Predicted Transcription Factor Binding Site Viewer

Andrew A Carbonetto *
CPSC 533, 2005 Final Project
University of British Columbia
Department of Computer Science
Vancouver Canada

ABSTRACT

It has been shown that transcription factors can play a significant role in gene expression. The ability to predict the location and clustering of binding sites for transcription factors, and later the confirmation of such sites, can play a pivotal role in medical research. Typical prediction methods produce a high level of false positives, sometimes in the area of 70% of the hits. These datasets are typically large (approx. 4000 hits for a 1000bp sequence), and verifying each hit individually can be expensive, and time consuming. TFBS viewer is a display tool to view and navigate these typically large sequences, but more importantly, the viewer provides several tools for comparison with itself and related species. The efficient speed and comparison capabilities of the viewer make it a good candidate for binding site searches.

Keywords: Information Visualization, Transcription Factor Binding Sites

1 INTRODUCTION

It has been known for quite some time, of the importance of DNA and DNA mutation. Small changes of single or few nucleotides can have detrimental effects to an individual. Some can cause fatal deceases, such as cancers, others have little to no known effect, still more can cause an advantageous mutation that can lead to evolution.

In studying these changes, we can classify them into three categories:

1. Deleterious: a mutation that reduces the chances (or the *fitness*) of an individual to survive (possibly even fatal).
2. Advantageous: a mutation that increases the chances (*fitness*) that an individual will survive
3. Neutral: a mutation that has no effect on the survival (*fitness*) of a species.

Kimura's [1963] "Neutral Theory" states that most mutations are usually Deleterious or Neutral, with very few being advantageous. Ohta expanded Kimura's theory [Ohta, 1992] to included a Nearly Neutral set. The effects of a Nearly Neutral mutation being very small to the individual's survival. Ohta stated that Nearly Neutral, Neutral and Deleterious mutations are by far the most common mutations, while Advantageous mutations are rare.

When comparing these theories on distantly related species, we can definitely see some long term effects of Neutral and Nearly-Neutral mutations. Most of these mutations are not important because they happen to lie within sections of the individual's DNA that are not important. Advantageous or Deleterious mutations normally lie within important areas of the DNA. If we compare two distantly related species, we can assume that areas of the DNA that are diverging are unimportant (the so-called "Junk DNA"), while areas that are conserved are important. This evolutionary phenomenon is called genetic drift.

*e-mail: acarbo@cs.ubc.ca

Researchers have made used of genetic drift when comparing related species, in an effort to understand the effects of evolution. Sorting through the Junk DNA to find interesting and important DNA can be frustrating, because of the level of noise that is often experienced. This noise can sometimes be confusing to an observer, and often takes some time to filter the set to manageable size.

1.1 Biological Data - Gene Transcription Regulation

DNA encodes for the building blocks of an individual, and none more so then an individual's gene's. The process that reads the gene and writes a new protein is called Transcription. If the transcription process is halted by a mutation, it can often have deleterious consequences. We can obviously see that genes will be well conserved between related species.

The mechanism that initializes and regulates transcription is built up of individual Transcription factors - proteins that factor into the transcription of a gene. These Transcription Factors first bind to the DNA sequence to have an effect on transcription. Common effects of Transcription Factors involve enhancing (an increase in) or silencing (a decrease in) the number of times that a gene is transcribed (called the expression of a gene or its protein product), or it can effect the protein end product (genes tend to code for more then one protein). It has been shown that the destruction or creation (by mutation) of these binding sites can influence the transcription of a gene. There exists motivation to detect and predict how and where these Transcription Factor Binding Sites (TFBS) are located.

There exist several large databases that maintain a collection of TFBS; The TRANSFAC database [at http://www.gene-regulation.com/] or JASPAR [Sandelin et al. (2004) - at http://phylofoot.org/]. These databases store the exact sequence of DNA that a Transcription Factor bound to. A summation of the frequency of such binding sites yields a frequency matrix, which can be used to predict and detect such Transcription Factors.

There are several TFBS predictors available (for non-commercial) for trial. MatInspector [Quandt K et al (1995) and Cartharius K et al. (2005) - http://www.genomatix.de/] takes a set of motifs (as default, the TFBS frequency matrices from TRANSFAC are used) and an input sequence of DNA to predict the location of TFBSs. Consite [Sandelin A (2004) - http://phylofoot.org/] is another TFBS predictor that uses the JASPAR data set as a default input. All of these programs will output a set of motifs (potential binding sites for transcription factors), many of which will be incorrect predictions. A difficult task, is reducing the number of outputted false positives, or optionally, finding a method to effectively sift through the output an find the more confident results.

1.2 Description of Task

Researchers are motivated to find efficient and precise methods for TFBS prediction, and I would like to motivate further that the analysis of such results needs an equally efficient method. Both MatInspector and Consite provide web-based, interactive, output, but the interaction is far from smooth (even with a fast internet connection, submission and retrieval of results are slower then real-time) and the display is limited. I would like to provide an offline solution to view predicted TFBS.

A new solution should not only be able to provide the functionalities that MatInspector and Consite's visualizations provide, but should be able to compare across several species and across multiple datasets, as well as being able to compare within a single dataset. These techniques can greatly aid a user in finding more confident results.

1.3 Contents

This paper is arranged in following manner:

Section 2, I mention related works, and other online implementations of TFBS viewers.

Section 3 talks about the goals of the projects, and how a solution could be implemented.

Section 4 gets into greater detail the Prefuse implementation strategy.

Section 5 gives examples of the use of the TFBS Viewer.

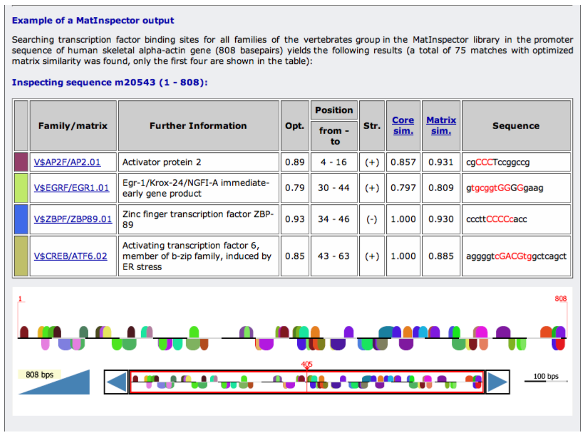
Section 6 to 8 conclude this paper with the strengths and weaknesses of the TFBS Viewer approach and future work that could be included.

2 Related Work

MatInspector [Quandt K et al (1995) and Cartharius K et al. (2005) - http://www.genomatix.de/] and Consite [Sandelin A (2004) - http://phylofoot.org/] provide online visualization of their predicted results (see figure 1 for matInspector view). The interaction is slow, there exists no smooth filtering of results, and the options are limited. Finally, the sequence results tend to static, and comparison across two or more species is sometimes not provided. This is partially due to the online capabilities.

The MatInspector visualization is a static display, with a limited viewable range, using a focus plus context visualization. There exists two images, one is a zoomed out display of the other. The zoomed out display (the *context display*) is useless for viewing the individual predicted motifs, because of clutter and thus occlusion. It is therefore used as a navigation tool. The sliding window on the zoomed out display can be moved horizontally, thus changing the focus of the second window (the *focus display*). There is no smooth transition for the window display, most probably due to memory and speed constraints (the visualization is fully online). The zoomed in display shows the individual motifs, as well as their relative location on the DNA sequence. Each motif is designated a color, although colors overlap when there are a lot (greater then approx. 50) of motifs to view. Clicking on the individual motifs will zoom to another link on the browser. The similarity score (see Cartharius K et al 2005) is represented as the height if each individual motif.

Although MatInspector is online, and provides a simple display of the motifs to a user, it lacks some very fundamental options. For one, the color scheme is bright and confusing. Unless there are only a few points (which would defeat the purpose of a visualization all



Example of a MatInspector output

Searching transcription factor binding sites for all families of the vertebrates group in the MatInspector library in the promoter sequence of human skeletal alpha-actin gene (808 basepairs) yields the following results (a total of 75 matches with optimized matrix similarity was found, only the first four are shown in the table):

Inspecting sequence m20543 (1 - 808):

	Family/matrix	Further Information	Opt.	Position	Str.	Core sim.	Matrix sim.	Sequence
				from - to				
🟪	V$AP2F/AP2.01	Activator protein 2	0.89	4 - 16	(+)	0.857	0.931	cgCCCTccggccg
🟩	V$EGRF/EGR1.01	Egr-1/Krox-24/NGFI-A immediate-early gene product	0.79	30 - 44	(+)	0.797	0.809	gtgcggtGGGGgaag
🟦	V$ZBPF/ZBP89.01	Zinc finger transcription factor ZBP-89	0.93	34 - 46	(-)	1.000	0.930	cccttCCCCcacc
🟨	V$CREB/ATF6.02	Activating transcription factor 6, member of b-zip family, induced by ER stress	0.85	43 - 63	(+)	1.000	0.885	aggggtcGACGtggctcagct

Figure 1: MatInspector introduction example. Included is an example listing of motifs and a visualization of the sequence. Each semi-circular item represents a motif. The bottom figure is a zoomed-out display of the entire sequence.

together), the colors overlap and become misleading. The only colormap is off screen. Using no colors might even be a better solution then the one implemented. Also, the colormap is bright, and distracting. Secondly, there is no way to view the information within each displayed motif at the same time as viewing the display. They could have implemented an extension to the display with allows for this option. Finally, there is no way to filter the results. This is unfortunate, because one application of the display is to help remove unwanted (and false with greater certainty) motifs from results to aid in further investigation.

One major drawback of the MatInspector view is that is does not provide any option to compare multiple sequences together. This option is, however, implemented in the Consite display (although the number of sequences is limited to two). Consite provides the user with some simple strategies to find significant results. One is the sequence wide filter. This helps reduce the number of displayed results (and the amount of clutter) that is visualized by the user. Consite does provide a parallel display of the motif information (a pop-up window is opened).

Although Consite's visualization is superior to MatInspectors in many respects, Consite lacks some other fundamental ideas. Individual motifs can still not be filtered. The display of sequences (in orientation and alignment) is final (once calculated, it is static and cannot be changed). Consite also does not display some addition information, include motif species or family type, nor does it readily show the motif similarity score (as a degree of confidence, this score can be very significant to a user).

rVista [Loots G. et al (2002) - http://genome.lbl.gov/vista/rvista/submit.shtml] is another predictor that provides a solution for multiple (equal to or greater then 2) sequence comparisons. However, rVista's solution is very simple, and the visualization cannot be altered at all - the alignment is also static. rVista has few options for filterering, but no dynamic movement, and limited individual motif information.

Softberry [http://www.softberry.com/] and Match [http://www.gene-regulation.com] provide TFBS predictions without displays. Because of this fact, their results tend to be hard to understand, and further data mining needs to be done on a case-by-case basis. Analyzing each individual motif is a long (and arduous) task.

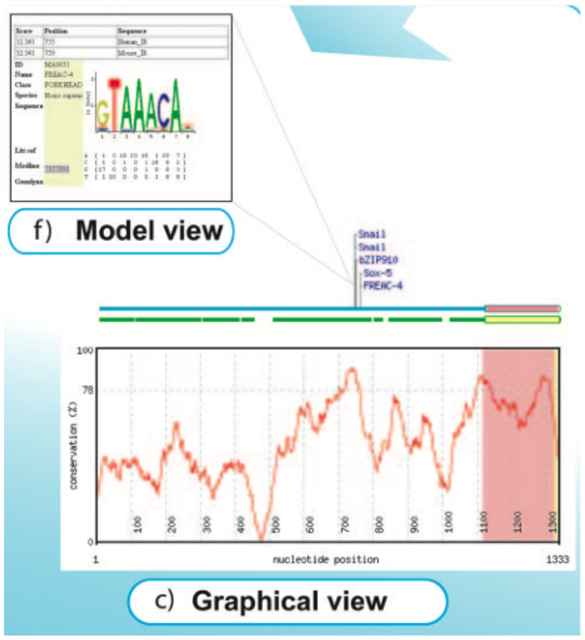


Figure 2: Consite example. The line graph represents the conservation level between the aligned sequences. Above are motifs. The Model view shows additional information for a selected motif. Picture copied from Sandelin A et al (2004)

3 TFBS Viewer: Information Visualization

TFBS Viewer is a implementation of Information Visualization, that takes XML sequences (with nodes TFBS predictions – motifs) as input, and then visualizes TFBS predictions using a parallel coordinates system. For the remainder, this paper will refer to each individual XML input file as a separate sequence, as it represents a single individual's DNA sequence. The implementation makes use of the 1 dimensional properties of DNA sequences as a layout, for a simple, symbolic representation of the motifs. Edges between the sequences represent common motifs across multiple sequences.

Sequences can be moved around the layout, and any number of sequences can be added or removed from the set (memory and space permitting). This provides a dynamic, user defined, alignment tool. The whole system runs at a decent place, and does not lag to inefficiency even when using large datasets.

TFBS Viewer also provides some simple, real-time, information of motifs. Information of individual motifs can be retrieved and highlighted. Users can also search through the list of families and highlight these families.

3.1 Parallel Coordinates

Parallel coordinates [Wegman EJ 1990] is a method for a 2 dimensional representation of high-dimensional data. The attributes are listed on the horizontal axis (or vertical axis if rotated). A set of edges from one attribute axis to another represent the individual entries in the dataset.

TFBS Viewer can sequentially load any (limited by memory) number of sequences into the display. Each file is read as an XML database, and each individual node represents an individual motif in the file. When two sequences are place side-by-side, all their motifs with the same family are connected with a darkened edge, while those that are farther away in proximity have lightly shaded edges (implementation of a dynamic alignment).

The parallel coordinates system provides some excellent features, that can be taken advantage of:

1. New attributes of a parallel coordinate system are easily added to a visualization. In the TFBS Viewer, this means that species, represented as a sequence, can be easily added to the visualization. The other sequences do not have to be touched or manipulated, nor do their locations on the display have to be moved. The advantages of this is that sequences can be added to the visualization without the user losing his sense of direction in the other sequences. As a bonus, sequences can be easily copied and compared against themselves. One sequences, with a duplicated subsequence, can be placed side-by-side, and compared. Optionally, one sequence can be compared to two locations of another sequences (although the parallel coordinates limits the number of comparisons to two).

2. The best way to find significant motifs is to find multiple, highly confident motifs together. The parallel coordinates system conserves linear spatial properties of a sequence. For the TFBS viewer, this means that if a pattern of motifs, with a constant distance in-between, is found in multiple sequences, a users confidence level for these motifs can rise significantly. This also works for long distant sequences. A user could find the presence of conserved patterns of motifs in multiple locations on a parallel sequence.

3. The number of attributes displayed on a parallel coordinate system is limited only by the amount of available space vertically. This implemented very well with the TFBS viewer. Multiple sequences can be easily viewed (in my experience, 12 sequences was still comprehensive) without lose of information displayed from clutter.

4. Simplicity, the parallel coordinates system is easy to understand and check. The target user audience for TFBS Viewer are users with a broad understanding of biology, but might have only a little experience with graph theory. It would be much desired to have a system that has a small learning curve.

3.2 Color-Brightness as Filter

Many visualization systems have color systems implemented. These systems can be well implemented on small datasets, where color can be easily associated with a discrete number of labels. It is well implemented in the TFBS Viewer for a small number of motifs. As the number gets larger, the significance of the colors fades. This is immediately apparent in the MatInspector tool, where color becomes distracting. Some visualizations (such as the Consite) leave color out all together when dealing with larger datasets.

I've implemented a darker coloring scheme for the motifs, so that the color doesn't get distracting, but is available if color is desired for the implementation as a labeling device. As the dataset get into the thousands, this labeling visualization becomes less useful, and can be ignored.

As the mouse passes over edges, motifs and clicks on the family references on the side (east) panel, the corresponding family of motifs gets highlighted. Highlighted motifs and edges appear in a bright color (magenta), a color that is significantly brighter the the rest of the dull motifs and edges. It might have been desirable to implement a "prioritizer"; a method that increases the priority (read: brightness) of a selected motif. This could be used to easily save desirable motifs. Because of the number of dull colors used, these prioritized motifs would be recognizable at first glance, for any dataset.

The second implementation of brightness is implemented using the edges between a pair of sequences. As the two nodes neighboring an edge get closer (and consequently, the edge becomes more perpendicular to the sequences), the edge gets darker (become black when it is fully perpendicular to the sequences). This becomes an aid for pattern searches. If many such edges becoming black on a single placement of the sequences, if means that there are many conserved (and more confident results of) motifs between those two sequences at that orientation.

3.3 Focus

Focus is another useful technique for larger datasets, to focus in on important sections of a graph, while leaving other areas in the background. This is an extremely useful tool for datasets that are easily categorized.

The implementation of TFBS viewer has a focus animation option. If desired, an edge of the display can be double-clicked, and the viewer with shift both sequences so that the edge is placed in the middle of the display, and lies perpendicular to both sequences. The transition has a slow-in-slow-out animation so that both sequences can be kept in context.

The motivation for implementing this is two fold: a) if a candidate edge, i.e. an edge that the user thinks has a high confidence value (determined empirically or otherwise), then it is nice to bring this edge into the center of the screen. b) if there are other nearby candidate motifs, each with conserved between the sequences, the focus will put each of their edges nearly perpendicular to the two sequences, thus highlighting them (see 3.2 Color-Brightness as filter). This functionality becomes a useful search tool.

3.4 Interactivity to Provide a Dynamic Alignment Tool

Interactivity is a powerful tool in Information Visualization, especially for large datasets, where confidence of a result is difficult to show. A simple static display might not be enough to show enough confidence in a result. Interactivity could solve this problem by giving a user the ability to navigate and manipulate the data until an ideal, or optimal representation is found.

Optimality in DNA tends to be found in small localities, rather then as a whole sequence. Running an alignment tool that finds global optimums might not be as desirable at a local alignment tool. Therefore, we propose a solution that allows users to find their own alignments, locally, that might be more similar to running a local alignment search then the global alignment search.

This is similar to searching for the near-optimal set. Although global optimal results are good useful, in practice, predicted TFBS datasets tend to behave more towards many near-optimal solutions that might be "better" results then the most optimal. This is true in many biological applications. For example, in the free energy model to predict the 3D structure of proteins, the optimal solution is many times computed, yet sometimes, is never seen in practices. This might be because "reaching" the optimal state might be very difficult.

Since the presence of many near-optimal configurations is a definite possibility, it would be very useful for a user to be able to navigate freely through each configuration. This is possible through an interactive environment that allows users to manipulate the number of sequences to view, alter their ordering, and then slide sequences of motifs horizontally. All of this can be done without the user losing their sense of direction. Even when the program starts to lag (for very large datasets), the program becomes slow, but still comprehensive.

4 HIGH-LEVEL IMPLEMENTATION

The majority of the TFBS Viewer was implemented in Java, using the Swing extension for interface design and organization, the Prefuse visualization toolbox for data organization, and Perl scripts to parse the data to XML format.

4.1 XML Dataset

HTML results were parsed using Perl scripts to make XML formated datasets. The datasets were constructed with the attributes described in Table 1.

Each of the attributes were used to develop a spatial layout that conserved location of a motif relative to its sequence, and kept individual sequences separate.

Table 1: Motif Attributes

Attribute	Description	Usage
id	Node ID number	Ignored
start	Motif start index	X position of motif
end	Motif end index	Ignored
species	Trans. Factor Species	Ignored
family	Motif Family	Clustered for highlighting
motif	Motif name and variation	Connected with edges
desc	Description of Trans. Factor	Ignored
matSim	Similarity Score of motif	Height of rendered node
seq	Binding Site sequence	Ignored
filename	Unique filename that motif was read from	Y positioning

4.2 Implementation of Prefuse

The Prefuse interactive information visualization [Heer J (2004)] has several new classes, methods and functions. VisualItems represent the visual structures that are painted onscreen, this includes Nodes, Graphs and Aggregated items. These VisualItems can be manipulated and customized easily, for example their rendering method can be extended or built separately. Each VisualItem has several boolean attributes that can decide its state, including visibility, focused or highlighted. Edges, Nodes and Aggregate Items can each be related to one another through appropriate function commends. Finally, each VisualItem is associated with a corresponding entity in the read graph.

The ItemRegistry is a repository for all the VisualItems, read graphs and displays. The registry is passed amongst the displays and actions performed, so that appropriate VisualItems and attributes can be manipulated as the states change.

The ActionLists contain a list of actions to perform. Appropriate actions include animation pacers, node and edge filters, display layouts, display panning and zooming, etc. ActionLists can be manipulated through timing functions, so that they provide smooth animation and visualizations.

ControlListeners are the interactive elements of an interface. These include mouse and key listeners that pass VisualItems as well as Mouse X and Y coordinates. They are important to any visualization that involved interaction.

In the following sections, I will explain how the TFBS Viewer differs from the standard or default implementation of Prefuse.

4.2.1 The SequenceItemRegistry class

The main implementation class of Prefuse is the ItemRegistry that stores the XML graphs, VisualItems, and displays. This class was extended to the SequenceItemRegistry class to include several other important variables needed for the TFBS Viewer. Some of the included features:

1. The order list contains the list of visible sequences, sorted by the filename attribute. By reordering this list, and then recalling the SequenceFilter and SequenceLayout action (described later), the sequences would be reordered.

2. The filename list contains all the loaded filenames. Files remain in memory even if all of their motifs are invisible, or removed. This allows for easy re-loading if needed. Unfortunately, this also reduces the available memory for subsequently loaded files.

3. Offsets stores the X coordinates for the entire sequence, and is loaded whenever the entire sequences if dragged to moved in any way.

4. FilterColorMap stored the color map for each motif family name. These colors are loaded when each node is painted. New colors are assigned randomly.

4.2.2 Dataset to VisualItems

Each individual dataset can be individually loaded, and read via the XMLGraphReader function. The newly created graph is concatenated to the existing graph (which resides in the ItemRegistry). Each item in the newly created graph is given the attribute of the filename. I've not allowed the same filename to be loaded more then once, which would create conflicting nodes attributes. Instead, included sets of motifs can be duplicated using the CopySequence ActionList.

Edges are automatically added to the graph using the SequenceFilter action. Edges are not necessary to be included in the XML file. Although they are not ignored, thus included Edges in the XML file should only be included if a user desires to include a special relationship between some motifs.

No Aggragate Items were used.

4.2.3 ActionLists

There are several new Actions include in this implementation of TFBS viewer. Some of the actions perform necessary tasks, such as the SequenceFilter that creates NodeItem representations of each of the motifs, and EdgeItem representations for each corresponding relationship. There is an AddSequence action that adds a new file to the graph. CopySequence and RemoveSequence respectively duplicated and remove existing sequences.

The animate actionlist is for animations that were included. The animations are all paced using the SlowInSlowOut pacer for smooth transitions. The update actionlist is a condensed repaint ActionList that does not create or update the number of NodeItem and EdgeItems. Instead it concentrates on color updates and repaint actions. Because the number of actions were minimized as much as possible, the update ActionList should be used almost exclusively for animation repainting.

4.3 Interface

This TFBS viewer demo uses the Java.swing toolbox for window implementation. There are three main divisions of the main JFrame: the main Display, the Control Panel and the Filter Panel.

The main display, named *Display*, displays the VisualItems in their current state. This Frame contains MouseListeners that listen and record where the mouse pointed is located. The HighlightSequence ControlListener waits for the mouse to hover over a VisualItem to highlight it. The VisualItem's color is highlighted, and the corresponding motif family is selected in the Filter Panel (see after).

If an EdgeItem is double clicked on, an animation is triggered that slides both Nodes neighboring the EdgeItem to the center of the display. The corresponding sequences slide with the NodeItems. If a NodeItem (a motif) is double clicked on, a new window pups up, that displays the motif's attributes.

Finally, a SequenceDragControl listens for a NodeItem drag. When a drag is recorded, then the entire sequence slides horizontally with the mouse.

The colors of motifs are determined randomly at creation. The darkness of an EdgeItem is determined depending on the neighboring NodeItem's X proximity. This is done as often as possible, so as to make dragged and animated sequences' EdgeItems change color are common motifs pass by each other.

The *Control Panel* controls several user trigger actions, including adding files, copying sequences and removing sequences. It also controls the ordering of sequences on the display. This ordering can be changed by using the MoveUp and MoveDown JButtons. Although there is no animation for the order change transitions, Users should not be confused by the new ordering, this is because all of the sequences were already on the screen.

The *Filter Panel* is a color changing panel that displays a listing of all the included motif family names. These family names can be selected to view the family highlighted on the display. Alternatively, when the mouse is passed over a sequence, its corresponding family is selected in the Filter Panel. The color of the family is displayed as a background of the Filter Panel.

5 RESULTS

There are two examples shown here of usage of the TFBS viewer on two datasets. It will provide a step by step usage of the various panels and options.

5.1 Test Data

We are using two example datasets. The raw DNA for each of the human (hg17 assembly), chimpanzee(assembly), mouse (), rat (assembly) and dog (assembly July 2004) sequences were extracted from UCSC's genome browser [Kent WJ et al. (2002) - http://genome.ucsc.edu/]. Predicted motifs were obtained using MatInspector (their algorithm for motif prediction works very well). The first dataset uses sequences of size approximately 1000bp. The second uses sequences of size approximately 6700bp.

The raw HTML files were parsed into XML files using a Perl parser script.

5.2 Example 1: Loading and Altering the Sequences

Figure 3 shows the initial setup for the Sequences. We see that 4 sequences have already been loaded using the argument: ¨./data/results/human_chr7.xml ./data/results/chimp_chr7.xml ./data/results/mouse_chr7.xml ./data/results/rat_chr7.xml¨. We can add an additional file using the add file button. By adding the file ./data/results/dog_chr7.xml we obtain the results as shown in figure 4.

We can play around by dragging the sequences. We have copied the human sequence using the JButton Copy, and moved it directly below the chimpanzee sequence (figure 5). One can move a selected sequence by using the JButtons Move Up and Move Down. We can immediately notice that the chimpanzee and human sequences are very common. There are several nodes that match up perfectly.

Now we remove the mouse, rat and dog sequences using the JButton Remove. We are left with two copies of the human sequence, and one copy of the chimpanzee sequences. For a comparison, we can observe that the human and chimpanzee sequences are not that similar when they are not perfectly aligned. Lets move the human sequence to the left by dragging the sequences. At once, we can see that many black edges fade away to grey with the movement.

If we'd like to perform further analysis of each individual motif, we can double click on it to bring up a feedback window. This window displays most of the attributes on the motif (see figure 7). We can also search through the alignment of the human and chimpanzee sequences by double clicking and centering individual edges. Such as what we did for the highlighted edge in figure 8.

5.3 Example 2: Finding Patterns

With almost a thousand nodes, this datasets of figure 9 and 10 are quite large. We can still find conserved groupings of motifs. This was performed on by dragging the mouse sequence of figure 9 to the right to get to figure 10. We see right away large *darkened* sets of edges. This represents groups of conserved patterns that might be important. The large the set of nodes, the more confident we become. The height of each individual motif represents the motif similarity score. If the motifs are also tall, we have found a set of interesting sites.

6 Strengths

Although there already exists some visualizations for TFBS datasets, they are very limited in their ability to provide two excellent visualization techniques: interactivity and multiple sequence alignments.

Multiple sequence alignments is the ability of a user to place multiple sequences side-by-side, and view where they are able to match up (where they are conserved between themselves). This technique, applied to TFBS prediction datasets, was used to view several distantly related sequences. This is a technique that has aided comparative geneticists. This implementation of TFBS viewer allows users to visualize several sequences side-by-side, without much clutter in between these sequences.

An interactive environment, including the ability to scan, move and slide sequences around each other is a big help. With real-time feed back on the alignment, this tool can help users quickly find alternatively conserved regions. The ability to copy sequences, reorder and add new files allows users non-stop customizability.

The real-time feedback for conserved regions works very well, and doesn't cause very much lag. Users don't have to worry very much about lag producing false or ghost results.

The real-time highlighting tool gives fast, quick, feedback for the motif's family. Further information can be extracted by double-clicking on the desired motif. This produces a new window, that doesn't need to be closed. That information will stay visible for as long as desired.

For large datasets, I have often found that color can either be used sparingly, or not at all. When color is relied upon, I find that it becomes more distracting then helpful. I'm fairly content with the color scheme used for the TFBS Viewer. The dull colors are forgotten ignored until a color legend is needed. Whereupon the color can actually become useful. The magenta highlights really stand out from the rest of the visualization, since they are the single most important and only bright object on the screen. I count the color as a success.

Although on first observation, the dataset looks very occluded. This is not always the case, and the important information (mainly the darkened EdgeItems) are easily seen. I have also avoided the use of text on the screen, which would add unnecessary clutter to the already overloaded display.

7 Weaknesses

There are still some minor (major?) bugs that have not been fixed. The ItemComparator does not compare based on whether an Item is highlighted or not, despite my best efforts to make it so. Secondly, highlighting overrides the visual visible property of edges, so that edges that run beyond adjacent parallel coordinates (and which correctly set to invisible by the SequenceFilter) are suddenly highlighted incorrectly.

Clutter is the main weakness of the TFBS Viewer. Because of the large datasets used, there are few good and successful approaches to this problem. The issue was place in the background for most of the project, as a strong implementation to form a comparison across sequences was deemed more important. Clutter should be addressed, however, and is a necessary obstacle to overcome if individual motif confidence levels need to be investigated.

Another weakness becomes evident if one wanted to include huge datasets. The reason for this is that Prefuse limits the number of nodes to $10'000$. Although this seems like a large number, there were greater then $3'000$ nodes in example on figure 9 and figure 10.

8 Future Work

8.1 Edge Filtering

Although this TFBS Viewer contains a load of helpful features, there are still many more that could be implemented. Clutter constantly remains a weakness in all large datasets. With motifs, there are a hundred and one ways to filter out data. It would have been great to implement them all, but due to time constraints, there were only a limited number that could be implemented.

The ability to filter out completely the edges that are long distance (instead of simply reducing their brightness) could be a strong aid for some users. This could have been implemented using a sliding bar. Edges are filtered out if they exceed the bar's distance.

8.2 Motif Occlusion

Motif as often occluded, since predicted sites tend to be cluster together. This can be overcome in several ways. The first suggestion would be to change the NodeRendering function to a value that makes the motifs less occluded. If the ItemComparator function is changed, it would not reduce the number of occluded motifs, but simply present the most significant motifs on top.

Node occlusion could have been reduced using a simple zoom function, a spacing function that multiplies the X position by a floating point factor ¿ 1 for a zoom-in, and ¡ 1 for a zoom-out. This, unfortunately, reduces the viewable space. I have tried to avoid doing this.

A slider bar could be used to control the filtering of motifs based on their similarity score. This would be a simple solution that relies on the user to pick an appropriate filter value.

Finally, the ability to select and remember edges and motifs could be significant. This could easily be implemented using MouseClickListener function, and store the list the ItemRegistry.

With the addition of several new filters, an option on the Control Panel turn the filters on and off should be implemented, for sanity's sake.

8.3 ToolTip controls

Further and readily available information could be returned using a tooltip control. This avoids the problem of adding additional text to the display, but gives users an added feedback control. The information displayed in the ToolTip could be editable, thereby reducing some of the clutter.

8.4 Aggregation

Although aggregation was not addressed in this paper, it could be a very helpful tool in reducing the clutter of motifs and as an aid in finding patterns. I leave this option open to interpretation, as I have little knowledge of how and where the user audience would understand the usage of aggregation.

8.5 Online Implementation

Most of the TFBS predictors are found online. Adding this implementation as an applet to be viewed by user online could be a huge benefit. Although this ability might be cumbersome to internet services, it could be widely appreciated by the global community. The implementation could also take advantage of of public online services, such as readily available Logo graphics of motifs, or download individual motifs from the TRANSFAC database.

8.6 User Study

Although this project is still far from being completed, it is at a point where feedback from the user target audience should be retrieved. Many trivial qualities for a Computer Scientist might be difficult concepts for some of the users.

Getting user feedback might dictate the future direction of the study.

9 Conclusion

Although there is a lot of biological data available on transcription factors, their DNA bindings sites, and their applications to medicine, microbiology and biochemistry, there still exists many of the hard, and probably even some easy questions to tackle. Researchers that tend to scan large sets of DNA are constantly frustrated by the limitations imposed by traditional software.

Several researches have dedicated many hours to scanning through large TFBS prediction datasets looking for anomalies, hoping that their understanding can lead to another breakthrough. Much of the researcher's time could be reduced if they have a reliable search tool (many of these researchers tend to rely more on their own hands and eyes, then on algorithms that find optimal solutions).

TFBS Viewer can provide a simple browsing tool so that research time can be spent in a more efficient manner. Although TFBS Viewer can be significantly improved upon, to search through more specialized datasets, it should aid research for general datasets.

10 Bibliography

1. Kimura, M (1968). *Evolutionary rate at molecular level.* Nature, 217: 624-626.

2. Kimura, M (1986). *The neutral theory of molecular evolution.* Cambridge University Press.

3. Ohta, T (1992). *The nearly neutral theory of molecular evolution.* Annual Review of Ecology and Systematics, 23: 263-286.

4. Ohta, T. (2002). *Near-neutrality in evolution of genes and gene regulation.* Proceedings of the National Academy of Sciences, 99: 16134-16137.

5. Sandelin A, Alkema W, Engstrom P, Wasserman WW, Lenhard B (2004). *JASPAR: an open-access database for eukaryotic transcription factor binding profiles.* Nucleic Acids Res 32(1):D91-4.

6. Quandt K, Frech K, Karas H, Wingender E, Werner T (1995). *MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data.* Nucleic Acids Res. Dec 11;23(23):4878-84.

7. Cartharius K, Frech K, Grote K, Klocke B, Haltmeier M, Klingenhoff A, Frisch M, Bayerlein M, Werner T (2005). *MatInspector and beyond: promoter analysis based on transcription factor binding sites.* Bioinformatics 21, 2933-42

8. Sandelin A,Wasserman WW,Lenhard B (2004). *ConSite: web-based prediction of regulatory elements using cross-species comparison.* Nucleic Acids Res 32W249-52.

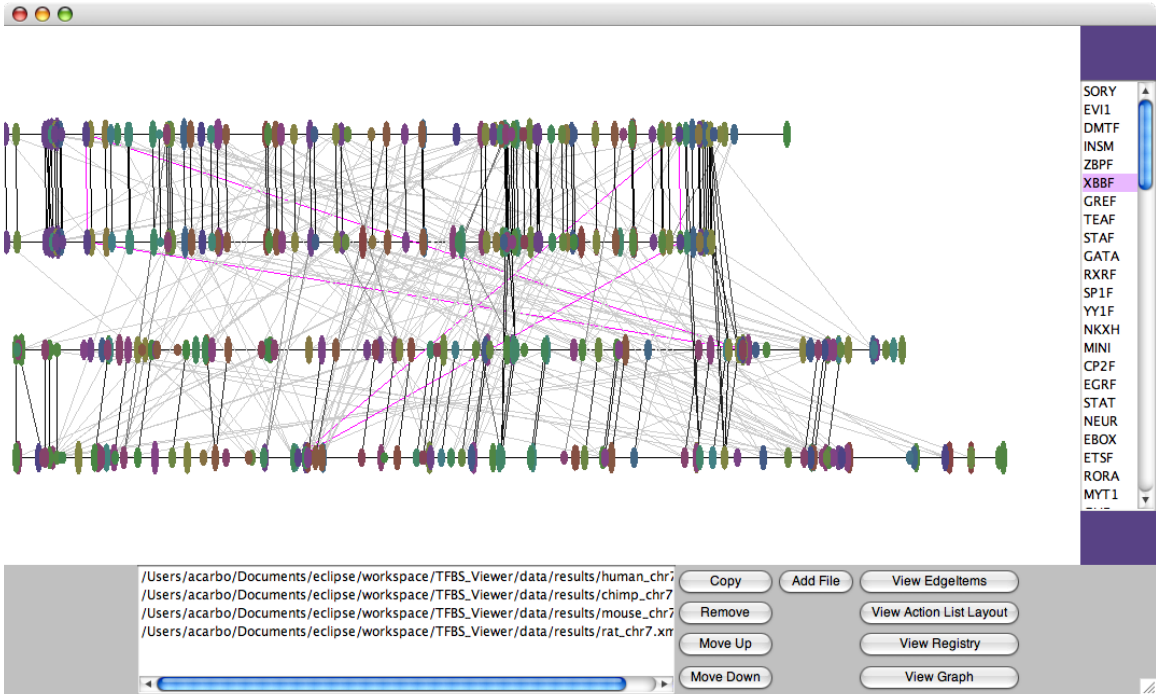
9. Loots G and Ovcharenko I (2004). *rVista 2.0: evolutionary analysis of transcription factor binding sites.*. Nucleic Acids Research, 32(Web Server Issue), W217-W221.

10. Wegman EJ (1990). *Hyperdimensional Data Analysis Using Parallel Coordinates*. Journal of the American Statistical Association, Vol. 85, No. 411., pp. 664-675

11. Heer J, Card SK, and Landay JA (2005). *Prefuse: a toolkit for interactive information visualization*. In CHI 2005, Human Factors in Computing Systems.

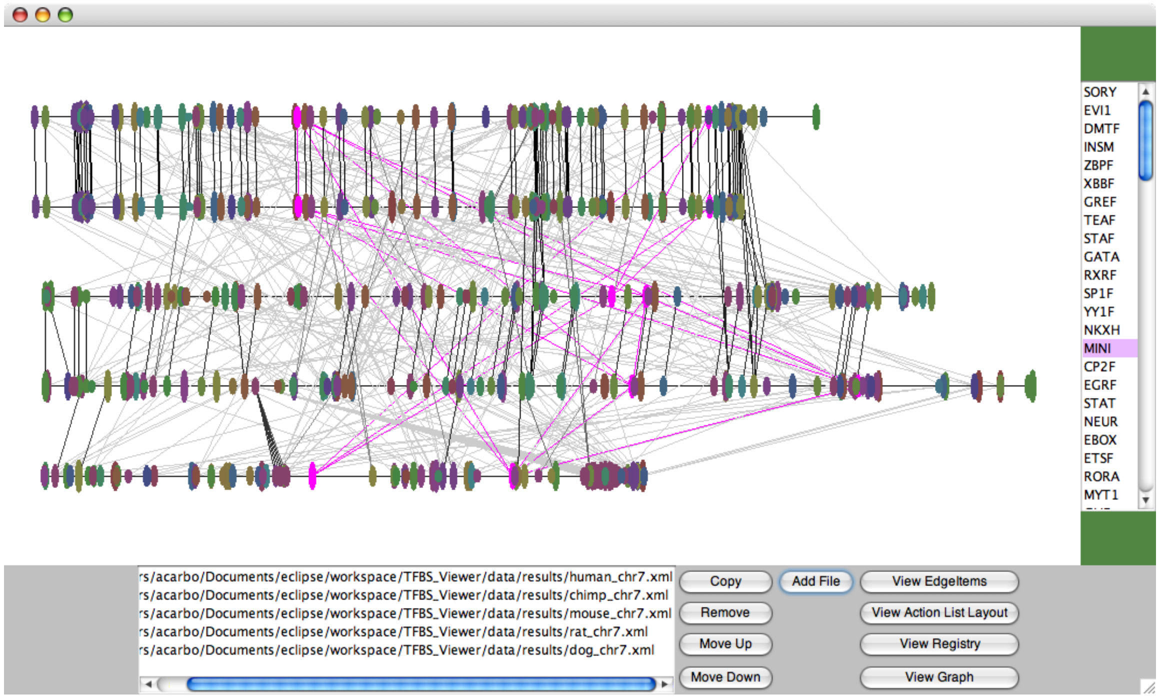
12. Heer J (2004). *Prefuse: a software framework for interactive information visualization*. Masters of Science, Computer Science Division, University of California, Berkeley.

13. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, and Haussler D (2002). *The Human Genome Browser at UCSC*. Genome Res. 12(6), 996-1006.



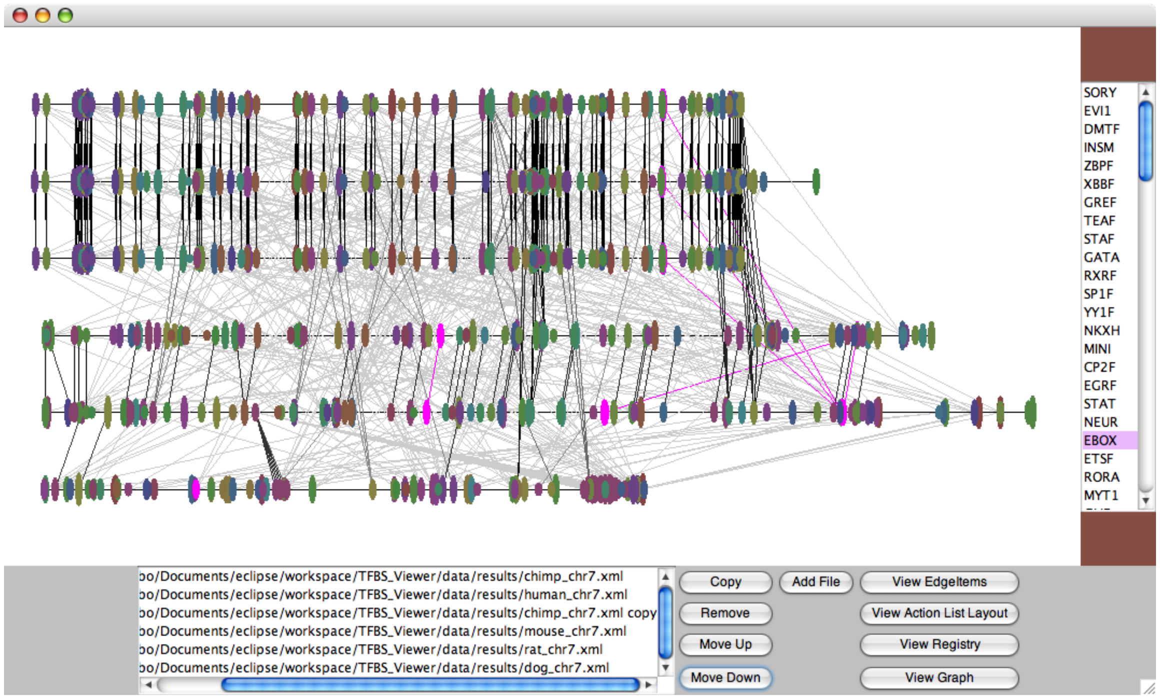
/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/human_chr7
/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7
/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/mouse_chr7
/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/rat_chr7.xm

Figure 3: Initial setup of Sequences



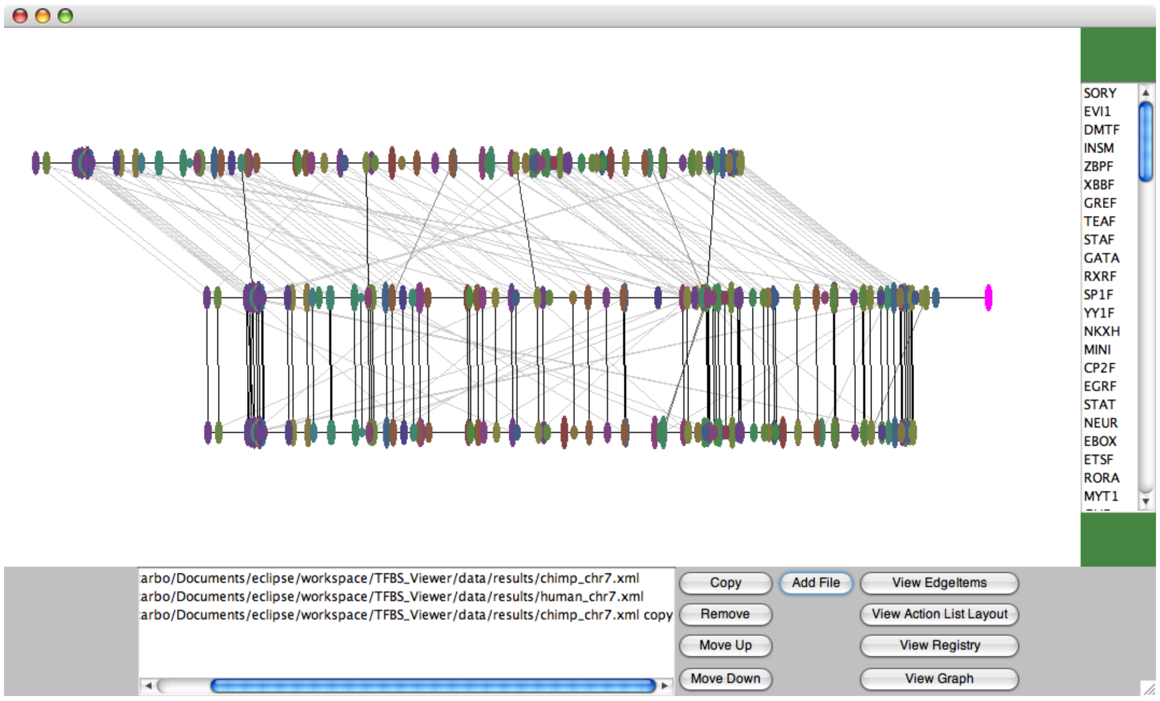
rs/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/human_chr7.xml
rs/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml
rs/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/mouse_chr7.xml
rs/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/rat_chr7.xml
rs/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/dog_chr7.xml

Figure 4: Add new file ":/data/results/dog_chr7.xml¨



```
oo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml
oo/Documents/eclipse/workspace/TFBS_Viewer/data/results/human_chr7.xml
oo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml copy
oo/Documents/eclipse/workspace/TFBS_Viewer/data/results/mouse_chr7.xml
oo/Documents/eclipse/workspace/TFBS_Viewer/data/results/rat_chr7.xml
oo/Documents/eclipse/workspace/TFBS_Viewer/data/results/dog_chr7.xml
```

Figure 5: Copied the Human sequence and reordered the sequences



```
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/human_chr7.xml
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml copy
```

Figure 6: Removed all but the human and chimp sequences, moved sequences

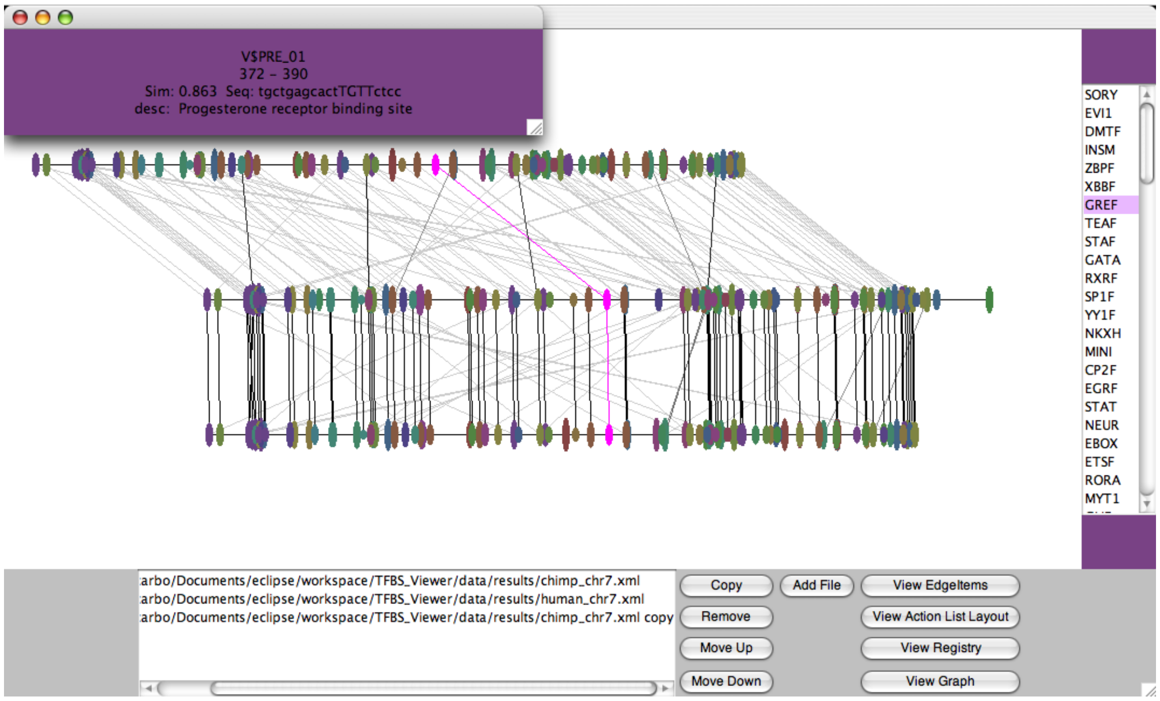

V$PRE_01
372 - 390
Sim: 0.863 Seq: tgctgagcactTGTTctcc
desc: Progesterone receptor binding site
SORY
EVI1
DMTF
INSM
ZBPF
XBBF
GREF
TEAF
STAF
GATA
RXRF
SP1F
YY1F
NKXH
MINI
CP2F
EGRF
STAT
NEUR
EBOX
ETSF
RORA
MYT1
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/human_chr7.xml
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml copy
Copy
Add File
View EdgeItems
Remove
View Action List Layout
Move Up
View Registry
Move Down
View Graph

Figure 7: Brought up feedback for motif V$PRE_01

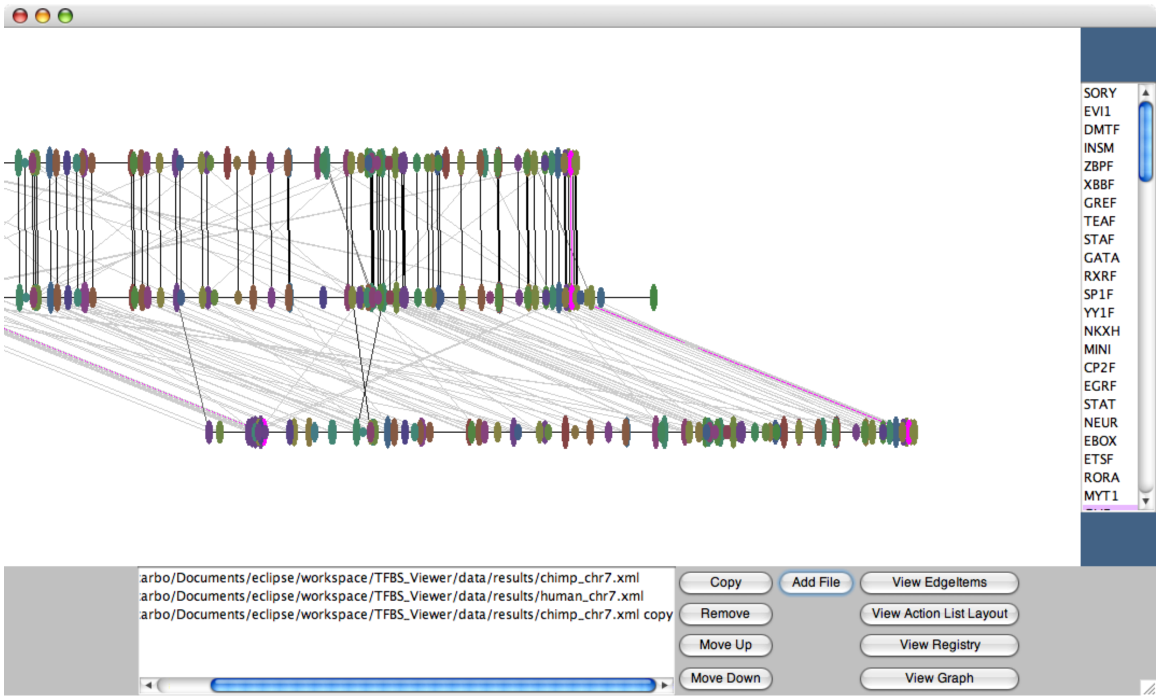
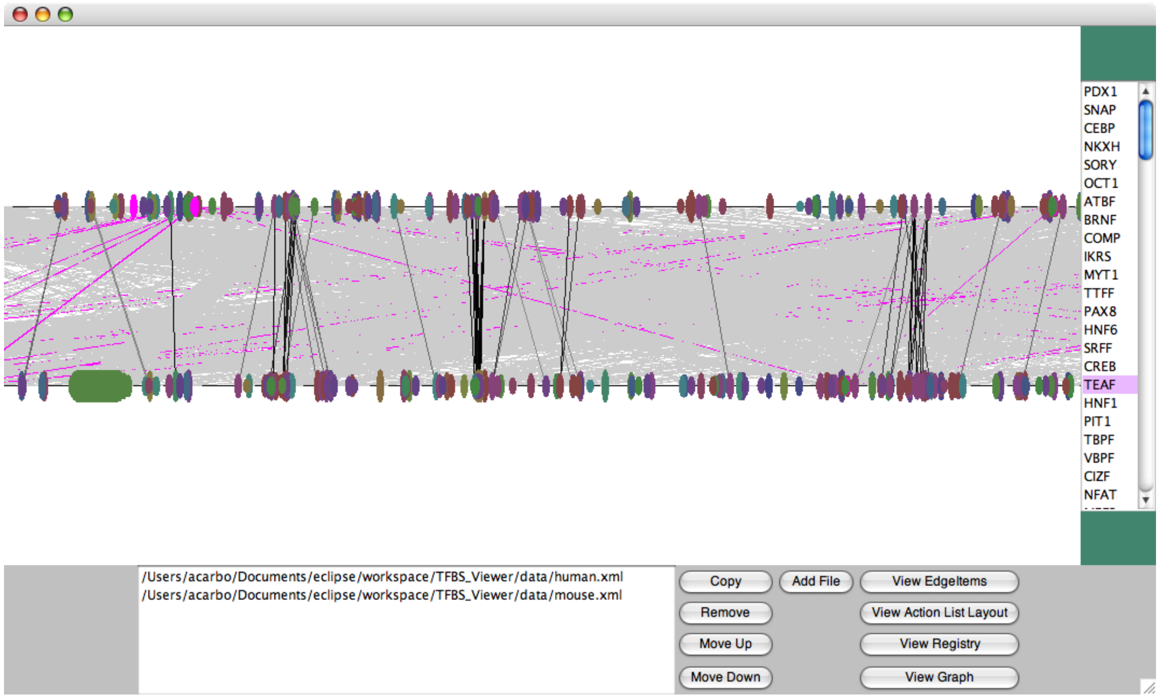
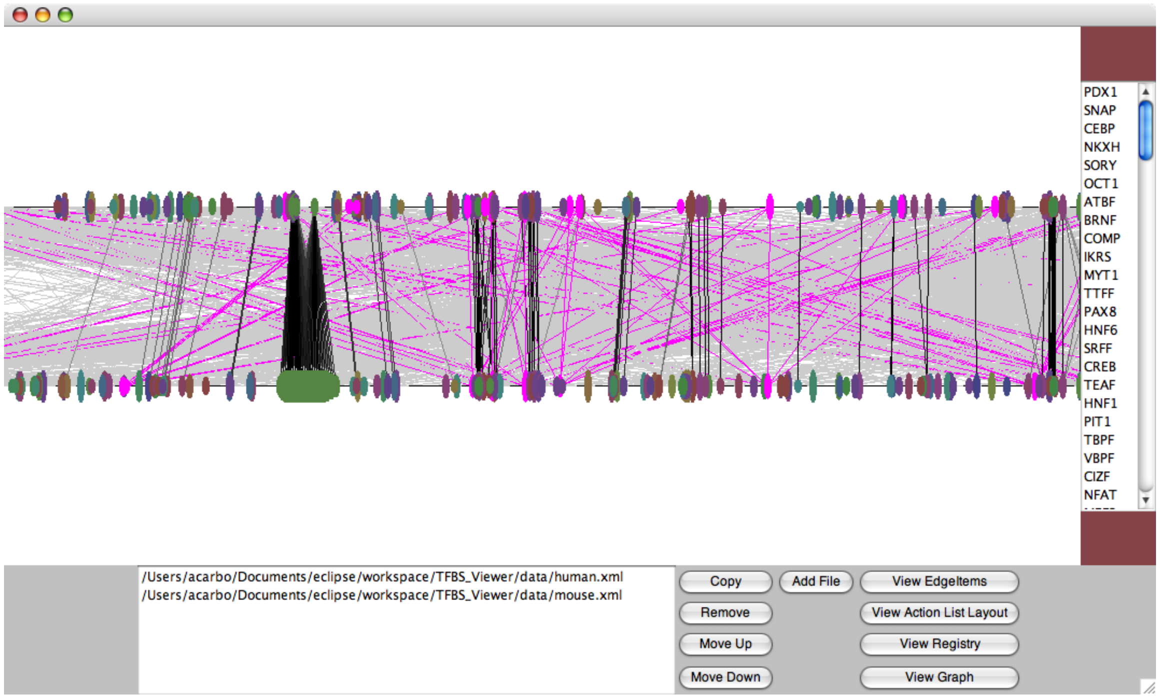

SORY
EVI1
DMTF
INSM
ZBPF
XBBF
GREF
TEAF
STAF
GATA
RXRF
SP1F
YY1F
NKXH
MINI
CP2F
EGRF
STAT
NEUR
EBOX
ETSF
RORA
MYT1
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/human_chr7.xml
arbo/Documents/eclipse/workspace/TFBS_Viewer/data/results/chimp_chr7.xml copy
Copy
Add File
View EdgeItems
Remove
View Action List Layout
Move Up
View Registry
Move Down
View Graph

Figure 8: Centered sequences around highlighted motif



/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/human.xml
/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/mouse.xml

Figure 9: Large human and mouse dataset



/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/human.xml
/Users/acarbo/Documents/eclipse/workspace/TFBS_Viewer/data/mouse.xml

Figure 10: New arrangement of sequences to get different matches