MATCH
Communications in Mathematical
and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **71** (2014) 445-456

ISSN 0340 - 6253

A New Protein Domain Assignment Algorithm Based on the Dominating Set of a Graph

C. Eslahchi[1,*], E. S. Ansari[2,3]

[1]*Department of Computer Science, Shahid Beheshti University, G.C., Tehran, Iran*

[2]*Department of Bioinformatics, Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran*

[3] *School of Computer Science, Institute for Research in Fundamental Science (IPM), Tehran, Iran*

(Received September 10, 2012)

Abstract

Assignment of structural domains in complex protein structures is an important task in bioinformatics researches. As the number of known protein structures grows rapidly, the need for automatic methods for determining protein domains based on the proteins tree-dimensional structure becomes more desirable. In this paper, we introduce a new domain decomposition algorithm which is based on the dominating set of the graph representation of a protein. To evaluate our method, we compare our results with the other computational methods on a commonly used benchmark of 55 proteins. It is shown that the performance of our algorithm is better than the other automatic methods.

Introduction

Proteins can be considered as a set of several structural domains. Each domain has a stable structure and can fold independently of the rest of the protein [1–3]. Structural domains are compact and should have a hydrophobic core. Each of these semi-independent units has a specific function [4].

[*] Corresponding author: ch-eslahchi@sbu.ac.ir

Structural domains are the basic components of the proteins. They should not necessarily be continues in the amino acid sequence and may consist of non-sequential segments [5, 6]. The assignment of structural domains is an important task in the classification of the proteins based on their three-dimensional structure [7, 8], understanding the proteins folding, function and evolution [9]. The concept of assigning protein domains has been proposed by Wetlaufer [6], Rossman and Liljas [10] in 1970. Domain decomposition can be done manually by human experts. There are several classifications of the protein structures based on structural domains like SCOP [7] and CATH [8]. SCOP classifications rely mainly on human experts. CATH uses both automatic methods and human experts' opinion for the classification of the protein structures. Due to the exponential rate of growth in the identification of the protein structures, the need for automatic methods for determining protein domains are required [11]. There are several automatic algorithms such as NCBI [12], DomainParser [13], PDP [14], PUU [15], DDomain [16], DHcl [17] and Dodis [18]. The computational approaches of these methods are different but they mainly focus on the fact that the residue contacts of amino acids within a domain are denser than between domains [19]. In this paper, we introduce a novel algorithm for determining protein domains, using the dominating set of the graph representation of a protein.

Method

A graph is usually shown by $G = (V, E)$ where V is a finite set of nodes and E is a finite set of edges, which are 2-element subsets of V. For constructing the graph of a protein, each amino acid residue of the protein is considered as a node of a graph. The edges of this graph are generated from the structural coordinates of the amino acid residues [20] that are obtained from the PDB (Protein Data Bank) [21]. Two nodes are connected by an edge if the distance between the C^{α} atoms of their corresponding amino acid residues is 4Å or less, following the definition of Holm and Sander [9].

A dominating set for a graph $G = (V, E)$ is a subset D of V such that every vertex not in D is a neighbor of at least one member of D. For example given the graph G shown in Figure 1, $D = \{3, 5\}$ is a dominating set for G.

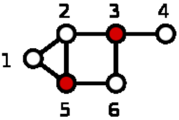
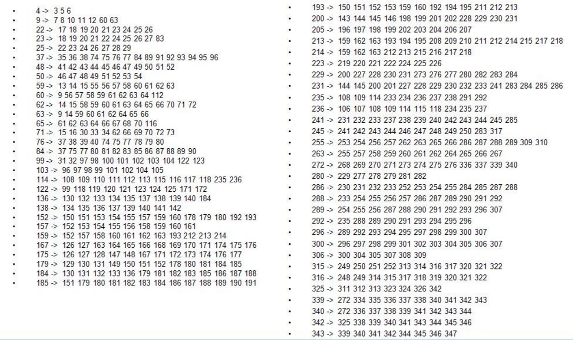


Figure 1. The set $\{3, 5\}$ denotes a dominating set for graph G.

For assigning the protein domains, we first construct a dominating set for the protein's graph. Let P be a protein with m amino acids and $G_P = (V_P, E_P)$ shows its representative graph. Finding the minimum dominating set in a graph is an NP-hard problem [22], so we use a greedy approach to obtain a dominating set D for the graph G_P. A dominating set for the protein $1A8Y$ is shown in Figure 2. The graph of this protein consists of 347 nodes while its dominating set has 60 nodes.



- 4 -> 3 5 6
- 9 -> 7 8 10 11 12 60 63
- 22 -> 17 18 19 20 21 23 24 25 26
- 23 -> 18 19 20 21 22 24 25 26 27 83
- 25 -> 22 23 24 26 27 28 29
- 37 -> 35 36 38 74 75 76 77 84 89 91 92 93 94 95 96
- 48 -> 41 42 43 44 45 46 47 49 50 51 52
- 50 -> 46 47 48 49 51 52 53 54
- 59 -> 13 14 15 55 56 57 58 60 61 62 63
- 60 -> 9 56 57 58 59 61 62 63 64 112
- 62 -> 14 15 58 59 60 61 63 64 65 66 70 71 72
- 63 -> 9 14 59 60 61 62 64 65 66
- 65 -> 61 62 63 64 66 67 68 70 116
- 71 -> 15 16 30 33 34 62 66 69 70 72 73
- 76 -> 37 38 39 40 74 75 77 78 79 80
- 84 -> 37 75 77 80 81 82 83 85 86 87 88 89 90
- 99 -> 31 32 97 98 100 101 102 103 104 122 123
- 103 -> 96 97 98 99 101 102 104 105
- 114 -> 108 109 110 111 112 113 115 116 117 118 235 236
- 122 -> 99 118 119 120 121 123 124 125 171 172
- 136 -> 130 132 133 134 135 137 138 139 140 184
- 138 -> 134 135 136 137 139 140 141 142
- 152 -> 150 151 153 154 155 157 159 160 178 179 180 192 193
- 157 -> 152 153 154 155 156 158 159 160 161
- 159 -> 152 157 158 160 161 162 163 193 212 213 214
- 167 -> 126 127 163 164 165 166 168 169 170 171 174 175 176
- 175 -> 126 127 128 147 148 167 171 172 173 174 176 177
- 179 -> 129 130 131 149 150 151 152 178 180 181 184 185
- 184 -> 130 131 132 133 136 179 181 182 183 185 186 187 188
- 185 -> 151 179 180 181 182 183 184 186 187 188 189 190 191
- 193 -> 150 151 152 153 159 160 192 194 195 211 212 213
- 200 -> 143 144 145 146 198 199 201 202 228 229 230 231
- 205 -> 196 197 198 199 202 203 204 206 207
- 213 -> 159 162 163 193 194 195 208 209 210 211 212 214 215 217 218
- 214 -> 159 162 163 212 213 215 216 217 218
- 223 -> 219 220 221 222 224 225 226
- 229 -> 200 227 228 230 231 273 276 277 280 282 283 284
- 231 -> 144 145 200 201 227 228 229 230 232 233 241 283 284 285 286
- 235 -> 108 109 114 233 234 236 237 238 291 292
- 236 -> 106 107 108 109 114 115 118 234 235 237
- 241 -> 231 232 233 237 238 239 240 242 243 244 245 285
- 245 -> 241 242 243 244 246 247 248 249 250 283 317
- 255 -> 253 254 256 257 262 263 265 266 286 287 288 289 309 310
- 263 -> 255 257 258 259 260 261 262 264 265 266 267
- 272 -> 268 269 270 271 273 274 275 276 336 337 339 340
- 280 -> 229 277 278 279 281 282
- 286 -> 230 231 232 233 252 253 254 255 284 285 287 288
- 288 -> 233 254 255 256 257 286 287 289 290 291 292
- 289 -> 254 255 256 287 288 290 291 292 293 296 307
- 292 -> 235 288 289 290 291 293 294 295 296
- 296 -> 289 292 293 294 295 297 298 299 300 307
- 300 -> 296 297 298 299 301 302 303 304 305 306 307
- 306 -> 300 304 305 307 308 309
- 315 -> 249 250 251 252 313 314 316 317 320 321 322
- 316 -> 248 249 314 315 317 318 319 320 321 322
- 325 -> 311 312 313 323 324 326 342
- 339 -> 272 334 335 336 337 338 340 341 342 343
- 340 -> 272 336 337 338 339 341 342 343 344
- 342 -> 325 338 339 340 341 343 344 345 346
- 343 -> 339 340 341 342 344 345 346 347

Figure 2. A dominating set of the protein $1A8Y$.

Next we construct a matrix for the obtained dominating set, $D = \{x_1, x_2, \dots, x_n\}$. We define a matrix $DS = [DS_{i,j}]$ by:

$$DS_{i,j} = \frac{|N(x_i) \cap N(x_j)|}{|N(x_i) \cup N(x_j)|}$$

where $N(x_i)$ denotes the set of the neighbors of the node x_i in G_P.

Figure 3 shows the matrix DS for the dominating set of the protein $1A8Y$. This protein has three domains which are shown by different colors in Figure 4. Its initial domains are also shown by different colors in the DS matrix. The entries of the colored parts of the DS matrix are almost none zero, while the rate of none zero elements in the white parts is small. The decomposition of the domains of this protein is (3-126), (127-228) and (229-347).

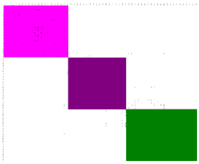


Figure 3. The matrix DS of the dominating set of the protein $1A8Y$.

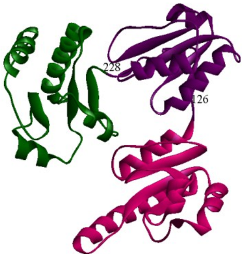


Figure 4. A solid ribbon diagram showing the three domains of the protein $1A8Y$.

For determining and merging the initial clusters, first we define the distance matrix $DIS = [d_{i,j}]$ from the matrix DS as follows:

$$d_{i,j} = \frac{\sum_k |DS_{i,k} - DS_{j,k}|}{n}.$$

The members of the dominating set are considered as initial clusters and are merged based on this distance matrix and the neighbor-joining algorithm [23]. For this purpose, first the array U of size n is obtained from the matrix DIS by:

$$U_i = \frac{1}{n-2}\sum_{i \neq k} d_{i,k}.$$

Then the matrix M is constructed from U and DIS:

$$M_{i,j} = d_{i,j} - U_i - U_j.$$

We define θ as:

$$\theta = \frac{\min[m_{i,j}] + \max[m_{i,j}]}{3}.$$

For merging the clusters, the minimum entry of M, $M_{x,y}$, is selected and the clusters x and y are merged together. Then the distance matrix DIS is updated by changing the row corresponding to the cluster x as:

$$d_{x,k} = \frac{d_{x,k} + d_{y,k} - d_{x,y}}{2}$$

and removing the row corresponding to y. The matrixes U and M are then computed from the matrix DIS in each step. This procedure is repeated until $M_{x,y}$ is less than θ.

In the next step, obtained clusters are merged based on their inter and intra densities; with respect to the fact that the residue interactions are denser within domains than between domains [19]. The density of the cluster C_i is computed by:

$$density(C_i) = \frac{|E(C_i)|}{|C_i|}$$

where $|E(C_i)|$ denotes the number of edges between the nodes of C_i. The intra-residue interactions of a cluster, which is the result of merging two clusters C_i and C_j is defined as:

$$intradensity(C_i, C_j) = \frac{|E(C_i \cup C_j)|}{|C_i \cup C_j|}.$$

The inter density between two clusters C_i and C_j is computed by:

$$interdensity(C_i, C_j) = \frac{|E(I(C_i \cup C_j))|}{|C_i \cup C_j|}$$

where $|E(I(C_i \cup C_j))|$ denotes the set of edges with one end in C_i and the other end in C_j. We define the total density of the two clusters C_i and C_j as:

$$totaldensity(C_i, C_j) = intradensity(C_i, C_j) - \frac{\left(density(C_i) + density(C_j)\right)}{2} + interdensity(C_i, C_j).$$

Two clusters C_i and C_j with the maximum total density are repeatedly merged together until the number of clusters become less than η.

Next unassigned vertices are determined and merged with the existing clusters based on their neighbors in each cluster.

In the next step, we assign a pattern to each cluster. Let $V_P = \{v_1, v_2, \dots, v_m\}$, we define the $m.m$ matrix NA as:

$$NA_{i,j} = |N(v_i) \cap N(v_j)|.$$

For a cluster C, the pattern $P(C)$ is defined by:

$$P(C) = \sum_{x \in C} r_x$$

where r_x is the row corresponds to the node x in NA. Then the similarity score $S(C, D)$ between two clusters, C and D, is defined by:

$$S(C,D) = \frac{|\{k | k \in C \ and \ P(D)_k \neq 0\}|}{|C|}.$$

Two clusters X and Y with the maximum similarity score are repeatedly merged until $S(X,Y)$ become less than a threshold δ.

Threshold determination

The thresholds that have been used in this algorithm are determined using a training set consisting of 50 proteins selected from a set of 135 proteins in the Balanced Domain Benchmark-3 of the pDomain resource introduced in [4]. This database is available at http://www.pdomains.sdsc.edu. Both expert methods, CATH and SCOP, agree on the domain decomposition of these 50 proteins, which are selected as the training set.

The obtained values for the parameters are: η =10 and $\delta = 45$. The minimum size of a domain is considered to be 32 residues in our algorithm.

Results and Discussion

The algorithm is applied to a frequently used benchmark consisting of 55 proteins introduced by Jones et al. [24]. A domain assignment is considered correct if the number of domains is the same as the assignment by the experts and the amino acid assignment of the domains is at least 85% in agreement with the experts' opinion [24]. In this paper, the domain decomposition of the automatic methods is compared with the assignments by the human experts, CATH or SCOP, similar to [4]. Using the above definition, the domain decomposition of each method is considered correct if it is consistent with the domain assignment of CATH or SCOP. It is noticeable that even the manual assignments of the protein domains, are sometimes different for the same proteins; since there is not a precise definition of protein domains [25–27]. This could also be the result of considering the function and evolutionary information of proteins in the domain decompositions by experts [28].

Our method correctly assigns 96.3% of the 55 proteins (Table 1). To compare our results with the assignments of other automatic domain assignment methods, we use dConsensus. dConsensus is a web resource which is available at http://pdomains.sdsc.edu/dConsensus [4] and displays the results of domain decompositions from multiple algorithmic methods. Using this software the results of six automatic domain assignment algorithms is calculated.

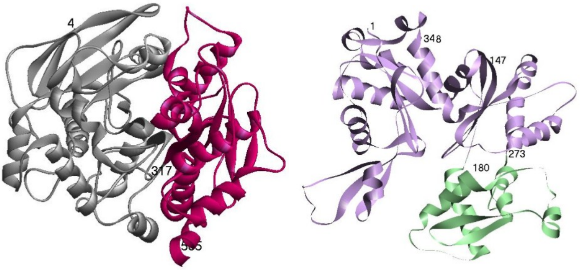
According to these results, the correct assignments by PDP, DomainParser, NCBI and PUU are 92.7%, 85.5%, 89% and 76.4% respectively. DHcL and DDomain run only on 41 and 50 proteins and their results are 70.7% and 84%.

Table 1. Protein PDB codes of 55 proteins, residue ranges of domains assigned by CATH, SCOP and our algorithm (fragments of domains are separated by ‘,’ and ‘/’ is used to separate domains).

Protein PDB ID	**CATH**	**SCOP**	**Our Algorithm**
8acna	2-202/ 203-315/ 316-490/ 534-754	**2-528/ 529-754**	**2-528/ 529-754**
3pmga	**1-197/ 198-300/ 301-400/ 401-561**	**1-190/ 191-303/ 304-420/ 421-561**	**1-188/ 189-300/ 301-420/ 421-561**
1phha	1-72, 96-180, 269-351/ 73-95, 181-268, 352-388	**1-173, 276-394/ 174-275**	**1-180, 267-394/ 181-266**
3grsa	**18-160, 290-365/ 161-289/ 366-478**	**18-165, 291-363/ 166-290/ 364-478**	**18-150, 290-363/ 151-289/ 364-478**
1atna	5-35, 72-135, 338-373/ 36-69/ 137-182, 272-333/ 183-268	2-147/ 148-373	1-179, 247-372/ 180-273
1ezma	**1-152/ 153-298**	1-301	**1-132/ 133-298**
1fnba	**19-151/ 152-314**	**19-154/ 155-314**	**19-163/ 164-314**
1gpba	**19-485 , 813-836/ 486-812**	1-842	**19-484, 813-841/ 485-812**
1lapa	**1-165/ 166-483**	**1-159/ 160-484**	**1-170/ 171-484**
1pfka	**1-142, 257-303/ 143-252, 304-319**	1-320	**1-137, 257-302/ 138-256, 303-319**
1ppna	**1-212**	**1-212**	**1-212**
1rhda	**1-156/ 157-293**	**1-149/ 150-293**	**1-151/ 152-293**
1sgta	1-12, 97-210/ 13-96, 211-223	**1-223**	**1-223**
1vsga	**1-33, 86-255/ 34-85 256-362**	1-364	**1-24, 86-253/ 42-85, 254-362**
1bksa	**1-267**	**1-268**	**1-267**
2cypa	**4-144, 266-294/ 145-265**	1-294	**2-140, 255-294/ 141-254**
2hada	**1-310**	**1-310**	**1-310**
3cd4a	**1-98/ 99-173**	**1-97/ 98-178**	**1-98/ 99-178**
1g6na	**10-138/ 139-207**	**8-138/ 139-207**	**7-137/ 138-206**
3pgka	**2-187/ 194-402**	1-416	**1-200/ 201-415**
4gcra	**1-83/ 84-174**	**1-85/ 86-174**	**1-80/ 81-174**

5fbpa	**7-199/ 200-334**	1-335	**6-200/ 201-334**
8adha	**1-178, 318-374/ 179-317**	**1-163, 340-374/ 164-339**	**1-189, 322-374/ 190-321**
8atca	**1-133, 292-310/ 134-291**	1-150/ 151-310	**1-134, 285-310/ 135-284**
8atcb	**8-100/ 101-153**	**8-100/ 101-153**	**8-97/ 101-153**
2acea	4-535	1-537	4-317/ 318-535
2buka	**13-196**	**13-196**	**26-195**
2aaka	**1-150**	**1-152**	**1-150**
1bbha	**1-131**	**1-131**	**1-131**
1bbpa	**1-173**	**1-173**	**1-173**
1brda	**8-226**	**1-248**	**8-226**
1fxia	**1-96**	**1-96**	**1-96**
1gkya	**2-33, 94-187/ 34-93**	1-187	**2-33, 82-186/ 34-81**
2gmfa	**4-124**	**1-127**	**4-124**
1gmpa	**1-96**	**1-96**	**1-96**
1goxa	**2-360**	**1-370**	**2-360**
1ofva	**1-169**	**1-169**	**1-169**
1pypa	**1-281**	**1-285**	**1-281**
1rbpa	**1-175**	**1-182**	**1-175**
1rcba	**1-129**	**1-129**	**1-129**
1rvea	**2-245**	**1-245**	**2-245**
1snca	**7-141**	**1-149**	**7-141**
1tiea	**1-170**	**1-172**	**1-170**
1tlka	**33-135**	**1-154**	**33-135**
1ulaa	**1-289**	**1-289**	**1-289**
1bksb	9-53, 87-205/ 54-86, 206-391	**1-397**	**3-394**
2azaa	**1-129**	**1-129**	**1-129**
2ceya	**1-306**	**1-306**	**1-306**
2m2a	**1-155**	**1-155**	**1-155**
2tmvp	**1-154**	**1-158**	**1-154**
3chya	**1-128**	**1-128**	**1-128**
3claa	**1-213**	**1-213**	**1-213**
3dfra	**1-213**	**1-213**	**1-213**
4blma	**1-162**	**1-162**	**1-162**
5p21a	**1-166**	**1-166**	**1-166**

The proteins that are decomposed incorrectly by our method are 1atna and 2acea (Figure 5).



(A) (B)

Figure 5. Domain decompositions of the proteins 2acea and 1atna which is obtained by our algorithm. Different domains are shown by different colors. (A) 2acea (4-317/ 318-535). (B) 1atna (1-179, 247-372/ 180-273).

The protein 2acea is considered as a one domain protein by the experts, but our algorithm assigns two domains to this protein (Figure 5(A)). Among automatic methods, DomainParser and DDomain consider this protein as a single domain protein. DHcL assigns two domains for this protein which is similar to our algorithm (Table 2).

Table 2. Residue ranges of domains assigned by different methods for protein PDB code 2acea (fragments of domains are separated by ‘,’ and ‘/’ is used to separate domains).

2acea					
SCOP	1-537	CATH	4-535	OUR Algorithm	4-317/ 318-535
pdp	4-315/ 332-394, 526-535/ 316-331 , 395-525	DomainParser	1-537	NCBI	1-230, 301-326, 415-516/ 231-300/ 327-414, 517-537
puu	1-233, 281-332, 396-508/ 234-280/ 333-395	DDomain	4-535	DHcl	4-315/ 316-535

For the protein 1atna, expert methods give different domain decompositions. SCOP considers this protein as a two-domain protein while CATH assigns four domains for this protein. Our algorithm considers two domains for this protein (Figure 5(B)) similar to SCOP but the fragments of our domains are inconsistent with the assignment by SCOP. Only pdp considers four domains for this protein similar to the CATH assignment (Table 3). Domain decomposition by DomainParser is also similar to our assignment.

Table 3. Residue ranges of domains assigned by different methods for protein PDB code 1atna (fragments of domains are separated by ‘,’ and ‘/’ is used to separate domains).

1atna					
SCOP	2-147/ 148-373	CATH	5-35, 72-135, 338-373/ 36-69/ 137-182, 272-333/ 183-268	OUR Algorithm	1-179, 247-372/ 180-273
pdp	2-34, 70-138, 340-373/ 35-69/ 139-185, 261-339/ 186-260	DomainParser	2-148, 338-373/ 149-337	NCBI	1-137, 353-372/ 138-182, 263-352/ 220-262
puu	1-33, 69-141, 336-372/ 142-179, 273-335/ 180-272	DDomain	2-103/ 104-373	DHcl	2-373

The above results show that our algorithm which is introduced in this paper performs better results compared to other automatic algorithms.

References

[1] M. Baron, I. D. Campbel, Protein modules, *Trends Biochem. Sci.* **16** (1991) 13–17.

[2] D. B. Wetlaufer, Nucleation, rapid folding, and globular intrachain regions in proteins, *Proc. Nail. Acad. Sci. USA*. **70** (1973) 697–701.

[3] K. Alden, S. Veretnik, P. E. Bourne, dConsensus: a tool for displaying domain assignments by multiple structure-based algorithms and for construction of a consensus assignment, *BMC Bioinf.* **11** (2010) 310–310.

[4] J. S. Richardson, The anatomy and taxonomy of protein structure, *Adv. Protein Chem.* **34** (1981) 246–253.

[5] R. B. Russell, Domain insertion, *Protein Eng.* **7** (1994) 1407–1411.

[6] D. B. Wetlaufer, Nucleation, rapid folding, and globular intrachain regions in proteins, *Proc. Nail. Acad. Sci. USA.* **70** (1973) 697–701.

[7] A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia, SCOP: A structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.* **247** (1995) 536–540.

[8] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, J. M. Thornton, CATH — A hierarchic classification of protein domain structures, *Structure* **5** (1997) 1093–1108.

[9] L. Holm, C. Sander, Parser for protein folding units, *Proteins* **19** (1994) 256–268.

[10] M. G. Rossman, A. Liljas, Letter: recognition of structural domains in globular proteins, *J. Mol. Biol.* **85** (1974) 177–181.

[11] S. Veretnik, I. N. Shindyalov, Computational methods for domain partitioning in protein structures, in: Y. Xu, D. Xu, J. Liang (Eds.), *Computational Methods for Structure Prediction and Modelling*, Springer, 2006, pp. 125–145.

[12] T. Madej, J. F. Gibrat, S. H. Bryant, Threading a database of protein cores, *Proteins* **23** (1995) 356–369.

[13] J. Guo, D. Xu, D. Kim, Y. Xu, Improving the performance of domain parser for structural domain partition using neural network, *Nucleic Acids Res.* **31** (2003) 944–952.

[14] N. Alexandrov, I. Shindyalov, PDP: Protein domain parser, *Bioinf. Appl. Note* **19** (2003) 429–430.

[15] L. Holm, C. Sander, Parser for protein folding units, *Proteins Struct. Funct. Genet.* **19** (1994) 256–268.

[16] H. Zhou, B. Xue, Y. Zhou, DDOMAIN: Dividing structures into domains using a normalized domain-domain interaction profile, *Protein Sci.* **16** (2007) 947–955.

[17] G. Koczyk, I. N. Berezovsky, Domain hierarchy and closed loops (DHcL): a server for exploring hierarchy of protein domain structure, *Nucleic Acids Res.* **36** (2008) 239–245.

[18] O. Carugo, Identification of domains in protein crystal structures, *J. Appl. Crystall.* **40** (2007) 778–781.

[19] Y. Xu, D. Xu, Protein domain decomposition using a graph–theoretic approach, *Bioinf.* **16** (2000) 1091–1104.

[20] M. Habibi, C. Eslahchi, M. Sadeghi, H. Pezeshk, The interpretation of protein structures based on graph theory and contact map, *Open Access Bioinf.* **2** (2010) 1–11.

[21] Protein data bank, http://www.pdb.org/.

[22] P. Crescenzi, V. Kann, M. Halldórsson, M. Karpinski, G. Woeginger, Minimum dominating set, *A Compendium of NP Optimization Problems*, 2000.

[23] N. Saitou, M. Nei, The neighbor–joining method: a new method for reconstructing phylogenetic trees, *Mol. Biol. Evolution* **4** (1987) 406–425.

[24] S. Jones, M. Stewart, A. Michie, M. B. Swindells, C. Orengo, J. M. Thornton, Domain assignment for protein structures using a consensus approach: characterization and analysis, *Protein Sci.* **7** (1998) 233–242.

[25] M.B. Swindells, A procedure for detecting structural domains in proteins, *Protein Sci.* **4** (1995) 103–112.

[26] L. Wernisch, M. Hunting, S. J. Wodak, Identifcation of structural domains in proteins by a graph heuristic, *Proteins* **35** (1999) 338–352.

[27] W. R. Taylor, Protein structural domain identification, *Protein Eng.* **12** (1999) 203–216.

[28] L. Lo Conte, S. E. Brenner, T. J. Hubbard, C. Chothia, A. G. Murzin, SCOP database in 2002: refinements accommodate structural genomics, *Nucleic Acids Res.* **30** (2002) 264–267.