# Accelerating Disease Gene Identification Through Integrated SNP Data Analysis

P. Missier*, S. Embury*, C. Hedeler*, M. Greenwood*, J. Pennock†, and A. Brass*†

*University of Manchester, School of Computer Science, Manchester, UK
†University of Manchester, School of Biological Sciences, Manchester, UK

**Abstract.** Information about small genetic variations in organisms, known as single nucleotide polymorphism (SNPs), is crucial to identify candidate genes that have a role in disease susceptibility, a long-standing research goal in biology. While a number of established public SNP databases are available, the specification of effective techniques for SNP analysis remains an open issue. We describe a secondary SNP database that integrates data from multiple public sources, designed to support various experimental ranking models for SNPs. By prioritizing SNPs within large regions of the genome, scientists are able to rapidly narrow their search for candidate genes. In the paper we describe the ranking models, the data integration architecture, and preliminary experimental results.

## 1 Introduction

The integration of scientific data sets can reveal opportunities for performing new forms of data analysis that cannot be supported by individual data sets, or which would otherwise lack sufficient coverage or depth. When the details of this analysis are known in advance, then we can design the integrated schema and the necessary data transformation steps with the needs of the intended application in mind. However, in many cases, converting the scientific ideas into concrete algorithms over the data is a non-obvious task. Different approaches must be prototyped and experimented with, before the most appropriate algorithm or model can be found. This requires a more flexible approach to data integration, since we cannot afford to lose information in the integration that may turn out to be critical to the implementation of the best analysis algorithm.

A problem in the life sciences that illustrates the need for experimentation, and consequent complication of the integration process, is the identification of the genes that are responsible for phenotypes in model organisms. A phenotypic trait is some observable behaviour or disease response and includes, for example, body size and susceptibility to some disease. Many phenotypes are typically the result of complex interactions among several genes, thus posing considerable challenges to the biologist wishing to understand their genetic origins.

Establishing the relationship between phenotype and one or more regions of the genome has been a research objective for quite some time [1]. The current methodology for establishing the genes which may be responsible for a quantitative trait uses elaborate breeding schemes to identify genomic regions where sequence differences among strains of the organism under study can be correlated to differences in the phenotype of interest. These regions are known as Quantitative Trait Loci (QTLs). They vary in size

but inevitably contain many genes (100's to 1000's), all with the potential to influence the trait by some means. The challenge for biologists is then to narrow this down to a more manageable set of candidate genes, the roles of which can then be investigated using less expensive and time consuming experimental techniques.

As a result of recent research on this problem, a large number of studies identifying genetic variations within the mouse genome are now available, for many inbred strains with documented phenotypes. Each variation takes the form of a Single Nucleotide Polymorphism (SNP) — that is, a difference in a single base pair between one strain and the reference strain of the model organism. SNPs thus provide a key tool for scientists wishing to target likely candidate genes within a QTL. If a variation in phenotype (such as susceptibility to a particular disease) has a genetic cause, then there should be clear differences in the SNPs of the strains exhibiting this variation. Moreover, the locations of the SNPs within the genome can indicate the genes that play a role in determining whether an individual will exhibit the phenotype of interest or not. While it is clear that some SNPs found in QTLs are more informative than others, the precise criteria needed to isolate these SNPs are not completely clear, and their investigation is part of current research. At the same time, the sheer volume of SNPs under consideration, typically of the order of tens of thousands for a single QTL region, calls for an automation of the analysis process. Our goal is to support this exploration by providing biologists with a software environment for the semi-automated SNP analysis of SNP "informativeness".

Recognition of the value of SNPs in detecting the genes involved in specific phenotypes has fuelled the development of several publicly-accessible SNP databases. Notable among these are Ensembl [5], dbSNP [14], the Perlegen Sciences database[1], MGD [4], UCSC [8], and Wellcome-CTC Mouse Strain SNP Genotype Set[2]. Each of these resources allows the retrieval of SNPs from a given chromosome region, but they are also highly heterogeneous, in terms of access mechanisms, structure, content and quality. For example, Ensembl contains high-quality data that has been assessed by expert curators, while dbSNP contains more recent but more speculative SNPs that have not been subjected to such rigorous quality control.

In order to get a good coverage of both strains and chromosomal regions for SNP analysis, therefore, it is necessary to integrate data from several sources. Since data volumes are high (there are currently around 8 million confirmed SNPs in the mouse genome, for example), and since the various resources do not all provide suitable programmatic access to data, a materialised integration is necessary. However, at present, the main purpose of this integration is not to support a specific known application but to allow experimentation with a variety of hypothesised algorithms for assessing the likely role of a SNP in producing a given phenotypic response. We do not know at the outset what quality or coverage of SNPs will be required to provide reliable analyses of this kind. Therefore, rather than a conventional, tight integration to a fixed common schema, with "one-time" data cleaning steps, we have instead adopted a loose integration approach, which allows the user to experiment with different combinations of sources and integration approaches.

[1] Perlegen: http://www.perlegen.com/

[2] http://www.well.ox.ac.uk/mouse/INBREDS/

The first results of this experimentation have been implemented in a web-accessible database called *SNPit*. The *SNPit* database is populated with a loose integration of SNP and strain data covering the entire mouse genome. This paper describes our experiences in constructing *SNPit* and the loose integration approach that supports it. We begin, in Section 3, by describing the kinds of SNP scoring models that must be supported by a system such as *SNPit*. We then discuss the integration problems that arise and our solutions for them (Section 4), and present experimental evidence for the usefulness of the resulting scores (Section 5). Finally, Section 6 concludes and outlines our plans for further exploitation of the *SNPit* database through the discovery and implementation of additional SNP scoring models.

## 2 Related work

While many examples of data integration projects can be found in bioinformatics, it is interesting to note the increased importance of automating SNP analysis, a sign that the role of SNPs in the discovery of genes responsible for particular phenotypes is widely recognized. It is no surprise, therefore, that a number of SNP searching tools are available in the public domain. A common goal of these tools is to perform large-scale searches through genome-wide collections of SNPs, in order to narrow the genotyping analysis to a small set of "optimal" SNPs. Where the tools differ is in the specific type of search filters, the analysis features offered, and the choice of primary data sources. *SNPHunter*, for example, retrieves SNPs that lie inside or around a given candidate gene [12]. The *SNPper* application described in [11] lets the user focus on highly polymorphic regions, and filter SNPs based on their submitter (since users may attribute different reliability to SNPs coming from different submitters). Some systems, like *PolyDoms* [7] and the *SNP function portal* [13], integrate multiple data sources, but only one of these is a SNP database (dbSNP). The former provides filter options for predicted functional properties of SNPs, such as "Damaging non-synonymous SNPs", while in the latter search criteria can be expressed on a long list of annotations obtained from various other databases, e.g. at the genome, protein, pathway levels. Others, like *PupaSuite* [2] and *SNPeffect* [9], add functionality to predict the functional effect of SNPs on the structure and function of the affected protein.

We note two important differences between these tools and our *SNPit* database. Firstly, we integrate multiple sources of SNP data, allowing users to perform searches on specific sources, or to compare analysis results across sources. Secondly, since all the cited tools are specific to the human genome, SNP analysis cannot be based on observed phenotype differences among strains (because no collections of strains are available for humans). In contrast, by targeting the mouse (an important model organism), we are able to exploit the complete genome sequencing of different mouse strains, along with the growing number of available QTLs already identified for the mouse. One secondary mouse SNP database, called *Mouse SNP Miner*, is indeed described in the recent literature [10]; but it is designed to perform batch analysis of potential damaging effect of SNPs, rather than for interactive search.

## 3 Capturing SNP "informativeness"

As mentioned, SNPs allow us to identify sets of candidate polymorphic genes within a QTL region which may be responsible for the disease response (or other behaviour) observed in various strains. The main intuition behind this process is the following: since different strains of the model organism exhibit the phenotype in different ways, if we can identify the regions of greatest genetic difference between those strains then we can prioritise the genes that are located in those regions for further investigation. In other words, we would like to rank the SNPs within a QTL in some way that indicates the likelihood that it contributes to the phenotypic differences observed between strains. From this, we can create a secondary ranking on the genes in which the SNPs appear.

In order to perform this ranking reliably, we need to gather together information about as many known SNPs in the QTL as possible. Since no one database, at present, can guarantee to provide this, we must collect data from several databases and integrate it. Unfortunately, there is no single way to translate the biologists' intuition regarding the informativeness of SNPs in identifying candidate genes into a procedure precise enough to be implemented in software. Therefore, we have proposed several different variants on the basic score model. The integrated data must be able to support experimentation with all these variants, so that their relative reliabilities can be explored.

The basic score model compares, for each SNP, the nucleotide base replacement observed in a single, user-selected strain, i.e., the strain that exhibits the phenotype under investigation, with those that occur in all other known strains. Each such alternative base is called an *allele*. A SNP in which the allele for the selected strain is different from that observed in all the others supports the hypothesis that the SNP plays a role in the phenotype associated with the selected strain; the SNP should therefore receive a high score.

To make this intuition precise, consider the set $\mathcal{S} = \{S_1, ..., S_N\}$ of all known mouse strains (about 60) for which SNPs have been sequenced. Ideally, we would like to have allele information about each SNP in the entire genome for each of the strains, i.e., $A_{i,j} \in \{\mathtt{G}, \mathtt{C}, \mathtt{A}, \mathtt{T}\}$ for each SNP $i$ and strain $S_j$. In reality, sequencing efforts focus on particular genome regions and on particular strains, so that this information is missing for some strains on some SNPs – we indicate missing alleles with $A_{i,j} = \mathtt{N}$. Note that the set $\mathcal{A}_i = \{A_{i,j}, j : 1..N\}$ of all alleles for a SNP is a bag, rather than as set, because alleles from different strains may coincide, as shown in the example of Table 3(a).

In the basic score model, the user selects a single strain $S_{ref} \in \mathcal{S}$ as the reference. For each SNP $i$, we compute a base score $s_{i,0}$ as the number of non-null alleles $A_{i,j} \neq \mathtt{N}$ that are distinct from the reference, or $j \neq ref$ and $A_{i,j} \neq A_{i,ref}$. This is then normalized by the number $n'_i$ of non-null, distinct alleles $A_{i,j}$ for each $j \neq ref$, to yield the final score:

$$s_{i,ref} = s_{i,0}/n'_i$$

This model gives a high score to SNPs for which the selected strain has a unique allele but where all other alleles are the same. Consider the example of Table 3(a), available from Perlegen for SNP `rs61647296` on chromosome 12. For selected strain `A/J`, the

allele G is indeed unique ($s_0 = 9$ because 9 non-reference strains have allele T), and furthermore, the only other known allele is T ($n' = 1$). This yields a score $s_{A/J} = \frac{s_0}{n'} = 9$. For comparison, in the SNP in Table 3(b) (rs61646963), the score for the reference strain (allele G) is only 0.5, because the allele appears among the non-selected strains, only one allele (A for strain BALB/cByJ) is different, and the non-selected strains contain two distinct values.

**Table 1.** Strains and alleles example 1

**(a)**

| Strain | StrainAllele |
|---|---|
| DBA/2J | T |
| A/J | G |
| BALB/cByJ | T |
| C3H/HeJ | N |
| AKR/J | T |
| FVB/NJ | T |
| 129S1/SvIm | N |
| NOD/LtJ | T |
| WSB/EiJ | N |
| PWD/PhJ | T |
| BTBR T+ tf | N |
| CAST/EiJ | T |
| MOLF/EiJ | T |
| NZW/LacJ | N |
| KK/HlJ | N |
| C57BL/6J | T |

**(b)**

| Strain | StrainAllele |
|---|---|
| DBA/2J | N |
| A/J | G |
| BALB/cByJ | A |
| C3H/HeJ | G |
| AKR/J | N |
| FVB/NJ | N |
| 129S1/SvIm | G |
| NOD/LtJ | N |
| WSB/EiJ | N |
| PWD/PhJ | N |
| BTBR T+ tf | G |
| CAST/EiJ | N |
| MOLF/EiJ | N |
| NZW/LacJ | G |
| KK/HlJ | G |
| C57BL/6J | G |

In summary, this simple model rewards SNPs where the selected strain is unique, and the alleles for all other strains are the same. Note that the score is 0 for SNPs where the allele is missing for the selected strain.

The second score model, called the *group score model*, generalises the first by allowing the comparison of two user-selected groups of strains, rather than comparing a single strain against all others. This is useful because it is often the case that a particular phenotype is observed in more than one strain. For example, it is common to want to compare strains which are known to be susceptible to a particular disease with those strains which are known to be resistant. There may be other strains for which we do not know the phenotype, and these should be excluded from the analysis. This score therefore rewards SNPs for which (i) the sets of alleles in the two selected groups are disjoint, and (ii) the alleles for each individual group are homogeneous — in the ideal case, the strains in one group will all exhibit one allele, while the strains in the other group all exhibit another allele.

Consider two disjoint sets of strains $\mathcal{S}_1 = \{S_1, ..., S_n\}$ and $\mathcal{S}_2 = \{S'_1, ..., S'_m\}$, and, for a given SNP, the corresponding bags of alleles $\mathcal{A}_1 = \{A_1, ..., A_n\}$ and $\mathcal{A}_2 = \{A'_1, ..., A'_m\}$ (the SNP index $i$ is omitted for simplicity). Let $\delta$ be the number of distinct, non-null alleles that are common to $\mathcal{A}_1$ and $\mathcal{A}_2$: $\delta = |\mathcal{A}_1 \cap \mathcal{A}_2|$, and $n'$, $m'$ the number of distinct alleles in $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. We define three variations for the group score. The simplest takes the form:

$$gs_0(\mathcal{A}_1, \mathcal{A}_2) = 1 - \frac{\delta}{n' + m'}$$

This model rewards disjoint sets of alleles, regardless of their internal homogeneity. Note however that, when using $gs_0$, one SNP for which one entire group of alleles is null gets a perfect score, because $\delta = 0$ in that case. This seems counter-intuitive, i.e., it would be misleading to give a high rank to SNPs for which a score simply cannot be computed. To counter this effect, the second variation of the model, $gs_1$, extends $gs_0$ by introducing penalty factors with values proportional to the number of null alleles in the groups under consideration:

$$p_1 = \frac{|\{A_j \in \mathcal{A}_1 | A_j = \texttt{N}\}|}{|\mathcal{A}_1|}$$

($p_2$ is defined similarly for $\mathcal{A}_2$). The resulting adjusted score is

$$gs_1(\mathcal{A}_1, \mathcal{A}_2) = gs_0(\mathcal{A}_1, \mathcal{A}_2) \cdot p_1 \cdot p_2$$

**Table 2.** Strains and alleles example 2

| **(a)** | | **(b)** | |
|---|---|---|---|
| **Strain** | **StrainAllele** | **Strain** | **StrainAllele** |
| DBA/2J | G | DBA/2J | A |
| A/J | A | A/J | N |
| BALB/cByJ | G | BALB/cByJ | A |
| C3H/HeJ | A | C3H/HeJ | N |
| AKR/J | G | AKR/J | N |
| FVB/NJ | G | FVB/NJ | A |
| 129S1/SvIm | A | 129S1/SvIm | N |
| NOD/LtJ | G | NOD/LtJ | A |
| WSB/EiJ | G | WSB/EiJ | N |
| PWD/PhJ | G | PWD/PhJ | T |
| BTBR T+ tf | A | BTBR T+ tf | N |
| CAST/EiJ | G | CAST/EiJ | A |
| MOLF/EiJ | G | MOLF/EiJ | N |
| NZW/LacJ | A | NZW/LacJ | N |
| KK/HlJ | A | KK/HlJ | N |
| C57BL/6J | G | C57BL/6J | A |

Note that the values of penalties decrease as expected (because they are multiplying factors) when the number of null alleles increases. Consider the example in Table 3(a), and the two groups $\{\texttt{A/J}, \texttt{BALB/cByJ}\}$ and $\{\texttt{AKR/J}, \texttt{C57BL/6J}\}$, corresponding to allele groups $\mathcal{A}_1 = \{\texttt{A}, \texttt{G}\}$ and $\mathcal{A}_2 = \{\texttt{G}, \texttt{G}\}$. We have $\delta = |\{\texttt{G}, \texttt{G}\}| = 1$, $n' = n = 2$, $m' = 1$, and $gs_1(\mathcal{A}_1, \mathcal{A}_2) = \frac{2}{3}$, with no penalties since there are no missing alleles. The effect of penalties can be observed in the example of Table 3(b), where the alleles for $\texttt{A/J}$ and $\texttt{AKR/J}$ are missing. Here $p_1 = p_2 = \frac{1}{2}$, and $gs_1(\mathcal{A}_1, \mathcal{A}_2) = (1 - \frac{1}{2}) \cdot p_1 \cdot p_2 = \frac{1}{8}$.

The third variation of this model accounts for the *heterogeneity* of each of the two groups, represented by the elements $h_1 = \frac{n'}{n}$ and $h_2 = \frac{m'}{m}$. The resulting score:

$$gs_2(\mathcal{A}_1, \mathcal{A}_2) = \frac{gs_0(\mathcal{A}_1, \mathcal{A}_2)}{h_1 + h_2}$$

is lower for highly heterogeneous groups. Using $gs_2()$, the score for the example of Table 3(b) would become $\frac{4}{9}$, because $h_1 = 1$, $h_2 = \frac{1}{2}$. It is possible, of course, to

combine $gs_1()$ and $gs_2()$ to take into account both penalties and group heterogeneity. Note also that the scores do not take into account the number of strains in each group, which is typically very small.

Preliminary results on the performance of one of these models, $gs_1()$, are presented in Section 5.

## 4 Gathering and Integrating SNP Data

From the SNP analysis described in the previous session, we derive a number of requirements and design decisions for the management of SNP data. First of all, there is choice of publicly-accessible databases containing SNP data for specific organisms — including the mouse. These databases partially overlap in structure and content, depending on the submission policy and procedures of the controlling organization. The update frequency of the data, and thus its currency, also varies. Users tend to choose among the available data sources based on their prior confidence in its reliability, possibly cross-referencing the retrieved data with other sources for validation afterwards.

There is currently no single reference data source for SNP data, and therefore SNPs from multiple sources must be combined in order to achieve the coverage levels required by the score models described in the previous section. We have selected three of the most prominent public SNP databases, on the basis of their completeness, authoritativeness, and the complementarity of their respective content. First is Ensembl Mouse[3], a well-known source for the mouse genome, which is regarded as being of high quality thanks to the team of expert curators who make sure that only confirmed and established data is included. The second database is dbSNP, maintained by NCBI[4]; the quality of its data is known to be less consistent, since the submission process involves relatively little quality control. This, at the same time, makes dbSNP a good source for recently discovered SNPs. Thirdly, we have selected the database from Perlegen Sciences, the result of a project devoted specifically to sequencing the whole mouse genome across 15 mouse strains with high accuracy.

Programmatic access to these data sources is provided through a range of different mechanisms, including Web services (for instance, NCBI's eUtils), direct data layer access (Ensembl accepts public connections to its mySQL database) and through bulk data download. Since each region-wide SNP analysis involves retrieving and joining data sets of the order of tens of thousands of elements, followed by the execution of ad hoc algorithms, the performance of frequent bulk queries on the remote sources is likely to be poor. There is thus a need for some form of data localization, and the potential for developing further of analysis algorithms also requires the design of an integrated schema.

### 4.1 Data Integration Approach

These considerations led us to the design of a new database for SNPs, called *SNPit*, which consolidates data from the three data sources mentioned. Unlike typical OLAP

[3] Ensembl Mouse genome: http://www.ensembl.org/Mus_musculus/

[4] dbSNP: http://www.ncbi.nlm.nih.gov/SNP/

integration projects, the new schema is designed so that individual relations are very similar in structure to the corresponding relations in the source schemas. In practice, the database consists of a collection of materialized views on the sources, which can be pairwise joined through the use of common identifiers for the SNPs. A sketch of the data integration scenario appears in Figure 1, where the flow of SNP data across the sources is highlighted (top half). Perlegen SNPs are gradually being submitted to dbSNP, making this one of its major contributors (although the process is not yet complete).[5] In turn, data from dbSNP is gradually incorporated into Ensembl, through a slower curation process. Ensembl also includes SNP data that has been discovered by the ongoing sequencing work of the Sanger Institute in the UK.[6] As the figure shows, independent loading procedures processes have been setup for each of the three sources, using various offline data transformation techniques. As a result, we expect some of the Perlegen SNPs to appear in our dbSNP and Ensembl tables.

Maintaining separate sets of relations for each data sources has several advantages. Firstly, by directing their queries to views on a specific source, users may limit the scope of their analysis to familiar data. Secondly, overlapping SNPs from different sources are retained as separate data items, thus avoiding the problem of having to resolve all possible inconsistencies (eg different alleles detected for the same SNP and strain) upon loading. Also, tracking the correct propagation of the same SNP information from one database to the next can be done at the application level. Thirdly, both dbSNP and Ensembl are subject to ongoing revision and using separate relations makes the reloading of updated versions more manageable. Finally, since there is built-in redundancy in the loosely integrated schema, additional data sources with partially overlapping data may be added without disrupting the schema. One minor shortcoming of this approach is the need to create additional views for each useful combination of sources that are frequently queried together. The schema is designed to model the following main aspects of SNP data:

- the one-to-many relationship between a SNP and the strains in which it is known to occur. The number of strains alleles available for each SNP varies, in Ensembl, between 1 and over 60, depending on the sequencing effort carried out by the originating lab. In general, we expect that the more alleles are available, the better the chance of correlating the SNPs to phenotype differences among the strains;
- the position of the SNP, expressed as the number of bases from the start of a chromosome. This translates into a one-to-many relationship between a gene (whose position is identified by an interval of bases within a chromosome) and the SNPs that occur within its boundaries.[7]
- SNP provenance, i.e., the submitter institution along with the version of the genome used to specify the SNP position (called "build"), and other similar data;
- SNP *location*, i.e., whether the SNP occurs in a DNA fragment that is involved in the translation process for protein synthesis (a *coding* region), or in a non-coding

[5] Perlegen currently contributes about 44% of the dbSNP SNPs.

[6] http://www.sanger.ac.uk/

[7] SNPs may also occur in between genes. For this reason, we have complemented the collection of genes with a set of labels corresponding to the intergenic regions, for the purpose of our study.

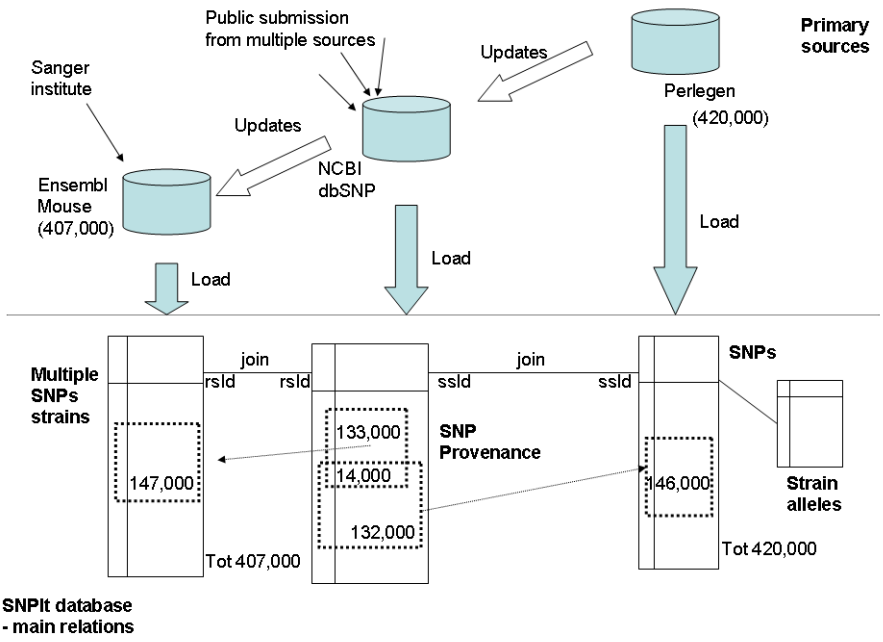



**Fig. 1.** Primary sources and main relations for the SNPit database. The number figures are data volumes for a single chromosome (12)

region. This is relevant in assessing the potential consequences of a single-base mutation.

Both the Ensembl and the Perlegen views include SNP-to-strain and SNP-to-gene relationships, and are used to calculate the score models for data in these sources. Native Perlegen data does not include gene information, however, and we have had to add it to our database separately, as part of the loading process. This was done using the mouse genome in the Ensembl gene database. The provenance data is currently being used in a separate study concerning methods to assess the reliability of SNPs (as opposed to their "biological informativeness"), and is not discussed further in this paper. We plan to exploit location data to improve upon our current score models for SNPs, as explained in our conclusions section.

Successful joins in our schema rely upon the use of common SNP identifiers. Unfortunately, SNPs are given different types of identifier at different stages of their "acceptance" (they are also known by different names, as described in [3]). While a reference ID "`rsId`" (for instance `rs61647296`) is issued by Ensembl curators for accepted SNPs, Perlegen uses its own private naming scheme. To complicate matters still further, dbSNP makes a distinction between the SNP reference ID (when available) and the *submitter ID* `ssID`, issued by dbSNP at the time the SNP is entered into the database. The purpose of using reference IDs is to represent SNPs that have been identified by more

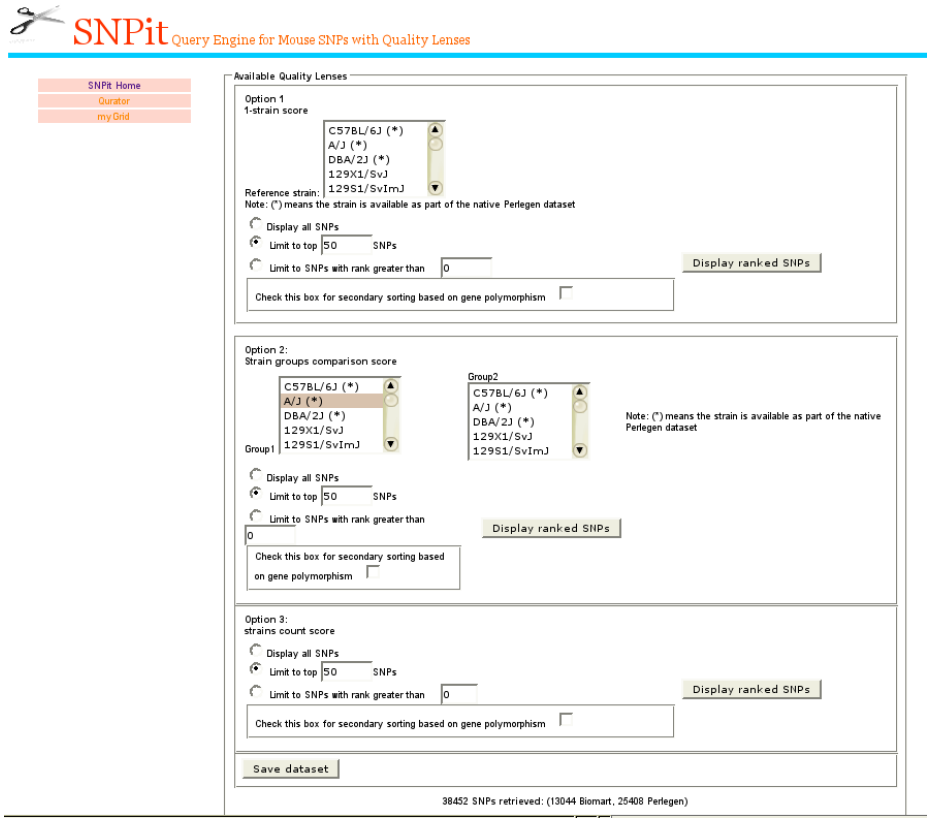



**Fig. 2.** Score model selection in the SNPit application

than one lab, using a submitter-independent numbering scheme. This complicates the task of tracking multiple occurrences of the same SNP in our schema, since, for example, only the Perlegen SNPs that have already reached Ensembl will have an `rsID`. The bottom part of Figure 1 shows how `rsID` and `ssID` are used in combination with the dbSNP view, to mediate between the Ensembl and Perlegen views. This means that the scope of a comparative analysis over the SNPs that occur in both views is limited to the subset indicated by the dotted box. The numbers in the figure provide an example, for one sample chromosome, of the amount of overlapping SNPs among the sources.

### 4.2 The *SNPit* web application

The *SNPit* MySQL database contains the entire set of known mouse SNPs from the three sources. A Web application (written using JSP technology) makes the score models available to end users. The application allows the biologist to (i) select SNPs for a region of interest, eg. an entire QTL, or for a set of individual genes, with some filtering capability, for example by selecting SNPs that belong to highly polymorphic regions; and (ii) repeatedly apply various score models on this selection. The available scoring

options are shown in Figure 2. At this stage, the application has already fetched about 38,000 SNPs from both Perlegen and Ensembl, for a user-specified region[8]. Next, users may select a score model (three of them are available in the Web form), along with the strain or strain groups they wish to analyse (Figure 2).

Once the user has selected their preferred ranking method, the SNPs are retrieved, scored and displayed, as illustrated in Figure 3. The ranked SNPs are shown in the table on the left (with the name of each associated gene shown in the leftmost column). On the right of the main SNP result table, we also show histograms of the score distribution for the returned SNPs. Ideally, we would hope to see a highly skewed distribution, with most of the SNPs receiving low scores, but with a long thin tail showing a small number of high scoring SNPs. In order to assist the user in understanding the characteristics of this tail, we also display the histogram using a log-linear scale, which amplifies the results in the tail. The application is scheduled to be released for public access in

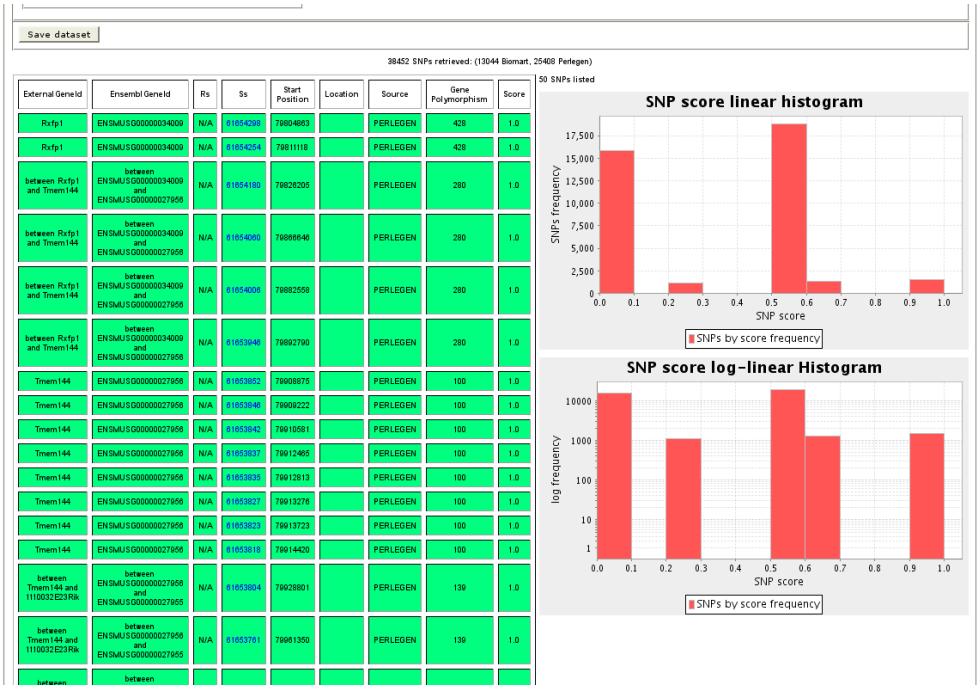



**Fig. 3.** Ranked SNPs in the SNPit application

the near future. In addition, Web Service access to the analysis functionality is also being implemented. This will make the score models available to scientific workflow

[8] The Ensembl data source is known in the application as "Biomart", since the Biomart version of the data has been used to populate the view. Biomart (http://www.biomart.org) is an open source project that makes data available as a data mart for analysis purposes.

applications, i.e., as part of the myGrid suite of services, which includes the Taverna workflow management system [6].

## 5 Experimental Evaluation of the Score Models

The integrated views of SNP data that we have created for the *SNPit* application are only of value if they can support experimentation with different score models. In this section, we describe how we have evaluated the $gs_1()$ model over the integrated views.

### 5.1 Experiment design

The goal of the experiments was twofold: firstly, to assess the performance of the $gs_1()$ model, i.e., to determine how well the resulting SNP ranking reflects an expert's judgment of their informativeness. More importantly, we also wanted to test the hypothesis that the SNP ranking induces a meaningful ranking on the genes themselves, by placing the best candidates at the top with sufficiently high accuracy.

The $gs_1()$ model was evaluated using three independent, manually selected test data sets consisting of SNPs from three separate, highly polymorphic QTL regions on the mouse genome, two on chromosome 12, with a size of 23 (denoted as `Chr12-A`) and 6 Mbases (`Chr12-B`), respectively, and one on chromosome 17 (8.2 Mbases), denoted as `Chr17`. In the experiment, the biologist selected a limited number of SNPs from a few genes that are known to be good candidates for a particular phenotype. The selection was made based on the known difference in phenotype between two groups of strains; the same two groups were then used to assign a $gs_1()$ score to all the SNPs in the selected regions. The main limitation factor for the size of the test sets, as is usually the case, is the amount of effort required to manually sift through the SNPs (the number of SNPs found in each of these regions ranges in the tens of thousands, as shown in Table 3).

### 5.2 SNP-level performance

A common way of assessing the performance of a score model is to compare the computed ranking with a correct binary classification (i.e., interesting vs. non-interesting) for a test data set. The performance can then be expressed in a standard way using a ROC curve, in which the ratios of false positives to true positives are plotted for various ranking thresholds.

| Data set | | SNPs in region | threshold criteria | SNPs above threshold | above threshold / total SNPs (selectivity) | genes count |
|---|---|---|---|---|---|---|
| Chr12A | Ensembl | 73242 | score > 0 | 231 | 0.3% | 81 |
| | Perlegen | 82281 | score >= 2/3 | 2656 | 3% | 145 |
| Chr12B | Ensembl | 13044 | score > 0 | 189 | 1.4% | 57 |
| | Perlegen | 25408 | score = 1 | 1471 | 5.8% | 111 |
| Chr17 | Ensembl | 40572 | score > 0 | 2849 | 7% | 64 |
| | Perlegen | 18916 | score = 1 | 2062 | 11% | 323 |

**Table 3.** SNPs and genes volume for the experiment QTL regions

In our case, two problems complicate this procedure. Firstly, biologists find that providing positive examples, i.e. for "definitely interesting SNPs", is much easier that providing negative examples. This reflects the nature of the experimental process, whereby the initial, large set of SNPs are all considered potentially interesting, and experimental evidence as well as prior experience is applied to make some of them stand out as genuinely important. Thus, while it is natural for the expert to indicate that a data element is of interest, ruling it out completely seems harder. The second problem is the high cost of manual SNP analysis, which results in a small test set (less than 100 SNPs for each of three experiments).

Given these limitations, we decided to perform only an informal SNP analysis, and instead invest additional expert time into higher-level gene-level analysis. Thus, we only count the user-selected SNPs which are found towards the top of our ranking (the true positives), normalized by the total number of user-selected positives. These rates are greater than 95% throughout (details are omitted due to space constraints), with the exception of one of the three experiments. In that case, SNP information was simply missing from the Ensembl database at the entire gene level. The ability to perform the same analysis on alternate data sources for the same region proved important in this case; indeed, the corresponding rate for the Perlegen SNPs, our second source, is unsurprisingly high.

### 5.3 Gene-level performance

In the second part of the performance assessment, the genes corresponding to the test SNPs were compared to the genes for the top-ranked SNPs. As we have mentioned, not all SNPs occur within genes – many occur in between genes, and indeed, these SNPs may be among the most important, since some of these inter-genic regions are responsible for controlling the transcription rates of the neighbouring genes. We use labels of the form "between X and Y" to record the location of each such SNP; these labels count as actual genes for the purposes of our study.

The comparison of the automatically and manually ordered genes was performed as follows. For each of the three test data sets (i.e. regions), the entire set of genes for that region was ranked according to the underlying ranking of their corresponding SNPs, using a novel metric that we call *density of interesting SNPs*. Specifically, suppose that $X$ SNPs are known for gene $G$, and that $x$ SNPs out of the $X$ are above a given threshold $t$, applied to the computed ranking. We say that $G$ has a density $x/X$ of interesting genes at threshold $t$. This choice of ranking metric follows the intuition that, from a biology perspective, a gene whose SNPs are considered for the most part informative, according to our definition based of strain differences, has a higher chance of explaining the phenotype than genes with only few interesting SNPs.

As in the case of the SNPs, we were again only given positive examples of strong candidate genes by the biologist, making it difficult to estimate the number of false negatives. In this case, however, the number of genes is much smaller than the number of SNPs (less than one hundred for each experiment). Thus we can afford to have our biologist manually analyse the top-ranked genes, in order to identify *additional positives* that may have escaped attention due to the size of the original list of genes. These

represent the real added-value information to the biologist: interesting genes that have been spotted only thanks to the ranking model.

Thus, our performance model is based on a two-step process, whereby the expert first provides an initial list of positive examples, which is used to plot a ROC curve where all the non-selected genes are assumed to be negatives. This is a pessimistic estimate, because each non-selected gene in the top ranks counts as a false positive. Then, the expert identifies additional positives from the list. These count as true positives if they lie above the threshold, and as false negatives otherwise. Although non-selected genes are again considered negatives, the new curve obtained from this list is more realistic.

Concerning the choice of threshold $t$ used to compute the gene SNP density, we observe that the model only assigns a handful of scores from the available [0,1] interval, namely 0, .25. .5, .67, and 1, effectively creating a discrete classification. This is due to the very small size (2) of the strain groups selected by the analyst for the comparison[9], which limits the possible overlaps among the alleles. By observing the frequency of occurrence of each score over all SNPs, we may select a suitable threshold that captures the majority of them – this is typically score = 1 for Perlegen, and score $> 0$ for Ensembl. With this assumption, we compute interesting SNP density as the ratio of SNPs that are above the threshold, to the total SNPs for the gene.

The resulting curves for each of the three experiments and for the two data sources are shown in Figure 4. In this type of chart, good results are represented by curves that rapidly reach the upper left corner, representing a region of many true positives and few false positives. Although not conclusive, our preliminary results are promising. The first chart shows the improvement of the additional expert selection (indicated as "second round"). This effect seems to be reverted in the last chart; this may be due to the relatively large number of false negatives, i.e., interesting genes with low ranking. The initial, subjective reaction from our users is that this level of accuracy may already be sufficient to significantly accelerate the search for candidate genes.

We are now experimenting with further score models that exploit some of the additional information associated to the SNPs, notably whether the SNP occurs in a coding region of the gene, and whether the base substitution actually causes a change in the corresponding amino acid. This additional knowledge can be used to improve upon our models, for example by adding weight factors to SNPs. Most of the required information for this study is already captured in our schema.

## 6 Conclusions

The problem of correlating phenotype with genotype information is important to determine the genetic cause of diseases. SNPs play an important role in current methodology, but their high volume limits the potential for their exploitation.

In this paper we have described an approach to partially automate SNP analysis, based on a data integration architecture that makes it easy to implement ranking models

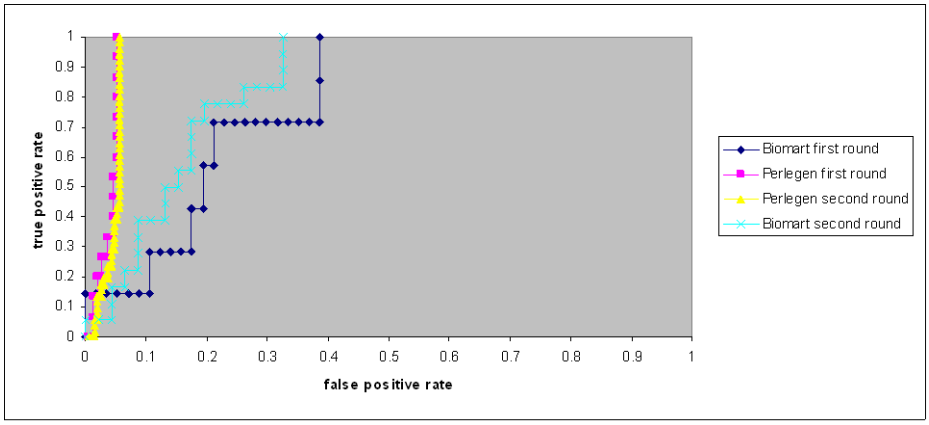
[9] This could be due to the complexity involved in manual analysis when larger groups are chosen, and we expect that automated support will encourage the biologists to investigate analyses involving more strains.

on large collections of SNPs, using multiple data sources. In our loose integration approach, we begin by capturing the essential attributes of SNPs as views on the primary sources, and then materialize the views into our new *SNPit* database.

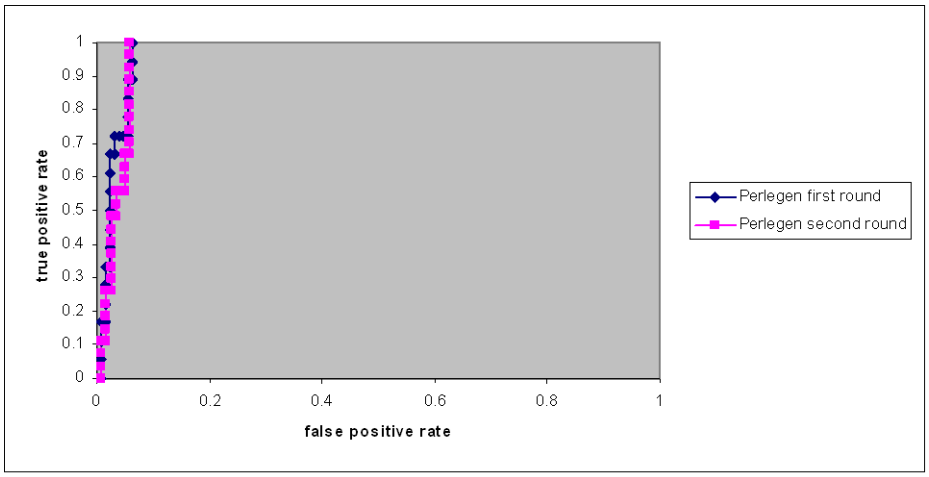
We have shown encouraging experimental results for the initial SNP ranking models implemented using the database. We are now experimenting with more elaborate models, that take into account the relative importance of individual SNPs, e.g. based on their location in the genome, as well as provenance information to assess their trustworthiness.

## References

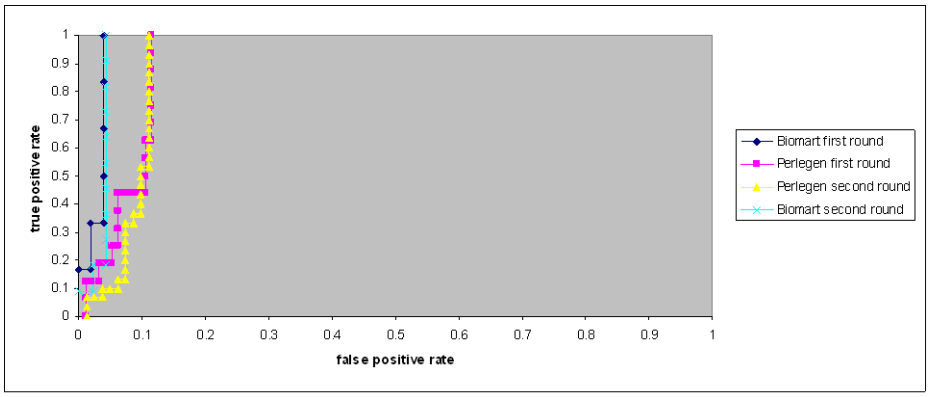

1. A. Chakravarti. Population genetics – making sense out of sequence. *Nature Genetics*, 21(Suppl. 1), January 1999.
2. L. Conde, J. M. Vaquerizas amd H. Dopazo, L. Arbiza, et al. PupaSuite: finding functional single nucleotide polymorphisms for large-scale genotyping purposes. *Nucleic Acids Res*, 34:W621 – W625, 2006.
3. A. Coulet, M. Smaïl-Tabbone, P. Benlian, A. Napoli, and M. Devignes. SNP-Converter: An ontology-based solution to reconcile heterogeneous SNP descriptions for pharmacogenomic studies. In U. Leser, F. Naumann, and B. A. Eckman, editors, *DILS*, volume 4075 of *Lecture Notes in Computer Science*, pages 82–93. Springer, 2006.
4. J. T. Eppig, J. A. Blake, C. J. Bult, J. A. Kadin, J. E. Richardson, and the Mouse Genome Database Group. The mouse genome database (MGD): new features facilitating a model system. *Nucl. Acids Res.*, 35(Database issue):D630–D637, 2007.
5. T. J. P. Hubbard, B. L. Aken, K. Beal, et al. Ensembl 2007. *Nucl. Acids Res.*, 35(suppl_1):D610–D617, 2007.
6. Duncan Hull, Katy Wolstencroft, Robert Stevens, Carole Goble, Matthew Pocock, Peter Li, and Tom Oinn. Taverna: a tool for building and running workflows of services. *Nucl. Acids Res.*, 34(Web Server issue):W729–W732, 2006.
7. A. G. Jegga, S. Gowrisankar, J. Chen, and B. J. Aronow. PolyDoms: a whole genome database for the identification of non-synonymous coding SNPs with the potential to impact disease. *Nucleic Acids Res*, 35:D700 – D706, January 2007.
8. R. M. Kuhn, D. Karolchik, A. S. Zweig, H. Trumbower, et al. The UCSC genome browser database: update 2007. *Nucl. Acids Res.*, 35(Database issue):D668–D673, 2007.
9. J. Reumers, S. Maurer-Stroh, J. Schymkowitz, and F. Rousseau. SNPeffect v2.0: a new step in investigating the molecular phenotypic effects of human non-synonymous SNPs. *Bioinformatics*, 22(17):2183 – 2185, 2006.
10. E. Reuveni, V. E. Ramensky, and C. Gross. Mouse SNP miner: An annotated database of mouse functional single nucleotide polymorphism. *BMC Genomics*, 8(24), 2007.
11. A. Riva and I.S.Kohane. SNPper: retrieval and analysis of human SNPs. *Bioinformatics*, 18(12):1681–1685, 2002.
12. L. Wang, S. Liu, T. Niu, and X. Xu. SNPHunter: a bioinformatic software for single nucleotide polymorphism data acquisition and management. *BMC Bioinformatics*, 6(60), 2005. doi:10.1186/1471-2105-6-60.
13. P. Wang, M. Dai, W. Xuan, R. C. McEachin, A. U. Jackson, L. J. Scott, B. Athey, et al. SNP function portal: a web database for exploring the function implication of SNP alleles. *Bioinformatics,*, 22(14):e523 – e529, 2006.
14. D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, et al. Database resources of the national center for biotechnology information. *Nucl. Acids Res.*, 35(Database issue):D5–D12, 2007.




(a) Chr17




(b) Chr12-A




(c) Chr12-B

**Fig. 4.** ROC curves for gene-level scores