

# A Biocomputational Platform for Template-based Protein-protein Docking

## Plataforma computacional de acoplamiento de proteínas basado en plantillas


Ricardo Román-Brenes<sup>1</sup>, Francisco Siles-Canales<sup>2</sup>,  
Daniel Zamora-Mata<sup>3</sup>

---


Román-Brenes, R; Siles-Canales, F; Zamora-Mata, D. A Biocomputational Platform for Template-based Protein-protein Docking. *Tecnología en Marcha*. Edición especial 2020. 6th Latin America High Performance Computing Conference (CARLA). Pág 96-100.

 <https://doi.org/10.18845/tm.v33i5.5084>


1 PRIS-Lab: Pattern Recognition and Intelligent Systems Laboratory School of Electrical Engineering, Faculty of Engineering, Universidad de Costa Rica (UCR). Costa Rica.  
E-mail: ricardo.roman@ucr.ac.cr.

 <https://orcid.org/0000-0002-6104-7561>

2 PRIS-Lab: Pattern Recognition and Intelligent Systems Laboratory School of Electrical Engineering, Faculty of Engineering, Universidad de Costa Rica. (UCR). Costa Rica.  
E-mail: francisco.siles@ucr.ac.cr.

 <https://orcid.org/0000-0002-6704-0600>

3 PRIS-Lab: Pattern Recognition and Intelligent Systems Laboratory School of Electrical Engineering, Faculty of Engineering, Universidad de Costa Rica (UCR). Costa Rica.  
E-mail: daniel.zamoramata@ucr.ac.cr.

 <https://orcid.org/0000-0001-8213-4974>



## Keywords

Clustering; protein-protein docking; template-based docking; HPC.

## Abstract

We propose the creation of a Biocomputational Platform for template-based protein-protein docking that aims reduce computational time by clustering data before the rigid body alignment. Using data from the Dockground project, models will be created using multiple clustering methods that will annotated each protein into a class, such that when performing the match search, not all of the databank needs to be inspected but just the class that resembles the most to the studied protein. This will reduce the time that conformation matching requires without incurring in lower precision.

## Palabras clave

Agrupamiento; acoplamiento de proteínas; Acoplamiento basado en plantillas; HPC.

## Resumen

Se propone la creación de una Plataforma Biocomputacional para el acoplamiento de proteínas que reduce el tiempo computacional al agrupar los datos previo al alineamiento de cuerpos rígidos. Utilizando datos del proyecto Dockground, se crearán modelos usando múltiples métodos de agrupación que asignarán cada proteína a un grupo para que, al realizar la búsqueda de pares, no se realice una búsqueda a fuerza bruta, sino una acotada. Esto reducirá el tiempo que requiere una conformación en ser procesada sin perder precisión.

## Introduction

Proteins are macromolecules that serve as catalyst for virtually all the cell's functions. In order to perform these functions, proteins need to interact with each other to form functional complexes [1]. Protein-protein docking (PPD) is an important area of study in molecular biology due to its importance in fields like drug discovery or precision medicine. This process can be done in silico or in vivo. The two general methods that exist for in silico PPD are de novo and template-based. The latter technique is called template-based protein docking (TBPD).

TBPD consists in finding out if two proteins form a complex by matching them by homology in a databank of complexes. If it is required to know if protein A and B can dock and form a functional complex, they would be iteratively aligned to all the proteins P in the databank. These alignments typically yield a score so that good matches can be selected. Finally, if protein A matches to a part of a complex and B matches to the other part of the complex, then it is said that A and B can dock [1].

In any case, the computational methods for docking proteins is very demanding on CPU resources [2] [3].

We propose here a computational platform with interchangeable modules to utilize a different set of descriptors, namely 3D geometrical descriptors [4] [5] to cluster, classify and validate TBPD in order to improve overall response time. The platform will use annotated data from the Dockground project [6] (full structure docking templates v1.1 databank) and run in the PRIS-Lab supercomputer TARÁ [7].

## Related Work

The use of clustering techniques for PPD is not a something new. There are numerous studies regarding this topic.

The Critical Assessment of Predicted Interactions (CAPRI) [8] is a community-wide experiment to measure the capabilities of protein-docking methods. In this experiment several teams try their algorithms to assess which is the best one for the task at hand. In 2005, most of the methods, 13 out of 20, used some sort of clustering technique in one point or another of their workflow [9].

By far the most common use for clustering is to discard decoys [10] [2] [11] [12] [13] [14] [15] [16]. Despite this extensive use of clustering, as far as this study goes, the importance of computational time has been relegated to a second plane of importance. We did not find any study that uses clustering to generate models prior to the mass comparison of each template candidate rigid body, in order to speed up the overall process.

## Methodology and Experiments

The platform will consist of several components. Items can be added to these components

- A set of paired template candidates  $TC = \{(tc00, tc01), (tc10, tc11), \dots, (tcn0, tcn1)\}$  that form a functional complex.
- A set of 3D geometrical feature extraction methods  $F = \{fe0, fe1, \dots, fen\}$  that will generate the features to be used in the clustering of the Dockground databank. Features like protein area, protein volume, protein circularity, 3D Zerkine descriptor for the protein (Daberdaku & Ferrari, 2018) and the 3 PCA components of the protein (en R3 solo hay 3 PCA components) will be used.
- A set of clustering methods  $C = \{c0, c1, \dots, cn\}$  that will form groups based on the 3D features. All the candidates TC will be clustered with each method in order to generate groups. Clustering methods like k-means, x-means, OPTICS and BIRCH will be used.
- A set of classification methods  $K = \{k0, k1, \dots, kn\}$  with which new proteins will be matched with subgroups in order to reduce run time. For starters the classifications methods to be used are SVM, k-nearest neighbors and LDA.

Each component of a set can be used interchangeably so that if for instance c0 does a better job than c3, the user or programmer can choose which one to use. A parallel and distributed implementation of the platform will be done so that multiple methods can be run simultaneously, since the process of each Fe, C, K, alignment and low-resolution docking by GRAMM don't have any kind of data dependency.

Each stage can be run by a pool of process or threads in TARÁ.

The Platform will have at least two modes of operation: databank clustering and docking prediction.

For the docking prediction, the normal workflow would be as follows

- Obtain the PDB files for the 2 proteins that want to be docked, p0 and p1.
- Choose one or more of the already clustered databanks generated with the features Fe extracted from TC using C methods.
- Using the groups generated in the previous step, choose one or more classification methods to find the best match for p0 and p1 in TC. The similarity of the match will be measured with TM-align (Zhang & Skolnick, 2005).

- The Platform shows, ordered by TM-align, the list of template candidates and the visualization of both the templates and p0 and p1.

For the databank clustering, the procedure would be

- Choose one of the clustering methods C.
- Select which databank to use, initially Dockground will be used. This databank will be split, first, in two sets for work and validation, and then the training set will be split in three, for training, testing and confirmation.
- Choose which 3D features from F will be used to perform the clustering.
- The Platform shows the results of the clustering and performs validation using a 10-fold cross-validation against the already annotated entries of the validation set from Dockground.
- If the user is satisfied with the results, the models will be saved as annotated data for classification, for future use in the docking prediction.

Aside from the automatization of the process of protein docking, run-time tests as well as a comparison of the docking solutions will be performed to ensure that the Platform behaves well against the current methods for TBPD, in particular using TM-align [17] and GRAMM-X [18]. The Platform's parallel implementation will ensure at least a higher throughput of data.

From an algorithmic point of view, this process can be seen as several time functions, all dependent on the size of the protein databank  $n$ .  $T(n)$  where  $n$  is the size of the protein databank. These functions can be seen in equationa 1, 2 and 3

$$T_0(n) = 2 \times \left( \sum_{i=1}^n lr(i) + \sum_{i=1}^n a(i) \right)$$

**Equation 1.** Time function of normal TBPD process, where  $lr(i)$  is the time taken to transform a PDB into a low-resolution representation and  $a(i)$  is the time taken to align a protein to another one in the databank.

$$T_1(n) = 2 \times \frac{\left( \sum_{i=1}^n lr(i) + \sum_{i=1}^n a(i) \right)}{P}$$

**Equation 2.** Time function of normal TBPD method proposed here, after the databank has been annotated with clustering. Here  $P$  is the size of the largest cluster,  $lr(i)$  is the time taken to transform a PDB into a low-resolution representation and  $a(i)$  is the time taken to align a protein to another one in the databank.

$$T_2(n) = \left( \sum_{i=1}^n fe(i) + \sum_{i=1}^n c(i) + \sum_{i=1}^n k(i) \right)$$

**Equation 3.** Time function for the clustering of the protein databank. Here  $fe(i)$  refers to the time taken by one of the feature extraction methods,  $c(i)$  is the take by a clustering method and  $k(i)$  the time taken by a classification method

## Expected Results

We expect that this study shows that run-times can be lowered if annotated data from clustering is used prior to the rigid body docking. The clustering model creation time can be significant but since this is a task that will be performed few times, it is negligible in the overall picture. We

don't expect that the docking of one particular pair of proteins will be faster than using TM-align and GRAMM-X but using our system for batch process large amounts of proteins will give much better response times than others. This means that Equation 1 might not take much more time than Equation 2, but after performing an amortized time analysis, the response-time will be much lower.

## References

- [1] A. Szilagyi and Y. Zhang, "Template-based structure modeling of protein-protein interactions.," *Current Opinion in Structural Biology*, pp. 10-23, 2014.
- [2] S. B. Abhishek K, "Protein-Protein Docking Using MultiDimensional Spherical Basis Functions on High Performance Computing Platform," *International Journal of Innovative Technology and Exploring Engineering*, 2019.
- [3] M. H. Avinash Mishra, "Computational Issues of Protein-Ligand Docking," *Journal of Biomolecular Research & Therapeutics*, 2018.
- [4] S. Daberdaku and C. Ferrari, "Exploring the potential of 3D Zernike descriptors and SVM for protein-protein interface prediction," *BMC Bioinformatics*, 2018.
- [5] I. Budowski-Tal, R. Kolodny and Y. Mandel-Gutfreund, "A Novel Geometry-Based Approach to InterProtein Interface Similarity," *Scientific Reports*, 2018.
- [6] P. J. Kundrotas, I. Anishchenko, T. Dauzhenka, I. Kotthoff, D. Mnevets, M. M. Copeland and I. A. Vakser, "Dockground: A comprehensive data resource for modeling of protein complexes," *Protein Science*, pp. 172-181, 2018.
- [7] PRIS-Lab, "TARÁ - Cluster HPC PRIS-Lab," PRIS-Lab/UCR, 1 8 2019. [Online]. Available: <https://wiki.prislab.org/doku.php?id=tara>. [Accessed 1 8 2019].
- [8] H. K. Janin J., "CAPRI: a Critical Assessment of PRedicted Interactions.," *Proteins*, 2003.
- [9] R. Méndez, R. Laplae, M. F. Lensink and S. J. Wodak, "Assessment of CAPRI predictions in rounds 3-5 shows progress in docking procedures," *Proteins*, 2005.
- [10] G. G.-P. Varela-Salinas, "Visual Clustering Approach for Docking Results from Vina and AutoDock," *Hybrid Artificial Intelligent Systems*, 2017.
- [11] S. R. Comeau, D. W. Gatchell, S. Vajda and C. J. Camacho, "ClusPro: a fully automated algorithm for protein-protein docking," *Nucleic Acids Research*, pp. 96-99, 2007.
- [12] J. J. Gary, S. W. C. S.-F. O. Moughon, B. Kuhlman, C. A. Rohl and D. Baker, "Protein-Protein Docking with Simultaneous Optimization of Rigid-Body Displacement and Side-Chain Conformations," *Journal of Molecular Biology*, pp. 281-299, 2003.
- [13] D. Kozakov, K. H. Clodfelter, S. Vajda and C. J. Camacho, "Optimal Clustering for Detecting Near-Native Conformations in Protein Docking," *Biophysical Journal*, pp. 867-875, 2005.
- [14] S. Vajda and D. Kozakov, "Convergence and combination of methods in protein-protein docking.," *Current Opinion in Structural Biology*, pp. 164-170, 2009.
- [15] M. Torchala, I. H. Moal, R. A. Chaleil, J. Fernandez-Recio and P. A. Bates, "SawrmDock: a server for flexible protein protein docking," *Bioinformatics*, pp. 807-809, 2013.
- [16] S. Lorenzen and Y. Zhang, "Identification of near-native structures by clustering protein docking conformations," *Proteins: Structure, Function and Bioinformatics.*, pp. 187-194, 2007.
- [17] Y. Zhang and J. Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," *Nucleic Acids Research*, pp. 2302-2309, 2005.
- [18] A. Tovchigrechko and I. A. Vakser, "GRAMM-X public web server for protein-protein docking," *Nucleic Acids Research*, 2006.