

Towards a Distributed Framework for Protein (Structure) Comparison, Knowledge, Similarity and Information (ProCKSI)

Gianlugi Folino¹, Azhar A. Shah², Daniel Barthel², and Natalio Krasnogor²

¹CNR-ICAR, Institute of High Performance Computing and Networking, Italy

²School of Computer Science, University of Nottingham, UK

Abstract

Protein structure comparison is an essential component of almost all structural proteomics activities such as structure prediction, classification and functional analysis which are key milestones in the road towards innovations in modern drug discovery and medicine. Many tools and methods have been developed to investigate the (dis)similarities among a given set of protein structures [1]. However, there is no agreement on how to optimally *define* what similarity/distance means as different definitions focus on different biological criterion such as sequence or structural relatedness, evolutionary relationships, chemical functions or biological roles. This observation calls for an explicit identification and understating of the various stages involved in the derivation of similarity/distance assessment as illustrated in Figure 1. The first four stages that have dominated the research in protein structure comparison so far are: similarity conception, model building, mathematical definition and method implementation. Interestingly, the fifth stage, where one would seek to leverage the strength of a variety of methods by using appropriate consensus and ensemble mechanisms has barely been investigated. One such approach has recently been introduced by means of the *Protein (Structure) Comparison, Knowledge, Similarity and Information* (ProCKSI) web server [2]. Using a set of modern optimization and decision making techniques including metaheuristics, ProCKSI automatically integrates the operation of a number of methods and provides consensus based results for the comparison, clustering, analysis and visualization of multiple protein structures through a simple unified web interface.

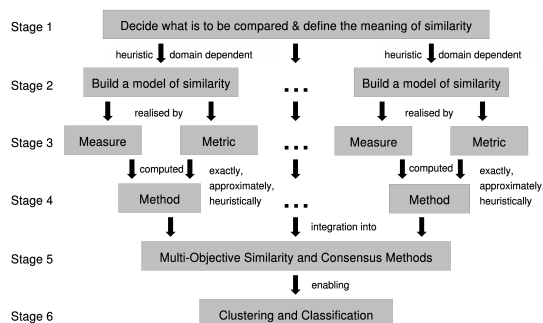


Figure 1: Stages in the derivation of a protein's classification: (1) Decide what "similarity" means, which, by nature, is a declarative and domain specific step. (2) Heuristically build a model of similarity based on l . This new similarity/distance conception will have its own bias, variance and outliers. (3) Decide whether this idealised model will be instantiated as a distance/similarity measure or metric. (4) One or more algorithms are implemented in order to calculate 3, which can be solved exactly and in polynomial time only in the simplest of cases. The more interesting similarity definitions, however, give rise to complex problems requiring heuristics/approximate algorithms for their solution. (5) Combining many different methods with different views of similarity produces a multi-competence pareto-front, from which a consensus picture might be derived. In turn, this allows the structural biologist to (6) cluster and classify proteins (more) reliably.

The integration of multiple methods for protein structure comparison in ProCKSI coupled with rapidly growing number of 3D structures in the Protein Data Bank (PDB) demands for compute and data storage resources that are far beyond the dimensions of single standard workstations. For example, a dataset with 1000 protein structures requires up to several days of computation and occupies several tens of mega bytes of main memory on a standard Pentium-4 uniprocessor machine when similarities are assessed by means of only one structure comparison method such as, e.g. MaxCMO-MSVNS [3]. What is more, for a consensus calculation from multiple similarity comparison methods with a (potentially very large) protein dataset (as demonstrated in [2]), a distributed framework becomes unavoidable in order to cope with vast amount of calculations. Ideally, an e-Science based infrastructure capable of performing m(b)illions of pairwise comparisons in parallel using cutting-edge *grid-styled* distributed computing technologies and services should be pursued.

In this paper, a distributed implementation was adopted for efficiently executing the different comparison methods used in ProCKSI including MaxCMO, USM, DaliLite, CE, TMAIalign, and FAST (see [2] for details).

The system is able to run both on a parallel environment using the MPI libraries and on a grid computing environment using the MPICH-G2 libraries [4].

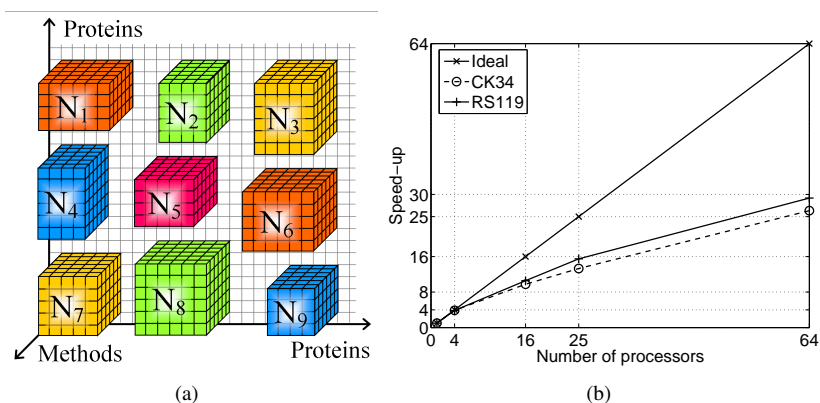


Figure 2: a) Distribution of the problem space (proteins x proteins x methods). Note that the different sizes take different protein sizes into account (e.g. one node only for a few big proteins, which take quite long to calculate; and one node for many smaller proteins, which are quicker to calculate). b) Speedup of the distributed implementation using the CK34 and RS119 datasets.

First experiments were performed on a Linux cluster with 64 dual-processors Itanium2 1.4GHz nodes each having 4 GB of main memory and being connected by a Qsnet high performance network. The performance of the parallel/distributed algorithm was measured in terms of *speedup*, i.e. the ratio between the time T_s taken by the best sequential implementation of an application measured on one processor and the execution time T_p taken by the same application running on p processors. In an ideal case, it should be equal to the number of processors. The problem space for pairwise similarity comparison with multiple methods ($proteins \times proteins \times methods$) has been decomposed along the first two dimensions (Figure 2 a), which has a twofold advantage if the number of proteins is sufficiently large: first, it reduces the execution time by an ideal factor of p balancing the methods having different execution times. Second, it reduces the memory needed by a factor of p (for the portion of matrix to be stored in a node) and $\frac{p}{2}$ (number of proteins for node).

Dataset	# Res. per Datasets	# Res. per Chain	USM	FAST	TM-Align	Dali	CE	MaxCMO
CK34	6102	179	0.52 ± 0.28	0.14 ± 0.07	0.28 ± 0.11	3.49 ± 1.53	3.20 ± 0.66	0.99 ± 0.34
RS119	23053	197	3.68 ± 0.31	2.16 ± 1.05	5.78 ± 2.86	44.59 ± 20.51	41.05 ± 20.41	20.13 ± 9.69

Table 1: Total number of residues and average number of chains per dataset and average execution times and standard deviation (minutes) of the different methods for the CK34 and RS119 datasets.

In our experiments, we used the first chain of the first model both for the Rost and Sander dataset (RS119) and for the Chew-Kedem (CK34) data set (see Table 1). For the RS119 (CK34) dataset, the entire execution time was reduced from about 6 days (6.2 hours), using the sequential implementation in one machine, to 4.8 hours (14.15 minutes) on 64 processors. For both datasets, the speed-up remains good using up to 16 processors, but using more processors does not help to speed up the total execution time as much as before (Figure 2 b). This is due to the structural differences of the proteins, as showed by the large variance in the execution times of the different methods (Table 1). Larger datasets should lessen this effect. Future works comprises a more extensive experimentation using very large datasets and the development of a distributed way of clustering and computing consensus.

References

- [1] R. Kolodny, P. Koehl, and M. Levitt, "Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures," *J Mol Biol*, vol. 346, pp. 1173–1188, 2005.
- [2] D. Barthel, J. Hirst, J. Blazewicz, E. K. Burke, and N. Krasnogor, "The ProCKSI server: a decision support system for protein (structure) comparison, knowledge, similarity and information," *BMC Bioinformatics*, vol. 8, p. 416, 2007.
- [3] D. A. Pelta, J. R. Gonzalez, and M. V. M., "A simple and fast heuristic for protein structure comparison," *BMC Bioinformatics*, vol. 9, p. 161, 2008.
- [4] N. T. Karonis, B. Toonen, and I. Foster, "Mpich-g2: a grid-enabled implementation of the message passing interface," *Journal of Parallel and Distributed Computing*, vol. 63, no. 5, pp. 551–563, 2003.