COCαDA - Large-Scale Protein Interatomic Contact Cutoff Optimization by Cα Distance Matrices

Rafael P. Lemos[1], **Diego Mariano**[1], **Sabrina A. Silveira**[2], **Raquel C. de Melo-Minardi** [1]

[1]Laboratory of Bioinformatics and Systems (LBS)
Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil

[2]Laboratory of Bioinformatics, Visualization and Systems (LaBio)
Universidade Federal de Viçosa (UFV), Viçosa, Brazil

rafaellemos42@gmail.com, raquel.minardi@gmail.com

Abstract. *Contacts, defined as inter- and intramolecular interactions predicted computationally, are typically detected using Euclidean distance and atom types. However, traditional methods can be computationally expensive and limit scalability. We introduce COCαDA (Contact Optimization by Cα Distance Analysis), a novel method that incorporates domain knowledge of amino acids to optimize distance cutoffs, simplifying implementation and enhancing efficiency. COCαDA outperforms traditional methods such as all-against-all, static cutoff (SC), and Biopython's NeighborSearch (NS), averaging 2.5x faster than SC and 6x faster than NS. COCαDA is well-suited for exploratory and large-scale analyses and is freely available at https://github.com/LBS-UFMG/COCaDA.*

1. Introduction

Proteins are macromolecules that are essential to life. In living beings, they perform several tasks such as composing cellular structure, defending against invaders, transporting nutrients, accelerating enzymatic reactions, among other functions [Nelson and Cox 2008]. These molecules are composed of amino acid residues interconnected by strong covalent peptide bonds. In addition to these bonds, weaker interactions like hydrogen bonds, electrostatic forces, and hydrophobic interactions contribute to the protein's final shape, which is crucial for its function [Smetana and Misra 2017]. Therefore, understanding a protein's shape, along with the molecular bonds and interactions that determine it, is essential for understanding its function.

Contacts can be defined as being protein inter- and intramolecular interactions (or bonds), that are predicted *in silico*. Using computational methods can assist in labor-intensive and costly experimental strategies, and facilitate large-scale approaches across protein families or other sets of entries [Ding and Kihara 2018]. There are many ways to determine if two residues, or in more refined methods, individual atoms, are in contact (reviewed in [da Silveira et al. 2009]). Inter-residue comparisons offer a more coarse-grained view, while interatomic analyses provide greater precision.

The two most common approaches to detect contacts are through the establishment of Euclidean distance thresholds (or cutoffs) in a discrete or continuous form [de Melo et al. 2006, Veloso et al. 2007, Sobieraj and Setny 2021], or using cutoff-independent methods like Voronoi [Voronoi 1908] and Delaunay Tessellations

[Delaunay 1934]. However, Pires and colleagues argued that cutoff-dependent approaches would yield more concise results than cutoff-independent ones, making data interpretation easier and reducing computational burden [Pires et al. 2011].

Over the years, several tools and databases were developed to elucidate and analyze protein contacts, ranging from simple command-line interfaces to web services offering multiple interaction types and visualization strategies [Wallace et al. 1995, Mancini et al. 2004, Lee and Blundell 2009, Bickerton et al. 2011, Pimentel et al. 2021, Pires et al. 2011, Fassio et al. 2020, Schreyer and Blundell 2013, Kasahara and Kinoshita 2014, Laskowski et al. 2018, Jubb et al. 2017, Laskowski and Swindells 2011]. However, some of these tools focus specifically on protein-ligand or residue-residue interactions.

To the best of our knowledge, these tools and databases suffer from one or more of the following limitations: a) they are static, based on predefined entries and conditions; b) they are computationally expensive, making large-scale analysis impractical; c) they are limited by server loads and bottlenecks like long queues, delaying processing of more than one file at a time; d) they only calculate inter-residue or interface contacts, limiting accuracy and coverage compared to interatomic or whole-protein approaches; e) they use cutoff-independent methods, that perform worse than cutoff-dependent methods; f) they are outdated, lacking support for refined definitions or newer file types like '.cif'; or g) they have been discontinued, making their services unavailable.

The Protein Data Bank (PDB, [Berman et al. 2000]) archive currently contains 238.922 entries, with about 92% being proteins[1]. Due to advancements in experimental resolution techniques, as well as computational hardware and software, the archive grows by approximately 6.5% annually (over 10,000 new entries)[2], highlighting the need for large-scale data analysis tools.

In this context, we propose COCαDA (Contact Optimization by Cα Distance Analysis), a new ingenious Python-based approach for large-scale analysis of inter- and intrachain atomic contacts between protein residues. This approach offers significant potential to accelerate research on data-heavy structural analyses, such as studies on protein evolution, pathogen lineage mutations, virtual screening of compounds, among others. COCαDA can be easily adapted to any existing analysis workflow, or be run independently for exploratory purposes.

COCαDA applies contact cutoffs based on the maximum possible Cα distance between a pair of residues, determined after analyzing all protein structures in the PDB. COCαDA includes a customized parser for both PDB and mmCIF files, containing functionalities for handling large files, filtering specific residues and interactions, and calculating geometric properties such as centroid and normal vectors for aromatic residues. In addition, the tool supports parallel execution across any selection of available CPU cores.

To compare and benchmark our approach to others in the literature, we conducted two case studies: a small dataset of gold standard enzyme superfamilies, aiming to address different types of proteins and benchmark against slower methods; and a very large dataset, encompassing all PDB protein entries under 10,000 residues, to accurately assess

[1]Available at https://www.rcsb.org/stats/explore/polymer_entity_type. Collected on August 23, 2024.
[2]Available at https://www.rcsb.org/stats/growth/growth-protein. Collected on August 23, 2024.

COCαDA performance and asymptotic growth.

2. Methodology

Figure 1 outlines the methodology for developing and benchmarking COCαDA. The process begins with defining contacts and applying a static cutoff distance to the full PDB dataset. COCαDA then uses the maximum possible Cα distance matrix for optimizing contact detection. The tool was benchmarked against similar methods using two datasets, focusing on processing time and computational complexity.

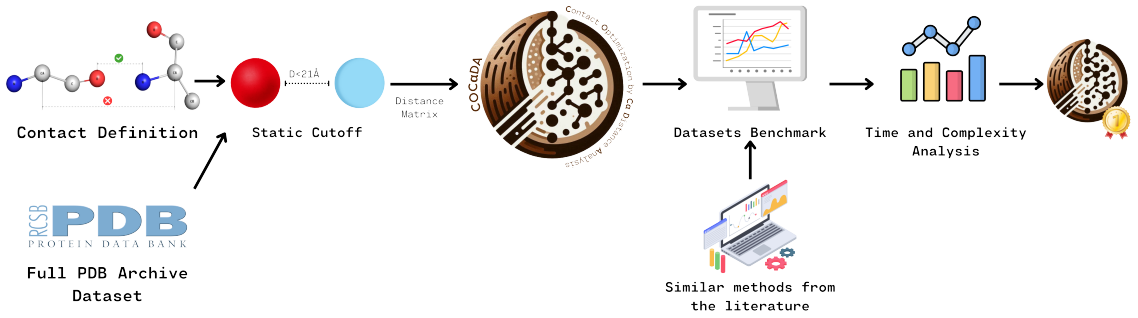

Figure 1. **Overview of the methodology used to create and benchmark COCαDA.**

2.1. Contact Definition

To store the interaction types and their conditions, we used a dictionary containing all heavy atoms from the 20 standard amino acids, as defined in [Sobolev et al. 1999, Fassio et al. 2020]. The possible interactions are: hydrogen and disulfide bonds; hydrophobic, attractive, repulsive, and salt bridge interactions; and aromatic stackings.

This dictionary also contains the conditions needed for the interaction (e.g., to form an attractive interaction, the atoms must be differently charged), and the range of Euclidean distances, in angstroms, for the interaction to occur (Table 1).

Table 1. **Summary of Types, Range and Conditions for contacts to occur.**
D_a = Euclidean distance between the atom pair.

Contact Type	Range (Å)	Condition (other than range)
Hydrogen Bond	$0 \leq D_a \leq 3.9$	Acceptor + Donor atoms
Disulfide Bond	$0 \leq D_a \leq 2.8$	Cys:SG + Cys:SG atoms
Hydrophobic	$2.0 \leq D_a \leq 4.5$	Hydrophobic + Hydrophobic atoms
Repulsive	$2.0 \leq D_a \leq 6.0$	Equally charged atoms
Attractive	$3.9 \leq D_a \leq 6.0$	Differently charged atoms
Salt Bridge	$0 \leq D_a \leq 3.9$	Differently charged atoms
Aromatic Stacking	$2.0 \leq D_a \leq 5.0$	Centroids of two aromatic rings in parallel or perpendicular angle

2.2. Protein Data Bank Archive

The full PDB protein archive, in '.cif' format, was obtained using in-house scripts to query and download entries directly from the PDB API. First, a script was used to

query the API for entries containing "Protein" as an exact match from the parameter "entity_poly.rcsb_entity_polymer_type".

To avoid rate limits and overwhelming the server, queries had a 1 second delay from one another, and only 25,000 IDs were obtained at a time. Then, a second script was used, together with the Biopython Bio.PDB module [Cock et al. 2009], to download all files that matched the IDs gathered in the first step. All files were downloaded between July 4th and July 10th, 2024.

2.3. Biopython Implementation

To serve as a comparison to our method, the Biopython package [Cock et al. 2009], largely used in bioinformatics, was used. The Bio.PDB module contains tools to parse a .pdb or .cif file, as well as the NeighborSearch (NS) class, which is useful in interatomic contact determination.

We then performed an all-atom neighbor search of 6Å radius, the maximum distance for contacts defined in our dictionary. Then, the neighbors were filtered based on their distance and physicochemical properties relative to the parent atom. The contacts obtained contained the following information: chain, residue number, and parent atom name; chain, residue number, and neighbor atom name (i.e. the atomic pair making the contact); type of interaction; and distance between the two atoms.

2.4. General Implementation and Static Cutoff

To analyze the PDB protein archive and obtain the maximum distances matrix used in the rest of this work, we first devised a Static Cutoff (SC) implementation, where the Cα cutoff distance was fixed. Akin to Biopython, proteins are treated as Python objects, containing chains, residues, and atoms. The package includes a customized .pdb/.cif parser, devised to rapidly extract only relevant information for contact determination, considering the following points as defaults:

- Only the first model of each protein is considered;
- Only atoms with occupancy ≥ 0.50 are considered;
- Hydrogen atoms and non-standard residues are not considered;
- DNA and RNA molecules are not considered.

After parsing, the protein object is passed to a contact calculation script, where the Cα distances for each pair of residues are obtained, and filtered based on the fixed cutoff. To calculate normal vectors and angles for aromatic stacking contacts, the Python Numpy package was used [Harris et al. 2020].

The atoms from the residues that are in range to interact are then compared to the dictionary previously described, based on their distance to each other, and their physicochemical properties. Finally, the contacts are returned in a custom object containing all their information, similar to the NS method.

2.5. Maximum Distances Matrix and COCαDA Implementation

During the SC approach, the maximum identified Cα distance for each amino acid pair was stored in a matrix, and we then updated the cutoff ranges to reflect the new values, herein called COCαDA.

The distance matrix $D = [d_{ij}]_{n \times n}$ is a square matrix of size $n \times n$, where n represents the number of standard amino acids. Each entry d_{ij} corresponds to the maximum distance between the Cα atoms of the amino acids at positions i and j (e.g., d_{11} represents an Alanine pair, and d_{nn} represents a Valine pair):

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}, \tag{1}$$

where each d_{ij} represents the maximum Euclidean distance between the Cα atoms of the amino acids at positions i and j.

A total of 210 distance values were obtained, representing each possible residue pair and excluding redundancies (e.g. Ala-Val is the same as Val-Ala) (Equation 2). All calculations and definitions are exactly equal to the SC Implementation.

$$P = \frac{n(n-1)}{2} + n, \tag{2}$$

where P is the number of non-redundant distance pairs, and n is the number of standard amino acids. In this case, as n = 20, then P = 210.

2.6. Datasets

Two datasets were selected to benchmark our results and compare them to other competitors ($n_{dataset1}$ = 896 and $n_{dataset2}$ = 215,716). The first is a modified gold-standard set of enzyme superfamilies [Brown et al. 2006], with 365 unique entries ranging from 194 to 6,208 residues. For a more balanced comparison, we split all chains in different files, and treated them separately. The new modified dataset contains 896 entries, ranging from one to 994 residues. The second dataset includes all PDB proteins with less than 10,000 residues, covering approximately 99.2% of all protein entries. This dataset contains 215,716 unique entries, ranging from three to 10,000 residues.

2.7. Benchmarks

To ensure fairness and eliminate bias, all benchmarks were conducted simultaneously using the same setup on a single core per process, to avoid memory overhead and parallelization issues. The second dataset was divided into nine batches of approximately 25,000 files, with each processed independently on a separate core, avoiding overlaps.

Although multithreading is available for all implementations using Python's 'concurrent.futures' standard module, it was not used; instead, each core handled a distinct batch to maintain consistency. The processing time measured for each entry included file reading, parsing, contact detection, and output generation.

3. Results and Discussion

3.1. Maximum Distance Matrix

In total, 217.454 PDB entries were downloaded in .cif format, totaling approximately 450 GB. Proteins ranged from three (PDB IDs: 1Q7O, 8DDG, 8DDH) to 503,221 (PDB ID:

8GLV) modeled amino acid residues. To obtain the values for the distance matrix, we processed all the downloaded files using a fixed cutoff of 21Å for all pairs of residues (SC). This value is comfortably above the maximum distance between the Cα of a pair of arginines, the biggest residues by length. To confirm this, we compared an all-atom approach (i.e., comparing every atom of the protein against each other, without cutoffs) to the SC approach, in a small test dataset, and no contacts were missed (data not shown).

Using the 217,454 entries from the PDB, over 211 million amino acid residues and 819 million contacts were processed and identified. We stored the maximum Cα distances for every pair of the 20 standard amino acids, and after merging redundancies, we obtained 210 values in a symmetric distance matrix (Figure 2, Equation 1).

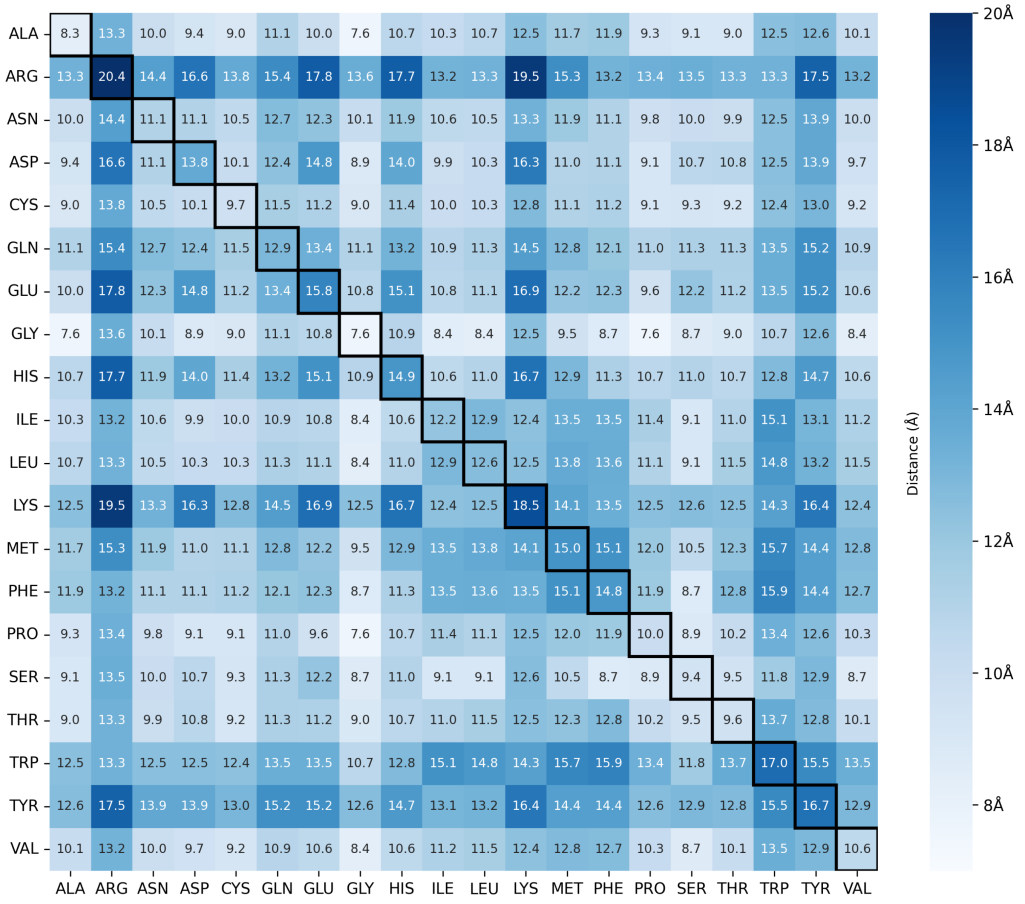

Figure 2. **Distance Matrix between the Cα of all pairs of residues.**
The intensity of the color indicates the scale of the value (from 7 to 20 Å). Highlighted diagonals represent pairs of the same residue.

As the distance matrix is color-coded based on the value of the maximum Cα distance, we can quickly spot the minimum and maximum values determined. A pair of two arginine residues, from chains G and H of PDB ID 3X0Y, represents the highest value encountered, of 20.46Å (Figure 3a). This is expected and corroborates, while also being lower, with the fixed distance of 21Å used in the SC approach, once again demonstrating that the fixed cutoff was appropriate to yield no missed contacts.

For the lowest value encountered, we found a pair consisting of an alanine and a glycine residue, both present in the HD chain of PDB ID 6QCM, with a distance of 7.65Å between their Cα's (Figure 3b). This is expected as well, as alanines and glycines

are two of the smallest amino acid residues, differing only by a single CH_3 group in the side chain of the alanine, while glycine has a hydrogen atom in its side chain. However, even with this difference, the presence of the CH_3 group on the side chain of the alanine does not impact the distance between their Cα atoms, only contributing to the chirality of the alanine residue. We can check the lack of influence of this side group by comparing the distance between this pair (7.65Å) with a pair of two glycine residues, which has a maximum value of 7.77Å, just a little above the first one.

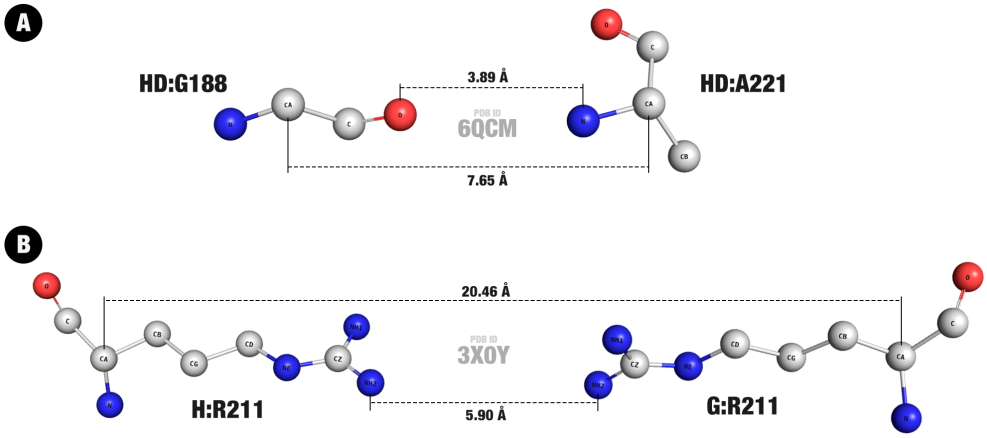

Figure 3. **Maximum and minimum entries in the Distance Matrix.**
a) Maximum value, in a contact between two arginine residues. b) Minimum value, in a contact between an alanine and a glycine residue. The higher number represents the distance between Cα, and the lower one the contact distance. The PDB IDs are shown in center, and the contact details are shown in the format Chain:Residue-Atom.

Now comparing the actual contacts of the maximum and minimum values encountered in the distance matrix, we can see that their values are almost to the limit defined in our dictionary. For the Arg-Arg pair, which makes a repulsive contact between their nitrogen (NH_2) atoms, the distance is 5.9Å, very close to the 6Å limit defined in the dictionary (Figure 3a). For the Ala-Gly pair, the only possible contact type is hydrogen bonds between their atoms, as the main chain atoms are only capable of donating (main chain nitrogen) or accepting (main chain oxygen) hydrogen atoms. The maximum contact value is then 3.89Å, even closer to the 3.9Å distance limit for hydrogen bonds (Figure 3b).

3.2. Benchmarks

With the maximum possible Cα distances properly established for all amino acid residue pairs, we updated the "distances" dictionary with the new values, creating the COCαDA (Contact Optimization by alpha-Carbon Distance Analysis) approach. To benchmark COCαDA against other approaches used in the literature, we selected the following: all atoms against all atoms (AllAtoms, used in [Pimentel et al. 2021]), Arpeggio Web[3] [Jubb et al. 2017], Arpeggio CLI[4] [Jubb et al. 2017], Biopython Neighbor Search (NS), and Static Cutoff (SC). Other methods, like nAPOLI [Fassio et al. 2020], STING Contacts [Mancini et al. 2004], and PICCOLO [Bickerton et al. 2011], were not available at the time of search, so they were not considered.

[3]Available at https://biosig.lab.uq.edu.au/arpeggioweb/.
[4]Available at https://github.com/PDBeurope/arpeggio/.

Both Arpeggio versions were too slow to process even small proteins, as our tests showed processing times of approximately 5 and 23 minutes for a single 1,000 residue protein (PDB ID 6RTH) for Arpeggio CLI and Arpeggio Web, respectively. This can be due to several factors, but we believe that the explanation lies in server load (for Arpeggio Web), and the several external libraries and computing time that are needed to run the more complex analysis (for both versions). So, as our goal is to make a fast, yet robust, tool to calculate a massive number of contacts for a large list of proteins, both Arpeggio versions were excluded from further analysis.

The first dataset used contains 896 entries, ranging from one to 994 residues. A smaller dataset was chosen first to be able to compare the slowest approach (AllAtoms) with the others, so we could get a sense of scale. In Figure 4, it is possible to see that the AllAtoms approach rapidly explodes in a quadratic curve compared to the three others, which maintain rather linear calculation times up to 1,000 residues. With this, we also removed AllAtoms from further analysis. Comparing the faster approaches, SC obtained calculation times 1.5x faster on average than NS, while COCαDA showed calculation times 3.8x faster on average, obtaining the same contacts.

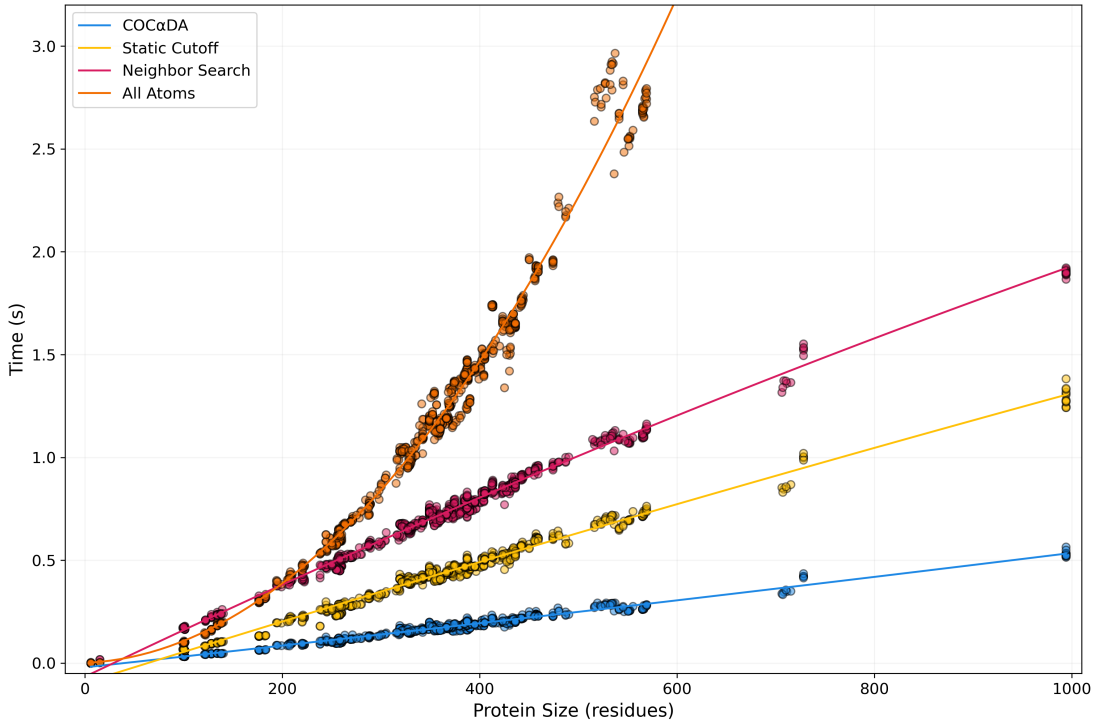

Figure 4. **Protein Size vs. Computation Time plot of Benchmark 1.**
In Benchmark 1, 896 files ranging from 1 to 994 residues were analyzed. COCαDA is shown in cyan, SC in yellow, NS in magenta, and AllAtoms in orange. Points represent individual entries, with lighter lines showing the fitted curves for the data.

As the results from the first, small dataset showed a significant difference in processing times between the 3 fastest approaches, we then moved to the second dataset, which contains 215,716 unique entries, ranging from three to 10,000 residues, making approximately 99.2% of the PDB protein archive. The choice to remove entries above 10,000 residues was made due to the nature of those entries, which are mostly protein complexes, containing several copies of each unique chain. This makes them not suitable for contact analysis directly, requiring some kind of pre-processing, like splitting only

the unique chains or working with each individual protein present in the complex separately. This can also be true for entries below 10,000 residues, but we believe that this slice correctly represents the diversity of experimentally resolved protein structures.

Figure 5 shows the results for dataset 2 when comparing the COCαDA, SC, and NS approaches. It is possible to see that COCαDA performs better for all proteins, averaging approximately 6x faster times than NS and 2.5x faster times than SC. The SC approach performs better than NS in proteins below 5,000 residues, equal between 5,000 and 7,000 residues, and worse above 7,000 residues. Outliers were considered as entries that had a processing time ± 5 times the Standard Deviation for each approach, with less than 1% of entries removed. After outlier removal, it is possible to see that COCαDA has a consistent time vs. size distribution, while the other two approaches have more variation. This can be due to the tight and precise definition of cutoff distances for COCαDA, which speeds up a lot of the computation, while also keeping little room for variations.

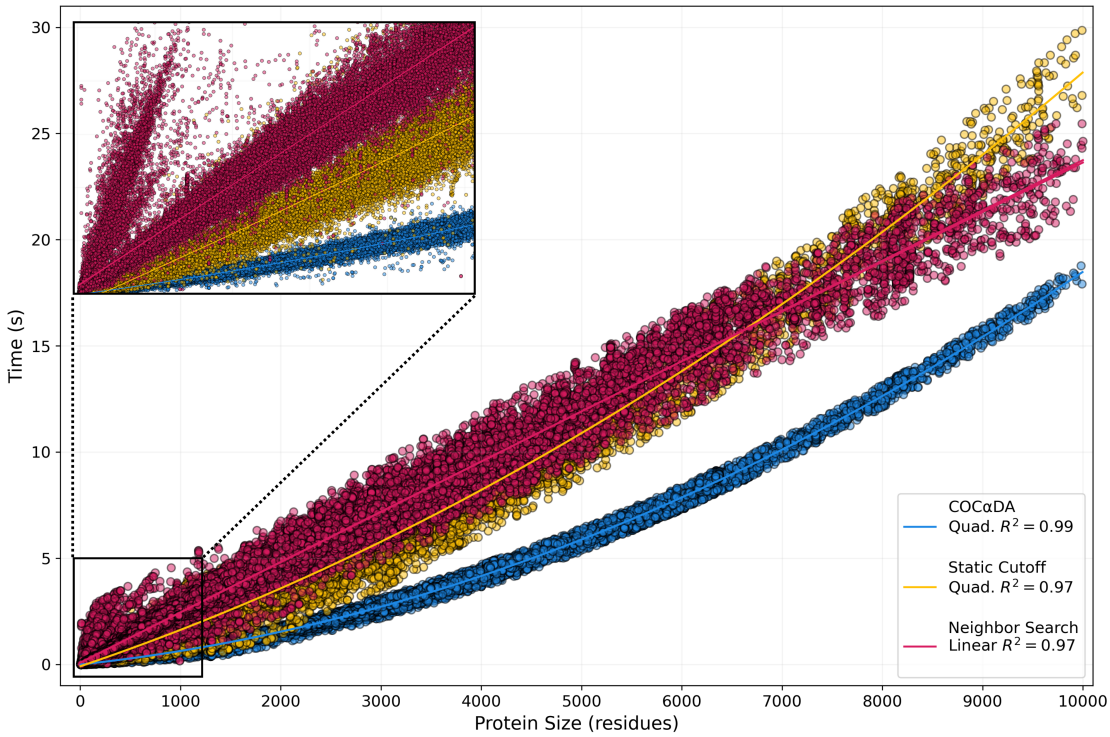

Figure 5. **Protein Size vs. Computation Time plot of Benchmark 2.**
In Benchmark 2, 215,716 files were analyzed, ranging from 3 to 10,000 residues. COCαDA is shown in cyan, SC in yellow, and NS in magenta. A detail of the 0-1,000 protein size range is shown in the upper left corner. Points represent individual entries. Curve fits are shown in the lower right corner and as darker lines on the data, and outliers were defined as ± 5 times the Standard Deviation for each approach.

The protein size range of 0-1,000 residues is noteworthy, as shown in detail in the upper left corner of Figure 5. At this range, we can see the time difference observed in dataset 1, while also identifying several divergent entries in the NS approach. The divergent spike is composed only of Nuclear Magnetic Resonance (NMR) resolved entries, and due to the nature of Biopython native parsing, all the models need to be parsed before the first one is selected. As these entries are small, the parsing time of several NMR models outpaces the contact calculation time of the first one, leading to a spike in processing time. This does not occur in the COCαDA and SC approaches, as the customized parser

handles only the selected model in the file (the default value is always the first model).

3.3. Complexity Analysis

Computing interatomic contacts is inherently a quadratic problem because it requires calculating the distance between every pair of atoms. However, sophisticated data structures, such as octrees, can be employed to avoid calculating distances between atoms/residues that are too far apart. This approach theoretically reduces the computational space by pruning irrelevant comparisons, leading to practical reductions in computation time and typically logarithmic or log-linear complexity. However, they can be more complex to implement and may perform poorly for small input sizes due to the overhead associated with allocating and populating the data structures.

In this study, we chose to evaluate the complexity of various algorithms empirically, by comparing standard methods commonly used in the structural bioinformatics community with the COCαDA method. These different methods were tested with inputs of increasing sizes (where n represents the number of residues, which have on average eight atoms each), and we analyzed the resulting fitted curves with real datasets.

The curve fittings of the three approaches against the second dataset demonstrate that both COCαDA (R^2 quadratic = 0.99, Equation 3) and SC (R^2 quadratic = 0.97, Equation 4) exhibit quadratic growth trends, while NS shows a linear growth trend (R^2 linear = 0.97, Equation 5). This difference arises from the nature of the contact identification functions, which are the most time-consuming operations. COCαDA and SC have a time complexity of $O(n^2)$, reflecting the quadratic growth in the number of contacts identified as the entry size increases. In contrast, NS contact identification function operates with a time complexity of $O(n)$, leading to its linear growth pattern.

$$f(n) = 1.35 \times 10^{-7} n^2 + 5.04 \times 10^{-4} n - 6.36 \times 10^{-3}; \tag{3}$$

$$g(n) = 1.20 \times 10^{-7} n^2 + 1.60 \times 10^{-3} n - 9.18 \times 10^{-2}; \tag{4}$$

$$h(n) = 2.37 \times 10^{-3} n + 7.94 \times 10^{-2}, \tag{5}$$

where $f(n)$, $g(n)$, and $h(n)$ are the best-fitted functions for COCαDA, SC, and NS, respectively, and n is the number of residues.

However, for proteins below 10,000 residues, the quadratic growth of COCαDA is so small that it outperforms the linear growth of NS in all entries. Examining Equation 3, we can see that the quadratic coefficient is orders of magnitude smaller than the linear coefficient and the constant term, which explains the slow rate of growth observed. These terms are so small that even when compared to Equation 5, from NS, only when the entry has approximately 14,000 residues (or, on average, 112,000 atoms) would the two curves intersect, far above the limit of the dataset and usual usage.

4. Conclusion

In the era of Big Data in bioinformatics, the need for efficient, robust, and scalable methods and tools is higher than ever. In structural bioinformatics, the continuous influx of experimentally resolved proteins into the PDB underscores the need for innovative data analysis solutions. We present COCαDA, a free, user-friendly tool to efficiently identify protein interatomic contacts on a large scale. COCαDA employs a novel method

for interaction ranges, based on the maximum Cα distances of two amino acid pairs collected from all proteins on the PDB. By incorporating amino-acid domain knowledge to set optimal cutoff values, we have effectively minimized temporal costs. Our approach simplifies implementation while improving efficiency in large-scale protein interaction analysis, and can be used in protein evolution studies and virtual screening campaigns, among other applications.

Currently, COCαDA outputs a '.txt' file, categorized by contact type. Contacts are detected inter- and intrachain, so usage for protein-protein and protein-ligand interactions are feasible. We plan to develop a web-based interface to improve usability and enable deeper insights into complex interatomic contact networks. COCαDA is implemented in Python and available at https://github.com/LBS-UFMG/COCaDA.

5. Acknowledgements

The authors would like to thank the research funding agencies CAPES, FAPEMIG, and CNPq. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Res.*, 28(1):235–242.

Bickerton, G. R., Higueruelo, A. P., and Blundell, T. L. (2011). Comprehensive, atomic-level characterization of structurally characterized protein-protein interactions: the PICCOLO database. *BMC Bioinformatics*, 12(1):313.

Brown, S. D., Gerlt, J. A., Seffernick, J. L., and Babbitt, P. C. (2006). A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biol.*, 7(1):R8.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., et al. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.

da Silveira, C. H., Pires, D. E. V., Minardi, R. C., Ribeiro, C., Veloso, C. J. M., Lopes, J. C. D., Meira, Jr, W., Neshich, G., Ramos, C. H. I., Habesch, R., and Santoro, M. M. (2009). Protein cutoff scanning: A comparative analysis of cutoff dependent and cutoff free methods for prospecting contacts in proteins. *Proteins*, 74(3):727–743.

de Melo, R. C., Lopes, C. E. R., Fernandes, Jr, F. A., da Silveira, C. H., Santoro, M. M., Carceroni, R. L., Meira, Jr, W., and Araújo, A. d. A. (2006). A contact map matching approach to protein structure similarity analysis. *Genet. Mol. Res.*, 5(2):284–308.

Delaunay, B. (1934). Sur la sphère vide. À la mémoire de georges voronoï. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et naturelles*, VII:793–800. Zbl 60.0946.06.

Ding, Z. and Kihara, D. (2018). Computational methods for predicting protein-protein interactions using various protein features. *Curr. Protoc. Protein Sci.*, 93(1):e62.

Fassio, A. V., Santos, L. H., Silveira, S. A., Ferreira, R. S., and de Melo-Minardi, R. C. (2020). napoli: A graph-based strategy to detect and visualize conserved protein-ligand interactions in large-scale. *IEEE/ACM Trans. Comp. Biol. Bioinf.*, 17(4):1317–1328.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., et al. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Jubb, H. C., Higueruelo, A. P., Ochoa-Montaño, B., Pitt, W. R., Ascher, D. B., and Blundell, T. L. (2017). Arpeggio: A web server for calculating and visualising interatomic interactions in protein structures. *Journal of Molecular Biology*, 429(3):365–371.

Kasahara, K. and Kinoshita, K. (2014). GIANT: pattern analysis of molecular interactions in 3D structures of protein-small ligand complexes. *BMC Bioinformatics*, 15(1):12.

Laskowski, R. A., Jabłońska, J., Pravda, L., Vařeková, R. S., and Thornton, J. M. (2018). PDBsum: Structural summaries of PDB entries. *Protein Sci.*, 27(1):129–134.

Laskowski, R. A. and Swindells, M. B. (2011). LigPlot+: multiple ligand-protein interaction diagrams for drug discovery. *J. Chem. Inf. Model.*, 51(10):2778–2786.

Lee, S. and Blundell, T. L. (2009). BIPA: a database for protein-nucleic acid interaction in 3D structures. *Bioinformatics*, 25(12):1559–1560.

Mancini, A. L., Higa, R. H., Oliveira, A., Dominiquini, F., Kuser, P. R., Yamagishi, M. E. B., Togawa, R. C., and Neshich, G. (2004). Sting contacts: a web-based application for identification and analysis of amino acid contacts within protein structure and across protein interfaces. *Bioinformatics*, 20(13):2145–2147.

Nelson, D. L. and Cox, M. M. (2008). *Lehninger principles of biochemistry*. W.H. Freeman, New York, NY, 5 edition.

Pimentel, V., Mariano, D., Cantão, L. X. S., Bastos, L. L., Fischer, P., de Lima, L. H. F., Fassio, A. V., and de Melo-Minardi, R. C. (2021). VTR: A web tool for identifying analogous contacts on protein structures and their complexes. *F. Bioinf.*, 1:730350.

Pires, D. E. V., de Melo-Minardi, R. C., dos Santos, M. A., da Silveira, C. H., Santoro, M. M., and Meira, Jr, W. (2011). Cutoff scanning matrix: structural classification and function prediction by protein inter-residue distance patterns. *BMC Gen.*, 12(S4):S12.

Schreyer, A. M. and Blundell, T. L. (2013). CREDO: a structural interactomics database for drug discovery. *Database (Oxford)*, 2013:bat049.

Smetana, J. H. C. and Misra, G. (2017). Principles of protein structure and function. In *Intro. to Biomol. Struct. and Biophys.*, pages 1–32. Springer, Singapore.

Sobieraj, M. and Setny, P. (2021). Entropy-based distance cutoff for protein internal contact networks. *Proteins*, 89(10):1333–1339.

Sobolev, V., Sorokine, A., Prilusky, J., Abola, E. E., and Edelman, M. (1999). Automated analysis of interatomic contacts in proteins. *Bioinformatics*, 15(4):327–332.

Veloso, C. J. M., Silveira, C. H., Melo, R. C., Ribeiro, C., Lopes, J. C. D., Santoro, M. M., and Meira, Jr, W. (2007). On the characterization of energy networks of proteins. *Genet. Mol. Res.*, 6(4):799–820.

Voronoi, G. (1908). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les paralléllòèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287.

Wallace, A. C., Laskowski, R. A., and Thornton, J. M. (1995). LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions. *Prot. Eng.*, 8:127–134.