10th Annual Bioinformatics Open Source Conference

BOSC 2009

1. Welcome

Welcome to BOSC 2009! This is the 10th annual Bioinformatics Open Source Conference held as a Special Interest Group (SIG) meeting in conjunction with the Intelligent Systems for Molecular Biology Conference.

BOSC is sponsored by the Open Bioinformatics Foundation (O|B|F), a non-profit group dedicated to promoting the practice and philosophy of Open Source software development within the biological research community. Many open source bioinformatics packages are widely used by the research community across many application areas and form a cornerstone in enabling research in the genomic and post-genomic era. Open source bioinformatics software has facilitated rapid innovation, dissemination, and wide adoption of new computational methods, reusable software components, and standards.

To celebrate the special occasion of BOSC's 10th Anniversary, the theme of this year's conference is "Looking Back and Looking Ahead: Open Source Solutions to Grand Challenges in Bioinformatics." Session topics under this umbrella include multicore and gpgpu computing, computational grids, visualization, and regulatory genomics. We also have a special shared session with the Data and Analysis Management SIG and a panel discussion on "The Applicability of Design Patterns for the Bioinformatics Open Source Community". In keeping with these sessions, we welcome our keynote speakers. Alan Ruttenberg, Principal Scientist at Science Commons, who will speak on the application of semantic web technologies to bioinformatics. Robert S. Hanmer of Alcatel-Lucent and President of the Hillside group will speak about "Software Patterns for Reusable Design". We will also hear updates about the main Open Source Bioinformatics Software suites.

One of the hallmarks of BOSC is the coming together of the open source developer community in one location to meet face-to-face. This creates synergy where participants can work together to create use cases, prototype working code, or run bootcamps for developers from other projects as short, informal, and hands-on tutorials in new software packages and emerging technologies. In short, BOSC is not just a conference for presentations of completed work, but is a dynamic meeting where collaborative work gets done and attendees can learn about new or on-going developments that they can directly apply to their own work. We have two formats to facilitate these collaborations: lightning talks (short, 5 minute talks for focused ideas and demos) and Birds of a Feather sessions where small groups gather around a common interest.

BOSC is made possible by a community effort. We thank the abstract reviewers and session chairs. We invite you to join the O|B|F (application forms are available) and the BOSC 2010 planning committee. Finally, thanks to support from an anonymous donor we will again be giving an award for the best student presentation.

Organizing Committee
Kam Dahlquist (Chair), Lonnie Welch (Co-chair), Hilmar Lapp, Jens Lichtenberg, Frank Drews, Andrew Dalke, Jim Procter, Anton Nekrutenko, Steffen Moeller

2. Schedule (Day 1)

9:00-9:15	Kam Dahlquist	[Welcome]
9:15-10:15	Alan Ruttenberg	[Keynote] Semantic Web Technologies
10:15-10:45	**Coffee Break**	
10:45-11:05	Jean-Stéphane Varré	[Multicore & GPGPU Computing] Biomanycores, a repository of interoperable open-source code for many-cores bioinformatics
11:05-11:25	Josh Buckner	[Multicore & GPGPU Computing] Enabling GPU Computing in the R Statistical Environment
11:25-11:40	Mikhail Fursov	[Multicore & GPGPU Computing] UGENE – A practical approach for complex computational analysis in molecular biology
11:40-11:48	Oswaldo Trelles	[Multicore & GPGPU Computing Lightning Talk] Qnorm: A library of parallel methods for gene-expression Q-normalization
11:48-12:08	Hajo N. Krabbenhöft	[Computational Grids] Taverna workflows across Grids, web services and the command line
12:08-12:23	Ann-Kristin Grimm	[Computational Grids] Grid-based expression QTL analysis
12:23-12:31	Joel Hedlund	[Computational Grids Lightning Talk] The Nordic BioGrid project -- Bioinformatics for the grid
12:31-2:00	**Lunch**	
2:00-2:20	Allan Kuchinsky	[Visualization] Cytoscape Springs Forward: Re-architecture for Version 3.0
2:20-2:33	Frederik Decouttere	[Visualization] Bioinformatics simplified with seqpad
2:33-2:46	Bernat Gel	[Visualization] DASGenExp: an interactive web-based DAS client with client-side rendering
2:46-2:59	Kazuharu Arakawa	[OS Software] Web Service Interface for G-language Genome Analysis Environment
2:59-3:12	Sylvain Brohée	[OS Software] Best of both worlds : combining the user-friendliness of Wikis and the rigor of biological databases
3:12-3:30	Bartek Wilczynski	[Regulatory Genomics] BNfinder: free software for effective Bayesian Network inference
3:30-4:00	**Coffee Break**	
4:00-4:20	Morris Swertz	[Data & Analysis Management] MOLGENIS by example: generating an extensible platform for genotype and phenotype experiments
4:20-4:40	Robert Murphy	[Data & Analysis Management] PSLID, the Protein Subcellular Location Image Database: Subcellular location assignments, annotated image collections, image analysis tools, and generative models of protein distributions
4:40-5:00	Mark Welsh	[Data & Analysis Management] BioHDF: Open binary file formats for large-scale data management – Project Update
5:00-5:15	Brad Chapman	[Data & Analysis Management] Lowering barriers to publishing biological data on the web
5:15-5:30	Kam Dahlquist	[Data & Analysis Management] XMLPipeDB: A Reusable, Open Source Tool Chain for Building Relational Databases from XML Sources
5:30-6:00	Lightning Talks and Birds of a Feather	
7:00	O|B|F Board Meeting and No-host Dinner (location TBA)	

3. Schedule (Day 2)

9:00-9:15	Kam Dahlquist	[Announcements]
9:15-10:15	Robert Hanmer	[Keynote] Software Patterns for Reusable Design
10:15-10:45	**Coffee Break**	
10:45-11:00	Peter Rice	[Bio* Update] EMBOSS: European Molecular Biology Open Software Suite
11:00-11:15	Peter Cock	[Bio* Update] Biopython Project Update
11:15-11:30	Andreas Prlic	[Bio* Update] BioJava 2009: an Open-Source Framework for Bioinformatics
11:30-11:45	Jim Procter	[Bio* Update] Application of VAMSAS enabled tools for the investigation of protein evolution
11:45-12:00	Martin Senger	[Bio* Update] Soaplab: Open Source Web Services Framework for Bioinformatics Programs
12:00-12:20	Pjotr Prins	[Bio* Update] BioLib: Sharing high performance code between BioPerl, BioPython, BioRuby, R/Bioconductor and BioJAVA
12:20-12:28	Steffen Möller	[OS Software Lightning Talk] Debian adopts and disseminates Bioinformatics Open Source Software
12:30-2:00	**Lunch**	
2:00-2:20	Quinn Snell	[OS Software] PSODA: Open Source Phylogenetic Search and DNA Analysis
2:20-2:40	Finn Drablos	[Regulatory Genomics] Computational discovery of composite motifs in DNA
2:40-2:55	François Fauteux	[Regulatory Genomics] SEEDER: PERL MODULES FOR CIS-REGULATORY MOTIF DISCOVERY
2:55-3:10	Matias Piipari	[Regulatory Genomics] Large-scale gene regulatory motif discovery and categorisation with NestedMICA
3:10-3:30	Sophie Schbath	[Regulatory Genomics] R'MES:
3:30-4:00	**Coffee Break**	
4:00-4:15	Lonnie Welch	[Regulatory Genomics] Open Source Implementation of Batch-Extraction for Coding and Non-coding Sequences/An Open Source Framework for Bioinformatics Word Enumeration and Scoring
4:15-4:50	Robert Hanmer (moderator), Lonnie Welch, Aleksi Kallio, other panelists TBA	[Panel Discussion] On the Applicability of Design Patterns for the Bioinformatics Open Source Community
4:50-5:30	Lightning Talks and Birds of a Feather	

4. Keynote Speakers

Alan Ruttenberg

Alan Ruttenberg is a Principal Scientist at. He works with Semantic Web technologies in computational biology, with an emphasis on the creation and application of structured biological knowledge to interpret experimental results. He is currently involved in a number of open biomedical ontology efforts, including: the Ontology for Biomedical Investigations (OBI), the Basic Formal Ontology (BFO) that will form the upper level ontology for the OBO foundry, the Infectious Disease Ontology (IDO), the Program on Ontologies of Neural Structures (PONS), the Information Artifact Ontology (IAO), and BioPAX-OBO for representing molecular and cellular pathways. These interests and efforts come together in my project at Science Commons - the Neurocommons, a large scale Semantic Web knowledge base of biological information aimed at supporting, initially, the neurosciences. He is also an active participant in W3C Semantic Web activities. In 2006 and 2007 he was a member of the Health Care Life Sciences Interest Group, and early work on the Neurocommons became the core of the prototype life sciences knowledge base that the group has documented. He is a chair of the OWL Working Group specifying OWL 2, and a coordinating editor of the OBO Foundry. His graduate work was at the MIT Media Lab in the Music and Cognition Group, and he has an undergraduate degree in Physics and Mathematics from Brandeis University.

Robert S. Hanmer

Robert S. Hanmer is a Consulting Member of Technical Staff in the Technical Component Management area in Alcatel-Lucent’s Operations area. He is based in Naperville, Illinois, USA. Current responsibilities include developing software-sourcing strategies for middleware and open source software. Previous positions within Lucent and Bell Laboratories have included development, architecture and evaluation of highly reliable systems focusing especially on the areas of reliability and performance. He is active in the software patterns community, including serving as program chair at several pattern conferences. He has authored or co-authored 14 journal articles, several book chapters and the book Patterns for Fault Tolerant Software. He is a member of the IEEE Computer Society, a Senior Member of the ACM and current President of The Hillside Group, the organization that sponsors the PLoP conference. He received his B.S. and M.S. degrees in Computer Science from Northwestern University in Evanston, Illinois.

5. Abstracts

Abstracts appear in the following pages in the order that they are presented at the conference.

Biomanycores, a repository of interoperable open-source code for many-cores bioinformatics

Jean-Stéphane Varré Stéphane Janot Mathieu Giraud

LIFL - UMR CNRS 8022, INRIA, Université Lille 1

jean-stephane.varre@lifl.fr, stephane.janot@lifl.fr, mathieu.giraud@lifl.fr

URL: `http://www.biomanycores.org`
Licences: Various open-source licences

Graphic processing units (GPUs) are a first step towards massively many-cores architectures, and recent trends blur the line between such GPUs and multicore processors. Those architectures enable efficient parallel processing at a very low cost. The current GPUs offer coarse-grained level parallelism (*work-groups* or *blocks* of independent computations) as well as SIMD-like parallelism (*work-items* or *threads*).

GPUs have been used in bioinformatics since 2005, at the beginning tweaking graphics primitives [3, 8]. The CUDA libraries, first released in 2007 [2], have deeply simplified the development on GPUs. In the two past years a growing number of applications have been proposed [13, 10, 7, 14, 5, 12, 9]. The new OpenCL standard [1] should increase this research field and improve the portability of many-cores applications. However, lots of those references are "proof-of-concept" papers, and do not get actually used.

We present Biomanycores, a collection of many-cores bioinformatics tools, designed to bridge the gap between researches in high-performance-computing and usual bioinformaticians and biologists. The goal is both to gather many-cores programs and to propose interfaces to Bio* projects. The language of choice should be OpenCL, but, while no public implementation of OpenCL is available, CUDA projects are included.

The project is still in an early stage of development, but already includes 3 different applications: Smith-Waterman [10], pKnotsRG [11] and Position-Weight-Matrix scan [5], with interfaces to Biojava 1.6 [6], Bioperl 1.52 [15], and Biopython 1.50b [4]. We wish to open as much as possible Biomanycores to other high-performance bioinformatics applications and to better integrate to Bio* projects.

References

[1] The Khronos Group, OpenCL 1.0 specification, 2008.

[2] Nvidia CUDA programming guide 2.0, 2008.

[3] M. Charalambous, P. Trancoso, and A. Stamatakis. Initial experiences porting a bioinformatics application to a graphics processor. *Adv. in Informatics*, pages 415–425, 2005.

[4] P. J. A. Cock, T. Antao, J. T. Chang, and al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, page btp163, 2009.

[5] M. Giraud and J.-S. Varré. Parallel position weight matrices algorithms. In *International Symposium on Parallel and Distributed Computing (ISPDC 2009)*, 2009.

[6] R. C. G. Holland, T. A. Down, M. Pocock, and al. BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097, 2008.

[7] L. Ligowski and W. Rudnicki. An efficient implementation of Smith-Waterman algorithm on GPU using CUDA, for massively parallel scanning of sequence databases. In *HiCOMB 2009*, 2009.

[8] W. Liu, B. Schmidt, G. Voss, and W. Müller-Wittig. GPU-ClustalW: using graphics hardware to accelerate multiple sequence alignment. In *High Performance Computing (HiPC 2006), LNCS 4297*, pages 363–374, 2006.

[9] Y. Liu, B. Schmidt, and D. Maskell. Parallel reconstruction of neighbor-joining trees for large multiple sequence alignments using CUDA. In *HiCOMB 2009*, 2009.

[10] S. A. Manavski and G. Valle. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. *BMC Bioinformatics*, 9 Suppl 2:S10, 2008.

[11] J. Reeder, P. Steffen, and R. Giegerich. pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows. *Nucl. Acids Res.*, 35(S2):W320–324, 2007.

[12] G. Rizk and D. Lavenier. GPU accelerated RNA folding algorithm. In *Using Emerging Parallel Architectures for Computational Science (ICCS 2009)*, 2009.

[13] M. C. Schatz, C. Trapnell, A. L. Delcher, and A. Varshney. High-throughput sequence alignment using graphics processing units. *BMC Bioinformatics*, 8:474, 2007.

[14] H. Shi, B. Schmidt, W. Liu, and W. Mueller-Wittig. Accelerating error correction in high-throughput short-read DNA sequencing data with CUDA. In *HiCOMB 2009*, 2009.

[15] J. E. Stajich, D. Block, K. Boulez, and al. The Bioperl toolkit: Perl modules for the life sciences. *Genome Research*, 12(10):1611–1618, 2002.

Enabling GPU Computing in the R Statistical Environment

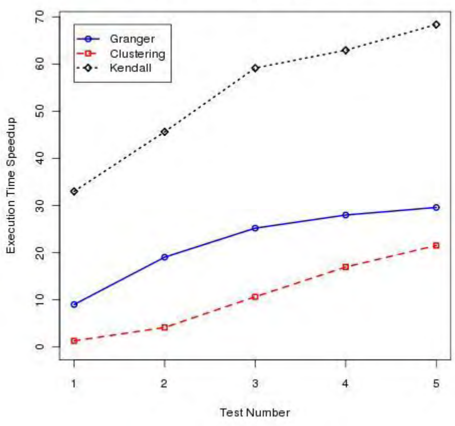
Josh Buckner[1], Manhong Dai[1], Brian Athey[1,2], Stanley Watson[1] and Fan Meng[1,2]
[1]Molecular & Behavioral Neuroscience Institute and Psychiatry Department
[2]National Center for Integrative Biomedical Informatics
University of Michigan, Ann Arbor, MI 48109, US
bucknerj@umich.edu

Project URL: http://brainarray.mbni.med.umich.edu/Brainarray/Rgpgpu/
Code URL: http://brainarray.mbni.med.umich.edu/Brainarray/Rgpgpu/gputools_0.1-0.tar.gz
License: GPLv3

R is the most popular open source statistical environment in the biomedical research community. However, most of the popular R function implementations involve no parallelism and they can only be executed as separate instances on multicore or cluster hardware for large data-parallel analysis tasks. The arrival of modern graphic processing units (GPUs) with user friendly programming tools, such as nVidia's CUDA toolkit (http://www.nvidia.com/cuda), provides a possibility of increasing the computational efficiency of many common tasks by more than one order of magnitude (http://gpgpu.org/). However, most R users are not trained to program a GPU, a key obstacle for the widespread adoption of GPUs in biomedical research.

To overcome this obstacle, we decided to devote efforts for moving frequently used R functions in our work to the GPU using CUDA. In the ideal solution, if a CUDA compatible GPU and driver is present on a user's machine, the user may only need to prefix "gpu" to the original function name to take advantage of the GPU implementation of the corresponding R function. We take achieving this ideal as one of our primary goals so that any biomedical researcher can harness the computational power of a GPU using a familiar tool. Since our code is open source, researchers may customize the R interfaces to their particular needs. In addition, because CUDA uses shared libraries and unobtrusive extensions to the C programming language, any experienced C programmer can easily customize the underlying code.

Fig. 1: Speedup (R GPU vs. CPU)



Using the CUDA extension to C and the shared linear algebra library CUBLAS, we have implemented a variety of statistical analysis functions with R interfaces that execute with different degrees of parallelism on a Graphics Processing Unit (GPU). If an algorithm is comprised of common vector or matrix operations each performed once, we involve the GPU by implementing those operations with calls to CUBLAS. If an algorithm involves computing the elements of a large matrix, we can often merely assign each thread executing on the GPU a portion of a row and/or column. Algorithms for which we have implemented GPU enabled versions include the calculations of distances between sets of points (R dist function), hierarchical clustering (R hclust function). Pearson and Kendall correlation coefficients (similar to R cor function), and the Granger test (granger.test in the R MSBVAR package).

Figure 1 provides performance comparison between original R functions assuming a four thread data parallel solution on Intel Core i7 920 and our GPU enabled R functions for a GTX 295 GPU. The speedup test consisted of testing each of three algorithms with five randomly generated data sets. The Granger causality algorithm was tested with a lag of 2 for 200, 400, 600, 800, and 1000 random variables with 10 observations each. Complete hierarchical clustering was tested with 1000, 2000, 4000, 6000, and 8000 points. Calculation of Kendall's correlation coefficient was tested with 20, 30, 40, 50, and 60 random variables with 10000 observations each.

We are committed to implement more R GPU functions, and we hope to contribute packages to the open source community via our project's website. We hope that others can contribute to the R-GPGPU effort and encourage any comments or suggestions.

Acknowledgements: J. Buckner, M. Dai, F. Meng, and S. J. Watson are members of the Pritzker Neuropsychiatric Disorders Research Consortium, which is supported by the Pritzker Neuropsychiatric Disorders Research Fund L.L.C. This work is also partly supported by the National Center for Integrated Biomedical Informatics through NIH grant 1U54DA021519-01A1 to the University of Michigan.

UGENE – A practical approach for complex computational analysis in molecular biology

Mikhail Fursov, Alexey Varlamov
mfursov@unipro.ru, varlax@unipro.ru
Center of Information Technologies "UniPro", Novosibirsk, Russia

UGENE (http://ugene.unipro.ru) is an open-source bioinformatics project. The ultimate goal of the project is to integrate popular bioinformatics tools and algorithms within a single visual interface that can be easily used by molecular biologists.

UGENE version v1.4 code base contains about 20 different plugins, each of them representing one of popular bioinformatics algorithms or methods. These include multiple alignment tools, Smith-Waterman algorithm implementation, HMMER2 tools, repeats and ORF analysis, restriction enzymes markup, search for transcription factor binding sites, integration with web databases like BLAST and CDD. For the complete list of plugins check the project web site: http://ugene.unipro.ru/plugins.html

In the next (1.5) version of UGENE we plan to support Primer3 package, add integrated BLAST tools and two secondary structure prediction algorithms: GOR and PSIPRED.

The key advantage of UGENE is a complete integration of the algorithms within a single visual interface. After installation UGENE does not require any additional software to be installed. From the developer's perspective it’s important that all algorithms are refactored to use universal data model and common threading API.

All of the algorithms in UGENE are tuned to utilize multi-core environment. For example, MUSCLE multiple sequence alignment package embedded into UGENE is able to utilize multi-core environment, while the original is single threaded. HMMER2 package contains special optimizations for IBM Cell Broadband platform, platforms with Altivec and SSE2 instruction sets. With the next version UGENE will also add GPU based version of HMMER2 algorithm.

To ensure that integration of the 3rd party algorithms is correct the project maintains test base. The compatibility test base for UGENE v1.4 contains about 1200 tests. The v1.5 test base will have more than 2000 tests.

The main component of the visual interface of UGENE is W*orkflow Designer.* The Workflow Designer is used for construction of computational diagrams from the predefined set of visual algorithmic blocks, or processes. The key idea of Workflow Designer is to make the process of automation of routine tasks as simple as it’s possible and make it available for non-programmers. The following page contains detailed description of Workflow Designer data model and interface: http://ugene.unipro.ru/plugin_workflow.html

UGENE is written with open-source QT4 C++ multi-platform library and QtScript scripting language. It is available for most of the popular platforms like Linux, Windows, MacOS X. UGENE is also included into Ubuntu and Fedora Linux distributions.

License: GNU General Public License v.2
Software and source code: http://ugene.unipro.ru/download.html

Qnorm: A library of parallel methods for gene-expression Q-normalization

José Manuel Mateos-Duran[1]; Pjotr Prins[2]; Andrés Rodríguez[1] and Oswaldo Trelles[1].
(1) Computer Architecture Dept.; University of Malaga; Campus de Teatinos, 29071; Spain.
(2) Lab. of Nematology, Wageningen University, The Netherlands
{jmateos, andres, ortrelles}@uma.es, Pjotr.Prins@wur.nl

Project web site:	http://www.bitlab-es.com/qnorm
Source code at:	http://github.com/ots/qnormalization
OSS License:	GPL version 3

Abstract for BOSC 2009 Session: Multicore and GPGPU computing

Qnorm is a library for exploring different strategies in parallelization of large scale computations, a generic approach in high performance computing (HPC).

The high amounts of molecular data produced by current high-throughput technologies in modern biology poses challenging problems in our capacity to process and understand data. Not only allows pyro-sequencing the production of overwhelming sets of data but even ultra high density microarrays jumped the previous thirty thousand genes contained, in a simple array, to more than 5 million genetic markers. Nowadays clinical studies include hundreds of thousand of patients instead of the thousands genetically fingerprinted a few years ago, in a typical study. Current sequential implementations of software are unable to deal with such enormous volumes. Here we show the impact of different high performance computing strategies using three different parallel approaches for shared memory, distributed memory and GPU architectures, that can be easily applied to other existing bioinformatics algorithms and show how benchmarking helps decide on strategy.

As proof of concept we chose the quantile based method[1] as it provides a fast and easy to understand procedure to normalize multiple gene-expression datasets, under the assumption of sharing a common distribution. The high computational cost and memory requirements ($p > 6$ millions and $N > 1000$ samples) of sequential Q-normalization in-core calculations are behind our interest in developing an HPC approach to this problem.

For shared and distributed memory architectures, we use a dynamic load distribution over the set of columns that are concurrently sorted and partially row averaged in a first step. A synchronization barrier is needed before global averaging in a second step. A number of indexes are managed to avoid a re-ordering of the experiments with good effects in the processing time. The same two steps approach can be mapped to a GPU solution. Every column is processed in parallel by the GPU and then the global average column is also computed in parallel. Performance results have shown a near perfect speed-up in the supercomputing parallel strategies. As expected, a good GPU (graphics card) can provide a working solution, obviously more modest than in a supercomputer.

By improving the quantile normalization algorithm large microarray datasets can now be normalized, previously not achievable on a single computer. These methods are generic, and our benchmarking strategy applies to all forms of parallelization of existing algorithms.

Our purpose is to provide mechanisms in a parallel library that compute static, dynamic and guided self scheduling and load distribution algorithms, as well as functions for matrix mapping on disk, in an open source package. These novel quantile normalization routines will be freely available with bindings for Perl, Python, Ruby and R, through Biolib mappings (http://biolib.open-bio.org/).

[1] Bolstad, B.M., Irizarry R. A., Astrand, M., and Speed, T.P. (2003), A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance. Bioinformatics 19(2):185-193

Taverna workflows across Grids, web services and the command line

Hajo N. Krabbenhöft, Daniel Bayer, Steffen Möller

University of Lübeck, Institute for Neuro- and Bioinformatics,
Ratzeburger Allee 160, 23538 Lübeck, Germany

Introduction

Technologies introduced for grid computing support the arbitrary sharing of resources. While the term 'Grid' is commonly associated with high performance computing, other aspects are access control, shared storage, and an increased variability in available data sources and applications that can be accessed via the command line.

Last year we have introduced an interface for Taverna to access resources of the Advanced Resource Collector (ARC) grid middleware (Krabbenhöft et al., Bioinformatics, 2008). It allowed the user of Taverna to upload the grid certificate and to retrieve from a database a series of use cases of common applications in bioinformatics that may be plugged into regular Taverna workflows. These are offered in Taverna like any other action, the user needs no knowledge of grid technology or technical details about the invocation of the respective program. All that is needed is the repository URL and a certificate to grant him computational resources.

Taverna has now evolved into version 2 and the ARC grid interface was adapted, accordingly. For testing purposes and as a fallback for offline operation, support for local invocation and a local distribution via SSH was implemented. Furthermore, the plug-in can now read use cases from local XML files or from custom repository URLs. This way, also internal applications can be offered as Taverna use cases and be intertwined with grid jobs.

Implementation

The ARC plug-in is written completely in Java and was tested on multiple Linux platforms, MacOS X and Windows. Uses cases are stored as an XML-based description for the assembly of

- UNIX command lines,
- Grid jobs,
- Taverna workflow elements

. Use cases are maintained collaboratively. The selection of compute sites is performed by a reference to runtime environments, which different subsets of the sites on a grid have installed.

One special addition to the regular Taverna introduced by the plug-in was the handling of data, which could also be addressed by reference. Since Taverna 2 offers referencing capabilities by default, the implementation was changed to use those with gsiftp URLs for grid storage elements.

Challenges

The biggest challenge is probably the handling of errors. In complex workflows, there should be a way to continue computations when a site is temporarily not available or a single computation has failed because of unforeseen hardware incompatibilities etc. With many redundant sites available on a grid, here grid computing differs from the direct accession of site-specific web services, error recovery mechanisms may be most beneficial. The plug-in extends Taverna's retry logic in a way that it will loop through different grid sites, thereby offering a lower failure rate by exploiting the inherent redundancy of computational grids.

ARC is currently evolving and offers web-services to control jobs and request status updates. Also, job migration and auto-resubmissions are coming. There is consequently the tendency wait for the middleware prior to implementing redundancies.

The availability of applications to be executed is another major concern. This will be helped by an automated deployment of runtime environments to grid sites that is currently under preparation. New software will then be available sooner and at a larger distribution across the grid, allowing for more complex workflows.

Availability

The Taverna plug-in to ARC is available on `http://grid.inb.uni-luebeck.de`. To join the NordGrid see `http://www.nordugrid.org`.

Grid-based expression QTL analysis

Ann-Kristin Grimm[a*], Jan Kolbaum[a], Saleh Ibrahim[b], Steffen Möller[a*]
[a] University of Lübeck, Institute for Neuro- und Bioinformatics, [b] University Medical Center
Schleswig-Holstein, Department of Dermatology, Ratzeburger Allee 160, 23538 Lübeck, Germany

Project URL: `http://eqtl.berlios.de/`
Code URL: `http://developer.berlios.de/git/?group_id=10090`
License: Affero GPL

Introduction

For finding new ways to treat diseases, an understanding of the molecular pathophysiology is extremely beneficial. For the analysis of complex diseases, the past years showed an increasing number of whole genome SNP association studies (GWAs). In animal models, microsatellites serve as genetic markers and combined with directed breeding of susceptible and resistant strains, disease-associated chromosomal loci (QTL) are determined.

For SNPs or QTLs, expression data of the genotyped samples may yield a functional characterisation of the genetic variation, allowing for the determination of expression QTL (eQTL). For QTL analyses, an increased density of genetic markers in regions with observed cross-overs and varying phenotypes may contribute to narrow down the disease causing regions. Later stages of a GWA may stimulate the sequencing of regions of interest. Data added demands a recomputation of previous analyses.

The computational effort for these computations and the amount of data are enormous. It needs to be organised and prepared for a computationally assisted interactive analysis with the researchers. The analyses demand many CPU years, which is not available to most research groups and is needed only between rounds of further wet-lab analyses, which should be as short as possible. Since the problem is data parallel when every gene is treated independently, there is barely a limit on the number of CPUs that can be employed for the computations.

Implementation

We present an approach that adopted the grid technology for the computations and established a workflow for the processing and presentation of the data. It uses the Advanced Resource Connector (ARC) grid middleware (M. Ellert *et al.*, 2007) to execute R scripts with the R/qtl library. Perl and MySQL organise the data, dynamic web pages with Apache/PHP allow for the presentation.

The SQL database is responsible for both, the representation of the computed data and the organisation of the jobs and data. The web interface to this data allows for browsing the computational results in an intuitive way and displaying additional biological information. This comprises local data on yet unpublished disease-associated regions and public data that is retrieved from Ensembl (T.J.P. Hubbard *et al.*, 2007) to faciliate the interpretation of the results.

Dependent on the computational results, new data is generated in the wet-lab and addtional computations are performed. Every result in the database has a reference to the compute task that yielded the finding. This way, refinements of previous results can be achieved incrementally: when there is demand, the internal task status table is indicating a recomputation that will substitute the earlier data.

The job submitted to the grid is a shell script, which starts an R script, which in turn request from a dynamic web page at a static URL a second R script. That second script directly corresponds to an internal compute task. It knows what data to retrieve and how to name the output files. When a time slot is used up or there are no more tasks to compute, the job ends. The results are continously downloaded from the grid and the database updated. New results lead to new information and trigger the production of more wet-lab data, a process to possibly be iterated multiple times.

Challenges

The grid technology helped establishing a direct interaction between wet-lab and computational analysis, i.e. the computed data can be supplied shortly after obtaining updates on the wet-lab data, and thus allows for refinements of the analyses. One remaining challenge is the complete automation of that process.

The current web interface performs analyses that are close to the data. The challenge here is to support the biological user more with machine learning techniques that attempt to model the disease, i.e. to provide access via pathways or functional classifications. One could use such generated hypothese to formulate more advanced analyses in statistical genetics and run these automatically.

[*] to whome correspondence should be adressed: {grimm, moeller}@inb.uni-luebeck.de

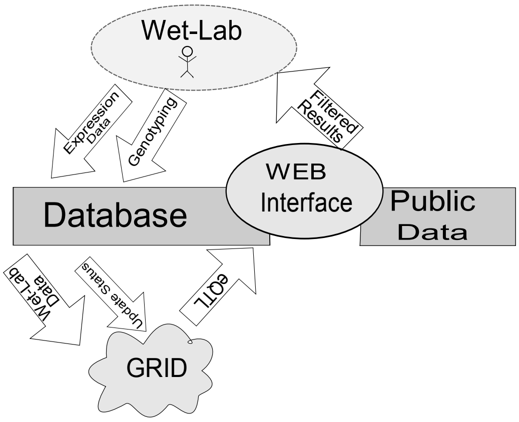


Figure 1: Graphical illustration of the data flow between wet-lab and computational analysis.

Acknowledgements

The authors thank Olli Tourunen and the NDGF BioGrid initiative for providing the ressources. Mélanie Thessén Hedreul and Maja Jagodic are thanked for testing and comments.

References

M. Ellert *et al.* (2007). Advanced resource connector moddleware for lightweight computational grids. *Future Gener. Comput. Syst.*, **23**, 219–617.

T.J.P. Hubbard *et al.* (2007). Ensembl 2007. *Nucleic Acids Res.*, **35**, D610–617.

The Nordic BioGrid project – Bioinformatics for the grid

Joel Hedlund[1], Olli Tourunen[2], Josva Kleist[2], Michael Grønager[2], Steffen Möller[3], Erik Sonnhammer[4], Ann-Charlotte Berglund Sonnhammer[5], Inge Jonassen[6], and Bengt Persson[1,7,8]

[1] IFM Bioinformatics, Linköping University, S-581 83 Linköping, Sweden
[2] Nordic Data Grid Facility, Kastruplundgade 22, DK-2770 Kastrup, Denmark
[3] Institut für Neuro- und Bioinformatik, University of Lübeck, D-23538 Lübeck, Germany
[4] Stockholm Bioinformatics Center (SBC), Stockholm University, S-106 91 Stockholm, Sweden
[5] Linnaeus Center for Bioinformatics (LCB), Uppsala University, S-751 05 Uppsala, Sweden
[6] CBU, Bergen Centre for Computational Science, N-5020 Bergen, Norway
[7] Dept of Cell and Molecular Biology, Karolinska Institutet, S-171 77 Stockholm, Sweden
[8] National Supercomputer Centre (NSC), S-581 83 Linköping, Sweden

E-mail: yohell@ifm.liu.se

Project web page: http://wiki.ndgf.org/display/ndgfwiki/BioGrid
Software web page: http://wiki.ndgf.org/display/ndgfwiki/BioGrid+software
Open source license: MIT license

Life sciences have undergone an immense transformation during the recent years, where advances in genomics, proteomics and other high-throughput techniques produce floods of raw data that need to be stored, analysed and interpreted in various ways. Biology and medicine have become information sciences and new areas of comparative biology have opened. Bioinformatics is crucial by providing tools to efficiently utilize these gold mines of data in order to better understand the roles of proteins and genes and to spark ideas for new experiments.

BioGrid is an effort to establish a Nordic grid infrastructure for bioinformatics, supported by NDGF (Nordic DataGrid Facility). BioGrid aims both to gridify computationally heavy tasks and to coordinate bioinformatic infrastructure efforts in order to use the Nordic resources more efficiently. Hitherto, the widely used bioinformatic software packages BLAST and HMMer have been gridified. Furthermore, the multiple sequence alignment programs ClustalW, MAFFT and MUSCLE have been made available on the grid. Regarding databases, the frequently used databases UniProtKB and UniRef have been made available on the distributed and cached storage system within the Nordic grid. A system for database updating has been deployed in a virtual machine hosted by NDGF. The database PairsDB updates are currently being run on BioGrid & M-grid resources. Further applications are in the pipeline to be gridified including molecule dynamics and phylogeny calculations.

The BioGrid has already contributed to provide computational power for analysis of the medium-chain dehydrogenase/reductase (MDR) superfamily. The size and complexity of this superfamily has recently been shown to far surpass the means of subclassification that have traditionally been employed for this task. Instead, more computationally demanding methods must be employed, such as profile Hidden Markov Models, implemented in the HMMer package.

The presentation will describe BioGrid from a user perspective.

Cytoscape Springs Forward: Re-architecture for Version 3.0

Allan Kuchinsky[1], Keiichiro Ono[2], Michael Smoot[2], Trey Ideker[2],Annette Adler[1]
[1] Agilent Technologies, 5301 Stevens Creek Blvd., Santa Clara, CA 95051 USA
[2] Department of Genetics, University of California, San Diego, CA 92093 USA
allan_kuchinsky@agilent.com

Website: http://cytoscape.org
License: The Cytoscape core distribution is available under GNU Lesser Public License (LGPL). Plugins each have their own licensing policies, most are freely available from the Cytoscape web site.

Cytoscape is an open source bioinformatics software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other data. The Cytoscape core distribution provides a basic set of features for data integration and visualization. Additional features are available as plugins. Plugins are available for network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases. Plugins may be developed by anyone using the Cytoscape open API based on Java™ technology and plugin community development is encouraged.

Over the past several years, Cytoscape has become a standard resource in academia and industry for bio-molecular network analysis. Cytoscape usage has increased at a rate of ~50% per year. To date, Cytoscape has been downloaded over 57,000 times, with a current download rate of 2500/month. There are 74 registered plugins, freely available from the Cytoscape Web site.

This success has engendered some growing pains. While powerful and feature-rich, Cytoscape is a monolithic application presenting a large and complex interface to developers. Like other open source projects with large developer communities, parts of the software have evolved organically with contributions from many programmers. Application components have become tightly-coupled and increasingly fragile. Small changes in one subsystem can have unintended consequences in another, making it difficult to add new features. Moreover, there are currently no mechanisms for plugins to communicate with each other or for supporting different versions of libraries.

To solve these problems, we are re-architecting Cytoscape to be more scalable and flexible and to ease the task of writing plugins, with the following design requirements:

- Organize the source code into clearly defined modules, each with clearly defined interfaces and function.
- separate interfaces (API) from implementation. This enables us to replace actual implementations without breaking existing plugins.
- Simplify the public plugin API. Currently, the Cytoscape public API contains >5000 classes.
- Function across various computing contexts. Such contexts include using Cytoscape to serve as a backend processor for a web service; to run network analyses within a GRID or cluster computing environment including effective use of modern multi-core CPUs; and control via a command line for batch processing.
- The modularization process must ensure that Cytoscape does not lose current functionality or performance.

We are re-architecting Cytoscape using the suite of tools and interfaces provided by the Open Services Gateway Initiative (OSGi, http://osgi.org) and the Spring Dynamic Modules framework (http://www.springsource.org/osgi). The goal is to provide a service architecture which allows modules to register services to be used by other modules and advertised based only on interfaces, so that implementation details are hidden. This re-architecture is the basis of the next major release of Cytoscape, version 3.0. As we proceed, we are addressing several tough design issues.

- Event handling: how do we optimize performance when large amounts of events are being fired, e.g. how to avoid unnecessary network redrawing without introducing dependencies between event producer and consumer.
- Model vs. view: what view information, such as nodes coordinate positions, really should be persisted.
- Handling multiple networks: when to share vs. copy attribute values for equivalent nodes/edges .
- How simple an API? If too simple, do we risk proliferation of ad hoc solutions for common operations?
- The use of new technologies/frameworks will increase the learning curve for plugin developers. We are trying to solve this by creating templates (Maven archetype) and writing tutorial documents.

I will discuss these design issues during my presentation. I will also illustrate these points from the perspective of a plugin, the Cytoscape network editing tool, which brings these issues to bear. I will be interested to learn how fellow participants may have addressed some of these issues.

Bioinformatics simplified with seqpad

De Beule, D.[1], Decouttere, F. [1], Trooskens, G. [2], Devisscher, M. [1] & Van Criekinge, W. [2]

[1] Genohm
Technologiepark 3 bus 9
B9052 Zwijnaarde, Belgium

www.genohm.com

{david|frederik|martijn}@genohm.com

[2] Laboratory for Bioinformatics and
Computational Genomics
Dept. Molecular Biotechnology
Fac. Bioscience Engineering, Ghent University
Coupure Links 653
B9000 Gent, Belgium
{geert.trooskens|wim.vancriekinge}@ugent.be

Url: www.seqpad.org
Source code: trac.seqpad.org | svn.seqpad.org
License: LGPL

Abstract

We present seqpad, an open source sequence visualisation and annotation suite. Seqpad has been available for several years as a commercial package from Genohm.com. Today, in line with our business shift from a product based to a service oriented model, we release our flagship product to the open source community.

Seqpad is a Swing application built in Java 1.6 and interfacing with BioJava 1.6 and BioSQL. It runs on Windows, Mac and Linux. Seqpad is available from www.seqpad.org, or can be checked out from our svn repository.

While several good visualisation and annotation packages exist in the open source community, we believe that Seqpad represents important contributions in usability, functionality, visualisation and automation.

First off, Seqpad has been developed with ease of use as one of the main design concepts, and targets as a user the biologist rather than the bioinformatician, while still leveraging the versatility and power of the Biojava platform. Data in Seqpad is organized in projects. Projects can contain a wide variety of typical bioinformatics files, such as sequences, multiple alignments, phylogenetic trees and protein structure models, but also spreadsheets and word processor documents. We feel this is a very intuitive way for scientists to organize data. The project view has very intuitive controls, e.g. copying a fasta sequence from a browser window and dropping it on the project view will automatically result in a file being created in the project view holding the sequence. The fasta format is automatically recognized, so the sequence can immediately be visualised in a visualisation dock.

The visualisation of seqences, alignments, trees and dotplots is based on the Piccolo 2D Graphics framework (http://www.piccolo2d.org). Piccolo is a framework that introduces the concept of Zoomable User Interfaces. A ZUI is a new kind of interface that enables showing a huge canvas of information on a traditional computer display by letting the user smoothly zoom in, to get more detailed information, and zoom out for an overview.

Seqpad features automated switching between protein/dna views, and also has a custom built openGL based 3D renderer for protein structure models. All visualisations are tightly coupled: for example, selecting a dna sequence in the zoomable sequence interface also highlights the relevant part in protein 3D structure, a very useful feature when interpreting biological relevance of conserved regions for example. The 3D viewer also facilitates quick docking experiments.

With the 'time line' feature it is possible to slide a virtual time line over a phylogenetic tree and see the matches in the coupled multiple alignment change over time.

Seqpad also features an automatic scheduler: Blast queries launched from seqpad for example automatically create a poll job on an internal stack, so the user is not required to poll for results, and results automatically end up in the project view.

The presentation will include a use case demonstrating features of the product. Our main developers will be present, so people wanting to get involved can get started immediately !

DASGenExp: an interactive web-based DAS client with client-side rendering

Bernat Gel[1,*] and Xavier Messeguer[1]

[1] *Dept. de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya UPC*

DASGenExp: `http://gralggen.lsi.upc.edu/recerca/DASgenexp`
Source Code: `http://www.lsi.upc.edu/~bgel/dasgenexp/dasgenexp.tar.gz`
License: GPL v3

The Distributed Annotation System (DAS) offers access to a great number of annotation sources worldwide in a seamless and integrated way. DAS is a client-server protocol designed around the "dumb servers and smart clients" paradigm so anyone with a network connected machine can easily set up a new DAS source. There are more than four hundred DAS servers currently active providing annotations for either genomic or protein sequences for tens of organisms.

As the amount of genomic data grows, the need of better visualisation tools increases. Although web-based DAS browsers exist both for genomic and protein DAS data, genomic oriented browsers generally lack direct interaction capabilities.

DASGenExp is a web based visualisation tool with client-side rendering designed to interactively explore the genomic data and genome based annotations available via DAS. DASGenExp is easy to use and intuitive and has a user interaction scheme similar to that of Google Maps. The user can explore the data by simply dragging and zooming with the mouse. It also offers some unique functions not found in other DAS clients: multiple and independent genomes at the same time, multiple zoom views, representation customization... DASGenExp can integrate annotations from any DAS server and create a graphical representation of the genomic features along with the reference sequence. A preliminary version of DASGenExp can be freely accessed at `http://gralggen.lsi.upc.edu/recerca/DASgenexp/`. DASGenExp has been released under the GNU General Public License version 3 and its cource code can be obtained from `http://www.lsi.upc.edu/~bgel/dasgenexp/dasgenexp.tar.gz`.

In contrast to other web based genomic browsers which display server created images, DASGenExp moves the rendering process to the client side. Raw data is transferred to the client machine and cached. Any further representation of that data does not trigger any network activity and so the overall responsiveness when zooming or panning is greatly increased. Client-side rendering also offers a good opportunity for the customization of data representation and DASGenExp allows the user to easily change colours, shapes, order and visibility of data tracks.

The DASGenExp client is pure javascript and takes advantage of some of the newest browser technologies like the canvas element in order to produce its data representation. Tracks are rendered in independent canvasses and redrawn every time any of the drawing parameters change. Due to the use of optimized data structures and some zoom dependant data preprocessing on the server, data rendering is fast and mostly unappreciable. When the rendering is finished, a canvas is treated like an image by the browser, so panning can be done without jumps or glitches. When dealing with potentially thousands of elements, the fully procedural API offered by canvas is much more convenient than the object oriented one offered by technologies like SVG, since it avoids most of its overhead.

Client-side rendering may be slow and resource intensive for some very dense data tracks but javascript is getting faster and lighter everyday and some big companies are working on it. Due to the use of the canvas element, at the moment, DASGenExp works in Firefox 2 and above but may have some problems in other browsers, mostly in old or non standards compliant browsers.

Due to its inherently highly-multiscale nature, genomic annotation data greatly benefits from zoomable interfaces able to produce zoom dependant representations which maximize the information given to the user. DASGenExp offers such an interface and uses client-side rendering techniques to produce informative, interactive and customizable representations of genomic DAS data.

[*] Electronic address: `bgel@lsi.upc.edu`

Web Service Interface for G-language Genome Analysis Environment

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

G-language Genome Analysis Environment (G-language GAE) is a set of Perl libraries for genome sequence analysis that is compatible with BioPerl, equipped with several software interfaces (interactive Perl/UNIX shell with persistent data, AJAX Web GUI, Perl API). The software package contains more than 100 original analysis programs especially focusing on bacterial genome analysis, including those for the identification of binding sites with information theory, analysis of nucleotide composition bias, analysis of the distribution of characteristic oligonucleotides, analysis of codons and prediction of expression levels, and visualization of genomic information. First version of G-language GAE was released in 2001, and the latest release is currently 1.8.8 (14th March, 2009).

In this lightning talk, we highlight the following recent implementations of web-service interfaces for G-language GAE, mainly developed during and after the BioHackathon 2009 to provide higher interoperability:

REST Services (http://rest.g-language.org/):
This interface provides RESTful URL-based access to all functions of G-language GAE, which is highly interoperable to be accessed from other online resources. For example, graphical result of the GC skew analysis of *Escherichia coli* K12 genome is given by http://rest.g-language.org/NC_000913/gcskew.

SOAP Services (http://soap.g-language.org/g-language.wsdl):
This interface provides language-independent access to more than 100 analysis programs. The WSDL file contains descriptions for all available programs in a single file, and can be readily loaded in Taverna 2 workbench to integrate with other services to construct workflows.

Lightweight Module (Bio::Glite):
This module is a wrapper around the REST services available at CPAN, with minimal number of external modules for easy installation, and with minimal computational resource requirement.

References:

1. Arakawa K, Mori K, Ikeda K, Matsuzaki T, Kobayashi Y, Tomita M, "G-language Genome Analysis Environment: a workbench for nucleotide sequence data mining", *Bioinformatics*, 2003, **19**(2):305-306.
2. Arakawa K, Tomita M, "G-language System as a platform for large-scale analysis of high-throughput omics data", *Journal of Pesticide Science*, 2006, **31**(3):282-288.
3. Arakawa K*, Suzuki H, Tomita M, "Computational Genome Analysis Using The G-language System", *Genes, Genomes and Genomics*, 2008, **2**(1): 1-13.

URL(project): http://www.g-language.org/
URL(code): http://sourceforge.jp/projects/glang/releases/
License: GNU General Public License

Best of both worlds : combining the user-friendliness of Wikis and the rigor of biological databases

Sylvain Brohée [1,*,+], Roland Barriot [1,2,3,+] and Yves Moreau [1]

(1) ESAT-SCD. Katholieke Universiteit Leuven. Kasteelpark Arenberg 10, B-3001 Leuven. Belgium.
(2) Université de Toulouse. UPS. Laboratoire de Microbiologie et Génétique Moléculaires. F-31000 Toulouse. France.
(3) Centre National de la Recherche Scientifique. LMGM. F-31000 Toulouse. France.

* e-mail : sylvain.brohee@esat.kuleuven.be
+ These authors contributed equally to this work.

Project URL : http://www.mediawiki.org/wiki/Extension:Inout
Code URL : http://www.esat.kuleuven.be/~bioiuser/chdwiki/inout.tar.gz
License : GNU - GPL

Wiki technology does not need to be introduced anymore. Indeed over these last few years, the number of wiki-based websites exploded, ranging from very general (e.g. the well known Wikipedia, Wiktionnary) to more specialized (WikiProteins (Mons *et al*, 2008), Wikipedia for genes (Huss *et al*, 2008), etc). Besides the main advantages of wikis (nice presentation, easy editing, large web community, etc), their main drawback is that knowledge contained in the wiki is represented as free text hosted in a unique underlying database. This is in total opposition to the current practice for biological databases where the available data come from well structured repositories. Indeed, a lot of web accessible biological databases mainly consist of a set of entries extracted from tables.

We present, *inout*, an extension to the MediaWiki PHP web software run by the majority of wiki websites and supported by a wide and very active community of developers and users. This extension allows the use of any other data source (structured or not), that is different from the main database, to populate the pages of the wiki. The advantages of this extension are numerous : e.g. the database user is confronted with an intuitive environment he is familiar with and as we keep the ergonomics of the wiki, it remains very easy for him to edit and to contribute to the database. As we also implemented a mean to include forms in the wiki itself, it offers one the possibility to edit in a very user-friendly manner the data present in the external databases on which the wiki may rely using *inout*. Moreover, as the user rights management is implemented in the MediaWiki software, the modifications can be made by some registered experts of the wiki application field.

Furthermore, our extension allows one to automatically include data on predefined pages. The data can be free wiki text as well as data obtained from external databases or webservices. Thus, if a page has not yet been manually created by the wiki contributors, external data may already fill it. Of course, the manual completion of pages is still possible afterwards.

Finally, via this extension, any web tool, like those found in classical biological databases (BLAST search, network analysis, gene prioritization, etc.) can also easily be integrated into the wiki.

Our extension has already been successfully deployed for two very different biological domains : a database dedicated to the study of congenital heart defects (CHDWiki : http://homes.esat.kuleuven.be/~bioiuser/chdwiki/) and a database devoted to the precise classification and annotation of yeast permeases (YTPdb : http://homes.esat.kuleuven.be/~sbrohee/ytpdb).

BNfinder: free software for effective Bayesian Network inference

Bartek Wilczynski <wilczyns@embl.de>*

URL http://launchpad.net/bnfinder

code http://code.launchpad.net/bnfinder/trunk

license GNU General Public License

In recent years, we have seen an increased interest in applications of Bayesian Networks (BNs) in modelling in molecular biology. It is not surprising, since BNs are very natural models for reconstructing dependencies between observables, especially when measurements are noisy. They have been particularly successful in the field of regulatory genomics[1, 6], where the need for uncovering causal relationships between different variables is crucial.

While BNs are very flexible and naturally interpretable models, they are inherently difficult to train[2]. So far, the most common approach to training Bayesian networks were methods based either on Monte Carlo Markov chains[5] or on Expectation Maximization[6]. These approaches share common problems – they only provide approximate solutions while being computationally intensive. Recently, it has been shown by Dojer[3] that, under additional assumptions, the optimal solution to BN reconstruction problem can be found in polynomial time.

BNfinder[7] is a python implementation of this exact and efficient algorithm for BN reconstruction. It is distributed under GNU General Public License, so it can be freely used or adapted by other researchers. Since our project is currently still in its early stages, we are very open to any comments, suggestions or code contributions coming from potential users.

In this talk, I will briefly describe current state of the project and its design. Then I will describe three different examples of real problems that can be solved with BNfinder:

- reconstructing small gene networks from expression measurements both wildtype[8] and under perturbation[4]
- finding informative sequence motifs for groups of coexpressed genes[1]
- predicting gene expression from cis-regulatory modules (in preparation)

References

[1] Michael A Beer and Saeed Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2):185–198, Apr 2004.

[2] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag, 1996.

[3] Norbert Dojer. Learning Bayesian networks does not have to be NP-hard. *LNCS*, 4162:305–314, 2006.

[4] Norbert Dojer, Anna Gambin, Bartek Wilczyński, and Jerzy Tiuryn. Applying dynamic Bayesian networks to perturbed gene expression data. *BMC Bioinformatics*, 7:249, 2006.

[5] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–82, 2003.

[6] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, 34(2):166–76, 2003.

[7] Bartek Wilczynski and Norbert Dojer. Bnfinder: Exact and efficient method for learning bayesian networks. *Bioinformatics*, page btn505, September 2008.

[8] Bartek Wilczyński and Jerzy Tiuryn. Reconstruction of mammalian cell cycle regulatory network from microarray data using stochastic logical networks. *Lecture Notes in Bioinformatics*, 4695/2007:121–135, 2007.

*EMBL Heidelberg and Institute of Informatics, University of Warsaw

MOLGENIS by example: generating an extensible platform for genotype and phenotype experiments.

Morris A. Swertz[1,2] (m.a.swertz@rug.nl), K Joeri van der Velde[2], Rudi Alberts[3], Bruno M. Tesson[2], Richard A. Scheltema[2], Gonzalo Vera[2], Damian Smedley[6], Katy Wolstencroft[7], Paul Schofield[4], Klaus Schughart[3], John M. Hancock[5], Engbert O. de Brock[2], Andrew R. Jones[8], Helen E. Parkinson[6], Ritsert C. Jansen[2] and members of the EU-CASIMIR consortium for mouse, EU-GEN2PHEN consortium for human, EU-Panacea consortium for C. elegans and NL-NWO QTL Express consortium for A. thaliana.

[1]Department of Genetics, University Medical Center Groningen, P.O. Box 30.001, 9700 RB Groningen, The Netherlands; [2]University of Groningen, The Netherlands; [3] Helmholtz Centre for Infection Research, Braunschweig, Germany; [4]MRC Harwell, United Kingdom; [5]University of Cambridge, United Kingdom; [6]European Bioinformatics Institute, United Kingdom; [7]University of Manchester, United Kingdom [8]University of Liverpool, United Kingdom.

Software and source:	http://molgenis.sourceforge.net and http://www.xgap.org/
Licence:	LGPLv3.

MOLGENIS is a general toolbox to auto-generate complete database software from compact models, including user and programmatic interfaces. Here we describe its use to generate XGAP, an eXtensible software platform for high-throughput Genotype And Phenotype experiments:

A growing array of genetic, transcript, protein and metabolite profiling technologies, natural and model organism populations, and GWA, GWL and mutagenesis experimental designs are at our disposal with each different strengths and weaknesses. Integrated analysis of all these genotypes and (molecular) phenotypes provides new opportunities to unravel the genome-to-phenome trajectory. However, the data formats and tool interfaces used by each experiment often also differ, impeding comparison, exchange and integration of data and tools within and between experiments, laboratories and consortia.

To mold existing and new data sets and analysis tools into a singular medium we developed an open software platform named XGAP benefitting both extremes on the user spectrum: experimentalists and computational researchers. To quickly generate a uniform looking software and ease extension with new profiling technologies and methods, we used the open source MOLGENIS software platform. In MOLGENIS, most parts of the software infrastructure can be blueprinted in a compact XML model file that is automatically converted into the many Java, SQL and R code files needed via generator templates written in Freemarker.

We blueprinted a minimal core data model building on standards such as FuGE, MAGE-TAB and OBO and added profiling method specific extensions. Then we auto-generated the basic platform having (i) an easy to create delimited file format to load and exchange data, (ii) a suitable database for storage and querying, (iii) programmatic interfaces to java, web services and the statistical R tools for computational researchers to plug-in tools and (iv) web user interfaces to manage and query data and run plugged-in analysis tools for experimentalists. Hand-written features were plugged into the generated software, such as import/export wizards and a large data matrix viewer, bridges to R/QTL, Ontology Lookup Service, and KEGG services; more plug-ins to GMOD/Gbrowse, standard GWA software and Bioconductor packages are planned.

Other researchers can (and have) edit(ed) the blueprint and add plug-ins to generate an extended XGAP version that suits their particular needs but still adheres to the standard XGAP formats. We are optimistic that XGAPs uniform data representation and MOLGENIS-based extensible software infrastructure will help the communities of genotype/phenotype researchers to share data and tools notwithstanding large variation between research aims.

PSLID, the Protein Subcellular Location Image Database: Subcellular location assignments, annotated image collections, image analysis tools, and generative models of protein distributions

Estelle Glory[123], Justin Newberg[123], Tao Peng[12], Ivan Cao-Berg[14], Robert F. Murphy[12345]

[1]Center for Bioimage Informatics, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[2]Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[3]Molecular Biosensor and Imaging Center, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[4]Department of Biological Sciences, Department of Machine Learning, Ray and Stephanie Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[5]External Fellow, Freiburg Institute for Advanced Studies, Freiburg, Germany

email: Robert F. Murphy (murphy@cmu.edu)
URL for the overall project web site: http://pslid.cbi.cmu.edu
URL for accessing the code: http://murphylab.web.cmu.edu/software
The particular Open Source License being used: GPL

Subcellular location is a fundamental characteristic of proteins, and knowledge of subcellular location on a proteome-wide basis will be critical to building accurate models of complex cellular behaviors. Subcellular patterns are often quite complex and difficult, if not impossible, to capture using text-based approaches. Since in many cases the primary data for determining subcellular location is in the form of images of fluorescently-tagged proteins, widespread availability of annotated protein images is essential. The Protein Subcellular Location Image Database (http://pslid.cbi.cmu.edu), which is currently at Release 4, is the earliest publicly available image database of subcellular patterns, with the first release occurring in 2001. It contains image collections from a number of research groups, in particular, large collections of 2D and 3D fluorescence microscope images that have been widely used to train and test systems for automatically analyzing subcellular patterns. Over the past decade, these systems have been demonstrated to be more sensitive at distinguishing subcellular patterns than visual analysis. PSLID provides access to either entire collections or to user-specified subsets of the collection via either an interactive search interface or a programmatic search interface. In addition to providing images, PSLID provides public access to a large suite of tools for automated analysis of those images. These include preparatory functions (such as segmentation into individual cells and extraction of numerical features to capture subcellular distributions), and core analysis functions (including the statistical comparison of location between image sets, training and using of machine classifiers, and clustering of images or proteins by their location patterns). Most critically, PSLID includes state-of-the-art methods for inference of patterns across multi-cell images using graphical models and tools for building and distributing generative models of subcellular patterns (models that allow synthesis of new images drawn from the same underlying distribution as a set of training images). The database and analysis software are not only accessible via web service but can be downloaded for local use. A number of the tools are also available separately so that they can be used without requiring installation of the PSLID database engine. These are available both as Matlab toolboxes and as standalone applications.

Title: BioHDF: Open binary file formats for large-scale data management – Project Update

Authors: Todd M. Smith [1], N. Eric Olson [1], Mark Welsh [1], Mike Folk [2]

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

Presenting Author: Mark Welsh – markw@geospiza.com

Project URL: http://www.hdfgroup.org/projects/bioinformatics/

Download URL: http://www.hdfgroup.org/projects/bioinformatics/bio_software.html

Licensing: BSD-style licensing for HDF5 libraries and BioHDF data access software;
GPL licensing for BioHDF data analysis software.

Abstract: The huge volume of data produced by novel sequencing technologies presents significant challenges around 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 is investigating the use of a mature technology for the storage and retrieval of very large scientific datasets – Hierarchical Data Format. Supported by The HDF Group, HDF5 provides a binary file format, a collection of APIs supporting data access (C-based, with bindings to Fortran and other languages), as well as comprehensive test cases to ensure the stability of this software. HDF has been in use for over 20 years in fields that have faced similar data challenges to those now being experienced in biology, such as aircraft flight test data (Boeing), Earth observation data (NASA, NOAA), and CAD/CAM data (EU).

BioHDF will extend HDF5 data structures and library routines with new features to support the high-performance data storage and computation requirements of Next Gen Sequencing. To enable its use within the bioinformatics and research communities, BioHDF will be delivered as an Open Source technology that will include: a data model supporting storage of sequences, their alignments against reference data sources, and annotations such as SNP or splice variation analysis; indexing for fast random-access into this data; analysis tools that target specific applications such as RNA-Seq, Tag Profiling, novel micro-RNA identification, and many others on multiple NGS platforms including Applied Biosystems SOLiD, Illumina Genome Analyzer, Roche 454, and Helicos; visualization tools that provide reporting and interactive display of these very large data sets.

Initial prototyping of BioHDF has demonstrated clear benefits. Sequences and their alignments against reference data sources, which occupy 10s of Gb stored as highly redundant text-format files, are just a few percent of that size when stored in BioHDF's compressed, structured binary format. Indexing in BioHDF enables very rapid (typically, few millisecond) random access into these sequence and alignment datasets, essentially independent of the overall HDF5 file size. Additionally, through our prototyping activities we have identified key architectural elements and tools that will form BioHDF.

Title	Lowering barriers to publishing biological data on the web
Authors	Brad Chapman, BioSQL and Biopython communities
Author affiliation	Massachusetts General Hospital, Boston, MA
Contact	`chapmanb@50mail.com`

URLs	`http://biopython.org`
	`http://www.biosql.org`
	`http://bcbio.wordpress.com`
	`http://biosqlweb.appspot.com`

Code URLs	`https://github.com/chapmanb/biosqlweb/tree`
Licenses	Biopython License, GNU LGPL

Talk summary

Scientists making their primary research data available on the web face an abundance of options. Many public databases and formats exist to solve similar problems, reflecting the complex nature of biological data. Additionally, researchers will often wait until the time of publication to format their data for public access. These factors can result in critical data being available as ad-hoc supplementary materials.

Data reuse is facilitated by standard web based presentation tools. An underlying shared architecture allows programmers to provide access points in widely used formats. Scientists gain an advantage from the visualization and organizational structure provided by these tools; this encourages collection of data from the start of a project, and allows data to be made available to other researchers earlier during the research process.

This talk describes a web based interface deployable on cloud computing resources. An existing database representation developed by the BioSQL project is utilized to store sequence data along with associated annotations and features. This data model is ported to the Google App Engine infrastructure, providing a full development stack for rapidly building and deploying web applications. A web interface designed with jQuery and jQueryUI serves as a framework for quickly implementing custom display and editing widgets. Finally, flat file data export is available via Generic Feature Format (GFF3); the structured format facilitates manipulation in Excel or text editors while being amenable to automated parsing.

More generally, the talk will discuss reusing open source frameworks to move towards a cloud-based approach to open data sharing. As data production continues to expand with the growth of next generation sequencing, individual research labs can help share the burden of data organization and presentation. In addition to relying on central repositories, we each take ownership of our data, presenting it in standard ways that encourage reuse and reanalysis.

Open source developers can encourage movement in this direction by simplifying the deployment of existing data models and frameworks. While the example discussed in this talk utilizes the BioSQL and Biopython infrastructures, the goals could be employed with a wide variety of data models and programming libraries. Our focus should be structuring our own open source work to be increasingly amenable to rapid integration.

XMLPipeDB: A Reusable, Open Source Tool Chain for Building Relational Databases from XML Sources

Kam D. Dahlquist[1], Alexandrea Alphonso[1], Derek Smith[2], Chad Villaflores[1],
John David N. Dionisio[2]

[1]Department of Biology, [2]Department of Electrical Engineering and Computer Science, Loyola Marymount University, 1 LMU Drive, Los Angeles, California, 90045 USA

Project Site: http://xmlpipedb.cs.lmu.edu
Source Code Availability: http://sourceforge.net/projects/xmlpipedb
Licence: GNU Library or Lesser General Public License

XMLPipeDB is an open source tool chain for building relational databases from XML sources with minimal manual processing of the data. While its applicability is intended to be general, the original motivation for XMLPipeDB was to create a solution for the management of biological data from different sources that are used to create Gene Databases for GenMAPP (Gene Map Annotator and Pathway Profiler), software for viewing and analyzing DNA microarray and other genomic and proteomic data on biological pathways. XMLPipeDB has a modular architecture with three components that may be used separately or together. XSD-to-DB reads an XSD (XML Schema Definition) and automatically generates an SQL schema, Java classes, and Hibernate mappings. XMLPipeDB Utilities provides functionality for configuring the database, importing data, and performing queries. GenMAPP Builder is based on the XMLPipeDB Utilities and exports GenMAPP-compatible Gene Databases based on data from UniProt and Gene Ontology (GO). We have previously used GenMAPP Builder to create Gene Databases for *Escherichia coli* K12 and *Arabidopsis thaliana*. We have extended GenMAPP Builder by automating the process of creating new Gene Databases for any additional species for which UniProt data are available. We have thus recently created Gene Databases for *Plasmodium falciparum* and *Vibrio cholerae*. We have also added functionality to GenMAPP Builder that automatically checks for data integrity from the original XML source, the intermediary PostgreSQL relational database, and the exported GenMAPP Gene Database. GenMAPP Builder has proved to be robust to changes in the XSDs from UniProt and GO, both of which have changed several times throughout the course of this project. We also report on the compatibility of other common bioinformatics XML formats with the XMLPipeDB suite.

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 wEMBOSS, EMBOSS Explorer, JEMBOSS, SoapLab, Pise, SRS, Taverna and several commercial workflow packages.

EMBOSS 6.1 will be released on 15th July 2009 (we always release on 15th July). New features include:

- C programming library development
- New data formats
- Use of ontologies
- Performance profiling
- DAS protocol support

The EMBOSS project has received 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)
- Closer integration with interfaces (DAS, Galaxy, SoapLab, wEMBOSS, Artemis)
- Genome-scale analysis and annotation
- Next-Generation Sequencing data processing
- Common methods for Open-Bio projects
- 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

Peter Cock*

Bioinformatics Open Source Conference (BOSC) 2009, Stockholm, Sweden

In this talk we present the current status of the Biopython project (www.biopython.org), focusing on features developed in the last year, and future plans. The Oxford University Press journal Bioinformatics has recently published an application note describing Biopython (Cock *et al.*, 2009).

Since BOSC 2008, Biopython has seen two releases. Biopython 1.49 (November 2008) was an important milestone in bringing support for Python 2.6, and in terms of our dependence on Numerical Python as we made the transition from the obsolete Numeric library to NumPy. Biopython 1.49 also added more biological methods to our core sequence object. April 2009 saw the release of Biopython 1.50, new features include:

- GenomeDiagram by Leighton Pritchard (Pritchard *et al.*, 2006) has been integrated into Biopython as the `Bio.Graphics.GenomeDiagram` module.
- A new module `Bio.Motif` has been added, which is intended to replace the existing `Bio.AlignAce` and `Bio.MEME` modules.
- `Bio.SeqIO` can now read and write FASTQ and QUAL files used in second generation sequencing work.

Biopython will celebrate its 10th Birthday later this year, and has now been cited or referred to in over one hundred scientific publications (a list is included on our website). We will present a brief history of the project and current work. This includes the evaluation of git (and github) as a possible distributed version control system (DVCS) to replace our existing very stable CVS server hosted by the Open Bioinformatics Foundation, which we hope will encourage more participation in the project. Also, we currently have two *Google Summer of Code* project students working on phylogenetics code for Biopython in conjunction with the National Evolutionary Synthesis Center (NESCent).

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* doi:10.1093/bioinformatics/btp163

Pritchard, L. *et al.* (2006) GenomeDiagram: A Python Package for the Visualisation of Large-Scale Genomic Data, *Bioinformatics* **22**(5), 616-617. doi:10.1093/bioinformatics/btk021

[*] Peter.Cock@scri.ac.uk – Plant Pathology, SCRI, Invergowrie, Dundee DD2 5DA, UK

BioJava 2009: an Open-Source Framework for Bioinformatics

Andreas Prlić (andreas.prlic@gmail.com), University of California San Diego, San Diego, CA, USA
Thomas Down, Wellcome Trust/Cancer Research UK Gurdon Institute UK
Andreas Dräger, Eberhard Karls University Tübingen, Center for Bioinformatics Tübingen (ZBIT) Germany
Sylvain Foisy, Laboratory in Genetics and Genomic Medicine of Inflammation, Montreal Heart Institute, Montreal, Québec, Canada
David Huen, University of Cambridge, UK
Jules Jacobsen, European Bioinformatics Institute, Cambridge, UK
Keith James, Wellcome Trust Sanger Institute, Cambridge, UK
Michael Heuer, Harbinger Partners, Inc., USA
Matthew Pocock, University Newcastle Upon Tyne, UK
Mark Schreiber,Novartis Institute for Tropical Diseases, Singapore
George Waldon, GeneInfinity, USA
Andy Yates, European Bioinformatics Institute, Cambridge, UK
Richard Holland, Eagle Genomics Ltd., Cambridge, UK

The BioJava website is http://biojava.org/.

The version 1.7 release can be downloaded from http://biojava.org/wiki/BioJava:Download

BioJava is available under the terms of version 2.1 of the GNU Lesser General Public License (LGPL).

BioJava is a mature open-source project that provides a framework for processing of biological data. BioJava contains powerful analysis and statistical routines, tools for parsing common file formats, and packages for manipulating sequences and 3D structures. It enables rapid bioinformatics application development in the Java programming language. Here we present the latest BioJava release (version 1.7, released in Apr 2009).

Besides numerous bug fixes and stability improvements, a lot of development has been going on in the protein structure modules. BioJava now provides a framework for parsing mmCif files. The parsing of PDB header information has been improved and a new tool to read the Chemical component dictionary is in place.

An ongoing project is the open sourcing of the RCSB protein structure viewers. These have been made available as open source as a stand-alone project. Work is currently under way to provide a more close integration with BioJava.

Application of VAMSAS enabled tools for the investigation of protein evolution

J. B. Procter[1*], I. Milne[2], F. Wright[3], P. Marguerite[4], A. M. Waterhouse[1,5], D. Lindner[2], D.M.A. Martin[1], T. Oldfield[4], D. Marshall[2], G. J. Barton[1].
[1]College of Life Sciences, University of Dundee, UK. [2]Scottish Crop Research Institute, UK.
[3]Biomathematics and Statistics Scotland, UK. [4]PDBe group, European Bioinformatics Institute (EBI).
[5]Genome Sciences Centre, RIKEN Yokohama Institute, Japan. [*]contact: j.procter@dundee.ac.uk

Website: http://www.vamsas.ac.uk (source, documentation and related software links)
License: The VAMSAS client library is implemented in Java and available under the **LGPL**.

Analysis of protein evolution involves the application of a combination of methods: nucleic acid and protein alignment, phylogenetic modelling, tree building, and functional annotation analysis. Whilst there are many interactive applications capable of applying one or more of these approaches, very few efficiently enable the user to perform all aspects of such a study. We present an example that demonstrates how this kind of analysis can be performed using applications that have been modified to dynamically exchange data; *via* the 'Visualization and Analysis of Molecular Sequences, Alignments, and Structures' (VAMSAS) framework.

The VAMSAS framework is a data exchange and interoperation framework for bioinformatics applications. It enables them to share sequences, alignments, phylogenetic trees and annotation, and rapidly exchange messages so that each one's visualization of the shared data can be synchronised. For example, messages are exchanged to indicate current display state such as the position of the mouse pointer within an amino acid sequence, or a newly selected area in an alignment. Shared data may also be exported, to provide a complete record of the current state of an analysis that can be exchanged with other researchers.

An LGPL licensed implementation of the VAMSAS framework was developed in Java to enable interoperation between three popular graphical applications: Jalview[1] is a widely used workbench for multiple sequence alignment, visualization, editing and analysis developed at the University of Dundee. TOPALi[2] is a program for evolutionary analysis of protein and nucleotide sequence alignments and phylogenetic tree construction, developed by Biomathematics and Statistics Scotland, and the Scottish Crop Research Institute. AV@MSD-EBI[3, 4] enables the visualization and analysis of protein and nucleic acid structures, and interfaces to web services provided by the Macromolecular Structure Database (MSD) at the European Bioinformatics Institute (EBI). The VAMSAS web site (see above) provides more information about the library, and links to the different programs.

1. Waterhouse AM, Procter JB, Martin DM, Clamp M, Barton GJ: **Jalview Version 2 - a multiple sequence alignment editor and analysis workbench**. *Bioinformatics* 2009, doi:10.1093/bioinformatics/btp033.
2. Milne I, Lindner D, Bayer M, Husmeier D, McGuire G, Marshall DF, Wright F: **TOPALi v2: a rich graphical interface for evolutionary analyses of multiple alignments on HPC clusters and multi-core desktops**. *Bioinformatics* 2009, **25**(1):126-127.
3. Hartshorn MJ: **AstexViewer: a visualisation aid for structure-based drug design**. *J Comput Aided Mol Des* 2002, **16**(12):871-881.
4. Oldfield TJ: **A Java applet for multiple linked visualization of protein structure and sequence**. *J Comput Aided Mol Des* 2004, **18**(4):225-234.

Soaplab: Open Source Web Services Framework for Bioinformatics Programs

Senger M.[1, 2] (m.senger@cgiar.org), Rice P. [1](pmr@ebi.ac.uk), Bleasby A. [1], Uludag M. [1]

[1] European Bioinformatics Institute, [2] Consultative Group on International Agricultural Research

Soaplab is a specialized Web Services framework for command-line bioinformatics programs. Its recent reincarnation Soaplab2 introduced enhancements that made Soaplab servers more reliable than before. However standard Web Services clients were not able to use their full capacities with Soaplab generic interface since it was not presenting complete input/output descriptions in standard way using XSD. In order to address this issue we have worked on a new Soaplab interface that includes input/output type definitions at WSDL level.

Soaplab prepares Web Services for command-line programs based on descriptions written using an extended version of the ACD language from the EMBOSS open source bioinformatics program suite. It includes routines for basic validation of inputs, making inputs available to command-line programs, running programs, and collecting and returning outputs of executed programs [1]. Soaplab's latest version, Soaplab2, was first released in 2007 after it had been refactored and enhanced in several ways, making Soaplab servers more reliable than before.

One of the unique features of Soaplab is its generic interface that makes it possible to use the same interface when accessing any Soaplab Web Services regardless of the command line interface of underlying programs. However Soaplab generic interface restricts the ways WSDL files of individual Soaplab Web Services can be tailored. For example Web Services specific input-output data types are not described at WSDL level; instead the interface accommodates methods to query this information. This difference from common WSDL interfaces, for example, doesn't allow standard Web Services clients to check input data types before sending a request or output data types after a response has been received.

There was a growing concern in Bioinformatics Web Services community that Soaplab should include input/output type descriptions at WSDL level. To address this concern we have extended Soaplab in such a way that service providers can deploy their Soaplab Web Services with WSDL files that includes type descriptions. The new interface, called typed interface, will also facilitate integration of service and data ontologies some service providers are working on.

A beta version of Soaplab typed interface for EMBOSS Web Services was made available on EBI Soaplab server last December. We are planning to include full support for typed interface in the next release of Soaplab (2.1.2) first week of June this year. The new release will also include support for load sharing system LSF.

References:

[1] Senger M., Rice P., Oinn T., “Soaplab - a unified Sesame door to analysis tools“, Proceedings, UK e-Science- All Hands Meeting 2003, p. 509-513, 2003

Open source license used: Apache License, Version 2.0

Project home page: http://soaplab.sourceforge.net/soaplab2/

Project download page: http://soaplab.sourceforge.net/soaplab2/Download.html

BioLib: Sharing high performance code between BioPerl, BioPython, BioRuby, R/Bioconductor and BioJAVA

by Pjotr Prins

Dept. of Nematology, Wageningen University, The Netherlands (pjotr.prins@wur.nl)

website: `http://biolib.open-bio.org/`
git repository: `http://github.com/pjotrp/biolib/`
LICENSE: BioLib defaults to the BSD license, embedded libraries may override

BioLib provides the infrastructure for mapping existing and novel C/C++ libraries against the major high level languages using a combination of SWIG and cmake. This allows writing code once and sharing it with all bioinformaticians - who, as it happens, are a diverse lot with diverse computing platforms and programming language preferences.

Bioinformatics is facing some major challenges handling IO data from microarrays, sequencers etc., where every innovation comes with new data formats and data size increases rapidly. Writing solutions in every language domain usually entails duplication of effort. At the same time developers are a scarce resource in every Bio* project. In practice this often means missing or incomplete functionality. For example microarray support is only satisfactory in R/Bioconductor, but lacking in all other domains. By mapping libraries using SWIG we can support all languages in one strike, thereby concentrating the effort of supporting IO in one place.

BioLib also provides an alternative to webservices. Webservices are often used to cross language barriers using a 'slow' network interface. BioLib provides an alternative when fast low overhead communication is desired, like for high throughput, high performance computing.

BioLib has mapped libraries for Affymetrix microarray IO (the Affyio implementation by Ben Bolstad) and the Staden IO lib (James Bonfield) for 454-sequencer trace files, amongst others, and made these available for the major languages on the major computing platforms (thanks to CMake). Current efforts include mapping of R/QTL analysis libraries and libsequence and/or BioC++ libraries. Also a solution is worked on for documenting API of mapped libraries automatically for all supported languages - something not provided by SWIG. Finally BioLib is attracting new code development with a focus on high performance computing including parallelization and GPU optimizations.

In this talk I will discuss the ins and outs of library mapping with SWIG, what it means, and how BioLib takes the pain away of deployment on different platforms, even with external dependencies.

Debian adopts and disseminates Bioinformatics Open Source Software

Steffen Möller, Charles Plessy, David Paleino, Andreas Tille and the Debian Community*

Looking back

In Bioinformatics we are used to associate technological progress with the advancements in wet-lab techniques that bring us a steadily increased influx of more and more novel data to manage and interpret. Over that we often forget that this is only possible since the IT sciences have evolved even more quickly. This allows us to keeping pace with the data stream while applying even more analyses.

When the first Bioinformatics Open Source Conferences were held in the late 90s, the Internet was still a recent event. To find data on the net was considered special. And that data came at no extra charge. GNU/Linux had emerged as the ubiquitous operating system, as free as the data that was analysed with it. And free were most tools for sequence analysis, with development often funded by the same institutions that funded the wet-lab production of the data. Free also became the Bio{Perl,Java,*} libraries that help analysing the sequence data.

These libraries and many accompanying tools are now being used in many different suites for the handling of biological data. Or they are being used by smallish scripts to help with analyses in smaller or larger research projects. They became a commodity. One has gained sufficient confidence in the community to always want the latest versions of these helpers. Many analysts take the existence of these tools for granted or use them as part of a larger tool while not being aware of them. And those distributing software that is depending on common libraries or tools need ways to ensure a trustworthy installation of the basic research infrastructure. They can give instructions to their users to install everything themselves, can ship precompiled binaries or – suggest a GNU/Linux distribution's packages.

Today

GNU/Linux distributions live from their users. Commercial distributions have opened up for packages organised by the community, i.e. via OpenSuSE and Fedora. Debian GNU/Linux has been a community-driven distribution ever since and with around 50 programs or libraries (plus dependencies) it is the distribution which ships the largest number of bioinformatics packages for the largest number of platforms.

In 2001 Debian introduced a concept now called Debian Blends, a platform for the presentation of software packages for communities with a distinct interest. Bioinformatics is well kept under the hood of the Debian-Med blend with some packages also being found under Debian-SciComp or Debian-Science. Individuals interested to see a bioinformatics software packaged will send an email to the mailing list or fill out a Request for Packaging. Quite often it is a Debian packager amongst the developers of the software or an enthusiastic user, who seeks his self-prepared package to be shipped with the distribution.

Looking forward

A major problem in bioinformatics is the local maintenance of remotely accessibly data. Every group performing research in this field is solving this to their local needs in some way, crafting a series of scripts, but this effort should somehow be shared.

Where Debian, like all other Linux distributions, need help, and this also is particularly obvious in the complex interplay of software in Bioinformatics, is the guidance of users in the interplay of multiple tools. There is yet no package for the education of users or for pre-assembled workflows that address frequently observed problems. This issues was raised in past discussions in the context of the possibility to prepare a BOSC liveCD, which may also be imaginable as an interplay of a regular Linux distribution with a set of Wiki pages to guide the users.

Availability

The Debian home page is `http://www.debian.org`, development on bioinformatics packages is best monitored on `http://debian-med.alioth.debian.org`.

* to whom correspondence should be addressed: debian-med@lists.debian.org

PSODA

Open Source Phylogenetic Search and DNA Analysis

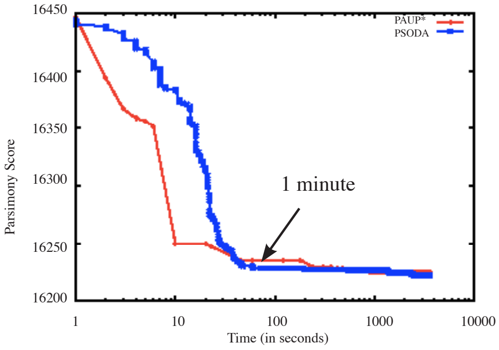
Quinn Snell, Mark Clement, Kenneth Sundberg
Brigham Young University
{snell,clement,kasundberg}@cs.byu.edu
http://dna.cs.byu.edu/psoda --- svn co http://dna.cs.byu.edu/opensvn/psoda

Abstract

PSODA (sō-də) is an open source (GPL v2) sequence analysis package that implements sequence alignment using biochemical properties, phylogeny search with parsimony or maximum likelihood criteria and selection detection using biochemical properties (TreeSAAP). PSODA is compatible with PAUP* and the search algorithms are competitive with those in PAUP*. PSODA also adds a basic scripting language to the PAUP block, making it possible to easily create advanced meta-searches. Because PSODA is open-source, we have also been able to easily add in advanced search techniques and characterize the benefits of various optimizations. PSODA is available for Macintosh OS X, Windows, and Linux.

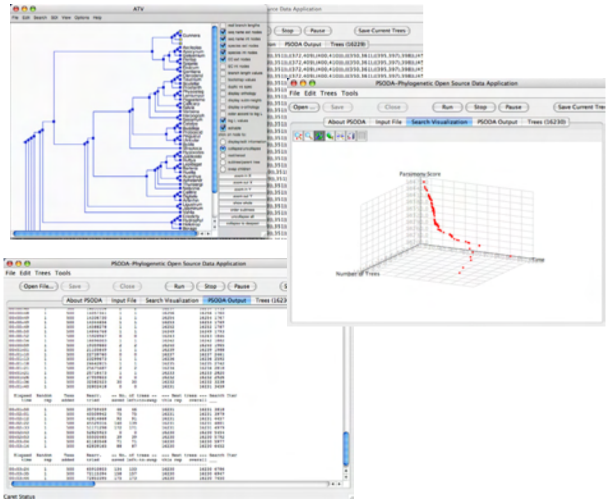
Advantages of PSODA

- High Performance
- Open Source (-: FREE :-)
- Modular Design (easy algorithm development)
- Advanced Scripting Language
  - makes advanced meta-searches simple
- Reads and Executes PAUP nexus files
- PSODA is competive with PAUP*



PSODA Features

- Parsimony and Likelihood (RAxML) search
- Baysian methods (Mr. Bayes)
- Consensus (strict and majority rules)
- Selection Detection (TreeSAAP)
- Graphical User Interface
- Binaries for Mac OS X, Windows and Linux
- Object-oriented C++
  - easy to contribute to new algorithm development
- Available via subversion
  - svn checkout http://dna.cs.byu.edu/opensvn/psoda


ATV
PSODA-Phylogenetic Open Source Data Application
File Edit Trees Tools
Open File... Save Close Run Stop Pause Save Current Trees
About PSODA Input File Search Visualization PSODA Output Trees
Parsimony Score
Number of Trees
Time
Caret Status

Advanced Scripting Language

- Added functionality for PAUP blocks.
- Decision Statements & Loops
- Advanced Functions & User-defined Functions
- Easily Extensible
- Easy scripting of advanced meta-searches such as:
  - Ratchet (Parsimony and Likelihood)
  - DCM and more.

PAUP* Ratchet

```
BEGIN PAUP;
set criterion=parsimony ;
set maxtrees=1
  increase = no;
hsearch start=stepwise swap=TBR ;
weights 2 : 7 14 17 26 27 31 34 45 50 52 54 57 60 63 64 65 73 77 86 91 92 102 103 107 115 117 121
122 124 127 131 133 134 140 142 155 156 163 173 176 183 185 187 195 197 198 202 204 209 219 221 222
225 226 230 237 238 240 241 244 248 252 254 255 258 269 276 279 283 284 291 294 295 297 309 310 315
319 321 323 325 326 342 346 349 350 351 353 354 355 359 360 363 366 370 377 378 390 393 394 398 408
412 413 418 419 423 425 426 428 429 432 450 456 467 475 479 482 484 489 492 493 497 498 500;
hsearch start=current swap=TBR
;
set maxtrees=1;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 17 20 21 29 33 35 39 42 50 57 59 69 70 71 73 80 86 93 96 97 99 100 102 111 121 125 126
141 142 148 149 155 157 163 168 174 179 180 184 187 188 194 203 208 212 216 219 220 225 226 231 242
243 251 253 254 256 257 260 265 269 272 274 277 282 286 287 288 291 293 296 297 305 306 309 315 316
318 324 326 327 329 334 340 346 351 352 356 360 366 370 372 373 379 380 384 390 396 399 401 408 412
414 421 423 426 428 429 441 442 444 448 455 459 460 462 472 473 475 477 478 487 489 493;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 2 9 11 14 16 18 20 24 28 36 40 43 45 47 59 60 62 64 65 70 72 84 90 92 105 107 111 112
118 122 129 130 135 138 146 148 152 153 156 161 162 167 172 173 177 190 197 200 207 209 213 215 217
218 226 234 245 249 250 251 253 254 255 257 266 275 276 281 284 288 293 303 305 315 319 322 324 329
336 347 349 351 352 355 356 358 360 365 367 368 371 376 377 378 381 382 407 409 410 411 412 413 418
421 424 425 426 437 442 444 447 448 452 459 461 462 466 474 475 477 478 480 484 488 491;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 1 5 17 22 26 31 34 39 44 46 58 65 71 73 79 80 85 92 93 95 98 100 101 107 114 119 127 131
137 138 139 142 145 146 150 151 152 155 161 162 167 170 175 176 177 188 191 197 198 211 212 217 220
228 232 234 236 238 239 241 242 243 245 252 257 268 269 270 271 281 283 287 300 305 308 309 311 318
323 324 326 333 334 335 342 345 349 350 351 361 363 367 368 371 372 374 384 388 395 406 408 410 425
426 427 428 430 432 434 435 439 443 445 447 452 455 457 461 465 467 470 474 483 485 490;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 2 3 9 20 22 24 25 26 28 30 32 34 38 43 46 49 58 60 66 68 70 72 75 87 88 91 97 99 103 105
109 112 115 118 123 139 140 144 146 152 160 162 165 168 180 184 194 197 203 204 207 214 217 220 222
223 225 233 240 244 249 250 251 257 260 262 270 275 280 286 290 291 292 297 298 301 307 310 313 314
318 321 322 326 334 336 338 339 343 347 357 369 372 373 378 380 381 389 395 396 397 401 403 404 407
410 413 425 430 447 454 455 460 461 464 472 478 482 483 487 488 489 492 494 498;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 2 3 6 12 25 29 30 37 50 56 60 61 63 64 67 68 69 70 75 79 85 87 95 99 102 103 109 110 120
123 125 127 129 131 142 143 145 157 164 168 170 172 175 176 180 190 191 194 195 196 197 201 202 204
205 206 211 214 216 227 230 235 236 240 241 246 250 252 255 257 263 266 272 281 289 304 307 308 309
310 327 328 329 336 338 339 342 350 353 354 356 357 358 368 380 382 387 389 397 400 405 407 408 410
413 418 421 422 423 424 440 444 446 448 451 452 455 457 463 468 470 476 480 485 495;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
;
weights 2 : 5 6 19 24 27 31 32 33 34 35 39 40 43 49 58 59 61 66 73 89 90 92 94 97 98 113 116 118 121
126 127 131 140 141 144 148 164 165 169 170 173 175 176 177 179 181 183 185 191 194 202 203 204 205
207 211 213 215 218 220 233 234 238 245 252 254 259 260 265 269 272 278 280 286 287 288 292 295 297
302 304 306 310 313 315 317 319 328 331 334 335 336 342 346 347 351 357 362 365 377 384 388 389 390
403 411 419 426 438 440 443 444 446 447 455 460 463 464 467 486 491 493 494 499 500;
hsearch start=current swap=TBR
;
weights 1:all;
hsearch start=current swap=TBR
```

Repeated text must continue. However, it is unclear when to stop.

PSODA Likelihood Ratchet

```
BEGIN PAUP;

  begin randomReweight
    numChars = getWeightsLength();
    numWeights = numChars / percent;

    j = 0;
    while (j < numweights)
      weight = random(max = range);
      col = random(max = numChars) + 1;
      weights weight:col;
      j++;
    endwhile;
  end;

  set maxtrees = 1 nreps = 5;
  set criterion=parsimony;
  hsearch start = stepwise swap = tbr;
  range = 3;

  while (true)
    randomReweight(percent = 10);
    set criterion=parsimony;
    hsearch start = current swap = tbr;

    weights reset;

    set criterion=likelihood;
    hsearch start = current swap = tbr;
  endwhile;

end;
```

http://dna.cs.byu.edu/psoda

Computational discovery of composite motifs in DNA

Geir Kjetil Sandve[1,4], Osman Abul[2] and Finn Drabløs[3]

1 Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
2 Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey
3 Department of Cancer Research and Molecular Medicine, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
4 Department of Informatics, University of Oslo, Norway

E-mail Presenting Author:	finn.drablos@ntnu.no
Project URL:	http://tare.medisin.ntnu.no/
Code URL:	http://tare.medisin.ntnu.no/compo/
Open Source License:	GPL

Computational discovery of motifs in biomolecular sequences is an established field, with applications both in the discovery of functional sites in proteins and regulatory sites in DNA. In recent years there has been increased attention towards the discovery of composite motifs, typically occurring in cis-regulatory regions of genes. Single motif discovery of transcription factor binding sites has been shown to generate a large number of false positive predictions [1]. Searching for composite motifs is interesting because it can give a more realistic prediction of regulatory binding sites by filtering out some of the false positives.

This presentation describes Compo: a discrete approach to composite motif discovery that supports richer modelling of composite motifs and a more realistic background model compared to previous methods [2]. Furthermore, multiple parameter and threshold settings are tested automatically, and the most interesting motifs across settings are selected. This avoids reliance on single hard thresholds, which has been a weakness of previous discrete methods. Comparison of motifs across parameter settings is made possible by the use of p-values as a general significance measure. Compo can either return an ordered list of motifs, ranked according to the general significance measure, or a Pareto front corresponding to a multi-objective evaluation on sensitivity, specificity and spatial clustering.

Compo performs very competitively compared to several existing methods on a collection of benchmark data sets. These benchmarks include a recently published, large benchmark suite [3] where the use of support across sequences allows Compo to correctly identify binding sites even when the relevant position weight matrices (PWMs) are mixed with a large number of noise matrices. Furthermore, the possibility of parameter-free running offers high usability, the support for multi-objective evaluation allows a rich view of potential regulators, and the discrete model allows flexibility in modelling and interpretation of motifs.

1. Tompa M, Li N, Bailey TL, Church GM, De Moor B, Eskin E, Favorov AV, Frith MC, Fu Y, Kent WJ *et al*: **Assessing computational tools for the discovery of transcription factor binding sites**. *Nat Biotechnol* 2005, **23**(1):137-144.
2. Sandve GK, Abul O, Drablos F: **Compo: composite motif discovery using discrete models**. *BMC Bioinformatics* 2008, **9**:527.
3. Klepper K, Sandve GK, Abul O, Johansen J, Drablos F: **Assessment of composite motif discovery methods**. *BMC Bioinformatics* 2008, **9**:123.

SEEDER: PERL MODULES FOR CIS-REGULATORY MOTIF DISCOVERY

François Fauteux[1,2], Mathieu Blanchette[2,3] and Martina V. Strömvik[1,**2**]

[1]Department of Plant Science, McGill University, Sainte-Anne-de-Bellevue, Canada, H9X 3V9
[2]McGill Centre for Bioinformatics, McGill University, Montreal, Canada, H3A 2B4
[3]School of Computer Science, McGill University, Montreal, Canada, H3A 2A7

The computational identification of *cis*-regulatory elements in DNA sequences is a notoriously difficult problem in contemporary bioinformatics. *Cis*-regulatory motifs are typically short (6-15 bp), somewhat degenerate and often hard to distinguish from the background sequences.

We have recently released the Seeder 0.01 suite of Perl modules implementing an exact, discriminative seeding algorithm designed for fast and reliable DNA motif discovery in the promoter sequences of co-regulated, functionally related or homologous eukaryotic genes. The algorithm outperforms popular motif discovery tools on biological benchmark data.

The Seeder algorithm uses an enumerative approach and an objective function based on the probability of the sum of Hamming Distances between words (combinations of nucleotide symbols) and best matching subsequences (substring minimal distance, SMD), given a word-specific background probability distribution. This computation is accelerated by using the SMD index, a data structure allowing an efficient lookup, in a given sequence, for a subsequence minimally distant to a given word.

We present an application of the algorithm to the identification of plant-family-specific, tissue-specific *cis*-regulatory promoter motifs. DNA motifs discovered in the promoter sequence of tissue-specific genes are highly similar to experimentally characterized plant *cis*-regulatory elements.

We also present a web server for motif discovery using the Seeder algorithm, with pre-computed background for six plant species. Various tools are also accessible on the Seeder website for post-processing operations including imaging, PWM scoring and database matching.

Reference: Fauteux F, Blanchette M, Stromvik MV: Seeder: Discriminative Seeding DNA Motif Discovery. Bioinformatics 2008, 24(20):2303-2307.

Seeder website: http://seeder.agrenv.mcgill.ca

Seeder 0.01 distribution: http://search.cpan.org/~ffauteux/Seeder-0.01/ (released under the terms of the Artistic License)

E-mail address of the presenting author: francois.fauteux2@mail.mcgill.ca

Large-scale gene regulatory motif discovery and categorisation with NestedMICA

Matias Piipari [1], Thomas Down [2], Tim Hubbard [1]

Background: Our understanding of DNA specificities of transcription factors is largely recorded in databases containing DNA sequence motifs (position frequence or weight matrices). Previously it has been shown that there are familial tendencies in these DNA sequence motifs that are predictive of the family of factors that binds them and indeed motifs belonging to different transcription factor families have been studied using unsupervised and supervised machine learning methods in an attempt to predict the binding domain for sequence motifs and to sensitively find motifs from novel sequence sets that fit these tendencies. However, a natural probabilistic model for recurring patterns in a set of sequence motifs is still an open research problem. Sequence motif discovery algorithms also often present the problem of reporting duplicate or closely related motifs multiple times. There is a clear need for a mathematical framework and tools for computational biologists used for profiling relatedness of gene regulatory sequence motifs.

Results: We propose a generative model for nucleotide sequence weight matrices termed the 'metamotif'. This model can be used to summarise recurring patterns in a set of weight matrices. A nested sampling based algorithm for parameter estimation of metamotifs from a set of motifs, as well as two practical uses for the model will be discussed: a motif classification task, as well as use as a weight matrix prior in a Bayesian model discovery algorithm.

Conclusions: The metamotif model is successfully applied to a weight matrix classification problem where sequence motif features are used to predict the type of a sequence motif on the level of its TRANSFAC family and superfamily. We also show that metamotifs can be applied as informative priors in a motif discovery algorithm to dramatically increase the sensitivity to discover motifs. Both the transcription factor type prediction tool and the informative prior are also made available for the use use of computational biologists through a web server and a new release of the NestedMICA motif discovery tool.

Project URL: `http://www.sanger.ac.uk/Software/analysis/nmica/`
SVN repository: `http://www.derkholm.net/svn/repos/nmica/`
License: LGPL

Affiliations:

1) Wellcome Trust Sanger Institute, Hinxton, Cambridgeshire (UK)
2) Wellcome Trust/Cancer Research UK Gurdon Institute, Cambridge, Cambridgeshire (UK)

R'MES: finding exceptional motifs in sequences

Mark Hoebeke and Sophie Schbath

INRA, Unité Mathématique, Informatique et Génome, 78352 Jouy-en-Josas, France.
Mark.Hoebeke@jouy.inra.fr, Sophie.Schbath@jouy.inra.fr.
R'MES home page: http://migale.jouy.inra.fr/outils/mig/rmes
URL for accessing the code: https://mulcyber.toulouse.inra.fr/projects/rmes/
License: GNU General Public License

The R'MES project started in 1995. This is now the 3rd version. The main question R'MES addresses is "does this motif occur in that biological sequence with an expected frequency?" In other words, can we observe it so many times, or so few times, just by chance? Usually, when the answer is no, such a motif is a candidate to have a particular biological meaning. To do so, we calculate an exceptionality score for each word of a given length (or for each given set of words); this score is a one-to-one transformation of the corresponding p-value. The p-value is the probability that a random sequence having the same 1- up to $(m + 1)$-letter word composition as the biological sequence contains as many occurrences of the given word. This probability is approximated thanks to rigorous statistical approximations of the word count distribution, namely either a Gaussian distribution (for frequent words) or a compound Poisson distribution (for rare words). Details about the statistical results on word counts in random sequences can be found in [1]. R'MES is getting enriched thanks to novel questions from the biologists. R'MES can now for instance compute an exceptionality score related to the skew of an oligonucleotide; the typical question is indeed "does this motif occur significantly more often on the leading strand than on the lagging strand?" At the moment, we are implementing the statistical tests proposed by [2] to compare motif exceptionalities between two different sequences. In the talk, we will illustrate how we have identified the Chi site of *Staphylococcus aureus* [3] and the matS site of *Escherichia coli* [4] thanks to R'MES.

[1] Robin, S., Rodolphe, F. and Schbath, S. (2005). *DNA, Words and Models.* Cambridge University Press.

[2] Robin, S., Schbath, S. and Vandewalle, V. (2007). Statistical tests to compare motif count exceptionalities. *BMC Bioinformatics.* 8:84 1–20.

[3] Halpern, D., Chiapello, H., Schbath, S., Robin, S., Hennequet-Antier, C., Gruss, A. and El Karoui, M. (2007). Identification of DNA motifs implicated in maintenance of bacterial core genomes by predictive modelling. *PLoS Genetics.* 3(9) e153.

[4] Mercier, R., Petit, M.-A., Schbath, S., Robin, S., El Karoui, M., Boccard, F. and Espeli, O. (2008). The MatP/matS site specific system organizes the Terminus region of the E. coli chromosome into a Macrodomain. *Cell*, volume 135, Issue 3, 475–485.

Title: "Open Source Implementation of Batch-Extraction for Coding and Non-coding Sequences"

Authors: Jens Lichtenberg, Lonnie Welch

Affiliations: School of EECS, Ohio University, Athens, Ohio 45701, USA, lichtenj@ohio.edu

URL for project: http://opensource.msseeker.org

URL for code: http://extractor.msseeker.org

Open Source License being used: The GNU General Public License (GPL)

Abstract

Regulatory genomics is focused on the discovery of general characteristics explaining the co-regulation of genes. Most applications used to discover co-regulation, return lists of putative co-regulated genes and their respective expression values. In order to determine shared regulatory elements it is necessary to analyze the non-coding and in some cases coding sequences of these genes.

While it is possible to extract promoter sequences, exons, introns, intergenic regions and untranslated regions for single genes or entire genomes through tools like the UCSC Genome Browser or directly through databases like Ensembl, it is difficult to extract these elements in an automated fashion for a batch of gene symbols.

Based on the available Ensembl API a Perl based open source solution to extract non-coding as well as coding sequences is presented. The tool reads in a list of gene symbols and extracts exons, introns, 5' and 3' untranslated regions, coding sequences and promoters (of variable length) for each gene in the list. The sequences are stored in fasta files separated based on the segment characteristic of the extracted sequence (e.g. 1 fasta file for all exons of all supplied gene symbols).

Due to limitations of the annotations within Ensembl it is sometimes not possible to extract the entire set of annotated sequences. In such cases the user is notified.

This application is integrated into an existing enumerative motif discovery framework (WordSeeker) in order to extract the desired sequences for a gene list automatically and supply it to the subsequent motif discovery phase. Due to its open source nature it is also possible to integrate this tool into other existing motif discovery frameworks (enumerative and alignment-based), sequence analysis frameworks, or as a post-processing stage in microarray, ChIP-chip or proteomics experiments.

To enhance the visibility of the tool, it will be published to the CPAN and BioPerl repositories in the near future.

An Open Source Framework for Bioinformatics Word Enumeration and Scoring

Kyle Kurz, Jens Lichtenberg, Lee Nau, Dr. Frank Drews, Dr. Lonnie Welch
Ohio University, welch@ohio.edu

Main Project Page: http://bio-s1.cs.ohiou.edu/~wordseek
Download Page: http://bio-s1.cs.ohiou.edu/~wordseek/download

GNU General Public License (GPLv3)

The software package presented here provides an open source framework for word enumeration (and subsequent scoring of those words) within biological sequence data. Using this framework, developers may choose to implement any of a number of algorithms to perform the enumeration, with no change to the execution logic of the framework.

Two major problems are addressed with our framework, one in the field of biology and one in computer science. Biological research creates enormous amounts of genomic data for each experiment performed. From there, the biologists must usually select a subset of the "words" (short DNA nucleotide subsequences) for further analysis. Bioinformatics provides computerized tools to aid biologists in the pruning process by providing information about statistically interesting words, under the assumption that over and under-represented words should provide some real biological function.

Through the creation of a modular framework, different algorithms can be used to accommodate the requirements of a specific job. Our framework allows a highly scalable software package, as it does not have the limitations of being tied to a single enumeration or scoring algorithm, and the best algorithm for a dataset can be selected at runtime. The implementation of multiple algorithms allows the user to focus on the job-specific optimizations such as high speed or low memory footprint. This flexibility is often not possible with single algorithm tools. By building a minimal set of requirements for functionality based on our WordSeeker [1] tool and analysis of other enumerative tools such as Weeder[2] and YMF[3], we have been able to provide a highly abstract system with interchangeable modules for the various stages, allowing the framework to grow and mature as new algorithms and methods are developed.

Using object-oriented software design and class abstraction, our framework is not only extensible, but easily modifiable. The base classes form a frame around two major phases, the enumeration of words and the scoring of those words. Built in C++, the framework utilizes virtual functions to provide consistent interfacing between components, regardless of the underlying algorithms. Well documented abstract class definitions provide a description of the minimal set of functions a developer must implement, as well as allowing the extension of algorithm specific functionality through standard interfaces. Future work will provide additional post-processing functionality through similar abstract interfaces.

In summary, our framework streamlines the development process for new techniques and modules in a general bioinformatics toolkit and facilitates the research process by allowing the selection of various implementations based on the biologists' needs for a given dataset.

1. J. Lichtenberg, M. Alam, T. Bitterman, F. Drews, K. Ecker, L. Elnitski, S. Evans, E. Grotewold, D. Gu, E. Jacox, K. Kurz, S. S. Lee, X. Liang, P. M. Majmudar, P. Morris, C. Nelson, E. Stockinger, J. D. Welch, S. Wyatt, A. Yilmaz, and L. R. Welch, "Construction of Genomic Regulatory Encyclopedias: Strategies and Case Studies," *Proceedings of the Ohio Collaborative Conference on Bioinformatics*, IEEE Computer Society press, June 2009

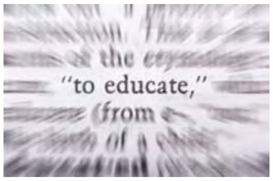
2. G. Pavesi, P. Mereghetti, G. Pesole, *Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes*. Nucleic Acids Research 2004 Jul 1;32(Web Server issue): W199-W203.

3. S. Sinha and M. Tompa. *YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation.* Nucleic Acids Research 2003 Vol. 31, No. 12 3586-3588

The *WORDIFIER* Pattern for Functional and Regulatory Genomics

Lonnie R. Welch, Jens Lichtenberg, and Frank Drews
Bioinformatics Laboratory, Ohio University, Athens, Ohio 45701, {welch|lichtenj}@ohio.edu

According to Christopher Alexander, the father of the 'design patterns' movement, a pattern "describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use the solution a million times over, without ever doing it the same way twice [1]." This paper use Alexander's pattern format to present *WORDIFIER*, a design pattern from the domain of functional and regulatory genomics.



With the availability of the genomic sequences of numerous organisms, life scientists are working in conjunction with bioinformaticians to decipher the meanings of the genomes. Projects such as Encyclopedia of Genomic Elements (ENCODE) [2] seek to identify and charatcetrize the functional elements in genomes. The functional elements are often referred to as *words*.

♦ ♦ ♦

Given a genomic sequence (or a set of sequences), an important problem is the enumeration of all subsequences (words) contained in the sequence (or the set of sequences).

The enumeration of all words in a sequence (or a set of sequences) is a fundamental operation in the study of genomic sequences. The enumerated words provide the basic elements that are further characterized and studied. Thus, the bioinformatics open source community has provided solutions to this problem.

To enumerate the words in a sequence, the sequence is decomposed into subsequences and the set of unique subsequences (words) is constructed. The number of instances of each unique word, the word frequency, is counted.

If multiple sequences are analyzed, the process is repeated for each sequence and the union of sets of unique words is constructed. The number of sequences containing each word, the word sequence frequency, is counted.

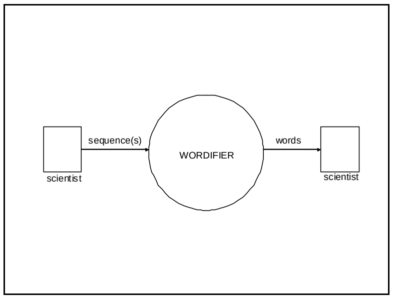


Figure 1. WORDIFIER solution diagram.

♦ ♦ ♦

The WORDIFIER makes use of the SEQUENCE pattern and the WORD pattern. Many operations are performed to characterize the product of WORDIFIER, such as SCORING words, ORDERING word sets, and EXTRACTING subsets of words based on various criteria.

Future work will include the use of POSA and GOF pattern formats to provide a more detailed description of the WORDIFIER pattern. Additionally, other patterns from the domain of regulatory genomics will be codified. A pattern language that unifies the set of patterns will also be developed.

REFERENCES

[1] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.
[2] E. Birney, Stamatoyannopoulos J. A., Dutta A., Guigo R., Gingeras T. R., Margulies E. H., Weng Z., Snyder M., Dermitzakis E. T., Thurman, R. E., et al., "Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project," Nature **447**:799–816, 2007.

Robust architectural patterns for bioinformatics: experiences with Chipster

Aleksi Kallio, Taavi Hupponen, Petri Klemelä, Jarno Tuimala, Eija Korpelainen
CSC - IT Center for Science Ltd. (contact address: aleksi.kallio@csc.fi)

Open source project homepage: http://chipster.sourceforge.net
Source code: http://chipster.sourceforge.net/sources.shtml
License: GPL version 3 or later.

Chipster (http://chipster.csc.fi) is data analysis platform which offers an intuitive graphical user interface to a comprehensive collection of DNA microarray data analysis methods, such as those developed in the R/Bioconductor project. As Chipster was developed for large scale, national use, the concerns for usable, robust and dependable technology had to be addressed carefully.

Based on our previous experience from developing and maintaining distributed systems we decided to opt for component based asynchronous architecture as a base paradigm. The Chipster environment consists of components that use message oriented middleware (Java JMS) for communication. The most important characteristic of the Chipster architecture is how distributed state is managed. Instead of complex patterns such as 2-phase commits we have designed components to be independent and to employ best effort service. This means that there are no hard state dependencies between the components and, for example, they can be restarted independently, making the system very adaptable to changes in the physical setup. Chipster allows compute node configuration to be changed at runtime and we have extended it to allow also data brokers to be configured in the same way. Hardware problems, smaller software problems and configuration changes can be taken care without affecting connected users.

For improving the perceived performance of the system, the most important step is to closely monitor performance. The message oriented middleware layer is used to communicate status messages and they are collected to a centralized database. We have implemented a manager tool that enables real time inspection of the status of the system and. By integration into Nagios monitoring platform we get real time alerts at system wide shortages.

Finally, one can never make the system bullet proof. However, robustness can be improved by putting effort into error management. For example, it is important to produce understandable information for users when the system not completely responsive. The benefit of a thick client architecture is that we can implement error management on the client layer, so that network or server side lags do not directly hinder user experience. Lately we have concentrated on improving client side robustness, for example by isolating visualization processes that can consume large amounts of CPU resources.

Taken together, after 2 years of public service on top of the Chipster platform we feel that our efforts with robust design and implementation have paid off. We have been able to concentrate on further development instead of nurturing an ill-behaving service. We feel that focusing on robust architectural patterns and their sharing will be beneficial for bioinformatics software developers.