Predicting the Effect of Mutations on a Genome-Wide Scale

by

Alexey Strokach

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Predicting the Effect of Mutations on a Genome-Wide Scale

Alexey Strokach
Master of Science
Graduate Department of Computer Science
University of Toronto
2016

Advances in DNA sequencing technology have led to an enormous growth in the amount of available genomic data. Interpreting this data to produce meaningful and actionable results remains a challenge. Tools currently in use for annotating discovered variants rely on a sequence conservation score and provide little mechanistic insight to explain why a particular variant may be deleterious. Tools for predicting the effect of mutations on the structure and function of a protein are laborious to use and require a crystal structure of the protein, severely limiting their coverage. ELASPIC, a tool recently developed in our lab, uses homology models instead of crystal structures to accurately predict the effect of mutation on protein stability and protein-protein interaction affinity. In this work we extend ELASPIC to predict the effect of mutations on a genome-wide scale. We discuss the importance of structural and sequential information in our ability to accurately predict the energetic effect of mutations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Advances in DNA sequencing technology have drastically lowered the cost and improved the accuracy of high-throughput sequencing [1]. Different sequencing techniques, including RNA-seq, whole-genome sequencing and whole-exome sequencing [2], now present as viable and cost-effective tools both in the laboratory, where they permits the study of individual cells and cell populations at an unprecedented level of detail [3], and in the clinic, where they can assist in the diagnosis and treatment of pediatric diseases [4] and in the design of targeted therapies against cancer [5]. However, while there has been enormous growth in the amount of genomic data that is generated and the number of sequence variants that are discovered, interpreting this data to produce meaningful and actionable results remains a challenge.

In vivo and *in vitro* experimental techniques remain the gold standard for elucidating the effect of DNA sequence variants. However, evaluating experimentally the effect of all discovered variants is not feasible, both in terms of time and resources that would be required. Computational techniques have been developed to predict the effect of different variants and to prioritize them for experimental validation. Those techniques can be loosely divided into three categories: sequence-based tools, structure-based tools, and tools that combine both sequential and structural information.

Sequence-based tools rely on some form of a conservation score, describing the frequency with which a particular nucleotide or amino acid is found at the given position in domain-, protein- or genome-level alignments, in order to make their prediction [6, 7, 8, 9, 10, 11, 12]. Due to their speed and scalability, sequence-based tools are the de-facto standard for annotating newly discovered variants. However, they remain limited in their accuracy and the type of information that they can provide [13]. In particular, they only predict whether or not a particular mutation is likely to be deleterious, and provide no information as to *why* that mutation may be deleterious. This makes it difficult to act upon those predictions, for example by designing drugs that would curtail the effect of disease-causing mutations or would take advantage of mutations found in cancer.

Structure-based tools predict the effect of mutations on protein structure and / or function using features describing the three-dimensional structure of the protein. They range from accurate but computationally expensive alchemical free energy calculations, which involve modelling the structural transition from wildtype to mutant proteins and using different integration techniques to calculate the energy of the transition [14], to quicker but more approximate techniques, which use semi-empirical or statistical potentials and assume that the backbone of the protein remains fixed [15, 16, 17, 18]. In theory, structure-based tools should be able to offer more insight into the effect of missense mutations

than sequence-based tools, since the effect is caused by changes in protein structure and function and not by changes in DNA sequence. However, since existing structure-based tools require manual setup and a crystal structure of the protein being mutated, there has not been a systematic, genome-wide comparison of the performance of sequence- and structure-based tools in the analysis of mutations.

Several tools have been developed that attempts to combine sequence- and structure-based information in order to make more accurate predictions about the deleteriousness [19] and the structural impact [20, 21, 22] of mutations. These tools generally are "meta-predictors" which integrate the results of several sequence- and structure-based tools using a machine learning algorithms trained on an appropriate dataset. ELASPIC, developed by Niklas Berliner *et al.*, is a particularly interesting example, because, while trained using homology models instead of crystal structures, it still achieves relatively high accuracy in predicting the effect of mutations on protein stability and protein-protein interaction affinity [21]. With the growth in the number of crystal structures deposited in the Protein Data Bank [23], it is now possible to create homology models of proteins and protein-protein interactions with high coverage of an entire proteome [24]. This suggests that ELASPIC could be extended to work on a genome-wide scale, offering a way to examine the contribution that structural information could make to our analysis and interpretation of mutations.

The aim of this project was to extend ELASPIC so that it could predict the effect of mutations on protein stability and protein-protein interaction affinity on a genome-wide scale. In Chapter 2, we describe modifications that had to be made to ELASPIC and the underlying pipelines in order to make the genome-wide analysis of mutations possible. We also discuss the implementation of the ELASPIC web service, which allows ELASPIC to be run through a webserver in a scalable way. In Chapter 3, we describe the performance of ELASPIC on the training, validation and test datasets, and compare its performance to other sequence- and structure-based tools. In Chapter 4, we discuss the results of this work and propose several directions for future study.

Chapter 2

Implementation

2.1 Profs

ELASPIC uses a domain-based approach for creating homology models of query proteins, and therefore requires access to accurate domain definitions. The most widely-used source of protein domain definitions is Pfam [25]. However, since Pfam domains definitions are based entirely on protein sequence, they correlate poorly with the structural fold of the protein. Using Pfam domain definitions when making homology models tends to produce unstable models of fragmented and / or truncated domains, and this would compromise our subsequent analysis of the structural impact of mutations.

In order to improve the structural accuracy of Pfam domains, Andres Felipe Giraldo Forero developed a pipeline that uses structural alignments and a set of heuristics to modify Pfam domain definitions and make them better aligned with the tertiary structure of the protein, as defined by CATH [26]. He named this pipeline Profs, for Protein families. A schematic of this pipeline is presented in Figure 2.1, and an R package implementing the pipeline is available at `https://bitbucket.org/afgiraldofo/profs`. Profs domains have an advantage over Pfam domains in that they have been corrected and expanded to match the structural fold of the protein. They have an advantage over CATH domains in that they are backed by large, manually-seeded alignments, and can be easily detected in any protein sequence using Pfam HMMs.

We used Andres' pipeline to annotate with Profs domains all proteins in the UniProt database. The resulting table of Profs domain definitions is available for download from the ELASPIC website (`http://elaspic.kimlab.org/static/download/`) and is included in the ELASPIC database (see **domain** and **domain_contact** tables in Figure 2.6 and Table 2.1). The following sections describe the procedure used to generate tables of Profs domain definitions and Profs domain-domain interactions that are used by ELASPIC.

2.1.1 Domains

We used Profs domain definitions, which had been calculated for all proteins in the PDB, to find Profs domains, and structural templates for those domains, for all proteins in UniProt (step 6 in Figure 2.1). To do this, we followed the same procedure that was used by the authors of Profs to annotate structures in the PDB that have Pfam domains but no CATH domains [27].

We started with Pfam domain definitions for all known protein sequences, which we download from

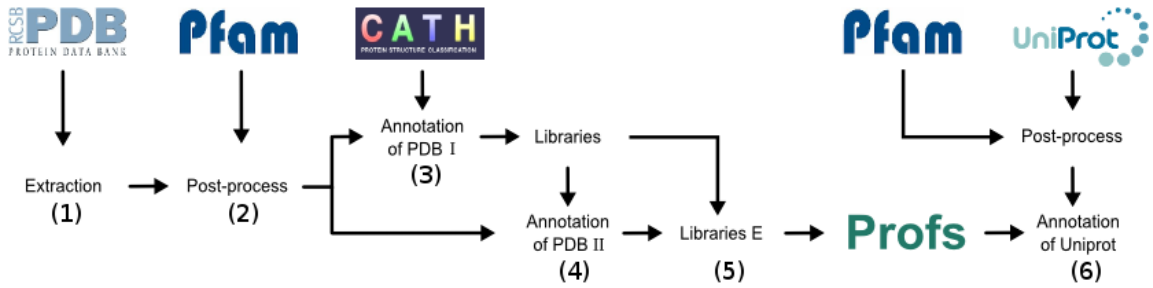


Figure 2.1: Flowchart illustrating the steps in the Profs pipeline (courtesy of Andres Felipe Giraldo Forero). **(1)** All structures in the PDB are parsed to extract protein sequences, and *hmmscan* is ran to find Pfam domains in those sequences. **(2)** Pfam domains of proteins in the PDB are processed in order to join and / or remove overlapping and repeating domains. **(3)** Pfam domain definitions are altered in order to make them compatible with CATH definitions, for structures that have been annotated by CATH. **(4)** Pfam domain definitions are altered in order to make them compatible with CATH definitions, for structures that have not been annotated by CATH. This is done by performing pairwise alignments with structures that do have CATH annotations. **(5)** Libraries of Profs domain definitions, and Profs domain-domain interactions, are generated for all proteins in the PDB. **(6)** Libraries of Profs domain definitions, and Profs domain-domain interactions, are generated for all proteins in Uniprot.

the SIMAP website [28]. We mapped those protein sequences to Uniprot using the MD5 hash of each sequence, and we joined or removed overlapping and repeating domains using a mapping table supplied with the Profs R package. Next, we tried to find a Profs structural template for each Pfam domain by running *blastp* against libraries of Profs domains, which also are included in the Profs R package. If a suitable template was found, we proceeded to do iterative global alignments using Muscle [29] while expanding domain boundaries of the Pfam domains to match domain boundaries of the Profs templates. If two Pfam domains were expanded to occupy the same region in the protein, that region was divided in half and attached to the preceding and the succeeding domains.

The results of this analysis are stored in the **uniprot_domain** and the **uniprot_domain_template** tables in the ELASPIC database (Figure 2.6). The **uniprot_domain** table contains all Pfam domains and supradomains that are obtained after removing repeating and overlapping domains, as outlined above. The *pdbfam_name* column contains the name of the Profs domain. The *alignment_def* column contains either the original Pfam domain definitions or, in the case of supradomains, the merged domain definitions of multiple Pfam domains. The **uniprot_domain_template** table contains information describing the alignment of the Pfam domain or supradomain with the corresponding Profs structural template, for domains for which a suitable Profs template could be found. The *cath_id* column identifies the Profs structural template that was selected, and the *domain_def* column contains the corrected and expanded domain definitions.

2.1.2 Comparison with other methods

The two most prominent methods for detecting structural domains in protein sequences are SUPERFAMILY [30], which is based on SCOP [31] domain definitions, and Gene3D [32], which is based on CATH [26] domain definitions. Both methods use hidden Markov models, trained on curated structural domains in proteins in the PDB, to detect structural domains in all other proteins. Gene3D further

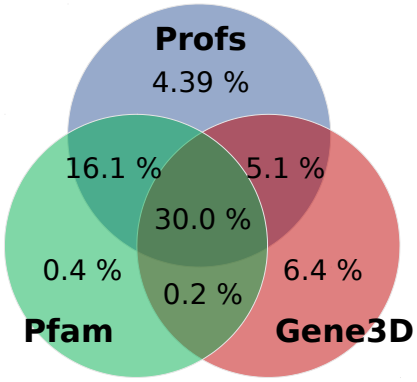


Figure 2.2: Venn diagram showing the overlap in domain definitions between Profs, Pfam, and Gene3D. Values represent the fraction of amino acids, of all human proteins in UniProt, which are covered by the particular domain or domains. A total of 18,828 human proteins and 10,868,810 amino acids were considered, after excluding proteins which had no predicted domains by any method. Profs has the highest coverage, with 55.7 % of amino acids being annotated by a Prof domain.

employs a set of heuristics to join repeating domains that have a similar sequence.

In order to ascertain the validity of Profs domain definitions, we compared Profs, Pfam and Gene3D in terms of sequence coverage (Figure 2.2) and domain size (Figure 2.3). While we did not include SUPERFAMILY in this comparison, it is unlikely that SUPERFAMILY would show better performance than Profs or Gene3D, as it is based on the SCOP database that has not been updated since 2009.

We downloaded Pfam and Gene3D domain definitions for all human proteins from SIMAP [28], and we calculated Profs domain definitions following the pipeline described above. The analysis was restricted to 18,828 human proteins from UniProt which are annotated with at least one Profs, Pfam or Gene3D domain.

In order to compare sequence coverage, we looked at the fraction of all protein sequences which are covered by each domain type (Figure 2.2). Overall, Profs has the highest sequence coverage, with 55.7 % of 10,868,810 amino acids in 18,828 proteins residing inside a Profs domain. Profs annotates $\approx$ 9% more amino acids than Pfam and $\approx$ 14% more amino acids than Gene3D, although the relatively low coverage by Gene3D is expected, as it can only detect domains which are represented in the PDB.

In order to compare domain size, we looked at the average number of domains per protein for each of the three methods (Figure 2.3). Profs has more proteins with only one domain per protein, while Pfam and Gene3D have more proteins with two or more domains per protein. This is consistent with Profs trying to join fragmented and repeating domains into consistent structural units. Gene3D does not detect domains in many proteins with Profs and Pfam domains, likely because those domains have not been crystallized.

The result of this analysis shows that, at least for human proteins, Profs achieves higher sequence coverage using fewer domains per protein than either Pfam or Gene3D. This makes Profs well-suited for the ELASPIC pipeline.

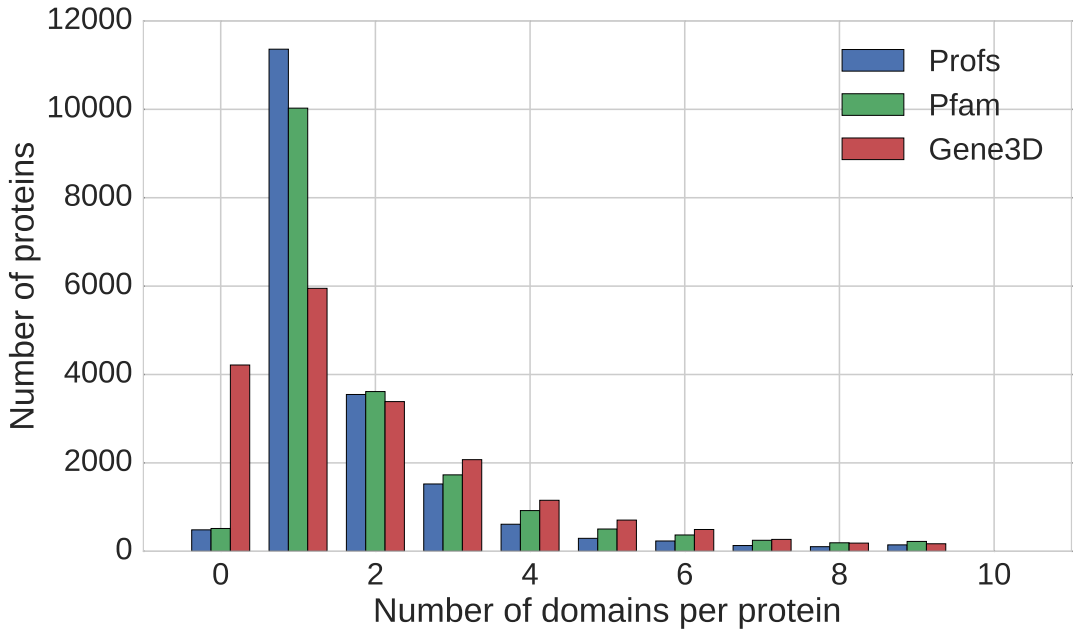


Figure 2.3: Average number of Profs, Pfam and Gene3D domains per protein, for all human proteins containing at least one domains. Profs tends to have fewer domains per protein then either Pfam or Gene3D, even though Profs domains have higher sequence coverage (see Figure 2.2). Gene3D lacks domain annotation for many proteins which contain at least one Pfam and Profs domain.

2.1.3 Domain interactions

We also created a table of domain-domain interactions for proteins that are known to interact and for which a homology model of the interaction can be created. We started by creating a comprehensive list of protein-protein interactions (PPIs), by taking the union of all PPIs listed in the HIPPIE database [33] and in the datasets hosted by the Harvard Center for Cancer Systems Biology (CCSB) [34]. The overlap in the PPIs obtained from each source is presented in Figure 2.4. We filtered those PPIs to select pairs of proteins where each protein has at least one domain with a structural template. This information is stored in the **uniprot_domain_pair** table in the ELASPIC database. For each of those domains, we perform a Blast search of the domain sequence against a library of Profs domains in the PDB (the **domain** table in Figure 2.6), and we selected only those templates that occur in the same crystal structure in both proteins and that interact according to the **domain_contact** table. In order to select the best template for the interaction, we calculate a quality score for each of the two domains using Equation 2.1, and chose the template with the highest geometric mean of the two scores (Equation 2.2).

$$alignment\_score = 0.95 \cdot seq\_identity \cdot coverage + 0.05 \cdot coverage \tag{2.1}$$

$$combined\_alignment\_score = \sqrt{alignment\_score\_1 \cdot alignment\_score\_2} \tag{2.2}$$

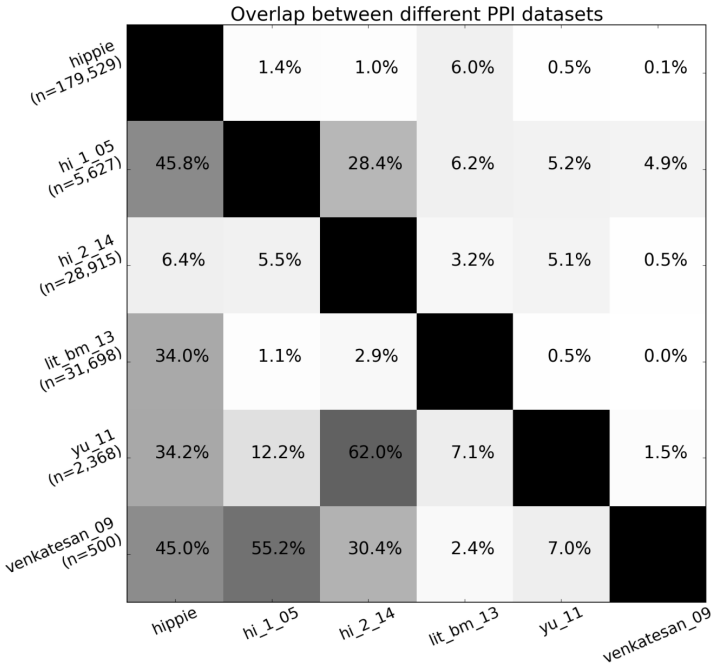


Figure 2.4: Overlap in protein-protein interaction (PPI) databases. The shade and value of each square denotes the percentage of PPIs in the database named on the y-axis that are also found in the database named on the x-axis. **hippie** is a meta-database, which integrates PPIs from many different sources [33]. **hi_1_05** contains PPIs discovered through a proteome-wide yeast two-hybrid experiment conducted by Rual *et al.* [35]. **hi_2_14** contains PPIs discovered through a proteome-wide yeast two-hybrid experiment conducted by Rolland *et al.* [34]. **lit_bm_13** contains PPIs obtained from the literature and supported by multiple pieces of evidence [34]. **yu_11** contains PPIs obtained using "stitch-seq", which combines PCR stitching with next-generation sequencing [36]. **venkatesan_09** corresponds to high-quality binary interactions found in repeat yeast two-hybrid assays conducted by Venkatesan *et al.* [37].
The **hippie** database was downloaded from the Hippie website: `http://cbdm-01.zdv.uni-mainz.de/~mschaefer/hippie/`. All other datasets were downloaded from the Harvard Center for Cancer Systems Biology: `http://interactome.dfci.harvard.edu/H_sapiens/`.

2.2 ELASPIC

The ELASPIC project was started by Niklas Berliner and others in 2014 [21]. ELASPIC uses Modeller [38] to construct homology models of domains and domain-domain interactions, FoldX to optimize those models and to introduce mutations [39], and the gradient boosting regressor algorithm [21], implemented in scikit-learn [40], to combine FoldX energy terms with the mutation deleteriousness score and other features and predict the energetic impact of a mutation on the stability of a single domain or the affinity between two domains.

Since the original publication, the ELASPIC pipeline was modified in several ways. First, instead of using SIFT [6] to calculate the mutation deleteriousness score, we now use Provean [12]. Provean is reported to achieve similar performance to SIFT [12], but it uses a more permissive GPLv3 license and is easier to compile, run and distribute to different machines. Like SIFT, Provean runs PSI-Blast to create a multiple sequence alignment for the query protein. However, instead of using the entire alignment, Provean runs CD-HIT to select under 50 representative sequences, referred to as the "supporting set", which capture the diversity of the alignment. The supporting set for a particular protein can be precalculated and stored for future use, allowing all subsequent mutations to be evaluated in seconds. Second, for calculating solvent-accessible surface area, we now use Maximal Speed Molecular Surface (MSMS) [41] instead of NACCESS [42]. We found that MSMS is more robust to different anomalies that occur PDB crystal structures. Third, when preparing features that are used by the ELASPIC core and interface predictors, we now include the values calculated for the wild-type structure and the *difference* between the values calculated for the wild-type and mutant structures. Previously, we used the values computed for the wild-type structure and the values computed for the mutant structure, but we found that the difference in those values correlates better with the experimental $\Delta\Delta G$ than either of the values itself.

The scripts making up the ELASPIC pipeline were restructured in order to allow for easy testing and reusability (Figure 2.5). ELASPIC now includes core "library" (Figure 2.5, centre), which contains code for aligning query sequences to structural templates, computing Provean supporting sets and mutation deleteriousness scores, constructing homology models, running FoldX, and predicting the $\Delta\Delta G$ of the mutations. It also includes a "standalone pipeline" (Figure 2.5, right), which provides a command-line interface for introducing mutations into individual structures or homology models, and a "database pipeline" (Figure 2.5, left), which provides a command-line interface for running mutations on a genome-wide scale using a database backend.

We distribute ELASPIC and all the programs that it requires as conda packages, which makes it easy to install ELASPIC on any Linux system. The source code for the ELASPIC pipeline is hosted on GitHub (`https://github.com/kimlaborg/elaspic`), ELASPIC documentation is hosted on ReadtheDocs (`http://elaspic.readthedocs.io/`), and precalculated data can be downloaded from the ELASPIC website (`http://elaspic.kimlab.org/static/download/`).

2.2.1 Standalone pipeline

The standalone pipeline works without downloading and installing a local copy of the ELASPIC and PDB databases, but requires, for every mutation, either a PDB file with the structure of the protein to be mutated, of a FASTA file with the sequence of the protein to be mutated and a PDB file with the structure to be used for homology modelling. The output of the pipeline is saved as JSON files inside

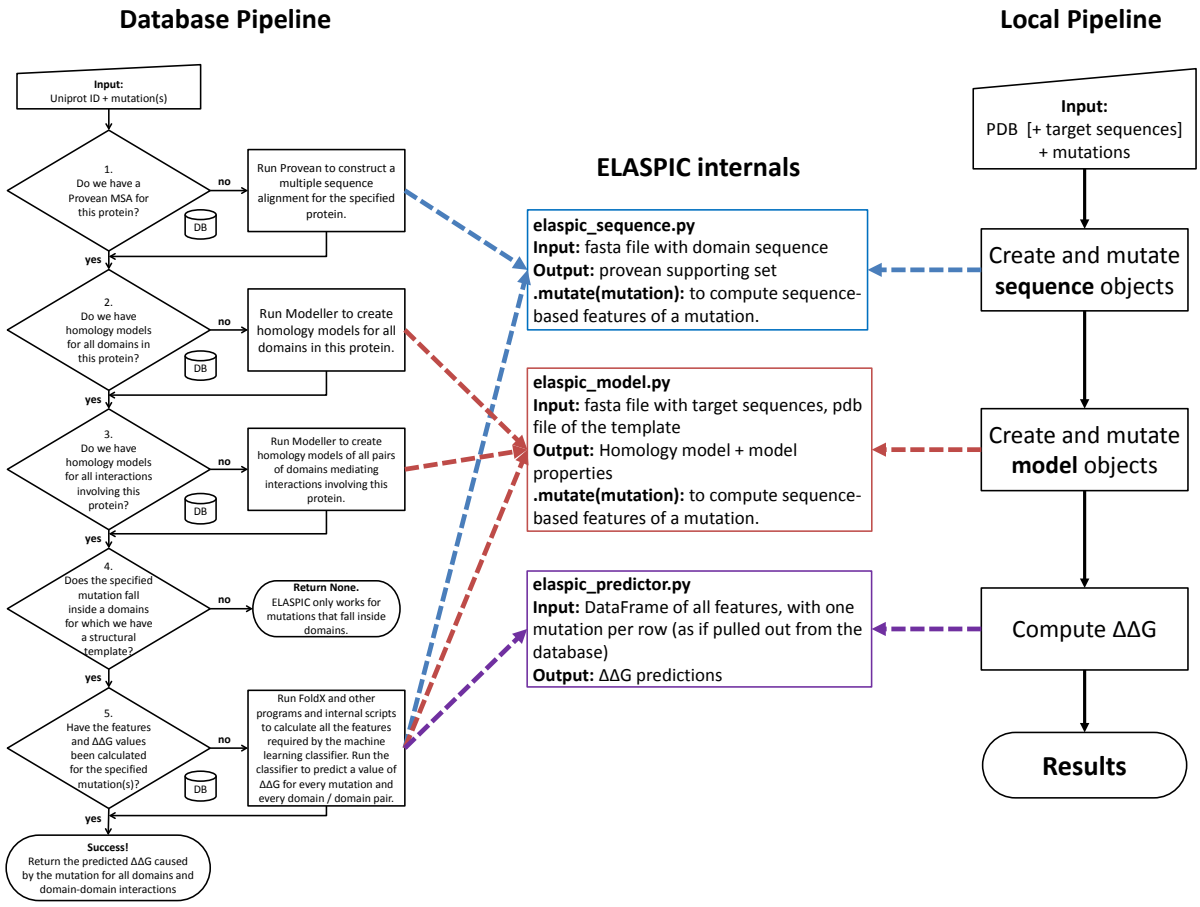


Figure 2.5: Overview of the ELASPIC pipeline. **Database Pipeline:** A user runs the ELASPIC pipeline by specifying the UniProt identifier of the protein being mutated, and one or more mutations affecting that protein. At each decision node, the pipeline queries the database (Figure 2.6) to check whether or not the required information has been calculated previously. If the required data has not been calculated, the pipeline calculates it on the fly and stores the results in the database for later retrieval. The pipeline proceeds until homology models of all domains in the protein, and all domain-domain interactions involving the protein, have been calculated, and the $\Delta\Delta G$ has been predicted for every specified mutation. **Local Pipeline:** A user runs the ELASPIC pipeline by specifying a PDB file with the structure of the protein that they wish to mutate and one or more mutations, or by specifying a FASTA file with the sequence of the protein that they wish to mutate, a PDB file with the structural template to be used for homology modelling and one or more mutations. ELASPIC runs Provean to calculate the supporting set, runs MODELLER to make the homology model, and runs FoldX to compute structural features describing the wildtype and mutant residues. Results are stored in a local *.elaspic* folder, and the Provean supporting set and homology models are *not* recalculated if the user decides to run more mutations.

the *.elaspic* subfolder created in the working directory. The general overview of the local pipleine is presented on the right side of Figure 2.5.

2.2.2 Database pipeline

The database pipeline allows mutations to be performed on a genome-wide scale, without having to specify a structural template for each protein. This pipeline requires a local installation of a relational

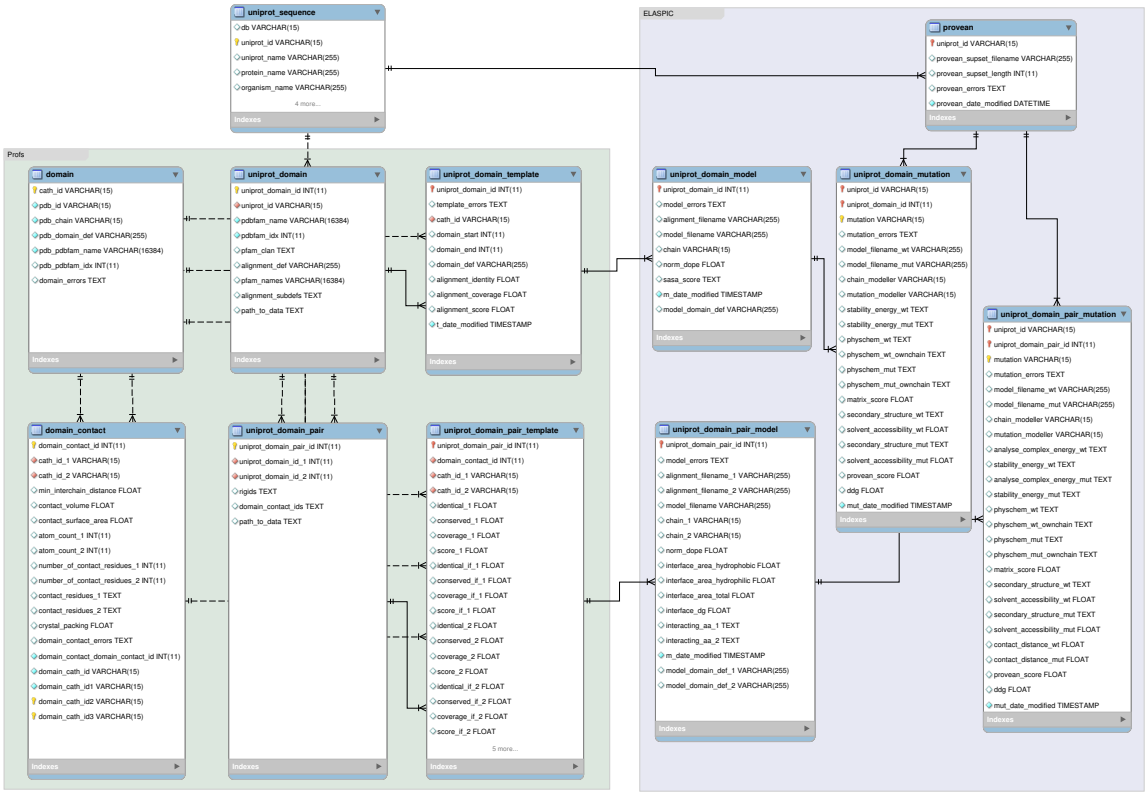


Figure 2.6: ELASPIC database schema. Tables on the green plate titled Profs are calculated using the Profs pipeline, following the procedure outlined in Figure 2.1. Tables on the purple plate titled ELASPIC are calculated using the ELASPIC pipeline, following the "database pipeline" shown in Figure 2.5. A detailed description of each table is provided in Table 2.1.

database containing ELASPIC domain definitions and templates, as well as a local copy of the BLAST and PDB databases.

The general overview of the database pipleine is presented on the left side of Figure 2.5. A user runs the ELASPIC pipeline by providing the UniProt ID of the protein being mutated, and one or more mutations affecting that protein. At each decision node, the pipeline queries the database to check whether or not the required information has been previously calculated. If the required data has not been calculated, the pipeline calculates it on the fly and stores the results in the database for later retrieval. The pipeline proceeds until homology models of all domains in the protein, and all domain-domain interactions involving the protein, have been calculated, and the $\Delta\Delta G$ has been predicted for every specified mutation.

Results of the database pipeline are store in the ELASPIC database. An overview of the ELASPIC database schema is presented in Figure 2.6, and a description of each database table is provided in Table 2.1.

Table 2.1: Description of the tables in the ELASPIC database schema (Figure 2.6).

Table name	Table description
domain	Contains Profs domain definitions for all proteins in the PDB.
domain_contact	Contains information about interactions between Profs domains in the PDB. Only interactions that are predicted to be real by NOXclass [43] are included in this table.
uniprot_sequence	Contains protein sequences for all proteins that are annotated with Profs domains in the **uniprot_domain** table. This table is constructed by downloading and parsing *uniprot_sprot_fasta.gz*, *uniprot_trembl_fasta.gz* and *homo_sapiens_variation.txt* files from Uniprot.
provean	Contains information about Provean [12] supporting set files. The construction of a supporting set is the longest part of running Provean. Thus, in order to speed up the evaluation of mutations, the supporting set is precalculated and stored for every protein.
uniprot_domain	Contains Profs domain definitions for proteins in the **uniprot_sequence** table. This table is obtained by downloading Pfam domain definitions for all known proteins from SIMAP [28], and mapping those proteins to Uniprot using the MD5 hash of each sequence. Overlapping and repeating domains are either merged or deleted, as described in [27].
uniprot_domain_template	Contains structural templates for domains in the **uniprot_domain** table. The *domain_def* column contains expanded and corrected domain definitions for every domain.
uniprot_domain_model	Contains information about the homology models that were created using structural templates in the **uniprot_domain_template** table.
uniprot_domain_mutation	Contains information about the structural impact of core mutations, calculated by introducing those mutations into homology models listed in the **uniprot_domain_model** table. The *ddg* column contains the predicted change in the Gibbs free energy of protein folding.
uniprot_domain_pair	Contains pairs of domains that are likely to mediate the interaction between pairs of proteins listed in Hippie [33] and Rolland *et al.* [34].
uniprot_domain_pair_template	Contains structural templates for domain pairs in the **uniprot_domain_pair** table.
uniprot_domain_pair_model	Contains information about homology models that were created using structural templates in the **uniprot_domain_pair** table.
uniprot_domain_pair	Contains information about the structural impact of interface mutations, calculated by introducing those mutations into homology models listed in the **uniprot_domain_pair_model** table. The *ddg* column contains the predicted change in the Gibbs free energy of protein-protein binding.

2.2.3 Jobsubmitter

In order to make ELASPIC accessible to a wider scientific audience, Daniel Witvliet created the ELASPIC webserver, which allows users to run ELASPIC for their protein and mutation of interest and to analyze interactively ELASPIC results [27].

One limitation of the webserver was that it spawned ELASPIC jobs on the same virtual machine as the webserver. This meant that only a few mutations could be analyzed at a time, and that the webserver could stall when running mutations in a protein lacking a precalculated Provean supporting set, since constructing a Provean supporting set could require more RAM than the virtual machine had available. In order to make the webserver scale to thousands of mutations, we decided to restructure the job execution backend to run ELASPIC on the local Sun Grid Engine (SGE) cluster. However, this design introduced several challenges.

First, since users can run multiple mutations affecting the same protein, we had to make sure that the Provean supporting sets and homology models are calculated first, before jobs for individual mutations are submitted to the cluster. Otherwise, each mutation would initiate the calculation of a Provean supporting set, which can require more than 5 GB of memory, and a homology model, which can take more than 30 minutes to complete. This would lead to many unnecessary jobs and would drastically lower our throughput.

Second, jobs running on a SGE cluster can die unexpectedly, if, for example, they exceed allocated resources, or if the node on which they are executing experiences a hardware failure. In most cases, jobs do not get an opportunity to send an error message before they are terminated. Therefore, we had to keep track of all running jobs, and resubmit jobs that do not finish successfully.

Third, in order to send a "Job Complete" email once all mutations submitted by a particular user have been evaluated, we had to keep track of the relationship between mutations and users that submitted those mutations.

One possible way to address those design requirements would be to use an asynchronous task queue, such as Celery. However, since different tasks inside the queue do not have a shared memory state, each task would have to periodically execute a *qstat* command on the SGE master node in order to monitor the status of the submitted jobs. Since we could have thousands of mutations running on the cluster at the same time, this would not be a scalable solution.

An alternative approach, which we used for the final design, was to create an independent web service which would submit ELASPIC jobs to the SGE cluster and would monitor their progress. We called this web service the ELASPIC "jobsubmitter". It was implemented using the *aiohttp* library, which leverages the *asyncio* event loop and improved support for asyncronous programming present in Python 3.5 (Figure 2.7a).

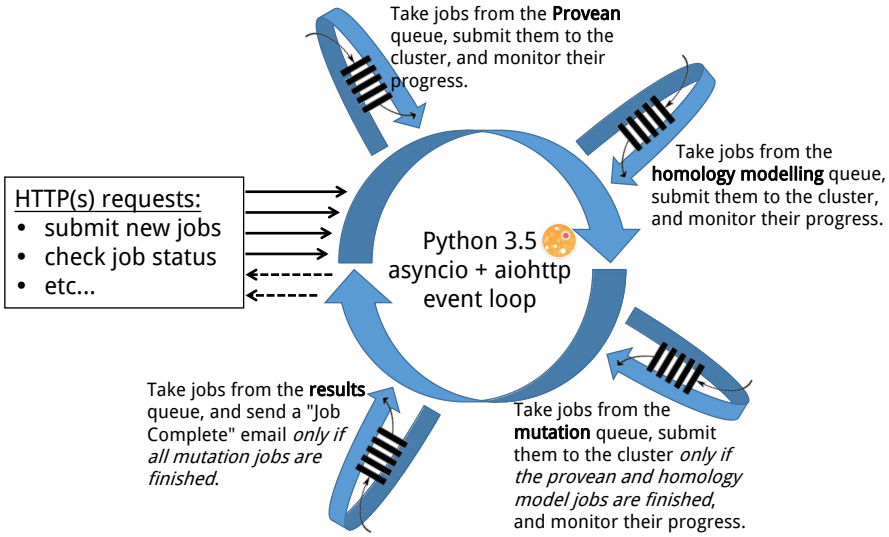
Once the jobsubmitter receives a *GET* or *POST* request containing a set of mutations, information concerning those mutations is distributed into the following queues:

- A "Provean" queue, which contains proteins for which a Provean supporting set has not been calculated.
- A "homology model" queue, which contains proteins for which a homology model has not been calculated.
- A "mutation" queue, which contains individual mutations.
- An "email" queue, which contains the set of mutations associated with each job.

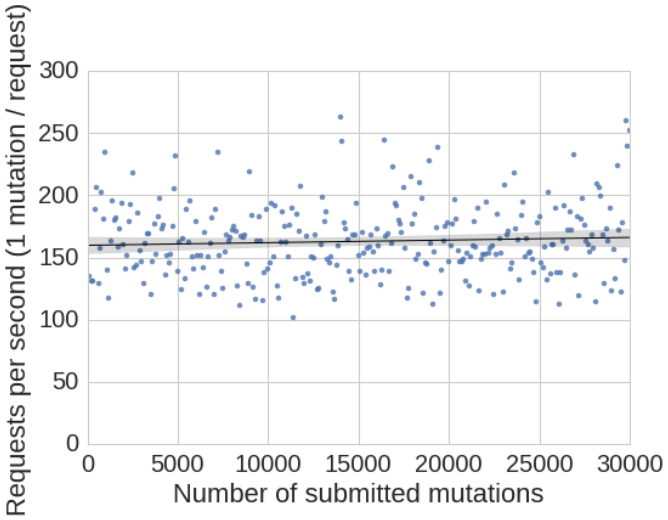
The information from those queues is then processed by the corresponding coroutines:

- For each protein in the "Provean" queue, a job is submitted to the SGE cluster, which calculates the Provean supporting set. If the Provean supporting set for the protein has already been calculated, the protein is taken of the "Provean" queue with no further action.
- For each protein in the "homology model" queue, a job is submitted to the SGE cluster, which calculates the homology model of the protein. If the homology model of the protein has already been calculated, the protein is taken of the "homology model" queue with no further action.
- For each mutation in the "mutation" queue, a job is submitted to the SGE cluster, which runs ELASPIC to calculate the $\Delta\Delta G$ of the mutation. This happens *only if the Provean supporting set and homology model for the protein have already been calculated.*
- For each job in the "email" queue, a "Job Complete" email is sent to the specified email address once all mutations for the associated job have been completed.

The ELASPIC jobsubmitter is able to handle over 150 requests per second, even with 30,000 mutations already being processed by the web service (Figure 2.7b). Therefore, it should not increase significantly the response time of the ELASPIC webserver and should not be the limiting factor in the number of jobs that can be submitted at the same time.



(a) Overview of the ELASPIC "jobsubmitter" web service. The web service was implemented using Python 3.5 and the *aiohttp* library. It contains a central *asyncio* event loop, data structures holding information about the mutations being processed, and coroutines which submit jobs to the SGE cluster, monitor job progress, and perform other maintenance tasks.



(b) Plot showing the number of requests per second handled by the ELASPIC jobsubmitter as a function of the number of mutations that are already being processed.

Figure 2.7: Implementation (a) and performance (b) of the ELASPIC "jobsubmitter".

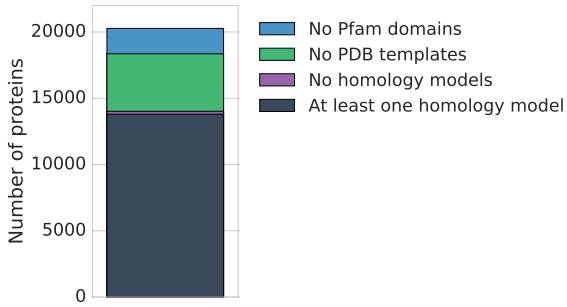
2.2.4 Precalculated data

In order to increase the speed with which the ELASPIC webserver generates results, we attempted to precalculate homology models and Provean supporting sets for all human proteins, and to precalculate mutations known to be involved in human disease.

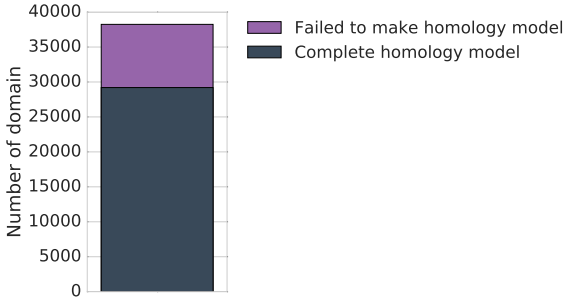
Out of 20,270 human proteins in the SwissProt database, 18,355 proteins have at least one Pfam domain, and 14,015 proteins have a Pfam domain for which we could find a structural template in the PDB (Figure 2.8a). We could create a homology model of at least one domain in 13,796 proteins, with the fraction of each protein covered by a homology model shown in Figure 2.8c. On the domain level, out of a total of 38,243 domains with a structural template, we could create a homology model for only 29,201 domains, or 76% (Figure 2.8b). The main reason for failing to calculate a homology model was low sequence identity between the domain being modelled and the structural template.

We also attempted to create a homology model for all protein pairs found in the HIPPIE [33] and CCSB [34] databases, keeping only the pairs where each protein has at least one domain with a homology model and where we would could find a structural template of the protein-protein interaction. Out of a total of 19,964 such protein pairs, we calculated a homology models for 18,956, or 95 % (Figure 2.9).

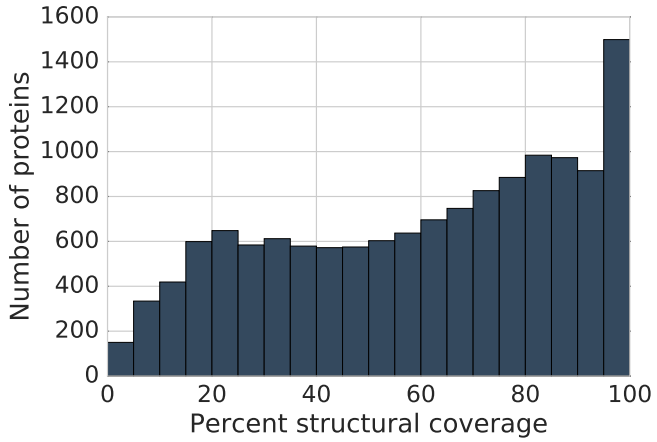
We successfully precalculated a Provean supporting set for all 14,015 human proteins with a Profs domain with a structural template. We also precalculated over 990,000 million mutations implicated in human diseases or found in human cancers, including nearly 600,000 mutations in different protein-protein interfaces.



(a) Diagram showing the number of *proteins* in the human SwissProt database that have no Pfam domains (blue), that have Pfam domains but no structural templates (green), that have Pfam domains and structural templates but no homology models (purple), and proteins with a homology model of at least one domain (grey).



(b) Diagram showing the number of *domains* in all proteins in the human SwissProt database for which we failed to create a homology model (purple) and for which we successfully created a homology model (grey). The most common reason for failing to create a homology model was low sequence identity between the Profs domain and the structural template.



(c) The percentage of protein sequence covered by Profs domains with homology models, for all proteins in the human SwissProt database that have a homology model of at least one domain.

Figure 2.8: Plots showing the number of proteins for which we could create a homology model (a), the number of domains for which we could create a homology model (b), and the structural coverage of proteins with at least one modelled domain (c). Plots were generated using all human proteins in the SwissProt database.

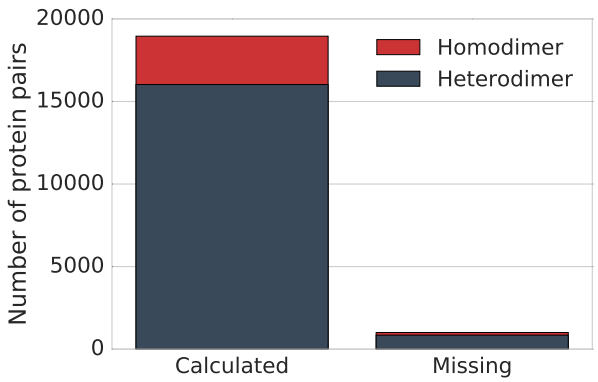


Figure 2.9: Number of homo-dimeric (red) and hetero-dimeric (grey) protein-protein interactions for which we created a homology model (left) and failed to create a homology model (right). In this figure, protein-protein interactions are defined as all pairs of proteins from the human SwissProt database that are found to interact according to one of the protein-protein interaction databases (see Figure 2.4) and that have at least one structural template of the interaction.

Chapter 3

Results

After making changes to the ELASPIC pipeline that are described in Section 2.2, we retrained ELASPIC core and interface predictors and validated them on new data. This involved curating high-quality training, validation and test datasets (Section 3.1), selecting the best hyperparameters for the machine learning algorithm using grid-search (Section 3.2), selecting the set of most informative features using feature elimination (Section 3.3), and testing the final predictors on the test datasets to compare their performance with competing methods (Section 3.4).

3.1 Datasets

The ELASPIC pipeline includes two machine learning predictors: a "core predictor", which predicts the change in the Gibbs free energy of folding ($\Delta\Delta G_{core}$) caused by mutations, and an "interface predictor", which predicts the change in the Gibbs free energy of protein-protein interaction ($\Delta\Delta G_{interface}$) caused by mutations. In order to train, validate and test those predictors, we compiled a number of datasets from different sources, as described in Table 3.1.

We used the "Protherm" dataset to train the core predictor, and the "Skempi" dataset to train the interface predictor. For the training datasets, we calculated features describing each mutation using the standalone pipeline (see Section 2.2.1) and the database pipeline (see Section 2.2.1) in order to make sure that both pipelines produce similar results and that the trained predictors perform well in both settings. In cases where only the PDB position of the mutation is provided in the dataset, we mapped the PDB position to the corresponding UniProt protein using SIFTS [44]. For the database pipeline, we attempted to construct four homology models of each domain and domain-domain interaction, with sequence identity to the template structure falling in each of the following bins: less than or equal to 40% sequence identity, greater than 40% but less than or equal to 60% sequence identity, greater than 60% but less than or equal to 80% sequence identity and greater than 80% sequence identity. We expected that including homology models with low sequence identity to the template structures would improve the ability of the predictor to generalize to external datasets, since both the Protherm and the Skempi datasets are over-represented in proteins that have a crystal structure deposited in the PDB.

We used the "Taipale" dataset, which measures mutation-induced change in protein stability using a chaperone interaction assay, to validate the core predictor, and the "Taipale PPI" and "Taipale GPCA" datasets, which measure mutation-induced changes in protein-protein interactions using yeast

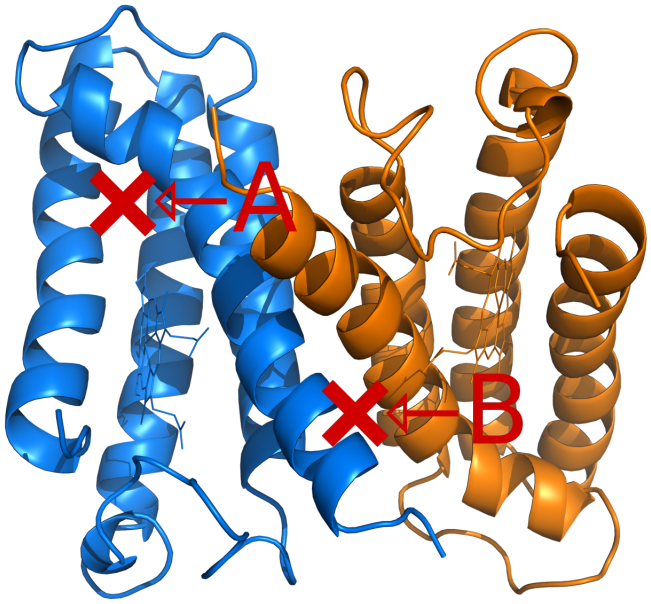


Figure 3.1: Diagram showing an example of a core (A) and an interface (B) mutation. The effect of core mutations is measured as the change in the Gibbs free energy of protein folding. The effect of interface mutations is measured as the change in the Gibbs free energy of protein-protein interaction. The protein depicted in the diagram is ferricytochrome c from *Rhodospirillum molischianum* (PDB entry `2ccy`).

two-hybrid and *Gaussia princeps* luciferase complementation assays, respectively, to validate the interface predictor [45]. We also selected a subset of mutations from the Humsavar [46], ClinVar [47], and COSMIC [48] datasets to validate both core and interface predictors. While mutation deleteriousness and $\Delta\Delta G$ are different metrics, it is expected that deleterious mutations, on average, should have a higher impact on protein structure that benign mutations. Therefore, accurate $\Delta\Delta G$ predictions should have a higher correlation with the deleteriousness score, defined as 1 for deleterious mutations and 0 for benign mutations, than inaccurate predictions.

For our test datasets, we used mutations from the Humsavar, ClinVar, and COSMIC datasets affecting proteins that do not appear in *any* of our training or validation datasets. In addition, we used the "SUMO Ligase" dataset, which measures the effect of mutations on the activity of SUMO Ligase, the "AB-Bind" dataset, which measures the effect of mutations on antibondy binding affinity, and the "Benedix" dataset, which measures the effect of mutations on the affinity between β-lactamase and β-lactamase-inhibitor. In all cases, the dataset for the core predictor was restricted to mutations that do not fall inside a protein-protein interface, and the dataset for the interface predictor was restricted to mutations that do fall inside a protein-protein interface.

The overlap in mutations found in different datasets is presented in Figures 3.2 and 3.3, for core and interface mutations, respectively. We made sure that no mutation from out test sets appears in our training and validation sets.

Table 3.1: Description of the datasets that were used in this study.

Name	Type	Description
Protherm	Train	Mutation-induced changes in the Gibbs free energy of protein folding ($\Delta\Delta G_{core}$) compiled from the Protherm database [49, 50] and from the datasets curated by Kellogg *et al.* [51].
Skempi	Train	Mutation-induced changes in the Gibbs free energy of protein-protein interaction ($\Delta\Delta G_{interface}$) compiled from the SKEMPI database [52] and the dataset curated by Kortemme and Baker [53].
Taipale	Validation	Interaction between chaperones and wildtype or mutant proteins, quantified using the LUMIER assay [45].
Taipale PPI	Validation	Results of yeast two-hybrid experiments, measuring the presence or absence of protein-protein interactions for wild-type and mutant proteins [45].
Taipale GPCA	Validation	*Gaussia princeps* luciferase complementation assay, measuring the effect of mutations on protein affinity [45].
Humsavar	Validation & Test	Disease-causing mutations and polymorphisms obtained from the UniProt *humsavar.txt* file [46]. Mutations annotated with at least one disease are assigned a value of 1. Mutations annotated as "polymorphisms" are assigned a value of 0.
ClinVar	Validation & Test	Disease-causing mutations and polymorphisms obtained from ClinVar [47]. Mutations found in the ClinVar *clinvar_20160531.vcf* file are assigned a value of 1. Mutations found in the ClinVar *common_no_known_medical_impact_20160531.vcf* file are assigned a value of 0.
COSMIC	Validation & Test	Mutations found in cancer [48]. Mutations classified by FATHMM [11] as cancer drivers are assigned a value of 1. Mutations classified by FATHMM as cancer passengers are assigned a value of 0.
SUMO Ligase	Test	Mutations affecting the activity of SUMO ligase, measured using a cell viability assay [54].
AB-Bind	Test	Mutations explored in antibody-antigen affinity maturation experiments [55].
Benedix	Test	Results of alanine scanning experiments of the TEM1 (β-lactamase) – BLIP (β-lactamase-inhibitor) interface [15].

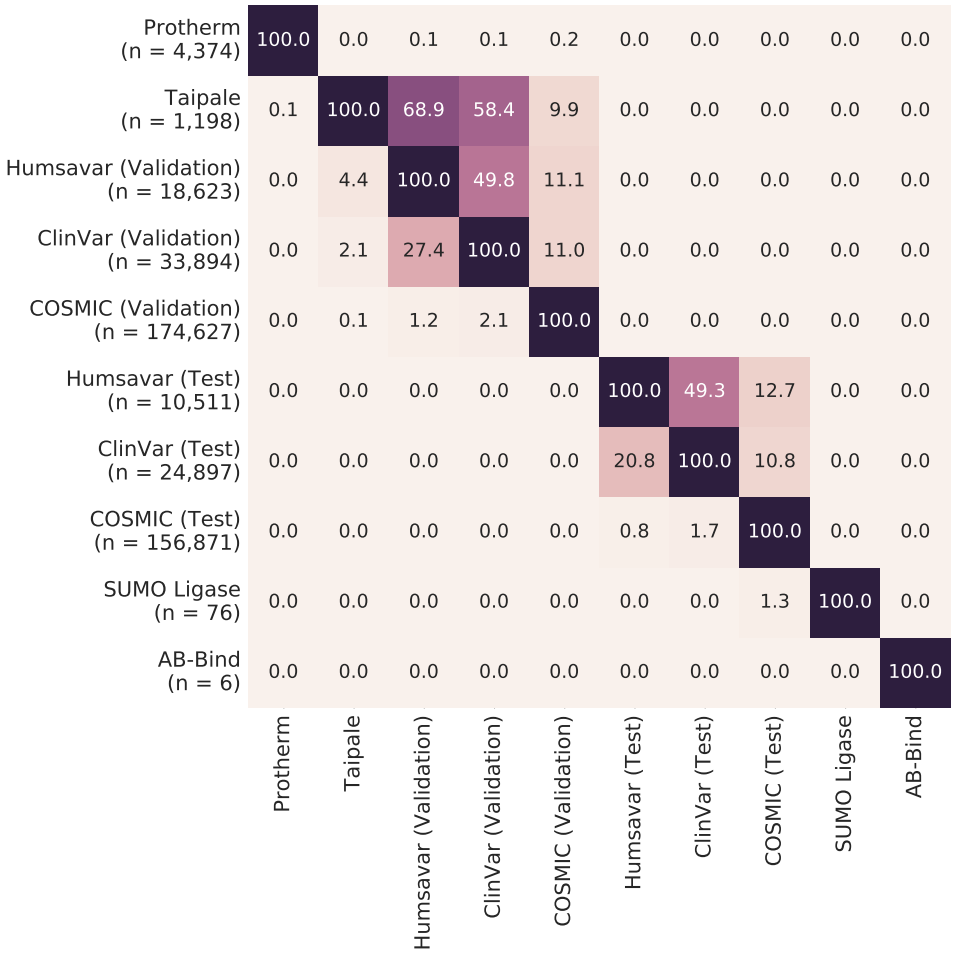


Figure 3.2: Overlap in core mutations between all the datasets used in this study. The shade and value inside each square denotes the percentage of mutations in the dataset named on the y-axis that are also found in the dataset named on the x-axis. Core mutations are defined as mutations that do not occur within 6 Å of a neighbouring chain in the provided PDB structure or protein-protein homology model. A description of each dataset can be found in Table 3.1.

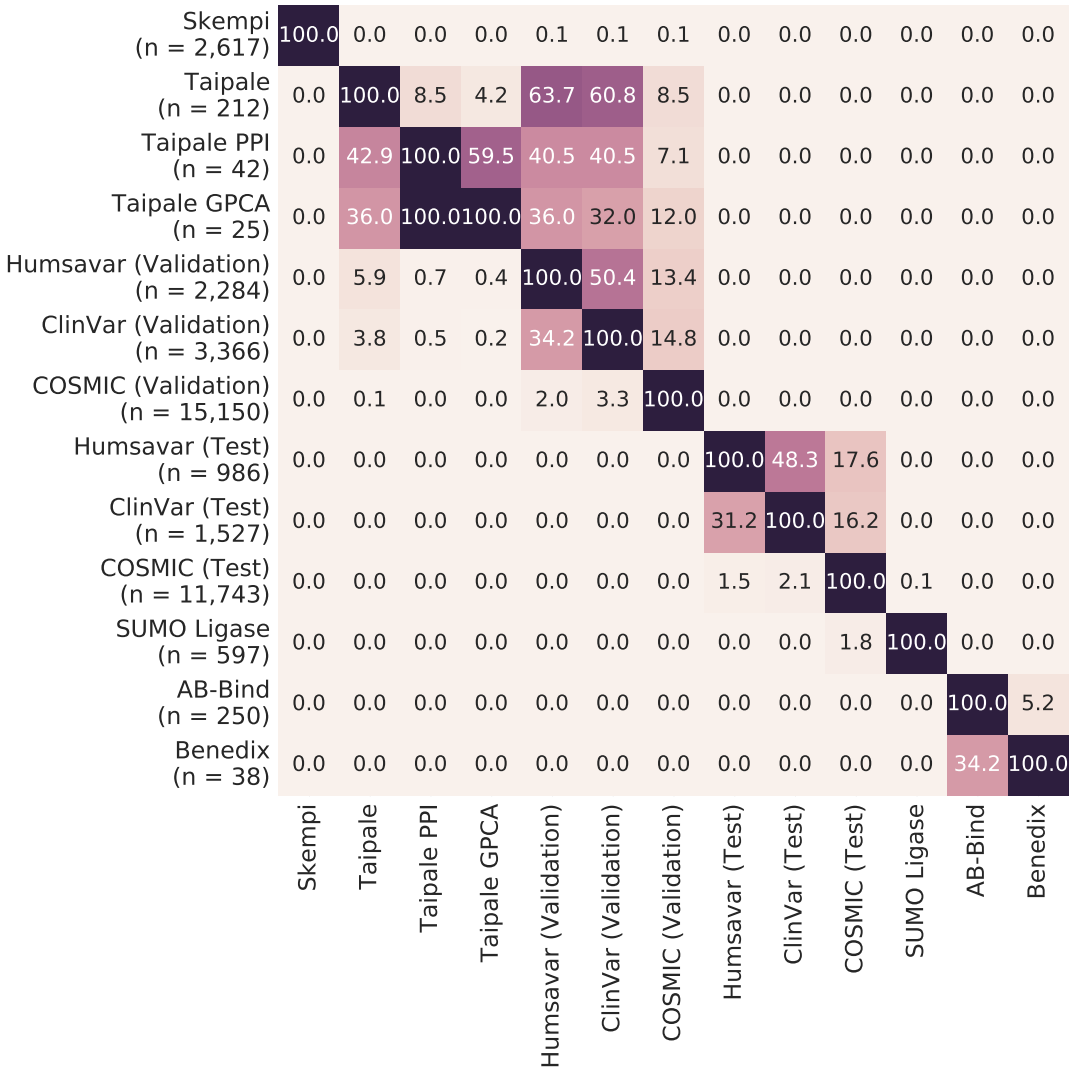


Figure 3.3: Overlap in interface mutations between all the datasets used in this study. The shade and value inside each square denotes the percentage of mutations in the dataset named on the y-axis that are also found in the dataset named on the x-axis. Interface mutations are defined as mutations that occur within 6 Å of a neighbouring chain in the provided PDB structure or protein-protein homology model. A description of each dataset can be found in Table 3.1.

3.2 Hyperparameter optimisation

ELASPIC uses the gradient boosting regressor (GBR) algorithm, implemented in scikit-learn [40], to combine over 70 different sequential and structural features and predict the change in the Gibbs free energy change of protein folding or protein-protein interaction. The GBR algorithm was selected because it achieved higher performance than linear regression, support vector machine and random forest algorithms, in 20-fold cross-validation over the training set [21].

In order to select the best set of GBR hyperparameters, we performed exhaustive "grid-search", where we measured the performance of the GBR algorithm for 3,600 different combinations of hyperparameters (Table 3.2). For each set of hyperparameters, we computed the Spearman correlation between predicted and experimental $\Delta\Delta G$ values for mutations in the training set, using 4-fold cross-validation. We also computed the Spearman correlation between predicted $\Delta\Delta G$ values and the experimental values recorded for our validation datasets. We used the combined scores CS_{core} (Equations 3.1) and $CS_{interface}$ (Equation 3.2) to select the best set of hyperparameters for the core and interface predictors, respectively. The contribution of each dataset to the combined score was assigned in an "ad-hoc" manner, giving more weight to large datasets than to small datasets, and making sure that the performance on energy-based datasets, including training and Taipale datasets, has a bigger overall impact on the combined score than performance on a deleteriousness-based datasets, such as Humsavar, ClinVar and COSMIC. We used combined scores instead of training set cross-validation alone because we wanted to select predictors that generalize well to external datasets. Since our training sets are limited and biased in the number and type of proteins and protein-protein interactions that they contain, the performance of the predictors on the training set may not be an accurate indicator of their performance in general. Our validation datasets contain many more distinct proteins and protein-protein interactions than our training datasets, and therefore the performance on the validation datasets should be indicative of how well the predictors generalize to other proteins in the human genome.

$$CS_{core} = \frac{3 \cdot Cross\_validation + Humsavar + ClinVar + COSMIC + Taipale}{7} \tag{3.1}$$

$$CS_{interface} = \frac{3 \cdot Cross\_validation + Humsavar + ClinVar + COSMIC + \frac{Taipale\_PPI}{4} + \frac{Taipale\_GPCA}{4}}{6.5} \tag{3.2}$$

In Figures 3.4 and 3.5, we show the performance of core and interface predictors, for different sets of hyperparameters, sorted by the combined score. For both the core and interface predictors, the performance of the predictor on the training dataset, measured through cross-validation, is highly correlated with its performance on the validation datasets. However, selecting hyperparameters solely based on cross-validation performance would result in a predictor that substantially underperforms on the validation datasets.

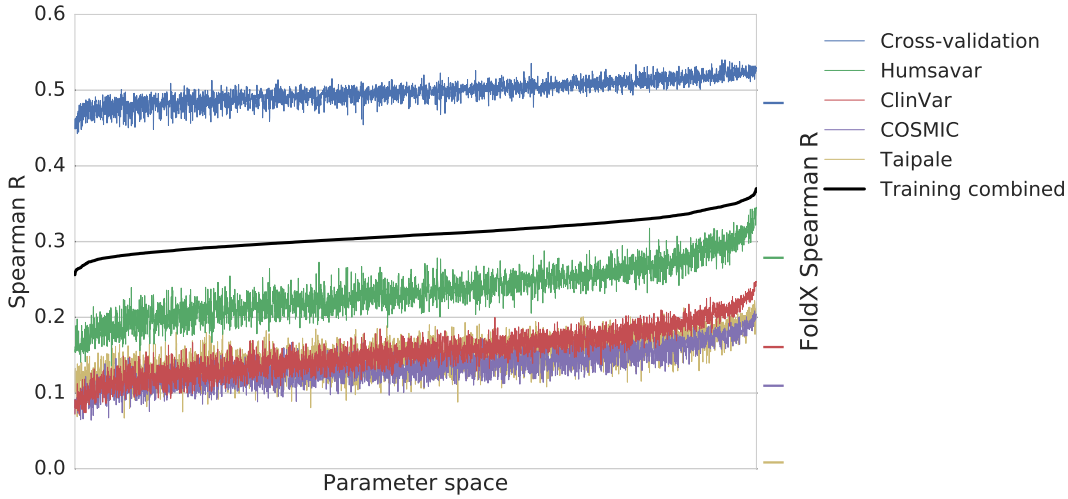


Figure 3.4: Core predictor hyperparameter optimization. The combined score (black line) was calculated using Equation 3.1. We chose the set of hyperparameters that correspond to the predictor with the highest combined score.

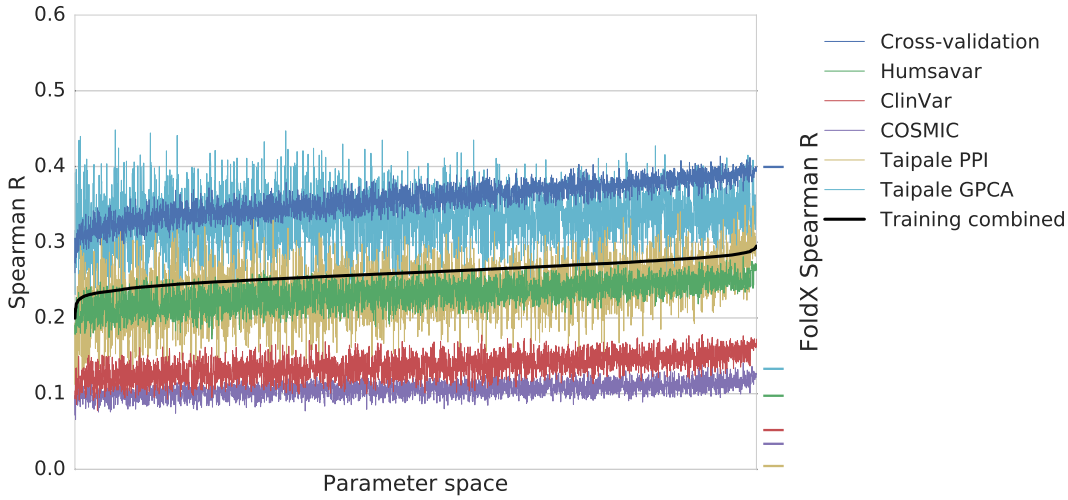


Figure 3.5: Interface predictor hyperparameter optimization. The combined score (black line) was calculated using Equation 3.2. We chose the set of hyperparameters that correspond to the predictor with the highest combined score.

Table 3.2: Hyperparameter search space for tuning the gradient boosting regressor algorithm used in the core and interface predictors. An all-by-all combination of those hyperparameters was explored in order to find the sets of hyperparameters that produce the best-performing predictors.

Parameter name	Parameter value
alpha	0.99, 0.95, 0.9, 0.8, 0.7, 0.5
learning_rate	0.1, 0.05, 0.02, 0.01
loss	huber
max_depth	10, 8, 6, 4
max_features	1.0, 0.8, 0.5, 0.3, 0.1,
min_samples_leaf	29, 21, 17, 13, 9, 5, 3
n_estimators	2000

Table 3.3: Hyperparameters selected for the core predictor.

Parameter name	Parameter value
alpha	0.5
learning_rate	0.01
loss	huber
max_depth	4
max_features	0.246
min_samples_leaf	17
n_estimators	2000

Table 3.4: Hyperparameters selected for the interface predictor.

Parameter name	Parameter value
alpha	0.9
learning_rate	0.01
loss	huber
max_depth	4
max_features	0.3
min_samples_leaf	13
n_estimators	2000

3.3 Feature elimination

After selecting the best set of hyperparameters for core and interface predictors, we performed feature elimination to evaluate the contribution of each feature to the overall accuracy of the predictor, and to select the sets of features that result in the most accurate predictions.

Feature elimination was performed using the following recursive strategy:

- Leave out each feature from the training set, one at a time.
- Train the predictor using all but the left out feature.
- Calculate the combined score (CS_{core} or $CS_{interface}$) evaluating the performance of the predictor.
- Discard the feature that, when left out, produced the predictor with the highest combined score.
- Repeat the process until only one feature remains.

Performances of the core and interface predictors at every step of feature elimination are shown in Figures 3.6 and 3.7. The sets of features that produced the best-performing predictors are described in Tables 3.5 and 3.6.

Most features play a surprisingly small role in the performance of the ELASPIC predictor. In fact, we can achieve near-optimal performance with both core and interface predictors by using only 6 features (displayed in bold in Tables 3.5 and 3.6). This suggests either that most features are not informative in predicting the energetic effect of mutations, or that the training set is too noisy for the contribution of those features to make a significant impact on the accuracy of the predictor.

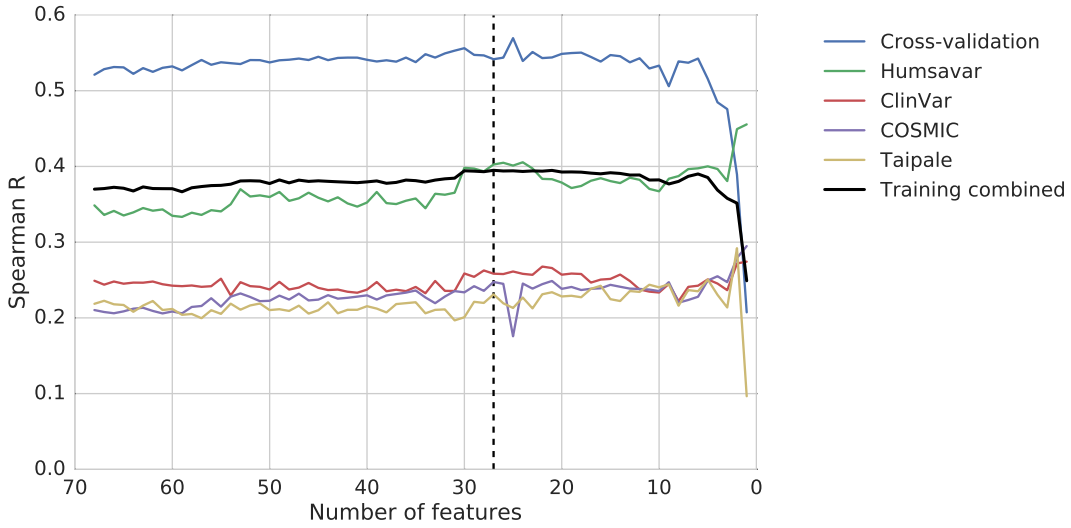


Figure 3.6: Performance of the core predictor at each step of feature elimination. The combined score (black line) was calculated using Equation 3.1. Predictor with the highest combined score is indicated by the vertical dashed line, and the features used to train that predictor are listed in Table 3.5.

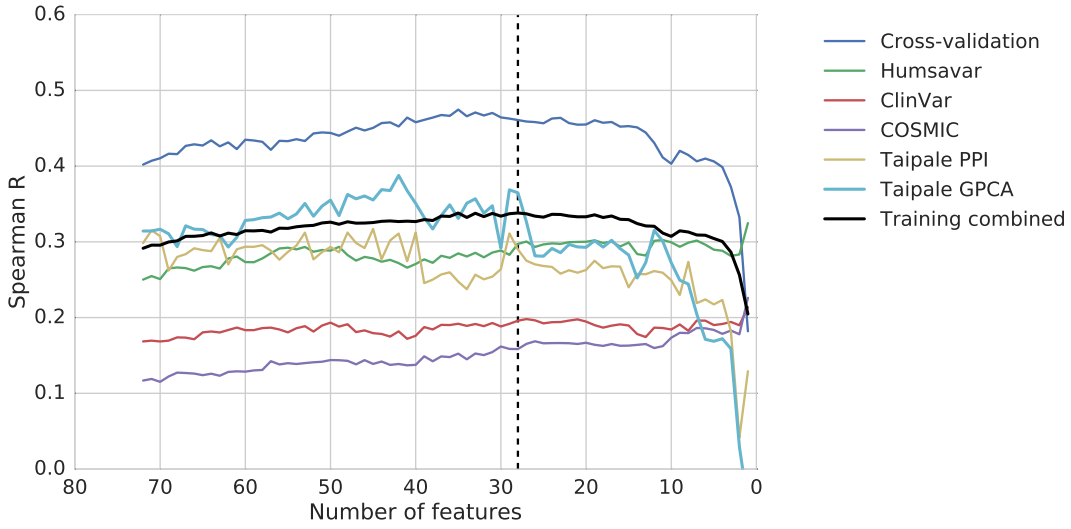


Figure 3.7: Performance of the interface predictor at each step of feature elimination. The combined score (black line) was calculated using Equation 3.2. Predictor with the highest combined score is indicated by the vertical dashed line, and the features used to train that predictor are listed in Table 3.6.

Table 3.5: Core predictor features. Features that end in *_wt* were computed for the wild-type structure. Features that end in *_change* correspond to the difference between the values computed for the wild-type and mutant structures. Rows in bold mark the 6 most important features. FoldX feature descriptions were taken from `http://foldxsuite.crg.eu/command/Stability`.

Feature name	Feature source	Feature description
alignment_coverage	ELASPIC	Alignment coverage.
alignment_identity	ELASPIC	Alignment sequence identity.
alignment_score	ELASPIC	Alignment quality (Equation 2.1).
backbone_hbond_change	FoldX	Backbone hydrogen bond energy.
backbone_hbond_wt	FoldX	Backbone hydrogen bond energy.
cis_bond_wt	FoldX	Cis peptide bond energy.
disulfide_wt	FoldX	Disulfide bond energy.
electrostatic_kon_change	FoldX	Electrostatic interaction between molecules in the pre-complex.
electrostatics_change	**FoldX**	**Electrostatic interactions.**
entropy_mainchain_change	FoldX	Entropy cost of fixing the main chain.
helix_dipole_wt	FoldX	Electrostatic contribution of the helix dipole.
matrix_score	ELASPIC	BLOSUM62 matrix score.
pcv_hbond_change	ELASPIC	Hydrogen-oxygen contacts involving atoms of the mutated residue and water molecules present in the crystal structure.
pcv_hbond_self_change	ELASPIC	Hydrogen-oxygen contacts involving atoms of the mutated residue and atoms of the mutated chain.
pcv_salt_equal_change	ELASPIC	Charge repulsions between atoms of the mutated residue and ions and heteroatoms present in the crystal structure.
pcv_salt_equal_self_wt	ELASPIC	Charge repulsions between atoms of the mutated residue and atoms of the mutated chain.
pcv_salt_equal_wt	ELASPIC	Charge repulsions between atoms of the mutated residue and ions and heteroatoms present in the crystal structure
pcv_salt_opposite_change	ELASPIC	Charge attractions between atoms of the mutated residue and ions and heteroatoms present in the crystal structure.
pcv_vdw_self_change	ELASPIC	Carbon carbon contacts between atoms of the mutated residue and atoms of the mutated chain.
provean_score	**Provean**	**Sequence conservation score.**
sloop_entropy_wt	FoldX	Entropic cost according to the SLoop database of loop conformations.
solvation_hydrophobic_change	**FoldX**	**Contribution of hydrophobic groups.**
solvation_polar_change	**FoldX**	**Energetic penalty for burying polar groups.**
solvent_accessibility_wt	**MSMS**	**Solvent-accessible surface area of the mutated residue.**
torsional_clash_change	FoldX	Intra-residue van der Waals torsional clashes.
van_der_waals_clashes_change	**FoldX**	**Energy penalization due to van der Waals clashes (interresidue).**
water_bridge_wt	FoldX	Contribution of water bridges.

Table 3.6: Interface predictor features. Features that end in *_wt* were computed for the wild-type structure. Features that end in *_change* correspond to the difference between the values computed for the wild-type and mutant structures. Rows in bold mark the 6 most important features. FoldX feature descriptions were taken from `http://foldxsuite.crg.eu/command/AnalyseComplex`.

Feature name	Feature source	Feature description
alignment_score	ELASPIC	Alignment quality (Equation 2.2).
backbone_clash_change	FoldX	Backbone-backbone van der Waals energy.
backbone_clash_wt	FoldX	Backbone-backbone van der Waals energy.
backbone_hbond_change	FoldX	Backbone hydrogen bond energy.
cis_bond_wt	FoldX	Cis peptide bond energy.
electrostatic_kon_wt	FoldX	Electrostatic interaction between molecules in the pre-complex.
energy_ionisation_wt	FoldX	Ionization energy.
entropy_complex_change	FoldX	Entropic cost of forming a complex.
entropy_sidechain_change	**FoldX**	**Entropic cost of fixing the side chain.**
intraclashes_energy_2_change	FoldX	van der Waals clashes of residues at the interface of the complex.
partial_covalent_bonds_wt	**FoldX**	**Interactions with bound metals.**
pcv_hbond_self_change	ELASPIC	Hydrogen-oxygen contacts involving atoms of the mutated residue and atoms of the mutated chain.
pcv_hbond_wt	ELASPIC	Hydrogen-oxygen contacts involving atoms of the mutated residue and atoms of the interacting chain.
pcv_salt_equal_self_change	ELASPIC	Charge repulsions involving atoms of the mutated residue and atoms of the mutated chain.
pcv_salt_equal_wt	ELASPIC	Charge repulsions involving atoms of the mutated residue and atoms of the interacting chain.
pcv_salt_opposite_change	ELASPIC	Charge attractions involving atoms of the mutated residue and atoms of the interacting chain.
pcv_salt_opposite_self_change	ELASPIC	Charge attractions involving atoms of the mutated residue and atoms of the mutated chain.
pcv_salt_opposite_self_wt	ELASPIC	Charge attractions involving atoms of the mutated residue and atoms of the mutated chain.
pcv_vdw_self_change	ELASPIC	Carbon carbon contacts involving atoms of the mutated residue and atoms of the mutated chain.
pcv_vdw_self_wt	**ELASPIC**	**Carbon carbon contacts involving atoms of the mutated residue and atoms of the mutated chain.**
pcv_vdw_wt	**ELASPIC**	**Carbon carbon contacts involving atoms of the mutated residue and atoms of the interacting chain.**
provean_score	**Provean**	**Sequence conservation score.**
sloop_entropy_change	FoldX	Entropic cost according to the SLoop database of loop conformations.
solvation_hydrophobic_change	FoldX	Contribution of hydrophobic groups.
solvation_polar_change	**FoldX**	**Energetic penalty for burying polar groups.**
solvation_polar_wt	FoldX	Energetic penalty for burying polar groups.
torsional_clash_change	FoldX	Intra-residue van der Waals torsional clashes.
water_bridge_change	FoldX	Contribution of water bridges.

3.4 Validation

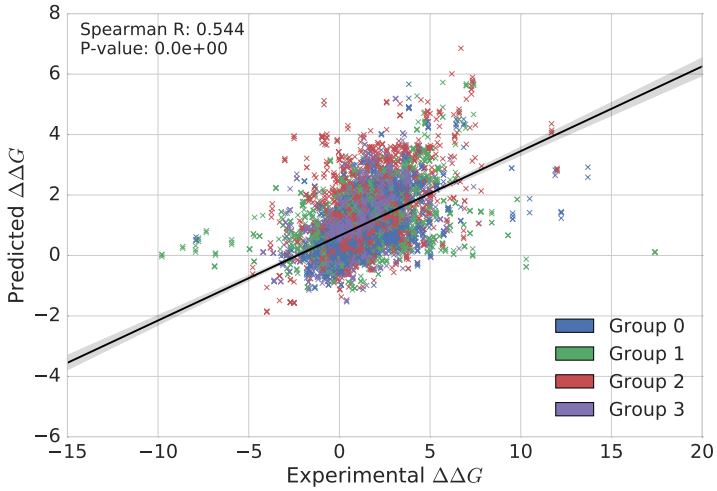
3.4.1 Performance on the training, validation and test datasets

The final ELASPIC core and interface predictors were trained using the best set of hyperparameters (Section 3.2) and the best set of features (Section 3.3) that were found for each predictor. Performance of the predictors on the training, validation and test datasets is shown in Figures 3.8 and 3.9, for core and interface predictors, respectively.

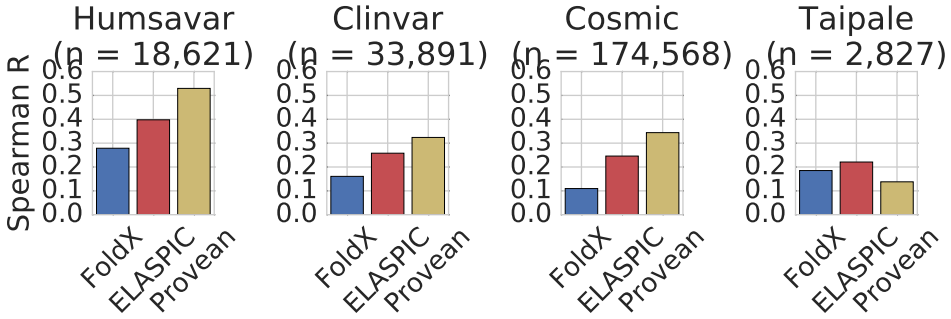
The ELASPIC core predictor outperforms FoldX and Provean on the Taipale dataset, which is the only validation dataset explicitly measuring the effect of mutations on protein stability rather than mutation deleteriousness (Figure 3.8b). It also outperforms FoldX and Provean on the core subsets of the SUMO and AB-Bind test datasets (Figure 3.8c). The core subsets of those datasets only contain mutations located more than 6 Å away from another chain in the PDB.

The ELASPIC interface predictor also outperforms FoldX and Provean on the Taipale GPCA dataset (Figure 3.9b). It performs marginally worse than Provean on the Taipale PPI dataset, but this is likely because the Taipale PPI dataset contains many known deleterious mutations, which are predicted well by Protherm. The ELASPIC interface predictor outperforms Protherm and FoldX on the SUMO Ligase and AB-Bind test datasets (Figure 3.9c) both alone and in combination with the core predictor. The ELASPIC interface predictor shows slightly lower performance than FoldX on the very small Benedix dataset.

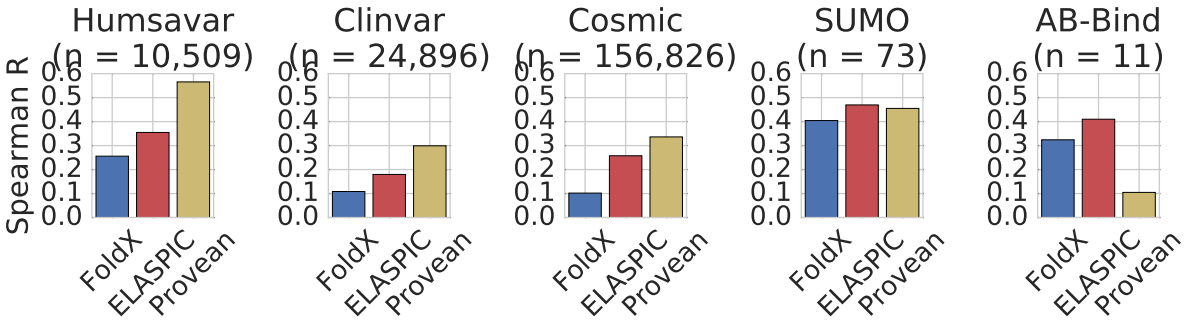
Both the core and interface predictors performs better than FoldX but worse than Provean on the validation and test subsets of the Humsavar, ClinVar and COSMIC (Figures 3.8c and 3.9c). The low performance of the ELASPIC compared to Provean can be justified, since ELASPIC attempts to model the effect of mutations on protein stability or protein-protein interaction affinity, and does not take into account other reasons that a mutation may be deleterious. For example, mutations can affect the active site or the signal sequence of a protein, which may prove to be highly deleterious to the organism but would have only a marginal effect on protein stability.



(a) Performance of the ELASPIC core predictor on the training dataset, evaluated using four-fold cross-validation. Colours indicate different cross-validation bins.

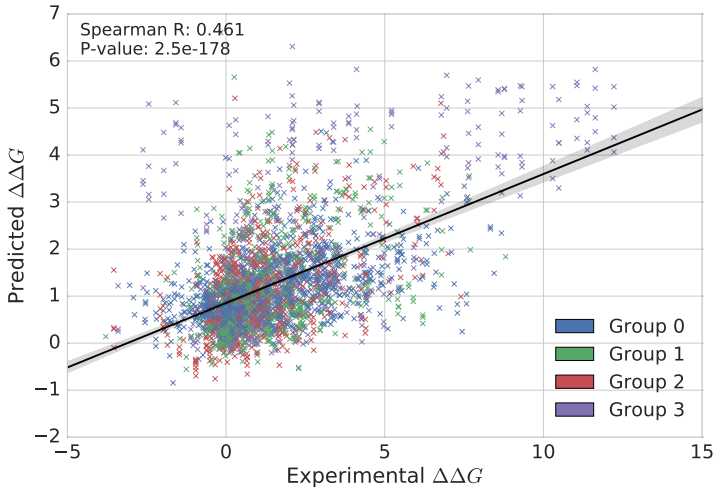


(b) Performance of the ELASPIC core predictor, FoldX and Provean on the validation datasets.

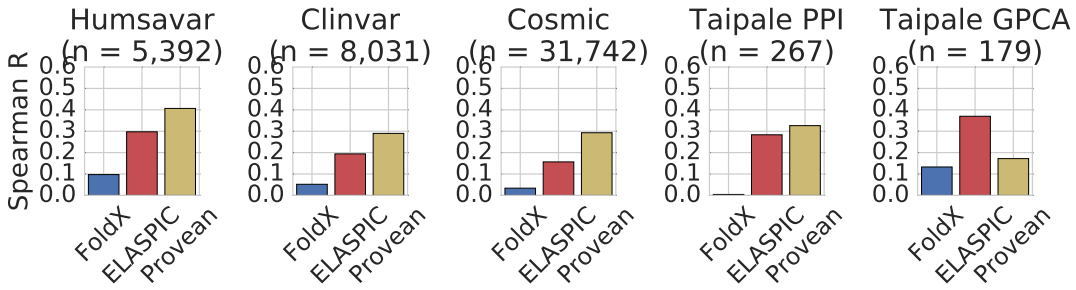


(c) Performance of the ELASPIC core predictor, FoldX and Provean on the test datasets. There is no overlap in mutations (or proteins for Humsavar, ClinVar and COSMIC) between the test datasets, and the training and validation datasets (see Figure 3.2).

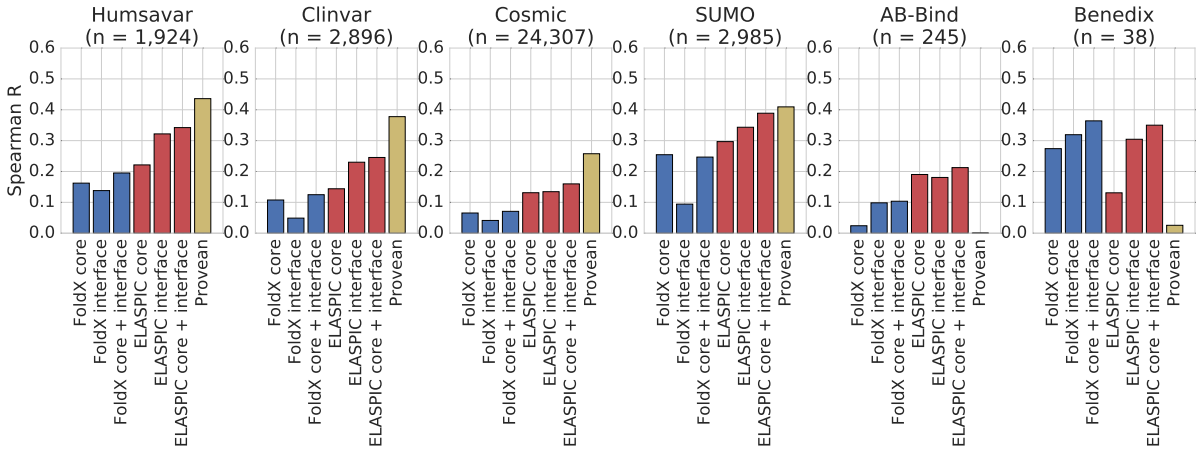
Figure 3.8: Performance of the ELASPIC core predictor on the training (a), validation (b) and test (c) datasets.



(a) Performance of the ELASPIC interface predictor on the training dataset, evaluated using four-fold cross-validation. Colours indicate different cross-validation bins.



(b) Performance of the ELASPIC interface predictor, FoldX and Provean on the validation datasets.

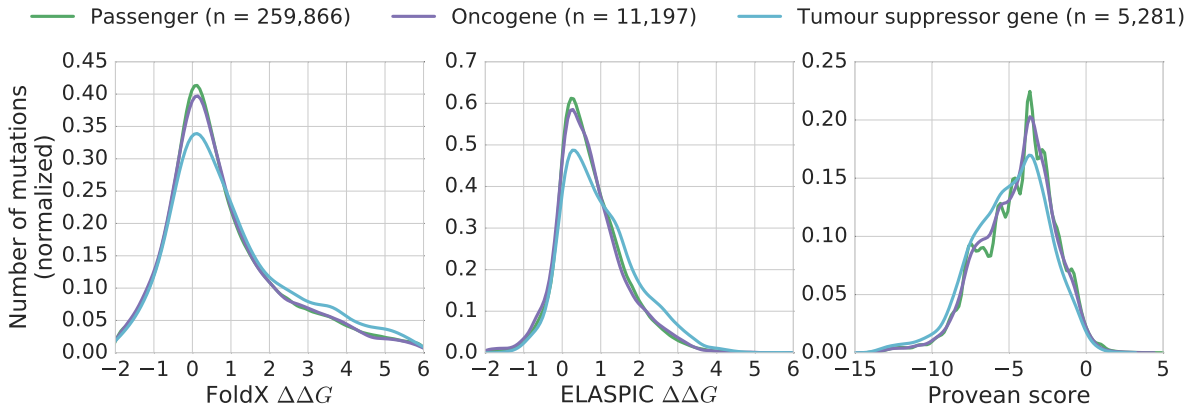


(c) Performance of ELASPIC, FoldX and Provean on the test datasets. Correlations are provided for core predictor outputs, interface predictor outputs, and the sum of core and interface predictor outputs, for ELASPIC and FoldX. There is no overlap in mutations (or proteins for Humsavar, ClinVar and COSMIC) between the test datasets, and the training and validation datasets (see Figure 3.3).

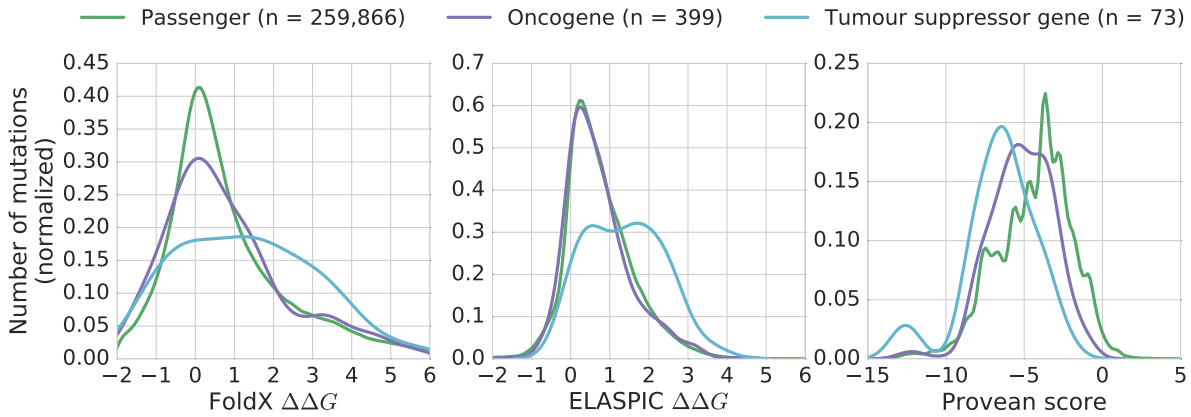
Figure 3.9: Performance of the ELASPIC interface predictor on the training (a), validation (b) and test (c) datasets.

3.4.2 Distinguishing gain-of-function and loss-of-function mutations

We explored the ability of FoldX, ELASPIC and Provean to differentiate between gain-of-function and loss-of-function mutations by comparing the output of the tools for cancer driver mutations falling inside oncogenes and tumour suppressor genes, respectively (Figures 3.10 and 3.11). We considered two sets of cancer driver mutations: i) mutations in the COSMIC database that are predicted to be cancer drivers by FATHMM, and ii) mutations in the database of curated mutations (DoCM) [56] that are known to be cancer drivers through *in vitro* and *in vivo* experiments. As a set of oncogenes and a set of tumour suppressor genes, we used genes in the cancer gene census database [57] that are marked as dominant or recessive, respectively, excluding genes that appear in the Protherm or Skempi datasets. We expected tools that predict the deleteriousness of mutations, such as Provean, to see mutations falling inside oncogenes and tumour suppressor genes as equally deleterious, since both types of mutations can lead to cancer. Conversely, we expected tools that predict the structural impact of mutations, such as FoldX and ELASPIC, to see mutations falling inside tumour suppressor genes as more destabilizing than mutations falling inside oncogenes, since the former, but not the latter, destroy the function of the protein. In fact, FoldX, ELASPIC and Provean all predict that mutations falling inside tumour suppressor genes are more destabilizing or deleterious than mutations falling inside oncogenes (Figure 3.10). The effect is more pronounced for curated driver mutations in DoCM (Figure 3.10b) than for predicted driver mutations in the COSMIC database (Figure 3.10a), likely because the COSMIC dataset contains many passenger mutations that are falsely predicted to be drivers. ELASPIC $\Delta\Delta G$ is the most informative score for predicting whether a mutation falls inside a tumour suppressor gene or an oncogene, as indicated by the areas under the receiver operating characteristic curves (Figures 3.11a and 3.11c). However, the advantage of ELASPIC over Provean largely disappears when we evaluate the ability of the tools to predict whether the *protein* is an oncogene or a tumour suppressor gene, using the mean of the scores for all mutations falling in that protein (Figures 3.11b and 3.11d). This suggests that ELASPIC is better than Provean at scoring mutations in proteins for which both tools make relatively good predictions, but it is not better than Provean for all other mutations.

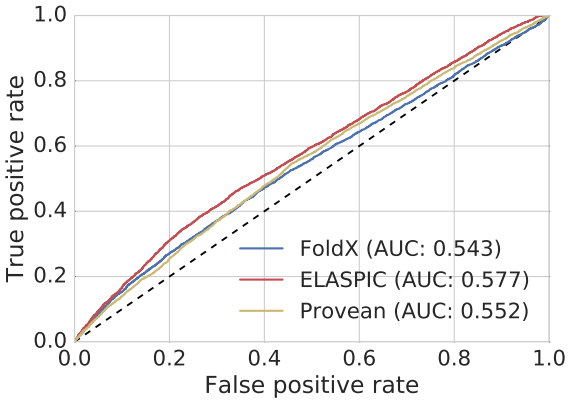


(a) Distribution of scores produced by FoldX (left), ELASPIC (middle) and Provean (right) for predicted cancer driver mutations in the COSMIC database. For all three programs, the scores obtained for mutations falling inside oncogenes are significantly different from the scores obtained for mutations falling inside tumour suppressor genes, according to the Mann–Whitney U test (FoldX p-value $< 1 \times 10^{-18}$, ELASPIC p-value: $< 1 \times 10^{-56}$, Provean p-value: $< 1 \times 10^{-25}$).

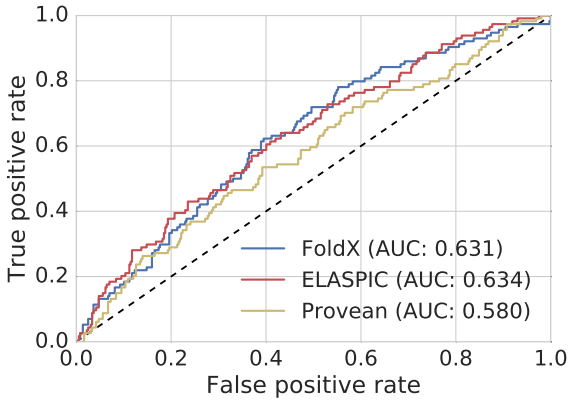


(b) Distribution of scores produced by FoldX (left), ELASPIC (middle) and Provean (right) for currated cancer driver mutations in DoCM. For ELASPIC and Provean, the scores obtained for mutations falling inside oncogenes are significantly different from the scores obtained for mutations falling inside tumour suppressor genes, according to the Mann–Whitney U test (FoldX p-value > 0.05, ELASPIC p-value: $< 1 \times 10^{-7}$, Provean p-value: $< 1 \times 10^{-5}$).

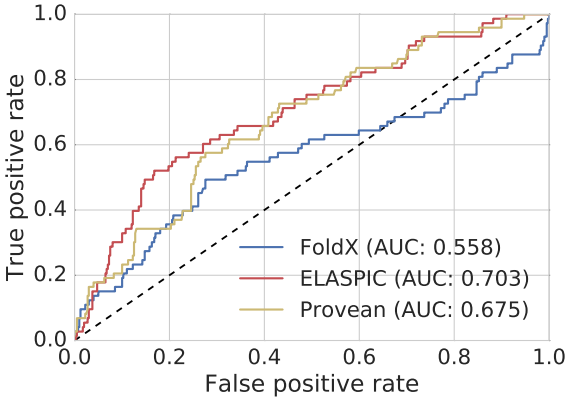
Figure 3.10: Distribution of scores produced by FoldX, ELASPIC and Provean for mutations in the COSMIC database predicted to be cancer drivers by FATHMM (a) and for mutations in the database of curated mutations (DoCM) known to be cancer drivers through *in vitro* and *in vivo* experiments.



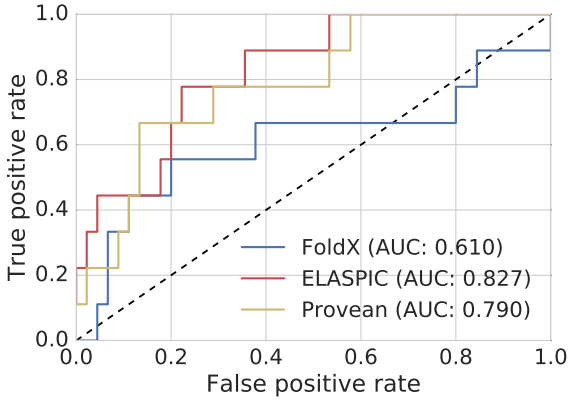
(a) Predicting whether a *mutation* falls inside an oncogene or a tumour suppressor gene, for all deleterious mutations in COSMIC.



(b) Predicting whether a *protein* is an oncogene or a tumour suppressor gene, using all deleterious mutations in the COSMIC database.



(c) Predicting whether a *mutation* falls inside an oncogene or a tumour suppressor gene, for all deleterious mutations in DoCM [56].



(d) Predicting whether a *protein* is an oncogene or a tumour suppressor gene, using all deleterious mutations in DoCM [56].

Figure 3.11: Receiver operating characteristic (ROC) curves showing the performance of FoldX, ELASPIC and Provean in predicting whether a *mutation* falls inside an oncogene or a tumour suppressor gene (a, c), or whether the *entire protein* is an oncogene or a tumour suppressor gene (b, d), using all deleterious mutations in the COSMIC database (a, b) or curated driver mutations in the DoCM database [56] (c, d). When predicting whether the entire protein is an oncogene or a tumour suppressor gene, we used the mean score for all mutations falling in that protein.

Chapter 4

Discussion

The primary objective of this project was to extend ELASPIC and make it possible to predict the effect of mutations on protein stability and protein-protein interaction affinity on a genome-wide scale. We were able to meet this objective with a reasonable amount of success. We implemented ELASPIC as an easy to install and fully automated pipeline, which can scale from a single mutation in a user-provided PDB file to hundreds of thousands of mutation affecting most proteins in the genome. We calculated $\Delta\Delta G$ values for almost one million mutations implicated in human diseases or found in cancers, which, to our knowledge, makes this the first study to evaluate the structural impact of mutations at such a large scale. It is now possible to use ELASPIC as part of a toolbox for annotating variants discovered through high-throughput sequencing, since evaluating a mutation takes under several minutes if the Provean supporting set and homology models have been precalculated. With the work of Daniel Witvliet *et al.* [27], it is also possible to use ELASPIC through a webserver.

The secondary objective of this project was to use structural information provided by ELASPIC to make accurate and informative predictions regarding the phenotypic effect of mutations. This objective was met with much less success. Provean remains more accurate than ELASPIC in predicting whether a mutation is associated with a disease, as seen in the performance of the two tools on the validation and test subsets of the Humsavar, ClinVar and COSMIC datasets (Figures 3.8 and 3.9). ELASPIC does achieves better performance than Provean in differentiating between loss-of-function and gain-of-function mutations (Figures 3.11a and 3.11c). However, the difference in performance is not substantial, and it does not improve our ability to distinguish between tumour suppressor genes and oncogenes (Figures 3.11b and 3.11d). The sequence conservation score already take into account most deleterious effects of mutations, including changes in protein stability and protein-protein interaction affinity, as well as other effects that not considered by ELASPIC, such as the disruption of active sites, ligand binding pockets and residues involved in cooperative interactions that regulate the activity of the protein. This is consistent with a recently published study, where the authors trained a machine learning algorithm called Variant Interpretation and Prediction Using Rosetta (VIPUR) to predict the deleteriousness of mutations using the Provean score, $\Delta\Delta G$ predicted by Rosetta and other structural and sequential features [19]. VIPUR achieves an area under the receiver operating characteristic (ROC) curve of 0.831, while Provean achieves an area under the ROC curve of 0.819 on the same dataset. Overall, it seems that the additional information provided by tools such as ELASPIC and VIPUR does not justify the extra complexity and computational costs required to evaluate the structural impact of mutations.

Several observations made during the training and validation of ELASPIC predictors warrant further discussion. We found that mutations falling inside protein-protein interfaces are more likely to be involved in disease, but the effect size that we observe is much smaller than what was reported previously. In the case of the UniProt humsavar database, we observe that 8.4% of deleterious mutations and 3.5% of polymorphic mutations fall inside protein-protein interfaces. If we restrict polymorphic mutations to the set of proteins also affected by deleterious mutations, the value increases to 4.3%, likely because proteins implicated in disease are better studied, have more structural templates and, consequently, have more homology models of protein-protein interaction. In contrast, a previous report indicates that 57% of disease mutations and 8% of polymorphic mutations disrupt protein-protein interactions [45]. It is likely that we underestimate the fraction of disease and polymorphic mutations that affect protein-protein interaction, since our structural coverage of the protein-protein interaction network is very incomplete. However, the discrepancy between the 2-fold over-representation of interface mutations in disease that we observe in our work, an the 7-fold over-representation of interface mutations in disease that are reported by Sahni *et al.*, should be examined further. One possible reason for the discrepancy is a bias in the experiments performed by Sahni *et al.*. For example, the HI-II-14 human interactome map used in yeast two-hybrid experiments may be enriched in protein-protein interactions that are involved in disease, making it appear that disease mutations break more protein-protein interactions than polymorphisms. Also, Sahni *et al.* report that disease mutation tend to alter the protein-chaperone interaction profile much more than polymorphic mutations, and increased interaction with chaperones could block some of the protein interaction interfaces, preventing native protein-protein interactions from occurring. Another explanation for the discrepancy could be that we create inaccurate homology models for many of the protein-protein interactions, and therefore do not capture accurately the effect of disease mutations on those interactions.

We found that interface mutations affecting proteins involved in multiple protein-protein interactions are usually "edgetic", in that they affect only one of the interactions. For example, in the case of proteins in the UniProt humsavar database that are involved in multiple interactions, 64% of deleterious mutations and 56% of polymorphic mutations affect only one of the interactions, and less than 1% of deleterious and polymorphic mutations affect all interactions. In the case of mutations affecting multiple interactions, we found that the average $\Delta\Delta G$ caused by a mutation is a better predictor of mutation deleteriousness than the difference in $\Delta\Delta G$ between the interface that is the most affected and the interface that is the least affected. This suggests that the "edgetic" nature of disease mutations is more due to the fact that proteins use distinct and non-overlapping interfaces to interacting with different partners, than due to an inherent propensity of those mutations to be selective in the interactions that they disrupt.

Results of feature elimination suggest that most features calculated by ELASPIC are either highly correlated or contain little information regarding protein stability or protein-protein interaction affinity (Figures 3.6 and 3.7). We do observe a consistent trend in that, for both core and interface predictors, the features that remain at the end of feature elimination include an electrostatic term, a van der Waals term, a solvation term and an entropic term (see Tables 3.5 and 3.6). This is consistent with a previous study by Benedix *et al.* [15], where the authors achieve good accuracy in predicting the effect of mutations on protein stability and protein-protein interaction affinity using only five energy terms: i) electrostatic energy between charged residues, calculated using Coulomb's law, ii) van der Waals energy between residues, calculated using the Lennard Jones equation, iii) solvation energy of polar residues, calculated by solving the Poisson-Boltzmann equation, iv) solvation energy of hydrophobic residues,

calculated as a linear function of the solvent accessible surface area, and v) entropic energy, calculated using the quasiharmonic approximation proposed by Schlitter [58]. This suggests that we could replace FoldX, which we currently use as a "black box" to calculate many of the ELASPIC features, with the energy terms described by Benedix *et al.*, while maintaining the same performance. This would grant us much better control over the ELASPIC pipeline, since we would know exactly how each feature is calculated and would be able to analyze in detail why ELASPIC makes incorrect predictions for some of the mutations.

Benedix *et al.* [15] also report that the accuracy of the predictions can be substantially improved by generating an ensemble of structures using CONCOORD, and using the average of the energies calculated for the wildtype and mutant structures to make the final prediction. They use the same dataset to fit the four parameters of their model, and to test their model, so it is difficult to compare directly the accuracy that they report to the accuracy of ELASPIC. Nevertheless, when using an ensemble of 300 structures, Benedix *et al.* achieve a Pearson correlation coefficient of 0.75 for core mutations and 0.79 for interface mutations, which is likely higher than what would be achieved by ELASPIC on the same dataset (see Figures 3.8a and 3.9a). It seems probable that the performance of ELASPIC could likewise be improved by averaging the energy terms calculated for an ensemble of protein conformations. The downside of this approach would be increased computational cost, although it should still be much faster than alchemical techniques such as thermodynamic integration, while achieving similar performance [59].

As discussed above, using structural information to assist with the prediction of the phenotypic effects of mutations offers little gain in accuracy and incurs a significant computational cost. On the other hand, using sequential information to assist with the prediction of the structural effects of mutations offers a significant gain in accuracy while adding relatively little computational cost. Case in point, the use of the Provean score is likely the main reason why ELASPIC can consistently outperform FoldX on different training, validation and test datasets. Calculating the Provean score can be done in seconds, if the Provean supporting set is available, and in about the same amount of time that it takes to create a homology model, if the Provean supporting set is not available. The Provean score appears to contain information that is distinct and complementary to the information contained by energy terms. While a mutation that has little effect on the structure of the protein may still be deleterious, in the majority of cases, a mutation is deleterious either because it disrupts the stability of the protein or because it disrupts the stability of interactions involving the protein. In those cases, the Provean score takes into account the effect of the mutation on all functional forms of the protein, rather than on a single snapshot that is the crystal structure or the homology model. Thus, the most promising avenue for future work appears to be in adding more sequential features to further improve the accuracy of the ELASPIC predictors. The performance of ELASPIC core and interface predictors on the training sets is highly correlated with their performance on the validation sets, as was shown in Sections 3.2 and 3.3. Since the validation sets contain 10 to 100 times more mutations than the training sets, and are much more diverse, they could contribute much information to the ELASPIC predictors. Some possible ways in which this could expand and incorporate more information into the ELASPIC training set are discussed in Chapter 5.

Chapter 5

Future directions

5.1 Predicting phenotypes

Train on ligand-directed signalling to try to predict the effect of drugs on receptors [60]. "Our longterm goal is to be able to predict proliferative or antiproliferative activity of a ligand in different tissues from its crystal structure by identifying different structural perturbations that lead to specific signaling outcomes."

5.2 Multitask learning of mutation deleteriousness and energetic effects

In this work, we attempted to improve the generalizability of ELASPIC core and interface predictors by keeping track of their performance on mutation deleteriousness datasets throughout cross-validation and feature elimination (Figures 3.4, 3.5, 3.6 and 3.7). While this approach allows us to discard predictors that overfit the training set, it does not improve the accuracy of any individual predictor.

We could improve the overall accuracy of the predictors by leveraging the information contained in mutation deleteriousness datasets to discover better and more useful features. Mutation deleteriousness datasets are much larger than the $\Delta\Delta G$ datasets, and they may allow sequential and structural features to "mix" in a more general environment, and produce combined features that are less noisy and better correlated with the actual effect of mutations.

We could learn those features by first training a boosted decision tree algorithm to predict mutation deleteriousness, and then use the output of those trees as input to a logistic regression model trained to predict mutation $\Delta\Delta G$ (Figure 5.1). The resulting predictor should not only have better accuracy, but should also have a better ability to extrapolate than the currently-used gradient boosted regressor algorithm, which never predicts values that are higher or lower than the maximum or minimum value observed in the training set.

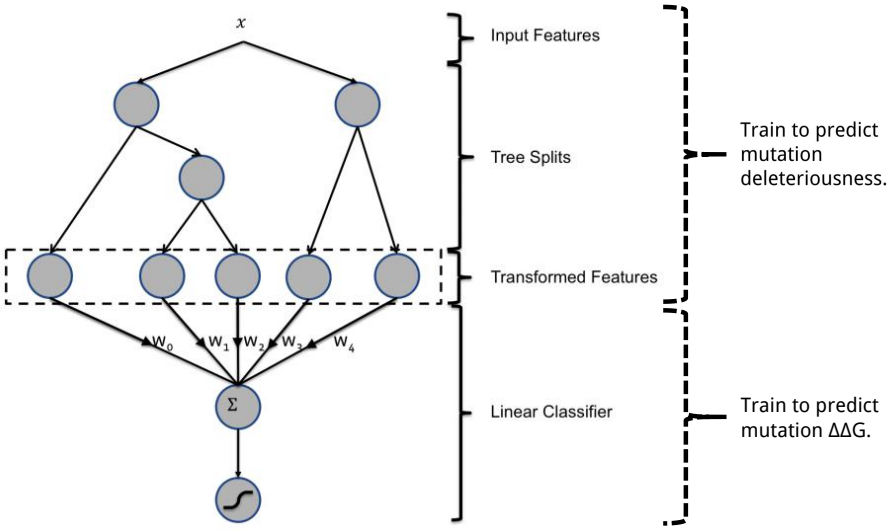


Figure 5.1: Multitask learning of mutation deleteriousness and $\Delta\Delta G$. The figure is adapted from He *et al.* [61], where it is used to describe an algorithm that couples boosted decision trees and linear regression to predict add click-trough rate. Boosted decision trees are used to learn a feature "manifold" that is provided as input to the linear classifier, which in turn makes the final predictions.
We propose to use a similar design, but train boosted decision trees to predict mutation deleteriousness, and fit a linear regressor to predict mutation $\Delta\Delta G$. We anticipate that the large training set of benign and deleterious mutations would allow the boosted decision tree algorithm to learn useful and generalizable features.

5.3 Adding support for multi-residue mutations

Around one third of experimentally determined $\Delta\Delta G$ values in the Protherm and Skempi databases correspond to changes in multiple amino acids. While ELASPIC currently supports only single residue mutations, we could leverage the information contained in multi-residue mutations by treating them as sets of single residue mutations with the experimental $\Delta\Delta G$ approximated using the following strategy:

1. Use ELASPIC to introduce each constituent single residue mutation, one at a time.
2. Keep the $\Delta\Delta G$ for the single residue mutation with the most stabilizing effect.
3. Repeat steps 1 and 2 for the remaining mutations, using, as the starting point, the structure containing the mutations selected in the previous steps.
4. Normalize the predicted $\Delta\Delta G$ of each single residue mutation in order to make the total equal to the experimental $\Delta\Delta G$ of the multi-residue mutation.
5. Add single residue mutations with the corresponding normalized $\Delta\Delta G$ values to the ELASPIC training set, retrain the classifiers, and repeat steps 1 to 5 until normalized $\Delta\Delta G$ values do not change between iterations.

If ELASPIC is able to predict the $\Delta\Delta G$ of multi-residue mutations in the Protherm and Skempi databases with reasonable accuracy, this strategy could be applied to many other datasets of multi-residue mutations. For example, we could create datasets of stabilizing, destabilizing and neutral mutations by looking at amino acid differences between proteins found in mesophilic and thermophilic bacteria, mesophilic and psychrophilic bacteria and different mesophilic bacteria, respectively. Following a similar strategy, we could also make use of the large amounts of data available from phage display experiments.

It is likely that the performance of the ELASPIC predictor would be lower for mutations affecting multiple amino acids than for mutations affecting a single amino acids, since changing multiple amino acids is more likely to alter the backbone of the protein, which is not modelled by ELASPIC. This drop in performance could in-part be ameliorated by including a backbone relaxation step in between each mutation, using molecular dynamics [62], Rosetta Backrub [63], or other algorithms [64]. Alternatively, we could construct multiple homology models using different templates, in an attempt to capture the backbone confirmation space of each domain. We could then introduce the mutation into each homology model, and average the results.

Bibliography

[1] KA. Wetterstrand. *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP).* May 24, 2016.

[2] Chee-Seng Ku et al. "Exome versus Transcriptome Sequencing in Identifying Coding Region Variants". In: *Expert Review of Molecular Diagnostics* 12.3 (April 1, 2012), pp. 241–251. DOI: 10.1586/erm.12.10.

[3] James Eberwine et al. "The Promise of Single-Cell Sequencing". In: *Nature Methods* 11.1 (January 2014), pp. 25–27.

[4] Caitlin C. Chrystoja and Eleftherios P. Diamandis. "Whole Genome Sequencing as a Diagnostic Test: Challenges and Opportunities". In: *Clinical Chemistry* 60.5 (May 2014), pp. 724–733. DOI: 10.1373/clinchem.2013.209213.

[5] Serena Nik-Zainal et al. "Landscape of Somatic Mutations in 560 Breast Cancer Whole-Genome Sequences". In: *Nature* 534.7605 (June 2, 2016), pp. 47–54. DOI: 10.1038/nature17676.

[6] Pauline C. Ng and Steven Henikoff. "SIFT: Predicting Amino Acid Changes that Affect Protein Function". In: *Nucleic Acids Research* 31.13 (July 1, 2003), pp. 3812–3814.

[7] Ivan Adzhubei et al. "Predicting Functional Effect of Human Missense Mutations Using PolyPhen-2". In: *Current Protocols in Human Genetics.* John Wiley & Sons, Inc., 2001.

[8] Biao Li et al. "Automated Inference of Molecular Mechanisms of Disease from Amino Acid Substitutions". In: *Bioinformatics* 25.21 (January 11, 2009), pp. 2744–2750. DOI: 10.1093/bioinformatics/btp528.

[9] The Cancer Genome Atlas Research Network. "Integrated Genomic Analyses of Ovarian Carcinoma". In: *Nature* 474.7353 (June 30, 2011), pp. 609–615. DOI: 10.1038/nature10166.

[10] Martin Kircher et al. "A General Framework for Estimating the Relative Pathogenicity of Human Genetic Variants". In: *Nature Genetics* 46.3 (March 2014), pp. 310–315. DOI: 10.1038/ng.2892.

[11] Hashem A. Shihab et al. "Ranking Non-Synonymous Single Nucleotide Polymorphisms Based on Disease Concepts". In: *Human Genomics* 8.1 (June 30, 2014). 00000, p. 11. DOI: 10.1186/1479-7364-8-11.

[12] Yongwook Choi et al. "Predicting the Functional Effect of Amino Acid Substitutions and Indels". In: *PLoS ONE* 7.10 (October 8, 2012). 00256, e46688. DOI: 10.1371/journal.pone.0046688.

[13] R Dorfman et al. "Do Common in Silico Tools Predict the Clinical Consequences of Amino-Acid Substitutions in the CFTR Gene?" In: *Clinical Genetics* 77.5 (May 1, 2010), pp. 464–473. DOI: 10.1111/j.1399-0004.2009.01351.x.

[14] MichaelR. Shirts and DavidL. Mobley. "An Introduction to Best Practices in Free Energy Calculations". In: *Biomolecular Simulations*. Ed. by Luca Monticelli and Emppu Salonen. Methods in Molecular Biology 924. Humana Press, January 1, 2013, pp. 271–311. DOI: 10.1007/978-1-62703-017-5_11.

[15] Alexander Benedix et al. "Predicting Free Energy Changes Using Structural Ensembles". In: *Nature Methods* 6.1 (January 2009), pp. 3–4. DOI: 10.1038/nmeth0109-3.

[16] Douglas E. V. Pires et al. "mCSM: Predicting the Effects of Mutations in Proteins Using Graph-Based Signatures". In: *Bioinformatics* 30.3 (January 2, 2014), pp. 335–342. DOI: 10.1093/bioinformatics/btt691.

[17] Josef Laimer et al. "MAESTRO - Multi Agent Stability Prediction upon Point Mutations". In: *BMC Bioinformatics* 16 (2015), p. 116. DOI: 10.1186/s12859-015-0548-6.

[18] Marharyta Petukh et al. "Predicting Binding Free Energy Change Caused by Point Mutations with Knowledge-Modified MM/PBSA Method". In: *PLOS Comput Biol* 11.7 (July 6, 2015), e1004276. DOI: 10.1371/journal.pcbi.1004276.

[19] Evan H. Baugh et al. "Robust Classification of Protein Variation Using Structural Modelling and Large-Scale Data Integration". In: *Nucleic Acids Research* 44.6 (July 4, 2016), pp. 2501–2513. DOI: 10.1093/nar/gkw120.

[20] Yves Dehouck et al. "Fast and Accurate Predictions of Protein Stability Changes upon Mutations Using Statistical Potentials and Neural Networks: PoPMuSiC-2.0". In: *Bioinformatics* 25.19 (January 10, 2009), pp. 2537–2543. DOI: 10.1093/bioinformatics/btp445.

[21] Niklas Berliner et al. "Combining Structural Modeling with Ensemble Machine Learning to Accurately Predict Protein Fold Stability and Binding Affinity Effects upon Mutation". In: *PLoS ONE* 9.9 (September 22, 2014), e107353. DOI: 10.1371/journal.pone.0107353.

[22] Minghui Li et al. "MutaBind Estimates and Interprets the Effects of Sequence Variants on Protein–protein Interactions". In: *Nucleic Acids Research* 44 (W1 August 7, 2016), W494–W501. DOI: 10.1093/nar/gkw374.

[23] Helen M. Berman et al. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (January 1, 2000), pp. 235–242. DOI: 10.1093/nar/28.1.235.

[24] Roberto Mosca et al. "Interactome3D: Adding Structural Details to Protein Networks". In: *Nature Methods* 10.1 (January 2013), pp. 47–53. DOI: 10.1038/nmeth.2289.

[25] Marco Punta et al. "The Pfam Protein Families Database". In: *Nucleic Acids Research* 40 (D1 January 1, 2012). 00002, pp. D290–D301. DOI: 10.1093/nar/gkr1065.

[26] Alison L. Cuff et al. "Extending CATH: Increasing Coverage of the Protein Structure Universe and Linking Structure with Function". In: *Nucleic Acids Research* 39 (Database issue January 2011). 00100, pp. D420–D426. DOI: 10.1093/nar/gkq1001.

[27] Daniel K. Witvliet et al. "ELASPIC Web-Server: Proteome-Wide Structure-Based Prediction of Mutation Effects on Protein Stability and Binding Affinity". In: *Bioinformatics* 32.10 (May 15, 2016), pp. 1589–1591. DOI: 10.1093/bioinformatics/btw031.

[28] Thomas Rattei et al. "SIMAP—a Comprehensive Database of Pre-Calculated Protein Sequence Similarities, Domains, Annotations and Clusters". In: *Nucleic Acids Research* 38 (suppl 1 January 1, 2010). 00031, pp. D223–D226. DOI: 10.1093/nar/gkp949.

[29] Robert C. Edgar. "MUSCLE: A Multiple Sequence Alignment Method with Reduced Time and Space Complexity". In: *BMC Bioinformatics* 5.1 (August 19, 2004). 02783, p. 113. DOI: 10.1186/1471-2105-5-113.

[30] Derek Wilson et al. "SUPERFAMILY—sophisticated Comparative Genomics, Data Mining, Visualization and Phylogeny". In: *Nucleic Acids Research* 37 (suppl 1 January 1, 2009), pp. D380–D386. DOI: 10.1093/nar/gkn762.

[31] Tim J. P. Hubbard et al. "SCOP: A Structural Classification of Proteins Database". In: *Nucleic Acids Research* 27.1 (January 1, 1999), pp. 254–256. DOI: 10.1093/nar/27.1.254.

[32] Su Datt Lam et al. "Gene3D: Expanding the Utility of Domain Assignments". In: *Nucleic Acids Research* 44 (D1 April 1, 2016), pp. D404–D409. DOI: 10.1093/nar/gkv1231.

[33] Martin H. Schaefer et al. "HIPPIE: Integrating Protein Interaction Networks with Experiment Based Quality Scores". In: *PLoS ONE* 7.2 (February 14, 2012), e31826. DOI: 10.1371/journal.pone.0031826.

[34] Thomas Rolland et al. "A Proteome-Scale Map of the Human Interactome Network". In: *Cell* 159.5 (November 20, 2014). 00006, pp. 1212–1226. DOI: 10.1016/j.cell.2014.10.050.

[35] Jean-François Rual et al. "Towards a Proteome-Scale Map of the Human Protein–protein Interaction Network". In: *Nature* 437.7062 (October 20, 2005). 02009, pp. 1173–1178. DOI: 10.1038/nature04209.

[36] Haiyuan Yu et al. "Next-Generation Sequencing to Generate Interactome Datasets". In: *Nature Methods* 8.6 (June 2011). 00070, pp. 478–480. DOI: 10.1038/nmeth.1597.

[37] Kavitha Venkatesan et al. "An Empirical Framework for Binary Interactome Mapping". In: *Nature Methods* 6.1 (January 2009). 00427, pp. 83–90. DOI: 10.1038/nmeth.1280.

[38] Benjamin Webb and Andrej Sali. "Comparative Protein Structure Modeling Using MODELLER". In: *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc., 2002.

[39] Joost Schymkowitz et al. "The FoldX Web Server: An Online Force Field". In: *Nucleic Acids Research* 33 (suppl 2 January 7, 2005), W382–W388. DOI: 10.1093/nar/gki387.

[40] F. Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[41] Michel F. Sanner et al. "Fast and Robust Computation of Molecular Surfaces". In: *Proceedings of the Eleventh Annual Symposium on Computational Geometry*. SCG '95. New York, NY, USA: ACM, 1995, pp. 406–407. DOI: 10.1145/220279.220324.

[42] S Hubbard and J Thornton. *NACCESS*. Department of Biochemistry Molecular Biology, University College London., 1993.

[43] Hongbo Zhu et al. "NOXclass: Prediction of Protein-Protein Interaction Types". In: *BMC Bioinformatics* 7.1 (January 19, 2006), p. 27. DOI: 10.1186/1471-2105-7-27.

[44] Sameer Velankar et al. "SIFTS: Structure Integration with Function, Taxonomy and Sequences Resource". In: *Nucleic Acids Research* 41 (D1 January 1, 2013), pp. D483–D489. DOI: 10.1093/nar/gks1258.

[45] Nidhi Sahni et al. "Widespread Macromolecular Interaction Perturbations in Human Genetic Disorders". In: *Cell* 161.3 (April 23, 2015), pp. 647–660. DOI: 10.1016/j.cell.2015.04.013.

[46] The UniProt Consortium. "UniProt: A Hub for Protein Information". In: *Nucleic Acids Research* 43 (D1 January 28, 2015), pp. D204–D212. DOI: 10.1093/nar/gku989.

[47] Melissa J. Landrum et al. "ClinVar: Public Archive of Interpretations of Clinically Relevant Variants". In: *Nucleic Acids Research* 44 (D1 April 1, 2016), pp. D862–D868. DOI: 10.1093/nar/gkv1222.

[48] Simon A. Forbes et al. "COSMIC: Exploring the World's Knowledge of Somatic Mutations in Human Cancer". In: *Nucleic Acids Research* 43 (D1 January 28, 2015), pp. D805–D811. DOI: 10.1093/nar/gku1075.

[49] K. Abdulla Bava et al. "ProTherm, Version 4.0: Thermodynamic Database for Proteins and Mutants". In: *Nucleic Acids Research* 32 (suppl 1 January 1, 2004). 00169, pp. D120–D121. DOI: 10.1093/nar/gkh082.

[50] M. D. Shaji Kumar et al. "ProTherm and ProNIT: Thermodynamic Databases for Proteins and Protein–nucleic Acid Interactions". In: *Nucleic Acids Research* 34 (suppl 1 January 1, 2006), pp. D204–D206. DOI: 10.1093/nar/gkj103.

[51] Elizabeth H. Kellogg et al. "Role of Conformational Sampling in Computing Mutation-Induced Changes in Protein Structure and Stability". In: *Proteins* 79.3 (March 2011). 00094, pp. 830–838. DOI: 10.1002/prot.22921.

[52] Iain H. Moal and Juan Fernández-Recio. "SKEMPI: A Structural Kinetic and Energetic Database of Mutant Protein Interactions and Its Use in Empirical Models". In: *Bioinformatics* 28.20 (October 15, 2012), pp. 2600–2607. DOI: 10.1093/bioinformatics/bts489.

[53] Tanja Kortemme and David Baker. "A Simple Physical Model for Binding Energy Hot Spots in Protein–protein Complexes". In: *Proceedings of the National Academy of Sciences of the United States of America* 99.22 (October 29, 2002), pp. 14116–14121. DOI: 10.1073/pnas.202485799.

[54] J. Weile et al. "An Atlas of Functional Amino Acid Changes in Human SUMO and SUMO Ligase." In: (In preparation).

[55] Sarah Sirin et al. "AB-Bind: Antibody Binding Mutational Database for Computational Affinity Predictions". In: *Protein Science* 25.2 (February 1, 2016), pp. 393–409. DOI: 10.1002/pro.2829.

[56] Malachi Griffith et al. "CIViC: A Knowledgebase for Expert-Crowdsourcing the Clinical Interpretation of Variants in Cancer." In: *bioRxiv* (September 1, 2016), p. 072892. DOI: 10.1101/072892.

[57] P. Andrew Futreal et al. "A Census of Human Cancer Genes". In: *Nature Reviews Cancer* 4.3 (March 2004), pp. 177–183. DOI: 10.1038/nrc1299.

[58] Jürgen Schlitter. "Estimation of Absolute and Relative Entropies of Macromolecules Using the Covariance Matrix". In: *Chemical Physics Letters* 215.6 (December 17, 1993), pp. 617–621. DOI: 10.1016/0009-2614(93)89366-P.

[59] Daniel Seeliger and Bert L. de Groot. "Protein Thermostability Calculations Using Alchemical Free Energy Simulations". In: *Biophysical Journal* 98.10 (May 19, 2010), pp. 2309–2316. DOI: 10.1016/j.bpj.2010.01.051.

[60] Jerome C. Nwachukwu et al. "Predictive Features of Ligand-specific Signaling through the Estrogen Receptor". In: *Molecular Systems Biology* 12.4 (April 1, 2016), p. 864. DOI: 10.15252/msb.20156701.

[61] Xinran He et al. "Practical Lessons from Predicting Clicks on Ads at Facebook". In: *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ADKDD'14. New York, NY, USA: ACM, 2014, 5:1–5:9. DOI: 10.1145/2648584.2648589.

[62] Mark James Abraham et al. "GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers". In: *SoftwareX* 1–2 (September 2015), pp. 19–25. DOI: 10.1016/j.softx.2015.06.001.

[63] Colin A. Smith and Tanja Kortemme. "Predicting the Tolerated Sequences for Proteins and Protein Interfaces Using RosettaBackrub Flexible Backbone Design". In: *PLOS ONE* 6.7 (July 18, 2011), e20451. DOI: 10.1371/journal.pone.0020451.

[64] Mark G. F. Sun et al. "Protein Engineering by Highly Parallel Screening of Computationally Designed Variants". In: *Science Advances* 2.7 (July 1, 2016), e1600692. DOI: 10.1126/sciadv.1600692.