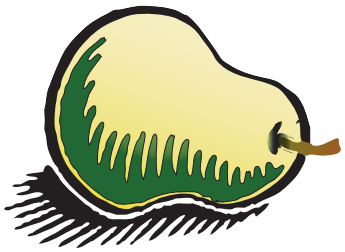
15^{th} Annual
Bioinformatics Open Source Conference
BOSC 2014



http://www.open-bio.org/wiki/BOSC_2014

11 & 12 July 2014, Boston, MA, USA

Welcome to BOSC 2014!

The Bioinformatics Open Source Conference, established in 2000, is held every year as a Special Interest Group (SIG) meeting in conjunction with the Intelligent Systems for Molecular Biology (ISMB) Conference. BOSC is organised by the Open Bioinformatics Foundation (O|B|F, www.open-bio.org), a non-profit group that promotes the practice and philosophy of Open Source software development and Open Science in the biological research community.

Talks and Posters

BOSC includes two full days of talks, posters, and Birds of a Feather interest groups (BOFs). Session topics this year include Software Interoperability, Visualization, Genome-Scale Data and Beyond, and Open Science and Reproducible Research, as well as the usual session on Bioinformatics Open Source Project Updates. Our panel topic this year is "Reproducibility: Rewards and Challenges". This year's keynote speakers will be Philip Bourne (NIH) and C. Titus Brown (Michigan State University).

There are poster sessions both days that start during the lunch hour. We have space for several last-minute posters, in addition to those listed in the program. Please contact us at bosc@open-bio.org if you'd like to present a last-minute poster.

Sponsors

We thank Eagle Genomics for sponsoring the BOSC Student Travel Awards again this year, and welcome GigaScience (an online open-access open-data journal for 'big-data' studies in the life and biomedical sciences), and Curoverse (the team behind the open source platform Arvados) as new sponsors for BOSC 2014. We also thank Google for their generous support for video-recording the talks at BOSC 2014.

BOSC Organizers

BOSC is a community effort – we thank all those who made it possible, including the organizing committee, the program committee, the session chairs, and the ISCB Conference Director, Steven Leard. If you are interested in helping to organize BOSC 2015, please email bosc@open-bio.org.

BOSC 2014 Organizing Committee

Nomi Harris and Peter Cock (Co-Chairs), Raoul Jean Pierre Bonnal, Brad Chapman, Robert Davey, Christopher Fields, Hans-Rudolf Hotz, Hilmar Lapp.

BOSC 2014 Program Committee

Tiago Antão, Kazuharu Arakawa, Raoul Bonnal, Timothy Booth, Brad Chapman, Peter Cock, Kam Dahlquist, Robert Davey, Thomas Down, Chris Fields, Björn Grüning, Nomi Harris, Michael Heuer, Hans-Rudolf Hotz, Amye Kenall, Juli Klemm, Hilmar Lapp, Heikki Lehväslaiho, Scott Markel, Hervé Ménager, Fiona Nielsen, Lorena Pantano, Michael Reich, Francesco Strozzi, Eric Talevich, Ronald Taylor, Ben Temperton.

ISCB Communities of Special Interest (COSIs)

Both the Open Bioinformatics Foundation (OBF, http://www.open-bio.org) and its annual Bioinformatics Open Source Conference (BOSC) were started in 2000. Since then BOSC has been held each year as one of the Special Interest Group (SIG) satellite meetings of the larger Intelligent Systems for Molecular Biology (ISMB) conference run by the International Society for Computational Biology (ISCB, http://www.iscb.org).

This year the ISCB is expanding the SIG scheme to recognise groups active all year round, rather than just at annual meetings in conjunction with the ISMB conference. The existing SIGs were invited to become ISCB Communities of Special Interest (COSIs).

The new COSI scheme is being formally launched at ISMB 2014, the day after BOSC, in a morning special session (SS02, Sunday, 13 July 2014, 10:30–13:25). The OBF (including BOSC) is one of the initial twelve COSIs, and OBF President (and BOSC committee member) Hilmar Lapp will give a brief introduction during the COSI special session.

Optional BOSC Dinner

We invite you to join BOSC organizers and attendees at a pay-your-own-way dinner the first evening of BOSC (Friday, 11 July, at 7pm) at The Asgard Irish Pub and Restaurant, 350 Massachusetts Avenue, Cambridge, MA. It is about 1.2 miles north of the Hynes convention center on Mass Ave. The #1 bus (towards Harvard Square) goes right there.

If you want to join us for dinner, RSVP at `http://bit.ly/BOSC2014-dinner` before Friday at 3pm. The restaurant has space for 30 BOSC guests; *only those who RSVP will be admitted.*

CodeFest

As in recent years, for the two days before BOSC we are holding the BOSC CodeFest, an informal "Coding Festival" or mini-hackathon: http://www.open-bio.org/wiki/Codefest_2014

The BOSC CodeFest 2014 is being hosted by hack/reduce, a wonderful hacker space in Cambridge.

Thanks to Amazon Web Services all participants will receive a $100 AWS credit to support work at the hackathon.

The CodeFest is also sponsored by Harbinger Partners, Inc. and Curoverse (the team behind the open source platform Arvados).

Keynote Speakers

Philip Bourne (Day Two)

Dr. Bourne will speak about "**Biomedical Research as an Open Digital Enterprise**":

> *The biomedical research lifecycle is fast becoming completely digital and increasingly open to the point that publishing could simply become changing the access control on given research objects comprising ideas, hypotheses, data, software, results, conclusions, reviews, grants and so on. This offers immense opportunities for software developers to enable the enterprise. I will describe a vision for the digital enterprise and what the NIH and others are doing to support the notion with the intent to accelerate scientific discovery.*

Philip E. Bourne, PhD, is the Associate Director for Data Science at the NIH and formerly Associate Vice Chancellor for Innovation and Industry Alliances, and Professor of Pharmacology at the University of California San Diego. Bourne's work at the NIH focuses on accelerating the rate of knowledge discovery from the ever-increasing amounts of biomedical data at all scales – from genomes to populations. Bourne's laboratory focuses on relevant biological and educational outcomes derived from computation and scholarly communication. This implies algorithms, text mining, machine learning, metalanguages, biological databases, and visualization applied to problems in systems pharmacology, evolution, cell signaling, apoptosis, immunology and scientific dissemination. He has published over 300 papers and 5 books. Bourne is the co-founder and founding Editor-in-Chief of the open access journal PLOS Computational Biology. He is a Past President of the International Society for Computational Biology, an elected fellow of the American Association for the Advancement of Science (AAAS), the International Society for Computational Biology (ISCB) and the American College of Medical Informatics. Bourne is committed to professional development through the Ten Simple Rules series of articles and a variety of lectures and video presentations. Awards include: the Jim Gray eScience Award (2010), the Benjamin Franklin Award (2009), the Flinders University Convocation Medal for Outstanding Achievement (2004), the Sun Microsystems Convergence Award (2002) and the CONNECT Award for new inventions (1996 & 97).

C. Titus Brown (Day One)

Dr. Brown's topic is "**A History of Bioinformatics (in the year 2039)**".

> *In 2039, I expect to look back at the last 25 years of biology and see both wonderful surprises and missed opportunities. In this talk, I will attempt to predict both some surprises and some of the opportunities I worry that we will have missed over the next 25 years.*

C. Titus Brown is an assistant professor in the Department of Computer Science and Engineering and the Department of Microbiology and Molecular Genetics. He earned his PhD ('06) in developmental molecular biology from the California Institute of Technology. Brown is director of the laboratory for Genomics, Evolution, and Development (GED) at Michigan State University. He is a member of the Python Software Foundation and an active contributor to the open source software community. His research interests include computational biology, bioinformatics, open source software development, and software engineering.

Main Talks

Talk abstracts are included in this program in the order in which they will be presented at the conference. These have been divided into five sessions:

- Genome-scale Data and Beyond
- Visualization
- Bioinformatics Open Source Project Updates
- Software Interoperability
- Open Science and Reproducible Research

This year we have allocated all the talks an equal slot, 15 minutes plus 3 minutes for questions (making 18 minutes), except for the project update session where this is reduced to 10 minutes plus 2 minutes for questions (making 12 minutes in all).

Lightning Talks

Lightning talks are 5 minutes only (no questions), and are intended for a brief introduction with the expectation that the expectation that people who are interested in the topic will find the speaker during a break or poster session to talk with them.

Posters

In addition to the talks, there will also be posters. Some, but not all, of the talks will also be presented as posters.

Authors should put up their posters in their assigned poster spot before the first poster session (which starts at 12:30 on the first day). After that time, any unused poster slots will be made available for last-minute posters.

The ISMB specifies that posters should not exceed the following dimensions: 46 inches (1.17m) wide by 45 inches (1.14m) high.

There will also be a few spaces available for last-minute posters. If you would like to present one, please email your abstract (which must meet the BOSC criteria of freely available source and recognized open source license) to bosc@open-bio.org.

Birds of a Feather (BOFs)

Birds of a Feather meetups are informal gatherings where participants group together based on common interests. We have set aside time when the room will be available to encourage these interactions, which will be co-ordinated via http://www.open-bio.org/wiki/BOSC_2014/BOFs on our wiki page.

BOSC Schedule - Day One - Friday, 11 July 2014

Time	Title	Author/Notes
7:30-9:00	**Registration**	
9:00-9:15	Introduction and Welcome	
9:15-10:15	**Keynote**: A History of Bioinformatics (in the Year 2039)	C. Titus Brown
10:15-10:45	**Coffee Break**	
10:45-12:30	***Session: Genome-scale Data and Beyond***	Chair: Chris Fields
10:45-11:03	ADAM: Fast, Scalable Genomic Analysis	Frank Austin Nothaft
11:03-11:21	A Framework for Benchmarking RNA-seq Pipelines	Rory Kirchner
11:21-11:39	New Frontiers of Genome Assembly with SPAdes 3.1	Andrey Prjibelski
11:39-11:57	SigSeeker: An Ensemble for Analysis of Epigenetic Data	Jens Lichtenberg
11:57-12:15	Galaxy as an Extensible Job Execution Platform	John Chilton
12:15-12:30	Open Bioinformatics Foundation (OBF) Update	Hilmar Lapp
12:30-13:30	**Lunch**	(Lunch and poster
13:00-14:00	**Poster Session and Birds of a Feather (BOFs)**	session overlap)
14:00-15:30	***Session: Visualization***	Chair: Rob Davey
14:00-14:18	WormGUIDES: an Interactive Informatic Developmental Atlas at Subcellular Resolution	Anthony Santella
14:18-14:36	BioJS: an Open Source Standard for Biological Visualisation	Manuel Corpas
14:36-14:54	Biodalliance: a Fast, Extensible Genome Browser	Thomas Down
14:54-15:12	TGAC Browser: Visualisation Solutions for Big Data in the Genomic Era	Anil S. Thanki
15:12-15:30	Explore, Analyze, and Share Genomic Data Using Integrated Genome Browser	Ann Loraine
15:30-16:00	**Coffee Break**	
16:00-17:00	***Session: Project Updates***	Chair: Peter Cock
16:00-16:12	BioMart 0.9 – Introducing Tools for Data Analysis and Visualisation	Luca Pandini
16:12-16:24	Biocaml: The OCaml Bioinformatics Library	Ashish Agarwal
16:24-16:36	BioRuby and Distributed Development	Pjotr Prins
16:36-16:48	Biopython Project Update	Wibowo Arindrarto
16:48-17:00	Shared Bioinformatics Database Within Unipro UGENE	Ivan Protsyuk
17:00-17:30	***Session: Lightning Talks***	Chair: Peter Cock
17:00-17:05	Fostering the Next Generation of Data-driven Open Science with R	Karthik Ram
17:07-17:12	Tripal: an Open Source Toolkit for Building Genomic and Genetic Data Websites and Databases	Margaret Staton
17:14-17:19	PLUTo: Phyloinformatic Literature Unlocking Tools	Ross Mounce
17:21-17:26	A Publication Model that Aligns with the Key Open Source Software Principles	Michael L. Markie
17:27-17:30	Announcements	
17:30-18:30	**Birds of a Feather (BOFs)**	
19:00–	***Pay-your-own-way BOSC dinner***, The Asgard Irish Pub and Restaurant	Limited seats, note RSVP only

BOSC Schedule - Day Two - Saturday, 12 July 2014

Time	Title	Author/Notes
8:55-9:00	Announcements	
9:00-9:15	Codefest 2014 Report	Brad Chapman
9:15-10:15	**Keynote**: Biomedical Research as an Open Digital Enterprise	Philip Bourne
10:15-10:45	**Coffee Break**	
10:45-12:30	***Session: Software Interoperability***	Chair: Raoul Bonnal
10:45-11:03	Pathview: an R/Bioconductor Package for Pathway-based Data Integration and Visualization	Weijun Luo
11:03-11:21	Use of Semantically Annotated Resources in the Mobyle2 Web Framework	Hervé Ménager
11:21-11:39	Towards Ubiquitous OWL Computing: Simplifying Programmatic Authoring of and Querying with OWL Axioms	Hilmar Lapp
11:39-11:57	Integrating Taverna Player into Scratchpads	Robert Haines
11:57-12:15	Small Tools for Bioinformatics	Pjotr Prins
12:30-13:30	**Lunch**	(Lunch and poster
13:00-14:00	**Poster Session and Birds of a Feather (BOFs)**	session overlap)
14:00-15:30	***Session: Open Science and Reproducible Research***	Chair: Hilmar Lapp
14:00-14:18	SEEK for Science: A Data Management Platform which Supports Open and Reproducible Science	Carole Goble
14:18-14:36	Arvados: Achieving Computational Reproducibility and Data Provenance in Large-Scale Genomic Analyses	Brett Smith
14:36-14:54	Enhancing the Galaxy Experience through Community Involvement	Daniel Blankenberg
14:54-15:12	Supporting Dynamic Community Developed Biological Pipelines	Brad Chapman
15:12-15:30	Open as a Strategy for Durability, Reproducibility and Scalability	Jonathan Rees
15:30-16:00	**Coffee Break**	
16:00-17:00	**Panel**: Reproducibility: Rewards and Challenges (Panelists: Phil Bourne, C. Titus Brown, Varsha Khodiyar, Kaitlin Thaney)	Chair: Brad Chapman
17:00-17:10	Presentation of Student Travel Awards and Concluding Remarks	
17:10-18:00	**Birds of a Feather (BOFs)**	

Please refer to http://www.open-bio.org/wiki/BOSC_2014_Schedule for the latest schedule, including any changes since this document was prepared. Abstracts follow later in this document.

The abstracts are included in the order the talks will be given, followed by the poster-only abstracts.

List of Posters

#	Poster Title	Presenter	Talk Session
1	ADAM: Fast, Scalable Genomic Analysis	Frank A. Nothaft	Genome Scale
2	New Frontiers of Genome Assembly with SPAdes 3.1	Andrey Prjibelski	Genome Scale
3	SigSeeker: An Ensemble for Analysis of Epigenetic Data	Jens Lichtenberg	Genome Scale
4	WormGUIDES: an Interactive Informatic Developmental Atlas at Subcellular Resolution	Anthony Santella	Visualization
5	BioJS: an open source standard for biological visualisation	Manuel Corpas	Visualization
6	TGAC Browser: visualisation solutions for big data in the genomic era	Anil S. Thanki	Visualization
7	Explore, analyze, and share genomic data using Integrated Genome Browser	Ann Loraine	Visualization
7	Shared bioinformatics databases within Unipro UGENE	Ivan Protsyuk	Updates
8	A publication model that aligns with the key Open Source Software principles	Michael L Markie	Lightning
9	Pathview: an R/Bioconductor package for pathway-based data integration and visualization	Weijun Luo	Interop.
10	Integrating Taverna Player into Scratchpads	Robert Haines	Interop.
11	SEEK for Science: A Data Management Platform which Supports Open and Reproducible Science	Carole Goble	Open Science
12	*Withdrawn*	*Withdrawn*	
13	Connecting computational steps for NGS, and beyond.	Laurent Gautier	
14	Updates to MISO, the open-source NGS LIMS project	Xingdong Bian	
15	Running Taverna Workflows within IPython Notebook	Alan Williams	
16	Reconstruction of ancestral genomes in presence of gene gain and loss	Shuai Jiang	
17	*Withdrawn*	*Withdrawn*	
18	GEPETTO Update: An Open Source Framework for Gene Prioritization	Hoan Nguyen	
19	NeoPipe: An Open Source Framework for Protein sequence analysis	Hoan Nguyen	
20	MyGene.info updates: scalable gene-centric web services with user contributions	Chunlei Wu	
21	Aiding the journey from data to publication in the plant sciences	Robert Davey	
22	Bio2RDF mobile: an app for biological semantic web databases	Maxime Déraspe	
23	Tripal: an open source toolkit for building genomic and genetic data websites and databases	Margaret Staton	
24	BioBuilds: A Model for Long Term Sustainability of Open Source Bioinformatics	Chris Mueller	
25	GigaGalaxy: A GigaSolution for reproducible and sustainable genomic data publication and analysis	Scott Edmunds	

This talk is accompanied by poster #1.

ADAM: Fast, Scalable Genomic Analysis

Frank Austin Nothaft[1], *, Matt Massie[1], *, Timothy Danford[4], Carl Yeksigian[4], Arun Ahuja[5], Neal Sidhwaney[5], Jey Kottalam[1], Christopher Hartl[2], Christos Kozanitis[1], André Schumacher[3], Jeff Hammerbacher[5], Michael D. Linderman[5], Anthony D. Joseph[1], and David Patterson[1]

[1]Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA
[2]The Broad Institute of MIT and Harvard, Cambridge, MA
[3]International Computer Science Institute (ICSI), University of California, Berkeley, CA
[4]GenomeBridge, Cambridge, MA
[5]Icahn Institute for Genomics and Multiscale Biology, Icahn School of Medicine at Mount Sinai, New York, NY
*These authors contributed equally.

`http://www.bdgenomics.org`
`http://www.github.com/bigdatagenomics/adam`
Apache 2 License

Abstract

ADAM is a high-performance distributed processing pipeline and API for DNA sequencing data. To allow computation to scale on clusters with more than a hundred nodes, ADAM uses Apache Spark as a computational engine and stores data using Apache Avro and the open-source Parquet columnar store. This scalability allows us to perform complex, computationally heavy tasks such as base quality score recalibration (BQSR), or duplicate marking on high coverage human genomes ($> 60\times$, 236GB) in under a half hour. In tests on the Amazon Elastic Compute platform, we achieve a $50\times$ speedup over current processing pipelines, and a lower processing cost.

To achieve scalability in a distributed setting, we rephrased conventional sequential DNA processing algorithms as data-parallel algorithms. In this talk, we'll discuss the general principles we used for making these algorithms scalable while achieving full concordance with the equivalent serial algorithms. Additionally, by adapting genomic analysis to a commodity distributed analytics platform like Apache Spark, it is easier to perform ad hoc analysis and machine learning on genomic data. We will discuss how this impacts the clinical use of DNA analysis pipelines, as well as population genomics.

A framework for benchmarking RNA-seq pipelines

Rory Kirchner[1], Brad Chapman[1], Oliver Hofmann[1], Winston Hide[1]
[1]Harvard School of Public Health, Harvard University (kirchner@hsph.harvard.edu)

Project page: http://bcbio-nextgen.readthedocs.org
Code: https://github.com/chapmanb/bcbio-nextgen and https://github.com/roryk/bcbio.rnaseq

Processing RNA-seq data to make differential expression calls requires selection of tools for filtering contamination, aligning to the genome, quantifying expression and calling differentially expressed genes. Understanding the tradeoffs between choices of tools for each step requires both a reference pipeline to test against and a set of known differentially expressed genes to use to test for accuracy. We have created a community-developed framework named bcbio-nextgen for analyzing NGS data that is easy to modify, install, and scales to thousands of samples. We have implemented a reference RNA-seq pipeline using the bcbio-nextgen framework and used a combination of test data from the Sequencing Quality Control (SEQC) project, simulated count data and simulated read data to benchmark components of the RNA-seq pipeline. We demonstrate the usefulness of the RNA-seq benchmarks by determining the best quality filtering cutoff, evaluating the performance of differential expression callers and determining appropriate sample sizes for given experimental questions. The simulator can be tuned to reflect a wide variety of experiments and can also be tuned to match a user-defined real dataset. This allows experimenters to simultaneously analyze their experiment and produce benchmarks of the differential expression callers on data similar in nature to their experiment. We hope that the bcbio-nextgen framework will be a useful tool to both analyze RNA-seq experiments and rapidly test new or updated pipeline components or callers.

This talk is accompanied by poster #2.

New Frontiers of Genome Assembly with SPAdes 3.1

Andrey D. Prjibelski[1,5], Dmitry Antipov[1], Anton Bankevich[1], Alexey Gurevich[1], Sergey Nurk[1], Yana Safonova[1], Irina Vasilinetc[1], Anton Korobeynikov[1,2], Alla Lapidus[1,3] and Pavel Pevzner[1,4]

[1] Algorithmic Biology Laboratory, St. Petersburg Academic University, St. Petersburg, Russia
[2] Department of Mathematics and Mechanics, St. Petersburg State University, St. Petersburg, Russia
[3] Theodosius Dobzhansky Center for Genome Bioinformatics, St. Petersburg State University, St. Petersburg, Russia
[4] Department of Computer Science and Engineering, University of California, San Diego, USA
[5] E-mail address: ap@bioinf.spbau.ru

Project web site: http://bioinf.spbau.ru/en/spades
Source code available at: http://bioinf.spbau.ru/en/spades
Licence: GPLv2

Despite all the efforts high quality genome assembly is a complex task that so far remains unsolved. It is well known that majority of problems caused by repeats present in all genomes of any nature. The usage of multiple methods of genomic DNA isolation, different sequencing technologies and different types o genomic libraries for research projects introduces additional levels of complication to the genome assembly. The assembler tool SPAdes was originally developed at the St. Petersburg Academic University for the purpose of overcoming the complications associated with single-cell microbial dat (uneven coverage and increased level of chimerical reads). The tool was able to successfully resolve these issues for Illumina reads and was recognized by the scientific community as one of the best assemblers working with both isolates and single-cell data. Even though the assembler was specifically designed to work solely with microbial genomes, scientists have tested the tool on a large number o different types of other data.

Their efforts and feedback have inspired us to extend the capabilities of SPAdes to include additional platforms (Ion Torrent, PacBio, Sanger), combinations of platforms, and to work with both paired-end and mate-pair libraries of different insert sizes. In this work we present novel features of SPAdes 3.1: hybrid assemblies including the combination of Illumina/IonTorrent with PacBio (or other long read technologies), improved algorithms for scaffolding and repeat resolution, and an approach for mate-pai only assembly using new Illumina NexteraMP protocol.

We also have noticeably improved both BayesHammer and SPAdes performance. For example, new version of BayesHammer corrects data set of 100 Mbp diploid genome (25 Gb, 310M reads) in 16 hours instead of 90 (16 threads on server with Intel Xeon 2.27GHz processors). As to SPAdes, the main performance improvements were done in the exSPAnder repeat resolution module. On 60 Mbp repeat-rich genome repeat resolution step takes only 2 hours comparing to 78 hours for SPAdes 3.0.

SPAdes is openly available as source code and as pre-built Linux and Mac OS binaries. Additionally you can use SPAdes on such online cloud services as DNAnexus and Illumina BaseSpace.

This talk is accompanied by poster #3.

SigSeeker: An Ensemble for Analysis of Epigenetic Data

Jens Lichtenberg, Elisabeth F. Heuston, David M. Bodine
National Human Genome Research Institute, National Institutes of Health, Bethesda, MD, USA
Project and Sourcecode: http://sigseeker.org (GNU General Public License, version 3.0)

Epigenetics is the study of the proteins associated with (e.g. DNA binding proteins) and modifications to (e.g. DNA methylation) the primary DNA sequence. The epigenetic landscape of a nucleus determines which genes will be expressed or silenced and regulates the different genetic program of cells in an organism. There are numerous approaches for the analysis of next generation sequencing data that can be applied to define the epigenetic landscape. We believe that the quality of epigenetic analyses can be increased through an integrative framework that correlates the user-generated output of multiple prediction tools with existing biological data. We found that few sequence analysis tools integrate multiple approaches and consequently the existing techniques suffer in their prediction quality.

To address this problem we developed SigSeeker, a computational framework designed for: 1) the mapping of sequencing reads against a reference genome, 2) the detection of epigenetic mark enrichment within a set of mapped reads, and 3) the correlation of these sites with previously annotated expression and epigenetic data. By considering the complete set of established expression and epigenetic data during the analysis process, SigSeeker overcomes the shortcomings of other single technique approaches. SigSeeker incorporates commonly applied epigenetic tools using ensembles for each analysis stage. SigSeeker allows comparisons of user-generated data, as well as correlations of these data to publicly available epigenetic and expression data. The predictions made by each of the modules in the SigSeeker framework are evaluated for their statistical significance during the each stage of the analysis process as well as in a final report. SigSeeker is validated using benchmarks for ChIPSeq and HistoneSeq benchmark. These comparisons indicate that our ensemble technique exceeds single approaches (300% sensitivity increase) and is highly relevant for epigenetic data analysis.

We applied SigSeeker to study genome wide patterns of DNA methylation and gene expression in primary mouse blood cells. We found that DNA methylation was most abundant in the most primitive hematopoietic stem cells (HSC), declined in more differentiated common myeloid progenitors and declined further in nucleated red blood cells. In contrast to nucleated red blood cells, DNA methylation was increased in platelet forming cells to levels similar to HSC. The adjacent regulatory regions of the genes transcribing RNA were found to be hypomethylated in all cell types, while DNA methylation in the gene itself positively correlated with gene expression. In genes transcribing non-coding (regulatory) RNAs, DNA methylation in the gene itself was not found. We expanded our analysis to include the DNA binding proteins GATA1 and NFE2. GATA1 and NFE2 occupancy was cell-type specific. In platelet forming cells DNA methylation and NFE2 binding in the gene itself was associated with RNA-producing genes, while in nucleated red blood cell genes, DNA methylation and GATA1 binding in the gene itself was associated with inactive genes.

In summary, using our novel SigSeeker ensemble, we demonstrate that epigenetic modifications change dramatically during hematopoiesis and we have identified critical regions of the genome that regulate hematopoietic cell fate. Our ensemble technique exceeds single technique approaches in prediction quality and is ideal for identifying high confidence epigenetic profiles.

Galaxy as an Extensible Job Execution Platform
John Chilton and the Galaxy Team
jmchilton@bx.psu.edu
Project: http://galaxyproject.org/ Code: https://bitbucket.org/galaxy/galaxy-central

Galaxy is a popular, open source platform enabling data intensive biomedical research with a vibrant development community. The core Galaxy project is maintained by many developers and researchers and includes diverse components for visual analytics, data management, and app store-like functionality… the core Galaxy concepts of tools and jobs underpin much this functionality. A Galaxy tool typically corresponds to a command-line driven application to perform some analysis and describes both the user interface presented to Galaxy users as well as how to transform the users inputs into a command-line for execution. Galaxy execution of this command-line is called a job. Despite the simplicity of this concept, Galaxy jobs running architecture is constantly growing richer and more extensible.

Galaxy deployers can configure any number of static job destinations representing various clusters or servers and submission parameters for jobs - this will be demonstrated as well as easy plugins (simple Python functions) termed dynamic job destinations that can be used to route jobs to these destinations, modify existing destinations for an existing job, or create entirely new destinations all based on the user submitting the job, the job inputs, or runtime conditions.

These destinations can be the local server hosting Galaxy, a variety of cluster resource managers, or remote servers running a highly configurable, light-weight application (which itself can target the local machine or resource managers). Communication with these remote job execution components can be securely configured over HTTP(S) or via message queue, and provides a myriad of easy to configure options for job staging.

Continuing this theme, Galaxy tool developers can describe abstract packages that a tools depends on. This talk will cover building plugin for resolving these including a discussion of the existing plugins for Galaxy's custom environment files, Galaxy tool shed packages, and standard Unix environment Modules.

Finally, Galaxy jobs can be instrumented for metric collection - the talk will discuss developing these plugins the stock ones - including Galaxy's collectl plugin providing detailed resource usage statistics for jobs (CPU, memory, etc…) in a resource manager agnostic fashion.

The extensibility and ease of configuration described above may answer the question of why application developer should want to build on Galaxy as a platform. This talk will discuss the how as well - including embedding applications into Galaxy via tools, building applications on top of Galaxy via the API, and simply running jobs like Galaxy using the remote job execution component to leverage these plugins and extensibility without needing to run Galaxy itself.

This talk is accompanied by poster #4.

WormGUIDES: an Interactive Informatic Developmental Atlas at Subcellular Resolution

Anthony Santella[1], Daniel Colón Ramos[2], Hari Shroff[3], Zhirong Bao[1], William A. Mohler[4]

1 Developmental Biology Program, Sloan Kettering Institute, 1275 York Avenue, New York, New York
santella@mskcc.org
2 Dept. of Cell Biology, Yale University, New Haven, CT
3 NIBIB, NIH, Bethesda, MD.
4 Dept. of Genetics & Dev Biology & Ctr. For Cell Analysis and Modeling, UConn Health Ctr, Farmington, CT

Source code URLs:
http://www.wormguides.org/open-source-software
http://sourceforge.net/projects/starrynite/

The BRAIN initiative aims to achieve a systematic understanding of human brain structure and function. To achieve this goal, there is a need for software to support creation of large-scale neural atlases. Open-source efforts like CATMAID provide important tools for annotating and sharing large electron microscopy image sets. However, live multi-dimensional fluorescence image acquisition and the goal of mapping the developmental dynamics of the nervous system present unique challenges and opportunities in data sharing and visualization. WormGUIDES EmbryoAtlas is an interactive developmental atlas designed to fulfill this need to explore the emergence of neural structures. Ultimately it will combine cell position information and detailed neurite-growth tracking to present the emergence of neural wiring in the model organism *C. elegans*. Cell identities, as well as the spatio-temporal details of cell's morphogenesis and other developmental behaviors, will be explorable within an augmented reality-inspired interface combining three dimensional image data, geometric annotation and hypertextual information in a unified 3D interface. WormGUIDES integrates data generated by three of our other open source development efforts: 1) CytoSHOW, a web-deployed interface that allows metadata-enhanced exploration of large multi-D imaging data sets; 2) AceTree, an interface for visualizing cell positions and lineages, and 3) StarryNite, robust cell tracking software that generates the cell position and identity information contained in WormGUIDES. Together, these programs run to 218,000 lines of open source code, including a customized version of the open source ImageJ libraries.

WormGUIDES will be the first developmental atlas that captures the entirety of a metazoan embryogenesis: every cell's behavior at minute-level time resolution. The *C. elegans* connectome, currently the only complete "wiring diagram" of a nervous system, has been a prime model in neuroscience. WormGUIDES will fill in a critical gap by showing how this invariant connectome arises during embryogenesis ("the living connectome") and facilitating inspection of *in toto* cell biological dynamics during development. Currently within WormGUIDES, nuclei of embryonic cells are plotted within a freely rotatable, time-animated model of the developing embryo. Each nucleus can be queried by the user for its identity and for information about its ancestry, fate, function, and genome-activity pattern. Any scene can be instantly shared with collaborators or published as a URL. This functionality is currently available in Android and iOS apps. An expanded desktop version and support for detailed annotation of 3D cell morphology are forthcoming. Open source since their inception, we hope to expand these projects into a truly collaborative development effort, and we welcome new partners and contributors.

This talk is accompanied by poster #5.

BioJS: an open source standard for biological visualisation

Manuel Corpas[1], Rafael Jimenez[2], Seth J Carbon[3], Alex García[4], Leyla Garcia[2], Tatyana Goldberg[5], John Gomez[2], Alexis Kalderimis[6], Suzanna E Lewis[3], Ian Mulvany[7], Aleksandra Pawlik[8], Francis Rowland[2], Gustavo Salazar[9], Fabian Schreiber[2,10], Ian Sillitoe[11], William H Spooner[12], Anil Thanki[1], José M Villaveces[13], Guy Yachdav[5,14,15], Henning Hermjakob[2]

[1] The Genome Analysis Centre, Norwich Research Park, Norwich, NR4 7UH, UK
[2] European Bioinformatics Institute EMBL-EBI, Hinxton, CB10 1SD, UK
[3] Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
[4] School of Library and Information Science, Florida State University, Tallahassee, FL, USA
[5] TUM, Department of Informatics, Bioinformatics & Computational Biology, 5748 Garching/ Munich, Germany
[6] Department of Genetics and Cambridge Systems Biology Centre, Cambridge University, Cambridge, CB2 3EH, UK
[7] eLife, Cambridge, CB2 1JP, UK
[8] Faculty of Mathematics, Computing and Technology, Open University, UK, Milton Keynes, MK7 6AA, UK
[9] Computational Biology Group, University of Cape Town, Cape Town, South Africa
[10] The Wellcome Trust Sanger Institute, Hinxton, Cambridge, CB10 1SD, UK
[11] Biomolecular Structure and Modelling Group Department of Biochemistry, University College London, London, UK
[12] Eagle Genomics Ltd, Cambridge, CB22 3AT, UK
[13] Max Planck Institute of Biochemistry, Am Klopferspitz 18, 82152, Germany
[14] TUM Graduate School of Information Science in Health (GSISH), 85748 Garching/Munich, Germany
[15] Biosof LLC, New York, NY, 10001, USA

**Corresponding authors*
manuel.corpas@tgac.ac.uk
hhe@ebi.ac.uk

ABSTRACT

BioJS is a community-based standard and repository of functional components to represent biological information on the web. The development of BioJS has been prompted by the growing need for bioinformatics visualisation tools to be easily shared, reused and discovered. Its modular architecture makes it easy for users to find a specific functionality without needing to know how it has been built, while components can be extended or created for implementing new functionality. The BioJS community of developers currently provides a range of functionality that is open access and freely available. A registry has been set up that categorises and provides installation instructions and testing facilities at http://www.ebi.ac.uk/tools/biojs/ The source code for all components is available for ready use at https://github.com/biojs/biojs

Biodalliance: a fast, extensible, genome browser

Thomas A. Down*[1,2] and Tim J. P. Hubbard[2,1]
[1] Wellcome Trust Sanger Institute, Cambridge, UK
[2] Department of Medical & Molecular Genetics, Kings College London, UK

* E-mail: thomas@biodalliance.org
Project website: http://www.biodalliance.org/
Source download: http://github.com/dasmoth/dalliance
License: BSD

Genome browsers are a vital part of the genomics workflow. Inspection of annotations and experimental data have proved indispensable for spotting unexpected correlations, formulating new hypotheses, or simply sanity-checking new data sets. The sequencing revolution has made this even more important. Today, even small labs are routinely applying techniques like ChIP-seq, RNA-seq, or genome sequencing, often with limited bioinformatic support. To keep pace with these trends we need tools that make integrating and visualisation large datasets as effortless as possible.

Dalliance [1] is a genome browser which makes aggressive use of modern web technologies to offer a high level of interactivity and powerful navigation and exploration tools while running within a normal web browser. The display can be freely scrolled and zoomed with mouse gestures and keyboard controls. Shortcuts for navigation between features allows sparse datasets to be rapidly explored. We have adopted a fully distributed approach, with no backend server between the datasets and the web-application client. Data can be accessed directly from a number of standard indexed binary file formats, such as BigWig, BigBed, BAM, and VCF. Direct file access makes integrating new genomic datasets, particularly the results of high-throughput sequencing experiments, very quick, and accessible to occasional bioinformaticians who often have limited sysadmin experience – and limited enthusiasm for installing extra server software. It also allows instant access to datasets in this format from remote web servers without time-consuming downloading (e.g. ENCODE datasets from EBI). Unusually for a web-based application, Dalliance also allows viewing of data directly from local disk on your own machine, without any helper processes or servers.

Recent developments include support for more file formats (VCF, non-binary BED and WIG), support for UCSC-style “track hub” metadata, and hooks to allow user code to interact with the displays and control the integration of data from multiple sources.

[1] Down TA, Piipari M, Hubbard TJ. *Dalliance: interactive genome viewing on the web* Bioinformatics (2011) 27:889-890

TGAC Browser: visualisation solutions for big data in the genomic era

Anil S. Thanki[1], Xingdong Bian[1], Robert P. Davey[1]

1. The Genome Analysis Centre, Norwich Research Park, Norwich NR4 7UH, UK

Email: Anil.Thanki@tgac.ac.uk

Demo: http://tgac-browser.tgac.ac.uk

Source Code: https://github.com/tgac/tgacbrowser

License: GPL v3

We present the TGAC Browser with novel rendering, annotation and analysis capabilities designed to overcome the shortcomings in available approaches. TGAC Browser, being a web-based client, utilises JavaScript libraries to provide a fast and intuitive genome browsing experience. We focus on harnessing Internet architectures as well as localised HPC hardware, concentrating on improved, more productive interfaces and analytical capabilities.

- **User-friendly:** Live data searching, track modification, and drag and drop selection; actions that are seamlessly powered by modern web browsers
- **Responsiveness:** Client-side rendering and caching, based on JSON fragments generated by server logic, helps decrease the server load and improves user experience
    - TGAC Browser visualises genomic data in different ways, based on the type and amount of data, which is more informative to the user and memory efficient.
- **Analysis Integration:** The ability to carry out heavyweight analysis tasks, using tools such as BLAST, via a dedicated extensible daemon
- **Annotation:** Users can edit annotations which can be persisted on the server, reloaded, and shared at a later date
- **Off-the-shelf Installation:** The only prerequisites are a web application container, such as Jetty or Tomcat, and a standard Ensembl database to host sequence features
- **Extensible**: Adaptable modular design to enable interfacing with other databases, e.g. GMOD
- **Data format**: TGAC Browser processes and visualises data directly from the Ensembl core schema as well as next-generation sequencing (NGS) data output, i.e. BAM/SAM, BigWig/wig, GFF, and VCF.

This talk is accompanied by poster #6.

Explore, analyze, and share genomic data using Integrated Genome Browser

Ann E. Loraine, David C. Norris, Tarun Kanaparthi, Hiral Vora, Ivory E. Clabaugh, Alyssa A. Gulledge, Kyle Suttlemyre
University of North Carolina at Charlotte, North Carolina Research Campus, Kannapolis,NC
contact: aloraine@uncc.edu

To benefit from high-throughput sequencing techniques, scientists need easy-to-use visual analytics software tools that support all aspects of the scientific process. To meet this need, we added new visualization capabilities to Integrated Genome Browser, a fast, flexible and free Java-based desktop software tool originally developed at Affymetrix. IGB is available from http://www.bioviz.org and https://bitbucket.org/lorainelab/integrated-genome-browser. IGB supports a wide range of interactions with data, from simple counting by selection to complex filtering operations that can highlight biologically meaningful aspects of data. With collaborators from Genentech, we re-architected IGB to use an OSGi-based framework that enables rapid addition of new visualization components and features via OSGi bundles, called plug-ins. Using this new architecture, we implemented new visual analytics tools, including a tool to quantify and visualize splicing support (FindJunctions), new linkouts to run blast searches at NCBI, a heatmap editor that enables color-coding features by score, and many others. For some new features, such as the heatmap editor and blast searches, we imported code from Cytoscape and Apollo, two other well-established Java-based open source projects. To enable data sharing, IGB implements a simple Web-based system called IGBQuickLoad. A new release of the blueberry genome, annotations, and fruit development RNA-Seq data set highlights the possibilities. IGB also supports ReST-style bookmark URLs that enable integration with sophisticated data sharing and analysis environments, such as Galaxy and GenomeSpace. IGB is released under the Common Public License, v1.0.

BioMart 0.9 – introducing tools for data analysis and visualisation.

Luca Pandini[1], Paolo Provero[1], Davide Cittaro[1], Jose Manuel Garcia Manteiba[1], Michela Riba[1], Arek Kasprzyk[1], Elia Stupka[1]

1. Center for Translational Genomics and Bioinformatics San Raffaele Scientific Institute. Milan, Italy.

The BioMart project provides free software and data services to the international scientific community in order to foster scientific collaboration and facilitate the scientific discovery process. The project adheres to the open source philosophy that promotes collaboration and code reuse.

The latest version of BioMart includes support for data analysis and visualisation tools exploiting the richness of the existing BioMart data sources and its query engine capabilities. The first of the BioMart tools has already been implemented and is accessible from BioMart Central Portal. This tool enables enrichment analysis of genes in all 66 species included in the latest Ensembl release and a broad range of gene identifiers for each species are also available. Furthermore, the tool supports cross-species analysis using Ensembl homology data. For instance, it is possible to perform a one step enrichment analysis against human disease dataset using experimental data from any of the species for which human homology data is available. Finally, the enrichment tool facilitates analysis of BED files containing genomic features such as Copy Number Variations (CNVs) or Differentially Methylated Regions (DMRs).

The BioMart tools offer different type of access tailored to different groups of users. For biologists, the BioMart tools offer an interactive and customisable web-based graphical user interface. For bioinformaticians, the BioMart tools provide programmatic access through a range of Application Programing Interfaces (APIs) such as REST, SOAP, SPARQL and JAVA. For service providers, the BioMart tools offer a highly customizable system that can be installed locally and tailored to support new types of data analysis. The BioMart tools are built as an extensible library that facilitates the implementation of new functionality and algorithms, while the backend can be extended with customized datasets.

Project URL http://www.biomart.org
BioMart Central Portal http://central.biomart.org
Software https://github.com/biomart

Biocaml: The OCaml Bioinformatics Library

Ashish Agarwal[1], Sebastien Mondet[2], Philippe Veber[4], Christophe Troestler[3]
[1]Solvuu, <agarwal1975@gmail.com>, [2]Mount Sinai, [3]Université de Mons
[4]Laboratoire de Biométrie et Biologie Evolutive, UMR 5558, CNRS, Université Lyon 1
Project Website: http://biocaml.org. *Source Code:* https://github.com/biocaml. *License:* LGPL.

Functional Programming has long served academic studies of software engineering and is now gaining traction in industrial settings. OCaml is a Functional Programming language that also has strong support for traditional imperative and object-oriented programming. Biocaml is OCaml's bioinformatics library, analogous to BioPerl, BioPython, etc, and has been used in several large genomics projects. Its current feature set can be split into 3 broad categories: i) parsing/printing of many data formats, ii) data structures on integer intervals, and iii) clients to public data repositories. Cross-cutting features include: asynchronous methods for concurrency, rigorous error handling, and comprehensive API documentation.

i) Parsers and printers for about 15 file formats are currently implemented, including: fastq, fasta, gff, sam, and bam. All can be zipped, and all support streaming. A format specification system, inspired by MIME types but more flexible, allows precise handling of all the minor variations in formats. ii) Integer intervals arise frequently in genomics. Biocaml provides sets and balanced binary trees with several variations, e.g. overlapping intervals are allowed or not, and nodes of a tree carry values or only the leaves. These serve as the basis for several other modules: memory efficient integer sets, ROC curve computations, gene model representations, and histograms. iii) Bioinformaticians rely on a multitude of public data repositories. Biocaml provides a client for several of the Entrez databases, and we hope to continue adding more.

Genomics algorithms are often IO intensive and access to remote data sources can lead to long delays. Biocaml provides an asynchronous API from the ground up; every system call is made in a non-blocking fashion. This can speed up your programs by allowing some computations to continue while other parts are waiting for a system call to return. Such concurrency handling is normally done by using an event loop or with callbacks, but Biocaml employs a concurrency monad, which allows this kind of code to be written more safely and more easily. System calls and many other computations are prone to errors. Biocaml takes error handling very seriously, again by using a monad. Sometimes, the majority of a function's implementation regards error handling.

We aim to accommodate a spectrum of programmers: from script writers to software engineers building industrial strength applications. Thus, we do not consider it reasonable to impose monadic programming on all of our users. For system calls, a simpler but blocking API is also provided. Fortunately, the burden on Biocaml's developers is minimized due to OCaml's functors, which provides the alternate API for free. For errors, an alternate exception-ful API is provided, which, although not free, is not difficult to provide. Combined with our focus on documentation, we hope all of these features will allow Biocaml to be widely used.

BioRuby and distributed development

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

Bioinformatics Open Source Conference (BOSC) 2014

Affiliations: The BioRuby Project
Contact E-mail: bioruby@lists.open-bio.org
Author E-mail: j.c.p.prins@umcutrecht.nl
URL: http://biogems.info/
Source code: Linked from biogems.info
License: All licenses are of type approved by the Free Software Foundation (FSF)

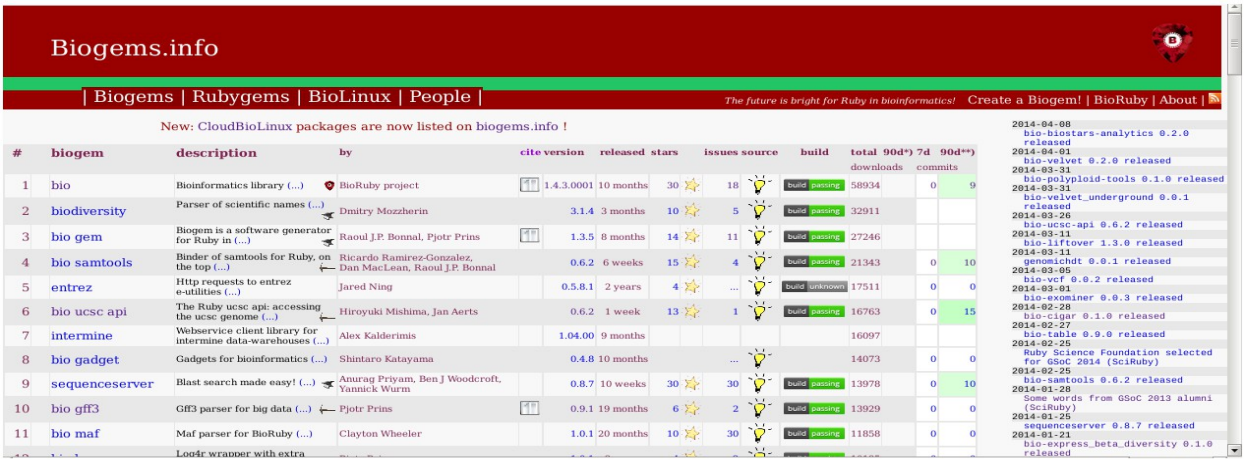
With the distributed development of biogems, listed on http://biogems.info/, the BioRuby community is one of the most active Open Bioinformatics Foundation (OBF) communities in terms of number of projects, git commits and library downloads.

In this talk we will quantify and visualize what it means to allow contributing independent small modules and tools to the bioinformatics community. Not only has the decision to distribute development led to a larger group of regular contributors, which puts less strain on the core maintainers, but it also led to a number of useful command line tools that have been introduced, such as bioruby-table, bioruby-samtools, bioruby-ngs and bioruby-vcf. The code generators that we use fast-track biogem development and drive the rapid creation of new software modules with support for command line interfaces, unit testing, and the Travis continuous integration service.

In addition we will discuss how the BioRuby community is making great strides in leveraging web technologies for biomedical data analysis using web services and linked data. This opens heterogeneous data sources and facilitates easier data integration in bioinformatic analysis pipelines.

We aim to expand tracking other Bio* projects in http://biogems.info/ and make it a comprehensive resource for the OBF. The BOSC meeting will allow us to fine tune such ideas and gather feedback from the bioinformatics community.

Figure 1: Snapshot of the biogems.info website which contains relevant information about BioRuby related modules



*) Department of Medical Genetics, Institute for Molecular Medicine, University Medical Center Utrecht, The Netherlands

Biopython Project Update 2014

Wibowo Arindrarto[*], Peter Cock[†], Eric Talevich[‡], Michiel de Hoon[§], Tiago Antao [¶], João Rodrigues[‖], and the Biopython Contributors

15th Bioinformatics Open Source Conference (BOSC) 2014, Boston, MA, USA

Website: http://biopython.org
Repository: https://github.com/biopython/biopython
License: Biopython License Agreement (MIT style, see http://www.biopython.org/DIST/LICENSE)

We present the latest updates from the Biopython project, a long-running, distributed collaboration producing a freely available Python library for biological computation [1]. Biopython is supported by the Open Bioinformatics Foundation (OBF).

Since BOSC 2013 there have been three Biopython releases: version 1.62, 1.63, and 1.64. New features in version 1.62 include parsing support for NeXML and CDAO in the Bio.Phylo module, parsing support for GAF, GPA, and GPI formats from UniProt-GOA in the Bio.UniPort module, and BioSQL support for Jython. In version 1.63, we added support for the population genetic tool fastsimcoal, a wrapper for samtools, and other significant enhancements to existing modules. Version 1.64 saw the addition of the Bio.CodonAlign module and enhancements to the Bio.Phylo module, contributed by our Google Summer of Code (GSoC) 2013 students. The upcoming version 1.65 is now under development. Moreover, since BOSC 2013 we have successfully supported Python 2, Python 3, PyPy, and Jython 2.7 with a single codebase. This change is also reflected in our Tutorial & Cookbook, which uses code compatible with all Python versions.

In addition to local installation on various operating systems, Biopython is now available at http://toolshed.g2.bx.psu.edu/view/biopython/ in the Galaxy Tool Shed [2] as a package dependency. Galaxy tools requiring Biopython can now specify this dependency explicitly and choose from Biopython version 1.61 onwards.

We participated in GSoC 2013 under the umbrella of the National Evolutionary Synthesis Center (NESCent) and selected two students to work on Biopython: Yanbo Ye extended the phylogenetics module Bio.Phylo with features for tree construction and analysis, and Zheng Ruan developed the new module Bio.CodonAlign for codon alignment and analysis support. Both projects were successfully integrated in Biopython are included in the latest Biopython release. We continue with GSoC 2014 under the OBF umbrella, with our student Evan Parker working to add lazy-parsing support to Bio.SeqIO.

References

[1] Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., de Hoon, M.J. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**(11) 1422-3. doi:10.1093/bioinformatics/btp163

[2] Blankenberg, D., Von Kuster, G., Bouvier, E., Baker, D., Afgan, E., Stoler N., The Galaxy Team, Taylor, J., Nekrutenko, A. (2014) Dissemination of scientific software with Galaxy ToolShed. *Genome Biology* **15** 403. doi:10.1186/gb4161

[*]Sequencing Analysis Support Core, Leiden University Medical Center, Leiden, NL. Email: bow@bow.web.id
[†]Information and Computational Sciences, James Hutton Institute (formerly SCRI), Invergowrie, Dundee, UK
[‡]Department of Dermatology, University of California San Francisco, San Francisco, CA, USA
[§]Division of Genomic Technologies, RIKEN Center for Life Science Technologies, Yokohama, JP
[¶]Vector Biology Department, Liverpool School of Tropical Medicine, Pembroke Place, UK
[‖]Bijvoet Center for Biomolecular Research, Utrecht University, Utrecht, NL

This talk is accompanied by poster #7.

Shared bioinformatics database within Unipro UGENE

Ivan Protsyuk[1], Mikhail Fursov[2]
iprotsyuk@unipro.ru, mfursov@unipro.ru
[1]Novosibirsk State University, Novosibirsk, Russia
[2]Center of Information Technologies "UniPro", Novosibirsk, Russia

Unipro UGENE [Okonechnikov et al. 2012] is an open-source bioinformatics toolkit that integrates popular tools along with original instruments for molecular biologists within a unified user interface. It has grown to a platform that can cope with a large variety of tasks including multiple alignment, phylogeny, functional annotation, *in silico* cloning, protein structure analysis and TFBSs recognition. To perform various types of analysis UGENE gathers computational algorithms, visualization capabilities and advanced workflows.

Nowadays most bioinformatics desktop applications, including UGENE, make use of local data models when processing different types of data. Namely, a user can work with files stored on his/her local machine only. Such an approach may cause inconvenience to scientists working cooperatively and relying on the same data. The most obvious issue arises from the need to make multiple copies of certain resources for every workplace. After that the problem of synchronization comes into play. There are tools that provide the needed capabilities, for instance, CLC Bioinformatics Database or Geneious but they are proprietary and quite expensive.

Recently we focused on delivering collaborative work into the UGENE user experience. Currently several UGENE installations can be connected to a designated shared database and use it simultaneously as if it was an ordinary file possibly containing a huge amount of data. Objects of each data type supported by UGENE such as sequences, annotations and multiple alignments can now be easily imported from or exported to remote storage. The storage itself is a regular database server (so far only MySQL is supported) available for external connections so even an inexpert user can deploy it. All its data are stored using a fully relational approach within a self-developed database schema. Thus, UGENE is able to provide access to shared data for a few users located, for example, in the same lab or institution.

Furthermore, UGENE maintains integrity with its visualization capabilities as well. Not only can data be transferred via the database between computers, but it also may be displayed and modified permanently just as if it were located in a normal file. Thereby smooth transition is achieved for the end-user between his local storage and the shared one.

This work itself presents a basis for further development in various directions. First, UGENE Workflow Designer coupled with a shared database may operate remotely and do jobs for clients aiming to process the shared data. Therefore, this opens an opportunity for using easy-to-install computational clusters requiring the UGENE suite only. Another feature is the saving of full user contexts between successive UGENE launches to an embedded local database. That implies tracking and permanent storing of states for all the open files, views and databases involved in a user session. In this way, UGENE may improve the convenience and productivity of a biologist's workplace even more.

The first version of the UGENE tool, supporting shared databases, is going to be released by the end of June 2014.

Project Web Site: http://ugene.unipro.ru/
Software and source code: http://ugene.unipro.ru/download.html

BOSC 2014 – *Five Minute Lightning Talks*
Day One, 11 July 2014, 17:00 – 17:30
Session chair: Peter Cock

Fostering the next generation of data-driven open science with R

Karthik Ram [1,2]

1. The rOpenSci Project
2. Berkeley Initiative in Global Change Biology, University of California, Berkeley.

karthik.ram@gmail.com

Abstract

Research is becoming increasingly data intensive and computation driven across various scientific domains, from the social and life sciences all the way to particle physics. Many new scientific insights will likely emerge from vast stores of existing data, rather than from new data collection efforts. In addition, funder and journal mandates now require that researchers share at least their final datasets at the time of publication. As a result of such changes, researchers not only need to continue maintaining their domain expertize, but also be proficient in skills necessary to acquire, manipulate, document and share their data.

rOpenSci is a community driven effort to foster such data driven science among research communities that use R. Our suite of tools (http://ropensci.org/packages/) allow access to these data repositories through a statistical programming environment that is already a familiar part of the workflow of many scientists. Our tools not only facilitate drawing data into an environment where it can readily be manipulated and visualized, but also one in which those analyses and methods can be easily shared, replicated, and extended by other researchers. In this poster and accompanying live demo I highlight some our our recent efforts in advancing open and transparent practices in the sciences.

Tripal: an open source toolkit for building genomic and genetic data websites and databases

Stephen Ficklin[1], Lacey Anne Sanderson[2], Margaret Staton[3]*, Chun Huai Cheng[1], Sook Jung[1], Kirstin Bett[2], Doreen Main[1]

1 Department of Horticulture, Washington State University
2 College of Agriculture and Bioresources, University of Saskatchewan
3 Department of Entomology and Plant Pathology, University of Tennessee
* mstaton1@utk.edu

URL:	http://tripal.info/
URL (code):	https://drupal.org/project/1337878/git-instructions
License:	GNU General Public License, Version 2

Community genomic databases fulfill a critical need by offering curated and mission-specific information to targeted audiences with shared basic and applied research goals. Tripal was created in response to the need for an open source, extensible software system to support and standardize the website development projects of diverse scientific communities with growing sequence resources and uniquely interlinked datasets. Tripal provides data pages and search tools for taxonomies, genomic sequences, genetic markers and maps, stocks, cultivars, DNA or clone libraries, and publications stored in Chado[1], the standard relational database schema for biological information. Tripal is written in PHP and provides an Application Programmers Interface (API) that allows other developers to extend the core modules and create new modules. As a result, extension modules have been developed for parsing and uploading data from common computational analyses of sequence data such as BLAST[2], InterProScan[3], KEGG[4], and blast2GO[5]. Tripal was developed for use with the popular open-source content management system Drupal[6], a PHP-based platform used to power millions of websites and applications worldwide. Drupal empowers non-technical users to easily add content and functionality without the need for programming, including writing news articles, announcements or blogs as well as designing the layout and content of new pages. This robust underlying structure is leveraged in Tripal to allow curators or community members to upload different types of data through an intuitive web interface and to create new ways of visualizing heterogeneous datasets. Drupal also provides built-in user management and content versioning, an ideal base for constructing robust community annotation capabilities. By bridging Chado and Drupal, Tripal marries the power of a biological data storage schema with a web development platform to decrease the cost and time associated with development of genomic, genetic and breeding databases for diverse biological research communities. Tripal is in use or being implemented by at least 24 different databases including the Genome Database for Rosaceae[7], CottonGen[8], the Hardwoods Genomic Database[9], and KnowPulse[10]. Future development is focusing on cross-site communication, adoption of community driven data standards, and integrated "big data" analysis.

1. http://gmod.org/wiki/Chado_-_Getting_Started
2. http://blast.ncbi.nlm.nih.gov/Blast.cgi
3. http://www.ebi.ac.uk/Tools/pfa/iprscan
4. http://www.genome.jp/kegg
5. http://www.blast2go.com
6. https://drupal.org
7. http://www.rosaceae.org
8. http://www.cottongen.org
9. http://www.hardwoodgenomics.org
10. http://knowpulse.usask.ca

PLUTo: Phyloinformatic Literature Unlocking Tools

Ross Mounce [1], Peter Murray-Rust [2], Matthew Wills [1]

[1] Department of Biology & Biochemistry, University of Bath, England, United Kingdom

[2] Shuttleworth Foundation Fellow

Contact Email: ross.mounce@gmail.com
Homepage: https://bitbucket.org/petermr/pluto/wiki/Home/
Repositories: https://bitbucket.org/petermr/crawlerrepo/
https://bitbucket.org/petermr/svg2xml-dev
https://bitbucket.org/petermr/imageanalysis
https://bitbucket.org/petermr/svgbuilder
https://bitbucket.org/petermr/xhtml2stm-dev/

Full Abstract

Approximately 4% of published phylogenetic analyses make their underlying data & results publicly available in an immediately re-usable, machine-readable form [1]. Furthermore, if one makes the effort to email the authors for the underlying data; only 16% of such requests are successful [2]. Phylogenetic data can be and *is* re-used in a multitude of different ways by other projects subsequent to the publication of the original analysis – it is well understood, valuable, and eminently re-usable data. This BBSRC-funded PLUTo project aims to develop software tools which can extract phylogenetic data directly from the PDFs in which these data are often siloed. Why PDF? Many biodiversity journals do not provide XML and are not deposited in PMC, indeed some like those published by Magnolia Press (e.g. *Zootaxa*, *Phytotaxa*) are *only* made available as PDF. Therefore tools that handle PDFs are needed if we are to discover, reclaim & re-use all the phylogenetic data that is otherwise buried in the literature. This talk will also discuss helpful changes to UK copyright law (to be introduced sometime this year) which will legally enable and empower this type of data discovery, liberation & re-use for 'non-commercial'[3] research purposes.

References

[1] Stoltzfus, A., O'Meara, B., Whitacre, J., Mounce, R., Gillespie, E., Kumar, S., Rosauer, D., and Vos, R. 2012. Sharing and re-use of phylogenetic trees (and associated data) to facilitate synthesis. *BMC Research Notes* http://dx.doi.org/10.1186/1756-0500-5-574
[2] Drew, B. T., Gazis, R., Cabezas, P., Swithers, K. S., Deng, J., Rodriguez, R., Katz, L. A., Crandall, K. A., Hibbett, D. S., and Soltis, D. E. 2013. Lost branches on the tree of life. *PLoS Biology* http://dx.doi.org/10.1371/journal.pbio.1001636
[3] Klimpel, P. 2013. Consequences, Risks, and side-effects of the license module Non-Commercial – NC [English translation] http://openglam.org/files/2013/01/iRights_CC-NC_Guide_English.pdf

This talk is accompanied by poster #8.

Author: Michael Markie (F1000 Research)

A publication model that aligns with the key Open Source Software principles

In recent years, software development has had a significant impact on scientific research and continues to play a major role in facilitating advances with the life sciences in particular. Building code using open repositories such as GitHub allows it to be continually improved both during the development phase and after the software has been more widely disseminated. However, the long term availability of code is important in order to be reproducible, and to enable future scientific research which may require further modification of existing code[1]. Documentation of code for scholarly purposes usually takes the form of a publication in a peer reviewed article. This allows the developer to provide context around their code for both fellow programmers and non-computational users. A published paper also contributes to the developer's formal academic output but also helps foster vibrant collaborative communities that help nurture and spread new ideas as well as reinforcing the quality of the code that is produced.

Releasing information in incremental steps is nothing new to software developers, who regularly release updates and patches that add new functionality to existing programmes. The launch of a new bioinformatics tool is often accompanied by a paper describing the software for new users. However, the paper describing the tool will be out-of-date as soon as a new software update is released but the changes are often not significant enough to warrant a whole new paper, and thus the most recent developments go undocumented for a sustained period of time. Trying to publish such dynamic information in traditional 'static' journals is much like fitting a square peg in a round hole.

The *F1000Research* (http://f1000research.com/) publishing model is much more in synch with the way software is developed. Each software tool published can be updated at any time as a new version (clearly linked to the original and previous versions of the article) allowing any new code, tweaks and features to be documented with relative ease. Furthermore, *F1000Research* ensures that all the code and related data are freely available from the paper. A usable copy of the code as it was at the time of publication remains available, with the code being forked into an archival *F1000Research* space within the same repository used by the authors. A copy of the code as at the time of publication is also assigned a persistent identifier to eliminate any ambiguity about the code that is described in the article. Additionally, *F1000Research* ensures the paper includes a link to the author's own working repository, so that readers can easily navigate to the latest version of the source code. By taking these measures, users are able to establish the provenance of the code and reuse it easily, hence supporting the reproducibility of the software, which ultimately contributes to making the software more robust. *F1000Research* also uses open peer review, providing an additional layer of validation for published software articles. Experts from the scientific community are invited to constructively critique the software and lay the foundations for any improvements. Having these reviews, together with any user comments, open to everyone helps to mirror the collaborative approach encouraged by open source initiatives and embraces the open source 'community' ethos.

By aligning with the requirements of publishing software, *F1000Research* has started to encourage computational science software developers to create an *F1000Research* Article Collection[2] to augment their open source software projects. In February 2014, we launched the BioJS Collection[3] which comprises individual software components, each of which are like a standard Lego-like pieces for building web applications that visualise biological data[4]. With this poster, we will discuss the novel requirements associated specifically with the needs of articles associated with open source software development, and discuss new publishing opportunities that better reflect and support those needs for the benefit of both software developers and scientific researchers as a whole.

1. Prlić A, Procter JB (2012) Ten Simple Rules for the Open Development of Scientific Software. PLoS Comput Biol 8(12): e1002802.
2. Markie ML (2014) F1000Research Article Collections (http://blog.f1000research.com/2014/02/13/f1000research-article-collections/)
3. BioJS Collection (2014) *F1000Research* doi/10.12688/f1000research.collections.2
4. Markie ML (2014) BioJS – visualising biological data: an interview with Manuel Corpas (http://blog.f1000research.com/2014/02/18/biojs-visualising-biological-data-an-interview-with-manuel-corpas/)

This talk is accompanied by poster #9.

Pathview: an R/Bioconductor package for pathway-based data integration and visualization

Weijun Luo[1,2*] and Cory Brouwer [1,2]
[1]Department of Bioinformatics and Genomics, UNC Charlotte, Charlotte, NC 28223
[2]UNC Charlotte Department of Bioinformatics and Genomics, North Carolina Research Campus, Kannapolis, NC 28081
[*]Correspondence: luo_weijun@yahoo.com
Project: http://pathview.r-forge.r-project.org/
BioC release (doc, code) http://bioconductor.org/packages/release/bioc/html/pathview.html
License: GPL (>=3.0)

Pathview is a novel tool set that maps, integrates and renders a large variety of biological data on pathways, and produces interpretable graphs with publication quality [1].

Pathview generates both native KEGG view and Graphviz view for pathways. KEGG view keeps all the pathway meta-data, including reaction and signaling contexts important for human reading and interpretation. Graphviz view provides better control of node and edge attributes, better view of pathway topology and analysis statistics.

Pathview provides strong support for data integration. It works with: 1) essentially all types of biological data mappable to pathways, 2) over 10 types of gene or protein IDs, and 20 types of compound or metabolite IDs, 3) pathways for over 2000 species as well as KEGG Orthology, 4) various data attributes and formats, i.e. continuous/discrete data, matrices/vectors, single/multiple samples or time-series etc.

Pathview is open source, fully automated and error-resistant. Although built as a stand-alone program, Pathview may seamlessly integrate with pathway and functional analysis tools for large-scale and fully automated analysis pipelines.

Pathview has been published with Bioinformatics [1], and ranked as a most-read among ALL Bioinformatics papers in 5 consecutive months (June-October, 2013). The software has been widely adopted by scientists worldwide and has downloaded over 6500 times within 12 months: http://bioconductor.org/packages/stats/bioc/pathview.html. In fact, it is a most used Bioconductor package released in 2013. Pathview has received hundreds of user inquiries or recommendations through emails, major mail-lists and bioinformatics forums, such as Bioconductor help list, seqanswers.com and biostars.org etc. All impact statistics available upon request.

1. Luo W, Brouwer C: **Pathview: an R/Bioconductor package for pathway-based data integration and visualization**. *Bioinformatics* 2013, **29**(14):1830-1831.

Use of semantically annotated resources in the Mobyle2 Web Framework

Hervé Ménager[1,#], Bertrand Néron[1], Olivia Doppelt-Azeroual[1], Olivier Sallou[2]
[1] Centre d'Informatique pour la Biologie, Institut Pasteur, Paris, France
[2] IRISA, Rennes, France
[#] Presenting author email : hmenager@pasteur.fr

Mobyle website : https://github.com/mobyle2
Mobyle license : BSD

Mobyle 2 is a project currently under development at the Institut Pasteur and the GenOuest platform. The Mobyle framework is a web-based workbench for bioinformatics analyses. Its interface allows scientists, without installing anything locally, to use command line-based bioinformatics tools to perform analyses on remote computing resources. The high level of integration between the different tools provided enables and guides users in the construction of potentially complex protocols, chaining interactively successive tasks in an exploratory mode, or automating their execution with workflows. Mobyle 2 is a major rewrite which adds new features such as collaborative work, secure data sharing, a REST API and the use of an ontology-based annotation mechanism. This presentation will focus on this last feature.

The current integration mechanisms of Mobyle are based on a custom vocabulary which is used to annotate services, biological data banks available in Mobyle, as well as the user data and workflows. These annotations, linking the actual resources to the concepts they represent, provide an integration layer used to guide the user by selecting and connecting semantically and/or syntactically compatible resources.

The upcoming version of Mobyle replaces this vocabulary with an ontology. The richness and focus of the EDAM[1] ontology allows for a more precise and consistent description of the resources that are integrated in Mobyle (command line programs, workflows, web services, interactive widgets), linked with external applications or shared through MobyleNet. Additionally, the use of this ontology as an abstraction layer for the classification of equivalent resources enables the development of a number of user-targeted features such as:

- the automatic suggestion of equivalent services, where a user can select alternative tools that perform the same task using a different method,
- the definition of abstract or semi-abstract workflows, where users can choose at runtime which specific tools will perform some or all of the tasks of a workflow,
- the automatic handling of an increased number of implicit tasks, improving the existing format detection and conversion mechanisms and adding new mechanisms such as implicit iterations.

[1] http://edamontology.org/

Towards ubiquitous OWL computing: Simplifying programmatic authoring of and querying with OWL axioms

Hilmar Lapp and Jim Balhoff, National Evolutionary Synthesis Center (NESCent), Durham, NC
Email: hlapp@nescent.org

Semantic web technologies have enjoyed a growing popularity in integrating and connecting data, especially in the life sciences. For example, RDF and triple stores have been used to connect large amounts of diverse biological data and knowledge by shared entities and properties. Ontologies have been applied with great success to harmonizing and defining terminologies for many areas of descriptive biology, most prominently gene function, and the processes and locations in which gene products act. OWL reasoning and OWL ontologies have been used to assess the similarity of biological observations described in natural language text by the degree they share semantics. Even though triple store, ontology authoring, and reasoning technologies have become substantially more powerful in recent years, applying them at scale and for complex discovery applications is still hampered by informatics challenges and limitations. Here, we present two generically useful tools that were developed in response to ontology computing challenges encountered within the Phenoscape project (http://phenoscape.org), an initiative that uses semantic web technologies to render evolutionary phenotype descriptions amenable to computational data mining and integration.

The first tool, named Scowl, greatly simplifies and thereby accelerates the kind of ad-hoc, in bulk OWL ontology axiom authoring that is often necessary as part of the build pipelines for OWL ontology-based knowledgebases. Specifically, Scowl provides a declarative API for creating OWL class expressions and axioms in a natural way that mirrors OWL Manchester Syntax. The tool was originally designed to allow better authoring and expert review of the axiom generations necessary for transforming datasets into pertinent OWL models. Due to the high human readability of OWL Manchester Syntax it could also be used for literate programming of ontologies in ways that much better support revision control, integration testing, and other collaborative authoring infrastructure available for source code-like text.

Owlet, the second tool, addresses the issue that the expressivity of reasoners built into RDF triple stores is much more limited compared to OWL reasoners. As a consequence, using OWL constructs more complex than simple named classes, such as disjunctive OWL class expressions composed ad-hoc in response to user input, often result in complex, error prone, and poorly performing queries in SPARQL (the query language for RDF), whereas an OWL reasoner could resolve them quickly and correctly. Owlet allows integrating OWL class expressions directly into SPARQL queries. It masquerades as a SPARQL query endpoint, recognizes an OWL expression embedded in OWL Manchester Syntax, expands the expression to a FILTER clause that enumerates the matching ontology classes, and passes the resulting expanded query on to the SPARQL endpoint for the triple store.

Both Scowl and Owlet are written in Scala, and available under the MIT License from Github at http://github.com/phenoscape/scowl and http://github.com/phenoscape/owlet, respectively.

This talk is accompanied by poster #10.

Integrating Taverna Player into Scratchpads

Robert Haines[*], **Simon Rycroft**[+], **Vince Smith**[+], **Carole Goble**[*]

rhaines@manchester.ac.uk, **s.rycroft@nhm.ac.uk**

[*]School of Computer Science, University of Manchester, UK; [+]Natural History Museum, London, UK

Project websites: http://www.taverna.org.uk and http://scratchpads.eu
Source code: https://github.com/myGrid/taverna-player and https://git.scratchpads.eu/git/scratchpads-2.0.git
Licence: Taverna Player – BSD; Scratchpads – GPL2

Scratchpads, developed as part of the ViBRANT[1] project, are an online virtual research environment for biodiversity, allowing anyone to share their data and create their own research networks. Sites are hosted at the Natural History Museum London, and offered freely to any scientist.

Sites can focus on specific taxonomic groups, or the biodiversity of a biogeographic region, or indeed any aspect of natural history. Scratchpads are also suitable for societies or for managing and presenting projects. Key features of Scratchpads include: tools to manage biological classifications, bibliography management, media (images, video and audio), rich taxon pages (with structured descriptions, specimen records, and distribution data), and character matrices. Scratchpads support various ways of communicating with site members and visitors such as blogs, forums, newsletters and a commenting system. There are currently 568 Scratchpads with 6,759 active users.

Taverna Player, developed as part of the BioVeL project[2], enables the running of a workflow within a Ruby-on-rails application. Taverna Player has a REST API that allows inputs to the workflow to be specified, a run to be started and monitored, and the resultant outputs to be retrieved. Any interactions the workflow includes are presented to the user for them to complete. Taverna Player has been released in the RubyGems registry[3] and is used within the BioVeL Portal[4] to run a wide range of biodiversity workflows.

As part of a collaboration between BioVeL and ViBRANT, Taverna Player has been integrated into Scratchpads in two ways. Firstly, workflows can be embedded in a page in the same way a video from YouTube would be embedded; the workflow itself is running on the BioVeL Portal but all set up and interaction is done in the embedded widget within the Scratchpads site. Secondly, the Scratchpads can use the Taverna Player REST API directly; this allows workflows to be run with a higher degree of control and results to be ingested back into the Scratchpads for further analysis. In both cases data can be automatically injected into the workflow run from the host Scratchpads site.

Security is handled at the individual Scratchpads level; each Scratchpads site has its own credentials to access the BioVeL Portal and run workflows. This allows the community within a Scratchpads site to create and share workflow runs that all members have access to by default while preserving privacy if required.

In this talk the Scratchpads system and Taverna Player are described and their integration will be demonstrated within a live Scratchpads site.

This work was enabled by BioVeL (Grant no. 283359) and ViBRANT (Grant no. 261532) funded by the European Commission 7th Framework Programme (FP7) as part of its e-Infrastructures activity.

[1] http://vbrant.eu
[2] http://www.biovel.eu
[3] https://rubygems.org/gems/taverna-player
[4] https://portal.biovel.eu

Small tools for Bioinformatics

Pjotr Prins[1], Artem Tarasov[2] and Konstantin Tretyakov[3]

Affiliations: 1. Medical Genetics, University Medical Center Utrecht, The Netherlands; 2. Department of Statistical Simulation, St. Petersburg State University, Russia; 3. Institute of Computer Science, University of Tartu, Estonia
Contact E-mail: j.c.p.prins@umcutrecht.nl
URL: https://github.com/pjotrp/bioinformatics
Source code: https://github.com/pjotrp
License: FOSS licenses approved by the Free Software Foundation (FSF)

Introduction
The small tools for Bioinformatics MANIFESTO is a grass-roots wake-up call for software developers which counters recent trends in providing largish 'monolithic' software solutions for bioinformatics without true free and open source software (FOSS) licenses. In this talk we present three tools together as a case study that represent the spirit of the MANIFESTO. These performance related tools have impact on designing and running NGS sequencing pipelines and are used today in major sequencing centers around the world. After speedily running ***sambamba once-only,*** using a **pfff** checksum, we discuss the overall philosophy of writing small tools for bioinformatics pipelines linking the design of these tools to the MANIFESTO.

Sambamba
With sambamba we set out to prove we could write an incarnation of samtools that is as efficient and can also make use of fine-grained parallelism to accelerate analysis. Sambamba is written in the D programming language which is a modern compiled programming language with run-time performance similar to that of C. D has powerful abstractions for parallel computing which it possible to easily scale sambamba with the number of cores until the point that input/output (I/O) hardware gets exhausted. Where samtools takes about 9 minutes to merge 3 BAM files of 1GB, sambamba takes 1 minute by utilising 12 cores. Sambamba is not only a fast alternative to samtools, but it also comes with extra functionality, a descriptive error handling system, a sophisticated look-ahead parser and powerful filtering.

Fast probabilistic file fingerprinting for big data
Biological data acquisition is raising new challenges both in data analysis and handling. Simply transferring files can be prohibitively slow due to their size. Common usage patterns, such as comparing and transferring, are proving computationally expensive and are tying down shared resources. Probabilistic Fast File Fingerprinting (pfff) exploits the variation present in biological data and computes fingerprints by sampling randomly from the file instead of reading it in full. Consequently, it has a flat performance characteristic correlated with data variation rather than file size. For file comparison probabilistic fingerprinting is as reliable as existing hashing techniques, such as MD5, with provably negligible risk of collisions.

Once-only
Once-only is inspired by the Lisp once-only function, which wraps another function and calculates a result only once if the inputs have not changed. Once-only makes a program or script only run once, provided the inputs do not change. This is very useful when running a range of jobs on a compute cluster or GRID. It may even be useful in the context of webservices. With once-only there are no worries about submitting serial jobs multiple times and rerunning a command when an input or output file changes. A mistake, an interruption, a hardware failure, or even a parameter tweak, does not mean everything has to be run again from scratch.

Discussion
Sambamba is a drop-in replacement of samtools. This is made possible by the fact that samtools adheres to the small tools design. Likewise, pfff replaces MD5 for large files. Once-only is incrementally beneficial for tools that run on the command line and can be submitted to a compute cluster. All three tools are small tools and were written in the Unix tradition by making software solutions self contained so they become modular and pluggable and can be easily replaced by a new generation of tools. As the MANIFESTO states: “Software is software. Software should be easy to change, replace and improve.”

This talk is accompanied by poster #11.

SEEK for Science: A Data Management Platform which Supports Open and Reproducible Science.

Stuart Owen[1*], Natalie J. Stanford[1], Katy Wolstencroft[4], Martin Golebiewski[2], Olga Krebs[2], Quyen Nguyen[2], Dawie van Niekerk[3], Lihua An[2], Meik Bittkowski[2], Ivan Savora[2], Jacky L. Snoep[3], Wolfgang Mueller[2], Carole Goble[1].

1. University of Manchester, UK; 2. Heidelberg Institute for Theoretical Science, Germany; 3. Stellenbosch University, South Africa; 4. Leiden Institute of Advanced Computer Science, Netherlands.

*stuart.owen@manchester.ac.uk

'Open Science' is vital for ensuring accessibility, and reproducibility of research. Publishing findings in open access journals is not enough; readers need access to the assets (data, models, SOPs etc.) made during, or used within, the research for validation and reuse.

SEEK for Science (http://www.seek4science.org/about) is an "Aggregated Asset Infrastructure", providing a suite of tools to support access-controlled asset sharing across large consortia of researchers working in systems biology. A scientist may register their assets, either through a direct upload, or through a web-link held in public databases. SEEK supports versioning of all assets in order to retain the history and life-cycle of the assets. The Investigation, Studies, Assay (ISA) framework is used so that registered assets can be interlinked to provide descriptions, scientific context, relationships between assets, and provide clear credit to the scientists and projects involved in the creation of the assets.

SEEK is open-source software, and is highly configurable and adaptable. Although its origins are for use in Systems Biology research, it is being used and adapted in several other projects including BioVeL (http://www.biovel.eu), and the Virtual Liver Network (http://www.virtual-liver.de/wordpress/en).

Code URL: https://bitbucket.org/seek4science/seek

Arvados: Achieving Computational Reproducibility and Data Provenance in Large-Scale Genomic Analyses

Brett Smith[1], Adam Berrey[1], Alexander Wait Zaranek[1,2]
(1) Curoverse, Inc; (2) Harvard Medical School, Boston, USA

Project: https://arvados.org
Code: https://github.com/curoverse/arvados
License: GNU AGPL v3 and Apache 2 for SDKs

Arvados is an open source platform for storing large genomic and biomedical data sets, executing distributed computations, and federated data sharing between private clouds. The platform implements a series of computing strategies to support data provenance and computational reproducibility:

- Content addressing to provide canonical, globally-unique, cryptographically-verifiable references to data sets;
- Manifests that provide a means to describe very large data sets (e.g. exabyte scale) in a compact, canonical, durable form for long term referencing;
- Computational job management that uses Docker containers and virtualization to capture exact configurations for distributed computations that utilize multiple nodes for parallel computation with the map/reduce programming pattern;
- Generation of a metadata graph that records the provenance of individual data sets and the usage of those data sets by computational jobs within the system;
- Graphical visualization of provenance for data sets stored in the system.

We will describe how Arvados can be used in a public cloud service such as Amazon Web Services or a private cloud using a hypervisor such as XenServer to store and analyze genomic data. We will show how the system can run pipelines created with common bioinformatics tools such as GATK and languages such as Python that result in clear and verifiable records of data provenance and outputs that are consistently reproducible over extended periods of time.

Arvados is based on software originally developed at Harvard Medical School for the Harvard Personal Genome project and deployed in a multi-cluster federated system for storing and analyzing those data. The software is developed primarily in Ruby and Go, licensed as free/open source software, and maintained by the Arvados project and community.

Enhancing the Galaxy Experience through Community Involvement

Daniel Blankenberg[1,2] and the Galaxy Team[2]
[1]Department of Biochemistry and Molecular Biology, Penn State University, University Park, PA 16801, USA
[2]http://galaxyproject.org

Project URL: http://galaxyproject.org

Galaxy (http://galaxyproject.org) is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research. Galaxy makes bioinformatics analyses accessible to users lacking programming experience by enabling them to easily specify parameters for running tools and workflows. Analyses are made transparent by allowing users simple access to share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis. Extending Galaxy with new tools, datasources, and external resources has been designed to be a plug-n-play process.

Among Galaxy's greatest strengths and assets is the involvement of its community. Here, we discuss several recent developments that are aimed at further engaging and incentivizing the involvement of community members.

Galaxy enables experimental biologist access to powerful analysis infrastructure through the web. However, this access is limited by the ability of users to obtain answers regarding the use of the available tools. Currently, the Galaxy Project provides this support through the use of several mailing lists. These mailing lists have been instrumental in supporting the Galaxy community. However, there are several areas that could be improved upon, including frequent reposting of common questions and better engagement of the Galaxy community in answering questions. Towards these ends, we have integrated BioStar (doi:10.1371/journal.pcbi.1002216) into the Galaxy framework as a Q&A support application. BioStar takes a Stack Exchange-based approach, where a participant asks a specific question and the other participants provide direct answers to the posed question. Other users can then vote on the correctness of each provided answer with the original poster given the option of approving one or more answers; the most positively voted answers rise to the top of the page. Participants are granted points and awards based upon the community assessment of their contributions. This has the positive effect of providing access to concise answers to specific questions, which can be easier for users to find and follow.

The Galaxy ToolShed (http://toolshed.g2.bx.psu.edu) serves as an appstore to all Galaxy instances worldwide. It is a free open service that hosts Galaxy Utilities including Tools and Workflows. The ToolShed allows Galaxy administrators to install thousands of freely available tools into their instances. It also manages the tool external dependencies and tool updates thus making their life easier. Moreover it allows the tool developers to easily share, update and manage their tools. There are dozens of Galaxy public servers and hundreds of private ones. By depositing an analysis tool within the ToolShed, developers gain free and instant click-to-install access to a large and active user base. More users → more citations → more grants.

Title	Supporting dynamic community developed biological pipelines
Author	*Brad Chapman*, Rory Kirchner, Oliver Hofmann, Winston Hide
Affiliation	Harvard School of Public Health
Contact	bchapman@hsph.harvard.edu
URL	https://github.com/chapmanb/bcbio-nextgen
License	MIT

bcbio-nextgen is a community developed set of validated, scalable pipelines for running variant calling and RNA-seq analyses. It creates an infrastructure from open source tools that is easy to install and run. The goal is to implement best-practice approaches that scale across multiple architectures ranging from single machines to large clusters, and combine this with automated validation of results for correctness against reference standards.

For example, the practical goal of the variant calling pipeline within bcbio-nextgen is to let biologists work with best-practice variant calls, instead of struggling with processing raw next-generation sequencing reads. To do this, we integrate aligners like bwa-mem and variant callers like GATK and FreeBayes alongside other BAM and variant manipulation tools, and then validate variants against reference callsets from the Genome in a Bottle consortium. The outcome is a push button analysis framework that parallelizes a complex set of tools to produce high quality variants as ready to analyze outputs.

The challenge associated with supporting bcbio-nextgen is that it relies on many open-source tools and works across a wide range of heterogeneous platforms. The result is a large amount of community time spent on installation issues, rather than answering biological questions. We produced an automated installer and updater using CloudBioLinux which solved many adoption issues but also requires work to maintain, extend and test.

At BOSC, we'll discuss two approaches designed to improve ease of use:

- Isolating dependencies within lightweight Docker containers. This provides a standard distribution environment containing all third-party code and tools. This avoids the need to compile and install these on a wide variety of systems. Additionally it provides a reproducible analysis environment for export, archival and sharing.
- Providing a Amazon Web Services implementation that is resilient to failure and makes using of spot instances. This helps overcome two major hurdles to cloud adoption: difficulty scaling on less reliable commodity hardware and justifying spending on external compute. We'll share our experiences redesigning bcbio-nextgen to run on non-shared filesystems and handle higher failure rates found in cloud environments.

These improvements move towards the goal of having shared community developed pipelines usable by researchers, clinical labs and the general public. By removing the separation between up to date research grade tools and validated clinical grade tools, we enable contributions from multiple communities and standardization around stable reliable tools for the overlapping needs of the diverse translational research community.

Title: **Open as a strategy for durability, reproducibility and scalability**
Authors: Jonathan A. Rees, Karen Cranston
Author affiliations: National Evolutionary Synthesis Center (NESCent), rees@nescent.org
URL for the overall project web site: http://opentreeoflife.org
URL for accessing the code: https://github.com/OpenTreeOfLife
Open source license: GPL v. 3 and BSD 2-clause

Open Tree of Life aims to create a complete and dynamic evolutionary history of all species by combining published phylogenetic trees with taxonomic hierarchies. Being a grant-funded academic project, our strategic decisions have been driven by the goals of scientific reproducibility of computational processes, scalability through automation that replaces what in other projects have been manual steps, and project durability beyond the end of the grant period. We have tried to make the project as open as possible because long-term success depends on scientific data sharing and volunteer curation, and because forks of the project may, we hope, eventually end up being as scientifically productive as the 'master' branch. The project practices the following:

- Free software, open access publications, and open data (CC0 when possible), with inputs, intermediate artifacts, and outputs available on the web
- Software development in the open on github, with biologist user/curators encouraged to use the issue tracker
- A largely open approach to scientific decision-making and progress reporting, using Google groups, Google drive, and similar tools
- Technical "opening" of artifacts through reliance on text files on github, as contrasted with a database, for live maintenance of data, using NeXML (an open standard for phylogenetic study data), json, tsv. In addition, provenance is tracked so that file origins are clear, improving transparency
- Transparency of study methods through scripting (mostly make and python); anyone can rebuild the outputs from the inputs. For example, taxonomy construction involves alignment of input taxonomies and correction of errors. Detecting errors is manual, but all correction steps are recorded, with provenance and evidence, as operations that are applied by the build script, similar to a log replay. This makes corrections reuseable.

As of April 2014 we are pre-launch, but already enjoying the benefits of this approach. The decentralized group of biologists and programmers can coordinate using open services and use resources on the web without having to worry about credentials or secrecy, and individuals not directly connected with the project can see what we're doing and contribute suggestions and data. In the future we are planning and hoping for community contribution of phylogenetic trees to our repository, taxonomy improvements, and of course research that builds on anything and everything we've done.

Poster #13.

Author: Laurent Gautier
Software Licence: 3-clause BSD
Project URL: To be released by BOSC

Connecting computational steps for NGS, and beyond.

The analysis of Next-Generation Sequencing (NGS) data consists in applying 3rd party tools in sequences of inter-dependent steps.

While this is often referred to as a "pipeline", we instead see the process as much less linear than a pipe. We propose here a framework to implicitly build and manipulate a directed graph of steps in an interactive and incremental way that we believe to be natural and intuitive for bioinformaticians and computational biologists.

Our "railroadtracks" framework is designed to be a modular ensemble of loosely coupled components, able to integrate arbitrary sets of 3rd party tools such as aligners, read counters, and differential expression methods in the case of RNA-Seq into models. The persistence of the graph connecting steps is also enabling reproducibility and the computation of variants of a process while keeping their computational cost relatively low.

Poster #14.

Updates to MISO, the open-source NGS LIMS project

Xingdong Bian, Anil Thanki, Robert Davey
The Genome Analysis Centre, Norwich Research Park, UK

Abstract

MISO ("Managing Information for Sequencing Operations") is a freely available open-source LIMS for recording next-generation sequencing (NGS) metadata for sequencing centres. Based on the common objects (projects, samples, libraries, pools and runs etc.) from the European Bioinformatics Institute (EBI) Sequence Read Archive schemas, MISO stores relevant metadata for typical lab workflows and automatically tracks run information from common NGS platforms (e.g. Illumina GA, HiSeq and MiSeq, Roche 454, ABI SOLiD and PacBio RS). MISO can also initiate HPC job submission for initial analysis and QC of sequencing data, and automatically generates public repository data submission schemas. Because MISO is modular, and it is designed to be extensible and customisable, MISO can by used by both large centres characterised by high-throughout data production and smaller scale laboratories with constrained expenditure for IT solutions.

We present the recent highlighted updates of MISO: Plate support (e.g. 96-well and 384-well), new visualisations in reporting, entity groups (i.e. grouping of objects for easier project management, such as sample groups), more flexible barcode printing, sequencing run QC analysis reporting and visualisation, support for traditional Excel/ODF/CSV input and output for bulk data import and export, continued support of new NGS platforms, and a new workflow system for customised lab processes.

Poster #15.

Running Taverna Workflows within IPython Notebook

Alan Williams, Aleksandra Pawlik, Carole Goble
{alan.r.williams, carole.goble}@manchester.ac.uk,
a.pawlik@software.ac.uk

School of Computer Science, University of Manchester, UK
Project's Web site: http://www.taverna.org.uk/

Source code: https://github.com/myGrid/DataHackLeiden.git
Licence: The MIT License (MIT)

IPython Notebook[1] is a browser-based environment for interactive computing. Users can write, edit and replay Python scripts. IPython Notebook has support for interactive data visualization and report presentation. A Notebook can be saved and shared. Notebooks can be replayed using the same or different data. The record of a notebook "run" can be saved and displayed in a Notebook Viewer[2].

The Taverna Player[3], developed as part of the BioVeL project[4], through its API and the use of iframes enables the running of a workflow within the Taverna Portal[5] to be included as part of another Web site. The Taverna Player has a REST API that allows workflow inputs to be specified, a workflow run started and monitored, and the resultant outputs retrieved.

As part of work enabled by pro-iBiosphere[6], the Taverna Player Client package[7] was developed for Python. This Client may be used to run workflows within an IPython Notebook; data can be passed from the Notebook as inputs to Taverna workflows to be executed, and results retrieved from the run back into the Notebook. Using Taverna's interaction service, the workflow run can be steered within the Notebook browser window. Reports can be generated for workflow runs using jinja2[8] templates.

The Taverna Player Client package is released in the PyPi registry[9] and has been used to orchestrate the running of BioVeL workflows for data refining and ecological niche modelling.

In this talk, an overview of the capabilities of IPython Notebook is given, the REST API to Taverna Player is described and the use of the Taverna Player Client package are demonstrated.

This work was enabled by BioVeL, a project (Grant no. 283359) funded by the European Commission 7th Framework Programme (FP7) as part of its e-Infrastructures activity, and by the Software Sustainability Institute supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through grant EP/H043160/1

[1] http://ipython.org/index.html
[2] http://nbviewer.ipython.org/
[3] https://github.com/myGrid/taverna-player
[4] http://www.biovel.eu/
[5] https://portal.biovel.eu/
[6] http://www.pro-ibiosphere.eu/
[7] https://github.com/myGrid/DataHackLeiden
[8] http://jinja.pocoo.org/
[9] https://pypi.python.org/pypi/tavernaPlayerClient

Poster #16.

Reconstruction of ancestral genomes in presence of gene gain and loss

Shuai Jiang[12*], Pavel Avdeyev[3], and Max A. Alekseyev[1]
[1]Computational Biology Institute, George Washington University, Ashburn, VA, USA
[2]Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA
[3]Bioinformatics Institute, Academic University, St. Petersburg, Russia

Genome rearrangements (such as *reversals*, *translocations*, *fusions*, and *fissions*) are evolutionary events that shuffle genomic material without altering it otherwise. One of the key computational problems in comparative genomics is reconstruction of genomes of common ancestors for genomes of living species and the sequence of evolutionary events (*evolutionary history*) between them. In rearrangement-based approaches (particularly employed by the MGRA tool published by the third author in *Genome Res.* 2009), ancestral genomes are reconstructed by minimizing the number of rearrangements along the branches of the phylogenetic tree.

In algorithmic studies of genome rearrangements, genomes are traditionally idealized to have equal gene content. However, in reality, instances of the same gene in different lineages may independently mutate, making it impossible to recognize them as orthologs. While for two genomes the information about missing (*deleted*) genes may not be that beneficial, this situation changes as more genomes come into consideration. In particular, genomes that share some genes are likely to be evolutionarily closer to each other than to a genome where these genes are absent. But most importantly, with the growing number of input genomes, the number of genes shared across all the genomes drops substantially. Therefore, in comparative studies of multiple genomes, it becomes crucial to maintain not only information about orthologous genes but also about deleted/inserted genes across the genomes.

We present a tool called MGRA2 that extends MGRA to support gene insertion and deletion (*indel*) operations. Given a set of genomes and their phylogenetic tree, MGRA2 reconstructs ancestral genomes at the internal nodes of the tree. MGRA2 not only organicly incorporates indels into the rearrangement analysis of multiple genomes but also generalizes algorithms employed by MGRA and make them applicable to "hard" genomic datasets inaccessible for MGRA and similar tools. To evaluate the performance of MGRA2, we conducted two sets of experiments for real and simulated genomes and compared the results of MGRA2 with other existing tools such as GapAdj and PMAG$^+$. These experiments demonstrated supremacy of MGRA2 in all comparisons.

The MGRA2 software is distributed under GNU GPL v2 license. The MGRA2 sources are available at GitHub repository `http://github.com/ablab/mgra/`.

This work is supported by the National Science Foundation under Grant No. IIS-1253614.

*Corresponding author. Email: `jiangs@email.sc.edu`

Poster #18.

GEPETTO (GEne PrioriTization Tool) is an open-source framework, distributed under the LGPL license. The source code is available at sourceforge.net/projects/gepetto/files/.

GEPETTO update: An Open Source Framework for Gene Prioritization

Hoan Nguyen
nguyen@igbmc.fr
Integrated structural Biology department, (IGBMC), Illkirch, France
Integrative Genomics and Bioinformatic Laboratory-LBGI, Strasbourg, France

Recently, the use of high-throughput biotechnologies has emphasized the need for new prioritization tools to identify the most promising genes/proteins among a list of candidates resulting from high-throughput experiments [1]. Large sets of genes must be evaluated, in order to score and rank them according to their similarity to known genes and their potential viability as candidates for important applications, such as diagnostic/prognostic markers, drug targets, etc. The biomedical community urgently needs a customizable and extensible framework for gene selection that can handle large-scale biological information from public, as well as private data resources.

GEPETTO (GEne PrioriTization Tool) is an original open-source framework, distributed under the LGPL license, for gene selection and prioritization on a desktop computer that ensures confidentiality of personal data. It takes advantage of the data integration capabilities from public database, combined with in-house developed gene prioritization methods. It currently incorporates six prioritization modules, based on gene sequence, protein-protein interactions, gene expression, disease-causing probabilities, protein evolution and genomic context). Each module integrates specialized evaluation or ranking approaches including K-means clustering, Pearson Correlation and Fisher's omnibus analysis and network-based approaches with neighbourhood evaluation of candidate or disease genes. The final overall prioritized candidate list is determined using several methods: order statistics [2], Robust Rank Aggregation[3] and GPSy's optimal weight [4].

GEPETTO is written in Java/Python and supported by an advanced modular architecture, which means that it can easily be modified and extended by the user, in order to include alternative scoring methods and new public/private data sources. Recently, we used the jBPM (JBoss Business Process Management) workflow engine to define and execute the prioritization process. The GEPETTO software and applications are available at sourceforge.net/projects/gepetto/files/ or decrypthon.igbmc.fr/sm2ph/cgi-bin/gepetto.

1) Moreau Y, Tranchevent LC. Computational tools for prioritizing candidate genes: boosting disease gene discovery. Nat Rev Genet. 2012 Jul 3;13(8):523-36.
2) Kolde R, Laur S, Adler P, Vilo J. Robust rank aggregation for gene list integration and meta-analysis. Bioinformatics. 2012 Feb 15;28(4):573-80. doi: 10.1093/bioinformatics/btr709. Epub 2012 Jan 12.
3) Britto R, Sallou O, Collin O, Michaux G, Primig M, Chalmel F. GPSy: a cross-species gene prioritization system for conserved biological processes--application in male gamete development. Nucleic Acids Res. 2012 Jul;40(Web Server issue):W458-65. doi: 10.1093/nar/gks380. Epub 2012 May 8.
4) Tranchevent LC, Capdevila FB, Nitsch D, De Moor B, De Causmaecker P, Moreau Y.A guide to web tools to prioritize candidate genes. Brief Bioinform. 2011 Jan;12(1):22-32. doi: 10.1093/bib/bbq007. Epub 2010 Mar 21.

Poster #19.

NeoPipe is an open-source framework, distributed under the LGPL license. The source code is available at http://sourceforge.net/projects/neopipealign/

NeoPipe: A workflow for protein family analysis

Vincent Walter, Julie Thompson, Olivier Poch and Hoan Nguyen
Contact: nguyen@igbmc.fr
Integrated structural Biology department, (IGBMC), Illkirch, France
Integrative Genomics and Bioinformatic Laboratory-LBGI, Strasbourg, France

New advanced technologies including the next-generation DNA, and information technology have significantly improved our capacities of developing biological knowledge and changing our understanding of diseases, phenotype and genotype. In this post-genomic context, protein sequence analysis is a key issue to better understand the evolutionary, structural and functional aspects. NeoPipe is a tool of analyzing a protein family , which consists of 8 steps concerning the search for homologous sequences in multiple databases (protein, 3D structures,…) and functional and structural annotations of clustered multiple alignment of complete sequences (MACS). Those indicate the relationship between the protein subfamilies. The emphasis is to get a high quality alignment by performing refinement and corrections (evaluated by a quality score at each steps) and giving a clustered and annotated alignment of potential subgroups**.** NeoPipe's application, APIs and REST Web Services are implemented in Java and supported on Linux. NeoPipe is open-source (under the LGPL license) and the source code is available on SourceForge at http://sourceforge.net/projects/neopipealign/ using a GIT repository. NeoPipe website is developed in Java, JavaScript (JQuery), AJAX with all major browser supported. The website is available at http://lbgi.fr/neopipe/.

Poster #20.

Title: MyGene.info updates: scalable gene-centric web services with user contributions
Authors: Chunlei Wu and Andrew I. Su (presenting author underlined)
Email: cwu@scripps.edu asu@scripps.edu
Affiliation: The Scripps Research Institute, 10550 N Torrey Pines Rd, La Jolla, CA 92037
Project web site: http://mygene.info
Source code: https://bitbucket.org/sulab/mygene.hub/src
https://bitbucket.org/sulab/mygene.info/src
Open Source License being used: **Apache License**
Considered for a talk, a poster, or both: **both**

Biological applications typically start with a query interface for users to search for their favorite genes. Building such an interface often requires developers to maintain a dedicated gene annotation database to translate user queries into the desired gene annotation objects. Setting up a database server and keeping it updated can be a time-consuming and cumbersome tasks. Since the majority of raw gene annotation data are coming from several large data centers like NCBI and Ensembl, developers are also duplicating their efforts to setup gene annotation databases from essentially the same data providers.

MyGene.info (http://mygene.info) is a cloud-based solution to abstract the task of building a gene annotation database into a set of scalable and extensible web services. End users have access to two simple-to-use REST web services for gene annotation query and retrieval, without worrying about designing, building and maintaining a dedicated database. The gene query service [1] takes the user query string and returns the matching gene objects with desired annotations; and the gene annotation service [2] returns annotation data for given gene IDs. Both services return JSON (Javascript Object Notation) formatted data, making them easy to integrate into applications.

Right before last year's BOSC, we released new v2 MyGene.info API [3]. Thanks to the scalable backend built upon MongoDB and Elasticsearch, we expanded our services to 16M genes from >13K species, while keeping the high query performance [4]. Since the release, MyGene.info has served over 60M requests (July 2013-March 2014). As of now, MyGene.info services steadily serve ~1.5M requests per month, and our Python client *mygene.py* module [5] also achieves ~600 downloads per month.

As a centralized resource hub, one important aspect of MyGene.info is the user contributions. Even though we, as core developers, are always adding more gene annotation data into our system, it's equally important to build a framework to allow our users to contribute data into MyGene.info. Users can now write a simple data importer script, based on the template we provide, to load their own data into MyGene.info, so that they will be accessible via our high-performance query engine. Given the flexibility of JSON data structure, new data under a new field name will not affect existing data, avoiding breaking existing applications.

Another way of user contributions is to write custom query filters. Our users often need to restrict their search scopes, e.g. for a given species, for ncRNA genes only, or simply a specific list of genes. We allow users to contribute their custom query filters (each with a unique name) at server-side, that way they can achieve both higher performance (server-side execution) and cleaner query syntax (with the filter name instead of the actual query). Moreover, other users can benefit from re-using those custom filters relevant to their use cases.

[1] http://mygene.info/doc/query_service.html
[2] http://mygene.info/doc/annotation_service.html
[3] http://sulab.org/2013/07/mygene-info-v2-api-goes-live/
[4] http://mygene.info/#what-s-new-in-v2-api
[5] https://pypi.python.org/pypi/mygene

Poster #21.

Aiding the journey from data to publication in the plant sciences

Robert Davey[1], Vicky Schneider-Gricar[1], Susanna Sansone[2], Paul Kersey[3], Jim Beynon[4], Ruth Bastow[4], Mario Caccamo[1]
[1] Presenting author: robert.davey@tgac.ac.uk
[1]The Genome Analysis Centre, Norwich, UK, [2]The University of Oxford, Oxford, UK, [3]The European Bioinformatics Institute, Hinxton, UK, [4]The University of Warwick, Coventry, UK

The development of new experimental technologies has opened the way to new, data-generative approaches to plant research, particularly within the genomics field but also high-throughput transcriptomics, proteomics and metabolomics.

Furthermore, publication models are moving away from the traditional "data late" approach, and are shifting towards the "data early", with particular pressure being made by funding agencies to see data publicised quickly. Alongside the wealth of these large plant science datasets held in public and private laboratories around the globe, there are a large number of tools to help researchers disseminate, analyse and publish those datasets. However, the disparate nature of the tools, data formats and scientific problems in light of this expansive experimental development has resulted in a lack of interoperable, production-quality software available for data analysis and dissemination.

We present a newly funded project, **Collaboratively Open Plant Omics** (COPO), to address this disparity in interoperability and easy access to important services in the 'omics data realm. We will develop a framework to utilise existing services to facilitate the description, deposition and publication of datasets, but also to enable the identification and citation of datasets, thereby increasing transparency and reproducibilty. Promoting reward for making data available is a central aim of the project.

By developing high quality, stable Application Programming Interfaces (APIs) and virtualised resources we will be tying together existing services such as: the ISATools metadata suite; EBI data repositories; Galaxy and iPlant analytical platforms; figshare, Research Object, Scientific Data and Gigascience platforms; to:

(i) develop and use community-accepted standards for data representation
(ii) facilitate submission to persistent archival resources, for data publication and citation
(iii) enable seamless transition from data to analysis platforms
(iv) provide aggregated provenance and suitable publication markup to link citable resources

All project code will be open source under an appropriate licence, such as GPL, LGPL or BSD, and made available from project outset on TGAC's GitHub repository.

Poster #22.

Title: Bio2RDF mobile: an app for biological semantic web databases

Authors:
Déraspe M[1], Rheault J-F[1], Joly-Beauparlant C[2], Emonet V[2], Belleau F, Droit A[2]
[1]Département de Biochimie, de microbiologie, et de bio-informatique, Université Laval, Québec, QC, Canada
[2]Département de médecine moléculaire, Faculté de Médecine, Université Laval, Québec, QC, Canada

Abstract

Bio2RDF provides one of the largest networks of Linked Data for Life Sciences. Herein, we describe a new way to navigate into the large flow of biological and medical databases through a mobile web application. Mobile applications are increasingly important tools in the everyday life of a scientist. Bringing a collection of reference databases uniformly into their pockets was the main motivation of this project. We therefore developed Bio2RDF-mobile and published it in the Android Play Store and the iOS App Store.

Since 2008, Bio2RDF is a well-recognized open-source project that provides linked data for life sciences using Semantic Web technologies (Belleau, 2008). More recently RDF[1] databases have grown in popularity with big data providers like the EBI[2] and the NCBI[3] institutions. One of the beauties of open RDF databases is they can be queried with the SPARQL query language directly throughout the Web. They offer a lot of flexibility on how scientists can fetch information of their interest. Moreover the data can come from several sources and be linked together; easy-to-use mashups can thereby be created. However, for the uninitiated biologist, SPARQL queries can be repellent and long URIs in the results are not always appealing. Mobile platforms are a good niche for which to propose simple, intuitive, uniform and user-friendly interfaces. To our knowledge, not a lot of attempts have been made to provide such an application combined with Semantic Web technologies. We can see agreements in Kumar (2012) for the usefulness of mobile apps in science by allowing a quicker access to scientific data mainly by avoiding the use of a personal computer. Several circumstances could take advantage of this : meetings, conferences, scientific discussions and so on.
The source code can be find at https://bitbucket.org/zorino/ionic2rdf/

References

Belleau, F., Nolin, M. A., Tourigny, N., Rigault, P., Morissette, J. (2008). "Bio2RDF: towards a mashup to build bioinformatics knowledge systems." J Biomed Inform 41:706-716.

Kumar, S., Boccia, K., McCutchan, M., Ye, J. (2012). "Exploring spatial patterns of gene expression from fruit fly embryogenesis on the iPhone." Bioinformatics 28:2847-2848.

[1] Resource Description Framework
[2] European Bioinformatics Institute
[3] National Center for Biotechnology Information

Poster #23.

Tripal: an open source toolkit for building genomic and genetic data websites and databases

Stephen Ficklin[1], Lacey Anne Sanderson[2], Margaret Staton[3]*, Chun Huai Cheng[1], Sook Jung[1], Kirstin Bett[2], Doreen Main[1]

1 Department of Horticulture, Washington State University
2 College of Agriculture and Bioresources, University of Saskatchewan
3 Department of Entomology and Plant Pathology, University of Tennessee
* mstaton1@utk.edu

URL: http://tripal.info/
URL (code): https://drupal.org/project/1337878/git-instructions
License: GNU General Public License, Version 2

Community genomic databases fulfill a critical need by offering curated and mission-specific information to targeted audiences with shared basic and applied research goals. Tripal was created in response to the need for an open source, extensible software system to support and standardize the website development projects of diverse scientific communities with growing sequence resources and uniquely interlinked datasets. Tripal provides data pages and search tools for taxonomies, genomic sequences, genetic markers and maps, stocks, cultivars, DNA or clone libraries, and publications stored in Chado[1], the standard relational database schema for biological information. Tripal is written in PHP and provides an Application Programmers Interface (API) that allows other developers to extend the core modules and create new modules. As a result, extension modules have been developed for parsing and uploading data from common computational analyses of sequence data such as BLAST[2], InterProScan[3], KEGG[4], and blast2GO[5]. Tripal was developed for use with the popular open-source content management system Drupal[6], a PHP-based platform used to power millions of websites and applications worldwide. Drupal empowers non-technical users to easily add content and functionality without the need for programming, including writing news articles, announcements or blogs as well as designing the layout and content of new pages. This robust underlying structure is leveraged in Tripal to allow curators or community members to upload different types of data through an intuitive web interface and to create new ways of visualizing heterogeneous datasets. Drupal also provides built-in user management and content versioning, an ideal base for constructing robust community annotation capabilities. By bridging Chado and Drupal, Tripal marries the power of a biological data storage schema with a web development platform to decrease the cost and time associated with development of genomic, genetic and breeding databases for diverse biological research communities. Tripal is in use or being implemented by at least 24 different databases including the Genome Database for Rosaceae[7], CottonGen[8], the Hardwoods Genomic Database[9], and KnowPulse[10]. Future development is focusing on cross-site communication, adoption of community driven data standards, and integrated "big data" analysis.

1. http://gmod.org/wiki/Chado_-_Getting_Started
2. http://blast.ncbi.nlm.nih.gov/Blast.cgi
3. http://www.ebi.ac.uk/Tools/pfa/iprscan
4. http://www.genome.jp/kegg
5. http://www.blast2go.com
6. https://drupal.org
7. http://www.rosaceae.org
8. http://www.cottongen.org
9. http://www.hardwoodgenomics.org
10. http://knowpulse.usask.ca

Poster #24.

Software Licence: Miscellaneous OSS licences (packaging project)
Project URL: http://www.lab7.io/test/solutions/biobuilds/

BioBuilds: A Model for Long Term Sustainability of Open Source Bioinformatics

Chris Mueller, Varshal Davé, Thomas Burnet, and Cheng Lee

From the original FASTA software suite to BLAST and the Human Genome Project, and now with the proliferation of tools for next generation sequencing data analysis, bioinformatics has a long history of using Open Source software. For many bioinformaticians, Open Source tools are a natural extension of their normal workflows and development philosophies. More importantly, the robust Open Source community enables the creation and sharing of common tools and methods that benefit everyone.

However, as bioinformatics extends beyond academic and research applications and gains traction in diagnostic and applied markets, there is a stronger focus on using, rather than developing, Open Source tools. This is a great evolutionary step for bioinformatics that highlights how the field is maturing. To maintain relevancy, it is important that the Open Source community grow with the new opportunities.

On one end, simple challenges such as the time and effort required to maintain installed Open Source tools must be addressed. Many Open Source tools require extensive hands-on time to install, with build and dependency errors often derailing the process entirely for all but the most committed, impacting adoption and productivity. Seemingly simple updates can require days of effort.

The problem is compounded in regulated environments, where tools must have clear provenance and additional support for verification and validation protocols, leading to a high level of uncertainty around using Open Source tools at all. It is unrealistic to expect every hospital pathology lab to employ a software engineer simply to maintain a collection of tools.

At a higher level, many of the tools developed in the last decade to support sequencing and other 'omics applications are nearing the end of their value as research targets. For example, short-read alignment methods, up until recently a go-to topic for Masters and PhD students, have very little research value left in them. As the funding agencies wind down support for this research to focus their resources on new topics, support for many important tools will fade.

To begin to address these issues, we have launched the BioBuilds project. BioBuilds is distribution of Open Source bioinformatics tools suitable for deployment in research, commercial, and regulated environments. BioBuilds' primary mission is to ensure long-term support for Open Source bioinformatics tools across a broad range of research, diagnostic, and industrial applications.

As a distribution of tools, BioBuilds provides pre-built binaries for many Open Source bioinformatics tools that "just work" out of the box on a number of platforms. All tools are built and tested and the package includes all dependencies. With vendor support, BioBuilds also includes binaries optimized for specific platforms. Like BioBrew, CloudBioLinux, and other projects that provide packaged tools, BioBuilds believes that the first step in long term support is ensuring ready access to tools.

As an organization, BioBuilds goes beyond just providing tools. To support our mission of long term support, we are actively working on funding and sponsorship models to ensure the resources are available to provide continued support for Open Source bioinformatics tools. We are working closely with academic, industry and non-profit partners, including hardware and instrument vendors, diagnostic suppliers, and other scientific software foundations such as NumFocus, Boost, and OpenMPI, to develop a model that ensures continued resources and financial support will be available for these tools long after their research support has ended.

Bioinformatics has recently reached the point where its value is clear outside of academic and research pursuits. For Open Source bioinformatics to maintain relevancy as applied markets develop, it is essential to development the mechanisms to ensure long term support is available for tools and develop healthy relationships with the industries that benefit from it. BioBuilds is a first step towards this goal.

Poster #25.

GigaGalaxy: A GigaSolution for reproducible and sustainable genomic data publication and analysis

Scott C Edmunds[1,2,*], Peter Li[1,2], Huayan Gao[3,4], Ruibang Luo[2], Dennis Chan[1], Alex Wong[1], Zhang Yong[2], Tin-Lap Lee[3,4]

1 BGI-Hong Kong Ltd., 16 Dai Fu Street, Tai Po Industrial Estate, NT, Hong Kong SAR, China.
2 BGI-Shenzhen, Beishan Industrial Zone, Yantian District, Shenzhen, China.
3 School of Biomedical Sciences, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China.
4 CUHK-BGI Innovation Institute of Trans-omics, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China.

* Corresponding Author: scott@gigasciencejournal.com

Today's next generation sequencing (NGS) experiments generate substantially more data and are more broadly applicable to previous high-throughput genomic assays. Despite the plummeting costs of sequencing, downstream data processing and analysis create financial and bioinformatics challenges for many biomedical scientists. It is therefore important to make NGS data interpretation as accessible as data generation. GigaGalaxy (http://galaxy.cbiit.cuhk.edu.hk) represents a NGS data interpretation solution towards the big sequencing data challenge. We have ported the popular Short Oligonucleotide Analysis Package (http://soap.genomics.org.cn) as well as supporting tools such as Contiguator2 (http://contiguator.sourceforge.net) into the Galaxy framework, to provide seamless NGS mapping, de novo assembly, NGS data format conversion and sequence alignment visualization. Our vision is to create an open publication, review and analysis environment by integrating GigaGalaxy into the publication platform at *GigaScience* and its GigaDB database that links to more than 40 Tetrabytes of genomic data. We have begun this effort by re-implementing the data procedures described by Luo et al., (*GigaScience* 1: 18, 2012) as Galaxy workflows so that they can be shared in a manner that can be visualized and executed in GigaGalaxy. We hope to revolutionize the publication model with the aim of executable publications, where data analyses can be reproduced and reused.

GigaGalaxy: http://galaxy.cbiit.cuhk.edu.hk
GigaDB: http://gigadb.org/
Code: https://github.com/gigascience

O|B|F - Open Bioinformatics Foundation

Membership Application

I wish to apply for membership in the Open Bioinformatics Foundation (O|B|F).

First and Last Name: _____

Street Address: _____

City, State, Zip Code: _____

Country of Residence: _____

Email Address: _____

All fields are mandatory. The O|B|F will treat all personal information as strictly confidential and will not share personal information with anyone except members of the O|B|F Board of Directors, or entities or persons appointed by the Board to administer membership communication. This may be subject to change; please see below.

I am an attendee of BOSC 201___: ☐ Yes ☐ No

If you answered No, please state why you meet the membership eligibility requirement of being interested in the objectives of the O|B|F:

(Use back of page if you need more space)

I understand that membership rights and duties are laid down in the O|B|F Bylaws which may be downloaded from the O|B|F homepage at http://www.open-bio.org/. I understand that if the O|B|F's privacy statement changes I will be notified at my email address (as known to O|B|F), and if I do not express disagreement with the proposed change(s) by terminating my membership within 10 days of receipt of the notification, I consent to the change(s).

_____

Signature