

# Computation and analysis of channels in protein dynamics

Petr Beneš, Petr Medek, Jiří Sochor

**Abstract**—When performing complex analysis of protein molecules, chemists need to analyze behavior of channels in protein molecules. For analyzing channels in a static molecule, there are various methods which can be used. However, no specialized method that would be able to follow an opening and closing of channels in a moving molecule exists. This paper surveys several possible methods, which are able to partition channels computed in a sequence of molecule snapshots into clusters representing progress of channels in the sequence. All presented methods are built on top of previous static methods of channel computation. They process molecule samples in time and are based on computing channels independently in these samples. Computed channels are classified subsequently. The methods were piloted on a real data. The results are discussed and main advantages and disadvantages of methods are mentioned.

**Keywords**—channel, cluster analysis, protein molecule, protein dynamics.

## I. INTRODUCTION

The behavior of protein molecules is dependent on the presence of channels connecting a cavity inside a protein with its surface. The channels serve as paths for transfer of small molecules in and out of proteins and they play an important role in chemical reactivity. The need of determination of channels and their properties led to the design and development of various methods for channel computation. However, almost all methods deal with static molecules and do not take dynamic properties of a protein into account.

Protein molecules are continuously moving in time. Chemists typically observe this behavior by computer simulations, in which mutual interactions and physical forces among atoms are considered [1]. Even though the movement, which is simulated, is in reality continuous, only samples of this movement in time are stored as results of the simulation. In given intervals the molecule state (i.e. atom positions) is saved and the samples are analyzed afterwards. These samples will be referred to as molecule snapshots.

As a molecule is moving in time, its channels are also changing. A wide channel computed in a random instance of time (a snapshot) might in reality be open only for a short time and thus biochemically unimportant. This is why a detailed analysis of channels in protein dynamics is desired. If channels and their parameters were analyzed in the dynamics, it would be easier to judge on substrate molecules which could pass through the channel into the specified cavity (the active site) inside a protein.

So far, chemists processed channels in snapshots manually by observing molecule surface and finding holes in it. Even

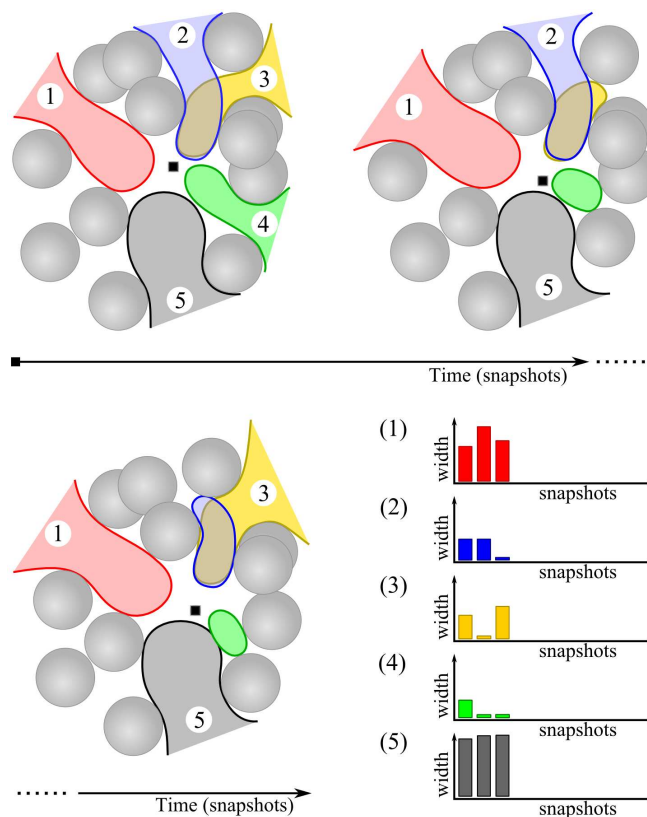


Fig. 1. Progress of channels in a sequence of molecule snapshots and charts depicting channel width.

though they were able to classify channels and gather information about channel properties this way, it was highly inaccurate and due to the manual processing also time consuming. This approach also required a prior experience with the molecule and its channels.

The aim of this paper is to present several methods which process all snapshots in a sequence obtained from simulation and provide an information about behavior of channels. The methods should be able to determine, whether the channels are opened or closed in a particular snapshot and provide an information about channel parameters such as bottleneck radius and channel stability in the sequence (see Fig. 1 and Fig. 2 for channel definition). The methods described in this paper do not rely on a particular algorithm of channel computation and any method able to compute channels could be used.

## II. RELATED WORK

Channels represent an empty space through which the active site is accessible from the protein surface. A formal definition

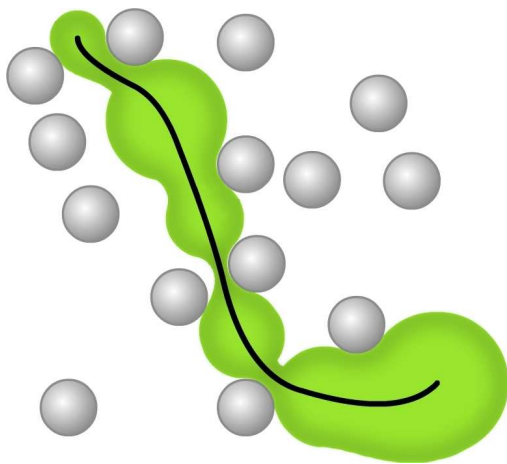


Fig. 2. Demonstration of channel definition. Channel centerline and volume.

of a channel (formerly tunnel) was proposed in [2] (see Fig. 2). The channel is defined as a centerline and a volume. The centerline is any curve leading from the active site to the surface. The volume is formed by the union of spheres inserted at each point of the centerline so that these spheres have maximum radius but do not cross any of the molecule atoms.

The first method of channel computation in a static molecule based on space rasterization was presented in [3]. The molecule was sampled and the three-dimensional grid was processed afterwards searching for the collision-free path.

More sophisticated methods which did not suffer from rasterization disadvantages were proposed later in [2], [4], [5]. These methods typically compute the Voronoi diagram or its dual, the Delaunay triangulation of the molecule and process it by searching for paths leading along Voronoi edges. This ensures that the widest channels would be found. Extensions proposed in [6], [7] allowed employing additional criteria for channel computation and a more accurate computation of more than one channel in a single molecule.

The most important method for analysing protein dynamic sequences is a simulation based on physical interactions of a certain substrate molecule with a protein. These simulations are in fact only the extensions which modify the general approach of computation of protein dynamics by positioning a certain substrate molecule in the active site. In addition, random forces are applied to the substrate molecule during the simulation to assist the substrate molecule in leaving the protein. A path used by the substrate for leaving the molecule is returned as a result of the simulation. The method is very accurate. The resulting channel is a real exit path regarding all physical properties. However, it is very time consuming, since a computation of a single channel takes hours. Furthermore, the method is random-based, therefore it doesn't mean that no channel exist in the molecule if no channel is computed.

The methods proposed in this paper are based on clustering principles [8]. Cluster analysis is well-known and many clustering schemes exist. Presented methods use K-means clustering [9], QT clustering [10] and graph cutting clustering [11].

### III. CLUSTERING METHODS

As mentioned above, the analysis of protein dynamics does not depend on a particular algorithm for computation of channels. Nevertheless, the method presented in [2] with extensions proposed in [6], [7] seems to be the most convenient when used to compute only relevant and possibly non-duplicate channels in each snapshot.

The aim of all presented clustering methods is to find biochemically relevant channels which could be used by a substrate molecule to penetrate to the active site. Factors which influence whether a channel is suitable or not, are the width of a channel during the dynamics and the number of snapshots in which a channel was open, i.e. its minimum width exceeded some threshold. With the information about channel behavior, chemists can invoke certain structural changes in the specific part of the molecule near a channel, so that in the modified molecule the channel may be wider and more stable.

An input of all methods is a set of channels computed in each snapshot of the sequence of protein dynamics. The channels are computed separately in each snapshot. A fixed number of channels could be computed or there could be computed all channels satisfying a specified condition. All methods have to deal with the issue of classification and annotation of channels. Classification of channels enables to determine that two channels – each of them computed in a different snapshot – actually represent the same channel.

More precisely, this can be seen as a progression of a certain channel in time. Having  $n$  snapshots and  $m_i, i = 1..n$  channels computed in each snapshot, a set  $S$  of size  $m = \sum_{i=1}^n m_i$  containing all these channels is computed. The classification can also be viewed as partitioning this large set of channels computed in all snapshots into clusters. As the result of cluster analysis, the sets of similar channels are formed. Since it is known in which snapshot the channel was computed, the results of cluster analysis can be interpreted as a progression of the specific channel in time.

All methods described in this section, except the last one (graph cutting), require prior knowledge of a set of channels, which are representatives of the final clusters. Computed channels are divided into clusters by measuring their similarity with these reference channels. There are several possible ways of obtaining the reference data. Selected channels computed in a single snapshot or substrate escape routes from protein dynamics simulations could be used.

#### A. Distance function

An important component of any clustering algorithm is the distance measure between data points. In this case, each data point corresponds to a channel. Despite the fact that a channel was defined as a centerline and a continuous volume, it is easier to maintain both the centerline and volume sampled. Each channel is then approximated by a set of spheres inserted into several points of the centerline. For measuring a distance between two channels  $A, B$  represented by sets of spheres  $sph(A)$  and  $sph(B)$ , the modified Hausdorff distance  $D$  is used:

$$D(A, B) = \min(dist_{ab}, dist_{ba})$$

where

$$dist_{ab} = \left( \sum_{a \in sph(A)} \min_{b \in sph(B)} (d(a, b)) \right) / |sph(A)|$$

$$dist_{ba} = \left( \sum_{b \in sph(B)} \min_{a \in sph(A)} (d(a, b)) \right) / |sph(B)|$$

and  $d(a, b)$  is the smallest euclidean distance between surfaces of spheres  $a$  and  $b$ . The distance function as is defined is purely geometric. However, it can be extended to take other criteria such as nearby residues or additional biochemical properties of the molecule near the channel into account.

The time needed to compute channels in each snapshot is considered constant all methods have to compute the channels. Assume we have  $n$  snapshots and after channel computation we get the set  $S$  of size  $m$  containing all computed channels.

### B. Simple clustering

The trivial method compares channels computed in each snapshot to the reference channels. Each computed channel is assigned to the cluster whose reference channel is the closest to the channel. If no reference channel can be found within a specified distance, the channel is assigned to no cluster (and eventually could become a reference channel for a new cluster). As will be demonstrated in the results, this method is inaccurate and it is difficult to set a threshold which would classify channels as similar with a reference channel or not.

Since this method has a fixed number of precomputed clusters to be checked in each snapshot, the method has the complexity of  $\mathcal{O}(m)$ .

### C. K-means clustering

A more advanced algorithm is based on K-means clustering. For channel clusterisation, a random selection of initial cluster centers is not suitable. Instead, reference channels are used as the initial centers of clusters.

As a first step of the algorithm, channels are assigned to the nearest cluster center. A channel belongs to the cluster whose center is the closest to the channel. When all channels are divided into clusters, a new center of each cluster is determined in the following way. For each channel belonging to the cluster, the sum of distances to all other channels in the cluster is computed. The channel with the minimal sum is selected as a new cluster center. If a convergence criterion is not met, the clusters are discarded and the whole process repeats.

A disadvantage of this method is a necessity to select an initial set of cluster centers. As will be demonstrated in the results, the only reliable way is to manually prepare a set of channels which represent clusters appropriately.

This method evaluates the distances of all channels within each of  $k$  clusters. If some of the clusters covers the whole set, the complexity is  $\mathcal{O}(m^2)$ . However, in real cases, channels are distributed into clusters uniformly. In the ideal case the complexity is  $\mathcal{O}((m/k)^2)$ .

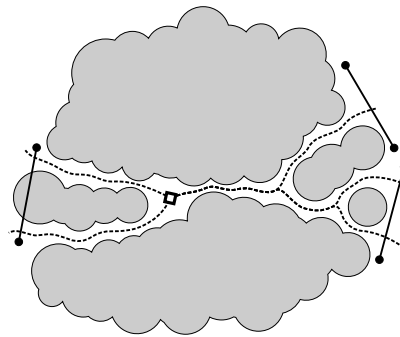


Fig. 3. Example of the problem of fixed cluster radius. Although the two channels on the left side are different they are associated to one cluster. On the other hand, channels on the right are similar and are assigned to separate clusters.

### D. QT based clustering

For the determination of reference channels, more sophisticated method can be used. This method analyzes all channels computed on the sequence of snapshots and selects the best representatives for each cluster. For each atom, the information about how many channels led near this particular atom is maintained. The channel, which is the closest to the atoms with the highest values, is taken as a reference channel. Another possibility is to use the channel which has the most channels in its neighborhood. The actual clustering algorithm is based on QT clustering method:

- **Input:** The set of all channels  $S$ . Each channel from  $S$  has a value denoting its importance computed according to the procedure described above. Threshold value  $t$ . Number of clusters  $k$ .
- **Output:** The set of clusters of channels.
- **Step 1.** Set the channel from  $S$  with the highest value as a center  $c_C$  of a new cluster  $C$ .
- **Step 2.** Assign all channels from  $S$  which are closer to  $c_C$  than  $t$  into  $C$ .
- **Step 3.** Remove channels assigned to  $C$  from  $S$ .
- **Step 4.** If the number of created clusters is lower than  $k$  (and  $|S| > 0$ ), go to **Step 1**.

This method has an obvious disadvantage. The radius of all clusters is fixed. However, some clusters could be very small but still important while others could be bigger than the selected radius (see Fig. 3).

The complexity of this method is  $\mathcal{O}(m * k)$  where  $k$  is fixed and it is expected to be  $m \gg k$ . Therefore the complexity could be considered linear with respect to  $m$ .

### E. Graph cutting clustering

We also propose an algorithm which does not require reference channels. The algorithm is based on finding isolated subgraphs in a graph. In the graph, the nodes are formed by computed channels. Value of an edge is equal to the distance of the two corresponding channels. The graph is complete and is represented by a distance matrix for all channels. The basic structure of the algorithm is as follows:

- **Input:** Weighted complete graph where nodes represent channels and the weight of an edge between two nodes

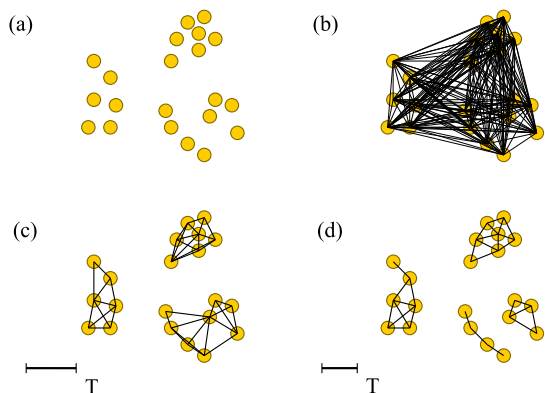


Fig. 4. Progress of the graph cutting clustering. (a) Set of nodes. (b) Graph in the initial state. (c), (d) Graph after cutting with the threshold value set to the length of line  $T$ .

(channels) is equal to the result of the distance function between this pair of channels. Threshold value  $t$ . Number of clusters  $k$ .

- **Output:** The set of clusters of channels.
- **Step1.** Remove edges with value higher than  $t$ .
- **Step2.** Perform depth-first search on a randomly selected node of the graph to find isolated clusters in the graph. When all neighboring nodes are processed, assign them into one cluster and repeat this action with another node randomly selected from the set of unprocessed nodes.
- **Step3.** If the number of clusters is smaller than  $k$ , lower  $t$  and go to **Step 1**.

A progress of the algorithm is demonstrated in Fig. 4. To be more intelligible, the graph in this figure is constructed so that the weight of an edge is equal to the length of the edge.

This method requires to compute the distances between all pairs of channels in  $S$  which requires  $\mathcal{O}(m^2)$ .

#### IV. RESULTS

The input data consisted of 36 sequences of haloalkane dehalogenase molecules. Each sequence contained 400 snapshots. All sequences were structurally similar, in fact they all were mutants of a wild type molecule where only a certain structural changes were applied. Chemists knew about four main channels in the wild type and expected these channels to appear also in the mutated structures. Therefore these four main channels are taken as reference channels for clustering methods for all analysed sequences.

The data provided by chemists also involved the results of manual processing of the sequences where for each snapshot and for each of the four main channels chemists have marked whether the channel is open or closed in the snapshot.

This information was obtained by observing the Connolly surface [12] of the molecule (pyMol<sup>1</sup> program implementation was used by chemists). If there was a hole in the surface which led into the active site, the channel was considered open. However, this procedure is quite inaccurate as the hole visible in the visualized surface does not predicate about channel bottleneck width. An example of this issue is demonstrated

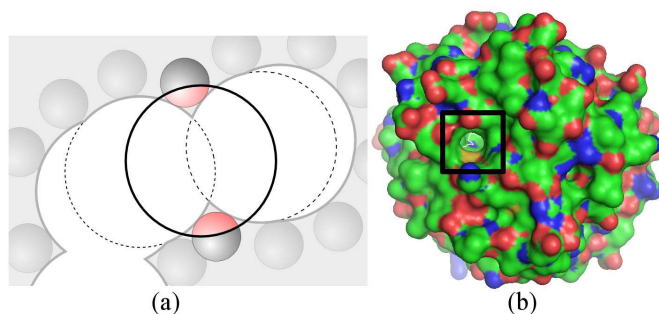


Fig. 5. Problems with the visualization of protein surface. (a) The hole would be visible although the bottleneck is lower. (b) Real visualization of protein surface with a hole.

in Fig. 5. The boundary of the Connolly surface is denoted by a thick grey line. According to the definition it is created as the union of all empty spheres of radius  $r$  which can be inserted into the molecule so that they do not cross any of the atoms. It is clear, that even though a hole is visible in the visualized surface (Fig 5 (a)), the actual bottleneck width is much smaller than  $r$ . The assumption "if a surface was generated with the value of  $r$ , then the small molecule of radius  $r$  should be able to pass through" is wrong. The example of visualized hole can be found in Fig. 5 (b).

It is expected that wrong annotations (additional channels) resulting from the visualization of Connolly surface did not occur frequently. However, it is not desired that our methods produce results which have 100% match with these annotated data. This would imply that the results contain the same error.

All sequences were processed by described methods and compared with manual annotations. Resulting clusters of channels were projected to progressions of certain channels in time and charts of channel parameters were generated (see Fig. 6).

As it can be seen in Table I and Table II, the algorithms based on QT and Graph cutting produced similar results. The results produced by the Simple method were in the most cases worse than these two methods. Regarding the K-means method, in case reference channels were selected appropriately, the results were approximately the same as the QT and Graph cutting. Nevertheless, K-means method is not suitable for automated processing. If the reference channels are selected randomly, the overall results of this method are even worse than the results of the Simple method (see Table II). As it can be seen in Fig. 6, if two cluster centers are located close to each other, the behavior of a real channel is represented by two disjoint clusters.

Similar issue could emerge when a cluster radius is set too small in QT based method. On the other hand, if the radius is too large, one cluster could describe the behavior of multiple real channels. Therefore the most suitable method for automated processing is the Graph cutting method which is able to adjust the size of the cluster according to the location of particular channels.

The theoretical complexities presented in the previous section were also verified on the real sequences. The results in Table III depict the computation time needed to process computed channels. The simple method is omitted as the

<sup>1</sup><http://pymol.sourceforge.net/>

TABLE I  
COMPARISON OF METHODS AGAINST MANUAL PROCESSING ON THE SELECTED SEQUENCES AND ON THE TWO MOST IMPORTANT CHANNELS

Molecule sequence	Channel 1				Channel 2			
	Simple	K-means	QT	Graph cutting	Simple	K-means	QT	Graph cutting
04_sdcl.cl	86.22%	69.60%	93.47%	92.46%	86.47%	78.89%	87.69%	87.19%
14_rdc1.cl	83.71%	89.45%	89.45%	89.45%	79.45%	79.15%	79.15%	79.15%
15_sdcl.cl	81.70%	83.42%	81.91%	83.42%	88.72%	88.69%	88.94%	88.69%
wt_rdc1.cl	74.94%	56.03%	76.63%	76.88%	88.47%	87.69%	83.92%	88.69%
wt_sdcl.cl	59.40%	63.07%	60.80%	62.31%	80.70%	83.42%	83.42%	83.41%

TABLE II  
OVERALL COMPARISON OF METHODS AGAINST MANUAL PROCESSING ON ALL 36 SEQUENCES AND IMPORTANT CHANNELS

Simple	K-means	QT	Graph cutting
80.95%	76.45%	82.23%	82.52%

TABLE III  
COMPARISON OF COMPUTATION TIME IN MILLISECONDS

Method/Sequence	04_sdcl.cl	14_rdc1.cl	15_sdcl.cl	wt_rdc1.cl	wt_sdcl.cl	average on all
K-means	49485	117750	101015	44171	142765	150505.26
QT	12734	12969	15031	13297	12468	14124.69
Graph cutting	180547	216766	303078	203172	193671	288208.89

time for this method is constant due to the fixed number of predefined channels.

It can be seen that average computation time (over 36 sequences) is the best in case of QT clustering and the worst in Graph cutting clustering. However, the graph cutting is the only method which is able to adjust cluster size dynamically.

## V. CONCLUSION

In this paper, various approaches to analysis of dynamic sequences of protein molecules were proposed. The comparison of methods indicates that the most promising results are achieved with the clustering methods which do not suffer from disadvantages caused by the need to choose reference channels.

Such methods are preferred by chemists as the methods do not need much input information. Presented methods are sufficiently reliable and offer a functionality which was not possible before – it is not only able to evaluate channels and their progression, but it can also determine channel parameters varying in the sequence. As an example of these parameters channel width and length could be mentioned; all these parameters can be for simplicity visualized in charts across the sequence. The methods are fast, robust and easy to implement.

In the future we plan to design more accurate distance functions which would include biochemical parameters of a channel. Resulting data are expected to be used in a new version of a visualization application CAVER 2.0<sup>2</sup> ([13], [2]) which would visualize channel geometry changing in time.

## ACKNOWLEDGMENT

This work was supported by The Ministry of Education of The Czech Republic, Contract No. LC06008 and by The Grant Agency of The Czech Republic, Contract No. 201/07/0927.

## REFERENCES

- [1] B. J. Alder and T. E. Wainwright, "Studies in molecular dynamics. i. general method," *The Journal of Chemical Physics*, vol. 31, no. 2, pp. 459–466, 1959. [Online]. Available: <http://link.aip.org/link/?JCP/31/459/1>
- [2] P. Medek, P. Benes, and J. Sochor, "Computation of tunnels in protein molecules using delaunay triangulation," *Journal of WSCG*, vol. 15, no. 1–3, pp. 107–114, 2007. [Online]. Available: [http://wscg.zcu.cz/WSCG2007/Papers\\_2007/journal/F53-full.pdf](http://wscg.zcu.cz/WSCG2007/Papers_2007/journal/F53-full.pdf)
- [3] M. Petrek, M. Otyepka, P. Banas, P. Kosinova, J. Koca, and J. Damborsky, "Caver: a new tool to explore routes from protein clefts, pockets and cavities," *BMC Bioinformatics*, vol. 7, pp. 316+, June 2006. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-7-316>
- [4] M. Petrek, P. Kosinová, J. Koča, and M. Otyepka, "Mole: A voronoi diagram-based explorer of molecular channels, pores, and tunnels," *Structure*, vol. 15, no. 11, pp. 1357–1363, November 2007. [Online]. Available: <http://dx.doi.org/10.1016/j.str.2007.10.007>
- [5] E. Yaffe, D. Fishelovitch, H. J. Wolfson, D. Halperin, and R. Nussinov, "Molaxis: Efficient and accurate identification of channels in macromolecules," *Proteins*, 2008. [Online]. Available: <http://www3.interscience.wiley.com/journal/117954527/abstract>
- [6] P. Medek, P. Benes, and J. Sochor, "Multicriteria tunnel computation," *Proceedings IASTED International Conference on Computer Graphics and Imaging (CGIM)*, 2008. [Online]. Available: <http://www.actapress.com/Abstract.aspx?paperId=32617>
- [7] P. Benes, P. Medek, and J. Sochor, "Computation of more channels in protein molecules," *Proceedings Conference Visual Computing For Biomedicine (VCBM)*, 2008.
- [8] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. [Online]. Available: <http://portal.acm.org/citation.cfm?id=42779>
- [9] H. Steinhaus, "Sur la division des corp materiels en parties," *Bull. Acad. Polon. Sci.*, vol. 1, pp. 801–804, 1956.
- [10] L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," *Genome Research*, vol. 9, no. 11, pp. 1106–1115, 1999. [Online]. Available: <http://genome.cshlp.org/content/9/11/1106.abstract>

<sup>2</sup><http://loschmidt.chemi.muni.cz/caver>



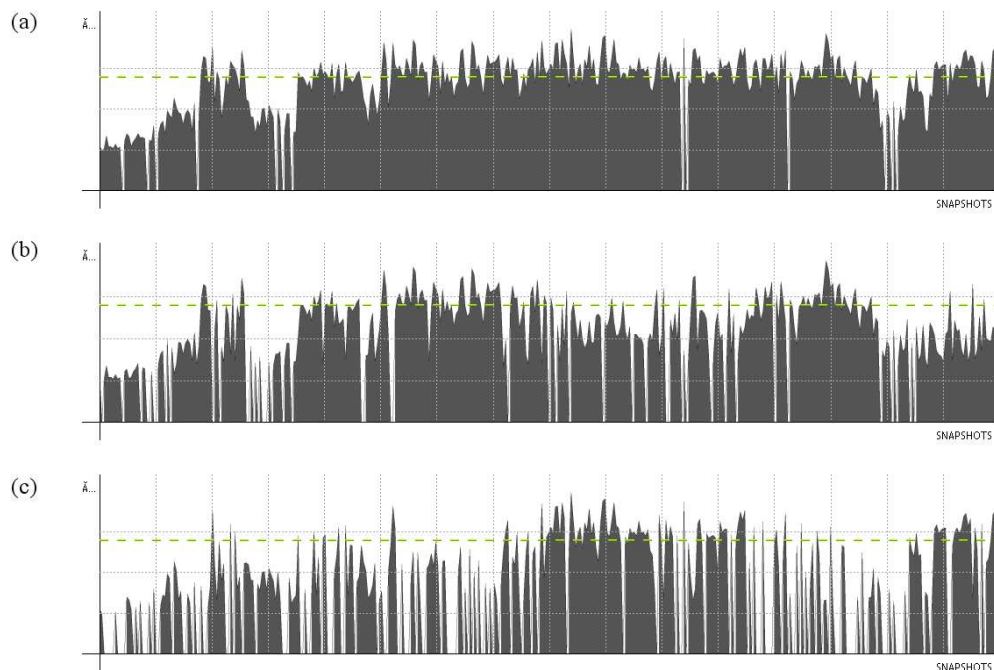


Fig. 6. The example of charts plotted for a channel in molecule 04\_sdcl.cl. (a) produced by Graph cutting method; (b), (c) by K-means based method. The dashed horizontal lines denote the threshold value below which the channel is considered to be closed. One channel in (a) is split into two clusters in (b), (c).

- [11] G. W. Flake, R. E. Tarjan, and K. Tsioutsoulis, "Graph clustering and minimum cut trees," *Internet Mathematics*, vol. 1, no. 4, pp. 385–408, 2004.
- [12] M. L. Connolly, "Analytical molecular surface calculation," *Journal of Applied Crystallography*, vol. 16, no. 5, pp. 548–558, Oct 1983. [Online]. Available: <http://dx.doi.org/10.1107/S0021889883010985>
- [13] B. Kozlíková, F. Andres, and J. Sochor, "Visualization of tunnels in protein molecules," in *AFRIGRAPH '07: Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. New York, NY, USA: ACM, 2007, pp. 111–118.