RAIRO-Oper. Res. 50 (2016) 363–374
DOI: 10.1051/ro/2015040

RAIRO Operations Research
www.rairo-ro.org

DOMGEN-GRAPH BASED METHOD FOR PROTEIN DOMAIN DELINEATION*

Maciej Milostan[1] and Piotr Lukasiak[1,2]

Abstract. The role of a protein depends heavily on its 3D shape, which is composed of semi-independent three-dimensional blocks called domains. Domains fold independently and constitute units of evolution. Most proteins contain multiple domains that are associated with a particular functions; moreover, the same domain can be found in different proteins. Automated recognition of domains can make prediction of proteins function easier and can support the analysis of proteins. Here, we propose a novel algorithm designed for domain recognition by identification of domain boundaries in the protein structure. The proposed algorithm uses a contact graph and an iterative approach to find meaningful clusters corresponding to the protein domains. The distinctive feature of the method is its effective complexity, that improves over other well-known methods, while holding a comparable level of correct domain assignments.

Mathematics Subject Classification. 68R10, 92-08.

Received September 8, 2015. Accepted September 21, 2015.

1. Introduction

The analysis of protein structure has created a fundamental challenge for many scientific institutes as well as for pharmaceutical companies. The progress in that area drives the development of new drugs and leads scientists to better understanding the machinery of life. The process of protein structure exploration is successfully supported by various computational techniques developed in operation research area [27]. Those techniques take into account various aspects of the protein universe, *e.g.* secondary structure prediction [5–7], tertiary structure prediction [4,44], protein function [8,9,30,35,45], or domain recognition [12,23,42]. One of the most important steps in a process of protein analysis is the prediction of its three-dimensional structure, because the shape is of crucial importance for identification of the function of particular protein [21]. Being more precise, the function of protein is usually constituted by semi-independent three-dimensional folds of proteins substructure, that may fold independently, called *domains*. Structural domains are regions that are either compact, globular modules, or are clearly distinguished from flanking regions of proteins [22]. Most proteins contain multiple structural domains and automatic recognition of that independent units can significantly improve the protein function prediction process.

Keywords. Graph theory, computational biology, protein structure.

* *The research has been supported by grant No. 2012/05/B/ST6/03026 from the National Science Centre, Poland.*

[1] Poznan University of Technology, Institute of Computing Science, ul. Piotrowo 2, 60-965 Poznan, Poland.
mmilostan@cs.put.poznan.pl; plukasiak@cs.put.poznan.pl

[2] Institute of Bioorganic Chemistry, Polish Academy of Sciences,ul. Noskowskiego 12/14, 61-704 Poznan, Poland

Nowadays, protein domains are classified using various methods and gathered in publicly available databases [17]. The most popular databases are SCOP [1, 28] (developed as an evolutionary classification, in which the main focus is to place the proteins in a coherent evolutionary framework, based on their conserved structural features), CATH [32] (hierarchical classification of domains into sequence and structure-based families and fold groups according to significant sequence similarity), FSSP (classification based on structure-structure alignment of proteins) [20] and DALI [18] (classification based on exhaustive all-against-all 3D structure comparison of protein structures currently deposited in the Protein Data Bank [3]).

Several computational methods to predict domains in proteins based on protein structural coordinates have been introduced, namely DOMARK [36], DETECTIVE [37], DomainParser [16], STRUDL [41], PUU [20]. The applied techniques are based mainly on the premise that the amino acids within a domain will make more internal (intra-domain) contacts than external (inter-domain) contacts. Other methods are based on graph theory − *e.g.* DomainParser, that employs Ford−Fulkerson algorithm [13] in a recursive way to partition the graph into semi-independent units, using neural networks to test the quality of domains and guide the partition process.

Sequence information is often insufficient to identify the structural domains in the protein, because the same structure can be reached from widely divergent sequence space (typically down to 30% sequence identity). Sequence based method has been presented in [2,26]. Therefore, knowledge of protein structure is often the only criterion to recognize structural domains. Although the problem has been risen more than 30 years ago, it is not completely resolved as of today.

In this paper we analyze only the problem of predicting domain boundaries from 3D protein structure. To our knowledge, currently existed approaches are not efficient for domain prediction process. To fill that gap we propose DomGen, a method which allows us to obtain similar results to the previous ones using less complex heuristics and simpler criteria. In our study, the idea of graph clustering [34] algorithms used for decomposition of protein contact graphs has been applied.

2. Problem definition

Domains are usually stabilized by specific set of interactions (*e.g.* chemical bonds or compact amino acid packings enforced by solvent) among amino acids. Thus, the basic step necessary to identify protein domains is the recognition of these interatomic interactions or spatial contacts between atoms in the protein under investigation. Unfortunately, such contacts could also appear between separate domains, what makes the problem harder to solve. It should be also mentioned that each protein domain consists of one or more continuous amino acid subsequences − called segments, but in the nature, discontinuous domains also exist. Discontinuous domain consist of multiple segments divided by at least one segment or subsequence not belonging to the considered domain. During the analysis of domains from the biological point of view, one should take into consideration additional information coming from so called secondary structure of protein. Secondary structure is a locally ordered structure brought about via hydrogen bonding mainly within the peptide backbone. The most common secondary structure elements in proteins are helices and strands.

As input of the DomGen approach the tertiary structure of protein (Figs. 1a) is provided; as output, domains assignments are obtained (Figs. 1b and 1c).

Although it is hard to define domain as a formal entity, it is possible to provide some basic features of the valid domain. According to literature [20, 42] a domain should have following properties:

(1) should have at least 40 residues (amino acids);
(2) in general β-strands should not be cut too frequently − at most one β-strand can be cut at the *interface* between two domains and a β-strand having more than 2 residues in each strand can belong to one domain only;

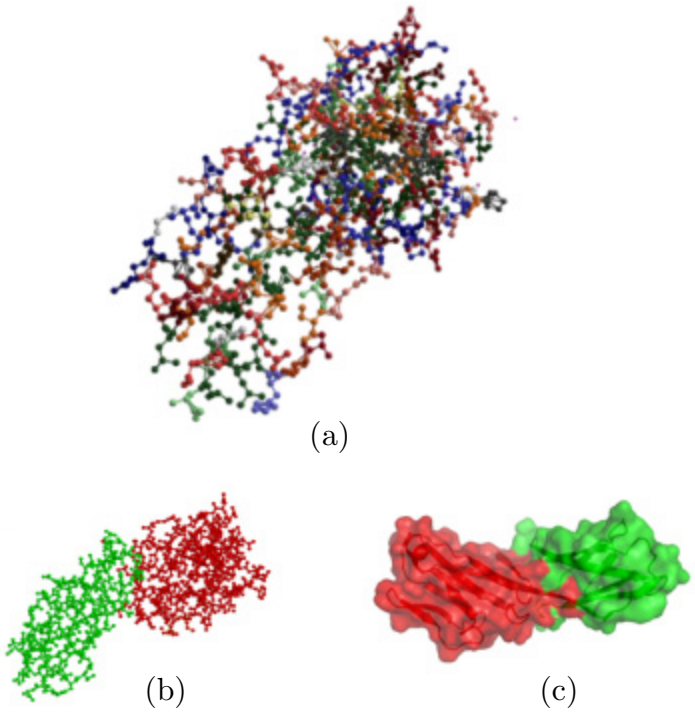

FIGURE 1. Structure of two-domain protein from PDB (id:3CD4) without domains highlighted (a); two domains of the same protein highlighted based on SCOP classification in atom view (b) and cartoon view (c).

(3) must be compact enough to satisfy the following condition [19]:

$$\frac{\sum_{i,j} p_{i,j}}{n_a} \geq g_m, \tag{2.1}$$

where i and j are any two atoms belonging to the domain separated by at least three other amino acids along the sequence; $p_{i,j} = 1$ if the distance between i and j is 4.0 Å or less, otherwise $p_{i,j} = 0$; n_a is the number of atoms in the domain; g_m is some threshold determined from known domains (*e.g.* $g_m = 0.54$).

(4) the interface between domains must be small enough, such that the number of intra-domain contacts or interactions between amino acids composing domain should be much larger than the number of inter-domain contacts that these amino acids have with other amino acids not belonging to the domain.

(5) the number of segments in a domain – D, should not be to great. More formally, the following condition should be fulfilled:

$$\frac{r(D)}{s(D)} \geq l_s \tag{2.2}$$

where $r(D)$ and $s(D)$ are the numbers of amino acids and segments in domain D respectively; l_s is some threshold determined from known domains (*e.g.* $l_s = 35$).

Most of the above conditions are not used directly in the method outlined below, but they could be used for validating its results.

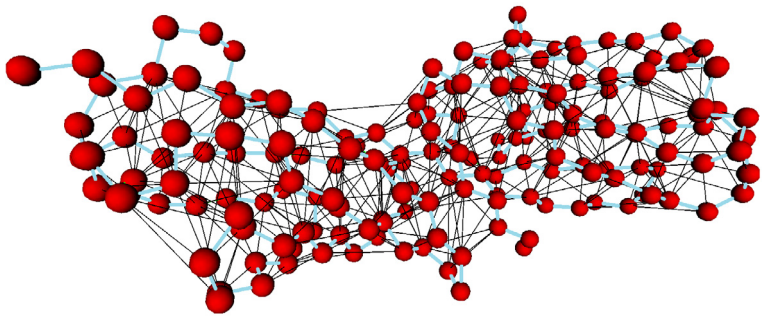

FIGURE 2. Graph representation of contacts between amino acids in protein; spheres correspond to amino acids, black lines denote contacts between particular amino acids, blue lines denote covalent bonds between consecutive amino acids in sequence.

3. Method description

Models based on graphs are very often used to solve biological problems, *e.g.* analysis of DNA [4], protein [25] and RNA [38] as well as for virus analysis [40]. Problems based on graphs are often computationally efficient, easy to understand and can be solved by powerful GPU computing [14, 15]. The key element in finding the solution to a particular biologically related graph problem is its correct transformation into a suited graph [39].

In order to solve the considered problem one has to use some mathematical abstraction to represent protein structure. The most straightforward approach is to represent protein structure as a graph of contacts (see Fig. 2). Each residue in protein chain is converted into a vertex in the graph, and each contact is presented as an edge in this graph. Additionally, a weight w can be assigned to an edge to denote the contact strength. In DomGen, the default weight of an edge is equal to 1 (if not given explicitly). Such a graph can be used to recognize potential domains by means of graph clustering, graph partitioning, or min-cut algorithms. Graph clustering methods [34] that have been applied include traditional graph partitioning algorithms based on max-flow/min-cut paradigm (as in Ford–Fulkerson algorithm [13]) or minimal spanning trees [43], as well as more complex methods such as spectral methods [29], kernel-based clustering [10], divide and merge strategies [46], random walks based methods, including Markov clustering [11, 33].

We now describe DomGen, a novel, parameterized, iterative algorithm based on the contacts between amino acids in protein. It incorporates information about secondary structures of proteins in the process of splitting a graph into clusters that corresponds to domains.

DomGen uses the definition of contact similar to the one proposed by Daniluk [9] to delineate protein domains. This definition of a contact is based on the following assumptions:

- each amino acid is represented as two points in three-dimensional space: the first one corresponds to C_α atom and the second one corresponds to the geometrical center of the side chain. For given position i in protein chain let us denote these points respectively as ${C_\alpha}^i$ and S^i;
- amino acids in the positions i and j in a given protein chain are in contact if any of the following conditions is fulfilled:
  (1) $\left\|S^i - S^j\right\| \leq r$Å,
  (2) rÅ $< \left\|S^i - S^j\right\| \leq r + 1.5$Å and $\left\|S^i - S^j\right\| < \left\|{C_\alpha}^i - {C_\alpha}^j\right\| - 0.75$Å, where r is method parameter – distance threshold (*e.g.* $r = 4.5$Å).

The visualization of a contact graph for a protein (PDB id:3CD4) is shown in Figure 2.

Existing efficient clustering methods need prior knowledge of a number of clusters, so to use these methods one has to know *a priori* the number of domains. To overcome this problem, the idea of coloring the structure iteratively using simple rules (see Fig. 3) has been proposed. Colors with sufficient coverage should point to

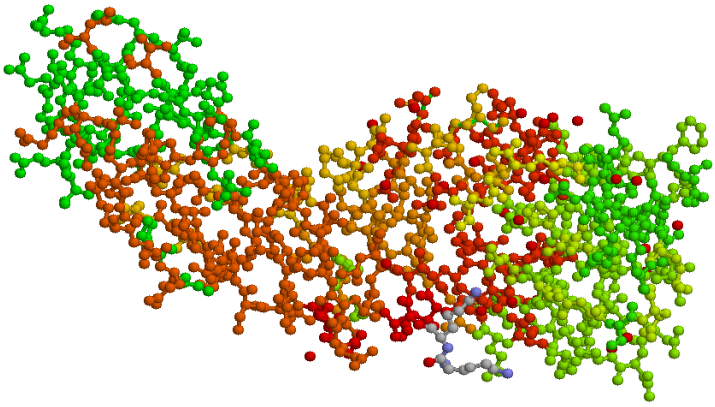

FIGURE 3. Initial coloring (clustering) of the protein structure based on contacts (different colors correspond to different clusters).

a core elements of the protein domains and give an estimated number of clusters. Moreover, it is possible to incorporate information about secondary structures into clustering strategy (protein secondary structure can be generated using DSSP algorithm [24]).

Our algorithm contains two main phases: graph construction phase and graph clustering phase. Clusters in a graph correspond to domains.

A cluster D_i in a given graph G is an induced subgraph that contains all vertices of a particular color c_i assigned by the clustering algorithm. Each vertex from G can belong to one cluster only.

The DomGen algorithm is described (Algorithm 1).

Clusters D_i and D_j are considered as *neighboring* in graph G if there exists an edge $e = \{v_i, v_j\}$ connecting two vertices $v_i \in D_i$ and $v_j \in D_j$. Let us denote $links(D_i, D_j)$ to be the sum of the edge weights w between nodes in subgraph D_i and subgraph D_j. In other words:

$$links(D_i, D - j) = \sum_{i \in D_i, j \in D_j} w\{i, j\}, \tag{3.1}$$

where $w\{i, j\}$ – is the weight of the edge $e = \{i, j\}$.

Also, let us assume that there are no loops in the considered graph G. This condition is enforced by the procedure constructing a contact graph.

As a result of the clustering stage of the Algorithm 1, one gets small, potentially very stable elements of the protein structure. These small elements can be treated as the potential seeds of the domain cores (Fig. 4). These seeds have then to be merged into larger entities to compose the domains. It is important to notice that during clustering, only contacts between nonconsecutive amino acids are considered. The exclusion of the contacts between consecutive amino acids make globally stable substructures more preferred than local motifs. The procedure assigning initial colors can assign colors misleadingly.

To avoid wrong clusters, a cluster quality measure q_i has been introduced:

$$q_i = \frac{\sum_{j \neq i} links(D_i, D_j)}{links(D_i, D_i)}. \tag{3.2}$$

Let t_{merge} be the threshold for merging clusters, and $t_{invalidate}$ the threshold for invalidating clusters. If the $q_i > t_{invalidate}$ then cluster i becomes invalid and all its vertices are "grey colored", and they are returned to the set of unassigned vertices. In the refinement procedure such vertices can be reassigned to the proper clusters.

Algorithm 1. Pseudocode of DomGen basic algorithm.

Require: Protein structure P, number v of contacts for amino acid to assign new color
Generates: Division of P into domains

```
G ← contacts(P)
//contacts(P) – generates graph of contacts
assign grey color to all vertices in G
(optional) assign colors to known secondary structures taking into account special presumptions about β-strands
L ← sort vertices V in G by their degree d (descending)
while (L is not empty) and (maximal degree of the vertex V' ∈ L > v) do
    pick the first element from L → l
    if l has color then
        assign l's color to all grey colored vertices in contact with l that are not consecutive to l in amino acid sequence
    else
        check the color of all neighbors
        if the most popular color among neighbors = grey then
            assign new color to l
        else
            assign to l the most popular color
        end if
    end if
    assign current color of l to all grey colored vertices in contact with l that are not consecutive to l in amino acid sequence
end while
G ← merge(G); //iterate through the graph G and merge clusters that have large cross-color contacts
G ← refine(G); //refine clusters
print the number of colors //(in ideal case the number of colors should correspond to number of domains)
print all vertices in G with assigned colors
```

Next, our algorithm merges small clusters into larger domains. The weights of the edges between consecutive amino acids belonging to different clusters are enlarged to some value w_c to increase the probability of merging clusters with large number of backbone contacts.

Two clusters D_i and D_j in G, that in fact are potential fragments of domains, can be merged when the following conditions are fulfilled:

- D_i and D_j $(i \neq j)$ are neighboring cluster in G,
- the ratio m given by the following equation:

$$m_{i,j} = \frac{links(D_i, D_j)}{links(D_i, D_i)} \tag{3.3}$$

  is greater than threshold t_{merge} set as a parameter.

This ratio is computed using an asymmetric function, so it cannot be treated as a measure of distance between the clusters. However it is sufficient to judge if clusters should be merged. The values of the ratio $m_{i,j}$ for the whole contact graph can be represented as the matrix M of the size $n \times n$, where n is the number of clusters. If for given cluster i exists more than one cluster that fulfills the above conditions then the cluster i is merged into cluster j, such that $m_{i,j} = \max_j(m_{i,j})$. The clusters are analyzed in descending order of $m_{i,j}$ values. If, for a given cluster i, exists more than one cluster that fulfills the above conditions, then the cluster i is merged into cluster j, so that $m_{i,j} = \max_j(m_{i,j})$ (i is fixed, so it is omitted in the max function). After merging the clusters according to matrix M, m ratios for a smaller number of clusters will be recalculated. This step is repeated until all m ratios in matrix M will be lower than t_{merge} threshold.

The detailed description of steps of merge(G) procedure are shown in pseudo-code as Algorithm 2. In the refinement step of Algorithm 1, elements from cluster i that are surrounded by elements from cluster j in the

Algorithm 2. Pseudocode of procedure merge(G).

for each edge $e = \{i, i+1\} \in G$, such that $i \in D_i$ and $i+1 \in D_j$ **do**
  $w\{i, i+1\} := w\{i, i+1\} + w_c$
  $//w_c$ method parameter
end for
for each color c_i **do**
  compute q_i
end for
for all i such that $q_i > t_{invalidate}$ **do**
  assign grey color to all $v_i \in D_i$
end for
for all pairs of c_i, c_j **do**
  compute $m_{i,j}$
end for
merge each pair of cluster D_i and D_j such that $m_{i,j} > t_{merge}$
if $\exists m_{i,j} > t_{merge}$ **then**
  change weights $w\{i, j\}$ to default values (*e.g.* equal to 1) and go to 1.
end if
return G

sequence are reassigned to cluster j. In Figure 4, an example of the results after *merge* and $refinement$ steps of the algorithm are shown. The improvement in assignment of domains (Fig. 4a) is noticeable.

For further improvement of the results quality, it is possible to incorporate knowledge about the secondary structures into the decomposition scheme. Assigning two different colors to two parallel fragments of strands can spoil the overall results, so special precautions must be incorporated. In case of tightly packed two interacting helices, the large number of geometrical contacts can appear although the chemical interactions could be much weaker. Overestimation of contacts can be misleading and can cause the merging of two individual domains together.

The algorithm has been tested on the data from SCOP [1,28] and CATH [31] databases. Both of these domain databases contain domains for protein structures deposited in PDB [3].

4. Complexity of the algorithm

The novel algorithm proposed above (see Algorithm 1) has polynomial complexity. The generation of contact graphs takes time $O(n^2)$, where n is the length of the amino acids sequence. The sorting of vertices has complexity $O(n \log(n))$, the generation of initial colors takes $O(n^2)$ in the worst case. The complexity of the merge procedure is $O(n^3)$ (to be precise it is $O(n^2 m)$, where m is the number of colors). The number m is not greater than $n/2$. The computation of the $links(D_i, D_j)$ takes $O(n^2)$ for all i, j in total. Computation of the quality measures q_i for all i takes additional $O(m)$ time assuming that $links(D_i, D_j)$ values have been computed. The $m_{i,j}$ ratios can be computed in $O(m^2)$ time. The merging procedure is iterated no more the $\frac{n}{2} - 1$ times, so it gives the overall complexity $O(n^3)$.

The complexity of the DomGen algorithm is very similar to the complexity of other popular graph clustering algorithms. Walktrap [33] algorithm has time complexity $O(kn^2)$, where k is the number of edges; MCL [11] has complexity $O(nl^2)$, where n is the number of nodes in the graph, and l is the number of resources allocated per node, for extremely tight and dense graphs this might become $O(n^2 \log(l))$. The MCL seems to have the best performance but it is not straightforward to incorporate additional biological information such as secondary structures or specific conditions, which is included in DomGen (*e.g.* in cluster merge procedure).

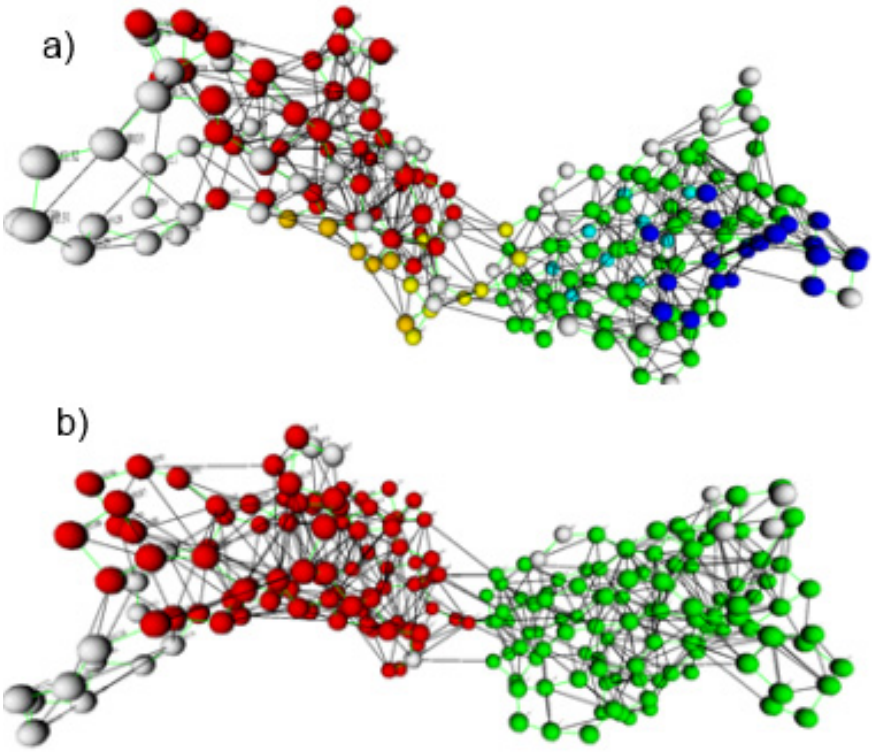

FIGURE 4. Domain assignments before merge and refinement procedures) of the Algorithm 1 for chain H of the protein with PDB id: 12e8 (a); Domain assignments after merge and refinement (b).

5. Parameter tuning in DomGen

In DomGen there are various parameters that have to be properly set up to give the best results. These parameters are:

- t_{merge} threshold for merging clusters,
- $t_{invalidate}$ threshold for invalidating clusters,
- w_c the weight increment and v the minimum contacts parameter.

The default value of the v parameter is 1, but it can be risen even up to 12 (more stable fragments of a structure are analyzed). As it has been presented in Figure 5 (based on 14 347 109 amino acids from proteins available in SCOP, CATH and DALI domain databases), the distribution of contacts is not a normal one and in most cases the amino acids have around 5 neighbors. In practice, it means that, if parameter v is set around 5, then the initial colors in the algorithm will be assigned to almost all amino acids. This information can be used to set up w_c parameter as well. For the initial tests let us assume that $w_c = 5$. The $t_{invalidate}$ is by default set up to 1 (the cluster is invalid if the sum of weights of external contacts is greater than a number of internal contacts). The parameter t_{merge} was determined experimentally, and for the purposes of initial validation it has been set to $t_{merge} = 0.41$.

A simple adjustment of these parameters leads to the correct domain assignments for the majority of the analyzed structures. Although adjustment of parameter is possible, in our approach we have used the same set of parameters after little tuning (pointed out above). The robustness of the results with respect to parameters

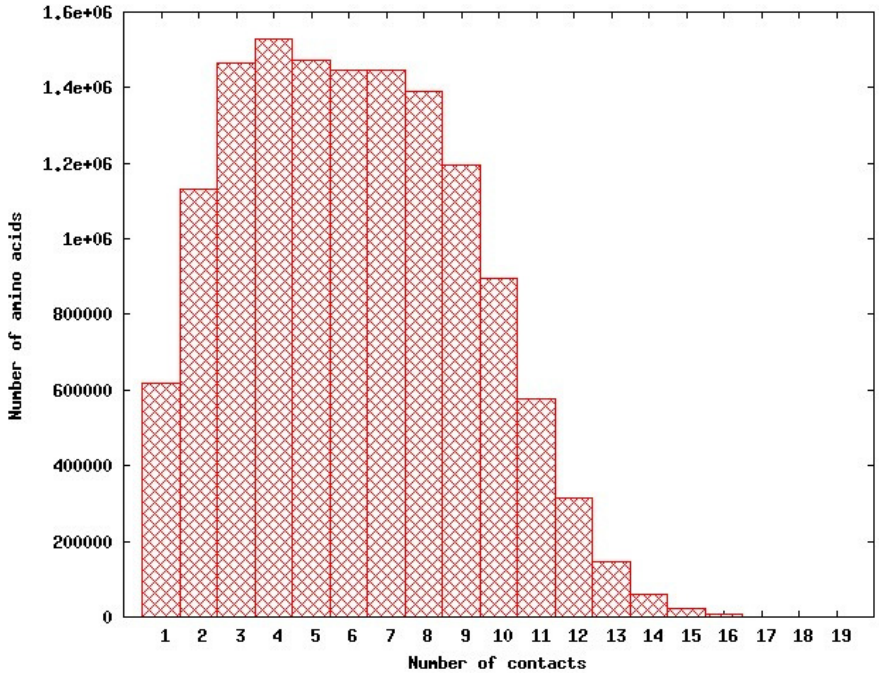

FIGURE 5. Distribution of contacts: numbers of amino acids with particular numbers of contacts.

adjustments can be consider as a strength of DomGen, where other methods appear to be very sensible to small parameters adjustments.

6. Comparison with the other methods

The comparison of the performances of DomGen algorithm with those of the other methods considered from the literature (DomainParser [20], Consensus approach [23], and SCOP is presented (Tab. 1). The DomainParser algorithm uses a network flow algorithm and the definition of contacts based on distances between the closest atoms of the amino acids. The Consensus method proposed by Jones uses results of other methods to generate the prediction. Results of the SCOP classification is also presented for the comparison. We have tested the performance of DomGen on a set of 55 proteins provided by Xu [42]. Among 55 proteins, 30 are single-domain proteins, 20 are two domain proteins, 3 are three-domain proteins, and 2 are four domain proteins. In the analyzed benchmark, domain decomposition is treated as correct if the number of decomposed domains is the same as in the literature (based on structure manual inspections), and the residue assignments is at least 85% in agreement with the structure [23]. Results presented in Table 1 consist of results for multi-domain proteins only (domains in one-domain proteins have been assigned correctly). We do not provide the validation of our algorithm against all SCOP domains, because the domain assignments tend to differ significantly between SCOP and other databases or methods [16], as well as those given by human experts. As it can be noticed, the level of domain recognition is at least on the same level as reported by other applied methods (78.2% – the best reported result for the considered data set [42]), and for the chosen set of multi-domain proteins is around 78.6%. The precision for two-domains proteins was 80.7%, for three-domains proteins 67.7%, and for four-domains proteins 82.3%.

TABLE 1. Test structures for domain splitting algorithm from [42] and [23] ('/' is used to separate domains).

Protein (PDB ID)	Consensus approach	DomainParser	DomGen	SCOP	DomGen *vs.* Scop (agreement of assignments)
1ezm	1-134/135-298	1-133/134-298	1-145/146-298	1 domain	51.01%
1fnr /1fnb	19-161/162-314	19-152/153-314	1-154/155-314	19-154/155-314	93.95%
1gpb	19-489/490-841	19-63/64-484;828-841/558-648;712-792/485-557;649-711;793-827	19-61;93-126/62-92;127-195;812-841/195-811	1 domain	73.25%
1lap	1-150/171-484	1-173/174-484	1-159/160-484	1-159/160-484	100.00%
1pfk_A	0-138;251-301/139-250;302-319	0-141;254-319/142-253	1-141;254-305/142-253;306-318	1 domain	60.31%
1ppn	1-20;112-208/21-111;209-212	1 domain	1 domain	1 domain	100.00%
1rhd	1-158/159-293	1-63;74-157/64-73;158-293	1-157/158-293	1-149/150-293	97.27%
1sgt	22-123;234-245/129-233	1 domain	1-121;234-245/123-233	1 domain	55.10%
1vsga	1-29;92-251/42-75;266-362	1-32;86-255/33-85;256-362	1-37;81-262/38-80;263-358	1 domain	61.73%
1wsy_B / 1bks_B	9-52;86-204/53-85;205-393	90-189/9-89;190-393	1 domain	1 domain	100.00%
2cyp	3-145;266-294/164-265	2-144;273-294/145-272	2-141/141-294 domain	1 domain	52.04%
2had	1-155;230-310/156-229	1 domain	1-122;186-221;279-310/123-185;222-278	1domain	61.61%
3cd4	1-98/99-178	1-98/99-178	1-101/102-178	1-97/98-178	97.75%
3gap_A / 1g6n	1-129/139-208	1 domain	1-133/134-208	7-137/138-206	94.71%
3pgk	1-185;403-415/200-392	0-188;402-415/189-401	1-195;390-415/196-389	1 domain	53.49%
4gcr	1-83/84-174	1-83/84-174	1-85/86-174	1-85/86-174	100.00%
5fbp_A	6-201/202-335	1domain	6-202;270-317/203-269;317-335	1 domain	74.63%
8adh	1-175;319-374/176-318	1-173;321-374/174-320	1 domain	1-163,340-374/164-339	62.03%
8atc_A	1-137;288-310/144-283	1-130;292-310/131-291	1-138/139-310	1-150/151-310	95.81%
8atc_B	8-97/101-152	8-97/101-153	8-100/101-153	8-100/101-153	100.00%
1phh	1-175/176-290/291-340	32-124/180-268/1-31;125-179;269-394	1-41;113-177/42-112;178-275	1-173,276-394/174-275	70.05%
3grs	8-157;294-364/158-293/365-478	8-161;290-368/162-289/369-478	18-65,108-158,291-364/66-107;159-290;365-478	18-165,291-363/166-290/364-478	65.27%
1atn_A	1-32;70-144;338-32/33-69/145-180;270-337/181-269	0-33;97-147;337-372/34-96/148-180;273-336/181-272	1-185;256-372/186-255	1-146/147-372	58.06%
2pmg_A / 3pmg_A	1-188/192-315/325-403/408-561	1-188/189-303/304-406/407-561	1-181/182-297/297-424/425-561	1-190/191-303/304-420/421-561	96.43%
8acn	2-200/201-317/320-513/538-754	2-530/531-754	2-586/587-754	2-528/529-754	92.31%

The difficult cases for which the proposed approach fail correspond to multi-domains proteins of different sizes (in some cases the number of domains and their limits do not agree with those in CATH or SCOP). Analyzed protein architectures consist of smaller structural units, often containing contiguous chain segments with a completely different architecture. It is hard to recognize them as completely separated units, because they usually form shapes, which are tightly associated with the main architectural motif.

Various disagreements between our assignments and the ones in the literature could be the result of the lack of precise definition of a structural domain, as mentioned by experts [41]. The manual assignments by experts can be also quite subjective, depending on their own interpretation of protein domain definition. It is worth to mention that different experts may propose different domain assignments for the same protein. Another uncertainty is related with the assignment of the short segments. If a short segment is in and out of one domain, while most of its flanks are in another domain, it can be assigned to the domain of its flanks depending on the size of the segment.

7. Conclusions

We have developed a novel algorithm for splitting tertiary structure of proteins into domains. Proposed solution uses specially crafted contact graph and graph clustering method to detect potentially stable substructures, and has been tested on the data from SCOP and CATH databases. Domain assignment has been done automatically, and, based on the results obtained, one can noticed that the proposed algorithm gives at least comparable result to the other currently used methods, but with polynominal time complexity. DomGen can be easily applied for the considered problem and extended by additional analytical modules devoted to recognition of similarity of domains cores in proteins. It can be also applied in protein quality assessment, especially in blind tests like CASP experiments, where fast and efficient analysis is one of the crucial points. Our approach can successfully support domain recognition, delivering plausible results of the prediction, together with fast and efficient computational time.

References

[1] A. Andreeva, D. Howorth, S.E. Brenner, T.J.P. Hubbard, C. Chothia and A.G. Murzin, Scop database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res.* **32** (2004) D226–D229.

[2] M. Antczak, J. Blazewicz, P. Lukasiak, M. Milostan, N. Krasnogor and G. Palik, Domanspattern based method for protein domain boundaries prediction and analysis. *Found. Comput. Decision Sci.* **36** (2011) 99.

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

[4] J. Blazewicz and M. Kasprzak, Complexity issues in computational biology. *Fund. Inform.* **118** (2012) 385–401.

[5] J. Blazewicz, P.L. Hammer and P. Lukasiak, Predicting secondary structures of proteins. *IEEE Eng. Med. Biol. Mag.* **24** (2005) 88–94.

[6] J. Blazewicz, P. Lukasiak and M. Milostan, Application of tabu search strategy for finding low energy structure of protein. Computational Intelligence Techniques in Bioinformatics. *Artif. Intell. Med.* **35** (2005) 135–145.

[7] J. Blazewicz, P. Lukasiak and S. Wilk, New machine learning methods for prediction of protein secondary structures. *Control and Cybernet.* **36** (2007) 183–201.

[8] M. Brylinski and J. Skolnick, A threading-based method (findsite) for ligand-binding site prediction and functional annotation. *Proc. Natl. Acad. Sci. USA* **105** (2008) 129–34.

[9] P. Daniluk and B. Lesyng, A novel method to compare protein structures using local descriptors. *BMC Bioinform.* **12** (2011) 344.

[10] I. Dhillon, Y. Guan and B. Kulis, A fast kernel-based multilevel algorithm for graph clustering. In *Proc. of the eleventh ACM SIGKDD international conference on Knowledge discovery in Data Mining*. ACM (2005) 629–634.

[11] A.J. Enright, S. Van Dongen and C.A Ouzounis, An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30** (2002) 1575–1584.

[12] I. Ezkurdia, O. Grana, J.M.G. Izarzugaza and M.L. Tress, Assessment of domain boundary predictions and the prediction of intramolecular contacts in casp8. *Proteins: Structure, Function and Bioinform.* **77** (2009) 196–209.

[13] L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Vol. 1962. Princeton Princeton University Press (1962).

[14] W. Frohmberg, M. Kierzynka, J. Blazewicz and P. Wojciechowski, G-pas 2.0–an improved version of protein alignment tool with an efficient backtracking routine on multiple gpus. *Bull. Pol. Acad. Sci.: Tech. Sci.* **60** (2012) 491–494.

[15] W. Frohmberg, M. Kierzynka, J. Blazewicz, P. Gawron and P. Wojciechowski, G-dna–a highly efficient multi-gpu/mpi tool for aligning nucleotide reads. *Bull. Pol. Acad. Sci.: Tech. Sci.* **61** (2013) 989–992.

[16] J. Guo, D. Xu, D. Kim and Y. Xu, Improving the performance of domainparser for structural domain partition using neural network. *Nucleic Acids Res.* **31** (2003) 944–952.
[17] C. Hadley and D.T. Jones, A systematic comparison of protein structure classifications: Scop, cath and fssp. *Structure* **7** (1999) 1099–1112.
[18] L. Holm and C. Sander, Protein structure comparison by alignment of distance matrices. *J. Molec. Biol.* **233** (1993) 123–138.
[19] L. Holm and C. Sander, The fssp database of structurally aligned protein fold families. *Nucleic Acids Res.* **22** (1994) 3600–9.
[20] L. Holm and C. Sander, Parser for protein folding units. *Proteins: Structure, Function and Bioinform.* **19** (1994) 256–268.
[21] L. Holm and C. Sander, Mapping the protein universe. *Science* **273** (1996) 595–602.
[22] T.R. Hvidsten, A. Kryshtafovych, J. Komorowski and K. Fidelis, A novel approach to fold recognition using sequence-derived properties from sets of structurally similar local fragments of proteins. *Bioinform.* **19** (2003) ii81–ii91.
[23] S. Jones, M. Stewart, A. Michie, M.B. Swindells, C. Orengo and J.M. Thornton, Domain assignment for protein structures using a consensus approach: characterization and analysis. *Protein Sci.* **7** (1998) 233–242.
[24] W. Kabsch and C. Sander, Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22** (1983) 2577–2637.
[25] N. Krasnogor, A.A. Shah, D. Barthel, P. Lukasiak and J. Blazewicz, Web and grid technologies in bioinformatics, computational and systems biology: A review. *Curr. Bioinform.* **3** (2008) 10–31.
[26] J. Liu and B. Rost, Sequence-based prediction of protein domains. *Nucleic Acids Res.* **32** (2004) 3522–3530.
[27] P. Lukasiak, J. Blazewicz and M. Milostan, Some operations research methods for analyzing protein sequences and structures. *Ann. Oper. Res.* **175** (2010) 9–35.
[28] A.G. Murzin, S.E. Brenner, T. Hubbard and C. Chothia, Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **247** (1995) 536–540.
[29] M.C.V. Nascimento and A.C.P.L.F. De Carvalho, Spectral methods for graph clustering–a survey. *Eur. J. Oper. Res.* **211** (2011) 221–231.
[30] M. Oh, K. Joo and J. Lee, Protein-binding site prediction based on three-dimensional protein modeling. *Proteins: Structure, Function, and Bioinform.* **77** (2009) 152–156.
[31] F. Pearl, A. Todd, I. Sillitoe, M. Dibley, O. Redfern, T. Lewis, C. Bennett, R. Marsden, A. Grant, D. Lee, A. Akpor, M. Maibaum, A. Harrison, T. Dallman, G. Reeves, I. Diboun, S. Addou, S. Lise, C. Johnston, A. Sillero, J. Thornton and C. Orengo, The cath domain structure database and related resources gene3d and dhs provide comprehensive domain family information for genome analysis. *Nucleic Acids Res.* **33** (2004) D247–D251.
[32] F. Pearl, *et al.* The cath domain structure database and related resources gene3d and dhs provide comprehensive domain family information for genome analysis. *Nucleic Acids Res.* **33** (2005) D247–D251.
[33] P. Pons and M. Latapy, Computing communities in large networks using random walks. *J. Graph Algorithms Appl.* **10** (2006) 191–218.
[34] S.E. Schaeffer, Graph clustering. *Comput. Sci. Rev.* **1** (2007) 27–64.
[35] T. Schmidt, J. Haas, T.G. Cassarino and T. Schwede, Assessment of ligand-binding residue predictions in casp9. *Proteins: Structure, Function, and Bioinform.* **79** (2011) 126–136.
[36] A.S. Siddiqui and G.J. Barton, Continuous and discontinuous domains: an algorithm for the automatic generation of reliable protein domain definitions. *Protein Sci.* **4** (1995) 872–884.
[37] M.B. Swindells, A procedure for detecting structural domains in proteins. *Protein science: a Publication of the Protein Society* **4** (1995) 103.
[38] M. Szachniuk, C.M. De Cola, G. Felici and J. Blazewicz, The orderly colored longest path problem – a survey of applications and new algorithms. *RAIRO: RO* **48** (2014) 25–51.
[39] S. Vishveshwara, K.V. Brinda and N. Kannan, Protein structure: insights from graph theory. *J. Theoret. Comput. Chem.* **1** (2002) 187–211.
[40] S. Wasik, P. Jackowiak, M. Figlerowicz and J. Blazewicz, Multi-agent model of hepatitis c virus infection. *Art. Intell. Med.* **60** (2014) 123–131.
[41] L. Wernisch, M. Hunting and S.J. Wodak, Identification of structural domains in proteins by a graph heuristic. *Proteins: Structure, Function, and Bioinform.* **35** (1999) 338–352.
[42] Y. Xu, D. Xu and H.N. Gabow, Protein domain decomposition using a graph-theoretic approach. *Bioinform.* **16** (2000) 1091–1104.
[43] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.* **100** (1971) 68–86.
[44] Y. Zhang, I-tasser server for protein 3d structure prediction. *BMC Bioinform.* **9** (2008) 40.
[45] Y. Zhang, Protein structure prediction: when is it useful? *Curr. Opin. Struct. Biol.* **19** (2009) 145–155.
[46] C. Zhong, D. Miao and P. Franti, Minimum spanning tree based split-and-merge: A hierarchical clustering method. *Inform. Sci.* **181** (2011) 3397–3410.