GALAXY, A WEB-BASED FRAMEWORK
FOR THE INTEGRATION OF GENOME ANALYSIS

A Dissertation in
Biochemistry, Microbiology and Molecular Biology
by
Daniel James Blankenberg

© 2009 Daniel James Blankenberg

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2009
The dissertation of Daniel James Blankenberg was reviewed and approved* by the following:

Anton Nekrutenko
Associate Professor of Biochemistry and Molecular Biology
Dissertation Adviser
Chair of Committee

Webb C. Miller
Professor of Biology
Professor of Computer Science and Engineering

Ross C. Hardison
T. Ming Chu Professor of Biochemistry and Molecular Biology

Stephan Schuster
Professor of Biochemistry and Molecular Biology

Andrey Krasilnikov
Assistant Professor of Biochemistry and Molecular Biology

Scott Selleck
Professor of Biochemistry and Molecular Biology
Head of the Department of Biochemistry and Molecular Biology

*Signatures are on file in the Graduate School.
ABSTRACT

The standardization and sharing of data and tools are among the biggest challenges facing large collaborative projects and small individual labs alike. Here a compact web application, Galaxy, is described which effectively addresses these issues. It provides an intuitive interface for the deposition and access of data and features a vast number of analysis tools including operations on genomic intervals, utilities for manipulation of multiple sequence alignments and molecular evolution algorithms. By providing a direct link between data and analysis tools, Galaxy allows addressing biological questions that are beyond the reach of existing software. Available both as (1) a publicly available web service providing tools for the analysis of genomic, comparative genomic and functional genomic data and (2) a downloadable package that can be deployed in individual labs, Galaxy attempts to serve both sides of the user distribution: experimental biologists and bioinformaticians.

For experimental biologists, it provides an intuitive interface for data deposition and access, features a large number of tools and makes analysis transparent by documenting every step in the Galaxy history system. Most importantly, it streamlines the path from data to analysis, as even complex tools can be applied to genomic data directly without manual parsing or preprocessing.

For bioinformaticians, Galaxy is a software system that provides informatics support through a platform that gives biologists simple interfaces to powerful tools, while automatically managing the computational details. Galaxy provides a framework that can integrate command-line tools with almost no effort. For each tool, Galaxy generates the interface and provides all computational housekeeping.

A prime example of a remarkable disconnect between genomic data and analysis tools is in the case of multiple-species whole genome alignments. Continuously expanding collections of freely downloadable multiple-species whole genome alignments have been made available to the scientific community, however, several issues exist which prevent experimental biologists from utilizing these important datasets. Simply put, these alignments are not only large enough to cause significant logistical problems just to download and store, but there are no tools available that allow command-line averse biologists to manipulate these alignments. Furthermore, current genome analysis packages, such as the phylogenetic software HyPhy, do not accept the Multiple Alignment Format (MAF) as input. A set of tools designed to address these challenges has been integrated into the Galaxy framework and is included as part of the standard software distribution. Short examples of tool usage as well as an in-depth sample analysis are presented along with descriptions of the individual tools. The step-by-step
sample analysis and toolset integration provide real-life examples of the utility of Galaxy both as (1) an effective and intuitive analysis platform for experimental biologists and (2) a tool and data source integration framework for bioinformaticians.
# TABLE OF CONTENTS

List of Figures..............................................................................................................................................vi

List of Tables....................................................................................................................................................vii

Acknowledgments..............................................................................................................................................viii

Chapter 1: Introduction......................................................................................................................................1

Chapter 2: A framework for collaborative analysis of ENCODE data: making large-scale analyses biologist-friendly......................................................................................................................................10

Chapter 3: Galaxy, a web-based genome analysis tool for experimentalists......................................................25

Chapter 4: Making whole genome alignments usable for biologists.................................................................59

Chapter 5: A genome-wide scan for synonymous SNPs affecting protein function............................................78

Appendix A: Supplemental Material for Chapter 2: A framework for collaborative analysis of ENCODE data: making large-scale analyses biologist-friendly..................................................................................89

Appendix B: Supplemental Material for Chapter 3. Galaxy, a web-based genome analysis tool for experimentalists........................................................................................................................................97

Appendix C: Supplemental Material for Chapter 4. Making whole genome alignments usable for biologists..................................................................................................................................................98

Appendix D: Supplemental Material for Chapter 5. A genome-wide scan for synonymous SNPs affecting protein function........................................................................................................................................109

v
LIST OF FIGURES

2.1 Genomic Interval Operations ........................................................................................................ 17
2.2 Types of Non-GENCODE ESTs .................................................................................................. 18
2.3 Steps in identification of Non-GENCODE ESTs .......................................................................... 19
3.1 Galaxy’s Analyze Data interface ................................................................................................. 49
3.2 Changing properties of a dataset in Galaxy .................................................................................. 50
3.3 Sending data to Galaxy from UCSC Table browser ...................................................................... 51
3.4 Get Flanks tool interface ............................................................................................................ 52
3.5 Genomic intervals Join tool interface ......................................................................................... 53
3.6 Build custom tracks tool interface and display at UCSC Genome browser ................................ 54
3.7 Dataset containing exons and overlapping SNPs ....................................................................... 55
3.8 Creating a Workflow from an existing History ........................................................................... 56
3.9 The Workflow editor .................................................................................................................. 57
3.10 Extract MAF blocks and Stitch MAF blocks tool interface ........................................................ 58
4.1 A typical example of a set of MAF blocks ................................................................................... 71
4.2 Extract MAF blocks tool ............................................................................................................. 72
4.3 MAF to FASTA converting tool ................................................................................................ 73
4.4 Stitch MAF blocks tool .............................................................................................................. 74
4.5 Filter MAF blocks by Species tool ............................................................................................. 75
4.6 Join MAF blocks by Species tool ............................................................................................... 76
4.7 Filter MAF blocks by Size tool .................................................................................................. 77
5.1 Distribution of \( \delta \) phyloP for 2,603 synonymous isoleucine wobble positions ....................... 82
A.1 Galaxy Interface ......................................................................................................................... 93
A.2 Genomic vicinity of intergenic Non-Gencode EST DR731323 ............................................... 94
A.3 Genomic vicinity of intergenic EST DB275065 .......................................................................... 96
# LIST OF TABLES

2.1  Descriptive Statistics for the three categories of Non-GENCODE ESTs .......................... 21
2.2  Overlap among Non-GENCODE EST exons, Affymetrix transfrags, and random intervals.. 21
2.3  Nucleotide substitution analysis of ENCODE ancestral repeats using HyPhy wrapper....... 22
A.1  Candidate list of synonymous isoleucine polymorphisms........................................ 109
ACKNOWLEDGMENTS

First, I thank my committee members and all of the teachers I have had thus far; foremost, I would like to specifically thank my advisor, Dr. Anton Nekrutenko –to whom I am dearly indebted, for all of his guidance and patience –without which, none of this work would be possible.

Second, I thank all my coauthors for their contributions to the papers included within this text as well as all of the members of the Galaxy development team; it is their tremendous efforts that allow Galaxy to be the successful genome analysis platform that it is today.

Third, I thank the entire Galaxy user base, particularly members of the Center for Comparative Genomics and Bioinformatics at Penn State, for their continued support of the Galaxy framework and public server through usage, comments, suggestions and bug reports.

Next, I thank everyone who makes data and analysis tools freely available to the scientific community; it is this free exchange of knowledge and resources which makes modern scientific investigation possible.

Finally, I thank my family and friends, especially Rachel, for the things they do.

Additional acknowledgements can be found inline within the individual chapters.
CHAPTER 1

INTRODUCTION

Recent development of several high-throughput experimental techniques, including microarrays and next-generation sequencing has led to a massive expansion in the quantity of available biological data. Much of this data is readily and freely accessible to all of the general public. However, for most experimental biologists there exists a void between accessing this wealth of information and translating it into useful biological knowledge. The first problem that biologists have to cope with is the immense size of genomic datasets. These datasets often comprise entire genomes worth of information: some contain information on specific genomic elements, such as the genome wide locations of a particular human transcription factor binding site, whereas other datasets, such as multiple-species whole genome alignments, can house information about several different organisms. Some of these datasets, such as the previously mentioned alignments, can easily occupy hundreds of gigabytes, causing many of these datasets, despite being freely and readily available, to go underutilized by the experimental community simply due to logistical issues related to storing massive quantities of information. Even if initial obstacles can be overcome, experimental biologists are left with few options to manipulate these data. Modern spreadsheet applications, for example, are not capable of loading a file containing all known human polymorphisms. Another problem that is encountered is the issue of data integration and format incompatibility. Beyond simply having different types of data such as sequences, alignments and genomic intervals, there is a seemingly endless supply of data formats for each of these different datatypes. This often leads to the creation of custom “one-off” scripts. These small scripts are generally developed by individual labs and might only perform simple functions such as pre-parsing a file, but the reliance on these sorts of scripts has led to the all-too-often inclusion of phrases such as “the data was transformed using a series of in-house Perl scripts.” These scripts may be simple, but when not readily available they prove to be a real hindrance to the reproducibility of research. In cases when preprocessing scripts are available, bioinformatic tools often come with confusing or command-line only interfaces. All of these interfaces are different and they are not designed to work together: rarely is it the case that the output of one tool can be fed directly as input into another tool. Furthermore, there are almost too many tools, it is hard for experimental biologists to know where to start or which tools are best suited for a particular analysis. These issues effectively prevent many biologists from utilizing
existing genome analysis software. The goal of the work discussed in this text is to address these issues. Available both as (1) a publicly available web service providing tools for the analysis of genomic, comparative genomic and functional genomic data and (2) a downloadable package that can be deployed in individual labs, Galaxy attempts to serve both sides of the user distribution: experimental biologists and bioinformaticians.

Galaxy is not simply about accessing data. FlyMine (Lyne, 2007), BioMart (Smedley, 2009), the UCSC Table browser (Karolchik, 2004) and many other projects offer access to genomic data, and they do it well. Galaxy is not meant as a replacement to data warehouses as the organizations that focus on this problem are able to more affectively address the issues of storing and querying their particular data schemas. Instead, Galaxy provides seamless integration of the native data warehouse interfaces where users are able to directly interact with the data in the fashion that the data resource designers and curators have intended, but instead of being forced to download potentially gigabytes of data, users can simply “send results to Galaxy.” Once data has been accessed by a user and placed into their history, it is immediately ready for analysis. Galaxy contains over a hundred analysis tools, with a concentration on providing tools that the genomics community has established as the “best of”, greatly reducing the struggle to find the proper tools for a particular analysis. Galaxy is able to automatically determine data formats and the data can only be used as input for bioinformatic tools that are able to accept a particular data’s format as input. In cases when the data is of the proper kind (e.g. an alignment) but the tool only accepts a particular format (e.g. a tool requires FASTA format, but the user’s data is in the MAF format), Galaxy has a collection of implicit datatype converters that handle converting the data into the format required by the tool without requiring any additional intervention by the user. By precluding the need for one-off scripts, Galaxy allows truly reproducible analysis. Furthermore, Galaxy allows users to not only share and publish data and results (data libraries and Galaxy user pages), but also entire analysis steps (a user’s history) as well as completely customizable plug-and-play analysis pipelines (workflows). Usability is a primary concern with the Galaxy framework causing the adaptation of various best of user design practices (e.g. Wroblewski and Rantanen, 2001; http://www.lukew.com/resources/articles/web_forms.html) with an emphasis on iterative interface design based upon user feedback.

Galaxy is not the first or newest application to be developed to address the problems facing experimental biologists and bioinformaticians. The attempts by several other groups to address the issues solved by Galaxy have met with varying success and some are still thriving while others have ceased development and have fallen by the wayside. As Galaxy is an ongoing software project and is
undergoing continual improvement and enhancement by an expanding group of developers, an examination of resources from both of these cases can allow Galaxy to avoid some of the pitfalls that have hindered these other projects. A few of these other notable applications include the Biology Workbench (Subramanian 1998), BioExtract Server (Lushbough, 2008) and GenePattern (Reich, 2006).

The Biology Workbench was among the first attempts at addressing the issue of database and tool integration. Like Galaxy, this web-based resource does provide a single interface for accessing database contents and running bioinformatic tools, however it suffers from several shortfalls. The data available in the Biology Workbench interface needs to be manually added to a database local to the server, there are no mechanisms allowing for the software to directly interact with primary data resources. Along with causing a significant bottleneck in providing users with the newest data available, this approach requires locally storing large quantities of data that may only be rarely or partially utilized. For example, the Biology Workbench website (http://workbench.sdsc.edu/update.html) states, several times, that entire databases had to be removed for disk space concerns. When these datasets are removed, analyses that had been previously possible are no longer possible, which is a great hindrance to reproducibility. Because Galaxy interacts directly with today’s most popular data sources, it does not generally suffer this issue. The exception, in Galaxy’s case, is the local caching of multiple-species whole genome alignments and assembled genomic DNA sequences (for use in the extract genomic DNA tool), but adding additional data is a streamlined process involving sets of flat files, which can be encoded or compressed for the sake of space and speed of access. The datasets are currently added by request and efforts to automate the process are on the horizon, but once they are made available to Galaxy they are never removed – reproducibility is paramount. Additionally, reliance on these locally cached data is not required, because, for example, the alignment extractors can operate directly on data in a user’s history.

A second issue with the Biology Workbench is that is entirely sequence based, and although much of Galaxy’s current functionality does operate on sequences, the framework is not restricted to any type of data. Datatypes can be any type of binary or text file and they can be composed of a single file or multiple files (known to Galaxy as “composite datatypes”). Another issue with the Biology Workbench is that it was not designed to handle genome sized datasets and server errors are common when attempting to analyze sequences larger than 100,000 nucleotides (http://workbench.sdsc.edu/faq.html) – Galaxy is unaffected by large sequence lengths and the only restrictions are due to issues with the underlying bioinformatic tools themselves or restrictions of the
platform it is installed upon (hardware, OS, file system, etc), not the framework. The primary issue with the Biology Workbench is its failure to adapt; while the operator of this resource has added databases and tools to meet requests of users, the framework itself was not updated to meet the modern demands of genome analysis. Causes of its failure to adapt are varied, but among them are difficulty of obtaining source code and allowing outside contributions. Galaxy does not suffer from these issues because it has been available in a publically downloadable version control system (first Subversion and now Mercurial) from the beginning. In fact, the change to Mercurial was prompted by the increase in ease of which “patches” can be generated and submitted from outside contributors to the core development team for review and possible inclusion in the software distribution. Galaxy is a community project and its development is fueled and aided by the suggestions and contributions of the genomics community at large. Furthermore, the Galaxy project has adopted a philosophy of agile software development including biweekly developer meetings, independent code reviews, short release cycles, fully functional and richly featured software favored over extensive documentation, a dynamic user driven development plan and a highly coherent and efficient core development team. Despite its downfalls, the Biology Workbench has had a lasting impact, spawning several other projects.

A second project worth discussing is the BioExtract Server (Lushbough, 2008), which allows users to query online sequence data, analyze it using bioinformatic tools and to create, save and share custom workflows. Upon first glance at these claims, this resource seems to echo the functionality of Galaxy and the BioExtract Server does in fact do all of these things, but has some notable issues. The first problem is that BioExtract requires JavaScript to be enabled for a user to have any functionality whatsoever and disabling JavaScript will present the user with empty boxes where none of the links nor buttons work; worse yet, this is not clearly indicated and it’s not even possible to access the help page. Galaxy does not have this pitfall as it is written using unobtrusive JavaScript with progressive enhancement. This allows Galaxy to have complete functionality when JavaScript is not enabled in a user’s browser, although, when enabled, JavaScript will provide a much cleaner and user-friendly interface. The single exception to this is Galaxy’s AJAX workflow editor, which allows interactive (drag and drop, point and click) building and modification of new or existing workflows. However, workflows can still be run, shared and created from an existing history when JavaScript is disabled. With JavaScript enabled, it is possible to use the BioExtract Server to query multiple datasources including GenBank databases which are local to the BioExtract Server and are updated monthly or every four months as well as data hosted via Web Services such as TrEMBL, SwissProt, UniProt and RefSeq. The single option available when querying is to compare a user provided string to a particular
field in a database, looking only for equality. Some databases support the use of wild-card characters, but the databases that allow this and the subset of modifiers for each database is not provided to the user. To enable searching across multiple databases, an administrator is required to define within the BioExtract server the searchable fields and how they link to each other. Galaxy is able to use the native interfaces for external databases, allowing much more complex querying than simply looking for string equality and also precluding the need for a local administrator to determine how databases should be linked. Once users have located sequence data in which they are interested (called “Extracts”), they can manually save their selections as “Saved Extracts” – forgetting this manual step will require the user to start all over again, should they end their session and come back later. These manual steps of saving are never required when using Galaxy, where every time a database is queried or a tool is run, a new history item is automatically created and saved. To run tools in BioExtract, the user selects a desired tool and then chooses input which they can upload directly from their computer, copy and paste or type in a box, the output from a previous tool run or they can use all the records found on the “Extracts” page. It is entirely up to the user to ensure that data is of the proper type (e.g. FASTA when using alignment tools such as ClustalW) as the BioExtract server has no internal knowledge of what sort of data is contained in each Extract or is required by a particular tool. Galaxy has an extensive set of easily expandable known datatypes, along with implicit datatype conversion facilities, which prevent a user from attempting to perform an analysis on improperly formatted data. One interesting aspect of the BioExtract Server is the ability for registered users to define their own tools. These tools can be external web services which use a standard web services API such as Soap or it can be an executable stored and run on the user’s own computer. Galaxy does not have the ability to allow users to add tools to the public Galaxy server and while this does initially seem like a really interesting feature, it becomes a great detriment to allowing reproducibility, as the versions of executables on an individual’s computer could differ greatly or even be different programs which share a common name such as “parseData.pl”. If a bioinformatician has a tool that they want to incorporate into Galaxy, they will need to run their own private server or submit the tool to the primary development team for integration into the public server. Running a private instance of Galaxy is encouraged, especially for bioinformaticians wishing to provide a set of tools to their research groups; it does not appear possible to obtain the source code in order for a research team to run their own private BioExtract Server (much of BioExtract Server’s functionality is proprietary whereas Galaxy is itself open source and the components which it utilizes are also open source). BioExtract Server allows the construction of complex workflows by recording the steps that a user takes to perform an analysis, similar to Galaxy’s ability to build a workflow from an existing history. However, in the BioExtract Server, users are
required to explicitly declare that they want to create a workflow; only then will the server start to track their actions. In Galaxy, this tracking happens organically: as the user interacts with tools and datasources their actions are recorded in their history, which can then be turned into a workflow with a single click. Another issue with the BioExtract server is the reporting of errors, due to either issues in the framework or with running a particular tool. For example, attempting to use the ClustalW tool with default parameters has “blosum” selected as the default matrix file, however this is not a valid matrix file on their system (“ERROR: Cannot find matrix file [blosum]” is reported in the output). The output appears exactly the same whether-or-not the tool completed successfully (i.e. the content differs, but the text output is always simple black on white text) and it is not clear how to report this error to the site maintainers (e.g. there is no email address listed for reporting bugs).

The final alternate platform to be discussed in this text is GenePattern. GenePattern is a well-designed web application, which, like Galaxy, appears to use JavaScript only for progressive enhancement. Two places where GenePattern excels relative to Galaxy are versioning support and an external API. Versioning support in Galaxy is still in its infancy and while, like GenePattern, tools have version information associated with them and this information is stored every time an analysis is performed, Galaxy does not currently have the ability to run older versions of tools, but this feature has been thoroughly considered and is on the agenda for implementation. GenePattern’s external API allows programmers to directly access a GenePattern server using JAVA, MATLAB and R programming languages, but in Galaxy this can currently only be accomplished by using one of the many generic cookie-aware URL libraries (similar to how Galaxy’s built-in functional test framework operates). In order to use the public GenePattern server users are required to register an account, whereas in the public Galaxy server non-registered users are able to use most features of the software and it is only the advanced features (e.g. permanently storing and sharing data, analyses and workflows) that require users to register an account. It should be noted that this is not a limitation of the software packages themselves, but the preference of the public service maintainers; the Galaxy and GenePattern platforms both allow administrators to configure whether-or-not to allow non-registered users access to the server within local installations. By requiring users to register (i.e. forced registration) before allowing them to try even the most basic features of the software, many potential users will be turned away for simple privacy concerns. It does, however, have the effect of artificially inflating the number of registered users, as a user must register just to see what the user interface looks like, even if that user never actually uses the software to run the most basic an analysis; another side effect of forced registration is users providing bogus or throwaway credentials (e.g. a mailinator.com email address), which will cause
them to register a second “real” account should they find the service useful. By allowing users access to the public Galaxy server without requiring them to register, users can run entire analyses and only register an account if they find the software to be acceptable for their needed purpose. Although GenePattern claims to be open-source, source code is only available by making a personal request via email, which effectively makes this project closed-source in practical purpose. Another issue relating to the development of GenePattern include a rather restrictive license, if one was able to obtain the source code and modify it, the developer is unable to share his work or modifications with the community or even allow access to their personal server to the community at large as this is strictly forbidden, unless prior approval is granted by the main developers. This is not an issue with Galaxy, which has a license that allows end-users to modify and distribute the original source as well as derivative works with the only requirement being the inclusion of the original copyright notice.

The remainder of this text is organized as follows. Chapters 2 and 3 address Galaxy as a web-based tool and data integration framework. For experimental biologists, it provides an intuitive interface for data deposition and access, features a large number of tools and makes analysis transparent by documenting every step in the Galaxy history system. Most importantly, it streamlines the path from data to analysis, as even complex tools can be applied to genomic data directly without manual parsing or preprocessing. For bioinformaticians, Galaxy is a software system that provides informatics support through a platform that gives biologists simple interfaces to powerful tools, while automatically managing the computational details. Galaxy provides a framework that can integrate command-line tools with almost no effort. For each tool, Galaxy generates the interface and provides all computational housekeeping.

Chapter 4 addresses the integration of a multiple-species whole genome alignment toolset into Galaxy. A prime example of a remarkable disconnect between genomic data and analysis tools is in the case of multiple-species whole genome alignments. Continuingly expanding collections of freely downloadable multiple-species whole genome alignments have been made available to the scientific community, however, several issues exist which prevent experimental biologists from utilizing these important datasets. Simply put, these alignments are not only large enough to cause significant logistical problems just to download and store, but there are no tools available that allow command-line averse biologists to manipulate these alignments. Furthermore, current genome analysis packages, such as the phylogenetic software PAML and HyPhy, do not accept the Multiple Alignment Format (MAF) as input. A set of tools designed to address these challenges has been integrated into the Galaxy framework and is included as part of the standard software distribution. This toolset allows
experimental biologists access to using and manipulating these previously out-of-reach and underutilized datasets. Short examples of tool usage as well as an in-depth sample analysis are presented along with descriptions of the individual tools. Chapter 5 contains a pilot study, conducted entirely within Galaxy, of “silent” synonymous single nucleotide polymorphisms (SNPs) appearing in isoleucine codons. The goal of this pilot study was to locate synonymous SNPs that are similar to a polymorphism found in the Multidrug Resistance 1 gene (MDR1), which has been previously shown to effect protein function. The step-by-step sample analyses and toolset integration provide real-life examples of the utility of Galaxy both as (1) an effective and intuitive analysis platform for experimental biologists and (2) a tool and data source integration framework for bioinformaticians.

Supplemental material for chapters 2, 3, 5 and 5 are included as appendices A, B, C and D, respectively. Appendix A contains implementation specifics of the Galaxy framework (at the time of original publication) that were too detailed to be included in the primary article; for the most up-to-date implementation details and feature documentation, users are encouraged to visit the Galaxy project homepage at http://galaxyproject.org. Appendix B contains links to screencasts (short, narrated videos) that demonstrate Galaxy’s effectiveness as an analysis platform for experimental biologists through a series of practical examples (protocols). Beyond the screencasts listed in Appendix B, additional screencasts are available and new ones are constantly being created which can be accessed at http://galaxycast.org. Appendix C contains step-by-step instructions on using the MAF toolset and for performing an in-depth sample analysis where multiple-species alignments of human regions containing synonymous protein coding SNPs are examined after filtering based upon changes in codon usage frequency. Appendix D contains a list of candidate synonymous SNPs that are similar to the previously identified “non-silent” synonymous SNP found in MDR1 and the steps used in Galaxy to produce this list.

References


CHAPTER 2

A FRAMEWORK FOR COLLABORATIVE ANALYSIS OF ENCODE DATA: MAKING LARGE-SCALE ANALYSES BIOLOGIST-FRIENDLY


Supplemental material is included as Appendix A.
Abstract

The standardization and sharing of data and tools are the biggest challenges of large collaborative projects such as the Encyclopedia of DNA Elements (ENCODEx). Here we describe a compact Web application, Galaxy2ENCODE, that effectively addresses these issues. It provides an intuitive interface for the deposition and access of data, and features a vast number of analysis tools including operations on genomic intervals, utilities for manipulation of multiple sequence alignments, and molecular evolution algorithms. By providing a direct link between data and analysis tools, Galaxy2ENCODE allows addressing biological questions that are beyond the reach of existing software. We use Galaxy2ENCODE to show that the ENCODE regions contain >2000 unannotated transcripts under strong purifying selection that are likely functional. We also show that the ENCODE regions are representative of the entire genome by estimating the rate of nucleotide substitution and comparing it to published data. Although each of these analyses is complex, none takes more than 15 min from beginning to end. Finally, we demonstrate how new tools can be added to Galaxy2ENCODE with almost no effort. Every section of the manuscript is supplemented with QuickTime screencasts. Galaxy2ENCODE and the screencasts can be accessed at http://g2.bx.psu.edu.

Analysis of data generated by The ENCODE Project Consortium (2004) for the Encyclopedia of DNA Elements (ENCODEx) is proving to be one of the most exciting collaborative events of the post-genomic era. The interpretation of enormous amounts of data generated by the ENCODE Consortium requires new methodologies for the sharing and standardization of data and new analysis tools. The system we describe here, Galaxy2ENCODE (http://g2.bx.psu.edu), is the first attempt to solve data and tool integration challenges for ENCODE-like projects and make data easily accessible for biomedical researchers. Galaxy2ENCODE attempts to serve both sides of the user distribution: experimental biologists and bioinformaticians. For experimental biologists, it provides an intuitive interface for data deposition and access, features a large number of tools, and makes analyses transparent by documenting every step in the history system. Most importantly, it streamlines the path from data to analyses, as even complex tools such as HyPhy (Pond et al. 2005) can be applied to genomic data directly without parsing or preprocessing. For computational biologists, Galaxy2ENCODE provides a framework that can integrate command-line tools with almost no effort. For each tool, Galaxy2ENCODE generates the interface and provides all housekeeping.

In this study, we demonstrate the utility of our system with examples using ENCODE data (the utility of our system is not limited to ENCODE). We show two complex analyses that can be conducted by using our system in <15 min. In the first example, we define and analyze all unannotated expressed
sequence tags (ESTs) in ENCODE regions. We show that over 2000 ESTs do not correspond to any annotated genes, yet show strong signature of purifying selection, indicating possible function. In the second example, we estimate the rate of nucleotide substitutions in ENCODE regions and demonstrate that it is consistent with genome-wide estimates. The two analyses are designed as “cookbook” examples for two distinct audiences. The first analysis is geared toward researchers studying the structure and function of the human genome. The second example is for researchers working in the area of evolutionary genomics. Finally, we show how easy it is to add new functionality to the Galaxy2\textsuperscript{ENCODE} toolbox and to use Galaxy2\textsuperscript{ENCODE} as a resource for sharing different analysis tools. This paper is supplemented with screencasts, short QuickTime movie clips. Each section of Results and Discussion features a screencast. The screencasts can be viewed directly from the main Galaxy2\textsuperscript{ENCODE} Web site (http://g2.bx.psu.edu) under the heading “Screencasts.”

Results and Discussion

Galaxy2\textsuperscript{ENCODE} interface and ENCODE data portal (Screencasts 1 and 2)

Galaxy2\textsuperscript{ENCODE} allows experimental biologists to retrieve and analyze data within a single unified interface. For this purpose, Galaxy2\textsuperscript{ENCODE} features a history system that stores data uploaded by the user as well as the results of all analyses. The concept of history was previously successfully deployed by our group (Giardine et al. 2005). The Galaxy2\textsuperscript{ENCODE} interface is shown in Supplemental Figure S1. The current version of Galaxy2\textsuperscript{ENCODE} allows users to create accounts and to have multiple histories (can be viewed at http://main.g2.bx.psu.edu).

To facilitate data exchange among different ENCODE groups during the analysis process, we implemented a local data repository at http://encode-upload.g2.bx.psu.edu. The repository is a Web application designed to (1) provide a user-friendly interface for data upload, (2) standardize naming of data files according to ENCODE guidelines, (3) automatically fragment the data into ENCODE analysis partitions, and (4) store the data for direct access through Galaxy2\textsuperscript{ENCODE} (http://encode.g2.bx.psu.edu) and ftp (ftp://encode:encode@g2.bx.psu.edu). See Methods for a description of the naming conventions and partition process.

Galaxy2\textsuperscript{ENCODE} tools (Screencasts 4–14)

The current version of Galaxy2\textsuperscript{ENCODE} provides access to >100 analysis tools. The functionality of each
category is detailed in tool screencasts (Screencasts 4–14). The most popular set of tools routinely used in genome analyses are operations on genomic intervals (Figure 1). These include the basic set operations of union, intersection, subtraction, and complement, as well as filters based on region size, proximity to regions from another query, and clustering by distance of regions within a single query. Many of these operations have options that allow the user to define what, for instance, “intersection” should mean when dealing with positional regions rather than atomic objects. The result is a new set of regions on which further processing can be performed. The Galaxy2 ENCODE toolset can be easily expanded. Developers can easily integrate any command-line tool as described below (see Screencast 19).

Galaxy2 ENCODE supports several variations of the basic set operations designed specifically for manipulation of genomic intervals.

Analysis of intronic, intergenic, and intertwined ESTs (Screencasts 15–17)

Here we define and characterize the 9191 transcripts that lie outside annotated genes within ENCODE regions. These are of considerable interest, as some may represent genes missed during the annotation process. We used GENCODE annotation as the source of gene data (http://genome.imim.es/gencode/). Genes are first predicted computationally and then experimentally verified using techniques such as RT-PCR, RACE, and direct sequencing of the products. As such, the gene predictions of GENCODE are the most reliable. In the following analysis, we define “genes” as the union of GENCODE Known Genes, GENCODE Putative Genes, and GENCODE pseudogenes annotations frozen during the Second ENCODE Workshop (University of California Santa Cruz, November 2005). Using genomic coordinates, we identified all ESTs that map outside GENCODE genes. We call such ESTs Non-GENCODE ESTs. Non-GENCODE ESTs belong to three categories (Figure 2): intronic, intergenic, and intertwined (or interleaved as suggested by Chen and Stein 2006). Figure 3 summarizes the steps of our analysis, which takes ~15 min to complete. See Screencast 15 and the Methods section for a step-by-step explanation of the procedure. Briefly, we first defined a set that includes all Non-GENCODE ESTs (Fig. 3A–D). Then, we classified Non-GENCODE ESTs into intronic, intergenic, and intertwined (Fig. 3E,F). Finally, we computed descriptive statistics as shown in Table 1.

Having defined Non-GENCODE ESTs in ENCODE regions, we can now use Galaxy2 ENCODE to look into the biology of these transcripts. How many Non-GENCODE ESTs correspond to missing protein-coding genes? What fraction of the Non-GENCODE ESTs are under purifying selection? Is there a
significant overlap between Non-GENCODE ESTs and transcriptional evidence produced by alternative methods? These are just some of the questions that can be easily answered with versatile Galaxy2\textsuperscript{ENCODE} tools.

To find out how many Non-GENCODE ESTs may represent missing or misannotated protein-coding genes, we computed the overlap between the EST exons and protein-coding regions predicted by Exoniphy. Exoniphy is an ab initio exon predictor that uses nucleotide substitution patterns and phylogenetic information to predict protein-coding regions with a high degree of accuracy (Siepel and Haussler 2004). First, we computed the overlap between exons of Non-GENCODE ESTs and exons predicted by Exoniphy using the Overlap tool. We then used the Base coverage tool to identify those Non-GENCODE EST exons that are covered by Exoniphy predictions for at least 75% of their length. Only one EST (accession no. DR731323) was found to overlap with three consecutive Exoniphy exons and represents a 3′-end extension of an Ensembl gene ENST00000355799 (Supplemental Fig. S2).

While only one of the Non-GENCODE ESTs appears to be protein-coding, others may be functional but non-coding. One of the ways to pinpoint functional non-coding regions is to measure the strength of purifying selection acting on the genomic region of interest. In Galaxy2\textsuperscript{ENCODE}, the strength of purifying selecting may be assessed using phastCons scores (Siepel et al. 2005). The phastCons score is one of the best measures of the strength of purifying selection acting on a DNA sequence. A high phastCons score (≥0.2) may be taken as strong evidence of the functional importance of a genomic region (Siepel and Haussler 2004; King et al. 2005). To perform these analyses, we “aggregated” phastCons scores for exons of Non-GENCODE ESTs using the Aggregate-datapoints tool (The aggregation is performed because phastCons scores are base-pair-specific; thus to obtain a phastCons score for an exon, phastCons values of individual nucleotides must be averaged for all nucleotides within that exon using the Aggregate tool.) After aggregation is complete, we filter out regions with average phastCons scores below 0.2. This leaves 3705 (14%) Non-GENCODE EST exons from the total of 27,202. At this point of the analyses, we operate with individual exons. However, in this case, it interesting to know which of the Non-GENCODE ESTs have all exons with the average phastCons score above 0.2. Using a combination of filtering and relational database operations implemented in Galaxy2\textsuperscript{ENCODE}, we identified 2180 such ESTs (942 intronic, 221 intergenic, and nine intertwined, respectively). An example of an intergenic EST from this set (accession no. DB275065) is shown in Supplemental Figure S3. Note the conservation peaks surrounding exons of this EST. Transcripts identified using this approach are strong candidates for further experimental validation.

If Non-GENCODE ESTs represent biologically relevant transcripts, there should be a significant
overlap between them and transcribed regions of the genome confirmed with other methods, such as transcribed fragments (transfrags) produced by the Affymetrix group (Kampa et al. 2004; Cheng et al. 2005). Galaxy2 ENCODE allows one to test the significance of the overlap between two sets of genomic features such as, for example, Non-GENCODE EST exons and transfrags. To perform this test, we designed a Random Intervals tool that generates a set of simulated regions that mimic a given set of intervals. In this example, we first (Experiment A) computed the intersection between exons of Non-GENCODE EST (including all three categories: Intertwined, Intergenic, and Intronic) and transfrags within ENCODE regions. Next (Experiment B), we used the Random Interval tool to generate a set of genomic intervals that mimic the length distribution of Non-GENCODE EST exons but lie outside transfrags. We then computed the intersection between exons of Non-GENCODE ESTs and the set of Random intervals. Comparing results of experiments A and B shows that the overlap between Non-GENCODE ESTs and transfrags is likely nonrandom (Table 2). The base-pair coverage in Experiment A is consistently higher than that in Experiment B. To obtain the empirical p-value, one can repeat Experiment B multiple times.

**Estimating mammalian substitution rates**

Since ENCODE regions have the highest depth of annotation, it is tempting to extrapolate their properties to the entire genome. However, is this legitimate? In other words, do ENCODE regions represent an unbiased sample of the genome? One way to answer this question is to compare evolutionary parameters of the ENCODE region with genome-wide estimates published elsewhere. We used ancestral repeats (ARs) (Hardison et al. 2003) to show that ENCODE regions are, indeed, representative of the remaining euchromatic portion of the genome. The AR coordinates were retrieved by using the ENCODE Multi-Species Sequence Analysis tool, and then the Filter tool was used to limit the results to ENCODE’s autosomal regions. Next, multiple alignments between mammalian genomes were extracted for the intervals and converted to FASTA-formatted sequences with the Maf-to-FASTA converter, where we also narrowed our species range to human, chimpanzee, mouse, rat, and dog. The total alignment length was 364 kb. We then applied a HyPhy wrapper (Pond et al. 2005) to this set using the general reversible model of nucleotide substitutions (REV) (Rodriguez et al. 1990; Yang et al. 1994) and obtained the following branch lengths: [(human: 0.006, chimp:0.007):0.098, (mouse:0.084, rat:0.112):0.276, dog:0.231] (Table 3). The analysis took 7 min to complete. These results are consistent with recent genomic studies (Gibbs et al. 2004; The Chimpanzee Sequencing and Analysis Consortium 2005; Lindblad-Toh et al. 2005). The 95% confidence intervals were derived with
the profile likelihood approach implemented in the HyPhy package (Pond et al. 2005).

**Galaxy2** ENCODE as a community resource for distributing tools (Screencasts 18 and 19)

ENCODE analysis groups have designed several innovative software tools that can be of great use to the rest of the genomic community. Galaxy2** ENCODE** can be used to provide unified, simple, and user-friendly interfaces for these tools. Adding tools does not require any knowledge about the internal operation of Galaxy2** ENCODE**. The entire tool deployment process consists of downloading a software distribution from [http://g2.bx.psu.edu](http://g2.bx.psu.edu), installing it (see the 3-min Screencast 18 that explains all steps of the installation process), and performing the two steps described in Supplemental Materials (also see Screencast 19).

**Conclusions**

We demonstrated that Galaxy2** ENCODE** serves as a new, critically needed environment that can foster interactions between experimental and computational biologists by providing a simple interface (important to the former) and a robust software integration environment (important for the latter). Galaxy allows data producers to deposit data and make them immediately available to the biological community. It features over 100 unique tools that allow the user to manipulate sequences, coordinates, and alignments on the genome-wide scale. The simplicity of Galaxy2** ENCODE**’s tool integration protocol allows developers and occasional scripters alike easily to integrate their programs and make them available to biologists.

**Methods**

Galaxy2** ENCODE** is a completely new compact implementation that combines the latest open-source technologies with ideas previously developed by our group (Giardine et al. 2005). A detailed description of (1) uploading and processing of ENCODE data, (2) finding Non-GENCODE ESTs, and (3) implementation details can be found in the Supplemental Material. In addition, our wiki page at [http://g2.bx.psu.edu](http://g2.bx.psu.edu) contains source code, written instructions, and screencasts on using, downloading, and developing Galaxy2. Usage-related questions should be directed to galaxy-bugs@bx.psu.edu.
Figures

Query 1: Highly conserved regions
Query 2: Known exons

Union, returning original regions:
Union, returning merged regions:
Intersection, returning whole regions from query 1:
Intersection, returning whole regions from query 2:
Intersection, returning only overlapping segments:
Subtraction 1-2, removing whole regions:
Subtraction 1-2, removing overlapping segments:
Complement query 1:

Figure 1. Galaxy\textsuperscript{ENCODE} supports several variations of the basic set operations designed specifically for manipulation of genomic intervals.
Figure 2. Types of Non-GENCODE ESTs.
A
Encode genes
| Encode exons

B
ESTs
1 2 3 4 5 6 7 8 9 10

EST Exons

(COUNT of EST exons AFTER subtraction)

D
<table>
<thead>
<tr>
<th>EST1</th>
<th>EST2</th>
<th>EST3</th>
<th>EST4</th>
<th>EST5</th>
<th>EST6</th>
<th>EST7</th>
<th>EST8</th>
<th>EST9</th>
<th>EST10</th>
</tr>
</thead>
<tbody>
<tr>
<td># exons</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

(COUNT of EST exons BEFORE subtraction)

<table>
<thead>
<tr>
<th>EST1</th>
<th>EST2</th>
<th>EST3</th>
<th>EST4</th>
<th>EST5</th>
<th>EST6</th>
<th>EST7</th>
<th>EST8</th>
<th>EST9</th>
<th>EST10</th>
</tr>
</thead>
<tbody>
<tr>
<td># exons</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

E
Mix of Intronic, Intergenic, and Intertwined ESTs

Result: INTERGenic ESTs

F
Mix of Intronic and Intertwined ESTs

Result: Intertwined ESTs

G
Using IDs subtract from all NonGenCode ESTs (1, 2, 4, 5, 6, 8, 9, and 10)
Intergenic (4 and 5) and Intertwined (1, 6, and 10). This leaves 2, 8, and 9 = Intronic ESTs:

Result: Intronic ESTs
**Figure 3.** Steps (A–G) in identification of Non-GENCODE ESTs. Galaxy2 makes such analyses transparent. See Methods and Screencast 15 for explanations of each step.
### Table 1. Descriptive statistics for the three categories of Non-GENCODE ESTs

<table>
<thead>
<tr>
<th>Non-GENCODE EST classes</th>
<th>Intrinsic</th>
<th>InterGenic</th>
<th>InterTwined</th>
</tr>
</thead>
<tbody>
<tr>
<td># ESTs</td>
<td>7441</td>
<td>1876</td>
<td>56</td>
</tr>
<tr>
<td># EST exons</td>
<td>21,692</td>
<td>5,242</td>
<td>268</td>
</tr>
<tr>
<td># merged EST exons</td>
<td>6532</td>
<td>1543</td>
<td>158</td>
</tr>
<tr>
<td>Base coverage</td>
<td>1,601,356</td>
<td>403,241</td>
<td>22,572</td>
</tr>
<tr>
<td>Overlap (bp) w/transfrags&lt;sup&gt;a&lt;/sup&gt;</td>
<td>82,194</td>
<td>20,125</td>
<td>1181</td>
</tr>
<tr>
<td>Overlap (bp) w/repeats&lt;sup&gt;a&lt;/sup&gt;</td>
<td>539,692</td>
<td>135,494</td>
<td>8962</td>
</tr>
</tbody>
</table>

<sup>a</sup>These values were obtained by first computing intersections between Non-GENCODE EST exons and transfrags and between Non-GENCODE EST exons and repetitive elements identified with RepeatMasker. Next, for each intersection we computed base coverage.

### Table 2. Overlap among Non-GENCODE EST exons, Affymetrix transfrags, and random intervals

<table>
<thead>
<tr>
<th>Overlap with</th>
<th>Total coverage</th>
<th>transfrags (Experiment A)</th>
<th>Random intervals (Experiment B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intertwined</td>
<td>22,572</td>
<td>1181</td>
<td>345</td>
</tr>
<tr>
<td>InterGenic</td>
<td>403,241</td>
<td>20,125</td>
<td>9376</td>
</tr>
<tr>
<td>Intrinsic</td>
<td>1,601,356</td>
<td>82,194</td>
<td>44,624</td>
</tr>
<tr>
<td>transfrags</td>
<td>1,373,896</td>
<td>—</td>
<td>24,302</td>
</tr>
</tbody>
</table>
### Table 3. Nucleotide substitution analysis of ENCODE ancestral repeats (located within autosomes) using *HyPhy* wrapper

<table>
<thead>
<tr>
<th>Branch</th>
<th>Mean</th>
<th>Lower-bound</th>
<th>Upper-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>0.0057</td>
<td>0.0056</td>
<td>0.0057</td>
</tr>
<tr>
<td>Chimpanzee</td>
<td>0.0072</td>
<td>0.0071</td>
<td>0.0073</td>
</tr>
<tr>
<td>Node1</td>
<td>0.0984</td>
<td>0.0978</td>
<td>0.0990</td>
</tr>
<tr>
<td>Mouse</td>
<td>0.0849</td>
<td>0.0843</td>
<td>0.0856</td>
</tr>
<tr>
<td>Rat</td>
<td>0.1122</td>
<td>0.1116</td>
<td>0.1129</td>
</tr>
<tr>
<td>Node4</td>
<td>0.2759</td>
<td>0.2749</td>
<td>0.2770</td>
</tr>
<tr>
<td>Dog</td>
<td>0.2305</td>
<td>0.2298</td>
<td>0.2313</td>
</tr>
<tr>
<td>Total tree length</td>
<td>0.8149</td>
<td>0.8135</td>
<td>0.8162</td>
</tr>
</tbody>
</table>

Nodes are numbered as given by the tree: [(human, chimpanzee), (mouse, rat), dog].
Acknowledgments

We thank David Haussler and Jim Kent for their continuing support of the project and the members of the Center for Comparative Genomics and Bioinformatics at Penn State for their input. Roderic Guigo, France Denoeud, Julien Lagarde, and Robert Castelo provided critical comments during software testing. Special thanks to Michael O’Connor for editing the wiki page content. This work is supported by funds provided by the Eberly College of Science, Huck Institutes of the Life Sciences, at Penn State University; NSF DBI grant 0543285 to A.N.; NIH R01 HG002238 to W.M.; and NIH R01 GM072264 to K.M.

References


Hardison, R.C., Roskin, K.M., Yang, S., Diekhans, M., Kent, W.J., Weber, R., Elnitski, L., Li, J,


CHAPTER 3

GALAXY, A WEB-BASED GENOME ANALYSIS TOOL FOR EXPERIMENTALISTS


Links to the screencasts referred to in this chapter are included in Appendix B.
Abstract

High-throughput data production has revolutionized molecular biology. However, massive increases in data generation capacity require analysis approaches that are more sophisticated, and often very computationally intensive. Thus making sense of high-throughput data requires informatics support. Galaxy (http://galaxyproject.org) is a software system that provides this support through a framework that gives experimentalists simple interfaces to powerful tools, while automatically managing the computational details. Galaxy is available both as a publicly available web service, which provides tools for the analysis of genomic, comparative genomic, and functional genomic data, or a downloadable package that can be deployed in individual labs. Either way, it allows experimentalists without informatics or programming expertise to perform complex large-scale analysis with just a web browser.

Introduction

Research in the life sciences continues to become more data intensive. With new high throughput experimental techniques, an individual lab can generate raw data of a scale that was unthinkable only a few years ago. These developments represent an enormous opportunity for basic and applied research. However, they are also creating a crisis for many experimental scientists, since making sense of this wealth of data requires significant analysis infrastructure. Without informatics support, experimental biologists, who possess key biological knowledge and experience, and thus the best potential for making novel discoveries, cannot effectively use the available data.

Galaxy (http://galaxyproject.org) rectifies this situation by providing the needed informatics infrastructure (Taylor et al, 2007). For experimentalists, it provides an analysis environment in which they can perform analysis interactively, while ensuring that the resulting analyses are transparent and reproducible. The Galaxy framework encapsulates high-end computational tools, and gives them intuitive user interfaces while hiding the details of compute and storage management. It thus eliminates the need for specialized informatics expertise when performing many common types of large-scale analysis.

This unit describes the functionality of Galaxy using a series of examples. The material in this section is directed primarily at experimentalists, and makes use only of analysis tools available at the public Galaxy service at http://usegalaxy.org. Various components and tools of the public Galaxy server will be explored by following several connected, but independent, protocols. Although the data being investigated in these protocols may not be of personal research interest, the techniques demonstrated
are useful in a wide-array of applications. Each of the protocols below is accompanied by a screencast (a real-time movie showing the steps of the protocol as they appear on the screen) available from http://galaxycast.org/CPMB. Following along with the screencasts is recommended, and they provide an alternate presentation of details not easily conveyed by text. This unit is divided into the following protocols:

- Basic Protocol 1. An introduction to the Galaxy approach: Finding promoters containing TAF1 binding sites identified from a CHiP-seq experiment
- Basic Protocol 2. A bit more data manipulation: Finding coding exons with most SNPs
- Support Protocol 2.1. Saving results in Galaxy and sharing data with others
- Basic Protocol 3. Generating a workflow from a history in Galaxy
- Support Protocol 3.1. Modify a parameter of the workflow in Galaxy
- Support Protocol 3.2. Running workflows with Galaxy
- Support Protocol 3.3. Sharing workflows with Galaxy
- Basic Protocol 4. Generating workflows from scratch with Galaxy
- Basic Protocol 5. Extracting sequences and alignments with Galaxy: A SNPs in exons example

These protocols cover the basic aspects of Galaxy’s functionality. They are sufficient for overcoming the initial learning curve, but Galaxy has much more to offer, including complex analyses of next generation sequencing data such as metagenomic applications or re-sequencing studies. Additionally, the Galaxy project is progressing rapidly with new tools and features added on a monthly basis. The best way to keep up with these enhancements is to regularly check the screencast page at http://galaxycast.org.

Before beginning the protocols it will be beneficial to review some terminology and concepts. Many of the formats ("datatypes") used in genomics are composed of rows of tab-delimited columns which contain varied data (known as tabular data and similar in function to a spreadsheet). One of these, known as interval, in which each row represents the position of a genomic feature in a particular genome. The interval format contains at least three columns: (1) the chromosome, (2) the start position within that chromosome and (3) the end position within that chromosome. Other columns commonly included are name, strand, score, and exon information (when the intervals are gene annotations). Additional formats beyond those composed of tabular columns are used, but the intricacies of their
formats can be largely ignored in this introductory text as Galaxy can handle most of the details needed for performing complex analysis. The practice of matching rows between tabular datasets with Galaxy is known as "joining." Two different Join tools are used here. The first Join tool works on interval datasets (using multiple columns to determine matching) and creates a dataset where rows are matched if their interval on the genome overlaps (by a user specified number of nucleotides) and combined into a single row. The second type of join works on a single column from each dataset and is useful for matching between identifiers. Every time a tool is run, one or more datasets are created in the user’s history. The box surrounding the dataset will change color based upon its state: a query in the queue will be indicated by a gray box, a running query will be yellow and a completed query will have a green box. Although a dataset is only ready to be viewed or used as input after it has turned green, additional analysis steps can be lined-up for non-completed queries by using the desired tools as normal; the tools will wait in the queue for the dataset needed to finish before running. Examining Figure 1 in detail will familiarize the user with the layout of Galaxy’s interface, including a user's history and the tools menu.

**Basic Protocol 1. An introduction to the Galaxy approach: Finding promoters containing TAF1 binding sites identified from a CHiP-seq experiment**

Suppose a CHiP-seq experiment was performed and identified a series of genomic regions that bind TAF1-protein. The next task is to identify a list of genes that contain such sites. This can be easily done with Galaxy in just a few steps. This protocol uses a file arranged by tab-delimited columns, where each column contains information about the genomic positions ("intervals"), as well as name and score data, for TAF1-binding sites from a CHiP-seq experiment. Each row in this file represents an individual TAF1-binding site by listing the chromosome and the start and end positions within that chromosome. Here we assume that the ChIP-seq data has already been processed into putative binding regions, since this procedure is currently very experiment and lab specific. However, as best-practices are defined for performing and evaluating the quality these procedures, appropriate tools will be added to Galaxy.

**Materials:**

- a file containing genomic coordinates for TAF1-binding sites from the CHiP-seq experiment (an example file can be downloaded at [http://galaxy.psu.edu/CPMB/TAF1_CHiP.txt](http://galaxy.psu.edu/CPMB/TAF1_CHiP.txt)). (Kim et al, 2005)

- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
Steps: *(Please note that the items to alter will be stated in the text. If other menus and options are not referenced, leave those settings in their default or existing condition.)*

1. Upload the TAF1 CHiP-seq data. Before beginning the analysis the CHiP-seq data needs to be uploaded into Galaxy’s workspace (known as a user’s "history" throughout this document).
   b. Click "Get Data".
   c. Click "Upload file".
   d. Middle panel of the interface will change and allow selection of the desired file (use the example file that can be downloaded from [http://galaxy.psu.edu/CPMB/TAF1_CHiP.txt](http://galaxy.psu.edu/CPMB/TAF1_CHiP.txt)).
      Note: It is possible to skip the downloading step and directly upload the data by entering the URL into the paste box, causing Galaxy to fetch the URL contents automatically.
   e. Click "Execute". The dataset will be uploaded and will appear as dataset #1 within the right panel.

2. Set properties of the TAF1 dataset. To begin the analysis, a number of properties for the CHiP-seq dataset need to be set.
   1. Expand the dataset by clicking on the name of the item in the history list (TAF1_CHiP.txt).
   2. Click "?" next to "database:". A new interface will appear in the middle panel.
   3. Use the "Database/Build" dropdown to select "Human Mar. 2006 (hg18)". Click the Save button. The dataset is now designated as originating from the human genome.
   4. Back in the right panel, click the pencil icon. A new interface will appear in the middle.
   5. Use the "New Type" dropdown within the "Change data type" box to select "interval". The "interval" datatype describes data representing genomic coordinates or "intervals" (chromosomes, start positions and end positions within chromosomes, as well as variable data, for a set of genomic features).
   6. Click the "Save" button immediately below the box. The upper part of the interface will change.
   7. Set "Chrom column", "Start column" and "End column" to 2, 3 and 4, respectively. Check the "Name" checkbox and select "5" from the adjacent dropdown.
   8. Click the "Save" button immediately below.
9. The appearance of dataset #1 within the right panel will change -- column headers will appear, and the format will change to "interval".

3. Upload gene annotations from the UCSC Table Browser (Karolchik et al, 2008 ; Karolchik et al, 2004). To identify which genes' promoters contain the TAF1 binding sites, the gene coordinates must first be uploaded.

   a. Click "Get Data" in the Tools menu list on the left panel.
   b. Click "UCSC Main". The UCSC Table Browser interface will be displayed in the middle panel.
   c. Because the data is of human annotations, make sure that "clade", "genome" and "assembly" are set to "Mammal", "Human" and "Mar. 2006", respectively.
   d. Set "group" to "Genes and Gene Prediction tracks" and "track" to "RefSeq Genes".
   e. Select the radio button "genome".
   f. Make sure "output format" is set to "BED - browser extensible data" and the checkbox by "Send output to Galaxy" is checked. The BED format is a specialized version of the interval format discussed earlier.
   g. Click "get output". A new interface will appear.
   h. Make sure the "Whole Gene" radio button is selected.
   i. Click "Send query to Galaxy". At this point a new dataset, #2, will appear in Galaxy's history on the right panel. This dataset contains the genomic positions for all RefSeq genes from the March 2006 human genome assembly.
   j. Rename dataset #2 to "RefSeq" by clicking the pencil icon and typing "RefSeq" in the "Name" field that appears within the center panel. Click the Save button.

4. Transform coordinates of genes into coordinates of putative promoters.

   a. Click "Operate on genomic intervals" in the Tools menu list on the left panel.
   b. Click "Get flanks". A new interface will appear in the center panel.
   c. Make sure the "Select data" dropdown is set to dataset #2 RefSeq.
   d. Set "Length of the flanking region/s" to "1000".
   e. Click "Execute". A new dataset, #3, containing the putative promoters, will appear within the right panel.
f. Rename dataset #3 to "Promoters" by clicking the pencil icon and typing "Promoters" in the "Name" field that will appear within the center panel. Save.

5. Remove unnecessary columns for dataset #3. Only five columns are needed from this dataset. Galaxy's Cut tool allows the removal of unwanted columns.
   a. Click "Text Manipulation" in the Tools menu list on the left panel.
   b. Click "Cut columns from a table". A new interface will appear in the center panel.
   c. Type "c1,c2,c3,c4,c6" in the "Cut columns" text box.
   d. Click "Execute". A new dataset, #4, will appear within the right panel, containing only columns one through four and column six, which correspond to the chromosome, start position, end position, name and strand, respectively, for the calculated promoter regions.
   e. Rename dataset #4 to "Clean Promoters" by clicking the pencil icon and typing "Clean Promoters" in the "Name" field that will appear within the center panel. Save.
   f. Because the Cut tool breaks column assignment, the pencil icon will need to be clicked. A new interface will appear in the middle.
   g. Use the "New Type" dropdown within the "Change data type" box to select "interval". The "interval" datatype describes data representing genomic coordinates (intervals).
   h. Click the "Save" button immediately below the box. The upper part of the interface will change.
   i. Set "Chrom column", "Start column" and "End column" to 1, 2 and 3, respectively. Check the "Strand" checkbox and select "5" from the adjacent dropdown. Check the "Name" checkbox and select "4" from adjacent dropdown.
   j. Click the "Save" button immediately below.
   k. The appearance of dataset #4 within the right panel will change -- column headers will appear, and the format will change to "interval".

6. Identify promoters containing the TAF1 binding sites. Now join the coordinates of TAF1 binding sites from dataset #1 with the coordinates of putative promoters from dataset #4. The Genomic Interval Join Tool matches two separate sets of genomic coordinates (intervals) according to their overlap, creating a single output containing the matched rows.
   a. Click "Operate on Genomic Intervals" in the Tools menu list on the left panel.
b. Click "Join". A new interface will appear in the center panel.

c. Select dataset #4 Clean Promoters from the first dropdown called "Join".

d. Select dataset #1 TAF1_CHiP.txt from the second dropdown called "with".

e. Make sure the "Return" dropdown is set to "Only records that are joined".

f. Click "Execute". A new dataset, #5, will appear in the right panel. This dataset will list coordinates of putative promoters and TAF1 binding sites side by side as shown in Figure 5.

7. Visualize results of this analysis using the UCSC Genome Browser.

   a. Click "Graph/Display Data" in the Tools menu list on the left panel.

   b. Click "Build custom track". A new interface will appear in the center panel.

   c. Click "Add new Track" and select dataset #1 TAF1_CHiP.txt from the "Dataset" dropdown.

   d. Type "TAF1" in the "Name" and "Description" boxes. Leave other settings as default.

   e. Click "Add new Track" to create Track 2, and select dataset #4 Clean Promoters from the "Dataset" dropdown.

   f. Type "Promoters" in the "Name" and "Description" boxes. Set the color to Blue.

   g. Click "Add new Track" to create Track 3 and select dataset #5 from the "Dataset" dropdown.

   h. Type "Overlap" in the "Name" and "Description" boxes. Set the color to Purple.

   i. Click "Execute". A new dataset, #6 Build custom track on data 5, data 4, and data 1, will appear within the right panel.

   j. Once dataset #6 becomes green in the History panel list, click on its name and then click the "display at UCSC main" link. The browser will open a new window or a tab with the UCSC Genome Browser interface displaying datasets 1, 4 and 5 as custom tracks.

Basic Protocol 2. Combining and filtering genome annotations: Finding Exons with the highest number of nucleotide polymorphisms

The objective of this protocol is to demonstrate joining, grouping, sorting, and filtering of genomic annotations in Galaxy. To explore these features using real data an illustrative example will be used: identification of exons containing the largest number of single nucleotide polymorphisms (SNPs).

Materials:
• an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).

Steps: *(Please note that it is beneficial to clear the current history and start re-numbering from 1 by accessing the History Options and selecting “Create a new empty history”. It simplifies following with the numbered steps.)*

1. Upload exon annotations from the UCSC Table Browser.
   a. Click "Get Data" in the Tools menu list on the left panel.
   b. Click "UCSC Main". The UCSC Table Browser interface will be displayed in the middle panel.
   c. Because the data of interest are human annotations make sure that "clade", "genome" and "assembly" are set to "Mammal", "Human" and "Mar. 2006", respectively.
   d. Set "group" to "Genes and Gene Prediction tracks" and "track" to "UCSC Genes".
   e. Select the radio button "position" and type "chr22" within the adjacent text box. This will limit the annotations to the entirety of chromosome 22.
   f. Make sure "output format" is set to "BED - browser extensible data" and the checkbox by "Set output to Galaxy" is checked. The BED format is a specialized version of the interval format discussed earlier; it contains the information required to represent a genomic position.
   g. Click "get output". A new interface will appear.
   h. Make sure the "Coding Exons" radio button is selected in the “Create one BED record per:” area. The genes on chromosome 22 will be divided into coding exons, with each exon having its own set of genomic intervals.
   i. Click "Send query to Galaxy". At this point a new dataset, #1, will appear in Galaxy's history on the right panel. A query in the queue will be indicated by a gray box, a running query will be yellow, and a completed query will have a green box.
   j. When the query has completed, rename dataset #1 to "exons" by clicking the pencil icon and typing "exons" in the "Name" field that will appear within the center panel. Click the Save button.

2. Upload SNP coordinates.
   a. Click "Get Data" from the Tools menu list on the left panel.
   b. Click "UCSC Main". The UCSC Table Browser interface will be displayed in the middle panel.
c. Because the data of interest are human annotations, make sure that "clade", "genome" and "assembly" are set to "Mammal", "Human" and "Mar. 2006", respectively.

d. Set "group" to "Variation and Repeats" and "track" to "SNPs (129)".

e. Select the radio button "position" and type "chr22" within the adjacent text box.

f. Make sure "output format" is set to "BED - browser extensible data" and a checkbox by "Set output to Galaxy" is checked.

g. Click "get output". A new interface will appear.

h. Make sure the "Whole Gene" radio button is selected in the “Create one BED record per:" area.

i. Click "Send query to Galaxy". At this point, a new dataset, #2, will appear in Galaxy's history in the right panel.

j. When the query has completed, rename dataset #2 to "snps" by clicking the pencil icon and typing "snps" in the "Name" field that will appear within the center panel. Save.

3. Join coordinates of exons with coordinates of SNPs to identify those exons that contain SNPs.

   a. Click "Operate on Genomic Intervals" in the Tools menu list on the left panel.

   b. Click "Join" in the Operate submenu. A new interface will appear in the center panel.

   c. Select dataset #1 exons from the first dropdown called "Join".

   d. Select dataset #2 snps from the second dropdown called "with".

   e. Make sure the "Return" dropdown is set to "Only records that are joined (INNER JOIN)".

   f. Click the "Execute" button. A new dataset, #3 Join on data 2 and data 1, will appear in the right panel. This dataset will list coordinates of exons and SNPs side by side as shown in Figure 7. The data can by examined by clicking on the name.

4. Count the number of SNPs per exon using the Group tool. In Figure 7, it is seen that if an exon contains multiple SNPs its name is repeated. It is possible to take advantage of this by using the Group tool. By counting the number of times each exon's name appears within dataset #3, the number of SNPs within that exon will be obtained.

   a. Click "Join, Subtract, and Group" from the Tools menu list on the left panel.

   b. Click "Group". A new interface will appear in the center panel.
c. Set the "Select data:" dropdown to dataset #3 Join on data 2 and data 1.
d. Set "Group by column" to "c4", as this column contains exon identifiers.
e. Click the "Add new operation" button. A new section of interface named "Operation 1" will appear below.
f. Within the new interface section, set "Type" to "Count" and "On column" to "c4".
g. Click the "Execute" button. A new dataset, #4 Group on data 3, will appear in the right panel. It will contain two columns: (1) exon id and (2) SNP count.

5. Sort exon by SNP count. To see the highest possible number of SNPs per exon in this dataset, sort the dataset from the previous step.
   a. Click "Filter and Sort" from the Tools menu list on the left panel.
   b. Click "Sort". A new interface will appear in the center panel.
   c. Set "Sort Query" to dataset #4 Group on data 3.
   d. Set "on column" to "c2" (the SNP count calculated above).
   e. Click "Execute". Dataset #5 Sort on data 4 will appear in the history panel.
   f. Click on the eye icon to see which exons have the highest SNP count.

6. Restrict dataset #5 to exons that have ten or more SNPs.
   a. Click "Filter and Sort" from the Tools menu list on the left panel.
   b. Click "Filter".
   c. Set "Filter" to dataset #5 Sort on data 4.
   d. Set "With following condition" to "c2 >= 10" (without the quotes). This is because column 2 (c2) contains the count of SNPs per exon; only rows in which the contents of column two is greater-than or equal-to ten will be kept.
   e. Click "Execute". Dataset #6 Filter on data 5 will appear in the history panel.

7. Restore genomic location for exons containing ten or more SNPs. The previous step has produced a list of exons containing ten or more SNPs, however, information about their genomic position, strand orientation, etc has been lost. Because dataset #6 contains the exon identifier field, it can be used to restore genomic context information by joining with dataset #1. The Join two Queries tool is different.
than the Genomic Operations Join which was used earlier; this tool matches two separate datasets by
matching column contents between any tab-delimited dataset (including interval datasets).

a. Click "Join, Subtract, and Group" from the Tools menu list on the left panel.
b. Click "Join two Queries". A new interface will appear within the center panel.
c. Set "Join" to dataset #1 exons.
d. Set "Using column" to "c4" as this column contains exon identifiers in dataset #1.
e. Set "with" to dataset #6 Filter on data 5.
f. Set "and column" to "c1" as this column contains exon identifiers in dataset #6.
g. Click "Execute". Dataset #7 Join two Queries on data 6 and data 1 will appear within the history
panel on the right. It contains full genomic context information about exons containing ten or
more SNPs in chromosome 22.

8. Visualize dataset #7 Join two Queries on data 6 and data 1 in UCSC Genome Browser.
   a. Go to the right (history) panel and expand dataset #7 by clicking on the name of the dataset.
   b. Click the "display at UCSC main" link. A new browser tab (or window) will open dataset #7
      within the UCSC Genome Browser. The query will be displayed as a track called "User
      Supplied Track". Access to menu control of this track is available in the menus area below, and
      the track will be available in the UCSC Table Browser for further query and manipulation.

9. To save the analysis and share it with colleagues continue on to Support Protocol 2.1

Support Protocol 2.1. Saving results in Galaxy and sharing data with others

How can researchers ensure that the analyses they have just conducted are safely stored and that they
are able to go back to them at anytime? They will need to create a free account within Galaxy. This is
the only requirement to save analyses. The protocol below explains this and also introduces sharing
analyses with colleagues.

Materials:

- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
- completed protocols – results of Basic Protocol 2.
- A Galaxy account (created by clicking “Register” in the Galaxy interface). Histories must be
  linked to a user to be stored and shared.
Steps:

1. Rename the history. All histories are given the default name "unnamed", which is obviously not very descriptive. This is easily changed by following the steps below.
   a. Click the pencil icon immediately above the history items (on a lavender background). A text box will appear to the left of the pencil.
   b. Type "Exons and SNPs" in the textbox and hit the Enter key or return key on the keyboard.

2. Click the "Options" button above the history panel. A list of history actions will appear in the middle panel.

3. Click the "Share current history" link.

4. Enter the e-mail address of an existing Galaxy user and click "Submit". This history is now shared.

**Basic Protocol 3. Generating a workflow from a history in Galaxy**

Protocols 1 and 2 demonstrate interactive analysis in Galaxy, the result is a "history" which documents each step of an analysis. Galaxy also allows the construction of reusable multi-step analysis "workflows". In this protocol, the creation of a workflow from an existing analysis history is demonstrated.

Materials:

- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
- completed protocols – the history that was created by following Basic Protocol 2.
- A Galaxy account (created by clicking “Register” in the Galaxy interface). All workflow in Galaxy requires the user to be logged in with an account.

Steps:

1. Ensure a non-empty history is loaded (for this example, the history resulting from the completion of Basic Protocol 2 is used).

2. In the header of the History panel (top-right of the Galaxy analysis interface) click "Options". This will load a menu of options, which apply to the current history, in the center panel.

3. In the center panel, click "Construct workflow from the current history". This will load a list of the actions (tool runs) which generated each dataset in the current history. A subset of tools can be selected by clicking the checkboxes on this page (e.g. if more than one analysis has been
performed in the current history, but a workflow is only to be created from one of them). Certain tools cannot be used in workflows (including most external data sources), in these cases the dataset can be treated as an input to the workflow. Here, a workflow is constructed from the entire history, so do not change any checkboxes.

4. Provide a name for the new workflow by entering a name of choice in the text box underneath the label "Workflow name".

5. Click the "Create Workflow" button to create the new workflow; a message will be displayed in the center panel confirming that the workflow was created.

Support Protocol 3.1. Modify a parameter of the workflow in Galaxy

After constructing a workflow from an existing analysis, the Workflow Editor can be used to modify tool parameters (or even add and remove steps).

Materials:

- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).

- completed protocols – the workflow that was created by following Basic Protocol 3.

Steps:

1. Move from the "Analyze Data" view to the "Workflow" view by clicking "Workflow" in the top panel of the Galaxy interface. The workflow view provides access to all workflow management functionality (editing, sharing, etc).

2. Load the workflow in the workflow editor.
   a. Click the triangle next to the name of the workflow that was just created.
   b. Select "edit" from the menu that appears.

3. Drag the workflow canvas (center panel) until the box labeled "Filter" is visible. Each box in the canvas represents a step of the workflow. The canvas viewport can be moved by dragging the background or by dragging the blue box in the overview panel (bottom right).

4. Click the box for the "Filter" step in the canvas, it will be outlined in blue, showing that it is the active step, and a form showing the tool parameters will appear in the right panel.

5. Modify this step to filter regions with 50 or more SNPs by entering the text "c2 >= 50" (without the quotes) in the textbox under the label "With the following condition" and clicking "Save" in
the right panel.

6. Save the changes to the workflow by clicking "Save" in the header of the center panel. It is important to save both the changes to the step and the changed step to the workflow.

Support Protocol 3.2 Running workflows with Galaxy

Once a workflow has been constructed, it can be run in the analysis view just like any other tool in Galaxy.

Materials:

• an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
• completed protocols – the workflow that was saved in Basic Protocol 3.

Steps:

1. Return to the Analyze Data view by clicking "Analyze Data" in the top panel.

2. Create a new empty history in which to store the result of running the workflow.
   a. In the header of the History panel (top-right of the Galaxy analysis interface) click "Options".
   b. In the center panel click "Create a new empty history"

3. Get exon and SNP annotations for human chromosome X from the UCSC Table Browser.
   a. Follow Steps 1 and 2 of Basic Protocol 2; however, enter "chrX" instead of "chr22".

4. Click "Workflow" at the bottom of the tool menu (left panel), then click "All workflows" in the list of options that appears.

5. In the center panel, click the name of the workflow created above in Basic Protocol 3. This will load the workflow in the center panel with prompts for parameters that need values.

6. Under "Step 1: Input Dataset" select the second item in the history (the SNPs).

7. Under "Step 2: Input Dataset" select the first item in the history (the exons).

8. Click "Run Workflow" at the bottom of the form in the center panel. A message will be displayed confirming that the workflow has been run, and the datasets for each workflow step will be added to the history (in the 'queued' state). At this point, the workflow is running, and each step will execute once the data it requires has been generated by previous steps. The box
surrounding the dataset will change color based upon its state as the steps progress: a query in
the queue will be indicated by a gray box, a running query will be yellow and a completed
query will have a green box.

**Support Protocol 3.3 Sharing workflows with Galaxy**

Galaxy allows researchers to share workflows with others. Workflows can either shared with a specific
Galaxy user, or made publicly accessible by a special link.

Materials:

- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
- completed protocols – the workflow that was created using Basic Protocol 3

Steps:

1. Move from the "Analyze Data" view to the "Workflow" view by clicking "Workflow" in the top
   panel of the Galaxy interface. This is located in the dark blue navigation banner across the top
   of the interface.

2. Click the triangle next to the name of the workflow to be shared and select "Sharing".

3. To allow others to import the workflow by following a link.
   a. Click "Enable import via link".
   b. Copy the link that is displayed.
   c. Provide the link to anyone wanting to access the workflow (e.g. via email, including in a
      publication, etc).

4. To share the workflow only with a specific user.
   a. Click "Share with another user".
   b. Enter the email address of a user to share with in the textbox.
   c. Click "Share". If the email address corresponds to another Galaxy user, they will now
      see the workflow in their workflow view.

**Basic Protocol 4. Generating workflows from scratch with Galaxy**

In addition to creating workflows from existing histories, Galaxy allows the creation of a workflow
from scratch. In this protocol a simple workflow that finds the 50 longest intervals from a dataset in a 6
A 6 column BED file is constructed. The BED format is a specialized version of the interval format discussed earlier; it contains the information required to represent a genomic position. A 6 column BED file contains the chromosome, start position in the chromosome, end position in the chromosome, name, score, and strand for a set of genomic positions.

Materials:
- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
- A Galaxy account (created by clicking “Register” in the Galaxy interface). All workflow in Galaxy requires the user to be logged in with an account.

Steps:
1. Move from the "Analyze Data" view to the "Workflow" view by clicking "Workflow" in the top panel of the Galaxy interface.
2. Create a new empty workflow.
   a. Click the "Add a new workflow" button near the top.
   b. Enter a name for the workflow in the text box under the label "Workflow name".
   c. Click "Create".
3. Load the (empty) workflow in the workflow editor.
   a. Click the triangle next to the name of the workflow that was just created.
   b. Select "edit" from the menu that appears.
4. Add an input dataset to the workflow.
   a. In the right panel, click "Inputs" from the Tools menu on the left, and then select "Input Dataset". A box (called a "node") will appear in the middle of the center panel (called the "Workflow canvas").
   b. Move the node representing the input dataset to the top-left of the editor canvas by dragging it by the title.
5. Add a step to the workflow to compute the length of each interval in the input dataset.
   a. Click "Text Manipulation" from the Tools menu on the left and then select "Compute" in the left panel.
b. Drag the newly created node up and place it to the right of "Input Dataset" (leaving some space).

Note: it's also possible to click "Layout" in the top of the center panel to automatically organize nodes in the canvas.

6. Create a connection between the input dataset and the Compute node. Outputs of a node are represented by circled arrowheads overlapping the right edge of a node, while data inputs are circled arrowheads overlapping the left edge of a node. Connections are made by dragging.
   a. Click and hold the arrowhead next to the label "output" in the "Input Dataset" node.
   b. Drag the mouse, a curve should follow the mouse pointer.
   c. Drag over the arrowhead next to the label "as a new column to" in the "Compute" node. The curve should turn green indicating that the connection is valid (datatypes are compatible).
   d. Release the mouse to make the connection.

7. Add a step to the workflow to sort the intervals by length.
   a. Click "Filter and Sort" from the Tools menu on the left and then choose "Sort" in the left panel.
   b. Position the new node to the right of the previously created nodes by dragging.
   c. Create a connection between the output labels "out_file1" of "Compute" with the input "Sort query" of "Sort".

8. Add a step to the workflow to select the longest intervals.
   a. Click "Text Manipulation" from the Tools menu on the left and then "Select first lines from a Query in the left panel.
   b. Position the new node to the right of the previously created nodes by dragging.
   c. Create a connection between the output labels "out_file1" of "Sort" with the input "from" of "Select first".

9. Edit the parameters of the "Compute" step to calculate interval length.
   a. Click the "Compute" node in the canvas. In the right panel a new form will appear. The parameters for this workflow action can be edited using the text boxes and menu
choices.

b. In the text box under "on column" enter "c3 - c2" (without the quotes) to subtract column 2 (start position) from column 3 (end position). Note that this may already be the default value for this field.

c. Click the "Save" button in the right panel to validate and save the parameters.

10. Edit the parameters of the "Sort" step to sort on the correct column.

a. Click the "Sort" node in the canvas. Its action options will appear in the right panel form.

b. In the text box under "on column" enter "7". Since the data is a 6 column BED file, the length computed in the "Compute" step will have been stored in column 7.

c. Click "Save" in the right panel to validate and save the parameters.

11. Edit the parameters of the "Select first" step to select the first 50 intervals.

a. Click the "Select first" node in the canvas. Its action options will appear in the right panel.

b. In the text box under "Select first" enter "50".

c. Click "Save" in the right panel to validate and save the parameters.

12. Click the "Save" button in the title bar header of the center workflow canvas panel to save the workflow as a whole.

13. Click "Close" in the header of the workflow canvas panel to return to the workflow list. This workflow can now be run in the same fashion as described in Support protocol 3.2.

**Basic Protocol 5. Extracting Sequences and alignments with Galaxy: A SNPs in exons example**

This protocol demonstrates how Galaxy is used to extract genomic sequences and multiple species alignments corresponding to regions of interest. It starts with the data that was generated in Basic Protocol 2, where human coding exons with high SNP counts were found. Two types of data will be extracted for these regions: the genomic sequence of each region, and pieces of a whole genome alignment between human and other species overlapping these regions. Because the whole genome alignment used here (produced by Multiz, a local aligner) is fragmented into pieces, these pieces will then be projected back onto the regions of interest (exons) to facilitate per-exon analysis of the alignments (the result is sometimes called a “pseudo-global” alignment. This protocol is a brief illustration of how easily the often tricky manipulation of these files is with Galaxy.
Materials:

- an internet accessible computer with any modern web browser (Firefox, Safari, Opera, IE).
- completed protocols – the completed and saved the history that was created by following Basic Protocol 2 and Support Protocol 2.1.

Steps:

1. Return to the main Galaxy interface by going to http://usegalaxy.org.

2. Load the history created in Basic Protocol 2:
   a. Click “Options”, located at the top right of the History Panel; History Options will appear in the middle panel.
   b. Click “List previously stored histories”; all histories associated with the current user will be displayed.
   c. Click on the history that was saved earlier as “Exons and SNPs”.

3. The history panel will refresh with the selected history, which should contain 7 steps. Dataset #7 Join two Queries on data 6 and data 1 contains the genomic coordinates of the exons of interest.

4. Extract Genomic DNA corresponding to each of the exons.
   a. In the Tools panel, click to expand the “Fetch Sequences” menu.
   b. Click the item titled “Extract Genomic DNA”; the tool’s interface will appear in the center.
   c. Select dataset #7 Join two Queries on data 6 and data 1 as the input query.
   d. Set output datatype to “FASTA”.
   e. Click “Execute”.

5. When the query finishes, a dataset containing one human sequence for each of the exons (total of 109 sequences) called “Extract Genomic DNA on data 7” is created. This dataset contains the human genomic DNA corresponding to each of the 109 exons in FASTA format, a very common format for storing multiple named sequences.

6. Extract multiple species alignment blocks for each of the human exon locations.
   a. In the Tools panel, click to expand the “Fetch Alignments” menu.
b. Click the item titled “Extract MAF blocks”; the tool’s interface will appear in the center. MAF stands for multiple alignment format, a standard format for large alignments of multiple genomes. A genome-wide local alignment of multiple genomes consists of many regions of high scoring alignment called “blocks”.

c. Select dataset #7 Join two Queries on data 6 and data 1 as the interval source.

d. Set MAF Source to “Locally Cached Alignments”.

e. Under Choose alignments, set “5-way multiZ” using the pulldown menu. This alignment contains 5 different mammals including human. Most of the locally cached multiple-species whole-genome alignments available in Galaxy were generated using the UCSC/Penn State Bioinformatics comparative genomic alignment pipeline; these original source alignments are available from the UCSC download site.

f. Click the “Select All” button to choose to extract all species.

g. Click the “Execute” button.

7. A new history item is created, #9 Extract MAF blocks on data 7, which contains the portions of the source alignment which overlap with the exon regions. For the 109 regions, 387 alignment blocks were retrieved, which is due to multiple local alignment blocks overlapping individual exons. Thus, the resulting dataset contains every local alignment block overlapping an exon, trimmed to just include the portion of the alignment that overlapped. This dataset is useful for examining the conservation of exons in aggregate, however the relationship between exons and alignments has been lost.

8. Create one projected alignment per human exon.

   a. In the Tools panel, ensure that the “Fetch Alignments” menu is still expanded.

   b. Click the item titled “Stitch MAF blocks”; the tool’s interface will appear in the center.

   c. Select dataset #7 Join two Queries on data 6 and data 1 as the input intervals.

   d. Set MAF Source to “Alignments in Your History”.

   e. Select dataset #9 Extract MAF blocks on data 7 for MAF File.

   f. Click the “Select All” button to choose to extract all species.

   g. Click “Execute”.

45
9. A new history item is created, #10 Stitch MAF blocks on data 7 and data 9, which contains one alignment block for each of the human exons, with regions where no alignment was found represented as gaps (-). Click the eye icon to examine the data in the center panel. The projected alignment is in FASTA format, suitable for downstream analysis in most phylogenetic software packages, including those available in Galaxy. For 109 regions, 545 FASTA sequences (109 regions each with sequences for 5 species) were generated in 109 alignment blocks.

**Commentary**

Galaxy successfully bridges the gap between data collection and analysis. The public Galaxy server allows researchers across the globe to perform computationally intensive, large-scale analyses with the only equipment requirement consisting of an internet-connected web browser. Users are not required to delve into the intricacies of how to execute a large collection of unrelated programs, but instead have access to a unified point and click interface. Galaxy provides both experimental biologists and their computational brethren with a framework to facilitate truly reproducible cutting edge science.

The protocols contained within this unit offer only a glimpse of possible analyses and tool functionality. The text contained here should only be considered an introduction to performing complex analysis with Galaxy. New datasets, tools, and features will be added regularly. Some new menu choices may arise or move. In addition to the screencasts that accompany these protocols; many more screencasts that demonstrate additional functionality are available at [http://galaxycast.org](http://galaxycast.org) and others will be added over time.

**Transparency and reproducibility**

Open and transparent research is essential to the process of science. Research papers cannot be published without making the protocols and generated experimental data publically available. Unfortunately, the same standards are often not applied to computational analysis. When analysis is performed within Galaxy, every detail is preserved in the “history” and can be inspected later. These histories can be shared or published, and can be reproduced (with or without modification) through the workflow system. Thus, without additional effort on the part of the user, Galaxy facilitates greater transparency and reproducibility of computational analyses.

**Collaboration**

While the scope of this unit is limited to introducing a user to performing data analysis with the public Galaxy server, Galaxy is also an excellent resource for collaborative analysis. Because it is web-based, collaborators at different locations can easily and rapidly share data and analysis. In particular,
Galaxy’s library system provides for sharing of datasets within research groups, with access controls and version histories.

Research groups that have their own collections of analysis scripts and binaries will find it worthwhile to download the open source framework, integrate their unique tools, and maintain a private server (a "Galaxy instance") for lab members to work on their projects. A local Galaxy server makes collaborations between computational and experimental researchers more efficient, since new analysis tools can be effortlessly made available to colleagues, allowing programmers to focus on method development. Although beyond the scope of this introduction to the user interface, documentation and assistance for programmers is also available on the Galaxy site.

The Galaxy Framework is easily downloaded, quickly configured and effortlessly deployed. Although written in Python, no knowledge of the Python programming language is required to deploy or maintain a personal Galaxy instance. This facilitates local development of new tools, the creation of new Galaxy instances with custom toolsets, and secure private Galaxy instances for analyzing protected data (e.g., genotype data obtained in clinical setting). To download the Galaxy Framework and view detailed installation documentation visit http://getgalaxy.org.

Help and feedback

Galaxy is under constant development and is improved based upon user suggestions. Extensive help is available in the form of screencasts as well as active public mailing lists, where both experimentalists and computationalists can request and receive advice. Discussion of feature requests is also encouraged. For links to these resources and to use Galaxy, visit http://galaxyproject.org.

Acknowledgements

A vision for Galaxy was originally articulated by Ross Hardison, who is also the major source of support and critical feedback. We thank Jim Kent and David Haussler for their continuing support and making UCSC Genome Browser uplink and connection possible. Istvan Albert pioneered initial aspects of Galaxy design. The following individuals contributed to Galaxy project at different stages: Richard Burhans, Laura Elnitski, Belinda Giardiane, Bob Harris, Jianbin He, Webb Miller, Cathy Riemer, and Yi Zhang. We thank Warren Lathe of OpenHelix for critical reading of the manuscript. Galaxy hardware is maintained by Nate Coraor. Robert Castelo, France Denoeud, Roderic Guigo, Erika Kvikstad, Julien Lagarde, and Kateryna Makova provided critical comments during software testing. This work is supported by funds provided by the Eberly College of Science, Huck Institutes of the Life Sciences at Penn State University, NSF BD&I grant 0543285 to AN and JT, NIH-NHGRI grant R01
HG004909 to AN and JT as well as funds from the Pennsylvania Department of Public Health.

**Literature cited**


Figure 1. Galaxy’s Analyze Data interface consists of four regions: the masthead (a) at the top, the tool menu (b) on the left-hand side, the work area (c) in the middle and the history panel (d) on the right. The Get Data section has been expanded in the tool menu and the Upload File tool has been selected. In the work area, a local file containing TAF1 CHiP-seq data has been chosen (Basic Protocol 1, step 1); clicking the “Execute” button will cause the data to be uploaded and appear in the history panel.
Figure 2. To change the properties of a dataset (Basic Protocol 1, step 2), click on the question mark (or the pencil icon) associated with our dataset in the history panel (a). This causes the Edit Attributes page to appear in the center panel (b) where the datatype has been changed from tabular to interval; clicking “Save” causes the page to refresh allowing additional interval-specific information to be set (c).
Figure 3. The UCSC Table browser tool has been selected and its interface (a) appears in the center panel; the refGene table has been selected and the output is marked to be sent to Galaxy (Basic Protocol 1, step 3). Once output style is specified (b), clicking “Send query to Galaxy” will create a new dataset in the history panel.
Figure 4. Selecting the Get flanks tool (Basic Protocol 1, step 4) from the Operate on Genomic Intervals Section (a) allows the creation of new data containing the region 1000 nucleotides upstream of our RefSeq genes (b).
Figure 5. The Join tool is used to create a dataset which contains the coordinates of putative promoters and TAF1 binding sites side by side (Basic Protocol 1, step 6).
Figure 6. The Build custom track tool (Basic Protocol 1, step 7) allows the user to design a custom track suitable for display at the UCSC Genome Browser (d) by progressively adding new tracks containing varying datasets (a-c).
Figure 7. A dataset containing exons and overlapping SNPs has been created (Basic Protocol 2, step 4) using the Join tool and has been displayed in the middle panel by clicking on the eye icon next to dataset 3. A red rectangle has been drawn around an exon which overlaps with 4 SNPs.
**Figure 8.** In order to create a workflow from an existing history (Basic Protocol 3), the user needs to make sure that they are logged in and then select "History Options" and click "Construct workflow from the current history". A new workflow will be populated from the current history as shown; the workflow can now be renamed and created.
Figure 9. The Workflow Editor allows users to click to add new tools and connect the output of one tool to the input of another by simple clicking and dragging. The output of the Sort tool is being connected to the Select first tool (Basic Protocol 4, step 9), as is shown by the green rope; when the mouse button is released, the connection will be created and the rope will become white.
Figure 10. Several options exist for obtaining multi-species alignments (Basic Protocol 5). The Extract MAF blocks tool (a) creates a MAF dataset which contains only the trimmed alignment blocks which overlap a specified set of intervals. The Stitch MAF blocks tool (b) creates a FASTA file which contains a single alignment block per provided interval.
CHAPTER 4

MAKING WHOLE GENOME ALIGNMENTS USABLE FOR BIOLOGISTS

To be submitted for publication as Blankenberg D, Taylor J, Nekrutenko A and the Galaxy Team. Making whole genome alignments usable for biologists.

Supplemental material for this chapter including Examples and the Sample Analysis, is found in Appendix C.
Abstract

Multiple-species whole genome alignments contain a wealth of information. Continually expanding collections of freely downloadable alignments have been made available to the scientific community, however, several issues exist which prevent experimental biologists from utilizing these important datasets. The first of these problems is the sheer size of these data; whole genome alignments can easily exceed hundreds of gigabytes. The other issues concern the format of the alignments themselves. The format of choice for multiple-species whole genome alignments is the Multiple Alignment Format (MAF). While this format is incredibly versatile, few genome analysis packages are able to accept these files as input. Even more challenging is the lack of programs that allow biologists to easily manipulate these alignments; even if current analysis software accepted MAF input, there is limited knowledge that can be gleaned by using the raw alignments in their entirety. Here we describe a set of tools designed to address these challenges. The toolset, which bridges the gap between the data and the analysis of multiple species alignments, is available through both a graphical user interface (GUI) and command-line interface. No downloads are required (but are available for advanced users) in order to use the GUI version of the tools, as they have been implemented into the web-based genome analysis platform Galaxy (http://galaxyproject.org).

Background

Data generation is no longer a challenge in life sciences as rapidly emerging next-generation technologies allow for complete sequencing of human-sized genomes in a matter of days. This implies that the next few years will see an exponential growth in the number of species with completely sequenced genomes. Unfortunately, the relative ease of sequencing does not automatically translate into the expanding of biological knowledge – it is still quite difficult to decipher the functional significance of genomic DNA. This is because the vast majority of functional studies have focused on genomes of human and model organisms. Whole genome alignments offer a solution to this challenge. By aligning newly sequenced genomes against well annotated sequences one can obtain a variety of functional, structural, and evolutionary insights. Whole genome alignments are critical for interpretation of genomic sequences: comparative and functional genomics simply cannot be done without the alignments. Yet there are still two formidable roadblocks preventing biomedical scientists, the ultimate “consumers” of whole-genome alignments, from effectively utilizing them in their research.
First, whole genome alignments are very large. For example, the existing alignment set of 28 mammalian species (Miller et al. 2007) occupies ~100 Gb of disk space and contains nearly 23,500,000 alignment blocks. Handling such massive amounts of data presents challenges even for researchers with extensive programming experience, while for most experimental biologists multiple genome alignments are simply beyond reach. Consider the following potential analysis: a researcher studies evolution of the \textit{INK4A} tumor suppressor locus in mammals and wants to obtain a 28-way alignment of its coding region. Although conceptually simple, this analysis cannot be performed using existing data mining resources. First, the 23,500,000 alignment blocks have to be filtered to find those corresponding to the coordinates of \textit{INK4A}. Next, the remaining blocks have to be trimmed at exon boundaries and all instances of overlapping blocks must be resolved. Finally, the exons need to be stitched together so that the reading frame is preserved in all 28 species.

Second, the alignments exist in a specialized format, the Multiple Alignment Format (MAF), created by multiple whole-genome alignment programs, such as Multiz/TBA (Blanchette et al. 2004), and which are available at the UCSC Table Browser (Karolchik et al. 2004). Although the MAF format is versatile and contains all information necessary for interpreting the alignments, it is not readily processed by downstream applications. This point is illustrated by continuing with the \textit{Ink4A} example from the previous paragraph. Suppose the researcher succeeded in obtaining the alignments for the locus. Now they want to estimate substitution rates from these sequences using existing molecular evolution software. As none of the popular packages such as PAML (Yang 2007) or HyPhy (Pond et al. 2005) accept the MAF format, this simple analysis will be nearly impossible for a command-line averse experimentalist.

Here we describe a set of tools designed to address these challenges. The tools, which bridge the gap between the data and the analysis of multiple species alignments, are available through both a graphical user interface (GUI) and command-line interface. No downloads are required (but are available for advanced users) in order to use the GUI version of the tools, as they have been implemented into the web-based genome analysis platform Galaxy (http://galaxyproject.org). The remainder of this manuscript will focus on using the tools via the GUI implementation. The \textit{Examples} and \textit{Sample Analysis} cited within this text are available in the accompanying supplemental material.

\textbf{MAF format in brief}

The multiple-species whole genome alignments are stored in a format known as Multiple Alignment Format (MAF). This format retains all the genomic position information for the aligning sequence
ranges. In Galaxy, each of these sequences are named, by convention, for the source species genome build and chromosome; the genome build and chromosome are separated by a period (.). For example (Figure 1), the sequence of chromosome 21 from the March 2006 human genome assembly (known as hg18) would be named “hg18.chr21”. Alignments are arranged in “blocks” separated by a blank line, where each block constitutes an individual set of sequence ranges; these ranges need not be unique as a MAF set can contain overlapping blocks. The difference between the coordinate system used in MAF alignments and genomic intervals is that, in the alignments, sequences located on the - strand have their positions listed relative to the reverse complement of the source sequence. This causes equivalent positions within a genome to be represented by a different numerical position depending upon which strand is being referenced (the notable exception would be the nucleotide located at the exact center of an odd-length sequence). This important difference in coordinate systems often becomes a roadblock to biologists trying to work with these files; luckily, these differences are resolved internally within this toolset without any additional effort or consideration on the part of users.

**Tool Descriptions**

**Alignment Extractors**

Often times using all of an alignment of entire genomes is not desired, nor practical, and one usually wants to look at specific regions of an alignment, such as those which correspond to genes or other genomic elements. Creating a subset of a genomic alignment where only desired regions are present is not a trivial matter; not only does one have to select only the alignment blocks which fall within the desired regions, but one also has to trim the alignment blocks, since the starts and ends of alignment blocks are not likely to correspond to the starts and ends of the desired regions. Trimming is not as simple as truncating a sequence string, as positions and lengths within an alignment block must also be updated for each species.

This presents the need for the first set of tools, known as alignment extractors (Figure 2). Alignment extractors allow the user to specify a list of genomic regions (chromosome, the start and end positions within the chromosome and strand for a particular species) and extract alignment blocks which correspond to the provided intervals. This requires two different types of annotations, one containing the genomic interval information, such as the BED format available directly from the UCSC table browser, and the multiple-species whole genome alignment set. Within the public web server implementation, users can extract from locally cached alignments (pre-indexed alignments saved
locally to the Galaxy server, allowing quick access) or from alignments which they provide via uploading from their computer or directly from an external data source, such as the UCSC table browser. In addition to the multiple-species alignments, there is a separate tool (which functions similarly and uses the same underlying code) that allows users to obtain locally cached pairwise whole genome alignments; extracting from the pairwise alignments has been separated from the multiple-species alignments for ease of use. Refer to Example 1 in the supplemental material for a demonstration of this tool.

**Format Converters**

**MAF-to-FASTA**

Current alignment analysis programs are unable to recognize MAF alignment files, which presents the need for the second set of tools: format converters. FASTA, often called the workhorse of sequence analysis, is the most widely accepted alignment format. Two flavors of MAF to FASTA converters (Figure 3) are provided: one which creates a multiple alignment FASTA file and another which creates a single alignment block. The converter that creates multiple FASTA blocks allows the user to exclude blocks which have missing species, allowing compatibility with analysis programs which require the presence of all species in each alignment block. The other MAF to FASTA converter concatenates the sequences for all blocks, resulting in one large alignment block, where each species has exactly one sequence. Blocks which lack a particular species will have the unaligned sequence filled with gaps; a separate filter tool, described later, can be used to remove blocks which are missing a desired species thereby preventing possibly long regions containing only gaps for the missing species. See Example 2 for a demonstration of the MAF to FASTA converter.

**MAF-to-BED**

A second type of converter, MAF to BED, allows users to extract genomic interval information from an alignment. As mentioned earlier, the coordinate system used within MAF files varies slightly from that used to represent genomic intervals. This converter takes the differences in coordinate systems into account without any additional intervention from the user. As an illustration of this tool in action, suppose there are two separate multiple alignments, one with human, mouse and rat and another with human, cow and dog. How can a researcher find the genomic positions of cow which correspond to mouse genes? The first step is to obtain genomic positions of mouse genes (i.e. from the UCSC Table Browser). Once mouse gene positions are in the history, these regions are used to extract MAF blocks from the human, mouse and rat alignment. Then the MAF to BED converter can be used to obtain the
human regions from the extracted alignment. Using these human regions, MAF blocks from the human, cow and dog alignment can be extracted. Converting this second extracted MAF set to BED will obtain genomic positions of cow which correspond to mouse gene positions – the extent of cow positions obtained is dependent upon the coverage of the alignments and it is possible that one mouse gene region can result in multiple cow regions (depending on alignment block boundaries).

**MAF-to-Interval**

Another format converter, MAF to Interval, functions almost identically to the MAF to BED converter, except the output produced is of interval type (not the more restrictive BED format) and has extra columns which contain the sequences for each species. See step 14 in the Sample Analysis for an example of this tool.

**Stitchers**

In addition to the alignment extractors and format converters, there is a type of tool which acts as a sort of hybrid of these two concepts. These tools are known as MAF “stitchers” (Figure 4), and exist in two forms: one which works on standard genomic intervals and another which works on genomic intervals having the additional fields that define protein coding exons (this format, consisting of a total of 12 columns, is known as BED12). Both versions also require a set of MAF alignments in addition to the genomic intervals; just as with the case of the extractors, users can select from either the locally cached alignments or from alignments existing within their history. Like the MAF to FASTA converter, the MAF stitching tools create FASTA output, but they do so in a much different manner. For each genomic interval that is specified, a single FASTA alignment block is created. This alignment block contains only the genomic positions which appear within the genome that the genomic intervals belong; insertions in aligning species, relative to this reference species, are discarded. Overlapping blocks, allowed within MAF alignments, must be taken into account. Conceptually, these blocks are split at the boundaries of the overlap, and the original score for the alignment block is used to determine which aligning sequences are used on a per species basis; if a sequence appears for a species in a lower scoring alignment block but not in any higher-scoring block, the sequence for that species is taken from the lower scoring alignment block. In the case when overlapping blocks have an equivalent alignment score, the sequence located last in the source MAF is, rather arbitrarily, used. When performing this operation on a gene basis, only the positions which are part of the protein coding sequence for the reference species are included in the output; individual coding exons are processed as separate genomic intervals, as above, and these exons are concatenated together to form a single FASTA alignment block.
for each requested gene. *Example 3* illustrates the difference between the MAF stitchers and the MAF to FASTA converter.

**Tools for MAF Manipulation**

*Reverse Complement*

A common manipulation that one may want to perform is simply reverse complementing a set of MAF blocks. The steps required to do this are more complicated than simply reversing the order of a sequence and substituting in the pairing nucleotide bases. Included with the information contained in a MAF block are the strand and start position; the start position is listed relative to the start of the stranded sequence (positions on the “-” strand are relative to the reverse complement of the source sequence). This causes equivalent positions within a genome to be represented by a different numerical position depending upon which strand is being referenced. These concerns are addressed transparently by the Reverse Complement a MAF file tool.

*Species Restriction*

The trend in the creation of multiple species alignments is driven by the idea that more species is better. While these large alignment sets are informative, often times one wants to work with a subset of these species (for example using a subset with high coverage). In order to make use of these publicly available alignments, but only work with a subset of species, a researcher would be required to manually remove, via a text editor or through the use of standard Linux commands, the undesired species. This process will result in an alignment which contains the proper species, but, beyond being time consuming, the alignments can contain extra gaps, such as those left over from an insertion found only in the genome of a species which was removed. The limiting of an alignment to only desired species (Figure 5) is accomplished in the toolset by simply clicking-to-select the desired species from a list of all the species found in a particular alignment set; this function is available as a separate tool or can be accomplished directly within the Extract MAF blocks tool interface. Limiting the species in a MAF file is particularly useful when one wants to visualize a subset of an alignment set in an alignment viewer such as Gmaj (http://globin.cse.psu.edu/dist/gmaj/), which is used in *Example 1.2*.

*Block Joiner*

A tool which is a more sophisticated kind of the filter species tool, is the Join MAF blocks by Species (Figure 6) tool. This tool not only removes undesired species, but it also concatenates adjacent MAF blocks. This requires the genomic positions of each species in the blocks-to-be-joined to have genomic positions that start and end directly next to each other. As more species are added to an alignment the
number of blocks increases (or stays the same), this is due to the added species having disparate genomic regions which align with this once-single block. When the species which caused a block to be divided is removed, the blocks can be rejoined together to create a single alignment block. When joining species, the strand of the sequence is important, as positions on opposing strands are considered different even if they are equivalent locations.

MAF filter

There are many times when there is no need to modify the contents of alignment blocks, but instead it is desired to apply a set of filters to remove blocks which do not match a set of conditions. Examples of this include removing blocks which lack species, removing blocks which have aligned species occurring between non-syntenous chromosomes or strands and removing block which fall outside of a desired size range (Figure 7). The Filter MAF by specified attributes tool allows this filtering. This tool allows building complex filters that are applied to each alignment block of a MAF file. Restraints can be defined on species based upon chromosome and strand. Options exist that allow blocks which contain only one species (this can happen when all but one species is removed from a block) or which are missing any of the desired species to be discarded; minimum and maximum block sizes can also be specified.

Alignment Coverage

After modifying or filtering an alignment (or without modification), it is often useful to view coverage information about the remaining blocks in reference to a particular set of intervals. The MAF Coverage Stats tool allows this. This tool requires a set of alignments and a set of genomic intervals to report upon. Like the alignment extractors, this tool can use locally cached alignments or alignments which are in a user's history. There are two different types of output, one that relays information on a per interval basis and another which provides a summary over all intervals provided. Similarly to the MAF stitching tools, only positions which exist in the genome of the supplied genomic intervals are included in the output.

Putting it all together: A real life example

Motivation

Kimchi-Sarfaty et al. (2007) describe a synonymous single nucleotide polymorphism (SNP) in the Multidrug Resistance 1 (MDRI) gene which results in a change of substrate specificity. This SNP,
described as C3435T, occurs in a codon which encodes for isoleucine and although this type of SNP is ordinarily classified as being “silent”, its presence causes a change in the substrate specificity of its protein product. This change, due to a conformational change in the protein, is purportedly caused by differences in codon usage frequencies affecting the rate of protein synthesis.

Sample Analysis

A real-life sample analysis that utilizes this toolset is now presented. In this example, multiple-species alignments of human regions containing synonymous protein coding SNPs are examined after filtering based upon the change in codon usage frequency. This analysis requires three different datasets: the genomic intervals and observed nucleotide bases for human SNPs, the genomic intervals (including coding exon information) of human genes and also a table containing the nucleotide sequence and human codon usage frequency for each amino acid. The steps to perform this analysis are included as the Sample Analysis in the supplemental material. Briefly: (1) genome-wide human SNPs and human genes are obtained from the UCSC Table browser, (2) codons are extracted from all genes and joined with the SNPs, (3) DNA sequence for these codons is obtained, (4) the DNA sequence is “mutated” according to the observed nucleotide values in the SNP data, (5) the reference codon and SNP codon sequences are joined to a table containing (a) amino acid names, (b) the encoding DNA sequences and (c) codon usage frequencies (Nakamura et al. 2000), (6) filtering is applied to (a) ensure that the amino acid encoded by the reference codon matches that of the SNP codon (removing non-synonymous SNPs) and (b) to include only those codons which have either a minor or significant change in codon usage frequency and (7) alignments are obtained for human, chimp, rhesus macaque, dog and mouse sequences corresponding to these regions. The result of the Sample Analysis steps contains the human genomic interval information and aligned sequences (from human, chimp, rhesus macaque, dog and mouse) for each of the codons containing synonymous SNPs that have a codon usage ratio which pass a specified filtering criteria.

Although the C3435T SNP (rs1045642) mentioned above did not pass the filtering criteria (designed to keep regions in the upper and lower halves of the distribution of the ratio of SNP codon / reference codon), there are two synonymous SNPs which overlap the MDRI gene, rs9332385 and rs41304191. Interestingly, both of these SNPs occur within the same codon (hg18 chr7:87067379-87067382 on the – strand; this is codon 19 within the first coding exon), rs9332385 is responsible for a change of CTG (codon usage 39.57%) → CTA (codon usage 7.14%) whereas rs41304191 causes CTG → TTG (codon usage 12.91%). Each of these codons code for the amino acid leucine, however, there is a significant difference between codon usage frequencies; even if both SNPs were to appear simultaneously (which
was not examined in this analysis), the resultant codon (TTA, codon usage 7.66%) would still be synonymous. Any effect that these “silent” mutations may have would need to be assayed experimentally. All three included primates (human:hg18, chimp:panTro2 and rhesus macaque:rheMac2) have the same reference sequence of CTG, which codes for leucine. The two included non-primate species (dog:canFam2 and mouse:mm9), however, have a reference sequence of ATG, which codes for methionine. Further investigation is required to check if these are valid codon locations within the non-human genomes (presumed codons could be affected by i.e. frameshift mutations).

Methods

Due to the size of these whole-genome alignments, searching through the entirety of their contents to locate desired blocks is not practical; this has led to the utilization of an indexing implementation which is a variation of the positional binning approach (Kent et al. 2002) that is stored on disk. Both the alignments which are available as MAF sources within Galaxy as well as the alignments appearing in a user's history are indexed. The indexes for MAFs appearing in a user's history are generated during history item creation; when an index is not available, the command-line tools will create temporary index files on-the-fly. For larger locally cached alignments, the source MAF files are compressed and an associated lookup table is created to allow the interoperability of the indexes with the compressed data; for a more in-depth explanation of this, see Miller et al. (2007).

It is worth mentioning that each of the alignment sets locally cached by the public Galaxy server are actually composed of several individual MAF files. These files tend to be split by and named for the chromosomes of the reference genome of the alignment. For example, the 28-way alignment is divided according to the human chromosome found within each alignment block; this results in 49 individual compressed MAF files, indexes and lookup tables (this number is larger than the number of human chromosomes due to the “random” chromosomal regions, several chromosomal haplotypes and the mitochondrial genome). It is not required that MAF sets be divided in this fashion, as the indexes indicate which blocks are found in a particular MAF file, but this is a common release practice of the labs creating the alignments.

The tools described here are implemented in Python, allowing seamless cross-platform compatibility. They utilize the bx-python package (http://bitbucket.org/james_taylor/bx-python/). The GUI version of this toolset has been made available through the public Galaxy server (http://usegalaxy.org) allowing
users to access not only the tools detailed above, but also additional genome analysis tools and data sources, all within one unified interface. The command-line tools and the graphical configuration files are distributed as part of the standard Galaxy distribution, available from http://getgalaxy.org. These tools and the entire Galaxy framework are released as open-source under an MIT style license – freely allowing developers to modify and distribute the applications with few restrictions. While all of the tools are designed to work directly out-of-the-box for personal Galaxy installations, additional steps are required to provide a collection of source alignments to the extract tools. These steps include obtaining source alignments (such as from UCSC), generating indexes and compressing the source MAFs (if desired); the steps required to perform these actions are outlined at the Galaxy wiki. Setting up these locally cached alignment sources is not required, as users are able to directly upload and use their own alignment files in any of the tools.

References


2. The bx-python project [http://bitbucket.org/james_taylor/bx-python/]

3. Gmaj: an Interactive Viewer for Multiple Sequence Alignments [http://globin.cse.psu.edu/dist/gmaj/]


Figures:

A typical example of a set of MAF blocks

Three MAF blocks are displayed. There are a total of five species included in this small alignment set, but one species (canFam2) is missing from the second block.
Figure 2 - Extract MAF blocks tool

This is a graphical representation of the Extract MAF blocks given a set of genomic intervals tool. Here, a single genomic interval is found to overlap with three MAF blocks in a source alignment set. MAF blocks 1 and 3 extend beyond the boundaries of the provided genomic interval and are trimmed before being included in the tool output.
Figure 3 - MAF to FASTA converting tool

This is a graphical representation of the MAF to FASTA converter tool. Two flavors of this tool exist, one which creates a one-to-one mapping of MAF blocks to FASTA blocks and another which creates a single concatenated multiple-species FASTA block, where species which are absent from a particular block have their sequence filled in with gap characters.
Figure 4 - Stitch MAF blocks tool

This is a graphical representation of the Stitch MAF blocks given a set of genomic intervals tool. Here four MAF alignment blocks are stitched into a single FASTA alignment block which contains only those positions which exist in the genome of the provided intervals.
Figure 5 - Filter MAF blocks by Species tool

This is a graphical representation of the Filter MAF blocks by Species tool. When species are removed from an alignment set, alignment columns which now contain only gaps are collapsed (excluded from the output).
Figure 6 - Join MAF blocks by Species tool

This is a graphical representation of the Join MAF blocks by Species tool. When genome 3 is removed from the alignment set, two of the three alignment blocks are joined together, resulting in only two output alignment blocks.
Figure 7 - Filter MAF blocks by Size tool

This is a graphical representation of the Filter MAF blocks by Size tool. Here all blocks which have more than 5 or less than 4 alignments columns are removed.
CHAPTER 5

A GENOME-WIDE SCAN FOR SYNONYMOUS SNPS AFFECTING PROTEIN FUNCTION

Supplemental material for this chapter including the list of candidate SNPs and genes along with the analysis steps used in Galaxy is found in Appendix D.
Introduction

Recent research has challenged the long held dogma that synonymous single nucleotide polymorphisms (SNPs), nucleotide substitutions that do not affect the resultant amino acid, have no affect on protein structure. Because these SNPs were thought to have no impact on protein structure and, therefore, no affect on function, they have been previously referred to as being “silent.” As it turns out, many of these synonymous mutations may not, in fact, be silent. Studies have shown that synonymous SNPs can have effects on the splicing, stability and structure of messenger RNA (mRNA) and also on protein folding itself. For a recent review of “silent” SNPs, Hunt et al (2009) is recommended.

In particular, Kimchi-Sarfaty et al (2007) has shown that a synonymous mutation in the Multidrug Resistance 1 gene (ABCB1 ATP-binding cassette, sub-family B (MDR/TAP), member 1), the gene coding for P-glycoprotein in humans, has an affect on substrate specificity. This particular non-silent synonymous SNP, C3435T (dbSNP130: rs1045642), codes for isoleucine. The authors hypothesize that the change in function is due to the use of a less frequent codon (ATC to ATT) causing a change in the rate of translation that manifests as a change in cotranslational folding. Using the Kimchi-Sarfaty study as an example, we wanted to find additional cases of synonymous SNPs that may potentially change protein function. To achieve this goal, we devised a simple initial approach for identification of SNPs that are similar to rs1045642. Specifically we used phyloP values (Siepel et al, 2006) computed for the human genome across placental mammals from the alignment of 44 vertebrate species. The phyloP program is part of the PHAST package and computes P-values of conservation or acceleration. The phyloP scores used here were computed for each individual alignment column using a likelihood ratio test (LRT) generating both conservation (positive values) as well as acceleration (negative values) scores; these scores reflect the conservation at a particular site and do not consider neighboring sites when computing the score for a position, with all branches of the phylogeny being considered instead of along subtrees or lineages. The LRT method compares an alternative evolutionary model having a free scale parameter to the provided neutral evolution model where P-values are computed by comparing test statistics to asymptotic chi-square null distributions, with the final phyloP score reported as –log(P-value); the absolute value of these phyloP scores is equal to a -log(P-value) under a null hypothesis of neutral evolution.

Results and Discussion

A visual investigation of the location of the non-silent synonymous MDR1 SNP (C3435T) at the UCSC
genome browser (Kent et al, 2002) reveals that this nucleotide position (a wobble base) has a drastic difference in phyloP value compared to the wobble position of its adjacent neighboring codons. For this study, phyloP scores were obtained from the UCSC download site (http://hgdownload.cse.ucsc.edu/goldenPath/hg18/phyloP44way/placentalMammals/) for placental mammals (44way alignment). The phyloP value for the position containing C3435T is -1.912, indicating accelerated evolution whereas the wobble positions of adjacent codons have values of 2.806 and 0.943, with both indicating conservation.

In this pilot study, synonymous isoleucine codons (all isoleucine codons vary only at the wobble position) were determined genome wide and the phyloP value of the SNP position was compared to the phyloP values of the nucleotides at the adjacent condons’ wobble positions. We used this as an initial approach and in the future we will attempt to compare SNPs against backgrounds of various genomic neighborhoods. This set includes 2,603 unique isoleucine codons with known synonymous SNPs that also have directly adjacent codons that are representable in interval format; isoleucine codons crossing splice points are excluded as are codons in which the neighboring codons cross or are on the other side of splice points. The phyloP score at the SNP position was subtracted from the average of the phyloP scores of the adjacent codons’ wobble positions; for C3435T this results in a delta of 3.7865 (i.e. (2.806 + 0.943) / 2 - -1.912). This calculation was performed on the set of 2,603 synonymous SNP containing isoleucine codons and a filter was applied to include only those codons that have a delta phyloP greater than or equal to that of C3435T in MDR1 (rs1045642); this results in 80 SNPs located in 80 unique codons found in 116 RefSeq gene accessions (79 unique genes by symbol) with one gene accession (NM_000778: cytochrome P450, family 4, subfamily A, polypeptide 11; CYP4A11) containing 2 reported SNPs (rs1126743 and rs17411656) which are actually the same polymorphism. A list of the 80 SNPs that are less conserved than their adjacent wobble position neighbors (in fact each these 80 candidate SNPs positions are actually undergoing accelerated evolution; i.e. the positions have negative phyloP score) and their corresponding gene symbols with RefSeq accession numbers as well as the analysis steps used in Galaxy can be found in Appendix D.

An initial literature search of the 80 SNPs, searching only by RefSNP accession ID (rs number), at PubMed (http://pubmed.gov) identified six SNPs as potentially interesting (rs1045642, rs12926669, rs3213607, rs893293, rs9682 and rs7121 found in RefSeq genes NM_000927: ABCB1 – ATP-binding cassette sub-family B (MDR/TAP) member 1, NM_001114331: CLCN7 – chloride channel 7, NM_021956: GRIK2 – glutamate receptor ionotropic kainite 2, NM_004526: MCM2 – minichromosome maintenance complex component 2, NM_006487: FBLN1 – fibulin 1 and
NM_080425: GNAS – GNAS complex locus, respectively). It should be noted that only papers which directly reference these SNPs by RefSNP ID have been located and papers which reference polymorphisms using only nomenclature (e.g. C3435T in MDR1, T393C in GNAS, etc.) require a much deeper literature search to find.

Results

Discussion of Comparison Metric

Let $\delta$ be the phyloP comparison metric, defined as $\delta = \left( \frac{\sigma_{-1} + \sigma_{+1}}{2} \right) - \sigma$, an arithmetic averaging on $\sigma$’s. Where $\sigma$ is the phyloP score at the wobble position of the isoleucine codons, $\sigma_{-1}$ is the phyloP score at the wobble position to the left of the isoleucine codon and $\sigma_{+1}$ is the phyloP score at the wobble position to the right of the isoleucine codon.

As the phyloP score is logarithmic, let $\sigma = \log(\pi)$, $\sigma_{-1} = \log(\pi_{-1})$ and $\sigma_{+1} = \log(\pi_{+1})$, where $\pi$, $\pi_{-1}$ and $\pi_{+1}$ are proportional to the P-values that are computed by comparing test statistics to asymptotic chi-square null distributions.

$\delta$ can be shown to be a ratio, a geometric averaging on $\mu$’s:

$$
\delta = \left( \frac{\sigma_{-1} + \sigma_{+1}}{2} \right) - \sigma = \frac{1}{2} \left[ (\sigma_{-1} - \sigma) + (\sigma_{+1} - \sigma) \right]
$$

$$
\delta = \frac{1}{2} \left[ \log \left( \frac{\pi_{-1}}{\pi} \right) + \log \left( \frac{\pi_{+1}}{\pi} \right) \right], \text{ where } \sigma_{-1} - \sigma = \log \left( \frac{\pi_{-1}}{\pi} \right) \quad \text{and} \quad \sigma_{+1} - \sigma = \log \left( \frac{\pi_{+1}}{\pi} \right).
$$

$$
\delta = \log \left( \frac{\pi_{-1}\pi_{+1}}{\pi^2} \right)^{\frac{1}{2}}
$$

$\delta$ was calculated for all synonymous SNP containing isoleucine codons ($\delta_i$) and a distribution (figure 1) was created. All SNPs with $\delta_i$ greater than or equal to that of the C3435T SNP located in MDR1 ($\delta_0 = 3.7865$) were subjected to a simple literature search via PubMed using RefSNP accession ID. Six SNPs were located in the initial literature search and the papers in which they appear are discussed below.
Distribution of $\partial$ phyloP

Figure 1. Distribution of $\partial$ phyloP for 2,603 synonymous isoleucine wobble positions. The red line indicates the location of the C3435T polymorphism in MDR1. The 80 SNPs found at locations with a $\partial$ phyloP value to the right of this line were subjected to a literature search via PubMed.

$rs1045642$

Papers containing references to the original SNP of interest, rs1045642 in ABCB1 (MDR1), include Scheiner et al (2009), Tripodi et al (2009), Lakhan et al (2009), Dai et al (2008), Bochud et al (2008), Levran et al (2008), Koo et al (2007), Kato et al (2008), Petrova et al (2008), Urayama et al (2007), Steinberg et al (2007), Osswald et al (2007), Frankfort et al (2006), Tate et al (2007) and Soranzo et al (2004). Not included in this list is the paper by Kimchi-Sarfaty et al (2007), which clearly demonstrates a functional change in protein function as the result of a synonymous polymorphism and was the motivation for this pilot study nor are any of the other papers cited by Kimchi-Sarfaty which have also linked rs1045642 to altered P-glycoprotein activity and reduced functionality. As this SNP is the inspiration for this investigation, it will not be discussed further, but the obvious limitations of this initial literature search should be noted.

$rs12926669$

Pettersson et al (2005) finds that the rs12926669 SNP is an independent predictor of femoral neck bone mineral density (BMD) in females. However, further investigation into this paper reveals there is possible referencing error. The authors claim that rs12926669 is a non-synonymous SNP (V418M) and that rs12926089 is a synonymous SNP (C/T), but the SNP database confirms that rs12926669 is a
synonymous (Ile T/C) SNP and rs12926089 is actually a non-synonymous SNP (Val to Met). If we ignore the labels and concentrate on the synonymous Ile codon (the details of which match those provided by the SNP database for rs12926669), the authors find that femoral neck BMD values are lower in T/C heterozygotes and C/C homozygotes when compared to TT homozygotes (p = 0.003). Due to the purported referencing error, it is not wise to make any conclusions until the study’s authors can clarify this issue, but this synonymous SNP is an interesting target for future investigation into a synonymous SNP having a phenotypic effect.

**rs3213607**

In a study examining 5 SNPs in GRIK2, Kim et al (2007) examine Korean trios and use the transmission disequilibrium test (TDT) to find that the rs3213607 SNP (C to A) in GRIK2 has preferential transmission of the C allele in autism spectrum disorder with a chi-squared of 15.110 and P-value < 0.001. As the A allele of rs3213607 is rare, the P-value for TDT using Liddell’s method is calculated, which is applicable to data with a small frequency, and still yields a significant result (P < 0.001). The authors find that rs3213607 is not in linkage disequilibrium (LD) with any of the other 4 SNPs that were examined, but that the other 4 SNPs show LD with each other. Kim et al admit that larger sample sizes are required to establish an association between rs3213607 and autism, but do discuss some possible pathway implications. The authors state that GRIK2 codes for the kianate glutamate receptor subunit 6 (GluR6) and has been shown to be a modulator of mossy fiber synaptic strength, responsible for learning and memory. It is theorized that this variation in GluR6 is a possible mechanism responsible for learning and memory deficits in autism spectrum disorder.

**Zeng et al (2006)** look at Huntington disease (HD), a neurodegenerative disorder caused by an expanded GAG trinucleotide repeat near the end of the huntingtin gene. The authors sequenced the GRIK2 gene, a known genetic modifier of HD, in eight HD afflicted individuals and the rs3213607 SNP as well as other SNPs were found in several individuals. However, their study theorizes that none of the SNPs, including rs3213607, found are the cause of the modification as there were several HD haplotypes, but rather a polymorphic TAA repeat found in the 3’ UTR of GRIK2 of each sequenced individual. It may be worthwhile to examine this polymorphic TAA repeat in the scope of Kim et al (2007).

**rs893293 and rs9682**

Two SNPs, rs893293 and rs9682, were found in a data mining paper using expressed sequence tags (EST) and HapMap data to search for allelic expression differences (Ge et al, 2005). Allele frequencies
were estimated from polymorphic ESTs from dbEST in UniGene clusters and compared to CEPH allele frequencies derived from HapMap. The deviation between EST and HapMap frequencies was used as a measure for allelic expression differences with rs893293 ($P = 0.03524$) and rs9682 ($P = 2.82E-06$) having significant \textit{in silico} allelic imbalance ($P < 0.05$), along with other SNPS from a total 976 genes. 40 genes, which are expressed in lymphoblastoid cell lines (LCL), were then selected and subjected to allelic expression assays. This assay used RNA from 60 unrelated LCLs with a cDNA library being constructed and then sequenced using PCR. Normalized heterozygote ratios of genomic DNA samples were used to create a confidence interval for each SNP and was used to determine if there was an allelic imbalance. The rs893293 SNP was one of those selected for the experimental assay, but it did not pass the experimental validation testing for differential allelic expression in LCL. The rs9682 SNP was not tested. Testing in different cell lines may have different results.

\textit{rs7121}

A study (Hirata et al, 2009) has shown there is a significant increase in the frequency of the C/C genotype of GNAS rs7121 in Japanese males afflicted with prostate cancer when compared to non-afflicted Japanese males. The authors note that G$\alpha$s has been shown to promote apoptosis through activation of the adenyl cyclase signal transduction cascade and that previous studies have shown that G$\alpha$s mRNA expression varied due to the rs7121 polymorphism with the T/T genotype having the highest level of expression. The authors note that the higher observed frequency of prostate cancer in C/C genotype individuals is consistent with a relative decrease in G$\alpha$s mRNA expression of the C/C genotype when compared to the T/T genotype, with the C/C polymorphism, therefore, having a lesser pro-apoptotic effect.

Simoni et al (2008) utilized a PubMed and EMBASE literature search looking for gene association studies relevant to Polycystic ovary syndrome (PCOS) and found that rs7121 failed to show an association with PCOS (Hahn et al, 2006), however, there was an association between the SNP and body mass index and insulin resistance in women with PCOS (Jones et al, 2007), with both of these studies looking at populations of German caucasian women. Neither of these two referenced papers appeared in the pilot PubMed search, because they do not include an RefSNP ID, but rather refer to the SNP as T393C.

Serre et al (2008) use the Illumina Allele-Specific Expression (ASE) assay to determine the extent of differential allelic expression for over 1,300 genes using more than 80 human lymphoblastoid cell lines (LCL). The authors determine the differential allelic expression for the gene containing rs7121 to be
50:50, which does not indicate any differential expression; this seems to contradict the study cited in Hirata et al (2009) but may be due to the use of different cell lines, etc.

Wing et al (2006) search for an association between a set of 41 candidate genes containing 68 SNPs and fetal growth. One of these genes is GNAS, which contains the rs7121 SNP. The authors use standardize birth weight (SBWT) as a metric to determine fetal growth restriction (FGR) with FGR defined as SBWT being less than the 10th percentile of the SBWT in the reference population. This study did not find an association between rs7121 and FGR.

Yang et al (2004) cite several papers where rs7121 has been found to be associated with hypertension, affective disorders, substance abuse and other complex diseases, but find that none of these studies has examined the assumption that rs7121 is in linkage disequilibrium with other possible contributing mutations elsewhere in the gene. This study determines that the haplotype structure of the Gαs region of the GNAS gene is composed of two haplotype blocks centered around a mutational hotspot region containing rs7121 and that rs7121 is in linkage equilibrium with SNPs found on either side of the hotspot; the authors suggest that previous studies which used rs7121 for association but did not look at rs7121 in the context of both sides of the hotspot should be reexamined.

This initial pilot study of SNPs that are similar in nature to the rs1045642 SNP (C3435T) found in MDR1 has notable limitations. For the initial study only synonymous isoleucine SNPs were investigated, but this analysis can be expanded to include other synonymous SNPs that result in the substitution of other rare codons. Due to technical limitations (i.e. the reliance on single values for start positions and end positions) only codons that do not cross splice junctions or have adjacent codons which cross splice junctions are included. The greatest limitation, however, is the depth of the initial literature search. Searching only for SNPs by RefSNP ID fails to find most of the literature relating to the set of 80 SNPs, including the paper that was the motivation for this study. A deeper literature search, utilizing common nomenclature (e.g. C3435T in MDR1/ABCB1, T393C in GNAS, etc.), should be conducted to more fully determine the effectiveness of the metric used to find SNPs similar to the C3435T SNP located in MDR1 (rs1045642), and is left as future work. Following an extensive literature search, possible functional changes to candidate proteins due to synonymous SNPs should be experimentally assayed. Despite the limitations of this pilot study, several of these “silent” SNPs have literature evidence for their association with phenotypes: femoral neck bone mineral density in females (rs12926669), autism spectrum disorder (rs3213607), purported in silico differential allelic expression (rs893293 and rs9682), prostate cancer (rs7121), body mass index and insulin resistance in women with Polycystic ovary syndrome (rs7121), hypertension (rs7121) and substance abuse (rs7121).
Acknowledgements

We thank Francesca Chiaromonte for her helpful discussion on the comparison metric used in this study.

References


Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome


Feb 29;4(2):e100006.


APPENDIX A

SUPPLEMENTAL MATERIAL FOR CHAPTER 2. GALAXY FOR COLLABORATIVE ANALYSIS OF ENCODE DATA: MAKING LARGE-SCALE ANALYSES BIOLOGIST-FRIENDLY.

SUPPLEMENT TO THE MATERIALS SECTION

Upload and processing of ENCODE data

At http://encode-upload.g2.bx.psu.edu, ENCODE members can upload a dataset in either Browser Extendible Format (BED) or Gene Feature Format (GFF). Users are required to: (1) provide a short feature name (e.g., Affymetrix_transfrag), (2) select an ENCODE analysis group (from a list of Chromatin and Chromosomes, Genes and Transcripts, Multi-species Sequences Analysis, or Transcription Regulation), and (3) paste an informative description of the dataset. Once data is submitted, this information is used to generate a filename in standard ENCODE format: group.feature.date.format (the group and feature parts are taken from the user input; data is added automatically, and the format is either BED or GFF. The Galaxy2ENCODE Upload tool ensures that both formats are available: if the user uploads data in BED format, Galaxy2ENCODE automatically generates GFF and vice versa). Next, the upload tool automatically partitions data against the standard set of protein-coding genes provided by the GENCODE group. In order of precedence, the categories of partitions are: (1) coding exons, (2) 5’UTR exons, (3) 3’UTR exons, (4) intronic proximal regions (within 5kb of an exon), (5) intergenic proximal regions (between genes and within 5kb of an exon), (6) intronic distal regions (greater than 5kb from an exon), and (7) intergenic distal regions (between genes but greater than 5kb from an exon). Finally, deposited data are immediately available through Galaxy2ENCODE (http://encode.g2.bx.psu.edu) or directly by ftp (ftp://encode:encode@g2.bx.psu.edu).

Accessing the ENCODE data from within Galaxy2ENCODE is intuitive (Screencast 3). The ENCODE Data tool lists all ENCODE analysis groups (Chromatin and Chromosomes, Genes and Transcripts, Multi-species Sequence Analysis, Transcription Regulation, and a set of combined latest datasets). After selecting a group of interest, the user is presented with a listing of all available datasets for that
particular group. The user is then able to select any number of the datasets at a time and add them to his/her history pane on the right side of Galaxy2ENCODE screen. Note that Galaxy2ENCODE stores all versions of the datasets, with the latest ones highlighted in bold font.

Finding Non-Gencode ESTs (Figure 3; Screencast 15)

First, we upload the coordinates of human ESTs and Gencode genes that fall within ENCODE regions from the UCSC Table Browser. To simplify the analysis, redundant ESTs (that appear more than once in the downloaded table) are removed from this dataset using a combination of Count, Filter, and Compare tools. Both downloaded tables are gene oriented: each line of the file represents a single EST of a single gene encoded together with all exons. By definition, a Non-Gencode ESTs must not have any exons that overlap Gencode exons. To find EST that satisfy this criterion we must first convert both tables into exon-based tables, where each line represents an exon rather than a gene. We use the Gene-BED-to-exon-BED tool to achieve this goal. Once gene tables are converted into exon-based tables, we use the Subtract tool to remove all EST exons that overlap Gencode exons (Figure 3C). We can use this information to locate those ESTs where none of the exons overlap Gencode exons in two steps. First, we count the number of exons per EST before subtraction and after subtraction. Next, we compare these two numbers. ESTs where the number of exons stays the same before and after subtraction are Non-Gencode ESTs (Figure 3D). This is done using a combination of count, join, filter, and compare tools. Once Non-Gencode ESTs are identified it is necessary to split them into three categories: intronic, intergenic, and intertwined. To find intergenic ESTs we can remove all Non-Gencode ESTs that overlap Gencode genes (Figure 3E) using the Subtract tool. To identify intertwined ESTs, we find the overlap between Non-Gencode ESTs and Gencode exons using the Intersect tool. Because we already have intergenic and intertwined ESTs we can easily define intronic ESTs by subtracting these two categories from the superset, identified in Figure 3D, using the Compare tool. The three sets can be downloaded from the http://www.bx.psu.edu/cgi-bin/trac.cgi/wiki/GenomeResearchSupp2006).

Implementation

Galaxy2ENCODE is a completely new compact implementation that combines the latest open-source technologies with ideas previously developed by our group (Giardine et al. 2005). LANGUAGE. The entire system is written in Python, a modern dynamic language that fully supports the object-oriented paradigm and is widely regarded as an optimal tool for fast and efficient software development. Thanks
to the work of hundreds of developers worldwide, programs written in Python can be run unaltered on
most platforms, from all variants of Unix to Mac OS X or Windows. Python also has one of the most
extensive default libraries of any programming language.

APPLICATION SERVER. The application server we use for Galaxy2ENCODE development is called
Paste. It is a simple and fully object-oriented web development framework that allows the creation of
“CGI-like” programs with ease. This application server has been in development for over three years
and has a number of production deployments around the world. The Paste framework provides us with
a lightweight, portable, multithreaded web server that can be started with minimal setup and allows for
a very flexible development environment. We believe that the web server is sufficiently robust to
handle any load that we can support. The fact that our services are primarily CPU-bound imposes a
limit on how many processes can be queued or run; this number will always be much smaller than the
number of simultaneous users the Paste framework can possibly serve. At the same time, the simplicity
of the server setup will be tremendously useful in allowing us to replicate the entire functionality of the
service with ease. We can easily start copies of the server on any system that supports Python. This
way we can distribute the computation load on different computers by assigning users to different
servers. The Paste server can be easily integrated with any other web server, including Apache. If a
need for load balancing, virtual hosts, or secure HTTP arises, it can be easily accomplished by running
Paste “behind” Apache and customizing the front web server to control the HTTP traffic as desired.

DATA STORAGE.

GALAXY2ENCODE deals with two kinds of data: (1) data from external sources and (2) its own
annotations attached to each of the results. Both types of data vary in size and type and are represented
as files on the file system. This is desirable since most tools operate on files and the file system is the
most efficient way of storing large objects. Data can have a number of annotations associated with
them, such as filename, display name, selection information, genome build, which user it belongs to,
etc. The database functionality is very loosely coupled to the actual storage mechanism and, therefore,
we could easily switch databases if the need arises (e.g., to MySQL or PostgreSQL; in fact, the current
production version of Galaxy uses PostgreSQL). Upon starting, GALAXY2ENCODE creates the
database-related storage if it does not already exist.

EXTERNAL DATA SOURCES.
GALAXY2ENCODE connects to external sources using a simple data-collection API. There are two main categories for data sources: (1) sources allowing synchronous access (where a data source can start streaming back data as soon as the user has finished his/her parameter selection) and (2) sources with asynchronous access that break off the connection and then later provide GALAXY2ENCODE with a URL pointing to the data. We have commitments from the developers of several data sources including UCSC, Ensembl (BioMart), and ENCODEdb to support our data access protocol.

GALAXY2ENCODE’s external data source protocols support HTTP calls and remote procedural calls via XML-RPC (for the exact protocols see the on-line documentation on the GALAXY2ENCODE wiki page).

TESTING.

In addition to manually testing each new feature that is put into GALAXY2ENCODE, we have developed an automated test suite. This serves two purposes: it checks that the older parts of GALAXY2ENCODE keep working when new changes are made, and it enables us to simulate many simultaneous users so we can examine GALAXY2ENCODE's performance under load. The suite is designed to be run constantly (on a nightly basis) and to automatically warn the developers of any errors. The suite is implemented as a Python script that submits a series of requests to GALAXY2ENCODE and monitors the responses. It contains tests for all of GALAXY2ENCODE's features: Table Browser queries, Featured Datasets, Tools, and History retrievals. Needed improvements to this facility include a more comprehensive set of feature tests, the ability to check the accuracy of the output, the ability to test the functioning of the user interface in addition to the GALAXY2ENCODE core, and the ability to determine if a request is taking too long.

SUPPLEMENTARY FIGURES
Figure S1. Galaxy interface contains four areas: the upper bar, tool frame (left column), detail frame (middle column), and history frame (right column). The upper bar contains user account controls as well as help and contact links. The left frame lists the analysis tools and data sources available to the user. The middle frame displays interfaces for tools selected by the user. The right frame (the history frame) shows data and the results of analyses performed by the user. Pictured here are three history items in different stages of completion: Green background = ready; Yellow background with rotating hourglass = computation (in this case upload) in progress; Gray with clock icon = queued (in this case it waits to be executed until history item 2 is finished uploading. This is because history item 3 will contain results of intersection between item 1 and 2). Every action by the user generates a new history item, which can then be used in subsequent analyses, downloaded, or visualized. The Galaxy history page can display results from multiple genome builds, and a single user can have multiple histories.
Figure S2. Genomic vicinity of intergenic Non-Gencode EST DR731323 (highlighted with red arrow). Exons of this EST overlap Exonify predictions and are well conserved in all mammals included in the conservation track of the UCSC Genome Browser.
Figure S3. Genomic vicinity of intergenic EST DB275065. Exons of this EST do not overlap with any protein-coding regions (including experimentally verified or computationally predicted) but coincide with region of high conservation.
Six screencasts demonstrating various protocols are available:

1. Finding promoters containing TAF1 binding sites identified from a CHiP-seq experiment.
   
   [http://galaxycast.org/cpmb-2009-1]

2. Finding Exons with the highest number of nucleotide polymorphisms.

   [http://galaxycast.org/cpmb-2009-2]

3. Saving your results and sharing data with others.

   [http://galaxycast.org/cpmb-2009-3]

4. Generating a workflow from a history.


5. Generating workflows from scratch.


6. Extracting Sequences and alignments: A SNPs in exons example.

APPENDIX C

SUPPLEMENTAL MATERIAL FOR CHAPTER 4. MAKING WHOLE GENOME ALIGNMENTS USABLE FOR BIOLOGISTS.

Examples:

Introduction to Examples

While these examples can be followed using any regions of interest, the genomic regions covered by the *ABCB1* ATP-binding cassette, sub-family B (MDR/TAP), member 1 gene (commonly known as *MDR1*) will be used within this text. To import these coordinates into Galaxy, use the *UCSC Main table browser* tool.

1. Within the UCSC table browser interface, set
   a. *clade* to 'Mammal',
   b. *genome* to 'Human',
   c. *assembly* to 'Mar. 2006',
   d. *group* to 'Genes and Gene Prediction Tracks',
   e. *track* to 'RefSeq Genes',
   f. *table* to 'refGene' and
   g. *region* to 'genome'.

2. Click the *paste list* button associated with the *identifiers (names/accessions)* setting.
   a. Type 'ABCB1' (without the quotes – this is the gene's official symbol) into the text box that appears and
   b. click submit; the output will now only include regions annotated as 'ABCB1'.

3. Make sure *Send output to Galaxy* is checked and that

4. the *output format* is set to BED and then

5. click *get output.*
6. On the new page that is displayed ensure that **one BED record per Whole Gene** is selected and then

7. click *Send query to Galaxy.*

A new dataset, *UCSC Main on Human: refGene (genome)*, appears in the history panel and will turn green when all data has been retrieved. A preview of the dataset can be expanded by clicking on the name or the entire dataset can be viewed by clicking on the eye icon. Doing either of these actions reveals a single genomic interval (in the BED12 format) that has 29 exons.

**Example 1: Alignment Extractors**

**Example 1.1:** Obtaining a subset of an alignment using the *Extract MAF blocks* tool.

Alignment blocks, from the 44-way alignment, which overlap the human *MDR1* gene are obtained.

1. Start with a set of genomic intervals, such as the coordinates of the *MDR1* gene.

2. In the tools menu, on the left hand side, click the heading *Fetch Alignments* to expand the tool section housing the multiple alignment tools.

3. The interface to the alignment extractor is accessed by clicking on the tool titled *Extract MAF blocks given a set of genomic intervals*, which appears within the newly expanded tool section.

4. Ensure that

   a. the genomic interval dataset (*UCSC Main on Human: refGene (genome)*) is selected under *Choose intervals* and

   b. *MAF Source* is set to 'Locally Cached Alignments'.

5. Under *Choose alignments*, select the 28-way multiZ alignment.

6. After the screen refreshes, the *Choose species* selector will have 28 different genome builds listed; this allows the optional exclusion of undesired species from the extracted MAF blocks. Either use the *Select All* button to select all species or select no species (in this tool, not selecting any species is the same as selecting them all) to cause the extracted MAF blocks to have all the species which appear in the original source blocks.

7. Clicking *Execute* adds a new dataset to the history, which contains 2,084 MAF blocks that fall within the genomic coordinates of the human *MDR1* gene.
Example 1.2: Viewing alignments graphically

An interactive java-based multiple sequence alignment viewer, Gmaj, has been made available as a tool under the Graph/Display Data tool section. To view the extracted alignment blocks in context with the MDRI exons, use the GMAJ Multiple Alignment Viewer tool.

1. Under Alignment File, select the extracted MAF blocks and then click Add new Annotations.

2. New configuration settings will appear:
   a. set Annotation Style to 'Galaxy',
   b. Species to 'hg18' and
   c. select the MDRI genomic intervals dataset.

3. A new history item will appear; clicking the dataset's name will reveal a button Launch GMAJ. Clicking on this button will cause the Gmaj applet to load with the exon annotated alignment blocks.

Example 2: Format Converters

Example 2.1: Creating a FASTA alignment from a MAF alignment

As an example, a single FASTA alignment block will be generated from the multiple alignments that were obtained in the extract MAF blocks example (Example 1.1). All format converters are located under a single tool heading, Convert Formats, this is a different location than the majority of the MAF toolset.

1. Clicking Convert Formats on the left hand side will reveal the tool titled MAF to FASTA

   Converts a MAF formatted file to FASTA format. Access this tool by clicking on its name.

2. Make sure that
   a. MAF file to convert is set to the MAF blocks obtained in the previous example and
   b. set Type of FASTA output to 'One Sequence per Species.'
   c. Click the Select All button under Species to extract in order to include all 28 species in the output FASTA block and then
d. click *Execute*.

3. A new dataset, having a single FASTA alignment block with 28 sequences will appear. This file is ready to be used in downstream analyses such as building a phylogenetic tree.

**Example 2.2: Building a phylogenetic tree**

A phylogenetic tree is built from the alignment of the human *MDR1* gene region using HyPhy.

1. To build a neighbor joining tree with HyPhy, select the tool section entitled *Evolution* and select the tool named *Neighbor Joining Tree Builder*.

2. Select the newly created FASTA alignment under *Fasta file* and choose a desired distance model.

3. Clicking execute on this tool creates two new datasets in the history:
   a. one is a PDF with a graphical representation of the tree and
   b. the other is a text file with the tree in the Newick format.

**Example 3: MAF Stitchers**

**Example 3.1: Stitching MAF blocks by genes**

An example of this can be best demonstrated by following the same steps used in the convert MAF to FASTA example above, but this time using the *Stitch Gene blocks given a set of coding exon intervals* tool. The stitching tools are located under the *Fetch Alignments* tool section. This tool requires the input intervals to be of the BED12 format, as exon and coding region information is encoded in additional columns not required of all interval files. All of the intervals returned from the refGene table of the UCSC table browser, of which the *MDR1* gene intervals retrieved earlier belong to, contain this information. Within the *Stitch Gene blocks* tool,

1. set *Gene BED file* to the gene interval,

2. set *MAF Source* to 'Alignments in Your History', and then

3. select the alignment dataset which was created in the extract MAF blocks example under *MAF file*.

4. Click *Select All* under *Choose species* and then
5. click *Execute*.

Alternatively, it is possible to directly use the locally cached 28-way alignment, just as was done in the extract MAF blocks example; the resultant stitched datasets will contain equivalent data. The FASTA dataset generated by this example can now be used in the HyPhy neighbor joining tree tool and the results can be compared to those obtained from following *Example 2.2*.

**Sample Analysis**

*Sample Analysis:* Investigating alignments corresponding to human codons containing synonymous SNPs.

In this example, multiple-species alignments of human regions containing synonymous protein coding single nucleotide polymorphisms (SNPs) will be examined after filtering based upon the change in codon usage frequency. This analysis requires three different datasets: the genomic intervals and observed nucleotide bases for human SNPs, the genomic intervals (including coding exon information) of human genes, and also a table containing the nucleotide sequence and human codon usage frequency for each amino acid.

1. The first two datasets can be retrieved using the *UCSC main table browser* tool.
   a. A dataset containing all human genes can be retrieved by following the steps used in the extract MAF blocks example (*Example 1.1*) to obtain the genomic interval for *MDR1*, but this time do not provide any *identifiers (names/accessions)*; be sure to *Clear* any identifiers which may already be set.
   b. To obtain a dataset containing all known human SNPs, access the UCSC table browser interface and
      i. set *clade* to 'Mammal',
      ii. *genome* to 'Human',
      iii. *assembly* to 'Mar. 2006',
      iv. *group* to 'Variation and Repeats',
      v. *track* to 'SNPs (130)',
      vi. *table* to 'snp130',

102
vii. *region* to 'genome' and

viii. *output format* to 'all fields from selected table'.

ix. Make sure *Send output to Galaxy* is checked and then click *get output.*

x. On the new page that is displayed click *Send query to Galaxy.* A new dataset, 

   *UCSC Main on Human: snp130 (genome)*, appears in the history panel and will 
   turn green when all data has been retrieved.

xi. UCSC informs Galaxy that this dataset is plain tabular data, but this dataset 
   contains all the information required of genomic intervals. This dataset can be 
   changed to genomic intervals by

   1. clicking on the pencil icon and

      a. setting *New Type* under *Change data type* to 'interval' and

      b. clicking *Save.*

   2. Make sure column assignment metadata is properly set (columns are 2, 3, 
      4, 7 and 5 for chrom, start, end, strand, and name columns, respectively).

   c. The last dataset needed, codon usage frequencies, can be obtained from the Galaxy 
      dataset library *Codon Usage Frequencies* by

      i. clicking on the *Data Library* link in the masthead of the Galaxy interface and 
         selecting the library by name.

      ii. Import the library item named *Human Codon Usage Frequencies* by filling in 
          the checkbox next to the name and clicking the *Go* button next to *Import into 
          your current history*.

      iii. Information about this file, including the original source, can be accessed by 
          clicking on the library item's name before importing.

2. Once all the datasets have been loaded into the history, it is time to start the analysis. The first 
   step is to filter out any SNPs that are not exactly one base long. This is done using the *Filter 
   data on any column using simple expressions* tool found under the tool section *Filter and Sort*:

   a. set *Filter* to the SNP dataset and

   b. enter 'c3 + 1 == c4' (without the quotes) for *With following condition*; only genomic
intervals with start position plus one equaling the end position will be included.

3. While the history has the genomic intervals for all human genes, it is really the genomic intervals for each codon that is needed. The codons can be obtained from the genomic intervals by using the tool *Gene BED To Exon/Intron/Codon BED expander* found in the *Extract Features* tool section.

   a. Set *Extract* to 'Codons' and
   b. *from* to the human genes dataset. The extended information of the BED12 format is used to generate a new dataset that contains all codons composed of nucleotides which appear adjacently within the human genome (codons crossing splice points cannot be represented with a single start and stop position and are discarded by this tool).

4. Next, the *Extract Genomic DNA using coordinates from assembled/unassembled genomes* tool, found in the *Fetch Sequences* tool section, is used to obtain the coding sequences for the human codons.

   a. Set *Fetch sequences corresponding to Query* to the codon genomic interval dataset and
   b. select 'interval' as the *Output data type*.
   c. The result is a new set of genomic intervals, exactly like the one for codons, except that a new column containing the genomic DNA sequence for each codon has been appended.

5. Now it is time to determine the codon usage frequencies for each of the codons, this is done by joining the codon sequence containing genomic intervals with the codon usage frequency table obtained earlier from the Galaxy library.

   a. Before this join can occur, it will be necessary to ensure that all the sequence characters extracted in the last step are uppercase. This is is done using the *Compute an expression on every row* tool found in the *Text Manipulation* tool section.
      i. Set *Add expression* to 'c7.upper()' (without the quotes, c7 is the column which contains the sequence information),
      ii. *as a new column to* to the dataset created in the previous step, and
      iii. leave *Round result?* as 'NO'.
   b. Now the genomic interval data is ready to be joined to the codon usage frequency data.
This is accomplished using the *Join two Queries side by side on a specified field* tool, which is found in the *Join, Subtract and Group* section.

i. Set *Join* to the genomic intervals containing all uppercase codon sequences,

ii. *using column* to 'c8' (the uppercased extracted sequences),

iii. *with* to the codon usage frequency table,

iv. *and column* to 'c2' (the column containing the nucleotide sequence for each codon) and

v. leave *Keep lines of first input that do not join with second input, Keep lines of first input that are incomplete* and *Fill empty columns* all set to 'No'.

vi. The resultant dataset now contains the genomic intervals of codons coupled with sequence and codon usage frequency information. A keen eye will notice that a handful of codons were lost in this process, this is due to the presence of ambiguous nucleotides (i.e. 'n's) in the human genome reference sequence (hg18 in this case).

6. Now that there is a dataset that contains the genomic intervals, sequences, and usage frequencies for human codons in the history, it is time to join the codons to overlapping SNPs. This is done using a different version of the join tool used earlier, named *Join the intervals of two queries side-by-side* and found under the *Operate on Genomic Intervals* tool section. This tool works by joining on overlapping genomic regions (using three columns: chromosome, start and end), whereas the previous join worked by comparing the contents of a single user-specified identifier column.

a. Set *Join* to the dataset that was created in the previous step,

b. *with* to the dataset containing human SNPs with a length of one,

c. *with min overlap* to '1' and

d. *Return* to 'Only records that are joined (INNER JOIN)'. The new dataset will have only those codons which overlap with a provided SNP.

7. The next step is to determine the sequence of the SNP containing codon. This is done using the *Mutate Codons with SNPs* tool found in the *Evolution* tool section. This tool will create a new dataset which contains the genomic intervals having different codons than were obtained from
the reference genome and which are based upon the nucleotides provided by the observed column of the SNP data.

a. Set Interval file with joined SNPs to the dataset created in the last step,

b. Codon Sequence column to 'c8',

c. SNP chromosome column to 'c17',

d. SNP start column to 'c18',

e. SNP end column to 'c19',

f. SNP strand column to 'c22' and
g. SNP observed column to 'c25'.

8. Now it is possible to join the SNP codons with the codon frequency table, as was done in step 6, but this time use 'c34' instead of 'c8'. After this join is complete, the Filter data tool is used to remove any non-synonymous codons. In the filter tool, set With following condition to 'c9 == c35' (without quotes), this will cause the new dataset to only have genomic intervals where the amino acid for the human reference genome sequence matches the amino acid of the SNP substituted sequence (removes non-synonymous codons).

9. The Compute tool is used to determine the ratio of the two codon usage frequencies by setting Add expression to 'c15 / c41'. After using the Count tool to calculate the counts of each occurrence of the ratios, the cutoffs for the upper and lower approximate 5% of the data was determined (the ratios are naturally binned into discrete values so obtaining exact 5% cutoffs was not possible). The upper and lower cutoff values are used in the Filter data tool by setting With following condition to 'c42 <= 0.360575241042 or c42 >= 3.06607911917' (without quotes).

10. Due to multiple SNPs overlapping individual codons, the filtered output should be collapsed to only include each codon once. This can be accomplished using the Count occurrences of each record tool by selecting the columns 'c1', 'c2', 'c3' and 'c6' (the chromosome, start, end and strand columns, respectively); the resultant dataset will have 5 fields: the number of times the codon was found, the chromosome, the start position, the end position and the strand. The output of this tool is tabular data, so it is necessary to change the datatype to 'interval' (as was done for the original SNP data retrieved from UCSC) and set the column assignment metadata to the proper values.
11. Now the Extract MAF blocks tool is used on this new dataset.
   a. Set MAF Source to 'Locally Cached Alignments' and
   b. Choose Alignments to the 44-way alignment.
   c. Limit the species in the extracted MAF blocks by selecting only 'hg18', 'panTro2',
      'rheMac2', 'mm9' and 'canFam2' under Choose species.

12. The Join MAF blocks by Species tool is now used to rejoin MAF blocks which may have been
    disrupted by rearrangements local to an excluded species. Use the Select All button to include
    each of the five species requested in the last step.

13. MAF blocks are again extracted, using the same genomic intervals as in step 11, but this time
    the joined-by-species MAF alignment set (created in step 12) is used as the MAF source.

14. The last three steps result in an alignment set which is not broken up by rearrangements
    occurring in species that are not of interest to this analysis and the majority of the blocks will
    have a length of 3 (matching the human codon locations). The Convert MAF to Interval tool is
    now used to create a set of genomic intervals from the extracted-joined-extracted alignment set.
    Only the genomic intervals for human are needed, so
    a. leave Select additional species blank,
    b. set Excluded blocks which have missing species to 'include blocks with missing species'
       and
    c. Remove Gap characters from sequences to 'keep gaps'.

15. The genomic intervals join tool is now used to join the SNP-codon interval dataset created
during step 9, where the ratio of codon usage frequencies was calculated with the interval file
created in the last step (from converting MAF blocks to intervals).

16. The Filter Data tool is used to make sure that the codon interval positions exactly match the
    intervals obtained from the extracted MAF blocks (rearrangements between the included
    species can cause these position to not match). Filter the joined SNP-codon-MAF dataset by
    using the expression 'c1 == c43 and c2 == c44 and c3 == c45 and c6 == c46' (without the
    quotes; this is checking that the chromosomes, starts, ends and strands are equal).

The resultant datasets contains the genomic interval information and aligned sequences for each of
the codons containing synonymous SNPs that have a condon usage ratio which met the specified
filtering criteria. Further steps that can be undertaken on this dataset, using only the already available data and the tools discussed, include determining whether the sequences of aligned species could encode synonymous or non-synonymous codons.
**APPENDIX D**

SUPPLEMENTAL MATERIAL FOR CHAPTER 5. A GENOME-WIDE SCAN FOR SYNONYMOUS SNPS AFFECTING PROTEIN FUNCTION.

*Table D.1. Candidate list of synonymous isoleucine polymorphisms.*

<table>
<thead>
<tr>
<th>RefSNP accession ID</th>
<th>RefSeq Gene Accession</th>
<th>Gene Symbol</th>
<th>delta phyloP</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs11538622</td>
<td>NM_032649</td>
<td>CNDP1</td>
<td>6.4445</td>
</tr>
<tr>
<td>rs17676820</td>
<td>NM_212557</td>
<td>AMTN</td>
<td>6.3445</td>
</tr>
<tr>
<td>rs1762114</td>
<td>NM_000350</td>
<td>ABCA4</td>
<td>6.1475</td>
</tr>
<tr>
<td>rs35922062</td>
<td>NM_001372</td>
<td>DNAH9</td>
<td>5.965</td>
</tr>
<tr>
<td>rs34625472</td>
<td>NM_014419</td>
<td>DKKL1</td>
<td>5.806</td>
</tr>
<tr>
<td>rs73857003</td>
<td>NM_017699</td>
<td>SIDT1</td>
<td>5.7205</td>
</tr>
<tr>
<td>rs11733989</td>
<td>NM_005980</td>
<td>S100P</td>
<td>5.5695</td>
</tr>
<tr>
<td>rs7121</td>
<td>NM_000516</td>
<td>GNAS</td>
<td>5.5565</td>
</tr>
<tr>
<td>rs7121</td>
<td>NM_001077488</td>
<td>GNAS</td>
<td>5.5565</td>
</tr>
<tr>
<td>rs7121</td>
<td>NM_001077489</td>
<td>GNAS</td>
<td>5.5565</td>
</tr>
<tr>
<td>rs7121</td>
<td>NM_080425</td>
<td>GNAS</td>
<td>5.5565</td>
</tr>
<tr>
<td>rs7121</td>
<td>NM_080426</td>
<td>GNAS</td>
<td>5.5565</td>
</tr>
<tr>
<td>rs16927166</td>
<td>NM_001042780</td>
<td>TNNT3</td>
<td>5.468</td>
</tr>
<tr>
<td>rs16927166</td>
<td>NM_001042781</td>
<td>TNNT3</td>
<td>5.468</td>
</tr>
<tr>
<td>rs16927166</td>
<td>NM_001042782</td>
<td>TNNT3</td>
<td>5.468</td>
</tr>
<tr>
<td>rs16927166</td>
<td>NM_006757</td>
<td>TNNT3</td>
<td>5.468</td>
</tr>
<tr>
<td>rs55845581</td>
<td>NM_003607</td>
<td>CDC42BPA</td>
<td>5.4275</td>
</tr>
<tr>
<td>rs55845581</td>
<td>NM_014826</td>
<td>CDC42BPA</td>
<td>5.4275</td>
</tr>
<tr>
<td>rs2276903</td>
<td>NM_020894</td>
<td>KIAA1530</td>
<td>5.366</td>
</tr>
<tr>
<td>rs34711743</td>
<td>NM_007353</td>
<td>GNA12</td>
<td>5.3655</td>
</tr>
<tr>
<td>rs34925006</td>
<td>NM_031476</td>
<td>CRISPLD2</td>
<td>5.351</td>
</tr>
<tr>
<td>rs58233547</td>
<td>NM_000050</td>
<td>ASS1</td>
<td>5.216</td>
</tr>
<tr>
<td>rs58233547</td>
<td>NM_054012</td>
<td>ASS1</td>
<td>5.216</td>
</tr>
<tr>
<td>rs33915456</td>
<td>NM_000343</td>
<td>SLC5A1</td>
<td>5.1115</td>
</tr>
<tr>
<td>rs9682</td>
<td>NM_001996</td>
<td>FBLN1</td>
<td>5.099</td>
</tr>
<tr>
<td>rs9682</td>
<td>NM_006485</td>
<td>FBLN1</td>
<td>5.099</td>
</tr>
<tr>
<td>rs9682</td>
<td>NM_006486</td>
<td>FBLN1</td>
<td>5.099</td>
</tr>
<tr>
<td>rs9682</td>
<td>NM_006487</td>
<td>FBLN1</td>
<td>5.099</td>
</tr>
<tr>
<td>rs41279700</td>
<td>NM_001408</td>
<td>CELSR2</td>
<td>4.9985</td>
</tr>
<tr>
<td>rs</td>
<td>NM</td>
<td>Affine</td>
<td>p-value</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>rs61747711</td>
<td>NM_003119</td>
<td>SPG7</td>
<td>4.9745</td>
</tr>
<tr>
<td>rs41287114</td>
<td>NM_002609</td>
<td>PDGFRB</td>
<td>4.9415</td>
</tr>
<tr>
<td>rs35020821</td>
<td>NM_001004434</td>
<td>SLC30A2</td>
<td>4.816</td>
</tr>
<tr>
<td>rs35020821</td>
<td>NM_032513</td>
<td>SLC30A2</td>
<td>4.816</td>
</tr>
<tr>
<td>rs12872010</td>
<td>NM_001676</td>
<td>ATP12A</td>
<td>4.809</td>
</tr>
<tr>
<td>rs4982408</td>
<td>NM_001146683</td>
<td>C14orf176</td>
<td>4.76</td>
</tr>
<tr>
<td>rs34070772</td>
<td>NM_005964</td>
<td>MYH10</td>
<td>4.753</td>
</tr>
<tr>
<td>rs17608368</td>
<td>NM_001999</td>
<td>FBN2</td>
<td>4.746</td>
</tr>
<tr>
<td>rs3212976</td>
<td>NM_001983</td>
<td>ERCC1</td>
<td>4.6975</td>
</tr>
<tr>
<td>rs3212976</td>
<td>NM_202001</td>
<td>ERCC1</td>
<td>4.6975</td>
</tr>
<tr>
<td>rs915909</td>
<td>NM_000773</td>
<td>CYP2E1</td>
<td>4.696</td>
</tr>
<tr>
<td>rs2910316</td>
<td>NM_019120</td>
<td>PCDHB8</td>
<td>4.6705</td>
</tr>
<tr>
<td>rs556386</td>
<td>NM_007344</td>
<td>TTF1</td>
<td>4.6115</td>
</tr>
<tr>
<td>rs2276778</td>
<td>NM_001031702</td>
<td>SEMA5B</td>
<td>4.577</td>
</tr>
<tr>
<td>rs17849927</td>
<td>NM_020247</td>
<td>ABCB1</td>
<td>4.5595</td>
</tr>
<tr>
<td>rs28437296</td>
<td>NM_001127487</td>
<td>FLNC</td>
<td>4.4775</td>
</tr>
<tr>
<td>rs28437296</td>
<td>NM_001458</td>
<td>FLNC</td>
<td>4.4775</td>
</tr>
<tr>
<td>rs41300169</td>
<td>NM_001105243</td>
<td>PCDH19</td>
<td>4.4245</td>
</tr>
<tr>
<td>rs41300169</td>
<td>NM_020766</td>
<td>PCDH19</td>
<td>4.4245</td>
</tr>
<tr>
<td>rs784498</td>
<td>NM_001105513</td>
<td>TTC21A</td>
<td>4.415</td>
</tr>
<tr>
<td>rs784498</td>
<td>NM_145755</td>
<td>TTC21A</td>
<td>4.415</td>
</tr>
<tr>
<td>rs61739279</td>
<td>NM_004911</td>
<td>PDIA4</td>
<td>4.406</td>
</tr>
<tr>
<td>rs8084</td>
<td>NM_019111</td>
<td>HLA-DRA</td>
<td>4.3935</td>
</tr>
<tr>
<td>rs7706444</td>
<td>NM_024867</td>
<td>SPEF2</td>
<td>4.264</td>
</tr>
<tr>
<td>rs7706444</td>
<td>NM_144722</td>
<td>SPEF2</td>
<td>4.264</td>
</tr>
<tr>
<td>rs17055101</td>
<td>NM_001829</td>
<td>CLCN3</td>
<td>4.2535</td>
</tr>
<tr>
<td>rs17055101</td>
<td>NM_173872</td>
<td>CLCN3</td>
<td>4.2535</td>
</tr>
<tr>
<td>rs45483700</td>
<td>NM_000548</td>
<td>TSC2</td>
<td>4.237</td>
</tr>
<tr>
<td>rs45483700</td>
<td>NM_001077183</td>
<td>TSC2</td>
<td>4.237</td>
</tr>
<tr>
<td>rs45483700</td>
<td>NM_001114382</td>
<td>TSC2</td>
<td>4.237</td>
</tr>
<tr>
<td>rs61737633</td>
<td>NM_015354</td>
<td>NUP188</td>
<td>4.2195</td>
</tr>
<tr>
<td>rs17806045</td>
<td>NM_020877</td>
<td>DNAH2</td>
<td>4.1995</td>
</tr>
<tr>
<td>rs17853024</td>
<td>NM_030630</td>
<td>C17orf28</td>
<td>4.1975</td>
</tr>
<tr>
<td>rs1065178</td>
<td>NM_002783</td>
<td>PSG7</td>
<td>4.1945</td>
</tr>
<tr>
<td>rs4869</td>
<td>NM_006829</td>
<td>C10orf116</td>
<td>4.1875</td>
</tr>
<tr>
<td>rs1064447</td>
<td>NM_001114</td>
<td>ADCY7</td>
<td>4.181</td>
</tr>
<tr>
<td>rs10460627</td>
<td>NM_001134771</td>
<td>SLC12A5</td>
<td>4.177</td>
</tr>
<tr>
<td>rs10460627</td>
<td>NM_020708</td>
<td>SLC12A5</td>
<td>4.177</td>
</tr>
<tr>
<td>rs11555757</td>
<td>NM_024946</td>
<td>FAM192A</td>
<td>4.165</td>
</tr>
<tr>
<td>rs34529052</td>
<td>NM_024897</td>
<td>PAQR6</td>
<td>4.155</td>
</tr>
<tr>
<td>rs34529052</td>
<td>NM_198406</td>
<td>PAQR6</td>
<td>4.155</td>
</tr>
<tr>
<td>rs2288509</td>
<td>NM_004706</td>
<td>ARHGEF1</td>
<td>4.144</td>
</tr>
<tr>
<td>rs2288509</td>
<td>NM_198977</td>
<td>ARHGEF1</td>
<td>4.144</td>
</tr>
<tr>
<td>rs2288509</td>
<td>NM_199002</td>
<td>ARHGEF1</td>
<td>4.144</td>
</tr>
<tr>
<td>rs34596202</td>
<td>NM_001142569</td>
<td>Clorfl06</td>
<td>4.133</td>
</tr>
<tr>
<td>rs34596202</td>
<td>NM_018265</td>
<td>Clorfl06</td>
<td>4.133</td>
</tr>
<tr>
<td>rs2289657</td>
<td>NM_001018064</td>
<td>NTRK2</td>
<td>4.1095</td>
</tr>
<tr>
<td>rs2289657</td>
<td>NM_006180</td>
<td>NTRK2</td>
<td>4.1095</td>
</tr>
<tr>
<td>rs34573899</td>
<td>NM_130386</td>
<td>COLEC12</td>
<td>4.0885</td>
</tr>
<tr>
<td>rs3846848</td>
<td>NM_006994</td>
<td>BTN3A3</td>
<td>4.0875</td>
</tr>
<tr>
<td>rs</td>
<td>NM_ or</td>
<td>Gene Name</td>
<td>p-value</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>rs3846848</td>
<td>NM_197974</td>
<td>BTN3A3</td>
<td>4.0875</td>
</tr>
<tr>
<td>rs848578</td>
<td>NM_001080830</td>
<td>PRAMEF12</td>
<td>4.087</td>
</tr>
<tr>
<td>rs2304035</td>
<td>NM_003062</td>
<td>SLIT3</td>
<td>4.0825</td>
</tr>
<tr>
<td>rs11078458</td>
<td>NM_145068</td>
<td>TRPV3</td>
<td>4.054</td>
</tr>
<tr>
<td>rs1043398</td>
<td>NM_006445</td>
<td>PRPF8</td>
<td>4.0115</td>
</tr>
<tr>
<td>rs12926669</td>
<td>NM_001114331</td>
<td>CLCN7</td>
<td>4.0095</td>
</tr>
<tr>
<td>rs12926669</td>
<td>NM_001287</td>
<td>CLCN7</td>
<td>4.0095</td>
</tr>
<tr>
<td>rs1774311</td>
<td>NM_032821</td>
<td>HYDIN</td>
<td>3.9745</td>
</tr>
<tr>
<td>rs11741186</td>
<td>NM_003966</td>
<td>SEMA5A</td>
<td>3.962</td>
</tr>
<tr>
<td>rs7517033</td>
<td>NM_015164</td>
<td>PLEKHM2</td>
<td>3.9595</td>
</tr>
<tr>
<td>rs6661253</td>
<td>NM_015047</td>
<td>KIAA0090</td>
<td>3.928</td>
</tr>
<tr>
<td>rs74000572</td>
<td>NM_173821</td>
<td>C2orf85</td>
<td>3.9245</td>
</tr>
<tr>
<td>rs3747504</td>
<td>NM_020893</td>
<td>KIAA1529</td>
<td>3.8885</td>
</tr>
<tr>
<td>rs1126743</td>
<td>NM_000778</td>
<td>CYP4A11</td>
<td>3.8865</td>
</tr>
<tr>
<td>rs17411656</td>
<td>NM_000778</td>
<td>CYP4A11</td>
<td>3.8865</td>
</tr>
<tr>
<td>rs3213607</td>
<td>NM_021956</td>
<td>GRIK2</td>
<td>3.8685</td>
</tr>
<tr>
<td>rs3213607</td>
<td>NM_175768</td>
<td>GRIK2</td>
<td>3.8685</td>
</tr>
<tr>
<td>rs17880183</td>
<td>NM_001012271</td>
<td>BIRC5</td>
<td>3.861</td>
</tr>
<tr>
<td>rs17880183</td>
<td>NM_001168</td>
<td>BIRC5</td>
<td>3.861</td>
</tr>
<tr>
<td>rs3730284</td>
<td>NM_000679</td>
<td>ADRA1B</td>
<td>3.8415</td>
</tr>
<tr>
<td>rs6420202</td>
<td>NM_003970</td>
<td>MYOM2</td>
<td>3.835</td>
</tr>
<tr>
<td>rs2228094</td>
<td>NM_000692</td>
<td>ALDH1B1</td>
<td>3.8275</td>
</tr>
<tr>
<td>rs6928338</td>
<td>NM_001003760</td>
<td>KLHL31</td>
<td>3.827</td>
</tr>
<tr>
<td>rs6729871</td>
<td>NM_001145909</td>
<td>TANC1</td>
<td>3.8225</td>
</tr>
<tr>
<td>rs6729871</td>
<td>NM_033394</td>
<td>TANC1</td>
<td>3.8225</td>
</tr>
<tr>
<td>rs5757761</td>
<td>NM_001003406</td>
<td>CACNA1I</td>
<td>3.8215</td>
</tr>
<tr>
<td>rs5757761</td>
<td>NM_021096</td>
<td>CACNA1I</td>
<td>3.8215</td>
</tr>
<tr>
<td>rs893293</td>
<td>NM_004526</td>
<td>MCM2</td>
<td>3.8195</td>
</tr>
<tr>
<td>rs34009998</td>
<td>NM_032588</td>
<td>TRIM63</td>
<td>3.814</td>
</tr>
<tr>
<td>rs2849793</td>
<td>NM_007028</td>
<td>TRIM31</td>
<td>3.8085</td>
</tr>
<tr>
<td>rs61731188</td>
<td>NM_023944</td>
<td>CYP4F12</td>
<td>3.802</td>
</tr>
<tr>
<td>rs2806455</td>
<td>NM_001012957</td>
<td>DISC1</td>
<td>3.8</td>
</tr>
<tr>
<td>rs2806455</td>
<td>NM_001164537</td>
<td>DISC1</td>
<td>3.8</td>
</tr>
<tr>
<td>rs2806455</td>
<td>NM_001164538</td>
<td>DISC1</td>
<td>3.8</td>
</tr>
<tr>
<td>rs2806455</td>
<td>NM_001164540</td>
<td>DISC1</td>
<td>3.8</td>
</tr>
<tr>
<td>rs2806455</td>
<td>NM_018662</td>
<td>DISC1</td>
<td>3.8</td>
</tr>
<tr>
<td>rs11544854</td>
<td>NM_003359</td>
<td>UGDH</td>
<td>3.7895</td>
</tr>
<tr>
<td>rs1045642</td>
<td>NM_00927</td>
<td>ABCB1</td>
<td>3.7865</td>
</tr>
</tbody>
</table>

**Analysis Steps**

The analysis was carried out in Galaxy as follows:

1.) Four initial datasets are required. A dataset containing the positions of RefSeq genes, a dataset containing SNP positions with observed nucleotides and a dataset containing phyloP scores for placental mammals from the 44way alignment, each for the entirety of the human genome.
Additionally, a dataset containing a list of all possible codons and their resultant amino acids will be used.

a. RefSeq genes can be loaded directly into Galaxy from the UCSC table browser by

i. going to the tool menu and expanding Get Data and

ii. clicking on the UCSC Main table browser tool.

iii. In the UCSC table browser interface that appears, make sure that

1. clade is set to “Mammal”,
2. genome is set to “Human”,
3. assembly is set to “Mar. 2006” (hg18),
4. group is set to “Genes and Gene Prediction Tracks”,
5. track is set to “RefSeq Genes”,
6. table is set to “refGene”,
7. region is set to “genome”,
8. output format is set to “BED – browser extensible data” and
9. “Send output to Galaxy” is selected.

iv. Click “get output”.

v. With Create one BED record per set to “Whole Gene”, click “Send query to Galaxy”.

vi. A new dataset appears in the user’s history that contains a list of all human RefSeq genes.

b. SNP positions can be loaded into Galaxy directly from the UCSC table browser by accessing the UCSC Main table browser tool as above.

i. But this time make sure that

1. clade is set to “Mammal”,
2. genome is set to “Human”,
3. assembly is set to “Mar. 2006” (hg18),
4. **group** is set to “Variation and Repeats”,

5. **track** is set to “SNPs (130)”,

6. **table** is set to “snp130”,

7. **region** is set to “genome”,

8. **output format** is set to “selected fields from primary and related tables” and

9. “Send output to Galaxy” is selected.

ii. Click “get output”.

iii. Select the following fields:

   1. “chrom”,
   2. “chromStart”,
   3. “chromEnd”,
   4. “name”,
   5. “strand” and
   6. “observed”.

iv. Click “done with selections” and then

v. Click “Send query to Galaxy”.

vi. A new dataset appears in the user’s history that contains a list of all known human SNPs.

c. The dataset containing phyloP base position scores cannot be obtained directly using the table browser, as the table browser aggregates these scores over a set of windows.

   i. To obtain a list of genome wide phyloP scores, the UCSC bulk download site must be used.

   1. Go to

      http://hgdownload.cse.ucsc.edu/goldenPath/hg18/phyloP44way/placental Mammals/

      a. and copy and paste the URLs for each of the chromosomes (e.g.
http://hgdownload.cse.ucsc.edu/goldenPath/hg18/phyloP44way/placentalMammals/chr1.phyloP44way.placental.wigFix.gz) into the Upload File from your computer tool’s URL/Text box, located under the Get Data section.

i. Set File Format to “wig”,

ii. Genome to “Human Mar. 2006 (hg18)” and

iii. click “Execute”.

2. A new dataset for each URL will appear in the history. To create one file containing all the genome wide data, use the Concatenate queries tail-to-head tool located in the Text Manipulation tool section.

   a. Select each of the datasets containing phyloP scores that were just uploaded, clicking “Add new Query” as needed to create enough dataset select lists to allow selection of all the score datasets.

   b. Clicking “Execute” will create one dataset that contains genome wide phyloP scores.

3. Convert these wiggle scores into intervals, because these scores will later be matched to base positions using the Join on Genomic intervals tool.

   a. Under Get Genomic Scores tool section, select the Wiggle-to-Interval converter tool:

      b. make sure that the file containing the genome wide phyloP scores is selected and

      c. click “Execute”.

   ii. or a dataset containing the phyloP scores which overlap exon positions can be obtained from Galaxy’s Data Libraries.

   d. A dataset containing a list of codons and the corresponding amino acid three-letter code (64 rows with 2 columns per row) can be obtained from Galaxy’s Data Libraries or you can upload your own.

2.) Obtaining codons from RefSeq genes.

   a. Expand the Extract Features tool section.
b. Click on the tool titled *Gene BED to Exon/Intron/Codon Bed expander*, the tool interface appears in the middle panel.

c. Under **Extract**, select “Codons”

d. Make sure **from** is set to the dataset containing RefSeq genes and

e. click “Execute”.

f. A new dataset will appear in the history, which will contain all codons from all RefSeq genes.

3.) Create a list of unique codons.

a. Expand the tool section titled **Statistics** and

b. click on the tool *Count occurrences of each record*, the tool interface appears in the middle panel.

c. Make sure that the dataset containing the RefSeq codons is selected and

d. set **Count occurrences of values in column(s)** to “c1”, “c2”, “c3” and a “c6” for chromosome, start position, end position and strand, respectively and then

e. click “Execute”. This new dataset contains a list of unique codons, as well as the number of times that this codon was found.

f. Use the *Cut columns from a table* tool, located under the **Text Manipulation** tool section,

g. make sure that **Delimited by** is set to “Tab”,

h. **From** is set the unique list of codons,

i. enter “c2,c3,c4,c5” under **Cut columns** and

j. click “Execute”. This dataset now contains a unique set of the chromosome, start, end and strand information for codons in RefSeq genes.

4.) Determine the set of codons that have neighboring codons. To do this, a set of labels will be created in the form of Chromosome_Start_End_Strand.

a. Expand the **Text Manipulation** tool section and

b. click on the **Compute an expression on every row** tool,

c. set **Add expression** to “str( c1 ) + ‘’ + str( c2 ) + ‘’ + str( c3 ) + ‘’ + str( c4 )” and
d. click “Execute”. This new dataset contains a unique label for each codon.

e. Now create labels for the neighboring codons.
   i. click on the *Compute an expression on every row* tool,
   ii. set **Add expression** to “str( c1 ) + ‘’ + str( c2 - 3 ) + ‘’ + str( c3 - 3) + ‘’ + str( c4 )” and
   iii. click “Execute”. This new dataset contains a unique label for each putative codon to the left as well as the actual codon.
   iv. click on the *Compute an expression on every row* tool,
   v. set **Add expression** to “str( c1 ) + ‘’ + str( c2 + 3 ) + ‘’ + str( c3 + 3 ) + ‘’ + str( c4 )” and
   vi. click “Execute”. This new dataset contains a unique label for each putative codon to the right as well as the left and the actual codon.

f. Using the *Join two Queries side by side on a specified field* tool found under *Join, Subtract and Group* twice to determine the unique set of codons that have neighboring codons.

   i. Set **Join** to the dataset containing labels for the codon and the codons to the left and right of the codon and
   ii. set **using column** to “c6” (the left label “str( c1 ) + ‘’ + str( c2 - 3 ) + ‘’ + str( c3 - 3) + ‘’ + str( c4 )”)
   iii. set **with** to the dataset containing only the label for the actual codon,
   iv. set and **column** to “c5” and
   v. click “Execute”. This new dataset contains only unique codons with a codon to the left.
   vi. Access the *Join two Queries side by side on a specified field* tool again.
   vii. Set **Join** to the dataset that was just created,
   viii. set **using column** to “c7” (the left label “str( c1 ) + ‘’ + str( c2 + 3 ) + ‘’ + str( c3 + 3) + ‘’ + str( c4 )”)
   ix. set **with** to the dataset containing only the label for the actual codon,
x. set and **column** to “c5” and

xi. click “Execute”. This new dataset contains only unique codons with a codon to the left and to the right.

xii. Using the *Cut columns from a table*, found in the *Text Manipulation* tool section, on the newest dataset,

xiii. set **Cut columns** to “c1,c2,c3,c4” and

xiv. click “Execute”.

xv. Click the *pencil icon* and **change datatype** to “interval” and set metadata to:

1. **Chrom column** to “1”,

2. **Start column** to “2”,

3. **End column** to “3” and

4. **Strand column** to “4”.

xvi. This new dataset now contains only unique codons having neighboring codons.

5.) Use the *Join the intervals of two queries side-by-side* tool found under the *Operate on Genomic Intervals* section to find the overlap between the codons and SNPs:

   a. set **Join** to the last created dataset containing codons with neighboring codons,

   b. with to the dataset containing the SNP data and

   c. click “Execute”. This new dataset contains codons with neighboring codons that have overlapping SNPs.

6.) Use the *Extract Genomic DNA using coordinates from assembled/unassembled genomes* tool located in the *Fetch Sequences* tool section to obtain the sequence data for the codons:

   a. Set **Fetch sequences corresponding to Query** to the dataset containing codons with neighboring codons that have overlapping SNPs,

   b. set **Output data type** to “Interval” and

   c. click “Execute”.

7.) Use the *Compute an expression on every row tool* located in the *Text Manipulation* tool section to make sure that sequences are using only uppercase characters (this is necessary for a later
step where the sequences are matched to amino acids):

a. set **Add expression** to “c11.upper()”,
b. set **as a new column to** to the interval file containing sequences created in the last step and
c. click “Execute”.

8.) Use the *Mutate Codons with SNPs* tool found in the *Evolution* tool section to calculate the SNP containing codon:

a. set **Interval file with joined SNPs** to the dataset created in the last step,
b. set **Codon Sequence column** to “c12”,
c. set **SNP chromosome column** to “c5”,
d. set **SNP start column** to “c6”,
e. set **SNP end column** to “c7”,
f. set **SNP strand column** to “c9”,
g. set **SNP observed column** to “c10” and
h. click “Execute”. This new file contains the original codon as well as the SNP codon.

9.) Using the *Join two Queries side by side on a specified field tool* found under *Join, Subtract and Group* twice to determine the amino acid for the reference and SNP containing codon:

a. Set **Join** to the dataset containing sequences for the reference codon and the SNP containing codons and
b. set **using column** to “c12” (containing the uppercased reference sequence),
c. set **with** to the dataset containing the list of amino acids and corresponding codons,
d. set **and column** to “c2” (column containing the codon) and
e. click “Execute”.
f. Reload the **Join** tool and set **Join** to the dataset containing sequences for the reference codon, the SNP containing codons and the amino acid of the reference codon and
g. set **using column** to “c13” (containing the SNP codon sequence),
h. set **with** to the dataset containing the list of amino acids and corresponding codons,
i. set **and column** to “c2” (column containing the codon) and
j. click “Execute”. This new dataset contains the list of reference codons with SNP codons and their corresponding amino acids

10.) Use the *Filter data on any column using simple expressions* tool located in the **Filter and Sort** tool section to obtain a list of synonymous isoleucine codons:
   a. set **Filter** to the dataset created in the last step,
   b. set **With the following condition** to “c14 == “Ile” and c14 == c16” and
   c. click “Execute”.

11.) Use the *Join the intervals of two queries side-by-side tool* found under the **Operate on Genomic Intervals** section to determine the phyloP score for each of the SNP locations:
   a. First, click the *pencil icon* to edit metadata and set the interval region to the SNP position:
      i. **Chrom column** to “5”,
      ii. **Start column** to “6”,
      iii. **End column** to “7” and
      iv. **Strand column** to “9”.
   b. Set **Join** to the dataset containing phyloP scores,
   c. **with** to the dataset created in the last step
   d. click “Execute”.

12.) Use the *Compute an expression on every row tool* located in the **Text Manipulation** tool section to determine the interval locations of the wobble positions of neighboring codons. This tool will be used 4 times in a row:
   a. Iteratively, set Add expression to
      i. “c11-3”
      ii. “c12-3”
      iii. “c11+3”
      iv. “c12+3”
b. set **as a new column** to the dataset created in the last (iterative) step and
c. set **Round result?** to “YES”
d. click “Execute” each time. The result of the last *Compute* tool run will contain the
interval starts and ends for the wobble positions to left and to the right.

13.) Click the *pencil icon* and set metadata to the positions of the wobble position to the left:
a. set **Chrom column** to “10”,
b. set **Start column** to “23”,
c. set **End column** to “24” and
d. set **Strand column** to “14”.

14.) Use the *Join the intervals of two queries side-by-side tool* found under the *Operate on Genomic Intervals* section to determine the phyloP score for the wobble position to the left:
a. set **Join** to the dataset containing phyloP scores,
b. **with** to the dataset created in the last step
c. click “Execute”.

15.) Click the *pencil icon* of the new dataset and set metadata to the wobble position to the right:
a. set **Chrom column** to “15”,
b. set **Start column** to “30”,
c. set **End column** to “31” and
d. set **Strand column** to “19”.

16.) Use the *Join the intervals of two queries side-by-side tool* found under the *Operate on Genomic Intervals* section to determine the phyloP score for the wobble position to the right:
a. set **Join** to the dataset containing phyloP scores,
b. **with** to the dataset created in the last step
c. click “Execute”.

17.) Use the *Cut columns from a table tool*, located under the *Text Manipulation* tool section:
a. make sure that **Delimited by** is set to “Tab”,

b. **From** is set the dataset containing the phyloP scores for the SNP position and the two neighboring positions,

c. enter “c16,c17,c18,c19,c20,c21,c22,c23,c24,c25,c27,c28,c29,c15,c10,c5” under **Cut columns** and

d. click “Execute”. This dataset now contains the codon interval and sequence, the SNP information and the phyloP scores for all 3 wobble positions.

18.) Use the **Compute an expression on every row** tool located in the **Text Manipulation** tool section to calculate the metric of phyloP scores (delta phyloP), (left wobble + right wobble) / 2 – SNP wobble:

a. set **Add expression** to “((c15 + c16) / 2) - c14”,

b. set **as a new column** to the file created in the last step and

c. click “Execute”.

19.) Use the **Sort data in ascending or descending order** tool found in the **Join, Subtract and Group** tool section to order the regions by delta phyloP:

a. set **Sort Query** to the dataset containing the delta phyloP scores,

b. set **on column** to “c17”, the column containing the delta phyloP scores,

c. set **with flavor** to “Numerical sort”,

d. set **everything in** to “Descending order” and

e. click “Execute”. This new dataset contains a list of synonymous SNP containing isoleucine codons ordered by delta phyloP.
VITA

Daniel James Blankenberg

EDUCATION
2004 – 2009  PhD Candidate Biochemistry, Microbiology and Molecular Biology
The Pennsylvania State University
University Park, PA

2000 – 2004  B.S. Biochemistry & Molecular Biology and Computer Science
Gettysburg College
Gettysburg, PA

PUBLICATIONS


