International Journal of Medical, Medicine and Health Sciences
ISSN: 2517-9969
Vol:1, No:8, 2007

Protein Graph Partitioning by Mutually Maximization of cycle-distributions

Frank Emmert Streib

***Abstract*— The classification of the protein structure is commonly not performed for the whole protein but for structural domains, i.e., compact functional units preserved during evolution. Hence, a first step to a protein structure classification is the separation of the protein into its domains. We approach the problem of protein domain identification by proposing a novel graph theoretical algorithm. We represent the protein structure as an undirected, unweighted and unlabeled graph which nodes correspond the secondary structure elements of the protein. This graph is call the protein graph. The domains are then identified as partitions of the graph corresponding to vertices sets obtained by the maximization of an objective function, which mutually maximizes the cycle distributions found in the partitions of the graph. Our algorithm does not utilize any other kind of information besides the cycle-distribution to find the partitions. If a partition is found, the algorithm is iteratively applied to each of the resulting subgraphs. As stop criterion, we calculate numerically a significance level which indicates the stability of the predicted partition against a random rewiring of the protein graph. Hence, our algorithm terminates automatically its iterative application. We present results for one and two domain proteins and compare our results with the manually assigned domains by the SCOP database and differences are discussed.**

***Keywords*— Graph partitioning, unweighted graph, protein domains.**

I. Introduction

THE investigation of the structural organization of proteins is important for understanding the mechanisms of protein folding and the evolution of the proteins. Direct determination of protein structures [10], [16], [2] as well as comparative sequence analysis [8] indicate that proteins have a modular structure, i.e., that a polypeptide chain may consist of several sequence elements that fold independently and may be inherited as discrete sequence fragments, which recombine to produce novel sequence and spatial architectures. This level of protein organization is called domain [17], [13], [4]. A formal definition of a domain is an interesting and still outstanding problem. In general, the notion of a structural domain of a protein is associated with its compactness and thermodynamical stability if excised, see, e.g., [15] for a more detailed discussion and references. Practically, a good definition of a protein domain is a prerequisite for any functional or evolutionary analysis of proteins and proteomes.

In this article, we present a novel algorithm for the automatic identification of structural domains of proteis which is based on a graph theoretical approach. In the next section we introduce this algorithm mathematically. In section III we present results for one and two domain proteins and compare our results with the manually assigned domains from the SCOP database [9]. We finish this article with a summary and some concluding remarks in section IV.

Frank Emmert-Streib is with the Stowers Institute for Medical Research, 1000 E. 50th Street, Kansas City, MO 64110, USA, e-mail: fes@stowers-institute.org.

II. The Model

The basic idea behind our algorithm for the identification of protein domains consists in two step. The first is, to represent a protein as graph. The second, to partition this graph and identify the partitions as domains. In the following subsections, we will describe both steps in detail.

A. Representation of Proteins

If one wants to identify the domains of a protein algorithmically, one has to represent the protein in a way which is accessible to mathematical methods. Hence, it is clear that this representation will inevitably disregard some known properties of proteins, because up to now there is no mathematical model for proteins available describing all of their known properties. We use as course-grained level of description the secondary structure elements of a protein. More precisely, we distinguish between three different types of secondary structure elements - helix, strand and loop. Based on the secondary structure elements of a protein we transform the information available about a protein in a PDB file from Protein Data Bank [3] in a graph by the following algorithm.

Algorithm 1: Representation of a protein as a graph:

1) Determine the secondary structure elements of a protein by using the information from a PDB file of a protein and enumerate them in a consecutive order. We differentiate between three types of secondary structure elements: helix, strand and loop.
2) Each secondary structure element represents one node in the protein graph.
3) Two nodes m and n in the protein graph are connected by an edge $e(m,n)=1$, if there exist two C_α-atoms, one from secondary structure element m and one from secondary structure element n whose spacial distance is below a threshold Θ

$$e(m,n)=\begin{cases} 1 & : |C_\alpha^m - C_\alpha^n| \leq \Theta \\ 0 & : |C_\alpha^m - C_\alpha^n| > \Theta \end{cases} \quad (1)$$

Additionally, we connect consecutive secondary structure elements along the backbone

$$e(m+1,m)=e(m,m+1)=1 \quad \forall\, m \in \{1,\ldots,N-1\} \quad (2)$$

All other entries in the adjacency matrix $e(,)$ of the protein graph remain zero[1].

This results in an undirected, unweighted and unlabeled graph for every protein. That means, we neglect the labels of the nodes representing a helix, a strand or a loop and we do not consider weights of edges resulting from multiple pairs of C_α-atoms whose reciprocal spacial distance is below the threshold Θ. A protein graph has zero entries on the main diagonal and ones on its first upper and lower diagonal representing the connections from the backbone. The remaining sites are sparsely occupied by ones representing, e.g., hydrogen bonds. We do not check if the connections determined by Eq. 3 are actually hydrogen bonds because we want to see, whether this course-grained approach is already appropriate to carry enough information for the domain identification. To choose the secondary structure elements of a protein as course-grained level is in contrast to other contributions dealing with the domain identification. Normally, the C_α's of the residues [14] or even the atoms of the backbone [18], [5] are selected to extract information for their methods. Our choice has the advantage to reduce the almost overwhelming complexity of information available for each protein whose structure has been chrystalised, provided by a PDB file [3], rigorously.

This leads to the following definition.

Definition 1: We call an unweighted, undirected and unlabeled graph obtained by the algorithm 1 a protein graph and denote it by $G_{\mathrm{III.}}$.

Based on this representation of a protein as a graph we will introduce now a method which partitions a protein graph. The obtained partitions will then be defined as the domains of a protein.

B. Partitioning of Protein Graphs

It is believed since a long time that the domains of a protein are in some form *compact* [12]. There are several suggestions to characterize the compactness of a domain in a more precise way. For example there are hypothesis that the domain should stay folded if the protein is cut into its domains or that the number of contacts between domains should be less than the number of internal domain contacts [13], [11]. If one takes a look to protein structures one gets immediately the feeling that the compactness of domains should not be interpreted in a strict mathematical sense as, e.g., the compactness of a chemical crystal structures like NaCl but in a less restrictive way. To make our point clear, we depicted in Fig. 1 three different domains. The line represents in all figures the backbone. Apparently, the intuitive notion of compactness varied between all figures significantly. However, a common property shared by all schematic domains is that the backbone has to 'fold back'. Each domain starts with a backbone piece for which this does not hold. It is clear, that such a piece can not be a domain at all. The degree to which the back-folding occurs differs for all three figures. This makes it from a mathematical point of view difficult to find a common characteristics. Interestingly, from a graph theoretical point of view one simple entity which can always distinguish between a back-folded and a non-backfolded backbone is a cycle. A cycle is a closed path which returns to its starting point and, hence, can be seen to represent compactness in a wider sense. This motivated us to derive an algorithm which mutually maximizes the cycle-distributions found in two protein subgraphs resulting from a single cut position of the backbone. In the following we present this algorithm in detail.

[1]The Cartesian coordinates are given in a PDB file of a protein

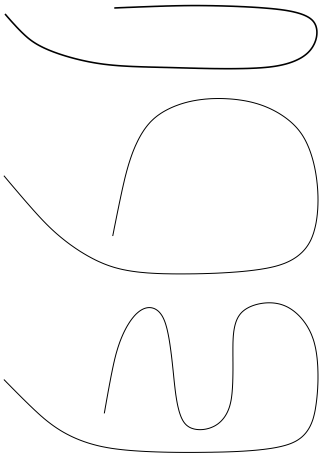

Fig. 1. Schematic domains of a protein. The full line corresponds to the backbone.

Algorithm 2: Partitioning of a protein graph G_{III} with N nodes.

1) Calculate the cycle set $\mathcal{CS}$ consisting of all cycles found in the graph G_{III} up to a length L.
2) Determine the cycle histograms $CH_L(i)$ and $CH_R(i)$ for $i \in \{1,\ldots,N-1\}$ by dividing the cycle set CS in three non-intersecting sets $\mathcal{CS}_L, \mathcal{CS}_R$ and $\mathcal{CS}_{\mathrm{LR}}$ defined by

$$\mathcal{CS}_L(i) = \{c \in \mathcal{CS} \mid c_j \leq i \,, \forall\, j \in |c|\} \quad (3)$$
$$\mathcal{CS}_R(i) = \{c \in \mathcal{CS} \mid c_j > i \,, \forall\, j \in |c|\} \quad (4)$$
$$\mathcal{CS}_{\mathrm{LR}}(i) = \mathcal{CS} \setminus \{\mathcal{CS}_R(i) \cup \mathcal{CS}_L(i)\} \quad (5)$$

We call i the boundary index of part L. The cycle histograms are now defined for the i-th index by

$$CH_L(i,j) = |\{v \in \mathcal{CS}_L(i) \mid |v| = j\}| \quad (6)$$
$$CH_R(i,j) = |\{v \in \mathcal{CS}_R(i) \mid |v| = j\}| \quad (7)$$

3) Normalize the cycle histograms along the cycle length index

$$\overline{CH}_L(i,j) = \frac{CH_L(i,j)}{\sum_{i'} CH_L(i',j)} \quad (8)$$
$$\overline{CH}_R(i,j) = \frac{CH_R(i,j)}{\sum_{i'} CH_R(i',j)} \quad (9)$$

4) Determine an objective function $E_{obj}(i)$ for $i \in \{1,\ldots,N-1\}$ by:

$$E_{obj}(i) = \sum_{j}^{L} \overline{CH}_L(i,j)\overline{CH}_R(i,j) \quad (10)$$

5) Determine the maximum of the objective Function

$$i_{cut} = \underset{i'}{\operatorname{argmax}}\, E_{obj}(i') \tag{11}$$

6) Determine a significance level of the cut and accept it, if

$$\alpha E_{obj}(i_{\rm cut}) > \overline{E}^{r}_{obj}(i_{\rm cut}) \tag{12}$$

with $\alpha \in [0,1]$.

As underlying visualization one should imagine a graph consisting only of the "backbone" connections given by Eq. 3 displayed as straight, horizontal line connecting the N nodes. The boundary index i determines uniquely two disjunct vertices sets $V_L(i) = \{1, 2, \ldots, i\}$ and $V_R(i) = \{i+1, i+2, \ldots, N\}$ separating the nodes on the "backbone" in a left L and right R part. The boundary index i can be seen as sliding cut-position along the "backbone" connections which results in V_L and V_R. The histograms of the cycle distributions are then given, e.g., for the left part and boundary index i, as the number of cycles of length j from $\mathcal{CS}$ which contain only vertices from $V_L(i)$. This is denoted by $CH_L(i,j)$. Our objective function E_{obj} determines the dot product between the normalized cycle histograms of the left and right part and measures by this their mutual overlap. We use the normalized cycle histograms along the cycle length index because the absolute number of cycles found is not of interest at all but only the relative number compared to other potential cut positions. The normalization transforms the absolute values into relative weights between different cut positions.

The crucial point of our procedure is to decide, if the suggested cut position i_{cut} is accepted or rejected. More precisely, we need to define a significance level of the suggested cut position. Intuitively, this could be done by calculating the objective function of a randomized protein graph. The cut position will then be accepted, if the value of the objective function of the randomized protein graph is significant lower than the value for the unperturbed protein graph. To apply this approach we have to define a randomized protein graph and a significance level mathematically. We suggest to define a randomized protein graph by randomly alternating $\Theta_r N$ entries of a protein graph of non diagonal and non first off-diagonal entries. This ensures, that the resulting graph has still its backbone connections and no self-connections[2]. By this we obtain a value $E^r_{obj}(i_{\rm cut})$ for this graph of the randomized objective function. The predicted cut position can be viewed as statistically significant, if it is stable against the averaged randomized objective functions

$$\overline{E}^{r}_{obj}(i_{\rm cut}) = \frac{1}{N_r} \sum E^r_{obj}(i_{\rm cut}) \tag{13}$$

of an ensemble of size N_r of randomized protein graphs. The parameter α in Eq. 12 is introduced as weight because our suggestion to define a randomized protein graph is plausible but certainly different to a 'natural' or 'real' randomized protein graph which is actually unknown due to the lack of our understanding of the organization of protein tertiary structures.

[2]The alterations of the matrix entries has to be done symmetrically.

If the cut condition in Eq. 12 is fulfilled then the protein is cut at this position. This cut results in two new protein graphs to which algorithm 2 is applied iteratively until the procedure eventually comes to an end. This means, our algorithm does not rely on prior information about the number of expected domains but stops automatically.

III. Results

We demonstrate the applicability of our method for the identification of protein domains for one and two domain proteins. Our test set consists of 100 one domain and 71 two domain proteins. The mutual sequence similarity between the proteins was below 30% to exclude redundancy. Our results are compared with the manually assigned domains in the SCOP [9] database. Table I gives the parameter values used for the following simulations. These values were estimated from the application of our method to a training set consisting of $20/20$ proteins.

TABLE I
Optimized parameters of our model found by the application of our model to the training set.

L	α	N_r	Θ_r
12	0.8	150	0.45

We found, that our method can detect one domain proteins with an accuracy of 80%. Inspection of the significance ratio $\frac{\alpha E_{obj}(i_{\rm cut})}{\overline{E}^{r}_{obj}(i_{\rm cut})}$ for the wrongly cut proteins revealed, that 8 of these 20 proteins are only less than 5% over our cut criterion. This means, some slightly additional refinements of our algorithm should easily solve this problem. The results for the 71 two domain proteins are shown in the left Fig. 2. The abscissae gives the distance, measured in the number of secondary structure elements, from our predicted cut position to the assigned position by SCOP. The ordinate gives the percentage of proteins with a distance $\#SSE$. One can clearly see, that the center of mass of our predictions is centered arround zero. More precisely, 78.3% of all two domain proteins within ± 6 secondary structure elements were correctly assigned by our method. We want to mention, that normally, e.g., between a helix and a strand is a loop. Hence, in average from 6 secondary structure elements 3 will be loops. That means, that a rescaling of the abscissae in Fig. 2 by a factor $\frac{1}{2}$ gives the distance in coordinates of the significant structure-determining secondary structure elements. Additionally, we found that 11 proteins were cut two and 2 proteins three times. No protein occured with more than three or zero cuts. This indicates, that our significance ciriteria in Eq. 12 which is based on the randomization of protein graphs works well despite its simplicity. Our results are comparable well as the results found by DomainParser [18], [5] which is currently the best algorithm available to identify the domains of a protein [15].

To demonstrate, that the predictions with a distance > 10 secondary structure elements from the position assigned by SCOP are not necessarily wrong we present in Fig. 3 two

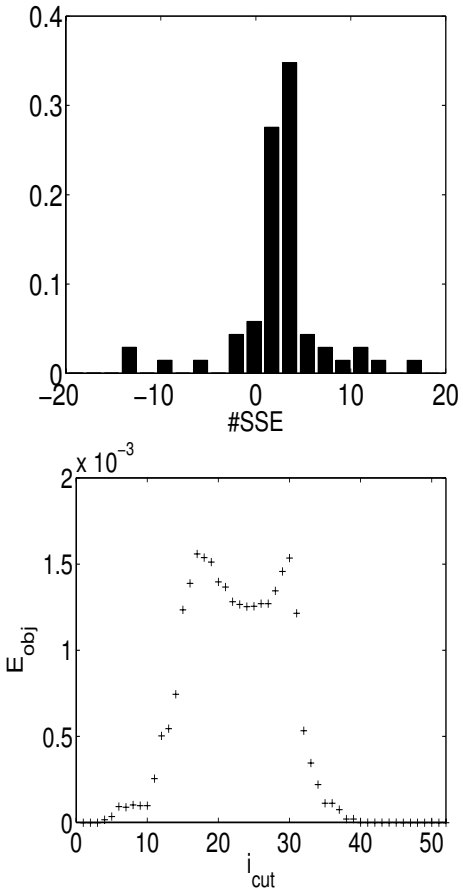

Fig. 2. Top: Normalized histogram for the predicted cut position of two domain proteins in relation to the assigned cut position by SCOP. Bottom: Objective Function E_{obj} for 1FTR chain A, shown in the right Fig. 3.

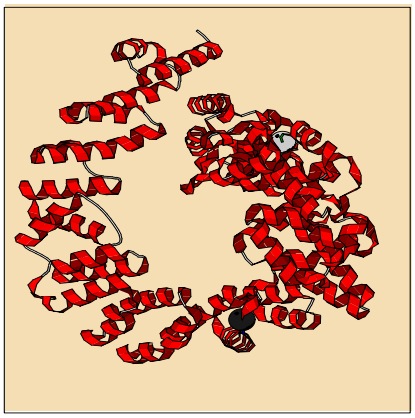

Fig. 3. Left: 1QSA chain A (soluble lytic transglycosylase SLT70). The assigned cut position by SCOP is indicated by the white ball, the predicted cut positon form our methode by a black ball. This figure was produced with Molscript [7].

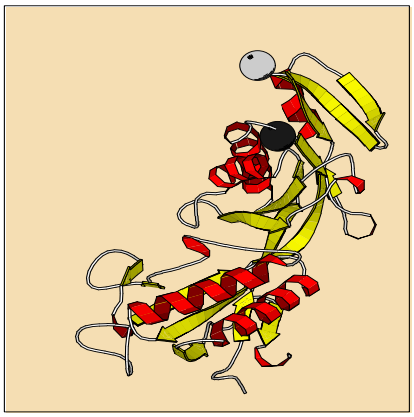

Fig. 4. 1FTR chain A (formylmethanofuran tetrahydomethanopterin formyltranferase). The assigned cut position by SCOP is indicated by the white ball, the predicted cut positon form our methode by a black ball. This figure was produced with Molscript [7].

such examples. The left Fig. 3 shows 1QSA chain A. The cut position assigned by SCOP is indicated by a white our predicted cut position by a black ball for clarity. One can see, that the helices are circularly arranged and our predicted cut position divides the circle roughly in two equal pieces separating the more loosely connected part on the left hand side of the figure from the more tangled helices on the righ hand side. This assignment, solely based on our graph theoretical measure is plausible. In contrast, the assigment by SCOP to cut the backbone in the tangled region can certainly not be found by our method because this cut destroys a huge amount of cycles as can be seen from E_{obj} (not shown). We think, the reason for this choice are beyond graph theoretical considerations and incorporate knowledge from biochemistry.

The second example in the right Fig. 3 shows 1FTR chain A. This case is interesting because the assigned cut position by SCOP dissects a beta-sheet. Again, our assigment is plausible and even preserves normally beta-sheets without the need of an explicite rule in opposite to other approaches for the protein domain identification, e.g., [14]. The reason therefore is, that the case shown in the right Fig. 3 namely a strand followed by a loop (or turn) followed by a strand results in a protein graph in a triangle connecting the corresponding secondary structure elements. A triangle is the smallest entity which can contain a cycle. Depending on the further progression of the backbone this triangle can be involved in a small or large number of cycles of different lengths within the protein graph. Again, there is no explicite rule preventing the cut of a beta-sheet but implicitely the dissection of a triangle will result in average in a loss of a certain number of cycles and, hence, the objective function in Eq. 10 which maximizes mutually the cycle-distributions found in both domains, will decrease. This can be see in the bottom Fig. 2 where we ploted E_{obj} for 1FTR chain A. This plot displays an interessting result. One can see, that the second highest E_{obj} value corresponds to the position of the SCOP assignment. The interpretation of this difference is enlightening because apparently the cut of the beta-sheet makes also sense for our graph-theoretical measure, however, as explained above this cut destroys inviteable a certain number of cycles and, hence, another position is favoured by our method which is difficult to find by visual inspection. This demonstrates a possible application of our

method assisting people to assin the domains of proteins, because our method provides in an objective way suggestions for such partitions.

IV. Conclusions

In this paper we introduced a novel algorithm for the identification of protein domains which design was data-driven. Our method is based on two steps. First, we represent each protein as undirected, unweighted and unlabeled graph which nodes correspond to the secondary structure elements of the protein. We call this graph the protein graph $G_{\mathrm{III.}}$ of a protein, because we distinguish between three types of secondary structure elements - helix, strand and loop. Second, we partition the protein graph by searching the cut position along the backbone which mutually maximizes the cycle-distributions found for this cut position for the remaining two protein subgraphs. The final decision, if this tentative cut position is accepted or rejected is made by comaring the value of the objective function for this position with a randomized objective function which is based on an ensemble of randomized protein graphs. This gives us a statistically significant decision criterion which accepts tentative cut positions only, if they are stable against a random rewiring of the protein graph under consideration. The algorithm we proposed is purely based on a graph theoretical entity namely a cycle. We want to mention explicitely, that we do not utilize other geometrical or physical parameters serving as additional rules to select cut positions as, e.g., in [18], [5]. Xu et al. proposed DomainParser [18] which is also based on a graph theoretical idea namely to find the minimum number of (weighted) cuts which separate a graph in two pieces. Due to the fact, that DomainParser needs to utilize half a dozen additional rules the merit of the graph theretical idea remains unclear. To our knowledge our approach is the only one which is soley based on a graph theoretical principle. This demonstrates not only that we found an algorithm which yields very good results but also that a treatment of the identification of protein domains is possible within the framework of graph theory enriched by a computational procedure. Hence, our proposed algorithm to approach the outstanding problem of the identification of protein domains is from a mathematical point of view even elegant.

Acknowledgments

We would like to thank Galina V. Glazko, Piotr Kozibal, Jing Liu and Arcady Mushegian for fruitful discussions and Mike Cooleman and Daniel Thomasset for computer support.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] A. Andreeva, D. Howorth, S.E. Brenner, T.J. Hubbard, C. Chothia and A.G. Murzin, SCOP database in 2004: refinements integrate structure and sequence fmily data, *Nucleic Acids Res.*, 32:D226-229, 2004.
[3] F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, E.F. Meyer Jr, M.D. Brice, J.R. Rodgers, O. Kennard, T. Shimanouchi, M. Tasumi, The Protein Data Bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535-542, 1977.
[4] R.F. Doolittle, The multiplicity of domains in proteins, *Annu. Rev. Biochem.*, 64:287-314, 1995.
[5] J-t. Guo, D. Xu, D. Kim and Y.Xu, Improving the performance of DomainParser for structural domain partition using neural network, *Nucl. Acids Res.*, 31(3)944-952, 2003.
[6] L. Holm and C. Sander, Dictionary of recurrent domains in protein structures, *Proteins: Structure, Function and Genetics*, 33:88-96, 1998.
[7] P.J. Kraulis, Molscript: A program to produce both detailed and schematic plots of protein sturctures, *J. of Appl. Chrystallograph*, 24:946-950, 1991.
[8] N.J. Mulder et al., InterPro, progress and status in 2005, *Nucleic Acids Res.*, 33:D201-205, 2005.
[9] A.G. Murzin, S.E. Brenner, T. Hubbard and C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.*, 247:536-540, 1995.
[10] D.C. Phillips, The dree-dimensional structure of an enzyme molecule, *Sci. Am.*, 215:78-90, 1966.
[11] A.S. Siddiqui and G.J. Barton, Contious and discontinuous domains: An algorithm for the automatic generation of reliable protein domain definietions, *Protein Science*, 4:872-884, 1995.
[12] G.D. Rose, Hierachic organization of domains in globular proteins, *J. Mol. Biol.*, 134(3):447-470, 1979.
[13] M.G. Rossmann and A. Liljas, Recognition of structural domains in globular proteins, *J. Mol. Biol.*, 85:177-181, 1974
[14] W.R. Taylor, Protein structural domain identification, *Protein Eng.*, 12(3):203-216, 1999.
[15] S. Veretnik, P.E. Bourne, N.N. Alexandrov and I.N Shindyalov, Toward consistent assignment of structural domains in proteins, *J. Mol. Biol.*, 339:647-678, 2004.
[16] D. Vitkup, E. Melamud, J. Moult and C. Sander, Completness in structural genomics, *Nat. Struct. Biol.*, 8(6):559-566, 2001.
[17] D. Wetlaufer, Nucleation, rapid folding, and globular intrachain regions in proteins, *Proc. Natl. Acad. Sci.*, 70:697-701, 1973
[18] Y. Xu, D. Xu and H.N. Gabow, Protein domain decomposition using a graph-theoretic apprach, *Bioinformatics*, 16(12):1091-1104, 2000.

Frank Emmert-Streib obtained his *Diploma* in Theoretical Physics in 1998 from the University of Siegen (Germany) and his Ph.D. in Theoretical Physics from the University of Bremen (Germany) in 2003. He is currently a postdoctoral research associate in Bioinformatics at the Stowers Institute for Medical Research (USA). His research interests include Computational Biology, Machine Learning and Systems Biology.