An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics

Author: Ronald Taylor, Pacific Northwest National Laboratory, Richland, WA (ronald.taylor@pnl.gov)

This talk will present an overview of Apache Hadoop (http://hadoop.apache.org/) and associated open source software projects. Current Hadoop usage within the bioinformatics community will be summarized, and the advantages of Hadoop will be discussed as they pertain to subareas of bioinformatics analysis.

Hadoop is a software framework that can be installed on a commodity Linux cluster to permit large scale distributed data analysis. Hadoop provides the robust Hadoop Distributed File System (HDFS) as well as a Java-based API that allows parallel processing across the nodes of the cluster. Programs employ a Map/Reduce execution engine which functions as a fault-tolerant distributed computing system over large data sets - a method popularized by use at Google. There are separate Map and Reduce steps, each step done in parallel, each operating on sets of key-value pairs. Processing can be parallelized over thousands of nodes working on terabyte or larger sized data sets. The Hadoop framework automatically schedules map tasks close to the data on which they will work, with "close" meaning the same node or, at least, the same rack. Node failures are also handled automatically.

In addition to Hadoop itself, which is a top-level Apache project, there are subprojects build on top of Hadoop, such as Hive (http://hadoop.apache.org/hive/), a data warehouse framework used for ad hoc querying (with an SQL type query language) and used for more complex analysis; and Pig (http://hadoop.apache.org/pig/), a high-level data-flow language and execution framework whose compiler produces sequences of Map/Reduce programs for execution within Hadoop. Also, I will discuss HBase (http://hadoop.apache.org/hbase/), another Hadoop subproject inspired by Google's BigTable. Each table is stored as a multidimensional sparse map, with rows and columns, each cell having a time stamp. HBase adds a distributed, fault-tolerant scalable database onto the Hadoop distributed file system, permitting random access to the stored data. Other "NoSQL" scalable databases (e.g., Hypertable, Cassandra) will be briefly introduced as HBase alternatives. Additional topics covered will include (1) the Apache Mahout project (http://lucene.apache.org/mahout), which is parallelizing many machine learning algorithms in Hadoop, (2) Cascading (http://www.cascading.org/), an API for defining and executing fault tolerant data processing workflows on a Hadoop cluster, and (3) use of the Amazon Elastic Compute Cloud to run Hadoop.

Recent applications of Hadoop in bioinformatics will also be summarized. Processing of high throughput sequencing data (for example, mapping extremely large numbers of short reads onto a reference genome) is an area where Hadoop-based software is making an impact. Thus, these examples will highlight, among others, the Cloudburst software (M. Schatz, 2009) and other Hadoop-based software from University of Maryland researchers for analysis of next-generation DNA sequencing data (http://www.cbcb.umd.edu/software/). Also, applications such as CloudBLAST (Matsunaga et al. 2008) will be presented. Dean and Ghemawat made the point in their recent article (Jan 2010, Comm. of the ACM) that Hadoop is well-suited to fill a need for analysis of high throughput data coming from heterogeneous systems. In conclusion, I will briefly describe a new project at Pacific Northwest National Laboratory wherein we are proposing Hadoop and HBase-based development of a scientific data management system that can scale into the petabyte range, that will accurately and reliably store data acquired from various instruments, and that will store the output of analysis software and relevant metadata, all in one central distributed file system. My concluding point, extracted from that project, follows Dean & Ghemawat. That is, for much bioinformatics work not only is the scalability permitted by Hadoop and HBase important, but also of consequence is the ease of integrating and analyzing various disparate data sources into one data warehouse under Hadoop, in relatively few HBase tables.

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data

McKenna, A.[1], Hanna, M.[1], Banks, E.[1], Sivachenko, A[1], Cibulskis, K.[1], Kernytsky, A.[1], Garimella, K.[1], Altshuler, D.[1,2], Gabriel, S. [1], Daly, M.[1,2], and DePristo, M.A.[1]
[1]Program in Medical and Population Genetics, The Broad Institute of Harvard and MIT, Five Cambridge Center, Cambridge, Massachusetts 02142.
[2]Center for Human Genetic Research, Massachusetts General Hospital, Richard B. Simches Research Center, Boston, Massachusetts 02114, USA

Contact: Matt Hanna hanna@broadinstitute.org
Overview: http://www.broadinstitute.org/gsa/wiki/index.php/Main_Page
Source: https://svn.broadinstitute.org/Sting
BSD licensed

Next-generation DNA sequencing (NGS) projects, such as the 1000 Genomes Project, are already revolutionizing our understanding of genetic variation among individuals. However, the massive data sets generated by NGS—the 1000 Genomes pilot alone includes nearly five terabases—make writing feature-rich, efficient and robust analysis tools difficult for even computationally sophisticated individuals. Indeed, many researchers are limited in the scope and the ease with which they can answer scientific questions by the complexity of accessing and manipulating the data produced by these machines.

Our solution to this issue is the Genome Analysis Toolkit (GATK), a structured programming framework designed to ease the development of analysis tools for next-generation DNA sequencers using the functional programming philosophy of MapReduce. The GATK provides a small but rich set of data access patterns that encompass the majority of analysis tool needs. Separating specific analysis calculations from common data management infrastructure enables us to optimize the GATK framework for correctness, stability, CPU and memory efficiency, and to enable distributed and shared memory parallelization. We highlight the capabilities of the GATK by describing the implementation and application of robust, scale-tolerant tools like coverage calculators and SNP calling. We present the techniques used to partition datasets, allowing the GATK to run effectively on both large-memory multiprocessor machines as well as the more restrictive but abundant nodes in a server farm. Finally, we outline the techniques we use to efficiently access and present genomic structure and variation data stored in a wide array of highly flexible file formats.

The GATK programming framework enables developers and analysts to quickly and easily write efficient, robust, and easy-to-use NGS tools. The GATK already underlies several critical tools in both the 1000 Genomes Project and The Cancer Genome Atlas, including: quality-score recalibration, multiple-sequence realignment, HLA typing, multiple sample SNP genotyping, and indel discovery. The GATK's robustness and efficiency has enabled these tools to be easily and rapidly deployed in recent projects to routinely process terabases of Solexa, SOLiD, and 454 sequencer data, as well as the hundreds of lanes processed each week in the production resequencing facilities at the Broad Institute.

SeqWare Query Engine: Storing and Searching Sequence Data in the Cloud

Brian D. O'Connor[1], Jordan Mendler[2], and Stanley F. Nelson[2]

1. Lineberger Comprehensive Cancer Center, UNC, Chapel Hill, NC, brianoc@email.unc.edu
2. Department of Human Genetics, UCLA, Los Angeles, CA

website: http://seqware.sourceforge.net
code: http://sourceforge.net/projects/seqware/develop
license: GPLv3

The SeqWare Query Engine project was started to provide a scalable means to store and query large amounts of data derived from massively parallel sequencing technologies. This project provides two components, first, a backend to store variant, coverage, and other key information in a scalable database cluster and, second, a web service to provide both a programmatic and web-based interface to query and filter this data. The SeqWare Query Engine backend is built using the open source HBase project which offers a truly distributed, NOSQL database that is capable of supporting thousands of genomes. HBase, as a Hadoop subproject, has tight integration with the Map/Reduce framework which provides a convenient and robust mechanism to traverse and process entries in the query engine database. The query engine webservice was written in Java using the RESTlet framework which provides both a user interface and a RESTful API for programmatic access. The SeqWare Query Engine has been successfully deployed on a small (4-8 node) HBase cluster within our group and is a key part of sequencing projects currently underway at UCLA as well as UNC. The SeqWare project (which includes SeqWare Query Engine) is freely available under the GNU General Public License (GPLv3) and has been written to allow for easy customization and community driven development through the project website (http://seqware.sourceforge.net).

Hybrid Cloud and Cluster Computing Paradigms for Life Science Applications

Judy Qiu[1,2], Thilina Gunarathne[1,2], Jaliya Ekanayake[1,2], Jong Youl Choi[1,2], Seung-Hee Bae[1,2],
Hui Li[1,2], Bingjing Zhang[1,2], Yang Ryan[1,2], Saliya Ekanayake[1,2], Tak-Lon Wu[1,2],
Scott Beason[2], Adam Hughes[2], Geoffrey Fox[1,2]

[1]School of Informatics and Computing, [2]Pervasive Technology Institute
Indiana University, Bloomington.
{xqiu, tgunarat, jekanaya, jychoi, sebae, lihui, zhangbj, yangruan, sekanaya, taklwu, smbeason, adalhugh, gcf}@indiana.edu

SALSA project http://salsahpc.indiana.edu/
Twister software and license http://www.iterativemapreduce.org/ and http://www.iterativemapreduce.org/license.html

Cloud computing [1] offers new approaches for scientific computing that leverage the major commercial hardware and software investment in this area. Closely coupled applications are still unclear in clouds as synchronization costs are still higher than on optimized MPI machines. However loosely coupled problems are very important in many fields and can achieve good cloud performance even when pleasingly parallel steps are followed by reduction operations as supported by MapReduce. However we can use clouds in several ways and we have compared five different approaches using two biomedical applications. We look at the cloud infrastructure service based virtual machine utility computing models of Amazon AWS and Microsoft Windows Azure; MapReduce based computing frameworks Apache Hadoop (deployed on raw hardware as well as on virtual machines) and Microsoft DryadLINQ. We compare performance showing strong variations in cost between different EC2 machine choices and comparable performance between the utility computing (spawn off a set of jobs) and managed parallelism (MapReduce). Our main emphasis is Cloud techniques as provider comparison is very subject to changes. The MapReduce approach offered the most user friendly approach. Typical results [2] are shown in Fig. 1.

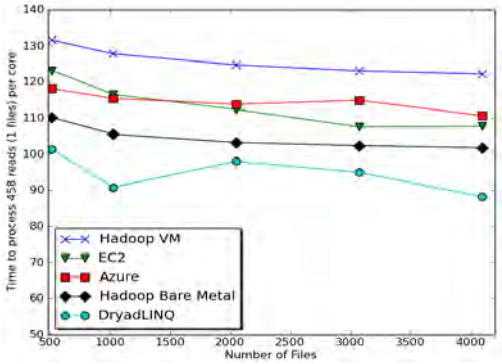

Figure 1 Time to process a single biology sequence file (458 reads) per core with different frameworks

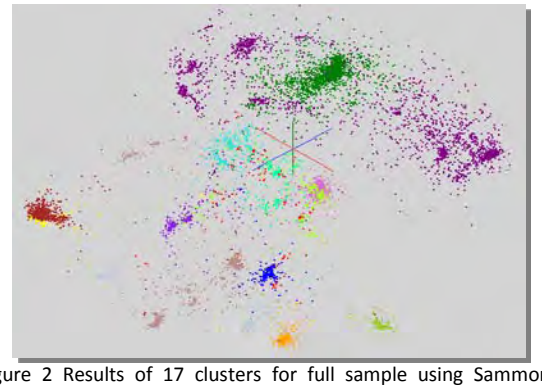

Figure 2 Results of 17 clusters for full sample using Sammon's version of MDS for visualization.

A typical bioinformatics pipeline of Smith-Waterman distance Computation, Deterministic Annealing Clustering and MDS visualization is shown below in Fig. 3, which can give results such as Fig. 2 where the results of 30,000 Metagenomics sequences in 3D are shown. The visualization uses dimension reduction where we have implemented two powerful methods GTM (Generative Topographic Mapping) and MDS (Multidimensional Scaling) [3] [4].

Only MDS can be used for DNA sequence visualization as GTM requires a vector representation of original high dimensional data whereas MDS only requires the N by N matrix of dissimilarity scores between sequences. Multiple Sequence Alignment needed to obtain a uniform vector representation of sequences is typically infeasible. The distance matrix calculation needed by MDS is very suitable for cloud implementation as the computations are independent. However both clustering and MDS require parallel implementation as they are expensive $O(N^2)$ computations; the run time of these on a 768 core cluster is about 3 hours for 30,000 sequences

with a speed up of 500. These parallel implementation run poorly on clouds or MapReduce as their iterative algorithms require the long running processes and low latency of MPI. Thus we see hybrid cluster-cloud architectures as needed for this class of problem where a complete workflow is gotten by linking separate services in clouds and closely coupled clusters.

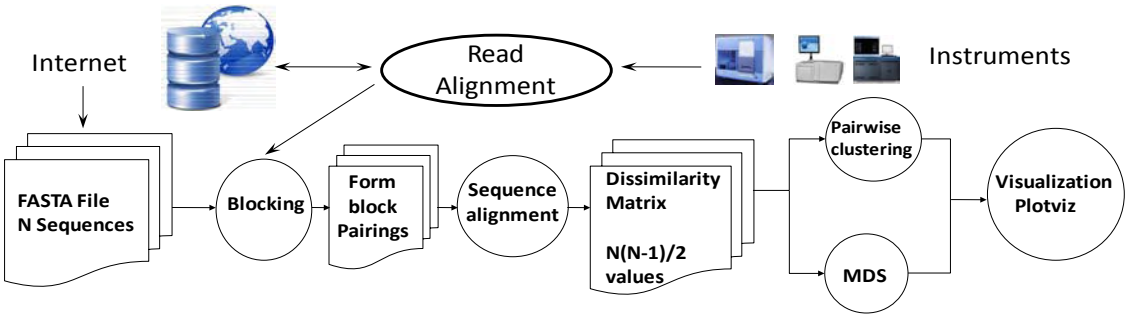

Figure 3 Pipeline for analysis of metagenomics Data

We have developed new interpolation algorithms for both MDS and GTM which can exploit clouds and MapReduce for the dominant part of the computation for large problems. These perform a basic dimension reduction for a sample of the data (20,000-100,000 points) which runs using MPI on a cluster; the remaining points are interpolated which is a pleasingly parallel cloud application. We will present performance results for run time and quality of dimension reduction.

Alternatively we have extended MapReduce in an open source system, Twister [5] [6], that supports iterative computations of the type needed in clustering, MDS and GTM. This programming paradigm is attractive as it supports all phases of the pipeline in Fig. 1. We present performance comparisons between MPI, MapReduce and Twister on kernel applications such as matrix multiplication as well as the core services of Fig. 1.

References

[1] Jaliya Ekanayake, Xiaohong Qiu, Thilina Gunarathne, Scott Beason, Geoffrey Fox High Performance Parallel Computing with Clouds and Cloud Technologies to appear as a book chapter to Cloud Computing and Software Services: Theory and Techniques, CRC Press (Taylor and Francis), ISBN-10: 1439803153.

[2] Thilina Gunarathne, Tak-Lon Wu, Judy Qiu, and Geoffrey Fox, Cloud Computing Paradigms for Pleasingly Parallel Biomedical Applications, Proceedings of Emerging Computational Methods for the Life Sciences Workshop of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[3] Seung-Hee Bae, Jong Youl Choi, Judy Qiu, Geoffrey Fox Dimension Reduction and Visualization of Large High-dimensional Data via Interpolation, Proceedings of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[4] Jong Youl Choi, Seung-Hee Bae, Judy Qiu, Geoffrey Fox, Bin Chen, and David Wild, Browsing Large Scale Cheminformatics Data with Dimension Reduction, Proceedings of Emerging Computational Methods for the Life Sciences Workshop of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[5] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, Geoffrey Fox Twister: A Runtime for Iterative MapReduce, Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[6] Twister software at http://www.iterativemapreduce.org/

Cloud-scale genomics: examples and lessons

Ben Langmead[1,3], Michael C. Schatz[1], Jimmy Lin[2], Mihai Pop[1], Steven L. Salzberg[1], Kasper Hansen[3], Jeff Leek[3]

1. Center for Bioinformatics and Computational Biology, University of Maryland, College Park, MD 20742, USA
2. The iSchool, College of Information Studies, University of Maryland, College Park, MD 20742, USA
3. Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, 615 North Wolfe Street, Baltimore, Maryland 21205, USA

Email: blangmea@jhsph.edu
Project URL: http://bowtie-bio.sf.net/myrna
Download URL: https://sourceforge.net/projects/bowtie-bio/files/myrna
License: Artistic License

In the race between DNA sequencing throughput and computer speed, sequencing is winning by a mile. While sequencing throughput has increased at a rate of about 5-fold per year, computer speed generally follows “Moore’s Law,” doubling every 18 or 24 months. As this gap widens, the question of how to design higher-throughput analysis pipelines becomes critical. Cloud computing, which allows any researcher equipped with a credit card to tap into the vast economies of scale afforded by some of the world's largest data centers, is increasingly seen as one way to combat this trend.

This talk examines and draws lessons from two tools that were designed from the ground up to run on a commercial cloud service. These tools are: Crossbow, a cloud-based tool for SNP genotyping from short reads, and Myrna, a cloud-based tool for calculating differential gene expression from large RNA-seq datasets. The design and performance characteristics of these tools provide some insight into how readily a range of comparative genomics applications can be adapted to the commercial cloud. The discussion will include assessments and advice regarding ease-of-use ease-of-development of cloud applications. The discussion will also highlight ways in which the development process for tools like Crossbow and Myrna could be improved in the future.

Title: Deploying Galaxy on the Cloud
Authors: Enis Afgan[1], Dannon Baker[1], Nate Coraor[3], The Galaxy Team[2], Anton Nekrutenko[3], James Taylor[1]
Author affiliations:
[1]Department of Biology and Department of Mathematics & Computer Science, Emory University {E.A. email: eafgan@emory.edu}
[2]http://galaxyproject.org
[3]Huck Institutes of the Life Sciences and Department of Biochemistry and Molecular Biology, The Pennsylvania State University
Project website: http://usegalaxy.org/cloud
Project source code: http://bitbucket.org/afgane/galaxy-central-gc2/
Open Source License used: Academic Free License

Abstract:
DNA sequencing has become one of the most transformative high-throughput techniques in life sciences. Novel sequence-based assays have made sequencing an indispensable tool for studying gene expression and regulation, chromatin structure, and sequence variation. What is most transformative however, is the wide availability of "next-generation" DNA sequencing (NGS) instruments, enabling any investigator, for a modest cost, to produce enormous amounts of DNA sequence data. However, working with raw data generated by next-generation sequencers and transforming it into biologically meaningful information requires significant computing infrastructure and informatics support. For the majority of experimentalists that lack needed computational support, just storing and managing the vast amount of data produced by these technologies presents a significant informatics burden As a result, for an experimental group with no computational expertise, simply running a data analysis program is a barrier, let alone building a compute and data storage infrastructure capable of dealing with volume and processing requirements of NGS data.

As a first step in combating the NGS data deluge, there exists an open-source system, Galaxy. Galaxy provides an integrated analysis environment where domain scientists can, without informatics expertise, interactively construct multi-step analyses, with outputs from one step feeding seamlessly to the next. In order to utilize Galaxy and ease NGS analyses, there is a need for availability of computational infrastructure. Fortunately, a computational model – cloud computing – has recently emerged and is well suited to the analysis of large-scale sequence data. However, cloud computing resources are not yet suitable for immediate "as is" use by experimental biologists. As a step in the direction of enabling seamless NGS analyses on the cloud, and thus removing limitations of local or publicly offered computational services and contentions that arise in such environments, we have developed Galaxy Cloud (GC). GC is a comprehensive manager for enabling, running, and scaling the Galaxy application on cloud computing infrastructures. It offers a simple web-based interface that allows anyone to acquire the desired computational and storage resources on a cloud infrastructure, and perform NGS analysis through the familiar Galaxy interface and corresponding tools (complete Galaxy functionality is supported). GC automatically handles all aspects of resource acquisition, configuration, and data persistence, thus entirely insulating a user from the low-level computational details. This talk will focus on the motivation, use cases, and available functionality within GC.

Community-driven computational biology with Debian and Taverna

S. Möller[1-3,*], H. Krabbenhöft[2], A. Tille[3], D. Paleino[3,4], A. Williams[5], K. Wolstencroft[5], C. Goble[5], C. Plessy[3]

[1]University Clinics of Schleswig-Holstein, Department of Dermatology, [2]University of Lübeck, Institute for Neuro- and Bioinformatics, Ratzeburger Allee 160, 23530 Lübeck. Germany; [3]Debian Linux Society, [4]Università degli Studi di Palermo, Dipartimento di Scienze Stomatologiche, Via del Vespro 129, 90127 Palermo, Italy, [5]University of Manchester, Oxford Road, Manchester, M13 9PL, UK

Availability: http://taverna.nordugrid.org and http://www.taverna.org.uk under the terms of the GPL, http://debian-med.alioth.debian.org

Computational biology manifests itself in many flavours. It comprises the data analysis and -management of sequences, structures, the observed and synthetical variants of the prior, static or dynamic interactions, and serves the modelling of biological processes in physiological and pathophysiological conditions. The field gained an enormous momentum over the past two decades. The information gathered today covers biological properties of many organisms and serves as a reference and general source for derived work also for neighbouring disciplines. Biologist, physicians and chemists all started using bioinformatics tools, data and models in their routine. The latest trend is to integrate the thinking of engineers and physicis, who construct compounds in silico to later prove the predicted function in the lab. The approach became known as synthetic biology and is perceived by many to allow a fluent transition towards nano-technologies. With research questions becoming increasingly complex, they demand the interaction of highly specialised disciplines. This leads to a steady increase in the number of non-redundant tools and databases that researchers need to interact with - both the computational developer and the biological users.

The dependency of the biological research community on such services will increase over the upcoming years. The strong computational demands of the services, and the sheer complexity of the research fosters the collaboration of individuals from many sites, computationally in form of grid and cloud computing, but also between computationally and biologically primed groups. To maintain the software installation consistently is barely achievable for dedicated individuals; the sharing of such across various platforms and institutional boundaries is the driving force behind the here presented work of the Debian Linux community.

Debian is an open society of enthusiasts around the globe who collaborate on packaging free software for the Linux and FreeBSD kernels. Packages are prepared by individuals and uploaded to the distribution's main servers for auto-building on today's most prominent platforms, thus rendering them available from mobiles to supercomputers and for all common processors. For complex suites or as a principle, packagers have an option to share their effort as part of a community. This process is aided by portalsauto-prepared by the infrastructure of the Debian blends. Packages invite feedback from users with the Bug Tracking System. Around 80000 users have allowed the counting of their applications via Debian's Popularity-Contest initiative. Separately counted are installations of packages that are forwarded to derived distributions. The most prominent of these is Ubuntu, for which more than 1.3 million users are reporting. Packages are described verbosely and are translated to many languages. More formally they may be selected by manual assignment of terms from a controlled vocabulary.

Technical constraints for the packaging are laid out in the Debian Policy document. Changes to it are discussed on the project's mailing lists and may be subject to voting by contributors to the distribution. The Ubuntu Linux distribution adopts the Debian packages for their own software "universe" and as such considerably contributes to the dissemination of the efforts. The computing world experiences continuous transitions, e.g. these days from 32 to 64 bit. Upcoming is an increased acceptance for energy-saving ARM- and MIPS-based operating systems of the mobile world and some special highly parallel systems. With Debian's packages being auto-built on all these different hardware platforms, one can expect continuity during such transitions, and similarly find consistent setups in the typical heterogeneous research infrastructures. This is of particu- lar benefit for distributed computations and contributes to the strong adoption of Debian and Ubuntu for cloud computing.

Packaging is most successful, i.e. up-to-date and tested, when it is derived from the packager's daily routine. For computational biology, the community now faces the challenge to scale with the steady increase in complexity: the number of contributors to the packaging needs to match the number of programs that users expect to be available. The group maintenance of applications is one such approach that seeks to lower the entry hurdle for the packaging by mutual training and the distribution of work according to expertise and interests. It also helps the integration of the software developers themselves with the community, e.g. for AutoDock and BALLView: the software developers follow the distribution's bug reports directly, and may contribute a description of their package or were invited to upload their own packages directly to the distribution's servers rather than offering them on their respective home page.

With an increasing number of packages available, the interaction between those tools becomes more and more of concern. This addresses the establishment of workflows comprising tools from many packages, but from the distribution's perspective it is also the challenge to work on the exact same version of public databases. The sharing of input between multiple applications is an ongoing work, for which many bioinformatics groups around the globe have provided solutions independently. To tap into that wealth of experiences and use it to share the effort to maintain the infrastructure is our impetus.

The distribution's software packages allows the tools included in those packages to be referenced and shared. The UseCase plugin developed as part of the EU KnowARC project extends the Taverna Workflow Workbench to take the description of such tools and include invocations of them within a Taverna workflow. The tools can be configured to run locally, or on a remote machine accessed via secure credentials such as ssh or grid certificates. Multiple invocations of a service can be achieved by the calling of the corresponding tool on a number of nodes at the same time, thus allowing faster running of the workflow over a distributed network of machines. So a workflow developer can write and test a workflow on small amounts of data locally and then by a simple change of configuration, run the workflow on a grid or cloud on much larger data sets. Workflows can include, not only tools within a packaged distribution, but also calls to other services such as WSDL operations, queries of a BioMart database or invocations of R scripts. The workflows can be uploaded to the myExperiment website and shared either publicly or with specific groups of people. The workflows can be downloaded and run, edited or included as part of a wider overall workflow. The development of workflows and the sharing of expertise via the myExperiment website based upon the creation of packaged distributions of tools, allows the collaboration of the Linux and Bioinformatics communities with great future potential.

With Taverna as a workflow engine and as a data transporter, to work locally in a most efficient manner, one also needs to have the data locally accessible - with the right indices and APIs and (especially) in the right version. For clouds, locally may now mean remote to the user's location, and it allows for the sharing of the data. The Debian community has prepared a small utility, getData, that knows how to download the most recent versions of a series of common databases, checks for the availability of a series of bioinformatics tools, and performs the respective indexing. When collaborating in clouds, the users can also ensure that any manual updates of databases are performed only once for the instant direct benefit of all other users.

To conclude, the dynamics of all three contributors, i.e. Linux distribution, Cloud infrastructure and workflow suite, are forming a symbiosis towards a readily usable infrastructure for performing and sharing biologically inspired research. The clouds bring considerable relief to smaller research groups, allowing them to think large, with the (optional) gained confidence through immediately available expert collaborators.

*Corresponding author: moeller@debian.org

Title: Dealing with the Data Deluge: What can the Robotics Community Teach us?
Authors: London, Darin - Institute for Genome Science and Policy, Duke University Medical Center (darin.london@duke.edu); Furey, Terry - Institute for Genome Science and Policy, Duke University Medical Center; Boyle, Alan - Institute for Genome Science and Policy, Duke University Medical Center
URL for Overall Project Website and Access to the Code: http://search.cpan.org/~dmlond/Google-Spreadsheet-Agent-0.01/lib/Google/Spreadsheet/Agent.pm

In 1986, Rodney Brooks published a paper which revolutionized the robotics industry (Brooks, 1986). In it, he presented work on a robot based on a new architecture, called the Subsumption architecture, which is composed of layers of very simple autonomous modules each designed with a very tight sense-response circuit, with modules in one layer able to mask/amplify (subsume) the inputs and outputs of modules in other layers. The critical advantage of this architecture is that it places many layers of intelligence between the 'data' of the world, and the overall action of the robot, without the need for any single internal representation of the world. This architecture was radically different from previous approaches which attempted to use a single, central processing system to 1. store a map or representation of the outside world based on the combined inputs from all sensors, 2. evaluate the state of the world, and decide which steps to take towards reaching the overall goals, 3. instruct all effectors how to act to achieve the current goal, 4. repeat. Though controversial, and highly debated within the robotics community at the time, no one now disputes that this architecture has single-handedly moved robotics from a theoretical academic adventure to a multi-billion dollar industry.

In many ways, we as bioinformaticians are now reaching a similar point. Most of us older folk grew up in a world of 1-server:1-dataset, whereby we wrote our scripts to store all of the tasks required to conduct our analyses along with all of the logic for performing those tasks, and very little intelligence for when (best) to conduct these tasks, or how to (re)prioritize tasks in different ways. All of us are now entering into a world of potentially infinite servers, and potentially infinite amounts of data to analyze. In order to make full use of the servers that will be available to us to analyze this data, our simple, dumb scripts are going to need to become more organic, responsive, and adaptive. We must also begin to move past the view that NextGen sequencing presents a problem of too much data, and begin to view it as a problem of not having enough layers of intelligence between the data and us to filter and amplify the signals that we are interested in from the noise that we wish to ignore.

This talk presents a module facilitating the creation of a subsumption architecture, called Google::Spreadsheet::Agent, available from the Comprehensive Perl Archive Network under the same terms as Perl itself. This module allows many highly focused intelligent agents, running on many different servers, to act autonomously to perform tasks on many datasets stored in the same, central Google Spreadsheet. Furthermore, by allowing concurrent access and editing to both humans and computer agents, the system presents a natural way of integrating the activities of human agents into the overall function of the system. This architecture has been implemented successfully into a pipeline to analyze Next Generation (Solexa and Illumina) Sequence Data associated with the Encyclopedia of DNA Elements (EncODE, http://www.genome.gov/10005107), and is currently being applied to a similar pipeline designed to research cross-species comparisons of Chimp, Rhesus, Mouse, and Human cell lines data.

References:
Brooks, Rodney, A., 1986. 'A Robust Layered Control System for a Mobile Robot'. IEEE Journal of Robotics and Automation, VOL RA-2, No. 1

The Goby framework: towards efficient next-generation sequencing data analysis

Nyasha Chambwe [1,2](nyc2003@med.cornell.edu), Kevin C. Dorff[1], Marko Srdanovic[1], Xutao Deng[1], Stuart J.D. Andrews[1,2], Fabien Campagne[1,2*] (fac2003@med.cornell.edu)
[1]The HRH Prince Alwaleed Bin Talal Bin Abdulaziz Alsaud Institute for Computational Biomedicine; Weill Medical College of Cornell University, New York, NY 10021; [2]Department of Physiology and Biophysics, Weill Medical College of Cornell University, New York, NY 10021

Project URL: http://goby.campagnelab.org/
Source Code URL: http://campagnelab.org/software/goby/download-goby/
Open Source License: GNU General Public License

Next-Generation Sequencing (NGS) technologies continue to evolve rapidly, generating massive amounts of short-read sequence data. The sheer volume of NGS data introduces computational challenges in its processing and storage.

NGS projects often require the ability to store sequence reads, alignments, base-resolution histograms, and to record specific subsets of reads. Current file formats support each of these types of information. NGS reads are often stored as Fasta or Fastq format. Alignment formats encode information about where reads map to a reference sequence as well as sequence variations (e.g., SAM/BAM format[1]). Base resolution histograms can be used to represent read abundance at particular positions of the reference sequence (e.g., Wig/Bed format are often shown as tracks in genome viewers). Files that represent a subset of reads help mark reads with specific properties, such as reads that do not match a reference and are often stored as text files. Files encoded in these formats can become very large when analyzing datasets that contain tens of millions to billions of reads. This creates scalability issues in most NGS analysis pipelines.

We have developed the Goby software framework to address these problems and provide a test-bed for alternative file formats and algorithms. Goby combines Gzip compression with the open-source Google Protocol Buffers library (http://code.google.com/p/protobuf/). Protocol buffers are advantageous because they support multiple languages (i.e., Python, Java, C++ and others), are cross-platform, flexible, and extensible. For instance, they allow older versions of software to read newer versions of a file format (forward compatibility) or new versions of the software to read older versions of a file format (backward compatibility). Goby files are organized as chunks and support semi-random access in a very large file, which is useful to retrieve only slices of a read or alignment file for processing on a cluster. Goby file formats are precisely specified and are often significantly smaller than current standards. For instance, we find that Goby alignment files can be 2 to 5 times smaller than an equivalent alignment encoded in BAM format (compressed binary version of a SAM file). The size ratio depends on the lengths of the reads, the amount of sequence variation and the platform and can only be estimated empirically. We have tested the Goby file formats by constructing an RNA-Seq analysis pipeline. This pipeline supports multiple aligners, including the Burrows Wheeler Aligner[2] (BWA) and the Last[3] aligner and has been tested using data from the Sequencing Quality Control (SEQC) (http://www.fda.gov/ScienceResearch/BioinformaticsTools/MicroarrayQualityControlProject/default.htm), from four major sequencing platforms. The talk will present compression ratios obtained with the Goby file formats and provide an overview of this Java software framework.

References:

1. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R; 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009 Aug 15;25(16):2078-9. Epub 2009 Jun 8
2. Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics, 25:1754-60.
3. Incorporating sequence quality data into alignment improves DNA read mapping Martin C. Frith, Raymond Wan, Paul Horton Nucleic Acids Research, 2010 (in press)

Title: BioHDF: Open binary file formats for NGS data – current status and future directions

Authors: Dana Robinson(derobins@hdfgroup.org) [1], Mark Welsh [2], Todd M. Smith [2], Mike Folk [1]

Affiliations:
[1] The HDF Group, 1901 S. First St., Suite C-2 Champaign IL 61820
[2] Geospiza, Inc. 100 West Harrison St. North Tower #330, Seattle WA 98119

Project URL: http://www.biohdf.org/

Download URL: http://www.biohdf.org/biohdf_downloads.html

Licensing: BSD-style licensing

Abstract:
The huge volume of data produced by the latest generation of sequencing technologies presents significant challenges in data transmission, storage, bioinformatics analysis, visualization, and archiving. Widespread adoption of next-generation DNA sequencing (NGS) will be hindered if bioinformatics software cannot scale to meet these challenges. The BioHDF project (http://www.biohdf.org) aims to solve some of these data storage and manipulation challenges by using the established, open-source HDF5 (http://www.hdfgroup.org) binary file format to store NGS data.

BioHDF extends HDF5 data structures and library routines with new features to support the high-performance data storage and computation requirements of next-generation sequencing. The open-source, BSD-licensed tools support the storage of sequences, their alignments against reference data sources and annotations such as SNP or splice variation analysis. Multiple NGS platforms are supported including Applied Biosystems SOLiD, Illumina Genome Analyzer, Roche 454, and Helicos.

Recent developments in the BioHDF toolset include NCList-like indexing(Alekseyenko and Lee 2007) for correct, efficient retrieval of gene annotation and alignment data and the ability to read and write SAM data(Li, Handsaker et al. 2009) for easier integration into existing toolchains. Our roadmap for the next year is also presented, including the new BioHDF API and scalability/performance improvements.

Alekseyenko, A. V. and C. J. Lee (2007). "Nested Containment List (NCList): a new algorithm for accelerating interval query of genome alignment and interval databases." Bioinformatics **23**(11): 1386-1393.

Li, H., B. Handsaker, et al. (2009). "The Sequence Alignment/Map format and SAMtools." Bioinformatics **25**(16): 2078-2079.

Concurrent Bioinformatics Software for Discovering Genome-wide Patterns and Word-based Genomic Signatures

Jens Lichtenberg[1], Kyle Kurz[1], Xiaoyu Liang[1], Rami Al-ouran[1], Lev Neiman[1], Lee Nau[1], Joshua Welch[1], Edwin Jacox[2], Thomas Bitterman[3], Klaus Ecker[1], Laura Elnitski[2], Frank Drews[1], Stephen Lee[4], Lonnie Welch[1,5,6,§]

1 - Bioinformatics Laboratory, School of EECS, Ohio University, Athens, Ohio, USA
2 - Genomic Functional Analysis Section, National Human Genome Research Institute, NIH, Rockville, MD, USA
3 - Department of Statistics, University of Idaho, Moscow, Idaho, USA
4 - Cyberinfrastructure Group, Ohio Supercomputer Center, Columbus, OH, USA
5 - Biomedical Engineering Program, Ohio University, Athens, Ohio, USA
6 - Molecular and Cellular Biology Program, Ohio University, Athens, Ohio, USA

§ - Presenting author: welch@ohio.edu

URL for the overall project web site: www.word-seeker.org
URL for accessing the code: http://word-seeker.googlecode.com/svn/trunk
Open Source License being used: GNU General Public License v3

The importance of discovering the patterns and features in genomic sequences is motivated by a number of problems in biology. The Encyclopedia of DNA Elements project, ENCODE, seeks 'to identify all functional elements in the human genome sequence'. The study of co-regulated genes involves the analysis of the promoter sequences, introns, and UTRs of genes that were determined, by microarray experiments, to be co-regulated. Similarly, transcription factor binding regions identified by ChIP-chip and ChIP-seq experiments are examined to identify genomic patterns. Genome-wide pattern discovery studies seek to identify vocabularies of genomes. The search for genomic signatures seeks unique elements that characterize specific organisms, tissues, pathways, and functions. A number of algorithms and software tools have been developed to address some of these problems. However, very few provide the scalability needed to process large (genome-scale) data sets. Furthermore, none provides the comprehensive, integrated set of capabilities required by the complete set of biological problems mentioned above.

This manuscript presents WordSeeker, a general purpose, concurrent software suite that addresses these shortcomings. The Open Word Enumeration Framework (OWEF) class performs a central role in WordSeeker. When it was presented at BOSC 2009, OWEF had been used only to support a single data structure for word enumeration. Since that time, it has been employed successfully with three different data structures (radix tree, suffix tree, and suffix array). Additionally, it has been deployed on multi-core and distributed computational platforms. The concurrent implementation of WordSeeker divides the data space among nodes by using nucleotide prefixes. A controller task coordinates the activities of the worker nodes, each of which enumerates a subset of the DNA *word space*. To build a distributed Markov chain model for the computation of word scores, the nodes communicate with each other to obtain word occurrence information. WordSeeker is deployed on the Ohio Supercomputer Center's *Glenn* cluster, which is an IBM e1350 system with more than 4200 Opteron processor cores that are connected by 20 Gbps Infiniband.

To measure the performance enhancement due to concurrency, WordSeeker's performance was evaluated on a 5 node distributed system and on a single node. Execution time was measured for the 27,167 core promoters of *A. thaliana*, and for the entire *E. coli* genome. Two different algorithms, radix tree and suffix tree, were compared. Table 1 shows the results for DNA words of length 20.

Table 1. Performance measurements for a distributed implementation of WordSeeker.

		5 Nodes						1 Node					
		Radix Tree			Suffix Tree			Radix Tree			Suffix Tree		
Data Set	Word Length	Complete (h:m:s)	Enum. (secs)	Scoring (secs)	Complete (h:m:s)	Enum. (secs)	Scoring (secs)	Complete (h:m:s)	Enum. (secs)	Scoring (secs)	Complete (h:m:s)	Enum. (secs)	Scoring (secs)
A. thaliana Core Prom.	20	0:05:10	3.74	303.51	0:05:39	20.74	315.43	0:45:56	29.69	6247.6	0:50:27	126.63	6218.3
E.coli Genome	20	0:01:18	5.49	68.89	0:01:27	13.03	70.15	0:03:48	60.4	165	0:03:23	78.65	134.57

Automated Annotation of NGS Transcriptome Data using ISGA and Ergatis

Aaron Buechlein, Jeong-Hyeon Choi_, Karthik Muthuraman, Chris Hemmerich,
Center for Genomics and Bioinformatics, Indiana University (chemmeri@indiana.edu)

Project: http://isga.cgb.indiana.edu
Download: https://cgb.indiana.edu/downloads/
License: Apache Version 2.0

With next-generation sequencing technologies, a single 454 run can generate roughly 400,000,000 base pairs of transcriptome reads and a Solexa run can generate 250 times as many bases for use against a reference genome. For high-throughput research, automated tools must be used for annotation and analysis to match the pace of sequencer output. Even for biologists working with a single sequencing run, automated annotation results can provide immediate insights while labor-intensive manual curation is performed. As an evolution of our previous work on EST analysis (http://estpiper.cgb.indiana.edu) and prokaryotic genome annotation (http://isga.cgb.indiana.edu), we have built a transcriptome annotation pipeline for suitable for use with NGS data.

In our pipeline, a transcriptome comprised of contigs and singletons is compared to public databases such as non-redundant protein and EST sequence databases using BlastX. Well matching contigs and singletons to those databases are further analyzed for taxonomic distribution, GO terms, metabolic pathways, orthology and paralogy, gene duplication, and alternative splicing variants. If a close, sequenced genome is available, comparison with the genome reveal a minimal gene set in the transcriptome. Unmatched transcriptomic sequences are fed to ORF prediction programs such as ORFpredictor to find possible protein coding frames. Since contigs and singletons are parts of a gene, they are clustered using splice junction reads and an orthology database. If a transcriptome is sequenced from different organs and individuals, the transcriptome can be used to identify sequence variants such as SNPs.

The Integrative Services for Genomic Analysis (ISGA) web application is a solid platform on which to develop this pipeline. ISGA was originally developed for prokaryotic genome annotation, and provides an intuitive interface for biologists to run and customize pipelines. ISGA has a simple account system to ensure that users data is private, and allows them to easily retrieve the results from previous experiments. In addition, ISGA provides a "Tool Box" for visualizing and further analyzing pipeline results using tools such as GBrowse and Blast.

ISGA uses Ergatis (http://ergatis.sourceforge.net/) to execute and manage the provided bioinformatics pipelines. Ergatis is a workflow management system for bioinformatics utilities used at the J. Craig Venter Institute, the Institute for Genome Sciences, and other institutions, and achieves high-throughput pipeline execution through automation and leveraging distributed computing resources. Ergatis provides the necessary tools for creating a complex bioinformatics pipeline, closely monitoring its execution, and efficiently recovering from errors.

From Moby to SADI - Modeling Semantic Web Services with the Semantic Automated Discovery and Integration Framework

Mark Wilkinson, Benjamin Vandervalk, Luke McCarthy, Edward Kawas, David Withers
Heart + Lung Institute at St. Paul's Hospital, University of British Columbia, Vancouver, BC, Canada
Email: markw@illuminae.com **Project**: http://sadiframework.org **Code**: http://sadiframework.org/content/links-and-docs/

In 2001 the BioMoby project was established. It's goal was to define a framework for modeling, annotating, and discovering Web Services to improve interoperability between bioinformatics resources. A key aspect of the Moby solution was a centralized ontology describing bioinformatics data structures; all data within the BioMoby system had to be represented as instances of these ontological classes. The Object Ontology, while being the key to Moby's success, was also the most mis-understood, mis-used, and widely disliked feature of the Moby solution. This monolithic structure conflated semantics and syntax in a manner that was confusing for newcomers to the project. In addition, Moby data, while being highly predictable in structure, could not be represented in XML Schema, thus making existing Web Service toolkits largely unusable. For these and other reasons BioMoby reached a peak of ~1500 Services in 2008, but has not seen any appreciable adoption since then.

In 2004, the W3C announced their recommendation of RDF (Resource Description Framework) and OWL (Web Ontology Language) for syntax and semantics , respectively, on the Semantic Web. Using our experiences with BioMoby as a requirements-guide, and utilizing these new Semantic Web technologies, we have designed and implemented a novel Semantic Web Service Framework – SADI (Semantic Automated Discovery and Integration). SADI adheres closely to existing standards, eliminates troublesome Web Service technologies such as SOAP, embraces the openness and distributed nature of the Web, while offering greater discovery power and interoperability than we achieved in BioMoby.

The core of the SADI "technology" are three simple, lightweight best-practices for modeling Web Services on the Semantic Web:

1. Web Services should consume and produce data in RDF syntax, consistent with how data is represented on the Semantic Web.
2. Web Services should consume OWL Individuals ("instances") of one ontological Class, and should generate OWL Individuals of another ontological Class. These two owl Classes, therefore, define the interface for that Web Service in much the same way as the "message" and "types" portion of a WSDL document define traditional Web Service interfaces. This OWL document is published openly on the Web for indexing by any mechanism.
3. The URI (Uniform Resource Identifier – usually a URL) of the OWL Individual passed as input to a service must be the same as the URI of the OWL Individual that is returned as output. **This best-practice is key to all of the semantic service discovery behaviours of SADI.**

The SADI project makes following open-source codebases available from its project homepage: http://sadiframework.org under the New BSD License. Perl modules are available from CPAN.

- SADI::SADI - Perl support for creating SADI-compliant services in Perl. The SADI libraries use our PLUTO module (also on CPAN) to automatically create Perl objects corresponding to OWL Class definitions. Consuming input and creating output data for a SADI service simply requires getting/setting values on these objects, with no direct manipulation of RDF.
- SADI for Java: tools for building SADI clients and services in Java, including Maven plugins for service generation and testing and a rich client API for discovering and invoking SADI services based on the Jena Semantic Web framework.
- SHARE for Java: a Java library for dynamic resolution of SPARQL queries using publicly-available SADI services.
- SADI Taverna Plugin: enables semantically-aided discovery of SADI services to add into Taverna workflows (LGPL 2.1 license; bundled with Taverna)
- SADI Protegé Plugin: Currently only in prototype, this integrates creation, testing, and deployment of SADI services in both Java and Perl into the Protegé ontology editor.

ONTO-ToolKit: enabling bio-ontology engineering via Galaxy

Erick Antezana, Aravind Venkatesan, Vladimir Mironov and Martin Kuiper

Department of Biology, Norwegian University of Science and Technology,
Høgskoleringen 5, N-7491 Trondheim, Norway.
venkatesan@nt.ntnu.no

Project page: http://bitbucket.org/easr/onto-toolkit/wiki/Home
Open Source License: GNU General Public License
http://bitbucket.org/easr/onto-toolkit/src/tip/README

Abstract

Biological data integration is a corner stone for systems biology approaches. Data integration is supported by a diverse series of tools, but still the lack of a consistent terminology to label these data presents significant hurdles, causing much of those biological data to remain disconnected or worse: to become misconnected. Bio-ontologies are being developed as a means to overcome those terminology issues. OBOF, RDF and OWL are among the most used ontology formats to capture terms and relationships in the Life Sciences. The Semantic Web also promises to support data integration and further exploitation of integrated resources via automated reasoning procedures.

Here we present ONTO-ToolKit, which is an extension to the existing PERL suite ONTO-PERL (http://search.cpan.org/dist/ONTO-PERL/), supporting the handling of OBO-formatted ontologies. ONTO-ToolKit is distributed as a set of Galaxy (http://galaxy.psu.edu/) tools. It provides not only a user friendly interface, via Galaxy, to manipulate OBO ontologies but also opens up the possibility to perform further biological (and ontological) analyses by using other tools provided within the Galaxy platform. Moreover, it provides some tools to translate OBO-formatted ontologies into Semantic Web formats such as RDF and OWL. Finally, it provides an interface (currently under development) to launch SPARQL queries from within Galaxy.

We present a couple of use cases to illustrate how the functionality of ONTO-PERL could be combined with the functionality of other tools from Galaxy. One of the use cases illustrates the functionality of ONTO-PERL to identify all the upstream terms (ancestors) of a particular Gene Ontology (GO) term to extend the information contained in the Gene Ontology Annotation (GOA) files. The other example illustrates how ONTO-ToolKit can be used to identify overlapping annotations for a given pair of proteins. Such an overlap would suggest that these proteins share a bio-molecular function, cellular localization or biological process.

G-language Bookmarklet: a gateway for Semantic Web, Linked Data, and Web Services

Kazuharu Arakawa[1] (gaou@sfc.keio.ac.jp), Nobuhiro Kido[1], Kazuki Oshita[1], and Masaru Tomita[1]

[1] Institute for Advanced Biosciences, Keio University, Fujisawa, 252-8520, Japan

In order to efficiently navigate and query through the huge masses of biological information, concepts of Linked Data and Semantic Web are gaining momentum as promising means for data integration. In light of the advent of these new data representation models, we here present a bookmarklet that provides an intuitive user interface for accessing the Linked Data and for querying the resources of Semantic Web, through a graphical ring-shaped menu on any webpage that the user is browsing. G-language Bookmarklet is implemented as a bookmarklet to seamlessly work with regular web browsing, runs on any modern browsers, and can be invoked from any websites, without the need for installation of software or any specialized browser plugins. By selecting keywords of interest within any webpage and by opening the G-language Bookmarklet, an array of icons in the shape of a ring appears with animation on top of the webpage that the user is currently browsing. Here the users can select the database to search with the selected keyword, such as Wikipedia, Google, NCBI Entrez, Pubmed, KEGG, and Bio2RDF. Results of the queries are readily shown as another ring of icons representing the top hits of the query, and when the query reaches a single entry, users are redirected to the webpage of that entry. Likewise the queries, users can also access web services such as BLAST and G-language REST services from the bookmarklet. The bookmarklet is freely available at: http://www.g-language.org/wiki/bookmarklet.

Semantic Web and Linked Data are generally accepted as the promising means for data integration in biology, lead by initiatives such as Concept Web Alliance, Banff Manifesto, SADI, and DBCLS BioHackathon 2010. G-language Bookmarklet aims to provide a gateway to these new technologies for the end-users, by providing an intuitive interface with icons and animations that works on any web pages without the need for installation.

Each data in Semantic Web is represented as the Triple (Subject - Predicate - Object), and the connection of these triples form a gigantic graph of linked data. Therefore, a feasible interface to query the data is to start from a keyword to find matching Subjects, display a list of related Predicates, and jump to the list of Objects. Ring interface is an effective implementation for this purpose, whereby the Predicates and Objects are displayed as a series of rings. Semantic Web would be a complement to existing data, as opposed to being an immediate replacement. We therefore provide a unique and consistent intuitive user interface for web search engines, linked data, semantic web, and web services. In this way, users can take advantage of the Semantic Web and Linked Data coherently with existing data.

URL(project): http://www.g-language.org/wiki/bookmarklet
URL(code): http://www.g-language.org/cube/cube.js
License: MIT License

Connecting TOPSAN to Computational Analysis

Christian M Zmasek[2,5], Kyle Ellrott[3], Dana Weekes[1,2], Constantina Bakolitsa[1,2], John Wooley[3], Adam Godzik[1,2,3,4]

[1] Joint Center for Structural Genomics, http://www.jcsg.org/
[2] Sanford-Burnham Medical Research Institute, La Jolla, California, USA
[3] University of California, San Diego, La Jolla, California, USA
[4] Joint Center for Molecular Modeling, http://jcmm.burnham.org/
[5] czmasek@burnham.org

Project Site: http://www.topsan.org
Software: http://www.topsan.org/Tools

The National Institute of Health's Protein Structure Initiative (PSI) has produced over four thousand protein structures. As a member of that initiative, the Joint Center of Structural Genomics (JCSG) uses TOPSAN to organize the annotations of these proteins. TOPSAN, the wiki based protein annotation system, sits at an important nexus between computationally generated protein annotations and human expertise. The information in TOPSAN is both the product of and a source to biological computational analysis. Unlike Protopedia, the goal of which is to catalog established literature about proteins in Wiki form, TOPSAN's primary purpose is to promote rapid theoretical discussion about protein structures.

TOPSAN has successfully provided a link from the library of protein information generated by the JCSG program to human curators. On the other hand, it is also important to close the loop and bring that human expertise to feed back into the database of information that JCSG is producing. To this end, we are attempting to draw more links from protein descriptions texts written in TOPSAN to the organized databases that are used to annotate proteins. One example of this is the effort to map TOPSAN proteins to standard ontologies, such as the Gene Onotology (GO) classifications. GO terms have been used to link protein families into large hierarchical groups related by function.

GO annotation will be done manually by the curators. In order to facilitate this effort we are beginning to provide automatic suggestions that the curators can review and approve. The first stage of this is to use the existing Pfam to GO mappings to take care of trivial cases. In situations where there are no existing mappings, we use automatic text scanning to generate suggestions for the user. Using published literature, GO terms will be suggested using standard indexing/searching strategies.

There is also a need to connect the work being done by the annotators to other sources of data. By linking annotations to semantic web compliant databases, TOPSAN becomes connected to a larger set of databases. This is done via a simple in-line notation embedded in the text of the wiki article, that is then automatically exported to RDFa and other semantic web compliant technologies.

More importantly, semantic web technologies will also allow TOPSAN to be interrogated computationally. A piece of software could, for example, easily extract a list of all proteins from a particular organism and/or of a given function. The program could use described protein interaction links to extract all available information about a pathway and then perform further analysis on the data delivered by TOPSAN.

The wiki platform provides a system that allows for rapid updates to data, while the semantic web enables a system for the representation of data in a way that is both easy to extract and malleable enough to allow a variety of new data sources to be integrated.

Musite: Global Prediction of General and Kinase-Specific Phosphorylation Sites

Jianjiong Gao[1,2,*], Jay J. Thelen[2,3], A. Keith Dunker[4], and Dong Xu[1,2]

[1]Department of Computer Science, [2]C.S. Bond Life Sciences Center, [3]Department of Biochemistry, University of Missouri, Columbia, Missouri 65211, [4]Center for Computational Biology and Bioinformatics, Indiana University Schools of Medicine and Informatics, Indianapolis, Indiana 46202, USA

* Email: jgao@mail.mizzou.edu
Project URL: http://musite.sourceforge.net/
Source code: http://musite.svn.sourceforge.net/viewvc/musite/musite/

Reversible protein phosphorylation is one of the most pervasive posttranslational modifications, regulating diverse cellular processes in various organisms. Since mass spectrometry-based experimental approaches for identifying phosphorylation events are costly, time consuming, and are biased towards abundant proteins and proteotypic peptides, *in silico* prediction of phosphorylation sites is an attractive alternative for whole proteome annotation. Due to various limitations, current phosphorylation-site prediction tools were not well-designed for comprehensive assessment of proteomes. Here, we present a novel software tool, Musite, specifically designed for large-scale prediction of both general and kinase-specific phosphorylation sites. We collected high confidence phosphoproteomics data from multiple organisms and used these to train prediction models by a comprehensive machine learning approach. Application of Musite on proteomes of *Homo sapiens*, *Mus musculus*, *Drosophila melanogaster*, *Caenorhabditis elegans*, *Saccharomyces cerevisiae*, and *Arabidopsis thaliana* yielded tens of thousands of phosphorylation-site predictions at a high stringency level. Cross-validation tests show that Musite significantly outperforms existing tools for predicting general phosphorylation sites and is at least comparable to those for predicting kinase-specific phosphorylation sites. Furthermore, Musite provides several other unique functionalities such as customized model training and continuous stringency selection by users. Musite provides a useful bioinformatics tool to biologists for predicting phosphorylation sites *en masse* and training prediction models from custom phosphorylation data. In addition, with its easily-extensible open-source application programming interface (API), Musite is aimed at being an open platform for community-based development of machine-learning based phosphorylation-site prediction applications. Musite is available at http://musite.sourceforge.net/.

Acknowledgement

This work was supported, in part, by the funding from the National Science Foundation-Plant Genome Research Program [grant number DBI-0604439 awarded to JJT] and the National Institute of Health [grant number R21/R33 GM078601 awarded to DX]. The authors thank Dr. Thorsten Joachims and Dr. Weizhong Li for permissions of integrating SVM^{light} and CD-HIT in Musite.

Reference

Gao, J., Thelen, J.J., Dunker, A.K., and Xu, D. Musite: a Tool for Global Prediction of General and Kinase-Specific Phosphorylation Sites. *Molecular & Cellular Proteomics*. 2010. Submitted.

Towards a Modern BioPerl: BioPerl Update 2010

Christopher Fields[1], Mark Jensen[2], Jason Stajich[3], and the BioPerl Core Developer Team

[1] The Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana IL 61801
[2] SRA International, Washington, DC
[3] Dept. of Plant Pathology and Microbiology, University of California-Riverside, Riverside CA, 92521

Bioinformatics Open Source Conference, Boston, MA, USA

The BioPerl project (http://bioperl.org, distributed under the Perl Artistic License) is now 15! In this talk, we will present the current state of BioPerl, outlining how BioPerl started, where we are now, and where we intend to go in the near future.

Initially focusing on the present, we will discuss how the BioPerl developers are currently addressing new biological and informatics-based problems, such as use of second-generation sequencing tools like MAQ and BowTie. Furthermore, productive collaborations with other OBF groups will be highlighted, such as the recent publication on FASTQ, ongoing work with BioLib, and the Google Summer of Code. A brief progress report will be given on the BioPerl Google Summer of Code by Jun Yin to refactor the BioPerl alignment architecture.

For the future, we will discuss possible strategies intended to address the current monolithic nature of BioPerl's core, including the possibility of moving towards a more modular design. The use of modern developer and Perl tools with BioPerl will be covered, such as git and GitHub, Moose, and DBIx::Class. Finally, we will give a brief glimpse of BioPerl6; yes, it exists.

BioPerl can now be found on GitHub at http://github.com/bioperl.

BioRuby 2010 updates: moving to agile bioinformatics

Raoul J.P. Bonnal[1], Naohisa Goto[2], Mitsuteru Nakao[3], Jan Aerts[4], Pjotr Prins[5] ,Toshiaki Katayama[6]

1 Fondazione INGM, Milan, Italy bonnalraoul@ingm.it; 2 Research Institute for Microbial Diseases, Osaka University, Japan; 3 Database Center for Life Science, Tokyo, Japan; 4 Wellcome Trust Sanger Institute, Cambridge, UK; 5 Wageningen University and Groningen Bioinformatics Center, Netherlands; 6 Institute of Medical Science, University of Tokyo, Japan

URL: http://bioruby.org
Code: http://github.com/bioruby/bioruby
License: The Ruby License

BioRuby provides tools and libraries for the Ruby programming language with the aim of providing an integrated environment for bioinformatics. The Ruby programming language was incepted around 1993 and is a reflective, general purpose object-oriented programming language that combines syntax inspired by Perl, with Smalltalk-like features (http://www.ruby-lang.org). Ruby has gained in popularity by virtue of its clean object oriented programming (OOP) design and its functional programming characteristics. These characteristics allow programming with less source code, thereby improving programmer productivity (Aerts and Law, 2009). The BioRuby project is started in 2000 and after the initial release in 2001, we made a steady development to its functionality. The project is supported by the Open Bioinformatics Foundation (OBF) initiative, which hosts the website and mailing list. The current software development tree is hosted on GitHub (https://github.com/), a public distributed source control system, which allows any developer to start contributing code to the BioRuby project.

Here we report recent developments in BioRuby. Since the last BOSC presentation in 2007 we had two major releases, 1.3.0 and 1.4.0. Firstly, we enhanced the support for web services as many bioinformatics resources are now being provided through SOAP and REST services. Notable efforts are made to utilize REST web services like the ones provided by TogoWS (http://togows.dbcls.jp/) and NCBI (http://eutils.ncbi.nlm.nih.gov/). Sponsored by the DBCLS during the BioHackathons in 2008, 2009, and 2010 we then tried to define next challenges for BioRuby. We made a lot of refactoring to ensure the correctness of file formats with conversion among rich annotated sequence objects. Internally, we introduced the ActiveRecord model provided by the Ruby on Rails (http://rubyonrails.org) framework for the BioSQL (http://biosql.org/) module to abstract the RDBMS layer inside. With the help of Google Summer of Code (http://code.google.com/soc/) and National Evolutionary Synthesis Center (NESCent http://www.nescent.org/) in 2009, BioRuby can handle PhyloXML data efficiently. A lot of bioinformatics centers are using or acquiring data from Next Generation Sequencers (NGS), so the FASTQ file formats is supported in all of its three variants, and supports for DNA chromatogram data in SCF and ABIF formats have been addded to the framework. Most recently, in collaboration with other Open Bio* projects like BioPython, we started to implement new Semantic Web technologies like RDF and SPARQL for the bioinformatics data mining. Finally, we adopted the Ruby Unit Testing framework to improve the quality of our code, and to break the barriers for newbies, all the documentation has been revised adding new tutorials and a better code description.

BioRuby hopes to have more contributors without influencing the stable core of the framework, so we'll introduce a plugin system to embrace the "agile" development reflecting the Rails' approach. This plugin system gives the chance for both BioRuby developers and other bioinformaticians to try the beta status code. This mechanism shows a potential when data needs not just to be analyzed in an efficient and easy way but, also presented in the right way; usually that means a fancy way. Bio::Graphics (http://bio-graphics.rubyforge.org/) is an example, which adds functionality to graphically represent biological objects. It fits exactly into the plugin philosophy and introducing the concept of view (in MVC model) into any BioRuby object that can be exported or published over the web. To handle the plugin system, we will improve the BioRuby shell and promote the integration with Rails. Heterogenous data integration is another aspect that should be more "agile", introducing new technologies like semantic web and supporting more web-services could reduce the needs to replicate information.

EMBOSS: The European Molecular Biology Open Software Suite

Peter Rice (pmr@ebi.ac.uk), Alan Bleasby, Jon Ison, Mahmut Uludag
European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, United Kingdom.

The European Molecular Biology Open Software Suite (EMBOSS) is a mature package of software tools developed for the molecular biology community. It includes a comprehensive set of applications for molecular sequence analysis and other tasks and integrates popular third-party software packages under a consistent interface. EMBOSS includes extensive C programming libraries and is a platform to develop and release software in the true open source spirit.

A major new stable version is released each year and the current source code tree can be downloaded via CVS. All code is open source and licensed for use by everyone under the GNU Software licenses (GPL with LGPL library code).

There have been many tens of thousands of downloads including site-wide installations all over the world since the project inception. EMBOSS is used extensively in production environments reflecting its mature status and has been incorporated into many web-based, standalone graphical and workflow interfaces including Galaxy, wEMBOSS, EMBOSS Explorer, JEMBOSS, SoapLab, Pise, SRS, Taverna and several commercial workflow packages.

EMBOSS 6.3 will be released on 15th July 2010 (we always release on 15th July). New features include:

- New data formats - standard FASTQ Open-Bio parsing
- Use of ontologies - new EDAM ontology for data types and methods in EMBOSS
- DAS, BioMart and Ensembl protocol support
- Integration support for Galaxy

Three books are to be published by Cambridge University Press for Users, Developers and Administrators of EMBOSS.

The EMBOSS project has significant new funding for an ambitious programme of extensions and new applications covering:

- Comprehensive coverage of public data (sequence data, linked data resources)
- Access methods for major public data repositories (Ensembl, UCSC, CHADO, BioMart, SOAP, REST)
- Persistent metadata (coordinates, taxonomy, gene ontology, keywords, citations)
- Genome-scale analysis and annotation
- Query language
- 100+ new applications

Project home page: http://emboss.sourceforge.net/
Release download site: ftp://emboss.open-bio.org/pub/EMBOSS/
Anonymous CVS server: http://www.open-bio.org/wiki/SourceCode

Biopython Project Update

Brad Chapman[*], Peter Cock[†], *et al.*
chapmanb@50mail.com

Bioinformatics Open Source Conference (BOSC) 2010, Boston, MA, USA

In this talk we present the current status of the Biopython project (www.biopython.org), described in a application note published last year (Cock *et al.*, 2009). Biopython celebrated its 10th Birthday last year, and has now been cited or referred to in over 150 scientific publications (a list is included on our website).

At the end of 2009, following an extended evaluation period, Biopython successfully migrated from using CVS for source code control to using git, hosted on github.com. This has helped our existing developers to work and test new features on publicly viewable branches before being merged, and has also encouraged new contributors to work on additions or improvements. Currently about fifty people have their own Biopython repository on GitHub.

In summer 2009 we had two Google Summer of Code (GSoC) project students working on phylogenetic code for Biopython in conjunction with the National Evolutionary Synthesis Center (NESCent). Eric Talevichs work on phylogenetic trees including phyloXML support (Han and Zamesk, 2009) was merged and included with Biopython 1.54, and he continues to be actively involved with Biopython. We hope to include Nick Matzkes module for biogeographical data from the Global Biodiversity Information Facility (GBIF) later this year. For summer 2010 we have Biopython related GSoC projects submitted via both NESCent and the Open Bioinformatics Foundation (OBF), and if all goes well we will again have summer students working on Biopython.

Since BOSC 2009, Biopython has seen four releases. Biopython 1.51 (August 2009) was an important milestone in dropping support for Python 2.3 and our legacy parsing infra-structure (Martel/Mindy), but was most noteworthy for FASTQ support (Cock *et al.*, 2010). Biopython 1.52 (September 2009) introduced indexing of most sequence file formats for random access, and made interconverting sequence and alignment files easier. Biopython 1.53 (December 2009) included wrappers for the new NCBI BLAST+ command line tools, and much improved support for running under Jython. Our latest release is Biopython 1.54 (May 2010), new features include Bio.Phylo for phylogenetic trees (GSoC project), and support for Standard Flowgram Format (SFF) files used for 454 Life Sciences (Roche) sequencing.

Biopython is free open source software available from www.biopython.org under the Biopython License Agreement (an MIT style license, `http://www.biopython.org/DIST/LICENSE`).

References

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

Han, M.V. and Zmasek, C.M. (2009) phyloXML: XML for evolutionary biology and comparative genomics. *BMC Bioinformatics* **10**:356. doi:10.1186/1471-2105-10-356

Cock, P.J.A., Fields, C.J., Goto N., Heuer, M.L., and Rice, P.M. (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res* **38**(6) 1767-71. doi:10.1093/nar/gkp1137

[*]Bioinformatics Core Facility, Molecular Biology Department, Massachusetts General Hospital, Boston, MA, USA
[†]Plant Pathology, SCRI, Invergowrie, Dundee DD2 5DA, UK

interPopula: Database and tool integration for population genetics using Python

Tiago Antao

Liverpool School of Tropical Medicine, University of Liverpool, UK. E-mail: tiagoantao@gmail.com

URL: http://popgen.eu/soft/interPop – Code URL: https://launchpad.net/interpopula – License: GPL v3

Population genetics suffers from a structural lack of Application Programming Interfaces (APIs) to interact with existing public databases. InterPopula address that problem by providing a Python library to interact with several of such databases. The initial version of the library and supporting scripts is mainly concerned with human population genetics and genomics by providing support to the HapMap project and also the UCSC Know Genes database (a part of the UCSC Genome Browser). HapMap – http://hapmap.ncbi.nlm.nih.gov/ – is a freely available dataset of human DNA sequence variation (SNP based) currently covering 11 different human populations around the globe. UCSC Know Genes is a database constructed from Swiss-Prot/TrEMBL (UniProt) and the associated mRNA data from Genbank using a fully automated process to predict genes along a genome.

InterPopula has three distinct objectives:

1. Provide an API to access the HapMap dataset and the UCSC Know Genes database (and, in the future, other UCSC Genome Browser tables). Currently, and as far as we know, there is no alternative API to interPopula, in any language, to access HapMap data. The Known Gene database (for which no API is also known, though it is possible to interact with the database with an SQL interface) supports more species, therefore making interPopula of (limited) use also for non-human studies. The API also allows for the export of HapMap data in Genepop file format. This format, being the de facto standard in non-sequence based population genetics, permits the analysis of HapMap data using a vast array of software applications which are able to import Genepop files.

2. Make available a set of scripts – based on the above library – that can serve not only as useful utilities, but also as examples of database and external tool integration. Currently we provide examples of integration with Entrez databases (nucleotide and SNP), the Genepop population genetics suite and charting libraries. Integration with Entrez databases and Genepop is achieved through Biopython.

3. A set of guidelines and scripts was developed in order to facilitate a consistent view across heterogeneous databases. HapMap, UCSC Known Gene and the Entrez databases might not be fully consistent among themselves and, if care is not taken, database integration efforts might lead to erroneous results.

InterPopula includes a core library, a set of scripts and documentation. The code includes unit testing support in order to maintain code quality. The development infrastructure is based on distributed version control (Bazaar) over the Launchpad hosting platform.

Future development plans include the support of more databases from the UCSC Genome Browser. As this repository supports multiple species, this will make a interPopula less centered on humans and more a general purpose (multi-species) population genetics suite. Long term efforts might include supporting other databases, depending on user feedback.

Title: Bioconductor with Python, What else ?

Authors: Laurent Gautier

Affiliations: CBS/DMAC, Department of Systems Biology, Technical University of Denmark, laurent@cbs.dtu.dk

URL (project): `http://pypi.python.org/pypi/rpy2-bioconductor-extensions/`

URL (downloads): `http://pypi.python.org/pypi/rpy2-bioconductor-extensions/`

License: AGPLv3.0

Abstract: The Bioconductor project has become a reference for the numerical processing and statistical analysis of data coming from high-throughput assays, providing a rich selection of methods and algorithms to the research community. Within the same time, Python has matured as a reliable platform for prototype development and data handling, with for example the biopython project providing the later for bioinformatics.

Building atop a bridge that allows the use of the R libraries from Python, we present an interface to Bioconductor data structures and function that uses an object-oriented paradigm familiar to Python users. This allows a seamless integration of the bioconductor project into existing Python infrastructure, be it for bioinformatics or web development with frameworks or GUI development.

The current implementation proposes Python-class representations for a large fraction of the core bioconductor infrastructure packages for the analysis of microarray and next-generation sequencing, and any other bioconductor package, or R package, is otherwise available through the underlying Python-R bridge *rpy2*. The design philosophy adopted allows to follow up with bioconductor as it continues developping, and we demonstrate with examples the benefits of the approach. In the face of exploding volumes for biological data generated, agile development can help link skills such as data analysis, data representation, user interfaces and prototype development, and we believe that Python will continue growing in that space.

Taking Python as a possible glue language, and now embedding what is probably the largest library of functions for data analysis and statistics in bioinformatics data, we outline perspectives for bioinformatics analysis frameworks in the age of omnipresent data.

Bio.Phylo: A unified phylogenetics toolkit for Biopython

Eric W Talevich*[1] and Brad A Chapman[2]

[1]Institute of Bioinformatics, University of Georgia, 120 Green Street, Athens, GA 30602
[2]Massachusetts General Hospital, 55 Fruit Street, Boston, MA 02114
*Presenting author e-mail: etal@uga.edu

Project web site: http://biopython.org
Source code: http://github.com/biopython/biopython
License: Biopython Public License

Background: The number and range of tools developed for phylogenetic analysis has expanded dramatically, creating new opportunities to combine results with other sources of information to obtain an enhanced evolutionary perspective. This leads to the additional challenge of integrating data found in a wide variety of formats. PhyloXML, an XML-based file format for richly annotated phylogenetic data, is one effort to address this problem. However, for researchers to benefit from standardization, there must also be a complementary software ecosystem that can read, manipulate and transform this information into the various forms required to build computational pipelines.

Results: We developed a Biopython software library capable of parsing common file formats for phylogenetic trees, performing basic transformations and manipulations, attaching rich annotations, and visualizing trees. Parsing and serialization code is separated from the internal tree object representation; this allows us to support a set of common operations on trees independent of the source format, as well as convert between formats. The complete phyloXML specification is implemented, providing full interoperability with popular tools such as Archaeopteryx. Nexus and Newick support was obtained through a refactoring of the Bio.Nexus module by Cymon J. Cox and Frank Kauff. Several mechanisms for displaying trees were also implemented, taking advantage of existing libraries for visualization (matplotlib) and graph manipulation (NetworkX and Graphviz).

Conclusions: Bio.Phylo meets a growing need in bioinformatics for working with heterogeneous types of evolutionary data. By supporting interoperability with multiple file formats and leveraging existing Biopython features, this library simplifies the construction of phylogenetic workflows.
We also acknowledge the role of Google Summer of Code and the NESCent Phyloinformatics program in sponsoring and initiating this project.

Availability: Bio.Phylo is included with Biopython version 1.54.

The Microsoft Biology Foundation

Simon Mercer, Michael Zyskowski and Bob Davidson
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA – msrerbio@microsoft.com

http://mbf.codeplex.com

The Microsoft Biology Foundation (MBF) is a library of common bioinformatics and genomics functionality built on top of the .NET framework. Functions include parsers and writers for common bioinformatics file formats, connectors to common web services, and algorithms for assembling and aligning DNA sequences. The project is released under the OSI-compliant MS-PL open source license (http://opensource.org/licenses/ms-pl.html) and is available for download from http://mbf.codeplex.com. The MBF project is guided by the user community through a Technical Advisory Board drawn from academia and commerce, with responsibility to maintain code quality and steer future development to respond to the needs of the scientific community. MBF is a community-led and community-curated project, and encourages bug fixes, feature requests and code contributions from all members of the commercial and academic life science community.

The .NET framework provides some advantages in terms of development and use of MBF – code can be added in any .NET compliant language (there are currently around 70), and the functions in MBF can be accessed in many different ways – compiled into an executable, wrapped as workflow activities, accessed on the command-line using a scripting language such as Python, or used in Microsoft Office plug-ins to add biological functionality to applications such as Microsoft Word and Excel.

Microsoft researchers frequently apply computer research to challenges in biology and this leads to the development of unique scientific tools – for example machine learning has been applied to the challenges of HIV vaccine design, leading to the creation of a range of tools relating to epitope binding prediction, phylogenetics, haplotyping and population studies (http://mscompbio.codeplex.com). MBF is being actively used and extended by Microsoft Research projects such as these, as well as by the wider academic and commercial community.

Title: Building Bioinformatics Web Applications with Clickframes
Authors: William Crawford, Vineet Manohar, Jonathan Abbett, Steven Boscarine, Nicole Zanetti, Evan Pankey
Email: william.crawford@childrens.harvard.edu
Affiliation: Children's Hospital Boston, Informatics Solutions Group
Website: http://clickframes.org/
Code: clickframes-php-0.9.0-php.zip

Developing effective clinical research tools requires exceptional coordination between a variety of stakeholders: investigators, software developers, testers, user interface designers, IRBs, information technology support departments and research administrators. User-centered design methodologies have proven to be effective at improving software quality and producing software that is both accessible and highly appropriate to task, but the training and manpower requirements of this approach has made adoption in healthcare research environments difficult. At the same time, development of clinical applications in hospitals and elsewhere traditionally proves expensive - the testing and quality assurance requirements of HIPAA, 21 CFR Part 11 and related regulations require development teams to accept substantial risk and expense.

The Clickframes platform, an open source development process toolkit developed by the Informatics Solutions Group at Children's Hospital Boston, was designed to address all of these issues, allowing research groups and hospital IT shops to develop higher quality, better tested, more reliable software faster, and at lower cost. Clickframes provides a collaborative reqiurements model that supports a user centered design process, allowing principal investigators, developers, testers, designers and other stakeholders to collaboratively develop an interactive specification for an application. Once the specification is complete, the Clickframes tools automate code generation and maintenance across multiple platforms, provide developers with up-to-date and easy to use documentation of additional requirements, and accelerate testing through automated script generation and execution. The result is both a better software design and working code that actually looks like the design.

This talk will introduce the Clickframes toolset, and describe how ISG has used Clickframes in conjunction with a user-centered design process to develop a range of applications, including a grant management system for the Harvard Catalyst CTSC, a novel patient reported outcomes tool for clinical trials, and a major clinical application for Children's Hospital.

MOLGENIS: rapid prototyping of biosoftware at the push of a button

Morris A. Swertz*[,1-6], K Joeri van der Velde[1], Alexandros Kanterakis[1], Juha Muilu[2,7] Tomasz Adamusiak[2,6], Martijn Dijkstra[1,7], Gudmundur A. Thorisson[2,9], George Byelas[1], Danny Arends[1], Anthony J. Brookes[2,9], Ritsert C. Jansen[1] and Helen Parkinson[2,3,6]

[1]Genomics Coordination Center, Groningen Bioinformatics Center, University of Groningen & Dept of Genetics, University Medical Center Groningen, The Netherlands. [2]EU-GEN2PHEN consortium. [3]EU-CASIMIR consortium. [4]BBMRI-NL. [5]NBIC/BioAssist consortium, [6]European Bioinformatics Institute, Hinxton, United Kingdom. [7]Institute for Molecular Medicine Finland (FIMM), Helsinki, Finland. [8]Center for Medical Biomics, University of Groningen, Groningen, The Netherlands. [9]Department of Genetics, University of Leicester, United Kingdom. Contact: Morris Swertz (m.a.swertz@rug.nl)

Project website: http://www.molgenis.org
Code: http://www.molgenis.org/svn/molgenis/3.3/
License: LGPLv3

Abstract

MOLGENIS provides bioinformaticians with a simple model to automatically generate flexible web platforms for all possible genomic, molecular and phenotypic experiments. Each generated MOLGENIS includes biologist friendly user interfaces, semantic interfaces to RDF, SPARQL and ontologies, processing interfaces to R and SOAP/REST web services, a text-file exchange format and full documentation. Galaxy compatible model extensions are currently being developed to integrate cloud computing into this mix.

Introduction

An increasing array of biotechnologies is producing unprecedented amounts of *omics data. There is a huge demand on bioinformaticians to provide their biologists with user friendly, scalable software infrastructures to capture, exchange, and exploit all these new data and provide semantic and programmatic interfaces to connect analysis tools [1]. While standardization is helpful, new research must be quickly accommodated for which efficient software variation mechanisms are needed [2]. We here present MOLGENIS, a model driven software toolkit to efficiently produce the software needed.

Methods

MOLGENIS uses templates to automatically convert a compact XML model into running software. Writing 500 lines of MOLGENIS model replaces 15.000 lines of code in Java, D2RQ-N3 [3], SQL and R. Existing databases can be quickly enriched with a MOLGENIS front-end using the 'ExtractModel' procedure. The standard generated platform of user interfaces, R interfaces, semantic interfaces, text file parsers and writers, and REST/SOAP web services can be extended via plug-ins to integrate research specific processing protocols. Obviously, the generator can be re-run often to accommodate new research (or when a new generator is added which features you want to add; like R and semantic web last year).

Results

We evaluated the MOLGENIS toolbox for many types of biomedical experiments ranging from sequencing to proteomics building on various community consultations [3], including:

XGAP: an eXtensible Genotype And Phenotype platform [4] for systems genetics (GWAS, GWL) on transcript, metabolic and protein data. See http://www.xgap.org

MAGETAB-OM: a microarray experiment data platform based on the MAGE-TAB data format standard. See http://magetab-om.sourceforge.net/

Pheno-OM: to integrate any phenotype data from locus specific annotations to rich cohort reports with the help of the OntoCAT ontology toolkit[4] . See http://wwwdev.ebi.ac.uk/microarray-srv/pheno/

FINDIS: a mutation database for monogenic diseases belonging to the Finnish disease heritage. See http://www.findis.org/molgenis_findis/

Conclusion

MOLGENIS enables rapid prototyping of biological software and eases sharing of data models and software components. New generators are added frequently; for example a Galaxy model parser to generate GridGain based cloud computing jobs that work MOLGENIS data entities. All this jazz greatly helps to reuse the best design patterns to timely compose intelligent software systems that "biologists want to have".

References

1. Thorisson GA, Muilu J, Brookes AJ (2009) Genotype-phenotype databases: challenges and solutions for the post-genomic era. **Nature Reviews Genetics.** 10(1):9-18.
2. Swertz MA, Jansen RC (2007) Beyond standardization: dynamic software infrastructures for systems genetics. **Nature Reviews Genetics**. 8(3).
3. Juha Muilu: schemalet website. http://www.schemalet.org
4. Swertz *et al* (2010) XGAP: a uniform and extensible data model and software platform for genotype and phenotype experiments. **Genome Biology**. 9;11(3):R2
5. Common Api for ontology Tasks.

Towards a federated microarray gene expression repository using MOLGENIS and MAGE-TAB

Alexandros Kanterakis[1,2], Tomasz Adamusiak[3,4], Juha Muilu[4,5], Helen Parkinson[3,4], Despoina Antonakaki[1]
Morris A. Swertz[1-4,6]

[1.] Genomics Coordination Center, Department of genetics, University Medical Center Groningen & Groningen Bioinformatics Center, Groningen Biomolecular Sciences and Biotechnology Institute, University of Groningen.[2.]BBMRI-NL consortium, http://www.bbmri.nl.
[3.]European Bioinformatics Institute, EMBL-EBI, Hinxton, UK[4.]EU-GEN2PHEN consortium, http://www.gen2phen.org[5.]Institute for Molecular Medicine Finland, University of Helsinki, Finland.[6.]NBIC BioAssist biobanking taskforce.
Contact: Alexandros Kanterakis: a.kanterakis@medgen.umcg.nl

Project website: http://magetab-om.sourceforge.net/
Code: http://sourceforge.net/projects/magetab-om/develop
License: LGPLv3

Abstract

The national consortium for Netherlands biobank research infrastructure BBMRI-NL is embarking on microarray meta-analyses aiming to exploit the wealth of microarray and GWAS data currently fragmented between individual biobanks (>6500 samples). To this end there was a need for an easy to populate, customize and federate infrastructure to submit, host and share data, annotations and tools between local and central installations. Here we report the first results.

Introduction

Primary objective is to (a) establish a web-based national repository for microarray gene expression data and (b) to populate it with well-annotated microarray experiment data from participating biobanks. Secondary objective is to (c) share the software as 'microarray database in-a-box' such that all BBMRI biobanks can reuse it locally and (d) can easily share/federate data and tools between local and central installations.

BBMRI-NL and GEN2PHEN

The envisioned system should include suitable user interfaces for researchers, programmatic interfaces for analysis protocols and data federation, and should be easily extended to accommodate diverging local needs. None of the available (open source) systems seemed to provide this and meanwhile GEN2PHEN [2] started 'database-in-a-box' projects including a microarray system based on the MAGE-TAB file format [3] and the MOLGENIS [4,5] biosoftware platform. BBMRI-NL chose to sponsor this project with the following results:

MAGE-TAB object model

We created an object model that matches the MAGE-TAB v1.1 [3] format which is adopted by many major institutes to share microarray investigations, samples, protocols, and data. As a tab-delimited, spreadsheet based format, it is easy to create targeting also BBMRI facilities that lack specialized bioinformatics support.

MOLGENIS implementation

We encoded the MAGE-TAB model in 850 lines of MOLGENIS [3] XML and auto-generated 60K lines of software code. The result is a MAGE-TAB data management suite including web user interfaces, programmatic interfaces to R, JAVA, SOAP, REST and semantic interfaces to RDF and SPARQL.

Data submission and sharing

We created data parsers to enable MAGET-TAB files submission and sharing, building on MOLGENIS import/export mechanisms which resolved much of the complicated foreign key dependency and performance issues when importing related data items into a database.

Future work

The next phase in this project is to populate the system with data, deploy local installs, plug-in analysis tools, and research how MOLGENIS' interfaces can be used in practice to enable Web 2.0 levels of data and tool sharing within the privacy sensitive biobanking community.

References

1. BBRMI-NL is a branch of Biobanking and Biomolecular Resources Research Infrastructure (BBMRI-EU).
2. GEN2PHEN aims to unify human databases towards holistic views into Genotype-To-Phenotype data.
3. Tim F. Rayner et al. (2006) A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB. BMC Bioinformatics, 7, 489
4. Swertz MA, Jansen RC (2007) Beyond standardization: dynamic software infrastructures for systems genetics. Nature Reviews Genetics 8(3).
5. MOLGENIS software: http://www.molgenis.org

Long-term availability of bioinformatics web services

Sebastian J. Schultheiss[1,*], Marc-Christian Münch[2], Gergana D. Andreeva[2], Gunnar Rätsch[1]

[1] Friedrich Miescher Laboratory of the Max Planck Society, Tübingen, Germany
[2] Wilhelm Schickard Institute for Computer Science, University of Tübingen, Germany
* E-mail address: sebastian.schultheiss@tuebingen.mpg.de or sebi@umich.edu

URL to the original data: http://tinyurl.com/fml-webservices or
http://www.fml.tuebingen.mpg.de/raetsch/suppl/long-term-availability-of-bioinformatics-web

We studied all 927 open-access web services found in the seven NAR Web Server Issues published between 2003 and 2009. We checked their availability, usability and functionality. A web service is defined here as an application usable through a web browser.

An astonishing number of 90% of services are still reachable. However, while the original web page may still be online, up to 62% of services no longer function as published when tested with their example data. Surprisingly, the services published in 2003 clearly stand out from the following years in terms of availability, as this issue contains many of the most accessed and cited services in computational biology developed before 2003. The requirements for publication of services in NAR have risen constantly, with services this year having to provide the following information directly on their web pages: contact information, example data, help texts, and version information. The presence of this information, and service usability, has risen constantly from year to year. Based on these qualities, we created the Long-Term-Score, which reflects the number of criteria fulfilled by a service, thus measuring service quality.

We counted the number of times a service's NAR publication was cited. We can show that bad design choices correlate with a low number of citations: the ratio of citations for services with high usability vs. services with low usability shows that better services are cited 2.2 times more often.

Given these findings, we can provide reviewers, editors, and most notably web service developers with guidelines that make the use of their service easier for everyone and simultaneously allow for time-saving maintenance of the service for the years after publication, hopefully leading to a higher number of citations. The most effective way of ensuring continued access to a service is a persistent web address, offered either by the publishing journal or instituted on the authors' own initiative.

GBrowse 2.0

Scott Cain[1], Ian Davies[2], Ben Faga[3], Sheldon McKay[4], Lincoln Stein[1,5]

[1]Ontario Institute for Cancer Research, Toronto, ON, Canada
[2]University of Waterloo, Waterloo, ON, Canada
[3]Center for Bioinformatics and Computational Biology, The University of Iowa, Iowa City, Iowa
[4]Cold Spring Harbor Laboratory, Cold Spring Harbor, NY
[5]Presenting author email address: lincoln.stein@gmail.com

Project URL: http://gmod.org/
Code URL: http://gmod.svn.sourceforge.net/viewvc/gmod/Generic-Genome-Browser/
License: Artistic License 2.0

The Generic Genome Browser (GBrowse) is an free, open source, web-based browser for displaying and navigating genome features. It is part of the Generic Model Organism Database (GMOD) project which aims to provide reusable components for working with genomic data, and is in use by hundreds of organizations around the world. Here we present GBrowse version 2.0 (GBrowse2), with numerous improvements to both the user interface and the back end architecture. Among the improvements are an AJAX interface that loads images without reloading the page, support for multiple database and rendering servers, support for SAM/BAM and BigWig formated data, and support for individual user accounts. GBrowse2 is available from CPAN or from the GMOD website, http://gmod.org/.

Title: Cytoscape Web: An interactive, customizable web-based network browser

Authors: Christian Lopes, Max Franz, Quaid Morris, Gary D. Bader

Author Affiliations: Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, Canada
gary.bader@utoronto.ca

URL for overall project website: http://cytoscapeweb.cytoscape.org
URL for accessing the code: http://cytoscapeweb.cytoscape.org/download

Open Source License being used: GNU Lesser General Public License

Cytoscape Web is an interactive, web-based, open source network visualization tool, which is freely available at http://cytoscapeweb.cytoscape.org. With some basic programming skills, Cytoscape Web can be customized and incorporated into any website. Because it is a pure client-side component, developers can choose any server-side technology, if necessary. The main network display is implemented in Flex/ActionScript, but a JavaScript API is also provided, so the website can rely on web standards (HTML, CSS, JavaScript) for embedding and interacting with Cytoscape Web.

Like Cytoscape, Cytoscape Web allows the client application to define custom node and edge attributes before loading the network data or even after it is rendered. Node and edge visual styles (e.g. color, size, opacity) can be dynamically changed by: (a) specifying default visual properties for all elements; (b) mapping node and edge attributes (name, interaction type, weight, etc.) to visual styles; (c) overriding default or mapped styles by setting a bypass style. These three options provide flexibility and allow each Cytoscape Web based application to have its own semantics, styles and features. For example, iRefWeb (http://wodaklab.org/iRefWeb/), an interface to the relational database interaction Reference Index (iRefIndex), uses a basic implementation of Cytoscape Web to display all interactions in which a single queried gene participates. Alternatively, GeneMANIA (http://www.genemania.org), a gene function prediction webserver, uses a more advanced implementation of Cytoscape Web to extend a users' input gene list and illustrate interactions among the genes. Cytoscape Web communicates with GeneMANIA to display gene or network specific highlights and associated information in real time.

A Cytoscape Web tutorial, with ready-to-use samples, and the API documentation can be accessed at the Cytoscape Web website. Developers will also find a demo application showcasing more advanced features that can be built around Cytoscape Web (http://cytoscapeweb.cytoscape.org/demo) and these can be freely used as a template for building similar web sites.

Pathway Projector: Web-Based Zoomable Pathway Browser Using KEGG Atlas and Google Maps API

Nobuaki Kono, Kazuharu Arakawa, Ryu Ogawa, Nobuhiro Kido, Kazuki Oshita, Keita Ikegami, Satoshi Tamaki, Masaru Tomita
Institute for Advanced Biosciences, Keio University, Endo 5322, Fujisawa, Kanagawa 252-8520, Japan
ciconia@sfc.keio.ac.jp

Biochemical pathways provide an essential context for understanding comprehensive experimental data and the systematic workings of a cell.Therefore, the availability of online pathway browsers will facilitate post-genomic research, just as genome browsers have contributed to genomics. Many pathway maps have been provided online as part of public pathway databases. Most of these maps, however, function as the gateway interface to a specific database, and the comprehensiveness of their represented entities, data mapping capabilities, and user interfaces are not always sufficient for generic usage. To this end, here we present a web-based pathway browser named Pathway Projector. Pathway Projector provides integrated pathway maps that are based upon the KEGG Atlas, with the addition of nodes for genes and enzymes, and is implemented as a scalable, zoomable map utilizing the Google Maps API. Users can search pathway-related data using keywords, molecular weights, nucleotide sequences, and amino acid sequences, or as possible routes between compounds. In addition, experimental data from transcriptomic, proteomic, and metabolomic analyses can be readily mapped.

Pathway Projector's pathway maps were based on the KEGG Atlas map, for the familiarity of its layout and for the availability of various analysis tools. However, because the KEGG Atlas only represents metabolite nodes, we added all gene and enzyme nodes semi-automatically on the reference pathway map. As a result, our pathway map contains 1572 metabolite nodes and 1813 enzyme nodes. In the organism specific pathway maps, the number of gene nodes are 1365 in *Escherichia coli*, and 2883 in human, for example. The software was implemented using AJAX (Asynchronous JavaScript + XML) programming paradigm and the main interface framework was built with Ext JS 2.0 library. For the representation of the global pathway map, we adopted zoomable user interface (ZUI) using Google Maps API through G-language Genome Analysis Environment. Detailed information of each node in pathway map is shown in an information window upon clicking. These windows include the common name, identifier, structural formula or chemical equation, and links to external databases. Organism specific pathway maps are currently available for 843 species.

Pathway Projector has four types of search functionalities, including those by keywords and identifiers, by molecular mass, by possible routes between two metabolites using PathComp, and by sequence similarity using BLAST. The results are listed in a search result panel and are also visually highlighted by red markers onto the respective components on the pathway map. The pathway mapping tool can change parameters for size, color, and labels of edges and nodes, and subsequently creates an overlay image. When values for time-series or multiple conditions are specified for nodes, graphs generated by the Google Chart API are displayed on the nodes. Quikmaps was utilized to implement manual annotation and editing capabilities.

The understanding of omics layers is important for systems biology, and biochemical pathways supply a necessity context for this purpose. Since pathways do not exist independently, but are rather interconnected *in vivo*, the observation of an integrated map is desirable, especially for the mapping of comprehensive experimental data. Pathway Projector has an intuitive interface by utilizing Google Maps and Ext JS and KEGG pathway maps. Moreover, capabilities of this software such as searching, editing, annotation, mapping and links to various databases, will be a useful gateway for pathway analysis.

Web server of Pathway Projector is freely available for academic users at http://www.g-language.org/PathwayProjector/.

Reference:

1. Kono N, Arakawa K, Ogawa R, Kido N, Oshita K, Ikegami K, Tamaki S, Tomita M, "Pathway Projector: Web-based Zoomable Pathway Browser using KEGG Atlas and Google Maps API", PLoS One, 2009, 4(11):e7710.

URL(project): http://www.g-language.org/PathwayProjector/
URL(code): http://www.g-language.org/PathwayProjector/install.html
License: GNU General Public License (source code), KEGG License (data)

Evoker: a visualization tool for genotype intensity data

James A. Morris [1], Joshua C. Randall [2],
Julian B. Maller [2] and Jeffrey C. Barrett [1],
[1]Wellcome Trust Sanger Institute,
Hinxton, Cambridge, CB10 1HH, UK
[2]Wellcome Trust Centre for Human Genetics,
University of Oxford, Roosevelt Drive, Oxford, OX3 7BN, UK

Email: jm20@sanger.ac.uk
Web site: `http://www.sanger.ac.uk/resources/software/evoker`
Source code URL: `http://sourceforge.net/projects/evoker/develop`
License: MIT

April 13, 2010

Genome-wide association studies (GWAS) are now widely used in complex human disease genetics, these approaches produce huge volumes of data, which has created a real need for user friendly tools for data quality control and analysis of GWAS datasets. One critical aspect of quality control in a GWAS is evaluating genotype cluster plots to verify sensible genotype calling in putatively associated SNPs. The normalized intensity files from which genotype cluster plots are generated are extremely large and unwieldy in the default formats from SNP chip providers (uncompressed text-format intensities from a GWAS of 10,000 individuals would be hundreds of gigabytes). Extracting subsets of data and plotting hundreds of SNPs of interest is typically a tedious procedure requiring some computational sophistication. Therefore, we have developed Evoker, a Java program which supports two simple and compact binary data formats and is designed to make genotype cluster plot inspection a highly efficient process.

Important features of the Evoker program include remote connection to data sources, which means users do not need to have local copies of the large genotype and intensity files saving greatly on local disk space. The program only needs to transfer the data for the SNPs of interest, so Evoker is able to remain responsive even when dealing with very large datasets. Users are able to load lists of SNPs of interest, such as those showing evidence of association. The user can then quickly view, asses and make a decision on the quality of the cluster plot for each SNP, with the decision recorded in a separate file. Evoker can also be used to visualize the impact that excluding samples (such as those with borderline QC results) has on structure of the clusters.

The main Evoker program has been written in Java and so will work on any platform with Java 1.5 or later installed. The Evoker software also includes easy to use scripts for the generation of binary format files.

Evoker is an easy to use tool for visualizing genotype cluster plots, and provides a solution to the computational and storage problems related to working with such large datasets.

Fiji Is Just ImageJ – an Open Source platform for biological image analysis

Pavel Tomancak[1], Stephan Saalfeld[1], Johannes Schindelin[1], Albert Cardona[2]
[1]Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG)
PfotenhauerStrasse 108, D-01307 Dresden, Germany
[2]Institute of Neuroinformatics, Uni/ETH Zurich Winterthurerstrasse 190 Bau 55, Zurich 8057
E-Mail: tomancak@mpi-cbg.de,
Web: http://pacific.mpi-cbg.de; License: GPL; Source code: http://pacific.mpi-cbg.de/cgi-bin/gitweb.cgi

KEY WORDS: Image Processing, Open Source development, biological image analysis

Substantial portions of primary data in biology are in the form of images. The volume of raw image data has increased dramatically in recent years with wider adoption of high-throughput and high-content imaging technologies. There is an urgent need for extracting quantitative information from these massive datasets in order to address important biological questions in particular from the systems biology point of view.

We contribute towards addressing this need by developing **Fiji** (Figure 1), an Open Source distribution of the popular biological image analysis software ImageJ written in Java. Fiji has recently gained substantial international recognition documented by almost 19,000 downloads from over 10,000 unique IP adresses over the past year and sustained increase in traffic on the Fiji Wiki reaching a record 37,000 hits in March [1]. The popularity of the platform comes from the fact that Fiji is developed according to modern software engineering practices, is extensively documented through active Wiki pages [2,3], offers a broad range of **scripting languages** (Python, Ruby, Javascript, BeanShell and Clojure) for algorithmic prototyping, provides transparent system for automatic updates and most importantly, because an active interdisciplinary community of developers has formed around Fiji who use the platform to solve real biological problems. Fiji developers come from all over the world yet they meet regularly at coding sprees called '**hackathons**' that dramatically speed up the development of the platform. The innovative Open Source development strategies of the Fiji community have been recognized by inclusion in the prestigious Google Summer of Code in 2009 [4].

The power of Fiji is highlighted by the **Fiji projects** that include rigid and elastic registration of large light and electron microscopy acquisitions, hardware accelerated 3d visualization, segmentation, neurite tracing, feature extraction and many more. The common denominators of Fiji projects are close connections to ongoing biological research and extensive sharing of algorithms and code through common software libraries. One example of such a shared code base is the dimension-, storage- and data type-independent image processing library that enables seamless manipulation of massive microscopy datasets in Java. We will describe the benefits of Fiji for users as well as developers interested in solving biologically motivated image analysis problems and hopefully attract more talent from the imaging community to this emerging Open Source platform.

Figure 1: Fiji logo

[1] Fiji Wiki statistics http://pacific.mpi-cbg.de/awesome/index.html#Unique visitors in each month

[2] Fiji Wiki pages : http://pacific.mpi-cbg.de/

[3] Fiji YouTube channel : http://www.youtube.com/user/fijichannel/

[4] GSoC Fiji http://socghop.appspot.com/gsoc/org/show/google/gsoc2009/fiji/

IPRStats: visualization and analysis of InterProScan Results

David E. Vincent[1] and Iddo Friedberg*[1,2]
1. Department of Computer Science and Software Engineering
2. Department of Microbiology
Miami University, Oxford OH, USA
*Corresponding author: i.friedberg@muohio.edu

URL: http://github.com/idoerg/IPRStats

Introduction: InterProScan is a popular tool used in the functional analysis of protein sequences; it is a powerful tool for identifying protein families, predicting protein function, as well as other features included in InterPro member databases. While the information generated by InterProScan is extremely useful it can be difficult to manage and interpret because of the vast amount of data it produces when analyzing genomic and metagenomic data.

We present IPRStats, a web server and standalone program that accepts the output of InterProScan and provides chart and tabular summaries of the data. These can be downloaded for further use and data analysis pipelining. The user uploads the InterProScan XML output to the server. This output gets incorporated into a MySQL database which is then queried by a series of scripts producing the output. Additional features planned are the correlation of sequence signatures with the abiotic conditions of the habitats from which they were produced. Graphs are produced using Google graphs.

Conclusions: we provide a useful web tool which will allows users to upload data generated by InterProScan and receive a graphical display summarizing the results. A page of information and charts is generated for each database which was included in the InterProScan search. These pages allow the users to see features such as the most common protein families and sequence signatures found in PFAM, TIGRFAM, PANTHER and common structures reported by Gene3D. Each page also provides links to more information about the data from sites such as GO and EBI.