Glyph-based Overviews of Large Datasets in Structural Bioinformatics

Aneesh Karve, Michael Gleicher
University of Wisconsin–Madison
{karve, gleicher}@cs.wisc.edu

Abstract

The exponential growth of structural bioinformatics data implies a need for visualization tools that provide informative overviews of large datasets. Nevertheless, online query interfaces and domain-specific visualization tools have neglected to provide suitable overviews. We identify four functional goals for large-scale overviews. We apply these goals to create glyph-based visualizations of query results from the Protein Data Bank.

Keywords—**overview, glyph, protein, ligand**

1 Introduction

Structural biologists aim to determine the shape and function of macromolecules like proteins and RNA. Increasing success in the field has flooded structure databases like the Protein Data Bank [2](PDB). At present the PDB contains more than 40,000 structures and is poised to continue growing at an exponential rate [28]. Databases like the PDB have query interfaces that support precise searches, but as structural data accumulates, even precise searches lead to large data collections. Structural biologists therefore need tools to help them explore and comprehend large data collections. These tools should support the discovery of trends, outliers and relevant subsets of data collections. Automated data mining can support some of these discoveries, but experts have emphasized the need for human insight supported by visualization [7, 18].

In this paper we show how visual overviews can be created to support information seeking over large collections of structural data. Overviews can increase the rate at which information is acquired, help to expose patterns and outliers [7, 21], reduce the need for search, and help the user to choose subsequent actions [6]. Although a variety of visualization tools and database interfaces enable the study of individual structures, few if any provide overviews of numerous structures in parallel (Section 2.2). In Section 3 we propose functional goals for large-scale overviews. In Section 4 we apply these goals to create novel, glyph-based visualizations of PDB query results. The PDB's web query interface has been chosen as the touchstone for our research due to the PDB's leading role in structural biology.

In a typical month the PDB's website serves half a terabyte of data and receives more than 100,000 visitors [14].

The motivations and methods for our research are illustrated by the following example. A PDB keyword search for "adenylate kinase," filtered for 90% sequence identity, returns 86 structure hits and 64 ligand hits. With the conventional display format, shown in Fig. 2, these hits span two tabs and more than twenty screens on a 17-inch display. With our display format, shown in Fig. 1, the same hits occupy less than half a screen. After reading a brief textual description, as in the caption to Fig. 1, users can quickly discover the smallest or most recent structures, the structures that bind adenosine diphosphate, etc. Users can also generalize about the result set: the structures span a wide range of molecular weights, every structure binds at least one ligand, no structure has more than six subunits, and so on. By way of comparison, the aforementioned discoveries and generalizations are not as easy to make with the PDB's web interface, as discussed in Section 2.3 and Section 3. In short, we hypothesize that overviews can reduce the cost of knowledge from structure databases like the PDB.

2 Related work

Prior research has explored overviews, glyphs, and the perceptual and cognitive issues that they engender. In this section we highlight relevant findings, then sketch the state of the art in bioinformatics visualization, and establish the need for overviews.

2.1 Glyphs, perception and cognition

Glyphs are graphics whose visual attributes are determined by data. They have been widely studied and applied in the visualization of multivariate data. Ward [25] provides a broad survey of glyph types and placement strategies. Ware [27] discusses the perceptual aspects of glyph design. For instance, Ware notes that preattentive visuals are processed more slowly as the the number of visual distractors increases. Distractors merit consideration in multi-glyph displays (Section 4) wherein numerous glyphs may crowd the visual field. We note that although distractors can slow the visual consumption of glyphs, individual glyphs remain informative in the context of thousands
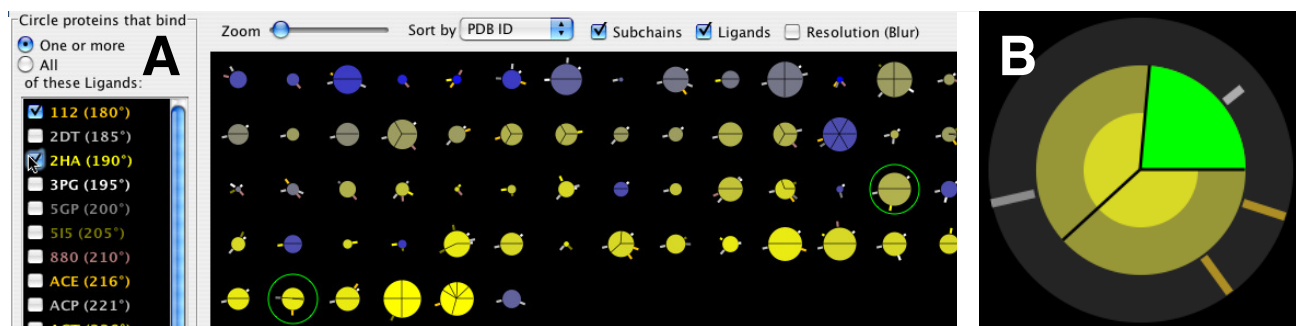
Figure 1: A) PDB structure and ligand hits in PDQVis. The dynamic query interface (left) enables users to select ligands and query for proteins that bind the same; query results are circled in green. **B)** Close-up of a structure glyph. Each glyph resents a PDB structure, its subunits (pie slices), number of residues (circle size), release date (blue is older, yellow newer), bound ligands (each radial whisker represents a ligand; the length of the whisker can encode the ligand's molecular weight), and resolution (blurred halo). User-selected subunits are brushed in green across the entire visualization.

of others and mega-glyph visualizations can produce useful gestalt effects which supersede the effect of individual glyphs [15].

A further consideration in multi-glyph displays is their impact on working memory. Any information that is retained for more than 0.3 seconds must pass through working memory, which has a limited capacity of about six chunks of information [27]. If the capacity of working memory is exceeded, "cognitive overload" [9] results. An important counterbalance to cognitive overload is integration. For example, Ware argues that a glyph that integrates multiple visual attributes makes more efficient use of working memory than a glyph based on a single visual attribute. (That said, for a multi-attribute glyph to be easy to read, the constituent visual attributes should be mutually separable [27]). Similarly, Meyer and Moreno [9] identify split attention as a source of cognitive load and recommend "integrated presentation" as an offset to cognitive load. Integrated presentation places related targets of attention, such as two glyphs that a user would be likely to compare, in close proximity (cf. [24, Ch. 4]). This echoes Tufte's thinking on small multiples, for which he recommends that comparisons be "enforced within the scope of the eyespan," [22]. Tufte further contends that small multiples should emphasize *what changes* in the data and should do so by providing variations on a repetitive, unifying theme. We conclude from the work mentioned in this paragraph that individual glyphs should combine as many separable visual attributes as possible and that multi-glyph displays should be dense, juxtapose related items, and employ repetitive design motifs that support inter-glyph comparison.

Siirtola [19] showed that users prefer data-related glyphs and can read them more accurately. We view this as a special case of what Norman calls "natural mapping" [12] and infer that a clear relationship between visual and data attributes will enhance the usability of glyphs. We exploit natural mappings to display quantiative variables like size and resolution (Fig. 1).

The glyphs we create for PDB structures and bound ligands (Fig. 1) combine elements of pie glyphs [26], whisker plots, and metroglyphs [1]. In Section 4.2 we explore the use of statistical graphics as data summaries and filter controls. The broader topics of statistical graphics and glyphs as widgets are explored by Tufte [23] and Calder et al. [5], respectively.

2.2 Visualization tools

Glyphmaker [15] allows users to create custom visualizations by specifying the visuals-to-data mapping or "visualization schema." Polaris [20] takes this a step further and provides an interactive query mechanism. That said, Polaris supports neither complex nor compound glyphs. (By "complex" we mean glyphs built from more than three visual attributes. By "compound" we mean glyphs built from two or more simpler glyphs.). Our research suggests that complex, compound glyphs are useful for visualizing bioinformatics data (Section 4.1). Kreuseler et al. [8] presented a generic framework for information visualization, but their approach to multivariate data yields a relatively limited set of glyphs: those that can be built from cylindrical primitives.

Although the startup cost of the aforementioned tools makes them impractical for working scientists, such tools could be used to prototype standard visualizations for data clearinghouses like the PDB. Table-based displays, proposed in Table Lens [13], are a natural method for organizing database visualizations. Table Lens also provides a focus+context mechanism for enhancing details. In Section 4 we present a table-based overview with an alternative focus+context technique.

Domain-specific visualization tools have focused on either structure or network visualization. In neither case has generic, large-scale overview been addressed. Chapter 7 of [3] provides a survey of structure visualization tools. Such tools are widely used and offer sophisticated features, but they primarily support the study of individual structures. Network visualization tools like Osprey [4] support the study and comparison of multi-molecular networks, but they focus on just one aspect (interactions) of specific datasets (interactomes). Our goal is to provide overviews of ad hoc datasets, wherein interactions may not be known or may not exist.
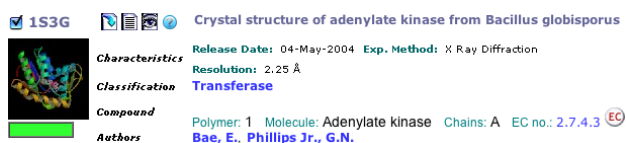

Figure 2: A structure hit from a PDB web search.

2.3   The PDB's web interface

Online queries to the PDB generate structure hits in the format shown in Fig. 2. Users with a 17-inch display can view about six of these hits in a browser window. As a result, large collections cannot be accommodated in a single view. A user must employ her memory to assemble an overview of the collection. Furthermore, the hit format relies on descriptive text, rather than visualization, to convey information. While such an approach is effective for conveying information about individual structures, it does not scale to large collections of structures.

The PDB also offers a "collage view" of structure hits. Collage view is a uniform tiling of thumbnail images, one for each hit, as shown in Fig. 3. This is a step in the direction of overview, but it fails to be synoptic and comparative. PDB collages are not synoptic since they expose only one facet of the dataset: small, fixed-perspective images of three-dimensional shapes. Such images can indicate the rough shape of molecules, but they do not afford accurate comparison, due to problems of perspective (Fig. 3).
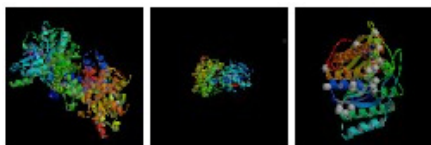
Figure 3: Thumbnail images from a PDB collage. The images belie the relative molecular weights of the structures: 66kDa (structure 2ACX), 60kDa (2AJ4), and 36kDa (2AJP).

2.4   Overview is missing

We have argued that current tools for structural bioinformatics do not provide adequate overviews. We propose to supplement current tools with new ones, explicitly designed for the large-scale overview of structural datasets. These new tools can act as a bridge between the growing stream of data and the existing crop of small-scale visualization tools. In the following section we present functional goals for large-scale overviews. We follow in Section 4 with examples of how we have applied these principles to create specialized displays for the visual overview of PDB query results.

3   Functional goals for large-scale overviews

A critique of the PDB hit format (Fig. 2) suggests functional goals for overview displays. First, since only a small number of hits appear on a page, the user will need to page back and forth to view the entire collection. Paging is likely to tax the user's working memory and thereby increase cognitive load. Second, the format is not well-suited for comparison between elements. Third, the format is not designed for quick, visual browsing since it consists primarily of descriptive text, which requires more time and effort to interpret than preattentive visual attributes (i.e. glyphs). Lastly, we note that the PDB website succeeds in offering methods to filter, sort and expose details of query results. These methods fall under the umbrella of "zoom and filter" operations, which are useful for overview displays (cf. [17]).

The four issues mentioned above suggest the following goals for overview displays:

1. Overview displays should be **synoptic**. They should provide monolithic views that summarize the entire dataset. In other words, overviews support the discovery of trends and outliers in the data. (In Section 4.2 we show how statistical graphics and brushing can be used to accomplish the former.) In order to support large-scale datasets, overviews should be built from information-dense elements that are interpretable at small sizes.

2. The elements of overview displays should be **comparative**. "Comparative" implies a consistent visualization schema that facilitates inter-element comparison. (PDB thumbnails, shown in Fig. 3, are not comparative: since there is no common perspective, structurally similar molecules may appear dissimilar, and vice versa.) In order to support comparison, we have found it useful to emphasize how the data relate to one another, rather than precise data measurements. Existing tools excel at the latter and fail at the former (Section 2.2). To some extent, an empha-

sis on inter-data comparison is a natural consequence of large-scale overview, which trades precision for screen space. We revisit this issue in Section 4.1.4.

3. Overview displays should be **perceptually efficient**, as discussed in Section 2.1. The goal is to make visual information seeking as quick and easy as possible.

4. Overview displays should provide **zoom and filter** controls that enable users to subset, sort, and expose details in the data. In light of integrated presentation (Section 2.1), focus+context is preferable to other overview+detail methods. [6] provides a survey of zoom and filter techniques.

Given the need for dense, comparative, perceptually efficient visual elements, and given the abundance of multivariate data in structural bioinformatics, we have chosen glyphs as the building blocks of our overview visualizations.

4 Glyph-based overviews

We have created two software prototypes for glyph-based overviews of PDB query results: PDQVis (Fig. 1) and BioSpark (Fig. 4). The former represents each structure and its attributes as a compound glyph. The latter represents each structure as a row of simple glyphs, one per data attribute. These simple glyphs are then organized into a table wherein each column represents a data field.

Pursuant to the functional goals established in Section 3, Section 4.1 proposes glyphs for data types common to structural bioinformatics. These glyphs may be combined into compund glyphs. We identify three considerations for designers of complex and compound glyphs: integral-separable dimension pairs, natural mappings, and the perceptual efficiency of the encoding. The former two considerations are mentioned in Section 2.1. We give an example of the latter consideration in Section 4.1.2. Fig. 1a shows a compound glyph design which applies the aforementioned considerations to combine six data attributes into a single graphic.

4.1 Glyphs by data type

4.1.1 Descriptive text. With the exception of short, critical strings such as the structure identifier, descriptive text can be excluded from the first layer of the overview. To read and interpret a description is a slow, post-attentive process that does not lend itself to rapid visual exploration. Descriptive text can be provided through tooltips or other zoom mechanisms.

4.1.2 Scalars. In table-based displays, continuous scalars such as structure resolution and molecular weight can be visualized as horizontal bars on a shared positional scale (Fig. 4, fourth column). This is the most accurate

method for visually encoding a continuous variable, followed in order of decreasing accuracy by interval length, slope, area, volume, and color [16]. Depth of field (blur) can also be used encode scalars (e.g. structure resolution in Fig. 1a). Small, integral scalars can be depicted by numerosity of marks, as with the number of subchains or ligands in Fig. 1a.

4.1.3 Indicator functions. Fields in a structure database may associate a structure with elements of a discrete set. For example, the "experimental method" field of a PDB record indicates one member of the set {X-Ray Diffraction, NMR}. Similarly, the set of ligands bound by a given structure is a subset of the set of all ligands in the database. We can therefore conceptualize the values of a field like "bound ligands" or "authors" as a family of indicator functions[1].

A family of indicator functions is comparative since its members share the same domain: the union of all observed values for the corresponding data field. For the purposes of visualization, we arrange the elements of the shared domain across a common axis. If the common axis is horizontal, and vertical marks are placed wherever the indicator function is nonzero, the visualization looks something like a barcode (see the author sparklines[2] in Fig. 4). If the common axis is radial, and each glyph has its own axis, the visualization may look like Fig. 1. When the common axis is tightly packed, color can be used to distinguish neighboring elements of the domain, as with the ligand whiskers in Fig 1.

It can be useful to expand the range of the indicator function (in which case it resembles a time series). For example, the length of the radial whiskers in Fig. 1 indicates the molecular weight of the corresponding ligand. We call the visual signature created by these whiskers a "ligand aura." Ligand auras, and indicator function glyphs in general, provide rapidly comparative visual signatures wherein shared marks indicate common traits (and conversely). The practical value of ligand auras is revealed in Fig. 1, wherein users can quickly infer that the first four structures in the second row all bind the same ligand (depicted by the gray whisker near the eight o'clock position).

4.1.4 Classification trees. The Enzyme Nomenclature tree [11] provides functional classifications for proteins via Enzyme Classification (E.C.) numbers. An E.C. number is of the form $a_0.a_1 \ldots a_n$ where $a_i \in \mathbb{N}, 1 \leq n \leq 3$. It identifies a path, rooted at a_0, descending the nomenclature tree. One can visualize E.C. numbers as glyphs according to a simple schema: the a_i are represented by n

---

[1] An indicator function for a set $B \subseteq A$ is of the form $f : A \rightarrow \{0, 1\}$ and obeys $f(a) = 1 \Leftrightarrow a \in B$.

[2] The term "sparkline" was introduced by Tufte [24]. In practice, a sparkline is simply a data-dense glyph.
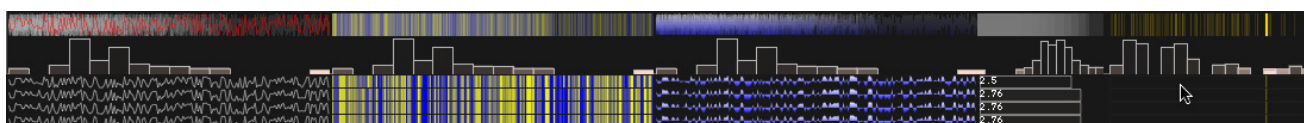
Figure 4: Biospark, a table-based display in which each data field is represented as a column of glyphs. From left to right the columns depict primary structure (three visualization styles), resolution, and authors. The focus+context technique is shown along the top of the figure: the author sparkline under the mouse pointer (right) is shown in context, in bold, at the top of the corresponding column; likewise for the red sparkline in context along the top of the first column.

rectangles in the same left-to-right order, the height and lightness of each rectangle is directly proportional to i, and the vertical position of each rectangle is proportional to a_i. Lastly, a polyline or "scribble"—we call these glyphs "scribble trees"—unifies the rectangles and provides a visual signature. The results are shown in Fig. 5. We note that scribble trees emphasize data comparison over data content (Section 3). For example, it is easy to ascertain that all of the E.C. numbers depicted in Fig. 5 share their top two levels, but it is relatively difficult to determine the precise E.C. number for a given glyph.
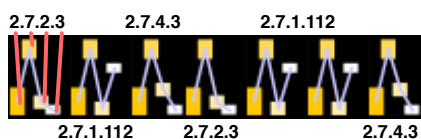
Figure 5: Scribble trees depicting Enzyme Classification numbers.

4.1.5 Protein structure. Sparklines are a natural method for summarizing primary or secondary structure. The former can be visualized with hydrophobicity plots. The first three columns of Fig. 4 show hydrophobicity plots as contours, height maps, and a combination of contour and height mapping. When displayed in columns, hydrophobicity sparklines make substitutions, shifts and insertions strikingly visual. (For instance, the yellow and blue primary structure sparklines at the bottom of Fig. 4 show three identical chains and one homologue, which appears to have diverged via substitution.) Since sparklines for primary and secondary structure can be aligned for easy comparison and can be rendered in 2D without occlusion, they circumvent some drawbacks of 3D structure visualization (Section 2.3 and Section 3) but convey some of the same information.

To visualize tertiary structure, scribble trees of a structural classification like SCOP [3] can be used. We hypothesize that such abstract representations of 3D structure are, at small sizes, easier to interpret than concrete 3D visualizations like PDB thumbnails (Fig. 3).

At low levels of zoom, 2D glyphs can be condensed into style boxes [10]. Each style box has an x- and y-axis di-

Figure 6: Style boxes summarizing the primary structure of proteins.

vided into n bins. This forms an $n \times n$ grid in which a single cell, representing one of n^2 styles, is highlighted. The 3×3 style boxes in Fig. 6 summarize the number of residues (horizontal bins) and average hydrophobicity (vertical bins) of 28 protein chains. Glancing over Fig. 6 reveals that every protein shown is of average length and that the majority of proteins have, comparatively speaking, a low mean hydrophobicity.

4.2 Statistical graphics, focus+context

Small statistical graphics can be displayed within overviews as data summaries. BioSpark provides a histogram at the top of each data column (Fig. 4). The histogram reveals how data in the column are distributed. Modal (tall) bars are filled with subdued colors while extremal (short) bars are, in inverse proportion to their height, filled with brighter colors. Each glyph in the column beneath is brushed with the color of its histogram bar. This "statistical brushing" may support the discovery of patterns and outliers. A summary of a data column can be generated by selecting a representative glyph from each bar in the histogram, then superimposing the representatives at high transparency. The result is an aggregate glyph summarizing the values of the data column. BioSpark's focus+context mechanism highlights the focus and renders it over the context (i.e. the aggregate glyph), as shown in Fig. 4.

Statistical graphics can double as filter widgets. In the case of histograms, or probability distribution functions, the user selects one or more regions of the graphic to isolate the corresponding data.

Conclusions and Future Work

We have argued that large-scale overviews are needed as an efficient interface to rapidly growing structure databases. We have proposed general design principles and concrete solutions for such overviews.

One limitation of the PDQVis visualization technique is

that it can display no more than about 2500 structures on a 17-inch display. Larger datasets may require a new level of logical zoom, which we have yet to design.

User studies are needed to assess the practical value of dataset overviews in structural bioinformatics. A first study might identify common knowledge-based tasks performed by PDB users and, using the PDB's web interface as a benchmark, measure the performance of these tasks in overview applications like PDQVis and BioSpark. A related study might determine at which levels of zoom, and for which types of data, table-based displays like BioSpark are more usable than data-driven glyph displays like PDQVis.

Acknowledgments

References

[1] E. Anderson. A semigraphical method for the analysis of complex problems. *PNAS*, 43:923–927, 1957.

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

[3] P. E. Bourne and H. Weissig, editors. *Structural Bioinformatics*, chapter 7, 9, 12, 13. Wiley-Liss, 2003.

[4] B.-J. Breitkreutz, C. Stark, and M. Tyers. Osprey: a network visualization system. *Genome Biology*, 4(3), 2003.

[5] P. R. Calder and M. A. Linton. Glyphs: flyweight objects for user interfaces. In *Proc. ACM SIGGRAPH UIST*, pages 92–101, 1990.

[6] S. K. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[7] D. A. Keim. Information visualization and visual data mining. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[8] M. Kreuseler, N. Lopez, and H. Schumann. A scalable framework for information visualization. In *Proc. IEEE InfoVis*, pages 27–36, 2000.

[9] R. E. Meyer and R. Moreno. Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, 38(1):43–52, 2003.

[10] Morningstar. Fact sheet: The new morningstar style box methodology. `http://news.morningstar.com/pdfs/FactSheet_StyleBox_Final.pdf`, 2002. [Online; accessed 16-Feb-2007].

[11] NC-IUBMB. Enzyme nomenclature. `http://www.chem.qmul.ac.uk/iubmb/enzyme/`, 1992. [Online, accessed 28-Jan-2007].

[12] D. A. Norman. *The Design of Everyday Things*. Basic Books, September 2002.

[13] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proc. ACM CHI*, 1994.

[14] RCSB PDB. PDB Newsletter. `ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/rcsb`, Winter 2007. [Online; accessed Feb-2007].

[15] W. Ribarsky, E. Ayers, J. Eble, and S. Mukherjea. Glyphmaker: Creating customized visualizations of complex data. *Computer*, 27(7):57–64, 1994.

[16] C. D. Shaw, J. A. Hall, C. Blahut, D. S. Ebert, and D. A. Roberts. Using shape to visualize multivariate data. In *Workshop on New Paradigms in Information Visualization and Manipulation*, pages 17–20, 1999.

[17] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. on Visual Languages*, 1996.

[18] B. Shneiderman. Inventing discovery tools: combining information visualization with data mining. *Information Visualization*, 1(1):5–12, 2002.

[19] H. Siirtola. The effect of data-relatedness in interactive glyphs. In *Proc. of the 9th International Conf. on Information Visualization*, pages 869–876, 2005.

[20] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[21] S. T. Teoh, K.-L. Ma, S. F. Wu, and T. J. Jankun-Kelly. Detecting flaws and intruders with visual data analysis. *IEEE Comput. Graph. Appl.*, 24(5):27–35, 2004.

[22] E. R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[23] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, second edition, 2001.

[24] E. R. Tufte. *Beautiful Evidence*. Graphics Press, July 2006.

[25] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

[26] M. O. Ward and B. N. Lipchak. A visualization tool for exploratory analysis of cyclic multivariate data. *Metrika*, 51(1):27–37, 2000.

[27] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2nd edition, 2004.

[28] Wikipedia. Protein data bank — wikipedia, the free encyclopedia. World Wide Web, 2006. [Online; accessed 3-December-2006].