BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 8 2005, pages 1443–1450
doi:10.1093/bioinformatics/bti220

Sequence analysis

Mapping SNPs to protein sequence and structure data

A. Cavallo and A. C. R. Martin*

Department of Biochemistry and Molecular Biology, University College London, Gower Street,
London WC1E 6BT, UK

Received on October 5, 2004; revised on November 18, 2004; accepted on December 12, 2004
Advance Access publication December 21, 2004

ABSTRACT

Motivation: Data on both single nucleotide polymorphisms and disease-related mutations are being collected at ever-increasing rates. To understand the structural effects of missense mutations, we consider both classes under the term single amino acid polymorphisms (SAAPs) and we wish to map these to protein structure where their effects can be analyzed. Our initial aim therefore is to create a completely automatically maintained database of SAAPs mapped to individual residues in the Protein Data Bank (PDB) updated as new mutations or structures become available.

Results: We present an integrated pipeline for the automated mapping of SAAP data from HGVbase to individual PDB residues. Achieving this in a completely automated and reliable manner is a complex task. Data extracted from HGVbase are mapped to EMBL entries to confirm whether the mutation occurs in an exon and, if so, where in the sequence it occurs. From there we map to Swiss-Prot entries and thence to the PDB.

Availability: The resulting database may be accessed over the web at http://www.bioinf.org.uk/saap/ or http://acrmwww.biochem.ucl.ac.uk/saap/

Contact: a.martin@biochem.ucl.ac.uk

1 INTRODUCTION

In the post-genomic era, interest is beginning to focus on the differences between the genomes of individuals and on the effects of mutations. Mutation data fall largely into two classes: single nucleotide polymorphisms (SNPs) and disease-related mutations. Further sub-divisions are possible including the split between non-coding, silent, mis-sense and nonsense mutations. Mutations occur naturally in DNA, but are generally corrected through DNA repair systems. Mutations are rarely maintained and inherited by daughter cells or future generations (Li *et al*., 1996). Clearly, mutations may affect organisms in very different ways; they may exhibit the complete range of phenotypes from drastic detrimental effects, through mild and completely phenotypically silent effects to minor improvements or the introduction of new function (Ohno, 1984). The larger the change brought about by the mutation, the more likely it is to have a drastically affected phenotype. For example, frameshifts, deletions, insertions, repetitions or nonsense codons leading to early termination of a protein are almost guaranteed to have a drastic effect, whereas single amino acid mutations may have a much more limited effect on phenotype. Because around 90% of sequence variants in humans are single DNA base changes (Collins *et al*., 1998), there is an increasing interest in this family of mutations (Casillas and Barbadilla, 2004; Capriotti *et al*., 2004).

Formally, SNPs can be defined as alleles which exist in normal individuals in a population with the least frequent allele having an abundance of at least 1% (Brookes, 1999). In principle, SNPs could be bi-, tri- or tetra-allelic variations, but tri- and tetra-allelic SNPs are very rare in humans. In practice, the term SNP is often applied in a more generic context and may encompass disease-causing mutations which are recessive, or low-penetration dominant alleles, the latter generally being present at much lower frequencies. For the purposes of this analysis we make no distinction on the basis of allelic frequency and many submissions to the major collection of SNP data, dbSNP (Sherry *et al*., 2001), are at a lower frequency or lack this information. We thus use the term SNP to refer to any observed single nucleotide DNA mutation and single amino acid polymorphism (SAAP) to refer to any resulting mis-sense protein mutation.

The likelihood of a SNP having an effect on phenotype depends on where it occurs and the nature of the change it induces. Table 1 summarizes the types of change possible and the likelihood of these changes having an effect on phenotype. Mutations in a non-coding region are less likely to have an effect on phenotype as they do not directly affect the structure of a protein. However, they may have effects on expression levels of adjacent proteins or influence alternative splicing. A silent mutation in a coding region may have an effect on phenotype for similar reasons. Recent work by Sanjuán *et al*. (2004) looked at single-nucleotide mutations in an RNA virus and identified 24 lethal mutations. While 19 were mis-sense, 3 were nonsense and 1 disrupted a start codon, one lethal mutation was a ‘silent’ mutation having no effect on the encoded protein sequence. Eight other synonymous mutations had no significant effect on viral fitness.

Thus the relationship between SNPs and phenotype is not straightforward, and we restrict our analysis to SNPs resulting in mis-sense mutations, i.e. SAAPs. In addition to SNP data, OMIM (McKusick, 2000, http://www.ncbi.nlm. nih.gov/omim/) provides a collection of information on mutations in disease and there are large numbers of so-called locus specific mutation databases (LSMDBs) (Claustres *et al*., 2002) which collect information on mutations in individual diseases (see http://www.genomic.unimelb.edu.au/mdi/dblist/glsdb.html). The advantage of small specialized LSMDBs is that the data and annotations are of high quality, being collected by experts in a particular gene or disease. The chief problem in using them is that their formats are arbitrary and to allow global analyses, there is a desperate need to define a common extensible format providing a subset of standardized annotations in a fashion analogous

*To whom correspondence should be addressed.

© The Author 2004. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oupjournals.org

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

Table 1. Point mutations and their likely effects on phenotype

Mutation type	Phenotype	
	Silent	Affected
Coding		
Silent	√	∼
Mis-sense	=	=
Nonsense	×	√
Non coding		
Silent	√	∼

√, High probability; =, Medium probability; ∼, Low probability; ×, Extremely low probability.

to the MIAME standard for microarray experiments (Brazma *et al*., 2001).

Our aim is to apply methods we have developed to predict the local structural consequences of protein mutations to all proteins for which structural data are available and mutations are known. These methods have been applied previously to the analysis of the tumour-suppressor protein, p53 (Martin *et al*., 2002) and G6PD, mutations which result in 'favism' (Kwok *et al*., 2002).

Previous work from other groups have attempted to address the question of the effects of mutations on protein structure. SNPs3D (http://www.snps3d.org) and PolyPhen (http://www.bork.embl-heidelberg. de/PolyPhen/), both address the problem of associating polymorphisms with protein structure alterations. PolyPhen (Ramensky *et al*., 2002; Sunyaev *et al*., 2000, 2001) provides a server on which one can assess the predicted effect of a mutation and see links from Swiss-Prot to the location of a mutation in a Protein Data Bank (PDB) file. The versions of Swiss-Prot and the PDB have not been updated since 2002. In addition, they provide a page (http://tux.embl-heidelberg.de/ramensky/data/) where predicted effects have been precomputed for mutations stored in the 2001 (V12.0) release of HGVbase (Fredman *et al*., 2002). SNPs3D, developed in John Moult's group, has been updated more recently, but at the time of writing, the most recent update was eight months old, the main source database, dbSNP, having been updated five times in that period, suggesting again that the system is not completely automatically maintained. No papers have been published about SNPs3D, so it is difficult to compare the methods used although it uses dbSNP, HGMD (Stenson *et al*., 2003) and OMIM (McKusick, 2000) as mutation data sources. The group of de la Cruz has also investigated the prediction of effects of single amino acid mutations on protein function and disease, either by looking at sequence and structure (Ferrer-Costa *et al*., 2002), or sequence alone (Ferrer-Costa *et al*., 2004). In addition, Conde *et al*. (2004) have very recently provided a web-based tool for finding SNPs with predicted effects at the transcriptional level.

The HGVbase project (Fredman *et al*., 2002) has tried to add an additional layer of validated annotation on top of dbSNP and aims to integrate data from LSMDBs. While dbSNP acts as a global, general repository for any SNP data, HGVbase has the goal of giving the scientific community a high quality, validated dataset (one can think of its relationship with dbSNP as being similar to the relationship between Swiss-Prot and trEMBL). In addition, the planned incorporation of LSMDB data eliminates the problems of extracting data from the diversity of formats used for these specialist databases. These two major advantages led us to decide initially to restrict our analysis to the high quality data in HGVbase.

At this stage, we have concentrated on the major prerequisite for this analysis, a mapping of mutations to individual residues in protein structures stored in the PDB (Berman *et al*., 2000). We set out to create a system which is completely automated and will update the mapping database as new mutation data, or new structures, become available. In the second phase of the project, our previously-developed analysis methods will be integrated into the system. In this paper we present a completely automated system for the collection of data from HGVbase in order to map mutations to individual mutated residues in PDB entries.

2 SYSTEMS AND METHODS

Conceptually, there are two distinct components: (1) a mapping process—a protocol for mapping a mutation to a residue in a PDB file and (2) an automated system for keeping the data updated and rerunning the mapping process as required.

The mapping process may be summarized as follows:

- extracting the required information from the primary database entry referenced in the mutation database,
- analyzing the gene information to determine whether this mutation will result in a mis-sense mutation and identifying the location of the mutation in a coding region, having accounted for the presence of introns,
- retrieving the wild-type and mutated version(s) of the protein sequence from the coding segment (CDS) of the gene sequence,
- identfying any known structure(s) for the protein sequence,
- extracting the location (chain name and residue number) of the mutation in a PDB file.

The automation is responsible for:

- keeping the source data (mutation databases, DNA and protein sequence databases and protein structure databases) updated,
- running the mapping process when data have changed,
- storing the results in a relational database,
- making the results available for access over the web.

We now describe the details of the mapping process. First, we define the minimal required input data as:

(1) the sequences upstream and downstream of the mutation site,

(2) the alleles at the mutation site (which must be the nucleotides a, t, c or g—unknown or ambiguous bases are not allowed),

(3) a link to a primary database.

HGVbase contains information on whether the SNP is coding and whether it generates a mis-sense mutation, but a preliminary analysis showed that these data were provided inconsistently and that there were errors in some entries. It was therefore decided to determine this information locally. By defining this very simple base requirement and determining the mapping to a protein sequence and the nature of any resulting change ourselves gives us the option to include less highly annotated SNP data sources, such as dbSNP, at a later date.

Strictly, we do not require the link to the primary database since we could use a BLAST search to identify the appropriate entry, but for practicality we require this to be present. HGVbase provides all these data together with various ancillary information (e.g. mutation accession code, gene symbol, product name, bibliography references, experimental conditions, allele frequency, etc.). Where available, we also store the gene symbol and product name to enable searches to be performed on these parameters.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

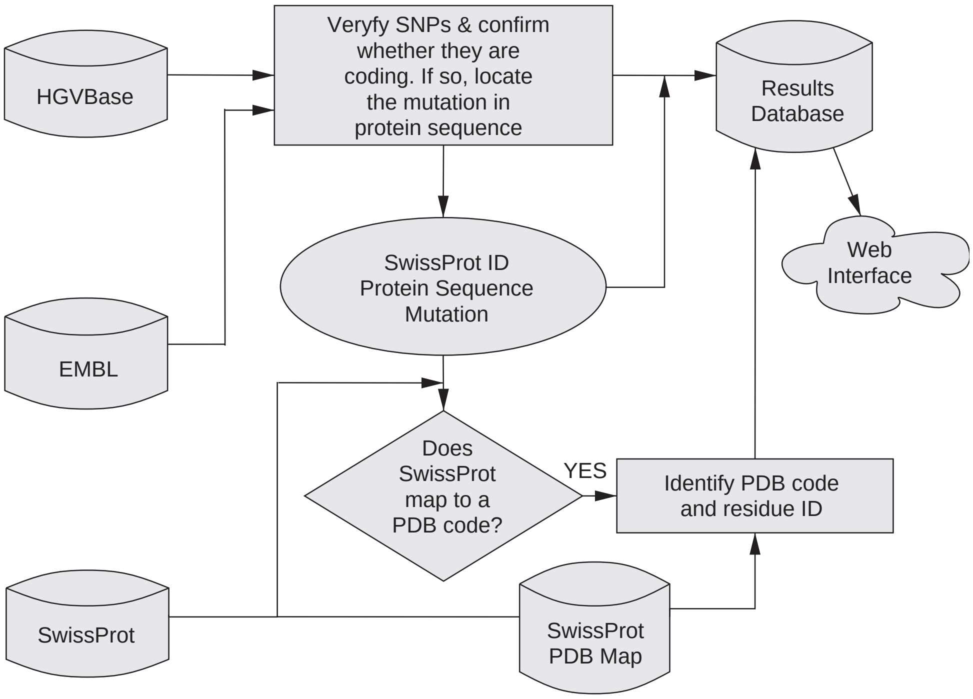
HGVBase
Veryfy SNPs & confirm whether they are coding. If so, locate the mutation in protein sequence
Results Database
Web Interface
SwissProt ID
Protein Sequence
Mutation
EMBL
Does SwissProt map to a PDB code?
YES
Identify PDB code and residue ID
SwissProt
SwissProt PDB Map

Fig. 1. The complete work flow. Coding SNPs are identified by comparing HGVBase entries with EMBL. Non-synonymous SNPs resulting in an amino acid mutation are then mapped to any available structural data in the PDB via Swiss-Prot.

Similarly, we define the final output of the system as:

(1) does the mutation occur in a coding region? If not, no further information is required;

(2) the protein sequence of the 'native' protein and its allelic variants where these result in an SAAP;

(3) where available, the residue number and chain label of the PDB structure file, or files, for this protein.

3 IMPLEMENTATION

The overall workflow is shown in Figure 1. This process is achieved through a four-layer architecture illustrated in Figure 2 and described in more detail in the following subsections. Once an 'interface agreement' has been made by the four architectural layers, any modification in one layer will be insulated from the other layers. The bottom 'data acquisition' layer encapsulates all the problems related to downloading and indexing datasets to enable data to be extracted. The lower middle 'data processing' layer is responsible for the major work of performing the mapping process and storing it in the database of the 'data storage' layer. Finally, the top 'data presentation' layer provides the web interface.

3.1 The data acquisition layer

HGVbase is available both in XML and a plain keyword-based format (similar to Swiss-Prot or EMBL). Formats are described in detail at http://hgvbase.cgb.ki.se/cgi-bin/main.pl?page=data_struct.htm. The plain-text format was used in preference to the XML since the latter was found to be around ten times slower to parse using the 'pulldom' library http://docs.python.org/lib/module-xml.dom.pulldom.html (a Streaming API for XML (StAX) implementation for Python).

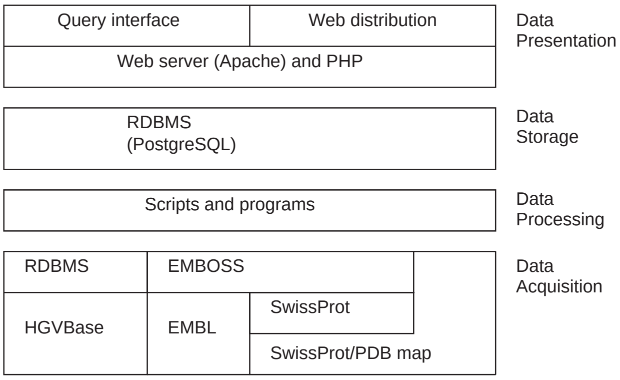

Fig. 2. The multi-layer architecture of the mapping system. The discrete layers encapsulate data collection, processing, storage and presentation respectively.

HGVbase is mirrored from ftp://ftp.ebi.ac.uk/pub/databases/variantdbs/hgbase/ using the Mirror package (http://sunsite.doc.ic.ac.uk/packages/mirror/, patched by ACRM to allow remotely compressed archives to be stored uncompressed locally). Data are then read into a relational database implemented in PostgreSQL (http://www.postgresql.org/). Data stored in the database consist of:

(1) *unique id* the mutation accession code from HGVbase stored in lowercase,

(2) *upstream sequence* bases preceding the mutation site,

(3) *downstream sequence* bases following the mutation site,

(4) *alleles* all alternative bases observed at the mutation site,

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

(5) *cross link* one or more references to a primary sequence database (EMBL or GenBank) and

(6) *text* the concatenated *genename*, *genesymbol* and *genetype* fields.

The format of the cross-link field consists of the database name followed by a colon and the unique id for that database. This format is used by the EMBOSS library (Mullan and Bleasby, 2002; Olson, 2002), which we use for data access and is a simplified version of the LSID specifications for a globally unique identifier to data resources. (See http://www.i3c.org for a full description of the LSID format and see http://emboss.sourceforge.net/docs/#Usa for a discussion of the format as used in EMBOSS.)[1]

The text field contains data from the *genesymbol* (http://ash.gene.ucl.ac.uk/nomenclature/) and *genename* fields in the SNP databases. Unfortunately, there are some errors in the current release of HGVbase such that the data contained in *genename*, *genesymbol* and *genetype* fields are confused. It was therefore decided to concatenate these three fields rather than try to untangle the data.

During data loading, a preliminary validation is performed in which entries not conforming to the minimal requirements stated above are rejected. Two additional fields are stored in the database indicating the validation status and any associated message to explain validation errors. The message field may also contain warnings in the case of ‘suspect’ entries in the SNP database.

This preliminary validation currently flags 26 313 out of 2 859 130 entries in HGVbase as ‘failed’. Typically, these are entries where an allele is ambiguous (i.e. a nucleotide symbol other than a, t, c or g; ambiguous bases are permitted in the upstream and downstream sequences), or is some other form of change such as an insertion, deletion or tandem repeat. A further 2003 entries are broken for other reasons such as invalid data in the database. For example, SNP000008183 reports an allele of ‘UnKnown’ where a single letter (in this case ‘n’) should be present to indicate a single base type.

The other databases used, EMBL ftp://ftp.ebi.ac.uk/pub/databases/embl/release, Swiss-Prot ftp://ftp.expasy.org/databases/uniprot/knowledgebase/ and the PDB ftp://ftp.ebi.ac.uk/pub/databases/rcsb/pdb/data/structures/all/pdb/ are also mirrored locally. Access to the sequence databases is handled via the AJAX library of EMBOSS (Olson, 2002; Mullan and Bleasby, 2002, http://emboss.sourceforge.net/) which is able to handle multiple data sources transparently.

3.2 The data processing layer

The data processing layer is responsible for:

- further validation of SNP data,
- mapping between the SNP with its upstream/downstream sequences to its location within a primary database entry (using the cross-link stored in the SNP database),
- identifying whether the mutation occurs in a coding region.

If the mutation is located in a CDS, then the data processing layer is also responsible for:

- determining whether the mutation is silent, mis-sense or nonsense,
- determining the location of the mutation in the protein sequence,
- reporting the wild type protein sequence,
- for a mis-sense mutation reporting the mutated sequence(s),
- identifying whether a structure is known for the protein,
- reporting the associated protein structure and the chain and residue identifier of the mutated residue.

The key parts of this process are the additional validation, mapping from mutation data to the primary database and mapping to a PDB entry. These stages are now described.

3.2.1 Additional validation Additional validation checks that the primary database cross-references link to EMBL. While HGVbase is supposed to link primarily to EMBL, almost half of the entries are described as GenBank references. However, in most of these cases, the accession code for EMBL and GenBank is identical, so a check is made to see whether the cross-link to GenBank is, in fact, a valid cross-link to EMBL. In addition, the following mapping stage validates the upstream and downstream sequences. If these (or their reverse complements) are not found in the cross-linked entry, then the SNP entry is marked as invalid.

3.2.2 Mapping between SNP data and the primary sequence database We employ EMBL as the primary sequence database since, in the main, links from HGVbase are to EMBL rather than to GenBank. The cross-linked sequence is extracted from EMBL using EMBOSS, as described above, and a lookup is performed inside that sequence to locate where the SNP occurs.

The mapping is performed using a program `findsnp3` implemented in C++ and using a locally implemented C++ wrapper to the EMBOSS library. This wrapper allows access to data entries in an object oriented fashion, simplifying the handling of large and complex datasets and hiding the inner complexities of the EMBOSS library from the programmer. The program takes an SNP sequence (expressed as the upstream sequence, an X to represent the mutation site and a downstream sequence), a list of alternative alleles at the mutation site and a pointer to the primary sequence database in the EMBOSS USA (abbreviated LSID) format.

HGVbase (like dbSNP) provides one or more links to a primary database and the presence of this cross-link is a requirement of our ‘minimum’ input. HGVbase also provides more detailed information including whether the SNP is coding and details of any amino acid change. However, as explained above, we decided not to rely on these additional annotations.

The mapping process must account for the fact that within one EMBL entry there may be zero, one or more sets of CDS records. Each set of CDS records indicates ranges of bases within the DNA sequence in the form of start and end points.

In addition, the mapping procedure is designed to take account of the fact that CDS records can contain external references to DNA segments which appear in other EMBL entries. These must be spliced in to assemble a complete coding sequence. This organization of

[1] A full LSID identifier has the form *URN:LSID:Authority:Namespace:Object:*[*Revision-ID*] where *URN:LSID* is a mandatory tag, *Authority* a name (such as, chemacx.cambridgesoft.com) for the authority in charge of the data indexed by the *Namespace:Object:*[*Revision-ID*], with [Revision-ID] being optional. The EMBOSS naming convention corresponds to the *Namespace* and *Object* fields.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

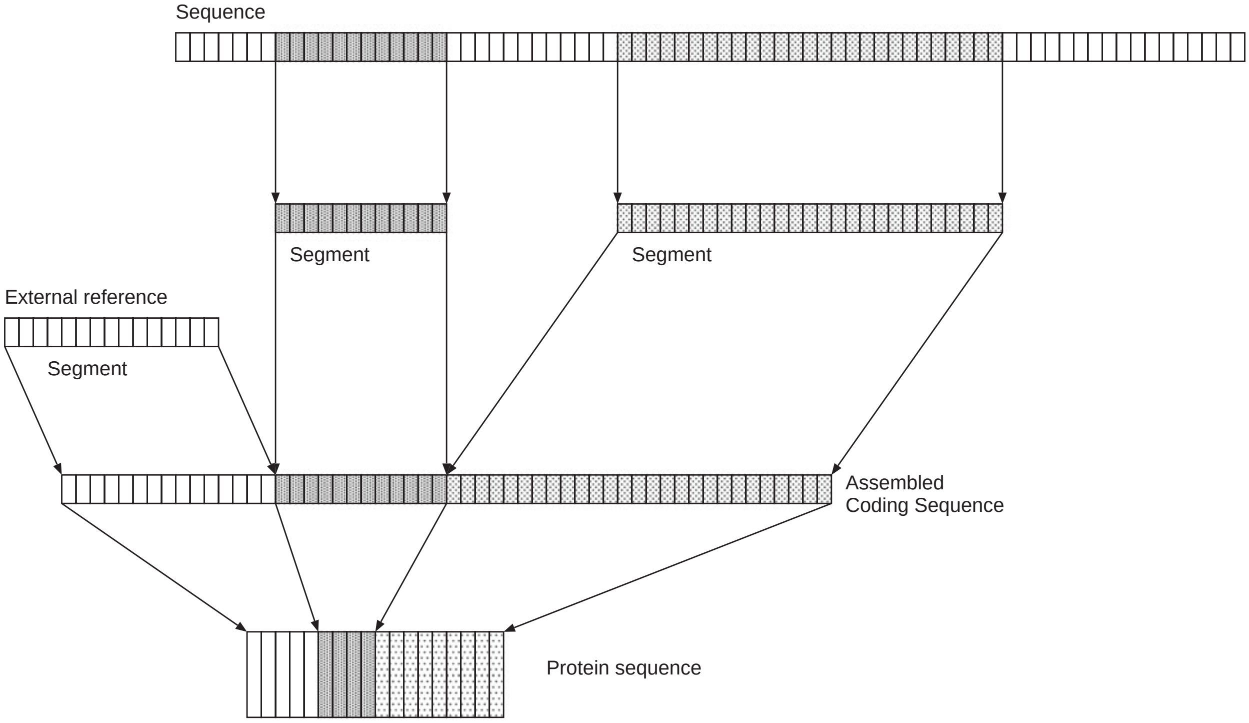

Fig. 3. The mapping procedure must take account of assembly of CDSs which may include segments from other primary database entries.

segments is illustrated in Figure 3. An EMBL entry has a sequence of nucleotide bases and CDS records which define the start- and end-points of 'segments' within this main sequence. These segments may also be in the form of external references to regions which are spliced in from other EMBL entries. The segments are spliced together to form a complete coding sequence which is translated to form a protein. The EMBL also contains a */translation* record which gives the complete translated protein.

Correct handling of the external references is the major problem with mapping the location of an SNP to its location within a protein sequence. In particular, the procedure is made more complex by the fact that a splice site may not coincide with the start of a codon. As shown in Figure 4, an SNP at the position labelled 'A' is at the first base of the sixth codon in reading frame 1 with reference to the whole sequence, or the first base of the first codon in reading frame 2 with respect to the CDS coming from the current EMBL file. When there is no external segment upstream of the SNP site, the reading frame is defined by the first segment specified in the CDS record, but when there is an upstream external segment, we have the problem of identifying the correct reading frame.

We wish to avoid including the actual sequence from the external reference since this requires either a database lookup (if the data are stored in a relational database), or loading and parsing from a flat file (if EMBOSS is used, as here). This would add significantly to the time required to process millions of SNPs and we therefore adopt a simple heuristic of assembling all the internal segments and identifying the continuous assembled segment containing the SNP. This is then translated in all three reading frames and the translations are compared with the */translation* record.

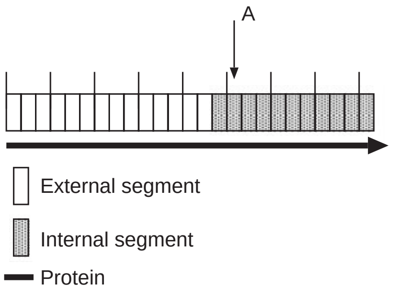

Fig. 4. The mapping is made more complex by the fact that the splice site between an external and internal segment may not coincide with the start of a codon.

`findsnp3` proceeds as follows:

(1) Normalize the SNP sequence, replacing all ambiguous bases with 'n'.

(2) If there are no CDS records, report an 'unknown' status and exit.

(3) Search the full gene sequence separately for the upstream and downstream regions of the SNP sequence. If both are found, ensure that the matches are separated by one base (the SNP site).

(4) If no match is found, or the separation is not one, then reverse-complement the SNP sequence, complement the alleles and repeat the search.

(5) If still no match is found, report an 'error' status and exit.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

(6) If the mutation site does not occur within a CDS, return a 'non-coding' status.

(7) Extract all internal segments from the CDS record of EMBL and assemble all CDSs (skipping external segments).

(8) Extract information about the CDS in which the SNP was found (start and end points, encoded gene name and translation).

(9) Map the SNP from the full gene sequence to the location within the assembled CDS.

(10) If there are external segments upstream of the SNP, check the reading frame.

(11) Extract the coded protein and identify the residue number in which the SNP occurs and the location within the codon.

(12) Finally, report these data.

The output is presented as a single row containing fields separated by vertical bars as described in Table 2 and these data are then stored in an SQL table. The identifier from HGVbase together with the primary sequence database identifier forms a compound primary key into the database.

3.2.3 Mapping between the protein sequence and the structure Having mapped an SNP to a mutation in an amino acid sequence, we now need to find out whether we can map the entry to a known protein structure. Rather than performing a direct search against the PDB, we use an indirect lookup via Swiss-Prot. This was done for two reasons. First, many of the entries in the primary sequence database (EMBL) contain cross-links to Swiss-Prot speeding the processing by removing the need for a search against sequences in the PDB. Second, Swiss-Prot is highly annotated providing a jumping-off point with cross-links to many other databases as well as itself containing detailed information.

The sequence from EMBL reported by the `findsnp3` program is aligned with the sequence extracted from the Swiss-Prot entry using `ssearch34` from the FASTA package (Pearson and Lipman, 1988). A wrapper to the program was written in Python to extract the required alignment. Although in principle the two sequences should be identical, an alignment is required in order to overcome problems caused by the N-terminal methionine which may or may not be present in either entry. In addition, if the sequences get out of synchronization owing to a correction to one of the entries, the alignment should ensure that the mapping remains correct.

After aligning the sequences, we need to extract the residue number for the mutation site within the Swiss-Prot file. We count along the EMBL sequence to the required residue (skipping any insertions) and identify the equivalent location in the Swiss-Prot sequence. We then identify the index of this residue in the Swiss-Prot sequence by again counting residues and skipping insertions.

Having found the location of the mutation in the Swiss-Prot entry, the corresponding location in a PDB file (if any) is found through a pre-calculated mapping of Swiss-Prot residues to PDB residues. This mapping is provided by Sameer Velankar, Phil McNeil and Virginie Mittard from the European Bioinformatics Institute (ftp://ftp.ebi.ac.uk/pub/contrib/mcneil/pdb_sws_mapping.lst.gz).

3.3 The data presentation layer

The web interface, implemented in PHP, is available via http://www.bioinf.org.uk/saap/. The front page presents the search options allowing search by SNP ID, PDB accession, EMBL accession, Swiss-Prot accession and gene symbol or product name extracted from EMBL.

A search by SNP ID takes the user to an intermediate page which summarizes the information available in the SNP entry and presents a pull-down menu allowing the user to select one of the primary database entries to which the SNP is linked. Selecting a primary database entry takes the user to a full extended version of the page which adds information about the protein mutation: the nature of the change (if any) and the location of the mutation in the translation of the primary database entry. Where available, the same information is presented for the associated Swiss-Prot entry and the location of the mutation in any PDB structure is provided. Cross-references to data sources are clickable enabling the original data to be viewed.

A search on any of the other fields presents a summary list of all relevant SNPs. The summary presents the text description, the SNP ID, the primary database entries it references and information on validation, irrespective of whether the SNP is coding or is mapped

Table 2. Fields output by the `findsnp3` program

Field	Meaning
status	Result of the mapping (`ok` or `failed`)
messages	Detailed report explaining a failed status
searchbits	Four boolean flags to indicate which parts of the SNP sequence matched: upstream, downstream, reverse-complement-downstream, reverse-complement-upstream
mutation	`unknown`, `coding`, `non-coding` or `none`
mutation_type	`silent`, `non-silent` (i.e. mis-sense or nonsense) or `none`
direction	Shows the orientation in which the SNP was found: `forward`, `revcomp` or `none`
protein	The protein sequence in which the SNP was found extracted from the */translation* record
cross_reference	Cross reference to Swiss-Prot from the *db_xref* record of EMBL
gene	The gene name from the */gene* or */product* record of EMBL
base_wildtype	The wild type base at the mutation site
aa_wildtype	The wild type amino acid associated with the mutation site
codon	The three-letter codon in which the mutation is located
snp_codon_position	The position of the mutation within the codon (1, 2, 3)
snp_sequence_position	The absolute base number of the mutation with respect to the complete DNA sequence in this database entry
snp_protein_position	The number of the codon containing the SNP in the protein sequence from the */translation* record
alleles	A comma-separated list of known alleles at the mutation site
allele_mutations	A comma-separated list of amino acids at the mutation site corresponding to the alleles
allele_status	A comma-separated list of the effects of each allele: (`nonsense`,`silent` or `mis-sense`)

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

to a known structure. From here, one can choose to view the full extended page described above.

3.4 Automated processing

Automation of the system relies on using Mirror to make local copies of the source databases and the Unix Make program to drive building the database. All parameters for the build (location of data files, database settings, location of the website, etc.) are set via a single configuration file. Excluding mirroring of the data, a complete build of the database takes ∼74 h on an Athlon XP 2800+ Linux machine with 512 Mb RAM. Of this ∼3 h are spent on importing HGVbase into a relational database and indexing EMBL, 70 h are spent on the mapping between HGVbase and EMBL, while the remaining time is taken for the final mapping via Swiss-Prot to the PDB. The process can easily be split across multiple machine should the need arise. At this stage, it was decided not to make the system 'update-able', but to rebuild the database on a weekly basis should any of the source databases have changed. At the end of the build, the website is automatically redirected to the new version of the database. Briefly, the build proceeds as follows:

- a global lock file is created to prevent concurrency problems,
- database tables are created to store the SNP data from HGVbase,
- the SNP data are loaded,
- the EMBL data repository is set up using the `dbiflat` program from EMBOSS,
- a database table is created to map SNPs to EMBL entries,
- SNP data are dumped from the database and processed with `findsnp3`, importing the results into the mapping table,
- each SNP mapped to EMBL is mapped to a Swiss-Prot entry and thence to a PDB structure,
- the global lock file is removed,
- the web pages are switched to access the new database.

4 DISCUSSION

We have presented a system to collect mutation data from HGVbase, process these data and present information about SAAPs, including their locations in known PDB structures in a completely automated fashion. The system performs validation of the mutation data at a number of levels, maps mutations to their locations in a primary sequence database and thence via Swiss-Prot to the PDB. Translation records in EMBL are remarkably inconsistent in their inclusion of N-terminal methionines and the same level of inconsistency is seen in Swiss-Prot records. There is no correlation between these: Swiss-Prot may contain the methionine while EMBL does not and vice versa. This forced us to use an alignment method rather than a direct lookup which will also account for any other minor differences between EMBL and Swiss-Prot entries which may result from an asynchronous update of one of the entries.

The major problems in performing the mapping of SNPs reliably are in handling splicing between EMBL entries. It is also important to realize that there is not a simple one-to-one mapping between SNP entries, EMBL entries and Swiss-Prot entries. One SNP can reference multiple EMBL entries (one might cover a whole chromosome or other large stretch of DNA while another may be a specific gene coding for an individual protein). Similarly, an EMBL entry may contain multiple distinct CDSs, each of which maps to a different Swiss-Prot entry.

An alternative implementation would have been to access (at least some of) the underlying databases remotely using web services such as SOAP. Such an approach may have simplified some of the data access requirements, but would have introduced an enormous time overhead. Thus regular complete rebuilds of the database would have been impractical and an update-able system would have been necessary, increasing the complexity of the system. Access to the underlying sequence databases has been performed with a C++ class wrapper to the EMBOSS libraries. This multilayer design does introduce a processing overhead and an alternative C++ EMBL parser is available (http://www.renatomancuso.com/software/phoenix/phoenix.htm). However, this parser does not support GenBank and we wish to retain the flexibility to access alternative data sources provided by EMBOSS. An increase in throughput can easily be achieved by spreading the processing across a computer farm.

As stated above, of the 2 859 130 entries in HGVbase, 26 313 (0.92%) were discarded as the mutation was undefined or was not a single base change. A further 2003 entries (0.07%) were discarded as invalid. Of the remaining 2 830 810 entries, only 4026 (0.14%), were linked to coding regions of proteins (either sense or mis-sense). Of these, 2384 had protein structure data available (0.08% of the valid SNP data). These numbers may appear very small, but if one assumes a normal protein chain length of between 100 and 500 amino acids, and assuming 25 000 genes in the human genome, then only between 0.23 and 1.17% of SNPs would be expected to code if they are evenly distributed. However, a proportion of missense mutations will be lethal to the cell or will not be present in normal populations surveyed in the SNP databases because they cause inherited disease. HGVbase is therefore expected to be skewed in favour of non-coding mutations. In addition, our validation of the SNP data in mapping to EMBL entries is currently very strict; we require that the upstream and downstream sequences contain only the four bases, ATCG. We plan to relax this constraint and allow ambiguous bases during the matching; preliminary results show this increases the number of SNPs matched to coding regions from 4026 to 15 481 (0.44% of the vald SNP data), well within the expected range.

Future development of the system will concentrate on five major aspects. (1) While HGVbase has a number of advantages over dbSNP, since starting this project, it appears that maintenance and development of HGVbase has, at least for the moment, come to a halt. We are therefore in the process of extending the above protocol to handle dbSNP and GenBank. (2) To support GenBank and to account for the fact that Swiss-Prot annotations are updated more regularly than those in EMBL, we will incorporate links from Swiss-Prot to the underlying primary databases and vice versa. (3) Incorporation of structural analysis. Wherever an SAAP is located in a protein of known structure, the types of automated analysis described by Martin *et al*. (2002) will be performed. (4) We will include further sources of mutation data—primarily data from OMIM (McKusick, 2000) and a simple facility to support LSMDBs. This will allow data to be read from a standardized format; filters can then be added on a gradual basis to convert LSMDBs to this standard form. (5) Addition of phenotype information to the database, i.e. associated disease information and severity, will come from OMIM and LSMDBs.

In conclusion, we believe this to be the first completely automatic and reliable implementation of an SNP to protein mapping

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018

system which correctly accounts for external references in CDSs. The method has been designed with future expansion in mind and will soon include data from dbSNP and GenBank as well as the results of structural analysis.

REFERENCES

Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Brazma,A., Hingamp,P., Quackenbush,J., Sherlock,G., Spellman,P., Stoeckert,C., Aach,J., Ansorge,W., Ball,C.A., Causton,H.C., *et al*. (2001) Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. *Nat. Genet.*, **29**, 365–371.

Brookes,A. (1999) The essence of SNPs. *Gene*, **234**, 177–186.

Capriotti,E., Fariselli,P. and Casadio,R. (2004) A neural-network-based method for predicting protein stability changes upon single point mutations. *Bioinformatics*, **20** (Suppl. 1), I63–I68.

Casillas,S. and Barbadilla,A. (2004) PDA: a pipeline to explore and estimate polymorphism in large DNA databases, *Nucleic Acids Res.*, **32**, W166-W169.

Claustres,M., Horaitis,O., Vanevski,M. and Cotton,R.G.H. (2002) Time for a unified system of mutation description and reporting: a review of locus-specific mutation databases. *Genome Res.*, **12**, 680–688.

Collins,F., Brooks,L. and Chakravarti,A. (1998) A DNA polymorphism discovery resource for research on human genetic variation. *Genome Res.*, **8**, 1229–1231.

Conde,L., Vaquerizas,J.M., Santoyo,J., Al-Shahrour,F., Ruiz-Llorente,S., Robledo,M. and Dopazo,J. (2004) PupaSNP Finder: a web tool for finding SNPs with putative effect at transcriptional level. *Nucleic Acids Res.*, **32**, W242–W248.

Ferrer-Costa,C., Orozco,M. and de la Cruz,X. (2002) Characterization of disease-associated single amino acid polymorphisms in terms of sequence and structure properties. *J. Mol. Biol.*, **315**, 771–786.

Ferrer-Costa,C., Orozco,M. and De La Cruz,X. (2004) Sequence-based prediction of pathological mutations. *Proteins*, **57**, 811–819.

Fredman,D., Siegfried,M., Yuan,Y.P., Bork,P., Lehväslaiho,H. and Brookes,A.J. (2002) HGVbase: a human sequence variation database emphasizing data quality and a broad spectrum of data sources. *Nucleic Acids Res.*, **30**, 387–391.

Kwok,C.J., Martin,A.C.R., Au,S.W.N. and Lam,V.M.S. (2002) G6PDdb, an integrated database of glucose-6-phosphate dehydrogenase (G6PD) mutations. *Human Mutat.*, **19**, 217–224.

Li,W., Ellsworth,D., Krushkal,J., Chang,B. and Hewett-Emmett,D. (1996) Rates of nucleotide substitution in primates and rodents and the generation-time effect hypothesis. *Mol. Phylogenet. Evol.*, **5**, 182–187.

Martin,A.C.R., Facchiano,A.M., Cuff,A.L., Hernandez-Boussard,T., Olivier,M., Hainaut,P. and Thornton,J.M. (2002) Integrating mutation data and structural analysis of the TP53 tumor-suppressor protein. *Human Mutat.*, **19**, 149–164.

McKusick,V.A. (2000) Online Mendelian Inheritance in Man (OMIM) (TM). McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD).

Mullan,L.J. and Bleasby,A.J. (2002) Short EMBOSS user guide. European Molecular Biology Open Software Suite. *Brief Bioinform.*, **3**, 92–94.

Ohno,S. (1984) Birth of a unique enzyme from an alternative reading frame of the preexisted, internally repetitious coding sequence. *Proc. Natl Acad. Sci. USA*, **81**, 2421–2425.

Olson,S.A. (2002) EMBOSS opens up sequence analysis. European Molecular Biology Open Software Suite. *Brief Bioinform.*, **3**, 87–91.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Ramensky,V., Bork,P. and Sunyaev,S. (2002) Human non-synonymous SNPs: server and survey. *Nucleic Acids Res.*, **30**, 3894–3900.

Sanjuán,R., Moya,A. and Elena,S.F. (2004) The distribution of fitness effects caused by single-nucleotide substitutions in an RNA virus. *Proc. Natl Acad. Sci. USA*, **101**, 8396–8401.

Sherry,S.T., Ward,M.H., Kholodov,M., Baker,J., Phan,L., Smigielski,E.M. and Sirotkin,K. (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, **29**, 308–311.

Stenson,P.D., Ball,E.V., Mort,M., Phillips,A.D., Shiel,J.A., Thomas,N.S.T., Abeysinghe,S., Krawczak,M. and Cooper,D.N. (2003) Human Gene Mutation Database (HGMD): 2003 update, *Human Mutat.*, **21**, 577–581.

Sunyaev,S., Ramensky,V. and Bork,P. (2000) Towards a structural basis of human non-synonymous single nucleotide polymorphisms. *Trends Genet.*, **16**, 198–200.

Sunyaev,S., Ramensky,V., Koch,I., Lathe,W., Kondrashov,A. and Bork,P. (2001) Prediction of deleterious human alleles. *Hum. Mol. Genet.*, **10**, 591–597.

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1443/250644
by guest
on 18 May 2018