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

Contents

List of Tables

List of Figures

Introduction

1.1 Background

Recent advances in DNA sequence technology have drastically lowered the cost and improved the accuracy of genome sequencing [1]. This has made exome and whole-genome sequencing a viable and cost-effective tool in both the laboratory and in the clinic to assist with the diagnosis and direct treatment of pediatric conditions [2] and cancers [3], and has led to an enormous growth in the amount of genomic data that is being generated. However, interpreting such genomic data to produce meaningful and actionable results remains a challenge.

In vitro and *in vivo* experiments remain the gold standard in elucidating the effect of mutations. However, evaluating experimentally the effect of all discovered mutants is not feasible. Computational techniques have been developed to predict the effect of different mutations and to prioritize them for experimental validation. Those techniques generally use conservation score describing the likelihood that a particular amino acid being found in the particular position in orthologous proteins.

The most widely-used program for predicting the deleteriousness of a mutation is Sorting Intolerant from Tolerant (SIFT) [4]. SIFT runs PSI-Blast to create a multiple sequence alignment for the query protein, and computes a conservation score by looking at the likelihood of the wildtype and mutant amino acids occurring at a given position in the alignment. While SIFT is a well-established tool in the field, it is difficult to compile and install on a local machine. Furthermore, multiple sequence alignments constructed by SIFT can be several megabytes in size, and caching this data for an entire proteome would require a non-trivial amount of storage space.

Another popular sequence-based algorithm is Provean [5]. Provean also calculates uses PSI-Blast to calculate a multiple sequence alignment. However, it then runs CD-HIT to select under 100 representative sequences capturing the diversity of the alignment, and then performs pairwise alignments with this "supporting set" to predict the final score. Provean is reported to achieve similar performance to SIFT. However, unlike SIFT, it is freely available under the GPLv3 license, it compiles easily and runs on modern Linux distributions. Furthermore, Provean is distributed under a license, and uses *supporting sets* of at most 45 sequences which can be precalculated and stored. If a supporting set is available, calculating the Provean score takes several seconds per mutation.

The performance of Provean is comparable to the leading mutation scoring programs, such as SITF, PolyPhen-2, Mutation Assessor, and CONDEL [5]. Furthermore, Provean is distributed under a GPLv3 license, and uses *supporting sets* of at most 45 sequences which can precalculated and stored. If a supporting set is available, calculating the Provean score takes several seconds per mutation.

Another widely-used mutaiton scoring tool is PolyPhen-2. It is one of the packages predicted for

Many other tools have been developed that offer various advantages over SIFT / Provean. PolyPhen-2

[6] uses support vector machines to combine a conservation score with different sequential and structural features of the wildtype and mutant residue. However, since PolyPhen-2 is trained on a dataset of human deleterious mutations, it is difficult to use in downstream applications, as one would have to make sure to exclude the PolyPhen-2 training set throughout the training and validation process. FATHMM [7] constructs a hidden Markov model based on the alignment, and is reported to achieve marginally higher accuracy than SIFT / Provean. Other techniques offering various advantages over SIFT / Provean include MutPred [8], MutationAssesor [9], CADD [**kircher˙general˙2014**], CONDEL [5], and others.

Despite the proliferation of tools predicting the deleteriousness of different SNPs, our ability to act on those predictions remains limited. Existing computational methods are limited in their accuracy and the type of information that they can provide. Most existing tools use a conservation score. While millions of single nucleotide polymorphisms (SNPs) have been implicated in thousands of diseases, approaches for predicting the phenotypic effect of newly-discovered mutations are still in their infancy. One of the reasons is that while sequence-based tools achieve reasonably good performance at predicting whether or not a given mutation is going to be deleterious, they fall short in predicting *why* that mutation is deleterious. This lack of actionable predictions limits the usability of the vast DNA sequencing data that has been generated. However, the etiology by which the mutations cause or contribute to a disease are often unknown.

1.2 Structural approaches to predicting the effect of mutations

Statistical potentials

Physics-based methods the electrostatic, van der Waals, solvent accessible surface area, and entropy terms

Concoord/Poisson-Boltzmann surface area (CC/PBSA server)

The central dogma of biology is that DNA is transcribed into RNA which is translated into Protein.

One reason for out lack of ability in interpreting is the focus on the sequence-level features, while in the majority of missense mutations, it is the alteration in the function of the transcribed protein which is responsible for the detrimental effect of mutations.

The field of protein science has generally been concerned with the broad questions of protein folding, protein design. Algorithms have been developed to predict the effect of mutations on protein folding and protein-protein affinity, but those tools are generally meant to be used on a case-by-case basis and have not been designed to be applied on a genome-wide scale to predict the effect of missense mutations from whole-genome sequencing studies.

While the growth in protein crystal structures has not seen the rapid rise that was observed in DNA sequencing, the number of resolved protein structures has also been increasing, with the Protein Data Bank (PDB) containing close to 125,000 structures as of 2016.

A related are of research is predicting the energetic effect of mutations.

The most accurate class of computational techniques are alchemical free energy calculations, which involve modelling the structural transition from the wildtype to the mutant protein and using the Bennett acceptance ratio (BAR) or thermodynamic integration (TI) to calculate the energetics of the transition [10]. However, alchemical free energy calculations are computationally expansive, and are generally used only in cases where the experimental characterization of mutants is particularly difficult, as in the case of D-amino acid peptide design [11].

Many algorithms have been developed which attempt to predict the effect of mutations on protein stability and / or on protein-protein interaction affinity. Those techniques generally use a rigid backbone representation of protein and use statistical potentials. For a review see XXX.

Mixed strategies which utilize both sequence- and structure-based approaches. Such algorithms include PoPMuSiC,

Structure-based tools which predict the effect of mutations on protein structure and / or function using features describing the three-dimensional structure of the protein. mCSM [12] (graph-based signatures), MAESTRO [13] (multi-agent machine learning), CC/PBSA (Concoord/Poisson-Boltzmann surface area) [14],

Some algorithms rely on the conservation of the residue in multiple sequence alignments.

Predicting protein thermal stability changes upon point mutations using statistical potentials: Introducing HoTMuSiC

- MAESTRO implements a multi-agent machine learning system.

- Structure based tools AUTO-MUTE [7], CUPSAT [8], Dmutant [9], FoldX [10], Eris [11], PoPMuSiC [12], SDM [13] or mCSM [14] usually perform better than the sequence based counterparts. Recently, SDM and mCSM have been integrated into a new method called DUET [15].

INPS: predicting the impact of non-synonymous variations on protein stability from sequence

- `http://bioinformatics.oxfordjournals.org/content/31/17/2816.long`

- Here, we describe INPS, a novel approach for annotating the effect of non-synonymous mutations on the protein stability from its sequence.

- [15]

FoldX

PoPMuSiC

RosettaCM

mCSM: predicting the effects of mutations in proteins using graph-based signatures.

- `http://www.ncbi.nlm.nih.gov/pubmed/24281696`

- “To understand the roles of mutations in disease, we have evaluated their impacts not only on protein stability but also on protein-protein and protein-nucleic acid interactions”.

- [12]

Predicting Binding Free Energy Change Caused by Point Mutations with Knowledge-Modified MM/PBSA Method

- `http://journals.plos.org/ploscompbiol/article?id=10.1371%2Fjournal.pcbi.1004276`

- “The core of the SAAMBE method is a modified molecular mechanics Poisson-Boltzmann Surface Area (MM/PBSA) method with residue specific dielectric constant”.

- [16]

Rosetta benchmark [17]

Benchmark showing Rosetta doing poorly: [18]

I-Mutant2, DMutant, CUPSAT, FoldX [19]

1.3 Goals and objectives

- Evaluate how well we can predict the deleteriousness of a mutation by measuring the effect of protein folding on protein stability.

- Assessing the impact of missense mutations.
- Protein engineering. For example generating biological therapeutics that are more thermostable and have a higher affinity for their target.
- Basic science: characterizing the forces that are most important in protein folding and binding, and the effect of mutations on those forces.
- In this work we examine how much sequence-based features can aid in the prediction of traditionally structural realms such as the prediction of $\Delta\Delta G$ scores of mutations, and how much structure-based features can aid with the prediction of mutation pathogenicity–a traditionally sequence based

1.4 Acknowledgements

This is a continuation of the work performed by Niklas Berliner *et al.* [20]. In 1.4 we discuss how we expand ELASPIC to work on the genome-wide scale. In 2.3 we discuss how we retrained ELASPIC while leveraging the information we extracted from genome-wide analysis.

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 [21]. 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 [22]. He named this pipeline Profs, for Protein families. A schematic of this pipeline is presented in Figure 2.1, and the 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 also 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 lists of Profs domain definitions and Profs domain-domain interactions that are used by ELASPIC.

2.1.1 Domains

We used Profs domain definitions calculated, as part of the Profs pipeline, for all proteins in the PDB, to find Profs domains, and structural templates for those domains, for all proteins in Uniprot. To do this, we followed a similar process to what was done to annotate with Profs domains structures in the PDB that lack CATH annotations [23].

We started with Pfam domain definitions for all known protein sequences, which we download from the SIMAP website [24]. 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

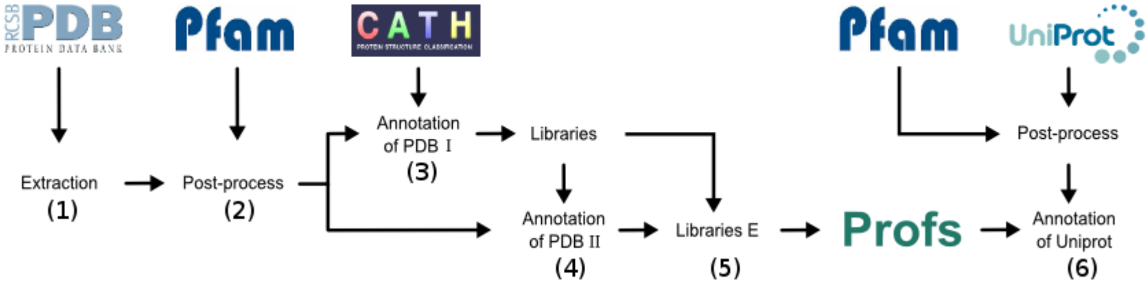


Figure 2.1: Flowchart illustrating steps in the Profs pipeline (courtesy of Andres Felipe Giraldo Forero). Each step in the flowchart is annotated with the section number where that step is explained. **(1)** All structures in the PDB are parsed to extract protein sequences, and HMMScan is ran to find 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.

running *blastp* against libraries of Profs domains, which are included in the Profs R package. If a suitable template was found, we proceeded to do iterative global alignments using Muscle [25] 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 equal parts 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 were 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 Comparing Profs with Pfam and Gene3D

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).

We downloaded Pfam and Gene3D domain definitions for all human proteins from SIMAP [24], 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.

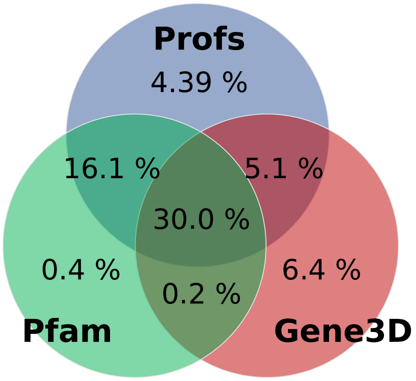


Figure 2.2: Venn diagram showing the overlap in domain definitions between Profs, Pfam, and Gene3D. Values represent the fraction of amino acids, of all human proteins in UniProt, which are covered the particular domain or domains. A total of 18,828 human proteins and 10,868,810 amino acids were considered, 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.

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 9 % more amino acids than Pfam and 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.

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 [26] and in the datasets hosted by the Harvard Center for Cancer Systems Biology (CCSB) [27]. 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. 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

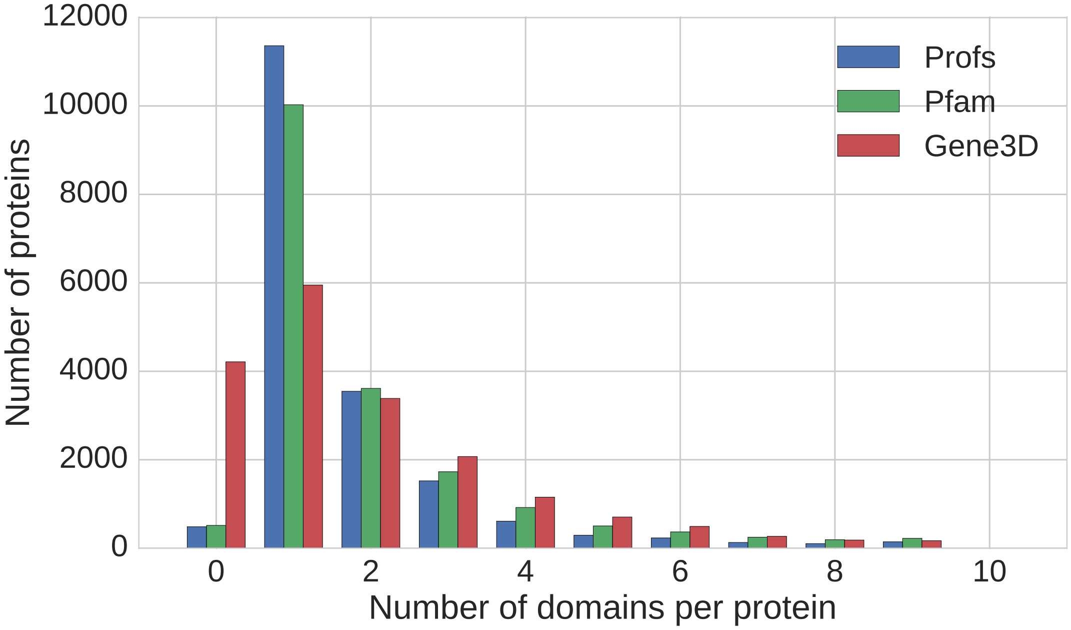


Figure 2.3: Average number of Profs, Pfam and Gene3D domains per protein, for all human proteins containing at least one domains. Profs tends to have fewer domains per protein then either Pfam or Gene3D. Gene3D lacks domain annotation for many proteins which contain at least one Pfam and Profs domain.

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.

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

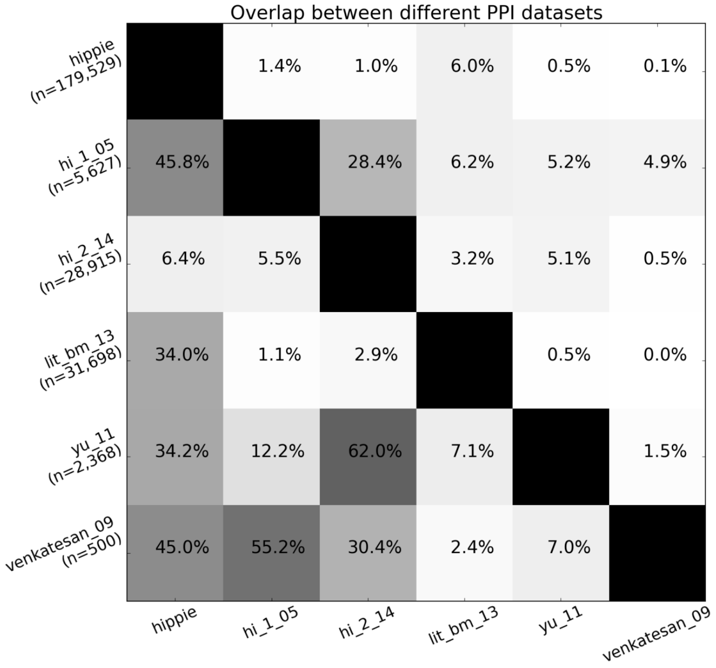


Figure 2.4: Overlap in protein-protein interaction (PPI) databases. The shade and value of each square denotes the percentage of PPIs in the database named on the y-axis that are also found in the database named on the x-axis. **hippie** is a meta-database, which integrates PPIs from many different sources [26]. **hi_1_05** contains PPIs discovered through a proteome-wide yeast two-hybrid experiment conducted by Rual *et al.* [28]. **hi_2_14** contains PPIs discovered through a proteome-wide yeast two-hybrid experiment conducted by Rolland *et al.* [27]. **lit_bm_13** contains PPIs obtained from the literature and supported by multiple pieces of evidence [27]. **yu_11** contains PPIs obtained using "stitch-seq", which combines PCR stitching with next-generation sequencing [29]. **venkatesan_09** corresponds to high-quality binary interactions found in repeat yeast two-hybrid assays conducted by Venkatesan *et al.* [30].

2.2 ELASPIC

The ELASPIC project was started by Niklas Berliner and others in 2014 [20].

ELASPIC uses Modeller [31] to construct homology models of domains and domain-domain interactions, FoldX to optimize those model and to introduce mutations [32], and the GradientBoostingRegressor from scikit-learn [33] to combine FoldX energy scores with sequence-based and other features and predict the energetic impact of a mutation on the stability of a single domain or the affinity between two domains. An overview of the ELASPIC pipeline is presented in Figure 2.5. ELASPIC includes a library Python scripts for construction sequence alignments, constructing Provean supporting sets and computing the Provean score, constructing homology models, running FoldX, and predicting the $\Delta\Delta G$ of the mutation. It also includes a “Standalone Pipeline” (Figure 2.5 right) and a “Database Pipeline” (Figure 2.5 left), which include command line options for mutating a protein structure.

2.2.1 Standalone pipeline

The standalone pipeline works without downloading and installing a local copy of the ELASPIC and PDB databases, but requires a PDB structure or template to be provided for every protein. Pipeline output is saves as JSON files inside the working directory, rather than being uploaded to the database as in the case of the database pipeline. The general overview of the local pipleine is presented in the figure to the right.

The local pipeline still requires a local copy of the Blast nr database.

We used the MODELLER software package to perform all homology modeling.

“MODELLER uses simulated annealing cycles along with a minimal forcefield and spatial restraints – generally Gaussian interatomic probability densities extracted from the template structure with database-derived statistics determining the distribution widthto rapidly generate candidate structures of the target sequence from the provided template sequence.”

2.2.2 Database pipeline

The database pipeline allows mutations to be performed on a proteome-wide scale, without having to specify a structural template for each protein. This pipeline requires a local copy of 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 in 2.5 left. A user runs the ELASPIC pipeline specifying 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.

At each step in the pipeline, a local database is queried to see if the required information has already been calculated. If the information is available, the pipeline moves to the next step. If the information is not available, the pipeline runs the module that generates the required information, stores the generated information in the database for future access, and then moves to the next step. If the specified mutation falls outside of every domain in the protein, no predictions are returned. Otherwise, the pipeline evaluates the impact of the mutation on the stability of the domain and, if the mutation falls in a domain interface,

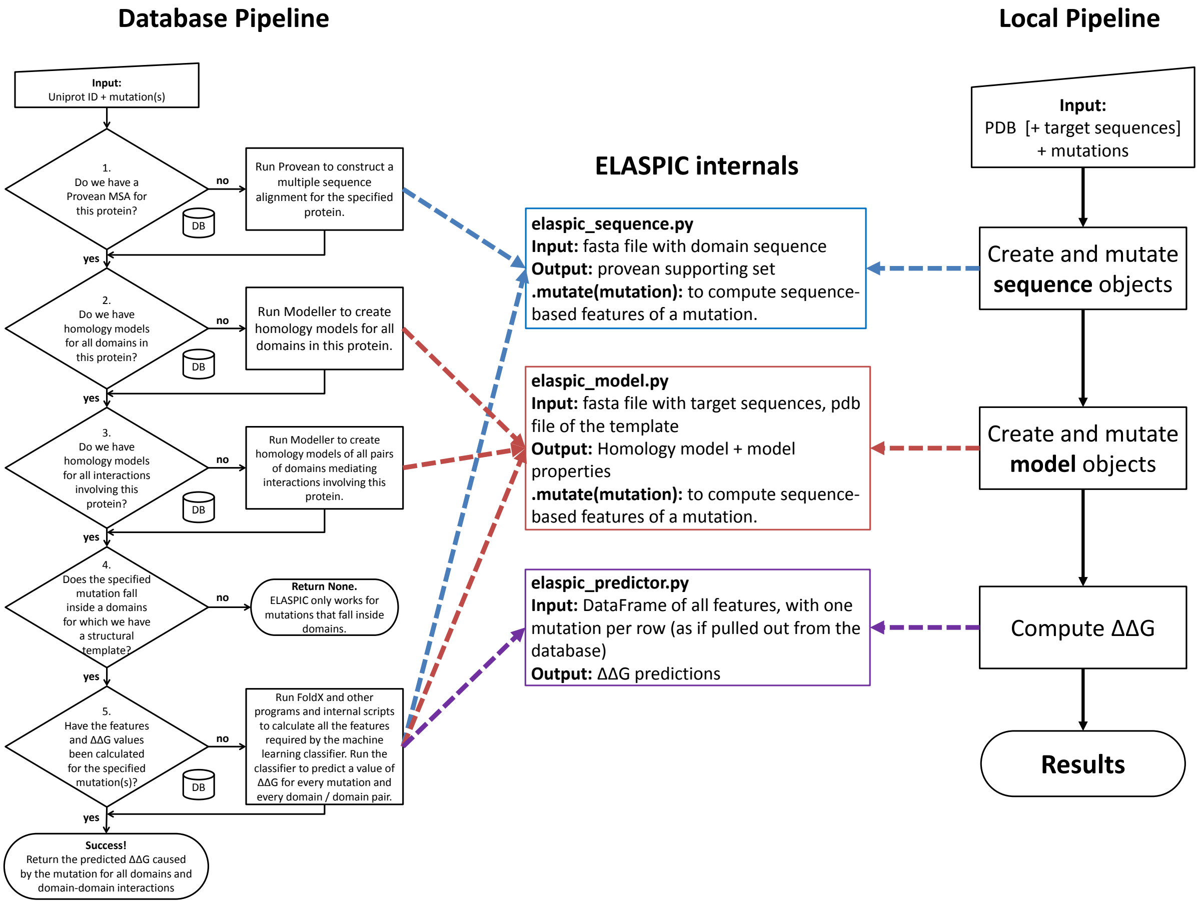

Database Pipeline
Input:
Uniprot ID + mutation(s)
1.
Do we have a
Provean MSA for
this protein?
DB
no
Run Provean to construct a multiple sequence alignment for the specified protein.
yes
2.
Do we have
homology models
for all domains in
this protein?
DB
no
Run Modeller to create homology models for all domains in this protein.
yes
3.
Do we have
homology models
for all interactions
involving this
protein?
DB
no
Run Modeller to create homology models of all pairs of domains mediating interactions involving this protein.
yes
4.
Does the specified
mutation fall
inside a domains
for which we have
a structural
template?
no
Return None.
ELASPIC only works for mutations that fall inside domains.
yes
5.
Have the features
and ΔΔG values
been calculated
for the specified
mutation(s)?
DB
no
Run FoldX and other programs and internal scripts to calculate all the features required by the machine learning classifier. Run the classifier to predict a value of ΔΔG for every mutation and every domain / domain pair.
yes
Success!
Return the predicted ΔΔG caused by the mutation for all domains and domain-domain interactions
ELASPIC internals
elaspic_sequence.py
Input: fasta file with domain sequence
Output: provean supporting set
.mutate(mutation): to compute sequence-based features of a mutation.
elaspic_model.py
Input: fasta file with target sequences, pdb file of the template
Output: Homology model + model properties
.mutate(mutation): to compute sequence-based features of a mutation.
elaspic_predictor.py
Input: DataFrame of all features, with one mutation per row (as if pulled out from the database)
Output: ΔΔG predictions
Local Pipeline
Input:
PDB [+ target sequences]
+ mutations
Create and mutate
sequence objects
Create and mutate
model objects
Compute ΔΔG
Results

Figure 2.5: Overview of the ELASPIC pipeline. **Database Pipeline:** A user runs the ELASPIC pipeline 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 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 the filename of the PDB that they wish to mutate and one or more mutations, or a filename specifying the sequence of the protein that they wish to model, the filename of the PDB that they wish to use as a template, 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 are not recalculated if the user decides to run more mutations.

on the affinity between two domains. In order to expedite the evaluation of mutations, we precalculated homology models and Provean supporting sets for all human proteins. Structural and sequential features, and predicted $\Delta\Delta G$ scores, have also been precalculated for the majority of mutations listed in the Uniprot humsavar file [34] and in the COSMIC [35] and ClinVar [36] databases.

Provean supporting sets, homology models and mutation $\Delta\Delta G$ scores are available from the ELASPIC downloads page: `http://elaspic.kimlab.org/static/download/`. The source code of the python package implementing the ELASPIC pipeline is available from `https://github.com/kimlaborg/elaspic`, and the documentation for the ELASPIC pipeline can be accessed online at `http://elaspic.readthedocs.`

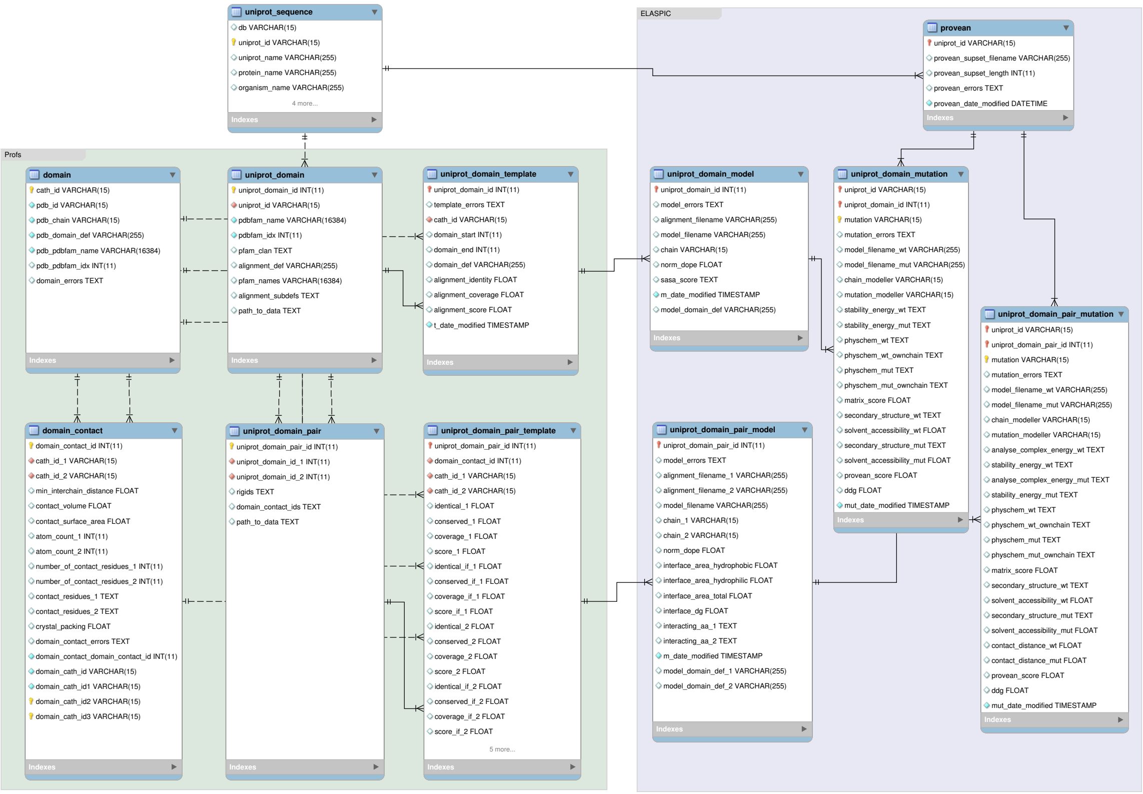

uniprot_sequence
db VARCHAR(15)
uniprot_id VARCHAR(15)
uniprot_name VARCHAR(255)
protein_name VARCHAR(255)
organism_name VARCHAR(255)
4 more...
Indexes
ELASPIC
provean
uniprot_id VARCHAR(15)
provean_supset_filename VARCHAR(255)
provean_supset_length INT(11)
provean_errors TEXT
provean_date_modified DATETIME
Indexes
Profs
domain
cath_id VARCHAR(15)
pdb_id VARCHAR(15)
pdb_chain VARCHAR(15)
pdb_domain_def VARCHAR(255)
pdb_pdbfam_name VARCHAR(16384)
pdb_pdbfam_idx INT(11)
domain_errors TEXT
Indexes
uniprot_domain
uniprot_domain_id INT(11)
uniprot_id VARCHAR(15)
pdbfam_name VARCHAR(16384)
pdbfam_idx INT(11)
pfam_clan TEXT
alignment_def VARCHAR(255)
pfam_names VARCHAR(16384)
alignment_subdefs TEXT
path_to_data TEXT
Indexes
uniprot_domain_template
uniprot_domain_id INT(11)
template_errors TEXT
cath_id VARCHAR(15)
domain_start INT(11)
domain_end INT(11)
domain_def VARCHAR(255)
alignment_identity FLOAT
alignment_coverage FLOAT
alignment_score FLOAT
t_date_modified TIMESTAMP
Indexes
uniprot_domain_model
uniprot_domain_id INT(11)
model_errors TEXT
alignment_filename VARCHAR(255)
model_filename VARCHAR(255)
chain VARCHAR(15)
norm_dope FLOAT
sasa_score TEXT
m_date_modified TIMESTAMP
model_domain_def VARCHAR(255)
Indexes
uniprot_domain_mutation
uniprot_id VARCHAR(15)
uniprot_domain_id INT(11)
mutation VARCHAR(15)
mutation_errors TEXT
model_filename_wt VARCHAR(255)
model_filename_mut VARCHAR(255)
chain_modeller VARCHAR(15)
mutation_modeller VARCHAR(15)
stability_energy_wt TEXT
stability_energy_mut TEXT
physchem_wt TEXT
physchem_wt_ownchain TEXT
physchem_mut TEXT
physchem_mut_ownchain TEXT
matrix_score FLOAT
secondary_structure_wt TEXT
solvent_accessibility_wt FLOAT
secondary_structure_mut TEXT
solvent_accessibility_mut FLOAT
provean_score FLOAT
ddg FLOAT
mut_date_modified TIMESTAMP
Indexes
uniprot_domain_pair_mutation
uniprot_id VARCHAR(15)
uniprot_domain_pair_id INT(11)
mutation VARCHAR(15)
mutation_errors TEXT
model_filename_wt VARCHAR(255)
model_filename_mut VARCHAR(255)
chain_modeller VARCHAR(15)
mutation_modeller VARCHAR(15)
analyse_complex_energy_wt TEXT
stability_energy_wt TEXT
analyse_complex_energy_mut TEXT
stability_energy_mut TEXT
physchem_wt TEXT
physchem_wt_ownchain TEXT
physchem_mut TEXT
physchem_mut_ownchain TEXT
matrix_score FLOAT
secondary_structure_wt TEXT
solvent_accessibility_wt FLOAT
secondary_structure_mut TEXT
solvent_accessibility_mut FLOAT
contact_distance_wt FLOAT
contact_distance_mut FLOAT
provean_score FLOAT
ddg FLOAT
mut_date_modified TIMESTAMP
Indexes
domain_contact
domain_contact_id INT(11)
cath_id_1 VARCHAR(15)
cath_id_2 VARCHAR(15)
min_interchain_distance FLOAT
contact_volume FLOAT
contact_surface_area FLOAT
atom_count_1 INT(11)
atom_count_2 INT(11)
number_of_contact_residues_1 INT(11)
number_of_contact_residues_2 INT(11)
contact_residues_1 TEXT
contact_residues_2 TEXT
crystal_packing FLOAT
domain_contact_errors TEXT
domain_contact_domain_contact_id INT(11)
domain_cath_id VARCHAR(15)
domain_cath_id1 VARCHAR(15)
domain_cath_id2 VARCHAR(15)
domain_cath_id3 VARCHAR(15)
Indexes
uniprot_domain_pair
uniprot_domain_pair_id INT(11)
uniprot_domain_id_1 INT(11)
uniprot_domain_id_2 INT(11)
rigids TEXT
domain_contact_ids TEXT
path_to_data TEXT
Indexes
uniprot_domain_pair_template
uniprot_domain_pair_id INT(11)
domain_contact_id INT(11)
cath_id_1 VARCHAR(15)
cath_id_2 VARCHAR(15)
identical_1 FLOAT
conserved_1 FLOAT
coverage_1 FLOAT
score_1 FLOAT
identical_if_1 FLOAT
conserved_if_1 FLOAT
coverage_if_1 FLOAT
score_if_1 FLOAT
identical_2 FLOAT
conserved_2 FLOAT
coverage_2 FLOAT
score_2 FLOAT
identical_if_2 FLOAT
conserved_if_2 FLOAT
coverage_if_2 FLOAT
score_if_2 FLOAT
5 more...
Indexes
uniprot_domain_pair_model
uniprot_domain_pair_id INT(11)
model_errors TEXT
alignment_filename_1 VARCHAR(255)
alignment_filename_2 VARCHAR(255)
model_filename VARCHAR(255)
chain_1 VARCHAR(15)
chain_2 VARCHAR(15)
norm_dope FLOAT
interface_area_hydrophobic FLOAT
interface_area_hydrophilic FLOAT
interface_area_total FLOAT
interface_dg FLOAT
interacting_aa_1 TEXT
interacting_aa_2 TEXT
m_date_modified TIMESTAMP
model_domain_def_1 VARCHAR(255)
model_domain_def_2 VARCHAR(255)
Indexes

Figure 2.6: ELASPIC database schema. Tables on the green plate titled Profs are calculated using the Profs pipeline, as described in [23]. Tables on the purple plate titled ELASPIC are calculated using the ELASPIC pipeline, following the procedure outlined in 2.5. A detailed description of each table can be found in 2.1.

`org/`.

2.2.3 Database schema

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.

In the ELASPIC database (Figure 2.6), Profs domain definitions produced by Andres' pipleine are contained in the **domain** and **domain_contact** tables, while Profs domain definitions for uniprot are contained in the **uniprot_domain**, **uniprot_domain_template**, **uniprot_domain_pair**, and **uniprot_domain_pair_t**
tables.

2.2.4 ELASPIC microservice

One of our aims was to make ELASPIC accessible to the general scientific community. Daniel Witvliet took up the task of developing a webserver that would allow users to submit mutations and analyze results interactively.

In order to make the webserver scale to thousands of mutations, we opted to run ELASPIC jobs on a local Sun Grid Engine (SGE) cluster. However, this introduced several unique objectives and challenges

Table 2.1: ELASPIC database schema.

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 [37] 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 the Uniprot.
provean	Contains information about Provean [5] 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 [24], 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 [23].
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 which 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 binding.
uniprot_domain_pair	Contains pairs of domains that are likely to mediate the interaction between known interacting partners, obtained from Hippie [26] and Rolland et al. [27].
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 which 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 binding.

to the design:

1. If users submitted multiple mutations affecting the same protein, we had to make sure that the Provean supporting sets and homology models for the protein are calculated before the mutation jobs are submitted to the cluster. Otherwise, each mutation would initiate the calculation of a Provean supporting set (which can require ¿ 5 GB of memory) and a homology model (which can

Table 2.2: ELASPIC web service API.

Method	**HTTP request**	**Description**
submitjob	POST /submitjob	Submit a job to be run on a SGE cluser.
jobstatus	GET /submitjob	View the results of a job.

require ¿ 10 minutes of CPU time), which lead to many un-neccesasry jobs, would drastically lower our throughput, and could lead to inconsistent results, since the supporting sets and homology models would not necessarily be the same even if they are of the same protein domain. A person submitting multiple mutations for the same job would not be unexpected; someone may decide do perform alanine scanning of an interface, etc. This requires the existence of a process which would monitor jobs and submit mutations only when Provean and MODELLER jobs have finished.

2. We have to keep track of all running mutations in order to be able to send a “Job complete” email when all mutations for a particular job have been calculated.

3. SGE jobs can die unexpectedly if they exceed allocated resources or if the node goes down. In most cases, the jobs do not have an opportunity to send an error message. Therefore, we could not assume that a job is finished successfully if it is no longer running on the cluster and if it has no error messages in the log files. The job may simply have been killed before it had a chance to print an ERROR message.

One possible way to address this problems would be to use an asyncronous task queue, such as Celery. However, Celery workers do not share the same memory state, and therefore each worker would have to execute its own *qstat* calls on the SGE master, quickly overloading it with connections and requests. It would not be possible to monitor all mutations for a particular job. The “job finished” commands could only be submitted as a cron job by querying the database and looking for finished and / or failed jobs.

and we ultimately opted to have a custom asynchronous task queue that was tailored to ELASPIC and would execute and monitor ELASPIC jobs on a local Sun Grid Engine (SGE) cluster rather than inside the same virtual machine as the webserver.

Would require dedicating significant computational resources solely to ELASPIC, // the server would not scale with the fluctuations indemand.

2.3 Precalculated data

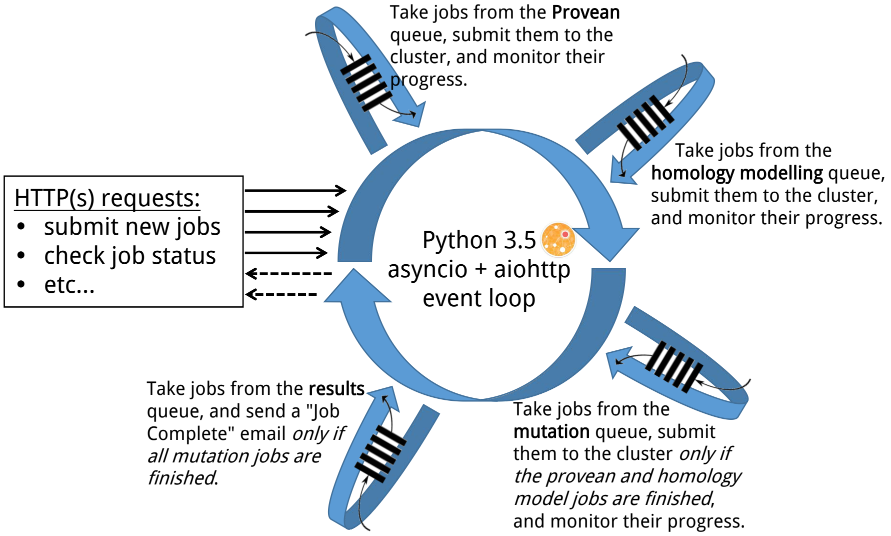
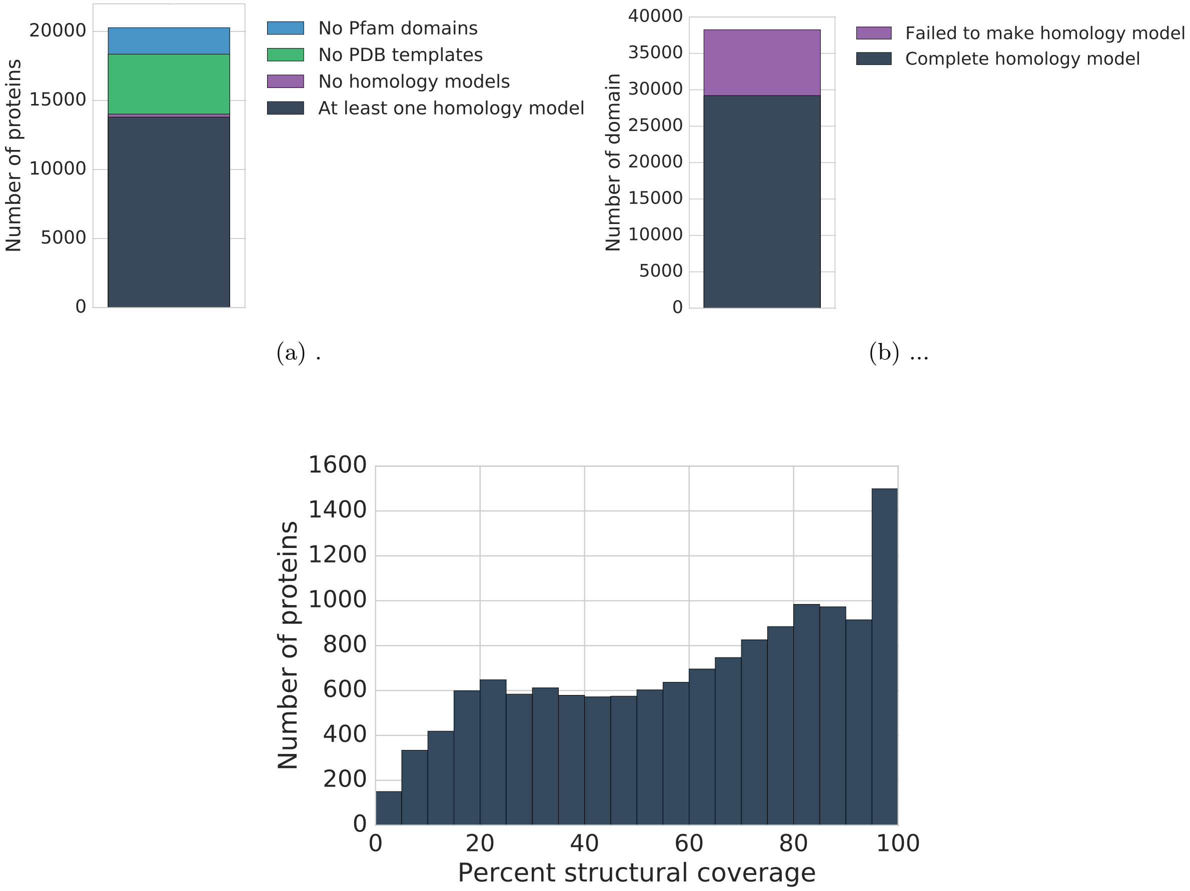

Take jobs from the **Provean** queue, submit them to the cluster, and monitor their progress.
Take jobs from the **homology modelling** queue, submit them to the cluster, and monitor their progress.
HTTP(s) requests:
• submit new jobs
• check job status
• etc...
Python 3.5
asyncio + aiohttp
event loop
Take jobs from the **results** queue, and send a "Job Complete" email *only if all mutation jobs are finished*.
Take jobs from the **mutation** queue, submit them to the cluster *only if the provean and homology model jobs are finished*, and monitor their progress.

Figure 2.7: Overview of the ELASPIC jobsubmitter.



(a) .

(b) ...

(c) ...

Figure 2.8: Statistics on homology modelling coverage.

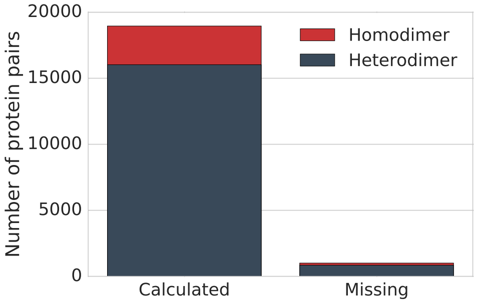


Figure 2.9: ...

Results

ELASPIC uses the gradient boosting of decision trees regressor (GBR). It was optimized in several ways.

ELASPIC described in output xxx features in total. 1. We calculated those features for the Provean and the Skempi training sets. 2. We removed features that were note different in any of the training cases (xxx for core mutations and yyy for interface mutations).

3. It has been reported that balancing the training set by including both positive and negative samples

As described in [], balancing the training set can significantly improve performance. However, with Provean balancing the training set can bias the result because most mutations are to unconserved amino acids (often alanine) and

We built two core predictors and two interface predictors:

1. No sequence features but a balanced training set.

2. Sequence features but no balanced training set.

- Accuracy over different sequence identity bins
- within protein correlation on the test set

3.1 Datasets

3.2 Predicting mutation induced $\Delta\Delta G$ of protein folding.

3.2.1 Gridsearch and feature elimination

3.2.2 Validation

3.3 Predicting mutation induced $\Delta\Delta G$ of protein-protein interactions.

3.3.1 Gridsearch and feature elimination

3.3.2 Validation

Table 3.3: Description of the datasets that were used in this study.

Name	Description	Type	Ref.
Protherm	Database of changes in the Gibbs free energy of protein folding caused by mutations ($\Delta\Delta G$).	Train	[38]
Skempi	Database of changes in the Gibbs free energy of protein folding caused by mutations ($\Delta\Delta G$).	Train	[39]
Taipale	Chaperone interaction assay measuring protein stability. Change in the interaction with various quality control factors (QCFs), measured using the LUMIER assay.	Validation	[40]
Taipale PPI	Yeast two hybrid studies measuring the effect of mutations on the presence / abscence of interactions.	Validation	[40]
Taipale GPCA	*Gaussia princeps* luciferase protein complementation assay measuring the effect of mutations on protein affinity.	Validation	[40]
Humsavar	Disease-causing mutations vs. polymorphisms. Mostly OMIM, old ClinVar, old COSMIC. 1 if the mutation is annotated with at least one disease in the UniProt *humsavar.txt* file. 0 if the mutation is annotated as “Polymorphism” in the UniProt *humsavar.txt* file.	Validation & Test	[34]
ClinVar	Disease-causing mutations with a weaker inheritence link than OMIM. 1 if the mutation is found in the ClinVar *clinvar_20160531.vcf* file. 0 if the mutation is found in the ClinVar *common_no_known_medical_impact_20160531.vcf* file.	Validation & Test	[36]
COSMIC	Mutations found in cancers. Use high-confidence FATHMM predictions. 1 if the mutation is predicted to be deleterious by FATHMM in the COSMIC database. 0 if the mutation is predicted to be benign by FATHMM in the COSMIC database.	Validation & Test	[35]
SUMO	Mutations affecting the activity of SUMO ligase, measured using a cell viability assey.	Test	[41]
AB-Bind	Antibody affinity maturation experiments.	Test	[42]
Benedix	Alanine scanning of the TEM1 (β-lactamase) – BLIP (β-lactamase-inhibitor) complex.	Test	[14]

Table 3.4: Core predictor parameters.

Parameter label	Parameter description	Parameter value
...	...	

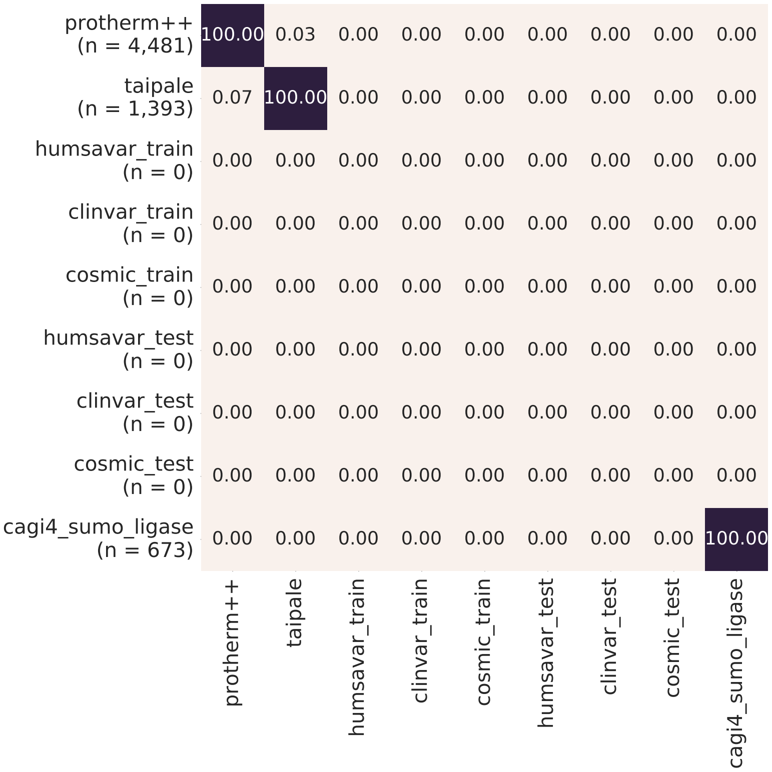


Figure 3.10: Size and overlap of core and interface datasets.

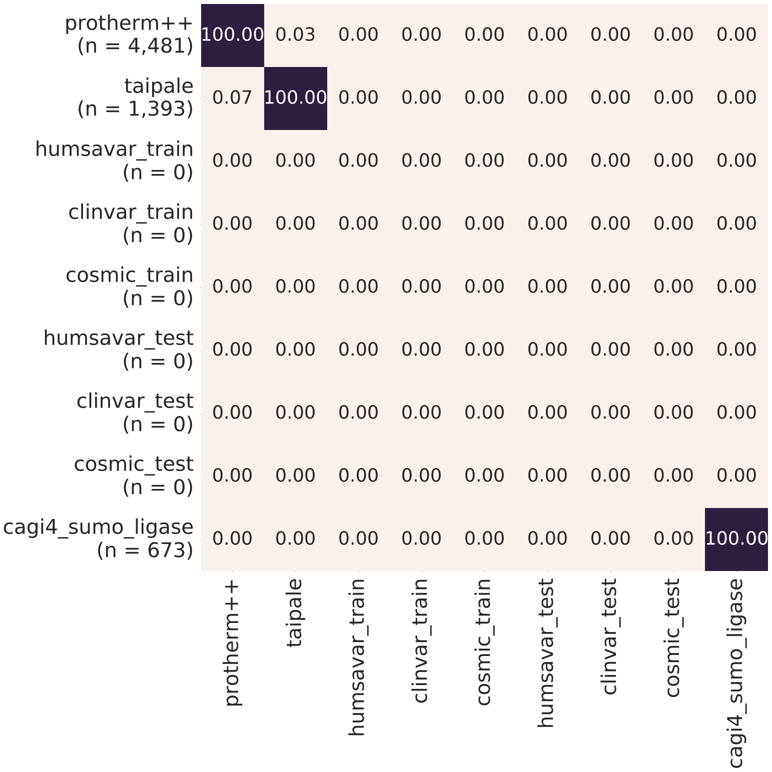
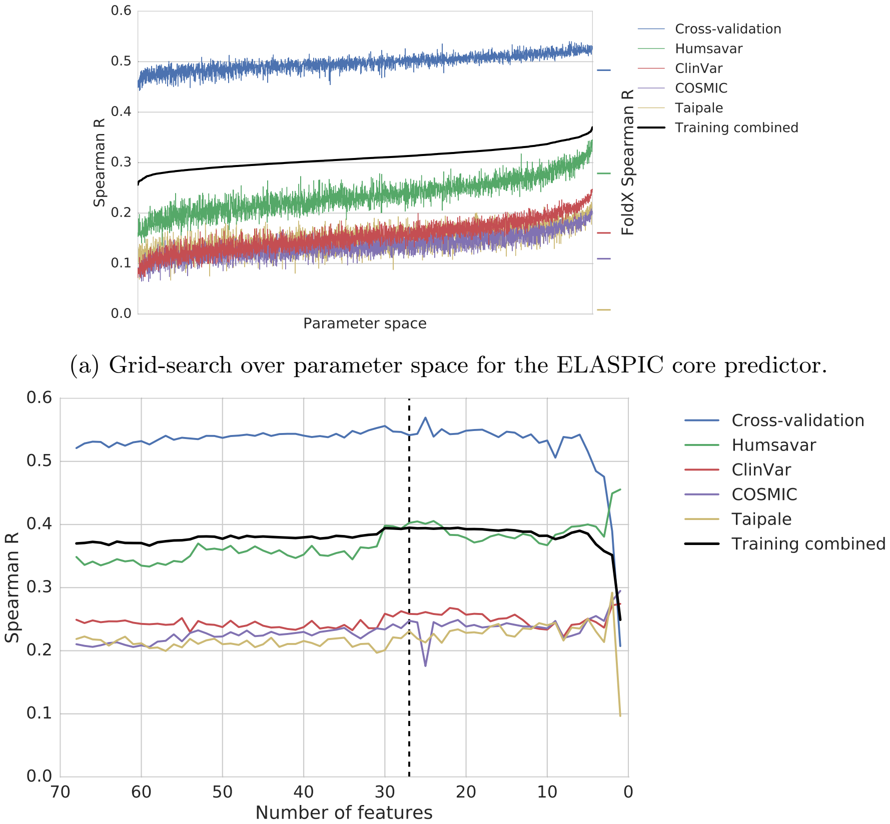


Figure 3.11: Size and overlap of core and interface datasets.



(a) Grid-search over parameter space for the ELASPIC core predictor.

(b) Feature elimination curve for the ELASPIC core predictor.

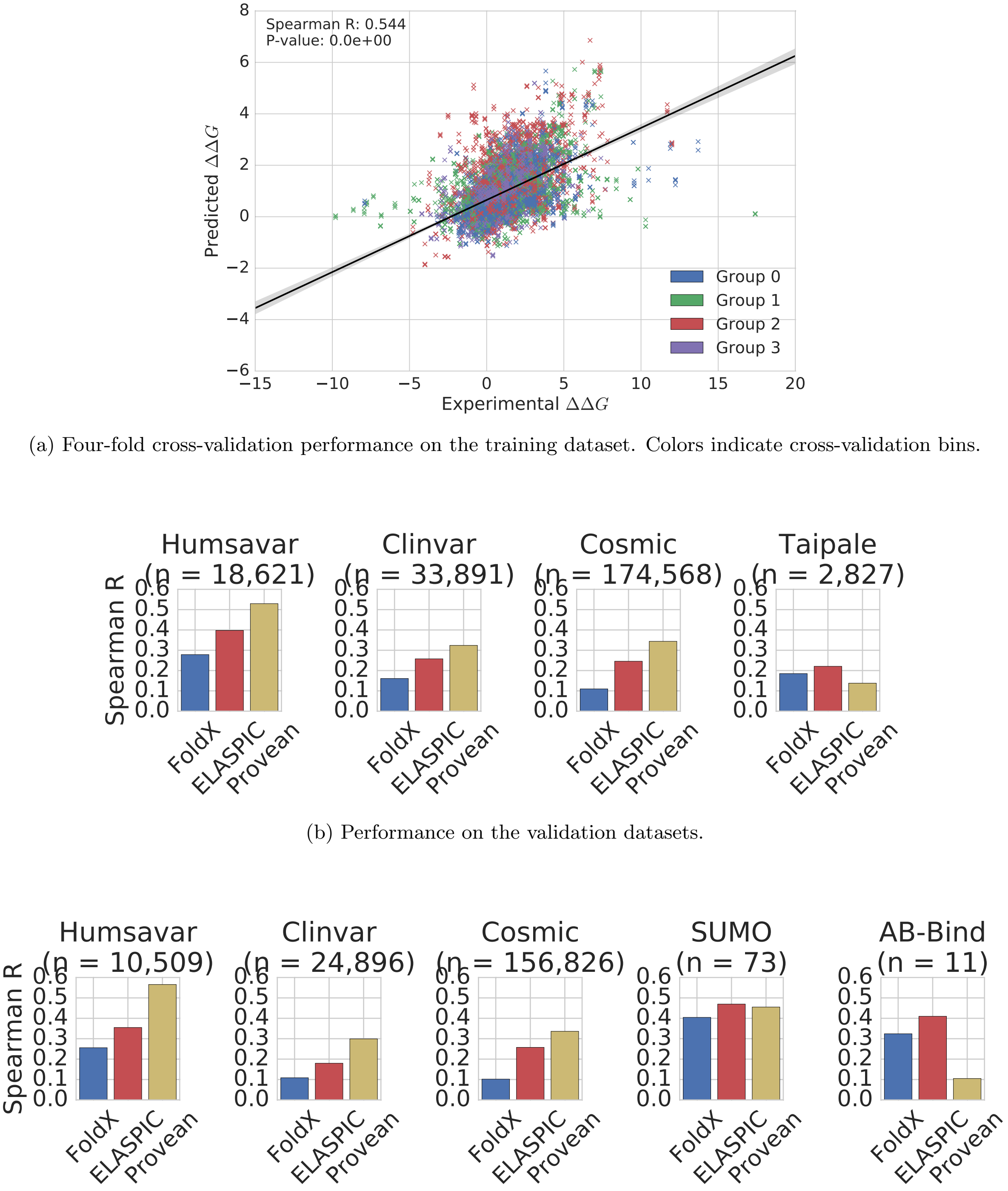
Figure 3.12: Training the core predictor.

Table 3.5: Core predictor features.

alignment_coverage	Alignment quality		
alignment_identity	Alignment quality		
alignment_score	Alignment quality		
backbone_hbond_change	FoldX		
backbone_hbond_wt	FoldX		
cis_bond_wt	FoldX		
disulfide_wt	FoldX		
electrostatic_kon_change	FoldX		
electrostatics_change	FoldX	*	
entropy_mainchain_change	FoldX		
helix_dipole_wt	FoldX		
matrix_score	Sequence conservation		
pcv_hbond_change	Physico-chemical features		
pcv_hbond_self_change	Physico-chemical features		
pcv_salt_equal_change	Physico-chemical features		
pcv_salt_equal_self_wt	Physico-chemical features		
pcv_salt_equal_wt	Physico-chemical features		
pcv_salt_opposite_change	Physico-chemical features		
pcv_vdw_self_change	Physico-chemical features		
provean_score	Sequence conservation	**	
sloop_entropy_wt	FoldX		
solvation_hydrophobic_change	FoldX	*	
solvation_polar_change	FoldX	**	
solvent_accessibility_wt	FoldX	*	
torsional_clash_change	FoldX		
van_der_waals_clashes_change	FoldX	*	
water_bridge_wt	FoldX		

Table 3.6: Interface predictor parameters.

Parameter label	Parameter description	Parameter value
...	...	



(a) Four-fold cross-validation performance on the training dataset. Colors indicate cross-validation bins.

(b) Performance on the validation datasets.

(c) Performance on the test datasets.

Figure 3.13: Performance of the core predictor on the training (a), validation (b) and test sets (c).

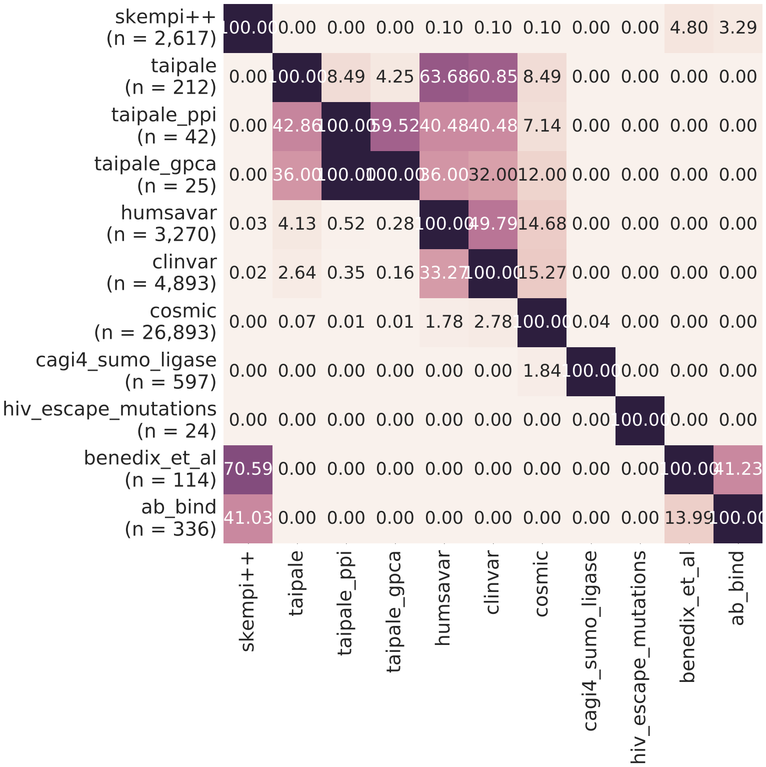
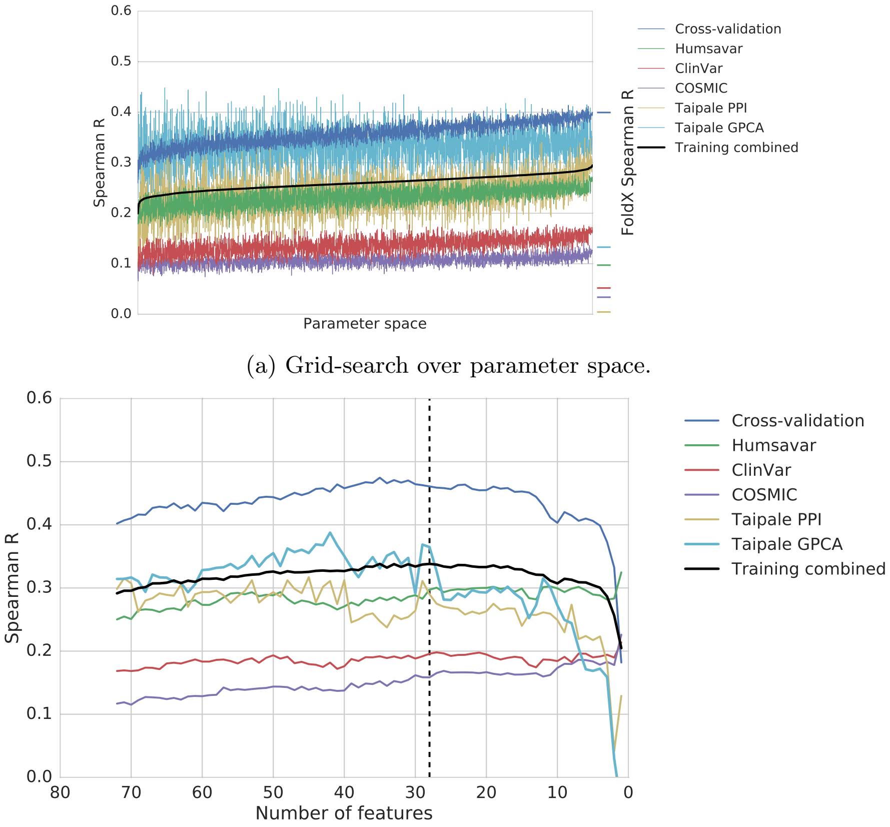


Figure 3.14: Size and overlap between the core and interface predictor datasets.



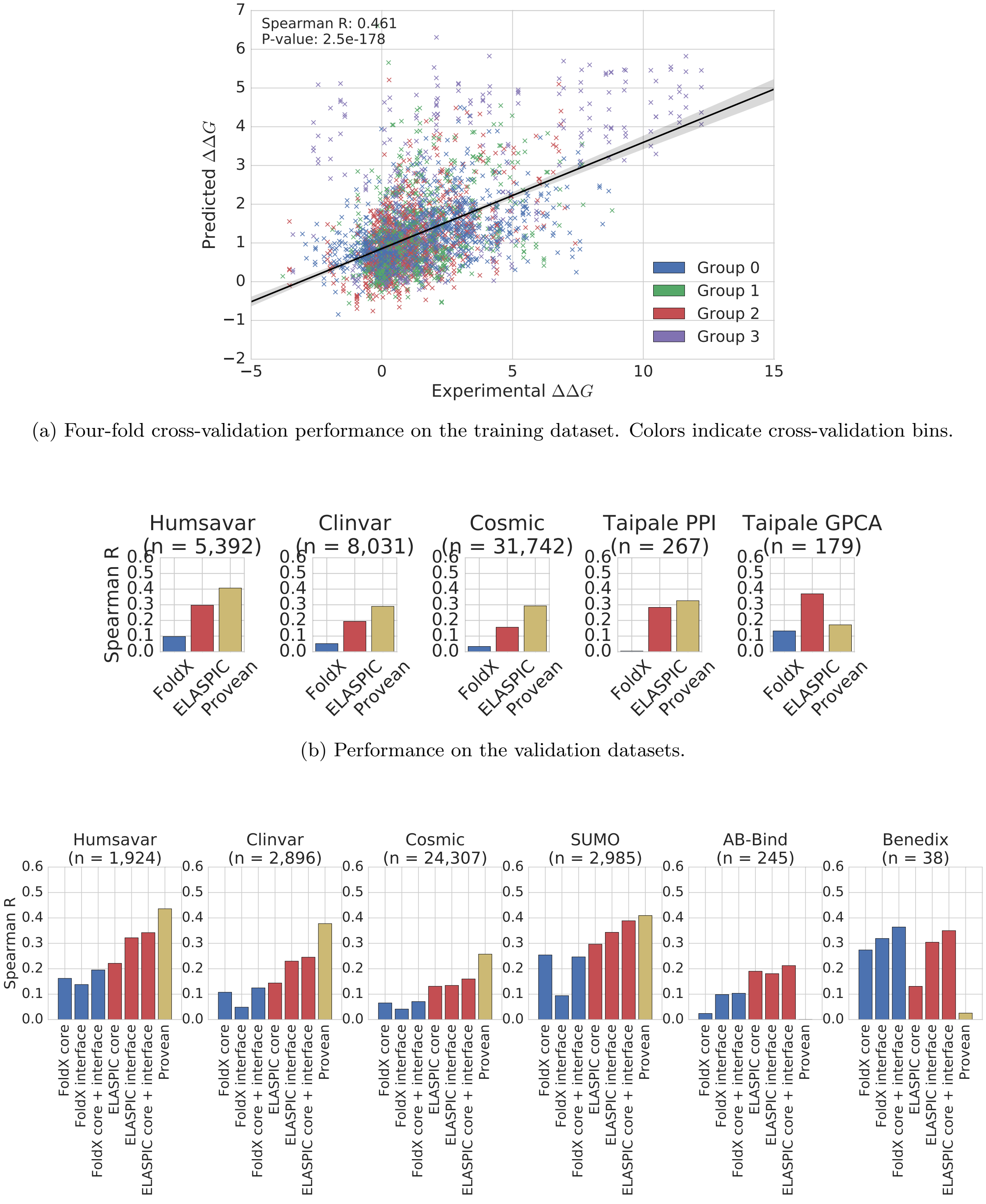
(a) Grid-search over parameter space.

(b) Feature elimination.

Figure 3.15: Training the interface predictor.

Table 3.7: Interface predictor features.

Feature name	Feature description	...
alignment_score	Alignment quality	
backbone_clash_change	FoldX	
backbone_clash_wt	FoldX	
backbone_hbond_change	FoldX	
cis_bond_wt	FoldX	
electrostatic_kon_wt	FoldX	
energy_ionisation_wt	FoldX	
entropy_complex_change	FoldX	
entropy_sidechain_change	FoldX	*
intraclashes_energy_2_change	FoldX	
partial_covalent_bonds_wt	FoldX	*
pcv_hbond_self_change	Physico-chemical features	
pcv_hbond_wt	Physico-chemical features	
pcv_salt_equal_self_change	Physico-chemical features	
pcv_salt_equal_wt	Physico-chemical features	
pcv_salt_opposite_change	Physico-chemical features	
pcv_salt_opposite_self_change	Physico-chemical features	
pcv_salt_opposite_self_wt	Physico-chemical features	
pcv_vdw_self_change	Physico-chemical features	
pcv_vdw_self_wt	Physico-chemical features	*
pcv_vdw_wt	Physico-chemical features	*
provean_score	Sequence conservation	*
sloop_entropy_change	FoldX	
solvation_hydrophobic_change	FoldX	
solvation_polar_change	FoldX	*
solvation_polar_wt	FoldX	
torsional_clash_change	FoldX	
water_bridge_change	FoldX	



(a) Four-fold cross-validation performance on the training dataset. Colors indicate cross-validation bins.

(b) Performance on the validation datasets.

(c) Performance on the test datasets.

Figure 3.16: Performance of the interface predictor on the training (a), validation (b) and test sets (c).

Discussions

We saw mixed results with the

3.1 Published post factum

VIPUR [43]

MutaBind [44].

3.2 Limitations

Cystic fibrosis

- Existing approaches remain limited in their ability to predict disease-causing variants. In a study of 1571 mutations of the CFTR gene causing cystic fibrosis, (SIFT, PolyPhen, PANTHER) [45]

Long QT syndrome

- Assessment of the predictive accuracy of five *in-silico* prediction tools, alone or in combination, and two meta-servers to classify long QT syndrome gene mutations.

- `http://www.ncbi.nlm.nih.gov/pubmed/25967940`

3.3 Protein science

Results of feature elimination support the view that electrostatics, van der waals forces and entropy are the main forces determining the effect of mutations, as suggested by

3.4 Future directions

eSCOP

Gene3D

- Use sequence profiles (e.g. Pfam or Gene3D) to guide the alignment.

3.5 Better features

Most structural features play a surprisingly small role in the performance of the ELASPIC predictor. Either those features are not informative, or our training set is too noisy for the contribution of those features to come through.

- Use covariation between amino acids in addition tho the conservation score to predict the impact of mutations, as described by Kowarsch et. al. [46].
- Standard conservation metrics, such as Provean, may predict a certain substitution to be benign because it occurs in other organisms. However, this does not take into account any potentially covarying mutations that mask the deleterious effect of the mutation in question.

- Use multiple templates when building the homology models.
- Create multiple models and choose the one with the highest DOPE score.
- Refine the model using molecular dynamics.

Long-term MD is not useful for optimizing structures in most cases [47].

3.5.1 Multi-task learning

Construct a shared representation for related problems in order to

In this work, we attempted to improve the performance of ELASPIC by keeping track of its performance on mutation deleteriousness datasets throughout cross-validation and feature selection. While this approach should prevent us from selecting a predictor which is over-fitted on the training dataset, it does not improve the pool of predictors from which we make this selection.

One way in which we could use information from the mutation deleteriousness datasets directly in the ELASPIC predictor is by training a boosted decision tree model to predict the mutation deleteriousness score, and using the output of the trained model as input to logistic regression which is trained to predict the $\Delta\Delta G$ of mutations. A similar approach was used successfully by a group at Facebook to predict clicks on adds [48]. This approach would have an additional advantage, in that since we use a linear model to predict the final $\Delta\Delta G$, it should be able to extrapolate outside the values present in our training set.

An additional advantage is that the feature learning part of the predictor would be done on a much larger dataset, allowing the sequential and structural features to "mix" in a more general environment.

"The resulting transformer has then learned a supervised, sparse, high-dimensional categorical embedding of the data."

`http://scikit-learn.org/stable/auto_examples/ensemble/plot_feature_transformation.html#example-ensemble-plot-feature-transformation-py`

3.6 Multi-residue mutations

ELASPIC can easily be extended to calculate the $\Delta\Delta G$ for mutations involving multiple amino acids. The tricky part is that the number of features changes with the number of amino acids that are mutated. We could address this by treating a mutation affecting multiple amino acids as a set of single amino acid mutations. For example, we could use the following recursive strategy:

1. Introdue each of the single amino acid mutations, one at a time.

2. Select the single amino acid mutation with the most stabilizing effect.

3. Repeat for the remaining mutations, using the structure containing the mutation selected in Step 2.

About one third on mutations in the Protherm and Skempi databases affect multiple amino acids. We could include those mutations in the training set by dividing them into single amino acid mutations and assigning to them a $\Delta\Delta G$ proportional to their contribution to the overall mutation score, as determined by the multiple amino acid substitution version of ELASPIC. This would require “bootstrapping” the ELASPIC predictor using single amin acid mutations, using the “bootstrapped” predictor to approximate the contribution of single amino acid mutaitons to the $\Delta\Delta G$ affecting mulitple amino acids, adding those mutations to the training set, and repeating.

In the case of the ELASPIC core predictor, we could create a dataset of multiple amino acid polymorphisms (MAAMs) from a thermophilic bacterium and it’s closest non-thermophilic relative (maybe such a database already exists?). Cross-validate ELASPIC making sure that we predict those MAAMs to be stabilizing. Incorporate those MAAMs into our training set, weighting them accordingly.

In the case of the ELASPIC interface predictor, we could construct a dataset from phage-display read counts, and cross-validate ELASPIC while keeping track of its performance on phage display counts. Could then recursively incorporate the phage display data into the training set, weighting it by how well the ELASPIC predictor does on those mutations, as determined through cross-validation.

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, as the former is more likely to induce changes in the conformation of the protein that are not modelled by ELASPIC. This drop in performance could in-part be ameliorated by including a backbone relaxation step between each mutation, using molecular dynamics [49], Rosetta Backrub [50], or other algorithms [51].

If the ELASPIC predictor can achieve reasonable results for mutations affecting multiple amino acids, it could be used “in reverse” to design protein domains with increased stability and protein interfaces with increased affinity.

FireProt: Energy- and Evolution-Based Computational Design of Thermostable Multiple-Point Mutants

- `http://journals.plos.org/ploscompbiol/article?id=10.1371%2Fjournal.pcbi.1004556`

- Predict the structural effect of multiple mutations.

- “Stability effects of all possible single-point mutations were estimated using the ¡BuildModel¿ module of FoldX”.

- We demonstrate that thermostability of the model enzymes haloalkane dehalogenase DhaA and -hexachlorocyclohexane dehydrochlorinase LinA can be substantially increased.

- [52]

HOPE THAT PROVEAN WOULD AT LEAST PARTIALLY MAKE UP FOR THE LIMITING ASSUMPTION THAT THE BACKBONE REMAINS STABLE BETWEEN MUTATIONS.

SCIENTIFICALLY INTERESTING TO SEE WHAT EFFECT MD RELAXATIONS WOULD HAVE ON THE PERFORMANCE OF THE ALGORITHM.

3.7 Additional interaction types

3.7.1 Protein-protein interactions

Predict PPIs: PRISM: Protein interaction by structure matching.

3.7.2 Protein-ligand interactions

- drugging protein-protein interfaces [53]

Platinum: Protein-ligand affinity change upon mutation database.

- `http://bleoberis.bioc.cam.ac.uk/platinum/`

BioLiP is a semi-manually curated database for high-quality, biologically relevant ligand-protein binding interactions.

- `http://zhanglab.ccmb.med.umich.edu/BioLiP/`

- The structure data are collected primarily from the Protein Data Bank, with biological insights mined from literature and other specific databases.

3.7.3 Protein-DNA/RNA interactions

ProNIT

RBPDB: a database of RNA-binding specificities

`http://rbpdb.ccbr.utoronto.ca`

Paper: `http://nar.oxfordjournals.org/content/39/suppl_1/D301`

3.7.4 Protein-peptide interactions

ELM

3.7.5 Phosphorylated residue-mediated interactions

Bibliography

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

[2] 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`.

[3] 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`.

[4] 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.

[5] 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`.

[6] 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.

[7] 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`.

[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] 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`.

[11] Brett D. Welch et al. “Potent D-Peptide Inhibitors of HIV-1 Entry”. In: *Proceedings of the National Academy of Sciences* 104.43 (October 23, 2007), pp. 16828–16833. DOI: `10.1073/pnas.0708109104`.

[12] 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`.

[13] 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.

[14] 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.

[15] Piero Fariselli et al. "INPS: Predicting the Impact of Non-Synonymous Variations on Protein Stability from Sequence". In: *Bioinformatics* 31.17 (January 9, 2015), pp. 2816–2821. DOI: 10.1093/bioinformatics/btv291.

[16] 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.

[17] Shane Ó Conchúir et al. "A Web Resource for Standardized Benchmark Datasets, Metrics, and Rosetta Protocols for Macromolecular Modeling and Design". In: *PloS One* 10.9 (2015), e0130433. DOI: 10.1371/journal.pone.0130433.

[18] Vladimir Potapov et al. "Assessing Computational Methods for Predicting Protein Stability upon Mutation: Good on Average but Not in the Details". In: *Protein Engineering Design and Selection* 22.9 (January 9, 2009), pp. 553–560. DOI: 10.1093/protein/gzp030.

[19] Sofia Khan and Mauno Vihinen. "Performance of Protein Stability Predictors". In: *Human Mutation* 31.6 (June 1, 2010), pp. 675–684. DOI: 10.1002/humu.21242.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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`.

[29] 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`.

[30] 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`.

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

[32] 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`.

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

[34] 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`.

[35] 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`.

[36] 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`.

[37] 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`.

[38] 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`.

[39] 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`.

[40] 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`.

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

[42] 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`.

[43] 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`.

[44] 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`.

[45] 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`.

[46] Andreas Kowarsch et al. "Correlated Mutations: A Hallmark of Phenotypic Amino Acid Substitutions". In: *PLoS Comput Biol* 6.9 (September 16, 2010), e1000923. DOI: `10.1371/journal.pcbi.1000923`.

[47] Alpan Raval et al. "Refinement of Protein Structure Homology Models via Long, All-Atom Molecular Dynamics Simulations". In: *Proteins: Structure, Function, and Bioinformatics* 80.8 (August 1, 2012), pp. 2071–2079. DOI: `10.1002/prot.24098`.

[48] 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`.

[49] 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`.

[50] 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`.

[51] 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`.

[52] David Bednar et al. "FireProt: Energy- and Evolution-Based Computational Design of Thermostable Multiple-Point Mutants". In: *PLOS Comput Biol* 11.11 (November 3, 2015), e1004556. DOI: `10.1371/journal.pcbi.1004556`.

[53] James A. Wells and Christopher L. McClendon. "Reaching for High-Hanging Fruit in Drug Discovery at Protein–protein Interfaces". In: *Nature* 450.7172 (December 13, 2007), pp. 1001–1009. DOI: `10.1038/nature06526`.