Fixing TIM: Identifying Functional Mutations in Protein Families through the Interactive Exploration of Sequence and Structural Data

John Wenskovitch* Timothy Luciani† Koonwah Chen‡ G. Elisabeta Marai§

University of Pittsburgh

ABSTRACT

We present the design and implementation of a visual mining and analysis tool to help identify protein mutations across family structural models, and to help discover the effect of these mutations on protein function. We follow a client-server approach in which distributed data sources for 3D structure and non-spatial sequence information are seamlessly integrated into a common visual interface. Multiple linked views and a computational backbone allow comparison at the molecular and atomic levels, while a trend-image visual abstraction allows for the sorting and mining of large collections of sequences and of their residues. We evaluate our tool on the triosephosphate isomerase (TIM) family structural models and sequence data, and show that our tool provides an effective, scalable way to navigate a family of proteins, as well as a means to inspect the structure and sequence of individual proteins.

Keywords: Molecular Sequence Analysis, Molecular Structural Biology, Computational Proteomics

1 INTRODUCTION

By determining the 3D structure and functionality of proteins, biologists can gain insight into the associated cellular processes, speed up the creation of pharmaceutical products, and develop drugs that are more effective in combating disease. A variety of protein-sequencing techniques are currently available; these techniques enable biologists to create, and later modify, sequences of amino acids linked to the structure of the protein. Mining this sequence information for internal structures may facilitate the discovery of correlations between specific structures and the protein functionality. However, the vast number of proteins sequenced by scientists make interactive visual mining tools necessary in solving this problem.

To improve the exploration process, many efforts have been made, from folding the sequences through classification [9][5], to tools for 3D view exploration [8] and to web-based applications which present large amounts of information to the users [7]. Nevertheless, challenges in solving this mining problem remain, from addressing scalability to spatial and non-spatial data integration to tool integration.

We introduce a novel visualization tool (Figure 1) to help identify protein mutations across families of structural models, and to help discover the effect of these mutations on protein function. Following a rigorous data and task analysis, we pursue a client-server approach in which distributed data sources for 3D structure and non-spatial sequence information are integrated. To better address scalability concerns, we aggregate family-sequence data into an interactive pixel-based abstraction called a trend image. Interactive exploration, multiple linked views, and details on demand further allow the generation of hypotheses regarding structure and functionality correlations in a diverse and fragmented space.

*e-mail: jwenskovitch@cs.pitt.edu

†e-mail:tbl8@cs.pitt.edu

‡e-mail:guc13@pitt.edu

§e-mail:marai@cs.pitt.edu

2 METHODS

2.1 Data and Task Analysis

Protein characteristics include structural information and amino acid sequence information. *The protein structure* — determined theoretically or experimentally – is typically stored in a PDB file [2], alongside references to the studies that determined the structure of the proteins, the residue sequence (the sequence of amino acids that make up the protein), and the positions of each atom in 3D space. The structural data can be visually mapped to a 3D representation of the protein, which includes atoms, bonds, amino acids and protein chains. *The amino acid sequence* of each protein is stored in remote databases, for example, Uniprot [6]. Each sequence consists of a string of capital letters, each letter representing an amino acid (also called a residue) in the protein. Sequences in the same protein family are usually manually aligned and expanded, with gaps introduced to better align common subsequences present across the family. A particular sequence family may include special sequences, some functionally-defective. Finally, external web services [6] may provide *additional relevant metadata and data*, such as model-quality ratings provided by domain experts.

Desirable features of a visual mining system, collected from the BioVis 2013 Data Contest, include:

- *Generate 3D protein structures* from sequence data
- *Inspect 3D* protein structures
- *Link to* online resources
- *Compare a single protein* to the rest of its family
- *Identify sequence mutation locations* on a family of proteins
- *Examine multiple sequence alignments*
- *Highlight specific residue locations* on the 3D protein structures
- *Examine residue distribution* across a protein family

2.2 Client-Server Framework

Given the variety and distributed nature of relevant domain data, we design and implement an overall client-server architecture (Figure 2). The server fetches and caches in a local MySQL database the protein sequences, alignment, and 3D structures from ModBase, Uniprot, and the NIH BLAST server [4]. The server provides sequence and 3D structure data to the client. If a 3D structure does not exist for the protein, the server computes an approximate model using the Sali Lab Modeller toolkit.

The client implements three core modules: a trend image module for exploring sequence data; an interaction manager for viewing; and an external reference module for access to online catalogues. The three modules are linked, allowing for simultaneous interaction with the data in each abstraction.

The back-end of the tool is implemented in Python, C, and MySQL. The front-end of the tool uses Python as the primary development language, with Qt for the GUI and the VTK-based Python Macromolecular Library [10] for rendering the protein structures.

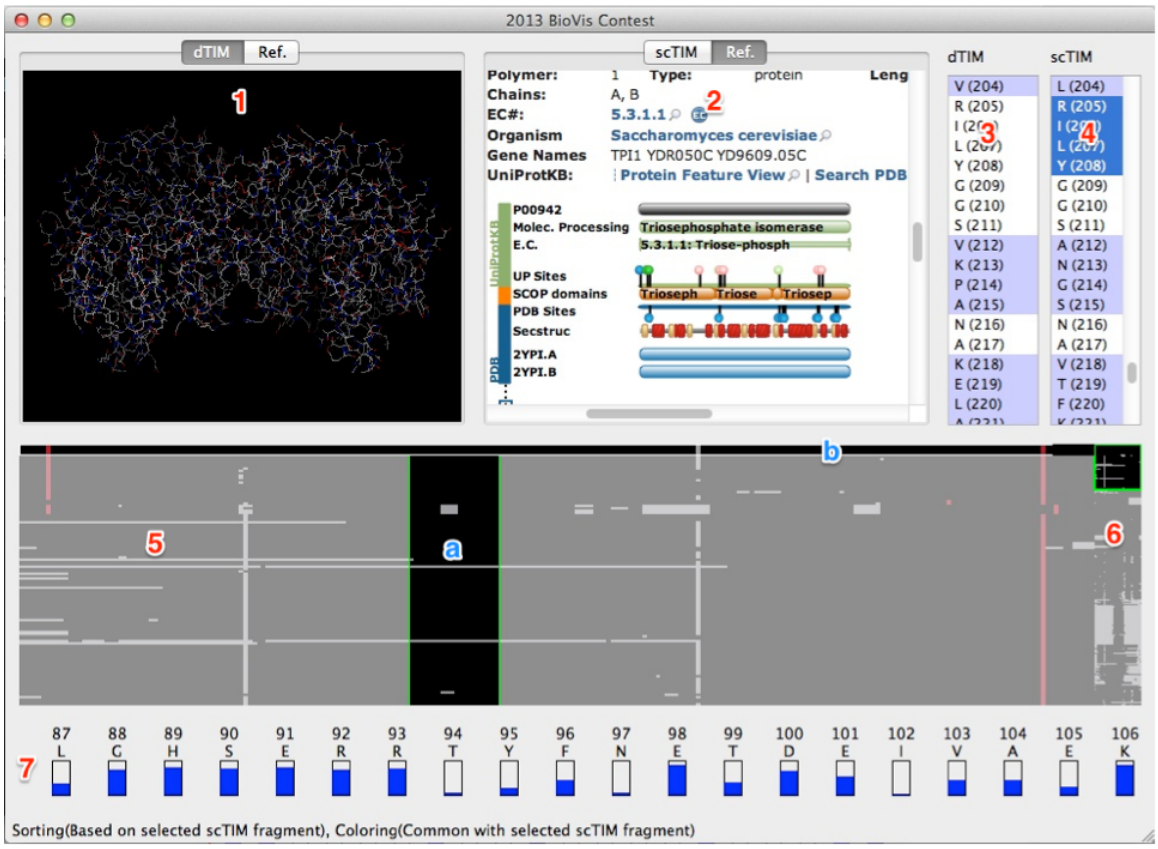

Figure 1: Visual interface with four panels: a 3D view and reference information panel (1 and 2); a protein sequence viewer (3 and 4); a trend image panel for aggregating protein families (5 and 6) with a fragment paddle (5a) and a sequence paddle (5b); and a residue view for residue distribution information (7).

2.3 Visual Design

Given the diversity and complementary nature of the data required, as well as the comparison nature of the domain-specific interactions, we pursue a linked multi-view top design. The visual interface consists of four linked panels (Fig. 1): a tabbed 3D structure and reference viewing panel (two side-by-side views); a side-by-side protein sequence viewer; a trend image panel; and a residue view for residue distribution information.

In this top design, the trend image view serves as the main anchor point of the interface. From this view, users can explore an entire protein family, and view the differences between family members. By right-clicking on a trend line, the user has the option of opening the structure file for this model in the 3D View, to compare it side-by-side with another model. Below, we describe each module in detail.

Trend Image Panel. The trend image view provides the ability to navigate and sort through large numbers of sequences. The trend-image is a pixel-based visual abstraction, in which each line represents the residues of a single residue sequence. The trend image summarizes an entire protein family, aligned by one of several sorting algorithms, and colored by one of several different color schemes. The trend image view further contains paddles for the selection of subsequences from a full family of protein sequences. These paddles also link to the residue distribution view at the bottom of the tool; this panel displays information about the distribution of amino acids, namely the fraction of proteins in the family that share the same residue as the selected protein.

A vertical overview pane (component 6 in Fig. 1) provides a high-level view of the full dataset; while the *fragment*-selection paddle allows narrowing the section of sequence considered for analysis and drilling for details. The sequence-selection paddle allows users to select a particular sequence. A selection event prompts the application to search for the 3D structure from online repositories; the 3D structure is presented if it already exists or it is generated on the fly if it does not.

To facilitate navigation of the trend image, we provide a set of sorting algorithms. The sorting algorithms calculate a weight for each sequence relative to one input member of a protein family, and then orders the sequences by their respective weights. We provide sorting by using the following measures as weights: fragment frequency; edit distance; weighted edit distance; number or percentage of common residues; number or percentage of common residues without regard to sequence position; number of residue subsequences of length N in common; and edit distance on selected residues.

In addition to the sorting algorithms, we also provide a ColorBrewer [3] set of coloring schemes to highlight a subset of the residues of each sequence. In each color scheme, black is used for residues that are not included in the scheme, whereas white is used to represent spacing in the sequence alignment. The residues in each internal class are given the same color. The list of coloring schemes is as follows: fragment frequency; general chemical characteristics; side-chain polarity; side-chain charge; and side-chain contact with polar solvent.

Residue Viewer. The residue distribution view displays the fragment ID, fragment name, and the percentage of each residue type found in the same column (corresponding fragment in each se-

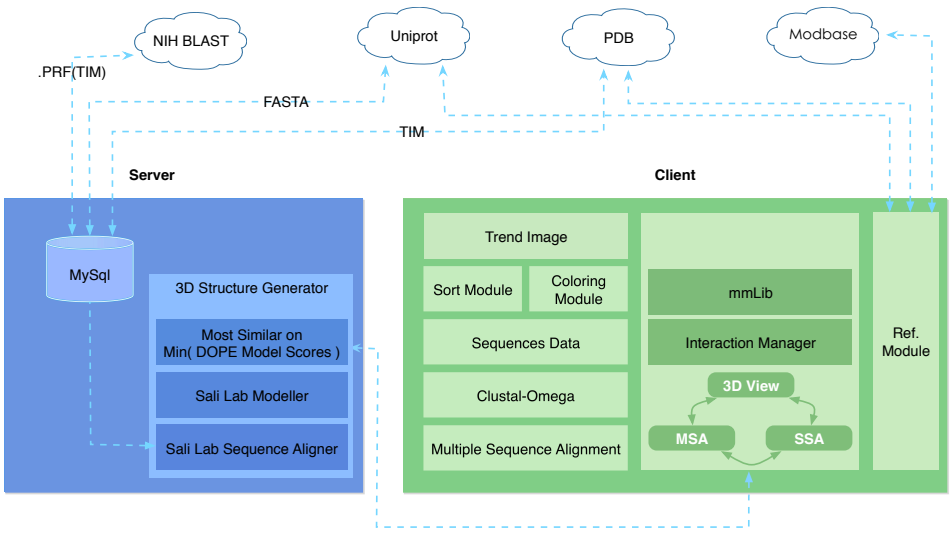

Figure 2: Client-server architecture diagram, with TIM-instantiation as an example application

quence) for all sequences.

3D Viewer. The top left panel of the tool provides two 3D structural views — one for the target protein, and one for the source protein, as well as a tabbed reference tool providing information about each protein. The structures can be examined at both the amino acid sequence level and at the atomic level. Through panning, zooming, rotation and details-on-demand operations (synchronized between the two views), users can observe different aspects of the two 3D structures.

Alternatively, the tabbed reference viewer allows users to access information from three complementary online data repositories: Uniprot, ModBase [1], and the RCSB Protein Data Bank. ModBase, for example, provides links to other databases, as well as ribbon diagrams for various models in the current sequence, and quality-criteria quantifying the reliability of certain model aspects.

Protein Sequence Viewer. To link in sequence information, the sequence panel lists the residue sequences for the selected structural models, with the differences between the two sequences highlighted in blue. The residues are selectable, and the selections are reflected in the 3D structure view.

3 Evaluation

We demonstrate our tool on a TIM protein-family application. These proteins play an important role in efficient energy production, and can be found in nearly every organism, including animals as well as fungi, plants, and bacteria.

The application examines the scTIM protein (saccharomyces cerevisiae triosephosphate isomerase), a member of the TIM family that was mutated towards the family consensus: a number of amino acids in the sequence were replaced by the most common residue found at that location in the TIM family. The resulting amino acid sequence is dTIM. Unfortunately, dTIM is functionally defective - one or more of the modifications made to scTIM caused the protein to lose its metabolic transport properties. Identifying which modifications caused the loss of functionality is an interesting open research problem.

For this application, we obtained the scTIM PDB, the TIM family sequence data and alignment information from the Battelle Center for Mathematical Medicine, through www.biovis.net. We used the tool to fetch 28 additional PDB files from RCSB, and to further generate more than 620 PDB files from the provided sequence data. We used the database backend to link PDB and FASTA IDs for preprocessing, and added data from ModBase & Uniprot.

Using our tool, we start by identifying the differences between the dTIM and scTIM sequences. There are 49 different subsequences of residues, encompassing 104 residues modified, created, or deleted in the creation of dTIM. By selecting some or all of these residues in the protein sequence viewer, we can highlight their locations on both 3D structures (Fig. 3). We can pan, zoom, and rotate the structures to more closely examine the distribution of these alterations on the protein structure. We can also adjust the rendering properties of the structure.

To determine which models from the TIM family are most similar to the original scTIM, we use the trend-image view in the lower panel. In Fig. 3, we can quickly see, for example, that only a few sequences have the same fragment in position 142 with scTIM. A step further, selecting any of the sorting modes from the menu allows comparisons to be made to scTIM. For example, when sorting by common residues, the TPIS_HAEDU, a Haemophilus bacteria, shares the greatest number of residues with scTIM. Selecting a particular coloring method displays specific information for each residue.

Manipulating the vertical selection paddle allows us to explore subsequences of the full TIM sequence. Distribution information about residues in the highlighted subsequences are displayed below the trend image, and show the most common amino acid in the TIM family at each sequence index. The bars in the residue viewer that are nearly empty imply that very few members of the TIM family share the same residue as scTIM, making it an ideal candidate for mutation towards the family consensus.

Manipulating the horizontal selection paddle allows us to further explore the individual TIMs in the family, with a fish-eye lens expanding the selected row to more clearly show the residue sequence and coloring. Right-clicking on a selected row allows us next to load the structure of that specific TIM into the structure view. If this TIM is unfamiliar to the user, a number of reference databases can be accessed.

4 Discussion and Conclusion

The TIM application shows that our tool can assist in the navigation of family of proteins, as well as in the exploration of individual protein structures. The side-by-side 3D views facilitate visual comparison, while the trend image abstraction provides an effec-

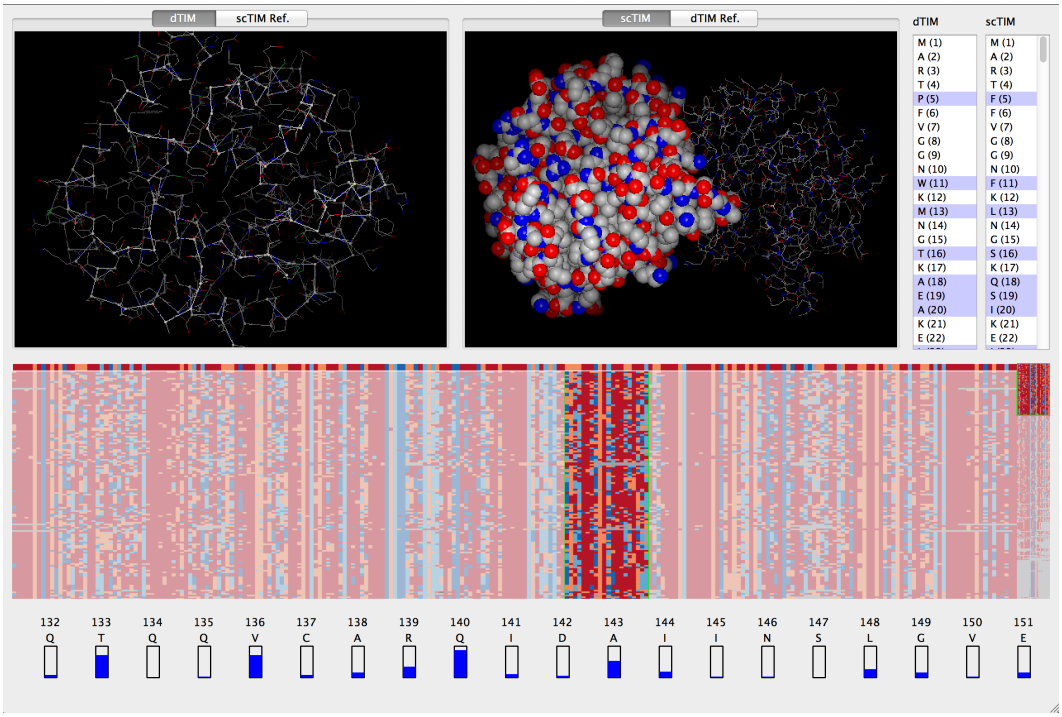

Figure 3: scTIM/dTIM comparison. The two volume views show the dTIM protein backbone (left), respectively a CPK sphere representation of scTIM (right). The trend image is sorted by the number of common residues. A side-chain polarity coloring is applied, and the vertical selection paddle is centered around position 142.

tive view and exploration of large collections of sequence data. Our tool further integrates successfully multiple sources of information, and both spatial and non-spatial data. Furthermore, a computational backbone facilitates sorting collections of sequences, as well as generates 3D structures for modified sequences.

In terms of limitations, while the trend image provides a scalable approach to viewing large amounts of sequence data, finding a particular sequence in a protein family remains a challenge. Similarly, attempting to code too much information into the color schemes results in an overload of colors, rendering the trend image unreadable and ineffective. A reduction in the number of colors restores readability to the view, at the cost of removing some information from the trend image.

In conclusion, we introduced a novel visualization tool that integrates 3D structural information and sequence information for a protein, with additional information from the multiple sequence alignment of the family of proteins with the same function, and with meta information extracted from the family data. In conjunction with domain expert knowledge, this interactive tool can help provide biophysical insight into why specific mutations affect function, and potentially suggest additional modifications to the protein that could be used to rescue functionality.

Acknowledgements

This work has been supported by grant NSF-IIS-0952720 and by the NSF Graduate Research Fellowship program. Many thanks to Adrian Maries, Xinghua Lu and the VisLab group for feedback and useful discussions.

References

[1] U. P. amd Narayanan Eswar, B. M. Webb, D. Eramian, L. Kelly, D. T. Barkan, H. Carter, P. Mankoo, R. Karchin, M. A. Marti-Renom, F. P. Davis, and A. Sali. Modbase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res*, 37(Database issue):D347–54, 2009.

[2] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[3] C. A. Brewer. 2009.

[4] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. Madden. Blast+: Architecture and applications. BMC Bioinformatics, 2009.

[5] N. Chen, T. W. Harris, I. Antoshechkin, C. Bastiani, T. Bieri, D. Blasiar, K. Bradnam, P. Canaran, J. Chan, C. kung Chen, W. J. Chen, F. Cunningham, P. Davis, E. Kenny, R. Kishore, D. Lawson, R. Lee, H. michael Muller, C. Nakamura, S. Pai, P. Ozersky, A. Petcherski, A. Rogers, A. Sabo, E. M. Schwarz, K. V. Auken, Q. Wang, R. Durbin, J. Spieth, P. W. Sternberg, and L. D. Stein. Wormbase: A comprehensive data resource for caenorhabditis biology and genomics. *Nucleic Acids Res*, 33:383–389, 2005.

[6] T. U. Consortium. Update on activities at the universal protein resource (uniprot) in 2013. *Nucleic Acids Research*, 41(D1):D43–D47, 2013.

[7] W. Kent, C. Sugnet, T. Furey, K. Roskin, T. Pringle, A. Zahler, and D. Haussler. The human genome browser at ucsc. *Genome Research*, 12(6):996–1006, 2002.

[8] S. B. Montgomery, T. Astakhova, M. Bilenky, E. Birney, T.Fu, M. Hassel, C. Melsopp, M. Rak, A. Robertson, M. Sleumer, A. S. Siddiqui, and S. Jones. Sockeye: A 3d environment for comparative genomics. *Genome Research*, 14:956–962, May 2004.

[9] W. Zhong, G. Altum, R. Harrison, P. C. Tai, and Y. Pan. Mining protein sequence motifs representing common 3d structures. In *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference - Workshops*, CSBW '05, pages 215–216, Washington, DC, USA, 2005. IEEE Computer Society.

[10] F. Zucker, P. Champ, and E. Merritt. Validation of crystallographic models containing tls or other descriptions of anisotropy. *Acta. Cryst.*, D61:889–900, 2010.