract great developers for GSoC

- OpenCL
- Naoki Nishida D3 (demo of interactive plotting
  https://github.com/domitry/Nyaplot.js)
‘BioRuby’

- Distributed development (BioRuby is in maintenance mode - note)
- Meta, bin and pipes
- Large data handling
- Standards JSON, RDF etc.
- D3 JS library also usable from Python etc. (in fact, Iruby-notebook is Python based)
- Sambamba D language (Artem Tarasov)
BioRuby community

In all ‘the project formerly known as BioRuby’ is

- into distributed development
  (http://biogems.info/)
- has no hangups about computer languages
- reaching out (SciRuby, RDF) and crossing over...
What is missing in Ruby

- Strong functional support
- Concurrent programming
- Full expression matching
- Macros

These are also missing in Python, Perl

Language of the year 2014/2015: Elixir
Where are we heading?

- The world is parallel and multi-core...
- Immutable data (Erlang, Scala, D, Clojure, Haskell)
- Higher level abstractions for multi-threading (actors, pipes, continuations/futures)
- More functional programming (expression matching, macros)
Elixir

- Elixir is a dynamic functional language for building concurrent, distributed and fault-tolerant applications (no OOP)
- Elixir has immutable data, pattern matching, higher level abstractions for multi-threading and macros(!) in a readable Ruby inspired language
- Elixir is the love child of Ruby and Erlang (and LISP)

http://elixir-lang.org/
Interested? Dave Thomas talk
Nothing central; distributed development; glue is biogems.info

Reaching out

Plan: more tools/libraries/languages

Plan: GNU Guix and Cloudbiolinux deployment

Ruby DNA: sticky, reaching out and crossing over!
Acknowledgements

BioRuby ML, SciRuby and OBF (Biopython, BioPerl) for supporting the project formerly known as BioRuby

Personal genomics and bioinformatics, University Medical Center Utrecht for supporting FOSS work of Pjotr Prins

Center for Biomedical Informatics Research, Stanford University, for supporting FOSS work of Joachim Baran

Bioinformatics Core Facility of Parco Tecnologico Padano for supporting the FOSS work of Francesco Strozzi

Istituto Nazionale Genetica Molecolare for supporting FOSS work of Raoul Bonnal

Australian Centre for Ecogenomics for supporting FOSS work of Ben Woodcroft

and many others...