## PHARMACOINFORMATICS: PROTEIN DOMAIN INTERACTIONS AND COMPARISON BETWEEN WILD TYPE AND ZINC ASSOCIATED ONE

[Pharmacoinformatics is the use of information system technologies to improve the quality and efficiency of pharmaceutical care processes and outcomes](https://www.cellsignal.com/learn-and-support/protein-domains-and-interactions)[One of the applications of pharmacoinformatics is to study the structure, function, and interactions of proteins involved in drug metabolism, transport, and action](https://en.wikipedia.org/wiki/Protein%E2%80%93protein_interaction)[Protein domain interactions are the physical contacts between specific regions of proteins that mediate their biological functions](https://www.eurekaselect.com/article/103637)[Understanding how protein domains interact with each other and with other molecules can help to design better drugs, identify potential targets, and predict drug responses](https://www.ebi.ac.uk/training/online/courses/protein-classification-intro-ebi-resources/protein-classification/what-are-protein-domains/)

[One of the proteins that has been extensively studied in pharmacoinformatics is poly(ADP-ribose) polymerase 1 (PARP1), a nuclear enzyme that participates in DNA repair, transcription, chromatin remodeling, and cell death PARP1 is composed of several domains, including a DNA-binding domain, a caspase-cleaved domain, an auto-modification domain, and a catalytic domain](https://en.wikipedia.org/wiki/Poly_%28ADP-ribose%29_polymerase)[PARP1 is activated by binding to DNA breaks and catalyzes the transfer of ADP-ribose units from NAD+ to itself and other acceptor proteins, forming poly(ADP-ribose) chains This process regulates the recruitment and activity of other proteins involved in DNA repair and signaling](https://www.cellsignal.com/learn-and-support/protein-domains-and-interactions)

[PARP1 is also a target of several drugs, especially for cancer therapy](https://en.wikipedia.org/wiki/PARP1)[PARP inhibitors are small molecules that bind to the catalytic domain of PARP1 and prevent its enzymatic activity PARP inhibitors have shown clinical efficacy in various types of cancers, such as breast, ovarian, prostate, and pancreatic cancers](https://www.cancerresearchuk.org/about-cancer/treatment/targeted-cancer-drugs/types/PARP-inhibitors)[However, the molecular mechanisms of PARP inhibitor resistance and sensitivity are still not fully understood and require further investigation](https://www.cellsignal.com/learn-and-support/protein-domains-and-interactions)

In this project, we aim to compare the protein domain interactions of PARP1 in its wild type and zinc associated forms. [Zinc is a metal ion that can bind to PARP1 and modulate its activity and interactions](https://www.cellsignal.com/learn-and-support/protein-domains-and-interactions) We hypothesize that zinc binding affects the conformation and stability of PARP1 domains and alters their interactions with other proteins and DNA. To test this hypothesis, we will use the following data and methods:

### Its wild and ligand attached types

The wild type PARP1 is the native form of the protein that does not have any modifications or ligands attached to it. The ligand attached PARP1 is the form of the protein that has a ligand, such as NAD+, poly(ADP-ribose), or a PARP inhibitor, bound to its catalytic domain. The ligand attached PARP1 can have different effects on the protein activity and interactions, depending on the type and affinity of the ligand. [For example, NAD+ is the substrate of PARP1 and activates its enzymatic function, while poly(ADP-ribose) and PARP inhibitors are inhibitors of PARP1 and block its enzymatic function](https://www.cellsignal.com/learn-and-support/protein-domains-and-interactions)

### The avg files

The avg files are text files that contain the average distance and angle between the acceptor and donor atoms of hydrogen bonds formed between PARP1 and its ligands. The files have the following columns:

* Acceptor: the atom name and residue number of the acceptor atom in PARP1
* DonorH: the atom name and residue number of the hydrogen atom in the ligand
* Donor: the atom name and residue number of the donor atom in the ligand
* Frames: the number of frames in which the hydrogen bond was observed
* Frac: the fraction of frames in which the hydrogen bond was observed
* AvgDist: the average distance between the acceptor and donor atoms in angstroms
* AvgAng: the average angle between the acceptor, hydrogen, and donor atoms in degrees

The avg files can be used to compare the hydrogen bonding patterns and strengths between PARP1 and its ligands in different conditions. For example, the file parp-dna-zn-nad\_All contains the hydrogen bonds between PARP1 and DNA, zinc, and NAD+, while the file PARP-DNA\_All contains the hydrogen bonds between PARP1 and DNA only. By comparing these two files, we can see how the presence of zinc and NAD+ affects the hydrogen bonding network of PARP1 and DNA.

### The intercaat residues of all domain interactions

## The intercaat residues are the amino acid residues of PARP1 that are involved in the interactions between different domains of the protein. The intercaat residues can be used to understand the functional significance of each domain of PARP1 and how they cooperate or compete with each other in different cellular contexts. [For example, the zinc finger domains are responsible for sensing and binding to DNA damage, the WGR domain is involved in protein-protein interactions and allosteric regulation, the helical domain is essential for catalytic activity and inhibitor binding, and the auto-modification domain is the target of PARylation and modulates the enzyme turnover](https://en.wikipedia.org/wiki/PARP1)[Zinc binding and ligand attachment can affect the conformation, stability, and affinity of these domains and alter their interactions with DNA and other proteins](https://www.genecards.org/cgi-bin/carddisp.pl?gene=PARP1) Therefore, studying the intercaat residues can provide insights into the molecular mechanisms of PARP1 regulation and its role in various biological processes.

The domain ranges can be used to map the intercaat residues and the avg files to the corresponding domains and to visualize the domain structures and interactions of PARP1.

### Step 0: File Preparation

In the initial phase of our pharmacoinformatics analysis, we focus on preparing the necessary files for subsequent investigations. The primary objectives of this step are to convert the provided ".avg" files into more accessible Excel (.xlsx) formats, and subsequently refine these files for ease of handling and analysis.

1. Conversion of .avg Files to Excel (.xlsx): We initiate the process by defining a function, avg\_to\_xlsx, designed to convert the raw ".avg" files into Excel files. These files, namely parp-dna-zn-nad\_All.UU.avg.dat and PARP-DNA\_All.UU.avg.dat, represent the wild type and zinc-associated protein domain interactions, respectively. The data is organized in columns such as Acceptor, DonorH, Donor, Frames, Frac, AvgDist, and AvgAng.

We employ the Pandas library to read the ".avg" files, remove any '#' characters from the header, and then write the cleaned data into corresponding Excel files

2. Refining Excel Files: Subsequently, we recognize the need to refine the Excel files for improved readability and analysis. To achieve this, we define a function, remove\_spaces\_add\_underscore, which addresses two key aspects: removing trailing whitespaces and introducing underscores between residue names and their positions. This refinement aims to enhance the consistency and uniformity of data representation.

The function reads the Excel file, iterates through columns and rows to remove unnecessary spaces, and replaces spaces with underscores in residue entries from the second row onwards. The modified DataFrame is then saved back to the Excel file.

By executing these file preparation steps, we establish a solid foundation for the subsequent phases of our pharmacoinformatics analysis, ensuring data clarity and compatibility for further exploration and comparisons between wild-type and zinc-associated protein domain interactions.

### Step 1: Filtration Based on Domain Ranges and Interacting Residues

In the second stage of our pharmacoinformatics analysis, we concentrate on refining the molecular dynamics (MD) simulation data by filtering it according to specified domain ranges and interacting residues. This step aims to extract relevant information that corresponds to the predefined domain interactions.

1. User Input: The user is prompted to provide the file paths or names for three essential input files in the Excel (.xlsx) format:

Residue File: Contains information about interacting residues in different domain interactions.

Domain Range File: Specifies the starting and ending positions of various protein domains.

MD Simulation Cleaned File: Represents the cleaned data obtained from MD simulations.

2. Data Loading: The Pandas library is utilized to read the contents of the provided Excel files into DataFrames, namely residues\_df, domain\_ranges\_df, and md\_simulation\_df.

3. Domain Ranges Dictionary Construction: A dictionary named domain\_ranges is created to store the start and end positions of various protein domains from the domain\_ranges\_df.

4. Filtering Function: The filter\_md\_simulation\_data\_corrected function is defined to perform the filtration process. It iterates through residues and their interactions, checks their validity, and filters the MD simulation data accordingly.

5. Output: The filtered data is saved to a new Excel file, incorporating the original file name with the addition of "\_filtered.xlsx". A confirmation message is displayed.

This step ensures that the subsequent analyses are conducted on a refined dataset, focusing specifically on the interactions within the specified domain ranges and involving the relevant residues identified in the previous step.

### Step 2: Filtration Based on a Threshold Condition for Significant Interactions

In the third phase of our pharmacoinformatics analysis, we introduce an additional filtration step to focus on significant interactions within the previously filtered data. This step involves applying a threshold condition specifically to the 'Frac' column, retaining interactions with a Frac value greater than or equal to 0.2.

1. Threshold Filtration Function Definition: We define the function filter\_significant\_interactions, which takes the path of the input Excel file and a threshold value as parameters. This function reads the previously filtered data, applies the specified threshold condition to the 'Frac' column, and saves the resulting significant interactions to a new Excel file.

2. User Input and Function Execution: Users are prompted to input the path/name of the previously filtered data file in the Excel (.xlsx) format and to provide a threshold value in the range of 0 to 1. The function filter\_significant\_interactions is then called to execute the filtration based on the specified threshold.

This step refines our dataset further by isolating interactions with a Frac value greater than or equal to 0.2, ensuring that subsequent analyses focus on interactions with potentially greater significance in the molecular dynamics simulations.

### Step 3: Pairwise Filtration

In the third segment of our pharmacoinformatics analysis, we introduce a pairwise filtration process to further refine the data by grouping interactions based on acceptor and donor residues within each domain interaction. The objective is to simplify the representation of interactions for downstream analyses, focusing on specific residue pairs.

1. Pairwise Domain Interaction Processing Function: We define the function process\_pairwise\_domain\_interaction to handle the processing of a specific domain interaction within the DataFrame. This function drops unnecessary columns, modifies residue names, groups data by acceptor and donor residues, updates frames, and calculates the fraction of frames in which the interaction occurs.

2. Pairwise File Processing Function: We define the function process\_file\_pairwise to process the entire input file, handling multiple domain interactions. This function iterates through rows, identifies domain interactions, and applies the pairwise processing function to each interaction. The results are concatenated to create a comprehensive DataFrame, which is then written to an output file.

3. Logic for Pairwise Filtration:

The function iterates through rows of the DataFrame, identifying the start of each domain interaction based on the 'Domain Interaction' column.

For each domain interaction, the process\_pairwise\_domain\_interaction function is called, handling the grouping and calculation logic for acceptor-donor residue pairs.

The results are stored in a list of DataFrames for each domain interaction.

After processing all domain interactions, the individual DataFrames are concatenated into a final result DataFrame.

The final result is then saved to an output file, following the naming convention for pairwise filtration.

This pairwise filtration step simplifies the dataset, making it more amenable to pairwise interaction analysis and subsequent exploration of specific residue pairs within each domain interaction.

### Step 4: Singular Filtration

In the fourth stage of our pharmacoinformatics analysis, we introduce the singular filtration process. This step focuses on identifying and processing individual residues within each domain interaction, providing detailed information about the contribution of each residue to the interaction dynamics.

1. Singular Domain Interaction Processing Function: We define the function process\_singular\_domain\_interaction to handle the processing of a specific domain interaction within the DataFrame. This function drops unnecessary columns, modifies residue names, extracts unique residues from both 'Acceptor' and 'Donor' columns, and calculates the frames and fraction of frames for each residue within the domain interaction.

2. Singular File Processing Function: We define the function process\_file\_singular to process the entire input file, handling multiple domain interactions. This function iterates through rows, identifies domain interactions, and applies the singular processing function to each interaction. The results are stored in a list of DataFrames for each domain interaction, which are then concatenated into a final result DataFrame and saved to an output file.

3. Logic for Singular Filtration:

The function iterates through rows of the DataFrame, identifying the start of each domain interaction based on the 'Domain Interaction' column.

For each domain interaction, the process\_singular\_domain\_interaction function is called, handling the extraction of unique residues, frames calculation, and fraction of frames calculation for each residue within the domain interaction.

The results are stored in a list of DataFrames for each domain interaction.

After processing all domain interactions, the individual DataFrames are concatenated into a final result DataFrame.

The final result is then saved to an output file, following the naming convention for singular filtration.

### Step 5: Pairwise Filtration (Revised)

In the fifth step of our pharmacoinformatics analysis, we revisit the pairwise filtration process. This step refines the data by grouping interactions based on acceptor and donor residues within each domain interaction, employing a modified approach compared to the previous pairwise filtration in Step 3. The goal remains to simplify the representation of interactions for downstream analyses, focusing on specific residue pairs.

1. Revised Pairwise Domain Interaction Processing Function: We retain the function process\_pairwise\_domain\_interaction, which is slightly modified to drop unnecessary columns, handle NaN values in residue names, group data by acceptor and donor residues, update frames, and calculate the fraction of frames in which the interaction occurs.

2. Revised Pairwise File Processing Function: The function process\_file\_pairwise is called for pairwise filtration. It reads the input file, iterates through rows, identifies domain interactions, and applies the revised pairwise processing function to each interaction. The results are stored in a list of DataFrames for each domain interaction, which are then concatenated into a final result DataFrame and saved to an output file.

3. Logic for Revised Pairwise Filtration:

The function iterates through rows of the DataFrame, identifying the start of each domain interaction based on the 'Domain Interaction' column.

For each domain interaction, the process\_pairwise\_domain\_interaction function is called, handling the grouping and calculation logic for acceptor-donor residue pairs.

The results are stored in a list of DataFrames for each domain interaction.

After processing all domain interactions, the individual DataFrames are concatenated into a final result DataFrame.

The final result is then saved to an output file, following the naming convention for revised pairwise filtration.

This revised pairwise filtration step maintains the goal of simplifying the dataset for a more focused analysis, employing an alternative approach to grouping interactions based on acceptor and donor residues.

### Step 6: Singular Filtration

In the sixth step of our pharmacoinformatics analysis, the focus is on singular filtration. This step aims to further simplify the dataset by considering each residue individually within the context of domain interactions. The process involves aggregating information for each unique residue, irrespective of domain interactions, providing a singular representation for each residue.

1. Singular Domain Interaction Processing Function:

The function process\_singular\_domain\_interaction is responsible for handling a specific domain interaction within the DataFrame.

Unnecessary columns (DonorH, AvgDist, AvgAng) are dropped, and residue names are modified, handling NaN values.

Unique residues are extracted from both the Acceptor and Donor columns.

A DataFrame is initialized to store results for each residue, including domain interaction, residue, frames, and fraction information.

The function processes each unique residue, calculating the sum of frames and fraction for all occurrences of the residue within the domain interaction.

Results are stored in the DataFrame.

2. Singular File Processing Function:

The function process\_file\_singular is called for singular filtration, reading the input file and iterating through rows.

For each domain interaction, the process\_singular\_domain\_interaction function is applied to aggregate information for each unique residue.

Results for each domain interaction are stored in a list of DataFrames.

After processing all domain interactions, the individual DataFrames are concatenated into a final result DataFrame.

The final result is then saved to an output file, following the naming convention for singular filtration.

3. Logic for Singular Filtration:

The process iterates through rows of the DataFrame, identifying the start of each domain interaction based on the 'Domain Interaction' column.

For each domain interaction, the process\_singular\_domain\_interaction function is called, handling the aggregation logic for unique residues.

The results are stored in a list of DataFrames for each domain interaction.

After processing all domain interactions, the individual DataFrames are concatenated into a final result DataFrame.

The final result is then saved to an output file, following the naming convention for singular filtration

### Step 7: Pairwise Comparison

In the seventh step of our pharmacoinformatics analysis, the focus is on performing a pairwise comparison between two sets of data obtained from MD simulations. The goal is to compare interactions between domain pairs and evaluate the impact of zinc association on these interactions.

1. Function merge\_and\_compare\_files:

Reads data from two Excel files (file1\_path and file2\_path) into DataFrames (df1 and df2).

Merges the DataFrames on common columns ('Domain Interaction', 'Acceptor', 'Donor') using an outer join.

Creates a new column ('Comparison') with the greater of the two 'Frac' values for each row.

Fills NaN values in 'Frac' columns with 0.

Replaces NaN values in the 'Comparison' column with 0 if either 'Frac' value is 0.

Selects relevant columns for the output and saves the result to a new Excel file (output\_file\_path).

2. Logic for Pairwise Comparison:

The data from two MD simulation files are merged based on common columns using an outer join.

The 'Comparison' column is created, containing the greater of the 'Frac' values from the two datasets.

NaN values in 'Frac' columns are filled with 0 to facilitate comparison.

NaN values in the 'Comparison' column are replaced with 0 if either 'Frac' value is 0.

The output includes columns for 'Domain Interaction', 'Acceptor', 'Donor', 'Frac\_with\_zinc', 'Frac\_wildtype', and 'Comparison'.

The result is saved to a new Excel file.

The pairwise comparison provides insights into the impact of zinc association on domain interactions, highlighting changes in the fraction values between the wildtype and zinc-associated scenarios.

Step 8: Singular Composition

In the eighth and final step of our pharmacoinformatics analysis, the focus is on composing singular data from two sets obtained from MD simulations, enabling a detailed comparison of residue-level interactions.

1. Function compare\_and\_save:

Reads data from two Excel files (input\_file1 and input\_file2) into DataFrames (df1 and df2).

Merges the DataFrames on common columns ('Domain Interaction', 'Residue') using an outer join.

Fills NaN values with 0 for effective comparison.

Adds a new column ('comparison') that considers the greater of the 'Frac' values from the two datasets for each row.

Sets 0s in specific columns to NaN based on certain conditions to retain relevance.

Drops duplicates based on 'Domain Interaction' and 'Residue'.

Selects relevant columns for the output and saves the result to a new Excel file (output\_file).

2. Logic for Singular Composition:

Data from two MD simulation files are merged based on common columns using an outer join.

NaN values are filled with 0 for effective comparison.

A new column ('comparison') is added, considering the greater of the 'Frac' values from the two datasets for each row.

0s in specific columns are set to NaN based on conditions to ensure the 'Domain Interaction' row and column remain meaningful.

Duplicate rows are dropped based on 'Domain Interaction' and 'Residue'.

The output includes columns for 'Domain Interaction', 'Residue', 'Frac\_wildtype', 'Frac\_withZinc', and 'comparison'.

The result is saved to a new Excel file.

The singular composition allows for a detailed examination of residue-level interactions, providing insights into how zinc association influences individual residue behaviors within domain interactions.