13th Annual Bioinformatics Open Source Conference

BOSC 2012

Long Beach, California
July 13-14, 2012

http://www.open-bio.org/wiki/BOSC_2012

Welcome to BOSC 2012! 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 sponsored by the Open Bioinformatics Foundation (O|B|F), a non-profit group dedicated to promoting the practice and philosophy of Open Source software development within the biological research community.

Our first keynote speaker this year is Jonathan Eisen (University of California, Davis), who is familiar to many in the community for his outspoken support of Open Science. Carole Goble (University of Manchester, UK), our second keynote speaker, is known for her leadership in distributed computing, semantic web, and social computing for scientific collaboration.

Software Interoperability and Linked Data and Translational Knowledge Discovery are new session topics this year, and several popular topics from last year (Cloud and Parallel Computing, and Genome-scale Data Management) are also on the schedule. We will hear project update reports from some of the ongoing open source bioinformatics projects. Our panel discussion on Day 2 will address Openness of Standards and Standards of Openness in the context of bioinformatics paper reviews.

Like last year, BOSC 2012 includes posters as well as talks. There are three scheduled poster sessions. We have space for several last-minute posters in addition to those listed in the program.

Thanks to generous support from Eagle Genomics, we were able to offer Student Fellowships to the authors of the three best student abstracts. Congratulations to the student winners, all of whom received free admission to BOSC: Spencer Bliven, Alexandros Kanterakis, and Sebastian Schoenherr.

BOSC is a community effort. We thank the organizing committee, the program committee, the session chairs, and the ISMB SIG chair for their help. If you are interested in participating in the organization of BOSC 2013, please email bosc@open-bio.org.

2012 Organizing Committee:
Nomi Harris (Chair), Jan Aerts, Brad Chapman, Peter Cock, Kam Dahlquist, Christopher Fields, Hilmar Lapp, Peter Rice

2012 Program Committee:
Jan Aerts, Enis Afgan, Kazuharu Arakawa, Raoul Bonnal, Timothy Booth, Brad Chapman, Peter Cock, Kam Dahlquist, Heiko Dietze, Thomas Down, Chris Fields, Erwin Frise, Jeremy Goecks, Nomi Harris, Hans-Rudolf Hotz, Konrad Karczewski, Hilmar Lapp, Heikki Lehvaslaiho, Scott Markel, Hervé Ménager, Peter Rice, Peter Robinson, Tiago Rodrigues Antão, Olivier Sallou, Ronald Taylor

BOSC 2012 Schedule

Day 1 (Friday, July 13, 2012)

Time	Title	Speaker or Session Chair
7:30-9:00	**Registration**	
9:00-9:15	**Introduction and Welcome**	Nomi Harris (Chair, BOSC 2012)
9:15-10:15	**Keynote: Science Wants to Be Open - If Only We Could Get Out of Its Way**	Jonathan Eisen
10:15-10:45	***Coffee Break***	
10:45-12:30	**Session: Cloud and Parallel Computing**	Chair: Richard Holland
10:45-11:10	Cloudgene - an execution platform for MapReduce programs in public and private clouds	Sebastian Schoenherr
11:10-11:35	Data reduction and division approaches for assembling short-read data in the cloud	C. Titus Brown
1135-12:00	MGTAXA - a toolkit and a Web server for predicting taxonomy of the metagenomic sequences with Galaxy front-end and parallel computational back-end	Andrey Tovchigrechko
12:00-12:15	Workflows on the Cloud - Scaling for National Service	Katy Wolstencroft
12:15-12:30	How to use BioJava to calculate one billion protein structure alignments at the RCSB PDB website	Andreas Prlic
12:30-1:30	***Lunch***	
12:30-2:00	**Poster Session I**	
2:00-3:30	**Session: Genome-scale Data Management**	Chair: Ronald Taylor
2:00-2:15	Using HDF5 to work with large quantities of biological data	Dana Robinson
2:15-2:30	Large scale data management in Chipster 2 workflow environment	Aleksi Kallio
2:30-2:45	JBrowse 2012	Ian Holmes
2:45-3:00	Khmer: A probabilistic approach for efficient counting of k-mers	Qingpeng Zhang
3:00-3:15	AmiGO 2: a document-oriented approach to ontology software and escaping the heartache of an SQL backend	Seth Carbon
3:15-3:30	Discovery of motif-based regulatory signatures in whole genome methylation experiments	Jens Lichtenberg
3:30-4:00	***Coffee Break***	

Time	Title	Speaker or Session Chair
4:00-5:30	**Session: Linked Data and Translational Knowledge Discovery**	Chair: Jan Aerts
4:00-4:25	The open source ISA metadata tracking framework: from data curation and management at the source, to the linked data universe	Philippe Rocca-Serra
4:25-4:50	Automated infrastructure for custom variant comparison and analysis	Brad Chapman
4:50-5:05	KUPKB: Sharing, Connecting and Exposing Kidney and Urinary Knowledge using RDF and OWL	Julie Klein
5:05-5:20	eagle-i: development and expansion of a scientific resource discovery network	Sophia Cheng
5:20-5:30	Report on Codefest 2012	Brad Chapman
5:30-5:40	Introduction to the Open Bioinformatics Foundation (O|B|F)	Hilmar Lapp (President, O|B|F)
5:40-6:40	**Poster Session II**	
5:40-6:40	**BOFs**	
7:00	Pay-your-own-way BOSC dinner ($25) RSVP at http://bit.ly/BOSC2012-dinner	George's Greek Café, 135 Pine Ave.

Day 2 (Saturday, July 14, 2012)

Time	Title	Speaker or Session Chair
8:45-8:50	**Announcements**	Nomi Harris
8:50-9:50	**Keynote: If I Build It Will They Come?**	Carole Goble
9:50-10:15	Pistoia Alliance Sequence Squeeze: Using a competition model to spur development of novel open-source algorithms	Richard Holland
10:15-10:45	***Coffee Break***	
10:45-12:30	**Session: Software Interoperability**	Chair: Hilmar Lapp
10:45-11:00	GenomeSpace: An open source environment for frictionless bioinformatics	Michael Reich
11:00-11:15	Galaxy Project Update	Dannon Baker
11:15-11:30	Zero to a Bioinformatics Analysis Platform in 4 Minutes	Enis Afgan
11:30-11:55	PyPedia: A python crowdsourcing development environment for bioinformatics and computational biology	Alexandros Kanterakis

Time	Title	Speaker or Session Chair
11:55-12:10	InterMine - Embeddable Data-Mining Components	Alexis Kalderimis
12:10-12:25	Creating biology pipelines with BioUno	Bruno Kinoshita
12:30-1:30	***Lunch***	
12:30-1:45	**Poster Session III**	
1:45-3:30	**Session: Bioinformatics Open Source Project Updates**	Chair: Jeremy Goecks
1:45-2:00	Mobyle Web Framework: New Features	Hervé Ménager
2:00-2:15	Biopython Project Update	Eric Talevich
2:15-2:40	Biogem,Ruby UCSC API, and BioRuby	Hiroyuki Mishima
2:40-2:55	A Framework for Interactive Visual Analysis of NGS Data using Galaxy	Jeremy Goecks
2:55-3:10	Why Scientists Should Contribute to Wikipedia	Spencer Bliven
3:10-3:25	scabio - a framework for bioinformatics algorithms in Scala	Markus Gumbel
3:30-4:00	***Coffee Break***	
4:00-4:30	**Session: Lightning Talks**	Chair: Kam Dahlquist
4:00-4:07	bioKepler: A Comprehensive Bioinformatics Scientific Workflow Module for Distributed Analysis of Large-Scale Biological Data	Jianwu Wang
4:07-4:14	(Last-minute lightning talk)	
4:14-4:21	(Last-minute lightning talk)	
4:21-4:28	Bioinformatics Testing Consortium: Codebase peer-review to improve robustness of bioinformatic pipelines	Ben Temperton
4:30-5:30	**Panel: Bioinformatics Paper Reviews--Openness of Standards and Standards of Openness**	*Moderator*: Brad Chapman *Panelists*: Titus Brown, Iain Hrynaszkiewicz, Hilmar Lapp, Scott Markel, Ben Temperton
5:30-5:40	Presentation of awards	Nomi Harris
5:40-6:40	**BOFs**	

Any last-minute schedule updates will be posted at http://www.open-bio.org/wiki/BOSC_2012_Schedule

Keynote Speakers

Jonathan Eisen

Dr. Eisen is a professor at the University of California, Davis, where he holds appointments in the Genome Center, the Department of Evolution and Ecology and the Department of Medical Microbiology and Immunology. In addition, he has an adjunct position at the Department of Energy Joint Genome Institute in Walnut Creek, CA. Prior to moving to UC Davis he was on the faculty at The Institute for Genomic Research (TIGR) for eight years. His research focuses on the genomic basis for the origin of novelty (how new processes and functions originate), in particular in microbes. Dr. Eisen is heavily involved in the Open Access publishing movement and is Academic Editor in Chief of PLoS Biology. He is also an active blogger and microblogger (e.g., see phylogenomics.blogspot.com and twitter.com/phylogenomics).

His talk is entitled *Science Wants to Be Open - If Only We Could Get Out of Its Way.*

> Scientific research and education is inherently an open activity. Yet the culture of scientific practice has inserted barriers in the way of this openness in every conceivable area from peer review, to publishing, to sharing resources, to education. I will argue that most or even all of these barriers are unnecessary and should be eliminated for scientific progress to be most efficient.

Carole Goble

Carole Goble is a full professor in the School of Computer Science at the University of Manchester, UK, where she co-leads the Information Management Group. She has an international reputation in Semantic Web, distributed computing, and social computing for scientific collaboration. She is the Director of the myGrid project, which has produced the widely-used Taverna open source software; myExperiment, a social web site that enables researchers to share scientific workflows; and the BioCatalogue of web services for the life sciences.

In 2008 Carole was awarded the inaugural Microsoft Jim Gray award for outstanding contributions to e-Science. In 2010 she was elected a Fellow of the Royal Academy of Engineering for her contributions to e-Science. In 2012 she was nominated for the Benjamin Franklin award for open science in Biology.

Carole's talk is entitled *If I build it will they come?*

> Over the years I have built a bunch of open source software and services for researchers: the Taverna workflow system, myExperiment for workflow sharing, BioCatalogue for services, SEEK for Systems Biology data and models, and most recently MethodBox for longitudinal data sets. As well as building software we built communities: development communities and user communities. So what drives/hinders adoption? What do I know now that I wished I had known before? How do we sustain communities on time-limited grants? How do we build it so they come, stay and join in?

Optional BOSC Dinner

We invite you to join BOSC organizers and attendees for dinner the first evening of BOSC (Friday, July 13, at 7pm) at a local restaurant: George's Greek Café, located at 135 Pine Avenue, a short walk from the convention center. For $25 per person (payable at the restaurant) gets you a lavish buffet of Greek appetizers and entrees (meat and vegetarian), and also includes unlimited soft drinks and/or coffee, as well as tax and tip. There will be an additional charge for any alcoholic drinks and desserts.

RSVP for the dinner at http://bit.ly/BOSC2012-dinner before Friday at noon.

O|B|F Membership

Professionals, scientists, students, and others active in the Open Source Software arena in the life sciences are invited to join the Open Bioinformatics Foundation (the O|B|F). The membership body was formally established at the 2005 Board of Directors meeting. As laid out in the bylaws, officers in the Board of Directors are elected by the membership among nominees, and candidates for future Directors will be nominated from the membership when seats are added or a term expires.

The eligibility criteria are met by anyone who is "interested in the objectives of the OBF", and there are no dues at present. You can join the O|B|F at BOSC by filling out the application form included in this program, signing it, and giving it to a Board member. You may also e-mail the scanned form to the current President, Hilmar Lapp, at hlapp@drycafe.net. (The O|B|F in its current form of incorporation is legally required to have signatures on record for all members.)

If you are interested in meeting and talking to some of the O|B|F Directors and members, please join us at the BOSC dinner (see above).

Talk and Poster Abstracts

Talk abstracts are included in this program in the order in which they will be presented at the conference. Some, but not all, of the talks will also be presented as posters. Posters that were submitted before the deadline are listed on the next page. There is also space available for last-minute posters. The ISMB staff specify that posters should not exceed the following dimensions: 46 inches wide by 45 inches high.

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.

Posters

Number	Poster Title and Author
1	Cloudgene - an execution platform for MapReduce programs in public and private clouds (Sebastian Schoenherr)
2	Data reduction and division approaches for assembling short-read data in the cloud (C. Titus Brown)
3	Workflows on the Cloud: Scaling for National Service (Katy Wolstencroft)
4	Using HDF5 to Work With Large Quantities of Biological Data (Dana Robinson)
5	Large scale data management in Chipster 2 workflow environment (Aleksi Kallio)
6	Khmer: A probabilistic approach for efficient counting of k-mers (Qingpeng Zhang)
7	Discovery of motif-based regulatory signatures in whole genome methylation experiments (Jens Lichtenberg)
8	Pistoia Alliance Sequence Squeeze: Using a competition model to spur development of novel open-source algorithms (Richard Holland)
9	GenomeSpace: An open source environment for frictionless bioinformatics (Michael Reich)
10	Zero to a Bioinformatics Analysis Platform in 4 Minutes (Enis Afgan)
11	PyPedia: A python crowdsourcing development environment for bioinformatics and computational biology (Alexandros Kanterakis)
12	InterMine - Embeddable Data-Mining Components (Alexis Kalderimis)
13	Creating biology pipelines with BioUno (Bruno Kinoshita)
14	Mobyle Web Framework: New Features (Hervé Ménager)
15	Biogem,Ruby UCSC API, and BioRuby (Hiroyuki Mishima)
16	Why Scientists Should Contribute to Wikipedia (Spencer Bliven)
17	scabio - a framework for bioinformatics algorithms in Scala (Markus Gumbel)
18	bioKepler: A Comprehensive Bioinformatics Scientific Workflow Module for Distributed Analysis of Large-Scale Biological Data (Jianwu Wang)
19	Bioinformatics Testing Consortium: Codebase peer-review to improve robustness of bioinformatic pipelines (Ben Temperton)
20-25	*Walk-in posters*

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

BOSC 2012

Talk and Poster Abstracts

Cloudgene - an execution platform for MapReduce programs in public and private clouds

L. Forer[1,2*], S. Schönherr[1,2,*], H. Weißensteiner[1,2], F. Kronenberg[1], G. Specht[2], A. Kloss-Brandstätter[1]

1 Division of Genetic Epidemiology; Department of Medical Genetics, Molecular and Clinical Pharmacology;
Innsbruck Medical University, Innsbruck, Austria

2 Department of Database and Information Systems; Institute of Computer Science;
University of Innsbruck, Innsbruck, Austria

*** contributed equally**

The MapReduce framework enables a scalable processing and analysing of large datasets by distributing the computational load on connected computer nodes, referred to as a cluster. The user is responsible to write the corresponding map and reduce task of an application and the framework itself is taking over the parallelization, fault tolerance of hardware and software and I/O scheduling. In Bioinformatics, MapReduce and especially its open-source implementation Hadoop [1] has been adopted to map next generation sequencing data to a reference genome [2], finding SNPs from short read data [3] or calculating differential gene expression in large RNA-Seq datasets [4]. Nevertheless, the execution of MapReduce jobs still includes non-trivial tasks like (1) installing and maintaining Hadoop on a cluster system, (2) importing data into the distributed file system HDFS, (3) executing jobs (often via the command line) and (4) exporting results to the local file system.

Here, we present an update on Cloudgene [5], a platform to improve the usability of MapReduce jobs by providing a graphical web interface for their execution, the fast import and export of data and the reproducibility of workflows. We show on different use cases that MapReduce programs can be integrated by writing a simple YAML configuration or by installing it from a provided web repository. Furthermore, Cloudgene allows the execution of Map/Reduce streaming jobs and job pipelines, by defining "steps" in the configuration. Data can be imported from S3/FTP/HTTP and a history of executed jobs with defined input/output parameters and results is accessible. For public clouds like Amazon, Cloudgene sets up user-defined cluster architectures, installs the Cloudgene web interface and all necessary data on it, and allows a simple processing directly in the cloud. Before a shutdown is fulfilled, all data is exported to a predefined S3 bucket. Compared to similar approaches like Elastic MapReduce (EMR), Cloudgene can be started on every private in-house cluster, avoiding any data transfer to the cloud in case of large or sensitive data. Currently six different MapReduce programs have already been integrated into Cloudgene. In our presentation we will give an overview on Cloudgene and demonstrate on a data intensive use case (a developed in-house solution for FastQ pre-processing) how an integration and execution works.

Project site: **http://cloudgene.uibk.ac.at**
Source code: **http://cloudgene.uibk.ac.at/downloads.html**
License: GNU GPL v3

[1] Apache Hadoop. http://hadoop.apache.org/

[2] Langmead B, Schatz MC, Lin J, Pop M, Salzberg SL. Searching for SNPs with cloud computing. Genome Biol 10:R134.

[3] Schatz MC. CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics, 25:1363–1369, Jun 2009.

[4] Langmead B, Hansen K, Leek J. Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biology 11:R83.

[5] Forer L, Schoenherr S, et al. Cloudgene. Poster presentation at BOSC 2011, Vienna, Austria

Data reduction and division approaches for assembling short-read data in the cloud

Adina Howe, Jason Pell, Qingpeng Zhang, Arend Hintze, Eric McDonald, and C. Titus Brown
Departments of Computer Science and Engineering / Microbiology and Molecular Genetics, Michigan State University, East Lansing, MI, 48824

E-mail: ctb@msu.edu

Software tutorial: http://ged.msu.edu/angus/
Software available at: http://github.com/ctb/khmer/
Available under the BSD License

De novo sequence assembly is one of the most common memory intensive tasks performed in bioinformatics. This limits our ability to use de novo assembly on cloud computing hardware, which is often memory constrained. We have developed an extensive suite of k-mer based filtering approaches, including digital normalization and partitioning, that enable the efficient and effective assembly of a wide variety of data sets in relatively low memory. We have applied these approaches to shotgun sequences from microbial genomes, MD-amplified single-cells, eukaryotic transcriptomes, eukaryotic genomes, and metagenomic data sets from the Human Microbiome Project.. After filtering, we typically achieve equivalent or better-than-equivalent assemblies with Velvet, while decreasing compute and memory requirements by factors of 10-100x. This allows significant assemblies to be performed on single cloud machines with generally available assemblers. I will discuss our current approaches and our future plans for this software.

All of our tools are freely available under an Open Source license (BSD) and are developed on github under github.com/ctb/khmer/.

MGTAXA – a toolkit and a Web server for predicting taxonomy of the metagenomic sequences with Galaxy front-end and parallel computational back-end

Andrey Tovchigrechko, Seung-Jin Sul, Timothy Prindle, Shannon J. Williamson and Shibu Yooseph
J. Craig Venter Institute, Rockville, MD, USA; atovtchi@jcvi.org

Website: http://mgtaxa.jcvi.org/
Source: http://andreyto.github.com/mgtaxa/
License: GPLv3

We describe the implementation of a software package and a free public computational Web server for predicting taxonomy of metagenomic sequences based on the nucleotide sequence composition for all domains of life as well as viruses. We describe a novel algorithm for selecting putative hosts for the bacteriophages represented by metagenomic contigs. The Web server component uses the popular and feature-rich Galaxy bioinformatics workbench as its Web front-end and integrates some of the standard Galaxy tools to help in pre- and post-processing of the taxonomic predictions. It provides a customized Galaxy job runner for the Gridway metascheduler, as well as a new middle access driver for the Gridway to provide an extra layer of security validation for the externally submitted jobs.

To perform high-throughput computations, the software implements a coarse-grained data-parallel backend where each single front-end user job generates a complex workflow of multiple batch jobs running in parallel on the compute cluster. The prediction method is based on the PhymmBL algorithm, extended and benchmarked here for viruses and eukaryotes. The server allows users to train additional models on their own reference sequences and combine these models with the main database of models trained on publicly available genomes.

The benchmarking of the novel host prediction algorithm was performed on 5Kbp-long contigs simulated from the NCBI viral RefSeq while selecting hosts from all available RefSeq prokaryotic genomes. On lysogenic phages, it resulted in 97% accuracy at the phylum level and 89% at the genus level of the predicted host taxonomy with 2% rejection rate of the testing samples, and on all phages regardless of the proliferation cycle - in 76% accuracy at the phylum level and 50% at the genus level after rejecting with 6% rejection rate.

All components of the framework including the server and the underlying computational methodology are released in open source.

The algorithm is implemented in Python and C++. It uses HDF5/PyTables to efficiently store and manipulate its working datasets. The outputs are tab-delimited files for downstream analysis as well as PDF and HTML5 Krona charts. Predictions are the assigned taxonomic lineages for every input sequence with the associated confidence scores.

The source code package also implements auxiliary tools such as a CRISPR annotation pipeline, a module to automatically build a database of known virus-host pairs based on the NCBI RefSeq data and a module to assign taxonomy to long metagenomic contigs based on a lowest common ancestor majority voting of BLASTP ORF matches.

Workflows on the Cloud – Scaling for National Service

Robert Haines, Helen Hulme, Shoaib Sufi, Katy Wolstencroft, Andy Brass and Carole Goble, *School of Computer Science, University of Manchester, UK*
{robert.haines, shoaib.sufi, helen.hulme, katherine.wolstencroft, andy.brass, carole.goble}@manchester.ac.uk
Madhu Donepudi, Nick James, *Eagle Genomics Ltd, UK*
{madhu.donepudi, nick.james}@eaglegenomics.com

Project site: http://www.taverna.org.uk/
Source code: http://taverna.googlecode.com/svn/taverna/products/uk.org.taverna.server/trunk
License: GNU Lesser General Public License (LGPL) 2.1

With the expansion of Next Generation Sequence (NGS) techniques and advances in clinical genetic testing, the UK National Health Service (NHS) has increasing need of fast, secure and accurate, disease-specific methods for analysing patient data. Single Nucleotide Polymorphisms (SNP) and other Genetic variation data require processing, annotating and filtering in preparation for interpretation by specialist clinicians.

Annotation tasks include collecting information from public databases and from tools that give estimates of deleterious effect (such as SIFT and Polyphen). Filtering includes patient consent constraints and criteria based on estimated severity of effect, as well as frequencies of occurrence of the variation in, for example, the 1000genomes project. The size of data involved is typically in the region of tens of thousands of data points per patient. The highly parallel nature of the task makes it an ideal candidate for using cloud computing. Also, as the tool is for use in a clinical context, provenance, reusability and reliability are key considerations, and the filtering system must be both track-able and scientifically rigorous.

Our approach is to use the Taverna workflow system to integrate information from public databases, and on-the-fly analysis tools, capturing provenance data and filtering SNPs, to give a concise, human readable report on those variations most likely to have clinical significance. This work uses the Cloud Analytics for Life Sciences (CA4LS) platform that is built on top of the Taverna server and the Amazon Cloud. The platform provides access to curated, certified workflows as applications that can be executed via a Web browser.

Through the web browser interface, users can choose analyses to run from a set of tested, optimized and certified workflows. Input data, such as SNP lists, and configuration options, such as filtering criteria, can be added through the web interface. The system then validates the provided inputs and submits the workflow instance to the workflow engine on the cloud for execution.

The workflows are deployed in the Amazon Cloud along with all of their dependencies, even though many of the services we use are available via public interfaces. We do this for a number of reasons:

- To track exactly which version of each service and each database that we use (for QA certified workflows)
- To minimise the latency when calling these Web Services.
- To avoid degrading the performance of public Web Services and/or breaking usage agreements.

The Cloud orchestration layer of the architecture is responsible for monitoring the resources in use and scaling them where required. Most workflows can be scaled in two ways. Firstly, the data can be divided up between multiple instances of the workflow. This is especially true of most Next Generation Sequencing pipelines due to the fact that individual SNPs can be processed entirely independently of one another. Secondly, the services used in the workflows can be scaled. The automatic scaling features of the Cloud provider can detect when the virtual machines hosting our services are getting close to capacity and replicate them as required.

Here we will introduce the CA4LS platform and architecture by presenting a motivating use case from the NHS. This will demonstrate how the system securely and efficiently analyses such data and thus scales to meet analytical demand.

Funded by: Technology Strategy Board-Harnessing Large and Diverse Sources of data–100932–Cloud Analytics for Life Sciences

How to use BioJava to calculate one billion protein structure alignments at the RCSB PDB website

Andreas Prlić[1], Spencer Bliven[2], Peter W. Rose[1], the BioJava development team, Philip E. Bourne[4]

[1]San Diego Supercomputer Center, and [2] Bioinformatics Program, University of California San Diego, La Jolla, California, USA. [3],http://www.biojava.org [4], Skaggs School of Pharmacy and Pharmaceutical Sciences University of California San Diego, La Jolla, California, USA *andreas.prlic@gmail.com

BioJava is an open-source project dedicated to providing a Java framework for processing biological data. It provides analytical and statistical routines, parsers for common file formats and allows the manipulation of sequences and 3D structures. The goal of the BioJava project is to facilitate rapid application development for bioinformatics. BioJava consists of several modules that are specialized for specific tasks.

The protein structure modules offer a flexible framework for structural biology. A biologically and chemically meaningful data representation provides a framework for developing protein structure algorithms. Built on top of this, open source implementations are available for some of the most widely used algorithms for protein structure alignment, namely CE and FATCAT. The RCSB PDB website uses these at the core of its Protein Comparison Tool.

Here we describe how we use BioJava for a systematic comparison of representative protein domains in the RCSB Protein Data Bank (PDB). With the continuous growth of the PDB, providing an up-to-date systematic structure comparison of all protein structures poses an ever growing challenge. We present a solution for distributing a large number of alignment calculations across virtual compute clouds. We use this approach to run 1 billion structure alignments at the OpenScienceGrid (OSG), consuming 260,000 CPU hours. The system is flexible enough to run across different data centers. After the initial bulk calculations run at the OSG, weekly incremental updates are computed using RCSB PDB specific in-house compute resources.

The results are made available as part of the RCSB PDB website on the 3D-similarity report for each structure. It lists related proteins that share structural similarity in a tabular display. This allows the discovery of novel relationships between proteins. The results are freely available for download.

Availability: http://www.biojava.org, http://www.rcsb.org, http://source.rcsb.org
License: LGPL 2.1, CC-BY

USING HDF5 TO WORK WITH LARGE QUANTITIES OF BIOLOGICAL DATA

Dana Robinson (derobins@hdfgroup.org)

The HDF Group (http://www.hdfgroup.org)

HDF5 web site: (http://www.hdfgroup.org/HDF5/)

HDF5 source code: (http://www.hdfgroup.org/HDF5/release/obtain5.html)

License: BSD

The HDF5 technology platform allows users to organize, store, share, and access large and complicated data. It consists of a data model, file format, library (C/C++/Java), and tools. HDF5 is used worldwide by government, industry, and academia in a wide range of science, engineering, and business disciplines. Prominent users include MathWorks (Matlab can read HDF5 files), NASA (HDF-EOS5), and Applied Biosystems (primary image data storage).

HDF5 combines the flexible and extensible data layout of a database with the portability and ease of access of individual files. As biology becomes more data-driven, with data sets of ever-increasing size, HDF5 can provide a way for researchers to work with their data via a high-performance, scalable, platform-independent technology suite.

Some aspects of HDF5:

- Facilities for item association, hierarchies, and annotation
- Flexible user-defined types
- Files are self-contained and self-describing
- Portable across platforms and architectures
- High I/O performance, parallel I/O
- Out-of-core data access (partial I/O)
- Unlimited file size support
- Support for compression and other custom filters
- Suitable for long-term data archiving
- Free and open source (BSD license)
- High-quality support, training, and documentation

In this talk, we will give a brief introduction to HDF5 and its capabilities, followed by some examples of how biological data can be stored in HDF5. Though there will probably not be time for a Q&A session given the short duration of the talks, the presenter will be available for questions throughout the conference.

There has been a good deal of interest in HDF5 in the life sciences. We want to support that interest any way we can, including working with communities that wish to adopt HDF5. Please talk to us about your data needs!

Large scale data management in Chipster 2 workflow environment

Aleksi Kallio (aleksi.kallio@csc.fi), Taavi Hupponen, Petri Klemelä, Mikael Karlsson,
Massimiliano Gentile, Eija Korpelainen / CSC – IT Center for Science, Finland
Web site: http://chipster.sourceforge.net, source code: http://code.google.com/p/chipster/source/

Chipster is a user friendly and cloud compatible data analysis environment for high-throughput biomedical data. As next-generation sequencing methods redefined the size of data that is being moved around and processed by the bioinformatics pipelines, it also required analysis environments such as Chipster to be extended.

Most importantly, Chipster was added a new concept: remote sessions. Previously sessions allowed the user to store the entire state of the analysis work into a compressed file, to continue work from or to share with colleagues. Remote sessions offer the same functionality, but data is stored to the server environment and not pulled to user's computer. Remote data storage exposes the user to the hurdles of remote data management: storage space need to be thought of and data removed to free space. In our approach, data is moved to long term storage only when a session is saved explicitly. We feel that this solution fits well the natural workflow of our users and decreases the effort required for data pruning.

On the server side, the file broker nodes of the system have two storage areas: cache and long term. The areas have different life cycle management strategies, allowing the file brokers to use disk space efficiently. Disk space management can be controlled via the file broker configuration and we are developing command line and web based administration tools also for centralised monitoring and management of disk usage.

Chipster data infrastructure was originally built around the principle of immutable files, which allows efficient caching and easy data sharing. However for big data another design principle was also required: move data only when absolutely necessary. All data is passed around the system as URL references and actual data is copied only when it is required for running analysis tools. If two components have a shared disk area, they can by-pass the network transfer and use data directly. If the system is installed to a data producing facility, it can also be configured to directly access raw data from a shared disk area.

To facilitate larger and faster analysis runs of NGS data, we have also been actively involved in development of Hadoop based tools (Hadoop-BAM, SeqPig). Major future development area will be interoperability between various systems. Our goal is to allow users to easily move data in and out from external data platforms, such as iRODS, implemented so that backend services talk directly to data providers.

Title: JBrowse

Authors: Robert Buels, Ian Holmes
Author affiliations: Department of Bioengineering, University of California, Berkeley
<ihh@berkeley.edu>

URL: http://jbrowse.org/
Code URL: https://github.com/GMOD/jbrowse
License: LGPL / Artistic

Abstract

JavaScript is coming of age as a platform, and JBrowse is coming of age as a genome browser. Dynamic HTML opens up exciting possibilities such as "faceted browsing", whereby complex Boolean queries can be interactively explored from within the web browser. JBrowse now incorporates faceted browsing into its track selector, so that previously rich-but-overwhelming lists of tracks (e.g. spanning multiple experimental protocols, computational analyses, cell lines, and collaborative groups) can be drilled into in real time. This is just one of several new user interface features in JBrowse - others include a help button (leading to full instructions) and a generally updated "look and feel".

As well as up-front user interface enhancements, one of the more significant improvements to JBrowse is enhanced scalability, achieved via 3-fold on-disk compression (among other optimizations). Many aspects of installation have been streamlined, including automatic dependency installation. Numerous bugs have been fixed and a substantial testing framework has been added.

In the talk we will discuss some of these in more detail as well as outlining some of our future plans, including a flexible event framework for customization of the browser client, and a highly adaptable data back-end on the server.

Khmer: A probabilistic approach for efficient counting of k-mers

Qingpeng Zhang, Jason Pell, Rose Canino-Koning, Adina Chuang Howe, C. Titus Brown
Department of Computer Science and Engineering
Michigan State University, East lansing, MI, 48824

Email: qingpeng@msu.edu
Software tutorial: http://ged.msu.edu/angus/
Software available at: http://github.com/ctb/khmer/
Available under the BSD License

K-mer counting has been widely used in many bioinformatics problems, including data preprocessing for de novo assembly, repeat detection, sequencing coverage estimation. However current available tools can not handle the high throughout data generated by next generation sequencing technology efficiently due to high memory requirements or impractically long running time. Here we present the khmer software package for fast and memory efficient counting of k-mers. Unlike previous methods bases on data structures including hash tables, suffix arrays, or trie structures, Khmer uses a simple probabilistic data structure, which is similar in concept to the CountMin Sketch data structure. It is highly scalable, effective and efficient in applications involving k-mer counting to analyze large next generation sequencing dataset, despite with certain false positive rate as tradeoff. We compared the memory usage, disk usage and time usage between our khmer software and other methods like tallymer and jellyfish to show the advantage of our method. The counting accuracy was also assessed theoretically and was validated using simulated data sets. We further showed applications of khmer software package in tackling problems like detecting sequencing errors in metagenomic reads, removing those erroneous reads to reduce data set size and repeat sequence discovery through efficient k-mer counting.

The Khmer software package is freely available under an Open Source license (BSD) and are developed on github under github.com/ctb/khmer/.

AmiGO 2: a document-oriented approach to ontology software and escaping the heartache of an SQL backend

Seth Carbon[1], Christopher J. Mungall[1], Heiko Dietze[1], Shahid Manzoor[2], Gene Ontology Consortium[3]

[1]Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA.
{sjcarbon, cjmungall, hdietze}@lbl.gov
[2]shahid.manzoor@gmail.com
[3]Gene Ontology Consortium
gohelp@geneontology.org

Website: `http://amigo2.berkeleybop.org`
Source: `http://geneontology.svn.sourceforge.net`
License: Modified BSD (`https://www.gnu.org/licenses/license-list.html#ModifiedBSD`)

AmiGO 2 is an open-source web application that allows users to query, browse, and visualize ontologies and related gene product annotation data. After years of using an SQL backend for its core--with an increasing number of tricks, extensions, and caches to keep the performance at an acceptable level--AmiGO development has turned towards Solr (a specialized HTTP server over the Lucene document store) for a document-oriented approach to handling core data and services. By making this change, AmiGO has greatly increased its features, flexibility, speed, and development turnaround time.

Complicated data in a flat document

While relational databases have long been used for storing the data necessary to run applications, in some cases they may not be as fast or as useful as a document store. While there is less possible structure in the field/value layout of our document store, thoughtful field creation and the use of a large number of simpler structures can cover most of the ground previously covered by relational databases. For example, common methods of modeling a graph (in our case the Gene Ontology) in a relational database require many joins and can become quite time consuming. However, a document store like Solr can still model some graph and structured data, often with extreme speed and usability improvements.

Development

Given that we want the features provided by a document store, there was some extra overhead to be dealt with. However, many backend implementation issues are mitigated by leveraging more off-the-shelf software leaving us with much simplified clients that act mostly as HTTP clients and do far less processing than their predecessors. In addition, Solr aggregate functions like facets and blob storage allow AmiGO 2 to replace complicated cache and storage handling with built-in functionality. Together, all of this decreases development time and client footprint while opening the doors to a whole new range of possible features.

What it got us

When a document store backend is applied to software like AmiGO 2, many operations that were previously time consuming become trivial. An easy example would be that of textual search: where instead of iterating across tables and collecting results in order to rank them outside of the relational database, all of those steps are performed as part of the Solr query. A more involved example would be modeling a gene product's annotation neighborhood in an ontology graph as unordered lists of the multiple graph closures, thereby allowing searches for information such as the number of gene products indirectly annotated to a term to be achieved quickly in a single step.

To leverage this shift to a frontend and backend separated by HTTP, we have created new perl libraries for communicating and processing data from Solr, increased the scope and availability of our JavaScript API, and started offering the Solr index of the Gene Ontology as a publicly available web service. Our goal is to have these services and APIs used by any group interested in working with the Gene Ontology.

Discovery of motif-based regulatory signatures in whole genome methylation experiments

Jens Lichtenberg, Amber Hogart, Stephanie Battle, and David Bodine
National Human Genome Research Institute, Bethesda, MD, 20892, USA
lichtenbergj@mail.nih.gov
http://code.google.com/p/nextgen-signatures (GNU General Public License 3.0 (GPLv3))

No generic framework exists that supports the mining of epigenetic high-throughput sequencing data extracted for different cell types in order to discover regulatory profiles. One epigenetic factor in the regulation of gene expression is the heritable and reversible methylation of cytosine in a cytosine guanine dinucleotide context. In order to study the role of methylation in development it is necessary to generate a complete snapshot of DNA methylation at a given point in a given cell. Using methyl-binding domain proteins (e.g. MBD2), which recognize methylated DNA, and high-throughput sequencing, it is possible to map in an unbiased fashion regions of methylated DNA on a genome wide level. Via a comparison against background sequence reads, peak calling applications (e.g. MACS) determine statistically significant methylated regions (so-called peaks), which enables correlation studies between gene expression, functionality and methylation on a genome-wide as well as within specific genomic partitions. For the purpose of correlating epigenetic marks with mRNA expression profiles for example, it is necessary to transform the data and query an outside database (e.g. BloodExpress). These data can then be used to gather sequences from promoter regions for targeted regulatory element discovery (e.g. WordSeeker). Given the huge number of testable hypotheses generated by the high-throughput sequencing technologies, a comprehensive platform allowing multiple analyses to be performed greatly facilitates data analysis.

Extending the conceptual approach of the EpiGRAPH framework, the SigSeeker approach presented here not only annotates epigenetic peak profiles with functional, regulatory and structural information but also provides a complete regulatory genomics analysis platform that infers regulatory genomic signatures that go hand in hand with the epigenetic profiles. Unlike the submission and resubmission based EpiGRAPH analysis pipeline, SigSeeker is designed to support epigenetic benchwork at a much closer level, allowing for frequent adjustments and parameter modification of various stages of the analysis. SigSeeker integrates high-throughput sequencing data and other available repositories of structural and functional annotations, compiled for a diverse set of biological problems, spanning the complete central dogma of molecular biology. The approach is designed to analyze multiple epigenetic data sets to determine the regulation and expression of genes unique to the cells being analyzed. SigSeeker not only mines high-throughput sequencing data for putative transcription factor binding sites but also integrates these data with databases containing inferred gene sets and functional annotations for the originating cell.

Two different hematopoietic case studies of methylation specific high-throughput sequencing data have been analyzed using SigSeeker. The first study focuses on the differences in the methylation profiles of differentiated hematopoietic stem cells, progenitor cells and fully committed cells following hematopoietic differentiation in *M. musculus*, while the second study analyses the effects of methylation inhibitors on myelomonocytic cells in Chronic Myelomonocytic Leukemia in *H. sapiens* at different time points during a chemotherapy treatment cycle. For both studies strong agreement with the existing body of literature could be found, as well as novel insights that were validated experimentally. These results demonstrate the applicability of the tool to large-scale systems biological problems and its use for the generation of targeted hypotheses for follow-up studies.

The source code for the pipeline presented here is made available under the GNU General Public License, version 3.0 (GPLv3) through the Google Project Hosting: http://code.google.com/p/nextgen-signatures

The open source ISA metadata tracking framework: from data curation and management at the source, to the linked data universe

Philippe Rocca-Serra, Eamonn Maguire, Alejandra Gonzalez-Beltran
and Susanna-Assunta Sansone, on behalf of the ISA community

University of Oxford, Oxford e-Research Center, Oxford, UK
isatools@googlegroups.com

Project website and community: http://www.isa-tools.org and http://www.isacommons.org
Code: https://github.com/ISA-tools
Open Source License: Mozilla Public License

Increased availability of the data generated is fuelling increased consumption, and a cascade of derived datasets that accelerate the cycle of discovery [1]. But the successful integration of heterogeneous data from multiple providers and scientific domains is already a major challenge within academia and industry [2]. Even when datasets are publicly available, published results are often not reusable due to incomplete description of the experimental details. Minimum reporting guidelines, terminologies and formats (referred to generally as community standards) are increasingly used in the structuring and curation of datasets, enabling data annotation to varying degrees. But in practice, achieving compliance is challenging, also because of the current wealth of domain-specific reporting standards, or their incompleteness and absence in other areas (www.biosharing.org).

In this unsettled status quo, how can we enable researchers to make use of existing community standards and maximize sharing and their subsequent reuse of richly annotated experimental information? A successful example is provided by the Investigation/Study/Assay (ISA) [3] open source, metadata tracking framework supported by the growing ISA Commons community [4]. The ISA framework includes both a general-purpose file format and a software suite to tackle the harmonization of the structure of experimental metadata (e.g., provenance of study materials, technology and measurement types, sample-to-data relationships) by enabling compliance with the community standards.

We will present the evolution of this exemplar ecosystem of data curation and sharing solutions - built on the common ISA framework - that serve to collect and manage heterogeneous experimental metadata in an increasingly diverse set of domains including environmental health, environmental genomics, metabolomics, (meta)genomics, proteomics, stem cell discovery, systems biology, transcriptomics and toxicogenomics, but also communities working to characterize nucleic acid structures and to build a library of cellular signatures [e.g. 5]. We will also discuss the experiences learned with usability of the community standards and provide an update on the next steps to use semantic web approaches to make existing knowledge available for linking, querying, and reasoning [6].

1. Field, Sansone, *et al.*, Omics data sharing, ***Science***, 9 (2009)
2. Harland, Larminie, Sansone *et al.*, Empowering Industrial Research With Shared Biomedical Vocabularies, ***Drug Discovery Today***, 16 (2011).
3. Rocca-Serra, *et a*l., ISA software suite, ***Bioinformatics***, 26 (2010).
4. Sansone, Rocca-Serra *et al.*, Toward interoperable bioscience data, ***Nature Genetics***, 27 (2012).
5. Ho Sui *et al.*, The Stem Cell Discovery Engine, ***Nucleic Acids Research***, 40 (D1) (2012)
6. Deus *et al.*, Translating standards into practice - One Semantic Web API for Gene Expression, ***Journal of Biomedical Informatics***, *in press* (2012)

Title	Automated infrastructure for custom variant comparison and analysis
Author	*Brad Chapman*, Oliver Hofmann, Anjana Varadarajan, Justin Zook, Marc Salit, Justin Johnson, Winston Hide
Affiliation	Harvard School of Public Health, EdgeBio, National Institute of Standards and Technology
Contact	bchapman@hsph.harvard.edu
URL	https://github.com/chapmanb/bcbio.variation
License	MIT

High throughput sequencing technologies such as Illumina, SOLiD and Complete Genomics allow researchers to assess variation across entire genomes. With this opportunity comes the challenge of reliably separating real biological changes from false positives. We describe an automated infrastructure, built on top of the Genome Analysis Toolkit (GATK) that simplifies the process of comparing variants from multiple platforms and calling approaches. Highlights of the system include:

- A command line interface that automates all steps of preparation and comparison, driven by an easily configurable YAML specification file.
- Normalization of Variant Call Format (VCF) files, including indel trimming, chromosome reordering and renaming for human comparisons.
- Comparisons between SNPs, indels and complex structural variations.
- High level summary tables of variant associated metrics for concordant and discordant calls, along with easy access to VCF files for downstream analysis.
- N-way comparisons between multiple approaches, enabling detailed resolution of true and false positives.
- Integration with GATK Variant Quality Score Recalibration and other filtering approaches, enabling re-filtering of variants based on comparison results.

The Archon Genomics XPrize presented by Medco, a 10 million dollar competition to sequence 100 human genomes to high accuracy and completeness, leverages this infrastructure to provide scoring and genome preparation for the contest. We will discuss the preparation of reference human genomes — a CEU daughter (NA12878) and YRI father (NA19239) — using four different technologies: Illumina, SOLiD, Complete Genomics and genotyping. These demonstrate the utility of this library for identifying platform specific differences and preparing a combined set of true variants.

At BOSC, we will describe:

- Development details: Coded in the Clojure programing language, the infrastructure leverages biological libraries on the JVM including GATK and Picard.
- Practical usage: Results obtained from comparisons of multiple variant detection technologies on public human genomes.
- Democratization: Efforts to develop an intuitive visual interface for running analysis pipelines and analyzing metrics in large variant files.

KUPKB: Sharing, Connecting and Exposing Kidney and Urinary Knowledge using RDF and OWL

Julie Klein[1,2$], Simon Jupp[3$], Panagiotis Moulos[2], Jean-Loup Bascands[2], Joost Schanstra[2], Robert Stevens[1]

[1] School of Computer Science, University of Manchester, UK.
[2] INSERM U1048, Institute of Cardiovascular and Metabolic Disease, Toulouse, France.
[3] European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD.
[$] Equal contributors.
Presenting author: julie.klein64@gmail.com
Project: http://www.kupkb.org/
Code: http://code.google.com/p/kupkb-dev/ Open Source License: GNU Lesser General Public License

The wealth of information produced by -omics experiments is still largely underexplored, as it is scattered across publications, comes in different formats and versions, and is hidden in figures and supplemental data. Research in the renal field is no exception. Several attempts have been made to make some of the data available via databases on the Web. However, these databases only provide access to subsets of information, are devoted to only one technology of one omics field, and are scattered across the Web. To facilitate the reuse and exploration of omics data in the field of renal research, we present an open-source publicly accessible and updatable repository that we have called the Kidney and Urinary Pathway Knowledge Base (KUPKB).

The KUPKB is built using a Semantic Web approach. It uses an ontology written in the Web Ontology Language (OWL) called the KUP Ontology (KUPO); this organizes the data about KUP entities and from experiments in the KUP domain. The KUPO reuses many existing ontologies, thus capitalizing on the work of others and reducing the effort involved in creating the KUPKB. The KUPKB is stored in the Resource Description Framework (RDF). The KUPKB integrates external resources, such as Human Metabolome Database (HMDB), National Center for Biotechnology Information (NCBI) Gene, Universal Protein Resource (UniProt), MicroCosm, and HomoloGene. Various mappings of identifiers integrate these different data sets into a single queryable resource. This additional knowledge layer provides an opportunity to expand user queries on a given gene to include both the genes and the proteins and also to link the orthologs across different species. An initial set of 220 experiments has been included in the KUPKB. These experiments span different biological levels (miRNA, mRNA, proteins, and metabolites), different techniques, different species, and different sample types. Data derived from open-source databases or published articles in Medline have been extracted manually from the figures or from the supplementary data where possible. To improve data integration, each entity has been represented in the KUPKB by a stable identifier, e.g. NCBI Gene ID for genes and UniProt ID for proteins.

We developed the iKUP browser (http://www.kupkb.org/) to provide a user-friendly interface to browse the RDF data in the KUPKB. It allows users to browse the KUPKB, focus their search on particular proteins or genes in particular locations, with particular functional attributes. More complex queries can be executed using an "Advanced browser", an interface dedicated for the users to query the data without actually having to write complex queries by themselves. The queries are converted into SPARQL Protocol and RDF Query Language (SPARQL). Users can submit data for inclusion in the KUPKB, using a RightField spreadsheet template where spreadsheet cells are constrained by the KUPO so the data can be annotated in a straightforward and reproducible way (see Submit Data tab).

In conclusion, the KUPKB combined with the iKUP browser provides a new resource for the KUP community in which previously scattered information has been integrated into a single repository for exploration. This knowledgebase has already proven to be of help to better to generate new hypotheses in the context of renal pathophysiology with in silico results confirmed in vivo. The KUPKB has been developed using semantic web technologies, used to ensure that the data are described in a way that enables reuse and integration of the data with future applications. This approach provides a showcase for the technology and a working framework that could be readily adapted for other biological domains (e.g. cardiovascular research and cancer).

This work has been funded by e-LICO project, an EU-FP7 Collaborative Project (2009-2012) Theme ICT-4.4: Intelligent Content and Semantics.

eagle-i: development and expansion of a scientific resource discovery network

D. Bourges-Waldegg PhD[a], S. K. Cheng, MS[a], T. Bashor[b], H. R. Frost, MS[a],
M.A. Haendel, PhD[c], C. Torniai, PhD[c] , J.A. McMurry, MPH[a] , D. MacFadden, MS[a]
[a]Harvard Medical School, Boston, MA, [b]Wonder Lake Software, [c]Oregon Health & Science University, Portland, OR

Abstract

eagle-i is an open-source web-based application suite enabling scientists to discover resources across a distributed network. The cornerstone of the eagle-i software is an ontology-driven data model that supports powerful search methods while maintaining flexibility and interoperability through linked data. We discuss our architecture and approach.

Introduction. Researchers in the medical field produce and consume a vast and varied amount of resources, such as cell lines, specialized services and animal models. Most of these existing research resources cannot be readily found using conventional methods. The eagle-i approach has been developed to address the complexities of this longstanding issue. In addition to research resources, groups such as VIVO, NeuroCommons and Chem2BioRDF aim to make other aspects of biomedical information available semantically. By removing barriers to resource discovery, eagle-i is helping scientists find existing resources more easily, thus reducing time-consuming and expensive duplication.

Technology. eagle-i is built around semantic web technologies and follows linked open data (LOD) principles. **The eagle-i architecture** comprises a set of ontology-driven software components deployed at each institution as well as a central search application that communicates with these federated components. At the core of each institutional deployment is an RDF repository.

While best LOD practice is to maintain a single record representing each entity, in a federated deployment model, this can be a challenge as some entities are common across the network. To address this issue, a separate deployment of eagle-i houses all **"global" resources** that have cross-institutional meaning—manufacturer, for instance. Providing a single authoritative version of each of these "global" resources saves effort and reduces inconsistency.

The **eagle-i data collection tool** produces well-structured resource descriptions that include text, annotations with ontology concepts, and links to other resource instances. The tool also provides a workspace for updating and managing resource descriptions individually and in bulk. Choices are dynamically generated for each resource type based on its corresponding ontology class.

The eagle-i search application backend is a Solr-based semantic search framework that enables rich functionality and inferencing to be applied. The search application front-end provides categorical search and synonym expansion as well as an autosuggest feature that provides real-time visibility into available resources and ontology terms.

To promote interoperability, the **eagle-i ontology** (http://bit.ly/yy3r8I) has been developed in ongoing collaboration with numerous other groups, including the OBO Foundry, VIVO and NIF. The ontology-driven application design does pose some challenges: ontologies are built to reflect domain knowledge—whereas applications need to know how to structure and display data to users. We bridge this gap by a) annotating the eagle-i ontology with application-specific information and b) importing only portions of domain ontologies that are relevant for the application. **The eagle-i glossary application,** in addition to providing a reference for users, also provides the ontologists with insight into how the application consumes ontology modifications.

Uniquely identifiable, linked data supports attribution as an incentive to share. Although critical to the advancement of science, resource sharing is under-attributed and not easy to measure systematically. In contrast, references to publications are a recognized metric of scientific importance. Others have proposed implementing a bioresource research impact factor (BRIF)[1] as an incentive to share human bioresources. BRIF could then be used for researchers much like the impact factor is used for journals. Because each resource in eagle-i has a globally-unique URI that can be referenced, eagle-i would be a natural choice for calculating and implementing a BRIF.

In the fall of 2011, eagle-i was released under an **open source (BSD-3) license**. Currently, adopters may build eagle-i from the source code or install from the binary packages. Since eagle-i is designed as a federated system, institutions have technical and administrative autonomy and can choose to make their resources locally discoverable, or globally discoverable at www.eagle-i.net. For more information on joining the network, visit http://open.med.harvard.edu/display/eaglei.

1. The BRIF workshop group, Cambon-Thomsen A et al. The role of a bioresource research impact factor as an incentive to share human bioresources. Nat Genet. 2011 43(6):503-504.

Pistoia Alliance Sequence Squeeze: Using a competition model to spur development of novel open-source algorithms

Richard Holland and Nick Lynch
Pistoia Alliance
richard.holland@eaglegenomics.com

Website: www.sequencesqueeze.org

Code URL: 100+ different locations, all linked from the front page of the website.

Licence: All code is distributed under the BSD-2 licence.

Storing millions of NGS reads and their quality scores uncompressed is impractical, yet current compression technologies are becoming inadequate. There is a need for a new and novel method of compressing sequence reads and their quality scores in a way that preserves 100% of the information whilst achieving much-improved linear (or, even better, non-linear) compression ratios.

The Pistoia Alliance, in the interests of promoting pre-competitive collaboration, put forward a prize fund of US$15,000 to the best novel open-source NGS compression algorithm submitted before the closing date of 15 March 2012. The winner of this Sequence Squeeze competition was announced and the prize awarded at the Pistoia Alliance Conference in Boston MA on 24 April 2012.

This talk will discuss the use of a competition model to spur innovation and development of new open-source solutions to industry bioinformatics problems. It will also look at the specific problem that the Sequence Squeeze contest addressed (FASTQ compression) and discuss technical aspects of some of the submissions made.

Finally this talk will review the selection of appropriate criteria for judging such contests and how the trade-off has to be made between performance (e.g. memory usage or compression ratio) vs. the requirements of typical (conflicting) real-world use-cases.

GenomeSpace: An environment for frictionless bioinformatics

Michael Reich, Ted Liefeld, Helga Thorvaldsdottir, Marco Ocana, Eliot Polk, Jill P. Mesirov
Broad Institute of MIT and Harvard, mreich@broadinstitute.org

Web site:	http://www.genomespace.org
Repository:	https://bitbucket.org/GenomeSpace/combined
License:	LGPL

GenomeSpace is an open source software environment that provides a connection layer between bioinformatics resources, whether they are Web-based applications, desktop packages, or simple scripts. GenomeSpace addresses the growing need for genomics researchers and bioinformaticians to have “frictionless” data transfer among the variety of analysis tools and data sources. GenomeSpace provides an open environment, which other bioinformatics resources can use to join the community of GenomeSpace-enabled tools. GenomeSpace is seeded by six prominent tools for genomics analysis: Cytoscape, Galaxy, GenePattern, Genomica, the Integrative Genomics Viewer (IGV), and the UCSC Genome Browser, and developed in collaboration with several biological research projects at the Broad Institute, Stanford University, and UCSD.

Galaxy Project Update

Dannon Baker (dannon.baker@emory.edu)[1], James Taylor[1], Anton Nekrutenko[2], The Galaxy Team[3]
[1]Departments of Biology and Math & Computer Sciences, Emory University
[2]Center for Comparative Genomics and Bioinformatics, Penn State University
[f3]http://galaxyproject.org

Website: http://galaxyproject.org
Code: http://bitbucket.org/galaxy/galaxy-central/
License: Academic Free License

Galaxy[1,2] is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research. In this talk we present an update on the current status of the Galaxy Project, focusing primarily on three particularly active areas of development. We cover infrastructure development efforts related to rearchitecting Galaxy in order to expose a fully capable RESTful API, a more tightly coupled cloud integration within the primary Galaxy interface, and efforts related to the transparent parallelization of common tools.

During the past year, we have undertaken a major effort to rearchitect Galaxy with a strong focus on RESTful services. The entire interface is being transitioned to use a single service layer that, while it does provide significant benefits to the core Galaxy interface, is also available via direct API calls. This allows developers to much more easily integrate Galaxy with both external applications and other instances of Galaxy.

Managing cloud computing resources can be a complicated task preventing novice users from taking advantage of significant computational infrastructure. We have created a seamless environment in which from a single Galaxy instance you can spawn additional instances in the cloud (specifically EC2 at this writing) to which your data can be automatically transferred. For example, this allows a user on the public Galaxy server at usegalaxy.org to easily migrate analyses to their own personal cloud instance on EC2 for further processing, or perhaps for integration with private data that cannot be copied to a public resource.

Lastly, we discuss efforts to implement a FUSE-based filesystem for rapid partitioning of datasets for analysis. This allows quick virtual subsetting of large datasets into ‘views’, or smaller files for use by external tools without copying any data into new files, reducing the analysis I/O burden and space required. For example, to parallelize a single BWA run 10x in Galaxy it is necessary for Galaxy to preprocess the original input file to create 10 individual files, passing each subset file to a single instance of BWA running on the cluster. This doubles the space required for input datasets, to say nothing of the I/O cost of actually duplicating all the data. Using FUSE, we are able to create a filesystem on the fly of 10 entries that, in this case, point directly to the offsets of the original file without any duplication.

References:

[1] Goecks, J, Nekrutenko, A, Taylor, J and The Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 2010 Aug 25;11(8):R86.

[2] Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J. "Galaxy: a web-based genome analysis tool for experimentalists". Current Protocols in Molecular Biology. 2010 Jan; Chapter 19:Unit 19.10.1-21.

Title: Zero to a Bioinformatics Analysis Platform in Four Minutes

Authors: Enis Afgan[1,2], Brad Chapman[3], Konstantinos Krampis[4], James Taylor[5]
Author affiliations:
[1]Center for Informatics and Computing, Ruđer Bošković Institute (RBI)
[2]VLSCI Life Science Computation Centre, University of Melbourne
{E.A. email: enis.afgan@irb.hr}
[3]Harvard School of Public Health, Bioinformatics Core
[4]J.Craig Venter Institute (JCVI)
[5]Department of Biology and Department of Mathematics & Computer Science, Emory University

Projects' websites: http://cloudbiolinux.org; http://usecloudman.org; http://biocloudcentral.org; http://usegalaxy.org
Projects' source code: https://github.com/chapmanb/cloudbiolinux; https://bitbucket.org/galaxy/cloudman; https://github.com/chapmanb/biocloudcentral; https://bitbucket.org/galaxy/galaxy-dist
Open Source License used: MIT (for Galaxy, see http://wiki.g2.bx.psu.edu/Admin/License)

Abstract

Many algorithms, toolkits and pipelines provide powerful approaches for manipulating biological data into meaningful information. However, it is still difficult to gain access to an up-to-date set of tools, data, and resources required to, for example, run a complete variant detection analysis pipeline. To do so, a researcher is required to have compute resources; configure those; find, install, and configure the currently recommended set of tools; upload reference data; upload research data and only then focus on using the analysis pipeline to answer biological questions.

To ease this process and greatly improve accessibility of biological tools, we have developed a set of services that automate many of the underlying steps and deliver complete analysis environments to the researcher. This talk will highlight the symbiosis between four coordinated projects that together provide accessibility to a variety of tools, improve ease of use, provide an opportunity for platform customization, and focus on building a sustainable community of users and developers:

1. **CloudBioLinux**, as a method for configuring bioinformatics analysis environments making it possible to prepare a machine image (in the cloud or otherwise) with over 100 bioinformatics tools and a large set of reference genome data.
2. **CloudMan**, as a platform for scalable tool execution, data integration, and sharing that works with multiple clouds, making it possible to scale the execution of jobs, customize individual environments, share those, and utilize commercial or research clouds (AWS, OpenNebula, OpenStack are operational while Eucalyptus support is forthcoming from JCVI).
3. **BioCloudCentral.org**, as a hub for easy access to the available cloud environments. It is a portal that allows a user to gain access to CloudBioLinux and CloudMan platform on any (compatible) cloud with a single mouse click.
4. **Galaxy**, as a graphical web-based tool execution framework capable of exploiting the software and infrastructure provided by CloudBioLinux and CloudMan. The available Galaxy is completely configured with all the underlying tools, reference data and infrastructure scalability needed for biologists to run computational pipelines.

PyPedia: A python crowdsourcing development environment for bioinformatics and computational biology

Alexandros Kanterakis[1], Morris Swertz[2]

[1,2]Genomics Coordination Center, University Medical Center Groningen, The Netherlands
[1]alexandros.kanterakis@gmail.com , *[2]m.a.swertz@gmail.com*

Availability: www.pypedia.com , www.molgenis.org
Source code: https://github.com/kantale/PyPedia_server
Software license: GPL v.3 , PyPedia contents license: Simplified BSD

The explosion of –omics data, the complexity of modern bioinformatics algorithms, and the necessity for validity, openness and reproducibility have raised the bar of expectations for software developers above the abilities of an average bioinformatics institute. To meet these expectations, the development should adhere to modern disciplines of professional application development that require excessive amount of skills and resources. Here we explore the use of collaborative and crowdsourced content management for the development and maintenance of bioinformatics algorithms that enables users to follow strict professional guidelines for qualitative and verifiable method implementation and documentation. Our showcase includes analysis of GWAS genotype data with methods indexed and delivered by PyPedia.

Crowdsourcing is the process where a loosely coupled community collaborates to perform related tasks under the coordination of a software application that imposes equal and simple rules for administration and quality control of all tasks. PyPedia facilitates this paradigm as a Wikipedia-like catalogue of bioinformatics algorithms where each article contains full source code of a function or class in the python programming language. Within the source code, any editor can call any function and instantiate any class that is defined in another article using the familiar wiki syntax, i.e. a call to foo() will reuse the method defined on wiki page foo. Furthermore each article, apart from the source code, contains the documentation, unit-tests and edit permissions. Every edit, triggers the execution of unit-tests that verify a certain behavior and quality for each article. The execution takes place in Google App Appspot that is a python sandbox suitable for testing of user provided and potentially unsafe code. Each page also provides an (editable) HTML form user interface where a user can execute the method online and have the results appear in browser. Furthermore a user can download a dependency-free, standalone version of the article (i.e. the Python code) that can be run or imported in any Python environment.

By applying this wiki based model for all the source code, PyPedia offers more than a commonly used version controlled code repository. It is a collaborative development environment that grows in a multidimensional way according to the most urgent needs of its user community. Furthermore, special attention was given to reproducibility. Through a simple and permanent URL a bioinformatician can share all the analysis software even if the articles where the analysis resides have been edited and altered. PyPedia is closely tightened with BioPython for basic biologic computation and can be used by systems like Galaxy for pipeline management. It also offers methods for statistical genomics and various format conversions. We also explore how data management tools such as MOLGENIS can benefit from PyPedia and demonstrate how the complete analysis can be reproduced in an external environment by sharing a simple URL. We conclude that PyPedia is an online, open source and collaborative IDE that focuses on verifiable and reproducible scientific development in the domains of bioinformatics and computational biology.

InterMine - Embeddable Data-Mining Components

Alexis Kalderimis (alex@intermine.org)[1], Richard N Smith (richard@flymine.org)[1], Daniela Butano (daniela@modencode.org)[1], Adrian Carr (adrian@flymine.org)[1], Sergio Contrino (sergio@modencode.org)[1], Fengyuan Hu (fengyuan@modencode.org)[1], Michael H Lyne (mike@intermine.org)[1], Rachel Lyne (rachel@flymine.org)[1], Radek Štépan (radek@intermine.org)[1], Julie Sullivan (julie@flymine.org)[1], Gos Micklem (g.micklem@gen.cam.ac.uk)[1].

[1] Department of Genetics, University of Cambridge, Downing Street, Cambridge, CB2 3EH, UK

Project Home	`http://www.intermine.org`
Code Repository	`svn://subversion.flymine.org/flymine/trunk`
API specification	`http://intermine.org/wiki/WebService`

A free and open-source software project, supported by the NIH and Wellcome Trust, licensed under the LGPL.

ABSTRACT:

InterMine (http://www.intermine.org) is a free and open-source biological data warehousing system used in projects worldwide. InterMine can integrate data from common biological formats and provides powerful query features through a web interface and RESTful web services, and now embeddable JavaScript components.

Having been adopted in a growing number of major Model Organism Databases (MODs), InterMine is coming to define a common query and web-services platform for integrated biological data (such as genome annotation, expression studies, ontologies, interactions and literature). Using InterMine, researchers can build custom *queries*, use pre-defined *template* searches, export results, and upload and analyse *lists* of data.

While many researchers continue to use the existing InterMine web application, a new embeddable UI framework means that the same functionality can increasingly be embedded elsewhere. The primary application of this is embedding InterMine functionality in MOD web-sites, which is one of the goals of the NIH InterMOD project. The framework is also designed to work from any web site and in a wide range of browsers, so that as many groups as possible can benefit from the MODs data integration and curation efforts. This talk looks at some ways that InterMine components can be used to enhance MOD and 3rd party pages.

These components are part of an actively developed set of client libraries, existing in Python, Perl, Ruby and Java to facilitate access to the web service API, supported by automatic code generation for any query. The JavaScript client libraries include both data presentation elements, and a more general set of modules for interacting with the web-services directly.

Creating biology pipelines with BioUno

Bruno P. Kinoshita, TupiLabs <bruno@tupilabs.com>

BioUno project consists of several plug-ins for Bioinformatics tools created for Jenkins, the leader Open Source continuous integration server. Each plug-in communicates with one or more tools using Jenkins features like job scheduling, distributed execution, tools interoperability, graphs and easy to use web interface. Using pipelines we can create workflows that can be saved in XML and later loaded by other users. Jenkins plug-in API provides access to operational system programs and executables, and an easy way to create configuration screens. This way, one can set up a simple workflow to run in a remote cluster very easily, with a standard UI. In this presentation we will see how to create a simple phylogenetic pipeline using MrBayes and FigTree, that loads the input files from a remote repository, store graphs for the job and e-mails the user. All this in less then 20 minutes.

Project URL: http://www.biouno.org
Source code: The project is composed of several plug-ins for Jenkins. At moment, the following plug-ins have been created:

- FigTree: https://github.com/tupilabs/figtree-plugin
- MrBayes: https://github.com/tupilabs/mrbayes-plugin
- Structure Plug-in: https://github.com/tupilabs/structure-plugin

License: MIT License

Mobyle Web Framework: New Features

Hervé Ménager[1], Bertrand Néron[1], Vivek Gopalan[2], Jie Li[2] and Yentram Huyen[2]
[1] Centre d'Informatique en Biologie, Institut Pasteur, Paris, France,
[2] Bioinformatics and Computational Biosciences Branch, Office of Cyber Infrastructure and Computational Biology,
National Institute of Allergy and Infectious Diseases, National Institutes of Health, Bethesda, MD 20892, USA
Presenting author email : hmenager@pasteur.fr

Mobyle Website	:	`https://projets.pasteur.fr/wiki/mobyle`
Mobyle Release Downloads	:	`ftp://ftp.pasteur.fr/pub/gensoft/projects/mobyle/`
Mobyle SVN Repository	:	`https://projets.pasteur.fr/svn/mobyle`
BMID/BMPS SVN Repository	:	`https://projets.pasteur.fr/svn/mobyle-niaid`
Open Source License	:	GNU GPLv2

Mobyle is an open-source framework and web portal specifically designed to facilitate the integration of bioinformatics software and databanks. Using Mobyle, researchers and bioinformaticians can leverage command-line applications from a web browser to seamlessly perform bioinformatics analyses on remote computing resources, such as high performance computing clusters. Here we present new modules that enhance the functionality of Mobyle, including newly-integrated tools and interfaces, web-based navigation of data, and user-defined analysis pipelines.

The **BCBB Mobyle Interface Designer (BMID)** application converts command-line driven applications to easy-to-use web interfaces. In Mobyle framework, single or groups of command-line arguments of an application are mapped to HTML form components through XML-based data model. In BMID, users create web form interface for an application by dragging and dropping new HTML form components, bound to XML data element, from existing templates and by creating new controls and validation rules using intuitive form wizard. BMID automatically generates valid Moble XML data model from the HTML form components that is used to deploy and to run the application in the Mobyle framework.

The **BCBB Mobyle Pipeline System (BMPS)** creates automated analysis pipelines by linking multiple applications within the Mobyle framework such that the output of one application becomes the input of the next. BMPS provides a graphical interface for creating user-defined scientific pipelines by dragging-and-dropping multiple applications onto a drawing canvas and linking file inputs and outputs using data type matching rules provided by the Mobyle framework. BMPS also provides an interface to manage and run pipeline jobs in remote compute environments.

Mobyle feature enhancements also include the integration of additional **visualization components** that enable users to embed HTML utilities within the Mobyle framework, such as Java applets and data-type-dependent file viewers, to provide a richer user experience, as well as the ability to edit and save files and associated metadata for future use in other Mobyle-hosted services.

By coupling enhanced functionality with simplified usage, Mobyle's BMID, BMPS, and the new visualization components are expected to promote broader usage of the Mobyle framework by the scientific research community.

Biopython Project Update

Eric Talevich[*], Peter Cock[†], Brad Chapman[‡], João Rodrigues[§], *et al.*

Bioinformatics Open Source Conference (BOSC) 2012, Long Beach, California, USA

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

In this talk we present the current status of 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 2011, we have made two releases. With Biopython 1.58 (August 2011), we included support for ABI chromatogram files, and also gained interoperability with PAML by merging an independent project, Brandon Invergo's pypaml. Biopython 1.59 (February 2012) added support for TogoWS, an integrated web resource for bioinformatics databases and services, and new features in GenomeDiagram [2]. Biopython 1.60 is expected to have been released by BOSC 2012. All releases have seen more unit tests, more documentation, and more new contributors. Additionally, we have submitted a paper describing the recent `Bio.Phylo` module for phylogenetics.

In Summer 2011 we had three Google Summer of Code (GSoC) students, all building features for protein structure analysis: Mikael Trellet (biomolecular interface analysis for `Bio.PDB`), Michele Silva (Python bridge for Mocapy++ [3] and linking it to `Bio.PDB` to enable statistical analysis of protein structures), and Justinas Daugmaudis (Python-based extension system for Mocapy++). Previous GSoC students João Rodrigues and Eric Talevich are working to refactor and merge the results of these projects under a unified API covering the PDB, mmCIF and PBDML formats for 3D molecular structures.

Two additional students are expected to work with us for GSoC 2012: Wibowo Arindrarto is implementing a Biopython equivalent to BioPerl's SearchIO (covering BLAST, HMMER, FASTA etc. search results), and Lenna Peterson is adding support for genomic variants (HGVS, GFF, VCF files).

We are now encouraging early adopters to help beta test Biopython under Python 3 and PyPy. The use of nightly unit tests via the OBF BuildBot server (http://http://testing.open-bio.org/) continue to be very helpful for cross-platform validation (covering Windows, Linux and Mac OS X for Python 2.5–2.7, Jython 2.5, plus Python 3.1–3.2 and soon PyPy) as well as catching general regressions.

Mailing list discussions continue to be active, with lots of new work and new contributors coming forward.

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] Pritchard, L., White, J.A., Birch, P.R., Toth, I. (2010) GenomeDiagram: a python package for the visualization of large-scale genomic data. *Bioinformatics* **2**(5) 616-7. doi:10.1093/bioinformatics/btk021

[3] Paluszewski, M., and Hamelryck, T. (2010) Mocapy++–a toolkit for inference and learning in dynamic Bayesian networks. *BMC Bioinformatics* **11** 126. doi:10.1186/1471-2105-11-126

[*] Institute of Bioinformatics, University of Georgia, Athens, GA, USA
[†] Information and Computational Sciences, James Hutton Institute (formerly SCRI), Invergowrie, Dundee DD2 5DA, UK
[‡] Bioinformatics Core Facility, Harvard School of Public Health, Harvard University, Boston, MA, USA
[§] Bijvoet Center for Biomolecular Research, Faculty of Science, Utrecht University, Utrecht, Netherlands

Biogem,Ruby UCSC API, and BioRuby

Hiroyuki Mishima[1], Raoul J.P. Bonnal[2], Naohisa Goto[3], Francesco Strozzi[4], Toshiaki Katayama[5], Pjotr Prins[6

1) Department of Human Genetics, Nagasaki University Graduate School of Biomedical Sciences, Japan. [hmishima@nagasaki-u.ac.jp]
2) Integrative Biology Program, Istituto Nazionale Genetica Molecolare, Milan 20122, Italy. [bonnal@ingm.org]
3) Department of Genome Informatics, Genome Information Research Center, Research Institute for Microbial Diseases, Osaka University, Japan, [ngoto@gen-info.osaka-u.ac.jp]
4) CeRSA, Parco Tecnologico Padano, Italy. [francesco.strozzi@tecnoparco.org]
5) Database Center for Life Science, Research Organization of Information and Systems, Japan. [ktym@dbcls.jp]
6) Laboratory of Nematology, Wageningen University, The Netherlands. [pjotr2012@thebird.nl]

BioRuby - http://bioruby.org/
Biogems.info - http://www.biogems.info/
Biogems - https://github.com/helios/bioruby-gem
Ruby UCSC API - https://github.com/misshie/bioruby-ucsc-api
Licence: The Ruby license

To scale up collaborative software development in BioRuby, we recognized that existing and new developers need to be encouraged to contribute more code. To achieve this, we created Biogem, a Ruby application framework for rapid creation of decentralized, internet published software modules written to lower the barrier to entry. Biogem, with its targeted modular and decentralized approach, software generator, tools and tight web integration, is an improved general model for scaling up collaborative open source software development in bioinformatics. The biogems.info website tracks published BioRuby modules. Popularity of each published module is tracked, as well as source code changes, updates, bugs and issues. This system encourages and motivates developers to improve modules.

To demonstrate the Biogem environment, we present the Ruby UCSC API, a library to access the UCSC genome database using Ruby. The API is designed as a Biogem and built on the ActiveRecord 3 framework for the object-relational mapping, making writing SQL statements unnecessary. The current version of the API supports databases of all organisms in the UCSC genome database including human, mammals, vertebrates, deuterostomes, insects, nematodes, and yeast. The API uses the bin index—if available—when querying for genomic intervals. The API also supports genomic sequence queries using locally downloaded *.2bit files that are not stored in the official MySQL database. The API is implemented in pure Ruby and is therefore available in different environments and with different Ruby interpreters (including JRuby). Assisted by the straightforward object-oriented design of Ruby and ActiveRecord, the Ruby UCSC API will enable biologists to query the UCSC genome database programmatically.

A Framework for Interactive Visual Analysis of NGS Data using Galaxy

Jeremy Goecks (jeremy.goecks@emory.edu)[1], The Galaxy Team[2], Anton Nekrutenko[3], and James Taylor[1]
[1]Departments of Biology and Math & Computer Sciences, Emory University
[2]http://galaxyproject.org
[3]Center for Comparative Genomics and Bioinformatics, Penn State University
Website: http://galaxyproject.org, Code: http://bitbucket.org/galaxy/galaxy-central/

Very large datasets are now the rule rather than the exception in next-generation genomic sequencing (NGS) experiments. Even simple NGS experiments routinely produce datasets that are many gigabytes in size. The size of NGS datasets limits the types and number of analyses that can be performed because each analysis take a significant amount of both time and computing resources. One approach for working more effectively with large datasets is to first identify a subset of meaningful data and perform experimental analyses on the data subset. Then, the lessons learned from the experimental analyses can be applied when analyzing the complete dataset. For example, it is often useful to try out different parameter settings for a tool or pipeline and then choose the settings that provide the best results.

We have implemented a framework that enables user-friendly experimentation on NGS data subsets using Galaxy [1,2] and its visual analysis environment, Trackster [3]. Using Trackster, genomic data can be visualized and analysis tools can be applied to generate and visualize new data *in real time*. For instance, transcript assembly from RNA-seq data and SNP calling can be done interactively with Trackster. Trackster enables interactive visual analysis of large NGS datasets by running analysis tools only on the subset of data that is visible to the user; running the tool on only visible data ensures that the tool runs quickly.

Trackster provides a generic framework for subsetting large datasets quickly and for reusing data subsets. Trackster is able to visualize and subset most major genomic dataset formats, including BED, GFF/GTF, SAM/BAM, and VCF. Any Galaxy tool that can run on subsets of data can be used in Trackster, and Galaxy's tool integration framework makes it possible to add nearly any tool into Galaxy without modifying its code. By providing fast data subsetting and access to Galaxy tools, Trackster provides a general platform for interactively analyzing NGS datasets visually. Trackster-Galaxy integration makes it easy to go from experimentation to dataset-wide analyses. Tools can be run quickly and repeatedly on visible data in Trackster; once suitable parameter values have been found, a tool can be run on a complete dataset and the tool's output is put in a Galaxy history for later use. The framework also enables collaborative visual analysis. Trackster visualizations can be shared, and collaborators can run tools in Trackster as well. Trackster and Galaxy are completely Web-based and require only a modern Web browser to use.

[1] Goecks, J., Nekrutenko, A., Taylor, J.Galaxy Team Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol* 11, R86 (2010).
[2] Blankenberg, D. *et al.* Galaxy: a web-based genome analysis tool for experimentalists. *Curr Protoc Mol Biol* Chapter 19, Unit 19.10.1–21 (2010).
[3] Goecks, J., Li, K., Clements, D., Team, T. G. & Taylor, J. The Galaxy Track Browser: Transforming the genome browser from visualization tool to analysis tool. 39–46 (2011).doi: 10.1109/BioVis.2011.6094046

Why Scientists Should Contribute to Wikipedia

Spencer Bliven,[1] Andreas Prlić,[2] and Philip E. Bourne[3]

[1] Bioinformatics Program, University of California, San Diego, La Jolla, California, USA
[2] San Diego Supercomputer Center, University of California San Diego, La Jolla, California, USA
[3] Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, La Jolla, California, USA

Email: sbliven@ucsd.edu
Website: http://topicpages.ploscompbiol.org

Wikipedia currently hosts over 21.5 million articles in 284 languages, and is the sixth most visited website on the internet.[1] Despite this popularity, the free encyclopedia is distrusted by schools and libraries, in part due to the lack of academic credentials of editors.[23] The journal *PLoS Computational Biology* recently introduced a new type of article called **Topic Pages** with the express goal of motivating academic experts to contribute to Wikipedia in the field of computational biology.[4] In this talk, we share our experience creating the first Topic Page, discuss why scientists should care about improving Wikipedia, and provide some tips for scientists who wish to contribute to Wikipedia articles.

A Topic Page is a detailed encyclopedic article which is published simultaneously in *PLoS Computational Biology* and on Wikipedia. This is possible due to the open creative commons license of both publications. Only articles not already present on Wikipedia are considered, and articles are written to conform both to Wikipedia's article guidelines and to PLoS's standards for scientific rigor. Only articles from computational biology are being considered by *PLoS Comp Biol*, but other journals have expressed interest in similar partnerships to improve the quality of Wikipedia articles in other fields.

This collaboration benefits all parties involved. The authors receive a *PLoS* publication and a wide audience for their article. Wikipedia gains a high-quality article that has passed peer-review for scientific accuracy. Finally, the computational biology community and the general public benefit by improving the coverage of computational biology articles in Wikipedia. Partnerships like Topic Pages are only feasible when journals publish content under permissive licenses like Creative Commons.

Despite the similarity in goals and licenses between Wikipedia and open-access journals, differences do exist. The Wikipedia community is active and passionate, and has developed specific guidelines for authors. Writing a Wikipedia article has many similarities to writing an academic review article, but there are subtle differences that authors should be aware of. We discuss cultural and technological differences between Wikipedia and *PLoS Comp Biol* which may surprise new authors of Topic Pages.

A 2005 study of *Nature* authors found that 17% consulted Wikipedia on a weekly basis, but only 10% had ever contributed.[5] Collaborations between open-access journals and Wikipedia have the potential to change that. We found publishing our own Topic Page[6] to be a very rewarding experience, and we encourage other scientists to seek out poorly represented scientific topics in Wikipedia and start editing!

[1] Alexa Internet rankings. http://www.alexa.com/siteinfo/en.wikipedia.org/wiki/Main_Page. Accessed 2012-04-12

[2] Lysa Chen (2007-03-28). "Several colleges push to ban Wikipedia as resource". *Duke Chronicle*.

[3] McHenry, Robert (2004-11-15). "The Faith-Based Encyclopedia". Tech Central Station.

[4] Wodak SJ, Mietchen D, Collings AM, Russell RB, Bourne PE (2012) Topic Pages: PLoS Computational Biology Meets Wikipedia. PLoS Comput Biol 8(3): e1002446. doi:10.1371/journal.pcbi.1002446

[5] Jim Giles. "Internet encyclopaedias go head to head." *Nature* **438**, 900-901 (15 December 2005) doi:10.1038/438900a;

[6] Bliven S, Prlić A (2012) Circular Permutation in Proteins. PLoS Comput Biol 8(3): e1002445. doi:10.1371/journal.pcbi.1002445
http://en.wikipedia.org/wiki/Circular_permutation_in_proteins

SCABIO – a framework for bioinformatics algorithms in Scala

Markus Gumbel[1]

[1]Mannheim University of Applied Sciences
Department of Computer Science, Institute for Medical Informatics,
Paul-Wittsack-Straße 10
D-68163 Mannheim, Germany
m.gumbel@hs-mannheim.de

Home:	http://www.mi.hs-mannheim.de/gumbel/en/forschung/scalabioalg
Source:	https://github.com/markusgumbel/scalabioalg
License:	Apache License Version 2.0

Overview/Motivation: SCABIO is a framework written in Scala ([7], [3]) that contains a collection of algorithms and methods for bioinformatics. It was originally developed to demonstrate algorithms for a lecture in bioinformatics but has grown in the meantime and become a comprehensive collection. Many of the algorithms are adapted from the text book "*Algorithmic Aspects of Bioinformatics*" by H. J. Böckenhauer and D. Bongartz [4]. The framework contains so far

- a generic (2D) extensible dynamic programing algorithm
- Global, semi-local and local pairwise alignment
- Multiple alignment
- Pattern recognition with the Viterbi-algorithm
- RNA 2D structure viewer based on Nussinov algorithm
- Greedy superstring algorithm for sequence assembly
- and much more...

Features: A key feature is a generic and extensible dynamic programming (DP) algorithm. Many bioinformatics algorithms rely on this method. However, often the DP approach within the algorithm is not explicitly apparent or even hidden. This is a pity as DP is a common algorithm paradigm which could and should have a clear programming interface. There are methods where the DP pattern (the recurrence equations) can be derived from an algebra (for instance, see [5]). Unfortunately, this requires an extra compiler and the integration into existing libraries appears to be laborious. SCABIO simply comes with a dynamic programming framework/interface which can be (re-)used or extended for many purposes, especially for all kind of bioinformatics algorithms. Within SCABIO, all pairwise alignment algorithms, the Viterbi-algorithm and the Nussinov-algorithm are based on these DP classes.

Requirements: SCABIO requires a Java virtual machine (Java 5 or above) and works with Scala 2.8 or above. It uses Maven 2 [1] and GIT [2] for development.

Discussion: Biojava [6] or bioscala [8] and many other mature open-source libraries for bioinformatics already exists. Why another framework and another language? Scala is a state of the art programming language for the Java virtual machine that combines object-oriented and functional programming paradigms and is 100% compatible with Java. SCABIO has a very compact code base as programming concepts like functional objects can improve the reusability remarkably. Also, Scala's `implicit def` feature can be used to wrap and seamless integrate other classes, e. g. those of the Biojava packages. In our opinion, these methods will simplify the development of scripting-like domain specific languages for the bioinformatics domain which are 1) very efficient and 2) type safe. As Scala also comes with an interpreter, interactive scripting is also possible.

The functional programming concepts are very interesting for concurrent algorithms and we will see how bioinformatics tools can benefit. Certainly, this question is already addressed withing the bioscala project [8].

Future work: SCABIO is still a very young project and especially more work on the documentation and unit tests needs to be done.

References

[1] http://maven.apache.org/.
[2] http://git-scm.com/.
[3] The scala programming language. http://www.scala-lang.org/.
[4] Hans-Joachim Böckenhauer and Dirk Bongartz. *Algorithmic Aspects of Bioinformatics*. Springer, 2007.
[5] Robert Giegerich, Carsten Meyer, and Peter Steffen. A discipline of dynamic programming over sequence data. *Science of Computer Programming*, 51, 2004.
[6] R. C G Holland, T. A. Down, M. Pocock, A. Prlic, D. Huen, K. James, S. Foisy, A. Dräger, A. Yates, M. Heuer, and M. J. Schreiber. Biojava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097, Sep 2008.
[7] Martin Odersky, Lex Spoon, and Bill Venners. *Programming in Scala*. artima developer, 2 edition, 2010.
[8] Pjotr Prins. bioscala. https://github.com/bioscala/bioscala.

bioKepler: A Comprehensive Bioinformatics Scientific Workflow Module for Distributed Analysis of Large-Scale Biological Data

Ilkay Altintas[1], Daniel Crawl[1], Weizhong Li[2], Shulei Sun[2], Jianwu Wang[1], Sitao Wu[2]
[1]San Diego Supercomputer Center, University of California, San Diego
{altintas, crawl, jianwu}@sdsc.edu
[2]Center for Research in Biological Systems, University of California, San Diego
{weli, s2sun, siw006}@ucsd.edu

Project Website: http://biokepler.org/
Accessing the code: http://www.biokepler.org/releases
Open Source License: BSD

The increase in the amount of DNA sequence data from the 2nd and the emerging 3rd generation sequencing technologies overwhelms current computational tools and resources. As datasets get larger, it poses challenges for distributed high-scale data analysis and transfer. This enormous data growth places unprecedented demands on traditional single-processor bioinformatics algorithms. Efficient and comprehensive analysis of the generated data requires distributed and parallel processing capabilities.

To address these challenges, the bioinformatics community often generates techniques for ad-hoc parallel computation for existing tools, such as, dividing their queries or databases to turn them into small jobs to be submitted into distributed environments, and then merge the output. However, most of these solutions are one-off developments that focus on a single bioinformatics tool. For widespread usage of distributed computing techniques in bioinformatics data analysis, new computational techniques and efficient execution mechanisms for the new data-intensive workload are needed. Data-intensive computing techniques and technologies that promote reuse and sharing like scientific workflows promise new capabilities to enable rapid and reproducible analysis of these next-generation sequence data. These technologies, when used together in an integrative architecture, have great promise to serve many projects with similar needs on the emerging distributed data-intensive computing resources.

For enabling bioinformaticians and computational biologists to conduct efficient analysis, there still remains a need for higher-level abstractions on top of scientific workflow systems and distributed computing methods. The **bioKepler project** (http://biokepler.org/) is motivated by the following three challenges that remain unsolved:

1. How can large-scale sequencing data be analyzed systematically in a way that incorporates and enables reuse of best practices by the scientific community?
2. How can such analysis be easily configured or programmed by end users with various skill levels to formulate actual bioinformatics workflows?
3. How can such workflows be executed in computing resources available to scientists in an efficient and intuitive manner?

bioKepler is a three year long project that builds scientific workflow components to execute a set of bioinformatics tools using distributed execution patterns, e.g., MapReduce and All-Pairs. Once customized, these components are executed on multiple distributed platforms including various Cloud and Grid computing platforms. In addition, we deliver virtual machines including a Kepler engine and all bioinformatics tools and applications we are building components for in bioKepler.

bioKepler is a module that will be distributed on top of the core Kepler scientific workflow system. Please refer to the Kepler collaboration website (https://kepler-project.org) for more information on Kepler in general.

Bioinformatics Testing Consortium: Codebase peer-review to improve robustness of bioinformatic pipelines

Dr. Ben Temperton, Oregon State University
btemperton@gmail.com

Over the last decade, biology has become one of the most data-rich sciences. A concomitant rapid expansion of the tools required for computational analysis of this data has also emerged, typically written by individuals or small teams and propagated to the bioinformatics community through publications as open-source programs. Whilst this code-sharing enables pipelines to be written once and become industry standards, the current review process for the publication describing a tool seldom involves a quality check of the tool itself, particularly if the tool is published as part of a broader biological study. Documentation is often sparse and code is often only tested on a single runtime environment, increasing the activation energy for future adopters. In the worst case, these difficulties can deter use and prevent broader adoption.

In the early 1990s the use of open source software libraries in the IT industry typically suffered from the issues seen in today's bioinformatics pipelines. However, a paradigm shift in the late 90s/early 2000s recognized the critical weakness of allowing software testing to be performed by the software developers themselves. When the industry started hiring professional testers, whose role consisted entirely of systematically running software as a naïve user down each possible path, these problems soon came to light and were quickly corrected. It is now commonplace for developers to release their code to testers, who then raise bugs in function and documentation for the developers to fix prior to general release.

While the use of professional testing in bioinformatics is undoubtedly out of the budgetary constraints of most projects, there are significant parallels to be drawn with the review process of manuscripts. Therefore, I would like to propose the establishment of a 'Bioinformatics Testing Consortium', to which pipelines could be voluntarily released for testing, and then subsequently tested by volunteer bioinformaticians to improve code robustness in different environments and raise standards of documentation.

The benefits to the developer of such a system would include fewer setup query emails; improved confidence in deployment; suggestions on how to improve the efficiency of algorithms used and exposure of the software to a broader range of data, improving applicability and ultimately, a stamp of approval that their codebase has been rigorously tested. End-users would ultimately benefit from better quality software with lower activation energy, broadening bioinformatics accessibility to those who are confounded at the first compilation error. Journals would benefit as the codebase presented in the manuscript would be more stable and thus less subject to changes from the reported methods.