

Ordering Protein Contact Matrices

Chuan Xu, Guillaume Bouvier, Benjamin Bardiaux, Michael Nilges, Thérèse Malliavin, Abdel Lisser

▶ **To cite this version:**

Chuan Xu, Guillaume Bouvier, Benjamin Bardiaux, Michael Nilges, Thérèse Malliavin, et al.. Ordering Protein Contact Matrices. Computational and Structural Biotechnology Journal, 2018, 16, pp.140 - 156. 10.1016/j.csbj.2018.03.001 . hal-01776351

HAL Id: hal-01776351
https://hal.science/hal-01776351

Submitted on 25 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Computational and Structural Biotechnology Journal 16 (2018) 140–156

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Ordering Protein Contact Matrices

Chuan Xu[a], Guillaume Bouvier[b,c], Benjamin Bardiaux[b,c], Michael Nilges[b,c], Thérèse Malliavin[b,c,*], Abdel Lisser[a]

[a] Laboratoire de Recherche en Informatique, Université Paris-Sud and CNRS UMR8623, France
[b] Unité de Bioinformatique Structurale, Institut Pasteur and CNRS UMR3528, France
[c] Centre de Bioinformatique, Biostatistique et Biologie Intégrative, Institut Pasteur and CNRS USR3756, France

ARTICLE INFO

Article history:
Received 1 October 2017
Received in revised form 28 February 2018
Accepted 1 March 2018
Available online 16 March 2018

Keywords:
Protein contact matrix
Fold prediction
Graph theory
Dynamic programming
Self-organizing map

ABSTRACT

Numerous biophysical approaches provide information about residues spatial proximity in proteins. However, correct assignment of the protein fold from this proximity information is not straightforward if the spatially close protein residues are not assigned to residues in the primary sequence. Here, we propose an algorithm to assign such residue numbers by ordering the columns and lines of the raw protein contact matrix directly obtained from proximity information between unassigned amino acids. The ordering problem is formatted as the search of a trail within a graph connecting protein residues through the nonzero contact values. The algorithm performs in two steps: (i) finding the longest trail of the graph using an original dynamic programming algorithm, (ii) clustering the individual ordered matrices using a self-organizing map (SOM) approach. The combination of the dynamic programming and self-organizing map approaches constitutes a quite innovative point of the present work. The algorithm was validated on a set of about 900 proteins, representative of the sizes and proportions of secondary structures observed in the Protein Data Bank. The algorithm was revealed to be efficient for noise levels up to 40%, obtaining average gaps of about 20% at maximum between ordered and initial matrices. The proposed approach paves the ways toward a method of fold prediction from noisy proximity information, as TM scores larger than 0.5 have been obtained for ten randomly chosen proteins, in the case of a noise level of 10%. The methods has been also validated on two experimental cases, on which it performed satisfactorily.

© 2018 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The application of few high-resolution approaches, such as X-ray crystallography and Nuclear Magnetic Resonance (NMR), to determine the structures of monomeric proteins and of small biomolecular complexes, constituted the starting point of structural biology. Recently, the development of techniques better adapted to the study of molecular assemblies, induced a transition of structural biology toward integrative structural biology [1,2], connecting eventually to cellular biology. In the frame of structural integrative biology, a large set of biophysical techniques is used, including: X-ray crystallography, NMR, cryo-EM, SAXS, FRET, mass-spectrometry coupled to cross-linking or affinity measurements [3,4,5]. Many of these techniques measure distances or distribution of distances between biomolecular atoms, residues or molecules.

In addition to the development of experimental techniques, the wealth of information accumulated on high-resolution structures as well as the development of bioinformatics, allows the widespread use of structure prediction approaches, based on proximity information between different parts of the modeled system. Such approaches include molecular homology modeling [6] to predict the fold of a protein, the use of approaches of protein structure prediction coupled to experimental measurements [7] and protein-protein docking methods [8,9] to predict complex structures.

In NMR, the observation of nuclear Overhauser effect between the observed nuclei in a biomolecule allows to have a quite precise description of the magnetization transfer between these nuclei. This measure is related to the spatial proximity of nuclei and vary along $1/r^6$, where r is the distance between the nuclei. It is indexed on the spectra by the chemical shifts of the two involved nuclei.

The spectral assignment [10,11,12] allows to realize a mapping of the NMR chemical shifts to the protein atoms observed by NMR. This task is realized by recording a varied set of heteronuclear NMR spectra [13]. A simplification of the assignment problem was proposed

* Corresponding author at: Unité de Bioinformatique Structurale, Institut Pasteur and CNRS UMR3528, France.
E-mail address: terez@pasteur.fr (T. Malliavin).

https://doi.org/10.1016/j.csbj.2018.03.001
2001-0370/© 2018 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

by [14], by regrouping the $^{15}N-{}^{1}H$ HSQC-NOESY signals of each protein residue to get numbers describing proximities between residues. Each of these residues is indexed by a couple of chemical shifts of Cα and Hα nuclei. The assignment problem can then be rewritten as a mapping of these spectral residues to the protein primary sequence, that we call assignment problem of residues. From the spectral proximities between unassigned residues, a contact matrix can be built, indexed by chemical shifts. But, due to the signal superposition on the spectra, false positive spectral proximities can be detected, and induce the appearance of spurious proximity elements in the contact matrix. The present work intends to test a procedure for ordering such noisy contact matrix in the frame of two additional hypotheses: (i) it is possible to detect proximities between all consecutive residues in the sequence, (ii) the spurious proximities are located on the matrix in an uncorrelated way. These hypotheses are strong, but can be expected to be matched in the case of high signal to noise ratio.

In the present work, we focused on the ordering problem of the contact matrix and we defined a synthetic *contact matrix*, indexed by the protein residues, as a matrix of zeros and ones, one being entered in the matrix for every elements i and j for which a proximity was measured between the corresponding residues i and j. A proximity can be also called a contact. From well-known features of proteins [15], the Cα atoms in residues i and $i+1$ are separated by about 3.8 Å, whereas the Cα atoms in residues i and $i+2$ are far apart at most 6 Å. Thus, using appropriate distance cutoffs, first and second sub-diagonals of contact matrices can be supposed to be filled or almost filled with contacts.

In this context, we have developed an approach capable of ordering a contact matrix using dynamic programming. As there is more chance to observed proximities between residues close in the sequence, we decided to use a dynamic programming base heuristic to maximize the number of contacts on the first three sub-diagonals of the matrix. The approach has been validated on a set of about 900 protein contact matrices, representing the full varieties of protein sizes, folds and secondary structure contents. The proposed approach displays a good level of robustness and precision.

One should notice that the method proposed here intends to assign each residue observed in the experiment to its correct relative position in the protein sequence. This ordering problem is different from filtering noise from an already ordered matrix [16] or from predicting contact matrix from protein sequence [17,18,19,20,21,22,23,24,25]. In particular, in the field of protein fold prediction, the proximities between consecutive residues are implicitly determined from the residue number in the sequence, whereas in the present approach, they have to be extracted from the set of spectral proximities.

The approach could be applied to solution state NMR, using the processing of rows extracted from a 3D $^{15}N-{}^{1}H$ HSQC-NOESY similar to the one performed by [14], or to solid state NMR spectra using a lighter pre-processing of PDSD spectra displaying correlations between α ^{13}C carbons [26]. In both cases, the indexes for input signal would be: the couples of NMR resonance frequencies of amide hydrogens and nitrogens for solution NMR, and the NMR resonance frequencies of α carbons for solid state NMR.

The objectives of the present work are to: (i) formalize the problem of ordering the contact matrix in such a way that it can be studied independently from the type of data experimentally measured, (ii) test the robustness to noise and the efficiency of reconstructing the fold, depending on various protein parameters: protein length, type and percentage of secondary structures.

2. Problem Formulation

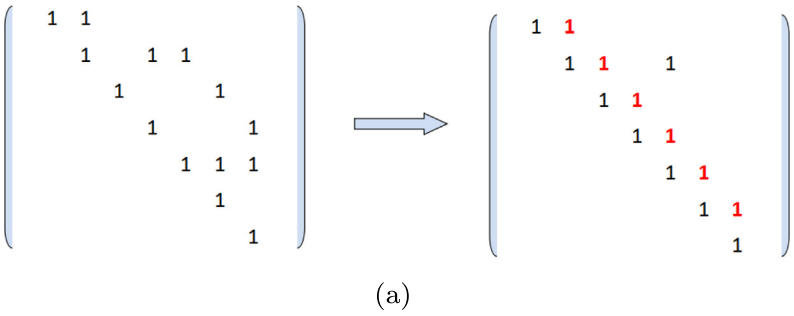
In this section, we present our algorithmic approach for the protein contact matrix ordering problem. After formulating the problem in terms of graph, a dynamic programming algorithm, searching for the longest trail in the graph to produce an individual ordered matrix, will be presented. Then, a clustering step, based on self-organizing maps (SOM) [27] and used to extract the final ordered matrix from the set of individual ordered matrices, will be described.

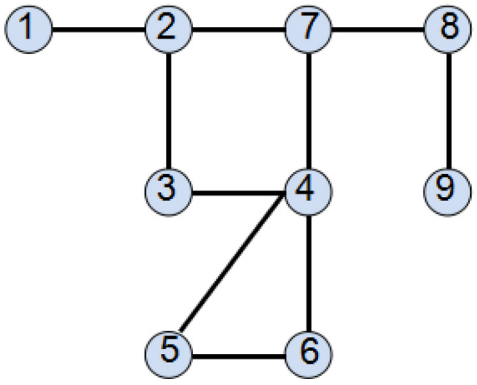
2.1. Problem Formulation

The ordering problem consists in finding an optimal permutation of the contact matrix in order to bring the maximum of nonzero components on the three first sub-diagonals (Fig. 1a). Thus, this problem can be modeled by graph theory tools where the optimal permutation problem is equivalent to the *longest trail problem* in the graph of the disordered contact matrix. A trail is a walk without repeated edges [28], and its length is the total number of traversed edges. In graph theory, the longest trail problem is different from the longest path problem, as the path may display repeated nodes [29], as illustrated in Fig. 1b. Indeed, in this example, the longest path is $1-2-3-4-5-6-4-7-8-9$ with length 9 while the longest trail is $1-2-3-4-7-8-9$ whose length is 6. Following the order of the nodes in the obtained longest trail, we affect each node consecutively and rebuild the contact matrix. In order to return a contact matrix with the maximum number of contacts on the first three sub-diagonals, we choose among the longest trails of the same length the one with the maximum number of non-zeros on the second diagonal. If multiple trails have the same maximum number of non-zeros on the second diagonal, we then choose the trail with maximum non-zero numbers on the third diagonal.

The studied problem belongs to the class of Hamiltonian paths, which in directed or undirected graphs visits each node exactly once. Hamiltonian path problem is NP-complete in general, for chordal, split and circle graphs [30,31,32]. Finding a Hamiltonian path in a given graph with length $n-n^{\epsilon}$ for any $\epsilon < 1$, where n is the number of the nodes of the graph, is a NP-complete problem [33]. There is no approximation polynomial time algorithm for solving this problem in a n-node Hamiltonian graph. Moreover, there is no polynomial-time constant-factor approximation algorithm for the longest path problem unless $P{=}NP$ [33]. However, the longest path problem is solvable in polynomial time for interval and bipartite permutation graphs [34,35]. Yet, to our best knowledge, no algorithm has ever been proposed and analyzed to solve directly the longest trail problem for arbitrary graphs. In the present work, we propose an efficient algorithm based on dynamic programming for solving this problem.

Nevertheless, some results have been already obtained in the literature concerning the longest trail problem. A lower bound on the length of the longest trail was proposed in [36] for enough dense graphs. The author proves that for a connected graph G of n nodes such that its average degree is at least $k \geq 12.5$, there exists a trail of length at least $\frac{(\lceil k \rceil+1)k}{2}$ or $n \leq \lceil k \rceil + 2$. In [37], a polynomial time algorithm was proposed to find a longest "properly edge-colored" trail in a specific class of edge-colored graphs. In [38], a similar problem, called the spanning trail problem, is first proposed and studied. The spanning trail is a trail that visits all nodes at least once while visits some edges at most once. A sufficient condition was derived for the existence of such a trail, and was further strengthened in [39]. In [40], it was proved that finding such a trail is NP-complete in general grid graphs, and, for the first time, an algorithm to find a spanning trail in a wide subclass of grid graphs was given. In [41,42], the spanning trails containing given edges in a graph G were studied. Other related but not equivalent problems (e.g. Eulerian extension problem [43], Chinese postman problem [44] and traveling salesman problem [45]), are considered and studied widely in the graph theory.





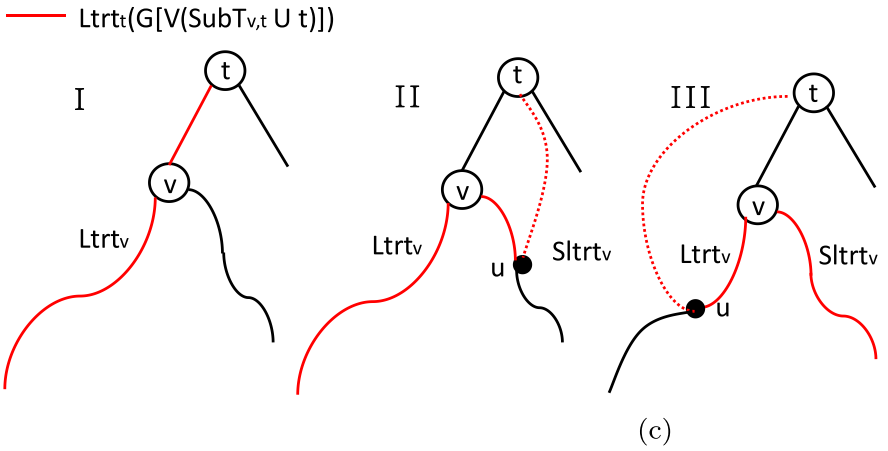


Fig. 1. (a) Matrix presentation of the ordering problem. (b) A graph with 9 vertices to illustrate the difference between the longest trail and the longest path. (c) Example of calculating $Ltrt_t(G[V(SubT_{v,t} \cup t)])$ (Eq. (5)) in dynamic programming with known $Ltrt_v$ and $Sltrt_v$.

2.2. Presentation of the Algorithm to Produce Individual Ordered Contact Matrices

We develop a heuristic that uses Depth First Search (DFS) [46], a well known graph traversal algorithm (Fig. 2a) to build a dynamic programming algorithm (Fig. 2c). Let $G = (V, E)$ be an undirected and connected graph with n nodes and m edges, where V is the set of nodes and E is the set of edges. In our heuristic, we apply DFS to construct a rooted tree for each node $v \in V$. During the construction of each rooted tree, dynamic programming scheme is used to compute the longest trail, and the best longest trail with the maximum nonzero components on the first three sub-diagonals is stored.

DFS is a traversal algorithm that systematically visits all nodes in G, starts at a chosen root node and returns a spanning tree with the edges used during the search. The strategy of DFS is to explore as deep as possible along the edges [46]. Let T_r be the rooted spanning tree obtained by running DFS at node r in graph G and C_r be the set of children nodes of node r in T_r. We denote $SubT_{c,r}$ the sub-tree of T_r with rooted node c, and $G[V(SubT_{c,r})]$ the induced sub-graph of G for nodes in $SubT_{c,r}$. An induced sub-graph is composed of a subset of the vertices and of the edges with both endpoints in the subset.

We define the longest trail in the graph G which contains node r by $Ltrc_r(G)$, and the longest trail in the graph G by $Ltr(G)$. For a given trail tr, we denote by $|tr|$ the length of the trail. It is easy to see that $|Ltr(G)| = \max_{r \in V}\{|Ltrc_r(G)|\}$. Notice that $Ltrc_r(G)$ has an optimal substructure. Suppose that $Ltrc_r(G)$ is composed of the vertices $a \sim c_1 \sim r \sim c_2 \sim b$ where $c_1 \neq c_2 \in V$. Since r is the rooted node, there must exist two nodes c_1, c_2 belonging to C_r. Moreover, since T_r is obtained

Algorithm: DFS(G)

```
1 for each v ∈ V do
2 |   col[v] ← white;
3 |   π[v] ← nul;
4 end
5 for each v ∈ V do
6 |   if col[v] == white then
7 |   |   DFS_Visit(v);
8 |   end
9 end
```

(a)

Algorithm: DFS_Visit(t)

```
1 col[t] ← gray;
2 for each v ∈ Adj[t] do
3 |   if col[v] == white then
4 |   |   π[v] ← t;
5 |   |   DFS_Visit(v);
6 |   end
7 end
8 col[t] ← black;
9 FindLongestTrail(t);
```

(b)

Algorithm: FindLongestTrail(t)

```
1  begin
2      l_1[t] ← 0, l_2[t] ← 0, T_1[t] ← ∅, T_2[t] ← ∅;
3      for each v ∈ Adj[t] do
4          l ← 0, T ← ∅;
5          if col[v] == black ∧ π[v] == t then
6              l ← l_1[v] + 1, T ← {v} + T_1[v];
7              for all inverse edges e = (u, t) whose cycle will pass v do
8                  if u ∈ T_2[v] then
9                      path ← T_2(v, u), d ← l(T_2(v, u));
10                     if d + 1 + l_1[v] > l then
11                         l ← d + 1 + l_1[v];
12                         T ← {u} + Inverse{path} + T_1[v];
13                 if u ∈ T_1[v] then
14                     path ← T_1(v, u), d ← l(T_1(v, u));
15                     if d + 1 + l_2[v] > l then
16                         l ← d + 1 + l_2[v];
17                         T ← {u} + Inverse{path} + T_2[v];
18             if l > l_1[t] then
19                 l_2[t] ← l_1[t], T_2[t] ← T_1[t], l_1[t] ← l, T_1[t] ← T;
20             else
21                 if l > l_2[t] then
22                     l_2[t] ← l, T_2[t] ← T;
23         else
24             if col[v] == gray ∧ π[t] ≠ v then
25                 Save Inverse Edge e = (t, v) to the node v;
```

(c)

Fig. 2. Design of algorithm for longest trail problem in graph G. The algorithm begins on (a) which performs a DFS procedure on graph G. During its exploration of edges with classical recursive implementation, shown in (b), the longest trail algorithm (c) is integrated to compute the longest trail of the induced graph of the nodes that have been visited. The following notations are used: $Adj(v)$ is the set of neighbors of node v, $\pi(v)$ represents the parent node of node v, $T_1[v]$, $T_2[v]$ and d correspond to $Ltrt_v$, $Sltrt_v$ and $d(u, v)$ defined in Eq. (5), $l_j[v]$ presents the length of longest trail in $T_j[v]$. $l(\bullet)$ returns the length of the path given as input.

by running DFS in the graph G, for any pair of nodes $c_1, c_2 \in C_r$ in $Ltrc_r(G)$, there is no edge between the nodes in $V(SubT_{c_1,r})$ and the nodes in $V(SubT_{c_2,r})$, where $V(SubT_{c_1,r})$ and $V(SubT_{c_2,r})$ are the sets of nodes in $SubT_{c_1,r}$ and $SubT_{c_2,r}$ respectively. This implies that $a \sim c_1 \sim r$ and $b \sim c_2 \sim r$ are the longest trails to node r in $G[V(SubT_{c_1,r}) \cup r]$ and $G[V(SubT_{c_2,r}) \cup r]$ respectively. We denote $Ltrt_r(G)$ the longest trail to node r in the graph G. Thus, we have

$$|Ltrt_r(G)| = \max_{c \in C_r} |Ltrt_r(G[V(SubT_{c,r} \cup r)])|. \tag{1}$$

Table 1
The number of contact matrice, kept after removal of non-connected ones, for the different types of contact matrices and noise levels.

Contact matrices	Noise level (%)				
	10%	20%	30%	40%	50%
All_avg_9	802	748	720	680	609
All_avg_11	855	855	848	840	839
All_min_5	831	779	712	649	575
All_min_7	560	553	550	535	522
Ca_7	673	471	354	383	328
Ca_9	847	813	781	422	440

And

$$|Ltrc_r(G)| = |Ltrt_r(G)| + \max_{\substack{c\in C_r \\ c\notin V(Ltrt_r(G))}} |Ltrt_r(G[V(SubT_{c,r}\cup r)])| \quad (2)$$

$$= |Ltrt_r(G)| + |Sltrt_r(G)| \quad (3)$$

where $Sltrt_r(G)$ represents the second longest trail to node r in G.

During the exploration of DFS that starts at node r, there are three possible states for each node, which are *not explored*, *partially explored* and *completely explored*. The colors of these three states are noted by white, gray and black labels respectively in Fig. 2. *Partially explored* means that the node has been reached but its children have not yet been explored. When a node t turns into a *completely explored* state (line 8 of Fig. 2b), a rooted sub-tree $SubT_{t,r}$ with nodes in *completely explored* states can be obtained. In this case, we are interested to see how to obtain $Ltrt_t(G[V(SubT_{t,r})])$ recursively by using dynamic programming. When node r turns to *completely explored* mode, we obtain $Ltrt_r(G)$ which leads to $Ltrc_r(G)$. For the sake of simplicity, we use hereafter $Ltrt_i$ for $Ltrt_i(G[V(SubT_{i,r})])$ and $Sltrt_i$ for $Sltrt_i(G[V(SubT_{i,r})])$.

The recursive function for calculating $Ltrt_t$ can be written as:

$$|Ltrt_t| = \max_{v\in C_t} |Ltrt_t(G[V(SubT_{v,t}\cup t)])| \quad (4)$$

$$= \max_{v\in C_t} \left\{ |Ltrt_v|+1,\ \max_{\substack{(u,t)\in E \\ u\in D_v \\ u\notin V(Ltrt_v)}} \{|Ltrt_v|+d(u,v)+1\},\ \max_{\substack{(u,t)\in E \\ u\in D_v \\ u\in V(Ltrt_v)}} \{|Sltrt_v|+d(u,v)+1\} \right\} \quad (5)$$

where D_v is the set of children of node v in T_r, $V(Ltrt_v)$ is the set of nodes in $Ltrt_v$ and $d(u,v)$ is the length of the longest trail between u and v.

In Fig. 1c, we give a graphical presentation of Eq. (5) to show the relationship between $Ltrt_t(G[V(SubT_{v,t}\cup t)])$ and $Ltrt_v$ where $v \in C_t$. The first sub-figure I shows one candidate trail in $G[V(SubT_{v,t}\cup t)]$ which is the trail $Ltrt_v$ plus the additional edge (v,t) (line 6 in Fig. 2c).

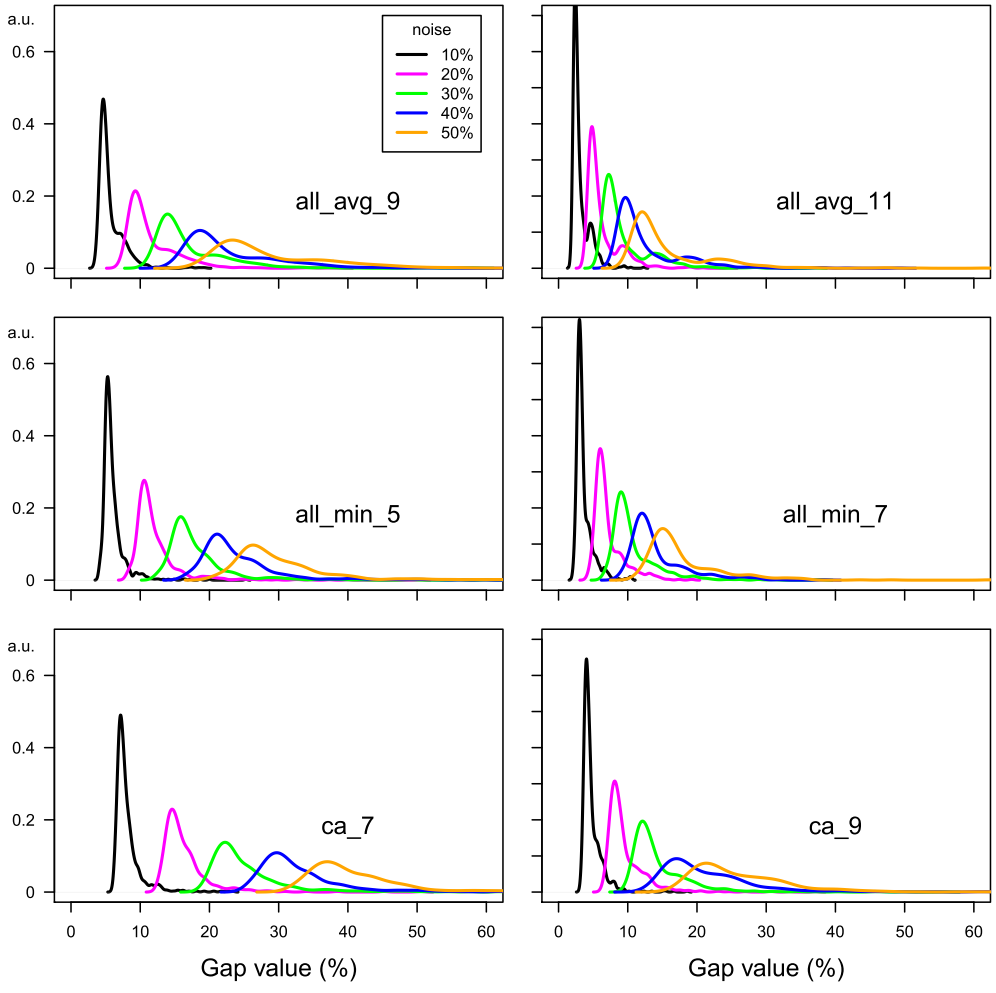


Fig. 3. Distribution of the *Gap* values calculated between the noisy and the initial matrices. The distributions have been plotted for the various matrices and noise levels.

```
   Input: Matrix A
   Output: Gap
 1 begin
 2 |  B_list ← nul;
 3 |  for i=1:500 do
 4 |  |  A′ ← A with p% noises ;
 5 |  |  A″ ← Randomly permuted matrix A′;
 6 |  |  B ← DFS(A″) ;
 7 |  |  B_list ← B_list + B;
 8 |  end
 9 |  C ← SOM(B_list);
10 |  for i=1:n do
11 |  |  for j=i+1:n do
12 |  |  |  if C[i, j] ≥ θ then
13 |  |  |  |  C[i, j] ← 1;
14 |  |  |  end
15 |  |  |  else
16 |  |  |  |  C[i, j] ← 0;
17 |  |  |  end
18 |  |  end
19 |  end
20 |  Gap ← sum(abs(C−A))/density(A);
21 end
```

Fig. 4. Procedure for algorithm validation.

In Fig. 2c, the schemes II and III correspond to the second and the third terms shown in Eq. (5) and calculated in lines 8–12 and lines 13–17 during the algorithm.

2.3. Self-organizing Maps

The **S**elf-**O**rganizing **M**aps (SOM) [47,48] are unsupervised neural networks, recently proposed [27,49] for clustering macromolecular conformations obtained during the sampling of protein conformational space. Here, this method has been used in a different context to cluster the contact matrices obtained by the ordering algorithm described above. Each ordered contact matrix obtained using a given random noise level, was reshaped as a vector to become the SOM input.

These vectors were used to train a periodic Euclidean self-organizing map (SOM), where the map is a three-dimensional matrix. The first two dimensions, defining the map size, were chosen as the integer part of $\sqrt{N}$, where N is the number of contact matrices obtained. The third dimension has the length of the input vectors. Each vector along the third dimension is called a neuron.

After a random initialization of the SOM map, the training is realized iteratively and each input vector is compared to all neurons. The neuron with the smallest Euclidean distance to the input vector, the so-called Best Matching Unit (BMU) is detected. The BMU as well as the neighbors of the BMU in the map are then modified toward the input vector. Indeed, SOM distributes data on the map so that points which are close or far in the descriptor space are also close or far,

Table 2
Determination of the optimal value for the threshold θ used to discretize the real values in range 0–1 in the matrix C to integers 0 or 1. The first column gives the noise levels used for the tests, whereas the five other columns correspond to the tested values of θ. Each table element contains the number of proteins for which a better *Gap* is observed than for all runs at the same noise level. The test of threshold was performed on a subset of 1951 proteins. Calculations were done using Ca_7 contact matrices.

Noise level(%)	Threshold θ				
	0.3	0.4	0.5	0.6	0.7
10%	388	1531	1399	1097	243
20%	110	1341	928	806	129
40%	92	918	646	666	140

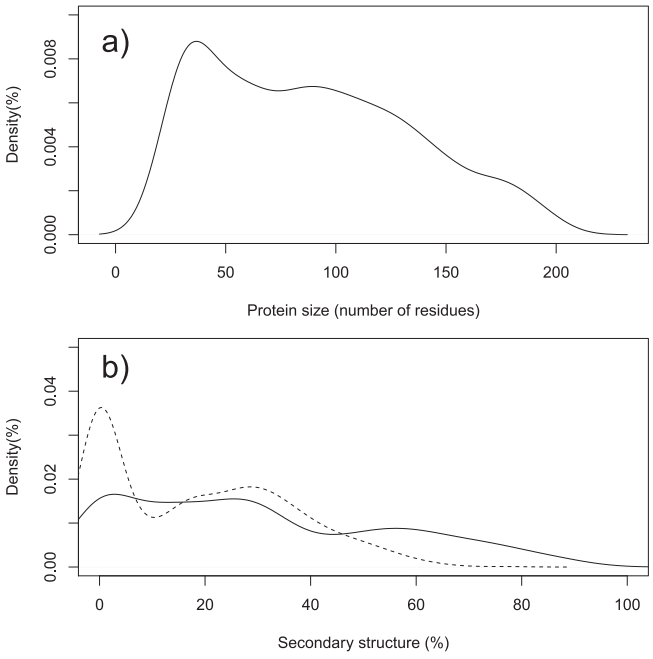


Fig. 5. Distribution of a) protein sizes and b) secondary structure percentages in the protein data-set. In b), the plain (respectively dashed) line represents the percentage of residues assigned to α helices (respectively β strands) in a protein. The secondary structure assignment, realized with DSSP [60], was downloaded from the PDB Web site: www.rcsb.org.

respectively, on the map. To obtain this result, the neurons neighboring the BMU are scaled by the learning rate α through the use of a neighborhood function, a 2D Gaussian, centered on the BMU, and with radius equal to $\frac{1}{8}\sqrt{N}$. The learning rate α decreases from 0.5 to 0.0 with the number of iterations to force convergence.

The conventional **U**nified distance-**matrix** (U-matrix) [49] was used to delineate clusters on the SOMs. For each neuron ν on the map, a corresponding U-matrix element is calculated as the average Euclidean distance between the neuron ν and its eight immediate neighbors:

$$\text{U-height}(\nu) = \frac{1}{8}\sum_{\mu \in N(\nu)} d(\nu,\mu) \quad (6)$$

where $N(\nu)$ is the set of neighbors, and $d(\nu,\mu)$ is the Euclidean distance between vectors ν and μ. In that way, the points of the U-matrix displaying the smallest values correspond to the most homogeneous clusters of neurons.

From the resulting U-matrix, the 3 SOM neurons displaying the 3 lowest minima in the U-matrix were selected and averaged to give the final result for the ordered contact matrix.

The SOM clustering procedure was repeated ten times, and the consensus gap value the most often observed consensus gap value, among these ten experiences, has been taken as the final gap value.

3. Materials and Methods

3.1. Generation of Synthetic Contact Matrices From PDB Structures

The proposed method was validated on a set of original contact matrices obtained from the 3111 protein folds present in the file recent.pdb_select25.nsigma3.0 from the PDBselect database [50]. PDBselect (homepages.thm.de/~hg12640/pdbselect.html) is a list of representative protein chains displaying mutual sequence similarity smaller than 25%. Eight proteins, for which side-chains coordinates are not available in the PDB structure, were removed from the set. Then a smaller subset of 897 proteins was randomly selected from the proteins smaller than 200 residues in order to respect the same distribution of size as in the full set.

Two types of original contact matrices were calculated from the PDB coordinates. The first type was based on the measurement of distances between α carbons of the protein residues, with different cutoffs. Each distance smaller than the cutoff corresponds to a contact. The calculations performed using such contact matrices are called "Ca" in the following: Ca_7 and Ca_9 were performed using distance cutoffs equal to 7 and 9 Å, respectively.

In the second type of contact matrix, the distance between two residues was obtained from the set of all distances between all possible pairs of atoms belonging to these two residues. Between two given residues, the average distance value or the minimum distance value was then compared to the given cutoff in order to detect a

Table 3
Coverage of the CATHv4.2 classification by PDBSelect 2009. Entries from the PDBSelect dataset cover 33 out 41 CATH Architectures. The 8 non-covered CATH Architectures (bold lines) contains only a limited number of CATH Topologies/Fold (11 out of 1391).

CATH class	CATH architecture	# in PDB select	# CATH topologies	# CATH superfamilies	# CATH domains
Mainly	Orthogonal bundle	449	291	582	60,694
alpha	Up-down bundle	222	104	208	25,152
	Alpha horseshoe	22	6	12	3466
	Alpha solenoid	**0**	**2**	**4**	**13**
	Alpha/alpha barrel	5	2	4	977
Mainly alpha	Ribbon	83	26	52	4097
	Single sheet	43	21	42	2426
	Roll	138	40	80	9827
	Beta barrel	147	48	96	28,939
	Clam	1	2	4	84
	Sandwich	266	44	88	51,931
	Distorted sandwich	15	18	36	4037
	Trefoil	11	2	4	1388
	Orthogonal prism	1	2	4	151
	Aligned prism	1	1	2	326
	3-Layer sandwich	3	3	6	439
	3 propeller	**0**	**1**	**2**	**1**
	4 propeller	1	1	2	55
	5 propeller	1	1	2	432
	6 propeller	6	1	2	1084
	7 propeller	2	1	2	1353
	8 propeller	**0**	**1**	**2**	**399**
	2 solenoid	**0**	**1**	**2**	**88**
	3 solenoid	6	3	6	1097
	Beta complex	12	26	52	2105
	Shell	**0**	**1**	**2**	**1**
Alpha beta	Roll	182	60	120	16,507
	Super roll	**0**	**3**	**6**	**56**
	Alpha-beta barrel	43	18	36	16,668
	2-Layer sandwich	495	224	448	66,583
	3-Layer(aba) sandwich	365	126	252	86,984
	3-Layer(bba) sandwich	12	11	22	4298
	3-Layer(bab) sandwich	1	6	12	90
	4-Layer sandwich	14	16	32	10,672
	Alpha-beta prism	2	1	2	458
	Box	**0**	**1**	**2**	**272**
	5-Stranded propeller	2	1	2	185
	Alpha-beta horseshoe	1	3	6	757
	Alpha-beta complex	84	163	326	25,780
	Ribosomal protein L15	**0**	**1**	**2**	**466**
Few secondary structures	Irregular	60	108	216	4519

possible contact. The two distances give rise to two types of contact matrices: All_min and All_avg. Cutoff values of 5 and 7 Å were used for the matrices All_min, whereas larger cutoff values of 9 and 11 Å were used for the matrices All_avg. Indeed, averaging induces a tendency to increase the obtained distance, and larger cutoff values are thus necessary to store sufficient three-dimensional information into the matrix. Overall, for each protein, six different contact matrices were prepared (Ca_7, Ca_9, All_min_5, All_min_7, All_avg_9, All_avg_11).

The Euclidean distance measured between atoms Cα, used in the matrices Ca, corresponds to what could be measured on a Carbon-Carbon dipolar correlation solid-state NMR spectrum (PDSD) [26] recorded in the absence of spin. The second type of distance, making use of all distances between all possible pairs of atoms belonging to these two residues, is related to the comparison of columns extracted for each residue from a 3D ^{1}H-^{15}N HSQC-NOESY spectrum performed by [14]. Indeed, these authors were estimating the differences between the columns by calculating their dot-product. Qualitatively, the averaging used in the matrices All_avg would correspond to the dot product averaged by the number of peaks whereas the minimum value used in the matrices All_min would correspond to the dot-product limited to spectral regions where the intensity is simultaneously maximum in the two columns.

3.2. Noise Introduction

The algorithm was tested on the prepared contact matrices using five different levels of artificial noise (10%, 20%, 30%, 40%, 50%). Introduction of noise to an original contact matrix A is realized in two steps, by addition and deletion of edges. For each iteration of noise introduction, one node v_1 is chosen randomly to run the addition and deletion of edges belonging to $\mathbf{E}(v_1)$, which is the set of edges directly connected to node v_1. During the addition process, one node v_2 whose $(v_1, v_2) \notin \mathbf{E}(v_1)$ is randomly selected, and the $\mathbf{E}(v_1)$ is updated to $\mathbf{E}'(v_1) \leftarrow \mathbf{E}(v_1) \cup (v_1, v_2)$. On the other hand, during the deletion process, one edge $e \in \mathbf{E}(v_1)$ is chosen randomly and the $\mathbf{E}(v_1)$ is updated to $\mathbf{E}''(v_1) \leftarrow \mathbf{E}(v_1) \backslash e$. At the end of an iteration, the set $\mathbf{E}(v_1)$ is replaced by $\mathbf{E}'(v_1) \cup \mathbf{E}''(v_1)$. The number of iterations for noise introduction is chosen as ρn, where n is the size of contact matrix, lr. the number of its rows or columns and ρ the percentage of noise level. The obtained noisy contact matrix is called A' (Fig. 4).

As the proposed algorithm for ordering the contact matrices searches the longest possible trail, it is mandatory that the corresponding graph is connected. But, depending on the protein topology and on the distance cutoff used to define the contact matrix, some contact matrices may be not connected at some noise levels and are removed from the processing. Table 1 displays the number of contact matrices kept along the type of contact matrices and the noise

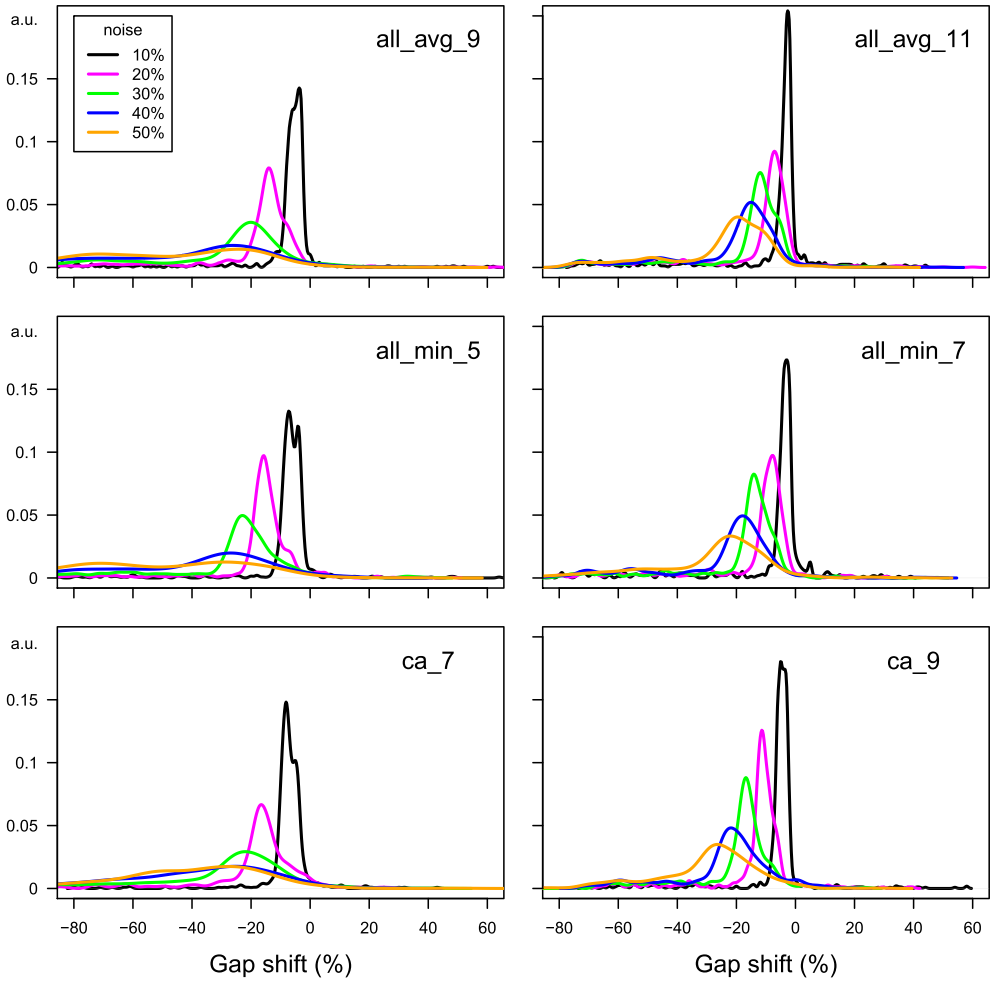


Fig. 6. Difference of values between the largest peak of the *Gap* distribution along the 500 noise realizations and the consensus *Gap* extracted from the repetitions of SOM clustering. The distributions of these differences have been plotted for the various matrices and noise levels and are mostly located in the domain of negative values, which proves that the SOM consensus decreases the gap value.

level. As expected, the number of kept matrices decreases with the noise level. Within a given noise level (a column of Table 1), the contact matrix Ca_7 displays the smallest data-set for all noise levels larger than 10%, and the contact matrix Ca_9 displays the second smallest data-set for noise levels of 40 and 50%. Beside, All_avg_11 retains the maximum number of matrices, rejecting at most 50 matrices.

To estimate the modifications introduced in the contact matrices by the noise addition, Fig. 3 displays the distribution of *Gap* values for each noise level and each type of contact matrix. The peaks of *Gap* values are between one third and one half of the nominal noise level.

3.3. Random Permutation

The input to our ordering problem is obtained by random permutation of the noisy contact matrix A'. The procedure of permutation is the following. One pair of residues is randomly chosen using a pseudo-random generator and their order numbers are exchanged. We repeat this process n times and obtain a disordered contact matrix A'' (line 5 in Fig. 4).

From the point of view of graph theory, permutation does not change the structure of the graph, it changes only the assignment of each node. The node assignment will be ordered using the algorithm described in section Theory (subsection "Presentation of the Algorithm to Produce Individual Ordered Contact Matrices").

3.4. Implementation

The longest trail algorithm was implemented in C++, the code is available at gitlri.lri.fr/chuan/LongestTrailAlgo. The random choice of node was completed using the pseudo-random generator rand() from the library cstlib (www.cplusplus.com/reference/cstdlib/rand/). The SOM clustering was implemented in python and is freely available at: github.com/bougui505/SOM.

3.5. Validation Procedure

The validation on the set of protein matrices was realized on a computer cluster in the following way (Fig. 4). For a given contact matrix A and a given noise level, we generate 500 different noisy and permuted contact matrices A'' (line 5), and run the longest trail algorithm on each of these matrices (line 6). The results are stored in a list of matrices *B_list* (line 7). We then analyze *B_list* using *SOM* which returns a matrix C filled with real numbers between 0 and 1 (line 9). In order to evaluate the result obtained by *SOM*, C is discretized to an integer matrix containing only 0 and 1 values (lines 10–19), using a threshold θ. The similarity between C and A

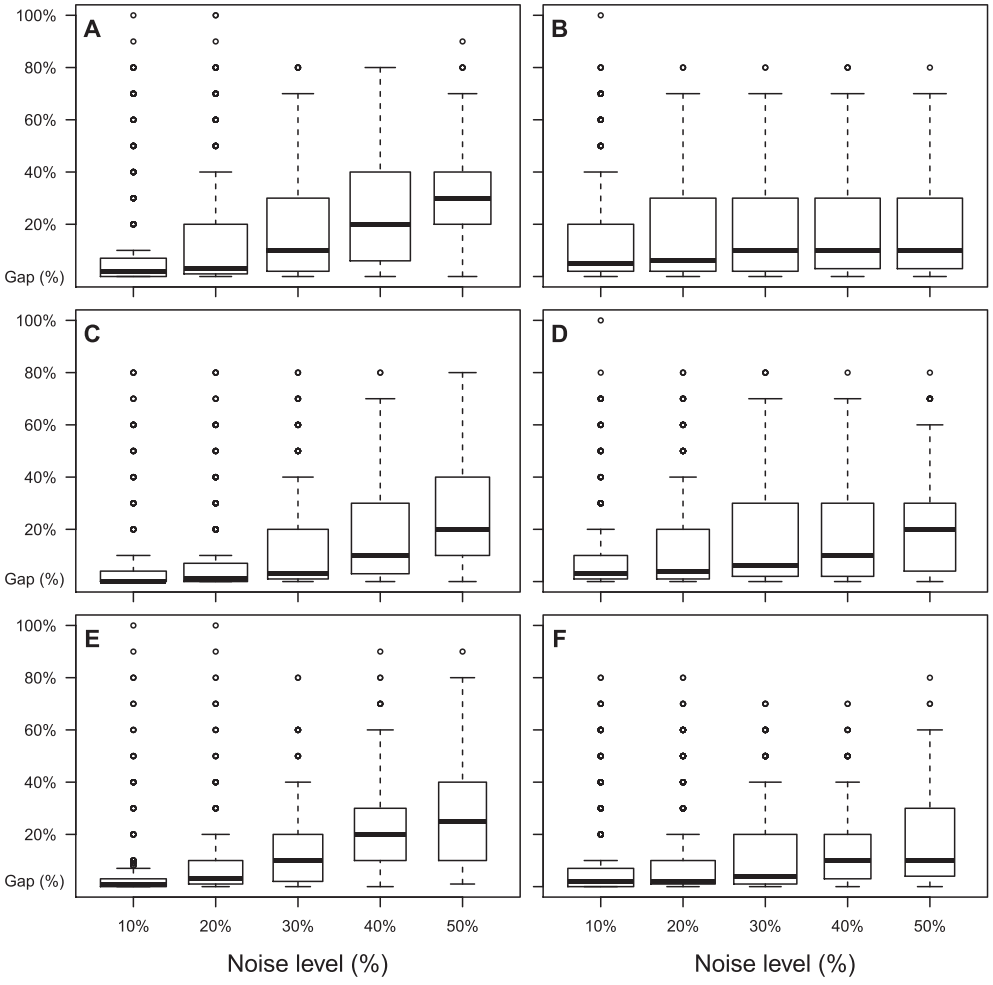


Fig. 7. Box plot of the distribution of consensus *Gap* values (%) for the different amounts of noise levels introduced in the original matrix. The contact matrices have been calculated by the methods: (A) All_avg_9, (B) All_avg_11, (C) All_min_5, (D) All_min_7, (E) Ca_7, (F) Ca_9, defined in section "Generation of Synthetic Contact Matrices From PDB Structures". The average *Gap* value is indicated with a black bar.

is evaluated using a metric *Gap* calculated as the percentage of different elements between *C* and *A* from the total number of nonzero elements of *A* (line 20):

$$Gap = \frac{\sum(|C - A|)}{\sum(|A|) - n}, \tag{7}$$

where n is the size of matrices A and C, $|X|$ returns a matrix with the absolute values of matrix X and $\Sigma(X)$ returns the sum of all values in matrix X.

The optimal threshold θ was determined by running the algorithm on Ca_7 contact matrices with various noise levels. The *Gap* values were calculated for a series of threshold values and all noise levels on a subset of proteins (Table 2). For each θ, the number of proteins which display a *Gap* better than for the other tested cutoffs, is computed. As this number is maximum for $\theta = 0.4$ for all noise levels, $\theta = 0.4$ was considered as the optimal threshold and used in the following. The self-organizing map clustering was repeated ten times and the most often observed *Gap* value was considered as the consensus *Gap* and further analyzed.

4. Results

4.1. Properties of the Processed Proteins

The processed proteins display a quite uniform distribution of sizes in the range of 50–150 residues, corresponding to small or medium folded proteins (Fig. 5a). The distributions of α helix and β strand percentages was compared to the global set of Protein Data Bank (PDB) entries to check whether the subset of protein structures used here is representative of the general knowledge on folded proteins. The processed proteins displaying more than 80% of α helix are relatively less numerous, similarly to the PDB entries in which, among the 15,633 entries corresponding to monomeric proteins with number of residues in the range 10–150, only 1434 (9.2%) contain more than 80% of α secondary structures. The processed proteins display a peak of structures containing no β strand, and the subset of structures with more than 50% of β strand is smaller than the subset of structures displaying less than 50% of β strands. Similarly, among the 15,633 PDB entries, 5514 entries (35.2%) contain less than 10% of β secondary structure, and 1372 entries (8.8%) contain more than 50% of β secondary structure. The set of proteins used here for

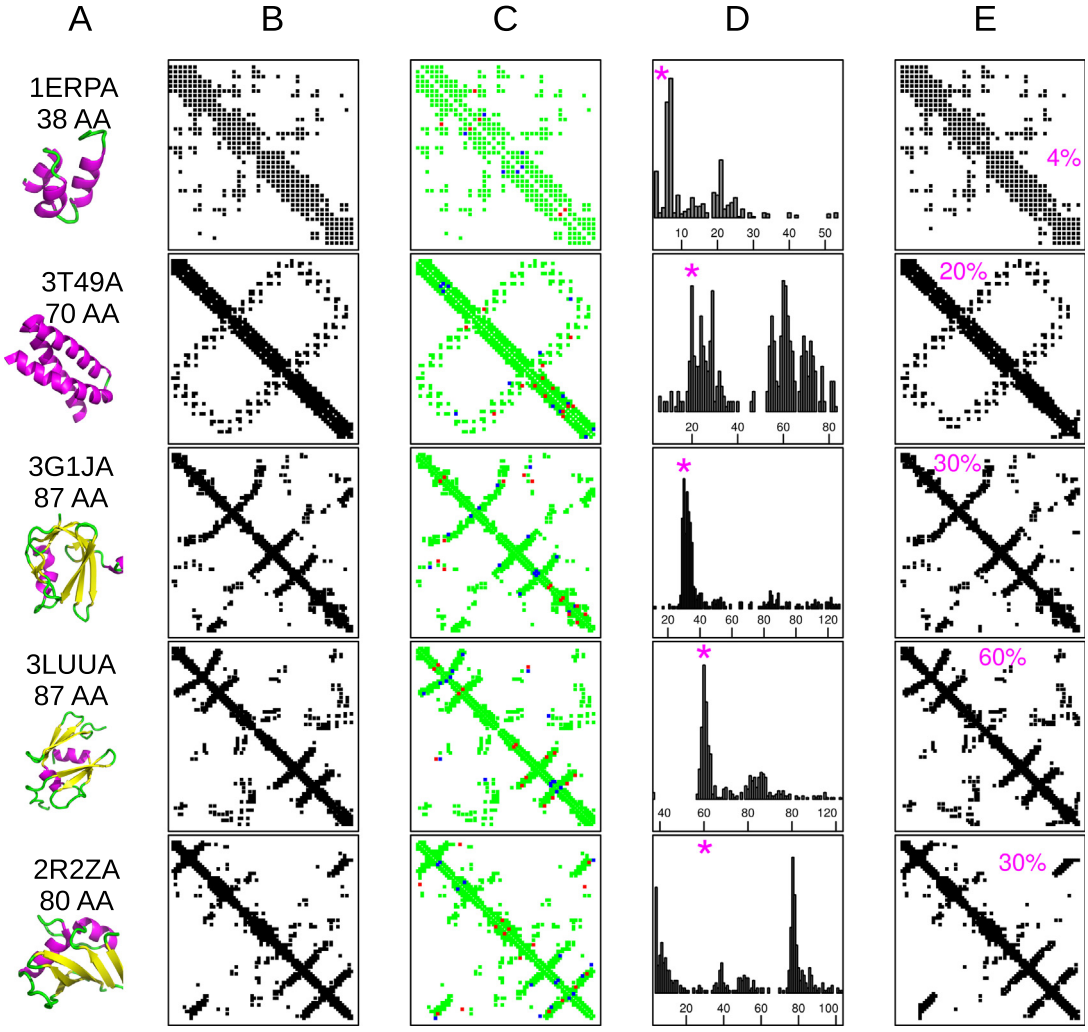


Fig. 8. Examples of calculations performed on contact matrices All_avg_9 with a noise level of 10%. Each line corresponds to the processing of a given protein, illustrated by: (A) cartoon representation of the protein structure with α helices colored in magenta, β strands colored in yellow and loops colored in green, (B) the original contact matrix $\mathcal{A}$, (C) an example of noisy contact matrix $\mathcal{A}'$, where the original data are drawn in green, the added spurious contacts are in blue and the removed contacts are in red, (D) the distribution of *Gap* values obtained for the 500 noise realizations, with the value obtained from SOM consensus marked using a magenta star, (E) the ordered matrix, result from the proposed algorithm, combining DFS and dynamic programming using 500 instances of noise level, followed by a SOM clustering and *Gap* consensus calculation. The corresponding gap value is written in magenta and corresponds to the star mark in the *Gap* distribution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

validation purposes thus displays secondary structure distributions similar to the ones of the whole PDB.

To get a deeper insight into the database quality, the PDBselect folds have been compared to the CATH architectures. Only 8 CATH architectures are not represented in PDB select (Table 3). These 8 architectures correspond only to 11 topologies over the 1391 topologies of CATH. Thus, the PDB select list corresponds to the majority of CATH folds.

4.2. Effect of the Consensus of SOM Results

For each processed contact matrix, the consensus *Gap* value obtained from the ten repetitions of SOM clustering, described in the Materials and Methods, was compared to the most frequent *Gap* value observed among the 500 noise realizations (Fig. 6). The distributions of these differences, calculated for each type of contact matrix, and each noise level, are mostly located in the range of negative values, which proves that the use of the consensus observed among the SOM repetition decreases the *Gap* value between the result and the initial contact matrices.

4.3. Effect of the Noise Level

In order to test the robustness of the proposed algorithm in the presence of spurious contacts, the distribution of obtained *Gap* values is plotted (Fig. 7) for the different noise levels and for the different types of contact matrices defined in Materials and Methods (subsection "Generation of Synthetic Contact Matrices From PDB Structures"). In each case, the *Gap* is smaller than the percentage of noise level. and of the same order of value than the *Gap* between the initial noisy matrix and the initial matrix. (Fig. 3). This shows that the proposed procedure attained the best possible *Gap* value.

Distance cutoff values of 5, 7, 9 and 11 Å used for calculating the original contact matrices Ca, All_avg and All_min have an important effect (Fig. 7) on the *Gap* distribution, in the case of noise levels larger than 30%. Indeed, within a given set of contact matrices (Ca, All_avg or All_min), shorter distance cutoffs (Fig. 7A, C, E) produce larger *Gap* values than longer distance cutoffs (Fig. 7B, D, F), as the average *Gap* values are more than doubled. By contrast, the number of outliers is smaller for shorter cutoffs. Thus, for large noise levels, the approach proposed here is very sensitive to the quality of initial input information, and contact matrices with larger cutoff values (7, 9, 11 Å) are more robust to noise introduction.

Among the different types of tested contact matrices, within a given range of distance cutoff values, the algorithm robustness decreases from All_avg (Fig. 7A, B) to All_min (Fig. 7C, D) and then to Ca (Fig. 7E, F). The increased robustness observed for All_avg with respect to All_min has the following origin: a minimum distance

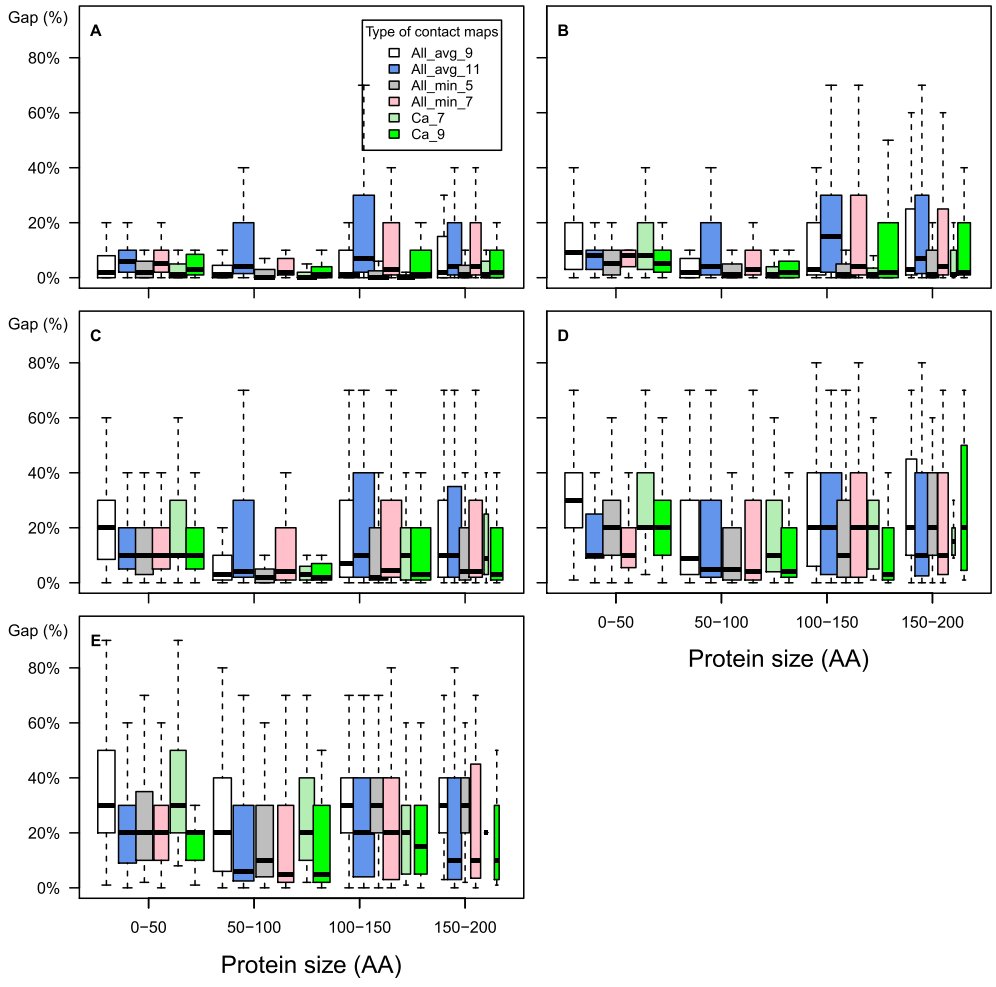


Fig. 9. Box plot of the distribution of consensus *Gap* values (%) according to the size of the processed protein (0–50, 50–100, 100–150, 150–200 residues). The plots (A) to (E) are given for the five levels of noise added to the original contact matrices (from 10% to 50%). For each noise level, the gap distributions are displayed along four ranges of protein size: 0–50, 50–100, 100–150 and 150–200 residues. For each range of protein size, the *Gap* distribution for six types of contact matrices described in subsection "Generation of Synthetic Contact Matrices From PDB Structures", are given. The average *Gap* value is indicated with a black bar.

value is more sensitive to outliers positions of atoms than an average distance value. Similarly, All_min matrices are more robust than Ca matrices, because All_min depends on the position of a larger number of atoms than Ca.

Examples of runs for five proteins, prepared with All_avg_9 and with a noise level of 10% are shown in Fig. 8. Consensus *Gap* values in the range of 4 to 60% are obtained between the initial and the final ordered contact matrices. It is remarkable that patterns of secondary structures and most of the tertiary structures are correctly detected for this whole range of *Gap* values. The distribution of *Gap* values along the 500 noise realizations (Fig. 8C) reveals that the SOM consensus *Gap* value, marked with a magenta star, is located close to the lower limit of all *Gap* values. In most of the displayed cases, this consensus values is not located at the maximum peak of the *Gap* distribution.

A closer examination of the cases where large *Gap* values have been obtained, reveals that these values arise from a shift of the ordered matrix by 1 or 2 residues with respect to the true contact matrix. This shift appears in protein structures, in which the N and the C terminal extremities are close in the 3D space, thus inducing the appearance of cyclic paths within the graphs. In the frame of the algorithm presented here, we do not have a way to overcome this problem.

4.4. Effect of the Protein Size and the Secondary Structure Content

The *Gap* values increase uniformly (Fig. 9) over all ranges of protein sizes except for small proteins (<50 residues), for which somehow larger average *Gap* values are observed than in other size ranges, except for the noise level of 10%. This increase of *Gap* values may arise from the small size *n* of the contact matrix, which is at the denominator in the calculation of the *Gap* (Eq. (7)). The range of 50–100 residues is the one resisting the best to the increase of noise level: this is a good point for the proposed approach as the proteins with 50–100 residues constitute a large proportion of the tested set of proteins (Fig. 5a).

The average *Gap* values and the *Gap* distribution increase for the small percentage of α helices, specially for noise levels larger than 40%. Increasing the percentage of α helices thus improves the robustness of ordering approach: this agrees with the general knowledge that NMR protein assignment is easier for α than for β strand structures. Within a given range of α percentage and for given types of matrices and noise levels, smaller *Gap* values are obtained for larger distance cutoffs.

Conversely, the distributions of *Gap* values along the percentage of β strand (Fig. 11) show that, for most of the contact matrix types, the average *Gap* value increases along the percentage of β strand,

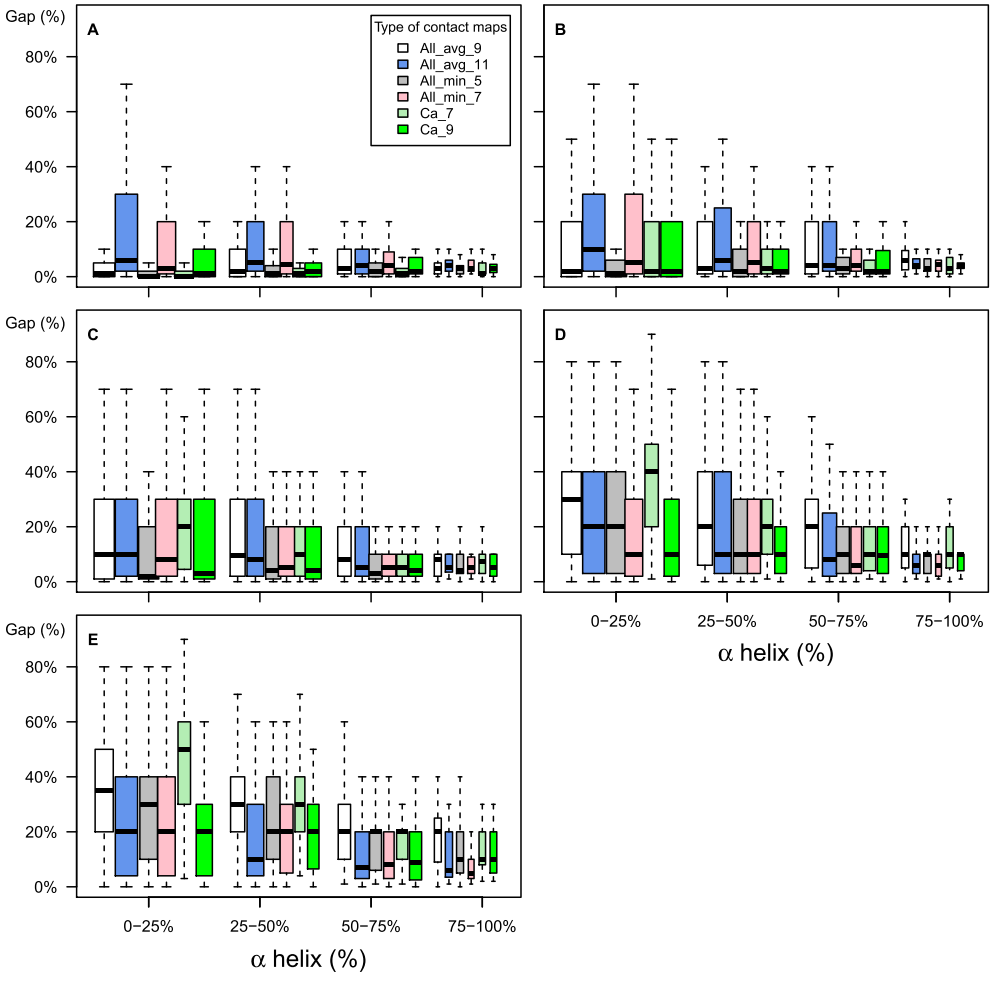


Fig. 10. Box plot of the distribution of consensus *Gap* values (%) along the percentage of α helices in the processed protein. The plots (A) to (E) are given for the five levels of noise added to the original contact matrices (from 10% to 50%). For each noise level, the *Gap* distributions are displayed for four ranges of α helix percentages: 0–25%, 25–50%, 50–75%, 75–100%. For each range, the *Gap* distribution of the six types of contact matrices described in Section "Generation of Synthetic Contact Matrices From PDB Structures", are given. The average *Gap* value is indicated with a black bar.

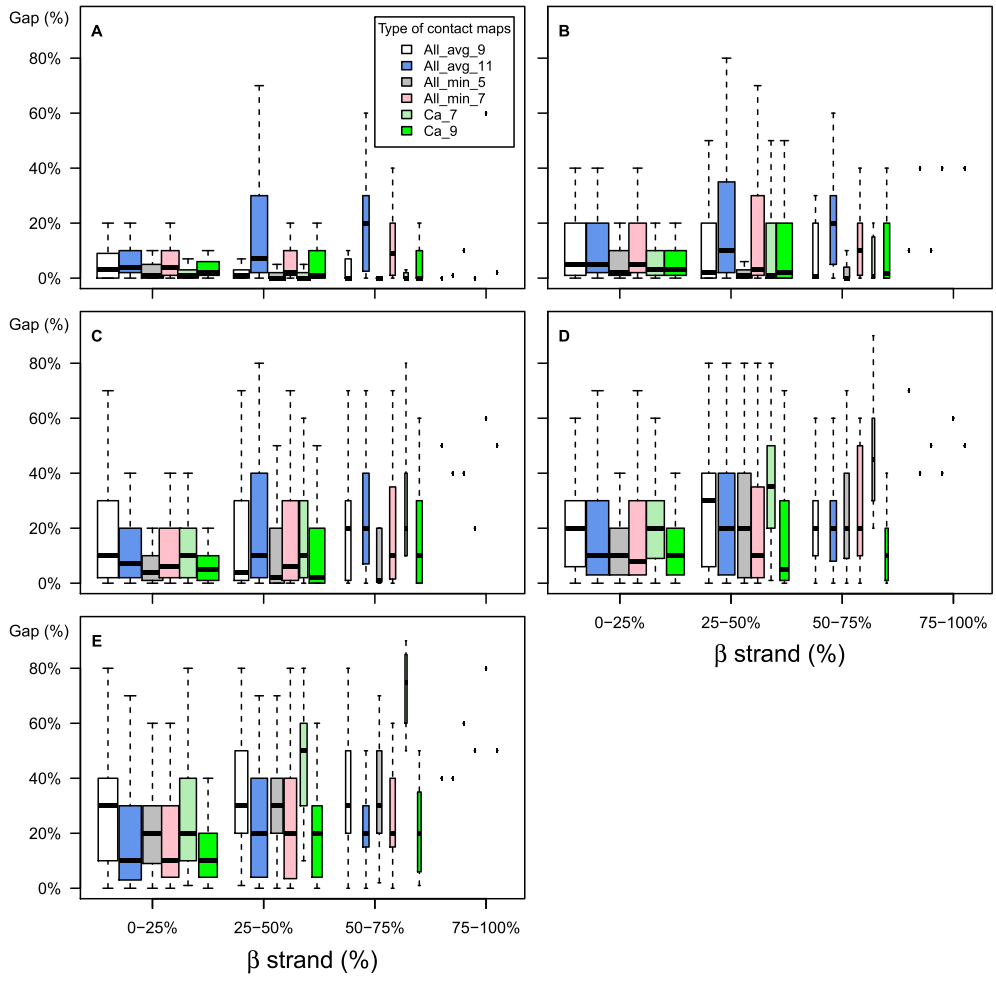


Fig. 11. Box plot of the distribution of consensus *Gap* values (%) along the percentage of β strand. The plot is similar to the one shown for percentage of α helices in Fig. 10.

particularly in the case Ca_9 and All_avg_11, for noise levels up to 40%. Matrices with larger cutoffs display average *Gap* smaller than or around 20%, for noise levels up to 40%. Matrices with shorter distance cutoffs constantly display *Gap* values smaller than 20%, for noise levels up to 20%. Most of the average *Gap* values eventually increase up to 50 % for noise levels between 30 and 50%. The reduced robustness of the proposed algorithm in the case of shorter cutoffs was already observed in Fig. 7. Overall, the average *Gap* values are the sign of a good robustness of the algorithm for percentages of β strands smaller than 75%.

For large percentages of β strands (75–100 %), the average *Gap* values are much worse. This difficulty might certainly stem from the starting hypothesis for the ordering. In this hypothesis, one searches for permutations that maximize the numbers of contacts in the first, second and third sub-diagonals of the contact matrix. However, long-range contacts in β sheets produce bands orthogonal to the diagonal of the contact matrix. For proteins displaying more than 75% of β strands, the presence of many such bands makes the algorithm heuristic irrelevant. By contrast, in Fig. 10, much better *Gap* values are obtained for proteins having more than 75% of α helix, which illustrates the better fit of algorithm heuristics to the pattern of α helices in contact matrices.

4.5. Reconstructing Protein Folds

The algorithm proposed here for ordering and filtering noisy contact matrices could find important applications in the field of protein structure determination, in particular using distance restraints. In order to test the efficiency of result contact matrices to produce reliable protein conformations, the following procedure was applied on 20 entries: 10 of them were taken from the run All_avg_9 with 10% of noise (Fig. 12A–B), and 10 from the run Ca_7 with 40% of noise (Fig. 12C–D), and all representative of different *Gap* values obtained.

For each entry, original and result contact matrices were converted into distance restraints with upper-bound of 9 Å (All_avg_9) or 7 Å (Ca_7). Using these distance restraints, 100 all-atoms conformers were generated with CNS [51] using the simulated annealing protocol [52] implemented in ARIA [53]. Force-field included terms for bond length, bond angles and improper angles, and non-bonded interactions were treated with a simple repulsive term. A conformational database potential was also used [54]. Reconstructed conformers were compared with the original PDB structure by computing the TM-scores [55] on the full PDB sequence. A TM-score larger than 0.5 means that the reconstructed conformers and the original PDB structure have the same fold.

In Fig. 12A, C, for each selected protein, TM-scores obtained for the original and result contact matrices are compared. In the case of the contact matrices defined with All_avg_9 and 10% noise, 6 proteins over 10 display a TM-score larger than 0.5 (Fig. 12A). TM-scores larger than 0.5 are obtained up to 60 % *Gap* and the TM-score decreases linearly with increasing *Gap* values (Fig. 12B). The situation is less favorable for contact matrices defined obtained with Ca_7 and 40% of noise, as only one selected protein displays

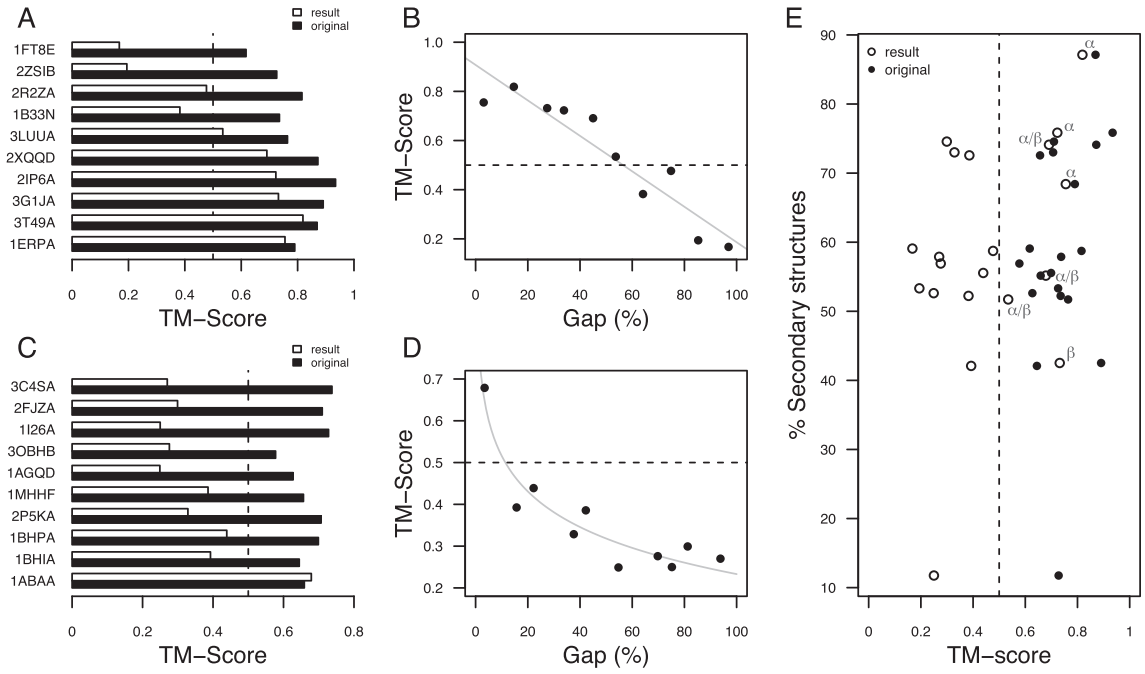


Fig. 12. TM-scores of reconstructed conformations from original and result contact matrices for 20 representative entries using All_avg_9 matrices and 10% noise (A–B) or Ca_7 matrices and 40% noise (C–D). (A, C) Best TM-score of the reconstructed conformations from the original contact matrix (filled black) or result matrix (white) for 10 entries ordered by *Gap* in the result matrix. (B, D) Best TM-score as a function of *Gap* in the result matrix. The linear (B) and exponential (D) fits are plotted in gray. The TM-score value of 0.5, corresponding to situation for which the reconstructed structures have globally the same fold as the initial PDB entry, is shown in all plots with a dashed line. (E) Plot of the percentage of secondary structure versus the TM-score. The original PDB structures and the reconstructed structures are displayed respectively as filled and empty bullets.

a TM-score larger than 0.5 (Fig. 12C). In that case, the TM-score decreases exponentially with the *Gap* (Fig. 12D).

As expected, the noise introduced in the original contact matrices as well as the distance cutoff have a striking influence on the efficiency of result contact matrices to reconstruct 3D conformations of proteins. But, it is encouraging to see that the use of All_avg_9 along with a noise level of 10%, allows to correctly predict protein folds for more than half of the selected proteins.

Overall, the TM-score improves with the increase in the overall percentage of residues belonging to any α or β secondary structure (Fig. 12E). But, the α or β secondary structure does not seem to play a specific role: indeed, 2IP6A and 3G1JA which are respectively rather α and β proteins, display quite similar results. For Ca_7 matrices and 40% noise, the situation is more complicated, but the proteins containing α percentages larger than 30% and β percentages smaller than 20% (2P5KA, 1BHPA, 1BHIA, 1ABAA) display the best TM-scores. The difference of behavior between All_avg_9 with 10% noise and Ca_7 with 40% noise can be put in parallel with the differences of average *Gap* observed for these two types of data-sets (Fig. 7A, E).

4.6. Application to NMR Experimental Data

The general workflow for applying of our approach to experimental NMR data is depicted in Fig. 13A. NMR experiments provide correlations between atoms that are close in space. However correlations are recorded in the frequency space, meaning that one has to assign each frequency (i.e. chemical shift) to the corresponding atoms. To alleviate this step, we propose to label chemical shifts arbitrarily (X_n) and construct an NMR correlation matrix from the peaks observed in the NMR correlation spectrum using the chemical shifts labels as indexes. Once re-ordered, the NMR correlation matrix allows to assign chemical shifts X_n to amino-acids of the protein sequence. Using these assignments, it becomes feasible to reconstruct 3D protein structures from the re-ordered NMR correlation matrix.

To assess the efficiency of this approach, we applied it to matrices obtained from solid-state NMR data on two proteins: MAVS [56] and YadA [57]. Conveniently, solid-state NMR can detect correlations between Carbon atoms which provide correlation matrices similar to contact matrices but that are incomplete and noisy owing to the experimental resolution. Using experimental NMR restraints from the Protein Data Bank (PDB entries 2MS7 for MAVS and 2LME for YadA), NMR correlation matrices were constructed using random labels (X_n) for assigned amino acids (Fig. 13B, E). The re-ordering step yielded matrices with *Gap* values of 39.3% and 0% for MAVS and YadA, respectively (Fig. 13D, F). For YadA, all amino-acids could be assigned correctly, while for MAVS only 10.9% could be correctly assigned but 75.0% were assigned to amino-acids next to the correct ones in the sequence. Yet, a trained NMR expert could easily correct these artefacts by checking NMR chemical shift signatures of given amino-acid types. Finally, using a single distance restraint per NMR correlation and a upper-bound of 7 Å, we were able to reconstruct 3D structures of MAVS and YadA with acceptable TM-scores when compared to the structures from the PDB (Fig. 13D, G). It is worth noting that the proteins analyzed here belong to larger assemblies (YadA trimer and MAVS filament) and that one should not exclude the possibility that inter-protein NMR correlations, incompatible with the 3D structure of an isolated protein, may be present in the data. Additionally, we anticipate that higher-accuracy 3D structures could be generated using a more quantitative restraint upper-bound derived from the intensity of NMR signals (so far not used here) and a more complete restraint set using assigned NMR chemical shifts (e.g. backbone dihedral angle restraints).

5. Discussion

In the present work, an efficient algorithm has been proposed to order protein contact matrices, and reconstruct their corresponding protein fold, given a set of spatial proximities between amino acids. The proposed approach is based on the successive application of two

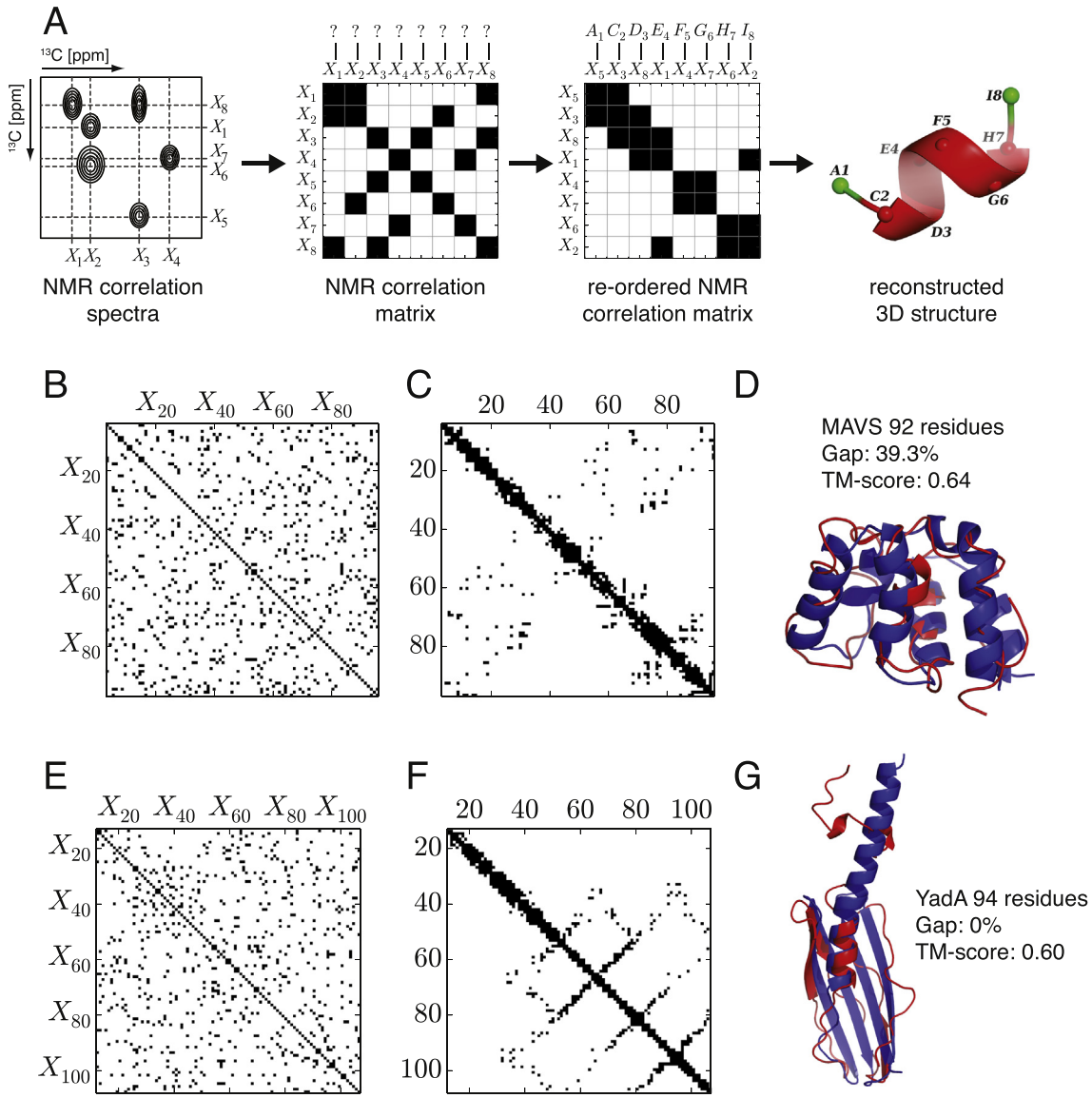


Fig. 13. (A) Workflow for application to NMR data. A NMR correlation spectrum is converted to a NMR correlation matrix by arbitrary labeling of chemical shifts. Re-ordering of the NMR correlation matrix yields assignments to amino-acids and allows 3D structure reconstruction. (B, E) Initial correlation matrices from solid-state NMR data for MAVS and YadA, respectively. (C, F) Re-ordered NMR correlation matrices for MAVS and YadA, respectively. (D, G) 3D structures of the PDB (blue) and reconstructed structures (red) for MAVS and YadA using the re-ordered NMR correlation matrices. Protein sizes, *Gap* values and TM-scores are also indicated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

algorithms: (i) ordering of the contact matrix by the concomitant use of DFS and dynamic programming algorithm, (ii) clustering of the obtained solutions by the self-organizing map (SOM) approach [27,49].

The assignment problem was formalized in the present work as the resolution of the longest trail problem [36,42]. To the best of our knowledge, this is the first time in the field of structural biology that this problem is formalized in a such way. The combined use of dynamic programming and self-organizing map approaches is one of the innovative aspects of the work and makes the proposed algorithm an original approach for the ordering of contact matrices of proteins.

The robustness of the approach has been extensively tested by randomly adding and removing contacts into the contact matrix, up to 50% of the matrix size. The repeated analysis of the noisy matrices allows to determine average *Gap* values on a set of about 900 proteins with sizes spanning the range of 20–200 residues and being a representative subset of the existing protein folds. Average *Gap* values smaller than 30% have been obtained for all protein sizes, when the distances between two residues are calculated using all atom positions from these two residues (contact matrices All_min and All_avg).

The proposed algorithm produces mostly similar *Gap* values along the percentages of α and β secondary structures. Nevertheless, a better robustness is overall observed for large percentages of α secondary structures than for large percentages of β secondary structures. This is due to the heuristics used to order the matrix, which fits better α secondary structure elements.

These results are quite encouraging, as the visual observation of calculation examples shows that a *Gap* value of 20% produces result contact matrices (Fig. 8E) close to the original ones (Fig. 8A). Also, the reconstruction of protein folds from result contact matrices and

the computation of TM-scores [55] confirm that correct protein folds can be predicted for the majority of proteins if the contact matrices All_avg_9 and a 10% noise level are used (Fig. 12).

The 3D reconstruction of the protein folds has been quickly evaluated, as the 3D fold reconstruction is not the main purpose of the present work, but rather a determination of the correct contact matrices. Nevertheless, reasonable 3D folds could be reconstructed from ordered matrices obtained from synthetic (Fig. 12) and experimental (Fig. 13) data.

An important aspect of the approach presented here concerns its applicability to real-life problems encountered in structural bioinformatics. Distance-based techniques in structural biology could benefit from the approach proposed here. Indeed, in quite different domains, an application example would be the metagenomics data for assembling DNA fragments [58] or photo-activated cross-linking detected by mass spectrometry [59] for assembling peptide fragments. For the second example, the peptides would be indexed by the mass-to-charge ratio. In both cases, the measurement of proximities does not give direct access to the position of observed residues in the protein or DNA sequence. The method proposed for processing contact matrices would provide such assignment, which would ease the subsequent approaches used for reconstructing 3D structures.

6. Conclusion

The algorithm proposed here represents an original approach to the problem of ordering protein contact matrices. It displays robust behavior with respect to random noise level introduced in the contact matrix, and with respect to the protein size and to the percentages of α and β secondary structures. The ordered contact matrices contain enough information to allow the correct prediction of protein fold for a low level of noise.

Conflicts of Interest

None.

Acknowledgments

This work was funded by the European Union (FP7-IDEAS-ERC 294809 to MN). Chuan Xu thanks the support for PhD thesis. University Paris-Sud, CNRS and Institut Pasteur are acknowledged for funding.

References

[1] Kyne C, Crowley PB. Grasping the nature of the cell interior: from physiological chemistry to chemical biology. FEBS J 2016;283:3016–28.
[2] Ferber M, Kosinski J, Ori A, Rashid UJ, Moreno-Morcillo M, Simon B. et al. Automated structure modeling of large protein assemblies using crosslinks as distance restraints. Nat Methods 2016;13:515–20.
[3] Purdy MD, Bennett BC, McIntire WE, Khan AK, Kasson PM, Yeager M. Function and dynamics of macromolecular complexes explored by integrative structural and computational biology. Curr Opin Struct Biol 2014;27:138–48.
[4] van den Bedem H, Fraser JS. Integrative, dynamic structural biology at atomic resolution-it's about time. Nat Methods 2015;12:307–18.
[5] Tyagi S, Lemke EA. Single-molecule FRET and crosslinking studies in structural biology enabled by noncanonical amino acids. Curr Opin Struct Biol 2015;32:66–73.
[6] Webb B, Sali A. Comparative protein structure modeling using modeller. Curr Protoc Bioinformatics 2014;5.6.1–5.6.32.
[7] Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R. et al. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 2011;487:545–74.
[8] van Zundert GC, Rodrigues JP, Trellet M, Schmitz C, Kastritis PL, Karaca E. et al. The HADDOCK2.2 web server: user-friendly integrative modeling of biomolecular complexes. J Mol Biol 2016;428:720–5.
[9] Yu J, Vavrusa M, Andreani J, Rey J, Tufféry P, Guerois R. InterEvDock: a docking server to predict the structure of protein-protein interactions using evolutionary information. Nucleic Acids Res 2016;44:W542–49.
[10] Zimmerman DE, Montelione GT. Automated analysis of nuclear magnetic resonance assignments for proteins. Curr Opin Struct Biol 1995;5:664–73.
[11] Güntert P. Automated structure determination from NMR spectra. Eur Biophys J 2009;38:129–43.
[12] Frueh DP. Practical aspects of NMR signal assignment in larger and challenging proteins. Prog Nucl Magn Reson Spectrosc 2014;78:47–75.
[13] Sattler M, Schleucher J, Griesinger C. Heteronuclear multidimensional nmr experiments for the structure determination of proteins in solution employing pulsed field gradients. Prog Nucl Magn Reson Spectrosc 1999;34:93–158.
[14] Bartels C, Wüthrich K. A spectral correlation function for efficient sequential NMR assignments of uniformly (15)N-labeled proteins. J Biomol NMR 1994;4:775–85.
[15] Branden C, Tooze J. Introduction to protein structure. New York: Garland Science; 1998.
[16] Vassura M, Lena PDi, Margara L, Mirto M, Aloisio G, Fariselli P. Blurring contact maps of thousands of proteins: what we can learn by reconstructing 3D structure. BioData Min 2011;4:1.
[17] Jones DT, Bryson K, Coleman A, McGuffin LJ, Sadowski MI, Sodhi JS. Prediction of novel and analogous folds using fragment assembly and fold recognition. Proteins 2005;61 Suppl 7:143–51.
[18] Vullo A, Walsh I, Pollastri G. A two-stage approach for improved prediction of residue contact maps. BMC Bioinformatics 2006;7:180.
[19] Marks DS, Colwell LJ, Sheridan R, Hopf TA, Pagnani A, Zecchina R. Protein 3D structure computed from evolutionary sequence variation. PloS One 2011;6:e28766.
[20] Ding W, Xie J, Dai D, Zhang H, Xie H, Zhang W. CNNcon: improved protein contact maps prediction using cascaded neural networks. PLoS One 2013;8:e61533.
[21] Kaján L, Hopf TA, Marks DS, Rost B. FreeContact: fast and free software for protein contact prediction from residue co-evolution. BMC Bioinformatics 2014;15:85.
[22] Kosciolek T, Jones DT. De novo structure prediction of globular proteins aided by sequence variation-derived contacts. PloS One 2014;9:e92197.
[23] Adhikari B, Bhattacharya D, Cao R, Cheng J. CONFOLD: residue-residue contact-guided ab initio protein folding. Proteins 2015;83:1436–49.
[24] Adhikari B, Nowotny J, Bhattacharya D, Hou J, Cheng J. ConEVA: a toolbox for comprehensive assessment of protein contacts. BMC Bioinformatics 2016;17:517.
[25] Yang J, Jin QY, Zhang B, Shen HB. R2C: improving ab initio residue contact map prediction using dynamic fusion strategy and Gaussian noise filter. Bioinformatics 2016;32:2435–43.
[26] Castellani F, van Rossum B, Diehl A, Schubert M, Rehbein K, Oschkinat H. Structure of a protein determined by solid-state magic-angle-spinning NMR spectroscopy. Nature 2002;420:98–102.
[27] Bouvier G, Duclert-Savatier N, Desdouits N, Meziane-Cherif D, Blondel A, Courvalin P. et al. Functional motions modulating VanA ligand binding unraveled by self-organizing maps. J Chem Inf Model 2014;54(1):289–301.
[28] DouglasBrent West. Introduction to graph theory.vol. 2. New Jersey: Prentice hall Upper Saddle River; 2001.
[29] Zamfirescu T. On longest paths and circuits in graphs. Math Scand 1976;38(2):211–39.
[30] Garey MR, Johnson DS, Tarjan R. The planar Hamiltonian circuit problem is NP-complete. SIAM J Comput 1976;5(4):704–14.
[31] Damaschke P. The Hamiltonian circuit problem for circle graphs is NP-complete. Inf Process Lett 1989;32(1):1–2.
[32] Müller H. Hamiltonian circuits in chordal bipartite graphs. Discret Math 1996;156(1):291–8.
[33] Karger D, Motwani R, Ramkumar G. On approximating the longest path in a graph. Algorithmica 1997;18(1):82–98.
[34] Ioannidou K, Mertzios GB, Nikolopoulos SD. The longest path problem has a polynomial solution on interval graphs. Algorithmica 2011;61(2):320–41.
[35] Gabow HN. Data structures for weighted matching and nearest common ancestors with linking. Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California. 1990. p. 434–43.
[36] Szécsi V. On the minimal length of the longest trail in a fixed edge-density graph. Cent Eur J Math 2013;11(10):1831–7.
[37] Abouelaoualim A, Das KC, Faria L, Manoussakis Y, Martinhon C, Saad R. Paths and trails in edge-colored graphs. Theor Comput Sci 2008;409(3):497–510.
[38] Catlin PA. Spanning trails. J Graph Theory 1987;11(2):161–7.
[39] Li H, Yang W. A note on collapsible graphs and super-Eulerian graphs. Discret Math 2012;312(15):2223–7.
[40] Cho HG, Zelikovsky A. Spanning closed trail and Hamiltonian cycle in grid graphs. In: J, Staples P, Eades N, Katoh A, Moffat, eds. Algorithms and computations. Berlin, Heidelberg: Springer Berlin Heidelberg1995. p. 342–51.
[41] Luo W, Chen ZH, Chen WG. Spanning trails containing given edges. Discret Math 2006;306(1):87–98.
[42] Yang W, Lai H, Wu B. A Note on Edge Degree and Spanning Trail Containing Given Edges.2017.arXiv preprint arXiv:170607274.
[43] Dorn F, Moser H, Niedermeier R, Weller M. Efficient algorithms for eulerian extension and rural postman. SIAM J Discret Math 2013;27(1):75–94.
[44] Eiselt HA, Gendreau M, Laporte G. Arc routing problems, part i: the Chinese postman problem. Oper Res 1995;43(2):231–42.

[45] Kruskal JB. On the shortest spanning subtree of a graph and the traveling salesman problem. Proc Am Math Soc 1956;7(1):48–50.
[46] Tarjan R. Depth-first search and linear graph algorithms. SIAM J Comput 1972;1(2):146–60.
[47] Kohonen T. Self-organized formation of topologically correct feature maps. Biol Cybern 1982;43:59–69.
[48] Kohonen T. Self-organizing maps. Heidelberg, Germany: Springer Series in Information Sciences; 2001.
[49] Bouvier G, Desdouits N, Ferber M, Blondel A, Nilges M. An automatic tool to analyze and cluster macromolecular conformations based on self-organizing maps. Bioinformatics 2015;31(9):1490–2.
[50] Griep S, Hobohm U. PDBselect 1992–2009 and PDBfilter-select. Nucleic Acids Res 2010;38:D318–9.
[51] Brünger AT, Adams PD, Clore GMarius, DeLano WL, Gros P, Grosse-Kunstleve RW. et al. Crystallography & NMR system: a new software suite for macromolecular structure determination. Acta Crystallogr D Biol Crystallogr 1998;54(Pt 5):905–21.
[52] Nilges M, Clore GM, Gronenborn AM. Determination of three-dimensional structures of proteins from interproton distance data by dynamical simulated annealing from a random array of atoms. Circumventing problems associated with folding. FEBS Lett 1988;239(1):129–36.
[53] Bardiaux B, Malliavin TE, Nilges M. ARIA for solution and solid-state NMR. Methods Mol Biol (Clifton, NJ) 2012;831:453–83.
[54] Kuszewski J, Gronenborn AM, Clore GM. Improving the quality of NMR and crystallographic protein structures by means of a conformational database potential derived from structure databases. Protein Sci 1996;5(6):1067–80.
[55] Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins Struct Funct Bioinf 2004;57(4):702–10.
[56] He L, Bardiaux B, Ahmed M, Spehr J, König R, Lünsdorf H. et al. Structure determination of helical filaments by solid-state NMR spectroscopy. Proc Natl Acad Sci U S A 2016;113(3):E272–81. https://doi.org/10.1073/pnas.1513119113.
[57] Shahid SA, Bardiaux B, Franks WT, Krabben L, Habeck M, van Rossum BJ. et al. Membrane-protein structure determination by solid-state NMR spectroscopy of microcrystals. Nature Methods 2012;9(12):1212–7. https://doi.org/10.1038/nmeth.2248.
[58] Marie-Nelly H, Marbouty M, Cournac A, Flot J, Liti G, Parodi D. et al. High-quality genome (re)assembly using chromosomal contact data. Nat Commun 2014;17:5695.
[59] Brodie N, Popov K, Petrotchenko E, Dokholyan N, Borchers CH. Solving protein structures using short-distance cross-linking constraints as a guide for discrete molecular dynamics simulations. Sci Adv 2017;3:e1700479.
[60] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983;22:2577–637.