MizBee: A Multiscale Synteny Browser

Miriah Meyer, Tamara Munzner, *Member, IEEE*, and Hanspeter Pfister, *Senior Member, IEEE*
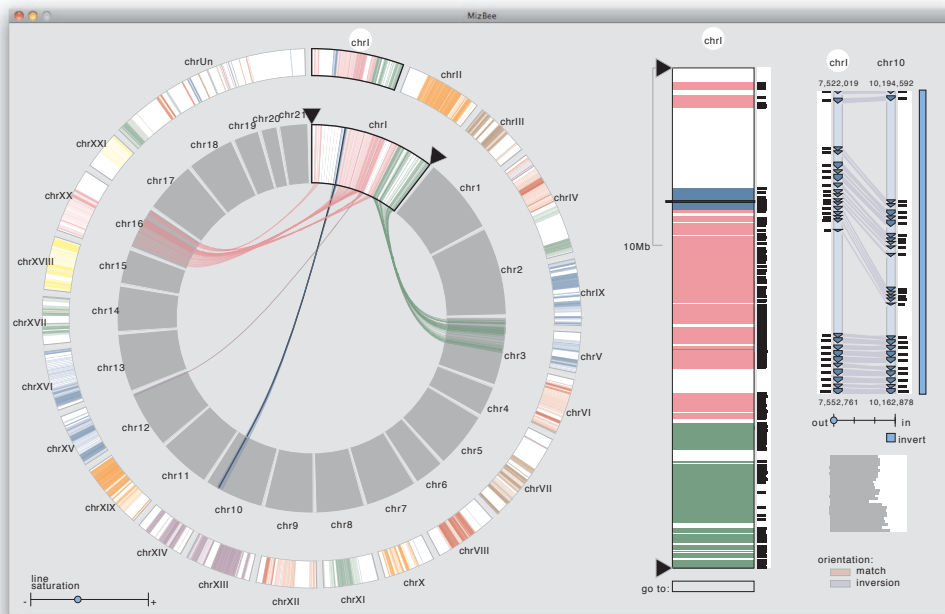

Fig. 1. The multiscale MizBee browser allows biologists to explore many kinds of conserved synteny relationships with linked views at the genome, chromosome, and block levels. Here we compare the genomes of two fish, the stickleback and the pufferfish.

Abstract—In the field of comparative genomics, scientists seek to answer questions about evolution and genomic function by comparing the genomes of species to find regions of shared sequences. Conserved syntenic blocks are an important biological data abstraction for indicating regions of shared sequences. The goal of this work is to show multiple types of relationships at multiple scales in a way that is visually comprehensible in accordance with known perceptual principles. We present a task analysis for this domain where the fundamental questions asked by biologists can be understood by a characterization of relationships into the four types of proximity/location, size, orientation, and similarity/strength, and the four scales of genome, chromosome, block, and genomic feature. We also propose a new taxonomy of the design space for visually encoding conservation data. We present MizBee, a multiscale synteny browser with the unique property of providing interactive side-by-side views of the data across the range of scales supporting exploration of all of these relationship types. We conclude with case studies from two biologists who used MizBee to augment their previous automatic analysis work flow, providing anecdotal evidence about the efficacy of the system for the visualization of syntenic data, the analysis of conservation relationships, and the communication of scientific insights.

Index Terms—Information visualization, design study, bioinformatics, synteny.

---

1 INTRODUCTION

In comparative genomics, scientists seek to answer questions about evolution and genomic function by comparing the genomes of different species. The comparison may shed light on evolutionary questions by providing evidence of shared ancestry between species. It can also indicate potential shared *function* where the sequences are similar. The effect of the genomic sequence on the functioning of an organism is a complex system involving many genes and regulatory elements working together in concert, a system which is difficult to understand by studying the genome of just a single species. Taken together, these

- *M. Meyer and H. Pfister are with Harvard University, E-mail: {miriah,pfister}@seas.harvard.edu.*
- *T. Munzner is with University of British Columbia, E-mail: tmm@cs.ubc.ca.*

indications allow for a range of biological insights, such as the relatedness of species in the Tree of Life, the discovery of new genes in the genome of a species, and the identification of sequences and mechanisms responsible for regulating the expression of functionally important genes.

To study the differences and similarities between genomes, biologists analyze relationships of conservation between genomic features. A *feature* is any genomic element of interest; genes are often the focus, but other possibilities are transposons, introns, and exons. The *similarity* of features is measured by how well their sequences match. *Conservation* refers to the similarity between genomic features in two different genomes, or sometimes within a single genome.

Synteny, which literally means "on the same ribbon", is the property that features occur on the same chromosome, and is often used to mean that they are contiguous within that chromosome. Because of the overwhelming number of features in many genomes, biologists abstract the idea of conservation by creating larger *syntenic blocks*, representing contiguous sets of features located on the same chromosome. Biologists use these blocks to look for several kinds of conservation relationships: proximity and location, size, orientation, and

similarity. Conserved synteny datasets are very large, with these relationships occurring across a wide range of scales from the level of the genome down through chromosomes and blocks to individual genomic features. Biologists use these relationships to infer answers to a broad range of questions related to evolution of and the functional effects of a specific genomic sequence.

Many algorithms have been proposed to compute blocks, but they all contain numerous parameters that must be tuned by a biologist, creating uncertainty in the data in the form of noise and false positives and negatives. While an algorithm can be written to answer any specific question about the reliability of the results or about a confirmed result, it is difficult to answer multiple questions across a range of scales using computational methods alone. Biologists incorporate visual data inspection into their work flow to augment relationship discovery algorithms, making effective visualization systems an important component of interpreting conserved syntenic relationships.

The goal of this work is to show different conservation relationships at different scales, expressed as comprehensible visual relationships. The first two contributions of this design study are a detailed characterization of the questions asked in this problem domain, and a taxonomic analysis of the visual encodings suitable for conserved syntenic data. Guided by this characterization and analysis, our third contribution is the design of the multiscale system MizBee, shown in Figure 1. MizBee is the first synteny browser to provide linked views across the genome, chromosome, and block levels, allowing the user to maintain context across all of these levels when exploring conserved syntenic data. In contrast to previous systems, we justify our design choices for spatial layout, color, and interaction in terms of known perceptual principles. MizBee uses the techniques of edge bundling and layering to reduce visual clutter, and also integrates quantitative statistical information in the context of spatial layouts showing genomic coordinate locations. The iterative design of MizBee was guided by close consultation with two target users. Our fourth contribution is two case studies that showcase how the design of MizBee evolved, and how it is currently used in their biological analysis workflow.

Next we discuss the biology behind, and computation of, conserved syntenic blocks, followed by our novel characterization of this data and description of the design space for visually encoding conservation. We then present MizBee, and discuss the features and implementation of the system. This discussion precedes the description of previous work in field. Finally, we present two case studies from users of MizBee, and finish with conclusions and directions for future work.

2 BIOLOGICAL BACKGROUND

The genome of a species is physically composed of multiple chromosomes, each of which is a long chain composed of the four nucleotides A, T, C, and G. Chromosomal rearrangements, in the form of deletions, inversions, or translocations, can occur within or between the chromosomes due to errors made by the cellular machinery responsible for maintaining the genome. Every so often rearrangements lead to an increase in the survival rate of an organism. Over time these changes accumulate, and sometimes lead to the divergence of species. Understanding how rearrangements could have occurred is a major topic in comparative genomics, as possible rearrangements inform biologists about the relatedness of species, genomically and functionally.

To find evidence of chromosomal rearrangements, biologists hunt for conserved sequences between the genomes of two species, or sometimes within the genome of a single species if it is thought that a duplication of the entire genome occurred. By analyzing the properties of these conserved sequences, biologists seek to answer a variety of questions, such as: Is there evidence of larger segments of conservation that could indicate a whole genome duplication? What changes to a genome can account for species variation? What segments of the genome account for the ability of a species to adapt to different environments? The answers to these types of questions not only enable scientists to determine the evolutionary relatedness of species, but to also help prioritize experimental analysis of genes in the search for true functional conservation between species.

As described in the previous section, biologists have proposed many

related algorithms for grouping features into larger syntenic blocks. Generally speaking, these algorithms first determine the most similar sequence in the *destination* genome for each feature in the *source* genome, as shown in Figure 2(a). Each of these *conserved pairs* has a *similarity score*, a percentage that indicates how similar one sequence is to the other, often referred to as the *strength* of the conservation. The similarity score is used to filter the pairs via a threshold value, a user-defined parameter that is often between 60 and 70 percent, as shown in Figure 2(b).

Blocks are then formed by combining source features, as shown with brackets in Figure 2(c). Features are grouped that are close to each other, that have matches on the same destination chromosome, and that also have the same *orientation* (sequence reading direction along the chromosome) relationship with their matches. Counterexamples to these grouping requirements are shown, respectively, with the orange, blue, and green ellipses in Figure 2(c).
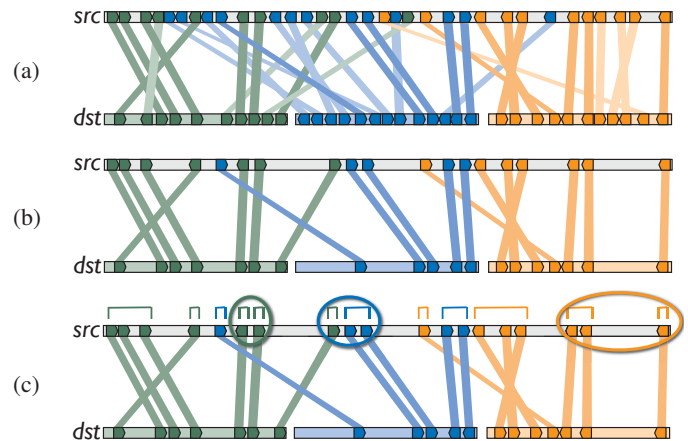

Fig. 2. Blocks are determined by (**a**) finding the most similar match for every feature in the source chromosome with the destination, where low saturation encodes low similarity scores; (**b**) filtering with a threshold on the similarity values; (**c**) combining features into blocks, denoted by brackets. Features must be close on the source chromosome, have matches on the same destination chromosome, and have matched orientation relationships. Counterexamples are circled in orange, blue, and green, respectively.

3 DATA AND TASK ABSTRACTION

We present the first contribution of this design study, a characterization of the problem domain. This characterization includes a description of conserved syntenic dataset structure, and a list of detailed questions about this data that the biologists ask to infer answers to higher level scientific questions. We gathered the raw data for this characterization by conducting a series of interviews with two target users, biologists who use conserved syntenic datasets as part of their analysis process.

The structure of datasets containing conserved syntenic blocks is broken into three main layers of scale. The highest level is the genome, which contains a list of chromosomes. The next level is the chromosome, which contains a list of blocks whose locations are specified in terms of the chromosome sequence coordinate system. The third block level contains a list of conserved features, which are specified with a chromosome id, coordinate along the sequence, length, orientation, tag, match on another chromosome, and similarity score. At an even lower level, a feature may contain the string of its constituent nucleotides. These datasets often contain *secondary* genomic features, whose location is interesting even though their individual names are not, as opposed to the named conserved features that are the direct objects of analysis.

The analysis of this data is challenging on two fronts. The first challenge is the size of these datasets and the range of scales of interest: they can have dozens of chromosomes, thousands of syntenic blocks, and hundreds of thousands of conserved features. Furthermore, the genome can be billions of nucleotides long, while some features of

Table 1. Questions for the analysis of conserved syntenic data, with the scale and relationship addressed by each. The scales are: g, genome; c, chromosome; b, block; and f, feature. The relationships are: p, proximity/location; z, size; o, orientation; and s, similarity.

	question	scale				relationship			
		g	c	b	f	p	z	o	s
1	Which chromosomes share conserved blocks?	X				X			
2	For one chromosome, how many other chromosomes does it share blocks with?	X	X			X			
3	What is the density of coverage and where are the gaps on: chromosomes? blocks?	X	X	X		X			
4	Where are the blocks: on chromosomes? around a specific location on a chromosome?	X	X			X			
5	What are the sizes and locations of other genomic features near a block?		X			X	X		
6	How large are the blocks?		X				X		
7	Do neighboring blocks go to the same: chromosomes? relative location on a chromosome?	X	X			X			
8	Are the orientations matched or inverted for: block pairs? feature pairs?		X	X				X	
9	Do the orientations match for pairs of: neighboring blocks? features within a block?		X	X				X	
10	Are similarity scores alike: with respect to neighboring blocks? within a block?		X	X					X
11	Are the paired features within a block contiguous?			X		X			
12	How large is a feature relative to other genes within a block?			X			X		
13	What are the sizes, locations, and names of features within a block?			X		X	X		
14	What are the differences between individual nucleotides of feature pairs?				X				X

interest are are less then a dozen nucleotides in length. The second analysis challenge is that there are multiple types of interesting conservation relationships. We have characterized these as addressing proximity/location, size, orientation, and similarity/strength. These appear across the entire range of scales, from the genome to a feature.

We have identified a set of 14 fundamental questions that biologists ask to gain scientific insight at different stages of the data analysis pipeline, shown in Table 1. These questions were gathered from two sources: interviews with our biologist collaborators about their data and analysis methods, and a study of problems addressed in the literature by other synteny visualization systems. We have organized them according to the scale at which they operate and the type of conservation relationship they address. Some of these questions pertain to the early data generation stage, probing the results of computational algorithms that determine the blocks. These algorithms have many parameters, such as the similarity score threshold for filtering feature pairs. While previous scientific insights might guide biologists in determining an initial range of parameter values, often they must be tuned for each individual dataset. Questions Q6 through Q11 attempt to determine whether the computed blocks are reliable, or if they are contaminated with noisy data due to poor parameter choices. Once the computed blocks are determined to be reliable, different questions are asked at later stages in the analysis pipeline to expose conservation relationships in the data. The relationships enable the inference of answers to higher level scientific questions. For example, questions Q1 through Q3 could lead to insights about possible chromosomal rearrangements.

We use these questions in our discussion of the capabilities of MizBee and previous systems. For example, MizBee is the first system to support all of the analysis questions Q1 through Q13, addressing the genome level, the chromosome level, and the block level. It does not address question Q14, however, since many previous systems address low-level nucleotide inspection, annotation, and editing.

4 VISUAL ENCODING OF CONSERVATION

The second contribution of this design study is a taxonomy of the design space that can be used to generate effective visual encodings of conserved syntenic data. We developed this generalized taxonomy from our critique of the design choices taken in other synteny browsers presented in the literature, as well as from the tools our biologist collaborators were using to visualize their data. All of the systems are designed around the representation of chromosomes because of their importance as a structural unit biologically, and also the inherent priority of chromosomes when talking about synteny. Chromosomes are a continuous piece of DNA, and are physically distinct structures, thus we establish our first design decision as the representation of chromosomes as segments. Also, a conserved feature is a segment of a source chromosome that has a similar match on a destination chromosome.

Hence, our second design decision is to represent conserved features as a segment on a chromosome with a matching segment on another.

The idea of representing a chromosome as a segment extends to blocks, which are also a continuous strand of DNA. At the block level, the number of conserved features is usually less than a few dozen, and conservation is a one-to-one relationship between only two blocks. The obvious and effective way to visually encode the matching relationships at this level is to show connections in the form of lines, curves, or ribbons between two parallel block segments, matching up the locations of conserved features on one block with their matching segments on the other.

At the next level up, the chromosome level, the matches become more complicated because there is a one-to-many relationship: a source chromosome can have conservation relationships with multiple destination chromosomes. At this level, encoding conservation relationships with connections is harder because the number of blocks on the chromosome can be large, leading to visual clutter from crossing connections. Color is a popular method for visually encoding conservation at this level, where a different color is mapped to each destination chromosome, and blocks on the source chromosome are colored according to their destination. Figure 3 shows examples of both the connection and color methods of visual encoding.
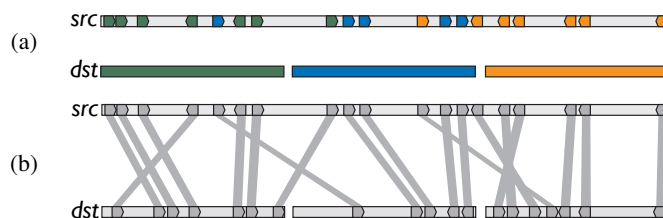

Fig. 3. At the chromosome and genome levels there are two methods for encoding conservation relationships: (a) color; (b) connection.
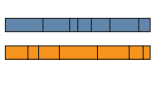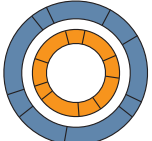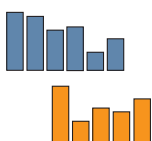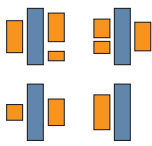
At the highest level, the genome level, there are complex many-to-many relationships: the full genome contains many source chromosomes, each of which can share conservation relationships with many of the destination chromosomes. At this level, both connection and color have been used in other systems to encode conservation.

Encoding conservation with connections allows for location information about the source *and* the destination to be shown. This method, however, does not scale well with the number of conserved features due to the visual clutter of numerous crossing lines. Encoding conservation with color does not entail this clutter problem because there are no crossings. Color encoding, though, only shows location information for the source, not the destination. Moreover, a basic perceptual principle is that less than one dozen colors are distinguishable when showing categorical data [23], and most genomes contain far

more chromosomes than that limit. Color encoding also has scalability problems as the number of conserved features increases, because color becomes more difficult to distinguish as the size of the colored region decreases. In MizBee, we make different choices about encoding with connection, color, or both depending on the level of the view, and limit the number of colors used to eight.

At the chromosome and genome levels, layout schemes must accommodate two different sets of chromosomes: source, and destination. From our analysis of other synteny browsers we classify possible layout schemes into two top-level categories, contiguous and discrete, as shown in Table 2. The *contiguous* scheme treats a set of chromosomes as contiguous elements, laying out the elements of a set end to end in a linear or circular pattern. In this scheme, the two sets of chromosomes can be separate or combined. For linear layouts, the source and destination sets are *combined* by placing the sets along a single line, whereas in the *separate* case the sets are laid out along two distinct lines. Similarly, for circular layouts, the source and destination sets are combined by placing both sets around a single circle, or placed around two individual circles in the separate case. The *discrete* scheme treats a set of chromosomes as individual elements, not requiring the chromosomes to lie end to end, and lays out the two sets of chromosomes in an interleaved or segregated pattern. In this scheme, *interleaved* layouts merge the sets, while *segregated* layouts isolate them. For any of these layouts, a different scheme can be applied to each set of chromosomes, creating hybrid layouts.

Table 2. A taxonomy of layouts for the two sets of chromosomes, distinguishing between the source in blue and the destination in orange.

While in theory, any of the layouts shown in Table 2 can be used for encoding conservation with color or connection, only a subset of the layouts are effective for each conservation scheme. For encoding with color, the visual representation of the destination chromosomes act as a color legend, thus these segments should be ordered and grouped to allow for easy understanding of the colormap. Thus, contiguous and segregated discrete layouts work best. When encoding with connection, there are fewer effective layout possibilities as there are more constraints: unique lines with minimal crossings, no obscuring of lines by segments, and minimal variance of line length. The effective layout schemes for connection encoding are thus linear separate, circular combined, and discrete interleaved. In MizBee, at the genome level we use a circular layout with connecting curves to reduce the amount of variation in the length of the curves, as well as to make conservation relationships of proximity more visually prominent.

5 MIZBEE

Our third contribution is the design of MizBee, a multiscale synteny browser that shows different conservation relationships at different scales, expressed as comprehensible visual relationships. This design was guided by the data characterization presented in Section 3, and was informed by the visual encoding taxonomy described in Section 4. It was iteratively refined in collaboration with the two biologists who were our target users. Their analysis needs motivated our highest level design decision of using multiple linked views [18], a visualization approach that is well suited for exploration of large datasets that

have clearly defined levels of structure. Below, we describe in detail the design and capabilities at each of the three levels shown in Figures 1 and 8: the genome view on the left, the chromosome view in the middle, and the block view on the right. The accompanying video tours the features of MizBee in action.

5.1 Genome View

The genome view, shown in Figure 4, provides a high-level overview of the many-to-many relationships between all the chromosomes. This view allows for the analysis of proximity/location and size relationships, answering questions Q1 through Q4 and Q7.

The view uses a separate-circular layout, with the source chromosomes on the outer ring. The inner ring shows the destination chromosomes arranged around a copy of the selected source chromosome at the top, with linked highlighting showing its location in the outer ring through a black outline. The one-to-many conservation information is encoded using connection, where blocks on the selected source chromosome are linked with blocks on destination chromosomes using B-spline curves. Conservation is also redundantly encoded with color, according to the destination chromosome at the end of the curve, to make proximity relationships more visually prominent (Q7). In the outer ring, the colors show the destinations of all the blocks, in an overview of the entire genome that provides answers to questions about coverage (Q3 and Q4) and about proximity (Q1 and Q2).
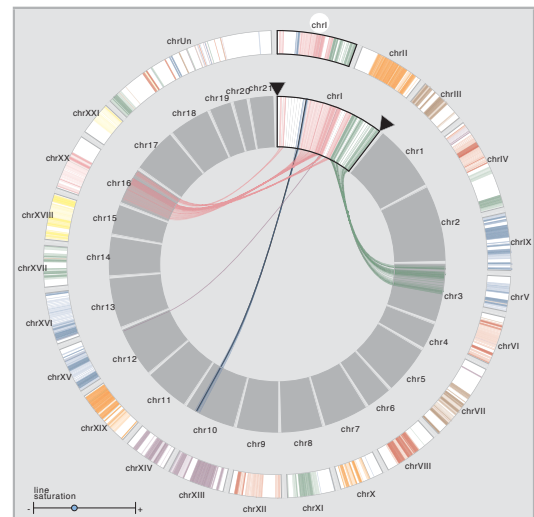

Fig. 4. In the genome view, all source chromosomes are shown on the outer ring. The inner ring has the destination chromosomes arranged around a copy of the currently selected source chromosome. Conservation is encoded with color for all the source chromosomes, and in more detail with connections for the selected one.

We use the *8-class qualitative Set1* colormap from ColorBrewer [1]. For genomes with more then eight chromosomes, the colors repeat. Our approach is to use color to accelerate scanning at the overview level, but we do not rely on it to tell the entire story. Details are shown on demand using connection for only the selected chromosome. This visual encoding design decision was motivated not only by our taxonomic and requirements analysis, but also by explicit feedback from our target users on the tradeoff between reducing information overload and visual clutter versus providing global overview information. The user can quickly browse by interactively selecting another source chromosome with mouse clicks or by using the left and right arrow keys.

Edge bundling is useful for generally reducing visual clutter, and more specifically for quickly pinpointing spurious blocks on a chromosome as shown in Figure 5. In MizBee, we use edge bundling to enhance the visual cues of proximity relationships (Q7) by bundling together connections from contiguous blocks that go to the same destination chromosome. Our implementation of edge bundling is inspired

by the work of Holten [7], specifically the ideas for using B-spline curves to render the connections between block pairs, as well the application of a parameter β to control how tightly curves are bundled together. Rather than use an external hierarchical structure [7] or a force-directed approach [8], in MizBee we exploit the geometry of layout itself to define control point locations. We produce the control points for the B-spline curves using information about blocks and their neighbors, and also about which chromosomes their matches reside on. The control points are generated such that contiguous blocks are bundled together very close to their origins on the source chromosome, and that bundles are clearly separated based on which destination chromosome they go to. Near the destination chromosome, control points ensure that a bundle splays out so the spatial extent of the bundle over the destination chromosome is clear.

The rendered blocks are filtered by moving the two triangles along the selected source chromosome to open or close the conservation viewing area, shown in Figure 6. The start and stop of the viewing area is reflected in the chromosome view as well, one of several linking mechanism between these views. Filtering allows users to home in on a region of interest, and to reduce the visual clutter in noisy data.
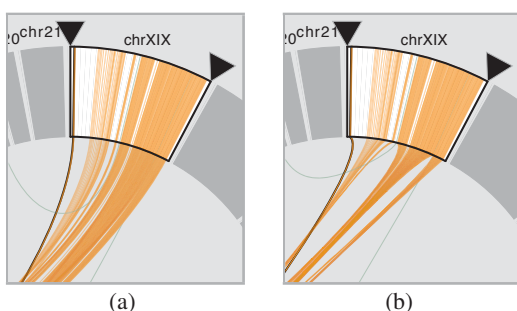

(a) (b)

Fig. 5. Edge bundling reduces clutter and makes spurious blocks easier to see. (a) Without edge bundling, the exact locations of the isolated green blocks are hard to see. (b) The locations of spurious blocks are more clearly visible because of breaks between the bundle groups.
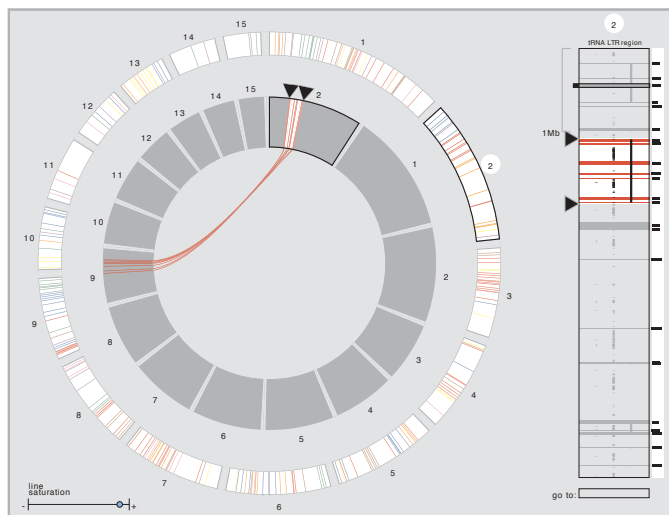

Fig. 6. The chromosome view on the right has more room for the details of block locations, and also shows statistical information and layered annotation tracks. The three tracks in this rhizopus dataset depict the location of tRNA and LTR transposons as well as larger conserved regions. Blocks can be filtered in either of the linked genome or chromosome views using the triangles.

5.2 Chromosome View

The chromosome view is a detailed look at the data at the block scale, showing the blocks within the selected source chromosome from the genome view. This view appears in the middle of the display, to the

right of the genome view. The chromosome view provides answers to questions about proximity/location, size, and similarity relationships of blocks within a chromosome (Q3 through Q6, Q10).

The chromosome view shows the location of blocks within the selected chromosome, color coded to correspond to the colormap of the genome view. Blocks are selected by directly clicking on a block in the chromosome view, or by using the up and down arrow keys. The selected block is outlined in the chromosome view, and drawn in black in the genome view. Once a block is selected, the block view is also updated with the selection. The chromosome view is a vital link between the highest and lowest level views of the data.

The chromosome view shows the same information as in the small curved chromosome segment in the genome view, but supports more precise spatial relationship judgments because the screen area available for the display is both rectilinear and several times larger. It also incorporates two additional information channels. On the right side of the chromosome is a histogram showing the average similarity score for each block. This statistical summary reveals similarity relationships of the blocks in the context of spatial location (Q10). Also, annotation tracks are layered on top of the blocks to indicate the presence and location of other interesting secondary genomic features (Q5). Figure 6 shows three annotation tracks: two kinds of transposons, tRNA and LTR, and larger conserved regions.

Filtering by moving the triangles in this view up and down is also mirrored in the genome view. The filtering enables an understanding of proximity and spatial relationships related to the location of layered annotations when using the genome and chromosome views together, shown in Figure 6 where the data is filtered based on a feature in the region annotation track.

The text box below the chromosome labeled go to accepts a location in chromosome sequence coordinates, and changes the selection to the block nearest that location, supporting question Q4.

5.3 Block View

The most detailed view of the data is shown in the block view, which provides details about the conservation relationships of features within the selected block related to proximity/location, size, orientation, and similarity. Using this view, shown in Figure 7, it is possible to answer questions Q8 through Q13.

The block view uses a one-to-one layout, rendering a conserved block pair as parallel segments. The coordinates of the start and stop of the block are printed above and below the segments. Conservation is encoded by linking features and their matches with ribbons, which are pale red when connecting features with the same orientation and pale blue when the orientations are inverted. The features are represented as oriented glyphs to indicate their orientation at their specific locations relative to the block start and stop, allowing for analysis of proximity, size, and orientation relationships (Q8, Q9, and Q11 through Q13). By selecting different blocks using the arrow keys, the analysis of these relationships can be extended to neighboring blocks as well.

The blocks and features have the same color coding as in the chromosome and genome views. The block view also contains statistical information about the similarity of each conserved feature pair, which is the detailed information underlying the averaged similarity score for the entire block. We again use histograms showing bars next to each block for context, and also have a second linked histogram showing the bars next to each other to enable precise length comparisons. Mousing over a feature highlights its similarity value in the lower histogram, and shows the value numerically, as shown in Figure 7(b). These views and interactions allow relationships of proximity/location and similarity to be analyzed (Q10 and Q13).

Analysis of blocks with many criss-crossings is made easier by flipping the entire paired block with the invert button, as shown in Figure 7(b). This functionality, also supported by previous work, is useful because of the high probability of inversions during evolution.

The size of features may be so small relative to the size of the block that important details cannot be seen, as shown in Figure 7(c). If the selected block contains any features that are smaller than five pixels, a zoom slider appears that allows all features to be represented by at
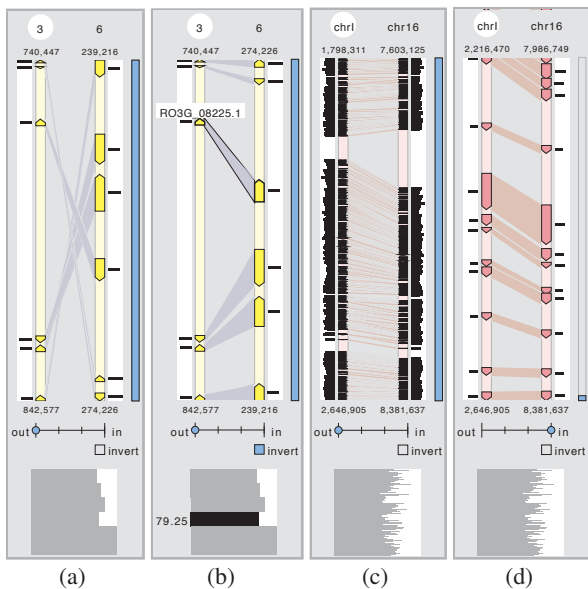
Fig. 7. The block view shows features and their matches using oriented glyphs and connecting ribbons. (a) An evolutionary inversion leads to many crossings. (b) Flipping the orientation of the entire block with the invert button solves the visual clutter problem. Also, mousing over a feature highlights its similarity value in the lower histogram, and shows the value numerically. (c) Blocks may have so many features that details cannot be seen. (d) When the view is zoomed, the scroll bar on the right allows panning.

least five pixels at the maximum zoom level, as shown in Figure 7(d). A user zooms in by double clicking on a location in the block view or by moving the slider; zooming out is controlled by the slider. A scroll bar to the right allows for panning up and down the zoomed view.

5.4 Implementation

MizBee is implemented in the Processing programming language [17]. Executables and source code are available at http://mizbee.org.

6 Previous Work

Many previous systems for analyzing conserved syntenic data are built on top of existing frameworks for browsing genomic data, which greatly constrains their designs. Ensemble [2] and SynBrowse [15] are two example systems that use nucleotide-oriented frameworks. These viewers use a separate-linear layout, and connection for encoding conservation. The feature-level views of these systems do not allow for answers to questions Q1 through Q10 at the chromosome and genome scales, and they suffer from visual clutter with many crossing lines when more then a few dozen conserved features are viewed.

Several viewers include chromosome level views, including SyntenyVista [9] and Sybil [21]. Both viewers use color for encoding conservation. SyntenyVista has a segregated-discrete layout, and Sybil a separate-linear layout. These viewers do not support the genome-level question Q1, and suffer from color distinguishability problems for genomes with more then eight or ten chromosomes. Sybil specifically targets small genomes, such as those of viruses.

Viewers that include a genome level view are Cinteny [20], Mauve [4], and Apollo [12]. Cinteny uses a segregated-discrete layout and encodes conservation with color, again making visualization challenging for genomes with average to large numbers of chromosomes. Mauve uses a separate-linear layout, encoding conservation with connections and using color to distinguish between blocks. This viewer is very challenging to interpret due to the visual clutter of many crossing lines and many colors, as well as the large variance in line length. Apollo takes a different approach at the genome level, laying out chromosomes in an interleaved-discrete scheme, and using connection to encode conservation. While this viewer succeeds in solving the visual clutter problem, it does have the problem that the layout

of destination chromosomes around the different source chromosomes is not spatially stable. The size and locations of destination chromosomes vary from one source chromosome to the next, undercutting the spatial memory of the user.

While the previous methods allow the user to drill down into more detailed views, Circos [10] only shows a genome level view of the data with a combined-circular layout, redundantly encoding conservation with connection and color. Although the non-interactive viewer provides an information-rich display, it does not show information at the block level, so questions Q3 and Q8 through Q13 are not supported.

None of the previously mentioned viewers show similarity values at the block or chromosome level, so they do not support question Q10. SynBrowse, however, encodes similarity with color at the low feature level. Biologists have used other visualization tools to analyze similarity/strength relationships. One approach is to *align* the genomes, namely to rearrange one genome relative to the conserved regions of another, and then plot similarity values above the aligned views [5, 14]. Another method is to use a scatter plot, where two genomes or chromosomes are placed along the x- and y-axes of the plot, and locations of conservation are encoded with dots, colored or sized according the strength of the conservation [11, 16]. Neither of these methods are able to answer the other questions related to proximity/location, size, or orientation, so they usually must be used in conjunction with another view of the data.

There is also previous work in the visualization community for showing connections using a circular layout, an early example of which is proposed by Salton *et al.* [19] for visualizing text data. The commercially available software Daisy [3] and NetMap [6] explicitly link nodes around a ring and show additional information at nodes such as histograms or metadata. Several systems augment the circular view with interactivity mechanisms that allow the placement of nodes in the center of the circle, such as TimeWheel [22] and VisAlert [13].

7 Case Studies

Our fourth contribution is to demonstrate the capabilities of MizBee on two datasets: one from each of our target user collaborators, both of whom are active research scientists. Executables containing the data from both of these case studies are at http://mizbee.org.

7.1 Rhizopus

Rhizopus oryzae is a fungus characterized by an extremely rapid reproduction growth rate, and is commonly found as fuzzy gray and white mold growing on fruits and vegetables. This fungus, studied by our first biologist collaborator, is a primary cause of mucormycosis, a potentially life-threatening fungal infection in immune-compromised individuals. By homing in on genes that are responsible for the rapid reproduction of rhizopus, as well as for the structural integrity of the organism, scientists hope to develop effective drug therapies that target the genetic origins of these mechanisms in order to stop the spread of infection in a patient. In the process of uncovering these genes, our collaborator discovered evidence for a *whole genome duplication* in the evolutionary history of the fungus.

This first collaborator was already in the late stages of the analysis process when the design of MizBee began. She had made her breakthrough by discovering a correlation between the presence of *transposons*, mobile genomic features that jump around the genome during evolution, and sections of conserved syntenic blocks, indicating the presence of much larger *regions* of conservation within the genome. This correlation is very clear when the location of the conserved blocks are shown along with the location of the transposons. The initial ideas for MizBee came out of discussions on how to effectively communicate her findings. She had difficulties in simultaneously presenting the correlation between transposons and conserved blocks and the characterization of gene pairs that define such regions through a static image. She hoped we could design a visualization that could be immediately understood. Figure 8(a) shows a region circled in red where the transposons in the annotation tracks tRNA and LTR exist in large numbers between some blocks that go to the same destination chromosome, a region that is also shown in Figure 6. By removing the transposons
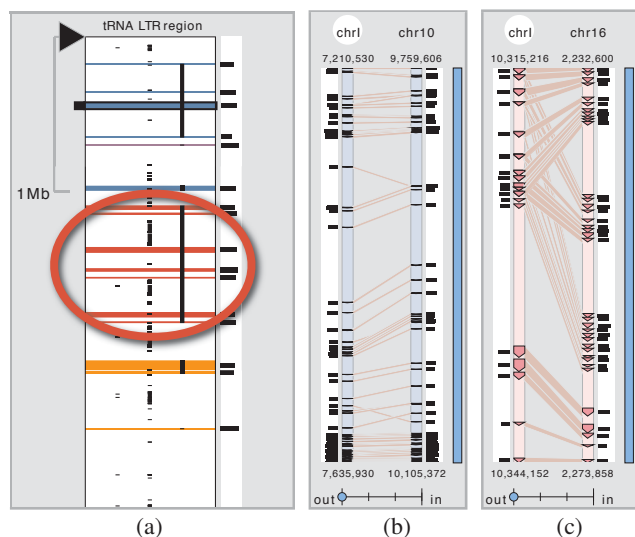
Fig. 8. **(a)** Our first collaborator found evidence of a whole genome duplication in *rhizopus* by observing large regions of conservation related to the location of transposons. An example is circled in red, and is also shown in Figure 6. MizBee successfully shows this relationship in a visually comprehensible way, and this late-stage collaborator plans to use it to communicate her findings. **(b)** The parameter that defines *acceptable* reordering for our second collaborator is fuzzy, and visual inspection of the data allowed him to verify his algorithm quickly. The amount of destination gene reordering here is acceptable. **(c)** An unacceptable amount of reordering, as well as a duplication event in the pufferfish genome.

from the sequence when computing syntenic blocks, she extended the conservation to larger regions, shown in the `region` annotation.

Our initial discussions on a visual encoding of her findings led us to the circular genome view that showed not only the location of source blocks, but also those of destination blocks, as she was using color to encode conservation which did not show this latter information. Also important in her work are the similarity scores within conserved blocks, as well as the number of genes in between conserved genes which she determines by looking at the tags supplied for each conserved gene. These two pieces of information allow her to ask questions about which genes were lost after the ancient duplication of this genome. Our collaborator also found the ability to visually invert a destination block useful for clarifying the contiguousness of her computed conservations. Although she provided feedback on the MizBee prototype during its refinement, she did not use it directly in her analysis process, which she completed prior to the development of MizBee. She plans to use MizBee to communicate her findings as well as those of future related projects.

7.2 Stickleback and Pufferfish

The stickleback is a highly adaptive fish species able to live in oceans, in rivers, and in lakes. Biologists believe that about 10 to 15 thousand years ago, the last Ice Age stranded formerly ocean-dwelling stickleback in freshwater systems, causing the fish to quickly adapt to the nonsalinity environments for survival. In this relatively short time span, the stickleback has diverged into a set of populations with very diverse morphologies and behaviors. By studying the adaption mechanisms in the stickleback genome, biologists hope to answer questions about evolution, such as: What kinds of genes underlby specific morphological differences? Does evolution use the same genes or different genes when evolving the same traits independently? What kinds of mutations lead to new traits?

To understand more details about the stickleback genome, biologists compare the stickleback with other well-characterized fish genomes, such as that of the pufferfish, to discover previously unknown or overlooked features in the stickleback genome. Figure 1 shows the source stickleback genome compared to the destination pufferfish genome. Our second collaborator used MizBee in the early

stages of analysis, while developing a new algorithm to find conserved syntenic blocks within these two species. This early-stage user focused on using the tool to understand the reliability of computations that generate conserved syntenic data, as discussed in Section 3.

He said *"The first time I saw my data in [MizBee] I was totally disappointed. The data was very noisy, and there were many small blocks that went to different chromosomes."* His previous data confirmation methods — using scatter plots and raw text analysis — hid away many of the small, noisy blocks generated by his algorithm. Figure 9 shows a series of three data sets that he generated through his algorithm refinement process. Figure 9(a) shows the first dataset he loaded into MizBee, containing many spurious blocks. Figure 9(b) shows one of his attempts to refine that approach, which shows only minimal improvement. After looking at several further refinement attempts in MizBee, he took an entirely different algorithmic approach, which resulted in the very clean dataset shown in Figure 9(c). When asked how long it would have taken to make the algorithmic breakthrough using his previous data-confirmation methods, he responded: *"Honestly, I don't know. I don't think I would ever have gotten here. The noise was very hard see in the scatter plots while [MizBee] is much more unforgiving."*

We received feedback from this collaborator during the later stages of MizBee development. For this biologist, the genome view was particularly useful due to the ability to see which chromosomes share relationships with multiple destination chromosomes by looking at the colors in the outer ring. He advocated for the single source chromosome in the inner ring to avoid information overload and too much visual clutter. He commented that the ability to quickly browse through all of the source chromosomes in this view was incredibly helpful, in stark contrast to his previous visualization methods that produced only a single, static chromosome view. The ability to interactively move from block to block in the chromosome view was similarly helpful. He also used the filtering method to home in on specific conservation regions, as well as edge bundling to quickly find small, spurious blocks. In the block view, he would quickly run through all of the blocks, looking for two things: one, inverted blocks are of particular interest for his algorithm; and two, he would quickly check whether all the destination genes in a block were contiguous. Using MizBee was a particular improvement from his previous methods for this latter task as he allows for some amount of reordering of the destination genes, with a fuzzy threshold for what "too much" means. Visual inspection of the blocks gave him a much clearer way to confirm his data then writing an algorithm to detect unacceptable amounts of reordering. Figure 8(b) shows an example of a block with an acceptable amount of gene reordering, while Figure 8(c) shows an unacceptable amount. While the example in Figure 8(c) is beyond the threshold for reordering, upon further investigation, it clearly reveals a duplication event in the pufferfish genome, a potentially interesting biological insight that is easily inferred from this view.

This collaborator's research is still in the initial stages, and he plans to use other features of MizBee for further downstream analysis. For example, he would like to use the zooming and `goto` box for a more refined analysis, such as an understanding of conservation relationships around a specific gene. He also plans to use MizBee for a more detailed investigation of small, isolated blocks in his latest dataset.

8 DISCUSSION

MizBee is a general visualization framework for analyzing conserved syntenic data. Our two collaborators used the same tool to analyze datasets with quite different characteristics. The first used the same fungus genome for both source and destination as she was analyzing a whole-genome duplication event, and used the tool very late in the analysis pipeline. Her dataset was relatively small with most blocks containing only a limited number of conserved features, but three annotation tracks showing secondary features were a critical part of the analysis process. The second compared two larger fish genomes, with both more features in each block and more total blocks, at an early stage of analysis. MizBee proved to be useful in both situations, providing some evidence that our design process, grounded in a careful
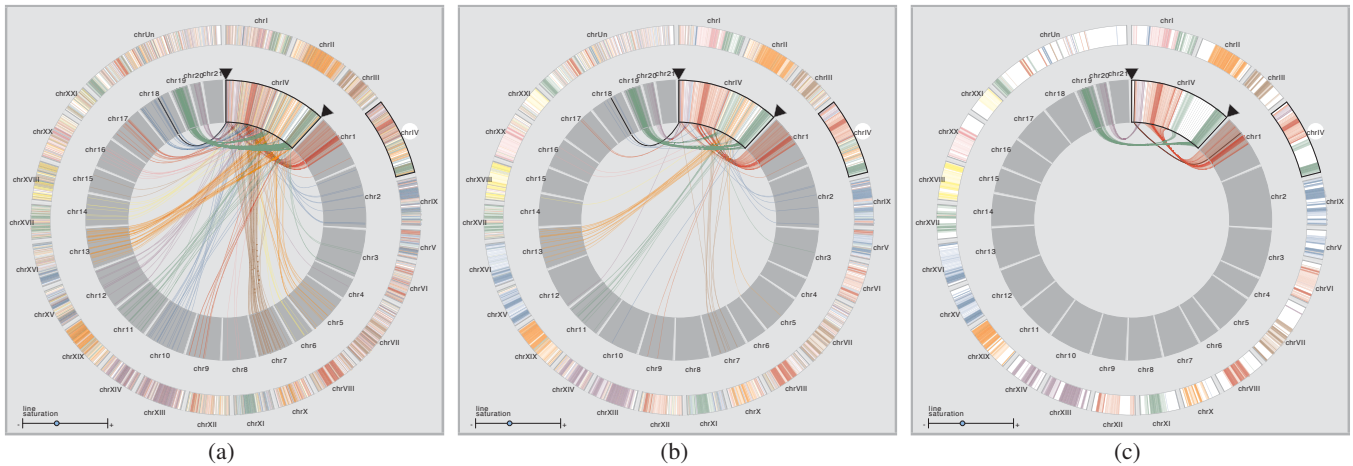
Fig. 9. Our second collaborator used MizBee during the creation of his block computation algorithm for a stickleback-pufferfish dataset. **(a)** The first algorithm created a surprising number of noisy blocks. **(b)** Attempts to refine the original algorithm led to only limited improvements. **(c)** An entirely different approach to computing conservation resulted in a very clean dataset.

characterization of the domain requirements and our taxonomy of conservation encodings, solved the intended problem. We believe that our characterization and taxonomy could provide effective guidelines for other future comparative genomics browsers.

9 CONCLUSIONS AND FUTURE WORK

Biologists working in the field of comparative genomics are faced with understanding large datasets that span a range of scales and contain numerous types of interesting relationships. Visualization is an important part of their workflow, augmenting computation algorithms to gain an understanding of these relationships. In this work, we target conserved synteny data with the goal of providing effective visual cues and intuitive interaction mechanisms that enable and speed up the scientific discovery process. To meet this goal, we present a novel characterization of the data and a taxonomy of the design space for visually encoding conservation. These two contributions guide our design of Mizbee, the first synteny browser to have side-by-side linked views that span a range of scales, from the genome to the feature. We present two case studies from our biologist collaborators, both of whom were active participants in the iterative refinement process of developing MizBee.

It would be interesting future work to adapt MizBee for use with on-the-fly parameter adjustments of conservation algorithms related to the rhizopus data set, rather than as a viewer for the static data. It would also be useful to support a more sophisticated pane management scheme that would allow users to modify the size of the views based on their current analysis needs.

10 ACKNOWLEDGMENTS

REFERENCES

[1] C. Brewer. Colorbrewer. http://colorbrewer.org, 2009 (accessed March 1, 2009).
[2] M. Clamp et al. Ensembl 2002: accommodating comparative genomics. *Nucleic Acids Research*, 31(1):38–42, 2003.
[3] Daisy Analysis Ltd. Daisy. http://www.daisy.co.uk/, 2009 (accessed March 1, 2009).
[4] A. C. Darling et al. Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *Genome Research*, 14(7):1394 – 1403, July 2004.
[5] K. A. Frazer, L. Pachter, A. Poliakov, E. M. Rubin, and I. Dubchak. Vista: computational tools for comparative genomics. *Nucleic Acids Research*, 32, July 2004.
[6] J. Galloway and S. J. Simoff. Network data mining: methods and techniques for discovering deep linkage between attributes. In *Proc. Asia-Pacific Conference on Conceptual Modelling (APCCM)*, pages 21–32. Australian Computer Society, 2006.
[7] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics,*, 12(5):741–748, Sept.-Oct. 2006.
[8] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum (Proc. EuroVis 09)*, 28(3):983–990, 2009.
[9] E. Hunt et al. The visual language of synteny. *OMICS: A Journal of Integrative Biology*, 8(4):289–305, 2004.
[10] M. Krzywinski. Circos. http://mkweb.bcgsc.ca/circos/, 2009 (accessed March 1, 2009).
[11] S. Kurtz et al. Versatile and open software for comparing large genomes. *Genome Biology*, 5(2):R12, 2004.
[12] S. Lewis et al. Apollo: a sequence annotation editor. *Genome Biology*, 3(12):research0082.1–0082.14, 2002.
[13] Y. Livnat, J. Agutter, S. Moon, and S. Foresti. Visual correlation for situational awareness. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pages 95–102, 2005.
[14] I. Ovcharenko et al. Mulan: Multiple-sequence local alignment and visualization for studying function and evolution. *Genome Research*, 15(1):184–194, 2005.
[15] X. Pan, L. Stein, and V. Brendel. Synbrowse: a synteny browser for comparative sequence analysis. *Bioinformatics*, 21(17):3461–3468, 2005.
[16] D. Rasko, G. Myers, and J. Ravel. Visualization of comparative genomic analyses by blast score ratio. *BMC Bioinformatics*, 6(1):2, 2005.
[17] C. Reas, B. Fry, and J. Maeda. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007.
[18] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. Intl. Conf. on Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pages 61–71. IEEE Computer Society, 2007.
[19] G. Salton, J. Allan, C. Buckley, and A. Singhal. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264(5164):1421–1426, June 1994.
[20] A. Sinha and J. Meller. Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. *BMC Bioinformatics*, 8(1):82, 2007.
[21] TIGR (The Institute for Genomic Research). Sybil: Web-based software for comparative genomics. http://sybil.sourceforge.net, 2009 (accessed March 1, 2009).
[22] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *Proc. ACM Symp. on Applied Computing (SAC)*, pages 1242–1247, 2004.
[23] C. Ware. *Information visualization: perception for design*, chapter 4. Morgan Kaufmann, 2000.