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

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

Advances in DNA sequencing technology have led to an enormous growth in the number of genome sequences that are available. Millions of single nucleotide polymorphisms (SNPs) have been implicated in thousands of diseases. However, the etiology by which the mutations cause or contribute to a disease are often unknown.

Interpreting the variation in those genome sequences in order arrive at actionable results remains a challenge. Evaluating experimentally the effect of all discovered mutations is not feasible both in terms of the time and the cost that would be required. Computational techniques have been developed to predict the effect of mutations and prioritize them for experimental validation.

Sequence-based tools are the de-facto standard for predicting whether variants in genome sequences are deleterious. Tools such as Ensembl VEP, ANNOVAR and SNPeff can leverage databases of pre-calculated scores (dbNSFP) to annotate VCF files with the score predicted by each of the tools. Such algorithms generally fall into three categories:

Sequence-based approaches which predict the effect of mutation by using different metrics describing the conservation of a particular residue. Examples include Mutation Assessor, CONDEL (ensemble approach)

The most widely-used program is Sorting Intolerant from Tolerant or **SIFT**. SIFT creates an extensive multiple sequence alignment for every protein, and produces a conservation score based on the likelihood of the wildtype and mutant amino acids occurring at a given position. However, we had difficulty compiling and running SIFT in a cluster setting. Furthermore

Another widely-used mutation scoring tool is **PolyPhen-2**. However, it is trained on a dataset of deleterious and neutral human mutations. This would make it difficult for us to run benchmarks, since we would have to be meticulous to ensure that the validation set that we are using does not contain mutations that are in the PolyPhen-2 training set.

Another popular sequence-based algorithm is **Provean**. Provean is comparable to the leading mutation scoring programs, such as SIFT, PolyPhen-2, Mutation Assessor, and CONDEL [1]. Furthermore, Provean is distributed under a GPLv3 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.

Other sequence-based algorithms include FATHMM, CONDEL, MutationAssesor, MutPred, and others.

Despite the proliferation of tools predicting the deleteriousness of different SNPs, our ability to act on those predictions remains limited. 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.

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 [2]. 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 [3].

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 [4] (graph-based signatures), MAESTRO [5] (multi-agent machine learning), CC/PBSA (Concoord/Poisson-Boltzmann surface area) [6],

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.

- [7]

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".

- [4]

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".

- [8]

1.0.1 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

1.1 Published post factum

VIPUR [9]

MutaBind [10].

1.2 Benchmarks

Rosetta benchmark [11]

Benchmark showing Rosetta doing poorly: [12]

I-Mutant2, DMutant, CUPSAT, FoldX [13]

1.3 Acknowledgements

This is a continuation of the work performed by Niklas Berliner *et al.* [14]. In 1.3 we discuss how we expand ELASPIC to work on the genome-wide scale. In 2.8.2 we discuss how we retrained ELASPIC while leveraging the information we extracted from genome-wide analysis.

Implementation

- Statistics on homology modelling coverage

2.1 ELASPIC pipeline

2.2 Homology modeling

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."

2.2.1 Standalone pipeline

2.1 right

An overview of the ELASPIC pipeline is presented in Figure 2.1. 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" and a "Database Pipeline", which include command line options for mutating a protein structure.

2.2.2 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.

2.2.3 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 2.1 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.

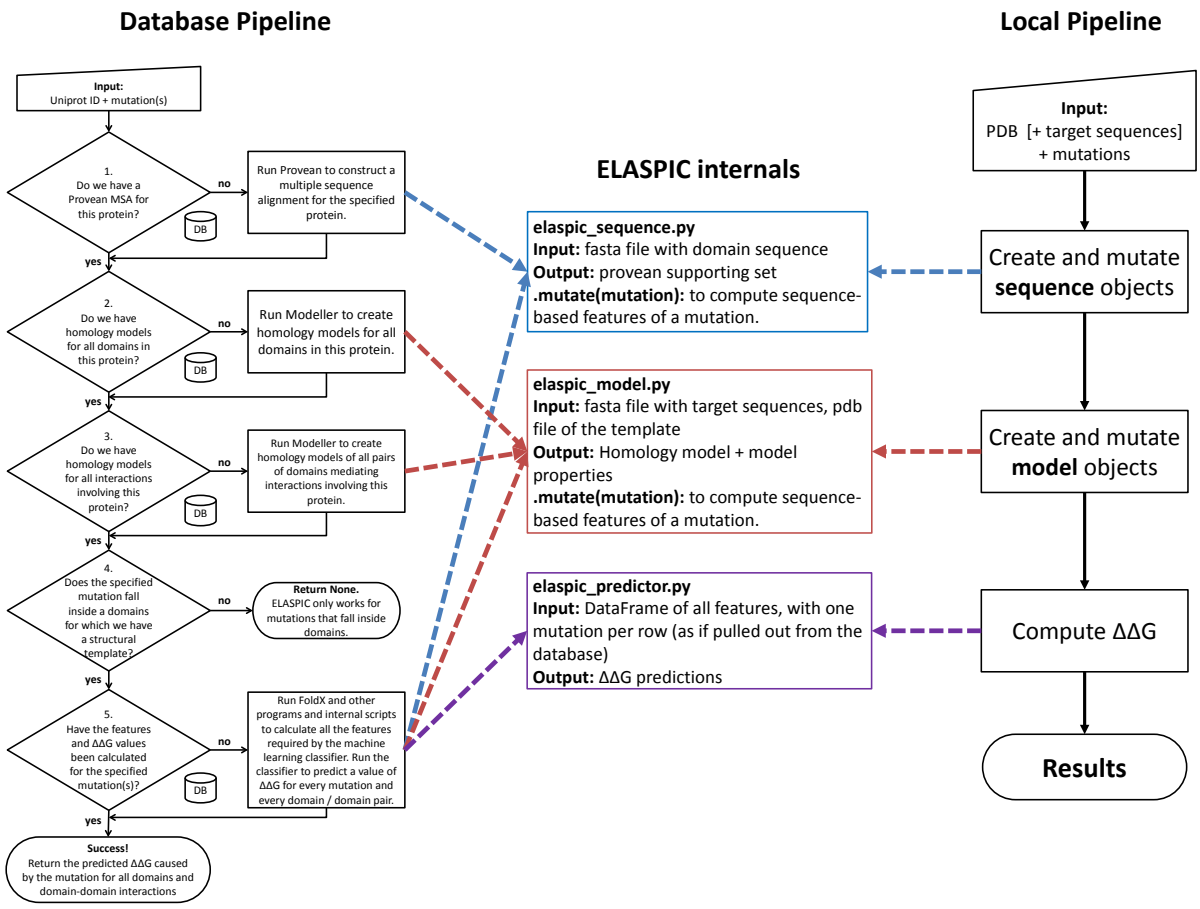


Figure 2.1: Overview of the ELASPIC pipeline. A user runs the ELASPIC pipeline specifying the UniProt id of the protein being mutated, and one or more mutations affecting that protein. At each decision node, the pipeline queries the database to check whether or not the required information has been calculated 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.

2.3 ELASPIC predictor

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

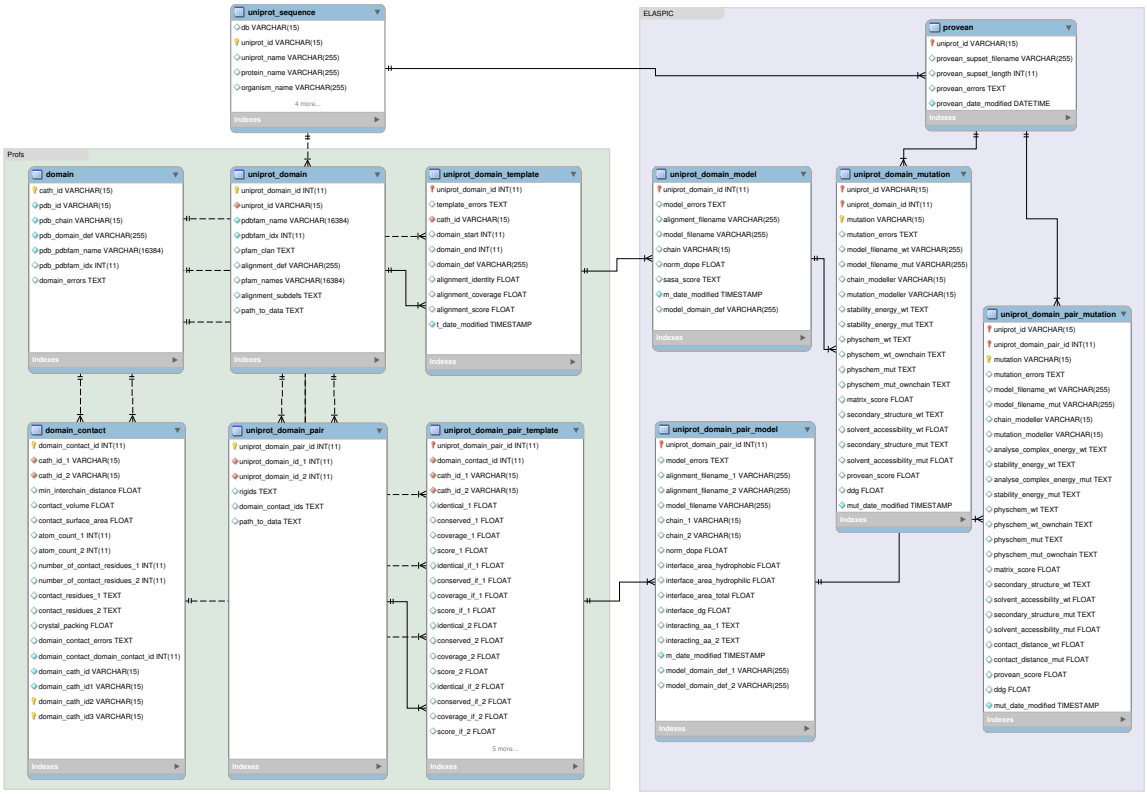


Figure 2.2: Database schema used by the ELASPIC pipeline. Tables on the green plate titled Profs are calculated using the Profs pipeline, as described in [15]. Tables on the purple plate titled ELASPIC are calculated using the ELASPIC pipeline, following the procedure outlined in 2.1. A detailed description of each table can be found in 2.1.

2.3.1 Training

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.

2.3.2 Validation

Compare how well Provean, FoldX, and 'ELASPIC with Provean' and 'ELASPIC without Provean' distinguish between the three different datasets for both core and interface mutations.

- Chaperone interaction data (core mutations) Luciferase complementation assay (interface mutations) (use Spearman correlation coefficient).
- Uniprot disease vs. polymorphism (use AUC / ROC / combination).
- COSMIC driver vs. passenger.

2.4 Structure features

The performance of Provean is comparable to the leading mutation scoring programs, such as SITF, PolyPhen-2, Mutation Assessor, and CONDEL [1]. 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

2.5 ELASPIC pipeline

The ELASPIC project was started by Niklas Berliner and others in 2014 [14].

ELASPIC uses Modeller [20] to construct homology models of domains and domain-domain interactions, FoldX to optimize those model and to introduce mutations [21], and the ELASPIC predictor 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. A flowchart describing the ELASPIC pipeline is presented in 2.1. 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, 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 G scores, have also been precalculated for the majority of mutations listed in the Uniprot humsavar file [22] and in the COSMIC [23] and ClinVar [24] databases.

Provean supporting sets, homology models and mutation 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.org/`.

2.6 Homology modelling of the human proteome

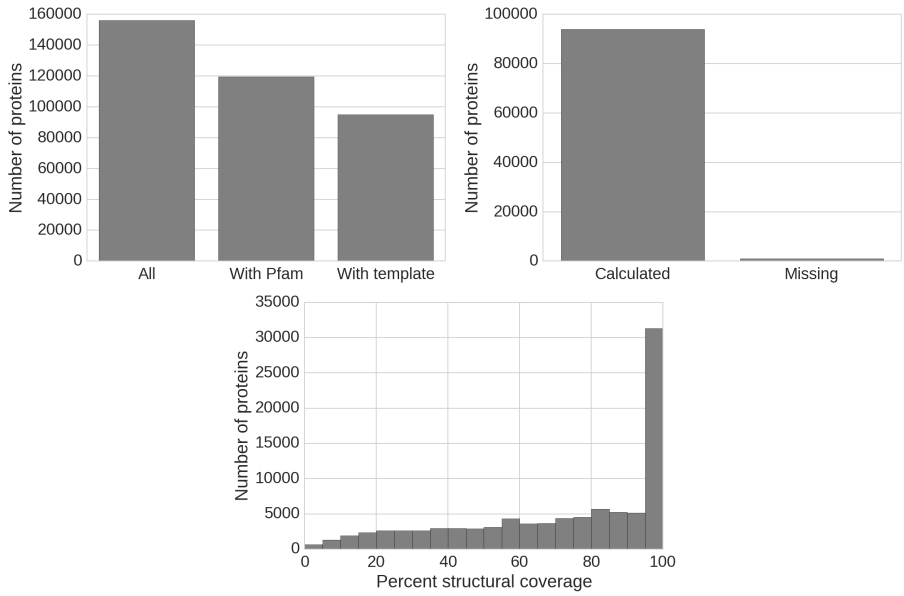


Figure 2.3: **Left**: statistics of how many homology models we were able to calculate. **Right**: structural coverage for proteins with at least one domain with a structural model.

2.7 ELASPIC web service

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.

2.8 Training sets

2.8.1 Core

Sanhi et al. databaset (taipale)

2.8.2 Interface

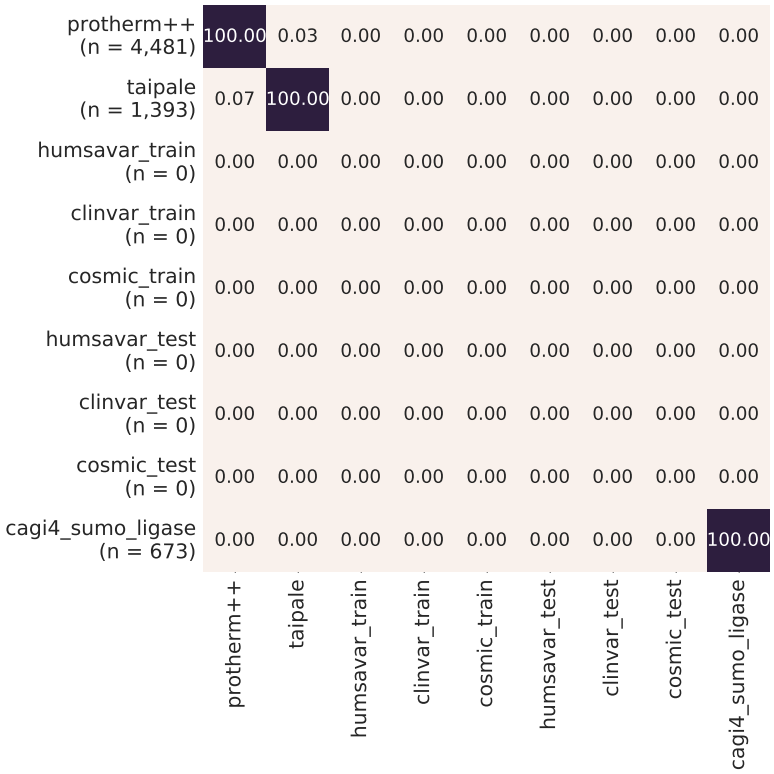


Figure 2.4: Size and overlap between the core and interface predictor datasets.

Table 2.1: ELASPIC database tables.

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 [16] 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 [1] 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 [17], 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 [15].
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 [18] and Rolland et al. [19].
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.

Results

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

3.1 Anti-proliferative peptides

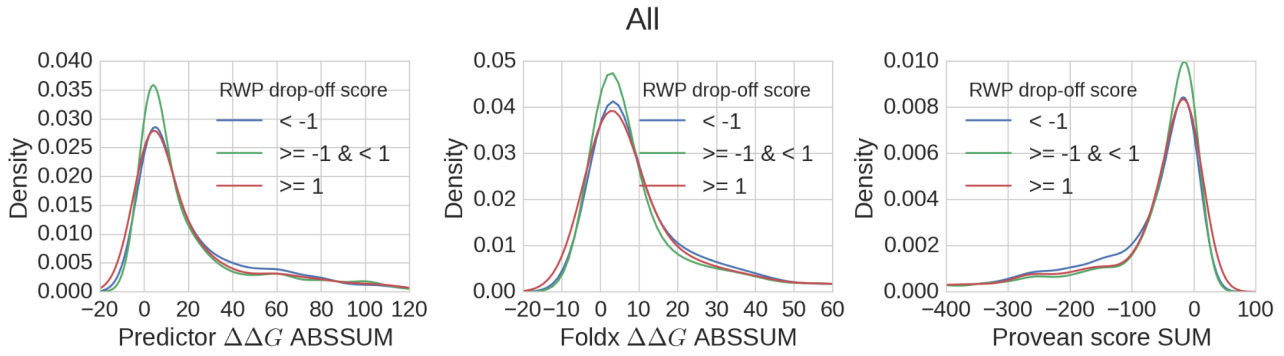
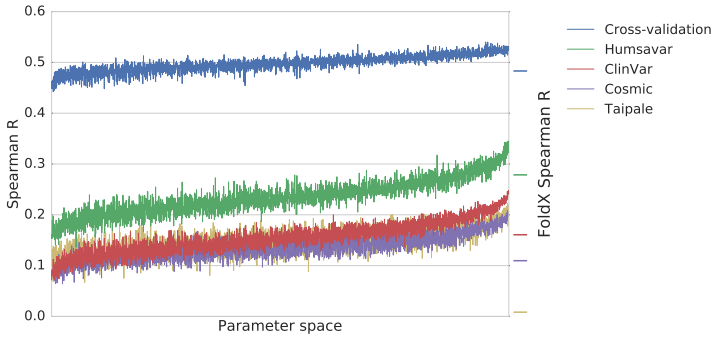


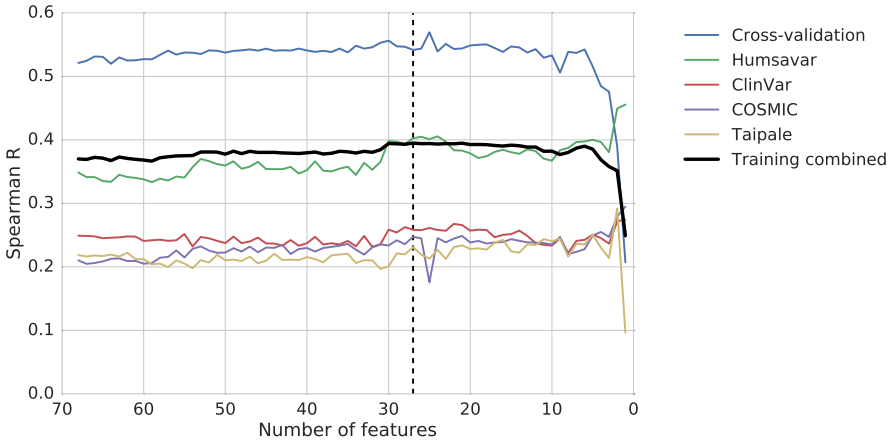
Figure 3.5: The drop-off score is higher for peptides that have a higher sum of the absolute interface FoldX energy.

Table 3.3: Description of the dataset used to train the core predictor.

Dataset name	Experimental feature
Protherm [25]	Change in the Gibbs free energy of protein folding ($\Delta\Delta G$).
Taipale [26]	Change in the interaction with various quality control factors (QCFs), measured using the LUMIER assay.
Humsavar [22]	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.
ClinVar [24]	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.
COSMIC [23]	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.



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



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

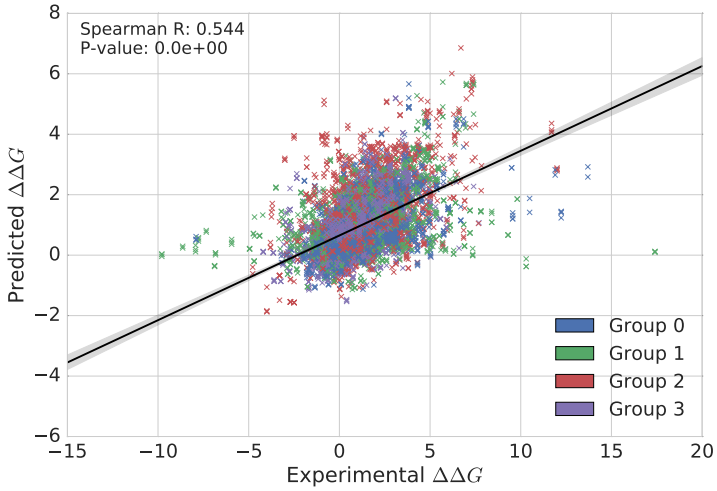
Figure 3.6: Core predictor training.

Table 3.4: Core features.

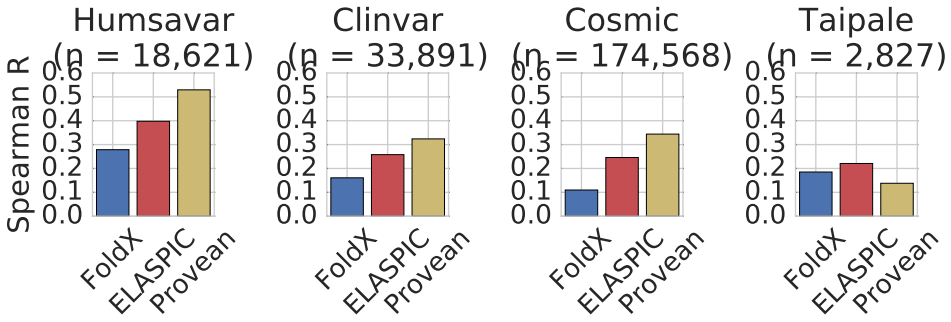
Feature name	Feature description
...	...

Table 3.5: Core parameters.

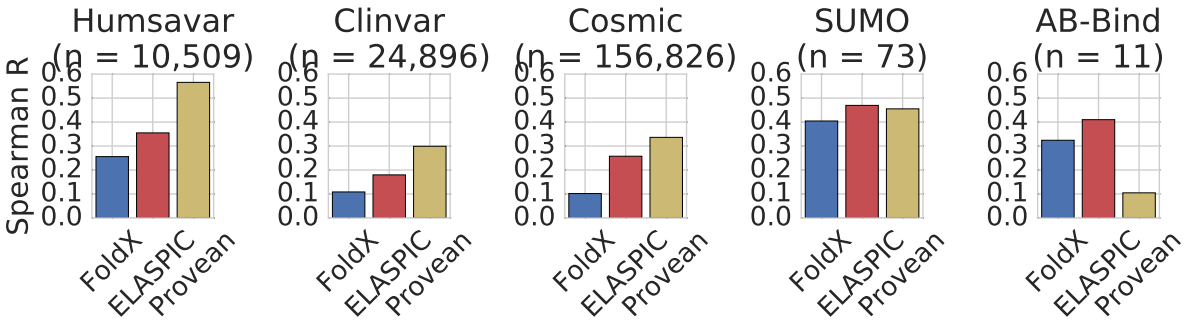
Parameter label	Parameter description	Parameter value
...	...	



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

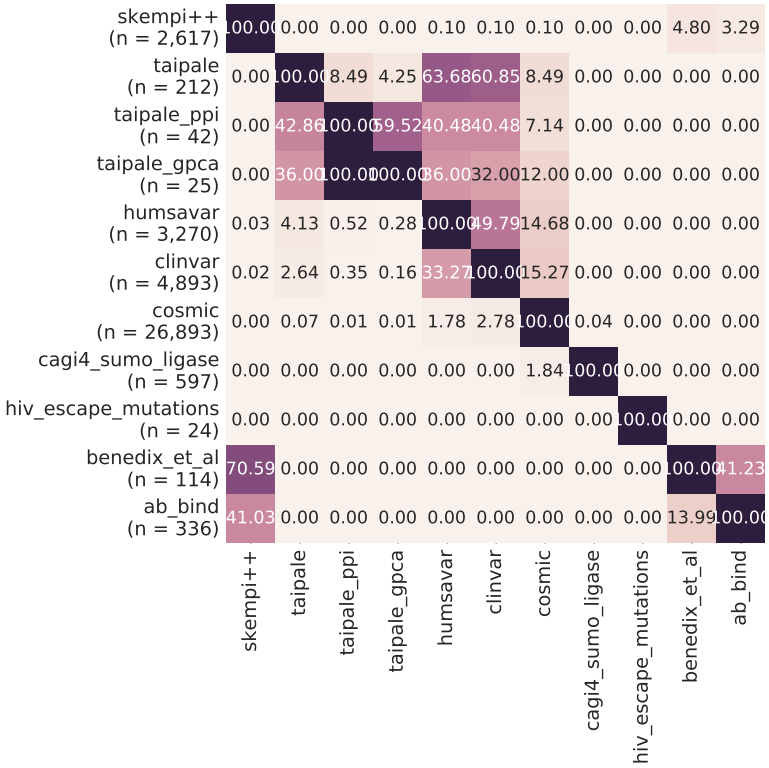


Figure 3.8: Size and overlap between the core and interface predictor datasets.

Table 3.6: Description of the dataset used to train the interface predictor.

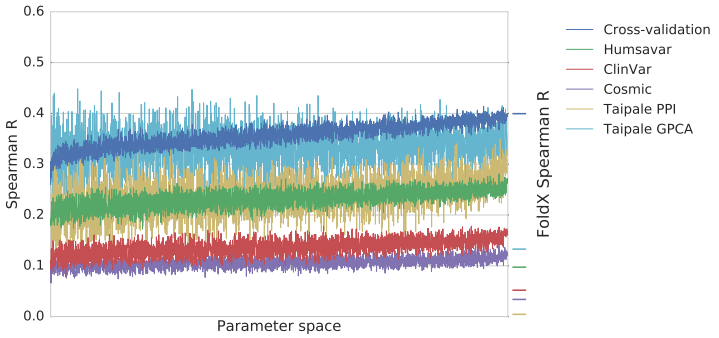
Dataset name	Experimental feature
Skempi [25]	Change in the Gibbs free energy of protein folding ($\Delta\Delta G$).
Taipale [26]	Change in the interaction with various quality control factors (QCFs), measured using the LUMIER assay.
Humsavar [22]	1 if the mutation is annotated with at least one disease in the UniProt *humsavar.txt* file. 0 if the mutation is annotated as "Polymorphism" in the UniProt *humsavar.txt* file.
ClinVar [24]	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.
COSMIC [23]	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.

Table 3.7: Interface features.

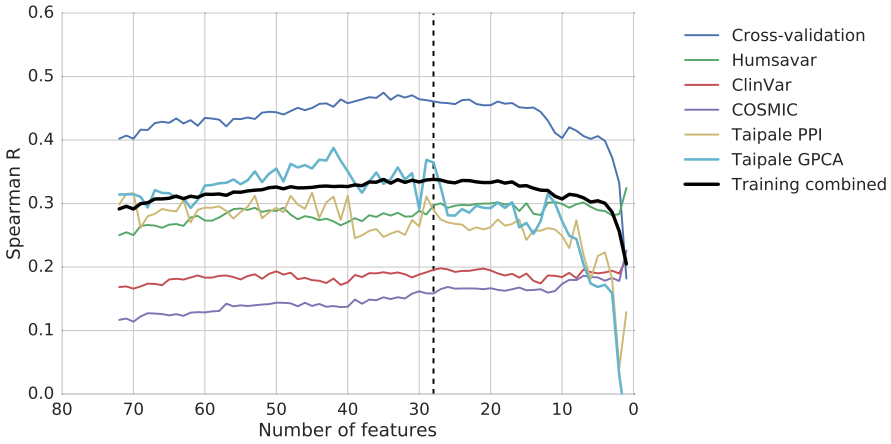
Feature name	Feature description
...	...

Table 3.8: Interface parameters.

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

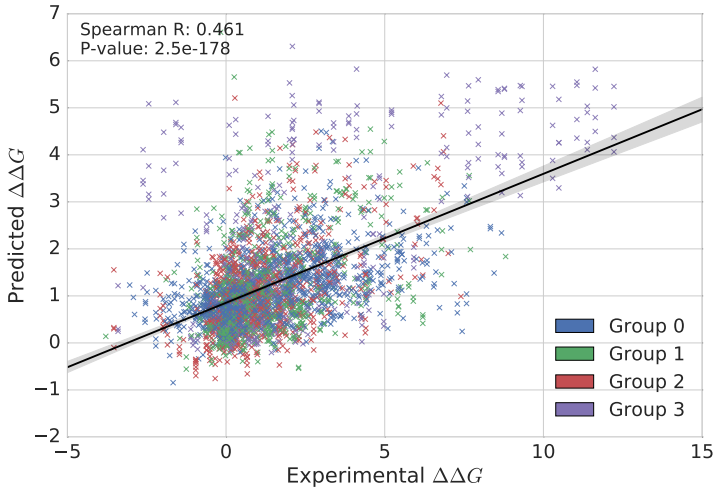


(a) Grid-search over parameter space.

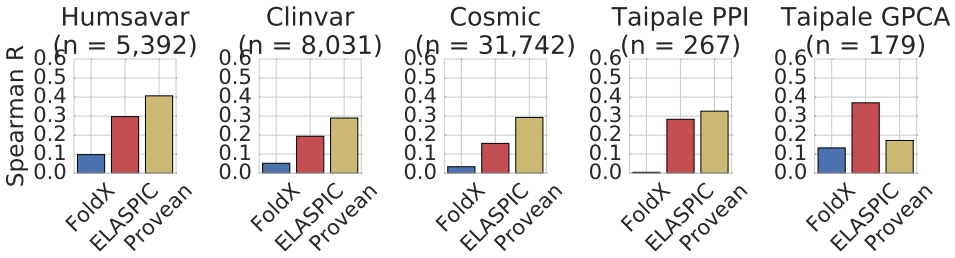


(b) Feature elimination.

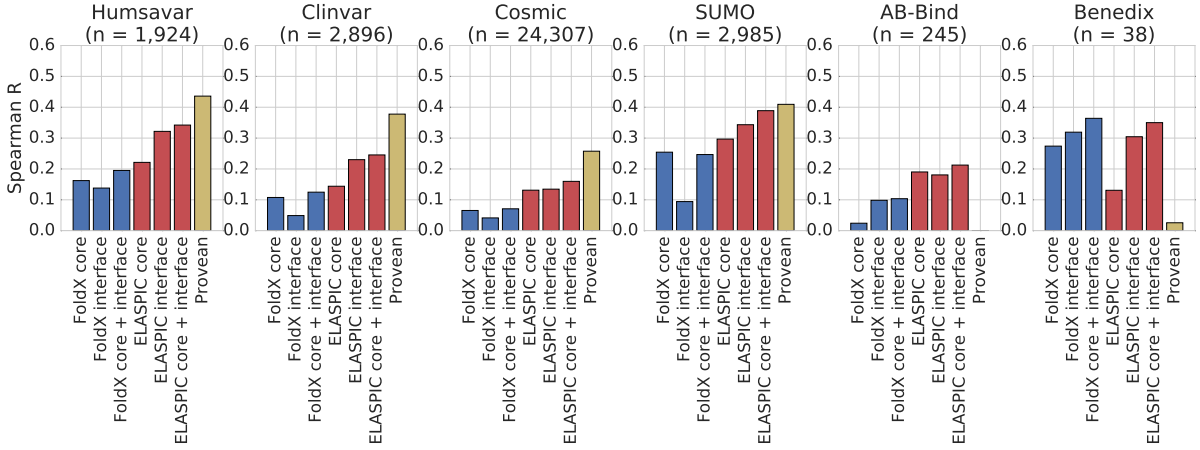
Figure 3.9: Interface predictor training.



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

Discussions

We saw mixed results with the

3.1 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) [27]

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`

3.2 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

3.3 Future Directions

eSCOP

Gene3D

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

3.4 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. [28].

- 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 [29].

3.4.1 Feature transformation using ensembles of trees

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 [30]. 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`

3.5 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 [31], Rosetta Backrub [32], or other algorithms [33].

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.

- [34]

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.

3.6 Additional interaction types

3.6.1 Protein-protein interactions

Predict PPIs: PRISM: Protein interaction by structure matching.

3.6.2 Protein-ligand interactions

- drugging protein-protein interfaces [35]

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.

3.6.3 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

3.6.4 Protein-peptide interactions

ELM

3.6.5 Phosphorylated residue-mediated interactions

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.