T.C
DOKUZ EYLÜL UNIVERSITY
IZMIR INTERNATIONAL BIOMEDICINE AND GENOME
INSTITUTE

# PROT-ON: STRUCTURE-BASED DETECTION OF CRITICAL MUTATIONS IN REDESIGNING PROTEIN-PROTEIN INTERFACES

MEHDİ KOŞACA

DEPARTMENT OF MOLECULAR BIOLOGY AND GENETICS

**MASTER OF SCIENCE THESIS**

IZMIR – 2022

**THESIS CODE:** DEU.IBG.MSc-2019850040

T.C
DOKUZ EYLÜL UNIVERSITY
IZMIR INTERNATIONAL BIOMEDICINE AND GENOME
INSTITUTE

# PROT-ON: STRUCTURE-BASED DETECTION OF CRITICAL MUTATIONS IN REDESIGNING PROTEIN-PROTEIN INTERFACES

DEPARTMENT OF MOLECULAR BIOLOGY AND GENETICS

**MASTER OF SCIENCE THESIS**

MEHDİ KOŞACA

Supervising Faculty Member: Asst. Prof. Dr. Ezgi KARACA EREK

(This research is supported by the Health Institutes of Türkiye (TÜSEB) Project code: 2019-TA-02, Project number: 3393)

**THESIS CODE:** DEU.IBG.MSc-2019850040

Dokuz Eylül University Izmir International Biomedicine and Genome Institute Department of Genomics and Molecular Biotechnology, Molecular Biology and Genetics graduate program, Master of Science student Mehdi KOŞACA has successfully completed his Master of Science thesis titled **'PROT-ON: STRUCTURE BASED DETECTION OF CRITICAL MUTATIONS IN REDESIGNING THE PROTEIN-PROTEIN INTERFACES'** on the date of 29/08/2022.

CHAIR
Asst. Prof. Dr. Ezgi KARACA EREK
Dokuz Eylül University
Izmir International Biomedicine and Genome Institute

MEMBER
Asst. Prof. Dr. Saliha Ece ACUNER
Istanbul Medeniyet University

MEMBER
Asst. Prof. Dr. Yavuz OKTAY
Dokuz Eylül University

# TABLE OF CONTENTS

## INDEX OF TABLES

## INDEX OF FIGURES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| ΔG | Gibbs free energy |
| 3D | 3 Dimensional |
| Å | Angstrom |
| DACUM | The Database of binding affinity Change Upon Mutations |
| DBD | DNA Binding Domain |
| EvoEF | EvoDesign Physics-based Energy Function |
| FNIII | Fibronectin type III |
| GAS6 | Growth Arrest-Specific protein 6 |
| HADDOCK | High Ambiguity Driven biomolecular DOCKing |
| IQR | Interquartile Range |
| iSEE | interface structure, evolution, and energy |
| $K_d$ | Dissociation constant |
| kDa | kilodalton |
| $K_{off}$ | Rate constant of dissociation |
| $K_{on}$ | Rate constant of association |
| Ig | Immunoglobulin |
| MDM2 | Mouse Double Minute 2 |
| p53 | Tumor protein 53 |
| PCC | Pearson's Correlation Coefficient |
| PDB | Protein Data Bank |
| PK | Protein Kinase |
| PPI | Protein-Protein Interactions |
| PSSM | Position Specific Scoring Matrix |
| Q1 | First Quartile |
| Q3 | Third Quartile |
| RTK | Receptor Tyrosin Kinase |
| SKEMPI | Structural Database of Kinetics and Energetics of Mutant Protein Interactions |
| TAD | Transcriptional Activation Domain |
| TD | Tetramerization Domain |

**ACKNOWLEDGEMENT**

Firstly, I am extremely grateful to my supervisor, Dr. Ezgi Karaca, for her guidance in my scientific research, patience, and endless support. This endeavor would not have been possible without her great mentorship.

I would like to thank TÜSEB for supporting and funding this study under the 2019-TA-02 support program with grant number 3393. Additionally, I would like to thank EMBO to support me financially.

I would also like to thank IBG servers for enabling the PROT-ON web server deployment. Additionally, special thanks to Ömür Dalan and Tahir Oktay Vural for taking the time and helping me in the deployment of the PROT-ON web server on the IBG servers.

I would like to extend my sincere thanks to my labmates, Ayşe Berçin Barlas, Burcu Özden, Büşra Savaş, Eda Şamiloğlu, İrem Yılmazbilek and Melis Oktayoğlu for sharing a wonderful work environment and coffee breaks. I had the pleasure of working with you. "You are good people!"

I would be remiss in not mentioning, Rabia Kayahan, Murtaza Yiğit, Zeynep Köse, Abdülmecit Karakaş, Taşkın Kerem Koçak, Serkan Polat and Seray Yetkin for emotional supports, and their friendship. Life is beautiful with you.

Last but not least, I am very grateful to my father Cevdet Koşaca, my mother Nurcan Koşaca, my beloved sister Nergiz Koşaca, and my beloved brother Yusuf Koşaca for love under all circumstances. I am deeply indebted for your support and belief in me throughout my education life. I could not have undertaken this thesis without your support. This thesis is dedicated to my family.

# PROT-ON: STRUCTURE-BASED DETECTION OF CRITICAL MUTATIONS IN REDESIGNING PROTEIN-PROTEIN INTERFACES

Mehdi Koşaca, Izmir International Biomedicine and Genome Institute, Dokuz Eylul University Health Campus, Balcova, 35340, Izmir / Turkey

## ABSTRACT

Many biological activities in the body are mediated through protein-protein interactions (PPI). Understanding the PPIs is crucial for developing vaccines and drugs. Therefore, it is of utmost importance to understand how interfacial mutations change the nature of protein-protein interactions. To understand this phenomenon, several computational approaches such as FoldX and EvoEF have been developed. These tools estimate the change in the interaction energy by using different mathematical algorithms. In this thesis, by making use of these programs, we developed PROT-ON, which can scan all the interfacial mutations to explore the mutational landscape of a protein-protein interface. PROT-ON then classifies the mutations as most depleting or enriching, based on several metrics. We tested PROT-ON on MDM2-p53 and AXL-GAS6 complexes. For these cases, our approach was able to predict mutations that are critical to the vitality of the complex. The standalone version of PROT-ON can be cloned for Linux and macOS systems from GitHub (https://github.com/CSB-KaracaLab/prot-on) or can be accessed through a user-friendly web server (http://proton.tools.ibg.edu.tr:8001).

**Keywords:** binding affinity prediction, protein-protein interaction, redesigning interface, hot spot residue prediction, detection of critical mutations, interactive web server, classification of mutation

PROTEİN-PROTEİN ARAYÜZLERİNİ YENİ YAKLAŞIM PROT-ON ALGORİTMASI İLE YENİDEN TASARLANMASI

Mehdi Koşaca, İzmir Uluslararası Biyotıp ve Genom Enstitüsü, Dokuz Eyül Universitesi, Sağlık Kampüsü, Balçova, 35340, Izmir / Turkey

## ÖZET

Vücudumuzdaki çoğu biyokimyasal olay protein-protein etkileşimleri (PPE) aracılığıyla gerçekleşir. PPE'leri anlamak ilaç ve aşı geliştirme çalışmaları için son derece önemlidir. Bu yüzden arayüz mutasyonlarının protein-protein etkileşimlerinin doğasını nasıl değiştirdiğini bilmek gerekmektedir. EvoEF ve FoldX gibi çeşitli hesaplamalı yaklaşımlar farklı matematiksel algoritmalar kullanarak etkileşim enerjisindeki değişimleri tahmin etmektedirler. Bu tezde, PPE'deki tüm mutasyonları tarayıp etkileşim enerjilerini bu programlarla hesaplayacak PROT-ON yaklaşımını geliştirdik. PROT-ON, arayüz mutasyonlarındaki etkileşim enerjileri değişimlerini çeşitli metrikler kullanarak bağlanmayı en iyileştiren ve en kötüleştiren mutasyonlar olarak sınıflandırmaktadır. PROT-ON'un performansını MDM2-P53 ve AXL-GAS6 kompleksleri üzerinde test ettik ve kompleks için kritik mutasyonları geliştirdiğimiz yaklaşım ile başarılı bir şekilde tahmin edebildik. PROT-ON'un kendi başına çalışan sürümünü GitHub'dan (https://github.com/CSB-KaracaLab/prot-on) Linux ve MacOS sistemleri için indirebilir ya da kullanıcı dostu bir web portalı (http://proton.tools.ibg.edu.tr:8001) üzerinden erişilebilirsiniz.

**Anahtar Kelimeler:** etkileşim enerjisi tahmini, protein-protein etkileşimleri, arayüzün yeniden tasarlanması, kritik mutasyonların tespit edilmesi, interaktif web server, mutasyon sınıflandırılması

# 1. INTRODUCTION

## 1.1 Statement and Importance of the Problem

Understanding the impact of the interfacial mutations on the binding affinity is crucial for redesigning protein-protein interfaces. There are many computational binding affinity predictors for estimating the impact of mutations. These tools use several metrics like evolutionary conservation and/or force field-related terms. The most accurate and widely used interfacial mutation predictors are FoldX and EvoEF1. These tools are often used to predict the impact of single-point mutations across a protein interface. However, the field still lacks a user-friendly tool that scans all possible interfacial mutations with the aim of finding designer mutations with potential therapeutic applications.

## 1.2 Aim of the Study

Expanding on the above-outlined problem, in this work, we developed PROT-ON, which uses EvoEF and FoldX to calculate all possible interfacial mutations of protein-protein interactions, followed by their statistical analysis to propose the most enriching and depleting mutations. These mutations can further be used to devise new protein-protein interactions.

## 1.3 The Hypothesis of the Study

**H1:** The computational mutational scanning of a protein-protein interface followed by their statistical analysis will provide experimentally testable mutation positions.

**H2:** A user-friendly web server that scans all the available protein interface mutations could aid the studies aiming to redesign an existing interface for the sake of developing drugs and vaccines.

## 2. GENERAL INFORMATION

### 2.1 Protein-Protein Interactions

Proteins are 3D-shaped macromolecules that play a crucial role in many cellular processes. Proteins mediate their functions by physical interactions with other molecules i.e., nucleic acids, proteins, and carbohydrates (Consortium, 2010). The interaction between the trypsin enzyme and its inhibitor was the first discovered protein-protein interaction (PPI) in the early 1900s (Northrop, 1922). Since then, many protein interactions were profiled while looking for the roots of infectious diseases. (Braun and Gingras, 2012). In recent years, the analysis of PPIs clarified the mechanism of a wide range of genetic diseases. Also, many therapeutic agents like drugs were developed to combat these genetic diseases by focusing on protein-protein interfaces (Moreira et al., 2007).

PPIs are mediated by the association and dissociation of interfacial amino acids over protein surfaces (Moreira et al., 2007). The interacting protein surfaces have geometric and chemical complementarity with each other (Fernández and Scheraga, 2003), as exemplified in Figure 1. Non-bonded interactions i.e., electrostatic, van der Waals and hydrogen bonding, play a major role in the association and dissociation of interfacial amino acids (Stites, 1997). In earlier studies, the interface size was calculated to range from 1150Å to 4660Å, depending on the type of protein and its function. (Conte et al., 1999). In 2002, Chakrabarti and Janin (Chakrabarti and Janin, 2002) suggested dissecting the PPIs in two regions, the core, and rim.

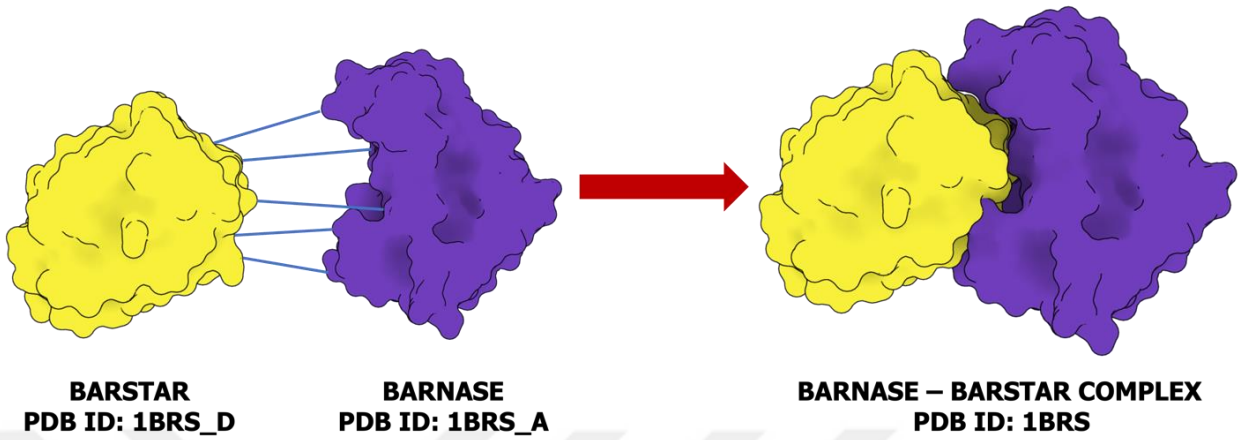



**Figure 1.** The schematic complex formation between barnase and barstar proteins.

### 2.1.1 Binding affinity calculation of PPIs

Binding affinity ($K_d$) measures the ratio between the dissociation ($k_{off}$) and association ($k_{on}$) rate constants under equilibrium. $K_d$ is a measure of binding strength in molar (M) units.

$$K_d = k_{off} / k_{on}$$

**Equation 1.** Equilibrium dissociation of a complex.

The binding affinity is related to Gibbs free energy through Equation 2, where "R" refers to the gas constant and its value is 8.3144 $J.K^{-1}.mol^{-1}$; "T" is the absolute temperature in the Kelvin metric; ΔH and ΔS refer to the change in enthalpy and entropy. For a spontaneous and thermodynamically favorable binding ΔG should be less than zero (Kastritis and Bonvin, 2013).

$$\Delta G = -RT\ln(K_d) = \Delta H - T\Delta S$$

**Equation 2.** The formula of Gibbs free energy.

### 2.1.2 Redesigning PPIs

There are different ways to develop protein-based drugs/vaccines. One of them is to engineer new proteins based on existing protein-protein interactions (Roy et al., 2017). Rational design and directed evolution are protein engineering methods used to protein design or redesign interfaces (Moreira et al., 2007).

Rational design requires prior information such as 3D protein structures, biophysical features, and functionality of the protein (Marshall et al., 2003). Based on the available structure, interfacial amino acids are mutated to their other possibilities. Rational design is used to design stable and efficient proteins or to alter the physical properties of proteins. Controlled mutations on the interfaces affect the protein function and indirectly its reactions. (Acuner et al., 2021). Directed evolution mimics natural selection (Lutz and Iamurri, 2018). Gene recombination and random mutagenesis are the most commonly used directed evolution techniques. In this technique, random mutations are imposed on the genome. Then, mutant gene libraries are generated, and each gene is amplified by recombination in host cells. All cells that include recombinant genes are screened with different experimental methods. and the desired recombinant is selected and amplificated (Acuner et al., 2021).

The number of possible mutations is too large to test experimentally. Therefore, many computational approaches have been developed to analyze the impact of mutations in a fast and efficient manner.

## 2.2 Binding affinity predictors

Computational tools were started to be developed to estimate the binding affinity at the beginning of the 20$^{th}$ century. They are used as an alternative to experimental techniques owing to experiments being time-consuming and challenging. Binding affinity predictors are divided into three types depending on their concepts physics -based,

statistical potential-based, and machine learning-based methods (Siebenmorgen and Zacharias, 2020).

Chemical and architectural knowledge of proteins are deposited into protein data bank files (PDB). Non-bonded interactions like electrostatic and van der Waals can be inferred through chemical knowledge. (Nicholls et al., 1991). Additionally, binding affinities of PPIs can be computationally predicted by determining the chemical environment of interfacial amino acids. Nonpolar and polar forces at the interface contribute favorably to the binding affinity. In order to determine forces between atoms and molecules, force fields like CHARMM (Vanommeslaeghe et al., 2010) and AMBER (Wang et al., 2004) have been developed (Halgren and Damm, 2001). They use bonded and non-bonded terms in order to estimate molecular forces. Bonded terms define atom-atom interactions that are covalently connected, whereas non-bonded terms represent noncovalent interactions like electrostatic interactions, hydrogen bonds, and van der Waals forces. Electrostatic interactions occur between charged groups of amino acid side chains. They are the most potent non-bonded interaction. Van der Waals interactions occur when two atoms of interfacial residues come close to each other. They are the weakest interactions Hydrogen bonds are strong forces that are linked between proton donors and proton acceptors. They are mediated through electronegative atoms (F, O, and N) (Bartell, 1968). An illustration of these forces is given in Figure 2.

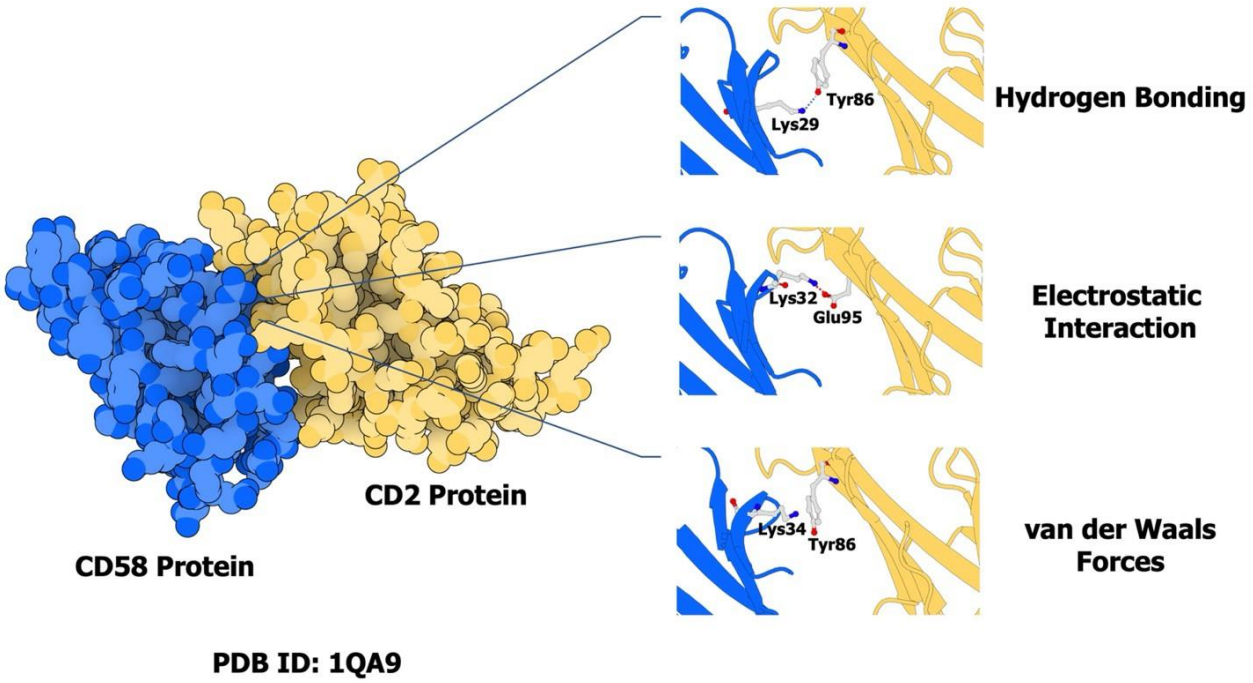



**Figure 2**. Illustration of non-bonded forces on 1QA9 structure.

The FoldX (https://foldxsuite.crg.eu/) (Schymkowitz et al., 2005) is one of the earliest physics -based approaches to estimate the Gibbs free energy for binding. It has a linear empirical formula consisting of a combination of non-bonded interactions. ΔΔG interaction energy is calculated by the differences between the mutant ΔG and wild-type ΔG score. It is formulated in Equation 3.

$$\Delta\Delta G = \Delta G_{Mutant} - \Delta G_{Wild\text{-}type}$$

**Equation 3.** Calculating the interaction energy by computational binding affinity predictors.

EvoEF (https://zhanggroup.org/EvoEF/) (Pearce et al., 2019) is another widely used physics based approach. It is used to estimate the interaction energies of proteins. It was developed particularly for redesigning the interface. EvoEF has also an empirical force field. It considers the combination of non-bonded interactions to estimate the binding affinity.

Statistical potential-based methods are used to estimate the Gibbs free energy. They are developed by analyzing large experimental binding affinity database such as SKEMPI v2.0 (Geng et al., 2019b). SKEMPI includes 7085 mutations and their measured experimental binding affinities. This dataset has information about the values of dissociation of constant, association rate, dissociation rate, and affinities for wild-type and mutant. It also includes properties of the binding site of mutation, PDB ID, and the experimental method that measured the binding affinity (Jankauskaitė et al., 2019).

BeAtMuSic (http://babylone.ulb.ac.be/beatmusic) (Dehouck et al., 2013) is a coarse-grained technique to estimate the ΔΔG interaction energies of a mutation by using the statistical potential-based method. It has yielded 0.67 Pearson's correlation coefficient (PCC) on 2648 single-point mutations, and 0.79 PCC on 90% of the SKEMPI v1.1 dataset.

Machine learning-based methods are also used in estimating Gibbs free energy. They are so dependent on the experimental binding affinity datasets. In the last few years, many machine learning techniques have been developed to predict the interaction energies for a PPI. Protein sequence/structure and evolutionary information are used as well as the binding potential in machine learning-based methods. They achieve high performance in estimating the interaction energy due to their large experimental data sets and their concepts to predict the structural and physicochemical changes that impact the mutation. the most common machine learning techniques are decision trees and Random Forest (RF). Especially RF techniques are widely used by ΔΔG estimating algorithms (Geng et al., 2019b).

Interface structure, evolution, and energy-based (iSEE) algorithm is one of the RF used machine-learning methods (Geng et al., 2019a). iSEE estimates the Gibbs free energy by using position-specific scoring matrix (PSSM) scores and the HADDOCK (Van Zundert et al., 2016) energy terms as features for both wild-type and mutant structures. It is developed by training the DACUM dataset (Geng et al., 2016) by using 31 features.

This dataset is a subset of SKEMPI and consists of 1102 mutations from 57 different protein complexes. iSEE achieves 0.80 PCC on the DACUM dataset.

PSSM scores are the most contributed iSEE feature to the predicted binding affinity. It is obtained by multiple sequence alignment data. PSSM scores define observing amino acid frequency in the evolutionary time for the current position. If a score is less than zero, the amino acid is less frequently present for that position, whereas, if the score is bigger, the situation is vice versa.

UEP is a fast classifier of interfacial mutations. It is a contact-based technique that classifies the mutations as enhancing and detrimental based on the three-body contacts. The interactome 3D (Mosca et al., 2013) dataset was used to analyze contact-based interactions. Two main criteria are used in analyzing the interactions. The first one is that an amino acid in a partner must be in contact with at least two amino acids of the other partner. Secondly, interactions must be within a 5.0Å cut-off distance. Interactions that do not conform to these criteria are missed. UEP is also inability building mutation models and makes any statistical analysis.

Expanding of the available tools, we have developed PROT-ON to classify the interfacial amino acids as enriching and depleting for PPIs. PROT-ON uses EvoEF1 or FoldX algorithms to estimate the binding free energy changes of a mutation. It can scan all the interfacial mutations to explore the mutational landscape of a protein-protein interface. Then classifies the mutations as most depleting or enriching, based on several metrics. PROT-ON codes are deposited on GitHub (https://github.com/CSB-KaracaLab/prot-on,) for Linux or Mac computers. We also offer a user-friendly web portal (http://proton.tools.ibg.edu.tr:8001/) without any registration.

# 3. MATERIALS AND METHODS

## 3.1 Type of the Study

A method development study that uses different services, websites, and Python packages in this thesis.

## 3.2 Time and Location of the Study

This study was conducted at Izmir Biomedicine and Genome Center and Izmir International Biomedicine and Genome Institute between March 2021 and August 2022.

## 3.3 Population and Sample of the Study

In this thesis, we considered the interfacial mutations of PPIs and their ΔΔG energy scores estimated by EvoEF1 or FoldX.

## 3.4 Materials of Study

In this thesis, we accept a PDB coordinate file that contains at least two distinct monomers as an input. We also used Python3 and its packages to develop the tool. All codes were pushed to GitHub as an open source. The web server was developed by Flask and Bootstrap. It was deployed from an HP Z2 Tower G4 workstation. It works with Linux 18.04.6 LTS operating system and has a 64-bit 3.40 GHz Intel Xeon E-2124 G CPU with 4 cores and 32 GB ram.

## 3.5 List of the Variables & Explanations

- Interface selection cut-off is a variable used to define interfacial amino acids of PPI.
- Mutation selection cut-off is a variable used to adjust the whisker length of the box-whisker in proposing designer mutations.
- Binding affinity predictors is a variable used to estimate binding affinities of interfacial mutations.

### 3.6 Utilities Used in PROT-ON

We have used the following packages, services, and websites in developing the tool and analyzing the test cases.

**PYTHON:** Python (https://www.python.org/) is object-oriented, easy to read, free to use, and open-source programming language. It is so preferred due to its simple syntax rule and extensive usage from web applications to data science. All codes of PROT-ON were written with Python3.

**PANDAS:** Pandas (https://pandas.pydata.org/) is one of the basis Python packages in data science. Dataframe and series can be manipulated and analyzed by Pandas. It was used to read the created data or output files of PROT-ON.

**NUMPY:** NumPy (https://numpy.org/) is a math python library that allows us to perform scientific calculations fast. It is based on the NumPy arrays. It was used to create lists or arrays, get the unique values in a list and calculate the quantile to draw a box plot.

**GITHUB:** GitHub (https://github.com/) is a cloud-based service that hosts the version control system (VCS) called Git. It serves developers to collaborate and follow up on the project's progress. It was used to posit the PROT-ON codes as an open-source for developers and users.

**VIRTUALENV:** Virtualenv (https://virtualenv.pypa.io/en/latest/#) is an isolated Python virtual environment that allows independently using different versions of the same libraries. It was used to avoid encountering possible errors after updating the package for the web server.

**FLASK:** Flask (https://flask.palletsprojects.com/en/2.1.x/) is a microframework written with Python to develop web applications. It uses the jinja2 template to render the HTML pages. Flask was used to build the backend of the web server.

**BOOTSTRAP:** Bootstrap (https://getbootstrap.com/) is an open-source and free web application development toolkit composed of utilizable code snippets. It was used to develop the front-end (i.e., navigator menu and grid system) of the web server.

**CELERY:** Celery (https://docs.celeryq.dev/en/stable/getting-started/introduction.html) is an open-source Python package that allows running tasks in the background asynchronously. It was used to prevent data accumulation in the workstation by running the scheduled task and to run the PROT-ON scripts with given users' inputs.

**RABBITMQ:** RabbitMQ (https://www.rabbitmq.com/) is a message broker that offers the transfer of any message or task from the first source to the target. It communicates with Celery background tasks whenever any error occurs or needs to check the job status.

**SQLALCHEMY:** Sqlalchemy (https://www.sqlalchemy.org/) is an ORM library developed for python. It was used for backend URLs and collaborates with Celery and RabbitMQ to store results in the database.

**PLOTLY:** Plotly (https://plotly.com/) is a library for R, Python, and JavaScript that consists of several functions to create a dashboard or data visualization. In contrast to classic plots, interactive graphics can be drawn with Plotly. It was used to create interactive heatmap and box plots for standalone and web server versions of the PROT-ON.

**NGINX:** Nginx (https://www.nginx.com/) is open-source web-service software that serves as a reverse proxy, load balancing for HTTP, and email proxy server for IMAP, POP3, and SMTP. It was used to deploy the PROT-ON web server.

**GUNICORN:** Gunicorn (https://gunicorn.org/) is a WSGI HTTP server written with Python. It was used to allow communication of the Flask PROT-ON project with Nginx by converting HTTP requests to Python.

**RCSB:** RCSB (https://www.rcsb.org/) is a biological structural bank for proteins and nucleic acids determined by experimental techniques like X-ray diffraction, NMR, and Cryo-EM. It was used to fetch the MDM2-p53 and AXL-GAS6 complex, which are test cases of PROT-ON.

**3DPROTEINIMAGING:** 3D Protein Imaging (https://3dproteinimaging.com/) is a free biological molecule visualization platform. It was used to export high-quality, unique figures for the test cases.

## 3.7 Methodology

### 3.7.1 Tools to sample & score interfacial mutations

#### 3.7.1.1 EvoEF1

EvoEF1 is a tool that is developed to predict the binding affinity changes by combining the evolutionary profile and physics-based potential. It is widely used in protein design studies and in identifying hot spot residues for PPI. FoldX was used to calculate physical energy changes in the previous version of EvoEF (EvoDesign). The EvoDesign team developed a new physical energy function due to FoldX being time-consuming to call from an external source. The new function applies the fundamental physical laws composed of five energy terms to describe the atomic interactions. Combining these terms is formulized in Equation 4.

$$E_{EvoEF} = \Sigma[E_{vdw}(i,j)+E_{elec}(i,j)+E_{HB}(i,j)+E_{solv}(i,j)]-E_{ref}$$

**Equation 4.** The empirical formula of the EvoEF1 algorithm.

EvoEF1 has several valuable functions to perform energy minimization, build mutation models and compute the stability & binding affinities. The "RepairStructure" command repairs the structure by removing clashes and minimizing the structure energetically. EvoEF team strongly recommends using it before building mutation or computing energy scores. The "BuildMutant" function is used to generate mutation models

in the mutant file with the mutation format (mentioned 3.6.2.1). The "ComputeStability" and "ComputeBinding" commands predict the stability and binding affinity scores, respectively.

#### 3.7.1.2 FoldX

FoldX is one of the earliest empirical effective energy functions. It was developed to predict the free energy changes of a mutation on the stability of a monomer or on the binding affinity for a complex. Estimating free energy is carried out with non-bonded interactions (i.e., van der Waals and electrostatic, hydrogen bonding), solvation energy, etc. In addition to these, it also considers the entropy by statistically analyzing the phi-psi distribution for each amino acid in the structure to calculate the free energy. The empirical formula of FoldX is given in Equation 5.

$$\Delta G = a.\Delta G_{vdw} + b.\Delta G_{solvH} + c.\Delta G_{solvP} + d.\Delta G_{wb} + e.\Delta G_{hbond} + f.\Delta G_{el} + g.\Delta G_{kon} + h.T\,\Delta S_{mc} + k.T\Delta S_{sc} + l.\Delta G_{clash}$$

**Equation 5.** The empirical formula of the FoldX algorithm.

The estimation of binding free energies for a protein-protein interaction is calculated with the formula that is given in Equation 5. Suppose that there is a complex with A and B proteins. FoldX computes the Gibbs energy of AB as a complex and then, subtracted it from the sum of Gibbs energies of A and B monomers. The empirical formula for the FoldX interaction energy can be found in Equation 6.

$$\Delta G_{binding} = \Delta G_{AB} - (\Delta G_A + \Delta G_B)$$

**Equation 6.** The formula to calculate the binding affinity in the FoldX.

FoldX has different versions and new features were added at each update. FoldX 4.0 provides 21 functions for protein-protein, protein-nucleic acid, or monomers. The "RepairPDB" function is a repair script for proteins. It repairs the residues that have van

der Waals' clashes and energetically less favorable torsion angles and gives a repaired structure as output. The "BuildModel" function is responsible to build mutation models. It creates a mutation model together with its wild-type reference as output. All mutations have to be written into the mutant file with the mutation format (mentioned in 3.6.2.1). The "AnalyseComplex" and "Stability" complex estimate the interaction energy and Gibbs free energy of a PPI or monomer respectively.

#### 3.7.1.3 Methods to analyze a set of mutation score

Box plot is a visualizing method to analyze the distribution of numerical data by is summarized it with five numbers the minimum score, first quartile, median, third quartile, and maximum score. Definitions of these five numbers are given below.

- **Minimum score:** The lowest whisker end.
- **First Quartile (Q1):** It is a value below which 25% of the data points are found when they are ranked ascending.
- **Median:** Marks the midpoint of the data and is indicated by the line dividing the box into two parts.
- **Third Quartile (Q3):** It is a value below which 75% of the data points are found when they are ranked ascending. Quartiles and median are calculated by the quantile function of the NumPy Python package.
- **Maximum score:** The highest whisker end.
- **Outliers:** Data points that are significantly different from the main population. If a data point is bigger than the third quartile or smaller than the first quartile, it is called an outlier. The distance of outliers is generally calculated by 1.5 times the interquartile range (IQR, which is calculated as Q3 – Q1) as a default, but it can be changed by the user from 1.5 to 2.5 in the PROT-ON server (Q1 – 1.5xIQR or Q3 + 1.5xIQR). The displacement of these numbers on a box plot is represented in Figure 3.

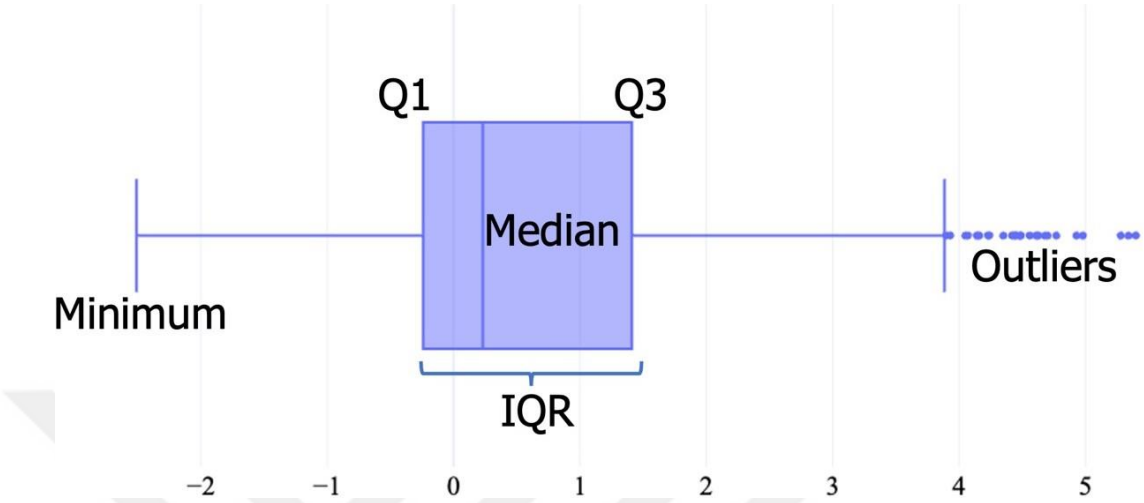



**Figure 3.** Five numbers of a box plot and displacement on the graph.

3.7.1.4 Metrics to analyze the probability of a mutation to occur

The most enriching and depleting mutations are filtered by ΔΔG stability scores and differences in PSSM scores. All ΔΔG is calculated by differencing the scores between mutant and wild-type by using the formula in Equation 3. Stability scores are predicted by the "ComputeStability" and "Stability" functions of EvoEF1 and FoldX respectively. PSSM scores are obtained from the uploading CSV file. Stability scores were calculated for each mutation. It is crucial to estimate the impact of the mutation on monomer stability. Especially some enriching mutations may destabilize the monomer despite having an enhancing effect on binding. Filtering these mutations help propose the most related and most accurate mutations for future search. On the other hand, PSSM scores give sequence information about the wild-type residue and its all-possible mutations on the evolutionary profile. Filtered mutations by these metrics may prove them both at structural and sequence levels.

### 3.7.2 Tools used in developing & deploying of PROT-ON as a web server

#### 3.7.2.1 Tools used in the back-end of the PROT-ON web server

Celery and RabbitMQ are used to run the jobs asynchronously in the background. Firstly, they were installed on the workstation that deployed the Flask web application. After installation of the RabbitMQ server, it was activated with a "detached" parameter. A new user was created, and a password is set. Finally, user permissions (configuration, writing, and reading) were given to the created user. Used commands to carry out these processes are found in Appendix A.

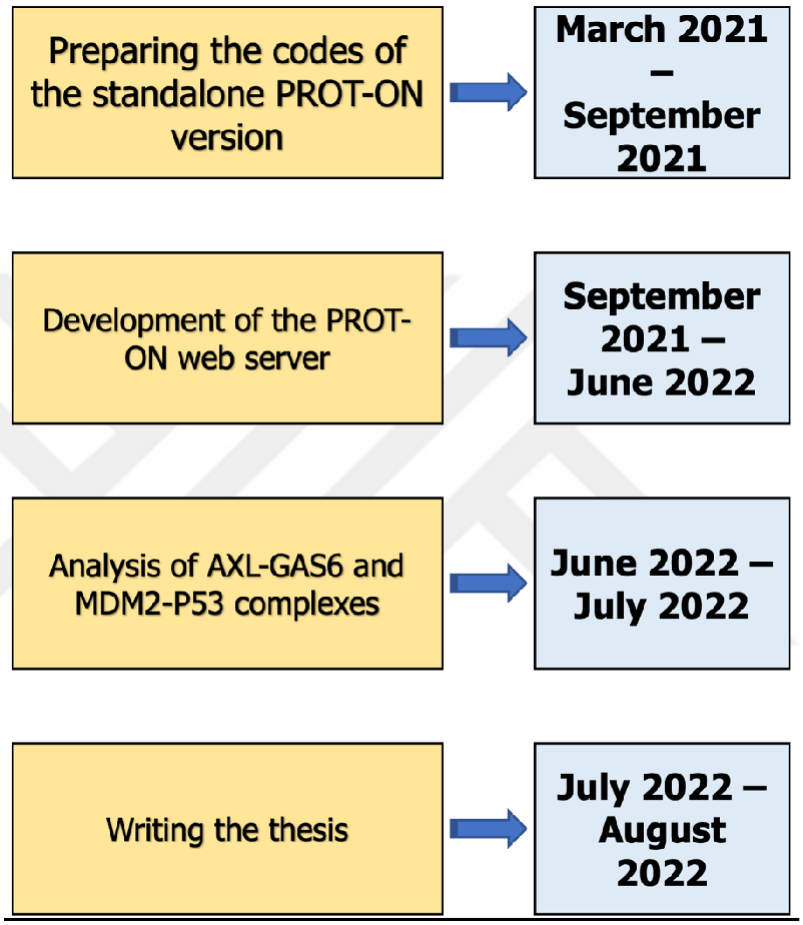
Celery is worked by communicating with backend and broker URLs to store the result or to transport the message. In this manner, RabbitMQ was used as a message broker and SQLAlchemy stored the results in the database. URLs are saved in an environment file (.env) with the created user information and database name. The environment file was called from the main python script (app.py) to update the configuration settings of Celery. The environment file was given in Appendix B.

#### 3.7.2.2 Tools used in the deployment of the PROT-ON web server

Deployment of PROT-ON in a workstation were performed by Gunicorn, Nginx, and Supervisor. All services and packages were installed on the Python virtual environment. The created Flask application can be run on the WSGI HTTP server safely. Gunicorn was used to provide safety for the application.

Gunicorn does not be used alone. It also requires communicating with another algorithm to receive and transmit the requests. Nginx is the best choice to do that. It was developed firstly as a mail SMTP server. Then it was used as a reverse proxy for many websites around the world. It was selected in PROT-ON for its easy usability and speed. It can be installed on Linux machines with the "apt-install" package.

## 3.7 Study Plan and Calendar




## 3.8 Data Evaluation

We have solved the problems that may arise as follows.

- FoldX and EvoEF1 are used to estimate the binding free energies accurately due to they are most accurate, fastest, and most correlated with the SKEMPI dataset to estimate the ΔΔG energies correctly.
- The cut-off distance and IQR range are set to user-dependent parameters to examine suggested mutations in different parameters. Thus, the most related mutations may be selected.

- All side chain atoms are considered when interfacial amino acids are defined with three distance formula. If only alpha carbon atoms are used, some interactions would be missed.
- A periodic task that removes the files in the web server was prepared to prevent data accumulation on the workstation.

## 3.9 Limitations of Study

PROT-ON was developed only to redesign the protein-protein interface. It is not compatible working with interfaces including small molecules and nucleic acids.

## 3.10 Ethics Committee Approval

This study was approved by the ethics committee approval with the 2021-007 protocol number.

# 4 RESULTS

## 4.1 The PROT-ON Pipeline

We developed an automatic pipeline to classify the interfacial mutations of PPIs. Our pipeline firstly defines interfacial residues and mutates each of them to other 19 amino acids. Then, FoldX or EvoEF1 build mutation models and estimate the binding affinities of each mutation. Finally, statistical analysis is made on the estimated binding affinity differences. Heatmap and box-whiskers plots are generated, and the most enriching and depleting mutations are proposed based on the mutational landscapes. This pipeline is illustrated in Figure 4 with a workflow.

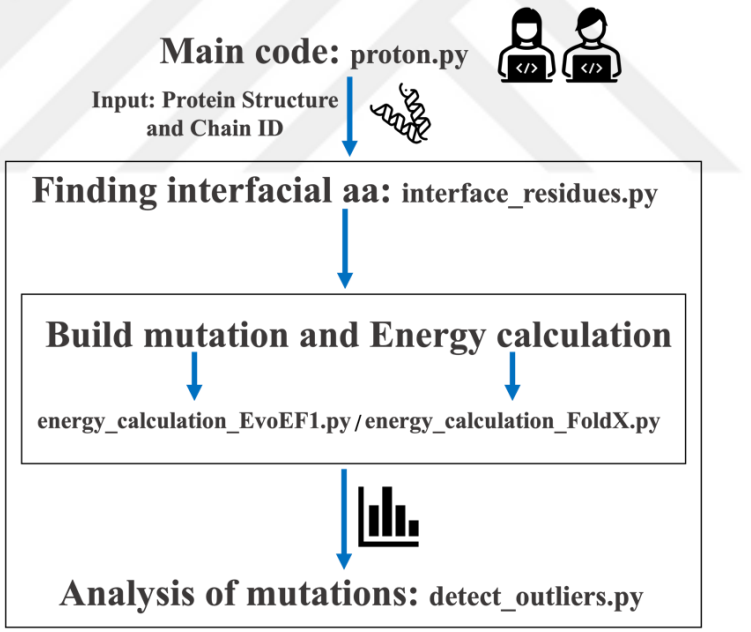



**Figure 4.** The code architecture of the standalone version of PROT-ON.

An initiator script (proton.py) was prepared to call all individual scripts. A job can be run using the "proton.py" script with several parameters. Its parameters are a structure file, a chain of interest, a cut-off distance, and an IQR range respectively. Example

command to run an example structure file is like that: "**python proton.py example.pdb A 5.0 1.5**" Note that it should be run with Python3.

In order to find the interfacial amino acids in the user's interesting cut-off, the "interface_residues.py" script is prepared. Firstly, the PDB file is read, and lines starting with "ATOM" are selected. Then atom number, atom id, residue ID, residue number, and x, y, and z coordinates of interacting chains are stored in lists. All these processes are performed with the "PDBParse" function. Before calculating the atom distances, x, y, and z coordinates are converted to floats from the string in the "ChainCoordinates" function. Distances between two atoms are calculated with the three-dimensional distance formula that is given in Equation 7.

A (x1, y1, z1) and B (x2, y2, z2)

$$|AB| = \&\overline{(x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2}$$

**Equation 7.** Three-point distance formula to calculate the atomic distance in the interface.

After applying the formula to all atoms in the interface, while pairwise atom information with the distance is written in a "pairwise_distance_list" file, interfacial residues are saved into the "interface_aa_list" file with the "FindDistances" function. All residues in the "interface_aa_list" file are mutated to other 19 amino acids and saved into the "mutation_list" file (i.e., KD22A-> **K**: Wild-type aa, **D**: Chain ID, **22**: aa position, **A**: Mutate aa), also "heatmap_mutation_list" (i.e., K22A) file is created. All these are performed in the "MutationFile" function.

An algorithm selection box is displayed to the user in the next step. After the algorithm selection, all mutations in the mutation file are built and their binding free energies are estimated by "energy_calculation_EvoEF1" or "energy_calculation_FoldX" scripts.

In the final step, the detect_outliers.py script gets the statistical analysis of all estimated ΔΔG interaction energies. Some plots are generated, and the most detrimental and enhancing mutations are proposed. PROT-ON creates three different box plots of the whole data, position-specific and mutation-specific distributions. The whole data box plot is useful for viewing how data is distributed and detecting the outliers. The most enriching and depleting mutations are identified by the whole-data box plot Codes for a drawing of the whole data box plot by the IQR range can be found in Appendix H. Position-specific and mutation-specific box plots illustrate the distribution of energy scores of each interfacial position or 20 amino acids with the default IQR range. On the other hand, a heatmap is created that displays the energy scores of all interfacial amino acids and their possible mutations except outliers. The reason for excluding outliers is that balance the heatmap colors from exceeding outlier points.

There are two main criteria to filter mutation for both. PSSM scores must be equal to or bigger than zero for enriching mutations and be equal to or less than zero for depleting mutations. On the other hand, stability scores must be bigger than zero for enriching mutations and less than zero for depleting mutations. Filtered mutations by these metrics can be found on the output folder with "stability filter" and "pssm filtering" files.

All codes were pushed to a GitHub repository. Also, an example run folder was added to the repository. The standalone version of GitHub is under the guard of an Apache 2.0 license. Detailed usage of PROT-ON was informed on the "readme" page. The standalone version of PROT-ON can be cloned from https://github.com/CSB-KaracaLab/prot-on for Linux and macOS systems.

Our automatic pipeline was tested on well-known MDM2-p53 interaction to detect the critical p53 amino acids for proofing the PROT-ON concept.

## 4.2 Proof of Concept Study 1: MDM2-p53

### 4.2.1 p53 protein

Accumulating mutations in DNA cause cancer in the cell. Cancer is uncontrolled cell division, and the organism may die if it is not treated. Some protective mechanisms evolved to inhibit uncontrolled cell division. The genes that regulate cell division are called tumor suppressor genes. p53 is a well-known and one of the most crucial tumor suppressor genes (Joerger and Fersht, 2010). It also regulates apoptosis, senescence, and the DNA repair (Teodoro et al., 2007). This gene encodes a protein with the same name. The p53 protein consists of 393 amino acids and three domains: The transcriptional activation domain (TAD) (residues: 6-29), the sequence-specific DNA-binding domain (DBD) (residues: 100-288), and the tetramerization domain (TD) (residues: 319-357) are the domains of p53 protein (Petitjean et al., 2007). A demonstration of the domain organization and 3D structures of p53's domains were shown in Figure 5.

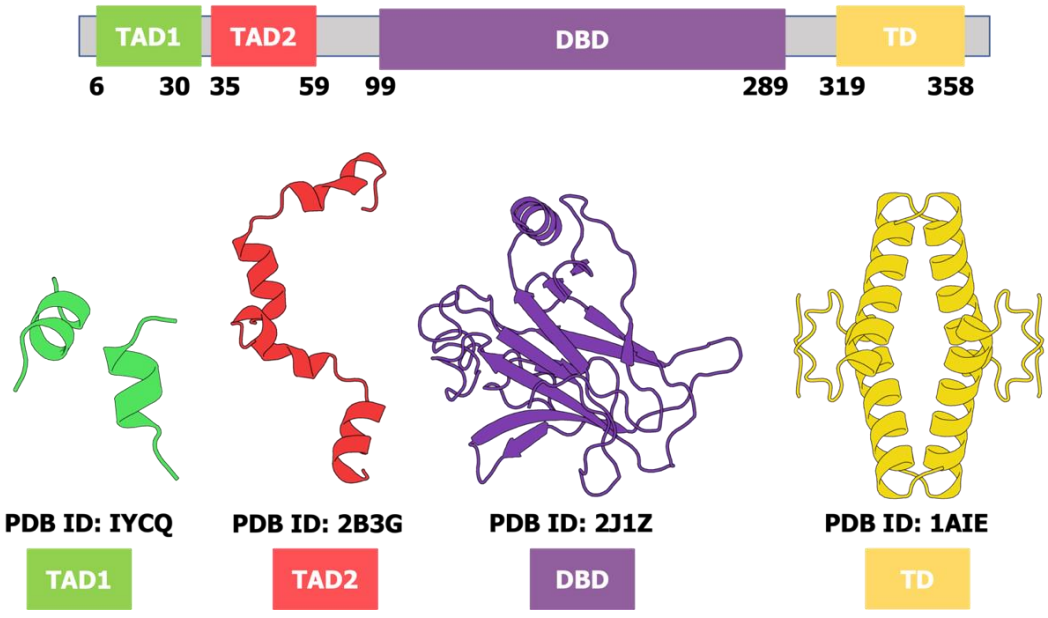



**Figure 5.** The domain organization and protein structures of the domain with their PDB IDs for p53 protein. Green is transcriptional activation domain 1 (TAD, PDB ID: 1YCQ).

Pink is transcriptional activation domain 2 (TAD2, PDB ID: 2B3G). Purple is DNA binding domain (DBD, PDB ID: 2J1Z). Yellow is the tetramerization domain (TD, PDB ID: 1AIE).

If DNA damage or toxicity occurs in the cell, the structure of p53 is stabilized by transduction pathways. It undergoes a conformational change to initiate the response programs such as apoptosis. On the other hand, the excessive level of p53 protein is controlled via MDM2 protein by the autoregulatory feedback loop in the unstressed cells (Moll and Petrenko, 2003).

### 4.2.2 MDM-p53 interaction

MDM2-p53 interaction is crucial for regulating the p53 protein level. MDM2 is an oncoprotein and acts as a ubiquitinate to prevent the accumulation of P53. It recognizes the N-terminal TAD1 of p53 and inhibits the transcriptional activity. MDM2 is overexpressed in cancer cells for blocking the p53 activation. Understanding the MDM2-p53 interaction may help to reactivate the p53 in cancer cells (Nag et al., 2013).

The structure of MDM2-p53 interaction was experimentally determined by X-ray crystallography in 2.6Å resolution and deposited in PDB with 1YCR id (Kussie et al., 1996). It consists of two chains that are MDM2 protein (109 residues) and the transactivation domain of p53 (15 residues). The interaction interface is highly hydrophobic, and three residues of p53 get deeper into the MDM2 cleft. These are F19, W23 and L26, and all are identified as key amino acids of p53 (Moll and Petrenko, 2003). 3D structure of the p53 protein and the location of key residues were shown in Figure 6.

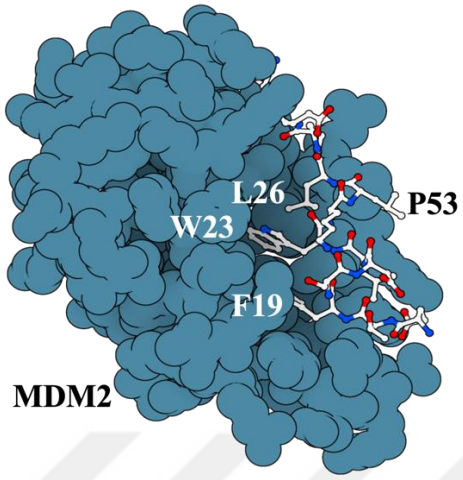



**Figure 6.** Structure of MDM2-p53 interaction. Critical residues are represented with one-letter amino acid codes on the structure. Petrol color represents the MDM2 protein and sticks formatting represents the p53 protein.

The MDM2-p53 interaction was chosen as the first case study to test the PROT-ON performance in detecting the critical residues with default parameters (5.0Å cut-off, 1.5 IQR, no PSSM file).

### 4.2.3 PROT-ON results for MDM2-p53

11 of 15 residues of p53 are identified as interfacial amino acids by the 5.0Å cut-off rule. These are E17, T18, F19, S20, L22, W23, L25, L26, P27, E28, and N29. All interfacial residues were mutated to other amino acids by EvoEF1 and FoldX algorithms. Totally 209 mutations were generated. Then their binding affinities were predicted. Overall mutation data were statistically analyzed. Heatmap and box-whisker plots were generated, and the most depleting and enriching mutations were proposed. Figure 7 shows the predicted ΔΔG interaction energies of each mutation with a heatmap. Each cell is colored according to the mutation's ΔΔG energy scores.

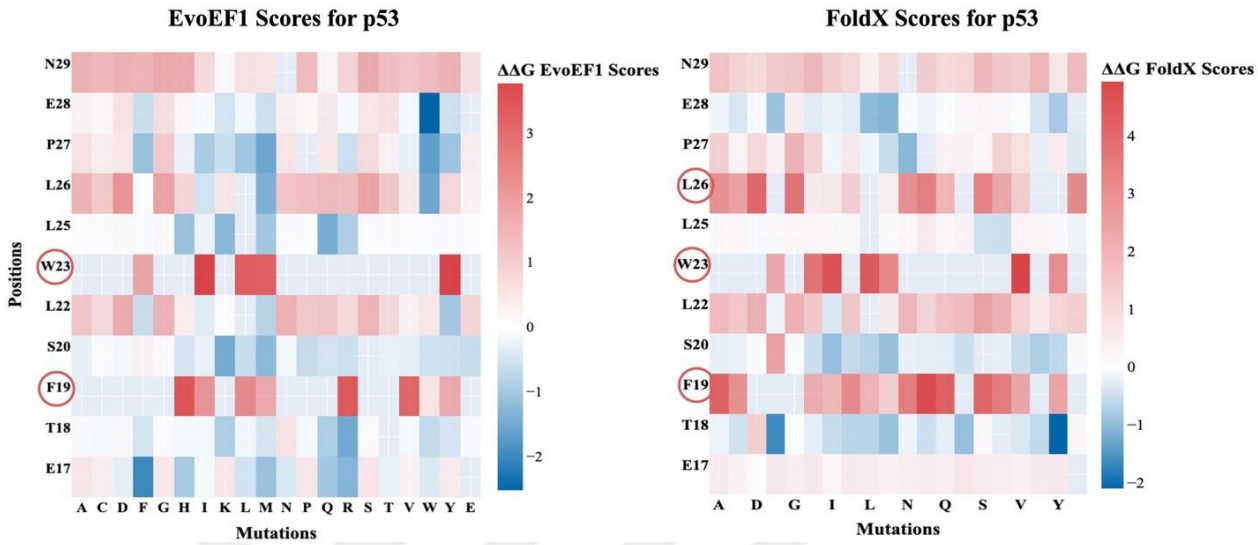



**Figure 7.** Heatmaps for EvoEF1 and FoldX respectively.

All interfacial amino acids and their mutations are displayed with estimated ΔΔG interactions on the heatmap red circle amino acids show the positive outlier positions. As mentioned in the method, heatmaps do not display the outliers of box-whiskers. They are represented as an empty cell in the heatmap. The red circles show the positive outlier mutation positions. These include more than one empty cell. All positive outliers belong to the F19 and W23 for EvoEF1 and additively L26 for FoldX. Notice that, these positions are key residues of p53. Also, most large volume hydrophobic mutations like isoleucine (I), and tyrosine (Y), are well-tolerated for these positions.

The distribution of ΔΔG scores of both algorithms is given with a box plot in Figure 8. These plots were drawn by default 1.5 IQR range. According to these plots, there were no detected negative outliers. Therefore, PROT-ON did not suggest any enriching mutations. On the other hand, there were 25 and 21 positive outliers detected for EvoEF1 and FoldX, respectively. These can be used as depleting mutations. All of these have an abolishing effect on the binding. The list of positive outliers is given in Appendix I and J for EvoEF1 and FoldX, together with ΔΔG interaction and ΔΔ stability scores. ΔΔG stability

scores can be used as a filtering mechanism for proposal mutations. Stabilizing depleting mutations were highlighted with light red. All FoldX depleting mutations are destabilizing effect on the p53 monomer except F19D and F19E. On the other hand, according to the EvoEF1, five W23 mutations have stabilizing effects: W23C, W23K, W23R, W23T, and W23V.

The most enriching and depleting mutations predicted by algorithms are given in Tables 1 and 2. Take note that, the most enriching mutations are converted to a hydrophobic residue for both algorithms.

**Table 1.** The most depleting and enriching mutations of p53 protein were estimated by EvoEF1.

| Mutation | ΔΔG EvoEF1 Score |
|---|---|
| F19G | 5.4 |
| E28W | -2.51 |

**Table 2.** The most depleting and enriching mutations of p53 protein were estimated by FoldX.

| Mutation | ΔΔG FoldX Score |
|---|---|
| L26Y | 11.41 |
| T18Y | -1.88 |

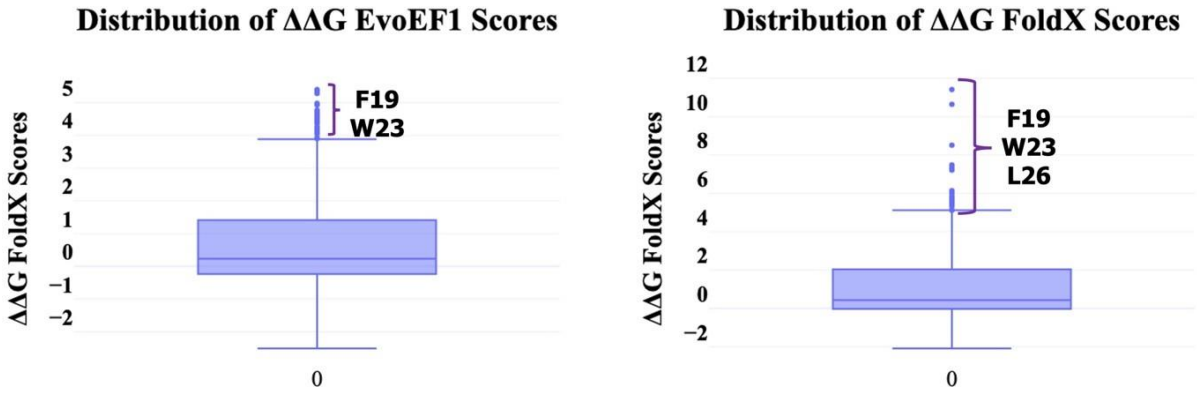



**Figure 8.** The distribution of the ΔΔG energy scores of mutations for EvoEF1 and FoldX.

The distribution of ΔΔG interaction energies for interfacial residues and all 20 amino acids are other outputs of PROT-ON. They are represented by box plots in Figure 9 and Figure 10 respectively. The red dash line in Figure 9 indicates the upper fence. The points above this border symbolize the critical positions for interaction.

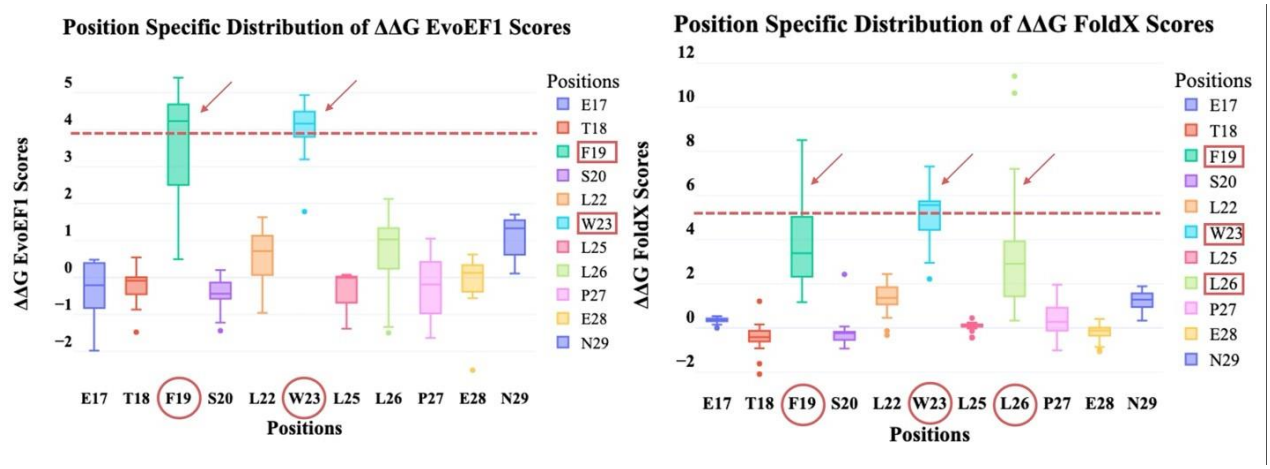



**Figure 9.** Position specific distribution of interfacial amino acids in p53 protein.

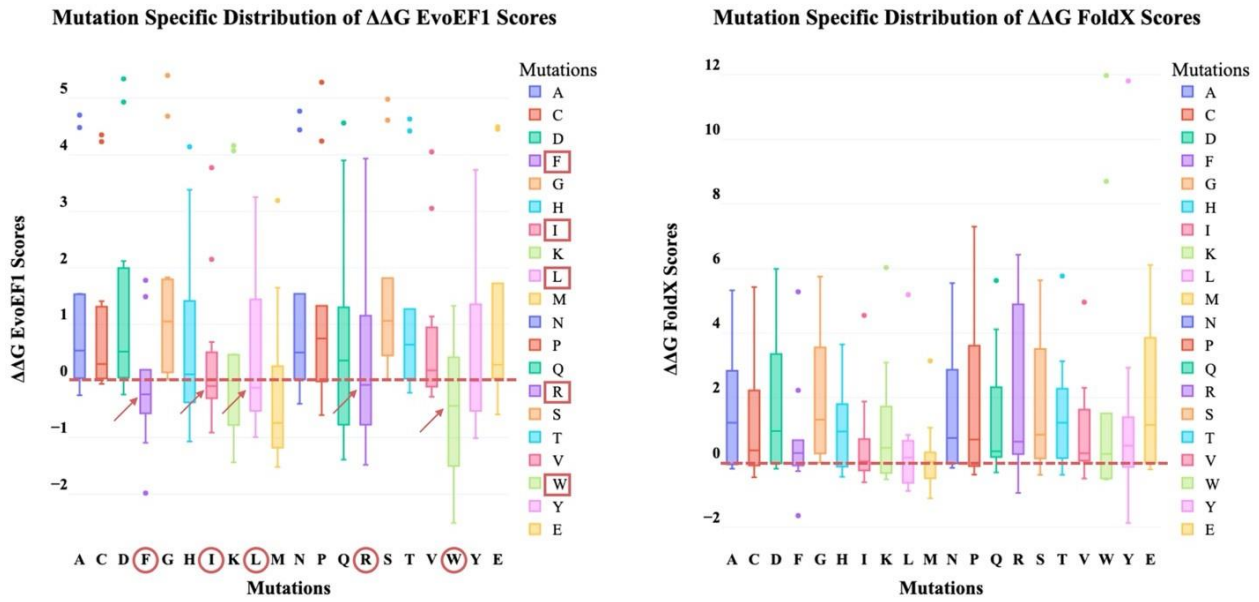



**Figure 10.** Mutation-specific distribution of interfacial amino acids in p53 protein.

Another essential point is investigating the distribution of ΔΔG interaction energies of 20 amino acids. It might give information about the most dominant interaction type in the interface but not always. The red dash line was drawn on a zero line. The red circles show the amino acids with a median of less than zero. Their analysis was not performed in this thesis.

## 4.3 PROT-ON Server Development

After successfully proofing our concept with MDM2-p53 interaction, we decided to develop a user-friendly web portal to be easily accessible for users.

### 4.3.1 Back-end development of the PROT-ON web server

#### 4.3.1.1 Creating a form to get the inputs from the user

PROT-ON has an input form in the "home" and "new query" pages to submit a job to the server. It includes twelve fields with seven different HTML input types, some of

which are required for running the job. Used HTML types are text, file, checkbox, range, radio, email, and submit. Except for the PSSM file option, all the rest are required fields.

Text fields are given in below:

- Run name: It is required to name the run. It is used in sending a mail to the user to inform the completion of which job.
- PDB ID: It is another required field. It is used in identifying a PDB ID to analyze the structure.
- E-mail: The run result is sent to the identified mail address.

File fields are given below:

- Structure file: If a protein structure file is present, it can be uploaded in this field.
- PSSM file: If a PSSM file is present in the CSV format, it can be uploaded here.

Checkbox: The checkboxes are positioned under the chains field. The checkbox below the chain of interest must be opened.

Range: There are two ranges in the input form. The first one is required to change the cut-off distance and the other is for the IQR range. The cut-off distance requires defining an interface. IQR range is used to scale the whisker length in box whiskers.

Radio: There are two radio buttons in the input form. These are used to select the algorithm to build mutations and estimate the binding free energies. EvoEF1 and FoldX are two options of PROT-ON. The details of these algorithms can be found in 3.7.1.1.1 and 3.7.1.1.2, respectively.

Cut-off distance, IQR range, and algorithms selections are called user-dependent parameters. The suggested mutations can be changed according to these selected

parameters. The default parameters of PROT-ON are 5.0Å cut-off distance, 1.5xIQR range, and EvoEF1 algorithm. The cut-off distance can be arranged from 3.0Å to 8.0Å and the IQR range from 1.5 to 2.5.

User preferences parameters of PROT-ON is illustrated with a job processing workflow in Figure 11.

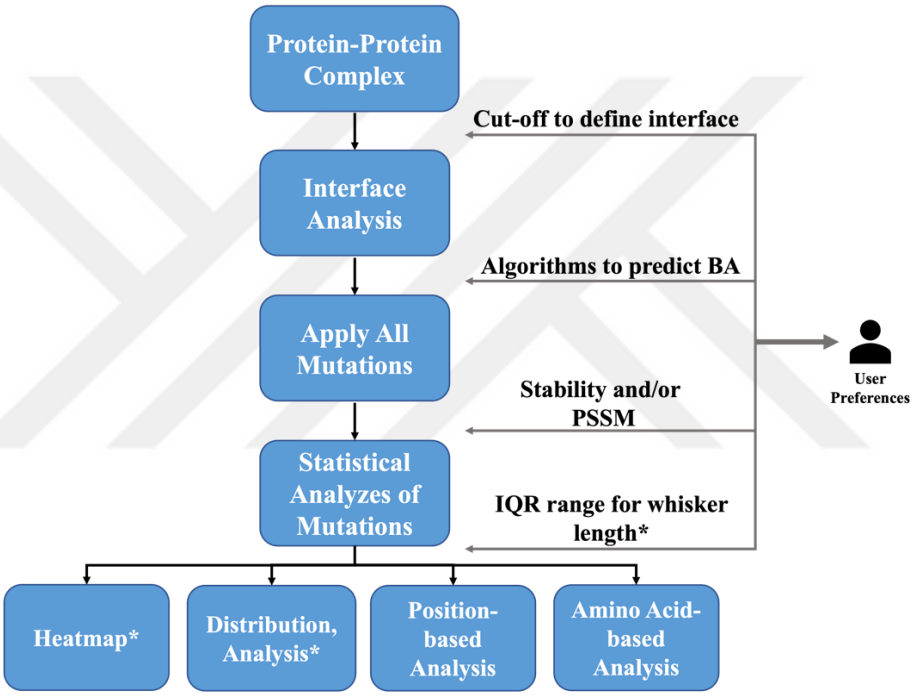



**Figure 11.** The automatic pipeline of PROT-ON and user-dependent parameters to change the proposal critical mutations.

#### 4.3.1.2 Input Alerts

PROT-ON implements flash messages to prevent any error before the job processing. Three functions in "alert.py" control the structure and/or PSSM inputs and if run into trouble, a danger message box is displayed on the homepage. The script takes structure, interacting chains, interesting chain, cut-off, and/or PSSM as parameters and functions operate following checks.

- If any chain that is specified as input is not found in the structure data, a flash message is screened to the user. (Message: Please check the chain IDs.)
- If the number of amino acids in an interesting protein sequence is not equal to the length of the first column (Position) of the PSSM file, a user warning appears on the homepage. (Message: Please check the PSSM file.)
- If there is no interfacial residue in the given cut-off distance, another user warning is displayed to the user. (Message: There are no interacting residues within that cut-off.)

These are the running preliminary control mechanisms to check if the job is submitted successfully. If one of the above errors occurs with the given inputs, job processing is stopped immediately, the uploaded file is removed from the system, a new query page is rendered, and a final flash message is displayed to warn the user. If given inputs do not encounter any warning during the checking process, a success flash message is displayed to the user and the job is run with further steps. (Message: Your run is queued. You will receive an email when your run is finished.)

#### 4.3.1.3 Processing the jobs in the background by Celery and RabbitMQ

In order to stand up the Celery on the Flask web server, a function is required. The function is used to manage or create the tasks or workers on Flask without requiring any reconfiguration. It firstly creates a "celery" object by the Celery function to update the backend and broker config data that will take from the Flask application. Then, it creates a subclass for running Celery tasks. This function is given in Appendix C.

After a Flask application is introduced by the Flask object, config data were updated as mentioned above. Also, a Celery application was created to define tasks. There are two Celery tasks: one runs the submitted jobs asynchronously and the other is a scheduled task working per night to remove expired run folders. All submitted jobs are sent to the message broker and then it is dispatched to an empty worker. The number of workers is

generally set depending on the CPU, cores, and task type. The "worker_concurrency" parameter is set to "1" in the config data to limit the worker size is only one to prevent uncontrolled file deletion during runs of the job. Flask application updating commands are given in Appendix D.

The asynchronous Celery task gets the structure file, interacting chain IDs, the interest of chain ID, cut-off value, IQR range, PSSM file, selected algorithm, user e-mail address, and run name as inputs to process the submitted job. All inputs are also parameters for the "task.py" script, which runs the jobs in the background. "Task.py" is called by the proton function in the app.py with the "apply_async" call option. Proton function also calls the main functions of the PROT-ON individual scripts to run the job. All created files are moved to the output folder and zipped in tar format. Finally, a link to show the result page is received to the user by email. While the "sudo celery -A app.celery worker –loglevel=info" command is used with the root user to run asynchronous tasks, "celery -A app.celery beat –loglevel=info" runs the scheduled tasks. The functions of asynchronous and scheduled tasks can be found in Appendix E. All these workflows are illustrated in Figure 12.

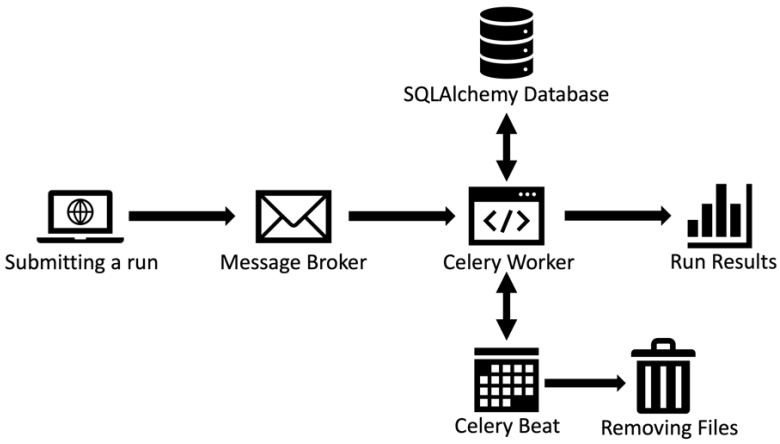



**Figure 12.** Workflow of the Celery and RabbitMQ servers to run tasks at the background.

### 4.3.2 Front-end development of the PROT-ON web server

#### 4.3.2.1 Grid System of the web server

Bootstrap grid layouts were used in the designing of homepage and result page for desktop and mobile versions of PROT-ON. These layouts consist of containers, rows, and columns, align the content., All website page layouts were inherited from the "layout.html" file. This file uses a container to fix the columns on the page. The logo and navbar menu are embedded in the header part of the container that is present on each page and columns are inserted into the rows. Mostly column 4 (it is indicated with col-4 in HTML.) was used to place the HTML input types like text, checkbox, submit, radio, slider, etc. Figure 13 indicates the grid layouts for the homepage and result page, respectively.

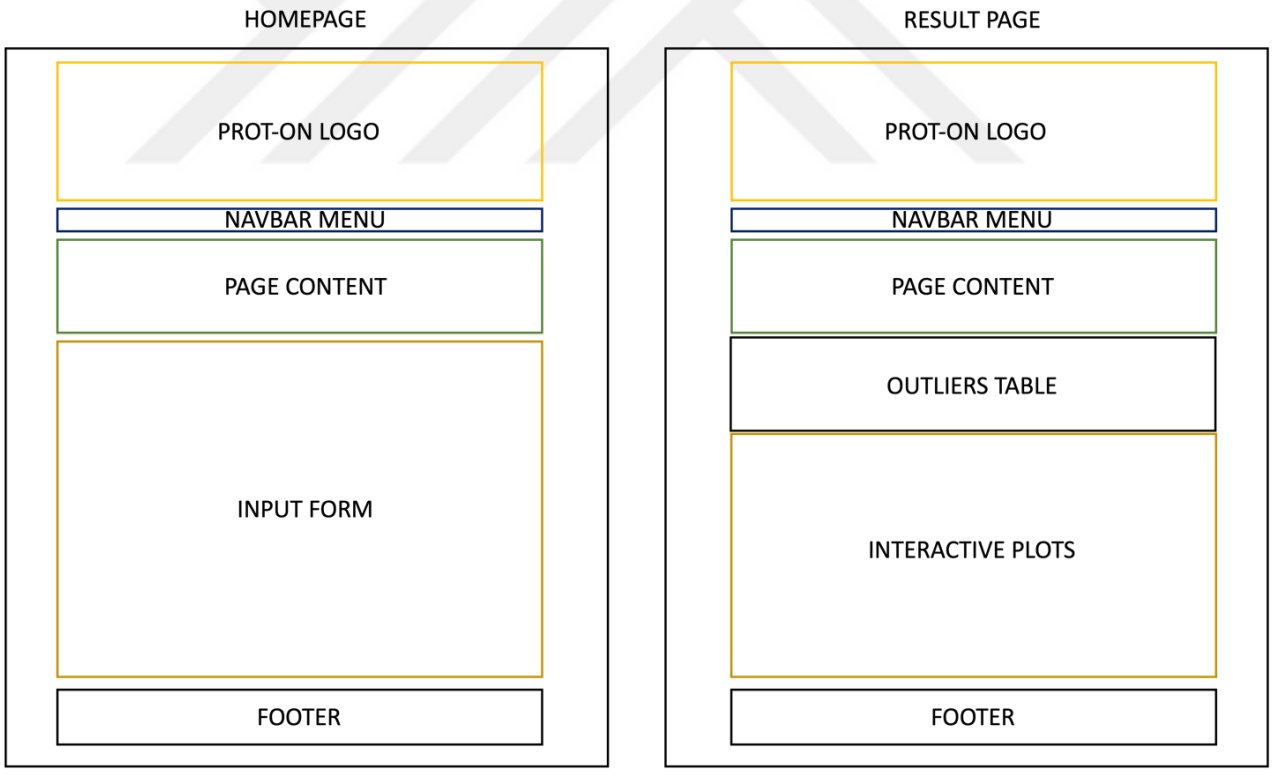



**Figure 13.** Grid system in homepage and result page of the PROT-ON webserver.

#### 4.3.2.2 Building navigator menu

Navigator menu was created by Bootstrap 3.4.1 version. It provides easy accessibility for pages. Four selections of the new query, help, download, and give feedback were placed into the navbar. The "new query" page gets a form to submit a new job to the server. The "help" page consists of a valuable tutorial explaining how to submit a job to the server. The "download" moves the user to the GitHub page to clone the PROT-ON standalone version for Linux or Mac OS computers. The last one "give feedback" section consists of a message form that delivers the users' positive or negative feedback to the developer team to improve the PROT-ON useability. "Modal" which is a JavaScript plugin of Bootstrap, was used for the last two selections to display the object into the dialog box on the current page without any page redirected.

#### 4.3.2.3 Displaying the result page

The result page was prepared with Flask, Bootstrap, and Python Plotly packages. It was divided into three sections. The first section contains several summarized information about the job result. The number of interfacial amino acids and generated mutation size is given in that section. Also, the most depleting and enriching mutations & positions are informed. The downloadable result link is present here. The second section consists of a table. It includes the positive and negative mutation data with their ΔΔG interaction and stability energies. It was created by JavaScript codes and has interactive features, i.e., focusing on rows and selecting or hiding columns. The last section contains interactive plots like heatmap and box-whiskers. All plots are created by Python Plotly. Wild-type positions, their mutations, and ΔΔG interaction energies are provided easily by holding the mouse on the interesting data points. Users can zoom in/out a part of the plots, autoscale, compare the data on hover, and save the interest graphs. URL link for the result page is shared with the user after the job is completed. This link is accessible for anyone who has the link for seven days. After seven days, the results are no longer available.

### 4.3.3 Deployment of the PROT-ON Web server

Gunicorn was set on the 8000th port for listening to the processes in the background. However, it is also required a supervisor monitor to restart the Gunicorn automatically when it is crashed. To do that, a supervisor file is created in "/etc/supervisor/conf.d/file_name." This directory does not change in all Linux systems. Then, the created file is configured with code snippets. This code snippet to configure the supervisor file is present in Appendix F. The file directory must show the PROT-ON path. app:app explains what your app name is. It was configured depending on the name of the Flask application script. The autostart and autorestart parameters serve for starting and restarting the program when the system is reopened, or the program encounters an error. The last two parameters are used to store error messages or log files. After reconfiguring the supervisor file, "sudo supervisorctl reread" command is run. It reread the supervisor files to control if any change is made. After that, "sudo service supervisor restart" is run to restart and confirm the changes on the system. Another command also may be run to check the server status: "sudo supervisorctl status". It shows how long your program has been running on the server.

After the installation of Nginx, an Nginx server's file must be configured. The configure file given in Appendix G. PROT-ON does not work with default Nginx settings. Normally, websites work with a default 80 port number. However, PROT-ON listens to the 8001-port number of IBG servers. Therefore, the port number should be written on the URL after the double dot. The server's name has to include the IP address of the workstation that works for PROT-ON and the DNS name (proton.tools.ibg.edu.tr). Remember that Gunicorn has listened to the 8000-port number of the workstation. It is also needed to state on the configuration file as a proxy pass. After completing all settings, the configuration file should be controlled if there is any error or not. It is controlled with "sudo nginx -t". If make sure there is no error, Nginx is required to restart to confirm the settings with "sudo service nginx restart". After all these processes, PROT-ON can be

accessible on the http:proton.tools.ibg.edu.tr:8001 website. Deployment of web site is given step by step in Figure 14.

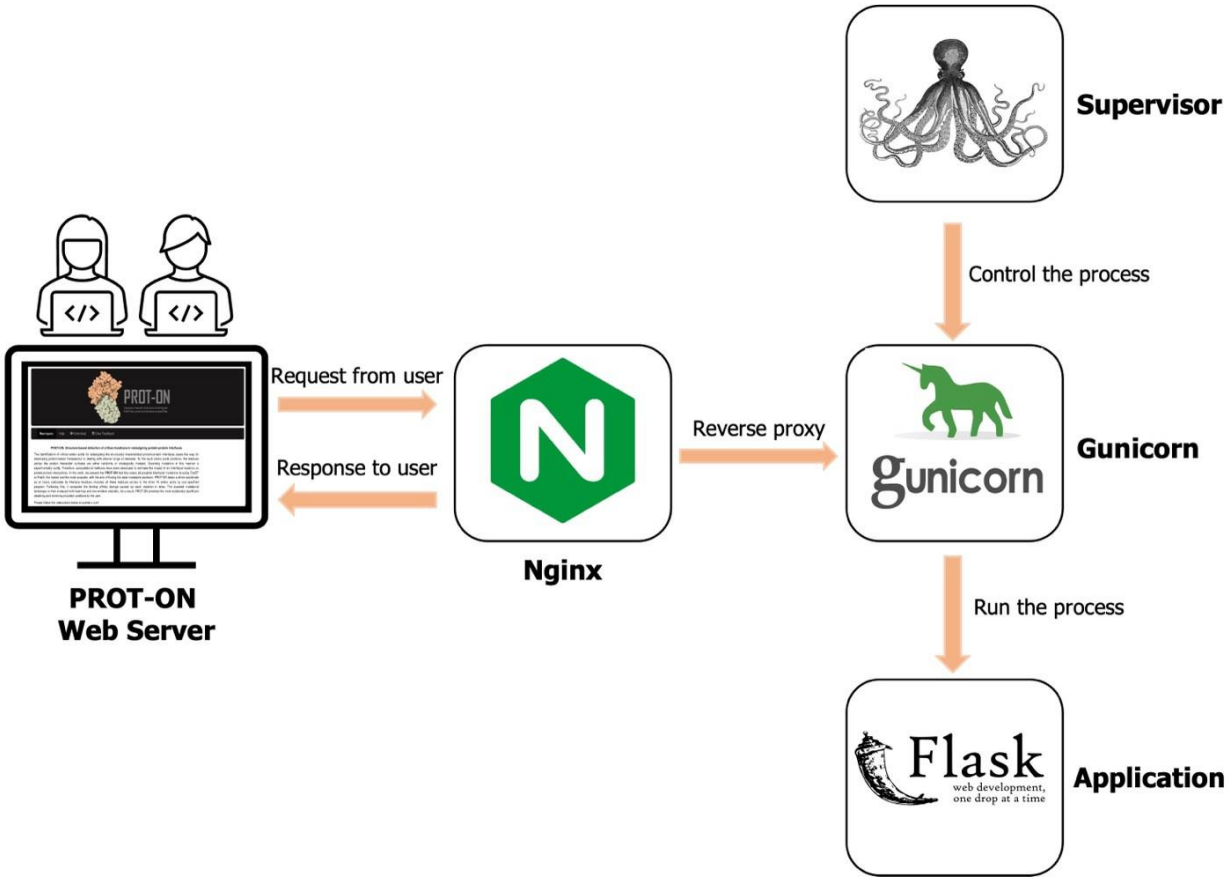



**Figure 14**. Workflow for the deployment of the PROT-ON web server.

## 4.4 Proof of Concept Study 2: AXL-GAS6 System

### 4.4.1 AXL protein

AXL is a transmembrane protein. It is embedded in the membrane and transduces signals by binding ligand molecules. Thus, it initiates crucial biological processes by activating other proteins (Zhang et al., 2018). It regulates many cellular mechanisms consisting of cell survival, proliferation, migration, and differentiation. AXL protein is encoded by the AXL gene. It is also called by other names like Tyro 7, UFO, ARK, or JTK11 (Li et al., 2009).

AXL is a member of the TAM family of receptor tyrosine kinase (RTK). TAM family includes Tyro 3, AXL, and Mer proteins. These have shared the same ligands consisting of growth arrest-specific 6 (GAS6) and S protein. They are called vitamin-K-dependent growth-potentiating factors and have a 43% amino acid sequence identity (Lee et al., 1999).

AXL protein consists of 894 amino acids. Its molecular weight is 104 kDa. It is also extended to 140 kDa by post-translational modifications like glycosylation. AXL has an extracellular transmembrane with an 85 kDa and an intracellular domain with 55 kDa molecular weights. Two immunoglobulin-like domains (Ig-like) (residues: 27-128 and 139-222) and two fibronectin type III (FNIII) (residues: 227-331 and 336-428) are constituted the extracellular domains. In the intracellular part, there is only kinase protein. It is so crucial for the auto-phosphorylation and kinase activity (Verma et al., 2011). A demonstration of their domain organization and structures of domains can be found in Figure 15.

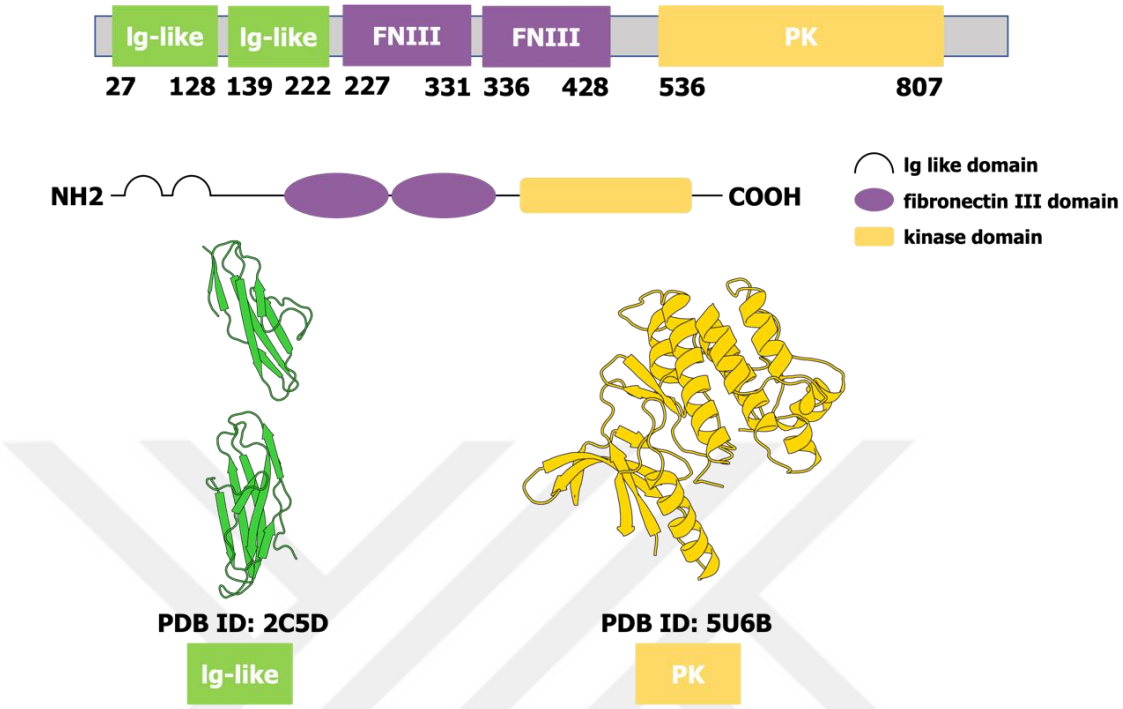



**Figure 15.** The domain organization and protein structures of the domain with their PDB IDs for AXL protein. Green is an immunoglobulin-like receptor (Ig-like, PDB ID:2C5D). Yellow is kinase domain (PK, PDB ID:5U6B).

### 4.4.2 AXL-GAS6 interaction

GAS6 is a ligand of the AXL receptor and activates the downstream signaling pathway like MAPK, PI3K, and AKT. It promotes the proliferation, migration, and survival of cancer cells. Also, it has been found to inhibit apoptosis in cancer cells. AXL is highly expressed in several metastatic carcinomas. All these shreds of evidence indicate the importance of understanding the pathogenesis of AXL-GAS6 binding for cancer cells (Cavet et al., 2010).

The structure of AXL-GAS6 interaction was experimentally determined with X-ray crystallography in 3.3Å resolution. It is deposited in the protein data bank with a 2C5D PDB ID (Sasaki et al., 2006). Interaction is mediated by the binding of immunoglobin domains of AXL and GAS6 proteins to each other. This complex has a 2:2 stoichiometry

and its global stoichiometry is hetero 4-mer - A2B2. While GAS6 protein contains 422 amino acids, AXL contains 195 amino acids. There are two distinct interaction regions as minor and major. Their size and binding affinities are so different. The minor interface is weaker than the major interface. There are three hydrogen bonds, one polar interaction, and several van der Waals contacts in the minor. On the other hand, many charged residues are located on the major interface, and they contact with GAS6. Also, six hydrogen bonds and several van der Waals contacts contribute to the interaction. The chains of interaction partners of the major interface are A and C for GAS6 and AXL respectively. The major interface was analyzed for this case study (Sasaki et al., 2006).

E56, E59, and T77 are identified as the critical residues for AXL protein (Sasaki et al., 2006). E56 and E59 interact with R308, R310, and K312 of GAS6 in the extensive region of the ß-sheet. T77 is also critical because of its presence in the hydrophobic abundance region (Duan et al., 2019).

It is demonstrated that the possible mutations of E59 and T77 dramatically reduced the binding affinity by surface plasmon resonance (Sasaki et al., 2006). According to the engineered decoy AXL protein paper (Kariolis et al., 2014), researchers suggest four mutations in the major sites. These mutations have an enhancing effect on binding to GAS6 protein if they are used as triple mutations. They do not strongly enhance individually. The Kd values of wild-type AXL protein and these mutations are given in Table 3 as individually and triple mutations.

**Table 3.** Reengineered AXL variants with their Kd values.

| Mutation | Kd (pM) |
|---|---|
| WT | 33.0 |
| G32S | 39.0 |
| D87G | 14.1 |

| | |
|---|---|
| V92A | 12.0 |
| G127R | 32.0 |
| D87G + V92A | 5.7 |
| G32S+D87G+V92A | 4.1 |

According to Karakulak T., etc., **R48, E56, E59, E70, D73, and E83** amino acids of AXL were found the most electrostatic contributor in the interface by HADDOCK refinement analysis. Detailed investigation of these residues with molecular dynamic simulations and evolutionary conservation energies suggested the **R48, E70, and E83** of AXL amino acids as specificity-determining positions (Karakulak et al., 2021).

The structure was given to the server by changing the cut-off distance and IQR range parameters in detecting of the critical positions and the enriching mutations illustrated above. Used parameter combinations are:

- Cut-off distance: 5.0Å IQR range: 1.5 (Default parameter)
- Cut-off distance: 6.0Å IQR range: 2.0
- Cut-off distance: 7.0Å IQR range: 2.5

PSSM was used to filter the suggested mutations for this case. An illustration of the structure of AXL-GAS6 interaction and critical residues can be found in Figure 16.

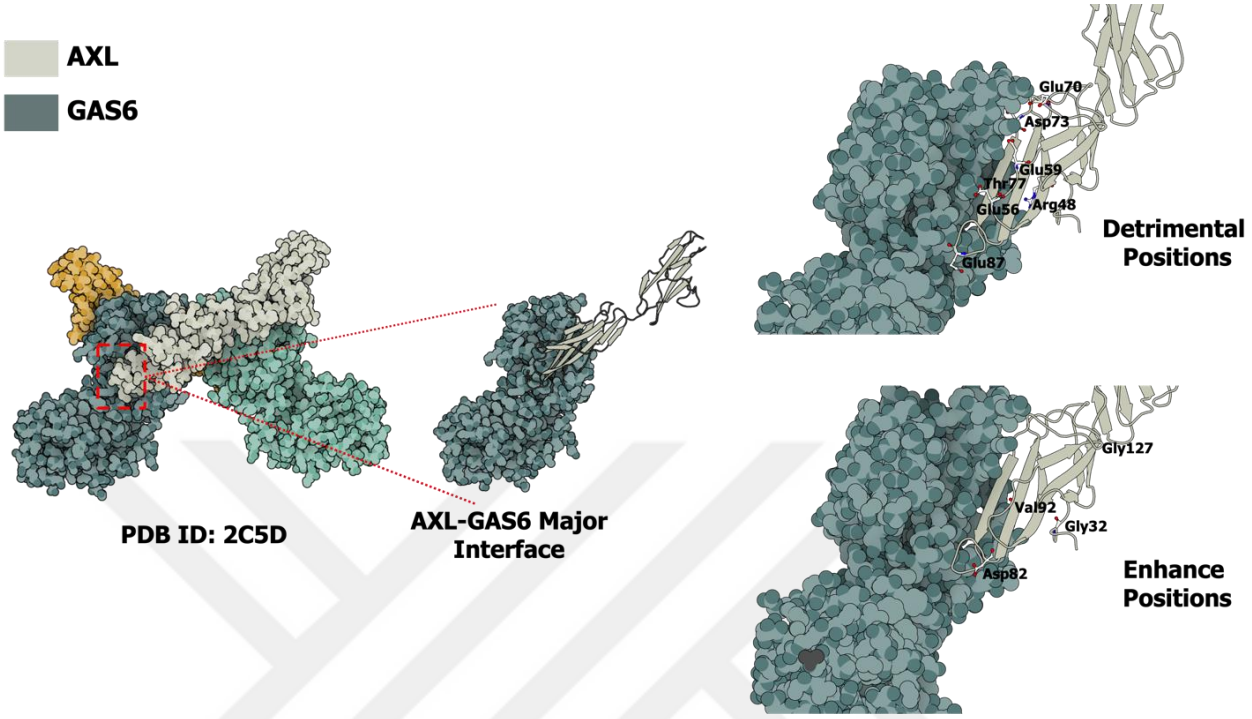



**Figure 16.** Structure of AXL-GAS6 interaction. Critical and engineered residues are represented with three-letter amino acid codes on the structure. Beige color represents the AXL protein and green represents the GAS6.

### 4.4.3 PROT-ON results for AXL-GAS6 interaction

According to the run results, 26, 30, and 34 interfacial residues were identified for 5.0Å, 6.0Å, and 7.0Å cut-off distances. T46, R48, E56, P57, E59, E70, L71, A72, D73, S74, T75, Q76, V79, P80, L81, G82, E83, D84, E85, W89, I90, V92, Q94, and R96 amino acids were identified for 5.0Å and they are common for other cut-off distances. In addition to these, while Q50, P58, V60, and H61 are identified for 6.0Å, L69, V91, S93, and T98 are identified for 7.0Å. Totally 494, 570, and 646 interfacial mutations were generated in these cut-off distances, respectively. Then their binding affinities were estimated. Overall mutation data were statistically analyzed. Finally, plots were generated according to the 1.5, 2.0, and 2.5 IQR ranges. Figure 17 shows predicted ΔΔG interaction energies of each

mutation with a heatmap for all cut-off distances. Each cell is colored according to the mutation's ΔΔG energy scores.

The arrows nearby the positions in heatmaps indicate the suggested depleting and enriching mutations by red and blue, respectively. According to the EvoEF1 scores, all mutations of R48 have an detrimental effect on binding. Therefore, it does not appear on the EvoEF1 heatmap for all cut-off distances, whereas it can be observed in the FoldX heatmap. None of the mutations of the R48 position were detected as the most depleting mutation by FoldX scores. However, they bind weaker than the wild-type. E56 and E59 were detected as one of the most depleting positions in all cut-off distances only by EvoEF1. None of these were detected as the most depleting by FoldX. However, they bind weaker than the wild-type like the R48 position. Please remember that R48, E56, and E59 are identified as critical residues. In addition to that, a mutation of S74 was detected as the most depleting by EvoEF1 in all cut-off distances. On the other hand, some mutations of P57, D73, T75, Q78, V79, and P80 were detected as the most depleting by FoldX in all cut-of distances.

The number of estimated enriching mutations by EvoEF1 is more than by FoldX. Enriching mutations are predominantly scattered between $70^{th}$ and $80^{th}$ positions for both algorithms. A few mutations of D73, T77, Q78, and E83 were proposed as enriching by EvoEF1 for all cut-off distances. On the other hand, a few mutations of T77 were detected as enriching for all cut-off distances by FoldX. Please remark that G32S, D87G, V92A, and G127R mutations are suggested by neither EvoEF1 nor FoldX. Because D32, D87, and G127 do not take place in the interface for all cut-off distances, PROT-ON did not analyze them. V92A was neither as depleting nor enriching by both algorithms in all cut-off distances. According to the designing decoy AXL protein paper, it has been observed that using the triple mutations of these mutations has increased the binding affinity. PROT-ON might make a botch of detecting these mutations because it does not perform triple

mutations. However, it can predict most of the detrimental positions correctly by both algorithms for this case.

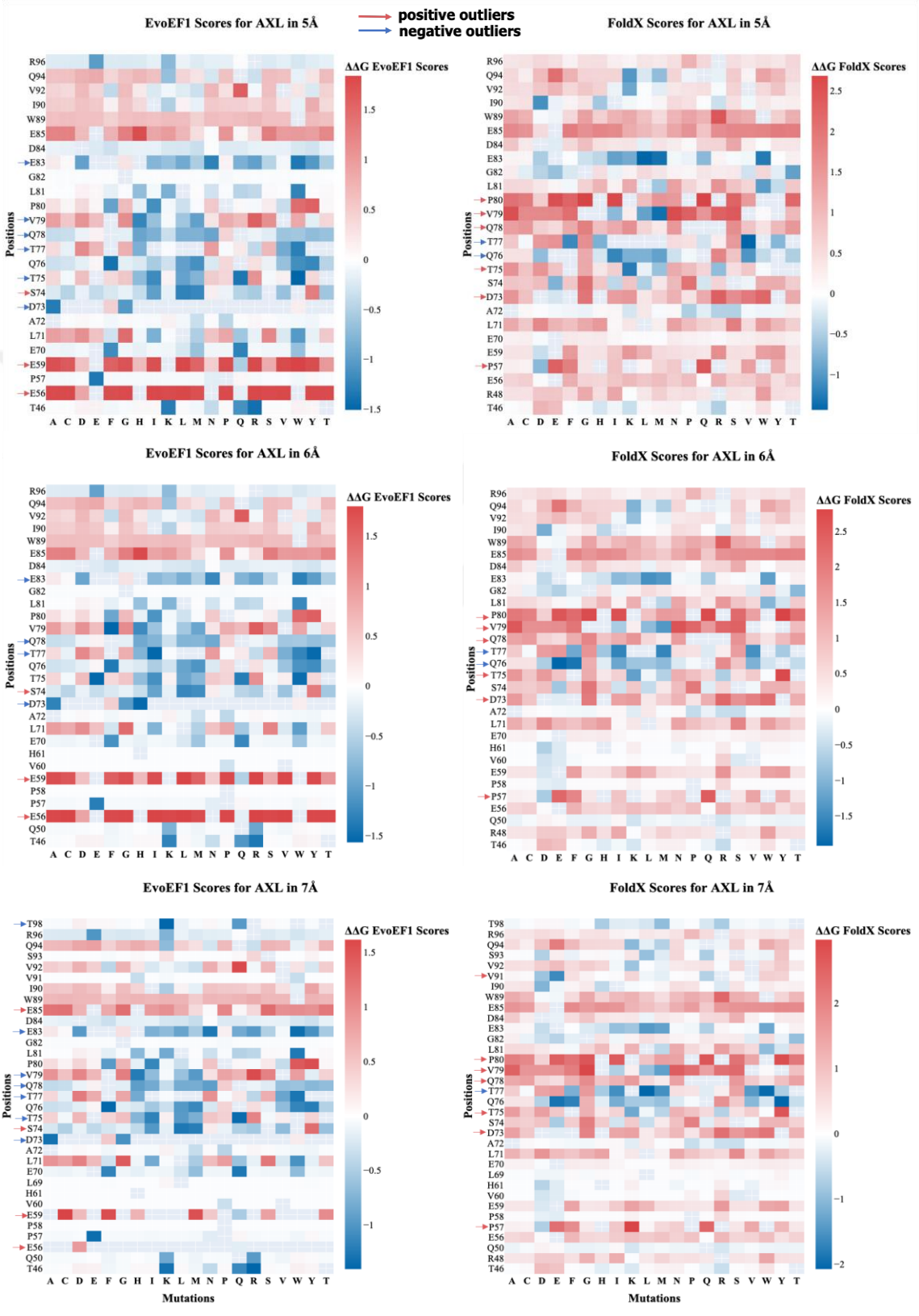

EvoEF1 Scores for AXL in 5Å
positive outliers
negative outliers
FoldX Scores for AXL in 5Å
ΔΔG EvoEF1 Scores
ΔΔG FoldX Scores
Positions
EvoEF1 Scores for AXL in 6Å
FoldX Scores for AXL in 6Å
EvoEF1 Scores for AXL in 7Å
FoldX Scores for AXL in 7Å
Mutations
A C D E F G H I K L M N P Q R S V W Y T

**Figure 17.** Heatmaps for EvoEF1 and FoldX in 5.0Å, 6.0Å, 7.0Å cut-off distances. All interfacial amino acids and their mutations are displayed with estimated ΔΔG interactions on the heatmap.

The distribution of ΔΔG scores of both algorithms is given with a box plot in Figure 18. These plots were drawn by 1.5, 2, and 2.5 IQR ranges respectively. According to these plots, many negative and positive outlier mutations were detected. Totally, 24 most depleting, and 25 most enriching mutations were suggested by EvoEF1 in 1.5 IQR. 19 most depleting and 1 most enriching mutation were proposed by FoldX in the same IQR. The number of proposed most depleting and most enriching mutations by other parameters is given in Appendix O. The list of all positive and negative outliers is given in Appendix K, and L for EvoEF1 and Appendix M and N FoldX together with ΔΔ interaction, ΔΔ stability, ΔΔ PSSM scores, cut-off distances, and IQR ranges. According to filtering mutations by both stability and PSSM are highlighted with red and blue in depleting and enriching mutations, respectively.

Most mutations of R48, E59, and E56 detrimental positions were filtered by stability and PSSM differences according to the EvoEF1 scores. Other depleting mutations can be filtered by either stability or PSSM. However, they cannot be filtered by both filtering mechanisms. Additionally, only T77I was filtered by stability and PSSM differences by EvoEF1 scores as enriching in 6.0Å and 7.0Å cut-off distances. Like depleting, most other enriching mutations can be filtered either by stability or PSSM.

D73F, D73Y, T75F, Q78R, and V89Y depleting mutations were filtered by both stability and PSSM score with FoldX in all cut-off distances. All other depleting mutations of FoldX can be filtered by PSSM but not by stability. In addition to depleting mutations, T77I was filtered by both filtering mechanisms in all cut-off distances. The most enriching and depleting mutations predicted by algorithms are given in Tables 4 and 5 with their cut-off distances.

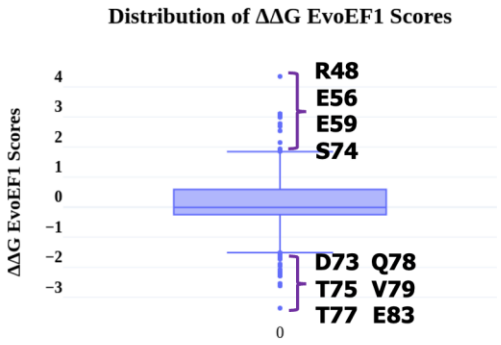
**Table 4.** The most depleting and enriching mutations of AXL protein estimated by EvoEF1 in different cut-off distances.

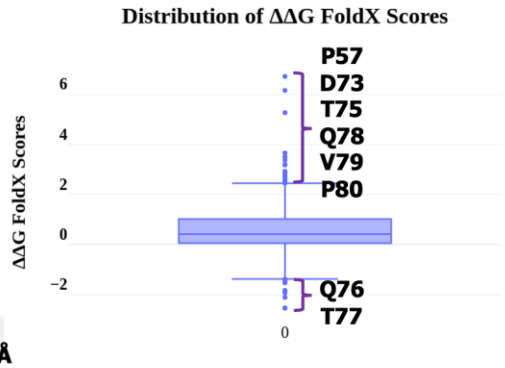
| Mutation | ΔΔG EvoEF1 Score | Cut-off Distance |
|---|---|---|
| S74W | 4.35 | 5.0, 6.0, 7.0 |
| D73M | -3.36 | 5.0, 6.0, 7.0 |

**Table 5.** The most depleting and enriching mutations of AXL protein estimated by FoldX in different cut-off distances.

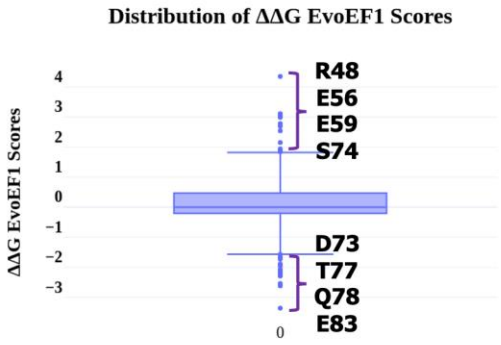
| Mutation | ΔΔG FoldX Score | Cut-off Distance |
|---|---|---|
| P80R | 8.07 | 5.0, 6.0, 7.0 |
| T77K | -2.23 | 5.0, 6.0, 7.0 |

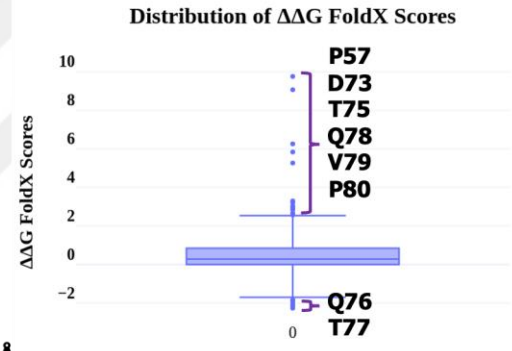
**Cut-off: 5.0Å**
**IQR Rule: 1.5**



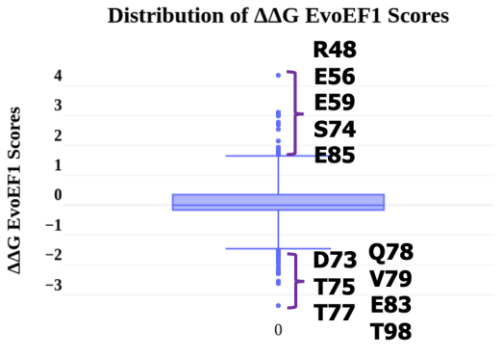




**Cut-off: 6.0Å**
**IQR Rule: 2.0**




**Cut-off: 7.0Å**
**IQR Rule: 2.5**



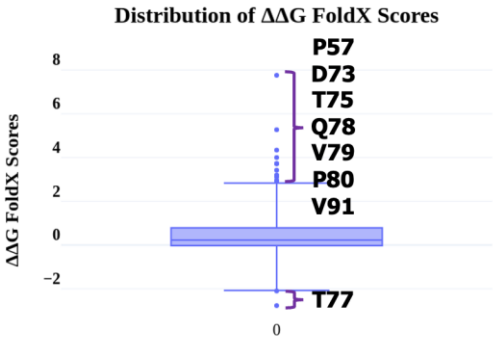



**Figure 18.** The distribution of the ΔΔG energy scores of mutations for EvoEF1 and FoldX in 5.0Å, 6.0Å, 7.0Å cut-off distances.

The distribution of ΔΔG interaction energies for interfacial residues and 20 amino acids was examined in Figure 19 and Figure 20. The red dash line indicates the upper fence. The points above this border symbolize the critical positions for interaction. Another essential point is investigating the distribution of ΔΔG interaction energies of 20 amino acids. It might give information about the most dominant interaction type in the interface. The red dash line was drawn on a zero line. The red circles show the amino acids with a median of less than zero. Their analysis was not performed in this thesis.

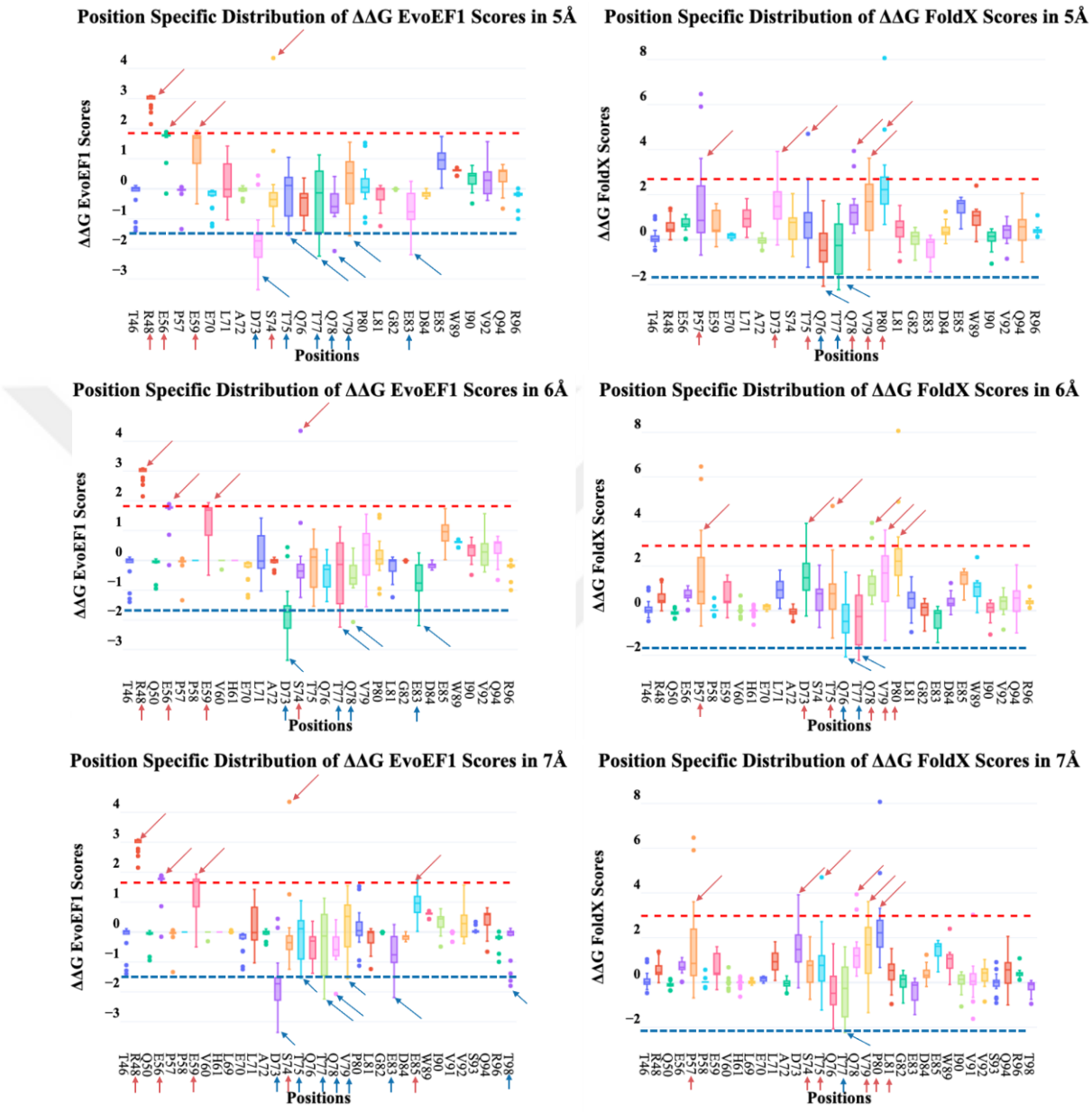



**Figure 19.** Position specific distribution of interfacial amino acids for p53 protein in 5.0Å, 6.0Å, 7.0Å cut-off distances.

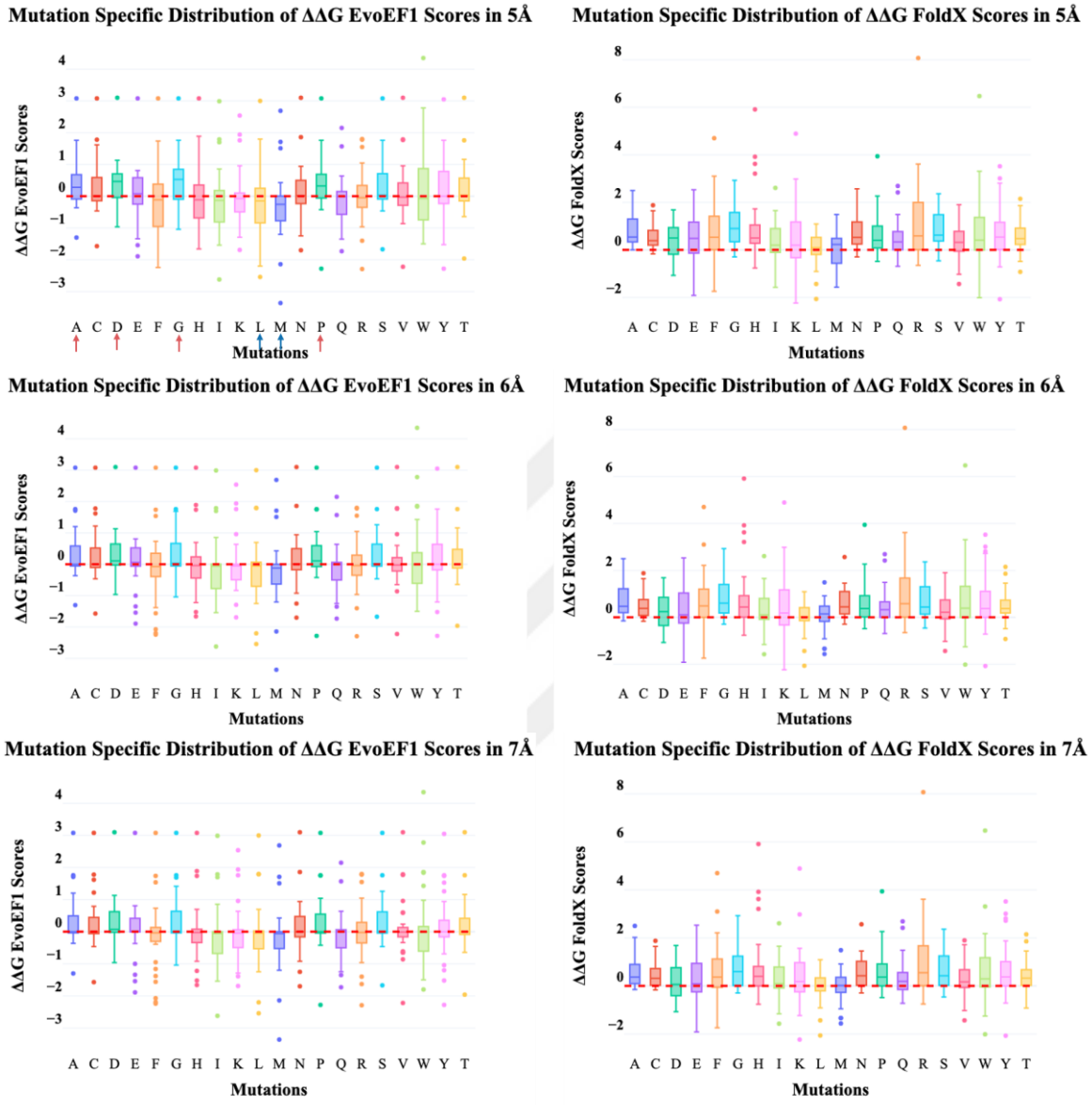



**Figure 20.** Mutation-specific distribution of interfacial amino acids for p53 protein in 5.0Å, 6.0Å, 7.0Å cut-off distances.

# 5 DISCUSSION

Computational approaches like EvoEF1, FoldX, BeAtMuSic, and iSEE estimate the binding affinities for PPIs. They are developed by using different methods such as empirical energy, statistical potential, or machine learning. FoldX and EvoEF1 are widely used and the most accurate empirical energy-based predictors. They are able to estimate the binding affinity by calculating the non-bonded interactions between atoms in the interface. However, they are incapable to classify all interfacial mutations statistically in a single run due to their manual pipeline and requiring additional scripts. On the other hand, the UEP algorithm has an automatic pipeline that can be used in classifying a set of mutations. However, it is insufficient to redesign the interface due to limitations in identifying the interfacial amino acids and does not make statistically analyze mutations. In this thesis, we have developed a novel automatic pipeline that proposes the designer mutations used in redesigning the interface and detects critical amino acids and mutations in the PPIs.

Our approach is named PROT-ON and can be accessed on a user-friendly web server through http://proton.tools.ibg.edu.tr or downloaded on Linux and macOS systems via https://github.com/CSB-KaracaLab/prot-on. PROT-ON is a structure-based technique and uses a protein dimer as an input. It then defines interfacial amino acids in the user of interest cut-off distance. We ranged the cut-off distance between 3.0Å and 8.0Å on the web server. It can be arranged manually on the standalone version. Later, all interfacial residues are converted to other 19 amino acids and their binding affinities are estimated by the user of interest algorithm: EvoEF1 or FoldX. PROT-ON makes a statistical analysis on estimated binding affinities of interfacial mutations and proposes the most enriching and depleting mutations that are outliers of the box-whisker plot. Proposing mutations may be changed by arranging the IQR range. We ranged the IQR range between 1.5 and 2.5 on the web server. It can be also arranged manually on the standalone version. PROT-ON generates three different box-whisker plots and a heatmap that displays the mutation

and their estimated energy scores except for proposing mutations in the last step. Generated box plots illustrate the distribution of all ΔΔG energy scores, position-specific and amino acid-specific. Our approach is not required user registration. It is fast, easy to use, user-friendly, and has user preferences parameters like cut-off distance, IQR range, and algorithm to change the suggested mutations. We have tested the PROT-ON performance to find the critical positions and mutations on MDM2-p53 and AXL-GAS6 complexes.

We used default PROT-ON parameters (5.0Å cut-off distance and 1.5 IQR range) to prove our concept on the p53 protein of the MDM2-p53 complex. PROT-ON identified E17, T18, F19, S20, L22, W23, L25, L26, P27, E28, and N29 amino acids as interfacial residues and generated 209 mutations. In the statistical analysis, it proposed 25 and 21 most depleting mutations according to EvoEF1 and FoldX, respectively. While F19 and W23 positions were detected as the most abolished position by EvoEF1, F19, W23, and L26 were detected as the most abolished by FoldX. Remember that F19, W23, and L26 are key residues of p53. PROT-ON successfully detected them as critical positions. Also suggested mutations were filtered by stability scores. W23C, W23K, W23R, W23T, and W23V were detected as stabilizing depleting mutations by using EvoEF1. On the other hand, F19D and F19E were stabilizing depleting by using FoldX. Other most depleting mutations had a destabilizer effect on the p53 protein.

We also tested the PROT-ON performance on the AXL protein of the AXL-GAS6 interaction. We used three different parameter combinations for AXL-GAS6 interaction. These were 5.0Å cut-off distance - 1.5 IQR range, 6.0Å cut-off distance - 2.0 IQR range, and 7.0Å cut-off distance - 2.5 IQR range. PROT-ON identified T46, R48, E56, P57, E59, E70, L71, A72, D73, S74, T75, Q76, V79, P80, L81, G82, E83, D84, E85, W89, I90, V92, Q94, and R96 interfacial residues in all cut-off distances. In addition to common amino acids, Q50, P58, V60, and H61 were identified in 6.0Å, and L69, V91, S93, and T98 were identified in 7.0Å cut-off distances. It generated 494, 570, and 646 mutations in these

cut-off distances respectively. In the statistical analysis, it proposed 24, 19, and 17 most depleting, 25, 20, and 28 most enriching mutations by using EvoEF1 in 1.5, 2.0, and 2.5 IQR ranges, respectively. On the other hand, it proposed 19, 17, and 16 most depleting, 1, 4, and 1 enriching mutation by using FoldX in the same IQR ranges. Many mutations on the R48, E56, E59, and S74 positions are suggested as detrimental positions by using EvoEF1 in all cut-off distances and IQR ranges. However, these positions were not suggested by using FoldX even though their mutations have a depleting effect on the binding. A few of the R48 and E56 most depleting mutations of EvoEF1 were filtered by both stability and PSSM scores in all cut-off distances and most mutations of E59 only in 7.0Å cut-off distance. S74W was not filtered by the stability score of EvoEF1. Meanwhile, a few mutations P57, D73, T75, Q78, V79, and P80 positions are suggested as abolishing positions by FoldX in all cut-off distances. However, FoldX depleting mutations D73, T75, Q78, and V79 were filtered by both stability and PSSM scores except mutations in P57 and P80 positions. Remember that R48, E56, E59, T70, D73, T77, and E83 positions of AXL protein are key residues, and R48, E56, and E59 are suggested by only EvoEF1, whereas D73 by FoldX as a detrimental effect on the binding through filtering mechanism. Surprisingly T77I, T77M, and T77Y were detected as enriching mutations by both EvoEF1 and FoldX. Also, T77I was filtered with stability and PSSM scores by both algorithms. Normally this position was identified as critical due to mutating it to arginine has no binder. However, mutating it to a hydrophobic amino acid may have the potential to increase the interaction capability due to its hydrophobic environment.

Algorithms may be conflicted on some mutations like in the D73 position of AXL. While EvoEF1 proposed the many mutations in the D73 position as most enriching, FoldX proposed them as most depleting. Remember that D73 is one of the most contributing amino acids located in the core region of the interface. It may have an abolishing effect by mutating to one of the other types of amino acids. We did not study to solve the conflict of algorithms problem to propose the most related mutations. On the other hand, some small molecules may be present on the interface and interact with a few interfacial

residues through complex formation. PROT-ON cannot predict the mutations accurately that interact with the small molecules due to EvoEF1 and FoldX being specialized for protein-protein interaction. Additionally, FoldX can also predict the impact of interfacial mutations that interact with nucleic acids. However, PROT-ON codes are not specialized for protein-nucleic acid interactions. In future studies, we are planning to offer a method to solve the conflicting algorithm problem to propose the most related mutations. Also, we aim to update PROT-ON codes for redesigning the protein-nucleic acid interface by using the FoldX functions.

PROT-ON was developed for researchers who study protein targeting drug development, redesigning interfaces, and detecting critical amino acids. It also provides flexibility to the users through the changeability parameters like the cut-off distances, IQR range, and binding affinity predictors. Our view is that PROT-ON will be a guide to protein engineering studies especially, regarding redesigning the interface.

# 6 CONCLUSION AND SUGGESTIONS

- In this study, we developed a structure-based mutation classification tool to detect the critical amino acid positions in a PPI and redesign the interface.
- We decided to use EvoEF1 and FoldX algorithms in building mutation models and estimating the ΔG energy for each mutation.
- We define three user-dependent parameters to change the suggested mutations in the statistical analyzing step. These are cut-off distances to define interfacial residues. IQR range is used to change the whisker length of the box plot and it mediately provides suggesting different outliers. The last user-dependent parameter is the selection of an algorithm to predict the interaction energies of mutations.
- We developed a user-friendly interactive web server. It is accessible through http://proton.tools.ibg.edu.tr. Also, we put into service a standalone PROT-ON version that can be cloned into a Linux or Mac operating system. It can be downloaded through https://github.com/CSB-KaracaLab/prot-on
- MDM2-p53 interaction was used as a proof-of-concept study with default parameters of PROT-ON. It successfully predicted the most critical amino acid positions of p53 protein are F19, W23, and L26 by both EvoEF1 and FoldX algorithms.
- AXL-GAS6 interaction was tested with different parameters on PROT-ON. Experimentally verified key residues of AXL that are R48, E56 and E59 were successfully detected as the most abolishing positions by EvoEF1.
- D73F, D73Y, T75F, Q78R, and V79R are detected as the most depleting mutation by FoldX. They are also filtering mutations by both stability and PSSM scores in all cut-off distances and IQR ranges.
- T77I mutation is suggested as an enriching mutation by both EvoEF1 and FoldX algorithms. It is only one enriching mutation filtered by stability and PSSM scores.

- We suggest that proposed mutations by PROT-ON can be experimentally validated to redesign a better binder or no binder AXL interface.

# 7 REFERENCES


ACUNER, S., KOCAER, A., ÇELIK, R. H., TURANLI, B., NIKEREL, E., KARACA, E., TATAR YILMAZ, G., ALOTAİBİ, H., DEMIRCI, H., ARGA, K., ERSOY TUNALI, N., KURKCUOGLU, O., ŞENSOY, Ö., OZBEK, P., KALYONCU, S. & ÇAKIR, T. 2021. Protein Yapısı, Mühendisliği, Etkileşimleri, Dinamiği ve İlaç Tasarımındaki Yeri ISBN : 9786257564038.

AMENGUAL-RIGO, P., FERNÁNDEZ-RECIO, J. & GUALLAR, V. 2021. UEP: an open-source and fast classifier for predicting the impact of mutations in protein–protein complexes. Bioinformatics, 37**,** 334-341.

BARTELL, L. 1968. Molecular geometry: Bonded versus nonbonded interactions. Journal of Chemical Education, 45**,** 754.

BRAUN, P. & GINGRAS, A. C. 2012. History of protein–protein interactions: From egg‐white to complex networks. Proteomics, 12**,** 1478-1498.

CAVET, M. E., SMOLOCK, E. M., MENON, P., KONISHI, A., KORSHUNOV, V. A. & BERK, B. C. 2010. Gas6-Axl pathway: the role of redox-dependent association of Axl with nonmuscle myosin IIB. Hypertension, 56**,** 105-111.

CHAKRABARTI, P. & JANIN, J. 2002. Dissecting protein–protein recognition sites. Proteins: Structure, Function, and Bioinformatics, 47**,** 334-343.

CONSORTIUM, G. P. 2010. A map of human genome variation from population scale sequencing. Nature, 467**,** 1061.

CONTE, L. L., CHOTHIA, C. & JANIN, J. 1999. The atomic structure of protein-protein recognition sites. Journal of molecular biology, 285**,** 2177-2198.

DEHOUCK, Y., KWASIGROCH, J. M., ROOMAN, M. & GILIS, D. 2013. BeAtMuSiC: prediction of changes in protein–protein binding affinity on mutations. Nucleic acids research, 41**,** W333-W339.

DUAN, Y., HU, B., QIAO, C., LUO, L., LI, X., WANG, J., LIU, H., ZHOU, T., SHEN, B. & LV, M. 2019. Engineered AXL-ECD-Fc variants that abolish the AXL/Gas6 interaction suppress tumor cell migration. Oncology letters, 17**,** 5784-5792.

FERNÁNDEZ, A. & SCHERAGA, H. A. 2003. Insufficiently dehydrated hydrogen bonds as determinants of protein interactions. Proceedings of the National Academy of Sciences, 100**,** 113-118.

GENG, C., VANGONE, A. & BONVIN, A. M. 2016. Exploring the interplay between experimental methods and the performance of predictors of binding affinity change upon mutations in protein complexes. Protein Engineering, Design and Selection, 29**,** 291-299.

GENG, C., VANGONE, A., FOLKERS, G. E., XUE, L. C. & BONVIN, A. M. 2019a. iSEE: Interface structure, evolution, and energy‐based machine learning predictor of binding affinity changes upon mutations. Proteins: Structure, Function, and Bioinformatics, 87**,** 110-119.

GENG, C., XUE, L. C., ROEL‐TOURIS, J. & BONVIN, A. M. 2019b. Finding the ΔΔG spot: Are predictors of binding affinity changes upon mutations in protein–protein

interactions ready for it? Wiley Interdisciplinary Reviews: Computational Molecular Science, 9**,** e1410.
HALGREN, T. A. & DAMM, W. 2001. Polarizable force fields. Current opinion in structural biology, 11**,** 236-242.
JANKAUSKAITĖ, J., JIMÉNEZ-GARCÍA, B., DAPKŪNAS, J., FERNÁNDEZ-RECIO, J. & MOAL, I. H. 2019. SKEMPI 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. Bioinformatics, 35**,** 462-469.
JOERGER, A. C. & FERSHT, A. R. 2010. The tumor suppressor p53: from structures to drug discovery. Cold Spring Harbor perspectives in biology, 2**,** a000919.
KARAKULAK, T., RIFAIOGLU, A. S., RODRIGUES, J. P. & KARACA, E. 2021. Predicting the Specificity-Determining Positions of Receptor Tyrosine Kinase Axl. Frontiers in molecular biosciences**,** 368.
KARIOLIS, M. S., MIAO, Y. R., JONES, D. S., KAPUR, S., MATHEWS, I. I., GIACCIA, A. J. & COCHRAN, J. R. 2014. An engineered Axl'decoy receptor'effectively silences the Gas6-Axl signaling axis. Nature chemical biology, 10**,** 977-983.
KASTRITIS, P. L. & BONVIN, A. M. 2013. On the binding affinity of macromolecular interactions: daring to ask why proteins interact. Journal of The Royal Society Interface, 10**,** 20120835.
KUSSIE, P. H., GORINA, S., MARECHAL, V., ELENBAAS, B., MOREAU, J., LEVINE, A. J. & PAVLETICH, N. P. 1996. Structure of the MDM2 oncoprotein bound to the p53 tumor suppressor transactivation domain. Science, 274**,** 948-953.
LEE, W.-P., LIAO, Y., ROBINSON, D., KUNG, H.-J., LIU, E. T. & HUNG, M.-C. 1999. Axl-gas6 interaction counteracts E1A-mediated cell growth suppression and proapoptotic activity. Molecular and cellular biology, 19**,** 8075-8082.
LI, Y., YE, X., TAN, C., HONGO, J. A., ZHA, J., LIU, J., KALLOP, D., LUDLAM, M. J. C. & PEI, L. 2009. Axl as a potential therapeutic target in cancer: role of Axl in tumor growth, metastasis and angiogenesis. Oncogene, 28**,** 3442-3455.
LUTZ, S. & IAMURRI, S. M. 2018. Protein engineering: past, present, and future. Protein Engineering**,** 1-12.
MARSHALL, S. A., LAZAR, G. A., CHIRINO, A. J. & DESJARLAIS, J. R. 2003. Rational design and engineering of therapeutic proteins. Drug discovery today, 8**,** 212-221.
MOLL, U. M. & PETRENKO, O. 2003. The MDM2-p53 Interaction. Molecular Cancer Research, 1**,** 1001-1008.
MOREIRA, I. S., FERNANDES, P. A. & RAMOS, M. J. 2007. Hot spots—A review of the protein–protein interface determinant amino‐acid residues. Proteins: Structure, Function, and Bioinformatics, 68**,** 803-812.
MOSCA, R., CÉOL, A. & ALOY, P. 2013. Interactome3D: adding structural details to protein networks. Nature methods, 10**,** 47-53.
NAG, S., QIN, J., SRIVENUGOPAL, K. S., WANG, M. & ZHANG, R. 2013. The MDM2-p53 pathway revisited. Journal of biomedical research, 27**,** 254-271.

NICHOLLS, A., SHARP, K. A. & HONIG, B. 1991. Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons. Proteins: Structure, Function, and Bioinformatics, 11**,** 281-296.

NORTHROP, J. H. 1922. THE INACTIVATION OF TRYPSIN: II. The Equilibrium between Trypsin and the Inhibiting Substance Formed by Its Action on Proteins. The Journal of general physiology, 4**,** 245.

PEARCE, R., HUANG, X., SETIAWAN, D. & ZHANG, Y. 2019. EvoDesign: designing protein–protein binding interactions using evolutionary interface profiles in conjunction with an optimized physical energy function. Journal of molecular biology, 431**,** 2467-2476.

PETITJEAN, A., MATHE, E., KATO, S., ISHIOKA, C., TAVTIGIAN, S. V., HAINAUT, P. & OLIVIER, M. 2007. Impact of mutant p53 functional properties on TP53 mutation patterns and tumor phenotype: lessons from recent developments in the IARC TP53 database. Human mutation, 28**,** 622-629.

ROY, A., NAIR, S., SEN, N., SONI, N. & MADHUSUDHAN, M. S. 2017. In silico methods for design of biological therapeutics. Methods, 131**,** 33-65.

SASAKI, T., KNYAZEV, P. G., CLOUT, N. J., CHEBURKIN, Y., GÖHRING, W., ULLRICH, A., TIMPL, R. & HOHENESTER, E. 2006. Structural basis for Gas6–Axl signalling. The EMBO Journal, 25**,** 80-87.

SCHYMKOWITZ, J., BORG, J., STRICHER, F., NYS, R., ROUSSEAU, F. & SERRANO, L. 2005. The FoldX web server: an online force field. Nucleic acids research, 33**,** W382-W388.

SIEBENMORGEN, T. & ZACHARIAS, M. 2020. Computational prediction of protein–protein binding affinities. WIREs Computational Molecular Science, 10**,** e1448.

STITES, W. E. 1997. Protein− protein interactions: interface structure, binding thermodynamics, and mutational analysis. Chemical reviews, 97**,** 1233-1250.

TEODORO, J. G., EVANS, S. K. & GREEN, M. R. 2007. Inhibition of tumor angiogenesis by p53: a new role for the guardian of the genome. Journal of molecular medicine, 85**,** 1175-1186.

VAN ZUNDERT, G., RODRIGUES, J., TRELLET, M., SCHMITZ, C., KASTRITIS, P., KARACA, E., MELQUIOND, A., VAN DIJK, M., DE VRIES, S. & BONVIN, A. 2016. The HADDOCK2. 2 web server: user-friendly integrative modeling of biomolecular complexes. Journal of molecular biology, 428**,** 720-725.

VANOMMESLAEGHE, K., HATCHER, E., ACHARYA, C., KUNDU, S., ZHONG, S., SHIM, J., DARIAN, E., GUVENCH, O., LOPES, P. & VOROBYOV, I. 2010. CHARMM general force field: A force field for drug‐like molecules compatible with the CHARMM all‐atom additive biological force fields. Journal of computational chemistry, 31**,** 671-690.

VERMA, A., WARNER, S. L., VANKAYALAPATI, H., BEARSS, D. J. & SHARMA, S. 2011. Targeting Axl and Mer Kinases in CancerAxl and Mer in Cancer. Molecular cancer therapeutics, 10**,** 1763-1773.

WANG, J., WOLF, R. M., CALDWELL, J. W., KOLLMAN, P. A. & CASE, D. A. 2004. Development and testing of a general amber force field. Journal of computational chemistry, 25**,** 1157-1174.
ZHANG, G., WANG, M., ZHAO, H. & CUI, W. 2018. Function of Axl receptor tyrosine kinase in non-small cell lung cancer. Oncology letters, 15**,** 2726-2734.

## APPENDIX A

Instruction for installing the rabbitmq-server, adding a user, creating host and set permissions for the RabbitMQ.

```
sudo apt-get install rabbitmq-server
sudo rabbitmq-server -detached
sudo rabbitmqctl add_user proton *******
sudo rabbitmqctl add_vhost myvhost
sudo rabbitmqctl set_permissions -p myvhost proton ".*" ".*" ".*"
```

## APPENDIX B

Creating Celery broker and Celery backend URLs for configuration of Celery.

```
CELERY_BROKER_URL=amqp://proton: *******@localhost/myvhost
CELERY_BACKEND_URL=db+sqlite:///proton.db
```

## APPENDIX C

Automatic Celery function for creating background tasks in a Flaks application

```
def make_celery(app):
    celery = Celery(
        app.import_name,
        backend=app.config["CELERY_BACKEND_URL"],
        broker=app.config["CELERY_BROKER_URL"]
    )
    celery.conf.update(app.config)


    class ContextTask(celery.Task):
        def__call__(self, *args, **kwargs):
            with app.app_context():
                return self.run(*args, **kwargs)

    celery.Task = ContextTask
    return celery
```

## APPENDIX D

Commands for updating the configuration file of the Flask application.

```
flask_app = Flask(__name__)
flask_app.config.update(
    CELERY_BROKER_URL=os.environ.get("CELERY_BROKER_URL"),
    CELERY_BACKEND_URL=os.environ.get("CELERY_BACKEND_URL"),
    SECRET_KEY = os.environ.get("SECRET_KEY"),
    MAIL_SERVER = "smtp-relay.gmail.com",
    MAIL_PORT = 587,
    MAIL_USE_TLS = True,
    MAIL_USERNAME = "proton.tools@ibg.edu.tr",
    MAIL_DEFAULT_SENDER = "proton.tools@ibg.edu.tr",
    worker_concurrency = 1,
    beat_schedule = {
        # Executes every night
        'periodic_task-every-night': {
            'task': 'periodic_task',
            'schedule': crontab(minute=0, hour=0)
        }
    }
)

celery = make_celery(flask_app)
```

## APPENDIX E

Celery tasks to run the PROT-ON in the background.

```
@celery.task
def
proton(chain_1,chain_2,selected_chain,cut_off,iqr,algorithm,protein_structure,pssm,run
_id,email,name):
    os.chdir(run_id)
    shutil.copy(protein_structure, "..")
    os.remove(protein_structure)
    if pssm != "":
        shutil.copy(pssm, "..")
        os.remove(pssm)
    os.chdir("..")
```

```
    pdb = protein_structure[:-4]
    os.mkdir("{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm))
    main_IR(protein_structure,chain_1,chain_2,selected_chain,cut_off)
    if algorithm == "FoldX":
        main_FoldX(protein_structure,selected_chain)
    else:
        main_EvoEF(protein_structure,selected_chain,algorithm)
    main_DO(protein_structure,selected_chain,algorithm,pssm,cut_off,iqr)
    shutil.move(UPLOAD_FOLDER + protein_structure,
"{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm))
    shutil.move(UPLOAD_FOLDER + "{}_pairwise_distance_list".format(pdb),
"{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm))
    shutil.move(UPLOAD_FOLDER +
"{}_chain_{}_interface_aa_list".format(pdb,selected_chain),
"{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm))
    shutil.move(UPLOAD_FOLDER +
"{}_chain_{}_proton_scores".format(pdb,selected_chain),
"{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm))
    os.remove(UPLOAD_FOLDER + "heatmap_mutation_list")
    os.system("rm *.pdb")
    os.chdir(UPLOAD_FOLDER +
"{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm))
    os.system("tar cvzf {}_chain_{}_{}_output.tar.gz
*".format(pdb,selected_chain,algorithm))
    os.chdir("../")
    shutil.move(UPLOAD_FOLDER +
"{}_chain_{}_{}_output".format(pdb,selected_chain,algorithm), run_id)
    shutil.move(UPLOAD_FOLDER + run_id, "run_results")
    os.system("chown username: username
run_results/{}/{}_chain_{}_{}_output".format(run_id,pdb,selected_chain,algorithm))
    os.system("chown username:username run_results/{}".format(run_id))
    os.system("chown username: username
run_results/{}/{}_chain_{}_{}_output/*".format(run_id,pdb,selected_chain,algorithm))
    SendMail(email,name,run_id)

@celery.task(name ="periodic_task")
def periodic_task():
    path = "run_results/"
    now = time.time()

    for filename in os.listdir(path):
        filestamp = os.stat(os.path.join(path, filename)).st_mtime
        filecompare = now - 7 * 86400
```

```
        if  filestamp < filecompare:
            os.system("rm -rf {}".format(path+filename))

    logger.info("Periodic run deletion initiated!")
```

**APPENDIX F**

Configuration settings of supervisor file creating for Gunicorn process. /PROT-ON/path/here illustrates the direction of the PROT-ON folder.

```
[program:PROT-ON]
directory=/PROT-ON/path/here
command=/PROT-ON/path/here/env/bin/gunicorn app:app -b localhost:8000
autostart=true
autorestart=true
stderr_logfile=/var/log/PROT-ON/hello_world.err.log
stdout_logfile=/var/log/PROT-ON/hello_world.out.log
```

**APPENDIX G**

Configuration settings of Nginx server for PROT-ON.

```
server {
    listen       8001;
    server_name  workstation_ip_address; proton.tools.ibg.edu.tr;

    location / {
        proxy_pass http://127.0.0.1:8000;
    }
}
```

**APPENDIX H**

Plotly box whisker plot drawing codes. Normally, Python Plotly does not have a parameter to identify whisker length according to IQR range. Below function draw a box plot user interest IQR range.

```
DDG_Scores = list(self.Scores_File["DDG_{}_Scores".format(self.algorithm)])
```

```
        q1 = np.quantile(DDG_Scores,0.25)
        q3 = np.quantile(DDG_Scores, 0.75)
        IQR = float(q3 - q1)
        upperfence = q3 + (self.iqr * IQR)
        lowerfence = q1 - (self.iqr * IQR)
        for i in DDG_Scores:
            if i < lowerfence:
                negative_outlier = q1 - (self.iqr * IQR)
                break
            else:
                negative_outlier = min(DDG_Scores)

        for i in DDG_Scores:
            if i > upperfence:
                positive_outlier = (q3 + self.iqr * IQR)
                break
            else:
                positive_outlier = max(DDG_Scores)
        boxplot = go.Figure()

boxplot.add_trace(go.Box(y=[DDG_Scores],boxpoints="outliers",hovertemplate='<b>Δ
ΔG Score</b>: %<extra></extra>'))
        boxplot.update_traces(q1=[np.quantile(DDG_Scores,0.25)],
            median=[np.quantile(DDG_Scores,0.50)],
            q3=[np.quantile(DDG_Scores,0.75)],
            lowerfence=[negative_outlier],
            upperfence=[positive_outlier])
        boxplot.update_layout(title="<b>Distribution of ΔΔG {}
Scores</b>".format(self.algorithm),
            title_x=0.5,
            yaxis={"title": '<b>ΔΔG {} Scores</b>'.format(self.algorithm)},
            template="plotly_white",
            font=dict(family='Times New Roman', size=20, color='black'))
        boxplot.update_yaxes(tickprefix="<b>",ticksuffix ="</b><br>")
```

## **APPENDIX I**

Outlier list for p53 protein of MDM2-p53 complex. ΔΔG scores are estimated by EvoEF1. Rows that are highlighted with red denote the depleting mutations filtering by stability score (Stabilizing depleting mutations).

**Table 6.** Suggested depleting mutations of p53 by EvoEF1

| Positions | Mutations | ΔΔG Binding | ΔΔG Stability |
|---|---|---|---|
| F19 | A | 4.7 | 0.88 |
| F19 | C | 4.23 | 0.16 |
| F19 | D | 5.34 | 0.87 |
| F19 | E | 4.45 | 0.44 |
| F19 | G | 5.4 | 1.51 |
| F19 | K | 4.07 | 0.53 |
| F19 | N | 4.44 | 0.31 |
| F19 | P | 5.28 | 0.2 |
| F19 | Q | 4.56 | 0.75 |
| F19 | S | 4.98 | 0.86 |
| F19 | T | 4.63 | 0.45 |
| W23 | A | 4.48 | 0.52 |
| W23 | C | 4.35 | -0.48 |
| W23 | D | 4.93 | 0.51 |
| W23 | E | 4.49 | 0.05 |
| W23 | G | 4.68 | 1.55 |
| W23 | H | 4.14 | 0.56 |
| W23 | K | 4.16 | -0.08 |
| W23 | N | 4.77 | 0.24 |
| W23 | P | 4.24 | 0.95 |
| W23 | Q | 3.9 | 0.35 |
| W23 | R | 3.93 | -0.45 |
| W23 | S | 4.61 | 0.51 |
| W23 | T | 4.42 | -0.11 |
| W23 | V | 4.05 | -0.98 |

## APPENDIX J

Outlier list for p53 protein of MDM2-p53 complex. ΔΔG scores are estimated by FoldX. Rows that are highlighted with red denote the depleting mutations filtering by stability score (Stabilizing depleting mutations).

**Table 7.** Suggested depleting mutations of p53 by FoldX

| Positions | Mutations | ΔΔG Binding | ΔΔG Stability |
|---|---|---|---|
| F19 | D | 6.03 | -0.36 |
| F19 | E | 5.12 | -0.33 |
| F19 | G | 5.49 | 1.43 |
| F19 | R | 7.48 | 0.90 |
| F19 | W | 8.51 | 0.13 |
| W23 | A | 5.4 | 0.61 |
| W23 | C | 5.43 | 0.91 |
| W23 | D | 5.73 | 1.04 |
| W23 | E | 5.84 | 0.53 |
| W23 | G | 5.75 | 1.87 |
| W23 | K | 6.14 | 0.26 |
| W23 | N | 5.57 | 1.19 |
| W23 | P | 7.32 | 4.95 |
| W23 | Q | 5.61 | 0.65 |
| W23 | R | 6.01 | 0.47 |
| W23 | S | 5.63 | 1.37 |
| W23 | T | 5.74 | 0.99 |
| L26 | F | 5.13 | 0.04 |
| L26 | R | 7.21 | 0.75 |
| L26 | W | 10.64 | 0.73 |
| L26 | Y | 11.41 | 0.12 |

## APPENDIX K

Outlier list for AXL protein of AXL-GAS6 complex. ΔΔG scores are estimated by EvoEF1. Rows that are highlighted with red denote the depleting mutations filtering by both stability and PSSM scores.

**Table 8.** Suggested depleting mutations of AXL by EvoEF1

| Positions | Mutations | ΔΔG Binding | ΔΔG Stability | Cut-off Distances | IQR Range | ΔΔ PSSM |
|---|---|---|---|---|---|---|
| R48 | A | 3.08 | 0.34 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| R48 | C | 3.08 | -0.53 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -6 |
| R48 | D | 3.1 | 0.52 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | E | 3.08 | 0.13 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | F | 3.08 | 0.64 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |
| R48 | G | 3.08 | 1.07 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |
| R48 | H | 3.08 | 0.78 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | I | 2.99 | -1.24 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |
| R48 | K | 2.54 | -0.06 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | 0 |
| R48 | L | 3 | -1.19 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |
| R48 | M | 2.69 | -0.7 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -4 |
| R48 | N | 3.1 | 0.17 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | 0 |
| R48 | P | 3.08 | -0.3 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | Q | 2.15 | 0.41 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | S | 3.08 | 0.37 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | T | 3.1 | 0.16 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| R48 | V | 3.1 | -0.34 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| R48 | W | 2.78 | 0.97 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -6 |
| R48 | Y | 3.05 | -0.55 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| E56 | A | 1.76 | -0.08 | 7.0 | 2.5 | -3 |
| E56 | C | 1.78 | -0.81 | 7.0 | 2.5 | -5 |
| E56 | F | 1.74 | -0.46 | 7.0 | 2.5 | -4 |
| E56 | G | 1.76 | 0.71 | 7.0 | 2.5 | -4 |

| | | | | | | |
|---|---|---|---|---|---|---|
| E56 | H | 1.89 | -0.02 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | 0 |
| E56 | I | 1.79 | -1.67 | 7.0 | 2.5 | -1 |
| E56 | K | 1.76 | -0.67 | 7.0 | 2.5 | -2 |
| E56 | L | 1.8 | -1.5 | 7.0 | 2.5 | -4 |
| E56 | M | 1.71 | -0.94 | 7.0 | 2.5 | -4 |
| E56 | N | 1.86 | -0.62 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | >0 |
| E56 | P | 1.76 | -0.36 | 7.0 | 2.5 | >0 |
| E56 | R | 1.8 | -1.73 | 7.0 | 2.5 | -3 |
| E56 | S | 1.76 | -0.03 | 7.0 | 2.5 | 0 |
| E56 | T | 1.76 | -0.46 | 7.0 | 2.5 | -1 |
| E56 | V | 1.79 | -1.18 | 7.0 | 2.5 | -1 |
| E56 | W | 1.86 | 0.05 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| E56 | Y | 1.76 | -0.65 | 7.0 | 2.5 | -2 |
| E59 | A | 1.71 | 0.24 | 7.0 | 2.5 | -4 |
| E59 | G | 1.69 | 0.94 | 7.0 | 2.5 | -5 |
| E59 | I | 1.7 | -1.2 | 7.0 | 2.5 | -4 |
| E59 | K | 1.94 | 0.04 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -1 |
| E59 | L | 1.79 | -1.11 | 7.0 | 2.5 | -5 |
| E59 | P | 1.75 | -0.4 | 7.0 | 2.5 | -6 |
| E59 | R | 1.78 | -0.54 | 7.0 | 2.5 | -2 |
| E59 | V | 1.77 | -0.74 | 7.0 | 2.5 | -3 |
| E59 | W | 1.84 | 1.01 | 6.0, 7.0 | 2.0, 2.5 | -6 |
| E59 | Y | 1.74 | 0.09 | 7.0 | 2.5 | -5 |
| S74 | W | 4.35 | 0.98 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |
| E85 | H | 1.74 | 0.54 | 7.0 | 2.5 | 0 |

## **APPENDIX I**

Outlier list for AXL protein of AXL-GAS6 complex. ΔΔG scores are estimated by EvoEF1. Rows that are highlighted with blue denote the enriching mutations filtering by both stability and PSSM scores.

**Table 9.** Suggested enriching mutations of AXL by EvoEF1.

| Positions | Mutations | ΔΔG Binding | ΔΔG Stability | Cut-off Distances | IQR Range | ΔΔ PSSM |
|---|---|---|---|---|---|---|
| D73 | C | -1.57 | -0.54 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | E | -1.89 | -0.42 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | H | -1.52 | 0.91 | 6.0, 7.0 | 2.0, 2.5 | <0 |
| D73 | I | -2.62 | -1.26 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | K | -1.69 | -0.12 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | L | -2.54 | -1.16 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | M | -3.36 | -0.55 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | N | -1.7 | -0.13 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | P | -2.28 | -0.33 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | Q | -1.73 | 0.04 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | R | -2.29 | -0.62 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | S | -1.67 | 0.13 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | T | -1.96 | -0.14 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | V | -2.22 | -0.83 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| D73 | Y | -2.28 | -0.44 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| T75 | E | -1.54 | 0.36 | 6.0, 7.0 | 2.0, 2.5 | <0 |
| T75 | W | -1.5 | 0.91 | 7.0 | 2.5 | <0 |
| T77 | F | -2.24 | -0.79 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| T77 | I | -1.54 | -1.38 | 6.0, 7.0 | 2.0, 2.5 | 0 |
| T77 | L | -2.2 | -1.23 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| T77 | M | -2.14 | -0.49 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| T77 | Y | -1.52 | -0.63 | 6.0, 7.0 | 2.0, 2.5 | <0 |
| Q78 | F | -2.07 | -1.41 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| V79 | F | -1.56 | 0.5 | 6.0, 7.0 | 2.0, 2.5 | 0 |
| E83 | F | -2.19 | 0.12 | 6.0, 7.0 | 2.0, 2.5 | <0 |
| E83 | H | -1.66 | 0.75 | 6.0, 7.0 | 2.0, 2.5 | <0 |
| T98 | R | -1.63 | -0.76 | 7.0 | 2.5 | <0 |
| T98 | W | -1.8 | 0.81 | 7.0 | 2.5 | 0 |

## APPENDIX M

Outlier list for AXL protein of AXL-GAS6 complex. ΔΔG scores are estimated by Foldx. Rows that are highlighted with red denote the depleting mutations filtering by both stability and PSSM scores.

**Table 10.** Suggested depleting mutations of AXL by FoldX

| Positions | Mutations | ΔΔG Binding | ΔΔG Stability | Cut-off Distances | IQR Range | ΔΔ PSSM |
|---|---|---|---|---|---|---|
| P57 | H | 5.91 | 11.74 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -11 |
| P57 | K | 2.98 | 4.97 | 5.0, 6.0 | 1.5, 2.0 | -9 |
| P57 | R | 3.61 | 4.74 | 5.0,6.0, 7.0 | 1.5,2.0, 2.5 | -10 |
| P57 | W | 6.47 | 4.82 | 5.0,6.0, 7.0 | 1.5,2.0, 2.5 | -15 |
| D73 | F | 3.1 | -0.32 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -7 |
| D73 | H | 3.92 | 0.26 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -4 |
| D73 | Y | 3.01 | -0.54 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -7 |
| T75 | F | 4.7 | -0.82 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| T75 | Y | 2.72 | -1.12 | 5.0 | 1.5 | -1 |
| Q78 | P | 3.94 | 0.58 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| Q78 | R | 3.26 | -0.23 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | 0 |
| V79 | G | 2.93 | 2.43 | 5.0, 6.0 | 1.5, 2.0 | -2 |
| V79 | H | 3.62 | 0.1 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -1 |
| V79 | Y | 3.52 | -0.76 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| P80 | H | 3.21 | 1.14 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| P80 | K | 4.89 | 0.78 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -3 |
| P80 | R | 8.07 | 1.02 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -2 |
| P80 | Y | 2.81 | 1.58 | 5.0 | 1.5 | -4 |
| P80 | W | 3.31 | 1.58 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | -4 |
| V91 | R | 3.03 | 2.16 | 7.0 | 2.5 | 0 |

## APPENDIX N

Outlier list for AXL protein of AXL-GAS6 complex. ΔΔG scores are estimated by Foldx. Rows that are highlighted with blue denote the enriching mutations filtering by both stability and PSSM scores.

**Table 11.** Suggested enriching mutations of AXL by FoldX

| Positions | Mutations | ΔΔG Binding | ΔΔG Stability | Cut-off Distances | IQR Range | ΔΔ PSSM |
|---|---|---|---|---|---|---|
| Q76 | E | -1.91 | -0.21 | 5.0 | 1.5 | <0 |
| Q76 | F | -1.74 | 0.57 | 5.0 | 1.5 | <0 |
| Q76 | Y | -2.07 | 0.33 | 5.0, 6.0 | 1.5, 2.0 | <0 |
| T77 | I | -1.57 | -1.2 | 5.0 | 1.5 | 0 |
| T77 | K | -2.23 | -0.68 | 5.0, 6.0, 7.0 | 1.5, 2.0, 2.5 | <0 |
| T77 | L | -2.06 | -0.98 | 5.0, 6.0 | 1.5, 2.0 | <0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| T77 | M | -1.56 | -1.09 | 5.0 | 1.5 | <0 |
| T77 | W | -2.01 | -0.69 | 5.0, 6.0 | 1.5, 2.0 | <0 |

## APPENDIX O

The number of outlier mutations of EvoEF1 and FoldX scores for 2.0 and 2.5 IQR ranges.

**Table 12.** Outlier mutation size by 2.0 and 2.5 IQR ranges.

| Algorithm | 6.0Å, 2.0 IQR - | 7.0Å, 2.5 IQR |
|---|---|---|
| EvoEF1 (Depleting) | 25 | 48 |
| EvoEF1 (Enriching) | 20 | 28 |
| FoldX (Depleting) | 17 | 16 |
| FoldX (Enriching) | 4 | 1 |

**İZMİR BİYOTIP VE GENOM MERKEZİ**
**GİRİŞİMSEL OLMAYAN ARAŞTIRMALAR ETİK KURULU (İBG-GOEK)**
**KARARI**

| | | | |
|---|---|---|---|
| **Toplantı Tarihi** | : 02/06/2022 | **Toplantı Günü** | : Perşembe |
| **Toplantı Sayısı** | : 8 | **Toplantı Saati** | : 10:30 |

**Sayın Dr. Ögr. Üyesi Ezgi KARACA,**

**2021-007 Protokol No'lu;** sorumlusu olduğunuz "Protein-Protein Arayüzlerini Yeni Yaklaşım PROT-ON Algoritması Ile Yeniden Tasarlanması" başlıklı araştırmanın uygulanmasında etik açıdan sakınca olmadığına oy birliği ile karar verilmiştir.

Bilgilerinizi ve gereğini rica ederiz.

| | |
|---|---|
| **Prof. Dr. H. Alper BAGRIYANIK**<br>**Başkan** | **Dr. Serap ERKEK ÖZHAN**<br>**Başkan Yardımcısı** |
| **Prof. Dr. Sedef AKGÜNGÖR**<br>**Üye** | **Prof. Dr. İnci ALBACAĞIOĞLU**<br>**Üye** |
| **Prof. Dr. Gülgün OKTAY**<br>**Üye** | **Prof. Dr. Hilmi ORHAN**<br>**Üye** |
| **Prof. Dr. Şermin GENÇ**<br>**Üye** | **Doç. Dr. Canan Aslı YILDIRIM**<br>**Üye** (Katılmadı) |
| **Doç. Dr. Can KÜÇÜK**<br>**Üye** (Katılmadı) | **Dr. Öğr. Üyesi Yavuz OKTAY**<br>**Üye** (Katılmadı) |

# Mehdi KOŞACA

## Personal Information

### General Information

**Date of births:**

**Telephone:** **(5**

**E-mail:**

**Education Information**

04 February 2020 – Now, Yüksek Lisans Tezli Programı, DOKUZ EYLÜL ÜNİVERSİTESİ, TÜRKİYE, İZMİR ULUSLARARASI BİYOTIP VE GENOM ENSTİTÜSÜ, GENOM BİLİMLERİ VE MOLEKÜLER BİYOTEKNOLOJİ ANABİLİM DALI, Cumulative Graduate Point Average: 3.57/4.00

1 September 2014 – 06 January 2020, Lisans Anadal/Normal Öğretim, ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ, TÜRKİYE, FEN EDEBİYAT FAKÜLTESİ, MOLEKÜLER BİYOLOJİ VE GENETİK BÖLÜMÜ, Cumulative Graduate Point Average: 3.03/4.00

**Experience Information**

11 May 2020 – 27 July 2022 ARAŞTIRMA ASİSTANI, İZMİR BİYOTIP VE GENOM MERKEZİ

**Foreing Language Information**

İNGİLİZCE (Reading: İyi, Writing: İyi, Speaking: İyi)

## Scientific Technological Activity Fields

### Keywords

Biyoinformatik
Hesaplamalı Biyoloji
Yapısal Biyoenformatik
Protein – Protein Etkileşimleri

## R&D Competency

### Conference Papers

M. KOŞACA T. BATUR, M. ÖZTÜRK, E. KARACA, PROT-ON: Structure-Based Detection of Critical Mutations in Redesigning Protein-Protein Interfaces, Sözlü Sunum, 8th International Congress of the Molecular Biology Association of Turkey, 9 June 2022, 12 June 2022

M. KOŞACA, B. BARLAS, E. KARACA, PROT-ON: A Python Package for Redesigning the Protein-Protein Interfaces Using EvoEF1, Poster Sunumu The 14th International Symposium on Health Informatics and Bioinformatics, 10 September 2021, 11 September 2021

M. KOŞACA, E. ŞAMİLOĞLU, M. ERGÜVEN, E. KARACA, Optimization of the HADDOCK Sampling for the Rapid Modeling of Interfacial Mutations, Poster Sunumu, The 13th International Symposium on Health Informatics and Bioinformatics, 22 October 2020, 23 October 2020

### Academic Paper

KOŞACA, M., ŞAMİLOĞLU, E., & KARACA, E. (2021). Arayüz Mutasyonlarının Protein Etkileşimlerine Tesirini Tahmin Eden Algoritmalarla HADDOCK'un

Performansının
Karşılaştırılması. International Journal of Advances in Engineering and Pure Sciences, 33(4), 592-608.

Samiloglu, Eda, Mehdi Kosaca, Melis Oktayoglu, Can Yukruk, Mehmet Erguven, Nazmiye Arslan, Gokhan Karakulah et al. "Benchmarking the Widely Used Structure-based Binding Affinity Predictors on the Spike-ACE2 Deep Mutational Interaction
Set." bioRxiv (2022)