Chapter 4

Development of a proteogenomics pipeline

4.1 Introduction

In chapter 3 the development and evaluation of Mascot Percolator was discussed, a powerful peptide scoring scheme with an implementation that can be automated and run in batch-mode, providing high-throughput capability. This system delivers sensitive peptide identification with accurate significance measures and thereby sets the foundation for a reliable proteogenomic pipeline. Mascot Percolator results can be written into a tab delimited text file or exported into the proprietary Mascot results file format. To use these data for genome annotation purposes, a proteogenomic pipeline is required that stores these data, maps the peptide identifications to the genome and enables comprehensive data analysis.

The currently available proteogenomics pipelines, Peptide Atlas (Desiere *et al.*, 2005) and GAPP (Shadforth *et al.*, 2006), which were described in detail in section 1.3, were found to be not suitable since these systems are highly specialised and the code bases are not in the public domain. The envisaged system should integrate peptide identifications available from Mascot Percolator, enable validation and refinement of Ensembl (Hubbard *et al.*, 2002) and Vega (Ashurst *et al.*, 2005) annotation, and provide a modular implementation together with a small code base for easy maintenance. Moreover, the pipeline should automatically and readily map the available peptide identifications onto the genome and provide comprehensive means to analyse and visualise these data. This chapter discusses the pipeline development, design and its individual components. Chapter 5 applies this pipeline in a proteogenomics pilot study.

Parts of this chapter will be published together with the next chapter by the author of this thesis (Markus Brosch) and my thesis supervisors (Tim Hubbard, Jyoti Choudhary).

4.2 Pipeline design and development

Figure 4.1 illustrates the pipeline design with its components. At the core of the system is a relational database "GenoMS-DB", which integrates all *in silico* digested peptides, each of which is associated with its genomic context. This approach offers several advantages:

- Non-redundant peptide-level FASTA files can be constructed, enabling more efficient Mascot searches.

- Gene level FASTA files can be constructed for gene centric viewing of Mascot results. Optionally, a peptide list that comprises peptides unique to a gene or gene isoform supports targeted proteomics experiment.

- Peptides identified with Mascot and processed with Mascot Percolator can be flagged and linked in the database with experimental and scoring information.

- The database provides readily available peptide-genome mapping, allowing immediate and direct mapping of peptides onto the genome.

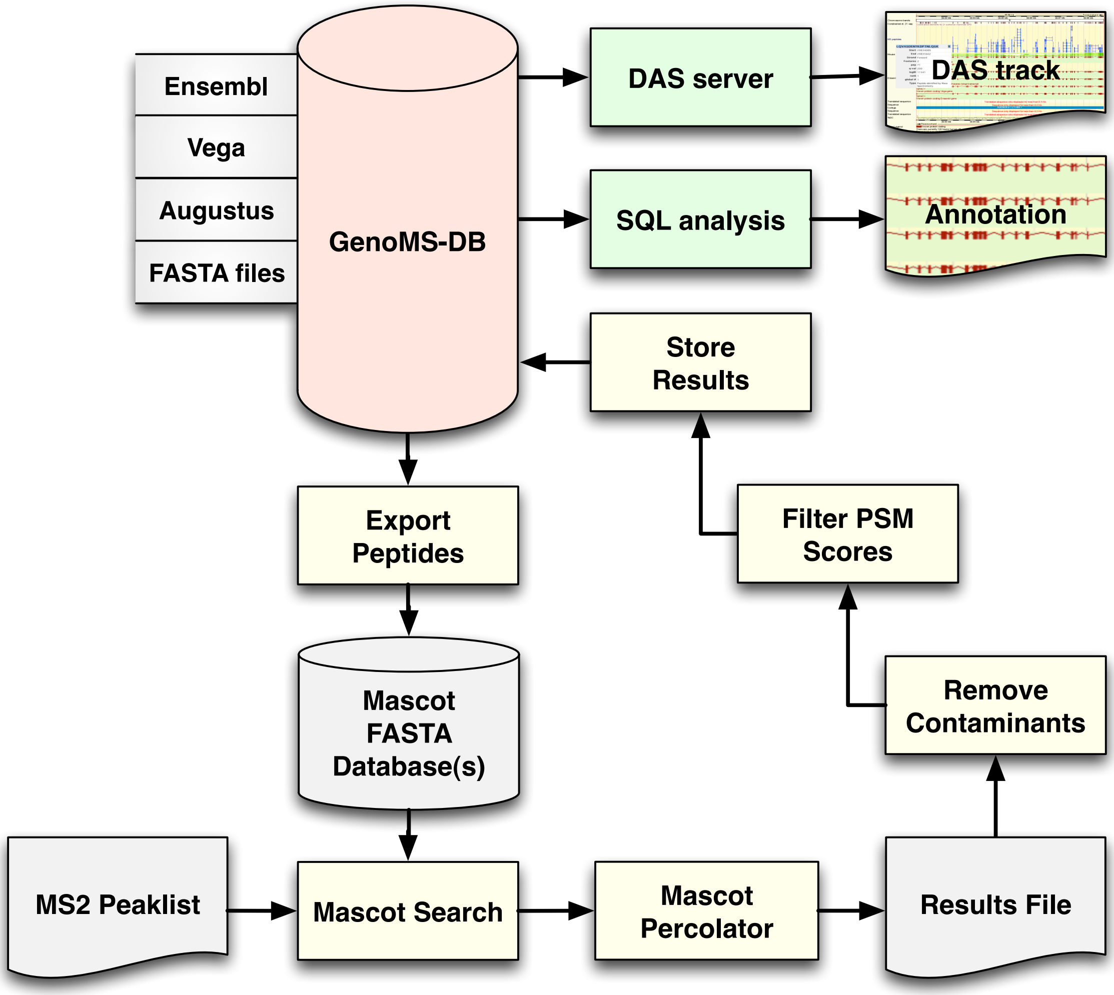


Figure 4.1: Schematic overview of the proteogenomics pipeline. The database at the core of the system, "GenoMS-DB", is built by integrating all peptides that are derived from an *in silico* digestion of available data sources. These can comprise Ensembl, Vega or Augustus gene predictions. Each peptide derived from these data-sources is associated with its genomic locus and context, such as gene, transcript, exon or splice site information. Peptides from FASTA protein databases can optionally be integrated, but lack genome mapping.
GenoMS-DB is then used to export a set of all non-redundant *in silico* digested peptides, which are used by Mascot as a search database. Tandem MS spectra are searched with Mascot and post processed with Mascot Percolator. This is followed by removing common contaminant sequences and low scoring peptide-spectrum matches (PSMs) from the results, prior to storing the remaining identifications into GenoMS-DB database. This integration of peptide-genome mapping together with peptide identifications enables streamlined analysis with standard SQL or visualisation via a DAS feature server.

- Comprehensive data analysis can be performed with fast and efficient Structured Query Language (SQL), a standard language for accessing and querying databases.
- The availability of the complete peptide-genome mapping enables theoretical studies such as genome coverage with a specific set of peptide.

The core of the system is written in Perl, comprising more than 2,000 lines of code, extensively relying on the Ensembl Perl API (Stabenau *et al.*, 2004), without which the codebase would have been significantly larger. This API provided all the core functionality required to establish the proteome-genome relationship, such as the coordinate conversions between translated gene products and the underlying genomic sequences. Therefore, none of the involved steps required any sequence alignment tools. The next few sections of this chapter briefly describe individual components.

4.2.1 Genome annotation data sources and integration

Section 1.2.5 discussed the Ensembl and Vega projects in detail, which together with Augustus gene predictions build the annotation data basis for this pipeline. Details and parameters are discussed in depth in the pilot study of chapter 5.

Conveniently, the Ensembl core API can be configured with the `Bio::EnsEMBL::Registry` module to handle either Ensembl or Vega data sources with the same platform. Moreover, the API module extension `Bio::EnsEMBL::Analysis::Runnable::Finished::Augustus` enables full API functionality for Augustus gene predictions (Stanke and Waack, 2003), which are otherwise only available as GFF files (`http://www.sanger.ac.uk/Software/formats/GFF/`). Lastly, standard text-based FASTA sequences can be integrated as supplemental data, such as a selected laboratory or contaminant protein sequences, but genome association is not possible.

During the database build process, the system performs the following simplified steps: (a) for each chromosome, get all protein coding genes; (b) for each gene, get all protein coding transcripts; (c) for each transcript, obtain the protein sequence; (d) determine all enzymatic cleavage sites within the protein sequence; (e) calculate the individual peptide start and end coordinates within the translation; (f) perform *in silico* digestion according to the available cleavage sites, allowing for the defined number of missed cleavage sites and sequence length constraints; (g) for each peptide, calculate the genomic coordinates, accounting for multiple loci if the peptide spans one or multiple splice sites; (h) store these loci in GenoMS-DB for each peptide, along with gene, transcript, exon and splice site information; (i) optionally account for coding SNPs (Schandorff *et al.*, 2007) and N-terminal methionine excision (Frottin *et al.*, 2006) by generating alternative peptide variants. In the current form, known post translational modifications or cleavages are not accounted for. It should be noted that the organism, enzyme settings, missed cleavages, peptide minimum and maximum length are user defined parameters, with the following default values: *mus musculus*, trypsin (cleavage after Arginine and Lysine), 2, 6, 50 respectively.

4.2.2 Database design

The database is an integral part of this pipeline and its relational model (schema) is illustrated in figure 4.2. The database is populated with one or multiple data sources, which were discussed in the previous section. Once built, the database is only used for querying with the only exception of some “PeptideSequence” table attributes, which are related to Mascot Percolator results integration (see section below).

GenoMS-DB was designed to allow the user to construct simple and fast SQL queries. This is done by selective denormalisation (Shin and Sanders, 2006) and choosing the peptide-genome mapping information as a central element, which is represented by the `PeptideMapping` table. This peptide-centric orientation, together with the involved denormalisation, led me to the development of a new schema instead of adapting the existing Ensembl database schema (Stabenau *et al.*, 2004).

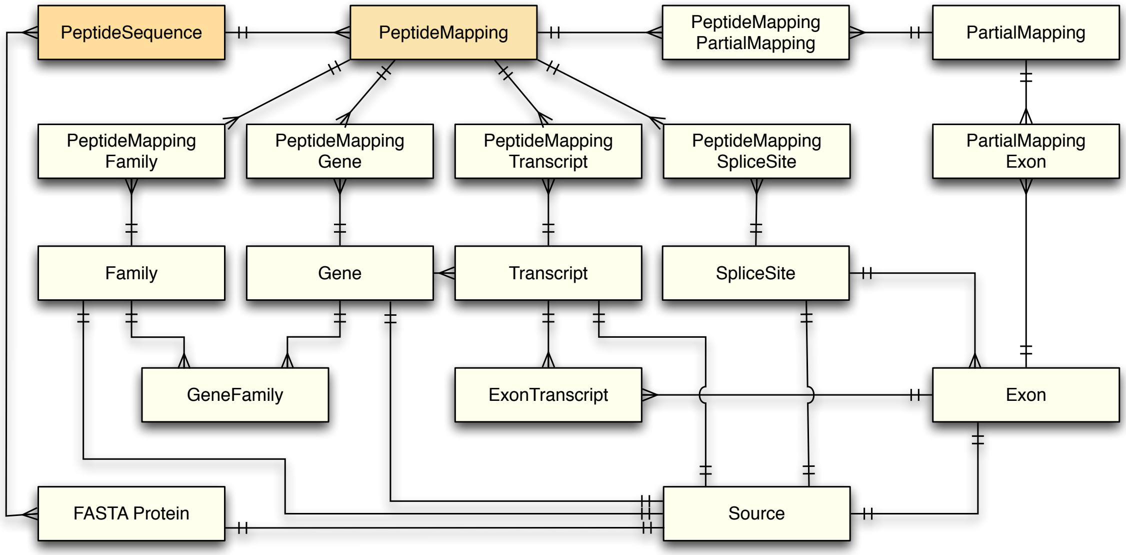


Figure 4.2: GenoMS-DB database schema. The `PeptideSeqeunce` together with the `PeptideMapping` table are the central elements in the schema (highlighted in orange). The former comprises peptide properties alongside peptide identification scores where available. The latter provides the genomic mapping coordinates for every peptide in the database. Note that this design is not fully normalised to provide optimal performance and ease of use. Yet, data integrity is guaranteed by the carefully controlled data integration process. Later use of the database only updates specific attributes of the `PeptideSequence` table that cannot lead to inconsistent data. The notation used in this schema is a simplified Crow's Foot notation.

In the following section the design of GenoMS-DB is discussed in more detail.

The `PeptideSequence` table stores all peptide related information, such as peptide sequence, the number of missed cleavages or sequence length. The attribute `seqKQLI` stores a sequence version with all Lysine (K) and Glutamine (Q) residues replaced with "1" and all Leucine (L) and Isoleucine (I) replaced with "2". Sequences that only differ in either K/Q or L/I residues cannot be differentiated in low resolution fragmentation spectra. Therefore, the uniqueness of every peptide within the proteome accounting for these ambiguities is tested in a post database build process. The results are stored as an integer number in the `ambigEnsVega` and `ambiguity` attributes, relating to the number of genomic loci these substituted peptides match within the tested space. The former attribute confines this space to Ensembl and Vega annotation only, whereas the latter also accounts for a much larger search space, such as predicted sequences generated by Augustus.

It should be noted that this approach is a simplification of the potential complexity involved in peptide-spectrum matching. For example, a modified residue such as deamidated Asparagine could result in a similar mass as an unmodified amino acid such as aspartic acid, which differs by only 3.6×10^{-5} Dalton. The selection of the wrong monoisotopic peak or the occurrence of a single-nucleotide polymorphism can lead to similar artefacts, hence sound MS data processing and database searching requires careful selection of parameters to avoid these caveats. Finally, the `PeptideSequence` table also stores identification information if the peptide was identified with Mascot and scored with Mascot Percolator. A separate table could have been directly linked with this table, but as pointed out earlier, the design is partly denormalised to minimise unnecessary table joins and yet provide a robust database schema.

To link peptide sequences with their genomic context, the `PeptideSequence` table has a one-to-many relationship with the `PeptideMapping` table. The latter represents the central proteogenomic element in the database, comprising attributes such as `start`, `end`, `chromosome` and `strand`. If a peptide spans one or multiple splice sites, the peptide maps partially to different genomic locations, which are stored in the `PartialMapping` table. If no splice site is spanned, the information of the `PeptideMapping` table and the `PartialMapping` table are redundant. If multiple alternative gene products give rise to multiple distinct peptide sequences, which have the same genomic start and end coordinate but differ in the partial genomic mapping due to variation in splicing, an alternative `PeptideMapping` entity is created in the table for each distinct case. Hence, a `PeptideMapping` entity maps to one and only one `PeptideSequence` entity.

`PartialMapping` entities are linked to the underlying `Exon`, which in turn links to related `SpliceSite` and `Transcript` entities. A `SpliceSite` entity maps to two distinct `Exon` entities, one representing the donor and one the acceptor exon. Since a transcript contains one or many exons and one exonic region can be part of multiple transcripts, the `Exon` table is linked to the `Transcript` table by a many-to-many relationship. Genes can give rise to multiple alternative transcripts, but a transcript always belongs to one gene, hence the relationship between the `gene` and `Transcript` table is one-to-many. To complete the genomic relationships, each `Gene` entity can belong to multiple `GeneFamily` entities and *vice versa.* The `Gene`, `Transcript` and `Exon` table store only the most relevant information, such as the chromosomal coordinates, identifiers, annotation status or short descriptions.

The `PeptideMapping` table has a many-to-many relationship with the `GeneFamily`, `Gene`, `Transcript` and `SpliceSite` tables. This redundant data representation minimises table joins to optimise performance and enables simplified query building.

The following example should clarify the rationale behind this design: to identify all splice sites that are associated with a set of peptides, the `PeptideSequence` table is joined with the table holding the peptide-genome mapping information, which in turn is joined with the table that maps to these requested splice sites. A fully normalised design would have required significantly more table joins, complicating the SQL query and slowing down the execution performance.

Lastly, the `PeptideSequence` table also has a many-to-many relationship with the `FastaProtein` table. This table is only populated if FASTA sequence information is integrated into the database, whereby no genomic mapping is available. This can be useful for protein contaminants or laboratory proteins as well as protein databases which are to be compared with Ensembl, Vega or Augustus. The `Source` table has a one-to-one relationship with `GeneFamily`, `Gene`, `Transcript`, `SpliceSite` and `FastaProtein` table. It comprises the version and type of the source database, which was used to build GenoMS-DB.

Overall the partly denormalised database represents a powerful tool to analyse proteogenomic data in an effective and efficient way. GenoMS-DB is individually built for genome annotation resources that are to be validated and refined with proteomics data, such as a specific Ensembl or Vega builds.

4.2.3 Mascot search database construction

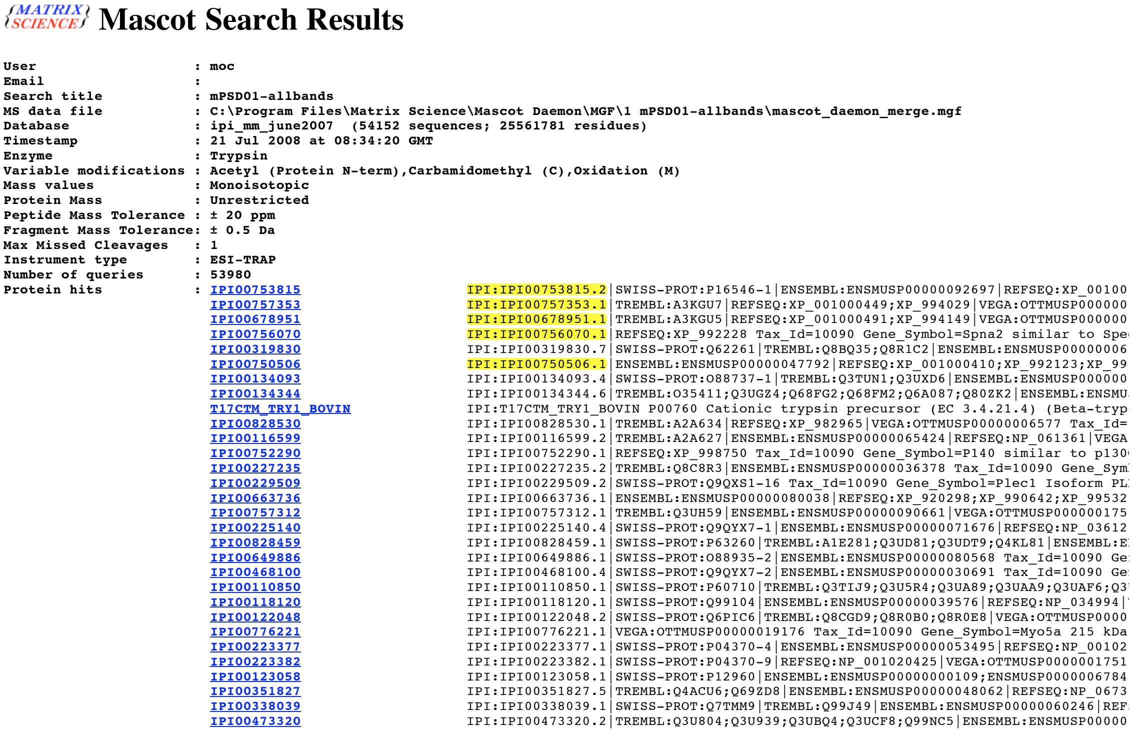
The search “database” of Mascot is simply a FASTA text file, where individual protein sequence entries are concatenated with the protein identifier as a delimiter, with alternative protein isoforms being handled as an individual protein entry.

Ensembl annotates alternative isoforms conservatively and as shown in chapter 5, of the almost 23 thousand protein coding Ensembl genes in mouse, only 8,877 had multiple protein coding isoforms annotated (Ensembl 54). However, a small subset of 1,542 protein coding Ensembl genes were predicted to code for 13,664 transcripts. Recently, Wang *et al.* (2008) has shown that more than 90% of human genes are expected to code for alternative isoforms. It is anticipated that manual annotation and improved automated annotation systems together with increased availability of expression data from different cell types or tissues will further increase the number of known alternative gene products.

Protein variants typically share most peptides (see section 5.3.2.4), which leads to significant peptide redundancy in these text based FASTA files. Moreover, when gene finding algorithms predict tens of alternative isoforms for a gene, a compact representation eliminating the inherent peptide redundancy is required.

Many database search tools, including Mascot, do not remove this redundancy since they sequentially cross correlate the *in silico* digested peptides (personal communication, John Cottrell, Matrix Science, London). Therefore search times scale linearly with database size.

To enable a compact representation, Martens *et al.* (2005b) proposed to digest the proteome into a peptide centric database that can be filtered and indexed to remove redundancy. For this, peptides are concatenated by an artificial residue that is used as a spacer element to separate individual peptides. The Mascot search enzyme settings need to be set accordingly to cleave at this artificial spacer element (see



Mascot Search Results

```
User                    : moc
Email                   :
Search title            : mPSD01-allbands
MS data file            : C:\Program Files\Matrix Science\Mascot Daemon\MGF\1 mPSD01-allbands\mascot_daemon_merge.mgf
Database                : ipi_mm_june2007  (54152 sequences; 25561781 residues)
Timestamp               : 21 Jul 2008 at 08:34:20 GMT
Enzyme                  : Trypsin
Variable modifications : Acetyl (Protein N-term),Carbamidomethyl (C),Oxidation (M)
Mass values             : Monoisotopic
Protein Mass            : Unrestricted
Peptide Mass Tolerance : ± 20 ppm
Fragment Mass Tolerance: ± 0.5 Da
Max Missed Cleavages    : 1
Instrument type         : ESI-TRAP
Number of queries       : 53980
Protein hits            : IPI00753815                   IPI:IPI00753815.2|SWISS-PROT:P16546-1|ENSEMBL:ENSMUSP00000092697|REFSEQ:XP_00100
                          IPI00757353                   IPI:IPI00757353.1|TREMBL:A3KGU7|REFSEQ:XP_001000449;XP_994029|VEGA:OTTMUSP000000
                          IPI00678951                   IPI:IPI00678951.1|TREMBL:A3KGU5|REFSEQ:XP_001000491;XP_994149|VEGA:OTTMUSP000000
                          IPI00756070                   IPI:IPI00756070.1|REFSEQ:XP_992228 Tax_Id=10090 Gene_Symbol=Spna2 similar to Spe
                          IPI00319830                   IPI:IPI00319830.7|SWISS-PROT:Q62261|TREMBL:Q8BQ35;Q8R1C2|ENSEMBL:ENSMUSP00000006
                          IPI00750506                   IPI:IPI00750506.1|ENSEMBL:ENSMUSP00000047792|REFSEQ:XP_001000410;XP_992123;XP_99
                          IPI00134093                   IPI:IPI00134093.4|SWISS-PROT:O88737-1|TREMBL:Q3TUN1;Q3UXD6|ENSEMBL:ENSMUSP000000
                          IPI00134344                   IPI:IPI00134344.6|TREMBL:O35411;Q3UGZ4;Q68FG2;Q68FM2;Q6A087;Q80ZK2|ENSEMBL:ENSMU
                          T17CTM_TRY1_BOVIN             IPI:T17CTM_TRY1_BOVIN P00760 Cationic trypsin precursor (EC 3.4.21.4) (Beta-tryp
                          IPI00828530                   IPI:IPI00828530.1|TREMBL:A2A634|REFSEQ:XP_982965|VEGA:OTTMUSP00000006577 Tax_Id=
                          IPI00116599                   IPI:IPI00116599.2|TREMBL:A2A627|ENSEMBL:ENSMUSP00000065424|REFSEQ:NP_061361|VEGA
                          IPI00752290                   IPI:IPI00752290.1|REFSEQ:XP_998750 Tax_Id=10090 Gene_Symbol=P140 similar to p130
                          IPI00227235                   IPI:IPI00227235.2|TREMBL:Q8C8R3|ENSEMBL:ENSMUSP00000036378 Tax_Id=10090 Gene_Sym
                          IPI00229509                   IPI:IPI00229509.2|SWISS-PROT:Q9QXS1-16 Tax_Id=10090 Gene_Symbol=Plec1 Isoform PL
                          IPI00663736                   IPI:IPI00663736.1|ENSEMBL:ENSMUSP00000080038|REFSEQ:XP_920298;XP_990642;XP_99532
                          IPI00757312                   IPI:IPI00757312.1|TREMBL:Q3UH59|ENSEMBL:ENSMUSP00000090661|VEGA:OTTMUSP000000175
                          IPI00225140                   IPI:IPI00225140.4|SWISS-PROT:Q9QYX7-1|ENSEMBL:ENSMUSP00000071676|REFSEQ:NP_03612
                          IPI00828459                   IPI:IPI00828459.1|SWISS-PROT:P63260|TREMBL:A1E281;Q3UD81;Q3UDT9;Q4KL81|ENSEMBL:E
                          IPI00649886                   IPI:IPI00649886.1|SWISS-PROT:O88935-2|ENSEMBL:ENSMUSP00000080568 Tax_Id=10090 Ge
                          IPI00468100                   IPI:IPI00468100.4|SWISS-PROT:Q9QYX7-2|ENSEMBL:ENSMUSP00000030691 Tax_Id=10090 Ge
                          IPI00110850                   IPI:IPI00110850.1|SWISS-PROT:P60710|TREMBL:Q3TIJ9;Q3U5R4;Q3UA89;Q3UAA9;Q3UAF6;Q3
                          IPI00118120                   IPI:IPI00118120.1|SWISS-PROT:Q99104|ENSEMBL:ENSMUSP00000039576|REFSEQ:NP_034994|
                          IPI00122048                   IPI:IPI00122048.2|SWISS-PROT:Q6PIC6|TREMBL:Q8CGD9;Q8R0B0;Q8R0E8|VEGA:OTTMUSP0000
                          IPI00776221                   IPI:IPI00776221.1|VEGA:OTTMUSP00000019176 Tax_Id=10090 Gene_Symbol=Myo5a 215 kDa
                          IPI00223377                   IPI:IPI00223377.1|SWISS-PROT:P04370-4|ENSEMBL:ENSMUSP00000053495|REFSEQ:NP_00102
                          IPI00223382                   IPI:IPI00223382.1|SWISS-PROT:P04370-9|REFSEQ:NP_001020425|VEGA:OTTMUSP000001751
                          IPI00123058                   IPI:IPI00123058.1|SWISS-PROT:P12960|ENSEMBL:ENSMUSP00000000109;ENSMUSP0000006784
                          IPI00351827                   IPI:IPI00351827.5|TREMBL:Q4ACU6;Q69ZD8|ENSEMBL:ENSMUSP00000048062|REFSEQ:NP_0673
                          IPI00338039                   IPI:IPI00338039.1|SWISS-PROT:Q7TMM9|TREMBL:Q99J49|ENSEMBL:ENSMUSP00000060246|REF
                          IPI00473320                   IPI:IPI00473320.2|TREMBL:Q3U804;Q3U939;Q3UBQ4;Q3UCF8;Q99NC5|ENSEMBL:ENSMUSP00000
```

Figure 4.3: Screenshot of a typical Mascot search result page. In this example, five of the top six protein entries belong to the same gene *spectrin alpha 2*, indicated in yellow.

section 5.2.2). In the following section I present two alternative peptide-level search databases, both of which can be directly derived from the GenoMS-DB database.

Gene centric peptide level database

The first approach concatenates all peptides from GenoMS-DB on a per gene basis in a non-redundant manner, thereby compressing the database size and reducing the database search times, since multiple occurrences of a peptide in alternative gene products are collapsed into one gene entry. This "gene-centric" search database can be useful to simplify the analysis of a complex sample, when there is no need to distinguish the individual isoforms.

To contrast this approach with the classic protein database search, figure 4.3 shows a typical Mascot search result page with a list of protein hits that were identified from a standard protein database (IPI). Browsing this list manually to analyse the dataset can be cumbersome, especially when multiple protein isoforms belong to the same underlying gene. In the illustrated example, five of the top six proteins belong to the same gene *spectrin alpha 2*. On the other hand, the search result against the gene-centric database introduced above only returned the gene of interest instead of the five individual entries, which can be useful for users who directly browse the search results with Mascot.

An alternative strategy would be to use GenoMS-DB to selectively export only peptides that map uniquely to one locus. This would enable a targeted proteomics experiment with the focus of gene identification. Even more complex scenarios could be designed, such as the selection of peptides that enable discrimination of protein isoforms.

Strict peptide level database

For my work however, no protein nor gene information was required from the Mascot search results, since the genomic context is established by integrating these peptide identifications into GenoMS-DB, which is discussed in section 4.2.4. Hence, this second approach simply concatenates all peptides stored in the database in a non-redundant manner, where 1000 peptides at a time are binned and concatenated into one FASTA entry, separated only by a spacer element (e.g. artificial residue “J”). The FASTA header is set to the number of the entry as a text string. By default, there is no selection for specific peptide parameters during the FASTA file build process, but optionally sequence length or number of missed cleavages could be restricted. By default only the database build parameters (see 4.2.1) limit the peptides available in the resulting FASTA file. Given a peptide is only stored once in the whole FASTA file, this method further reduces peptide-level redundancy that is caused by peptides that are present in multiple genes. This database type is used in the pilot study conducted in chapter 5.

Listing 4.1: A simple SQL query example that selects the list of Ensembl protein coding genes from GenoMS-DB with the corresponding number of identified peptides per gene that exceed a PEP score of 20 and match a unique genomic locus within the Ensembl annotation.

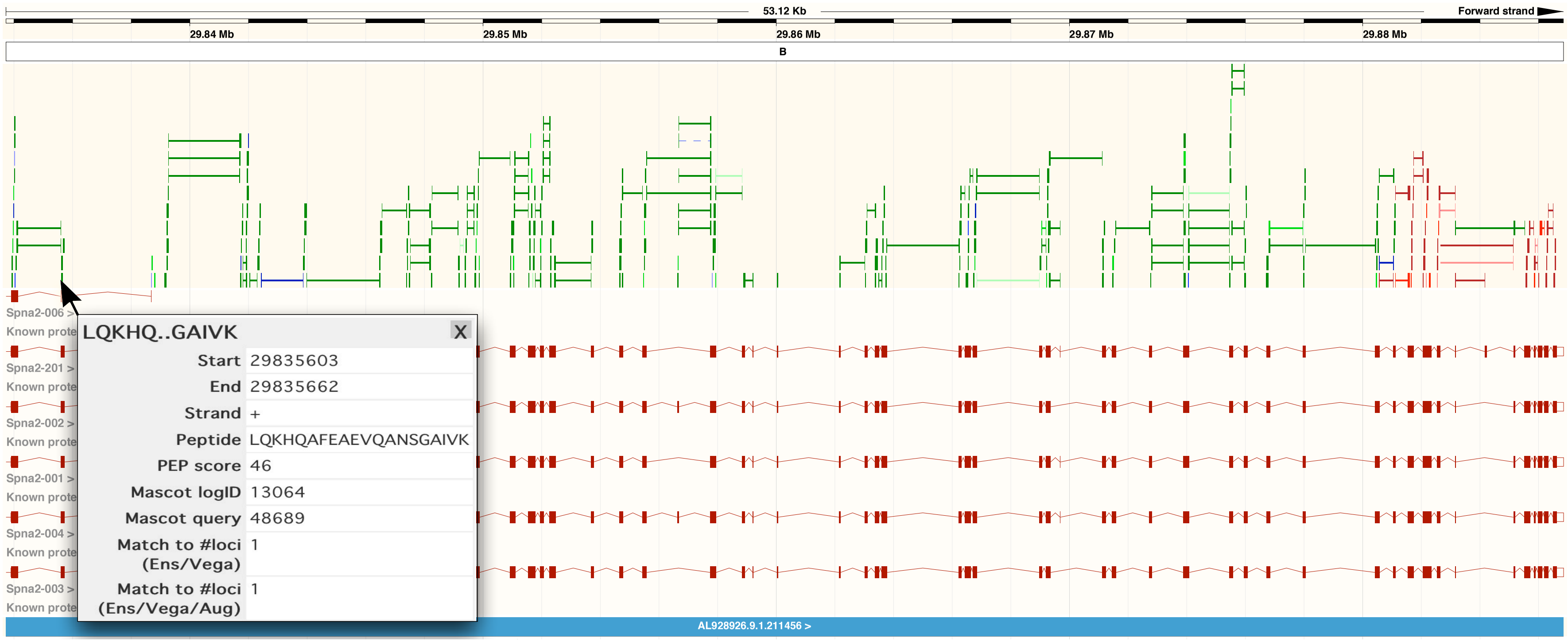
```
select geneID, count(distinct peptideSequenceID)
  from PeptideSequence
  inner join PeptideMapping using(peptideSequenceID)
  inner join PeptideMapping_Gene using(peptideMappingID)
  inner join Gene using(geneID)
  inner join Source using(sourceID)
  where PeptideSequence.score > 20
  and PeptideSequence.ambigEnsVega = 1
  and Source.db = "ensembl"
  and Gene.biotype = "protein_coding"
  group by geneID
```

4.2.4 Results integration

After searching tandem MS data with Mascot against the peptide level database described above, search results are post-processed by Mascot Percolator and stored in a specified folder. A Perl script, which can be executed on a regular basis with a job scheduler such as “Cron” on Unix-like operating systems, scans this results folder and processes the Mascot Percolator result files in the following way: (a) firstly peptides that match to user defined contaminants or laboratory proteins are filtered out from the search results; (b) all remaining peptide identifications that exceed user defined scoring criteria are then persisted in the PeptideSequence table of GenoMS-DB (figure 4.2) by updating the relevant attributes. Currently the system only keeps the best scoring peptide identification in GenoMS-DB.

4.2.5 SQL analysis and DAS server implementation

After results integration, GenoMS-DB can be leveraged for large scale proteogenomics analysis, employing standard SQL queries. For each analysis, a custom SQL query statement can be designed and executed, providing efficient means to an otherwise complex manual analysis process. In listing 4.1 a simple SQL query example is



LQKHQ..GAIVK X

Start	29835603
End	29835662
Strand	+
Peptide	LQKHQAFEAEVQANSGAIVK
PEP score	46
Mascot logID	13064
Mascot query	48689
Match to #loci (Ens/Vega)	1
Match to #loci (Ens/Vega/Aug)	1

Figure 4.4: The Ensembl browser representing the *spectrin alpha 2* locus. The page shows the Ensembl annotation tracks (red tracks on lower half) as well as a peptide identifications (upper half), which are integrated into Ensembl via the DAS server that queries GenoMS-DB and renders the peptide mapping accordingly. Peptide features can be selected and meta-information can be displayed (see illustrated information window), such as the exact genomic mapping coordinates, scoring information, number of genomic loci the peptide matches perfectly or the Mascot logging identifier and the spectrum query number, enabling to track back into the original Mascot results. The type of colour of these peptide features depends on the uniqueness of the peptide within the genome: unique within Ensembl/Vega/Augustus (green), unique within Ensembl/Vega (blue), ambiguous and multiple matches (red). The brightness of the peptide features is correlated to the Mascot Percolator Posterior Error Probability (PEP) score: $5 \times 10^{-2} \leq PEP < 1 \times 10^{-2}$ (light), $1 \times 10^{-2} \leq PEP < 1 \times 10^{-3}$ (medium), $PEP \geq 1 \times 10^{-3}$ (dark). Although these data are also available through the information window, the Vega annotation pipeline cannot show meta-data and hence colour coding was a way to provide uniqueness and scoring information to annotation curators. This gene is also used as an example in figure 4.3 and is further discussed in section 4.2.3

Listing 4.2: A more complex SQL query example that selects the list of Augustus genes that were validated by identified peptides that exceed a PEP score of 30 and map to an unique genomic locus and not match either the Ensembl or the Vega annotation in order to only select truly novel coding regions. The resulting list provides all gene details, the matching peptide sequences and the genomic peptide mapping information.

```
select G.*, S.seq, P.*
  from PeptideSequence as S
  inner join PeptideMapping as P using(sequenceID)
  inner join PeptideMapping_Gene using(peptideMappingID)
  inner join Gene as G using(geneID)
  inner join Source using(sourceID)
  where PEP > 30
  and db = "augustus"
  and ambiguity = 1
  and not exists (
    select * from PeptideSequence
      inner join PeptideMapping using(sequenceID)
      inner join PeptideMapping_Gene using(peptideMappingID)
      inner join Gene using(geneID)
      inner join Source using(sourceID)
      where PEP > 30
      and db in (’ensembl’, ’vega’)
      and S.seq = seq
  )
```

provided that demonstrates how the list of Ensembl genes with the corresponding peptide matches can be selected. Listing 4.2 is a more complex query that demonstrates the simplicity with which relatively complex questions can be answered.

SQL queries were also used for the development of a stand-alone proteogenomic distributed annotation server (DAS) (Dowell *et al.*, 2001) that is accessing the integrated data of GenoMS-DB. When the DAS server receives a request for a specific genomic region, all peptides together with their associated genomic mapping data are selected within the defined region and returned as DAS features. The required SQL statements were implemented into ProServer, an extendable Perl based DAS feature server (Finn *et al.*, 2007). Peptide features provided through the DAS server can then be attached to genome browsers, such as Ensembl (figure 4.4). Features can be supplemented with meta-information that are required for automatic or manual genome annotation: peptide mapping details, uniqueness of peptide within the genome, scoring details or original Mascot spectrum ID.

4.3 Conclusion

In this work I have developed a proteogenomic pipeline that enables efficient and effective large scale genome wide data analysis. It leverages the power of a relational database, which is at the core of the system. My database design allows high performance analysis with easy to construct SQL statements. GenoMS-DB, integrates all relevant information for subsequent proteogenomic analysis. This database accepts annotation data from Ensembl, Vega or Augustus, as well as supplemental data from FASTA databases. Therefore, proteins or protein isoforms not present in these databases cannot be identified with this pipeline, which is generally true for any database search algorithm (see section 1.1.1). These data are digested *in silico* and stored in GenoMS-DB together with their genomic context. The genomic mapping coordinates are calculated, enabling the ad-hoc mapping of millions of peptides with GenoMS-DB, since alignment tools to map peptide sequences against the genome are not required. Integrated peptides can be exported to non-redundant peptide collections, which can in turn be used by Mascot as efficient search databases. Results from Mascot Percolator can also be stored in GenoMS-DB. This complete integration enables proteogenomic analysis with standard SQL. Even complex questions can be formulated in a few lines of SQL code, whereby analysis is fully automated, avoiding any manual intervention. Large scale studies can be carried out since genome mapping is readily available through GenoMS-DB. This also allows the analysis of theoretical peptide collections, such as the whole proteome. The next chapter tests and uses this pipeline in a pilot study.