SPACE-PARTITION BASED IDENTIFICATION OF PROTEIN DOCKSITES

Ling Wei Lee and Andrzej Bargiela
School of Computer Science
The University of Nottingham Malaysia Campus
Jalan Broga, 43500 Semenyih, Selangor Darul Ehsan, Malaysia
Ling-Wei.Lee@nottingham.edu.my, Andrzej.Bargiela@nottingham.ac.uk

KEYWORDS
Protein Docksites / Information Granules, Space-Partitioning, Multi-Resolution Modelling

ABSTRACT

A new method for identifying protein docksites is presented here. We introduce a space-partition based technique for evaluating the binding areas of a protein and present the final output in a 3-dimensional environment. The experimental space is tessellated through bi-partitions and the occupancy of each atom within the tessellations is approximated using evaluated constraints. A series of preprocessing steps are taken to ensure that the subject of experiment is independent of orientations and thus gives reproducible results. The final modelling of the protein shows the docksites granulated using a compactness criterion. Controls of the compactness limit the clusters of binding sites to be displayed on the 3D projections with more prominent areas linked to a higher compactness of 3D structures.

INTRODUCTION

With the increasing number of proteins being added into the Protein Data Bank (PDB) there is a need for fast and efficient algorithms to process these files and derive important information for further analysis. Each protein – formed from a series of amino acids – may fold to reveal a completely different structured protein although the base chain may be similar. The binding sites of a protein are the areas that contribute to the protein's functions. Hence it is important to identify these docksites and conduct analysis upon them to gain a better understanding of the protein.

There are many methods available today for the identification of protein docksites. For example, POCKET – developed by Levitt and Banaszak [Levitt, Banaszak, 1992], is able to identify the docksites of a protein through the use of cubic grids without the need for prior knowledge of proteins. However, disorientations in the protein will lead to a reduced efficiency of the algorithm. LIGSITE [Hendlich et al., 1997] remedied the weaknesses of POCKET by using rigorous scans. SURFNET, created by Laskowski [Laskowski, 1995], visualises molecular surfaces and the gaps in between which indicate potential cavities. The program is interactive and is compatible with other graphics packages. Although each algorithm has its own strong points, none of the current solutions is fully satisfactory. Research is still continuously being conducted to find better solutions [Ritchie, 2008].

Here we present a new way of looking at the problem of identifying protein docksites. The notion of space-partitioning is introduced here. Our technique attempts to recursively 'slice' a fixed experimental space into smaller and smaller subunits until the minimum threshold is achieved. The protein is then cast into the experimental space and a checking is made to verify if a particular voxel is occupied by any of the protein's atoms. We do not use exact measurements but rather a way of generalising the data, which greatly increases the processing speed but also maintains the vital information required.

The final modelling of the predicted docksites of the protein is carried out on a multi-resolution basis. Information granules [Bargiela at al, 2002, 2004, 2005, 2009] contribute to and determine how the modelling algorithm runs. By using the same set of original data, a series of processing is carried out to extract and compose a range of data which can be studied on different levels. The information granules gathered may be clustered together such that a more generalised representation is obtained. Alternatively, the granules may also be broken down for more refined data analysis. Multi-resolution modelling is a visualisation of these granules and hence, our modelling program displays the detected docksites of a protein in terms of coarse and detailed representations.

BACKGROUND

It can be seen from previous research that cubic grids offer an efficient means for protein analysis. Such grids also form the underlying structures for our space-partitioning algorithm. In space-partitioning, a given experimental space is tessellated into subunits of smaller spaces, whereby the size of the subunits are dependent on the type of partitioning algorithm employed. For this research, the bi-partitioning method is used. This technique is fast and executes in $\log_2(n)$ time. For example, a test space of size 80 will be subdivided into units of size 40, followed by units of

doi:10.7148/2009-0848-0854

size 20 and so on. The algorithm stops when the minimum threshold has been reached.

Bi-partitioning may seem similar to the concept of fractals but it should be noted that there are noticeable differences in both. The partitioning of a space has constraints applied to it and the end result should be somewhat balanced. It is a recursive process similar to the generation of fractals but unlike the latter it generates regular shapes, which in this case are hyperboxes. Therefore, one can say that space-partitioning has a fixed scale applied to ensure that the subdivisions are always consistent.

We can say that information displays the attributes of fractals – one can granulate the data and analyse the output on different stages in terms of fuzzy modelling [Pedrycz, Bargiela, 2002, 2003], [Bargiela et al, 2002, 2003, 2004, 2005, 2009]. Wang et al [Wang et al, 1990] did a study and identified the relationship between fractal dimensions and the tertiary structure of proteins, proving that such an approach is applicable within this domain. Hu and Peng [Hu, Peng, 2005] developed a measurement termed Volume Fractal Dimensionlity (VFD) which may be employed in the study of protein evolution and prediction.

Our approach to identify the docksites on a protein attempts to process an original PDB file – obtained from RCSB Protein Data Bank (http://www.rcsb.org) – in several stages. A PDB file contains all vital information related to the protein, including the spatial coordinates of each atom, the atom type, the occupancy of each atom etc. However only 4 items of data are required in the algorithm – the X, Y, Z coordinates as well as the Van der Waals radius for each of the atom. The VDW radius gives an estimation of the size of the entire atom, which corresponds well to our objective of approximated analysis and modelling. By using the above data, we first preprocess the protein so that it is invariant to translation and rotation. It should be noted that a change in the orientation of the protein has posed problems for researchers over the years. We hereby introduce a novel method of aligning the protein to overcome all disorientations. Further details are elaborated in the Preprocessing Section.

The new set of preprocessed data is then sent for the generation of 2-dimensional templates whereby the space-partitioning algorithm is duly applied. We choose to partition in 2D instead of 3D as the additional dimension will increase the computation and memory usage significantly. To overcome the problem of the missing dimension, we selected the dimension not portrayed in 2D for unit slicing. By unit slicing we mean that the dimension concerned will be split based on the smallest unit used. This is similar to z-buffer algorithm in 3D graphics. A list of images are then generated using the other two dimensions. The number of images is related to the count of segments obtained from the sliced dimension. More elaboration is given in the Space-Partitioning Section.

The final stage of the program involves projecting the processed data into the 3-dimensional space. The Java 3D package is harnessed here and the 2 smallest partition levels are included in the program for visualising the concept of multi-resolution modelling. Each partition level is given a set of controls whereby the occurrences of higher or lower weighted docksites can be manipulated by changing the values. A more detailed description is presented in the section on modelling.

PREPROCESSING

The objective of the preprocessing stage is to ensure that the chosen protein is aligned with the axes in the experimental space. This method is an attempt to overcome the problem of disorientations encountered in previous research works. The algorithm has been tested for the same protein rotated in all X, Y and Z planes and has proved to be effective in realigning the protein consistently.

The first step in the preprocessing stage is the calculation of the maximum-valued cross sectional cord between all atoms in the protein. The calculation is carried out in 3D space whereby the equation is given by Equation (1).

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \qquad (1)$$

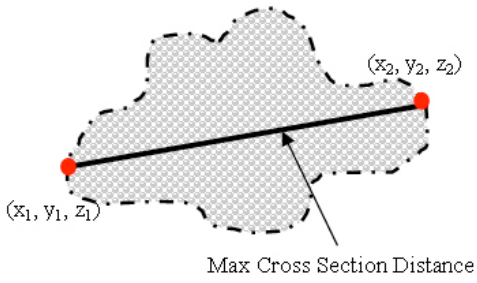

Figure 1: Illustration of the Maximum-Valued Cross Sectional Cord

Once this value has been obtained, the algorithm proceeds to align the cord such that it is parallel with the X axis in 3D space. The maximum cord always gives the same value regardless of how the protein was initially positioned. Affine transformations are used here and the matrices are given in the equations below.

doi:10.7148/2009-0848-0854

$$T = \begin{bmatrix} 1 & 0 & 0 & dx_i \\ 0 & 1 & 0 & dy_i \\ 0 & 0 & 1 & dz_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -dx_i \\ 0 & 1 & 0 & -dy_i \\ 0 & 0 & 1 & -dz_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

$$Rz = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

$$Ry = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

In affine transformation all the spatial coordinates of the object are first translated with relation to the origin. Subsequently rotations around the z and y-axis are carried out. The final step involves moving the object back into the original space. Equation (2) gives matrices for translation to the origin and back to experimental space. In the program, one of the points of the maximum cord is selected as the point of reference for translation to origin. All atoms are translated with relation to this point. Following that the protein is first rotated about the Z-axis followed by a rotation about the Y-axis. Equations (3) and (4) are the corresponding transformations. The angles for each of the matrices are illustrated in Figure 2.

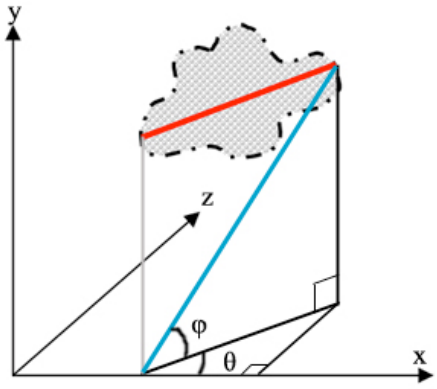

Figure 2: Angles of Rotation Around Y-axis and Z-axis

Applying the above rotations will ensure that the maximum cord gets aligned with the X axis, resulting in a value of 0 for both Y and Z axes for all points on the cord. However this does not solve the protein missalignment completely. The protein may need to be rotated around the X axis to a position that ensures reproducible positioning of the protein.

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5)$$

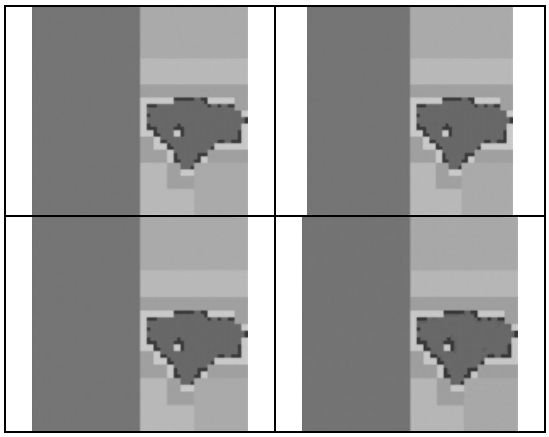

Figure 3: Results of PDB300d after Preprocessing. Originals (clockwise from top left) are rotated 60º about Z, 120º about Z, 10º about X, 20º about Y, 30º about Z and 30º about Y, Z.

Our approach is to identify an atom that is furthest away from the cord and to align the orthogonal projection from this point onto the cord with the Y axis. The matrix representation for rotation about X axis is given in Equation (5). Therefore, the complete affine transformation of the protein involves translation to the origin, carrying out the rotations about all the axes, and translating the protein back into the experimental space by a fixed amount. This preprocessing has been tried on different inputs and the results show consistency regardless of the initial positioning of the protein. Figure 3 shows a few of the output generated in 2D.

SPACE-PARTITIONING

The preprocessed datasets are sent for the generation of 2-dimensional templates in the form of images. As mentioned in Background, one of the dimensions is selected for unit slicing, which in this case is the Z dimension. Take for example the protein that has atoms occupying units 300 to 900 in Z dimension and the smallest unit space used is 10 units. Unit slicing will proceed to generate an analysis image for every 10-unit interval within the occupied range. The output images are then generated using the other 2 dimensions following space-partitioning. The images in Figure 4 show output of different layers after bi-partitioning has been applied.

Notice that the images contain different shades of boxes. These boxes are generated from the bi-partitioning algorithm. The darker gray shades on the left of the images are obtained from the first run of bi-

doi:10.7148/2009-0848-0854

partitioning. As no part of the protein occupy these spaces, hence they are shaded.

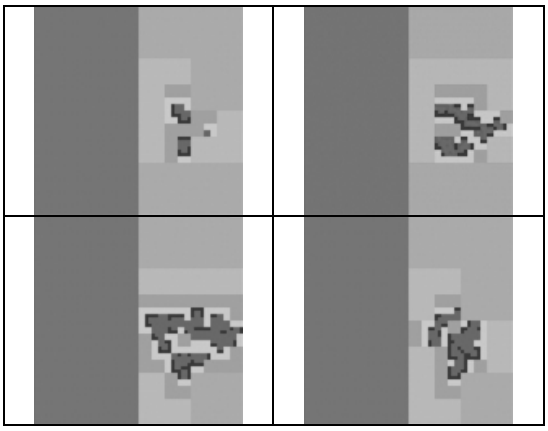

Figure 4: Slicing of Protein PDB300d with Partitioning Applied. The slice number is (clockwise from top left) 620, 650, 760 and 680 respectively.

The bi-partitioning algorithm starts from the whole of the experimental space and continues executing until the smallest unit is reached. The smallest unit is chosen based on an analysis of the sizes of atoms in proteins. There is no merit in subdividing the units beyond this size as this would increase the processing and decrease the efficiency of the algorithm without adding any significant information about the topology of the binding sites. The objective is to keep the voxels sizes just small enough such that information on the protein is not lost. Generating the voxels on a subatomic basis does not necessarily increase the details needed. Such space-partitioning method looks at the predictions of the docksites on a multi-resolution level. The first few runs executed at coarse resolution generally do not tell the user where the docksites are, but they do limit the experiment space to where the protein resides. Further runs of the algorithm with increasing resolution begin to indicate to the user the potential docksites and the final run with the highest resolution gives a clear definition of the binding areas by lining the edges of the docksites with boxes of the smallest unit.

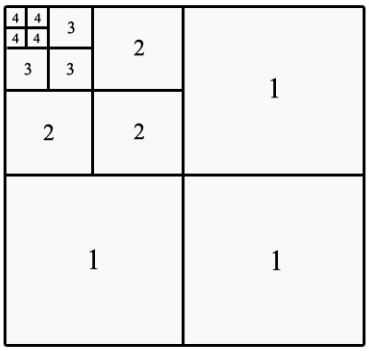

Figure 5: How The Bi-Partitioning Algorithm Works

Figure 5 shows how the bi-partitioning algorithm works. The numbers in the grids correspond to the runs, i.e. 1 is Run 1, 2 for Run 2 and so on.

There are some techniques of image processing employed here to increase the speed of the space-partitioning algorithm. First of all an imprinting of the protein is created on a temporary image in a solid colour. Then the pixels of a partitioned space from the temporary image are checked to see if it is being occupied by any atom. If it is empty, then the partition is given a colour or shade on another new image.

```
PartitionSize = The Test Space Dimension

FOR 0 to Test Space Dimension
        FOR each PartitionSize

                Check if partition contains atom
                    IF YES
                       Leave partition unshaded
                    ELSE
                      Shade partition based on
                      current partition color

                END OF CHECK
        END FOR

        Decrease PartitionSize by half

END FOR
```

Figure 6: Pseudocode for the Bi-Partitioning Algorithm

If however, an atom partially occupies a partition, then a constraint is applied to decide if the protein should be given the partition or if the partial occupancy should be excluded. This is a method of estimation which maintains the necessary information while discarding insignificant ones.

The checking for the images is carried out for each and every slice and the unit information for two of the smallest partitions are exported to files for use in modelling. By using comparison between images, the amount of time used is decreased as the process involves only simple comparisons between pixels. The bi-partitioning algorithm is given in pseudocode in Figure 6.

MODELLING THE IDENTIFIED PROTEIN DOCKSITES

After the space-partitioning algorithm has been applied to every image generated, there is a list of datasets available for visualisation in the 3-dimensional space. Here we harness the power of the Java3D package as our modelling and visualisation tool. The program output has a universe showing groups of unit

doi:10.7148/2009-0848-0854

voxels which fill the potential docksites and a control panel for users to update the compactness factor. This factor characterizes individual voxels and enables control over granulation of voxels into docksites. The images below show comparisons between the output from different levels of compactness factors at the same sites. Higher compactness factor implies display of only the voxels that are highly connected between themselves and lower compactness factor permits display of less well connected voxels.

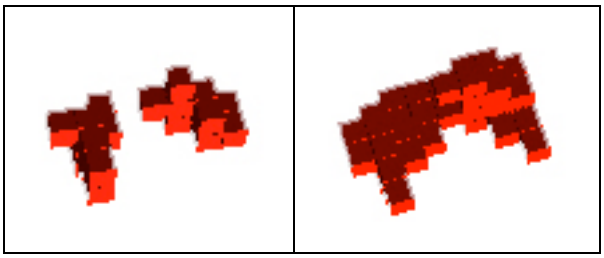

Figure 7: Voxels of Unit 10, with Compactness Factor of 60 and 55 respectively

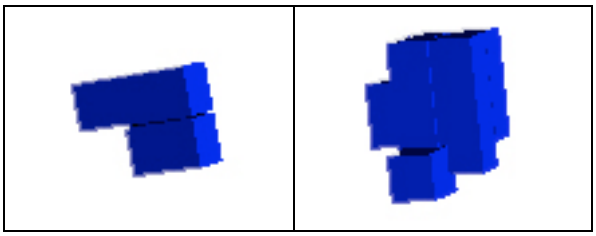

Figure 8: Voxels of Unit 20, with Compactness Factor of 60 and 50 respectively

A method for the calculation of the connectivity of the voxels has been devised so as to enable the user to control the display of the docksites according to their compactness factor. By using the information from the datasets generated from space-partitioning, a list is constructed to store unique entries of the points of cubes existing for each partition group. At the same time, the number of times each point is used is also stored into memory. A standalone cube has each of its corners given a value of 0 as none of the points are interconnected with other cubes. The sum of the points determine the total weight for that cube, which in this case is 0, making the cube a least significant voxel. This weight is interpreted as a compactness factor of the docksite. A higher total weight indicates that the cube is highly interconnected with other cubes, suggesting that it contributes more to the group of voxels which collectively form the binding agent for a potential docksite. By allowing the user to select which compactness factor should be used for visualizing docksites, the program becomes an analysis tool for revealing potential crevices in the protein. The images below provide a clearer idea of this approach.

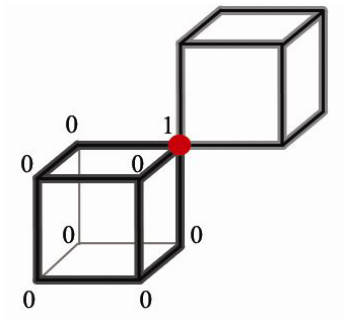

Figure 9: Illustration of A Single Corner On A Voxel Connected to Another Voxel

2

Figure 10: Illustration of A Moderately Connected Voxel with Total Weight 10

Figure 9 shows a single point connectivity between the main subject and a neighbouring voxel. The total weight for the voxel is 1. In Figure 10, the main voxel has 4 neighbouring cubes. Three of its corners are each shared with two other cubes and four corners are each shared with one cube; thus making the total weight for this cube equal 10. Such sharing does not violate the uniqueness of the points, but rather contributes to the total weight of the voxel concerned and defines the level of significance of the voxel within a cluster. The higher the total weight, the heavier the role of the voxel in a particular docksite group. The maximum sum any corner of a voxel can achieve is 7, which is a full neighbourhood connectivy, making the maximum total weight 56.

In our visualisation program, we have added controls for users to update the compactness factor of the docksites. Users are given the opportunity to increase or decrease the weight while observing how differences in the values affect the voxels in the 3-dimensional universe. As the weight value decreases, voxels with lesser neighbourhood cubes are gradually introduced. The higher the factor value, the more compact and smaller a docksite binding cluster becomes. However it should be noted that there is an optimum threshold for the clusters. Once beyond this threshold, the docksites

doi:10.7148/2009-0848-0854

information may tend to overcluster, leading to a less than accurate output. Likewise an overly high compactness value may also lead to underclustering. Determination of this optimum value is a study in itself and requires further analysis.

As of current, it is sufficient to show that given the flexibility to control the weights of the clusters, such a visualisation program is able to aid researchers and scientists in the investigation of potential protein docksites and in the prediction of possible docking elements through the analysis of the models of the displayed clusters.

DISCUSSION

There is much merit in the use of cubic units or voxels in identifying potential protein docksites. This could be due to its simplicity and the reduced processing time as it does not require complex mathematics. The approach proposed here is considerably fast, with the space-partitioning stage completing in a matter of minutes on a normal workstation. Increasing the processing power of the machine will reduce the time involved in partitioning the spaces. The modelling of the voxels in 3-dimension is based on the results generated and computation for determining the voxels to be displayed is fast enough such that a change in the compactness factor updates the 3D universe in seconds.

An advantage of using cubic units in studying the surface of the protein is the ability to shortlist the corresponding units for further analysis. Methods like convex hull assignments [Stout et al, 2008] or Bezier splines do provide a mapping of the protein structure and surface to some extent, but some crevices in the protein may tend to get ignored or generalised unless richer input is supplied. Compared to these approaches, the use of cubic units and space-partitioning is able to cater for a more refined level of analysis, allowing for the extraction of particular sections. Cubic units also provide a clearer estimation of the shapes of possible binding agents.

In our approach we have overcome the problem of misalignment of the protein in the 3D space. The method is straightforward and fast, and results have shown to be promising in terms of reproducible alignment of the protein. The use of multi-resolution modelling here provides a new insight into the analysis of data from a single protein, especially in locating the docksites of a protein. Through simple controls one can choose to show or hide a particular set of information granules, allowing for different levels of studies. Coarser granules give general representations while refined ones display more information. With different resolutions one can create programs that work on different datasets – often a coarser granulation will lead to faster processing. Such approach works on the approximation plane while retaining the right amount of information for the user.

Granular modelling is emerging as a new paradigm for the classification and processing of information. By introducing this concept into the field of bioinformatics new methods can be produced whereby the use of fuzzy processing may lead to a comparable if not better achievement of results. In our research it is important to correctly identify the docksites on a protein and by utlilising an outside-in approach the shapes of potential binding agents can be estimated. Granular modelling generates different levels of granularites for these predicted binding agents and the visualisation program has included 2 sets of controls for viewing the agents in different levels. For each level of granularity ranging from coarse to fine, we can obtain varying degrees of information – a course representation provides us with clues on the locations of the docksites and as the representation becomes more and more refined the shape of the binding agent becomes more well described.

CONCLUSION

The space-partitioning method provides a new and novel approach of identifying docksites in a protein. By using a newly devised algorithm, the protein file is preprocessed such that it is invariant to disorientations. The bi-partitioning algorithm is then executed on the preprocessed data, resulting in a new list of images and voxel information which is then used for the modelling of the located docksites. Controls for the compactness factor are given in the visualisation program. By updating and changing these values, users may observe an immediate update of identified docksites being projected or removed from the 3-dimensional universe. These docksites are represented on 2 levels of resolution, providing information on a rough and refined basis. The use of this multi-resolution approach not only extracts the necessary information from the data through estimation, but it also preserves the vital information needed. There are significant advantages of using a cubic grid system over more conventional methods which employ surface curves in the analysis of proteins. Cubic grids are able to locate more precisely the potential docksites and possibly provide a predicted solution to the docksite.

REFERENCES

Bargiela A, W. Pedrycz, Granular Mappings, IEEE Transactions on Systems, Man, and Cybernetics - Part A, 35, 2 ,292-297, March 2005, doi: 10.1109/TSMCA.2005.843381

Bargiela A, W Pedrycz, From numbers to information granules: A study in unsupervised learning and feature analysis, SERIES IN MACHINE PERCEPTION AND ARTIFICIAL INTELLIGENCE 47, 75-112, 2002, doi: 10.1142/9789812778147_0004

doi:10.7148/2009-0848-0854

Bargiela A, W Pedrycz, Granular computing: an introduction, Springer, 2003,

Bargiela A, W. Pedrycz, A model of granular data: a design problem with the Tchebyschev FCM, Soft Computing, 9(3):155-163, March 2005, doi: 10.1007/s00500-003-0339-2

Bargiela A, W. Pedrycz, K. Hirota, Granular Prototyping in Fuzzy Clustering, IEEE Transactions on Fuzzy Systems, 12(5):697-709, 2004, doi: 10.1109/TFUZZ.2004.834808

Bargiela A, W. Pedrycz, Human-Centric Information Processing Through Granular Modelling, Studies in Computational Intelligence 182, Springer Berlin Heidelberg, 2009, doi: 10.1007/978-3-540-92916-1

Hendlich, M.; F. Rippmann; G. Barnickel; 1997. "LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins" *Journal of Molecular Graphics and Modelling 15* 359-363.

Hu, M.; Q. Peng; "Volume Fractal Dimensionality: A Useful Parameter for Measuring the Complexity of 3D Protein Spatial Structures" In *Proceedings of the 2005 ACM Symposium on Applied Computing* (Santa Fe, New Mexico, Mar 13-17). 172-176.

Laskowski, R. A. 1995. "SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions" *Journal of Molecular Graphics* Vol. 13, Issue 5, 323-330.

Levitt, D.G.; L.J. Banaszak; 1992. "POCKET: a computer graphics method for identifying and displaying protein cavities and their surrounding amino acids" *Journal of Molecular Graphics* Vol. 10, Issue 4, 229-234.

Pedrycz, W.; A. Bargiela; 2003. "Fuzzy fractal dimensions and fuzzy modeling" *Information Sciences: an International Journal* Vol. 153, Issue 1, 199-216, doi: 10.1016/S0020-0255(03)00075-6

Pedrycz W, A Bargiela, Granular clustering: a granular signature of data, Part B: Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics, 32(2):212-224, April 2002, doi: 10.1109/3477.990878

Ritchie, D.W. 2008. "Recent progress and future directions in protein-protein docking" *Current Protein and Peptide Science* Vol. 9, No. 1, 1-15.

Stout, M.; J. Bacardit; J.D. Hirst; N. Krasnogor; 2008. "Prediction of Recursive Convex Hull Class Assignments for Protein Residues" *Bioinformatics* Vol. 24, Issue 7, 916-923.

Wang, C.X.; Y.Y. Shi; F.H. Huang; 1990. "Fractal study of tertiary structure of proteins" *Physical Review A* Vol. 41, Num. 12, 7043-7048.

AUTHOR BIOGRAPHIES

LING WEI LEE was born in Kuala Lumpur and studied at the University of Nottingham Malaysia Campus where she took up Computer Science and obtained her honours degree in 2007. She worked for about a year as an analyst programmer with a local company before deciding to pursue her postgraduate studies. Her current research focuses on the applications of granular computing methods in bioinformatics in the area of proteins. She can be reached at Ling-Wei.Lee@nottingham.edu.my.

ANDRZEJ BARGIELA is Professor and Director of Computer Science at the University of Nottingham, Malaysia Campus. He is member of the Automated Scheduling and Planning research group in the School of Computer Science at the University of Nottingham. Since 1978 he has pursued research focused on processing of uncertainty in the context of modelling and simulation of various physical and engineering systems. His current research falls under the general heading of Computational Intelligence and involve mathematical modelling, information abstraction, parallel computing, artificial intelligence, fuzzy sets and neurocomputing.

doi:10.7148/2009-0848-0854