Visualizing Conserved Gene Location across Microbe Genomes

Chris D Shaw

School of Interactive Arts & Technology, Simon Fraser University, Surrey, BC, Canada

Abstract—This paper introduces an analysis-based zoomable visualization technique for displaying the location of genes across many related species of microbes. The purpose of this visualizatiuon is to enable a biologist to examine the layout of genes in the organism of interest with respect to the gene organization of related organisms. During the genomic annotation process, the ability to observe gene organization in common with previously annotated genomes can help a biologist better confirm the structure and function of newly analyzed microbe DNA sequences. We have developed a visualization and analysis tool that enables the biologist to observe and examine gene organization among genomes, in the context of the primary sequence of interest. This paper describes the visualization and analysis steps, and presents a case study using a number of *Rickettsia* genomes.

Index Terms— Bioinformatics, Sequence Analysis and Visualization, Visual Analytics

--- ✦ ---

1 INTRODUCTION

A newly sequenced Bacterial genome requires a significant amount of effort to determine the structure and function of its genes and related sequence products. Once genes are found, their function can often be determined by analyzing the similarity of the Gene's DNA sequence to similar sequences in the international public databases such as NCBI, EMBL or DDBJ. The standard interface to these databases is to provide a search facility with a type-in box in which the user may input a DNA sequence. In response to such a query, the researcher receives a report of the sequences in the remote database that are sufficiently similar to the query, which the user can then save into a local file collection. Such web-based search and analysis services have the advantage of portability and convenient access, but with the disadvantage of requiring the user to organize all of the piecemeal results.

The analytical process typically followed by the biologist is to divide and conquer. The genome is divided into a set of individual genes, and each gene is analyzed using the same modest pipeline of analytical steps [7, 13,18]. The review articles by Stein [18] and Lee et al. [13] explain the following methods in more detail. In some cases, this pipeline can be semi-automated, but the drawback of this approach is that the separation of analytical results gene by gene implicitly obscures the relationships that may exist between genes in close neighbourhoods of DNA within the genome.

We have developed a system that is aimed at smoothing the analytical task of analyzing biological sequences. IMAS (Interactive Multigenomic Analysis System) [16] combines sequence analysis tools such as Glimmer, Blast, and Clustal-W into a visualization and analysis workbench that enables rapid analysis of genomic sequences without the heretofore standard difficulties of having to manage a large number of printed outputs from websites.

This paper introduces a new analysis-based visualization technique that enables a researcher to analyze higher-level organization of related bacterial genomes. Although the primary unit of inheritance and corresponding biochemical activity in a genome is the individual gene, in microbes there is often a significant amount of coordinated activity among genes that are part of a closely related neighborhood of genes. It is frequently the case that genes are expressed (activated) together in a coordinated unit of expression called an *Operon* [2], which consists of regulatory and protein-coding regions co-located along a contiguous strand of DNA. The observation that neighboring genes A,B, and C occur in the same order in many genomes can thus be used to confirm that putative genes in such a neighborhood are real, even though some of the genes in the neighborhood may have unknown function.

The traditional divide-and-conquer approach to gene analysis that analyzes genes individually outside of their DNA context makes it difficult to observe consistent organization among genomes. Thus, we have developed a visualization and analysis tool that enables the biologist to observe and examine gene organization among genomes, in the context of the primary sequence of interest.

1.1 Analyzing Microbe DNA Sequences

The primary bioinformatic method of gaining knowledge about a DNA sequence is to compare sequences of biomolecules such as DNA or Amino Acid chains for *similar* sequences, since similar sequences of DNA tend to be translated by the organism into similar Amino Acid (AA) sequences, which in turn fold themselves into proteins with similar or identical function. So, finding a sufficiently similar DNA sequence in an organism's DNA to some known gene/protein allows one to infer that the resulting protein exists in the organism, and thus probably has some role to play. In a microbe, replicating DNA takes a nontrivial amount to energy and materials, so the presence of a gene likely means that it is being used, since otherwise the microbe would be wasting resources.

To start this analysis process, DNA sequences are generated by DNA sequencing hardware in segments of 500 nucleotides (DNA molecules) each. Many overlapping nucleotide (NT) segments are then assembled together into a draft sequence. The genome analysis pipeline that follows typically has at least the following major analytical steps.

First, the researcher will identify various genetic markers that are known to be present in the DNA sequence. These are short

DNA sequences that have been previously found in the organism and need to be located within the newly assembled sequence [18].

Second, the researcher must find genes within NT sequence using a gene finder such as Glimmer [6]. These gene finders identify the *Coding Sequence* (CDS) of nucleotides that are transcribed and translated into a corresponding Amino Acid (AA) sequence. The resulting chain of AAs folds itself into a biologically active protein molecule.

Third, each gene (CDS) is analyzed for similarity with other known NT or AA sequences. An organism will transcribe and translate an NT sequence into an AA sequence using a well-known translation scheme, so the process of determining the AA sequence is simply one of stepping through the NT sequence 3 NTs at a time and looking up the NT triplet in the translation table to generate each corresponding AA value. The function of a putative gene can then be inferred by analyzing its NT sequence or AA sequence for similarity with other known sequences. The international public databases (NCBI, EMBL, DDBJ) contain NT and AA sequence information that has been discovered by biologists around the world. Searching for similar sequences in these databases is typically done by running a similarity search and alignment tool called BLAST [1], which takes an NT sequence or an AA sequence query and searches for the most similar sequences in its database. BLAST will also run on specially-formatted sequence databases on a local computer.

In some cases, the NT alignment query to BLAST may yield nothing sufficiently similar, while the translated AA sequence yields a significant alignment. In this situation, the DNA sequence has evolved a lot while the protein sequences they encode have been largely conserved. Thus, both NT and AA similarity search is useful.

BLAST returns a list of similar sequences that it found, each with a pairwise string alignment between the query and the found sequence. This detailed alignment is of significant interest because the alignment may or may not be of uniform similarity along its length. In about half of the cases, the similarity between the query and the best-aligned sequence found by BLAST is so good that there is not much new to be learned from further analysis.

For the genes that are not obviously similar to genes in the public databases, a fourth major analysis step is to multialign the query sequence with other possibly similar sequences. Typically, BLAST will have found that a fragment of the query is similar to a fragment of one or more sequences in the database, and multialignment is aimed at discovering if key regions of the query sequence are similar enough to discover similar function.

Other analyses that the biologists may wish to take are to identify repetitive elements such as long stretches of the same few NTs repeated many times; identify duplicated genes within the DNA sequence; identify genes that have Single Nucleotide Polymorphisms (SNPs) [18].

The analytical process outlined above follows a divide and conquer approach to genome analysis. Each putative gene is taken in isolation and analyzed independently of the other genes in the organism, both the primary organism under consideration, and the other organisms that may be discovered to have similar sequence information. The visual analysis we are proposing in this paper takes the approach of analyzing genes at the NT neighbourhood level, where neighbouring genes are analysed for similarity of sequence as well as similarity of location. Since small collections of genes are often transcribed and translated together, expressing a few related proteins, this analysis can help

the biologist infer function by virtue of the presence of related neighbouring genes.

The term *Synteny* is often used to denote the situation where gene order and location is conserved from one organism's genome to another. That is, the genes are conserved across species, and their neighbourhoods are also conserved.

2   VISUALIZING GENE NEIGHBORHOODS

A well-established method for displaying similar gene organization across genomes is to use the spatial layout of the genes along a chromosome of primary interest, and display similarity links to one or two secondary genomes of interest.

A bottom-up approach is to use the Dot Plot, which is a scatterplot technique that displays a sequence along the X and a compared sequence along the Y axis. A dot is drawn where there is sufficient sequence similarity between the two corresponding horizontal and vertical NT fragments. This allows one to infer structure, but does not typically encode meaning at the gene level. Another drawback of this technique is that dotplots can become quite cluttered in genomes with many repeated elements (identical fragments of DNA sequence occurring in many places in the genome).

GeneOrder 3.0 [5] extends this idea by comparing *genes* across genomes, enabling the user to infer structure more clearly. That is, instead of using genomic DNA to drive a dot plot, similarity is computed on pairs of proteins sequences, and a mark is plotted based on the level of similarity.

In SyntenyVista [9], Hunt et al developed methods of displaying similar genes and other loci between chromosomes of human, rat, and mouse. The overview shows a primary organism with each chromosome assigned a unique color, and the secondary chromosomes are displayed with corresponding color patches. As single secondary chromosome may have multiple color patches from the primary genome, indicating that genes have moved from chromosome to chromosome. As the user drills down, pairs of chromosomes are lined up vertically, with line segments drawn between each to show similarity.

Cinteny [17] uses a similar color-coding approach to presenting an overview display of similarity between chromosomes, and a line-segment linkage for similarity at higher levels of detail.

SynBrowse [15] presents genomes in parallel horizontal tracks, with similarity linkages from genome to genome drawn as line segments when zoomed out, and as pairs of line segments showing the range of similarity when zoomed in.

Apollo [14] is a sequence annotation editor, which enables users to view genomic data in a zoomable interface and view the text content of annotations in pop-up windows. Users can connect to web-based services by clicking on menus that open a web browser to the desired page. Users then communicate findings to and from the web site via cut and paste. Apollo has a synteny viewer that allows the user to view links between genes similar to Cinteny and SynBrowse.

ACT, the Artemis Comparison Tool [4] displays synteny using a similar visual approach to Apollo, using a quadrilateral to link pairs of similar genes, and displaying identity value with color tone.

YGOB, the Yeast Gene Order Browser [3] displays similarities among yeast genomes, where the sequence is laid out horizontally with an equal-sized box per gene. The user focuses on a particular gene of interest and its 6 neighbours to the left and right are drawn in sequence. This establishes a grid in which each related organism is assigned its own horizontal track, and in each vertical slice is placed the corresponding gene to the primary

genes. Only the primary organism's genome is presented in order, so similar secondary genes are colored according to their source chromosome.

Whole-genome browsers like the UCSC genome browser [10] and others tend to have a similar organizational idea. The primary sequence creates the order, and the secondary sequences are presented in the same order as the primary. In essence, the secondary genomes are chopped up in order to allow the user to examine similarity in detail, with the trade-off that organizational information is therefore not shown.

The primary advantage of this spatial layout is that the ordering and size relationships can be directly visualized (with the exception of YGOB). If the secondary sequence is presented in primary order, the assumption is that gene order is not as important as the direct examination of sequence alignment. For the Synteny viewers like SyntenyVista, Cinteny and SynBrowse, secondary order is important, so similarity linkage is drawn using color patches or line segments or a combination of both. For a pair of genomes, this works well, but it does not scale well to multiple genomes. Also, reversals in gene order cause an unfortunate crossing of many linkage line segments, which can be very hard to follow, except as an overview. The DotPlot handles this issue, but only works for a pair of genomes.

3 ANALYZING SYNTENY IN CONTEXT

The analytical approach we take to visualizing synteny between related genomes is as follows. First, the user runs IMAS to analyze and visualize a *Primary Sequence*. This is the sequence that the user is primarily investigating to determine structure and function. The user runs Glimmer 3.02 to discover genes in the sequence, which are anchors for further analysis. The user can run Blast queries from IMAS to trigger sequence search against the NSCI NonRedundant (nr) database, or against local BLAST databases that have been pre-formatted. The drawback of this technique is that BLAST returns sequence alignments with an alignment and a database key, which is difficult to transform into a query that will reliably generate access to sequence data in its genomic context.

Instead, we have developed a *Secondary Sequence* function, which takes a complete annotated sequence data file from NCBI in ASN.1 format and transforms it into a set of BLAST-accessible databases for both NT and AA BLAST queries. The secondary sequence is also written in IMAS's own internal XML format so that it can be analyzed and visualized within the system. Having a Secondary Sequence allows IMAS to reliably determine what neighbor genes may exist to genes found by BLAST. The user can then link the appropriate BLAST-aligned secondary genes to the primary sequence. We have found that the Secondary Sequence notion is useful to maintain control over the data collection of the secondary sequence, since processing by IMAS guarantees that identities of genes and other sequence parts can be accurately maintained. The former approach of trying to guarantee that the header lines of FASTA-format data files input to BLAST be maintained with a clean nomenclature simply proved to be too much wasted effort.

To generate the information required to analyze similar gene organization, the user may run a BLAST query on each gene in the primary sequence against each of the secondary sequences. This generates a BLAST report for each secondary sequence that was found aligned below the querying primary gene. When the user is interested in examining gene organization between genomes, the user selects each of the Secondary Sequences that they wish to compare and displays their genes below the Primary sequence's genes.

IMAS enables the biologist to identify Orthologs [11] in one of two ways. First, the biologist can examine the BLAST pairwise alignments and select the BLAST alignment that is judged to be the highest quality (usually, the longest alignment with the highest identity). IMAS provides a menu item on each BLAST alignment to link a Secondary gene to the approriate Primary gene. The metadata in each BLAST alignment is used to link the correct Primary and Secondary genes.
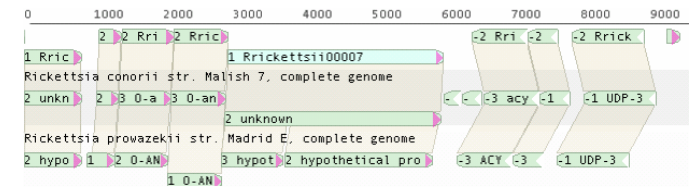

Figure 1. Visual display of common gene organization in *Rickettsia Rickettsii, Rickettsia Conorii* and *Rickettsia Prowazekii*. The top line of the display shows the NT index. The next pair of rows with the white background contain the genes of *R.Rickettsii*, with the remaining organisms arranged below. The orange link marks indicate the range if sequence similarity and the fact that the genes in *R.Conorii* and *R.Prowazekii* are each linked to the Primary sequence.

IMAS also provides a facility to compute the Reciprocal BLAST Hit on a Primary gene. The selected Primary gene runs BLAST on the Secondary sequence database, and the best-scoring BLAST-aligned secondary gene is in turn used as a query into the Primary sequence's genes. If the triggering Primary Gene is found with this Secondary query then a high-quality Ortholog is deemed to have been discovered [11,19]. The drawback of RBH technique is that it will find both false positives and negatives, which means that this automated process will need to be examined by hand for accuracy.

In the genomes discussed the following case study, a particular kind of mistake that may be made by RBH is the failure to discover a gene fission/fusion event [11], where a gene in one organism splits into two closely-related neighbouring genes in a related organism. Our manual linkage approach can then be used to fix these cases.

3.1 Visual Design

Figure 1 shows the results of linking together thee genomes *Rickettsia Rickettsii* (the primary sequence) to two previously

annotated and characterized genomes *Rickettsia Conorii* and *Rickettsia Prowazekii*. The genes in the first 9000 NT are shown. Immediately evident in this picture is the strong alignment between the three organisms. However, *Rricketsii0007* has been linked to a single gene in *R.Conorii* and to two split genes in *R.Prowazekii*.

The visual design of this display required a certain amount of effort to balance the salience of the links with respect to the salience of the features (Genes) in the scene.

The goals of the linkage marks are as follows:
1) Display linkage between primary and secondary genes.
2) Display range of linkage between genes
3) Do not conflict visually with gene icons (green bars in Figure 1). That is, take less visual attention than the genes.
4) In the case where links cross, be able to distinguish the routes of crossing links.
5) Enable the distinction between Secondary genomes.

3

6) Display pairwise sequence similarity on link elements.

The first visual design, shown at the top of figure 2, uses simple black line segments to connect Primary genes to their secondary counterparts. This display had the problem of significant overdraw at the primary gene, and caused a problem with following linkages to the appropriate secondary target.

The next design iteration used a chaining approach, where the primary is linked to the secondary below, and then a link is drawn from the first secondary to the next, and so on. This significantly reduces difficulty in following links, although it has the drawback of suggesting that the similarity is between secondary genes instead of between Primary and Secondary. This redesign also draws a quadrilateral filled with a dull orange color, which is a hue that is not used elsewhere in IMAS.

The third design iteration sought to control the salience of the bounding line segments to the link polygons. The gene boxes have a 100% black outline, and they need to be more salient than the links, so the link outlines were weakened to a 25% darker orange color than the fill with a 60%.

The fourth design iteration dealt with the issue of being able to distinguish genomes from each other. This was accomplished by two related approaches: A title was placed above each Secondary sequence, and a contrasting background wash was used to frame each sequence. The background wash alternates between 100% grey (white) and 80% grey.

To make the background washes visible under the link polygons, the fifth design iteration changed the link fill to be 60% opaque. This also has the beneficial effect that links drawn on top of each other can be followed through each other.

The sixth design iteration tackled the issue of pairwise similarity. Since each similarity link has access to the pairwise BLAST alignment that triggered the linkage, we took the opportunity to optionally draw a Percent Identity Plot on the link polygons. The Percent Identity Plot scores the level of identity using a non-overlapping 3-character window for AA alignments that scores the level of similarity per character. Identical AA letters score 1, and similar letters score a smaller value. The IMAS BLAST visualization uses Percent Identity to visually score alignments, and we use the same approach to shading links. The idea is to draw a shaded vertical stripe whose transparency is controlled by the Percent Identity score. Higher scores get higher opacity, while low scores are more transparent.

Figure 2 Design iterations

3.2 Interaction

The basic interaction method provided by IMAS is the zoomable display along the horizontal dimension. Because the vertical dimension is already quite compact, and because the horizontal extent is somewhat larger than the vertical extent, IMAS continuously zooms along the horizontal dimension. As IMAS zooms out, sequence details are hidden as their available screen real estate shrinks to invisibility.

For the gene similarity links shown on this page, the amount of detail is quite low, so we have been showing example images at a zoom level of 1/100 or so.

Along highly similar genomes, the level of similarity is such that the display of similarity links is well-ordered and neat, and can be examined in detail and in overview with equal ease. However, when the similarity structure is more complex, the linkage shapes grow to be more and more oblique, and the detailed structure becomes somewhat harder to follow. To deal
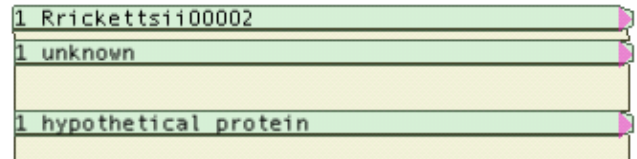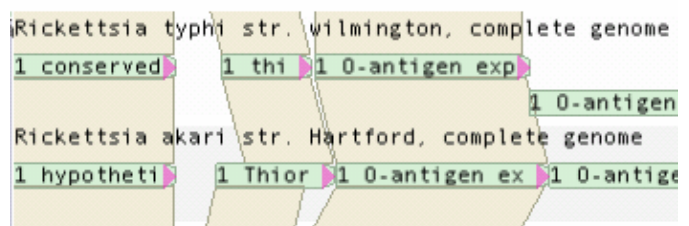

Iteration 1: Line segments
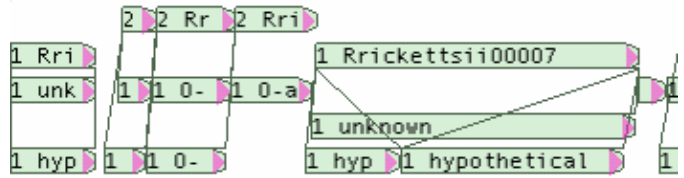
Iteration 2: Filled Link polygon chains
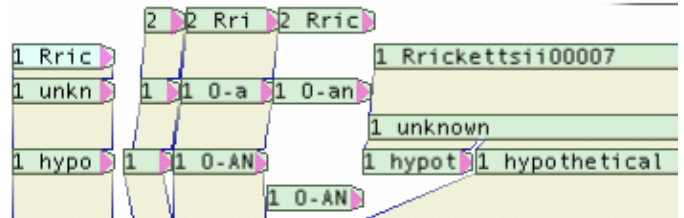
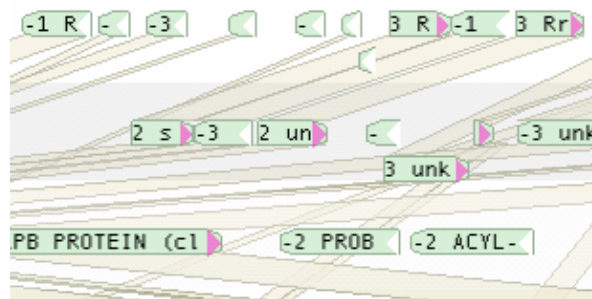Iteration 3: Semitransparent Link Edges
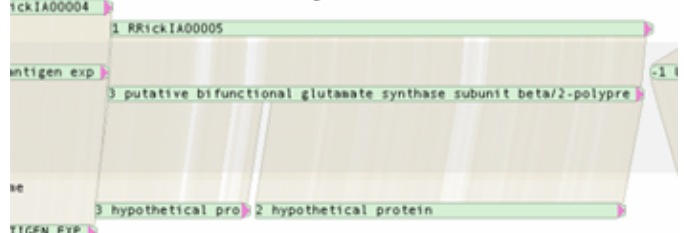
Iteration 4: Labels + Background Wash

Iteration 5: Semitransparent Links

Iteration 6: Similarity PIP Score on Links

with this issue, we provide a re-center operation that adjusts the horizontal location of each secondary genome to lie directly underneath the corresponding gene of the primary sequence. A user simply selects the primary gene of interest and selects the re-center menu item. All linked genomes are slid to the left or right to align the linked genes vertically.

For temporary adjustments, we also provide a mode that readjusts all secondary genomes on the fly in response to the current cursor location. When the user holds the ALT key and moves the cursor over the secondary sequences, they are automatically shifted to the left or right to align with the primary sequence. This approach is consistent with the technique we developed to visually shift pairwise sequence alignments to account for gaps in the alignment.

Kosara et al.'s Parallel Sets [12] visualization has a similar appearance to the IMAS Synteny display, except in their case, the individual data items are represented as members of a number of parallel categories. For example, people on the Titanic are passengers or crew, men or women, survived or not. Their system supports interaction techniques to reorder categories, merge categories, delete categories and highlight categories. To interpret the IMAS display according to Parallel Sets, each organism forms its own parallel organizational structure, and each gene may or may not appear in an organism. However, reorganization of genes cannot be done arbitrarily, since moving a gene would take it out of its genomic context.

IMAS enables a neater and more visually comprehensible Synteny display by analyzing the linkage relationships and visually gathering and reordering genes according to the order of the Primary sequence. This is logically equivalent to the reordering operation in Parallel Sets.

3.3 Reversals

As the case study in the next section illustrates, a common occurrence in comparing Bacterial genomes is a *Reversal*, in which a segment of DNA in one organism occurs in reverse order in a related genome. Figure 3 shows an example of this, in which *Rickettsia Akari* has genes 12-59 reversed with respect to the other genomes. While a large reversal like this is easy to see and understand, it has the disadvantage of being very hard to follow in detail, since many of the linked genes are by definition very far away from each other through most of the reversal. When this is coupled with horizontal shifts due to absent or extra putative genes in the Primary sequence, visual inspection of local linkage becomes very difficult. Moreover the shift-alignment interaction in the previous section does not help with reversals.

It turns out that such reversals can be detected automatically once each secondary genome has had its corresponding primary genes identified. We have developed an algorithm to detect and reverse segments of a Secondary sequence that are reversed with respect to the Primary sequence. This algorithm, based on those by Hannenhalli & Pevzner [8], compares the location and orientation of each secondary gene with respect to the primary gene, and seeks to compute a permutation of the secondary genome that puts the secondary genes in the same order and orientation as the primary genes. The algorithm also seeks to identify NT regions of the Secondary genome that should accompany any reversal, which means in practice that Secondary genes that have no corresponding Primary gene are included in any reversal of its neighbours. For example, the table shows a number of genes labelled by the Primary index number. The secondary sequence has a gene called "3.5" that does not exist in the primary. The orange highlighted region in the second row is reversed, with the result shown in the bottom row. Gene 3.5 goes

along with the reversal because its neighbours have been reversed.

Primary	1	2	3	4	5		6	7	8
Secondary	1		3	-6	-5	-4.5	-4	7	8

Primary	1	2	3	4		5	6	7	8
Secondary	1		3	4	4.5	5	6	7	8

The algorithm takes three phases. The first phase fills a table similar to the one shown above with all of the Primary genes in Primary order, along with its linked Secondary genes. Each secondary gene is given a positive sign if it is in the same reading direction as the Primary gene, and a negative sign if it is in the opposite reading direction. At this point, the table contains all of the primary genes, and most Secondary genes. Gene 2 in the table above is an example where there is no corresponding secondary gene.

The next pass inserts secondary genes into the table that are not linked to a Primary gene. Gene 5.5 is an example. Such secondary genes are inserted beside their nearest neighbor, and are given the same sign as their neighbor.

In the second phase, reversals are detected by searching through the array for *breakpoints* [8], which are points in the array where neighbouring secondary gene numbers are non-contiguous and change sign. There are two breakpoints in the table: one to the left and one to the right of the orange highlight. The number of breakpoints is indicative of the number of reversals that are needed to transform the secondary sequence into the order of the Primary. Thus, the goal of the sort-by-reversals algorithm is to search for a set of reversals that step-by-step reduces the number of breakpoints. Each reversal can reduce the breakpoint count by at most 2 [8], so the task is to find breakpoints that make progress towards the correct order.

Each breakpoint induces an *oriented pair*, which is a pair of sequential values that are not neighbours. There are two oriented pairs in the table: (3, -4) and (-6,7).

The second phase repeats the following steps:
1) Identify breakpoints and oriented pairs in the secondary permuted sequence.
2) For each oriented pair, compute a reversal that places the paired numbers next to each other.
   In the table, (-6, -5, -4.5, -4) -> (4, 4.5, 5, 6)
3) If the reversal yields zero breakpoints, we are done.
4) Otherwise, select the reversal that yields the *most* oriented pairs, and permute the secondary sequence with this reversal.

The algorithm makes progress because at each loop, an oriented pair range is flipped. This reduces breakpoints by at least 1, because the reversal moves the two non-contiguous genes next to each other and gives them the same sign. The idea behind selecting the reversal with the *most* oriented pairs is to make sure that the available search space for remaining reversals is as unconstrained as possible.

The output of the second phase is a series of reversals that are applied in order to the gene layout of the secondary sequence.

The third phase identifies gaps in the Primary sequence that have no corresponding unmatched genes in the secondary sequence. A "gap" permutation is created for these cases, which is simply a space filler that highlights a region of non-correspondence in the secondary sequence. The purpose of this is to automatically re-align secondary sequences that are missing genes that are present in the Primary sequence.

Each permutation is used to generate a visual container that holds the genes that it permutes, and displays it in the Secondary sequence's channel. Reversals are drawn with a light red background wash so that they can be distinguished visually from the non-reversed segments of the secondary sequence.

The first phase stores the set of genes in an array that contains the primary and secondary genes. The user may deactivate reversals in a region to get a visual indication of the real gene order situation, and my reactivate reversals if they wish to examine gene-by-gene linkage.

4 CASE STUDY

The IMAS gene organization visualization has been used to examine the similarity relationships among 5 closely-related species of *Rickettsia*: *Rickettsia Rickettsii*, *Rickettsia Prowazekii*. *Rickettsia Conorii*, *Rickettsia Typhi*, and *Rickettsia Akari*.
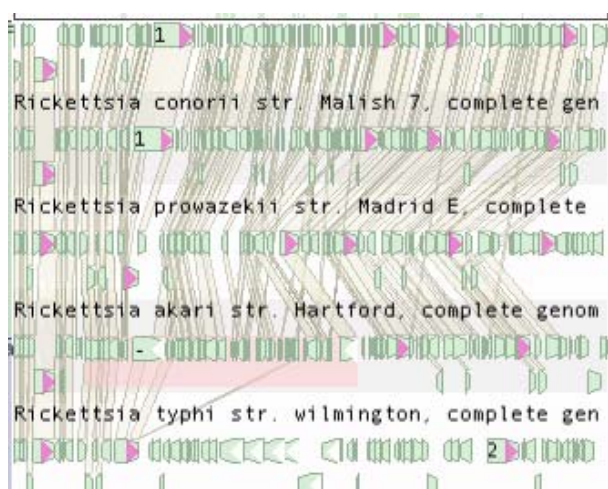

Figure 3. Five Rickettsial genomes at 1/2000 scale.

These organisms are bacteria that are the cause Rocky Mountain Spotted Fever, Typhus, Mediterranean Spotted Fever, Murine Typhus, and Rickettsialpox, respectively. These organisms have about 1.2 million nucleotides each, and a significant amount of similarity across their genomes. We have analyzed the first 85000NT of *R.Rickettsii* and displayed the genes occurring in first 85000NT of the other 4 genomes. In this subset, we have detected the closest alignment of each gene in *R.Rickettsii* to any gene in the first 85000NT of the other sequences. The following discussion is thus restricted to links found in these subsets.

Figure 3 shows an overview of the first 85000NT at a zoom factor of 1/2000. The top pared row of genes represents the Primary Sequence, *Rickettsia Rickettsii*. The system has detected a major reversal of about 40000NT in *Rickettsia Akari*, as denoted by the red wash in the 4th row. At this scale, most of the similarity links are reduced to only a few pixels in width. The other major points to notice in this visualization is that the *Rickettsia Prowazekii* is shorter, and has less genes than the other genomes, and *Rickettsia Typhi* shared genes in common only with the first 12 or so genes. There may be other similarity links to other parts of the Primary Sequence, but the 80000NT cutoff precludes detecting or showing that.
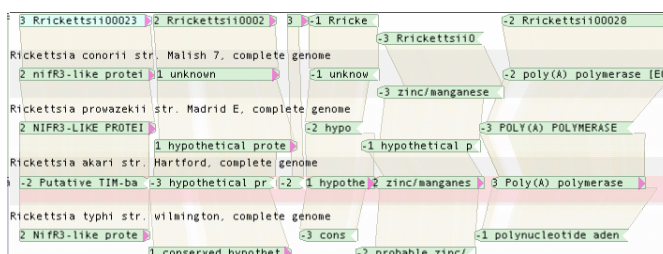

Figure 4. Five Rickettsial genomes at 1/50 scale. This is at the start of the *Rickettsia Akari* reversal.

Figure 4 shows the *Rickettsia Rickettsii* genes 23 through 28 ad about 1/50 scale. Interesting items to note are that the third gene from the left (Rrick0025) has no corresponding match in *R.Conorii, R.Prowazekii,* or *R.Typhi*. Also, the gene to the left (upstream) in *R.Prowazekii* and *R.Typhi* has a patch of low similarity, indicated by the transparent stripe in the links above these respective genes. Finally, the linked genes on the right (Rrick0028) each have a patch of low similarity at the left ends in *R.Prowazekii, R.Akari,* and *R.Typhi,* whereas this gene is quite similar in *R.Conorii.*

Figure 5 shows genes 4 through 7 inclusive. We have arranged the genomes in this image in order from most similar to least similar in this segment. The interesting thing to note is the "progressive dissimilarity" of gene Rrick007 from top bottom. The most similar is *R.Conorii,* next is *R.Typhi,* then *R.Akari,* then *R.Prowazekii.* This gene-aligned display shows that the regions of low similarity seem to be maintained between *R.Rickettsii* and these 4 other genomes. In fact, in *R.Prowazekii,* this gene has split into two parts, where the division point corresponds to the second patch of low similarity in the *R.Typhi* and *R.Akari* genes.
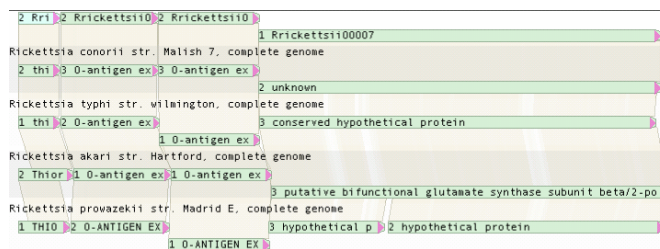

Figure 5. Five Rickettsial genomes. The gene corresponding gene to Rrickttsii0007in *Rickettsia Prowazekii* has been split into two genes (bottom right).

5 CONCLUSIONS

We have introduced an improved visualization and analysis system for the analysis of gene organization in microbial genomes. This system makes links between genes in the Primary Sequence of interest and Secondary sequences under analysis, reorders reversals detected in an aligned sequence, and displays these reversals in a coherent zoomable genomic display. The functionality described in this paper is part of the IMAS Interactive Multigenomic Analysis System, which presents a zoomable visualization and analysis framework for sequences analysis of microbes.

The visualization scheme presented in this paper makes careful use of color and background washes to present a coherent

view of the collection of genomes. Unlike the standard genomic analysis pipeline, where sequence analysis of necessity separates analysis gene-by-gene, this visualization and analysis method enables the biologist to analyze gene organization by drawing together the separate gene-based analysis into a coherent display.

A potential usage scenario for this visualization technique is to display the evolution of programming code, where the code could be organized along a horizontal axis, and major revisions are represented as new "organisms". The gene would correspond to an object's code. Major code refactoring events could be visualized in a similar way to gene fission/fusion events. An important aspect of this display is that local ordering of the sequence should not be disturbed. Any reorderings performed for the task of enhancing visual coherence must be un-doable so that the original order is preserved.

REFERENCES

[1] Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Research, 1997, 25(17):3389-3402.

[2] JM Berg, JL Tymoczko, L Stryer, "Biochemistry, 5th Ed.", W.H. Freeman and Company, New York, 2002.

[3] Byrne, K.P., and K.H. Wolfe, "Visualizing syntenic relationships among the hemiascomycetes with the Yeast Genome Order Browser", Nucleic Acids Research, Vol 34 Database issue, 2006, pp. D452-D455.

[4] Carver TJ, KM Rutherford, M Berriman, MA Rajandream, BG Barrell, and J Parkhill. "ACT: the Artemis comparison tool", Bioinformatics. 2005 Aug 15;21(16):3422-3.

[5] Celamkoti, S., S. Kundeti, A. Purkayashtha, R. Mazumderm C. Buck, D. Seto, "GeneOrder3.0: Software for comparing the order of genes in pairs of small bacterial genomes", BMC Bioinformatics, 2004, 5:52.

[6] Delcher, AL, D Harmon, S Kasif, O White and SL Salzberg, "Improved microbial gene identification with GLIMMER", Nuc.Acid.Rsrch, 1999, 27(23):4636-4641.

[7] Gibas, Cynthia, & Per Jambeck, Developing Bioinformatics Computer Skills, 2001, O'Reilly Press.

[8] Hannenhalli, S., and Pevzner, P.A, "Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals", Journal of the ACM, Vol. 46, No. 1, January 1999, pp. 1–27.

[9] Hunt. E., N. Hanlon, D.P. Leader, H. Bryce, and A. F. Dominaczak, "The visual language of synteny", OMICS, 2004 Winter;8(4):289-305

[10] Karolchik, D., R. Baertsch, M. Diekhans, T. S. Furey, A. Hinrichs, Y. T. Lu, K. M. Roskin, M. Schwartz, C. W. Sugnet, D. J. Thomas, R. J. Weber, D. Haussler and W. J. Kent. The UCSC Genome Browser Database. In Nucleic Acids Research, 2003, 31(1):51-54.

[11] Koonin, E.V., "Orthologs, Paralogs, and Evolutionary Genomics", Annual Review of Genetics 2005. 39:309–38.

[12] Kosara R., F. Bendix, and H. Hauser, "Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data", Trans. On Visualization and Computer Graphics, Jul 2006, 12(4):558-568.

[13] Lee D., O. Redfern, and C. Orengo, "Predicting protein function from sequence and structure", Nature Reviews Molecular Cell Biology, Dec. 2007, 8(12):995-1005.

[14] Lewis, SE, SMJ Searle, N Harris, M Gibson, V Iyer, J Richter, C Wiel, L Bayraktaroglu, E Birney, MA Crosby, JS Kaminker , BB Matthews, SE Prochnik, CD Smith, JL Tupy, GM Rubin, S Misra, CJ Mungall and ME Clamp, "Apollo: a sequence annotation editor", Genome Biology 2002, 3:research0082.1-0082.14, doi:10.1186/gb-2002-3-12-research0082

[15] Pan, X,, L. Stein, V. Brendel, "SynBrowse: a synteny browser for comparative sequence analysis", Bioinformatics21(7):3461-3468.

[16] Shaw., C.D., Gregory A. Dasch, and Marina E. Eremeeva, IEEE Symposium on Visual Analytics Science and Technology 2007 (VAST), Sacramento, CA, Oct 30-Nov 1, 2007, pp 59-66.

[17] Sinha, A.U., and J. Meller, "Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms". BMC Bioinformatics, 2007, 8:82.

[18] Stein, L., "Genome Annotation: From Sequence to Biology", Nature Reviews Genetics, July 2001, 2(7):493-503.

[19] D. P. Wall, H. B. Fraser and A. E. Hirsh, "Detecting Putative Orthologs", Bioinformatics Vol. 19 no. 13 2003, Pages 1710-1711