MU3DSP

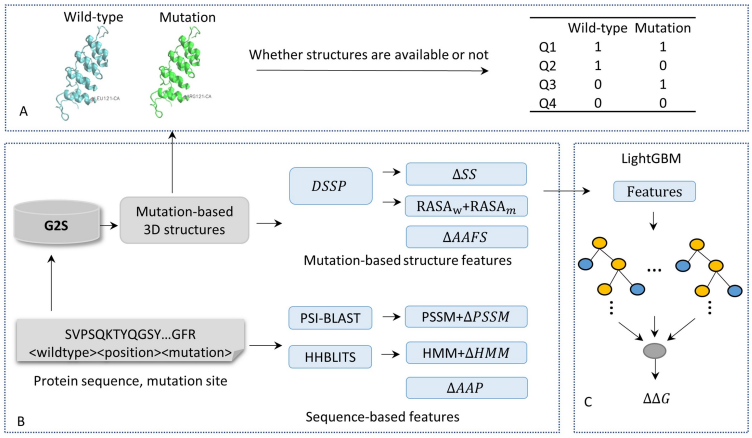
Release 0.1

Jianting Gong

Nov 18, 2022

MAIN

Figure 2. Overview of feature extraction and feature processing.



MU3DSP, a residue level 3D structure-based prediction tool to assess single point mutation effects on protein thermodynamic stability and applying to dingle-domain monomeric proteins. Given protein sequence with single mutations as the input, the proposed model integrated both sequence level features of mutant residues and residue level mutation-based 3D structure features.

CHAPTER

ONE

MU3DSP'S HIGHLIGHTS

- We propose a fast computational method, MU3DSP, based on LightGBM to predict stability changes upon single-residue mutations on proteins by fusing information from 3D structure profiles. The tertiary structure of the protein does not have to be available.
- MU3DSP uses a structure-based feature from either homology models of query variants if they are available or the annotated genomic variants database G2S (Genome to Structure).
- MU3DSP can achieve real-time prediction, and it only takes less than 1 minute on average to compute the same mutated position on one protein. A software tool is also provided to allow easy use of the tool.
- MU3DSP achieves state-of-the-art performance on two independent testing datasets. It is a reliable tool to assess both somatic and germline substitution mutations and assist in protein design.

CHAPTER

TWO

REFERENCE

CHAPTER

THREE

ACKNOWLEDGEMENTS

CHAPTER

FOUR

SUPPORT

Feel free to submit an issue or send us an email. Your help to improve MU3DSP is highly appreciated.

4.1 About

Mutation-induced protein thermodynamic stability changes (DDG) are crucial for understand protein biophysics, genomic variant interpretation, and mutation-related diseases. We introduce MU3DSP, a residue level 3D structure-based prediction tool to assess the effects of single point mutation on protein thermodynamic stability and applying to dingle-domain monomeric proteins. Given protein sequence with single mutations as the input, the proposed model integrated both sequence-level features of mutant residues and mutation-based structure features. Our stability predictor outperformed previously published methods on various benchmarks. MU3DSP will be a dedicated resource for assessing both somatic and germline substitution mutations in biological and medical research on genomics and proteomics.

4.1.1 What is DDG?

Protein thermodynamic stability changes of single point mutation are changes of the Gibbes free energy for the biophysical process of protein folding between two states before and after single point mutation on the protein[1]. A quantified change of Gibbs free energy of a protein between the folding and unfolding status is usually represented as DG. When a point mutation is present and a residue is substituted in a protein, the original protein would be a “reference state”, likely called “wild-type protein”. The protein mutated is called “mutation protein”.

4.2 Installation

4.2.1 Clone the package

```
git clone https://github.com/hurraygong/MU3DSP.git
cd MU3DSP
```

4.2.2 Install dependencies

1. Install HHsuite

Using conda install hhsuite. It is also can be installed by other methods shown on official website, If you have installed, pass this step.

```
conda install -c conda-forge -c bioconda hhsuite
```

Install HHsuite database.

```
mkdir HHsuitDB
cd HHsuitDB
wget http://wwwuser.gwdg.de/~compbiol/uniclust/2020_06/UniRef30_2020_06_hhsuite.tar.gz
mkdir -p UniRef30_2020_06
tar xfz UniRef30_2020_06_hhsuite.tar.gz -C ./UniRef30_2020_06
rm UniRef30_2020_06_hhsuite.tar.gz
```

2. Install DSSP

Using conda install DSSP programe (https://anaconda.org/salilab/dssp). It is also can be installed by other methods shown on official website, If you have installed, pass this step.

```
conda install -c salilab dssp
```

Then to find the execute programe `mkdssp`. You can change the `mkdssp` to `dssp` using `cp` command by yourself.

```
whereis mkdssp
```

3.Install BLAST++

Install BLAST++ using `conda`. To install this package with conda run one of the following:

```
conda install -c bioconda blast
conda install -c bioconda/label/cf201901 blast
```

It is also can be installed by other methods shown on official website, If you have installed, pass this step.

Install BLAST++ database.

```
mkdir PsiblastDB/
cd PsiblastDB
wget https://ftp.ncbi.nlm.nih.gov/blast/db/swissprot.tar.gz
wget https://ftp.ncbi.nlm.nih.gov/blast/db/swissprot.tar.gz.md5
tar -zxvf swissprot.tar.gz
rm swissprot.tar.gz
rm swissprot.tar.gz.md5
```

Then you can proceed to the next step.

4.3 Datasets

4.3.1 Datasets for training, testing and casestudy

Our work used datasets S1676, S236, S543 to investigate the prediction of stability changes on protein (Table S1).

Table S1 Datasets used to build, evaluate and independently test in MU3DSP

Dataset	Total Variants (Proteins)	Destabilizing Variants (Proteins)	Stabilizing Variants (Proteins)	Stabilizing Variants (Proteins)	Additional Details
S1676	1676 (67)	1,223 (64)	424 (53)	29(4)	Unique Variants/Averaged DDG
S236	236 (22)	192 (18)	42 (14)	2(2)	Unique Variants/Averaged DDG
S543	543(55)	426(48)	107(37)	10(6)	Unique Variants/Averaged DDG
p53	42 (1)	31(1)	11 (1)	0	One Protein

4.3.2 Databases and references

Folkman, L., et al., EASE-MM: Sequence-Based Prediction of Mutation-Induced Stability Changes with Feature-Based Multiple Models. J Mol Biol, 2016. 428(6): p. 1394-1405.

Pires, D.E., D.B. Ascher, and T.L. Blundell, mCSM: predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics, 2014. 30(3): p. 335-42.

4.3.3 Datasets with features

Downloading

- S1676

4.4 Quickstart

4.4.1 Quikly Run MU3DSP

This is the parameters to run MU3DSP.

```
parser = argparse.ArgumentParser(description='Download PDBfiles from G2s & get features')
# parser.add_argument('-l', '--variant-site', dest='variant_list', type=str, help='A list of␣
↪variants, one per line in the format "POS WT MUT", a file')

parser.add_argument('-w', '--variant-wildtype', dest='variant_wildtype', type=str,
                    required=True, help='wild-type residue, for example residue "A"')
parser.add_argument('-m', '--variant-mutation', dest='variant_mutation', type=str,
                    required=True, help='mutation residue, for example residue "A"')
parser.add_argument('-p', '--variant-position', dest='variant_position', type=int,
```

(continues on next page)

(continued from previous page)

```
                    required=True, help='variant position in sequence')

parser.add_argument('-s', '--sequence',  type=str,
                    required=True, help='A protein primary sequence in a file in the␣
↪format fasta.')
parser.add_argument('--hhmfile', type=str,help='MSA files with hhm format from HHblits')
# background mutation Features from G2s

parser.add_argument('--pdbpath',  type=str,default='/storage/htc/joshilab/jghhd/SC/
↪stability_change1/datasets_s1676_seq/PDB/',
                    required=True, help='The path for mutation-based structures')



parser.add_argument('--dssppath', type=str,default='/storage/htc/joshilab/jghhd/SC/
↪stability_change1/datasets_s1676_seq/dssp/',
                    required=True, help='The path for DSSP output files')
parser.add_argument('--dsspbin', type=str, default='mkdssp',
                    required=True, help='DSSP binary executable.')
parser.add_argument('--psiblastbin', type=str, default='psiblast',
                    required=True, help='psiblast binary executable.')
parser.add_argument('--hhblitsbin', type=str, default='hhblits',help='Binary executable␣
↪for computing hhblits profile, "hhblits" for fasta input file and "hhmake" for A3M,')
parser.add_argument('--hhsuite', type=bool, default=False,help='HHM file')

parser.add_argument('--psiblastout',  type=str,default='./Sequence/psiout',
                    required=True, help='psiblast output files, a path')
parser.add_argument('--psiblastpssm',  type=str,default='./Sequence/pssmout',
                    required=True, help='A path for PSSM files')
parser.add_argument('--psiblastdb',  type=str,default='/home/gongjianting/tools/
↪PsiblastDB/swissprot',
                    required=True, help='background database for align in psiblast')


parser.add_argument('--hhblitsdb',  type=str,default='/home/gongjianting/tools/HHsuitDB/
↪UniRef30_2020_06', help='background database for align in tools hhblits')

parser.add_argument('--hhblitsout',  type=str,default='./Sequence/hhblitout',
                    required=True, help='A path for hhblits output files')
parser.add_argument('--hhblitshhm',  type=str,default='./Sequence/hhmout',
                    required=True, help='A path for storing HHM files')
parser.add_argument("-v", "--version", action="version")
parser.add_argument("-o", "--outfile", default='False', action='store_true',help=
↪'Whether save the result or not')
parser.add_argument("--printout",  default='False', action='store_true', help='Whether␣
↪print the result or not')
parser.add_argument("--outfilepath", type=str,default='./input.npy',help='Output file␣
↪path')
parser.add_argument("-G","--G2s",  default='False', action='store_true', help='Fast␣
↪Version, structures are unavailable in Q4')
```

Take mutation Q to H at position 104 of p53 protein as example. Its position in sequence is 9. So the run command as follows:

If provide MSA files with a3m format from HHblits. It can run with:

```
python MU3DSP.py -p residueposition -w wildtyperesidue -m mutationresidue -o outfilename␣
↪-s sequencepath --pdbpath pdbfilepath --dssppath dsspfilepath --dsspbin mkdssp-path --
↪psiblastbin  psiblast-path --hhblitsbin hhblits-path --psiblastout psioutfile-path --
↪psiblastpssm pssmoutfile-path --psiblastdb swissprot-path --hhblitshhm hhmoutfile-path␣
↪--seqa3m MSA-a3m-file
```

For example:

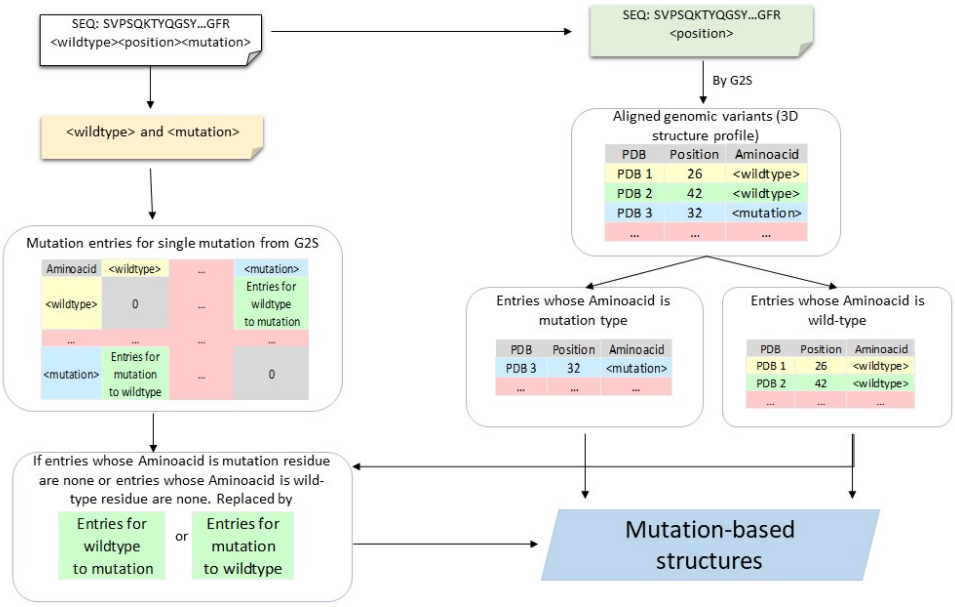
```
python MU3DSP.py -p 9 -w Q -m H -o True -s ./examples/SEQ/2ocj_A.fasta --pdbpath ./
↪examples/PDBtest --dssppath ./examples/DSSPtest --dsspbin dssp --psiblastbin  psiblast␣
↪--hhblitsbin hhmake --psiblastout ./examples/psiout --psiblastpssm ./examples/pssmout -
↪-psiblastdb ./PsiblastDB/swissprot --hhblitshhm ./examples/hhmout --outfilepath ./
↪examples/2ocj_Q9H.npy --seqa3m ./examples/a3m/p53.a3m --printout True
```

If provide fasta files and HHblits is accessable.

```
python MU3DSP.py -p residueposition -w wildtyperesidue -m mutationresidue -o outfilename␣
↪-s sequencepath --pdbpath pdbfilepath --dssppath dsspfilepath --dsspbin mkdssp-path --
↪psiblastbin  psiblast-path --hhblitsbin hhblits-path --psiblastout psioutfile-path --
↪psiblastpssm pssmoutfile-path --psiblastdb swissprot-path --hhblitsdb UniRef30_2020_06-
↪path --hhblitsout hhblitoutpath --hhblitshhm hhmoutfilepath
```

4.5 Mutation structure preparation

Our in-house [G2S](https://g2s.genomenexus.org) provides a real-time web Application Programming Interface (API) that automatically maps genomic variants on 3D protein structures. Giving a protein sequence and the position of a variant as the query, G2S searches similar sequence fragments (covering the surrounding regions of the mutation) in PDB to get a 3D structure profile from a list of protein structures with similar local sequences. G2S then chooses protein structures containing either wild-type amino acid or mutant amino acid at the aligned position of the mutant (queried residue; Supplementary Figure S2). According to the availability of PDB structures, four different strategies (Q1-Q4) may be adopted (Figure 2A). Q1: Tertiary structures of the wild-type residue and mutant residue are available. Q2: Tertiary structures of the wild-type residue are available, and tertiary structures of the mutant residue are unavailable. Q3: Tertiary structures of the wild-type residue are unavailable, and tertiary structures of the mutant residue are available. Q4: Neither tertiary structures of the wild-type residue nor tertiary structures of the mutant residue are available.



The samples count for 4 situations when mutation-based structures are available or not on datasets S1676, S543 and S236.

Dataset	Q1	Q2	Q3	Q4
S543	172	366	1	4
S236	46	159	0	31
S1676	706	950	0	20

4.6 Results Analysis

4.6.1 Performance on multiple models according to mutation-based structure information

To build a robust predicting protein stability changes model, dataset S1676 is used to train a model using 10-fold cross-validation. The results of our methods are replicated 10-fold cross-validation 20 times with shuffling the training data. Based on the availability of the PDB structure, we proposed four models MU3DSP^str, MU3DSP^seq, MU3DSP and MU3DSP^* to conduct comparative tests. When assuming that structures of wild-type and mutation proteins are unavailable for the training dataset, mutation-based structure features would get from G2S (MU3DSP*).

Table 1. Performance of 10-Fold cross-validation on Dataset S1676

Method	r_p	*RMSE*	*MAE*
EASE-AA	0.52	1.56	1.04
EASE-MM	0.56	1.52	1.00
SCpre-seq^{str}	0.53	1.58	1.09
SCpre-seq^{seq}	0.66	1.46	0.99
SCpre-seq	**0.72**	**1.33**	**0.88**
SCpre-seq*	0.70	1.36	0.90

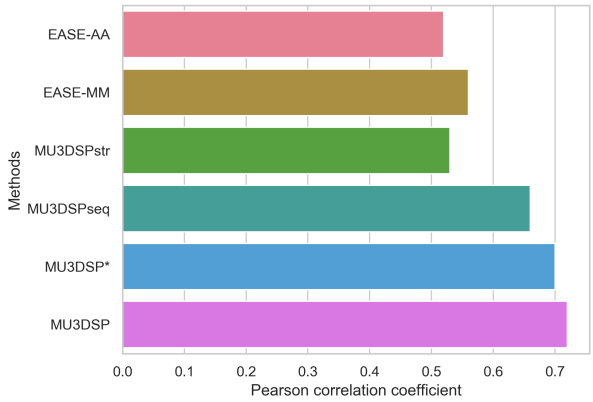


Figure 1. The Pearson correlation coefficient of 10-fold cross-validation for six models EASE-AA, EASE-MM, MU3DSP^str, MU3DSP^seq, MU3DSP and MU3DSP^* on the training dataset

4.6.2 MU3DSP achieves state-of-the-art performance on testing sets

To evaluate the robustness of our model, we used S236, S543 as the independent testing datasets. MU3DSP compared with nine commonly used methods including EASE-AA, EASE-MM, MUpro, I-Mutant2.0, INPS, SAAFEC-SEQ, DDGun and PoPMuSiC, MAESTRO on different testing datasets. Among them, EASE-AA, EASE-MM(EASE-MM-web), MUpro, sequence-based version of I-Mutant2.0(I-MutantA), INPS, SAAFEC-SEQ, DDGun predict G starting from sequence and mutation residues while the structure-based version of I-Mutant2.0(I-MutantB), PoPMuSiC, MAESTRO required structures as input. The results are shown in Figures 2 and 3 for S236 and S543, respectively.

Figure S10. Multiple bivariate plots for 11 comparative methods and MU3DSP with marginal histograms.

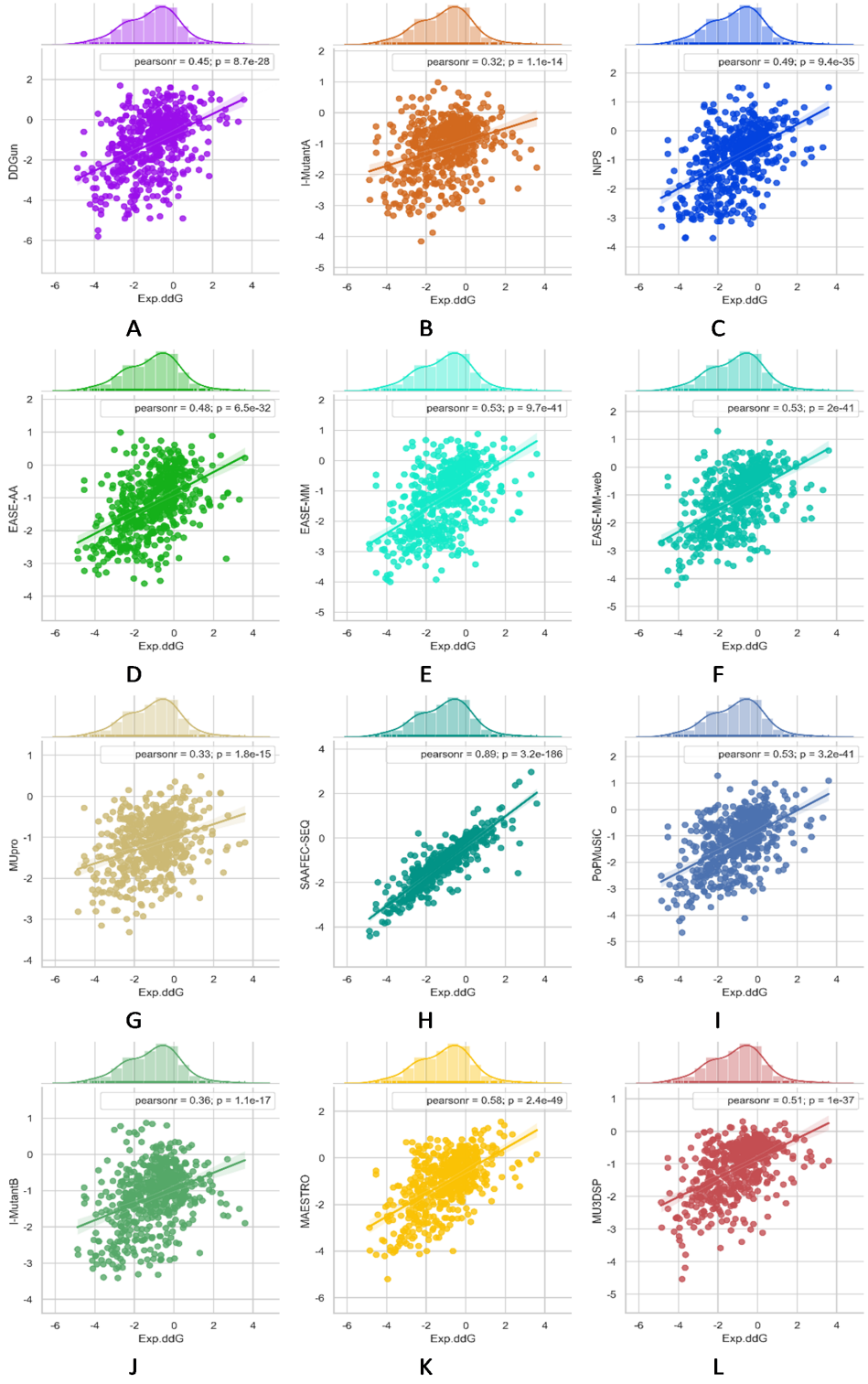


Figure 2. Multiple bivariate plots for 11 comparative methods and MU3DSP with marginal histograms on dataset S236.

Figure S9. Multiple bivariate plots for 11 comparative methods and MU3DSP with marginal histograms

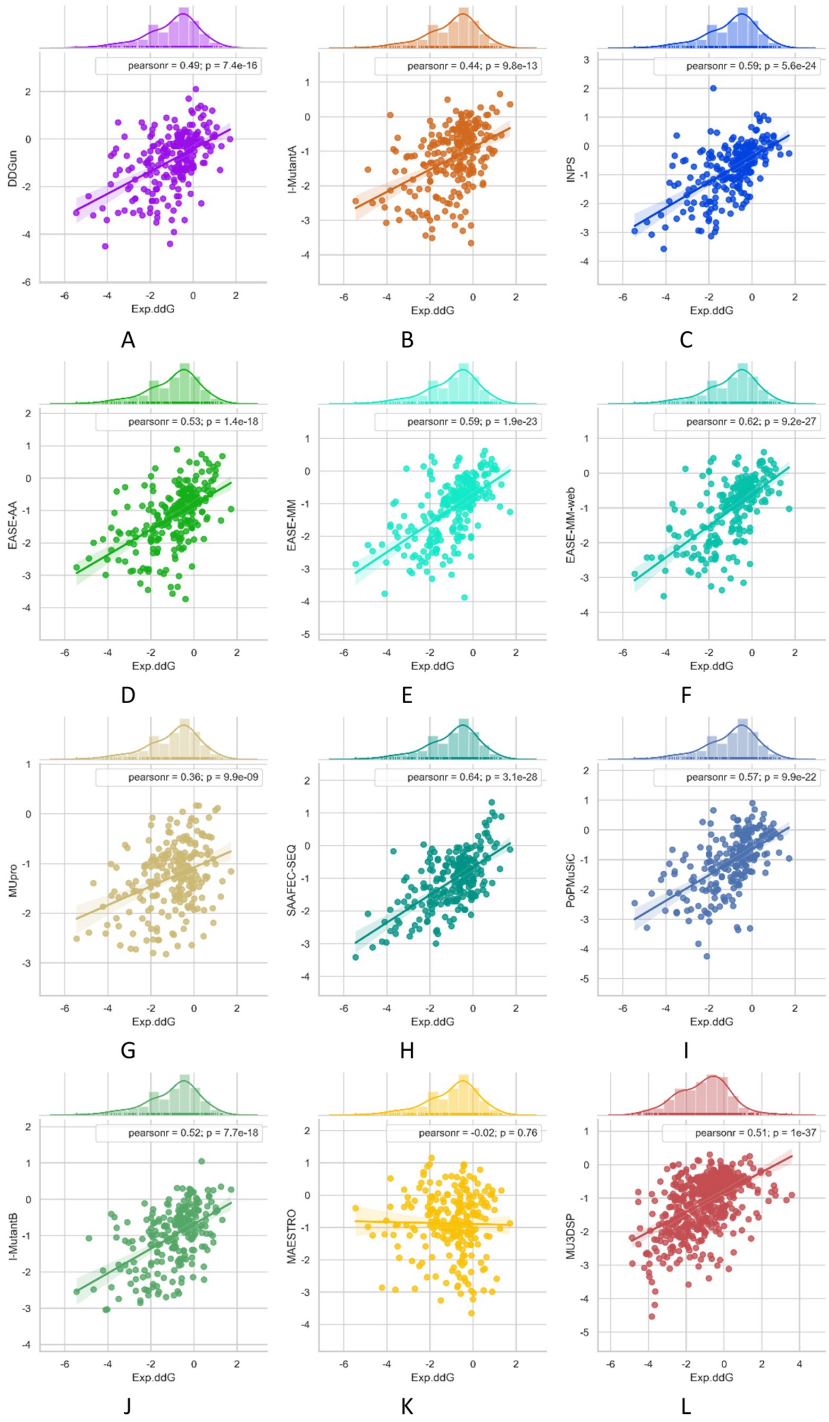


Figure 3. Multiple bivariate plots for 11 comparative methods and MU3DSP with marginal histograms on dataset S543.

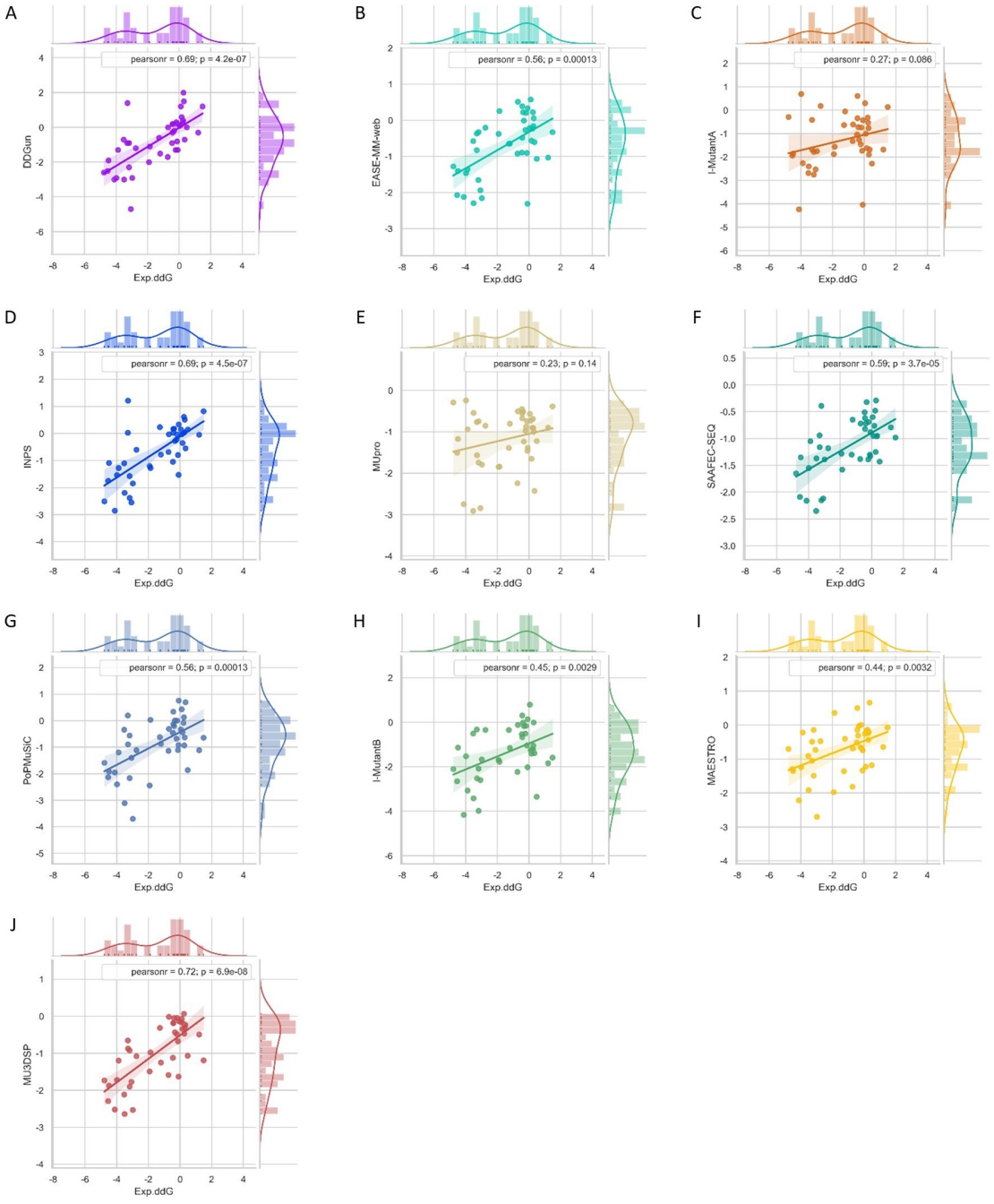
4.7 Casestudy

4.7.1 Predicting the impact of single-residue mutations on p53 thermodynamic stability

To further demonstrate the applicative power of MU3DSP, we applied it to a disease-related protein p53 containing 42 mutations. The mutations include 31 stabilizing mutations and 11 destabilizing mutations. p53 case datasets can be downloaded from p53.

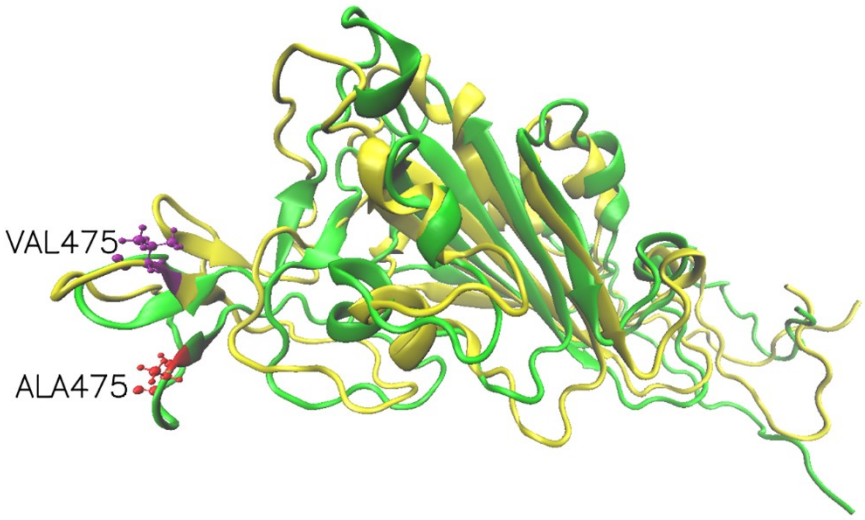
Multiple bivariate plots for nine comparative methods and MU3DSP with marginal histograms. DDG predicted with the three structure-based methods (G, H, I) and six sequence-based methods (A, B, C, D, E, F) including MU3DSP as function of experimentally measured stability changes (Exp.ddG) from the p53 dataset. The lines are the linear regression fits. Pearsonr represents Pearson correlation coefficient.

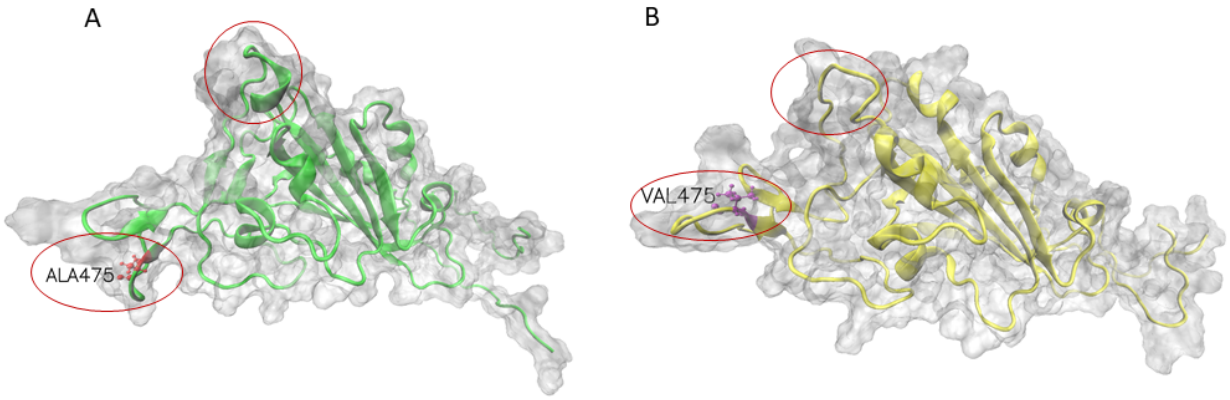
Figure 4. Multiple bivariate plots for nine comparative methods and MU3DSP with marginal histograms.



4.7.2 Mutation effects on for SARS-CoV-2 variants by stability changes perspective

The S proteins form homo-trimers on the virus surface and are necessary for the entrance of the virus into the host. Here, we only explored the effects of mutation on monomeric spike protein. The receptor-binding domain (RBD) is from 319 to 541 (UniProt ID: P0DTC2). Due to three PDB structures i.e. 6VXX (Closed state), 6VYB (Open state) and 6VSB (Prefusion) being discontinuous, we try to demonstrate wild-type and mutation RBD of spike protein 3D structures by AlphaFold2. The structures had visible changes after the A475V mutation and their structures align shown as follows.





4.8 Release

1.0

4.9 References

Please cite:

The codes are available at https://github.com/hurraygong/MU3DSP

4.10 Contact

Jianting Gong: gongjt057@nenu.edu.cn

Juexin Wang: wangjue@missouri.edu

Dong Xu: xudong@missouri.edu