# Brief Description of Each Table

## blastresults

This table records the results of running PSI-BLAST on the entire human proteome. The table can be used to determine redundancy as it records the sequence identity returned by PSI-BLAST.

## cosmicvariants

This table records somatic variants from COSMIC. Any somatic variants in COSMIC which Ensembl records as being in or adjacent to a gene coding for a human UniProt protein are recorded in this table. Information about the location of the variant and the amino acid change it causes are also recorded.

## drugs

This table records information about drug-target interactions, and the tightness of the binding between the drug and target. There is one record in this table for every drug-interaction, along with the tightest known binding affinity and the dissociation constant.

## ensemblgene

This table records information about the number and type of transcripts. Each Ensembl gene which is determined to generate a UniProt human protein is present, along with the number and type of transcripts generated by the gene.

## entrezgene

This table records information about diseases linked to Entrez genes. Every Entrez gene is recorded, along with the diseases it is determined to be involved in.

## germvariants

This table records germline variants. Any germline variants which Ensembl records as being in a gene coding for a human UniProt protein, are recorded in this table. Information about the location of the variant and the amino acid change it causes are also recorded.

## goinfo

This table contains information about each GO term. The information consists of the type of the term (biological process, cellular component and molecular function), the path from the current term to the root term and the terms one and two levels below the root term.

## homologs

This table contains information about all the homologs of each Ensembl gene which codes for a UniProt human protein. This includes paralogs of the human protein genes, along with orthologs of differing evolutionary distance.

## nonredundant

This table records which proteins are in the non-redundant set of different target and non-target subsets of the human proteome (e.g. GPCR targets/non-targets, cancer targets/non-target, etc.).

## pathways

This table records information about the number of pathways that each UniProt human protein tales part in.

## ppi

This table records binary protein protein interactions for each protein in the human proteome. Information about the isoform of the protein involved in the interaction, whether the two proteins are both human proteins and the number of experiments which have shown the interaction to occur is also recorded.

## proteininfo

This table records basic information about each human protein in UniProt. Additional information predicated or calculated from the sequence of the protein is also recorded. Whether or not the protein is a target of an approved drug is also recorded here. Examples of the type of information recorded are sequence composition statistics, isoelectric point, subcellular location and secondary structure information. This table is the hub of the database, with every other table extending the basic information recorded here.

## stability

This table records the predicted *in vivo* half-life of the protein, and the *in vitro* stability of the protein.

## unigene

This table records information about the expression of the human proteins. This includes information about the developmental stage, health state and body site where the protein is expressed.

## unigenetotals

This table records the total number of ESTs recorded for each developmental stage, health state and body site. This data can be used to calculate relative expression profiles for each protein.

## uniprot2ensembl

This table records the mapping of UniProt accessions to Ensembl gene/transcript/protein IDs.

## uniprot2entrez

This table records the mapping of UniProt accessions to Entrez gene IDs.

## uniprot2go

This table records the mapping of UniProt accessions to GO term IDs.

## uniprot2unigene

The table records the mapping of UniProt accessions to UniGene IDs.

# Table Columns, and How The Data is Generated

No column value will ever be NULL.

If there is no default value specified for a column, then the column will always have a non-default value.

Many definitions for Ensembl terms (such as the transcript types) can be found in the Ensembl Glossary <http://www.ensembl.org/Help/Glossary?>

The format of the columns is:

1. Name (attribute TYPE) – Description.

## blastresults

1. ProteinA (VARCHAR(10)) – The accession of one of the two proteins whose similarity is recorded. Foreign key from proteininfo.UPAccession.
2. ProteinB (VARCHAR(10)) – The accession of the other of the two proteins whose similarity is recorded. Foreign key from proteininfo.UPAccession.
3. Similarity (FLOAT) – The percentage sequence identity of the two proteins. Calculated.
4. Length (INT) – The length of the local alignment returned by BLAST.
5. EValue (FLOAT) – The E-Value for the alignment returned by BLAST.

### Extra Information

PSI-BLAST from the BLAST+ package by NCBI (<http://www.ncbi.nlm.nih.gov/books/NBK1763/>) is used to determine the percentage sequence identity of the proteins. The proteins in the database are BLASTed one by one against every other protein to determine all N2 possible sequence identities. Not every pair of proteins will have a recorded sequence identity. This is because the restrictions placed on the BLASTing ensure that weak alignments (e.g. alignments with large E-Values) are not recorded. The arguments used for the BLASTing are:

* -evalue 1
* -inclusion\_ethresh 0.0001
* -num\_iterations 3
* -gap\_trigger 18
* -num\_descriptions 10000
* -num\_alignments 10000
* -dbsize 0
* -outfmt "7 qseqid sseqid pident length evalue"
* -num\_threads 2

## cosmicvariants

The descriptions of the different variant types are taken from Ensembl’s description of the variants recorded there (<http://www.ensembl.org/info/docs/variation/index.html>).

1. EnsemblTranscriptID (VARCHAR(45)) – The ID of the Ensembl transcript that the variant occurs in.
2. VariantID (VARCHAR(45)) – The ID of the variant.
3. EnsemblGeneID (VARCHAR(45)) – The ID of the Ensembl gene that the transcript is generated from.
4. InitialAminoAcid (VARCHAR(45)) – The amino acid(s) changed by the variant.
5. FinalAminoAcid (VARCHAR(45)) – The resulting amino acid(s) of the variant.
6. CDSStart (INT) – The location in the gene where the variant starts.
7. CDSEnd (INT) – The location in the gene where the variant ends.
8. 3Untranslated (INT) – 1 if the variant occurs in a 3’ untranslated region, otherwise 0.
9. 5Untranslated (INT) – 1 if the variant occurs in a 5’ untranslated region, otherwise 0.
10. CodingUnknown (INT) – 1 if the variant occurs in a coding sequence with indeterminate effect, otherwise 0.
11. ComplexInDel (INT) – 1 if the variant is an insertion or deletion that spans an exon/intron or coding sequence/UTR border, otherwise 0.
12. Downstream (INT) – 1 if the variant occurs within 5 kb downstream of the 3 prime end of a transcript, otherwise 0.
13. EssentialSpliceSite (INT) – 1 if the variant occurs in the first 2 or the last 2 basepairs of an intron, otherwise 0.
14. FrameshiftCoding (INT) – 1 if the variant occurs in a coding sequence and results in a frameshift, otherwise 0.
15. Intergenic (INT) – 1 if the variant occurs more than 5 kb either upstream or downstream of a transcript, otherwise 0.
16. Intronic (INT) – 1 if the variant occurs in an intron, otherwise 0.
17. NMDTranscript – 1 if the variant occurs within a transcript predicted to undergo nonsense-mediated decay, otherwise 0.
18. NonSynonymousCoding (INT) – 1 if the variant occurs in a coding sequence and results in an amino acid change in the encoded peptide sequence, otherwise 0.
19. PartialCodon (INT) – 1 if the variant occurs in within the final, incomplete codon of a transcript whose end coordinate is unknown, otherwise 0.
20. RegulatoryRegion (INT) – 1 if the variant occurs in a regulatory region annotated by Ensembl, otherwise 0.
21. SpliceSite (INT) – 1 if the variant occurs 1-3 bps into an exon or 3-8 bps into an intron, otherwise 0.
22. StopGained (INT) – 1 if the variant occurs in a coding sequence, resulting in the gain of a stop codon, otherwise 0.
23. StopLost (INT) – 1 if the variant occurs in a coding sequence, resulting in the loss of a stop codon, otherwise 0.
24. SynonymousCoding (INT) – 1 if the variant occurs in a 3’ untranslated region, otherwise 0.
25. TranscriptionFactorBindingMotif (INT) – 1 if the variant occurs in a 3’ untranslated region, otherwise 0.
26. Upstream (INT) – 1 if the variant occurs within 5 kb upstream of the 5 prime end of a transcript, otherwise 0.
27. WithinMatureMIRNA (INT) – 1 if the variant occurs within a microRNA, otherwise 0.
28. WithinNonCodingGene (INT) – 1 if the variant occurs within a gene that does not code for a protein, otherwise 0.

### Extra Information

The steps by which the data for this table are gathered are as follows:

1. The IDs of the Ensembl genes which correspond to the UniProt proteins are determined. This is done using the Ensembl BioMart
2. The somatic variants that Ensembl records as being linked to the gene IDs recorded in 1 are determined. This is done using the Ensembl BioMart.
3. From the variants recorded in 2, only those that are from COSMIC have their information recorded. The COSMIC IDs of these variants are also recorded separately.
4. The initial amino acid(s), final amino acid(s), CDS start position and CDS end position for all the COSMIC IDs found in 3 are determined. This is done using the COSMIC BioMart.

## drugs

1. UPAccession (VARCHAR(10)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession. This is the protein which is targeted by the drug in question.
2. DrugID (VARCHAR(45)) – The unique ID of the drug. This can be either the PubChem CID, or blank.
3. DrugName (VARCHAR(255)) – The name of the drug.
4. KiValue (FLOAT) – The Ki (binding affinity) value for the binding between the drug and the protein. The value is given in nM. If there is no known Ki value for the drug-target interaction, then the value is -1.
5. KdValue (FLOAT) – The Kd (dissociation constant) for the binding between the drug and the protein. The value is given in nM. If there is no known Kd value for the drug-target interaction, then the value is -1.

### Extra Information

The drug IDs and names are extracted from DrugBank, the TTD and CheMBL. Each of the IDs is then converted to the corresponding PubChem CID. For DrugBank and the TTD there is a record of the mapping to PubChem CIDs in DrugBank and the TTD. For ChEMBL the Entrez E-Utilities.

The Ki and Kd values come from ChEMBL and BindingDB. For ChEMBL the extraction process was:

1. Query the local ChEMBL mirror to extract all drug IDs (molregno in ChEMBL terms) and UniProt accessions, where at least one assay linking the drug has a confidence score of 4 or more. This means that the assay has been assigned a protein target (<https://www.ebi.ac.uk/chembldb/index.php/faq#faq4>).
2. In addition extract the quantitative information about all the assays where the confidence score is at least 4. This includes the value and type of the activity measurements from the assay. Type being IC50, EC50, Ki, Kd, etc.
3. This left a mapping between the UniProt targets (not necessarily human) and drug IDs, and therefore a list of all drug-target interactions with activity information about the interaction.

For BindingDB the extraction process was:

1. Go through the local copy of the database collecting all compound-target interaction, and keep only those interactions that contained human proteins.
2. Next the compound from BindingDB had to be crosschecked against ChEMBL, DrugBank and the TTD to determine if they were approved drugs. Any compounds found not to be approved drugs were discarded, along with their compound-target interactions.
3. This left a mapping between the human proteins and the compounds known to be drugs, along with the binding information, only Ki and Kd were recorded, about the compound-target interaction.

Both the final mapping from ChEMBL and the final mapping from BindingDB had to be cross-referenced against the known UniProt human protein accessions. Consideration was given to the fact that the accession extracted from ChEMBL/BindingDB may be an old one, and so all extracted accessions were mapped to current UniProt accessions if possible. This left only mappings between known drug compounds and valid current UniProt human protein accessions. If there was more than one mapping between a given drug and a protein, then only the smallest Ki and Kd values were recorded. This was because the tightest biding is sought.

## ensemblgene

1. EnsemblGeneID (VARCHAR(45)) – The ID of the Ensembl gene to which the transcripts correspond.
2. NumberTranscripts (INT) – The total number of transcripts. Type: INT.
3. ProteinCodingTranscripts (INT) – The number of transcripts which are a spliced mRNA that leads to a protein product.
4. RetainedIntronTranscripts (INT) – The number of noncoding transcripts that contain an intronic sequence.
5. ProcessedTranscripts (INT) – The number of noncoding transcripts that do not contain an open reading frame.
6. NonsenseMediatedDecayTranscripts (INT) – The number of transcript is thought to undergo nonsense mediated decay.

## entrezgene

1. GeneID (VARCHAR(45)) – The ID of the Entrez gene.
2. Cancer\_Ovarian (VARCHAR(1)) – Y if the gene is involved in ovarian cancer, else N.
3. Cancer\_Melanoma (VARCHAR(1)) – Y if the gene is involved in melanoma, else N.
4. Cancer\_Breast (VARCHAR(1)) – Y if the gene is involved in breast cancer, else N.
5. Cancer\_Lung (VARCHAR(1)) – Y if the gene is involved in lung cancer, else N.
6. Cancer\_Pancreatic (VARCHAR(1)) – Y if the gene is involved in pancreatic cancer, else N.
7. Cancer\_Liver (VARCHAR(1)) – Y if the gene is involved in liver cancer, else N.
8. Cancer\_Colon (VARCHAR(1)) – Y if the gene is involved in colon cancer, else N.
9. Cancer\_Prostate (VARCHAR(1)) – Y if the gene is involved in prostate cancer, else N.
10. Cancer\_Testicular (VARCHAR(1)) – Y if the gene is involved in testicular cancer, else N.
11. Cancer\_Oesophageal (VARCHAR(1)) – Y if the gene is involved in oesophageal cancer, else N.
12. Cancer\_Stomach (VARCHAR(1)) – Y if the gene is involved in stomach cancer, else N.
13. Cancer\_Heart (VARCHAR(1)) – Y if the gene is involved in heart cancer, else N.
14. Cancer\_Oral\_Cavity (VARCHAR(1)) – Y if the gene is involved in oral cavity cancer, else N.
15. Cancer\_Leukaemia (VARCHAR(1)) – Y if the gene is involved in leukaemia, else N.
16. Cancer\_Lymphoma (VARCHAR(1)) – Y if the gene is involved in lymphoma, else N.
17. Cancer\_Intestinal (VARCHAR(1)) – Y if the gene is involved in intestinal cancer, else N.
18. Cancer\_All (VARCHAR(1)) – Y if the gene is involved in any form of cancer, else N.

### Extra Information

The information for this table is gathered as follows:

1. A local copy of the GeneRIF database is parsed to record all GeneRIF entries relating to each human Entrez gene.
2. The disease ontology is parsed to extract the set of descriptions of all terms that are a descendant of one of the desired disease terms. This generates a number of sets of disease ontology terms, which can be searched for in the GeneRIF entries. The desired Disease Ontology term IDs are:
   1. Ovarian Cancer – 2394
   2. Melanoma – 1909
   3. Breast cancer – 1612
   4. Lung cancer – 1324
   5. Pancreatic cancer – 1793
   6. Liver cancer – 3571
   7. Colon cancer – 219
   8. Prostate cancer – 10283
   9. Testicular cancer – 2998
   10. Oesophageal cancer – 5041
   11. Stomach cancer – 10534
   12. Heart cancer – 117
   13. Oral cavity cancer – 8618
   14. Leukimia – 1240
   15. Lymphoma – 0060058
   16. Intestinal cancer – 10155
   17. Brain cancer – 1319
   18. cancer – 162
3. This step is repeated once for every set of Disease Ontology terms, as an example ovarian cancer will be used. The GeneRIF entries for each human gene are searched to see if they contain any of the ovarian cancer descendant terms, for example ovarian malignant mesothelioma (DOID:2143). If any of the GeneRIF entries for a gene have a term from the ovarian cancer term set, then the gene is marked as being involved in ovarian cancer. This is repeated for every Entrez gene.

## germvariants

The descriptions of the different variant types are taken from Ensembl’s description of the variants recorded there (<http://www.ensembl.org/info/docs/variation/index.html>).

1. EnsemblTranscriptID (VARCHAR(45)) – The ID of the Ensembl transcript that the variant occurs in.
2. VariantID (VARCHAR(45)) – The ID of the variant.
3. EnsemblGeneID (VARCHAR(45)) – The ID of the Ensembl gene that the transcript is generated from.
4. AAChange (VARCHAR(255)) – The amino acid(s) changed by the variant. The initial amino acid(s) and the variant amin acids(s) are separated by a ‘/’. If the amino acid change caused by the variant is not known, then this field is ‘-/-‘.
5. TranslationStart (INT) – The location in the amino acid sequence where the variant begins. If the start position of the variant is not known, then is field is -1.
6. TranslationEnd (INT) – The location in the amino acid sequence where the variant ends. If the end position of the variant is not known, then is field is -1.
7. CDSStart (INT) – The location in the CDS sequence of the gene where the variant starts. If the start position of the variant is not known, then is field is -1.
8. CDSEnd (INT) – The location in the CDS sequence of the gene where the variant ends. If the end position of the variant is not known, then is field is -1.
9. 3Untranslated (INT) – 1 if the variant occurs in a 3’ untranslated region, otherwise 0.
10. 5Untranslated (INT) – 1 if the variant occurs in a 5’ untranslated region, otherwise 0.
11. CodingUnknown (INT) – 1 if the variant occurs in a coding sequence with indeterminate effect, otherwise 0.
12. ComplexInDel (INT) – 1 if the variant is an insertion or deletion that spans an exon/intron or coding sequence/UTR border, otherwise 0.
13. Downstream (INT) – 1 if the variant occurs within 5 kb downstream of the 3 prime end of a transcript, otherwise 0.
14. EssentialSpliceSite (INT) – 1 if the variant occurs in the first 2 or the last 2 basepairs of an intron, otherwise 0.
15. FrameshiftCoding (INT) – 1 if the variant occurs in a coding sequence and results in a frameshift, otherwise 0.
16. Intergenic (INT) – 1 if the variant occurs more than 5 kb either upstream or downstream of a transcript, otherwise 0.
17. Intronic (INT) – 1 if the variant occurs in an intron, otherwise 0.
18. NMDTranscript – 1 if the variant occurs within a transcript predicted to undergo nonsense-mediated decay, otherwise 0.
19. NonSynonymousCoding (INT) – 1 if the variant occurs in a coding sequence and results in an amino acid change in the encoded peptide sequence, otherwise 0.
20. PartialCodon (INT) – 1 if the variant occurs in within the final, incomplete codon of a transcript whose end coordinate is unknown, otherwise 0.
21. RegulatoryRegion (INT) – 1 if the variant occurs in a regulatory region annotated by Ensembl, otherwise 0.
22. SpliceSite (INT) – 1 if the variant occurs 1-3 bps into an exon or 3-8 bps into an intron, otherwise 0.
23. StopGained (INT) – 1 if the variant occurs in a coding sequence, resulting in the gain of a stop codon, otherwise 0.
24. StopLost (INT) – 1 if the variant occurs in a coding sequence, resulting in the loss of a stop codon, otherwise 0.
25. SynonymousCoding (INT) – 1 if the variant occurs in a 3’ untranslated region, otherwise 0.
26. TranscriptionFactorBindingMotif (INT) – 1 if the variant occurs in a 3’ untranslated region, otherwise 0.
27. Upstream (INT) – 1 if the variant occurs within 5 kb upstream of the 5 prime end of a transcript, otherwise 0.
28. WithinMatureMIRNA (INT) – 1 if the variant occurs within a microRNA, otherwise 0.
29. WithinNonCodingGene (INT) – 1 if the variant occurs within a gene that does not code for a protein, otherwise 0.

### Extra Information

The steps by which the data for this table are gathered are as follows:

1. The IDs of the Ensembl genes which correspond to the UniProt proteins are determined. This is done using the Ensembl BioMart.
2. The germ variants that Ensembl records as being linked to the gene IDs recorded in 1 are determined. This is done using the Ensembl Variation Perl API.

If the consequence to transcript for a variant is intronic, upstream or downstream (or any of these three terms in combination with other terms), then the (variant, transcript) pair is not recorded. The method used to determine the genetic variants returns failed variants, in addition to validated ones. Ensembl classifies a variant as failed for multiple reasons (<http://www.ensembl.org/info/docs/variation/data_description.html>), but primarily because a variant maps to over three locations. In order to combat this, any variant that maps to greater than one location is analysed further. The variant is only kept if it maps to only one transcript, and all of the recorded consequences to the transcript are the same. The amino acid change does not have to be identical. This is because it is the type of change induced by the variant that is of interest, not the actual sequence location/amino acid change.

## goinfo

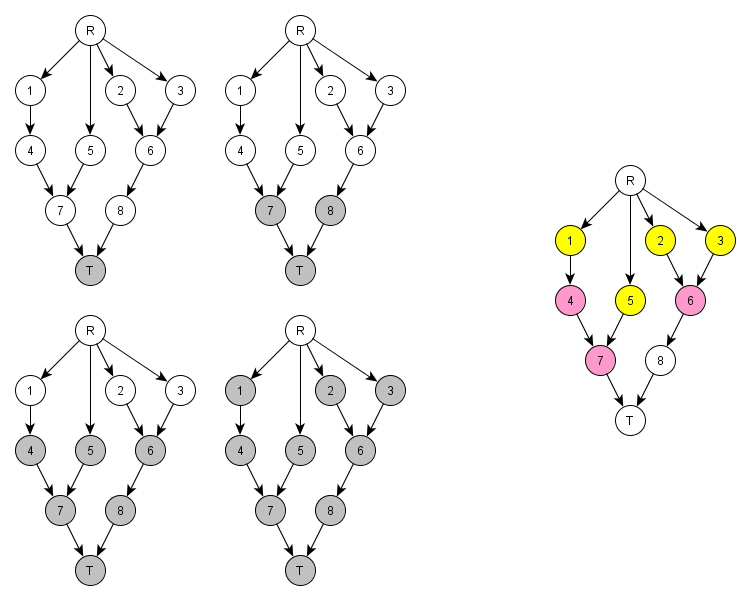
1. GOTermID (INT) – The accession (GO:...) of the GO term.
2. GOName (LONGTEXT) – The name if the GO term.
3. GOType (VARCHAR(45)) – The type of the GO term.
4. GOPaths (LONGTEXT) – The list(s) of terms that lead from the current term to the root term (biological process, cellular location or molecular function). Each term in a single path is separated by a #, and each path is separated by a semi-colon.
5. LevelOne (LONGTEXT) – The term(s) along the path(s) to the root term that are direct descendants of the root term. If there are multiple level one terms, then each term is separated by a semi-colon. If there are no level one terms, then the value of rhis column is NA.
6. LevelTwo (LONGTEXT) – The term(s) along the path(s) to the root term that are grandchildren of the root term. If there are multiple level two terms, then each one is separated by a semi-colon. If there are no level two terms, then the value of this column is NA.

### Extra Information

The paths are calculated by treating the GO as a directed acyclic graph (DAG) (which it is). Starting from the current term T, the DAG is traversed backwards to the root node R. Every possible path backwards from T to R is determined using a breadth-first search. Starting with T, the ancestors of T are determined, generating paths of length two (i.e. contain two terms). The ancestors of the ancestors are then determined, generating paths of three terms. Once a path has reached the root it is no longer extended, this is because there are no more ancestors for that path. This process of adding ancestors is continued until all paths have reached the root.

For the level one and level two terms, a term can only appear in either set a maximum of one time. In other words, neither the level one nor the level two set are permitted to have duplicates.

The diagram shows how the DAG is traversed backwards. The path T->7->5 finished before any of the other paths, as it reaches R first. We can also see the level one terms in yellow, and the level two terms in pink. Term 6 would be in the level two term set twice, as it is the level two term along both the T->8->6->2 and T->8->6->3 paths.



## homologs

1. HumanGene (VARCHAR(45)) – The Ensembl ID of the human gene. Foreign key from ensemblgene.EnsemblGeneID.
2. HomologGene (VARCHAR(45)) – The Ensembl ID of the homologous gene.
3. HomologSpecies (VARCHAR(255)) – The species that the homologous gene comes from.
4. HomologyType (VARCHAR(45)) – The type of homology.
5. Ancestor (VARCHAR(255)) – The most recent common ancestor of the two genes.
6. dN (FLOAT) - The number of non-synonymous substitutions per non-synonymous site.
7. dS (FLOAT) - The number of synonymous substitutions per synonymous site.
8. PeptideAligned (FLOAT) – The percentage of the peptide which has been aligned.
9. Identity (FLOAT) – The percentage sequence identity between both homologs
10. Positivity (FLOAT) - The percentage of positivity (similarity) between both homologs.

### Extra Information

The possible values for HomologyType are:

* ortholog\_one2one
* apparent\_ortholog\_one2one
* ortholog\_one2many
* ortholog\_many2many
* possible\_ortholog
* between\_species\_paralog
* within\_species\_paralog
* other\_paralog
* contiguous\_gene\_split
* putative\_gene\_split

Definitions for what these mean, and how Ensembl predicts homologs, can be found [here](http://www.ensembl.org/info/docs/compara/homology_method.html).

The steps by which the data for this table are gathered are as follows:

1. The IDs of the Ensembl genes which correspond to the UniProt proteins are determined. This is done using The Ensembl Biomart
2. The IDs found in 1 are used to query the Ensembl Compara database via the Ensembl Compara API. The Multi species name is used to access the information which the Ensembl BioMart provides a view of. For every ID found in 1, all of the genes it is homologous to are extracted.
3. The IDs found in 1 are used to query the Ensembl Genome Pan-Taxonomic database via the Ensembl Compara API. For every ID found in 1, all of the genes it is homologous to are extracted.

The difference between steps 2 and 3 is that step 2 only provides access to the data which Ensembl BioMart does. Step 3 provides access to the Pan-Taxonomic database, which enables the extraction of homologies between human genes and organisms such as *E. Coli*, *A. Thanliana* etc. which are not in the standard Ensembl database.

## nonredundant

1. UPAccession (VARCHAR(10)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. AllTargetPositive (VARCHAR(1)) – Y if the protein is a non-redundant drug target, else N.
3. AllTargetNegative (VARCHAR(1)) – Y if the protein is a non-redundant non-target, else N.
4. GPCRTargetPositive (VARCHAR(1)) – Y if the protein is a non-redundant GPCR drug target, else N.
5. GPCRTargetNegative (VARCHAR(1)) – Y if the protein is a non-redundant GPCR non-target, else N.
6. IonChannelTargetPositive (VARCHAR(1)) – Y if the protein is a non-redundant ion channel drug target, else N.
7. IonChannelTargetNegative (VARCHAR(1)) – Y if the protein is a non-redundant ion channel non-target, else N.
8. KinaseTargetPositive (VARCHAR(1)) – Y if the protein is a non-redundant kinase drug target, else N.
9. KinaseTargetNegative (VARCHAR(1)) – Y if the protein is a non-redundant kinase non-target, else N.
10. ProteaseTargetPositive (VARCHAR(1)) – Y if the protein is a non-redundant protease drug target, else N.
11. ProteaseTargetNegative (VARCHAR(1)) – Y if the protein is a non-redundant protein non-target, else N.
12. CancerTargetPositive (VARCHAR(1)) – Y if the protein is a non-redundant cancer drug target, else N
13. CancerTargetNegative (VARCHAR(1)) – Y if the protein is involved in cancer, and is a non-redundant non-target, else N.
14. CancerTypePositive (VARCHAR(1)) – Y if the protein is a non-redundant cancer drug target, else N.
15. CancerTypeNegative (VARCHAR(1)) – Y if the protein is a non-redundant non-cancer drug target, else N.
16. CancerProteinPositive (VARCHAR(1)) – Y if the protein is a non-redundant cancer protein, else N.
17. CancerProteinNegative (VARCHAR(1)) – Y if the protein is a non-redundant non-cancer protein, else N.

### Extra Information

The non-redundancy of the proteins is determined by using the Leaf algorithm. Datasets for all of the different options (e.g. target, non-target, GPCR target, GPCR non-target, etc.) are generated, and the redundancy removed from those datasets using the Leaf algorithm and the data in the proteinInfo.blastresults table. A ‘cancer protein’ is a protein that is determined to be involved in cancer. A cancer drug target is a protein that is the target of a cancer drug.

## pathways

1. UPAccession (VARCHAR(10)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. NumberOfPathways (INT) – The number of pathways that the protein is involved in.

### Extra Information

The PathwayCommons database is used to determine the number of pathways that a protein is involved in. The *Homo sapiens* gene sets file from (<http://www.pathwaycommons.org/pc-snapshot/current-release/gene_sets/by_species/>) was downloaded and parsed. This file contains information about the proteins/genes present in each pathway.

## ppi

1. PPIProteinOne (VARCHAR(10)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. PPIProteinTwo (VARCHAR(10)) – The UniProt accession for the protein. Does not have to be a human accession, and therefore is not a foreign key.
3. IsoformID (VARCHAR(45)) – The number identifying the isoform of PPIProteinTwo that interacts with PPIProteinOne, else ‘No Isoform’. For example, if the protein which interacts with PPIProteinOne is Q9P0K1-3, then PPIProteinTwo is Q9P0K1 and IsoformID is 3. Whereas, if the protein which interacts with PPIProteinOne is Q9P0K1, then PPIProteinTwo is Q9P0K1 and IsoformID is ‘No Isoform’.
4. OrganismsDiffer (VARCHAR(10)) – false if PPIProteinOne and PPIProteinTwo come from the same organism, else true. If the proteins come from the same organisms it means that they are both human.
5. NumExperiments (INT) – The number of experiments in the IntAct database which support the interaction.

### Extra Information

The PPI data is extracted from the UniProt human proteome files. The data is extracted from uniprot.entry.comment when the type attribute is interaction. PPIProteinTwo and IsoformID comes from uniprot.entry.comment.interactant.id (the second uniprot.entry.comment.interactant, as the first one corresponds to the InterAct ID of PPIProteinOne). OrganismsDiffer comes from uniprot.entry.comment.organismsDiffer, and NumExperiments comes from uniprot.entry.comment.experiments. If both uniprot.entry.comment.interactant entries are the same, then the interaction is homotypic. In this case both PPIProteinOne and PPIProteinTwo are the same, and IsoformID is ‘No Isoform’.

## proteininfo

1. UPAccession (VARCHAR(10)) – The UniProt accession for the protein. Extracted from uniprot.entry.accession in the UniProt XML file.
2. ProteinName (VARCHAR(45)) – The name of the protein. Extracted from uniprot.entry.name in the UniProt XML file.
3. A (FLOAT) – The fraction of the protein sequence composed of alanine. Calculated.
4. C (FLOAT) – The fraction of the protein sequence composed of cysteine. Calculated.
5. D (FLOAT) – The fraction of the protein sequence composed of aspartic acid. Calculated.
6. E (FLOAT) – The fraction of the protein sequence composed of glutamic acid. Calculated.
7. F (FLOAT) – The fraction of the protein sequence composed of phenylalanine. Calculated.
8. G (FLOAT) – The fraction of the protein sequence composed of glycine. Calculated.
9. H (FLOAT) – The fraction of the protein sequence composed of histidine. Calculated.
10. I (FLOAT) – The fraction of the protein sequence composed of isoleucine. Calculated.
11. K (FLOAT) – The fraction of the protein sequence composed of lysine. Calculated.
12. L (FLOAT) – The fraction of the protein sequence composed of leucine. Calculated.
13. M (FLOAT) – The fraction of the protein sequence composed of methionine. Calculated.
14. N (FLOAT) – The fraction of the protein sequence composed of asparagine. Calculated.
15. P (FLOAT) – The fraction of the protein sequence composed of proline. Calculated.
16. Q (FLOAT) – The fraction of the protein sequence composed of glutamine. Calculated.
17. R (FLOAT) – The fraction of the protein sequence composed of arginine. Calculated.
18. S (FLOAT) – The fraction of the protein sequence composed of serine. Calculated.
19. T (FLOAT) – The fraction of the protein sequence composed of threonine. Calculated.
20. V (FLOAT) – The fraction of the protein sequence composed of valine. Calculated.
21. W (FLOAT) – The fraction of the protein sequence composed of tryptophan. Calculated.
22. Y (FLOAT) – The fraction of the protein sequence composed of tyrosine. Calculated.
23. NegativelyCharged (FLOAT) – The fraction of the protein sequence composed of negatively charged residues. Calculated.
24. PositivelyCharged (FLOAT) – The fraction of the protein sequence composed of positively charged residues. Calculated.
25. Basic (FLOAT) – The fraction of the protein sequence composed of basic residues. Calculated.
26. Charged (FLOAT) – The fraction of the protein sequence composed of charged residues. Calculated.
27. Polar (FLOAT) – The fraction of the protein sequence composed of polar residues. Calculated.
28. NonPolar (FLOAT) – The fraction of the protein sequence composed of non-polar residues. Calculated.
29. Aromatic (FLOAT) – The fraction of the protein sequence composed of aromatic residues. Calculated.
30. Aliphatic (FLOAT) – The fraction of the protein sequence composed of aliphatic residues. Calculated.
31. Small (FLOAT) – The fraction of the protein sequence composed of small residues. Calculated.
32. Tiny (FLOAT) – The fraction of the protein sequence composed of tiny residues. Calculated.
33. PESTMotif (INT) – The number of PEST motifs present in the protein sequence. Calculated.
34. LowComplexity (INT) – The number of low complexity regions in the protein sequence. Calculated.
35. Hydrophobicity (FLOAT) – The hydrophobicity of a protein. Calculated.
36. Isoelectric (FLOAT) – The isoelectric point of the protein. Calculated.
37. ModeOfAction (VARCHAR(45)) – The type of protein. Recorded types are GPCR, ion channel, kinase and protease/peptidase. Extracted from UniProt files. If the protein does not fall into one of the four categories, then this field will be NA.
38. ECNumber (VARCHAR(45)) – The EC number of the protein, if it has one. Extracted from uniprot.entry.dbReference in the UniProt XML file, where the type attribute is EC. If the protein does not have an EC number, then this field will be NA.
39. O-Glycosylation (LONGTEXT) – The sites along the protein sequence where O-linked glycosylation occurs. Extracted from uniprot.entry.feature.position in the UniProt XML file, when the type attribute of uniprot.entry.feature is glycosylation site and the description attribute begins with O-linked. Recorded as a list of the sequence positions where the glycosylation occurs (i.e. a list of integers), split by semi-colons (e.g. 12;136;210;349). If there are no O-linked glycosylation sites, then this field will be NA.
40. N-Glycosylation (LONGTEXT) – The sites along the protein sequence where N-linked glycosylation occurs. Extracted from uniprot.entry.feature.position in the UniProt XML file, when the type attribute of uniprot.entry.feature is glycosylation site and the description attribute begins with N-linked. Recorded as a list of the sequence positions where the glycosylation occurs (i.e. a list of integers), split by semi-colons (e.g. 12;136;210;349). If there are no N-linked glycosylation sites, then this field will be NA.
41. Phosphoserine (LONGTEXT) – The sites along the protein sequence where serine is phosphorylated. Extracted from uniprot.entry.feature.position in the UniProt XML file, when the type attribute of uniprot.entry.feature is modified residue and the description attribute is Psosphoserine. Recorded as a list of the sequence positions where the phosphorylation occurs (i.e. a list of integers), split by semi-colons (e.g. 12;136;210;349). If there are no phosphoserine sites, then this field will be NA.
42. Phosphothreonine (LONGTEXT) – The sites along the protein sequence where threonine is phosphorylated. Extracted from uniprot.entry.feature.position in the UniProt XML file, when the type attribute of uniprot.entry.feature is modified residue and the description attribute is Phosphothreonine. Recorded as a list of the sequence positions where the phosphorylation occurs (i.e. a list of integers), split by semi-colons (e.g. 12;136;210;349). If there are no phosphothreonine sites, then this field will be NA.
43. Phosphotyrosine (LONGTEXT) – The sites along the protein sequence where tyrosine is phosphorylated. Extracted from uniprot.entry.feature.position in the UniProt XML file, when the type attribute of uniprot.entry.feature is modified residue and the description attribute is Phosphotyrosine. Recorded as a list of the sequence positions where the phosphorylation occurs (i.e. a list of integers), split by semi-colons (e.g. 12;136;210;349). If there are no phosphotyrosine sites, then this field will be NA.
44. SubcellularLocation (LONGTEXT) – The subcellular location of the protein. Extracted from uniprot.entry.comment, when the type attribute of uniprot.entry.comment is subcellular location. Multiple locations are split by semi-colons (e.g. membrane;golgi;other loc). The information is recorded as a tuple, with the elements split by commas. The first element is the isoform name (extracted from uniprot.entry.comment.molecule), or ‘MatureProtein’ if there is no isoform, and the second is the subcellular location (extracted from uniprot.entry.comment.subcellularLocation.location). If there are multiple subcellular locations, then multiple tuples are separated by semi-colons (e.g. MatureProtein,locm1,iso1,loc11,iso1,loc12,iso2,loc21). It is only the top element of the location ontology that is extracted, and no topology or orientation information is extracted. If there is no subcellular location information for the protein, then this field will be NA.
45. TopologicalDomain (LONGTEXT) – The subcellular compartment where each non-membrane region of a membrane spanning protein is found. Extracted from uniprot.entry.feature in the UniProt XML file, when the uniprot.entry.feature type attribute is topological domain. The information is recorded as a 3-tuple, with the elements split by commas. The first element is the domain name (the value of the description attribute in uniprot.entry.feature), the second the position in the protein sequence where the domain starts (the value of the position attribute for uniprot.entry.feature.location.begin) and the third the position in the sequence where the domain ends (the value of the position attribute for uniprot.entry.feature.location.end). If there are multiple domains, then multiple 3-tuples are separated by semi-colons (e.g. dom1,s1,e1;dom2,s2,e2;dom3,s3,e3). If there is no topological domain information for the protein, then this field will be NA.
46. PredictedSubcellularLocation (LONGTEXT) – The results of predicting subcellular localisations using the WoLFPSORT algorithm. Predicted locations, along with the prediction confidence for each location, are recorded. The format for the data is loc1,conf1;loc2,conf2;loc3;conf3... If there is no predicted value for a protein, then this field will be NA.
47. SignalPeptide (LONGTEXT) – The location of signal peptides within the protein. Extracted from uniprot.entry.feature in the UniProt XML file, when the uniprot.entry.feature type attribute is signal peptide. The information is recorded as a 3-tuple, with the elements split by commas. The first element is the position in the protein sequence where the signal peptide starts (the value of the position attribute for uniprot.entry.feature.location.begin), the second is the position in the sequence where the signal peptide ends (the value of the position attribute for uniprot.entry.feature.location.end) and the third is Y or N depending on whether there is any experimental evidence for the signal peptide (the presence or absence of an evidence attribute in the uniprot.entry.feature tag). If there are multiple domains, then multiple 3-tuples are separated by semi-colons (e.g. s1,e1,N;s2,e2,Y,s3,e3,N). If there is no signal peptide information for the protein, then this field will be NA.
48. TransmembraneHelices (LONGTEXT) – The location of alpha-helical transmembrane regions. Extracted from uniprot.entry.feature in the UniProt XML file, when the uniprot.entry.feature type attribute is transmembrane region and the description attribute begins with Helical. The information is recorded as a tuple, with the elements split by commas. The first element is the position in the protein sequence where the transmembrane region starts (the value of the position attribute for uniprot.entry.feature.location.begin) and the second is the position in the sequence where the region ends (the value of the position attribute for uniprot.entry.feature.location.end). If there are multiple transmembrane regions, then multiple tuples are separated by semi-colons (e.g. s1,e1;s2,e2,s3,e3). If there is no transmembrane helix information for the protein, then this field will be NA.
49. Turns (LONGTEXT) – The positions of hydrogen-bonded turns in the protein structure. Corresponds to the DSSP secondary structure code T. Extracted from uniprot.entry.feature in the UniProt XML file, when the type attribute it turn. The information is recorded as a tuple, with the elements split by commas. The first element is the position in the protein sequence where the turn starts (the value of the position attribute for uniprot.entry.feature.location.begin) and the second is the position in the sequence where the turn ends (the value of the position attribute for uniprot.entry.feature.location.end). If there are multiple turns, then multiple tuples are separated by semi-colons (e.g. s1,e1;s2,e2,s3,e3). If there is no turn information for the protein, then this field will be NA.
50. AlphaHelices (LONGTEXT) – The positions of helical regions in the protein structure. Corresponds to the DSSP secondary structure codes G, H and I. Extracted from uniprot.entry.feature in the UniProt XML file, when the type attribute it helix. The information is recorded as a tuple, with the elements split by commas. The first element is the position in the protein sequence where the helix starts (the value of the position attribute for uniprot.entry.feature.location.begin) and the second is the position in the sequence where the helix ends (the value of the position attribute for uniprot.entry.feature.location.end). If there are multiple helices, then multiple tuples are separated by semi-colons (e.g. s1,e1;s2,e2,s3,e3). If there is no alpha helix information for the protein, then this field will be NA.
51. BetaStrands (LONGTEXT) – The positions of beta strands in the protein structure. Corresponds to the DSSP secondary structure codes B and E. Extracted from uniprot.entry.feature in the UniProt XML file, when the type attribute it strand. The information is recorded as a tuple, with the elements split by commas. The first element is the position in the protein sequence where the strand starts (the value of the position attribute for uniprot.entry.feature.location.begin) and the second is the position in the sequence where the strand ends (the value of the position attribute for uniprot.entry.feature.location.end). If there are multiple strands, then multiple tuples are separated by semi-colons (e.g. s1,e1;s2,e2,s3,e3). If there is no beta strand information for the protein, then this field will be NA.
52. PredictedAlphaHelices (LONGTEXT) – Currently not used due to the sequence length restrictions of leading secondary structure prediction methods.
53. PredictedBetaStrands (LONGTEXT) – Currently not used due to the sequence length restrictions of leading secondary structure prediction methods.
54. Isoforms (LONGTEXT) – The alternative protein sequences that can be generated from the same gene. Extracted from uniprot.entry.comment in the UniProt XML file, when the type attribute is alternative products. The information is recorded as a tuple, with the elements split by commas. The first element is the accession of the isoform (extracted from uniprot.entry.comment.isoform.id) and the second is the name of the isoform (extracted from the first instance of uniprot.entry.comment.isoform.name). If there are multiple isoforms, then multiple tuples are separated by semi-colons (e.g. acc1,name1;acc2,name2,acc3,name3). If there is no isoform information for the protein, then this field will be NA.
55. Target (VARCHAR(1)) – Whether or not the protein in question is a drug target. Calculated.
56. Sequence (LONGTEXT) – The amino acid sequence of the protein. Extracted from uniprot.entry.sequence in the UniProt XML file.

### Extra Information

1. The amino acid composition of each protein sequence is calculated without third party software. Amino acids O, U and X are ignored in the calculation, but not removed from the count of the total length of the sequence. For example, a sequence of 200 amino acids would have a length of 200 irrespective of how many O, U and X amino acids it contains. Amino acids B, J and Z, which indicate an inability to determine between two residues, are counted as half an amino acid for each of the two possibilities. For example, if there are ten B amino acids in a sequence, this will be counted as five N and five D. The count of each type of the twenty amino acids is divided by the length of the sequence to get the fraction.
2. The ten categories (e.g. polar, basic, etc.) of amino acid are calculated without third party software. The number of residues that fall into each of the ten categories is summed over the entire length of the sequence, and then divided by the length of the sequence.
3. The number of PEST motifs is calculated using epestfind (<http://emboss.bioinformatics.nl/cgi-bin/emboss/epestfind>). The program is run with the following parameters:

* -auto
* -window 10
* -order score
* -graph none

epestfind returns potential, poor and invalid PEST motifs. Only potential PEST motifs are counted.

1. The number of low complexity regions is calculated using segmasker (included with BLAST+). The program is run with the default algorithm parameters, and default input/output options:

* -infmt fasta
* -outfmt interval

The number of low complexity intervals returned by segmasker is summed to get the total number of low complexity regions.

1. The hydrophobicity of a protein is calculated to be the mean of the hydrophobicity values of all amino acids in the protein sequence. The Kyte and Doolittle index is used for the hydrophobicity values for individual amino acids. The sum of all the hydrophobicity values for the individual residues is divided by the length of the protein sequence. Only the twenty amino acids encoded by the genome are given hydrophobicity values in the Kyte and Doolittle index. Due to this, the O, U and X amino acids are not counted in the summation, but are in the sequence length. Additionally, the B, J and Z amino acids are again counted as half of a D/N, I/L and E/Q respectively.
2. The isoelectric point of each protein is calculated using the pepstats program (<http://emboss.sourceforge.net/apps/cvs/emboss/apps/pepstats.html>). The program is run using the parameters:

* -auto

1. Information about the type of a protein is gathered from four sources. The XML file of the entire human proteome contains information about the protein type in the uniprot.entry.keyword data. UniProt records the controlled vocabulary of keywords that are permissible to use (<http://www.uniprot.org/docs/keywlist>). In the UniProt XML file keywords are recorded along with the accession of the keyword phrase (e.g. KW-0002 for 3D-structure). The accessions of the keywords that correspond to the four protein types of interest are as follows:

* GPCR – KW-0297
* Ion channel – KW-1071, KW-0851, KW-0107, KW-0869, KW-0407, KW-0631, KW-0894
* Kinase – KW-0418, KW-0723, KW-0829
* Protease – KW-0031, KW-0064, KW-0121, KW-0224, KW-0482, KW-0645, KW-0720, KW-0788, KW-0888

Three additional files are used to help determine GPCRs, kinases and proteases. A file containing the human GPCRs in UniProt can also be accessed from: <http://www.uniprot.org/docs/7tmrlist>. A file containing the human kinases in UniProt can also be accessed from: <http://www.uniprot.org/docs/pkinfam>. A file containing the human proteases in UniProt can also be accessed from: <http://www.uniprot.org/docs/peptidas>. These three files are parsed to determine if there are any GPCRs, kinases or proteases not found in the XML file, that are in fact GPCRs, kinases or proteases. There is no additional source of ion channel proteins.

1. For the information about topological domain, signal peptides, transmembrane helices and secondary structure it is possible that there will be no information about the start position of the feature, the end position or either position. In this case the position is recorded as an empty string. This means the data for these columns will look more like s1,’’,X1;s2,e2,X2;’’,’’,X3 for the topological domain, signal peptides and transmembrane helices, and like s1,’’;’’,e2;,’’,’’;,s4,e4 for the secondary structure. It is also possible that there will be no information for the Xs, i.e. no domain, peptide or helix information. In these cases the Xs will also be replaced by empty strings.
2. The subcellular localisation predictions are performed using WoLFPSORT (<http://wolfpsort.org/>, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1933216/>). This method requires a minimum sequence length of thirty amino acids, therefore any proteins with fewer than thirty amino acids have no predictions.
3. The targets are calculated using data from UniProt, DrugBank, the Therapeutic Target Database (TTD) and ChEMBL. The only drugs considered are small molecule drugs, biotech and nutraceutical are not counted. First the target-drug relationships extracted from UniProt are cross-referenced with DrugBank. This serves to determine which target-drug relationships actually involve an approved drug target. Any proteins which are involved in a target-drug relationship with a valid drug are recorded as approved drug targets. The second step is to determine which approved targets from DrugBank are: (1) the target of an approved small molecule drug and (2) human proteins. It is necessary to check if the DrugBank approved targets are the target of an approved small molecule drug as otherwise a target which is only targeted by approved biotech drugs could be included. Whether the target is a human protein is checked by first turning the DrugBank target IDs into UniProt accessions, this is done using the DrugBank ‘Links to external databases and external identifiers for drug targets’ file (found at <http://drugbank.ca/system/downloads/current/target_links.csv.zip>). These UniProt accessions may not be current accessions, e.g. an old accession that has been merged into a new one, so they are converted to current UniProt accessions. This will lose some approved targets from DrugBank, either because there is no UniProt accession recorded for the target or because the accession cannot be mapped to a current human one. This may indicate that the target is not a human protein, or that it simply has a bad accession. The next step is to cross-reference the TTD with UniProt. Every target of an approved drug is extracted from the TTD along with the UniProt accessions it maps to, which may be more than one. For each target extracted in this manner, the UniProt accessions that it maps to are checked to see if they are or can be converted to current accessions. Next ChEMBL is cross-referenced with UniProt. The UniProt accessions of targets of approved drugs are extracted from ChEMBL. The targets are found by linking approved drugs to targets using assay information. A target is only accepted when the assay2target confidence score is at least 4. This means that the assay has been assigned a protein target (<https://www.ebi.ac.uk/chembldb/index.php/faq#faq4>) These accessions are checked to see if they are or can be converted to current accessions. As with DrugBank and the TTD, only those accessions which can be converted to current accessions are kept. Once the current accessions of the approved drug targets from UniProt, DrugBank, the TTD and ChEMBL have been determined, the accessions are combined into one set, and duplicates removed. This leaves one large set with all the approved targets that could be adequately cross-referenced with UniProt.

## stability

1. UPAccession (VARCHAR(10)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. HalfLife (FLOAT) – The estimation of the time it takes *in vivo* for half of the amount of protein in a cell to disappear after its synthesis. Calculated using the table from <http://www.ncbi.nlm.nih.gov/pubmed/2506181> (Universality and Structure of the N-end Rule). The proline half life was altered from >20 hours to 20 hours in order to ensure all values were floats. Additionally, the N-terminus of a few of the proteins is not one of the amino acids described in the paper (e.g. B, J, O, U, X and Z). In this case the protein half life is recorded as -1.
3. InstabilityIndex (FLOAT) – An estimation of the *in vitro* stability of the protein. An instability index value of less than 40 is predicted as stable. Calculated using the BioPython implementation of the ExPasy ProtParam tool (<http://biopython.org/wiki/ProtParam>). For details of the calculation method see the ExPasy documentation (<http://web.expasy.org/protparam/protparam-doc.html>) and the paper it is based on (<http://www.ncbi.nlm.nih.gov/pubmed/2075190?dopt=Abstract>). When the protein contains one of the amino acids not covered by this publication (e.g. B, J, O, U, X and Z) the instability index can not be calculated using BioPython. In these cases the instability index is calculated using the ProtParam tool on the ExPASy servers (<http://web.expasy.org/protparam/>).

## unigene

1. UniGeneID (INT) – The ID of the UniGene cluster.
2. DS\_Embryoid\_Body (INT) – The number of ESTs in the cluster from embryoid body sources.
3. DS\_Blastocyst (INT) – The number of ESTs in the cluster from blastocyst sources.
4. DS\_Fetus (INT) – The number of ESTs in the cluster from fetal sources.
5. DS\_Neonate (INT) – The number of ESTs in the cluster from neonatal sources.
6. DS\_Infant (INT) – The number of ESTs in the cluster from infant sources.
7. DS\_Juvenile (INT) – The number of ESTs in the cluster from juvenile sources.
8. DS\_Adult (INT) – The number of ESTs in the cluster from adult sources.
9. HS\_Adrenal\_Tumor (INT) – The number of ESTs in the cluster from adrenal tumor sources.
10. HS\_Bladder\_Carcinoma (INT) – The number of ESTs in the cluster from bladder carcinoma sources.
11. HS\_Breast\_Mammary\_Gland\_Tumor (INT) – The number of ESTs in the cluster from breast mammary gland tumor sources.
12. HS\_Cervical\_Tumor (INT) – The number of ESTs in the cluster from cervical tumor sources.
13. HS\_Chondrosarcoma (INT) – The number of ESTs in the cluster from chondrosarcoma sources.
14. HS\_Colorectal\_Tumor (INT) – The number of ESTs in the cluster from colorectal tumor sources.
15. HS\_Esophageal\_Tumor (INT) – The number of ESTs in the cluster from esophageal tumor sources.
16. HS\_Gastrointestinal\_Tumor (INT) – The number of ESTs in the cluster from gastrointestinal tumor sources.
17. HS\_Germ\_Cell\_Tumor (INT) – The number of ESTs in the cluster from germ cell tumor sources.
18. HS\_Glioma (INT) – The number of ESTs in the cluster from glioma sources.
19. HS\_Head\_And\_Neck\_Tumor (INT) – The number of ESTs in the cluster from head and neck tumor sources.
20. HS\_Kidney\_Tumor (INT) – The number of ESTs in the cluster from kidney tumor sources.
21. HS\_Leukemia\_Tumor (INT) – The number of ESTs in the cluster from leukemia tumor sources.
22. HS\_Liver\_Tumor (INT) – The number of ESTs in the cluster from liver tumor sources.
23. HS\_Lung\_Tumor (INT) – The number of ESTs in the cluster from lung tumor sources.
24. HS\_Lymphoma (INT) – The number of ESTs in the cluster from lymphoma sources.
25. HS\_Non-neoplasia (INT) – The number of ESTs in the cluster from neo-neoplasia sources.
26. HS\_Normal (INT) – The number of ESTs in the cluster from normal sources.
27. HS\_Ovarian\_Tumor (INT) – The number of ESTs in the cluster from ovarian tumor sources.
28. HS\_Pancreatic\_Tumor (INT) – The number of ESTs in the cluster from pancreatic tumor sources.
29. HS\_Primitive\_Neuroectodermal\_Tumor\_Of\_The\_CNS (INT) – The number of ESTs in the cluster from primitive neuroectodermal tumor of the CNS sources.
30. HS\_Prostate\_Cancer (INT) – The number of ESTs in the cluster from prostate cancer sources.
31. HS\_Retinoblastoma (INT) – The number of ESTs in the cluster from retinoblastoma sources.
32. HS\_Skin\_Tumor (INT) – The number of ESTs in the cluster from skin tumor sources.
33. HS\_Soft\_Tissue/Muscle\_Tissue\_Tumor (INT) – The number of ESTs in the cluster from soft tissure/muscle tissue sources.
34. HS\_Uterine\_Tumor (INT) – The number of ESTs in the cluster from uterine tumor sources.
35. BS\_Adipose\_Tissue (INT) – The number of ESTs in the cluster from adipose tissue sources.
36. BS\_Adrenal\_Gland (INT) – The number of ESTs in the cluster from adrenal gland sources.
37. BS\_Ascites (INT) – The number of ESTs in the cluster from ascites sources.
38. BS\_Bladder (INT) – The number of ESTs in the cluster from bladder sources.
39. BS\_Blood (INT) – The number of ESTs in the cluster from blood sources.
40. BS\_Bone (INT) – The number of ESTs in the cluster from bone sources.
41. BS\_Bone\_Marrow (INT) – The number of ESTs in the cluster from bone marrow sources.
42. BS\_Brain (INT) – The number of ESTs in the cluster from brain sources.
43. BS\_Cervix (INT) – The number of ESTs in the cluster from cervix sources.
44. BS\_Connective\_Tissue (INT) – The number of ESTs in the cluster from connective tissue sources.
45. BS\_Ear (INT) – The number of ESTs in the cluster from ear sources.
46. BS\_Embryonic\_Tissue (INT) – The number of ESTs in the cluster from embryonic tissue sources.
47. BS\_Esophagus (INT) – The number of ESTs in the cluster from esophagus sources.
48. BS\_Eye (INT) – The number of ESTs in the cluster from eye sources.
49. BS\_Heart (INT) – The number of ESTs in the cluster from heart sources.
50. BS\_Intestine (INT) – The number of ESTs in the cluster from intestine sources.
51. BS\_Kidney (INT) – The number of ESTs in the cluster from kidney sources.
52. BS\_Larynx (INT) – The number of ESTs in the cluster from larynx sources.
53. BS\_Liver (INT) – The number of ESTs in the cluster from liver sources.
54. BS\_Lung (INT) – The number of ESTs in the cluster from lung sources.
55. BS\_Lymph (INT) – The number of ESTs in the cluster from lymph sources.
56. BS\_Lymph\_Node (INT) – The number of ESTs in the cluster from lymph node sources.
57. BS\_Mammary\_Gland (INT) – The number of ESTs in the cluster from mammary gland sources.
58. BS\_Mouth (INT) – The number of ESTs in the cluster from mouth sources.
59. BS\_Muscle (INT) – The number of ESTs in the cluster from muscle sources.
60. BS\_Nerve (INT) – The number of ESTs in the cluster from nerve sources.
61. BS\_Ovary (INT) – The number of ESTs in the cluster from ovary sources.
62. BS\_Pancreas (INT) – The number of ESTs in the cluster from pancreas sources.
63. BS\_Parathyroid (INT) – The number of ESTs in the cluster from parathyroid sources.
64. BS\_Pharynx (INT) – The number of ESTs in the cluster from pharynx sources.
65. BS\_Pituitary\_Gland (INT) – The number of ESTs in the cluster from pituitary gland sources.
66. BS\_Placenta (INT) – The number of ESTs in the cluster from placenta sources.
67. BS\_Prostate (INT) – The number of ESTs in the cluster from prostate sources.
68. BS\_Salivary\_Gland (INT) – The number of ESTs in the cluster from salivary gland sources.
69. BS\_Skin (INT) – The number of ESTs in the cluster from skin sources.
70. BS\_Spleen (INT) – The number of ESTs in the cluster from spleen sources.
71. BS\_Stomach (INT) – The number of ESTs in the cluster from stomach sources.
72. BS\_Testis (INT) – The number of ESTs in the cluster from testis sources.
73. BS\_Thymus (INT) – The number of ESTs in the cluster from thymus sources.
74. BS\_Thyroid (INT) – The number of ESTs in the cluster from thyroid sources.
75. BS\_Tonsil (INT) – The number of ESTs in the cluster from tonsil sources.
76. BS\_Trachea (INT) – The number of ESTs in the cluster from trachea sources.
77. BS\_Umbilical\_Cord (INT) – The number of ESTs in the cluster from umbilical cord sources.
78. BS\_Uterus (INT) – The number of ESTs in the cluster from uterus sources.
79. BS\_Vascular (INT) – The number of ESTs in the cluster from vascular sources.

## unigenetotals

1. StageStateSite (VARCHAR(100)) – The name of the column in proteininfo.unigene to which the total corresponds.
2. Total (INT) – The total number of ESTs recorded for the developmental stage, health state or body site.

## uniprot2ensembl

1. UPAccession (VARCHAR(45)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. EnsemblGeneID (VARCHAR(45)) – The Ensembl gene ID for the Ensembl gene which corresponds to the UniProt protein. Foreign key from proteininfo.ensemblgene.
3. EnsemblTranscriptID (VARCHAR(45)) – The Ensembl transcript ID for the Ensembl transcript which corresponds to the UniProt protein.
4. EnsemblProteinID (VARCHAR(45)) – The Ensembl protein ID for the Ensembl protein which corresponds to the UniProt protein.

## uniprote2entrez

1. UPAccession (VARCHAR(45)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. GeneID (VARCHAR(45)) – The ID of the Entrez gene which corresponds to the UniProt protein. Foreign key from proteininfo.entrezgene.

## uniprot2go

1. UPAccession (VARCHAR(45)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. GOTermID (INT) – The GO term ID which corresponds to the UniProt protein. Foreign key from proteininfo.goinfo.

## uniprot2unigene

1. UPAccession (VARCHAR(45)) – The UniProt accession for the protein. Foreign key from proteininfo.UPAccession.
2. UniGeneID (INT) – The ID of the UniGene cluster that the protein is a member of. Foreign key from proteininfo.unigene.

# Brief Description of Each View

# View Columns, and Where the Data Comes From

There are three primary types of view used. The first is a simple mapping between UniProt accessions and attributes from the spoke tables (e.g. PPIs, variants, etc.). All these views begin with upacc\_. The second is a mapping between UniProt accessions and their sequences. These views are used to generate FASTA files for redundancy removal, and are all subsets of the entire human proteome. For example, there are subsets based on type of protein or involvement in a particular disease. All these views end in r\_n or r\_p. The final type of view is a collection of all the data in the database about individual proteins. The proteins in the view are all the non-redundant proteins for a particular dataset. For example, all the data about non-redundant drug target GPCRs. All these views end in nr\_n or nr\_p.

The redundant and non-redundant dataset views have a well defined format for naming. The format is A\_B\_C\_D\_E, where each element has the following significance:

1. The subdivision type. Two possibilities are ‘type’ and ‘ill’. If A is ‘type’, then this indicates that the dataset is a subdivision of the proteome based on the type of the protein (e.g. GPCR, kinase, etc.). If A is ‘ill’, then this indicates that the dataset is a subdivision of the proteome based on whether or not the proteins are involved in a certain disease.
2. The identifier of the specific subdivision. Two examples of this element would be ‘cancer’ and ‘gpcr’. If B is ‘cancer’ (A would be ‘ill’), then the subdivision is based on proteins involved in cancer. If B is ‘gpcr’, then the subdivision is based on proteins that are GPCRs.
3. The type of comparison that is being made. The four possibilities are ‘ctncnt’, ‘targ’, ‘type’ and ‘prot’. If C is ‘ctncnt’ (criterion targets non-criterion non-targets), then the comparison is being performed between targets in the B subdivision and non-targets not in the B subdivision (e.g. cancer targets and non-cancer non-targets). If C is ‘targ’, then the comparison being performed is between target and non-target proteins in the B subdivision. If C is ‘type’, then the comparison being performed is between targets in the B subdivision and targets not in the B subdivision. If C is ‘prot’, then the comparison is between proteins in the B subdivision and proteins not in the B subdivision.
4. Whether the view contains redundant or non-redundant data points. D is ‘r’ if the data in the view is redundant, and ‘nr’ if it is non-redundant.
5. Whether the view contains the positive or negative data points. Each testing/training dataset is composed of two views. One view consists of the positive data points, and one consists of the negative data points. For example, if you are comparing GPCR targets with GPCR non-targets (A=type, B=gpcr and C=targ), then one view will contain all GPCR targets, and another view will contain all GPCR non-targets.

## Accession to Attribute Mappings

### upacc\_cancer

1. UPAccession – The UniProt accession from proteininfo.UPAccession.

### upacc\_expression

1. UPAccession – The UniProt accession from proteininfo.UPAccession.
2. DS\_Embryoid\_Body – The sum of the number of ESTs expressed in the embryoid body developmental stage, in UniGene clusters linked to the UniProt accession.
3. DS\_Blastocyst – The sum of the number of ESTs expressed in the blastocyst developmental stage, in UniGene clusters linked to the UniProt accession.
4. DS\_Fetus – The sum of the number of ESTs expressed in the fetus developmental stage, in UniGene clusters linked to the UniProt accession.
5. DS\_Neonate – The sum of the number of ESTs expressed in the neonate developmental stage, in UniGene clusters linked to the UniProt accession.
6. DS\_Infant – The sum of the number of ESTs expressed in the infant developmental stage, in UniGene clusters linked to the UniProt accession.
7. DS\_Juvenile – The sum of the number of ESTs expressed in the juvenile developmental stage, in UniGene clusters linked to the UniProt accession.
8. DS\_Adult – The sum of the number of ESTs expressed in the adult developmental stage, in UniGene clusters linked to the UniProt accession.
9. HS\_Adrenal\_Tumor – The sum of the number of ESTs expressed in the adrenal tumour health state, in UniGene clusters linked to the UniProt accession.
10. HS\_Bladder\_Carcinoma – The sum of the number of ESTs expressed in the bladder carcinoma health state, in UniGene clusters linked to the UniProt accession.
11. HS\_Breast\_Mammary\_Gland\_Tumor – The sum of the number of ESTs expressed in the breast and mammary gland tumour health state, in UniGene clusters linked to the UniProt accession.
12. HS\_Cervical\_Tumor – The sum of the number of ESTs expressed in the cervical tumour health state, in UniGene clusters linked to the UniProt accession.
13. HS\_Chondrosarcoma – The sum of the number of ESTs expressed in the chondrosarcoma health state, in UniGene clusters linked to the UniProt accession.
14. HS\_Colorectal\_Tumor – The sum of the number of ESTs expressed in the colorectal tumour health state, in UniGene clusters linked to the UniProt accession.
15. HS\_Esophageal\_Tumor – The sum of the number of ESTs expressed in the oesophageal tumour health state, in UniGene clusters linked to the UniProt accession.
16. HS\_Gastrointestinal\_Tumor – The sum of the number of ESTs expressed in the gastrointestinal tumour health state, in UniGene clusters linked to the UniProt accession.
17. HS\_Germ\_Cell\_Tumor – The sum of the number of ESTs expressed in the germ cell tumour health state, in UniGene clusters linked to the UniProt accession.
18. HS\_Glioma – The sum of the number of ESTs expressed in the glioma health state, in UniGene clusters linked to the UniProt accession.
19. HS\_Head\_And\_Neck\_Tumor – The sum of the number of ESTs expressed in the head and neck tumour health state, in UniGene clusters linked to the UniProt accession.
20. HS\_Kidney\_Tumor – The sum of the number of ESTs expressed in the kidney tumour health state, in UniGene clusters linked to the UniProt accession.
21. HS\_Leukemia\_Tumor – The sum of the number of ESTs expressed in the leukemia tumour health state, in UniGene clusters linked to the UniProt accession.
22. HS\_Liver\_Tumor – The sum of the number of ESTs expressed in the liver tumour health state, in UniGene clusters linked to the UniProt accession.
23. HS\_Lung\_Tumor – The sum of the number of ESTs expressed in the lung tumour health state, in UniGene clusters linked to the UniProt accession.
24. HS\_Lymphoma – The sum of the number of ESTs expressed in the lymphoma health state, in UniGene clusters linked to the UniProt accession.
25. HS\_Non\_neoplasia – The sum of the number of ESTs expressed in the non-neoplasia health state, in UniGene clusters linked to the UniProt accession.
26. HS\_Normal – The sum of the number of ESTs expressed in the normal health state, in UniGene clusters linked to the UniProt accession.
27. HS\_Ovarian\_Tumor – The sum of the number of ESTs expressed in the ovarian tumour health state, in UniGene clusters linked to the UniProt accession.
28. HS\_Pancreatic\_Tumor – The sum of the number of ESTs expressed in the pancreatic tumour health state, in UniGene clusters linked to the UniProt accession.
29. HS\_Primitive\_Neuroectodermal\_Tumor\_Of\_The\_CNS – The sum of the number of ESTs expressed in the primitive neuroectodermal tumour of the CNS health state, in UniGene clusters linked to the UniProt accession.
30. HS\_Prostate\_Cancer – The sum of the number of ESTs expressed in the prostate cancer health state, in UniGene clusters linked to the UniProt accession.
31. HS\_Retinoblastoma – The sum of the number of ESTs expressed in the retinoblastoma health state, in UniGene clusters linked to the UniProt accession.
32. HS\_Skin\_Tumor – The sum of the number of ESTs expressed in the skin tumour health state, in UniGene clusters linked to the UniProt accession.
33. HS\_Soft\_Tissue\_Muscle\_Tissue\_Tumor – The sum of the number of ESTs expressed in the soft tissue and muscle tissue tumour health state, in UniGene clusters linked to the UniProt accession.
34. HS\_Uterine\_Tumor – The sum of the number of ESTs expressed in the uterine tumour health state, in UniGene clusters linked to the UniProt accession.
35. BS\_Adipose\_Tissue – The sum of the number of ESTs expressed in the adipose tissue body site, in UniGene clusters linked to the UniProt accession.
36. BS\_Adrenal\_Gland – The sum of the number of ESTs expressed in the adrenal gland body site, in UniGene clusters linked to the UniProt accession.
37. BS\_Ascites – The sum of the number of ESTs expressed in the ascites body site, in UniGene clusters linked to the UniProt accession.
38. BS\_Bladder – The sum of the number of ESTs expressed in the bladder body site, in UniGene clusters linked to the UniProt accession.
39. BS\_Blood – The sum of the number of ESTs expressed in the blood body site, in UniGene clusters linked to the UniProt accession.
40. BS\_Bone – The sum of the number of ESTs expressed in the bone body site, in UniGene clusters linked to the UniProt accession.
41. BS\_Bone\_Marrow – The sum of the number of ESTs expressed in the bone marrow body site, in UniGene clusters linked to the UniProt accession.
42. BS\_Brain – The sum of the number of ESTs expressed in the brain body site, in UniGene clusters linked to the UniProt accession.
43. BS\_Cervix – The sum of the number of ESTs expressed in the cervix body site, in UniGene clusters linked to the UniProt accession.
44. BS\_Connective\_Tissue – The sum of the number of ESTs expressed in the connective tissue body site, in UniGene clusters linked to the UniProt accession.
45. BS\_Ear – The sum of the number of ESTs expressed in the ear body site, in UniGene clusters linked to the UniProt accession.
46. BS\_Embryonic\_Tissue – The sum of the number of ESTs expressed in the embryonic tissue body site, in UniGene clusters linked to the UniProt accession.
47. BS\_Esophagus – The sum of the number of ESTs expressed in the oesophagus body site, in UniGene clusters linked to the UniProt accession.
48. BS\_Eye – The sum of the number of ESTs expressed in the eye body site, in UniGene clusters linked to the UniProt accession.
49. BS\_Heart – The sum of the number of ESTs expressed in the heart body site, in UniGene clusters linked to the UniProt accession.
50. BS\_Intestine – The sum of the number of ESTs expressed in the intestine body site, in UniGene clusters linked to the UniProt accession.
51. BS\_Kidney – The sum of the number of ESTs expressed in the kidney body site, in UniGene clusters linked to the UniProt accession.
52. BS\_Larynx – The sum of the number of ESTs expressed in the larynx body site, in UniGene clusters linked to the UniProt accession.
53. BS\_Liver – The sum of the number of ESTs expressed in the liver body site, in UniGene clusters linked to the UniProt accession.
54. BS\_Lung – The sum of the number of ESTs expressed in the lung body site, in UniGene clusters linked to the UniProt accession.
55. BS\_Lymph – The sum of the number of ESTs expressed in the lymph body site, in UniGene clusters linked to the UniProt accession.
56. BS\_Lymph\_Node – The sum of the number of ESTs expressed in the lymph node body site, in UniGene clusters linked to the UniProt accession.
57. BS\_Mammary\_Gland – The sum of the number of ESTs expressed in the mammary gland body site, in UniGene clusters linked to the UniProt accession.
58. BS\_Mouth – The sum of the number of ESTs expressed in the mouth body site, in UniGene clusters linked to the UniProt accession.
59. BS\_Muscle – The sum of the number of ESTs expressed in the muscle body site, in UniGene clusters linked to the UniProt accession.
60. BS\_Nerve – The sum of the number of ESTs expressed in the nerve body site, in UniGene clusters linked to the UniProt accession.
61. BS\_Ovary – The sum of the number of ESTs expressed in the ovary body site, in UniGene clusters linked to the UniProt accession.
62. BS\_Pancreas – The sum of the number of ESTs expressed in the pancreas body site, in UniGene clusters linked to the UniProt accession.
63. BS\_Parathyroid – The sum of the number of ESTs expressed in the parathyroid body site, in UniGene clusters linked to the UniProt accession.
64. BS\_Pharynx – The sum of the number of ESTs expressed in the pharynx body site, in UniGene clusters linked to the UniProt accession.
65. BS\_Pituitary\_Gland – The sum of the number of ESTs expressed in the pituitary gland body site, in UniGene clusters linked to the UniProt accession.
66. BS\_Placenta – The sum of the number of ESTs expressed in the placenta body site, in UniGene clusters linked to the UniProt accession.
67. BS\_Prostate – The sum of the number of ESTs expressed in the prostate body site, in UniGene clusters linked to the UniProt accession.
68. BS\_Salivary\_Gland – The sum of the number of ESTs expressed in the salivary gland body site, in UniGene clusters linked to the UniProt accession.
69. BS\_Skin – The sum of the number of ESTs expressed in the skin body site, in UniGene clusters linked to the UniProt accession.
70. BS\_Spleen – The sum of the number of ESTs expressed in the spleen body site, in UniGene clusters linked to the UniProt accession.
71. BS\_Stomach – The sum of the number of ESTs expressed in the stomach body site, in UniGene clusters linked to the UniProt accession.
72. BS\_Testis – The sum of the number of ESTs expressed in the testis body site, in UniGene clusters linked to the UniProt accession.
73. BS\_Thymus – The sum of the number of ESTs expressed in the thymus body site, in UniGene clusters linked to the UniProt accession.
74. BS\_Thyroid – The sum of the number of ESTs expressed in the thyroid body site, in UniGene clusters linked to the UniProt accession.
75. BS\_Tonsil – The sum of the number of ESTs expressed in the tonsil body site, in UniGene clusters linked to the UniProt accession.
76. BS\_Trachea – The sum of the number of ESTs expressed in the trachea body site, in UniGene clusters linked to the UniProt accession.
77. BS\_Umbilical\_Cord – The sum of the number of ESTs expressed in the umbilical cord body site, in UniGene clusters linked to the UniProt accession.
78. BS\_Uterus – The sum of the number of ESTs expressed in the uterus body site, in UniGene clusters linked to the UniProt accession.
79. BS\_Vascular – The sum of the number of ESTs expressed in the vascular body site, in UniGene clusters linked to the UniProt accession.

A small sample of the SQL code to generate this view can be seen below. The sample covers only the embryoid body developmental stage.

CREATE VIEW `proteindatabase`.`upacc\_expression` AS

SELECT

prot.UPAccession,

IFNULL((SELECT SUM(unigene.DS\_Embryoid\_Body) FROM uniprot2unigene AS u2u, unigene WHERE prot.UPAccession = u2u.UPAccession AND u2u.UniGeneID = unigene.UniGeneID), 0) AS DS\_Embryoid\_Body,

...

FROM

proteininfo AS prot

GROUP BY

prot.UPAccession

### upacc\_genetrans

1. UPAccession – The UniProt accession from uniprot2ensembl.UPAccession.
2. EnsemblGeneID – The Ensembl gene linked to the UniProt accession. Extracted from ensemblgene.EnsemblGeneID.
3. EnsemblTranscriptID – The ID of a transcript linked to the Ensembl gene . Extracted from uniprot2ensembl.EnsemblTranscriptID.
4. ProteinCodingTranscripts – The number of protein coding transcripts linked to the Ensembl gene. Extracted from ensemblgene.ProteinCodingTranscripts.

The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`upacc\_genetrans` AS

SELECT

UPAccession,

ensemblgene.EnsemblGeneID,

uniprot2ensembl.EnsemblTranscriptID,

ensemblgene.ProteinCodingTranscripts

FROM

uniprot2ensembl

JOIN ensemblgene

ON uniprot2ensembl.EnsemblGeneID = ensemblgene.EnsemblGeneID

Each (UPAccession, EnsemblGeneID) pair may appear in the view multiple times. This is because an individual UniProt protein might be linked to multiple transcripts from a gene. It is also possible for an individual UniProt protein to be linked to more than one Ensembl gene. There is no expectation/requirement for the number of protein coding transcripts to be equal to the number of times an individual (UPAccession, EnsemblGeneID) pair appears in the view.

### upacc\_germvariants

1. UPAccession – The UniProt accession from upacc\_transcripts.UPAccession.
2. VariantID – The ID of the variant (generally a dbSNP ID), extracted from germvariants.VariantID.
3. 3Untranslated – 1 if the variant affects 3 untranslated region of the mRNA transcribed from the gene that codes for the UPAccession protein. Extracted from germvariants.3Untranslated.
4. 5Untranslated – 1 if the variant affects the 5 untranslated region of the mRNA transcribed from the gene that codes for the UPAccession protein. Extracted from germvariants.5Untranslated.
5. NonSynonymousCoding – 1 if the variant causes a non-synonymous alteration to the coding region of the gene that codes for the UPAccession protein. Extracted from germvariants.NonSynonymousCoding.
6. SynonymousCoding – 1 if the variant causes a synonymous alteration to the coding region of the gene that codes for the UPAccession protein. Extracted from germvariants.SynonymousCoding.

The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`upacc\_germvariants` AS

SELECT

DISTINCT

trans.UPAccession,

germ.VariantID,

germ.3Untranslated,

germ.5Untranslated,

germ.NonSynonymousCoding,

germ.SynonymousCoding

FROM

upacc\_transcripts AS trans,

germvariants as germ

WHERE

trans.EnsemblTranscriptID = germ.EnsemblTranscriptID

### upacc\_paralogs

1. UPAccession – The UniProt accession from upacc\_ transcripts.UPAccession.
2. Paralogs – The number of paralogs the protein has.

The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`upacc\_paralogs` AS

SELECT

trans.UPAccession,

IFNULL((SELECT COUNT(DISTINCT hom.HomologGene) FROM homologs AS hom WHERE trans.EnsemblGeneID = hom.HumanGene AND hom.HomologyType = "within\_species\_paralog"), 0) AS Paralogs

FROM

upacc\_transcripts AS trans

GROUP BY

trans.UPAccession

### upacc\_ppi

1. UPAccession – The UniProt accession from proteininfo.UPAccession.
2. BinaryPPI – The number of binary protein-protein interactions that the protein participates in.

The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`upacc\_ppi` AS

SELECT

prot.UPAccession,

IFNULL((SELECT COUNT(\*) FROM ppi WHERE prot.UPAccession = ppi.PPIProteinOne AND ppi.OrganismsDiffer = 'false'), 0) As BinaryPPI

FROM

proteininfo AS prot

GROUP BY

prot.UPAccession

### upacc\_transcripts

1. UPAccession – The UniProt accession from upacc\_genetrans.UPAccession.
2. EnsemblGeneID – The Ensembl gene linked to the UniProt accession, from upacc\_genetrans.EnsemblGeneID.
3. EnsemblTranscriptID – The ID of a transcript linked to the Ensembl gene upacc\_genetrans.EnsemblTranscriptID.
4. ProteinCodingTranscripts – The number of protein coding transcripts linked to the Ensembl gene upacc\_genetrans.ProteinCodingTranscripts.

The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`upacc\_transcripts` AS

SELECT

maxi.\*

FROM

upacc\_genetrans AS maxi

LEFT OUTER JOIN

upacc\_genetrans AS checker

ON

checker.UPAccession = maxi.UPAccession AND

checker.ProteinCodingTranscripts > maxi.ProteinCodingTranscripts

WHERE

checker.ProteinCodingTranscripts IS NULL

Each (UPAccession, EnsemblGeneID) pair may appear in the view multiple times. This is because an individual UniProt protein might be linked to multiple transcripts from a gene. Unlike upacc\_genetrans, it is not possible for a UniProt accession to be linked to more than one Ensembl gene in this view. If there is more than one gene linked to the UniProt accession, then the gene with the most protein coding transcripts appears in this view. If more than one gene has the largest number of protein coding transcripts, then the choice of which (UPAccession, EnsemblGeneID) pair appears in the view is arbitrary.

### upacc\_variantsnumber

1. UPAccession – The UniProt accession from upacc\_transcripts.UPAccession.
2. 3Untranslated – The number of genetic variants linked to the protein that affect the 3’ untranslated region of the mRNA transcribed from the gene that codes for the UPAccession protein.
3. 5Untranslated – The number of genetic variants linked to the protein that affect the 5’ untranslated region of the mRNA transcribed from the gene that codes for the UPAccession protein.
4. NonSynonymousCoding – The number of genetic variants that cause non-synonymous changes to the coding region of the gene that codes for the UPAccession protein.
5. SynonymousCoding – The number of genetic variants that cause synonymous changes to the coding region of the gene that codes for the UPAccession protein.
6. Total – The total number of genetic variants of the protein that fall into the 3’ untranslated, 5’ untranslated, non-synonymous coding or synonymous coding categories.

The SQL code for generating the view is as follows:

CREATE VIEW `proteindatabase`.`upacc\_variantsnumber` AS

SELECT

trans.UPAccession,

IFNULL((SELECT SUM(3Untranslated) FROM upacc\_germvariants AS germ WHERE trans.UPAccession = germ.UPAccession), 0) AS 3Untranslated,

IFNULL((SELECT SUM(5Untranslated) FROM upacc\_germvariants AS germ WHERE trans.UPAccession = germ.UPAccession), 0) AS 5Untranslated,

IFNULL((SELECT SUM(NonSynonymousCoding) FROM upacc\_germvariants AS germ WHERE trans.UPAccession = germ.UPAccession), 0) AS NonSynonymousCoding,

IFNULL((SELECT SUM(SynonymousCoding) FROM upacc\_germvariants AS germ WHERE trans.UPAccession = germ.UPAccession), 0) AS SynonymousCoding,

IFNULL((SELECT SUM(3Untranslated) + SUM(5Untranslated) + SUM(NonSynonymousCoding) + SUM(SynonymousCoding) FROM upacc\_germvariants AS germ WHERE trans.UPAccession = germ.UPAccession), 0) AS Total

FROM

upacc\_transcripts AS trans

GROUP BY

trans.UPAccession

## Redundant Dataset Views

All of these views contain two columns, UPAccession and Sequence. The UPAccession column is the UniProt accession for the protein in question, and the Sequence is the protein’s amino acid sequence. As the columns are identical for each of the views, only the name of each view, the SQL code used to generate the view and an explanation of the name will be provided.

### all\_all\_targ\_r\_n

This view is used for the comparison of all the target proteins and all the non-target proteins. It contains the redundant non-target proteins. The SQL code to generate the view is as follows:

CREATE VIEW `proteindatabase`.`all\_all\_targ\_r\_n` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "N"

### all\_all\_targ\_r\_p

This view is used for the comparison of all the target proteins and all the non-target proteins. It contains the redundant target proteins. The SQL code to generate the view is as follows:

CREATE VIEW `proteindatabase`.`all\_all\_targ\_r\_p` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "Y"

### ill\_cancer\_prot\_r\_n

This view is used for the comparison of all the cancer proteins and all the non-cancer proteins. It contains the redundant non-cancer proteins. The SQL code to generate the view is as follows:

CREATE VIEW `proteindatabase`.`ill\_cancer\_prot\_r\_n` AS

SELECT

prot.`UPAccession`, prot.`Sequence`

FROM

`proteindatabase`.`proteininfo` AS prot LEFT JOIN `proteindatabase`.`upacc\_cancer` AS cancer

ON

prot.`UPAccession` = cancer.`UPAccession`

WHERE

cancer.`UPAccession` IS NULL

### ill\_cancer\_prot\_r\_p

This view is used for the comparison of all the cancer proteins and all the non-cancer proteins. It contains the redundant cancer proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`ill\_cancer\_prot\_r\_p` AS

SELECT

`proteininfo`.`UPAccession`, `Sequence`

FROM

`proteindatabase`.`proteininfo`

INNER JOIN

`proteindatabase`.`upacc\_cancer`

ON `proteininfo`.`UPAccession` = `upacc\_cancer`.`UPAccession`

### ill\_cancer\_targ\_r\_n

This view is used for the comparison of all the cancer target proteins and all the cancer non-target proteins. It contains the redundant cancer non-target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`ill\_cancer\_targ\_r\_n` AS

SELECT

`proteininfo`.`UPAccession`, `Sequence`

FROM

`proteindatabase`.`proteininfo`

INNER JOIN

`proteindatabase`.`upacc\_cancer`

ON `proteininfo`.`UPAccession` = `upacc\_cancer`.`UPAccession`

WHERE

`proteininfo`.`Target` = "N"

### ill\_cancer\_targ\_r\_p

This view is used for the comparison of all the cancer target proteins and all the cancer non-target proteins. It contains the redundant cancer target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`ill\_cancer\_targ\_r\_p` AS

SELECT

`proteininfo`.`UPAccession`, `Sequence`

FROM

`proteindatabase`.`proteininfo`

INNER JOIN

`proteindatabase`.`upacc\_cancer`

ON `proteininfo`.`UPAccession` = `upacc\_cancer`.`UPAccession`

WHERE

`proteininfo`.`Target` = "Y"

### ill\_cancer\_type\_r\_n

This view is used for the comparison of all the cancer target proteins and all the non-cancer target proteins. It contains the redundant non-cancer target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`ill\_cancer\_type\_r\_n` AS

SELECT

prot.`UPAccession`, prot.`Sequence`

FROM

`proteindatabase`.`proteininfo` AS prot LEFT JOIN `proteindatabase`.`upacc\_cancer` AS cancer

ON

prot.`UPAccession` = cancer.`UPAccession`

WHERE

cancer.`UPAccession` IS NULL AND

prot.`Target` = "Y"

### ill\_cancer\_type\_r\_p

This view is used for the comparison of all the cancer target proteins and all the non-cancer target proteins. It contains the redundant cancer target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`ill\_cancer\_type\_r\_p` AS

SELECT

`proteininfo`.`UPAccession`, `Sequence`

FROM

`proteindatabase`.`proteininfo`

INNER JOIN

`proteindatabase`.`upacc\_cancer`

ON `proteininfo`.`UPAccession` = `upacc\_cancer`.`UPAccession`

WHERE

`proteininfo`.`Target` = "Y"

### type\_gpcr\_targ\_r\_n

This view is used for the comparison of all the GPCR target proteins and all the GPCR non-target proteins. It contains the redundant GPCR non-target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_gpcr\_targ\_r\_n` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "N" AND `ModeOfAction` = "G-protein coupled receptor"

### type\_gpcr\_targ\_r\_p

This view is used for the comparison of all the GPCR target proteins and all the GPCR non-target proteins. It contains the redundant GPCR target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_gpcr\_targ\_r\_p` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "Y" AND `ModeOfAction` = "G-protein coupled receptor"

### type\_ionchannel\_targ\_r\_n

This view is used for the comparison of all the ion channel target proteins and all the ion channel non-target proteins. It contains the redundant ion channel non-target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_ionchannel\_targ\_r\_n` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "N" AND `ModeOfAction` = "Ion Channel"

### type\_ionchannel\_targ\_r\_p

This view is used for the comparison of all the ion channel target proteins and all the ion channel non-target proteins. It contains the redundant ion channel target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_ionchannel\_targ\_r\_p` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "Y" AND `ModeOfAction` = "Ion Channel"

### type\_kinase\_targ\_r\_n

This view is used for the comparison of all the kinase target proteins and all the kinase non-target proteins. It contains the redundant kinase non-target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_kinase\_targ\_r\_n` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "N" AND `ModeOfAction` = "Kinase"

### type\_kinase\_targ\_r\_p

This view is used for the comparison of all the kinase target proteins and all the kinase non-target proteins. It contains the redundant kinase target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_kinase\_targ\_r\_p` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "Y" AND `ModeOfAction` = "Kinase"

### type\_protease\_targ\_r\_n

This view is used for the comparison of all the protease target proteins and all the protease non-target proteins. It contains the redundant protease non-target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_protease\_targ\_r\_n` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "N" AND `ModeOfAction` = "Protease"

### type\_protease\_targ\_r\_p

This view is used for the comparison of all the protease target proteins and all the protease non-target proteins. It contains the redundant protease target proteins. The SQL code used to generate the view is as follows:

CREATE VIEW `proteindatabase`.`type\_protease\_targ\_r\_p` AS

SELECT `UPAccession`, `Sequence` FROM `proteindatabase`.`proteininfo`

WHERE `Target` = "Y" AND `ModeOfAction` = "Protease"

## Non-Redundant Dataset Views

Each non-redundant view corresponds to one of the redundant dataset views. The corresponding views have the same name, except that the fourth element of the non-redundant views is ‘nr’ rather than ‘r’. The SQL code used to generate the non-redundant views, minus the SELECT portion of the query, is as follows:

CREATE VIEW `proteindatabase`.`all\_all\_targ\_nr\_n` AS

SELECT

DISTINCT

...

FROM

proteininfo AS prot,

nonredundant AS nr,

stability AS stab,

upacc\_expression AS exp,

upacc\_variantsnumber AS gv,

upacc\_paralogs AS para,

upacc\_ppi AS ppi,

upacc\_transcripts AS trans

WHERE

prot.UPAccession = nr.UPAccession AND

VIEW\_DEPENDENT AND

prot.UPAccession = stab.UPAccession AND

prot.UPAccession = exp.UPAccession AND

prot.UPAccession = gv.UPAccession AND

prot.UPAccession = para.UPAccession AND

prot.UPAccession = ppi.UPAccession AND

prot.UPAccession = trans.UPAccession

GROUP BY

prot.UPAccession

The SELECT portion of the query is described below in the description of the columns. The VIEW\_DEPENDENT portion of the WHERE clause is different for each non-redundant dataset view, and will be given below for each view. The tables and views in the FROM clause are described above in the table and upacc view sections.

All of these views contain 138 columns. The columns are as follows:

1. UPAccession – Extracted from prot.UPAccession.
2. A – Extracted from prot.A.
3. C – Extracted from prot.C.
4. D – Extracted from prot.D.
5. E – Extracted from prot.E.
6. F – Extracted from prot.F.
7. G – Extracted from prot.G.
8. H – Extracted from prot.H.
9. I – Extracted from prot.I.
10. K – Extracted from prot.K.
11. L – Extracted from prot.L.
12. M – Extracted from prot.M.
13. P – Extracted from prot.P.
14. N – Extracted from prot.N.
15. Q – Extracted from prot.Q.
16. R – Extracted from prot.R.
17. S – Extracted from prot.S.
18. T – Extracted from prot.T.
19. V – Extracted from prot.V.
20. W – Extracted from prot.W.
21. Y – Extracted from prot.Y.
22. NegativelyCharged – Extracted from prot.NegativelyCharged.
23. PositivelyCharged – Extracted from prot.PositivelyCharged.
24. Basic – Extracted from prot.Basic.
25. Charged – Extracted from prot.Charged.
26. Polar – Extracted from prot.Polar.
27. NonPolar – Extracted from prot.NonPolar.
28. Aromatic – Extracted from prot.Aromatic.
29. Aliphatic – Extracted from prot.Aliphatic.
30. Small – Extracted from prot.Small.
31. Tiny – Extracted from prot.Tiny.
32. PESTMotif – Extracted from prot.PESTMotif.
33. LowComplexity – Extracted from prot.LowComplexity.
34. Hydrophobicity – Extracted from prot.Hydrophobicity.
35. Isoelectric – Extracted from prot.Isoelectric.
36. ECNumber – Extracted from prot.ECNumber.
37. OGlycosylation – Extracted from prot.OGlycosylation.
38. NGlycosylation – Extracted from prot.NGlycosylation.
39. Phosphoserine – Extracted from prot.Phosphoserine.
40. Phosphothreonine – Extracted from prot.Phosphothreonine.
41. Phosphotyrosine – Extracted from prot.Phosphotyrosine.
42. SubcellularLocation – Extracted from prot.SubcellularLocation.
43. TopologicalDomain – Extracted from prot.TopologicalDomain.
44. PredictedSubcellularLocation – Extracted from prot.PredictedSubcellularLocation.
45. SignalPeptide – Extracted from prot.SignalPeptide.
46. TransmembraneHelices – Extracted from prot.TransmembraneHelices.
47. AlphaHelices – Extracted from prot.AlphaHelices.
48. BetaStrands – Extracted from prot.BetaStrands.
49. PredictedAlphaHelices – Extracted from prot.PredictedAlphaHelices.
50. PredictedBetaSheets – Extracted from prot.PredictedBetaSheets.
51. Sequence – Extracted from prot.Sequence.
52. 3Untranslated – Extracted using IFNULL(gv.3Untranslated / NULLIF(gv.Total, 0), 0) AS 3Untranslated.
53. 5Untranslated – Extracted using IFNULL(gv.5Untranslated / NULLIF(gv.Total, 0), 0) AS 5Untranslated.
54. NonSynonymousCoding – Extracted using IFNULL(gv.NonSynonymousCoding / NULLIF(gv.Total, 0), 0) AS NonSynonymousCoding.
55. SynonymousCoding – Extracted using IFNULL(gv.SynonymousCoding / NULLIF(gv.Total, 0), 0) AS SynonymousCoding.
56. Paralogs – Extracted using IFNULL(para.Paralogs, 0) AS Paralogs.
57. BinaryPPI – Extracted using IFNULL(ppi.BinaryPPI, 0) AS BinaryPPI.
58. AlternativeTranscripts – Extracted using IFNULL(trans.ProteinCodingTranscripts, 1) AS AlternativeTranscripts.
59. DS\_Embryoid\_Body – Extracted from exp.DS\_Embryoid\_Body.
60. DS\_Blastocyst – Extracted from exp.DS\_Blastocyst.
61. DS\_Fetus – Extracted from exp.DS\_Fetus.
62. DS\_Neonate – Extracted from exp.DS\_Neonate.
63. DS\_Infant – Extracted from exp.DS\_Infant.
64. DS\_Juvenile – Extracted from exp.DS\_Juvenile.
65. DS\_Adult – Extracted from exp.DS\_Adult.
66. HS\_Adrenal\_Tumor – Extracted from exp.HS\_Adrenal\_Tumor.
67. HS\_Bladder\_Carcinoma – Extracted from exp.HS\_Bladder\_Carcinoma.
68. HS\_Breast\_Mammary\_Gland\_Tumor – Extracted from exp.HS\_Breast\_Mammary\_Gland\_Tumor.
69. HS\_Cervical\_Tumor – Extracted from exp.HS\_Cervical\_Tumor.
70. HS\_Chondrosarcoma – Extracted from exp.HS\_Chondrosarcoma.
71. HS\_Colorectal\_Tumor – Extracted from exp.HS\_Colorectal\_Tumor.
72. HS\_Esophageal\_Tumor – Extracted from exp.HS\_Esophageal\_Tumor.
73. HS\_Gastrointestinal\_Tumor – Extracted from exp.HS\_Gastrointestinal\_Tumor.
74. HS\_Germ\_Cell\_Tumor – Extracted from exp.HS\_Germ\_Cell\_Tumor.
75. HS\_Glioma – Extracted from exp.HS\_Glioma.
76. HS\_Head\_And\_Neck\_Tumor – Extracted from exp.HS\_Head\_And\_Neck\_Tumor.
77. HS\_Kidney\_Tumor – Extracted from exp.HS\_Kidney\_Tumor.
78. HS\_Leukemia\_Tumor – Extracted from exp.HS\_Leukemia\_Tumor.
79. HS\_Liver\_Tumor – Extracted from exp.HS\_Liver\_Tumor.
80. HS\_Lung\_Tumor – Extracted from exp.HS\_Lung\_Tumor.
81. HS\_Lymphoma – Extracted from exp.HS\_Lymphoma.
82. HS\_Non\_neoplasia – Extracted from exp.HS\_Non\_neoplasia.
83. HS\_Normal – Extracted from exp.HS\_Normal.
84. HS\_Ovarian\_Tumor – Extracted from exp.HS\_Ovarian\_Tumor.
85. HS\_Pancreatic\_Tumor – Extracted from exp.HS\_Pancreatic\_Tumor.
86. HS\_Primitive\_Neuroectodermal\_Tumor\_Of\_The\_CNS – Extracted from exp.HS\_Primitive\_Neuroectodermal\_Tumor\_Of\_The\_CNS.
87. HS\_Prostate\_Cancer – Extracted from exp.HS\_Prostate\_Cancer.
88. HS\_Retinoblastoma – Extracted from exp.HS\_Retinoblastoma.
89. HS\_Skin\_Tumor – Extracted from exp.HS\_Skin\_Tumor.
90. HS\_Soft\_Tissue\_Muscle\_Tissue\_Tumor – Extracted from exp.HS\_Soft\_Tissue\_Muscle\_Tissue\_Tumor.
91. HS\_Uterine\_Tumor – Extracted from exp.HS\_Uterine\_Tumor.
92. BS\_Adipose\_Tissue – Extracted from exp.BS\_Adipose\_Tissue.
93. BS\_Adrenal\_Gland – Extracted from exp.BS\_Adrenal\_Gland.
94. BS\_Ascites – Extracted from exp.BS\_Ascites.
95. BS\_Bladder – Extracted from exp.BS\_Bladder.
96. BS\_Blood – Extracted from exp.BS\_Blood.
97. BS\_Bone – Extracted from exp.BS\_Bone.
98. BS\_Bone\_Marrow – Extracted from exp.BS\_Bone\_Marrow.
99. BS\_Brain – Extracted from exp.BS\_Brain.
100. BS\_Cervix – Extracted from exp.BS\_Cervix.
101. BS\_Connective\_Tissue – Extracted from exp.BS\_Connective\_Tissue.
102. BS\_Ear – Extracted from exp.BS\_Ear.
103. BS\_Embryonic\_Tissue – Extracted from exp.BS\_Embryonic\_Tissue.
104. BS\_Esophagus – Extracted from exp.BS\_Esophagus.
105. BS\_Eye – Extracted from exp.BS\_Eye.
106. BS\_Heart – Extracted from exp.BS\_Heart.
107. BS\_Intestine – Extracted from exp.BS\_Intestine.
108. BS\_Kidney – Extracted from exp.BS\_Kidney.
109. BS\_Larynx – Extracted from exp.BS\_Larynx.
110. BS\_Liver – Extracted from exp.BS\_Liver.
111. BS\_Lung – Extracted from exp.BS\_Lung.
112. BS\_Lymph – Extracted from exp.BS\_Lymph.
113. BS\_Lymph\_Node – Extracted from exp.BS\_Lymph\_Node.
114. BS\_Mammary\_Gland – Extracted from exp.BS\_Mammary\_Gland.
115. BS\_Mouth – Extracted from exp.BS\_Mouth.
116. BS\_Muscle – Extracted from exp.BS\_Muscle.
117. BS\_Nerve – Extracted from exp.BS\_Nerve.
118. BS\_Ovary – Extracted from exp.BS\_Ovary.
119. BS\_Pancreas – Extracted from exp.BS\_Pancreas.
120. BS\_Parathyroid – Extracted from exp.BS\_Parathyroid.
121. BS\_Pharynx – Extracted from exp.BS\_Pharynx.
122. BS\_Pituitary\_Gland – Extracted from exp.BS\_Pituitary\_Gland.
123. BS\_Placenta – Extracted from exp.BS\_Placenta.
124. BS\_Prostate – Extracted from exp.BS\_Prostate.
125. BS\_Salivary\_Gland – Extracted from exp.BS\_Salivary\_Gland.
126. BS\_Skin – Extracted from exp.BS\_Skin.
127. BS\_Spleen – Extracted from exp.BS\_Spleen.
128. BS\_Stomach – Extracted from exp.BS\_Stomach.
129. BS\_Testis – Extracted from exp.BS\_Testis.
130. BS\_Thymus – Extracted from exp.BS\_Thymus.
131. BS\_Thyroid – Extracted from exp.BS\_Thyroid.
132. BS\_Tonsil – Extracted from exp.BS\_Tonsil.
133. BS\_Trachea – Extracted from exp.BS\_Trachea.
134. BS\_Umbilical\_Cord – Extracted from exp.BS\_Umbilical\_Cord.
135. BS\_Uterus – Extracted from exp.BS\_Uterus.
136. BS\_Vascular – Extracted from exp.BS\_Vascular.
137. HalfLife – Extracted from stab.HalfLife.
138. InstabilityIndex – Extracted from stab.InstabilityIndex.

### all\_all\_targ\_nr\_n

This view is used for the comparison of all the target proteins and all the non-target proteins. It contains the redundant non-target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.AllTargetNegative = "Y".

### all\_all\_targ\_nr\_p

This view is used for the comparison of all the target proteins and all the non-target proteins. It contains the non-redundant target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.AllTargetPositive = "Y".

### ill\_cancer\_prot\_nr\_n

This view is used for the comparison of all the cancer proteins and all the non-cancer proteins. It contains the non-redundant non-cancer proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.CancerProteinNegative = "Y".

### ill\_cancer\_prot\_nr\_p

This view is used for the comparison of all the cancer proteins and all the non-cancer proteins. It contains the non-redundant cancer proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.CancerProteinPositive = "Y".

### ill\_cancer\_targ\_nr\_n

This view is used for the comparison of all the cancer target proteins and all the cancer non-target proteins. It contains the non-redundant cancer non-target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.CancerTargetNegative = "Y".

### ill\_cancer\_targ\_nr\_p

This view is used for the comparison of all the cancer target proteins and all the cancer non-target proteins. It contains the non-redundant cancer target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.CancerTargetPositive = "Y".

### ill\_cancer\_type\_nr\_n

This view is used for the comparison of all the cancer target proteins and all the non-cancer target proteins. It contains the non-redundant non-cancer target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.CancerTypeNegative = "Y".

### ill\_cancer\_type\_nr\_p

This view is used for the comparison of all the cancer target proteins and all the non-cancer target proteins. It contains the non-redundant cancer target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.CancerTypePositive = "Y".

### type\_gpcr\_targ\_nr\_n

This view is used for the comparison of all the GPCR target proteins and all the GPCR non-target proteins. It contains the non-redundant GPCR non-target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.GPCRTargetNegative = "Y".

### type\_gpcr\_targ\_nr\_p

This view is used for the comparison of all the GPCR target proteins and all the GPCR non-target proteins. It contains the non-redundant GPCR target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.GPCRTargetPositive = "Y".

### type\_ionchannel\_targ\_nr\_n

This view is used for the comparison of all the ion channel target proteins and all the ion channel non-target proteins. It contains the non-redundant ion channel non-target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.IonChannelTargetNegative = "Y".

### type\_ionchannel\_targ\_nr\_p

This view is used for the comparison of all the ion channel target proteins and all the ion channel non-target proteins. It contains the non-redundant ion channel target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.IonChannelTargetPositive = "Y".

### type\_kinase\_targ\_nr\_n

This view is used for the comparison of all the kinase target proteins and all the kinase non-target proteins. It contains the non-redundant kinase non-target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.KinaseTargetNegative = "Y".

### type\_kinase\_targ\_nr\_p

This view is used for the comparison of all the kinase target proteins and all the kinase non-target proteins. It contains the non-redundant kinase target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.KinaseTargetPositive = "Y".

### type\_protease\_targ\_nr\_n

This view is used for the comparison of all the protease target proteins and all the protease non-target proteins. It contains the non-redundant protease non-target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.ProteaseTargetNegative = "Y".

### type\_protease\_targ\_nr\_p

This view is used for the comparison of all the protease target proteins and all the protease non-target proteins. It contains the non-redundant protease target proteins. For this view, the VIEW\_DEPENDENT portion of the view generation query is as follows: nr.ProteaseTargetPositive = "Y".