GACAI-PCP: Cellular Automata based Tool for Contact Map Prediction

***Abstract*– We describe a software tool that we developed to build contact map predictors. This is a novel approach that exploits the capabilities of cellular automata to offer complex behavior from local interactions only. Our tool identifies cellular automata able to classify predicted protein contacts that are more likely real contacts. The core of our cellular automata identification tool is a genetic algorithm that optimizes the prediction of balanced contact maps. With our tool, we proved that is possible to improve contact map prediction by means of cellular automata models.**

***Keywords*-- Protein Contact Prediction, Protein Contact Map, Cellular Automata, Inverse Design.**

I. Introduction

Protein contact maps prediction (CMP) and protein contact prediction (PCP) fields arose in 1970's [1]. Nowadays, the field has shown a great development particularly since the inception of the analysis of correlated mutations [2]. Though, the first PCP tools that implements correlated mutations were developed more than a decade ago, just recently were solved the issue of false correlations that dampened the success of this kind of tools [3]. We realized that cellular automata (CA), can be used as a tool to improve CMP.

PCP has acquired importance because of its helpfulness in template-free protein structure prediction [4]. PCP provides spatial constraints derived from the protein chain that can be used in tertiary structure reconstruction or in pipelines that predict more detailed native structures [5]. PCP has been a tool used since the 1970s when Tanaka and Scheraga used protein contacts in an approach for protein folding [1]. After several decades of advancement, PCP has taken a prominent place in protein folding and protein structure prediction, especially for proteins that have few homologs [6]. Despite the current progress, there is room for PCP improvement in the way of enhancing the contacts in the context of the overall protein structure and the contact map that represents the protein. In Fig. 1, we show an example of a predicted protein contact map and a real contact map. RaptorX-Contact [7] was the tool used to predict the contact map in Fig. 1. The protein used for the Fig. 1 is part of the benchmark data set of the biennial Critical Assessment of Protein Structure Prediction (CASP) for year 2016 (CASP12). The predicted contact map in **Error! Reference source not found.** includes a high proportion of all the contacts in real contact maps (true positives or TP), but the proportion of false contacts (false positives or FP) is bigger.

Digital Object Identifier: (to be inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).

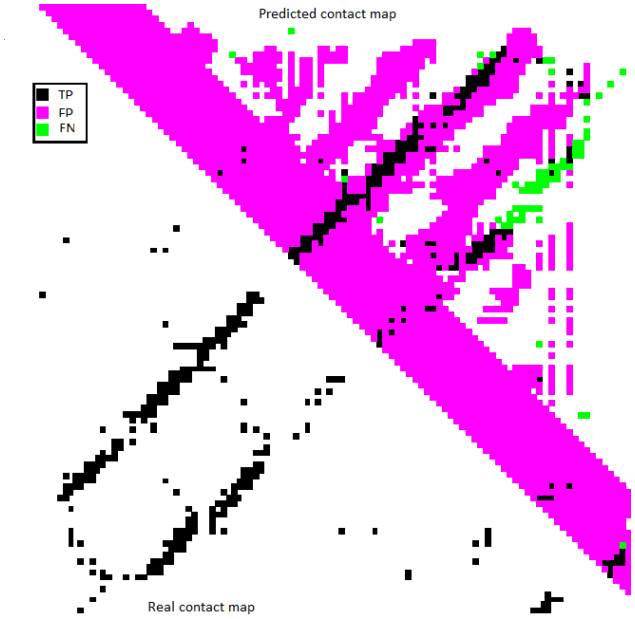

Fig. 1. RaptorX-Contact predicted contact map and actual contact map for target protein T0900 (CASP12). Upper triangular: RaptorX-Contact predicted contacts. Lower triangular: Actual contact map. TP denotes true positives (true contacts predicted); FP indicates false contacts predicted (false positives); FN (false negatives) represents actual contacts that were not predicted.

We propose a software tool that can identify CAs that transform a PCP to a contact map that is closest to a real one. Researchers rely on approaches for CA identification when the knowledge about the inherent mechanisms that define transitions is insufficient [8]. In the case of PCP, many prediction tools contesting in the CASP, use the idea that local effects govern the presence of contacts for amino acid pairs. Usually this concept is implemented by defining a window and the prediction tools analyse information from MSAs and sequence related features (i.e., solvent accessibility, secondary structure). This window is analogous to an arbitrary neighborhood in a CA, but there is little evidence supporting that this is the right neighborhood that determines residue-residue contacts.

In this paper, we describe the methods we used for CA identification for PCP. In section II we describe the databases, datasets, and tools used in our CA identification process. In section III, we compare a CA identified by our approach with CAs defined by commonly used neighborhoods. Finally, in section IV we state some conclusions about our methods for CA identification for PCP.

II. Materials and Methods

A. Initial Conditions Dataset

We required several protein reference databases to build MSAs and from these we obtain initial PCPs using CCMpred. The reference databases that we used were Uniprot20 [9] and Uniref100 [10]. To generate MSAs on Uniprot20 we used HHblits [11] and JackHHMER [12] on Uniref100. In Fig. 2 we show the pipeline that we defined to build a dataset of contact maps. These contact maps are used as initial conditions to extract rules for CAs. The phase of MSAs generation ends with the best MSA as input for CCMpred.

We build the dataset of initial conditions from the hundred and fifty protein dataset that was described in [13]. This set of proteins contains biological macromolecules, with lengths in the range [50, 275] amino acids, with high resolution (≤ 1.9Å), and unique Pfam domains [14].

CCMpred takes as input the optimal MSA for each sequence in the training dataset (Fig. 2) and returns a matrix that predicts the coupling scores for each pair of amino acids in the sequence. We use ConKit [15] to extract a contact from the CCMpred coupling scores matrix.

B. Cellular Automata Identification Framework

We used the architectural framework described in [16] to implement a genetic algorithm that identifies CAs that evolve predicted contact maps. This framework provides a core architecture that eases the process of implementation of several kinds of algorithms for CA identification. By using this framework, we can put the focus on the details of the CAs that we want to obtain, because there are many tasks common in CA identification ready to use.

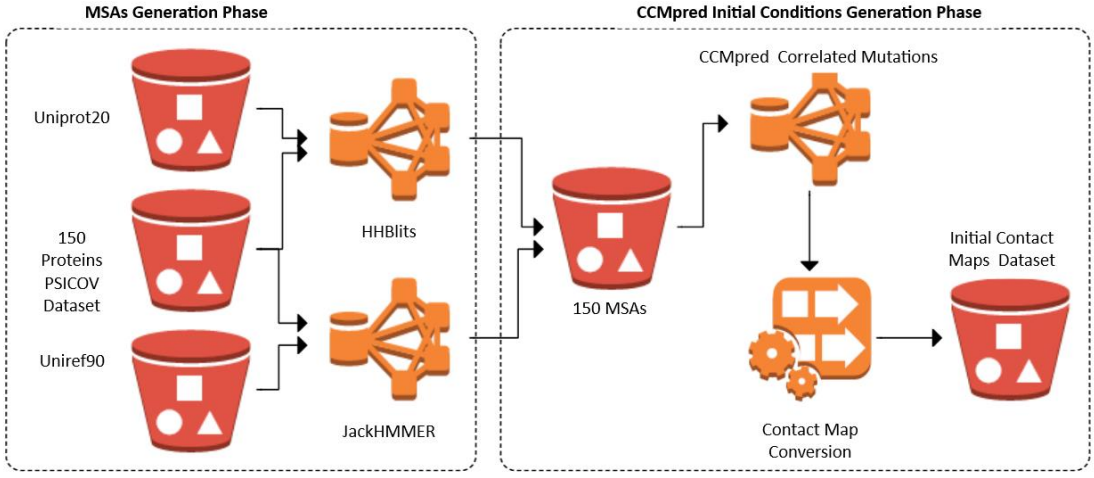

Fig. 2. Pipeline for the generation of the dataset of initial conditions.

C. Genetic Algorithm for Identification of CAs that evolve Contact Maps

CAs identification is an optimization task. In our case we require an algorithm that searches for an optimal CA that improves a predicted contact map by reducing the proportion of false contacts while keeps real contacts. Our approach implements a genetic algorithm (GA). The search space for our GA includes 5.63×10^{14} possible CAs. The size of the search space depends of the size of the maximum neighborhood, which is a matrix of 7×7 around each cell in the lattice.

In Fig. 3 we show a high-level description of the steps that implements our GA. The first step is to arrange the training dataset (pairs of predicted/real contact map) in data strata, which allow us to reduce computational cost for an iteration of the GA. By splitting the data, we can use a stratum in each GA iteration to avoid model overfitting and increase diversity in the set of CA transitions. In each iteration half of the contact maps are used for transitions identification and the remaining contact maps are used for testing. When the dataset has been consumed by the GA, it continues using the strata in the same order.

For each iteration the GA creates a new population of one hundred CAs. For the first iteration the population is generated in a random way, and for the following iterations the new population is generated by GA operators of selection and searching. Each CA in the population is evaluated in parallel, exploiting the high-level library DASK [17] that allows to run parallel process in modern computing clusters. Once each CA is evaluated in a parallel and independent process, the master node retrieves all the results (CAs and individual scores). Then the global evaluation arranges the best individuals to be used in the next iteration. A new iteration is executed until the stop condition is met, i.e., two-thousand iterations are done, or a perfect CA is found.

For quality evaluation of each CA we used the Matthews Correlation Coefficient [2], which is insensitive to class bias. PCP is a highly biased problem, because the proportion of non-contacts is very high, so that measures that are sensitive to

class bias could search for models that prefers to predict non-contacts in most of the cases.

We named our algorithm GACAI-PCP as an acronym for Genetic Algorithm for Cellular Automata Identification for Protein Contact Prediction.

In DCT we can use small lattices, datasets with lattices of 21 × 21 initial conditions are very common, so that less processing is required for parameters tuning. In contrast, in PCP an average contact map can easily have a size of 300×300.

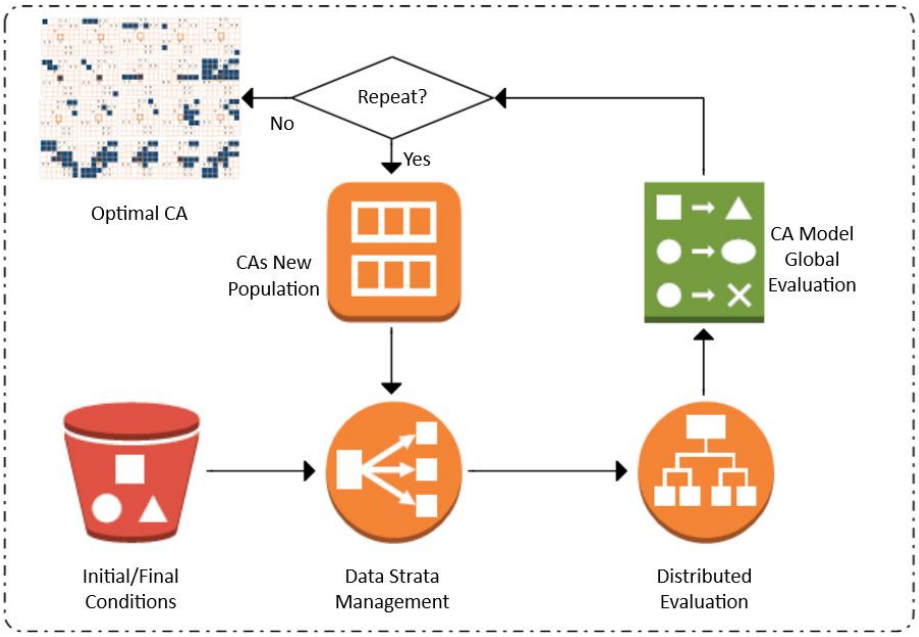

Fig. 3. Genetic Algorithm Process for CA identification.

D. Density Classification Task as Benchmark for GACAI-PCP

We named our algorithm GACAI-PCP as an acronym for Genetic Algorithm for Cellular Automata Identification for Protein Contact Prediction.

GACAI-PCP as described in the above section, was adapted from our solution for the density classification task (DCT). GACAI-DCT was the solution that we proposed for DCT [18]. We used DCT as a benchmark for our approach to DCT, because the two problems are similar in some aspects. DCT is a theoretical problem that requires the identification of CAs that can evolve an initial condition that has majority of a class (0 or 1) to a final condition where all the cells in the lattice have the state of the majority class. DCT requires that the CA implements mechanisms of global coordination expressed just by the local transitions included in the CA rule. CA global coordination is a property that emerges when the CA rule is applied to an initial condition iteratively and it is hard to design manually. For the 2D DCT case, we have a problem similar to PCP: 1) The initial and final conditions are known; 2) The optimal neighborhood is unknown; 3) The rule transitions can be deterministic or stochastic (there are no explicit restriction about this); 4) The lattice can be a regular n × n 2D binary matrix; 5) the border condition can be cyclic.

In GACAI-PCP we used the same algorithm as for GACAI-DCT and the only important difference is that the MCC used in GACAI-PCP is adjusted to evaluate only the upper triangular of the contact map to reduce the processing time.

III. Discussion

GACAI-PCP search space includes CA's neighborhoods that are common in CAs modelling. Perhaps, the more widely used neighborhood used is the Moore's neighborhood of radius one (Fig. 4.a). We compare some obvious CAs using the Moore's neighborhood of radius one, two and three (Fig. 4.a-c), as well as a random neighborhood (Fig. 4.d) against a CA evolved by GACAI-PCP.

For evaluation in this paper we used the target proteins in the CASP12 dataset, which is comprised of 39 target proteins (predictioncenter.org). In Fig. 5, we show the precision of the five CAs defined by the neighborhoods of the Fig. 4, measured for the full list of predicted contacts. The model identified by GACAI-PCP outperforms every other CA. The only target protein where our model was outperformed was T0862. The CAs that use Moore's neighborhood of radius one and two, and the CA with random neighborhood show a similar performance with precision in the range [0.0, 0.1]. The CA of radius three is the only model in the comparison that surpasses the threshold precision at 0.1, for twelve out of the 39 proteins

in the evaluation dataset. But, in the overall comparison is evident that GACAI-PCP obtain better contact predictors.

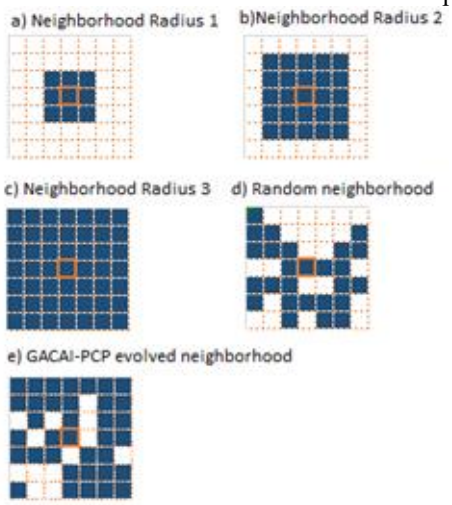

Fig. 4. Examples of Moore's CA neighborhood. a) Radius one Moore's CA neighborhood. b) Radius two Moore's CA neighborhood. c) Radius three Moore's CA neighborhood. d) Random neighborhood. e) Blue cells indicate cells that affects CA's transitions. The cell with bold border is the one that is updated by the CA evolution. White cells have no effect in the CA transitions.

Target protein T0862 is the exception where our identified CA is outperformed by the CA with Moore's neighborhood radius three. For eight target proteins, the CA with neighborhood radius three obtains the worst prediction result. In Fig. 6, we illustrate the differences of the five CAs in the comparison. To assess the performance of the predictors set, we used Friedman's test and Nemenyi's post-hoc test. Friedman's test compares the precision achieved for each predictor in each protein target and determines its average ranking. If the differences in rankings are significant, Friedman's test reports a small p-value and rejects the null hypothesis (there is no difference in performance). If the null hypothesis is rejected, is necessary to perform a post-hoc test to find out the predictors that perform better than others.

Nemenyi's post-hoc test compares all predictor pairs and allows us to identify those with significant difference in performance. For this comparison (39 targets and 5 predictors), by Nemenyi's test we conclude that predictors with rank differences greater than the critical difference (CD ≥ 0.9768), are significantly different and the one with the best ranking (lowest value) is the dominant in the set of target proteins. In Fig. 6, arrows start in the dominant predictor. Our evolved CA dominates all others, which allows to assume that our approach gets better CAs than those defined with traditional neighborhoods. The radius three CA surpassed only the radius two CA.

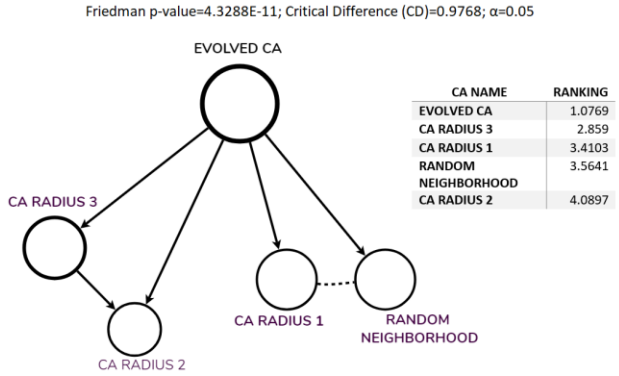

Fig. 6. Dominance graph analysis with statistical significance comparison by Friedman's test and post-hoc Nemenyi's test.

IV. Conclusions

We proposed a framework based on GAs to identify CAs that we used to improve protein contact maps prediction. GACAI-PCP identified a CA that exceled three obvious CAs and random CA, providing evidence that our approach identifies optimal CAs for PCP.

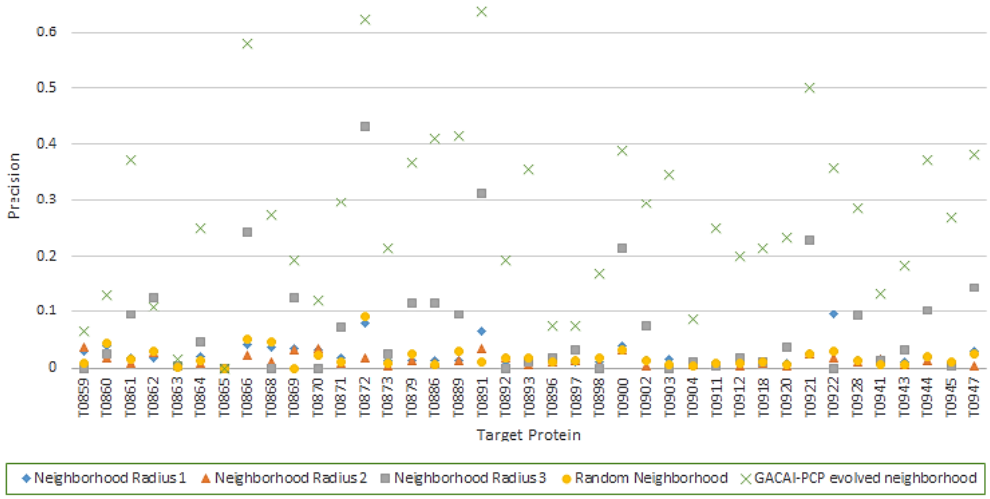

Fig. 5. Performance of the five CAs that implement on the CASP12 evaluation dataset.

We tuned the GA parameters using DCT as a problem that shares similar specifications to PCP. This allowed us to try several configurations and design options in a computationally less expensive setting.

In protein tertiary structure determination PCP is currently a habitual step. We proved that CAs can help in PCP. Our approach identifies CAs that are suited for the search of feasible local arrangement of contacts. The neighborhoods of the CAs for PCP showed an irregular shape that was found by our machine learning based approach. The several irregular neighborhood shapes capture the relationships which have effect in the identification of real contacts from false positives.

ACKNOWLEDGMENT

This work was partially supported by COLCIENCIAS PhD. Scholarship (Call 567 – 2012).

V. REFERENCES

[1] S. Tanaka and H. a Scheraga, "Model of protein folding: inclusion of short-, medium-, and long-range interactions.," Proc. Natl. Acad. Sci. U. S. A., vol. 72, no. 10, pp. 3802–6, 1975.

[2] B. Monastyrskyy, D. D'Andrea, K. Fidelis, A. Tramontano, and A. Kryshtafovych, "Evaluation of residue-residue contact prediction in CASP10," Proteins Struct. Funct. Bioinforma., vol. 82, no. SUPPL.2, pp. 138–153, 2014.

[3] B. Monastyrskyy, D. D'Andrea, K. Fidelis, A. Tramontano, and A. Kryshtafovych, "New encouraging developments in contact prediction: Assessment of the CASP11 results.," Proteins, no. October, pp. 1–14, 2015.

[4] B. Adhikari and J. Cheng, "Protein Residue Contacts and Prediction Methods," in Data Mining Techniques for the Life Sciences, vol. 1415, O. Carugo and F. Eisenhaber, Eds. Humana Press, New York, NY, 2016, pp. 463–476.

[5] C. Zhang, S. M. Mortuza, B. He, Y. Wang, and Y. Zhang, "Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12," Proteins Struct. Funct. Bioinforma., vol. 86, pp. 136–151, 2018.

[6] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP)—Round XII," Proteins Struct. Funct. Bioinforma., vol. 86, pp. 7–15, 2018.

[7] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, "Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model," PLoS Comput. Biol., vol. 13, no. 1, pp. 1–34, 2017.

[8] W. Bolt, J. M. Baetens, and B. De Baets, "An evolutionary approach to the identification of Cellular Automata based on partial observations," 2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc., pp. 2966–2972, 2015.

[9] A. Bateman et al., "UniProt: The universal protein knowledgebase," Nucleic Acids Res., vol. 45, no. D1, pp. D158–D169, 2017.

[10] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, and C. H. Wu, "UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches," Bioinformatics, vol. 31, no. 6, pp. 926–932, 2015.

[11] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment," Nat. Methods, vol. 9, no. 2, pp. 173–175, 2012.

[12] L. S. Johnson, S. R. Eddy, and E. Portugaly, "Hidden Markov model speed heuristic and iterative HMM search procedure," BMC Bioinformatics, vol. 11, no. 431, pp. 1–8, 2010.

[13] D. T. Jones, D. W. a Buchan, D. Cozzetto, and M. Pontil, "PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments.," Bioinformatics, vol. 28, no. 2, pp. 184–90, Jan. 2012.

[14] S. El-Gebali et al., "The Pfam protein families database in 2019," Nucleic Acids Res., vol. 47, no. October 2018, pp. 427–432, 2018.

[15] F. Simkovic, J. M. H. Thomas, and D. J. Rigden, "ConKit: A python interface to contact predictions," Bioinformatics, vol. 33, no. 14, pp. 2209–2211, 2017.

[16] N. Díaz and I. Tischer, "Generic framework for mining cellular automata models on protein-folding simulations," Genet. Mol. Res., vol. 15, no. 2, pp. 1–16, 2016.

[17] M. Rocklin, "Dask: Parallel Computation with Blocked algorithms and Task Scheduling," in SciPy in Science, 2015, no. 14, pp. 130–136.

[18] N. Diaz and I. Tischer, "Mining Stochastic Cellular Automata to Solve Density Classification Task in Two Dimensions," 2019.