MAVEN: A TOOL FOR VISUALIZATION AND FUNCTIONAL ANALYSIS OF
GENOME-WIDE ASSOCIATION STUDIES

by

KANCHANA NARAYANAN

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science

Thesis Adviser: Dr. Jing Li

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

May, 2010
We hereby approve the thesis/dissertation of

Kanchana Narayanan

candidate for the Master of Science degree *

(signed) Jing Li

(Chair of the committee)

Gultekin Özsoyoglu

Guo-Qiang Zhang

(date) March 15, 2010

*We also certify that written approval has been obtained for any proprietary material contained therein.
Dedicated to my Mother for loving and supporting me all through my life!
# Table of Contents

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

Abstract ......................................................................................................................................... viii

Chapter 1: Introduction .............................................................................................................. 1
  1.1 Related Work ................................................................................................................... 3

Chapter 2: Functionality ............................................................................................................ 6
  2.1 User Management ............................................................................................................ 7
  2.2 Data Management ............................................................................................................ 9
  2.3 Filtering Capabilities ...................................................................................................... 13
  2.4 Data Visualization and Download ................................................................................. 16
  2.5 Functional annotation of SNPs ...................................................................................... 22
  2.6 Manhattan Plots ............................................................................................................. 29

Chapter 3: Database Design ..................................................................................................... 31

Chapter 4: Implementation ...................................................................................................... 41
  4.1 MAVEN Architecture and Design ................................................................................. 41
  4.2 Object Model ................................................................................................................. 43
    4.2.1 Business Objects .................................................................................................... 44
    4.2.2 ActionForms .......................................................................................................... 50
  4.3 MAVEN's Authorization framework ............................................................................. 55
  4.4 Implementation Classes ................................................................................................. 56
  4.5 Search Query Implementation ....................................................................................... 66
List of Figures

Figure 1: Login Page........................................................................................................................ 7
Figure 2: Home Page ........................................................................................................................ 8
Figure 3: User Registration page ................................................................. ................................. 8
Figure 4: Manage Study Data page.................................................................................................. 9
Figure 5: Upload Page ................................................................................................................... 10
Figure 6: Upload multiple phenotypes and/or multiple statistics......................... ........................ 11
Figure 7: Upload SNPs page.......................................................................................................... 12
Figure 8: Search Page .................................................................................................................... 13
Figure 9: Search Results Screen .................................................................................................... 17
Figure 10: Study Details Section ................................................................................................... 18
Figure 11: Filter Criteria Section ................................................................................................... 18
Figure 12: Search result table sorted by PVALUE ........................................................................ 19
Figure 13: Search returning more than 500 records with pagination ......................... ........................ 19
Figure 14: Graph with more than one chromosome; tool tip for highlighted point .......... 20
Figure 15: Graph showing results for a single chromosome...................................................... 21
Figure 16: Download Data feature................................................................................................. 22
Figure 17: SNP Summary data section .......................................................................................... 23
Figure 18: Transcript Section ......................................................................................................... 24
Figure 19: Find Neighboring Genes section .................................................................................. 25
Figure 20: Neighboring Gene Info ................................................................................................. 25
Figure 21: Find SNPs in LD ............................................................................................................. 26
Figure 22: LD SNPs Information.................................................................................................... 27
Figure 23: Genome Browser link section ...................................................................................... 27
Figure 24: Gene related Links ........................................................................................................ 29
Figure 25: Manhattan Plot ................................................................. 30
Figure 26: User Data Table ................................................................. 32
Figure 27: Study Meta data tables ....................................................... 33
Figure 28: Study Tables ................................................................. 34
Figure 29: Chromosome Table .......................................................... 34
Figure 30: SNP Meta data table ........................................................ 35
Figure 31: Gene Data table ............................................................... 36
Figure 32: KEGG Data Tables ............................................................ 37
Figure 33: Transcript data table ....................................................... 38
Figure 34: LD Data Tables ............................................................... 39
Figure 35: HPRD Data table ............................................................... 39
Figure 36: Coding Function table ...................................................... 40
Figure 37: MAVEN Architecture ....................................................... 43
Figure 38: Study Business Object ..................................................... 45
Figure 39: Base Action form .............................................................. 51
Figure 40: Data Parser Tool ............................................................... 80
Figure 41: Visitors Overview ............................................................. 81
Figure 42: Map Overlay of Visitors .................................................. 82
MAVEN: a tool for Visualization and Functional Analysis of Genome-Wide Association Studies

Abstract

by

KANCHANA NARAYANAN

Genome-wide association studies (GWAS) provide a new and powerful approach to investigate the effect of inherited genetic variation on risks of complex diseases. With recent advances in genotyping technology, GWAS are now becoming a reality. Within the past two years, scientists have successfully replicated genetic risks of several complex diseases including cancers, obesity, and type 2 diabetes using GWAS. And more data from GWAS are expected at an accelerated rate. However, management, analysis, visualization, and interpretation of GWAS data are particularly difficult, primarily because GWAS may consist of hundreds of thousands SNPs (single nucleotide polymorphisms) from thousands of individuals. This thesis describes the functionality and implementation of a web application tool, MAVEN for Management, Analysis, Visualization and rEsults shariNg of GWAS data. MAVEN seamlessly integrates users own data with publicly available databases like NCBI, to provide several functional annotations of SNPs and genes that are relevant to their research interests.
Chapter 1: Introduction

Most common diseases such as neurodegenerative diseases (e.g., Alzheimer's disease (AD) and Parkinson's disease), cardiovascular diseases, various cancers, diabetes, and osteoporoses are complex diseases that involve multiple genes, their interactions, environmental factors, and gene-environment interactions. Two most commonly used approaches for disease gene mapping are linkage and association studies. Linkage studies examine the co-segregation patterns of genetic markers and the trait of interest within families, and establish linkage of disease genes and markers through recombination fractions. Association studies evaluate correlations between genetic polymorphisms and phenotypes at the population level with the aim to identify genetic loci that are in linkage disequilibrium with causal variants. Recently, genome-wide association studies (GWAS) have shown to be powerful in investigating the effect of inherited genetic variations on risks of complex diseases (W.T.C.C Consortium 2007) (McCarthy, et al. 2008). Within the last two years, scientists have successfully replicated genetic risks of many complex diseases such as cancers, heart diseases and diabetes using GWAS. As of this writing, there have been ~502 published GWAS studies with ~2400 SNPs that have been identified as being associated with various diseases/conditions (Hindorff, et al. 2009).

Though GWAS have shown some initial success, they also bring tremendous challenges to the community, not only in computations (i.e. hundreds of thousands of SNPs) and statistical analyses (e.g. multiple testing) but also in data management, access control, results visualization and sharing, integration with existing public resources (e.g. dbSNP, PubMed databases at NCBI) and interpretation of results. In this thesis, we
present a web application tool named MAVEN - for Management, Analysis, Visualization and results sharing of GWA data, implemented using cutting edge technologies.

The current version of MAVEN (version 1.1) allows users to directly upload their own GWAS results to be stored in a relational database, and to search their data for significant SNPs using powerful and user-friendly filtering capabilities. The search results will be presented in tabular as well as graphical formats. MAVEN also integrates information about SNPs and genes directly from NCBI, KEGG, OMIM and HapMap databases and provides direct links to detailed information of each selected SNP and/or gene, which will allow researchers to easily narrow down interesting candidate disease genes for further functional analysis. Hence, MAVEN can conveniently assist genetic epidemiologists to visualize and analyze their GWAS results and to share results within their own group or with colleagues across the world as MAVEN is available over the web.

There has been some work done in the area of GWAS analysis tools prior to MAVEN. But, none of the existing tools, including those recently developed such as GenABEL (Aulchenko, et al. 2007), SNPassoc (Gonzalez, et al. 2007) and SNPStats (Sole, et al. 2006), that are specific for GWAS, have fully considered or incorporated all the features that MAVEN has implemented. Almost all of these earlier tools focus primarily on data analysis functions such as quality control and cleaning, and single SNP- or haplotype-based statistical tests, which are greatly needed at earlier stages of GWAS. However, they provide limited, if any, functionalities on result sharing, visualization and integration, which are also greatly needed at later stages of GWAS. An exception is TAMAL (Hemminger, Saelim and Sullivan 2006), which provides some query functions
on SNPs given a list of genes. However, it does not integrate users’ data with online information.

Another recently developed tool, GWAS GUI (Chen, Liang and Abecasis 2008), also aims to visualize GWA results and thereby provides independent evidence for the need of such tools. In comparison to GWAS GUI, MAVEN has the advantage that all annotations of SNPs and genes have been either stored in local databases or can be accessed directly from NCBI through links provided by MAVEN. In contrast, users themselves have to upload annotation information into GWAS GUI, which is difficult and tedious for many researchers in the epidemiology community. Another major difference is that MAVEN is available over the web, which makes sharing of results among colleagues from different locations extremely easy. In addition, MAVEN offers powerful filtering capabilities using different criteria, interactive selections, visualization of results and seamless integration with several different annotation databases.

Note: Our Application Note on MAVEN has been reviewed, accepted and published in the Bioinformatics journal in 2009. The citation is as follows:


1.1 Related Work

Before describing the features of MAVEN, we would like to mention some of other work done in the area of visualization and curation of annotation information as part of a review exercise to get a better perspective about the kind of work done in MAVEN. We will be briefly discussing four tools – Integrative Genomics Viewer (Broad

**Integrative Genomics Viewer (IGV)** (Broad Institute 2009) - The IGV is a high-performance visualization tool for interactive exploration of large, integrated datasets. It supports a wide variety of data types including sequence alignments, microarrays, and genomic annotations. Users can upload their data and visualize it as tracks. Each track represents a sample or an experiment. Tracks can be viewed as Heat Maps, Bar Charts, Scatter plots and Line plots. Several other types of configuration are available as well like setting the data range, changing track color, height and name. It provides users with a whole genome view through which we can choose to view data on a particular chromosome, and also view all the genes present in the chosen base pair region.

**GBrowse** (Stein, et al. 2002): The Generic Genome Browser (GBrowse) is a web application used to visualize and explore annotated genes. GBrowse provides a bird’s eye view and a detailed view of the genome at the same time. Each type of annotation information is viewed as a track and hence tracks can be enabled or disabled. Users can scroll and zoom through regions of the genome and search for a genome region by entering a landmark. Users can also upload their own annotations and compare them with pre-existing public annotations.

**SNP Function Portal** (Wang, et al. 2006): SNP Function portal is a web database for exploring the function implication of SNP alleles as it is important for researchers to be able to understand the potential functional significance of SNPs to be able to understand the genome–wide SNP scanning results better. The SNP Function portal contains
annotation information from publicly available data sources and also has in-house functional annotations derived from other data sources. It allows users to upload SNPs and also has a powerful search functionality that can accept different genetic markers and obtain the SNPs related to those markers. Users can apply HapMap LD criteria to include SNPs that are in LD with the input list of SNPs. Hence the SNP function portal allows users to identify the potential biological impact of genetic markers and complex relationships among genetic markers and genes, and facilitates knowledge discovery in genome-wide SNP scanning experiments.

**H-Invitational Database (H-InvDB)** (Yamasaki et al. 2008): The H-Inv DB is an integrated database of human genes and transcripts. For all these transcripts and genes, H-Inv DB provides comprehensive annotation including gene structures, gene functions, alternative splicing variants, functional non-protein-coding RNAs, functional domains, predicted sub cellular localizations, metabolic pathways, predictions of protein 3D structure, mapping of SNPs and microsatellite repeat motifs, co-localization with orphan diseases, gene expression profiles, orthologous genes, protein–protein interactions (PPI) and annotation for gene families. The current H-InvDB annotation resources consist of two main views: Transcript view and Locus view and eight sub-databases: the DiseaseInfo Viewer, H-ANGEL, the Clustering Viewer, G-integra, the TOPO Viewer, Evola, the PPI view and the Gene family/group.
Chapter 2: Functionality

As discussed in the introduction, MAVEN provides several useful features for biologists to analyze and visualize the SNPs present in their GWA studies. In this chapter we describe each of these features in detail. To begin with, MAVEN allows users to register / or create a user account with the application and gain access to additional features to manage their study data, as discussed in the User Management system in Section 2.1.

The data stored in MAVEN is study-oriented and all functionality is implemented around this concept. The Data Management section (Section 2.2) in this chapter describes how MAVEN allows users to upload their GWAS study data and also their own SNPs into the system.

Once a study is uploaded users can apply several filter criteria to search for signification SNPs which is described under the Filtering Capabilities section (Section 2.3).

Visualization capabilities include: viewing the distribution of filtered significant SNPs as a plotted graph and as a table (Section 2.4), functional annotation information about SNPs and genes including several links to other data sources which can help users analyze significant SNPs further (Section 2.5), and viewing the entire GWAS study data as a Manhattan plot to get an overview of data distribution (Section 2.6).
2.1 User Management

MAVEN has a simple user management system incorporated to manage user accounts and access privileges. All features that allow users to upload and analyze GWAS data are available to all users of MAVEN whether or not they choose to create a user account. Hence, users can access MAVEN either:

a) anonymously, without registering themselves, in which case all study data uploaded by them will be available to all MAVEN users, or

b) register themselves with a login and password, thereby allowing them manage their study data, i.e. allowing them to choose whether or not to make a particular study available to all users or not.

Registered users can log into MAVEN by clicking on the "Sign in" link in the Top Navigation bar which will lead them to the Login Page (Figure 1).

![Login Page](image)

**Figure 1: Login Page**

Users need to enter their user id and password in the respective input fields. User IDs and passwords are case sensitive. Once the users are logged in, they are presented with the Home Page (Figure 2). A Welcome message with the logged in user's First and Last name is displayed above the top navigation.
Figure 2: Home Page

The Login page also allows new users to register themselves by clicking on the “Register” button which would lead them to the User Registration page (Figure 3). This page requires users to enter their first and last names, an email address and choose a user name and password. We have also protected this page with a CAPTCHA verification mechanism to prevent computer bots from creating thousands of user accounts every minute and causing a system overload. Once the users have completed the registration process, they will be able to log in to MAVEN with their user name and password.
Any study uploaded by *registered users* into MAVEN will be owned by them and can be managed by them through the Manage Study List page (Figure 4). This page lists all the studies created by the logged in user, along with a check box against each study row that will allow them to make a particular study “public”. This implies that this study can be viewed by all users of MAVEN, registered or anonymous. Updated Study List data can then be saved into the database. Anonymous users will be able to upload study data but do not have the privileges to make their studies private. They can also analyze any public studies created by registered users.

![Figure 4: Manage Study Data page](image)

### 2.2 Data Management

The current implementation of MAVEN accepts and stores GWAS results based on *single-locus* analysis methods, not the raw data. Summary results in general are adequate for many users and at the same time, protection of data privacy can be enforced because no personal information will be available at any time. The system allows users to upload their own results into MAVEN, with the hope that eventually, MAVEN will
become a database that collects most existing published GWAS results. All data is stored and managed using a relational database management system.

MAVEN's Upload Study page (Figure 5) allows users to upload their study data into the application. Users can prepare their analysis results in a tabular text format and upload them along with a study name, a description about their study, including references to their publications. To make it more convenient, users can directly use the output file of the program PLINK (Purcell, et al. 2007), which is a popular tool for analyzing GWAS data. The "Columns" section in the Upload Page allows users to configure the MAVEN database to accept study data in either the default file format, from PLINK (Section 6.1 - PLINK File Format), or files with a different format thereby requiring users to create the column configurations. Each study has a unique ID and MAVEN maintains all information about the study in its database. This page is also protected by the CAPTCHA verification mechanism.

![Figure 5: Upload Page](image-url)
The current version of MAVEN does not restrict users’ data to a single phenotype or a single statistics. For example, users can upload data with several P-values for a single phenotype, each based on a different test statistic or several P-values for different phenotypes. As shown in Figure 6 users can define any number of columns by specifying a column name, the data type of the column (Character, Integer or Double) and also whether or not to make them filterable (refer Section 2.4 to understand how columns configured in this page are used). When the default column configuration is used, the system automatically sets the p-Value, CHISQ and OR-Value columns as filterable. Filterable columns appear in the “Test Data” filter drop down box on the Search Screen (Figure 9). There are also a number data validations that are performed on this page for e.g., Study Name and File Name cannot be empty, column names cannot be duplicated and cannot belong to the list of SQL commands and that at least one column should be added to the Test Data filter for user defined configurations.

![Figure 6: Upload multiple phenotypes and/or multiple statistics](image)
The database for MAVEN is study oriented but also maintains a lot of common annotation information about SNPs and Genes (refer Chapter 3: Database Design). SNP annotation information has been downloaded from NCBI beforehand by parsing the ASN.1 flat files into tables. They are identified by NCBI build number and will be updated as necessary. We have also compiled SNPs from different SNP-chip platforms of two major vendors, Affymetrix and Illumina. Apart from this, users can also upload SNPs of their customized chips through the Upload SNPs page (Figure 7), also protected by CAPTCHA's. For gene annotations all three assemblies - reference, HuRef and Celera - have been considered. Also, KEGG pathway data, OMIM and transcript data from NCBI, miRNA data from PolymiRTS database and LD data from HapMap have been downloaded and stored in the database. A description of the data sources and the information used by MAVEN from these sources is discussed in detail in Section 6.1 - Data Sources.

![Upload SNP Data](image)

**Figure 7: Upload SNPs page**

Apart from the GWA Study data and the annotation information, MAVEN also stores the credentials of registered users as part of the User management system. This data allows MAVEN to obtain information about the users and also about the studies that belong to these users.
2.3 Filtering Capabilities

Once users have uploaded their data they can analyze this data to identify significant SNPs from their studies. MAVEN offers several filtering capabilities for users to retrieve such interesting SNP and gene regions. Figure 8 shows a screen shot of the Search Page containing various input fields for different search criteria that include filters using significant test parameters, chromosome, SNP rs#, gene id / gene name, functional class and disease pathways. In this section we describe each of these filters in detail.

![Figure 8: Search Page](image)

a) **Study selection:** Users are allowed choose a particular study to filter, from a list of public or owned studies, and view the results. MAVEN does not allow users to choose more than one study at a time to filter.
b) **Test Data:** The Test data filter allows users to filter data based on the significant parameters / test statistics of the GWAS study. For e.g. $p$-Value (Significance value of case/control association test), CHISQ (CHI Square Value), OR (Odds Ratio). The dropdown values for this filter are basically those Columns that the users chose to “add to Test Data Filter” on the Upload screen. Users can also specify a threshold value to this filter and specify conditions like $PVALUE < \text{Less than or Equal to} > 0.0001$. The “Apply Transformation” check box is used to transform the users data, e.g. $PVALUE$’s to “$-\log_{10}$” to better visualize these values on a graph plotted with the $PVALUE$ as it y-axis in the Search Results screen (Figure 9).

c) **Physical Position:** This filter allows users to select SNPs based on the Chromosome their physical position within the Chromosome, i.e. the base pair values. So this filter would choose for e.g. SNPs from Chromosome 1 with location between 10000000 and 300000000 base pairs.

d) **SNP rs#:** Users can provide the rs# of a SNP and a range parameter in base-pairs to obtain all SNPs in that neighborhood.
e) **Gene ID / Gene Name**: Users can provide one or more comma separated gene ID’s or gene names in this filter and a range parameter in base-pairs to obtain all SNPs in the neighborhood of the list of genes that were entered. For e.g. Gene Id: PTEN, TRIM10 and in the neighborhood of 1000000.

f) **Functional role of SNPs**: In many cases, users are more interested in SNPs that are in gene regions. MAVEN allows users to limit their search to only those SNPs in gene regions. To facilitate this, users can view SNPs from any of the functional classes shown in the figure below and be able to choose more than one class for any given search.

![Functional SNPs](image)

- Nonsense
- Missense
- Non-Synonymous
- Synonymous
- Exon
- Intron
- 5'-UTR
- 3'-UTR
- Near-Gene-5'
- Near-Gene-3'

g) **KEGG pathway filter**: This filter allows users to select all SNPs belonging to the genes present in the selected pathway. The KEGG Search consists of three levels of drop down selections. Users should start with the drop down for “Class” selection. This drop down basically lists all the different Biological Processes in Humans. Once the user has made a Class selection, the second drop down “Sub class” is populated with all the sub-classes under this particular class. The next step is to choose a particular Sub class, which will in turn populate the third drop down for “Pathway” which would list all pathway maps for the chosen Class and sub-class. This data has been downloaded from the KEGG Pathway database which is a collection if manually drawn pathway maps representing human knowledge on the molecular interaction and
reaction networks. The various drop downs for this filter are shown in the screen shot below.

All the above mentioned filters can be applied in conjunction with one another. For e.g. users can choose to view:

1. All SNPs with PVALUE $\leq 0.001$ AND in Chromosome 1 BETWEEN 10000000 AND 30000000 base pairs.
2. All SNPs in the neighborhood of SNP “rs16861613” and within the range of 10000 base pairs.
3. All SNPs that are "Non-Synonymous or Exons or Near-gene-5’" AND present in pathway “Alzheimer’s disease – Homo Sapiens (human)” AND with “PVALUE < 0.01”.
4. All SNPs present in genes “PTEN or PTPN22 or TRIM10”.

The result will contain all SNPs which satisfy all of the conditions specified in the filter. If there are no SNPs that match the all filter criteria, no results will be returned to the user.

2.4 Data Visualization and Download

As mentioned in the previous section, users can search their study data in order to filter out significant SNPs. The search results from these filter queries are then visualized
on the Search Results page (Figure 9) after the user hits the "Search" button. This page displays meta data about the study currently being filtered, the specific filter criteria chosen for this request, the list of filtered records in a tabular form and a graphical format and also summary information about SNP selected from the table or the graph. It also provides users the capability to download search results data and retrieve additional information about SNPs and genes, such as neighboring genes and SNPs in LD with the selected SNP. Search results are displayed with pagination showing 500 records per page to ensure quick turn-around time for any search query. This section describes each of these visualization features in detail.

**Figure 9: Search Results Screen**

1. **Study details:** This section at the top of the results page displays all the information the user entered about the study - Study Name, Created By, Description, Citation and File Name - while uploading the Study.
2. **Filter Criteria:** The filter criteria section displays all the filter conditions that the user chose for this request. For e.g. if the "Test Data" filter was used it would display the chosen significant parameter, say PVALUE, the threshold condition and the threshold value. If a KEGG pathway search was used, in addition to displaying the chosen pathway, this section also displays all the genes that are present in this pathway and specifically in the current page being viewed.

3. **Search results:** MAVEN displays search results (SNP records from the study) satisfying the filter criteria, in two different formats: a *tabular format* and a *graphical format*.

   - The *tabular format* displays all the columns of data from the study file that the user uploaded. For e.g. if the uploaded study contains the default data format (Figure 12) from PLINK, the table would display the following columns - SNP, CHR, A1 - (First allele code, not necessarily minor allele), F_A (Allele 1 frequency in cases), F_U (Allele 1 frequency in controls), A2 - (Second allele code, not necessarily major allele), CHISQ - CHI Square Value, *p-Value*: 

---

**Figure 10: Study Details Section**

**Figure 11: Filter Criteria Section**
Significance value of case/control association test, **OR** - Odds ratio. If the study has user defined columns, then the table displays those configured columns. The table can be sorted based on the values of a selected column by clicking on the column headers. Above the top right corner of the table we display the number of records that the query returned.

**Figure 12: Search result table sorted by PVALUE**

If a search query returns more than 500 records, MAVEN uses pagination to display the search results (Figure 13). The reason for this is to be able to provide users with the search results as quickly as possible and to also use lesser memory during run time. For all pages, MAVEN displays the current page number and the range of records being displayed.

**Figure 13: Search returning more than 500 records with pagination**
The graphical format plots each SNP in the search result on a graph, using the selected significant parameter / test data (PVALUE, ORVALUE, CHISQ etc.) as the y-axis and the chromosome base position of the SNP as the x-axis. This format facilitates users to better visualize the distribution of SNPs in their GWA study and easily identify the most significant SNPs based on the value of the selected significant parameter. If the search query returns SNPs from different chromosomes, the plot displays all chromosomes: 1-22, X and Y (if present in the data set), on the x-axis starting from 0 to the combined length of all the chromosomes and partitioned by each chromosome, each partition proportionate to the base length of the chromosome (Figure 14). If the search returns only one chromosome, then the plot's x-axis displays only one chromosome starting from 0 to the length of that chromosome in base pairs (Figure 15). When the user hovers the mouse over a plot point a tooltip showing the base position, chromosome and test statistic value, is displayed. Users can also zoom in to a particular plot area to better view the distribution of the values at a area and then reset the chart back to its original state by clicking on the "Reset Zoom" button once they have finished analyzing.

Figure 14: Graph with more than one chromosome; tool tip for highlighted point.
Both the table and the chart support interactive and synchronized selections. Any row in the table or any plot point can be clicked to view the summary information about the selected SNP (refer Figure 17: SNP Summary data section). Also, both the table and the chart are synchronized in selection, i.e. the data point / data row of the selected SNP is highlighted simultaneously in both the chart and the graph by clicking on either.

4. **Download Data**: The Search Results screen allows users the capability to download the filtered SNPs along with their annotation information into a comma separated (.csv) file. The Download Data section (Figure 16) gives users the option to download data, sorted in ascending or descending order on a particular test statistic and also select the number of records that they want to download. The downloaded file contains all the columns that the user's study contains and also other annotation information from NCBI like - Function class, Gene Name, OMIM Id, Allele change and Residue change.
5. **Functional annotation of SNPs**: The Search Results page also displays a lot of functional annotation information about SNPs in the SNP Summary section which will be discussed in detail in the next section.

2.5 **Functional annotation of SNPs**

This section talks about all the different functional annotation information that can be obtained through MAVEN for a particular SNP. MAVEN displays information from NCBI along with a lot of annotation information that was downloaded beforehand into MAVEN and links to other major online data sources so that users to get a better picture of a particular SNP.

1. **SNP Summary data**: The SNP Summary Data section (Figure 17) displays summary information about the selected SNP and contains a lot of annotation information about the selected SNP. When the user selects a SNP by clicking on a table row or graph point, an AJAX request is sent to the server. The server in turn sends a eUtils (NCBI eUtils 2009) request to NCBI and obtains the summary information about the SNP like chromosome, gene name, functional class and base position. Then it accesses the MAVEN database to obtain other annotation information like – Allele Change,
Residue Change, OMIM Id, and the KEGG pathways (Kanehisa Laboratories 2000) - which have been downloaded beforehand into the MAVEN database. The SNP Id, Gene Name, OMIM all link to the NCBI website containing information about them. The URL examples are shown below:


The KEGG pathway links to the KEGG website’s pathway page (Kanehisa Laboratories 2000). This section also provides links to data sources like GEO (Edgar, Domrachev and Lash 2001), HPRD (Prasad, et al. 2009) and MutDB (Mooney and Altman 2003) using the gene ids of the selected SNP which will be discussed in detail later.

![SNP Summary Data](image1)

![Transcript Section](image2)

2. **Transcript section:** As the name suggests, this section displays transcripts to which the selected SNP belongs (Figure 18). The figure below shows the transcripts that contain SNP rs2072505. The transcript Ids have been obtained from NCBI and stored.
in MAVEN's database. Each transcript Id is displayed as a link to the UCSC Genome Browser website, for e.g.:

http://genome.ucsc.edu/cgi-bin/hgGene?db=hg18&hgg_gene=NM_012411

This section also displays the corresponding miRNA ids for which the listed transcripts are target sites along with allele information. The miRNA id -> SNP -> transcript mappings have been obtained from the PolymiRTS database (Bao, et al. 2006) (PolymiRTS Database 2006). Each miRNA id links to the mirBase database (Griffiths-Jones, et al. 2006) with the following URL:

http://microrna.sanger.ac.uk/cgi-bin/sequences/mirna_entry.pl?id=hsa-miR-422b

<table>
<thead>
<tr>
<th>SNP Summary Data</th>
<th>Transcript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcript Id</td>
<td>Allele</td>
</tr>
<tr>
<td>NM_001635</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>C</td>
</tr>
<tr>
<td>NM_139316</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

**Figure 18: Transcript Section**

3. **Find Neighboring Genes:** This section (Figure 19) allows users to query NCBI ESearch (NCBI ESearch Entrez Utility 2010) to retrieve those genes neighboring the selected SNP. This feature is particularly useful when the selected SNP does not belong to any genomic region and the information about surrounding genes may be of importance. Users need to specify a range value in base pairs to represent the neighborhood of the search values. Once the NCBI query returns the list of genes, MAVEN initiates another query to NCBI (NCBI EFetch Entrez Utility 2010), to figure out which of the genes are real and which ones are hypothetical. Figure 19
shows the list of genes closest to SNP rs2072505 in the neighborhood of 1000000 base pairs. The user can also specify the number of closest genes that they wish to see. MAVEN then sorts the genes based on their distance from the selected SNP and returns the genes on the top of the list satisfying the count threshold. Clicking on the "more info.." opens a new window / popup (Figure 20) which contains links for all these genes to NCBI, GEO (Edgar, Domrachev and Lash 2001), HPRD (Prasad, et al. 2009) and MutDB (Mooney and Altman 2003) thus providing access to a lot of information about these genes.
4. **Find SNPs in LD**: This section (Figure 21) allows users to query MAVEN to obtain all SNPs that are in LD with the selected SNP. The LD data has been downloaded into our database from the HapMap (Thorisson, et al. 2005) (Barnes 2006). We have uploaded only those records which had a $r^2$ value greater than or equal to 0.5 so as to be able to limit the number of records that would be stored in the database. MAVEN stores the D', $r^2$ and LOD values for all four populations available in HapMap - European CEPH; Han Chinese Beijing, China; Japanese in Tokyo, Japan; Yoruba in Ibadan, Nigeria. The figure below shows all SNPs in LD with SNP rs2072505 containing a D' score between 0.5 and 0.9 in the European CEPH population.

![Neighboring Genes](image)

**Figure 21: Find SNPs in LD**

The "more info.." link as shown in the snapshot above opens a new window / popup (Figure 22) displaying annotation information about all the SNPs that were obtained from the query.
Figure 22: LD SNPs Information

5. **Genome Browser Link:** Users can also visualize at the SNPs from the search results in UCSC Genome Browser (Genome Browser 2010) as a customized track. The query to Genome browser requires the user to select a chromosome from drop down that lists all the chromosomes that were returned in the search result. When the user clicks the "Generate" button, a request is sent to server. This request generates ".bed" file which contains all the SNPs present in the selected chromosome from the search results in the format as prescribed by UCSC (BED Format 2010). Once this file is generated, a link containing the path to this bed file as an argument to the genome browser URL is created and displayed on the page. Users can then click on this link to go the Genome Browser (Genome Browser 2010) and view the SNP tracks.


Figure 23: Genome Browser link section
6. Additional links associated with Genes:

- **GEO Profiles:** MAVEN provides links to the NCBI GEO profiles of Genes. The GEO database (Edgar, Domrachev and Lash 2001) is a comprehensive database for obtaining information about micro-arrays and next generation sequencing. The URL generated links to the NCBI Entrez website using the GEO database and gene id as the query parameters as shown below:


- **Protein-Protein Interaction / HPRD Database:** Genes associated in protein-protein interactions are linked to the HPRD website which is a database of all known PPI interactions. The HPRD Ids and the associated Gene Ids have been downloaded before hand into the MAVEN database. The summary section provides a link to the HPRD website by obtaining the HPRD Id corresponding to the Gene in question and using this as the query parameter in the URL as given below:

  http://www.hprd.org/alternate?&hprd_id=06100&isoform_id=06100_1&isoform_name=

- **Post Translational Modification(PTM) Links:** Genes associated with PTM link to the MUTDB (Mooney and Altman 2003) website which requires the Gene Name and Gene Id as the query parameters as follows -

  http://mutdb.org/cgi-bin/mutdb.pl?id=PTPN22&geneid=26191.
7. **miRNA SNPs:** MAVEN also contains information that tells users whether a particular SNP is actually in an miRNA. It displays the name of the miRNA and links it to the mirBase database.

### 2.6 Manhattan Plots

MAVEN also generates Manhattan plots for an uploaded GWAS study (Figure 25). A Manhattan plot is a type of scatter plot, usually used to display data with a large number of data points - many of non-zero amplitude, and with a distribution of higher-magnitude values, which is typical of genome-wide association studies. It can be generated by clicking on the link on the Search page next to the Study List drop down. It is very useful for understanding the distribution of the entire Study data based on the obtained $p$-Values for each SNP. The plot is generated by using $-\log_{10}$ value of the significant Test Data parameter chosen in the Search page as the y-axis and the chromosome base lengths of all chromosomes as the x-axis. The parameter that is typically used on the y-axis of a Manhattan plot is the $p$-Value. Since users can create custom column configurations in MAVEN, there by being able to call the $p$-Value data...
with a custom name, we use the Test Data filter value to determine which column should be used as the y-axis in the plot.

Figure 25: Manhattan Plot
Chapter 3: Database Design

In this section we describe the database schema used by MAVEN. MAVEN uses SQL Server 2005 as its database server which is a well known relational database management system. The database is accessed from the server by making JDBC calls to insert, update and delete records. MAVEN contains tables that store information about various genetic components like Chromosomes, Genes, SNPs, LD Information and Transcripts for Humans only. We will define some of the main biological data below:

A chromosome is an organized structure of DNA and protein that is found in cells. It is a single piece of coiled DNA containing many Genes.

A gene is the basic unit of heredity in a living organism. All living things depend on genes. Genes hold the information to build and maintain an organism's cells and pass genetic traits to offspring.

A single-nucleotide polymorphism (SNP) is a DNA sequence variation occurring when a single nucleotide — A, T, C, or G — in the genome (or other shared sequence) differs between members of a species (or between paired chromosomes in an individual). For example, two sequenced DNA fragments from different individuals, AAGCCTA to AAGCTTA, contain a difference in a single nucleotide. In this case we say that there are two alleles: C and T. Almost all common SNPs have only two alleles.

In genetic epidemiology, a genome-wide association study (GWA study, or GWAS) - also known as whole genome association study (WGA study) - is an examination of genetic variation across a given genome, designed to identify genetic associations with observable traits. In human studies, this might include traits such as blood pressure or
weight, or why some people get a disease or condition. These studies normally require two groups of participants: people with the disease (cases) and similar people without (controls). After genotyping each participant, the set of markers, such as SNPs, are scanned into computers. Then bioinformatics is applied to survey participants’ genomes for markers of genetic variation.

Apart from the biological data, MAVEN stores a lot of data to manage the study and user information which is then used to manipulate the display of data on MAVEN as described in the Data Management section. Also, each study that is uploaded is inserted into a new database table created dynamically while uploading the study. In this section we describe in detail the tables that store all of the above mentioned types of data.

1. **User Data:** The *user_data_tbl* table (Figure 26) holds all information about the users who have created an account with MAVEN. The *userid* column is the primary key of this table. This table also stores the password, first name and last name, email id and role of the user. This table is also used by MAVEN's User management system for authentication and authorization purposes.

![user_data_tbl](image)

**Figure 26: User Data Table**
2. **Study Meta Data tables**: These tables (Figure 27) hold information about each of the studies uploaded by the user. The two tables that are used to hold this information are – *study_meta_data_tbl* and *study_column_data*. The *study_meta_data_tbl* contains all information about the study that user keyed in - study name, summary, citations and upload file name. Apart from this, the *study_meta_data_tbl* also contains the *study_table_id* (primary key) which is an automatically generated sequence number and is actually the name of the table created for the study, the userid of the user who created this study (this column value is set to “default” if an anonymous user created this study), a flag column to determine whether or not this study uses the default column configuration(*default_columns*), and whether this study is public and available to all MAVEN users (*is_public*). The *study_column_data* table contains the actual column configurations for all the studies. It contains the *study_id* which is actually the name of the study table, a column id, a column name, its data type and whether or not this column should be added to the test data filter. The primary key for this table is a compound key comprising of the study id and the column id.

![Figure 27: Study Meta data tables](image-url)
3. **Study Data tables:** The Study data tables (Figure 28) are created dynamically based on the columns present in the study data uploaded by the user. The table names are generated as a sequence by MAVEN. The users need to define the data type for each of the columns they are uploading if they are not using the default column configuration. Character columns are of type varchar(50), Numeric are of type numeric(18,0) and Double are of type float. In the figure below we show the study tables created for study S00001 which uses the default column configuration and study S00012 which has a user defined column configuration.

![Figure 28: Study Tables](image)

4. **Chromosome Data table:** The *chromosome_tbl* (Figure 29) stores the chromosome numbers (chromosomes 1 to 22, 23 for X, 24 for Y) and their corresponding base pair lengths with chromosome number as the primary key.

![Figure 29: Chromosome Table](image)
5. **SNP Meta Data table:** The main goal of MAVEN is to be able to provide users with information about the SNPs that are present in their GWAS studies and be able to analyze the effect of these SNPs further using different kinds of annotation information about them. The `snp_meta_data_tbl` (Figure 30) stores some basic annotation information about SNPs belonging to various SNP chip assemblies from Illumina, Affymetrix as well as Whole chromosome assembly from Illumina and Affymetrix (for more information refer Data Sources section). This table contains SNP Id or SNP rs# which is the primary key, the chromosome that the SNP belongs to, the position of this SNP in base pair values as specified by the reference, HuRef and Celera assemblies, the gene symbols of the genes that this SNP belongs to, its functional classes, allele and residue change information and an OMIM Id and allelic variant if this SNP is present in the OMIM database. All the above annotation information for the SNPs has been obtained from parsed ASN.1 files downloaded from NCBI.

![snp_meta_data_tbl](image.png)

*Figure 30: SNP Meta data table*
6. **Gene Data table:** The *gene_data* (Figure 31) table stores information about all the genes available in NCBI for Humans. This information includes the Gene Ids and Gene Symbol (gene name) as specified in NCBI, the chromosome to which this gene belongs, the starting and ending positions of this gene in the chromosome specified in base pairs, and the group or assembly (reference, HuRef or Celera) which identified this gene. The primary key is a compound key comprising the *gene_id* and *chromosome* as some genes can be found in both X and Y chromosomes.

![Figure 31: Gene Data table](image)

7. **KEGG Data tables:** The KEGG data is represented by four relations (Figure 32). This data contains pathway information about various metabolic networks in the Human body. The pathways are classified under the following hierarchy: A top level Class (*kegg_pathway_class* table), a subclass (*kegg_pathway_subclass* table) which belongs to a Class, a pathway (*kegg_pathway* table) which belongs to a specific subclass and a relation connecting pathways to genes (*kegg_pathway_gene* table), i.e. listing all genes that are contained in a pathway. The *kegg_pathway_class* relation containing Class information uses the *class_id* as the primary key and also stores a
description for the class. The `kegg_pathway_subclass` table relates a class and subclass, and has a foreign key reference to a `class_id` from the class table. The primary key for this table is a compound key made up of `class_id` and `subclass_id`. It also contains the description of the subclass. The `kegg_pathway` table relates the pathways to a particular class-subclass combination. Each pathway has a unique id which is the primary key of this table. This table also has a compound foreign key reference comprising `class_id` and `subclass_id` combinations of the `kegg_pathway_subclass` table. And finally, the `kegg_pathway_gene` relates pathway ids to gene names, and thereby gives us the list of genes present in a particular pathway.

![Diagram of KEGG Data Tables](image)

**Figure 32: KEGG Data Tables**

8. **Transcript data table:** The `transcript_tbl` (Figure 33) relates SNPs to transcript ids and the miRNA Ids to which the transcripts are target sites. A SNP can be a part of many transcripts and a transcript can contain several SNPs. Hence the primary key for this table is a compound key comprising of SNP id and transcript Id. A clustered index based on these two columns is automatically created by SQL server greatly
which improves query performance. The transcript ids have been obtained from NCBI. This table also contains miRNA information related to a SNP-Transcript combination. This miRNA information includes the miRNA id (as defined in the mirBASE database) and the miRNA allele, for both alleles of the SNP.

![Transcript data table](image)

**Figure 33: Transcript data table**

9. **Linkage Disequilibrium (LD) Data tables:** The LD data is stored in four different tables (Figure 34) each representing a population – Japanese (LD_JPT_TBL), European (LD_CEU_TBL), Yoruba (LD_YRI_TBL) and Chinese (LD_CHB_TBL). This data has been downloaded from the HapMap database. Each relation contains the following information – chromosome number of the SNPs in LD, marker1 and marker2 columns which contain the SNP rs# of the SNPs in LD, the base positions of each of these SNPs in the chromosome, the D’, r2 and LOD values of the LD for these two SNPs. We have also defined indexes based on the combination of marker id and score values as these tables contain more than 10 million records and we are interested in querying these tables using a marker id and a score as the columns in the where clause. These indexes greatly improve query performance.
10. **Protein-protein Interaction / HPRD Data table**: HPRD data is stored in the `hprd_data_tbl` (Figure 35). It relates gene names to a specific HPRD (Human Protein Reference Database) Id. These IDs are then used to access a specific record / web page on the HPRD website, which in turn displays all the protein-protein interactions that a particular gene is involved in.

![Figure 35: HPRD Data table](image)

11. **SNP Meta Data table** (main): The `snp_meta_data_tbl_big` table contains data about all the SNPs from NCBI. When the users try to upload their SNPs from their customized chip sets, MAVEN refers to this table to obtain data about the SNP. It has the same columns as `snp_meta_data_tbl` (Figure 30). We have created several indexes on this table to improve query performance. There are 3 indexes based on SNP Id,
chromosome and a finally a combination of chromosome and bp_ref (reference base pair value).

12. **Coding Function**: This table stores the different coding functions / functional classes defined for SNPs in ENSEMBL. The primary key is a sequence starting from 1 which is used solely for display purposes. This table contains the specific coding function id and descriptions obtained from ENSEMBL.

![Figure 36: Coding Function table](image)

13. **miRNA SNPs**: The `mirna_snp_tbl` contains all the SNPs that are found in Human miRNA's. It relates the SNP rs# and the miRNA id as defined by the mirBase database.

14. **Reserved SQL Keywords table**: The `reserved_sql_keywords` table contains a single column listing all the keywords so as to be able to verify and prevent users from naming study columns using SQL keywords. The data from this table is used when the user is trying to upload a new study.
Chapter 4: Implementation

In this chapter we first describe the overall architectural pattern of MAVEN. Then we describe the various objects used by MAVEN for data transfer and communication followed by the back-end implementation classes that use these objects. We also describe the authorization framework implemented in MAVEN, a very detailed look at how MAVEN implements the Search query, the Java package structure and the various components of the User Interface. We finally discuss some performance tests that were conducted on MAVEN and present the results that were obtained.

4.1 MAVEN Architecture and Design

In this section we describe in detail the various architectural components used by MAVEN. The architecture of MAVEN is a combination of the MVC (Model, View and Controller) and the 3-Tier architecture patterns. We begin by describing each of these architecture patterns and then look at how they have been amalgamated by MAVEN.

**Model-View-Controller Architecture:** MAVEN's User Interface has been implemented using Struts 1.1. Struts (Apache Corporation - Struts n.d.) is a well known framework which follows the Model-View-Controller (MVC) architecture. As the name suggests, the three parts of this architecture are, the Model, the View and the Controller. A diagrammatic representation of this architecture and the interactions between the components is shown below:
The Model is refers to the domain specific data that will be used by the application. The View renders the model in a way that is understandable to the user. The Controller acts as the mediator between the View and Model, in that it accepts user input and updates/retrieves the required data for the user, i.e. it determines the flow or path of the request.

3-Tier architecture: This architectural pattern divides the application into three tiers – Presentation tier which handles user input and implements the user interface, Business Logic tier which handles all business logic and validations and the Data Access Logic tier which communicates directly with the database. The Presentation and Data Access logic tiers only communicate with the Business Logic tier. Hence, it is the responsibility of the Business Logic tier to pass the data back and forth between these two layers, and implement all business logic. Data is passed between the tiers in the form of Business Objects. A diagrammatic representation of this architecture is shown below:
MAVEN's Architecture: MAVEN combines the two architectural patterns that have been described above (Figure 37). The MVC framework is implemented in MAVEN as follows: Struts provides and implements the Controller which is called the ActionServlet. The ActionServlet is responsible to route requests from the user to the server. The View consists of Action classes, JSPs and ActionForm beans for each screen/ action in the application which implemented as part of the Struts framework. The Model is made of the business/service and data access layers. Similarly, the 3-tier architecture is implemented as follows: The Presentation Layer being the Struts Action, ActionForm, JSP pages, and the ActionServlet (Controller); the Service Layer classes implementing the business logic; and the DAO classes accessing the database. This is depicted in figure below. MAVEN uses Business Objects to retrieve and store information from the database as described in the following section.

![Figure 37: MAVEN Architecture](image)

4.2 Object Model

In this section we will be describing the various objects that are used in MAVEN for data transfer between the different layers and also for representing the information finally displayed to the user. Hence the objects can be separated into two categories:
– **Business Objects** used by the back-end to transfer data between various layers
– **ActionForms** that contain Business Objects and make them available to the user.

The next two sub sections describe the various objects that are used by MAVEN under these two categories.

### 4.2.1 Business Objects

Business Objects are basically JavaBeans that hold information that needs to be transferred between the various layers in MAVEN. They are the logical representations of the data stored in the MAVEN database. In this section we will describe in detail the various business objects that are used by MAVEN. All Business objects are serializable, i.e. they implement the `Serializable` interface provided by the JDK (Java Development Kit).

1. **LabelValueBean:** The `LabelValueBean` is a utility object. It basically represents a name-value pair and is used by other Business Objects and `ActionForm's` that are required to represent this kind of data. For e.g. most of the drop down boxes used in MAVEN can be viewed as name-value pairs containing an ID and a description of each entry in the drop down list. Apart from the ID and Description, the `LabelValueBean` can be enhanced to include any other property that might be required for a specific case.

```
com.cwru.maven.vo.LabelValueBean
- label: String
- value: String
```

2. **Study:** The Study object basically represents all the information about a GWAS study that is stored in MAVEN. Looking at this object one can understand how MAVEN
looks at study data from a logical perspective. A Study object is perceived as something that contains meta information about itself and the actual study data rows uploaded by the user into MAVEN. Hence the Study object aggregates or encapsulates two important business objects - the StudyMetaData object and an array of StudyData objects. Each of these objects will be described in detail below. Figure 38 represents the aggregation of all the objects that are associated with a GWAS study in MAVEN.

Figure 38: Study Business Object
3. **StudyMetaData**: The StudyMetaData represents the meta data about the Study as keyed in by the user in the Upload Study page. It contains getters and setters for

- studyId, which is the name of the database table in which study data is stored, study name, summary and citation fields.
- the file name of the study data file
- the userid of the logged in user
- an array of StudyColumnData objects to hold data about the study columns created by the user to hold the data they are uploading. The StudyColumnData object contains getters and setters to for the column name, data type and a flag stating whether or not the column is filterable.
- defaultColumns flag to tell the back-end whether or not the study is using the default column configuration.
- makePublic flag to determine whether or not this particular study is to be made public to all MAVEN users.
4. **StudyData**: The *StudyData* business object holds information about each row of data present in the GWAS study. All studies uploaded by the user should contain the chromosome number and SNP rs#, and hence this object contains getters and setters for these two fields. It contains the *studyColumns* attribute, which is an array of *LabelValueBeans* (name-value pairs), which holds the names of the user defined columns added by the user or the rest of the columns from the default configuration, and the value present in that row for this column. The attributes *bpValue* and *geneName* hold the base pair position of the SNP in the chromosome and the names of the genes it belongs to.

![com.cwru.maven.vo.StudyData class diagram]

5. **GeneData**: The *GeneData* object represents the gene information stored in MAVEN. It contains getters and setters for Gene Id, Gene Name, Chromosome, Start position of this gene in the chromosome, end position of this gene in the chromosome, the organism or tax id used by NCBI and the assembly to which this gene belongs (reference, HuRef, Celera).

![com.cwru.maven.vo.GeneData class diagram]
6. **SearchCriteria:** This is an important object which holds all the search criteria, as displayed in the Search screen, that MAVEN provides users so as to be able to filter out significant SNPs from their GWAS studies. It contains getters and setters for study list, test filters and their threshold values, chromosome number and base pair positions, SNP id filter with threshold, gene id/ gene name filter with threshold, functional class filter, KEGG pathway filters. It also maintains the current pagination information, which is required while retrieving results for the next/previous page when the user is trying to navigate through the pages. Pagination is maintained using a page count parameter along with the page number of the page currently being viewed by the user. It also maintains the download paging counters, used when trying to download the search results. These two counters are separate because the number records per page differ based on whether the user is viewing or downloading the results.

7. **DownloadCriteria:** This object contains getters and setters for the download filter conditions like the Significant parameter based on which data will be sorted, the number of records that need to be downloaded and the sort order: ascending or descending.
8. **LDCriteria**: Stores the search conditions for retrieving SNPs that are in LD with the currently selected SNP. The attributes include the following - Population code, LD score type (D', r², LOD), minimum and maximum thresholds for the score parameter.

9. **TranscriptData**: Stores transcript details for the currently selected SNP. Attributes include transcript Id, allele 1 and allele 2 of the SNP and miRNA ids for both alleles.

10. **User**: This business object stores the user data of the currently logged in user. It contains getters and setters for user id, first name, last name, email and a flag that tells us whether or not this user is authorized.
4.2.2 ActionForms

ActionForms are basically objects that are used by the Struts Framework to transfer data between the Model and View. ActionForms are JavaBeans that contain getters and setters for the various properties or user input elements that are displayed on the screen for the user. Every form needs to extend the "ActionForm" class provided by the Struts Framework. Each request has an associated ActionForm which is populated by the JSP on initiation of the submit request. The ActionForm instance is then used by the Action class to process the user input. We describe each of the main ActionForm classes used by MAVEN below.

1. **BaseActionForm**: The BaseActionForm (Figure 39) is the base class for all forms (pages) that require user authentication. It extends the ActionForm class and provides getters and to access information about the logged in user through the User business object and also maintains a flag to confirm whether a user is authorized or not. In the current implementation, three forms, MavenMainForm, ManageStudyListForm and UploadDataForm extend this form. The BaseActionForm is populated by the BaseActionClass, when the user has logged in and tries to access a page that requires user authentication. The BaseActionForm was designed so that all forms that require user authentication would automatically contain the user information that is required and not have to access the HttpSession object every time it is required. The authorization framework in described in detail in Section 4.3.
2. **LoginActionForm**: The `LoginActionForm` is used by the `LoginAction` and is populated when users enter their credentials on the login page and hit the "Login" button. It contains getters and setters for the `userid` and `password` input fields. It also maintains two flags - `login` and `invalidUser`. The `login` flag basically establishes whether the user is actually trying to login or is just visiting the login page. The `invalidUser` flag is set to true when authentication for this user has failed when they try to log in. This class extends the `ActionForm` class.

3. **UserRegistrationForm**: The `UserRegistrationForm` is used by the `UserRegistrationAction` and is populated when a new user tries to create an account.
for them from the User Registration page. It contains getters and setters for the User business object that holds the information about the user, the password field where the user enters their chosen password, a confirmPassword field that holds the reconfirmed password, and a register flag that determines whether or not the user is actually trying to register or is just visiting the registration page.

4. **MavenMainForm**: The MavenMainForm is the main form that is used by the SearchAction and the SearchResultsAction classes. It extends the BaseActionForm class, as it contains fields that behave differently based on user credentials or basically fields that require user authentication. It is populated and used by the Search page and the Results page. This form contains getters and setters for the following business objects: SearchCriteria which holds all the fields available to filter study data, DownloadCriteria which holds fields for filtering rows that the user wants to download from the search results, Study which holds the study data, and several arrays for various drop down lists on the Search screen - KEGG pathway, class, subclass lists, list of studies uploaded in MAVEN, list of chromosomes, a string to hold SNP summary information that is returned to the form as an AJAX response and a field to hold the SNP Id for which the user is currently requesting information. The various fields are populated when the user accesses the Search page and hits the search button. Figure explains the aggregation of various data types into MavenMainForm.
5. **ManageStudyListForm**: This form is associated with the `ManageStudyListAction` and is populated when the user updates or accesses the Manage Study List page. It contains getters and setters for the `studyList` parameter which is an array of `StudyMetaData` business objects, holding the list of studies for the logged in user. It extends the `BaseActionForm`, as viewing Manage Study List page requires user authentication, so as to be able to obtain the list of studies managed by this user. The user can update the "makePublic" property in the `StudyMetaData` object when they update the study list on this webpage.
6. **UploadDataForm**: The *UploadDataForm* is used by the *UploadDataAction* class and is populated when the user is uploading a study into MAVEN, i.e. when the user is accessing the Upload Study page. This form contains getters and setters for the *StudyMetaData* business object which holds information about the study as keyed in by the user, a list of *StudyColumnData* objects to manage the study column configuration entered by the user, a *FormFile* object to hold the uploaded file stream from the user's machine, and the *upload* flag which tell the *UploadAction* whether the user is actually performing an upload operation or is just visiting the Upload Study page. This form extends the *BaseActionForm*, because this page needs the user credentials to relate the uploaded study to logged in user.

7. **UploadSNPDataForm**: The *UploadSNPDataForm* is a simple form associated with the *UploadSNPDataAction* and is populated when the user is trying to upload SNPs into the MAVEN database, i.e. when the user is trying to access the Upload SNPs
page in MAVEN. It has getters and setters for the *FormFile* object to hold the file stream being uploaded and a flag to determine whether the user is performing and upload operation or is just visiting the page.

![Image of FormFile object](image)

#### 4.3 MAVEN's Authorization framework

In this section we will be describing the user authorization framework employed in MAVEN and describe the various classes associated with authorization. To facilitate authenticating the user at one place in the application, all action classes that contain some operation that requires a user to be logged in would extend a *BaseActionClass*. The Struts Framework provides an organized way to configure a specific action as requiring authorization by using the action mappings section of the struts-config.xml file. The framework is as follows: the requirement for authorization can be viewed as a new user defined property of the action-mapping. When an application requires a custom property to be added to an action, it can extend the *ActionMapping* class provided by struts and create a custom *ActionMapping* class which holds getters and setters for that property. So if a particular action requires authorization, we can specify the action mapping in the struts-config.xml as follows:

```xml
<action path="/getStudyList" type="com.cwru.maven.ui.ManageStudyListAction" name="manageStudyListForm" parameter="getStudyList" className="com.cwru.maven.authentication.MavenActionMapping">
  <set-property property="authorizationRequired" value="true"/>
  <forward name="success" path="doc.manageStudyList"/>
</action>
```
In the above action mapping, we see that the `getStudyList` action uses the `MavenActionMapping` class defined in the "className" attribute of the `<action>` tag. `MavenActionMapping` class extends the `ActionMapping` class provided by the struts framework and adds the "authorizationRequired" property to the mapping. If this property is set to true for an action, using the `<set-property/>` tag, it would imply that this particular action requires a user to be logged in. The `BaseActionClass`, which is extended by all Action classes requiring authentication, receives requests directed to all such Action classes. The `execute()` method of the `BaseActionClass` checks for the authenticity of the user if a particular action that requires authentication has been invoked. If user is not authorized, then the control is forwarded to the login page, else, the requested action is executed thereby preventing unauthorized access.

4.4 Implementation Classes

In this section we describe in detail the various classes implemented in the different layers of MAVEN. The Presentation layer classes mainly include the Struts Action classes. These classes extend the "Action" class provided by the Struts Framework and are responsible for processing user requests and communicating with the Service Layer to obtain the required response. Once the user's request has been processed the Action classes forward the response back to user. The Action classes make calls to the Service Layer methods to request the required data and make the necessary updates. The Service Layer methods in turn make calls to various DAO classes to perform the necessary database updates and retrievals. We describe these implementation classes by grouping them based on the various features of MAVEN that are supported by them.
Also, some of the complicated features are described with the help of UML Sequence Diagrams to better illustrate and understand the control flow.

1. **Login / Logout**: The two action classes associated with login and logout are `LoginAction` and `LogoutAction`. The `LoginAction` class extends Action and overrides the execute method provided by the Action class. In the execute method, this class performs the following actions: If the user is going to visit the login page, this action resets the form data and then forwards the request to the login JSP page. If the user is trying to log in, it makes a call to the `AuthenticationService.authorizeUser()` method, belonging to the Service layer to authenticate the user. If the user could not be authenticated, it returns to the login page with an error message; otherwise, it sets the user details into the session, and forwards to the Home Page. The `authorizeUser()` method in turn uses the `UserDataDao.getUser()` method to retrieve user details based on login id and password.

   The `LogoutAction` class, extends the Action class and overrides the execute method. It basically removes the user information from the session and then invalidates it.

2. **User Registration**: The feature is handled by `UserRegistrationAction` that extends the Action class and overrides the `execute()` method. The User Registration page uses CAPTCHA to prevent overloading the system with computer generated requests. So the `execute()` method first validates the CAPTCHA string entered by the user against the system generated string. If validated, it calls the `AuthenticationService.registerUser()` Service layer method which in turn calls the `UserDataDao.insertUser()` DAO layer method to insert the user details into the `user_data_tbl`. 

57
3. **Manage Study List**: The feature is handled by the `ManageStudyListAction` which extends the `BaseActionClass` as viewing this page requires user authentication (refer Section 4.3 for more information). It implements two actions: `getStudyList` and `updateStudyList`.

- `getStudyList` basically retrieves the list of studies assigned to this user. This action method invokes the `MavenMainService.getStudyListMetaData(userId)` Service layer method which in turn invokes the `StudyMetaDataDao.getStudyListForUser(userId)` DAO layer method which queries the `study_meta_data_tbl` to get the required records.

- `updateStudyList` saves the changes made to the study on this page. The users can modify the `makePublic` flag for all the studies on their list. The `updateStudyList` action invokes the `MavenMainService.updateStudyMetaData()` Service layer method which in turn invokes the `StudyMetaDataDao.updateStudyList()` DAO layer method. The `StudyMetaDataDao.updateStudyList()` executes a batch update query to update all the records in the `study_meta_data_tbl`.

4. **Search Page**: The `SearchAction` responds to the user’s request to view the Search page. The `SearchAction` class extends the `BaseActionClass` since the study list drop down displayed on this page requires user details so as to determine which of the studies need to be displayed to the user. It contains two main action methods:

- The `search()` method uses the `MavenMainService` to obtain the list of studies, and the list of filterable columns for the first study in the list from the `study_meta_data_tbl` and the `study_column_data` tables respectively, when the page is requested. It brings all the studies which are public if the user is
anonymous; public (owned and anonymous) and private studies if the user is registered. It checks the allow_filtering column of the study_column_data table to retrieve the list of filterable columns for the "Test Data" filter. It uses the KeggDataService methods to obtain the list of KEGG Classes from the kegg_pathway_class table. A sequence diagram depicting the flow of control is shown below:

− The filterColumns() action method receives AJAX requests from the Search page. It populates: the Test Data Filter drop down when the user selects a different Study to analyze, the KEGG Pathway Sub class drop down when the Pathway Class is chosen and the KEGG Pathway drop down when the KEGG Sub class is chosen. It uses the MavenMainService and KeggDataService classes to retrieve
the required values form the database. The Dao’s involved in this action are

*StudyMetaDataDao* and *KeggDataDao*.

5. **Results:** The *SearchResultsAction* class is responsible for handling requests that are sent to and from the Search Results page. It contains the following action methods:

− *results()* action method is an important method in this class. When the user hits the Search button on the Search page, the request is forwarded to this action. The main responsibility of this action is to retrieve the records of the selected study matching the filter criteria. It performs two validations before beginning the process of retrieving search records. If user is not logged in and the study chosen is not created by anonymous user or is not public, return with error message. If user is logged in, but is trying to view another users private table, return with an error message. If both the above conditions have not been violated, this action calls the *MavenMainService.getStudyData()* Service layer method by passing the SearchCriteria as the parameter. The *getStudyData* method first determines the columns that are defined for this study table by querying the *study_column_data* table, and then using the SearchCriteria and the list of columns, queries the respective study table to retrieve study data. The method used for this purpose is the *StudyDataDao.select()* DAO layer method. After retrieving the study data, this method retrieves summary information about the first SNP on the list, to be displayed on the screen when it loads. SNP Summary information is obtained by first posting a request to NCBI. MavenMainService and KeggDataService classes are used to obtain other annotation information about the SNP as follows: *MavenMainService* is invoked to obtain data pertaining to the SNP, like OMIM
Ids, allele and residue changes etc from the `snp_meta_data_tbl` using the `StudyDataDao`, and also to obtain any HPRD Ids for PPI links. The `KeggDataService` is used to obtain the list of pathways that this SNP belongs to by using the names of the genes containing this SNP. Apart from this, the `results()` action also formats the StudyData into a string representation that can be used by the JQuery library ([Section 4.7 - User Interface Implementation](#)) to create the graphical representation of this data. The data for plotting the graph is prepared in a two step process: first, we scale the base pair lengths of the chromosomes which are in the order of 100,000,000 base pairs so as to be able to represent them in the graph and then we determine the position of the SNPs based on it base pair position and these scaled lengths. Chromosome 1 is the longest chromosome. So we start by assigning a fixed base length value of 2 for this chromosome. The lengths of all other chromosomes are then calculated based on this base length value, i.e. in proportion to the length of Chromosome 1 and are stored in a map. These values are then applied to the base pair position of each SNP in the study data to determine the actual position of a SNP in the scaled chromosome. The final operation performed by this action method is to obtain Transcript information about this SNP using the `MavenMainService` to query the `transcript_tbl` using the `TranscriptDataDao`. The figure below shows the sequence diagram for these operations:
getSNPSummary() action method receives the AJAX request from the Search Results page when the user makes a SNP selection, by either clicking on a table row or selecting a different plot point. The selected SNP's rs# is sent as a request parameter to this method. It uses the same methods used by the results() action to retrieve the SNP Summary information, other annotation information along with transcript information for the selected SNP.

download() action method receives the request to download data. This method invokes the MavenMainService.downloadStudyData() to get the required records to download. The downloadStudyData() method uses the SearchCriteria and the DownloadCriteria to perform the download operation. The SearchCriteria is applied because we need to download a subset or all of the search results. The DownloadCriteria contains the download instructions needed to determine the
sort order and the significant parameter on which this sort order should be applied and also the number of records that need to be downloaded. The download operation is makes multiple calls to the `MavenMainService.downloadStudyData()` Service layer method, retrieving 1000 records at a time. This is to avoid encountering an `OutOfMemory` Exception from the JVM in case there are several 100,000 records to download and also to properly buffer the output.

- `geneData()` action method receives the AJAX request to retrieve the closest neighbors of the Gene to which the SNP belongs. This action accepts the Window Size and number of closest genes as request parameters and uses the `NCBIService` to obtain the list of closest genes. The `NCBIService.getGeneData()` sends requests to the ESearch utility of NCBI EUtils to obtain the list of neighbors of the gene. From this list of genes it determines which of the genes is the closest to this gene by ordering them based on their distance in chromosome base pairs. Once this list is determined, it sends a request using EFetch utility of NCBI EUtils to obtain the list of Pseudo genes.

- `snpsInLD()` method receives the AJAX request to obtain the list of SNPs that in LD with the selected SNP. It invokes the `MavenMainService.getSnpsInLD()` Service layer method using the `LDCriteria` object to query the LD tables using the `LDDataDao.getSnpsInLD()` DAO layer method. The specific LD table to be queried is determined based on the chosen population for e.g. if the European population was chosen then the LD_CEO_TBL is queried to obtain the required results.
- `genomeBrowserLink()` action method receives the AJAX request to generate the .BED file and the required link to access the UCSC Genome Browser. The URL is supplied with the location of the .BED file as a query parameter. The .BED file is generated using the `GenomeBrowserService.generateBedFile()` method. The `SearchCriteria` business object is passed to this method so as to be able to apply these criteria in addition to the chromosome which the user requested to be viewed in the genome browser.

6. **Upload Study Data:** The `UploadDataAction` handles requests to view the Upload Data Page and also requests to upload a study into the MAVEN Database. The input file is read as a stream and the upload operation is split into multiple database calls uploading 1000 records at a time to avoid getting an `OutOfMemory` exception. The `UploadService.uploadData()` Service layer method is used to perform this operation. The `uploadData()` method first determines if default or user defined columns are being used to upload the study. Once this is established, it makes call to the `StudyDataDao.createStudyTable()` method by passing the `StudyMetaData` object. This method creates a database table with table names generated by a sequence starting with “S00001” and columns as defined in the `StudyMetaData` object; and then returns the name of the newly created table. Once the table has been created, the `uploadData()` method inserts the `StudyMetaData` into the `study_meta_data_tbl` and the `study_column_data` tables. After this, it parses the input file stream and inserts 1000 records at a time into the newly created study table. If any file format errors are encountered during upload, the study table is dropped and the meta data information is deleted and an `InvalidFileFormat` exception, which is a custom exception, is
thrown to the action method, which in turn displays the error message to the user. If the upload was successful, the control is passed on to the Search Page.

7. **Upload SNPs:** The *UploadSNPDataAction* handles requests to upload SNPs into the database. It uses the *UploadService* to insert the records. The uploaded file consists only of SNP rs numbers. The *UploadService.uploadSnpData()* method uploads 1000 SNPs at a time. It invokes the *SnpDataDao.insertSnpData()* method which is implemented as follows: we first identify those SNPs that are already present in the *snp_meta_data_tbl*. After discarding these SNPs from this list, this method creates a insert statement which selects data for the remaining SNPs from the *snp_meta_data_tbl_big*, as this table contains information about all SNPs that have been identified in NCBI, and then inserts it into the *snp_meta_data_tbl*.

8. **Manhattan Plot:** The *ManhattanPlotAction* generates the Manhattan plot for the entire GWA study by using the JFreeChart open source library (JFreeChart 2009). The generatePlot() action receives the request from the user and uses the generateGraph() method of this action class to generate the graph. The generateGraph() method first obtains the list of chromosomes from the database. It then calculates the scaled length for each of these chromosomes by assigning a base fixed value of 1 for Chromosome 1 and then calculates the length of the rest of the chromosomes based on this value. Then it retrieves the SNP records from the study belonging to each chromosome to calculate the position of each of these SNPs on the scaled graph. The computed data for SNPs in each chromosome are stored in the JFreeChart *XYSeries* object and then added to the *XYDataCollection* object which is then passed as argument to the *ChartUtilities.saveChartAsJPEG()* method defined in
the JFreeChart library. This method generates a "jpg" image file under the "temp" folder of our web application with a unique application generated file name. This image is then embedded into the Manhattan plot web page for display.

4.5 Search Query Implementation

In this section we describe how the search query is determined with the various search parameters. We mainly describe the implementation of the StudyDataDao’s select() method which takes the SearchCriteria and StudyColumnData as input parameters and returns the set of rows satisfying this search criteria.

This method begins by first constructing the SELECT SQL query using the various search parameters by invoking the retrieveSelectQuery() method. The retrieveSelectQuery() method combines all the query conditions with the AND clause. The Test Data, Chromosome, SNP rs# and Functional class conditions are fairly simple to handle. The method checks if these conditions are present and constructs the respective condition based on the selected parameter and threshold values for the WHERE clause and combines the different conditions with an AND clause. However, when we need to combine these conditions with the Gene search or the KEGG pathway search the query becomes more complicated.

When the KEGG pathway search condition is involved, we first need to determine the genes that are present in the selected pathway. The major point to be considered is that a pathway can contain several hundreds of genes, and hence cannot be added to a single query as SQL Server has restrictions placed on the length of a query. Apart from the fact that there can be several hundreds of genes, we need to take into account if a
particular Chromosome and/or if Gene Ids/Names were added to the filter. This would imply that we cannot consider all the genes that are listed for the selected pathway.

Therefore when a combination of these conditions is presented, we first retrieve the pathway genes by applying the chromosome condition if present. Then, if Gene Ids were entered, we retrieve information about those genes as well. Once we have both these lists, we determine which of the genes need to be used by obtaining all genes common to both these lists. After this is determined, we start building the SELECT query.

The SELECT query basically needs to JOIN the study table with the `snp_meta_data_tbl` to retrieve all information about the required SNPs. These two tables are joined on SNP id and chromosome columns. To this JOIN we also apply the WHERE clause conditions. If KEGG pathway condition has been applied, the query determines if the number of genes in question is greater than 100. This condition is applied because of the query length restrictions as stated before, and definitely a SELECT query involving hundreds of genes is extremely long. Since there is more than one gene involved, this method constructs a UNION query where each SELECT contains the JOIN clause and all the filter conditions present in the constructed WHERE clause apart from the gene data specific conditions which are the chromosome number and the base position start and end. If there are more than a 100 genes, this method creates a temp table with the required result columns, and creates a SQL UNION query with the first 100 genes, and inserts the data into the temp table. This is repeated for the next 100 genes until all the genes in the list have been processed. Thus when the KEGG pathway condition is present, the SELECT query that is returned would be querying the temp table. If KEGG
pathway condition was not chosen, the SELECT query with the JOIN including all chosen filters in the WHERE clause is returned.

Once the SELECT query is obtained, the select() method first determines the number of records that the query will return. This record count is then used to determine the number of pages that this particular search result will contain. Once the page count is determined, it is stored in the SearchCriteria, and then applied to the select query in order to establish the pagination mechanism. When the user tries to navigate the search results by clicking on the next / previous buttons, this method is invoked again and this time, it does not need to determine the page count again. It remembers the current page that the user is in and returns the next set of records accordingly.

4.6 Java Package Structure
MAVEN follows a simple package structure demarcating the different layers. In this section we list all the packages and the classes that they contain.

- **ActionForm and Action Classes**: com.cwru.maven.ui
- **Business Objects**: com.cwru.maven.vo
- **Service Layer Classes**: com.cwru.maven.service
- **Exceptions**: com.cwru.maven.exception
- **Data Access Layer classes**: com.cwru.maven.dao
- **Common Utility classes and objects**: com.cwru.maven.common
- **Authentication Classes**: com.cwru.maven.authentication
4.7 User Interface Implementation

MAVEN's user interface has been implemented using the Struts and Tiles frameworks (Apache Corporation - Tiles n.d.). In this section we will be describing how MAVEN uses and configures these two frameworks.

The Struts Framework defines all the user actions (called action mappings) in the struts-config.xml file. Each action mapping is associated with an action form, an action class and an action mapping class. Action mappings also define action forwards i.e. the path of control flow once the action is executed. An action can define several such forwards based on what needs to be done. Each of these forwards define a path which is the name of the tiles definition for the particular page. Example of struts-config.xml action mapping is shown below:

```xml
<action path="/results" type="com.cwru.maven.ui.SearchResultsAction" name="mavenMainForm" parameter="results">
    <forward name="results" path="/doc.resultsLayout" />
    <forward name="searchPage" path="/search.do" />
</action>
```

The Tiles framework allows us to split the UI into different components, so that the same code can be reused on different pages. It basically allows us to share layout information between different pages. We can also extend a layout to change content for different pages. Tiles definitions are configured in the tiles-def.xml. MAVEN uses two layouts - the mainLayout, which is defined in the basicLayout.jsp file and the resultsLayout which is defined in the resultsLayout.jsp file. The mainLayout contains the following tiles: title, header, top navigation and the data container which displays the contents of the specific page. The mainLayout is used by the Login / Logout, User
Registration, Home, Search, Upload Data, Upload SNPs and the Help pages. The definition for each of the pages extends the mainLayout and overrides the title and the dataContainer tiles of the layout.

Sample tiles-def.xml definitions for mainLayout:

```xml
<definition name="doc.mainLayout" path="/layout/basicLayout.jsp">
  <put name="title" value="" />
  <put name="header" value="/common/header.jsp" />
  <put name="topNavigation" value="/pages/topNavigation.jsp" />
  <put name="dataContainer" value="" />
</definition>

<definition name="doc.loginPage" extends="doc.mainLayout">
  <put name="title" value="MAVEN - Login" />
  <put name="dataContainer" value="/pages/login.jsp" />
</definition>
```

The Search Results page uses the resultsLayout format. The resultsLayout contains the following tiles: title, header, top navigation, study meta data, download, search results table, graph data and SNP Summary Data tiles. The definition of this layout looks as follows:

```xml
<definition name="doc.resultsLayout" path="/layout/resultsLayout.jsp">
  <put name="title" value="MAVEN - Results" />
  <put name="header" value="/common/header.jsp" />
  <put name="topNavigation" value="/pages/topNavigation.jsp" />
  <put name="studymetadata" value="/pages/studyMetaData.jsp" />
  <put name="download" value="/pages/download.jsp" />
  <put name="searchResults" value="/pages/searchResults.jsp" />
  <put name="graphdata" value="/pages/graphData.jsp" />
  <put name="snpSummaryData" value="/pages/snpSummaryData.jsp" />
</definition>
```

Each tile is actually a JSP page defining the content of that particular tile: the header content is defined in header.jsp and top navigation content is defined in topNavigation.jsp etc. The various WEB Components, i.e. the JSP pages, Script files, images, and CSS files are all placed in the WebContent folder under different sub-folders - pages for JSPs,
scripts for JavaScript, images for Image files, styles for CSS style sheets and layouts for tiles layout JSPs. A JSP file will have an associated JavaScript file by the same name if it requires JavaScript functions. All script methods are defined mostly in ".js" files only. MAVEN uses CSS style sheets to style the various UI elements on the screen.

The graphical format displayed in the Search Results page has been implemented using the jQuery-flot library (jQuery flot 2009). jQuery (jQuery 2009) is a JavaScript Library that simplifies HTML document traversing, event handling, animating and AJAX interactions for rapid web development. Flot is a pure JavaScript plotting library for jQuery. It produces graphical plots of arbitrary datasets on-the-fly in the client-side. Manhattan Plots have been generated using the open source JFreeChart library (JFreeChart 2009)

The CAPTCHA's used in the User Registration, Upload Study and Upload SNPs pages, have been implemented using the JCaptcha library (Jcaptcha 2010) which is an open source framework for CAPTCHA definition and integration.

4.8 Performance Tests and Results

We have tested MAVEN using a GWA Study consisting of ~1 million SNP records. The test results on our local network show that MAVEN can efficiently handle such large datasets. This can be substantiated as follows. The data upload and database creation steps are by far the most time consuming steps in MAVEN. Uploading a study with ~1 million records takes about 4-5 minutes. But this does not pose an issue because it is only a one-time operation. The main performance measure for MAVEN is based on how quickly the application is able to retrieve Search results. Our tests have shown that this operation typically takes less than a few seconds irrespective of the number of
records returned by the query, or that the study table itself may contain a million records, because we use pagination to always limit the number of records to 500 per page. However, we have noticed that when a search criteria contains a KEGG pathway with hundreds of genes, it takes a few minutes to retrieve the search results. But this is mainly due to the complexity of the query itself and we have made taken all measures to optimize this query so as to reduce the query processing time. This proves that MAVEN is able to handle large datasets efficiently.
Chapter 5: Future Work

There is a lot of scope for adding new features to MAVEN in terms of new annotations or data sources. These features can be added as additional Search criteria and / or can be included in the SNP annotation information in the Search Results page. Some of the new data sources that can be considered are discussed in this section.

Gene Ontology (Gene Ontology 2010): The Gene Ontology database is a very popular database in the research community and is a very useful database because it addresses the effort to provide biologists with a consistent description of gene products in different databases. Data is organized in terms of cellular components, biological processes and molecular function.

BioCarta (BioCarta 2010) is another important data source that provides researchers with a graphical depiction of gene interactions using dynamic models. It also catalogs and summarizes important resources providing information for over 120,000 genes from multiple species and includes both classical pathways as well as current suggestions for new pathways.

We could also integrate ENSEMBL database (ENSEMBL 2010) information into MAVEN. ENSEMBL is a joint project between EMBL-EBL and Wellcome Trust Sanger Institute which produces and maintains automatic annotation software on selected eukaryotic genomes.

Gene Ontology, BioCarta and ENSEMBL databases will be interesting additions to the Gene annotation in MAVEN and will give a deeper understanding about the significance of the SNPs present in the GWA study.
Another interesting addition to the SNP annotation could be including details about Protein Structure and Function information from the modBase LS-SNP database (LS-SNP: Large Scale Human SNP Annotation 2010). This database catalogs information about coding region polymorphisms that result in amino-acid residue changes [ie, non-synonymous cSNPS (nsSNPs)] are of critical importance in human disease and drug sensitivity and will help understand the functional consequence of these SNPs better.

Generating analysis reports for significant SNPs could be another useful feature to add to MAVEN. Since MAVEN contains a lot of annotation information about SNPs and Genes it would be interesting to look at an overall picture of the GWA study data.
Chapter 6: Appendix

6.1 PLINK File Format

<table>
<thead>
<tr>
<th>CHR</th>
<th>SNP</th>
<th>BP</th>
<th>A1</th>
<th>F_A</th>
<th>F_U</th>
<th>A2</th>
<th>CHISQ</th>
<th>P</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rs11590169</td>
<td>6720947</td>
<td>G</td>
<td>0.04957</td>
<td>0.08425</td>
<td>A</td>
<td>17.83</td>
<td>2.42e-05</td>
<td>0.5669</td>
</tr>
<tr>
<td>1</td>
<td>rs2336385</td>
<td>11875750</td>
<td>A</td>
<td>0.1221</td>
<td>0.1774</td>
<td>C</td>
<td>21.51</td>
<td>3.522e-06</td>
<td>0.6452</td>
</tr>
<tr>
<td>1</td>
<td>rs1110400</td>
<td>14036153</td>
<td>G</td>
<td>0.1566</td>
<td>0.2150</td>
<td>A</td>
<td>20.53</td>
<td>5.862e-06</td>
<td>0.6760</td>
</tr>
<tr>
<td>1</td>
<td>rs6615483</td>
<td>14974407</td>
<td>A</td>
<td>0.08333</td>
<td>0.129</td>
<td>C</td>
<td>19.72</td>
<td>8.988e-06</td>
<td>0.6136</td>
</tr>
<tr>
<td>1</td>
<td>rs3793757</td>
<td>18133379</td>
<td>A</td>
<td>0.3623</td>
<td>0.2888</td>
<td>G</td>
<td>18.57</td>
<td>1.638e-05</td>
<td>1.34</td>
</tr>
<tr>
<td>1</td>
<td>rs16861613</td>
<td>18199291</td>
<td>A</td>
<td>0.02992</td>
<td>0.07434</td>
<td>C</td>
<td>33.37</td>
<td>7.619e-09</td>
<td>0.3941</td>
</tr>
<tr>
<td>1</td>
<td>rs422422</td>
<td>33033188</td>
<td>G</td>
<td>0.2055</td>
<td>0.246</td>
<td>A</td>
<td>20.05</td>
<td>7.543e-06</td>
<td>0.7051</td>
</tr>
<tr>
<td>1</td>
<td>rs1282070</td>
<td>37314092</td>
<td>C</td>
<td>0.01582</td>
<td>0.03953</td>
<td>A</td>
<td>18.17</td>
<td>2.024e-05</td>
<td>0.3908</td>
</tr>
<tr>
<td>1</td>
<td>rs2062629</td>
<td>69601120</td>
<td>G</td>
<td>0.07628</td>
<td>0.1404</td>
<td>G</td>
<td>35.34</td>
<td>2.768e-09</td>
<td>0.52</td>
</tr>
<tr>
<td>1</td>
<td>rs1059316</td>
<td>66918744</td>
<td>C</td>
<td>0.07086</td>
<td>0.1134</td>
<td>A</td>
<td>18.67</td>
<td>1.555e-05</td>
<td>0.5961</td>
</tr>
</tbody>
</table>

6.2 Data Sources

In this chapter we list down the various data sources and exact file locations of the data that were used to obtain the annotation data stored in MAVEN.

- **SNP Meta Data** - The data available in the snp_meta_data_tbl has been compiled from various Illumina and Affymetrix chip compilations. These files can be obtained from the following location:

  ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606/GWAS_arrays/. We have used the following files to populate this table:

  - ILLUMINA.ILLUMINA_Human_1M.xml
  - ILLUMINA.HumanHap650Yv3.0.xml
  - ILLUMINA.HumanHap550v3.0.xml
  - ILLUMINA.HumanHap300v2.0.xml
  - AFFY.AFFY_6_1M.xml

This table contains about 2 million SNP records. We have also used the whole chromosome information from Affymetrix and Illumina to populate this table.
The `snp_meta_data_tbl_big` contains all SNPs defined by NCBI and has been downloaded from the NCBI ftp site. We have parsed the ASN.1 files from this site and extracted the relevant information. The ASN.1 files can be found in the following location - ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606/ASN1_flat/. Each chromosome has a well compiled list of SNPs along with annotation information. An entry for a SNP in the ASN.1 flat file is shown in the figure below with highlighted portions depicting lines from which data is extracted for our database:

![ASN.1 file example](image)

We extract the SNP rs#, chromosome base position in all three assemblies, gene names, allele change, residue change and functional classes from this file.

- **OMIM**: The OMIM ids have been extracted from the following file in NCBI: ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606/database/organism_data/OmimVar LocusIdSNP.bcp.gz This flat file contains mappings between OMIM Id and the SNP rs#. Data extracted from this table is used to update the omim_id and omim_allelic_variant columns in the `snp_meta_data_tbl` and the `snp_meta_data_tbl_big`. 
**Gene Data:** The Gene Data has been downloaded from the following file on the NCBI ftp website: [ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/mapview/seq_gene.md](ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/mapview/seq_gene.md). This file contains information about the Genes in all three assemblies Reference, HuRef and Celera. After parsing this file, we need to some post-processing which involves removing duplicate data for the genes, so as to retain only one copy of every gene, i.e. for any given gene we retain data only from one assembly. The data from this file is inserted into the gene_data table. The figure below shows the entries in this file. We are interested in those entries which identify "GENE" and do not consider those entries which identify "RNA" or "UTR" or "CDS".

![Gene Data Table](image)

We extract the taxid, chr_start, chr_stop, feature_name, feature_id and group_label columns from this file.

**Linkage Disequilibrium Data:** The LD data has been downloaded from the following location: [http://hapmap.ncbi.nlm.nih.gov/downloads/ld_data/latest/](http://hapmap.ncbi.nlm.nih.gov/downloads/ld_data/latest/). This ftp location contains LD data for all 4 types of population for all chromosomes. Since there several millions of records in these files, we upload only those records value of $r^2 \geq 0.5$ into the MAVEN database. The LD data flat file contains the following data
as shown below with the columns: bp_marker1, bp_marker2, population code, snp_marker1, snp_marker2, D', r^2, LOD, as shown in the figure below:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1523</td>
<td>1543</td>
<td>CEU</td>
<td>rs7235596</td>
<td>rs7235612</td>
</tr>
<tr>
<td>1.0</td>
<td>0.931</td>
<td>17.09</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1523</td>
<td>2183</td>
<td>CEU</td>
<td>rs7235596</td>
<td>rs9950836</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>18.9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1523</td>
<td>2716</td>
<td>CEU</td>
<td>rs7235596</td>
<td>rs9966762</td>
</tr>
<tr>
<td>0.017</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>1523</td>
<td>2842</td>
<td>CEU</td>
<td>rs7235596</td>
<td>rs10853286</td>
</tr>
<tr>
<td>0.922</td>
<td>0.851</td>
<td>13.7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1523</td>
<td>3034</td>
<td>CEU</td>
<td>rs7235596</td>
<td>rs12455984</td>
</tr>
<tr>
<td>0.703</td>
<td>0.044</td>
<td>0.76</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

We extract bp_marker1, bp_marker2, snp_marker1, snp_marker2, D', r^2 and LOD from these files.

- **Transcript Data:** Transcript ids have been extracted from the NCBI ASN.1 flat files found at: ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606/ASN1_flat/. The figure below shows a typical SNP record from these files and highlights the lines from which transcript ids have been extracted.

```
rs538 | human | W086 | snp | genotype=NO | submitter=LinkYES | updated=2009-02-15T03:02
ss542 | ORCID | Contig1 | orient=+ | ss_pick=NO
ss3137607 | LOC | 69329 | orient=+ | ss_pick=NO
ss47924509 | LOC | 699329 | orient=+ | ss_pick=NO
ss162098618 | ss_pick-GA1 | 147179 | orient=+ | ss_pick=NO
ss20498048 | SsPick-MR | W6GA-20044-chr1.chr1.WT_021937.16208017 | orient=+ | ss_pick=NO
c05648312 | ASET | NCVA008250 | orient=+ | ss_pick=NO
c52976708 | si1 | 517197.17 | 608828 | orient=+ | ss_pick=NO
ss77156528 | MRSA | Cor12567_NG_20070510.chr19952240 | orient=+ | ss_pick=NO
ss70858639 | MRSA | Cor12567_NG_20070510.chr19952240 | orient=+ | ss_pick=NO
ss91147661 | MRSA | Cor12567_NG_20070510.chr19952240 | orient=+ | ss_pick=NO
ss81154216 | MRSA | Cor12567_NG_20070510.chr19952240 | orient=+ | ss_pick=NO
ss16723482 | DCGVGDC_N4 | Jh-00990612 | orient=+ | ss_pick=NO
ss07934695 | HPMKGMRP_EVE | 1229675923210 | orient=+ | ss_pick=NO
ss187957518 | 1000GENOMES | CEU.trio.12.15.2008_7599_chr1_0003540 | orient=+ | ss_pick=NO
ss118015369 | 1000GENOMES | NA12896_2008_12_16_7873_chr1_0003540 | orient=+ | ss_pick=NO
ss11807835 | EURLUna-UK | NA12897_000007555533916_chr1_0003540 | orient=+ | ss_pick=NO
SNP | Alleles| "G+/T" | het? | se[het]?

<table>
<thead>
<tr>
<th>Validated</th>
<th>min_prob</th>
<th>max_prob</th>
<th>NOT withdrawn</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTG</td>
<td>assembly*Celera</td>
<td>chr1</td>
<td>chr-pos=525777</td>
</tr>
<tr>
<td>CTG</td>
<td>assembly*Rhef</td>
<td>chr1</td>
<td>chr-pos=530975</td>
</tr>
<tr>
<td>CTG</td>
<td>assembly*Rhef</td>
<td>chr1</td>
<td>chr-pos=530975</td>
</tr>
<tr>
<td>CTG</td>
<td>assembly*Reference</td>
<td>chr1</td>
<td>chr-pos=530975</td>
</tr>
</tbody>
</table>

LOC | KKN632 | Gubac_id=4514 | fan-classes=mur-gene-3 | anna_acc=0015212.3
LOC | KKN632 | Gubac_id=4514 | fan-classes=mur-gene-3 | anna_acc=0015212.3
```

- **KEGG Data:** has been downloaded and parsed by using the keggapi.jar offered by the KEGG website. We use the KeggLocator class in the keggapi library to access the KEGG database and download the data into a "dat" file. Once the data has been downloaded, we parse the downloaded "dat" file to upload into the pathway information into MAVEN database.
− **SNPs in miRNA target sites:** This data has been downloaded from the PolymiRTS database (PolymiRTS Database 2006) at the following location: http://compbio.utmem.edu/miRSNP/download/miRSNP_human.txt. This file maps SNP rs#, transcript id of the target site, miRNA Ids and the corresponding alleles.

− **SNPs in miRNA:** SNPs present in miRNA have been obtained by a two step procedure. We first obtained the genomic coordinates of miRNA's from the mirBase database (Griffiths-Jones, et al. 2006). Using these coordinates we queried the `snp_meta_data_tbl_big`, which contains all SNPs defined in NCBI, to obtain SNPs present within these coordinates which would in turn identify the SNPs present in miRNA's.

− **HPRD Data:** HPRD Ids have been downloaded from the following website / file: http://www.hprd.org/download/HPRD_Release_8_070609.tar.gz. This file maps gene names to the HPRD ids. These HPRD Ids are then used to link to the HPRD website and display the Protein-protein interactions of the selected gene.

### 6.3 Data Parser

The Data Parser tool (Figure 40) will be used by the administrator of MAVEN to make future updates to the annotation information stored in MAVEN's database whenever a new release of NCBI SNP build or any of the other data sources is announced. This tool has been developed using Microsoft Visual Studio .NET and C#. It is a simple Windows Form application, which provides the administrator the ability to choose the physical location of the downloaded data files and parse them. Each data source needs to be uploaded separately; hence each category is separated as a group box. The users can choose the file / folder from which data needs to be extracted. The main
The intent of the parser is to collect the necessary data from the source format and convert it into a tabular format with the first row being the column header, and delimited by a space or any special character. The output file can then be used to upload the data into the database table using the Import Data feature of Microsoft SQL Server Management Studio. This turns out to be less time consuming than inserting row by row into the database using ODBC calls from the parser application as each data source contains millions of records to be uploaded.

![Figure 40: Data Parser Tool](image)

### 6.4 Installation

In this section we describe the installation requirements for MAVEN. MAVEN currently runs on a powerful Windows 2008 Server - [cbc.case.edu](http://cbc.case.edu). The web site is hosted using the Apache Tomcat web container and the Apache Web server under the following address [http://cbc.case.edu/maven](http://cbc.case.edu/maven). The Tomcat container and Apache web
server interface with each other using mod_jk. MAVEN uses Microsoft SQL Server 2005 as its database server. JDBC calls are made to the SQL Server Database using the sqljdbc.jar which has been downloaded from java.sun.com and placed in Tomcat's library folder. The server.xml of the web application needs to be configured to access the CBC database. The version of JVM used is 1.06. Struts 1.1 and Tiles 1.1 are used for the UI.

6.5 Usage Statistics from Google Analytics

As MAVEN is a web based application, we decided to obtain usage statistics by integrating our website with Google Analytics. In this section we would like to discuss some of these results. MAVEN was integrated with Google Analytics on the 11th of January 2010. We will discuss usage statistics obtained for the period between January 11th and February 22nd, 2010.

![Figure 41: Visitors Overview](image)

The above graph displays the average number of visitors to our website for every week since January 11th. Overall, there have been 269 visits containing 181 absolutely unique visitors. The average time spent on the website has been 2.22 minutes. These visits are not limited to the United States alone. Visitors from 25 different countries have used MAVEN as shown in Figure 42.
The above figure shows the map overlay of visitors from the various countries. The maximum number of visits has been from the United States with about 108 visits, followed by France with 41 visits and Australia with 27 visits. Most of these visits are new visits as the application has been online only for a little over a month.

Also, MAVEN has been adapted to work with only Firefox and Internet Explorer at this point of time as these are the most commonly used web browsers. This point is also proven by the following usage statistic we obtained from analytics. The table below clearly shows that a majority of the users, about 86% in total, use Firefox and Internet Explorer.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Visits</th>
<th>% visits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firefox</td>
<td>165</td>
<td>61.34%</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>70</td>
<td>26.02%</td>
</tr>
<tr>
<td>Chrome</td>
<td>18</td>
<td>6.69%</td>
</tr>
<tr>
<td>Safari</td>
<td>8</td>
<td>2.97%</td>
</tr>
<tr>
<td>Mozilla</td>
<td>4</td>
<td>1.49%</td>
</tr>
</tbody>
</table>
6.6 Conclusion

In conclusion, we hope that MAVEN will be useful to biologists / researchers all over the world as it acts as portal for information gathering and analyses of SNPs in Genome-wide association studies, truly helping in understanding the relationship between complex diseases and genetic markers, and one day will become a true repository for GWA studies conducted world-wide.
Bibliography


