

GLYPH-BASED OVERVIEWS OF LARGE DATASETS
IN STRUCTURAL BIOINFORMATICS

by
Aneesh Padmakar Karve

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
(Computer Sciences)

at the
UNIVERSITY OF WISCONSIN-MADISON
2007

Approval

This thesis has been approved by the following faculty members.

Advisor

Michael Gleicher, Associate Professor of Computer Sciences

Signature _____ Date _____

Second reader

George N. Phillips, Jr., Professor of Biochemistry with an appointment in
Computer Sciences

Signature _____ Date _____

Contents

Acknowledgments	v
Abstract	1
1 Introduction	3
1.1 Contribution and organization of this thesis	4
2 Related Work	5
2.1 Glyphs, perception and cognition	5
2.1.1 Retinal attributes, preattention and pop-out	6
2.1.2 Perceptual color sequences	8
2.2 Working memory and cognitive load	9
2.2.1 Readable, meaningful glyphs	9
2.3 Overview and visual data mining	11
2.3.1 Visual data mining	12
2.4 Visualization tools	12
2.4.1 Tools for glyphs, databases and query results	13
2.4.2 Bioinformatics visualization tools	13
3 Functional Goals for Large-scale Overviews	17
3.1 Four goals for overviews	17
3.2 Glyphs and overview	18
4 Research Software	19
4.1 Glyphs by data type	19
4.1.1 Images	19
4.1.2 Descriptive text	19
4.1.3 Scalars	24
4.1.4 Indicator functions	24
4.1.5 Classification trees	25
4.1.6 Protein structure	25

4.1.7	Statistical graphics, focus+context	26
4.2	Complex and compound glyph design	26
5	Conclusions and Future Work	31
	Bibliography	35
	List of Figures	38
	List of Tables	40

Acknowledgments

Mike Gleicher advised my thesis work. He contributed ideas, directions, and motivation throughout. Mike's support made it possible for me to pursue Information Visualization and Human-Computer Interaction as part of my graduate studies. I'm grateful for the opportunity.

George N. Phillips, Jr. lent his expertise in structural biology to the project. George's domain knowledge helped steer the research in directions relevant to working scientists.

Rachel Heck and Nicholas Penwarden critiqued a short paper based on this thesis. Their suggestions were applied to the thesis as well.

In the past I've wondered why authors thank loved ones in the Acknowledgments. Now I get it. Mom, Dad, Sonu, Bharatee, Thakar, and R.S., you got me here and I thank you for it.

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.

Chapter 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 Bank¹ (PDB). At present the PDB contains more than 40,000 structures and is poised to continue growing at an exponential rate [Wik06b]. 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 provide some insight into large collections but experts have emphasized the need for human insight supported by visualization [Kei02, Shn02].

In this thesis 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 [Kei02, TMWJK04], reduce the need for search, and help the user to choose subsequent actions [CMS99]. 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 2.4). In Section 3.1 we propose functional goals for large-scale overviews. In Chapter 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 [RCS07].

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. 2.3, these hits span two tab pages and more than twenty screens on a 17-inch display. With our display format, shown in Fig. 1.1, the same hits occupy less than

¹<http://www.pdb.org/>

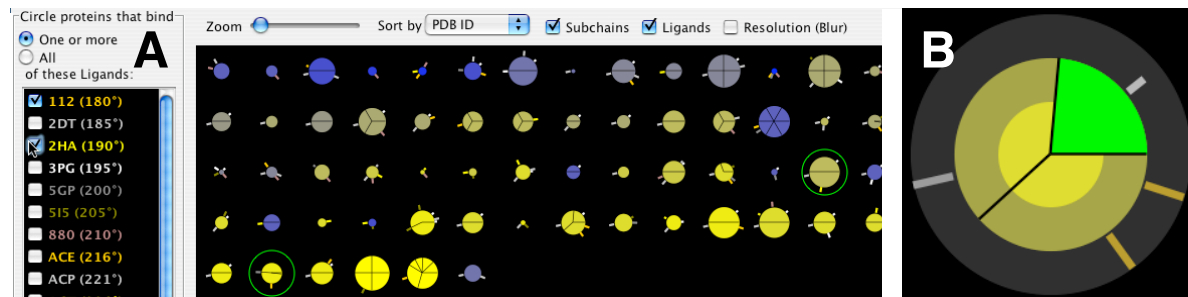


Figure 1.1: **A)** PDB structure and ligand hits in PDQVis. The dynamic query interface (left) enables users to select ligands and query for proteins that bind the same; query results are circled in green. **B)** Close-up of a structure glyph. Each glyph represents a PDB structure, its subunits (pie 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.

half a screen. After reading a brief textual description, as in the caption to Fig. 1.1, 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 require more time and effort to obtain with the PDB’s web interface, as discussed in Section 2.4.2 and in Section 3.1. In general we hypothesize that overviews can reduce the cost of knowledge from structure databases like the PDB. The “cost” of knowledge refers to factors such as time and cognitive effort (see the Glossary for more).

1.1 Contribution and organization of this thesis

The contribution of this thesis consists of design principles, software prototypes, and glyph designs for perceptually efficient overview displays of structure hits from databases like the PDB.

Above we have given an example of why our research is needed and how it might be useful. In Chapter 2 we survey the existing body of work in information visualization and bioinformatics visualization to achieve the following ends. First, to establish the need for overview displays in structural bioinformatics; in part by showing how prior visualization tools have neglected this need. Second, to gather general principles that can inform the design of effective overviews. In Chapter 3 we present these general principles. We follow in Chapter 4 with software prototypes designed according to these principles.

Chapter 2

Related Work

Users of structure databases face specific challenges. In Chapter 3 and Chapter 4 we propose glyph-based overviews as solutions to some of these challenges. We therefore begin the present chapter with an exploration of existing research in overviews, glyphs, and the perceptual and cognitive issues that they engender. Our goal is to synthesize and extend this research to create overviews of structural bioinformatics data. In Section 2.4 we explore the state of the art in bioinformatics visualization and query result visualization.

In this Chapter we also introduce the concept of *visual data mining*, which is the process of extracting useful information from a visualization. Visual data mining dovetails with our research goal, to create informative overviews of structural bioinformatics data, since the purpose of an overview is to provide *insight* into the dataset at hand. We therefore draw on the literature in visual data mining to find recommendations for overview displays.

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. In Chapter 3 we show that glyphs are a natural building block for overview visualizations in structural bioinformatics. In Chapter 4 we present glyph-based visualizations of PDB query results.

Ward [War02] provides a broad survey of glyph types and placement strategies. The glyphs we create for PDB structures and bound ligands (Fig. 1.1) combine elements of pie glyphs [WL00], whisker plots, and metroglyphs [And57]. Anderson [And57] recommends against the full-circumference whisker layout we employ, but does not substantiate his recommendation. We hypothesize that a full-circumference whisker layout results makes more efficient use of screen space and is at least as comparative as Anderson's method. User studies are needed to resolve the issue. In Section 4.1.7 we explore the use of

Retinal category	Preattentive features
Form	Line orientation Line length Line width Line collinearity Size Curvature Spatial grouping Blur Added marks Numerosity
Color	Hue Intensity
Motion	Flicker Direction of motion
Spatial position	2D position Stereoscopic depth Convex/concave shape from shading

Table 2.1: Preattentively processed retinal attributes [War04, 151-152].

statistical graphics as data summaries and filter controls. The broader topics of statistical graphics and glyphs as widgets are explored by Tufte [Tuf01] and Calder et al. [CL90], respectively.

2.1.1 Retinal attributes, preattention and pop-out

To support visual data mining we wish to create glyphs that can be quickly identified and interpreted. A first principle of glyph design is to develop a visualization schema that uses *preattentively processed* retinal attributes to encode important data. Preattentive processing precedes conscious attention. It is a theoretical mechanism to explain pop-out, or visual salience. Visuals that pop out seem to grab the attention, quickly distinguishing themselves from the rest of the visual field. (Consider how a flashing red light “demands” attention.) In general, anything processed at a rate faster than 10 msec per item is considered preattentive [War04]. A list of retinal attributes that are preattentively processed shown in Table 2.1.

It is impractical to rank preattentive features by strength, since the strength of a given feature depends upon its context [War04]. This consideration is especially relevant in dense, multi-glyph visualizations where glyphs are likely to occur in the context of numerous others. In fact the performance of preattentive attributes declines in proportion

to the number of neighboring distractors. Nevertheless, individual glyphs can be perceived, used, and analyzed even in the context of thousands of other glyphs [RAEM94]. Such multi-glyph displays also produce gestalt effects where the overall impression conveyed by clusters of glyphs supersedes the effects of any individual glyph [RAEM94]. It is worth mentioning that some retinal attributes encode quantitative variables better than others. [SHB⁺99] cites evidence that position along a scale is the most effective method of displaying a scalar quantity in 2D. This is followed, in order of decreasing effectiveness, by length, slope, area, volume, and color.

Integral-separable dimension pairs

Some pairs of preattentive features are *integrally* or holistically perceived. When integral features occur together it is difficult for the visual apparatus to tease them apart. Motion and flicker, for instance, are integrally perceived. Spatial position and color, on the other hand, are *separable*; they are perceived independently. From the integral-separable dichotomy we learn that using one or more integral dimensions in a single glyph is likely to obfuscate the underlying data. In designing effective glyphs one should therefore employ a set of mutually separable preattentive attributes. [War04, 180] provides further details on integral-separable dimension pairs and notes that the integral/separable dichotomy encompasses a continuum of effects ranging from fully integral to fully separable .

For each retinal dimension, there is also a limit to the number of different steps that can be rapidly resolved. [War04] suggests that most dimensions support about four discrete steps (two bits of information), with color supporting eight steps (three bits).

If we now step back and consider all retinal dimensions, not just the preattentive ones, we end up with the following eight: size, position, color, shape, orientation, motion, blink, and texture. Assuming four rapidly distinguishable steps per dimension, one might expect $4^8 \approx 65,536$ rapidly distinguishable glyphs. But, once we factor out integral dimensions, we are left with a much humbler number, 32 [War04]. The implication is that a carefully designed visualization schema can represent no more than 32 rapidly distinguishable alternatives. At first glance that seems like a crushing limitation for glyph-based visualizations of large databases, which may contain thousands of unique data entries; but, for the reasons outlined below, this need not be the case.

Rethinking the dimensionality of glyphs. First, we note that the visualization schema designer is free to use as many steps per retinal dimension as are *available*—let’s forget, for a moment, about how many of those dimensions are *resolvable*. On modern displays the color dimension alone provides millions of steps. Likewise for dimensions such as shape, orientation, and texture. So the number of *possible* glyphs is extremely large. The fact that only 32 of these can be rapidly distinguished is not a total loss. For one thing, these 32 alternatives may help to segment large sets of glyphs into as many disjoint subsets. It is reasonable to assume that these subsets would be *logical*

clusters of related data points. Users looking for fine-grained information, beyond the one-of-32 alias for a given glyph, could study the glyphs in detail, on time scales much longer than the 10 msec needed for preattentive processing. At these longer time scales the resolving power of the eye goes far beyond four steps per dimension and the number of perceptually distinct glyphs approaches the number of logically distinct ones. An alternative solution is to provide *compound glyphs*. Instead of compressing all of the data for a single concept into one glyph, compound glyphs distribute the data across a family of adjacent or superposed glyphs. Since compound glyphs consist of multiple sub-glyphs they may provide more than 32 rapidly distinguishable flavors. We will revisit compound glyphs in Chapter 4 where they are used to represent structural and chemical properties of proteins.

2.1.2 Perceptual color sequences



Figure 2.1: Perceptual color sequences.

Ware [War04] notes that quantitative (or ordinal) variables are easiest to interpret if they are encoded with a *perceptual* color sequence. Elements of a perceptual color sequence can be ordered by the brain without additional semantic information. Figure 2.1 shows examples of perceptual color sequences. We make use of the yellow-blue and saturation sequences in Chapter 4 to encode continuous and ordinal variables from protein databases.

2.2 Working memory and cognitive load

Any information retained longer than 300 msec must first pass through working memory [War04]. For our purposes, the key feature of working memory is its limited capacity. Research indicates that working memory can hold somewhere between four and seven chunks of information [Wik06c]. These chunks may be words, letters, or other symbols. When the amount of incoming data exceeds the capacity of working memory, *cognitive overload* occurs. Cognitive overload may well be an issue in mega-glyph displays, where users are faced with hundreds or thousands of glyphs at a time. We therefore turn our attention to cognitive load and its mitigation.

Meyer and Moreno [MM03] identify *split attention* as a source of cognitive load. Split attention occurs when the user is required to split his attention between remote targets. (A user simultaneously viewing a movie in one window and reading a webpage in another has split his attention.) Split attention taxes working memory by requiring the user to devote cognitive capacity to *finding* the targets as he switches rapidly between them. Not surprisingly, [MM03] advises *integrated presentation* to reduce cognitive load. Integrated presentation places multiple targets—text and images, for instance—in close proximity for ease of consumption. Tufte makes similar recommendations in [Tuf90] and [Tuf06a]. The former contains a chapter on *small multiples*, series of small, related images designed to convey trends and changes in the data. In the spirit of integrated presentation, Tufte emphasizes that “comparisons must be enforced within the scope of the eyespan” [Tuf90, 76]. The implications for visual data mining are clear, and support our emphasis on dense visualizations in which information for many data entities is concentrated in a small area. More recently, [Tuf06a] espouses the integral presentation of words, numbers, and images (Fig. 2.2). This is a principle motivation behind sparklines, which can be placed in and among text.

2.2.1 Readable, meaningful glyphs

As Shaw et. al observe, “one of the most difficult problems in glyph visualization is the design of meaningful glyphs,” [SHB⁺99]. In addition to the perceptual guidelines outlined above, we identify three approaches to creating meaningful glyphs:

user specification [RAEM94] provides a flexible framework with which users can completely specify the glyph-to-data bindings (i.e. the visualization schema).

automatic generation Systems that automatically generate icons¹ are propounded in [PPWS95] and [SABG⁺05]. [PPWS95] outlines a model to abstract and visualize data and applies the model to tensors of scientific data. [SABG⁺05] presents a system that automatically mines the name, location and contents of a file to produce meaningful desktop icons or “semanticons.”

¹Here, as in [PPWS95], the terms “icon” and “glyph” are roughly equivalent .

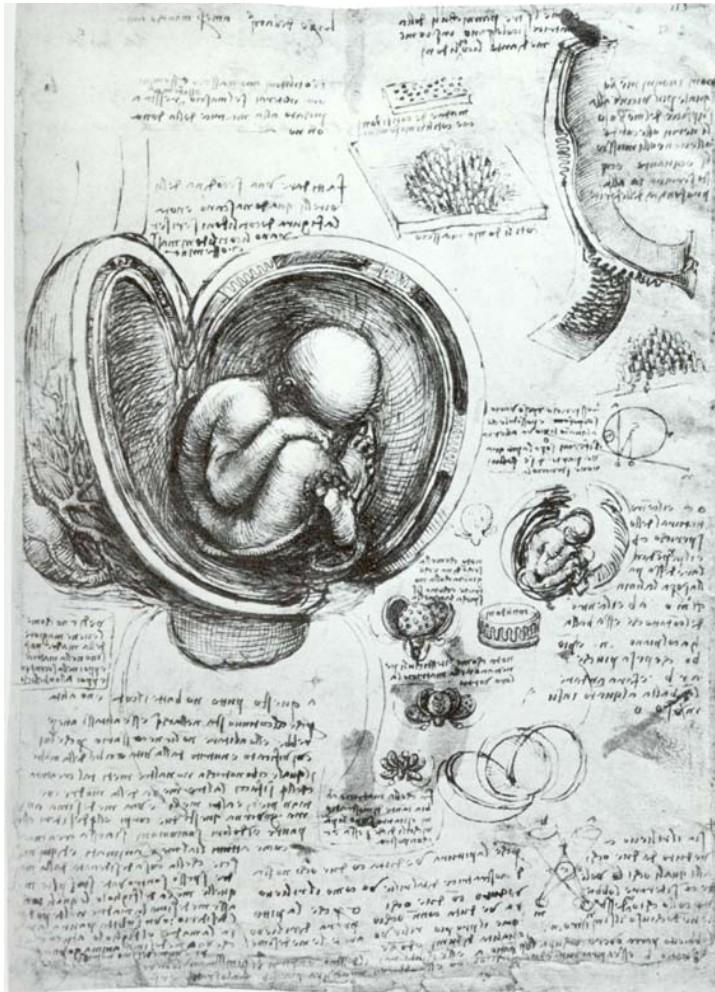


Figure 2.2: *Studies of Embryos* by Lenardo DaVinci [Dav10]. Notice the tight integration of word and image, if not number as well. [Tuf06a] contains further, striking examples of such integrative data art.

data relation [Sii05] explores the effects of data-relatedness on interactive glyphs and shows that users prefer, and perform better with, data-related glyphs (e.g. use car glyphs when exploring automobile data). This technique can be viewed as a handrolled version of *automatic generation* since the design of the glyph is influenced by the semantics of the data.

Norman [Nor02] defines a *natural mapping* as a clear relation between user intentions and possible actions, between possible actions and their effect on a system. Norman argues that natural mappings foster usability. We can extend this idea to glyphs and say that a glyph is *natural* when its retinal attributes have a clear relationship to the underlying data.

We employ data relation to design glyphs for protein structures in Chapter 4.

2.3 Overview and visual data mining

The so-called *visual information-seeking mantra* was popularized by Shneiderman [Shn96] as a useful starting point for advanced graphical interfaces. The mantra reads as follows: “Overview first, zoom and filter, and then details-on-demand.” It goes without saying that the overview operation is central to visual data mining, where a prime goal is the identification of trends and outliers. As such, infovis systems commonly provide *overview+detail*, in which detailed information for specific items of interest is displayed alongside an overview of the entire data set. A related technique, *focus+context*, provides detailed information about a subset of items under focus and simultaneously conveys the broader context of these items. Focus+context differs from overview+detail in that focus and context are understood to be integrally presented, whereas overview and detail are, in general, presented separately (e.g. in two separate windows). In Section 2.2 we identified integral presentation as a means to reduce cognitive load. For this reason, when circumstances permit, focus+context may be preferable to overview+detail.

Both overview+detail and focus+context imply the concept of *zoom*. Zoom entails not only changes in the visible dimensions of an object, but changes to its *logical* or *semantic* structure as well. Logical zoom is simply a mechanism for revealing or hiding details as the depth of the zoom changes. At the lowest level of zoom, only broad, abstract features of the data are visible. At the highest level of zoom all available details are visible for the item(s) in focus.

Linking and brushing is the use of highlighting to identify common elements across multiple views. Highlighting may be accomplished with color, contours, or other methods.

We apply the aforementioned techniques in our research software (Chapter 4). We now turn our attention to visual data mining, a subarea of information visualization which can provide heuristics for the design of information-rich overview displays.

2.3.1 Visual data mining

At the core of visual exploration are the search and browse operations. In search, the user proceeds from a query to the corresponding results. Shneiderman [Sch94] pioneered an exploration technique called *dynamic queries*. Dynamic queries rapidly update search results in response to user requests made through graphical widgets. Browsing, as distinguished from search, is an ad hoc process in which the user has no preset query in mind but is nevertheless free to flag or pursue items of interest. Keim [Kei02] identified the agnostic nature of browsing as a key advantage of visual data mining over automatic data mining (a.k.a. “machine learning”). That is to say, browsing remains applicable in the absence of *a priori* knowledge about the data. Keim further argues that visual data exploration is robust to inhomogenous or noisy data and scales gracefully to large data sets. In light of these advantages, and in light of the human capacity for creativity and domain knowledge, Keim advocates for the inclusion of human beings in the data mining process (enter visual data mining). He concludes that visual data mining is likely to be faster and, in many cases, produce better results than automatic data mining alone. Keim’s advocacy of agnosticism and visual data mining are echoed in [TMWJK04]. The aforementioned conclusions suggest that visual overviews can help scientists to understand and organize large collections (of bioinformatics data).

Shneiderman [Shn02] notes that information visualization and automatic data mining have developed, by and large, in isolation from one another. Shneiderman favors the integration of the two as a path to deeper understanding of data. He makes the following three recommendations for visual data mining systems:

1. “allow users to specify what they are seeking and what they find interesting”
2. support exchange, discussion, and other forms of social networking over the data
3. respect human responsibility by providing tools that are “comprehensible, predictable, and controllable.”

Though automatic data mining is beyond the scope of our research, some brief comments are in order. [KLS00] and [MDH⁺03] outline automated preprocessing techniques for “unstructured” data. They apply algorithmic measures of information, such as entropy and self-organizing maps, to create self-structured visualizations. Such techniques may facilitate the discovery of patterns in the data, but one must simultaneously guard against imposing interpretations upon the data [Tuf06b]. (In some sense all automated data mining techniques impose their own inductive bias upon the data.)

2.4 Visualization tools

In this section we give a brief overview of existing research tools for glyph-based information visualization, database exploration, and structural bioinformatics visualization.

In each of these areas we examine a handful of leading tools that are germane to our research.

2.4.1 Tools for glyphs, databases and query results

Glyphmaker [RAEM94] allows users to create custom visualizations by specifying the visuals-to-data mapping or “visualization schema.” Polaris [STH02] 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). Our research suggests that complex, compound glyphs are useful for visualizing bioinformatics data (Chapter 4). Kreuzler et al. [KLS00] 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.

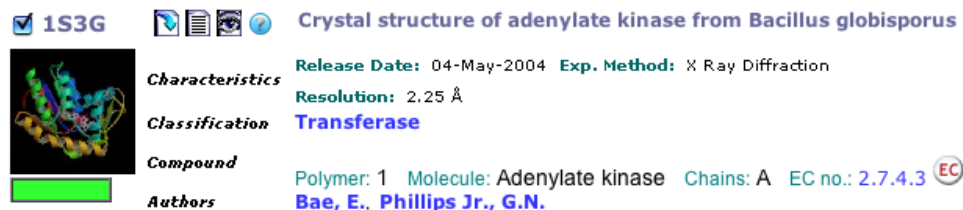
Sparkler [HHP⁺01] is relevant to our work since it was designed to visualize the results of multiple database queries via a data-driven glyph layout wherein glyphs are organized according to relevance and parent query. Sparkler does not expose the data content of individual results. Our approach differs from Sparkler in that we focus on content-intensive overviews of a single result set. In Chapter 5 we explore the possibility of integrating the Sparkler visualization method with one of our own.

Table-based displays, proposed in Table Lens [RC94], are a natural method for organizing database visualizations. Table Lens also provides a focus+context mechanism for enhancing details. In Chapter 4 we present a table-based overview with an alternative focus+context technique.

2.4.2 Bioinformatics visualization tools

Structural biologists make frequent use of visual abstractions to summarize data. This fact is reflected in biochemical structure diagrams and, since the advent of software, a rich and varied use of visualization tools to support problem-solving.

In general, bioinformatics visualization tools have focused on either structure or network visualization. That said, in neither case has generic, large-scale overviews been tackled. Chapter 7 of [BW03] provides a survey of structure visualization tools. Such tools are widely used and offer sophisticated features, but primarily support the study of individual structures. Network visualization tools like Osprey [BST03] 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.



1S3G Crystal structure of adenylate kinase from *Bacillus globisporus*

Characteristics Release Date: 04-May-2004 Exp. Method: X Ray Diffraction
Resolution: 2.25 Å

Classification Transferase

Compound Polymer: 1 Molecule: Adenylate kinase Chains: A EC no.: 2.7.4.3

Authors Bae, E., Phillips Jr., G.N.

Figure 2.3: A structure hit from a PDB web search.

The PDB's web interface

Online queries to the PDB return structure hits in the format shown in Fig. 2.3. 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 (Fig. 2.4). Collage view is a uniform tiling of thumbnail images, one for each hit, as shown in Fig. 2.4. This is a step in the direction of overview, but it fails to be synoptic and comparative (in Section 3.1 we will argue that these are desirable characteristics of overviews). PDB collages are not synoptic since they expose only one facet of the dataset: small, fixed-perspective images of each structures three-dimensional shape. Such images can indicate the rough shape of molecules, but they do not afford accurate comparison due to problems of perspective (Fig. 2.5).

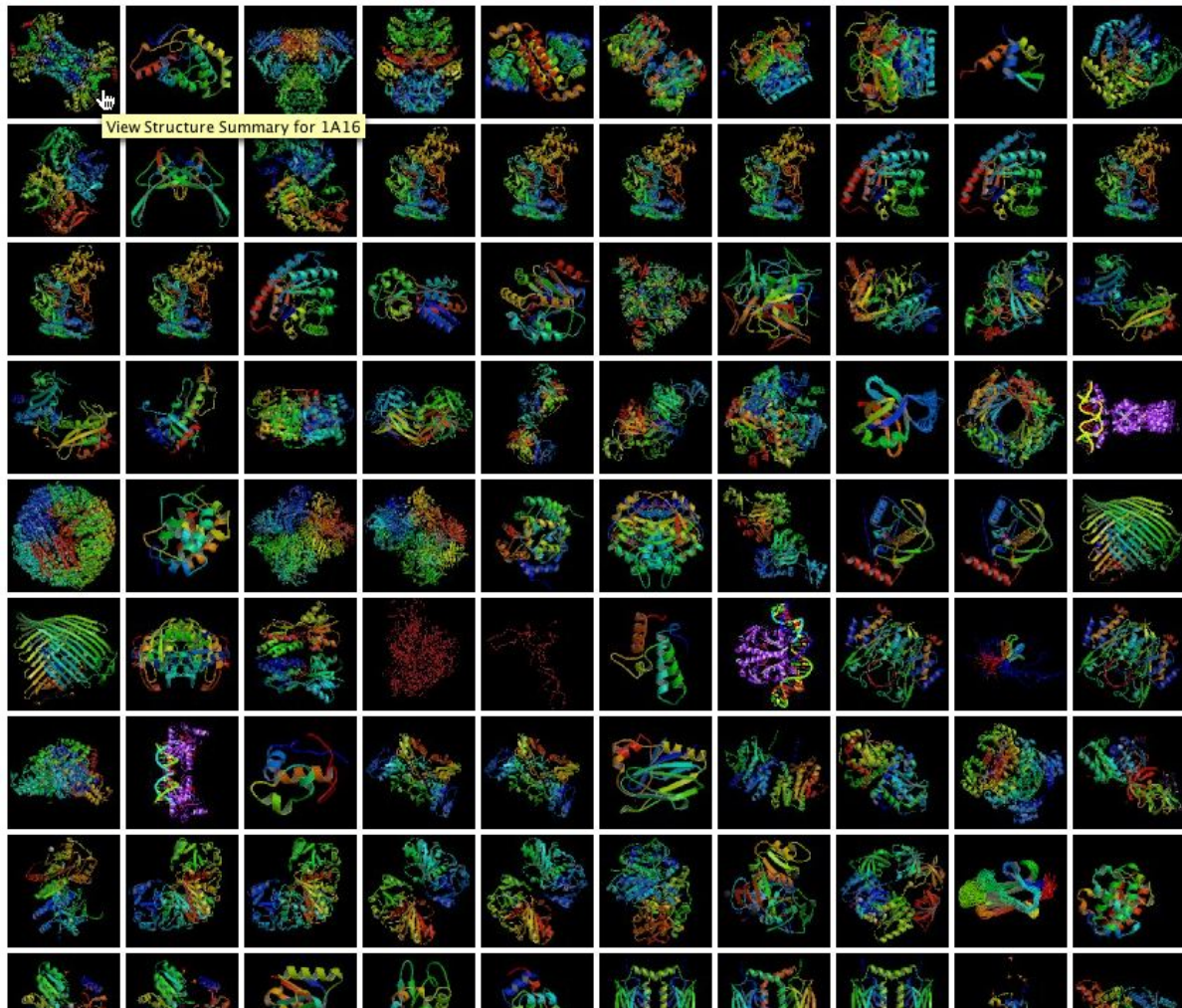


Figure 2.4: Collage View from the PDB's website.

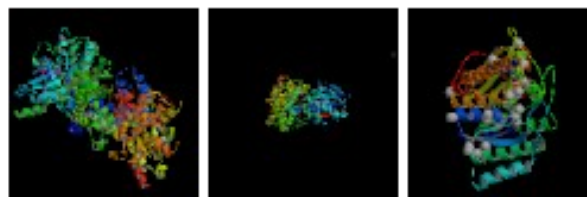


Figure 2.5: Thumbnail images from a PDB collage. The images—which depict structures 2ACX, 2AJ4 and 2AJP in left-to-right order—belie the relative molecular weights of the corresponding structures: 66kDa, 60kDa, and 36kDa, respectively.

Chapter 3

Functional Goals for Large-scale Overviews

In Chapter 3 we established the importance of overview. In Section 2.4 we argued that current tools for structural bioinformatics do not provide adequate overviews. We propose to supplement current tools with new ones, explicitly designed for 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 this section we present functional goals for large-scale overviews. We follow in Chapter 4 with examples of how we have applied these principles to create specialized displays for the visual overview of PDB query results.

3.1 Four goals for overviews

A critique of the PDB hit format (Fig. 2.3) 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. It consists primarily of descriptive text, which requires more time and effort to interpret than preattentive visual attributes or glyphs. Finally, 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. [Shn96]).

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 4.1.7 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 informative 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. 2.4, are not comparative: without a common frame of reference, similar molecules appear different, and vice versa. In designing overview displays, 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 2.4). To some extent, an emphasis on inter-data comparison, as opposed to precise data values, is a natural consequence of large-scale overview, which trades precision for space. We revisit these issues in Section 4.1.5.’
3. Overview displays should be **perceptually efficient**, as discussed in Chapter 3. The goal is to make visual information seeking as quick and easy as possible. Put another way, the goal is to reduce the cost of knowledge from the underlying dataset.
4. Overview displays should support **zoom and filter** controls that enable users to subset, sort, and expose details in the data. In light of integrated presentation (Section 2.2), focus+context is preferable to other overview+detail methods. Zoom and filter *per se* are beyond the scope of this paper; [CMS99] provides a survey.

3.2 Glyphs and overview

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.

Chapter 4

Research Software

We have created two software prototypes for glyph-based overviews of PDB query results: PDQVis (Fig. 1.1) and BioSpark (Fig. 4.5). 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 3.1, Section 4.1 proposes glyphs for data types common to structural bioinformatics. Section 4.2 discusses the design of compound glyphs and the rationale behind the glyph design we chose for PDQVis.

4.1 Glyphs by data type

4.1.1 Images

In the upper left of Figure 2.3 are three icons. Moving from left to right, these icons are buttons to download the PDB file, view the PDB file for the result, and visualize the result in three dimensions. These functions may be vital, but probably do not belong in a data-dense visualization for the simple reason that they are common to every result in the data set and provide no distinguishing information. Such functionality is better exposed in contextual or pull-down menus. If a menu is unacceptable, said icons could be readily added to BioSpark. Since this addition would require little if any modification to the icons, we will not discuss these icons further.

4.1.2 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 since reading and interpreting descriptions 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.

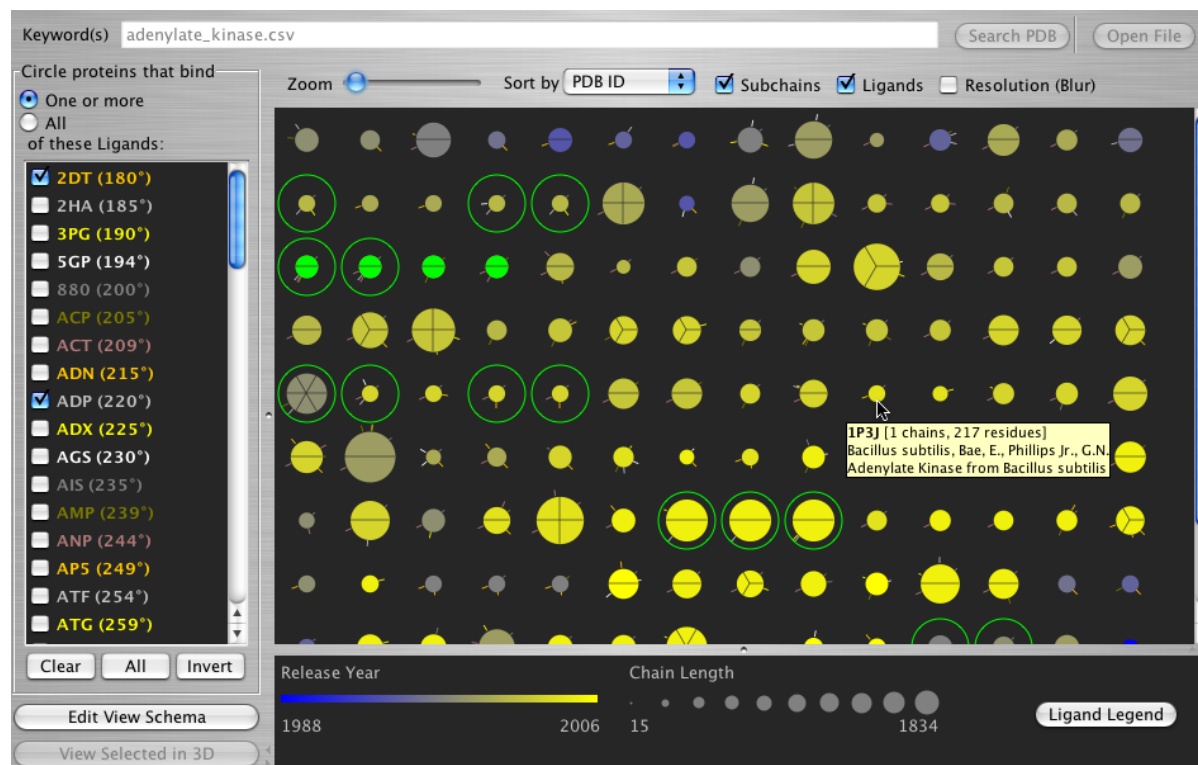


Figure 4.1: Protein structures and bound ligands displayed in PDQVis. Each glyph represents a PDB structure, its subunits (pie slices), number of residues (size), release date (blue is older, yellow newer), and bound ligands (radial whiskers); resolution (blur) is not shown. If the user selects a subchain, it is brushed in green. For example, the two green slices of structure 20OX (lower right) represent identical subunits. The dynamic query interface (left) enables users to select ligands and query for proteins that bind the same; query results are circled in green.

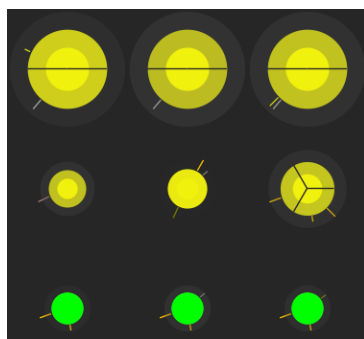
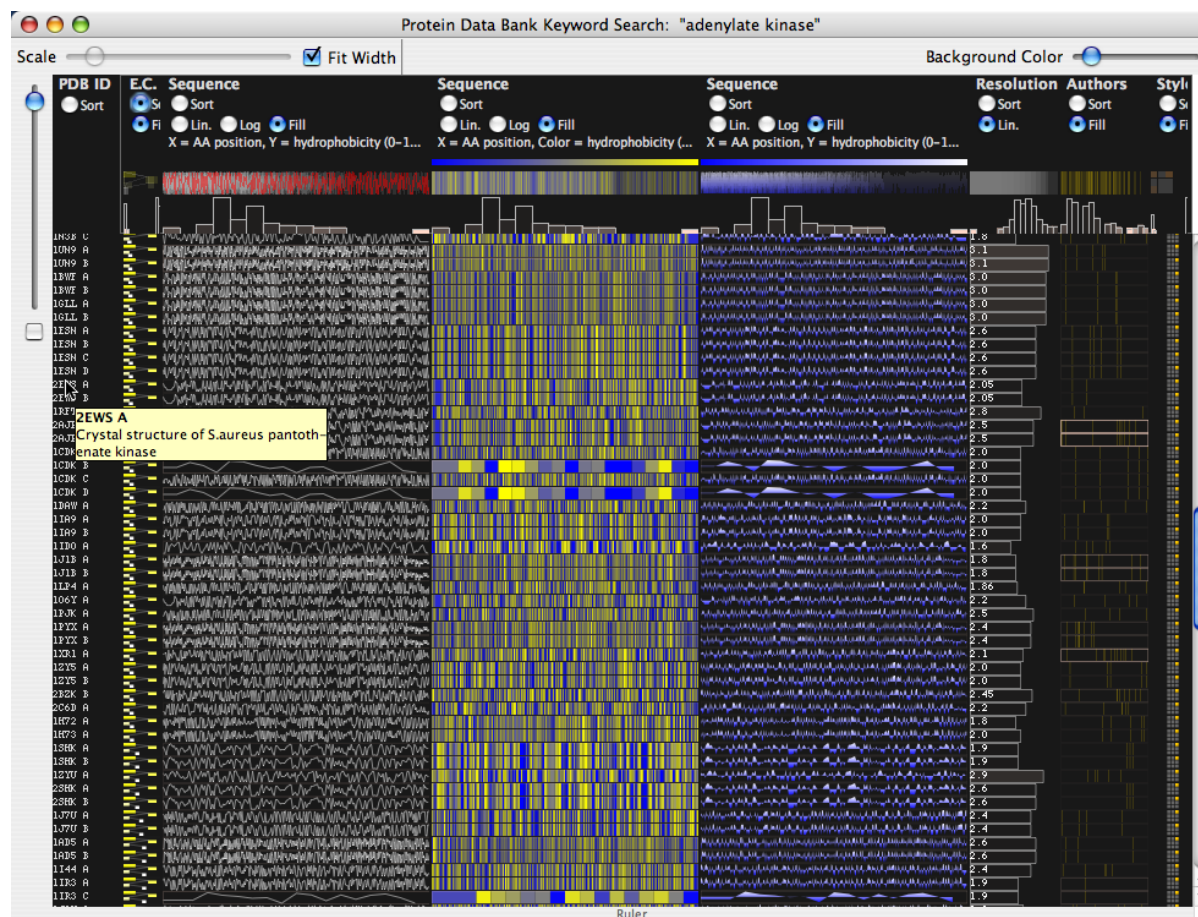


Figure 4.2: Close-up view of glyphs in PDQVis.



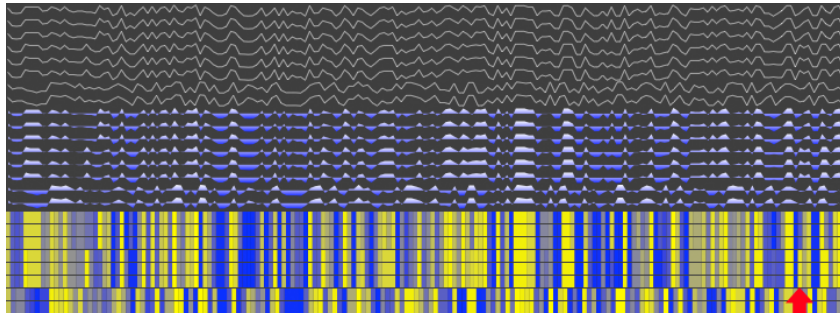


Figure 4.6: Sparklines representing the primary structure of protein subchains. In top-to-bottom order the subchains belong to the following PDB structures: 1KHT (3 subchains), 1K19 (3 subchains), and 1KOF (2 subchains). This pattern repeats for each of the three visualization styles shown above for a total of 24 sparklines. (The red arrow shown on the right is for reference and is not part of the visualization.)

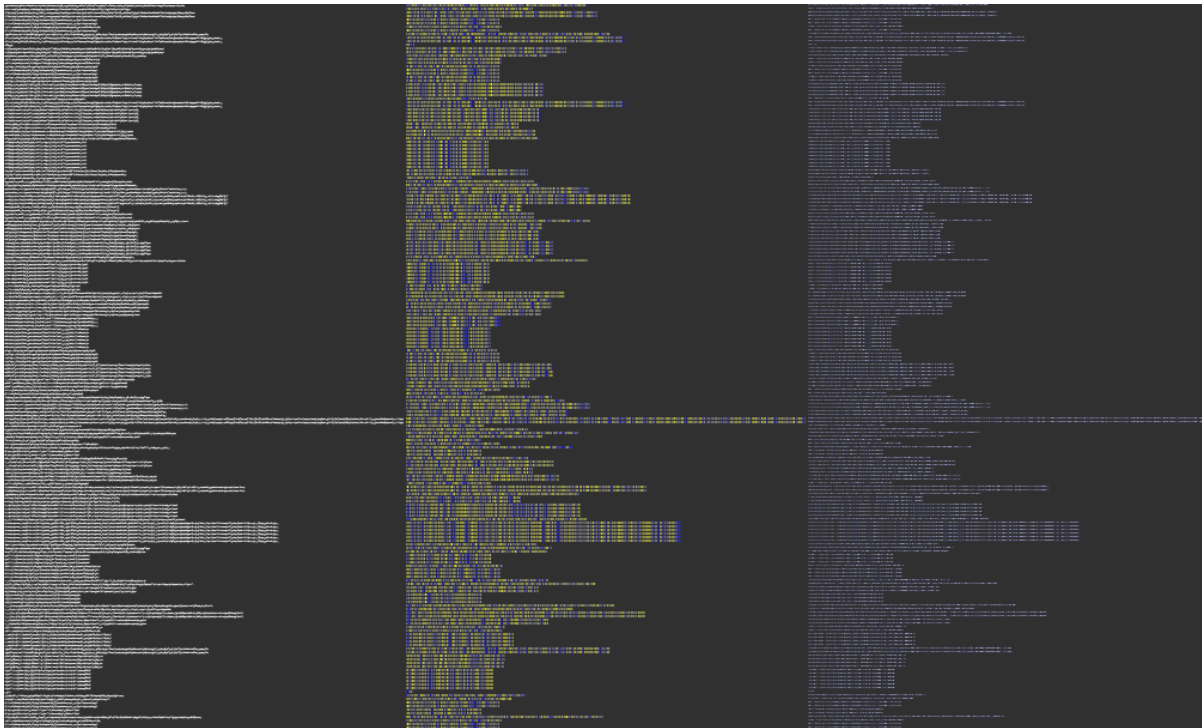


Figure 4.7: 606 sparklines representing the primary structure of proteins. From left to right the sparklines are rendered using contour only, height mapping, and filled contours.

4.1.3 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. 4.5, 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 [SHB⁺99]. Depth of field (blur) can also be used to encode scalars (e.g. structure resolution in Fig. 1.1a).

4.1.4 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 functions¹.

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 in Fig. 4.5). If the common axis is radial, and each glyph has its own axis, the visualization may look like Fig. 1.1. 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 1.1. The legend shown in Figure 4.3 provides users with detailed information about the name, color, and radial position of each ligand.

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. 1.1 indicates the molecular weight of the corresponding ligand². 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. 4.1, wherein users can quickly infer that three of the first four structures in the top row bind the same ligand (depicted by the yellow whisker near the nine o’clock position).

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

²In reality the length of the whisker is determined by the number of letters in the ligand’s abbreviation code. This is a sloppy heuristic based on the assumption that lone atoms tend to have one- or two-letter codes whereas molecules tend to have three-letter codes (and that shorter abbreviations therefore correlate to lower molecular weight). The important point is that biologically relevant information can be encoded in the length of the whisker.

4.1.5 Classification trees

The Enzyme Nomenclature tree [NC-92] provides functional classifications for proteins via Enzyme Classification (E.C.) numbers. An E.C. number is of the form $a_0.a_1 \dots a_n$ 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 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. 4.8. Given that size, position, color, and line orientation are preattentively processed, users can quickly compare and contrast scribble trees.

We note that scribble trees emphasize data comparison over data content (Section 3.1). For example, it is easy to ascertain that all of the E.C. numbers depicted in Fig. 4.8 share their top two levels, but it is relatively difficult to determine the precise E.C. number for a given glyph.

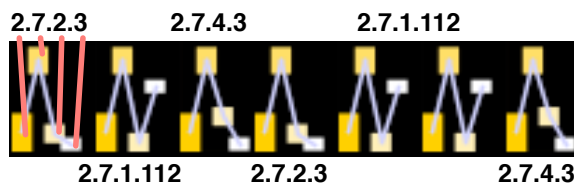


Figure 4.8: Scribble trees depicting Enzyme Classification numbers.

4.1.6 Protein structure

Sparklines are a natural method for summarizing primary and even secondary structure. The former can be visualized with hydrophobicity plots. The first three columns of Fig. 4.5 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 example, the amino acid position indicated by the red arrow in Fig. 4.6, or the yellow and blue primary structure sparklines at the bottom of Fig. 4.5 which show three identical chains plus one homologue which appears to have diverged via substitution.) Sparklines can be downsampled to produce data dense visualizations, as shown in Fig. 4.7. As discussed in Section 3.1, the PDB’s current result format does not lend itself to such data-dense displays.

Since sparklines for primary and secondary structure can be aligned, with well-known algorithms, for easy comparison and since they can be rendered in 2D without occlusion, they circumvent some drawbacks of 3D structure visualization (Section 2.4.2 and Section 3.1) but convey some of the same information.

One method for visualizing tertiary structure at the overview level is with scribble trees of a structural classification like SCOP [BW03]. We hypothesize that such abstract representations of 3D structure are, at small sizes, easier to interpret than actual 3D visualizations like PDB thumbnails (Fig. 2.4).



Figure 4.9: Style boxes summarizing the primary structure of proteins.

At low levels of zoom, 2D glyphs can be condensed into style boxes [Mor02]. Each style box has an x - and y -axis divided 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. 4.9 summarize the number of residues (horizontal bins) and average hydrophobicity (vertical bins) of 28 protein chains. Glancing over Fig. 4.9 reveals that every protein shown is of average length and that the majority of proteins have, comparatively speaking, a low mean hydrophobicity.

4.1.7 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. 4.5). 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. 4.5.

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.

4.2 Complex and compound glyph design

The aforementioned glyphs may be combined into compound glyphs to achieve higher data density. Designers of multi-attribute glyphs should consider constraints such as the following: integral-separable dimension pairs, natural mappings, and the perceptual efficiency of the encoding. (The former two considerations are mentioned in Section 2.1.

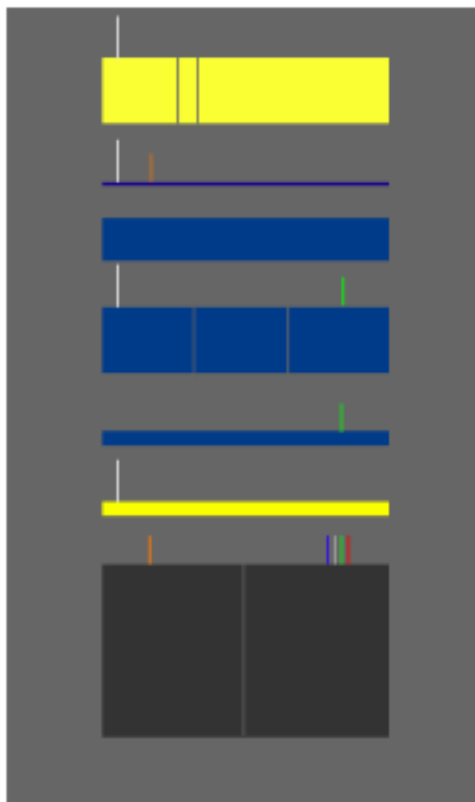


Figure 4.10: An alternative glyph design to that of PDQVis (Fig. 1.1). Rectangles, rather than circles are used to represent protein chains. The ligand auras are layed out horizonally on a shared scale.



Figure 4.11: A ligand aura that used shape-based encoding to display information about one ligand, or a group of ligands.

We give an example of the latter consideration in Section 4.1.3.) The compound glyph in Fig. 1.1 combines six data attributes into a single graphic, and was created by applying the aforementioned constraints. In Fig. 1.1 the data attributes of chain length, bound ligands, ligand weight, number and size of subunits, resolution and release date are bound, respectively, to the following retinal attributes: size, line orientation, line length, slice numerosity and size, blur³, and color. We note that these attributes are mutually separable, preattentive and natural (i.e. chain length maps to size, resolution maps to blur). As discussed in Chapter 2, these factors make for efficient visual consumption. The glyph-layout is data-driven which means that the two-dimensional position of the glyph encodes yet another attribute, such as source organism (i.e. the glyphs can be spatially sorted according to data attributes). We believe that the compound glyph design outlined in this paragraph is efficient, but it is by no means the only possible design.

A look at the preattentive attribute list in Table 2.1.1 shows that a handful of preattentive attributes remain unused in Fig. 1.1. Namely, under “form,” curvature and line width; under “motion,” flicker and direction of motion; under “spatial position,” convex/concave shape from shading. We discuss each of these briefly. Line width might have been used to encode a ligand attribute, such as solubility, with the proviso that making ligand whiskers very thick would limit the number of whiskers that could be packed onto a small glyph. (In practice, datasets contain several hundred ligands. That said, whisker layouts could be improved by using an optimization algorithm to determine the layout that results in the least crowding.) Curvature and convex/concave shape from shading might have been used to display further data attributes, as explored in [SHB⁺99], or to provide logical zoom for bound ligands, as shown in Fig. 4.11. The reason that bound ligands might benefit from logical zoom is that, when PDQVis glyphs are displayed at small sizes or the number of bound ligands is large, individual ligand whiskers become difficult to discern. Therefore, several similar ligands, or sets of ligands could be combined into a summarizing shape (Fig. 4.11). We decided to forego motion-based retinal attributes under the assumption that they would be distracting and would interfere with the perception of other retinal attributes.

Fig. 4.10 and Fig. 4.11 show alternatives to the compound glyph from Fig. 1.1. They use squares instead of circles to represent proteins and their subchains, and show alternative layouts for the ligand auras. Some potential advantages of the design in Fig. 4.10 are the following. First, size (chain length) is now a function of one retinal attribute: height. It may be easier for users to compare rectangle heights than circle areas (as in PDQVis). Second, for massive datasets, rendering millions of rectangles may be quicker than rendering the same number of circles or quadrics. Third, the ligand whiskers all appear on the same side of the protein glyph. It may be easier to compare and contrast

³Our method of producing blur is to superimpose a halo which is the same color as the visualization background and is transparent in direct proportion to the desired blur. This method could be improved using, for instance, a gaussian filter on the glyph raster.

whiskers with this layout, versus the radial one used in PDQVis, since there is a single shared, positional scale for every glyph in a column (as opposed to a separate radial scale for each glyph in PDQVis). As discussed above, the design in Fig. 4.11 makes it possible to provide logical zooming for ligand auras. It also makes it possible to express more information about each ligand, (i.e. now each ligand is represented by a higher-dimensional mark and not just a line); but this is subject to space constraints.

In the ideal case an optimal glyph design would be chosen from among several candidates through user studies.

Chapter 5

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.

User studies are needed to assess the 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-savvy applications like PDQVis and BioSpark. A related question is: at which levels of zoom, and for which types of data, are table-based displays preferable to other glyph layout strategies (e.g. PDQVis).

A third, more general question to be settled by user studies was hinted at in [WAM01] and has been foregrounded by our work on indicator function glyphs: How does axis orientation (e.g. radial or horizontal) affect the interpretation and comparison of time series and indicator functions?

The Sparkler [HHP⁺01] method for visualizing query results, discussed in Section 2.4, could be hybridized with PDQVis as follows. Replace Sparkler’s document glyphs with structure glyphs from PDQVis, while retaining Sparkler’s multi-query, relevance-based layout. Said hybrid could enable users to interpret and utilize data, such as query results from the PDB, more efficiently than either visualization in isolation.

Glossary

The terms of the glossary are organized in order from simple to complex.

Data are raw symbols, devoid of meaning [BCM04].

Data density or resolution is a measure of how much data is visible relative to the amount of space used to display it.

The Protein Data Bank (PDB) is an online repository of structural data for proteins and nucleic acids [KXdlC⁺06]. We sometimes refer to individual entries in the PDB as **structures**. The PDB is a key resource for biologists worldwide.

Information is data that has become meaningful through semantic associations [BCM04].

Data mining is the extraction of useful information from data [Shn02]. Data mining can be performed by human or machine. Visual data mining, as we will use the term, is unique to the former. As with traditional forms of mining, data mining entails *exploration*.

Knowledge, understanding, and wisdom are higher levels of cognition that build upon information [BCM04]. The science of supporting these cognitive functions through visualization is closer to *visual analytics* than information visualization. We touch upon knowledge, understanding and wisdom only peripherally.

Knowledge crystallization is the process by which a person gathers data, makes sense of it, and then arranges it for communication or action [CMS99, 10].

Retinal attributes are visual attributes of graphical objects. Examples include position, size, color, orientation, motion and texture.

Glyphs are graphical objects whose retinal attributes are determined by data.

Sparklines are “intense, simple, word-sized graphics” [Tuf06a]. In principle, a sparkline is a simple glyph. In practice, as established by Tufte, sparklines are distinguished by their integral placement within larger blocks of text, compact size, or tendency to depict time series data.

Information design is the art and science of preparing data for efficient consumption.

Information visualization (infovis) is “the use of computer-supported, interactive, visual representations of abstract data to amplify cognition” [CMS99]. Information visualization pivots on the human visual system as a parallel, high-bandwidth channel from computer to human [NH06]. This channel can be leveraged to acquire and process information.

Informally, information visualization is “using vision to think” [CMS99].

Scientific visualization (scivis) is the use of “interactive visual representations of *scientific* data, typically *physically based*, to amplify cognition” [CMS99].

A visualization schema or transfer function is a map from data attributes to retinal attributes. The development of an effective visualization schema is a core challenge in information visualization where the data are typically abstract. Unlike the physical data used in scientific visualization, abstract data do not imply a natural transfer to spatial position and other retinal attributes.

Working memory is the set of cognitive structures used to temporarily store and manipulate information. The precise theoretical and biological components of working memory are open to debate, but it is widely accepted that working memory has a small, finite capacity [Wik06c].

Cognitive load is a measure of how much of the total working memory is in use [Wik06a].

Information Foraging Theory “holds that in seeking information, people act adaptively, attempting to arrange their activities so as to increase the amount of information gained per unit cost consistent with the tools and information they have at hand,” [CMS99, 579].

A Cost of Knowledge Characteristic Function measures the amount of information available as a function of cost, typically measured in time [CMS99].

A knowledge crystallization task is one in which a person gathers data, makes sense of it, and then expresses the result [CMS99].

Bibliography

- [And57] Edgar Anderson. A semigraphical method for the analysis of complex problems. *PNAS*, 43:923–927, 1957.
- [BCM04] Gene Bellinger, Durval Castro, and Anthony Mills. Data, information, knowledge, and wisdom. <http://www.systems-thinking.org/dikw/dikw.htm>, 2004. [Online, accessed 2-Dec-2006].
- [BST03] Bobby-Joe Breitzkreutz, Chris Stark, and Mike Tyers. Osprey: a network visualization system. *Genome Biology*, 4(3), 2003.
- [BW03] Philip E. Bourne and Helge Weissig, editors. *Structural Bioinformatics*, chapter 7, 9, 12, 13. Wiley-Liss, 2003.
- [CL90] Paul R. Calder and Mark A. Linton. Glyphs: flyweight objects for user interfaces. In *Proc. ACM SIGGRAPH UIST*, pages 92–101, 1990.
- [CMS99] Stuart K. Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
- [Dav10] Leonardo Davinci. Studies of embryos. http://www.theartgallery.com.au/ArtEducation/greatartists/DaVinci/14_Studies_of_Embryos/index.html, ca. 1510. Original in Royal Library, Windsor Castle; [Online; accessed 20-December-2006].
- [Haw06] David R. Hawkins. *Discovery of the Presence of God: Devotional Nonduality*. Veritas Publishing, 2006.
- [HHP⁺01] Susan Havre, Elizabeth Hetzler, Ken Perrine, Elizabeth Jurrus, and Nancy Miller. Interactive visualization of multiple query results. In *Proc. IEEE InfoVis*, page 105, 2001.
- [Kei02] Daniel A. Keim. Information visualization and visual data mining. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):1–8, 2002.

- [KLS00] Matthias Kreuzeler, Norma Lopez, and Heidrun Schumann. A scalable framework for information visualization. In *Proc. IEEE InfoVis*, pages 27–36, 2000.
- [KXdlC⁺06] A. Kouranov, L. Xie, J. de la Cruz, L. Chen, J. Westbrook, P. E. Bourne, and H. M. Berman. The RCSB PDB information portal for structural genomics. *Nucleic Acids Res*, 34(Database issue), January 2006.
- [MDH⁺03] Alan MacEachren, Xiping Dai, Frank Hardisty, Diansheng Guo, and Gene Lengerich. Exploring high-d spaces with multiform matrices and small multiples. *infovis*, 00:5, 2003.
- [MM03] Richard E. Meyer and Roxana Moreno. Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, 38(1):43–52, 2003.
- [Mor02] Morningstar. Fact sheet: The new morningstar style box methodology. http://news.morningstar.com/pdfs/FactSheet_StyleBox_Final.pdf, 2002. [Online; accessed 16-Feb-2007].
- [NC-92] NC-IUBMB. Enzyme nomenclature. <http://www.chem.qmul.ac.uk/iubmb/enzyme/>, 1992. [Online, accessed 28-Jan-2007].
- [NH06] Matej Novotny and Helwig Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.
- [Nor02] Donald A. Norman. *The Design of Everyday Things*. Basic Books, September 2002.
- [PPWS95] Frits H. Post, Frank J. Post, Theo Van Walsum, and Deborah Silver. Iconic techniques for feature visualization. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 288, Washington, DC, USA, 1995. IEEE Computer Society.
- [RAEM94] William Ribarsky, Eric Ayers, John Eble, and Sougata Mukherjea. Glyph-maker: Creating customized visualizations of complex data. *Computer*, 27(7):57–64, 1994.
- [RC94] Ramana Rao and Stuart K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proc. ACM CHI*, 1994.
- [RCS07] RCSB PDB. PDB Newsletter. <ftp://ftp.rcsb.org/pub/pdb/doc/newsletters/rcsb>, Winter 2007. [Online; accessed Feb-2007].

- [Rob99] Andrew Roberts. Getting to grips with latex — tables, 1999. [Online; accessed 20-December-2006].
- [SABG⁺05] Vidya Setlur, Conrad Albrecht-Buehler, Amy Gooch, Sam Rossoff, and Bruce Gooch. Semanticons: Visual metaphors as file icons. *Computer Graphics Forum*, 2005.
- [Sch94] Ben Schneiderman. Dynamic queries for visual information seeking. *IEEE Softw.*, 11(6):70–77, November 1994.
- [SHB⁺99] Christopher D. Shaw, James A. Hall, Christine Blahut, David S. Ebert, and D. Aaron Roberts. Using shape to visualize multivariate data. In *Workshop on New Paradigms in Information Visualization and Manipulation*, pages 17–20, 1999.
- [Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. on Visual Languages*, 1996.
- [Shn02] Ben Shneiderman. Inventing discovery tools: combining information visualization with data mining. *Information Visualization*, 1(1):5–12, 2002.
- [Sii05] Harri Siirtola. The effect of data-relatedness in interactive glyphs. In *Proc. of the 9th International Conf. on Information Visualization*, pages 869–876, 2005.
- [SP05] Ben Schneiderman and Catherine Plaisant. *Designing the User Interface*. Addison Wesley, 4th edition, 2005.
- [STH02] Chris Stolte, Diane Tang, and Pat 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.
- [TMWJK04] Soon Tee Teoh, Kwan-Liu Ma, Soon Felix Wu, and T. J. Jankun-Kelly. Detecting flaws and intruders with visual data analysis. *IEEE Comput. Graph. Appl.*, 24(5):27–35, 2004.
- [Tuf90] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1990.
- [Tuf01] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, second edition, 2001.
- [Tuf06a] Edward R. Tufte. *Beautiful Evidence*. Graphics Press, July 2006.

- [Tuf06b] Edward R. Tufte. Presenting data and information. [live course; Chicago, IL], October 2006.
- [WAM01] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *Proc. IEEE InfoVis*, page 7, 2001.
- [War02] Matthew O. Ward. A taxonomy of glyph placement strategies for multi-dimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.
- [War04] Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2nd edition, 2004.
- [Wik06a] Wikipedia. Cognitive load — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Cognitive_load&oldid=95293005, 2006. [Online; accessed 20-December-2006].
- [Wik06b] Wikipedia. Protein data bank — wikipedia, the free encyclopedia. World Wide Web, 2006. [Online; accessed 3-December-2006].
- [Wik06c] Wikipedia. Working memory — wikipedia, the free encyclopedia. World Wide Web, 2006. [Online; accessed 20-December-2006].
- [WL00] Matthew O. Ward and Benjamin N. Lipchak. A visualization tool for exploratory analysis of cyclic multivariate data. *Metrika*, 51(1):27–37, 2000.

List of Figures

1.1	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.	4
2.1	Perceptual color sequences.	8
2.2	<i>Studies of Embryos</i> by Lenardo DaVinci [Dav10]. Notice the tight integration of word and image, if not number as well. [Tuf06a] contains further, striking examples of such integrative data art.	10
2.3	A structure hit from a PDB web search.	14
2.4	Collage View from the PDB’s website.	15
2.5	Thumbnail images from a PDB collage. The images—which depict structures 2ACX, 2AJ4 and 2AJP in left-to-right order—belie the relative molecular weights of the corresponding structures: 66kDa, 60kDa, and 36kDa, respectively.	15
4.1	Protein structures and bound ligands displayed in PDQVis. Each glyph resents a PDB structure, its subunits (pie slices), number of residues (size), release date (blue is older, yellow newer), and bound ligands (radial whiskers); resolution (blur) is not shown. If the user selects a subchain, it is brushed in green. For example, the two green slices of structure 2OOX (lower right) represent identical subunits. The dynamic query interface (left) enables users to select ligands and query for proteins that bind the same; query results are circled in green.	20
4.2	Close-up view of glyphs in PDQVis.	20

4.3	The ligand legend in PDQVis provides users with detailed information about the name, color, and radial position of each ligand.	21
4.4	BioSpark is a table-based display in which each data field is represented as a column of glyphs.	22
4.5	Data columns in BioSpark. 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.	22
4.6	Sparklines representing the primary structure of protein subchains. . . .	23
4.7	606 sparklines representing the primary structure of proteins. From left to right the sparklines are rendered using contour only, height mapping, and filled contours.	23
4.8	Scribble trees depicting Enzyme Classification numbers.	25
4.9	Style boxes summarizing the primary structure of proteins.	26
4.10	An alternative glyph design to that of PDQVis (Fig. 1.1). Rectangles, rather than circles are used to represent protein chains. The ligand auras are layed out horiztonally on a shared scale.	27
4.11	A ligand aura that used shape-based encoding to display information about one ligand, or a group of ligands.	27

List of Tables

2.1	Preattentively processed retinal attributes [War04, 151-152].	6
-----	---	---