IMAS: The Interactive Multigenomic Analysis System

Christopher D. Shaw[1]
School of Interactive Arts & Technology,
Simon Fraser University Surrey,
Surrey, BC, Canada

Gregory A. Dasch[2]
Viral and Rickettsial Zoonoses Branch,
National Center for Infectious Diseases,
Centers for Disease Control and
Prevention, Atlanta, Georgia 30333

Marina E. Eremeeva[3]
Viral and Rickettsial Zoonoses Branch,
National Center for Infectious Diseases,
Centers for Disease Control and
Prevention, Atlanta, Georgia 30333

ABSTRACT

This paper introduces a new Visual Analysis tool named IMAS (Interactive Multigenomic Analysis System), which combines common analysis tools such as Glimmer, BLAST, and Clustal-W into a unified Visual Analytic framework. IMAS displays the primary DNA sequence being analyzed by the biologist in a highly interactive, zoomable visual display. The user may analyze the sequence in a number of ways, and visualize these analyses in a coherent, sequence aligned form, with all related analysis products grouped together. This enables the user to rapidly perform analyses of DNA sequences without the need for tedious and error-prone cutting and pasting of sequence data from text files to and from web-based databases and data analysis services, as is now common practice.

CR Categories and Subject Descriptors: J.3 [Life and Medical Sciences] Biology and genetics, I.3.3 [Computer Graphics]: Picture/Image Generation - Viewing Algorithms; I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques.

Additional Keywords: Bioinformatics, Visual Analytics

1 INTRODUCTION

The process of determining and annotating the structure and function of a newly sequenced microbial genome is a slow process because a researcher must first determine the nucleotide sequence of the microorganism of interest, then find genes, their protein product(s), and then the function of these products. While bioinformatics tools are available to automate elements of this process for single genomes, manual examination of the results is necessary because different analytical tools give different results, and because the quality of the data in the international genome databases (NCBI, EMBL, and DDBJ) is of varying accuracy. The current tools available for analyzing and understanding microbial genomic data are somewhat lacking in user-friendliness, consisting largely of web-based tools for either visualizing a fixed database of findings, or for analyzing sequence data supplied by the biologist in a type-in box.

This paper introduces a new Visual Analysis tool named IMAS (Interactive Multigenomic Analysis System), which combines common analysis tools such as Glimmer, BLAST, and Clustal-W into a unified Visual Analytic framework. IMAS displays the primary DNA sequence being analyzed by the biologist in a highly interactive, zoomable visual display. The user may analyze the sequence in a number of ways, and visualize these analyses in a coherent, sequence aligned form, with all related analysis products grouped together. This enables the user to rapidly perform analyses of DNA sequences without the need for tedious and error-prone cutting and pasting of sequence data from text files to and from web-based databases and data analysis services offered by different research organizations.

2 BIOLOGICAL SEQUENCE ANALYSIS

IMAS is aimed at reducing the drudgery that is currently entailed in the analysis of a microbial sequence for the structure and function of its genome. The primary job is to find what parts of the DNA of the organism encode biologically active molecules, and to discover what each molecule does within the organism. The following steps are typically taken by a biologist to discover this.

1. *Find genes.* A gene is a locatable region of DNA sequence, corresponding to a unit of inheritance, with regulatory regions, transcribed regions and/or other functional sequence regions. The transcribed region is the stretch of DNA that is transcribed by the organism into a biologically active molecule. To create a protein, transcription from DNA to RNA is followed by translation from that RNA sequence into the corresponding Amino Acid sequence. The resulting chain of amino acids folds itself into a biologically active protein molecule. A few genes create an RNA molecule that is itself active in the organism, which means that only transcription takes place for these. The regulatory region influences how much of its associated molecule is created.

Gene finding consists primarily of identifying protein-coding regions. In bacteria, such coding sequences are contiguous stretches of DNA. Since there are 20 possible amino acids to code for, and 4 types of DNA molecule (G, C, A, T), each amino acid is encoded using 3 DNA letters. Such a 3-letter sequence is called a codon. Gene finding programs such as Glimmer [6], GeneMark [13], and others typically locate a stretch of DNA that has a start codon and a stop codon in the same phase, with appropriate statistical properties.

2. *Identify gene/protein function by sequence similarity analysis.* Since an identical string of DNA yields an identical amino acid chain, gene function can be inferred by searching for a similar DNA or AA (Amino Acid) sequence in the international sequence databases. These databases store DNA and AA sequences of whole genomes, as well as of known proteins and their DNA coding sequences.

The most widely used tool for similarity search is named BLAST [1], which takes as a query the DNA or the AA sequence to search for. The BLAST program can be run on one's own computer to search in a local database. Also, the National

1: shaw<at>sfu.ca 2:ged4<at>cdc.gov 3: mge6<at>cdc.gov

Center for Biotechnology Information (NCBI) stores a copy of all of the publicly available DNA and AA sequences, and provides a web-based search service that runs BLAST to find similar sequences in its repository. The NCBI databases are quite large, about 150Gigabytes and growing by about 30GB per year. Running AA queries involves a subtle similarity computation per amino acid, since some amino acids share common chemical properties, and is frequently substituted for each other.

BLAST finds similar sequences, and presents a report of each relevant sequence alignment, which is a report of the character-by-character similarities and differences aligning the query sequence with the subject sequence found in the database. Both AA and DNA searches are needed because mutations that may change the DNA sequence may result in AA sequence changes. Codon redundancy means that some DNA change yield no changes in AA sequence.

In about 50 percent of cases, the query coding sequence is so similar to a well-known gene that the analysis for it need go no further.

3. *Multialignment.* For the remaining genes, the found sequences may be similar to others only to a very limited level, with one in 4 or 5 AAs being similar enough chemically to the query AA sequence. One analysis technique is to collect the hypothesized similar sequences together with the new sequence and perform a multiple sequence alignment. This process attempts to align all input sequences in an optimal arrangement that places identical or similar AA letters in the same column. As with pairwise alignment, gaps may need to be inserted in one or more sequences to enable this alignment.

Multialignment enables the biologist to observe regions where each sequence has conserved sequence. This implies that the region is biologically important, where other nonconserved regions are not important; thus mutations do not matter to the survival of the organism. Multialignments are often used to infer what the active region of an AA sequence is, and are often used to derive probabilistic models of membership in a family of proteins. The PFAM database contains a set of protein families that are each represented as a Hidden Markov Model of the family's AA sequence. This database can be searched by using a tool like HmmPFAM, which checks the query sequence against each HMM in the database.

2.1 Biological Sequence Visualization

Many visualization systems are aimed at the job of showing the results of analyses that have been conducted by the researchers on a particular set of genomic data. Some standard systems are in common use today. The UCSC genome browser [10] presents a web-based form that enables users to select a segment of stored sequence to view. At scales of tens of thousands of DNA letters (nucleotides, or NT), the browser presents genes over the selected NT range, along with channels representing other organisms and the presence of similar genes in those organisms. Included in this display are other annotations, or ranges of the DNA sequence with biological importance. Interaction is primarily through submitting queries to a web-based back end.

Lawrence Livermore Lab's ECR browser [15] enables visualization of similarity of multiple species. Arranged below the query sequence, a horizontal channel is allocated to each organism, with the percentage similarity to the source organism displayed per channel.

NCBI also offers a web-based gene visualization service, which again uses web-based queries to show genes and other annotations stored in NCBI's database. Each of these sequence viewers provides a comprehensive view of the hosting organization's database, with an interaction style that is typical of the web: users fill in forms or click buttons to submit a query, which the system responds to within a few seconds, depending on system load.

Another category of visualization systems are the bioinformatics workbenches such as Bluejay [18] and Apollo [12], among others. Bluejay is a Java-based genome visualization system that enables users to browse existing XML-formatted genomic data files, and to execute queries from the user's local machine that access remote databases. These include the genome databases as well as information about each gene in the Genome Ontology, among others. When a genome is being visualized, the user can query a gene by popping up a customized menu of analyses that are offered via a remote web services. The web services include BLAST, HmmPFAM, and many others.

Apollo [12] 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. Like Bluejay, there is a note-taking facility where users fill in free-text information about sequence ranges of interest.

Commercial tools such as Vector NTI and the CLC workbench also offer genomic analysis, although they tend to be focused on the management of laboratory processes such as gel electrophoresis, cloning, primer design and the like. (For example, Vector NTI is offered by Invitrogen, which is a company that sells biological equipment and reagents.)

Most of these systems have a common approach to the management of genomic data, which is to present the analysis process as a sequence of independent steps that generate independent results, not integrated into a coherent display. For example, Bluejay presents each analysis on a separate window. Apollo supports analysis by cut-and-paste. The CLC workbench presents each sequence alignment in a separate pane. Vector NTI performs simple analyses such as GC percentage analysis internally in a stacked set of line plots, but its access to external services such as sequence alignment and motif analysis are accomplished by submitting a URL to the user's web browser. Users must cut and paste the web-based analysis into Vector NTI.

This slows the analysis process because assessments of putative functions typically require that the researcher access multiple analysis tools found on different platforms with different types of databases and interfaces, read and interpret their output, and infer relationships between genetic elements based on rather variable primary biochemical data.

The analysis and accurate annotation of genomic sequences thus requires many analytical steps, each of which results in additional pieces of information that must be related to the growing database that the researcher is gathering. Current efforts to integrate promoter and termination recognition programs -- as well as evolutionarily conserved nearest-neighbor gene analyses and transcriptional analyses that identify both proteins and RNA regulatory elements -- have added new elements to the annotation process that are less-well honed and automated. The researcher is thus faced with an increasingly complex task of annotation document management, where each document contains results of many types of analyses of DNA or protein sequences that will need to expand and change as the total genomic sequence database expands and as new experimental data becomes available. Furthermore, this segmented information must be integrated into a network of higher level relationships including coordinated environmentally responsive transduced signals, transcriptional regulatory signals, and specific metabolite-regulated nucleic acid-protein and protein-protein interactions.

The drawback of separating each analysis into its own window is that there is no opportunity to visually integrate these analyses, and thus the opportunity to visually discover interesting correlations is absent. The benefits of an integrated display are that many of the analytical pieces can be brought together and integrated visually, enabling more rapid analysis of the researcher's data collection.

3 Integrated Display With IMAS

To enable integrated visual analysis, we have developed a zoomable visual analysis system named the Interactive Multigenomic Analysis System. This work was developed after our experience with another zoomable system named GVis [11]. As the primary architect of GVis, one of us (CDS) discovered that although GVis visually displayed evolutionary relationships of many microbial genomes, the GVis technology did not really aid biological analysis. That is, looking at the pictures was nice, but the real payoff of such a system would be to allow users to view the information and discover something new. The collaboration of the biologists from the CDC with the Computer Scientists at the School of Interactive Arts and Technology enabled a participatory design process that helped drive the development of IMAS towards analytical processes that really enabled a new type of analysis system.

Figure 1 shows a screenshot of the IMAS sequence analysis visualization. IMAS shows a number of channels of information about the DNA sequence of an organism (Nucleotide sequence, or NT sequence for short). Figure 1 shows the display and the most zoomed-in scale, where all sequence details are shown. The figure has been cropped, as the figure caption explains.

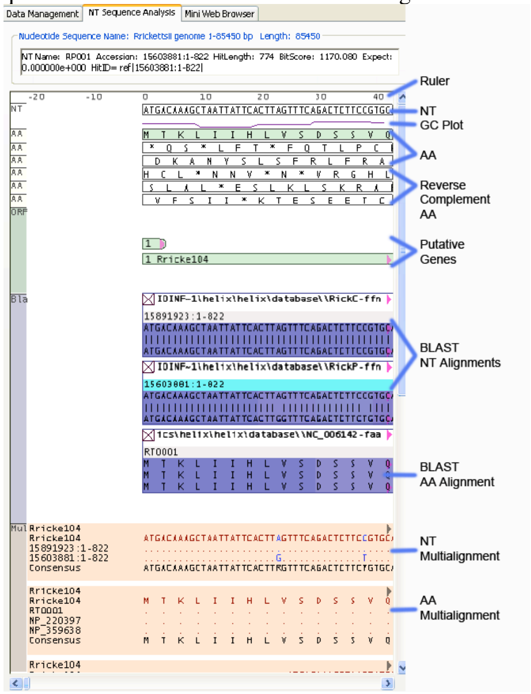
At the top of the display is a ruler that shows the index number of the NT sequence. Below the ruler is the NT sequence itself. We show the *forward* strand of the NT sequence. The *reverse complement* strand can be optionally displayed. The fundamental concept of visual display in IMAS is to align all sequence analyses with the NT sequence, and to layer analyses in a non-overlapping manner below the sequence. There are a number of categories of analysis, and each category is allocated its own piece of vertical screen real estate across the entire sequence.

Thus, the line graph below the NT sequence is a plot of GC content (percentage of nucleotides that are either Guanine or Cytosine), using a sliding window of 10 NTs.

Below this line plot are 3 channels dedicated to displaying the 3 possible forward reading frames of AA sequence. That is, if the first codon starts at the first nucleotide, the first channel of AAs would result. If the first codon starts at the second NT, the second channel would result. Every combination of 3NTs yields either an amino acid or is the STOP codon, which is commonly displayed as an asterisk.

Below the 3 forward *reading frames* are the 3 *reverse complement* reading frames, which are genes read from the complementary strand of the DNA double helix in the reverse order. In IMAS, the user would read the forward strand from left to right, and the reverse complement strand from right to left.

Below the AA reading frames are shown genes that have been found by running the Glimmer gene finder [6]. In IMAS, we run a local copy of Glimmer on the currently displayed NT sequence and interpret the sequence ranges as *features* to be displayed. There is a Feature channel in the IMAS display that is dedicated to the various sequence annotations that a biologist may be interested in. A putative gene named Rricke104 is shown in Figure 1 as a horizontal green box. The reading direction of the gene is displayed by a magenta arrow at the rightmost edge of the display, and is also displayed as a green highlight in the appropriate AA channel. The reading frame is also shown numerically as a single digit in the gene box. Thus, Rricke104 has an AA sequence of MTKLIIH...KYNK*. Because the display is zoomed in, the entire sequence cannot be seen without scrolling horizontally.



Nucleotide Sequence Name: Rrickettsii genome 1-85450 bp Length: 85450

NT Name: RP001 Accession: 15603881:1-822 HitLength: 774 BitScore: 1170.080 Expect: 0.000000e+000 HitID= ref|15603881:1-822|

```
-20       -10       0         10        20        30        40
NT                  ATGACAAAGCTAATTATTCACTTAGTTTCAGACTCTTCCGTGC
AA                  M  T  K  L  I  I  H  L  V  S  D  S  S  V  Q
AA                   *  Q  S  *  L  F  T  *  F  Q  T  L  P  C
AA                    D  K  A  N  Y  S  L  S  F  R  L  F  R  A
AA                  H  C  L  *  N  N  V  *  N  *  V  R  G  H  L
AA                   S  L  A  L  *  E  S  L  K  L  S  K  R  A
AA                    V  F  S  I  I  *  K  T  E  S  E  E  T  C
ORF
                    1
                    1 Rricke104
Bla                 IDINF-1\helix\helix\database\\RickC-ffn
                    15891923:1-822
                    ATGACAAAGCTAATTATTCACTTAGTTTCAGACTCTTCCGTGC
                    |||||||||||||||||||||||||||||||||||||||||||
                    ATGACAAAGCTAATTATTCACTTAGTTTCAGACTCTTCCGTGC
                    IDINF~1\helix\helix\database\\RickP-ffn
                    15603881:1-822
                    ATGACAAAGCTAATTATTCACTTAGTTTCAGACTCTTCCGTGC
                    ||||||||||||||||||||||| |||||||||||||| ||||
                    ATGACAAAGCTAATTATTCACTTGGTTTCAGACTCTTCTGTGC
                    1cs\helix\helix\database\\NC_006142-faa
                    RT0001
                    M  T  K  L  I  I  H  L  V  S  D  S  S  V  Q
                    M  T  K  L  I  I  H  L  V  S  D  S  S  V  Q
                    M  T  K  L  I  I  H  L  V  S  D  S  S  V  Q
Mul Rricke104
    Rricke104       ATGACAAAGCTAATTATTCACTTAGTTTCAGACTCTTCCGTGC
    15891923:1-822  ...........................................
    15603881:1-822  .......................G..............T....
    Consensus       ATGACAAAGCTAATTATTCACTTRGTTTCAGACTCTTCYGTGC

    Rricke104
    Rricke104       M  T  K  L  I  I  H  L  V  S  D  S  S  V  Q
    RT0001         .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
    NP_220397      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
    NP_359638      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
    Consensus      M  T  K  L  I  I  H  L  V  S  D  S  S  V  Q

    Rricke104
```

Figure 1. IMAS visualization at closest zoom. Starting from the top, the displayed elements are: NT location ruler, the NT sequence, GC plot, possible forward AA reading frames, 3 reverse complement reading frames (AA letters), a gene fragment and a putative gene named Rricke104 (green), 2 BLAST pairwise NT alignments (blue), a BLAST pairwise AA alignment (blue), an NT multiple alignment (orange), and an AA multiple alignment (orange). On the left is a color-coded vertical navigation widget. Cropped out of this picture are the menu bar at the top, the frame, and a debug reporting pane at the bottom.

Aligned below the Rricke105 gene are two NT and one AA pairwise sequence alignments, which each have dark blue background colors. Each of these was created by selecting the gene box and issuing a local BLAST query to locally-stored databases of NT or AA genomic data, respectively. In this case, each alignment is displayed inside a box that shows what database the alignment came from. From top to bottom, the databases were RickC-ffn, RickP-ffn, and NC_006412-faa. These databases were prepared by downloading data from NCBI and running the BLAST formatdb program. BLAST queries can also be done to the NCBI "nr" (non-redundant) database using BLAST's network query feature. This is a very large database, so queries take a minute or two.

The middle (RickP-ffn) pairwise NT alignment has been selected by the user, as shown by the title line's cyan highlight. The BLAST visualization strategy will be explained further in section 4.

Below the BLAST alignments are shown three multiple sequence alignments, with orange background color. The upper alignment is a NT alignment that was created by submitting 3 NT sequences to a local copy of the Clustal-W multiple alignment program [17]. The three NT sequences are the NT sequence of the gene Rricke104, and the two sequences found by BLAST and displayed in the BLAST area (15891923:1-822 and 15603881:1-822, respectively). To activate this, the user simply selects the desired BLAST alignments and pops up a menu on one of the BLAST boxes to execute the multialignment. The gene data to use is inferred from the gene that was used as the BLAST query.

For the AA multialignment, the interaction is similar. The user selects the desired AA pairwise alignments and pops up a menu to multialign these with the gene that the pairwise alignments are linked to. The Multialignment visualization strategy will be explained further in section 5.

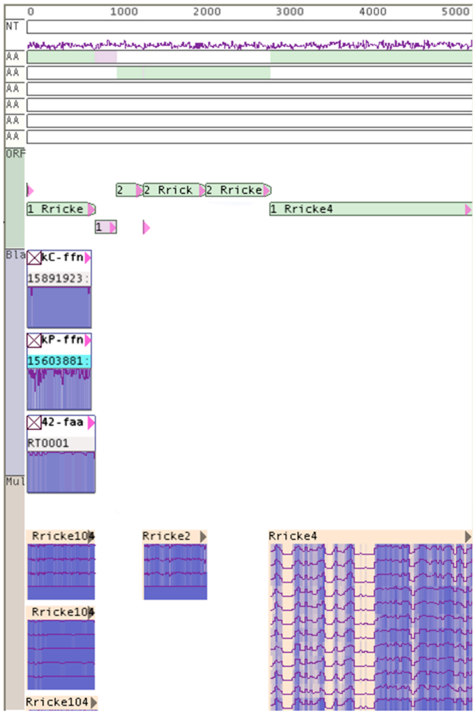


Figure 2. IMAS shown at zoom scale approximately 1/75. The rescaling happens only horizontally, enabling more than one gene to be shown with its associated analyses. Here 5 genes are shown (green) along with a magenta intergenic region. AA and NT sequence details are impossible to read at this scale, and are thus not drawn.

3.1 Zooming

The fundamental technique for managing the large amount of detail that can be rapidly accumulated is to perform a semantic zoom [8,16] in the contents of the IMAS display. Unlike many zoomable frameworks such as Pad++ [2,16], Jazz [3] or Piccolo [4], zooming in IMAS is in the horizontal direction only. The reasons for this are as follows: (1) By far the largest dimension is the horizontal sequence. The example shown in Figure 1 contains 85000NTs, which at the fully zoomed-in scale shown here would measure 118 meters. In contrast, our experience with using IMAS shows that the vertical dimension occupies about 3 meters at full scale. Second, the vertical space is allocated to separate functions that are usually accessed one at a time. That is, many BLAST runs are performed, followed by some multialignments, with respect to a single gene. Rescaling in both dimensions would result in making the individual BLAST runs difficult to interact with or to visually identify due to their small size.

We mean Semantic Zoom in the sense defined by Perlin and Fox's definition, which is the augmentation of detail of a semantically rich document as the level of magnification of an item increases [16]. Perlin and Fox give the example of a text document revealing first its title, then its abstract, then its outline, then finally the document's full content. These more detailed items are not per se geometric. The pairwise sequence similarities drawn by IMAS are geometrically located, but the nucleotide letters are not geometric themselves. What is being approximated is *similarity*, not the sequence itself, so as zooming out occurs, there is a semantic shift from similarity + sequence detail to just similarity visualization.

IMAS uses two levels of semantic zoom. A third level of semantic zoom could be to selectively hide insignificant items such as low-significance BLAST results as zoom-out continues.

Figure 2 shows a view of IMAS zoomed out to display 5000NTs horizontally, or about 500NTs per centimeter. In this case, all of the NT and AA sequence detail is not drawn because it would be unreadable at this scale. The labels are still drawn to fit within the space available, so that the user can see what each item is. At the bottom right is shown a multialignment with 11 aligned sequences. Geometrically, the zooming is accomplished by imposing a scale factor that shrinks only the horizontal dimension.

At zoom levels below certain thresholds, the NT and AA sequences are no longer drawn. For NT sequence data, the cutoff is about 0.8, and for AA, the cutoff is about 0.25. All text is drawn using texturing, so rescaled text is still functional, but gets hard to read at finer scales. To deal with sequence text scaling, our text drawing system performs an anti-scale for AA sequences that has the result of reducing the spaces between the AA letters at scales between 1 and 0.33. That is, since AA letters are drawn as a single letter followed by 2 spaces, as the scale shrinks, the blank space gets used up first. Once the blanks are used, AA letters are then scaled uniformly.

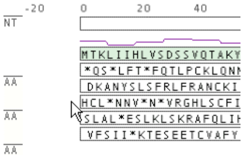


Figure 3. IMAS shown at zoom scale approximately 1/2. The text drawing function consumes the AA whitespace before the AA letters get rescaled.

The user sets the scale factor by either pressing the plus or minus keys on the keyboard, by turning the mouse wheel, or by double-clicking on a visual box in the display.

3.2 Zoom Navigation

Our first design for scale navigation used the + and – keys to scale up or down at appropriate discrete steps. The visual transition is immediate, to enable rapid response. The scaling center is the horizontal center of the display. The only drawback of this technique is that scaling and navigation are separated, so that scaling requires a key to be pressed, while scrolling requires that the users select a standard scrollbar.

To enable better vertical scrolling, we developed the *Sidebar*, which displays the major channels of analysis information at the left, with a fixed horizontal size; the vertical size is the log of the main display's channel. The minimum size of each sidebar channel is 1 character height. To scroll vertically, the user left-clicks in the sidebar to immediately jump to the corresponding place in the display. To move the top of the Feature/Gene channel to the top of the display, the user would click at the top of the green section of the sidebar (labelled ORF for Open Reading Frame, a common abbreviation for coding region). To jump to multialignments, the user would click on the orange region labelled "Mul". The user can hold the left mouse button down and drag the cursor in the sidebar to scroll vertically. Since the sidebar sizes are usually the log of the real sizes, scrolling rates are variable per channel.

For channels with small vertical extent such as the NT and AA sequence, sidebar elements are the same size as the corresponding main item. We allocate space in the sidebar first to the small-size items. For the larger channels, the log of the vertical size is used to compute an initial size, and the remaining sidebar space is divided up as a ratio of an item's log size divided by the sum of log sizes.

For very large regions (typically the BLAST channel), too much precision was required to align the pairwise alignments with the top of the display, so we included a dead band at the top of each sidebar channel to allow easy selection of the top of the channel.

IMAS's most effective navigation technique is to enable horizontal scaling and XY scrolling in a single gesture attached to the mouse wheel. When the cursor is in the display and the wheel is rolled forward, the display is scaled up by a factor of 1.05 per wheel unit, and horizontal and vertical scrolling takes place according the cursor's position relative to the center of the screen. Rolling the wheel toward the user scales by 0.95 per unit and scrolling occurs the same way.

Scrolling occurs according the following function, where, *dist* is the normalized distance from the center of the window, and *size* is horizontal or vertical size.

$\text{delta} = (\text{size}/2) * \text{dist}^{4.3}$

The resulting *delta* is in the range (0..size/2). The exponent 4.3 has the effect of reducing the rate of scrolling to very small amounts near the center. The same equation is computed for X and Y scrolling.

This wheel scroll/zoom function enables the user to rapidly scroll and zoom about the display using one unified technique, and is the primary means by which most users navigate. The user can rapidly scroll left or right by rocking the wheel backward and forward. This has no net effect on scale, but rapidly scrolls in the direction of the mouse.

A drawback or this technique is that it is not very precise, but tools such as the scrollbars are more suitable for this.

A requirement for the success of this technique is immediate graphical refresh, which takes place within 62ms or less on our system (see section 6.1).

4 PAIRWISE ALIGNMENT VISUALIZATION

As a consequence of semantic zoom, each of the alignment displays needs a visual summarization procedure that enables the researcher to rapidly gauge the quality of the pairwise alignments or multiple alignments.

At maximum detail, the pairwise alignments are presented similarly to standard human-readable BLAST, except that the pairwise alignment is presented continuously from left to right. Scrolling left or right will allow the user to see more of the alignment. Zooming out will allow the user to see a percent identity plot of alignment quality, and allow the user to see the entire alignment in one image. Instead of presenting index numbers of Query and Subject, each alignment is anchored at the left end, so that it can be located in the context of the query Feature. Gaps in the alignment will cause the results to fall out of alignment with the top channel NT sequence.

Each BLAST pairwise alignment is presented in a four-row blue box. The top row shows the name of the aligned sequence. The next 3 rows show the query, alignment midline, and subject (hit) sequence. We chose darker blue to indicate stronger similarity because of the requirement to distinguish figure from ground. Because we use a white background, the darker blue contrasts the most, and therefore shows strongest similarity. The semantic relation is that light blue fades into the background, which means that it is unimportant.

Figures 1 and 2 show pairwise alignments. The dark blue backgrounds of the 3 detail lines of the alignment are a percent identity plot of the alignment. Percent identity is a computation of the percentage of a segment of a pairwise alignment that is identical. Pairwise alignments are displayed in a number of genomic workbenches, including many of those reviewed in section 2.1.

IMAS does two visualizations of percent identity. The first is shown at all levels of zoom, where the dark blue color represents 100% identity, and very light blue represents 0 identity. These background colors are shown at all zoom levels so that the user can visually locate an interesting part of the alignment and maintain focus on it while zooming in and out. In figures 1 and 2, the level of identity is quite high throughout, so there is a lot of dark blue.

Figure 2 shows the same alignments with the display zoomed out. When the zoom level leaves no space to show NT or AA text, the alignment details are replaced with a line plot of percent identity, which visually duplicates the blue background, but allows more accurate visualization. When zooming, the text labels are scaled appropriately to be visible, but they may be cut off on the right due to lack of space, as can be seen in the top left three alignments.

The cyan highlight in the label line of the middle alignment indicates that it has been selected. Detailed information is displayed in a text window above the visualization display. This includes many of the important details of the selected alignment, such as Name, Accession number (database key), Hit length, plus various quality scores. The HitID is usually something like gi|28262418|gb|EAA25922.1|, which can be used for data lookup by selecting the text in the text window and searching for it at the NCBI website.

4.1 Computing Percent Identity

The Percent Identity Plot for pairwise alignments is computed by sliding a non-overlapping 10 NT character window over the pairwise alignment and scoring each Query-Subject letter pair according to its similarity. The sum of scores per letter is divided by the number of letters, resulting in a value between 0 and 1, which is plotted.

For NT pairwise alignment, we have built two schemes. The first is very simple. Identical NTs score 1, everything else scores 0, and the number of letters is always 10. The problem with this simple scheme is that some NT differences matter more than others, because they code for the same Amino Acid.

Our more complex NT scoring scheme accounts for codon identity and codon redundancy, as well as for the resulting Amino Acid similarity according to the standard substitution matrices. For this codon-based similarity, IMAS scores each codon according to its level of identity. There are three situations:

- First, if the Query & Subject codons are identical, the score is 3, which is numerically compatible with the previous scheme of scoring 1 per identical nucleotide.
- Second, if the codons are different, but code for the same amino acid, the score is 2.
- Third, if the codons code for different amino acids, then the BLOSUM62 matrix is consulted. Each codon is translated into its Amino Acid, the BLOSUM62 matrix [9] is used to look up the substitution log-odds probability for these two AAs. This value subst is plugged into the formula

score = (subst – min) / (max – min).

The value min is the minimum log-odds ratio for the matrix, and max is the maximum for the Query AA, which is on the matrix diagonal. This formula yields a codon score between 0 and 1. The sum of codon scores is divided by the 3 times the number of codons used.

Since the size of the sliding window is not necessarily divisible by 3, a codon is included in the current window only if all 3 letters fit. If not, the current window is completed, and the non-fitting codon is used as the first codon in the next window. If necessary, the scoring procedure accounts for slippage by "catching up" with an extra codon every 3 windows. For example, with window size 10, the first window uses 9 NTs, the next uses 9 NTs, and the next uses 12 NTs. 9+9+12 = 30.

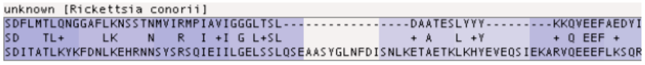
For AA pairwise alignments, the same BLOSUM-based scoring scheme is used, with the obvious exception that no NT translation needs to take place. Similarly, for window sizes not a factor of 3, the catch-up procedure is used. The score per window is scaled to a value between 0 and 1 according to the number of AAs used.

4.2 Geometry of BLAST Results

The presentation of each pairwise alignment starts immediately below the feature/gene channel. BLAST reports pairwise alignments in order from most to least significant pairwise alignment, and we use this strategy to lay results out spatially. The significance ratings are generated by BLAST's computation of Bit Score and Expected Probability Value. Expected Value is computed by BLAST as the probability of randomly finding a sequence alignment of the same length in a database of the size used.

The first pairwise alignment is placed as close to the top as possible, and the next is stacked below as close to the top as possible. If there is space available, a pairwise alignment will be placed in the top row, which would happen if there were two non-overlapping alignments along the NT sequence. Pairwise alignments are stacked from top to bottom in order from most to least significant. If there is space available horizontally for a pairwise alignment near the top, it will be placed there.

BLAST runs for a single database (which are a set of pairwise alignments) have a similar stacking rule of being fitted in the first available space. Because alignment gaps tend to make results larger than the query, BLAST runs on closely-spaced genes will give badly-stacked results with lots of empty space. We are investigating layout techniques to solve this problem.



```
unknown [Rickettsia conorii]
SDFLMTLQNGGAFLKNSSTNMVIRMPIAVIGGGLTSL-----------------DAATESLYYY---------KKQVEEFAEDYI
SD   TL+     LK    N   R  I +I G L+SL                 + A   L +Y         + Q EEF +
SDITATLKYKFDNLKEHRNNSYSRSQIEIILGELSSLQSEAASYGLNFDISNLKETAETKLKHYEVEQSIEKARVQEEEFLKSQR
```

Figure 4. AA BLAST alignment with gaps in the query sequence.

Many BLAST alignments have one or more gaps, where one sequence contains a small inserted sequence of letters not contained in the other. Figure 4 shows an example of an AA sequence in which the query sequence has ended up with gaps because the subject sequence (found and aligned by BLAST) has two small inserted sequences in those regions.

For visualization purposes, this causes the problem of misaligning the query sequence against the top-level AA or NT sequence. We solve this problem with a brushing technique. IMAS initially lays out all alignments according to the leftmost boundary. When the user holds down the ALT key and moves the mouse cursor in the region of the BLAST alignment, IMAS examines each alignment for gaps in the query, and moves the alignment to the left by the number of gap letters that are present to the left of the cursor. In effect, each alignment "skips over" each gap so that the query sequence is aligned with main AA or NT sequence.

All BLAST results are placed within a dedicated rectangle that is as wide as the Primary Sequence, and as tall as is needed to fit all the results. Other analyses, such as multialignments, are placed below this pairwise alignment channel.

4.3 Filtering Pairwise Alignments

Many of the pairwise alignments reported by BLAST can be of low significance. IMAS currently calls BLAST with default values and filters out the results. IMAS provides a dialog box to filter out insignificant pairwise alignments. The user types in minimum values of length or bit score (smaller means less significant), or maximum values of probability or expected value (larger means less significant). IMAS applies the minima/maxima globally and cumulatively to filter out small or insignificant alignments from the display. Note that this filtering operation controls Display only. The BLAST pairwise alignments that are hidden are still present in the system, and can be un-hidden at any time.

5 Multiple Sequence Alignment

To run the Clustal-W multialignment program, the user selects one or more pairwise alignments in the display to multialign. IMAS uses the Subject (Hit) sequences and the NT or AA sequence of the related Feature for multialignment. The feature/gene does not need to be explicitly selected, since this will be determined by which BLAST pairwise alignments were selected. It only makes sense to multialign the gene with the pairwise alignments that were derived that gene. Selections of pairwise alignments from other genes will be ignored.

5.1 Multialignment Display

Multialignment results are displayed on an orange background with the letters of the alignment colored according to their similarity to the consensus. The consensus line is computed by IMAS, and is simply the letter that is represented above 50% in the column. Columns of identical letters in the multialignment are colored red, while non-identities are blue. All sequences in multialignments are presented in Forward order. Thus, if a reverse-complement Feature or BLAST pairwise alignment is to be used, it is transformed into forward order and multialigned.

The whole multialignment is labelled with the gene that was used in the multialignment. Each row of the multialignment is labelled with which sequence was used to create it. Since we allocate a fixed amount of space on the left for the row labels, as IMAS is zoomed out, that space is also shrunk, which leaves less and less room for row labels.

At less detailed zoom levels, similarity to the consensus is computed row-by-row in the same way as the Percent Identity Plot for BLAST pairwise alignments.

Figure 2 shows 4 multialignments near the bottom of the image. The three on the left are of quite high quality throughout, so each line has a blue background. The bottom right multialignment has 5 regions of medium to large similarity across each sequence, separated by low-similarity regions, particularly at the left and the center. The visualization gives the general impression of the level of consensus across the multialignment at areas of coarse zoom.

5.2 Multialignment Source Data

The source data used in multialignments comes from databases that were used to generate BLAST pairwise alignments. There are two categories of multialignment input data: Gene data comes from either the NT sequence as derived from the extent of the gene, or the appropriate reading frame of the AA sequence for this gene.

Pairwise alignment Subject sequences are used as a database accession query into the source database that generated the pairwise alignment. That is, instead of just copying the sequence from the Subject line in the pairwise alignment that IMAS displays, IMAS takes the Accession key from the BLAST pairwise alignment and searches for the “full-size” sequence in the local BLAST database where the pairwise alignment was found. The length of this sequence is trimmed to the length of the Feature, so that BLAST pairwise alignments that came from whole-genome NT sequences are not multialigned over millions of nucleotides. The result is that BLAST is used to identify the corresponding gene in another organism, and that full gene is used in multialignments.

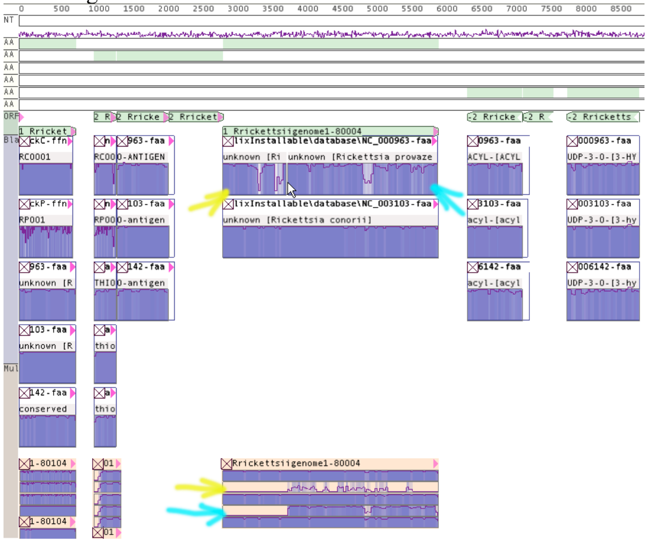


Figure 5. This shows a brief analysis of the first 6 putative genes of *Rickettsia Rickettsii* at a zoom factor of 1/75. The mouse cursor shows where BLAST has found two significant alignments for the long gene Rrickettsiigenome1-80004 (between upper yellow and cyan arrows). Interestingly, the multialignment below finds a better alignment of the left fragment (yellow) with the part of the sequence on the right two thirds. In the source of these two BLAST hits, *Rickettsia Prowazekii*, these are two separate genes that align with fragments of a single corresponding gene in *Rickettsia Rickettsii.* The relocation in the multialignment indicates that there is a repeated subsequence.

This function does not work for BLAST pairwise alignments derived from the network-based NCBI “nr” database, because fastacmd (a BLAST program that performs this lookup query) does not allow this type of retrieval to the NCBI nr database. Of course, the user could download the desired database and run queries locally. BLAST database formatting must be done correctly for this function to retrieve data from local BLAST databases to work. Source data requires that each entry have some kind of accession number or other unique identifier. If this is not present, then the database lookup process for multialignments will fail, and the user sees only the Subject sequence.

6 Results and Conclusions

IMAS has been successfully employed at the CDC to analyze genomic data of *Orientia tsutsugamushi* [7], which is the causative agent of a tropical disease named *Scrub Typhus*. IMAS analysis led to the recognition of some significant attributes of the *Orientia* genome. First, a large number of repeated elements are present in the *Orientia* genome, most notably including 19 SMART domains found in more than one Open Reading Frame: most frequently the 56 histidine kinase-like ATPases domains (HATPase_C) and the 37 domains found in ATPases associated with a variety of cellular activities (AAA). The number and type of SMART domains detected suggest that *Orientia* has a much larger range of regulatory proteins involved in responses to environmental stimuli than any known for *Rickettsia* including *R. felis*.

6.1 Performance Results

IMAS was constructed with the goal of supporting rapid genomic analysis, and to that end, we have developed a visualization front end to IMAS that performs reasonably quickly in terms of graphic update. For a database of 10megabytes, IMAS is able to draw visual updates quite rapidly at all zoom levels. This 10MByte database consists of an 85,450NT sequence with 104 genes, 131 BLAST runs, and 10 multialignments.

To measure graphics update times, we instrumented the IMAS code to count the number of graphical operations performed per update. We also measured the real time elapsed in milliseconds for 100 single updates, since single-frame update times were too close to the millisecond resolution of the Windows XP clock. The following measurements were performed on a Dell Optiplex GX280 with a single HyperThreading Pentium 4 with a 3.00GHz clock and 800MHz Front Side Bus, and 1.0GB of 400MHz DDR2 SDRAM. The graphics card was an ATI X800 XL with a 400MHz core clock, and 256MB of memory, released in early 2005. The rendered area was 1622 x 700 pixels.

Table 1 shows various levels of zoom for this database, counting the number of quads actually drawn, the number of visits to unique scene graph nodes, the number of geometric transformations, and the number of textured letters drawn.

Table 1. Runtimes for a moderately large database at various levels of zoom.

Zoom level ⇨	1	1/3	1/15	1/75	1/225
Quads Drawn	257	593	2373	4489	12653
Scene Graph Visits	1213	1342	3268	7404	19014
Transforms	820	1065	3382	7621	20714
Text Drawn	4924	2498	1367	10060	681
Time/Update (ms)	4.07	3.75	9.22	25.47	62.50

At the closest zoom, the update time is about 4.07ms, or 245 updates/second. The update time increases as the zoom factor decreases to accommodate a larger proportion of the scene. The farthest zoom, 1/225, yields an update time of 62.5ms, or 16 updates/second. For practical purposes, the navigation response time is instant when viewing details, and within the interactive limit of 100ms when navigating at the coarsest level of zoom [5].

6.2 Discussion

A major strength of IMAS is the ease with which genomic sequence analyses can be dispatched and viewed. Correspondences between many similar sequences can be readily detected in the IMAS display and further examined by mouse based panning and zooming.

BLAST analyses can run in about one or two seconds on local small databases of 1-10 million nucleotides. We recommend running against a local database, if possible,.

IMAS currently has some drawbacks. First is the problem of inadequate vertical screen real estate. Figure 5 shows that a few analyses can overwhelm the available space, and some sort of approximation procedure would be useful. Our current solution to the problem for BLAST runs is to use the global dialog to shrink the vertical extent of all BLAST alignments. Local control of visibility would be more useful.

Another significant drawback is the stacking layout. In following the axiom that all items must align to the main sequence, it is frequently the case that BLAST runs from neighboring genes overlap. In fact, gene overlap is also common. The stacking layout responds to this by stacking BLAST runs below others, even if the overlap is 1 NT. From an algorithm perspective, this decision means that we can compute on BLAST *runs*, not the entire collection of *alignments*, but this performance benefit is not without usability drawbacks.

Finally, the interaction with other BioInformatics tools could be more flexible. As many Bioinformaticists are aware, the task of integrating many analyses into a single workspace is quite labor intensive. Workflow software such as Taverna [14] promises to help with this by providing a protocol for accessing SOAP-enabled analyses on the Web.

6.3 Conclusions

Operationally, IMAS enables the rapid analysis of genomic data either before or after sequence assembly. IMAS allows the user to analyze DNA fragments that are generated at the first stage of a sequencing project. The stages of analysis outlined above occur with much greater speed than the current various web-based tools allow for. This is primarily due to process and display integration. All analytic products are displayed together in a coherent manner, which allows for significant time savings for the researcher.

ACKNOWLEDGEMENTS

We would like to acknowledge the help of Jason Chih, Hemal Shah, Athena Deng, Carl Zhang, Buddy Moore, Alex Tsai, and Jessie Pledger in the development of IMAS.

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] Bederson, B. B., & Hollan, J. D. “Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics”. In Proceedings of User Interface and Software Technology (UIST 94) ACM Press, 1994 pp. 17-26.

[3] Bederson, B. B., Meyer, J., & Good, L. “Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java”. UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters, 2(2), pp. 171-180.

[4] Bederson, B. B., Grosjean, J., Meyer, J., “Toolkit Design for Interactive Structured Graphics”, Transactions on Software Engineering, New York: IEEE, 2004, 30 (8), pp. 535-546.

[5] Card, S.K, Moran, T. & Newell, A. The Psychology of Human-Computer Interaction, Hillsdale, NJ: 1983, Lawrence Erlbaum Associates.

[6] Delcher, AL, D Harmon, S Kasif, O White and SL Salzberg, “Improved microbial gene identification with GLIMMER”, Nucleic Acids Research, 1999, 27(23):4636-4641.

[7] Eremeeva, Marina E., Anup Madan, Chris D. Shaw, Kevin Tang, and Gregory A. Dasch, “New Perspectives on Rickettsial Evolution from New Genome Sequences of Rickettsia, particularly *R. canadensis*, and *Orientia tsutsugamushi*, Annals of the New York Academy of Sciences, Vol 1063, pp 47-63 (2005).

[8] Furnas, George W. “Effective View Navigation”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta, GA, ACM Press, 1997, pp. 367–374.

[9] Gibas, Cynthia, & Per Jambeck, *Developing Bioinformatics Computer Skills*, 2001, O’Reilly Press

[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] Hong, Jin, Dong Hyun Jeong, Chris D Shaw, William Ribarsky, Mark Borodovsky, and Chang Song, “GVis: A Scalable Visualization Framework for Genomic Data”, EUROVIS 2005: Eurographics / IEEE VGTC Symposium on Visualization 2005 June, 2005, Leeds, UK. pages 191-198, 2005.

[12] 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

[13] Lukashin, AV, and M Borodovsky , “GeneMark.hmm: new solutions for gene finding”, Nucleic Acids Research, 1998, 26(4):1107-1115.

[14] Oinn, Tom, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat, and Peter Li, “Taverna: a tool for the composition and enactment of bioinformatics workflows”, Bioinformatics 2004 20(17):3045-3054; doi:10.1093/bioinformatics/bth361

[15] Ovcharenko, I, Nobrega MA, Loots GG, Stubbs L. “ECR Browser: a tool for visualizing and accessing data from comparisons of multiple vertebrate genomes”. Nucleic Acids Res. 2004 Jul 1;32(Web Server issue):W280-6.

[16] Perlin, Ken, and David Fox, "Pad: An Alternative Approach to the Computer Interface", Proceedings of SIGGRAPH 1993, Aug. 1-6, 1993 Anaheim, CA, ACM Press, pages 57-64.

[17] Thompson, J.D., Higgins, D.G. and Gibson, T.J. “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice”. Nucleic Acids Research, Nov. 1994, 22(22):4673-4680.

[18] Turinsky, A.L., Ah-Seng, A.C., Gordon, P.M.K., Stromer, J.N., Taschuk, M.L., Xu, E.W., and Sensen, C.W. (2005) “Bioinformatics Visualization and Integration with Open Standards: The Bluejay Genomic Browser”. In Silico Biology, 5(2):187-198.