A Top-Down Approach to Whole Genome Visualization

Heumann K., Harris C., Mewes H.W.

Max-Planck-Institut für Biochemie, MIPS,
Am Klopferspitz, 82152 Martinsried, Germany
Phone: +49 89 8578 2451 FAX: +49 89 8578 2655
heumann@mips.embnet.org

Abstract

The investigation of large DNA contigs like complete chromosomes or genomes requires novel methods of data visualization. The complex information contained in a genome, particularly the relation of its individual genetic elements, needs to be accessible in a comprehensive, intelligent and intelligible manner. The yeast genome is expected to contain more than 6,000 Open Reading Frames (ORFs). As yet, the function of many of these ORFs has not been characterized satisfactorily. Also, many ORFs are found to have redundant copies elsewhere in the genome that originated from common ancestors. Other genetic elements (e.g. Tss, delta-elements, t-RNAs) are present in multiple copies. To visualize these relationships, a top-down "genome browser" is introduced that enables inspection of genomic data at different levels of abstraction (e.g. chromosomes, coding/non-coding regions, high/low levels of similarity). This novel tool is a key component for the integrated services approach to biological sequence data management (Heumann et al. 1995) and is accessible through the world wide web (WWW). This work demonstrates how the genome browser visualizes the results of an all-against-all comparison of the elements in the yeast genome as a graph. Interactive navigational queries across yeast chromosomes along the lines of sequence similarity open versatile options for the detailed investigation of genome properties. For sequence comparison the hashed position tree HPT (Mewes & Heumann 1995) is applied. Sequence similarity relationships are represented using the genome similarity graph (GSG) (Heumann & Mewes 1996c).

Introduction

The volume of biological sequence data published grows at exponential rates. Systematic sequencing projects contribute to this not only by the volume of data[1], but also by providing a natural organizational framework for the management of large genome oriented data sets. No comprehensive tools to access large contigs or whole genomes with sensible queries have been presented[2]. An interactive tool is introduced here that allows for comprehensive visualization of multiple chromosomes or genomes as a starting point for a more refined analysis following a top-down strategy.

A major goal of in genome analysis is the identification of ORFs that are translated into mature proteins that make up the living cell. In a first step of the analysis, coding DNA regions are separated from non-coding regions and sequence comparison techniques are applied to the translated ORFs to characterize their function by homology (Dujon et al. 1994). In many cases the precise biological function of the expressed proteins remains uncertain or entirely unknown (Galibert et al. 1995). The non-coding regions of a genome are also subject to independent analysis. Known sequence patterns (i.e. DNA motifs like ARS-Elements, promotors, etc.) and other genomic elements like t-RNAs, Tys, delta-elements (Feldmann et al. 1994) can be found.

The genetic elements identified are the basic logical units to perform sequence data analysis. Global properties of contiguous genomic data have rarely been investigated (Ozier-Kaleogeropoulos 1995). Thus, relationships between independent units are limited to isolated, accidental findings. The systematic investigation of relationships between distant regions based on feature tables is tedious and relies on annotated information which might be incomplete or inaccurate. In addition, since the redundancy within the yeast genome reflected by the intra-genomic similarity is high (ca. 30% (Ozier-Kaleogeropoulos 1995), these relationships are often complex. Adequate visualization is an appropriate approach to the systematic, exhaustive genome analysis.

The main objective of the work is to demonstrate a mechanism for visualization as a specific, novel approach to genome analysis. The method is best described as a "top-down" approach (Heumann & Mewes 1996c). The starting point is a global view of the whole genome. From this general view the user can focus on specific elements or features of the genome. In contrast, current approaches to genome analysis follow a "bottom-up" approach. They

[1] More than 70% of the yeast genome has been completed. The sequences of chromosomes I, II, III, V, VI, VIII, IX, X, XI and XIII have been published. In addition other small genomes, haemophilus influenzae and mycoplasma genitalium, have been completed.

[2] The well known database tool ACeDB is limited to the graphical presentation of the underlying dataset.

rely on identification of isolated independent findings. The set of all findings is reported as a result of the analysis (Bork et al. 1992). By definition, this methodology is always local, static and rarely exhaustive. Since more than 80% of the S. cerevisiae genome is presently known, a top down approach to whole genome analysis based on chromosomal units becomes feasible for the first time.

The method of sequence comparison and data representation will be described. The hashed position tree (HPT) is applied as index data structure to compute the all-against-all comparison of the genomic sequence data efficiently. The result of this comparison is represented as a graph of sequence similarities. This genome similarity graph (GSG) is used by the genome browser for the purpose of visualization. We will discuss different applications of the genome browser to systematic whole genome analysis.

Method

The WWW based genome browser renders the set of all sequence similarities within a whole genome accessible in an interactive way. As a prerequisite, the all-against-all comparison of the data set is required. Suffix-trees (Weiner 1973, McCreight 1976, Ukkonen 1993), pat-trees (Gonnet et al. 1992) and position trees (Aho et al. 1974) are well known data structures that allow for efficient string comparisons, provided that both text and tree fit into main memory. Under these conditions, search times depend on the length m of the query string for searches subsequent to index construction.

The first attempt at an all-against-all comparison of a large data set was published by Gonnet (Gonnet et al. 1992), but the processing time required to keep up with the increase in data volume is prohibitive. Furthermore, the question of organizing the data so as to limit the accesses to slow, persistent memory (i.e. disks), was not addressed. Recent implementations have adopted suffix tree variants to large data sets: HPT (Mewes & Heumann 1995), SB-tree (Ferragina & Grossi 1995), and huge-tree (Bieganski 1995). The huge-tree relies on a compression of a generalized suffix tree, but the algorithm was not analyzed according to the number of secondary memory accesses. The alignment algorithm described using the huge-tree operates efficiently compared to programs such as BLAST only for closely related sequences up to PAM[3] 20 (Bieganski 1995) and is not suitable for any sensitive sequence similarity searches. The SB-tree addresses the specific concerns of secondary storage representation of suffix trees and presents a B-tree variant as solution. However, no alignment algorithm has yet been outlined to operate on SB-trees. It is questionable if the SB-tree is suitable for efficient approximate string matching at high sensitivity. The HPT is the first tree structure proven to be suitable for approximate substring matching as required for sensitive sequence alignments. The details of this approach are given elsewhere (Heumann & Mewes 1996a). Several examples of the applications of the HPT in whole genome analysis, such as the comparison of data collections to a complete genome (e.g. human ESTs) (Heumann & Mewes 1996b) and sequence fragment assembly (Heumann & Mewes 1995) were described. Therefore, only a very brief summary of the HPT is given.

The underlying principle of a hashed position tree (HPT) is to group position identifiers that have a common partial position identifier as prefix. Specific partial position identifiers are encoded as paths in the HPT pointing to a set of positions.

Definition: A substring Y of length m of sequence X is called a *position identifier* for position i, if the following conditions are met:
(1) Y is not empty.
(2) No substring Z exists in X starting at position j with $i \neq j$ and $Y=Z$.
(3) Y has no proper prefix satisfying (2).

Definition: Every non-empty prefix of a position identifier is a *partial position identifier*.

The HPT groups such a set of position identifiers onto a single page. The HPT is balanced to allow for the retrieval of all occurrences of an encoded partial position identifier in a single disk access.

Definition: An *HPT* is a position tree encoding partial position identifiers according to the parameter set $\{\Sigma, \tau, h, d, k_{max}, \mu\}$ with:
(1) Alphabet Σ and terminal symbol t.
(2) Hash function h with depth d.
(3) A maximal segment size k_{max} and a maximal segment number per page μ.

The HPT can be generated in $O(n)$ space and $O(n \log n)$ time for a biological sequence data set of size n. Matching a substring Y of size m requires $O(m \log n)$ time subsequent to index construction. With respect to secondary storage access, the HPT clusters large data sets efficiently, i.e. any substring Y of size m can be found in at most $O(m/\log n + occ/b)$ disk accesses, where occ is the number of occurrences of Y and b represents the page size (depending on k_{max} and μ).

[3] Equivalent to about 85% sequence identity.

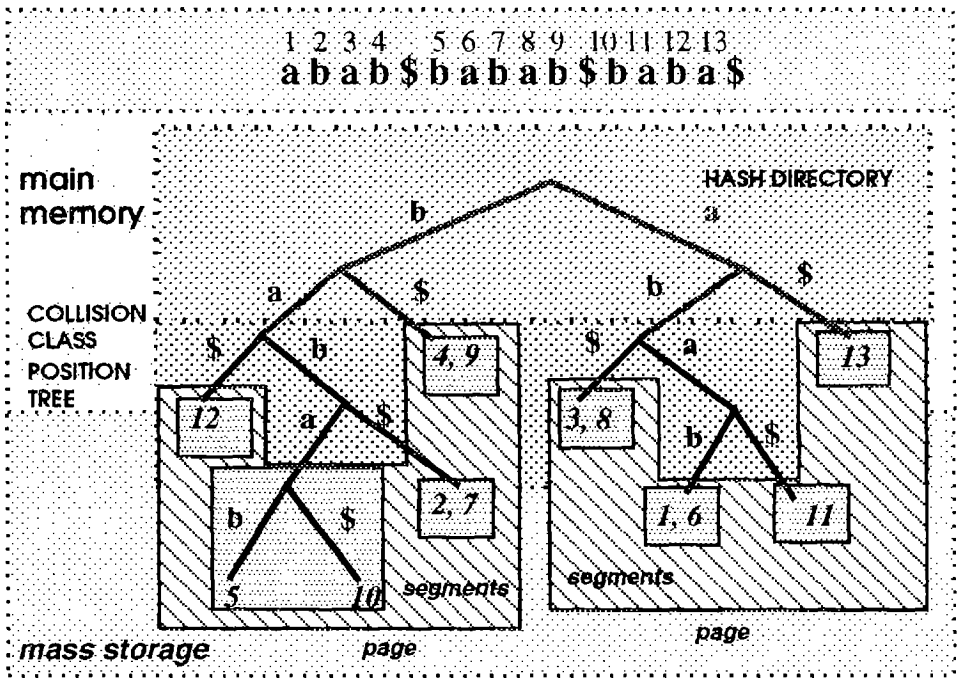

Figure 1: Example of an HPT={Σ={*a,b*},τ=`*$*`,*h,d=2,k*$_{max}$=2,μ=*4*} taken from (Mewes & Heumann 1995).

The example presented in figure 1 demonstrates how a text is represented by an HPT index. The HPT is realized as a hybrid data structure that combines the following components:

- *Hash-directory:* The hash-directory contains the part of a position tree that is complete.
- *Collision class position tree:* The collision class position tree refines a collision class of the hash-directory
- *Segment:* A segment contains the set of positions associated to the corresponding partial position identifier encoded in the collision class position tree.
- *Data page:* A data page is the logical transport unit of the secondary storage device. A page contains a set of segments.

According to the definition outlined above, the HPT can be parameterized to allow for the optimization of individualized search strategies (e.g. for protein and DNA data respective for coding and non-coding regions in DNA). For example, the comparison on the DNA level, focusing on similarities of coding regions was realized by the application of specialized hash functions. A function with ***d=7*** was selected that allows for *don't cares* at position ***3*** and ***6***, reflecting the ambiguity of the genetic code. In case of the analysis of intergenic regions, other hash functions appeared to reflect the biological significance of the signals better. This adaptability of the HPT is employed for the systematic analysis of the yeast genome by applying specialized hash functions.

The all-against-all comparison of the yeast genome is done by pairwise sequence alignments of fixed size blocks of genomic data (e.g. more than 9000 blocks with 500 nucleotides per block). In order to allow for direct comparison of genomic DNA and protein similarities for each block also, the 6-frame translation into protein sequences is generated. The details of the HPT-based pairwise alignment algorithm are given elsewhere (Heumann & Mewes 1996a). The basic idea of the approach is to sample a query sequence by a set of partial position identifiers encoded in the HPT. This is done according to an objective function that also accounts for conservative exchanges of amino acids for protein data. Note that this first step is done fully in main memory. In a second step segments related to high scoring partial position identifiers are retrieved from secondary storage. Sets of partial position identifiers that belong to the same sequence are compiled to generate an alignment with the query sequence. Note that this alignment also allows for gaps. This alignment score is presented to the user. The overall complexity of the all-against-all comparison accumulates to $O(m^2 n \log n)$ time with block size m and size n of the data set. In practice this corresponds to 3-4 hours CPU time for the yeast data set depending on the block size, using a single DEC station 3000.

The GSG represents the result of the all-against-all comparison. In order to be exhaustive we choose to calculate similarity relationships down to a level of about 15% sequence identity. This results on the order of 10^4 distinct relationships identified on protein level. Thus, on average, every block has 10 related blocks at this sensitivity and

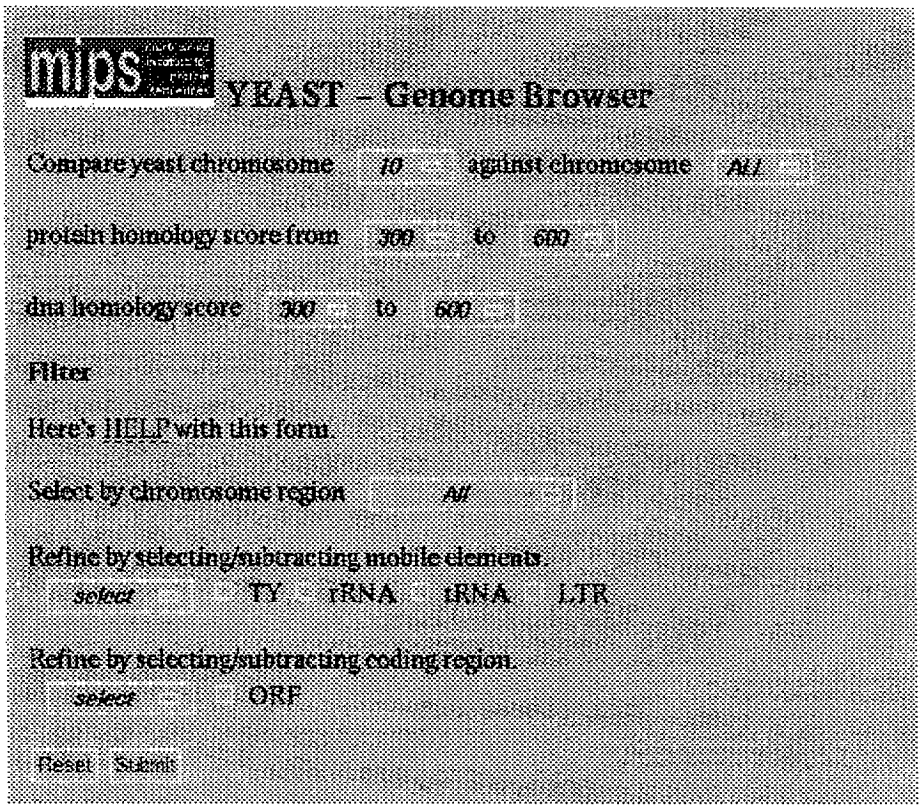

Figure 2: WWW form of the genome browser to specify a view of the genome similarity graph GSG. The resource is accessible through the MIPS home page: http://www.mips.biochem.mpg.de.

block size. Once a GSG is generated, all standard transformations of graphs can be efficiently applied. A graph is a network of vertices connected by edges. Each vertex of the GSG represents a DNA-block[4]. An edge connecting to vertices represents a similarity relationship between two blocks. Each vertex contains information about its position on a distinct chromosome. Edges are labeled according to the scalar similarity score. In terms of relational algebra, it is possible to formulate complex SELECT-statements on the relation underlying the GSG. A SELECT specifies a filter operation f applied to a GSG and generates a subgraph GSG_f.

Results and Discussion

Once the all-against-all comparison of the genomic data is completed for a given block size, the resulting relation is represented as a GSG. This allows for visualization and investigation of the genome using the graphical genome browser and provides intuitive access to the data. The user may explore groups of genetic elements without prior knowledge of the data's properties. For example, gene duplications are of great interest for the functional analysis in yeast. Duplications are frequently described as isolated discoveries. The genome browser allows for the investigation of such local findings with respect to the complete genome on a high level of abstraction and elucidates these events by visualization. It supports data mining within the scope of the complete genome. The user can surf across the genome, express hypotheses and validate them by filtering. Importantly, annotated genetic elements can be selected to be correlated with specific features of the GSG. Figure 2 shows the input form that enables the user to define a specific view of the genome.

The following three applications exemplify how the genome browser can be used:

- Service integration and interoperability
- Systematic investigation of ORF duplications (sliding window technique)
- Systematic investigation of intergenic regions

[4] An analogous GSG for protein data represents the ORFs by nodes of the graph.

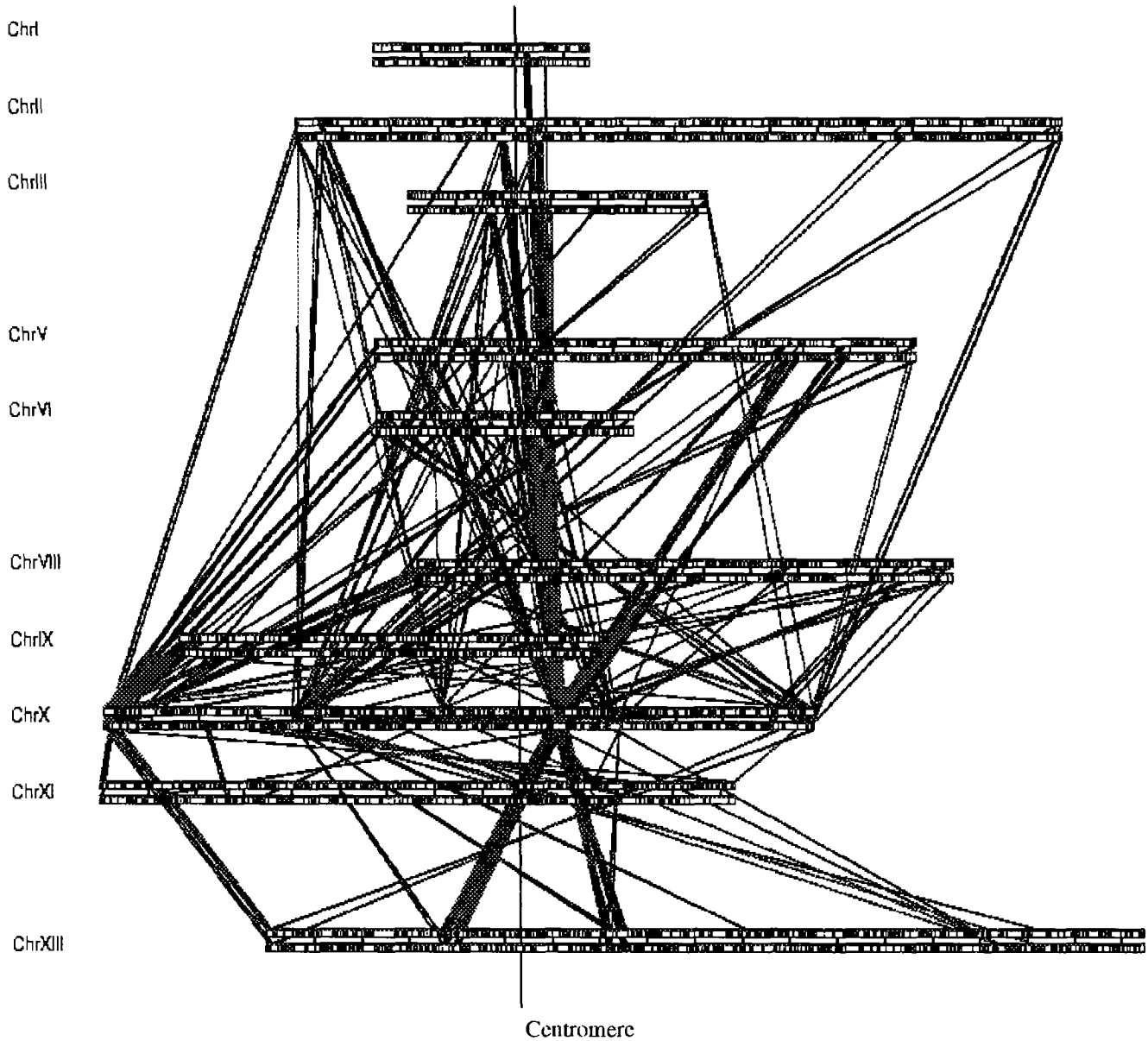

Figure 3: View of the yeast genome as specified in the genome browser form (figure 2). The example shows the matching blocks of chromosome X against all chromosomes. Each pair of horizontal bands represents the two strands of a chromosome. After every 50,000 nucleotides a black tig mark is set between the bands. The bands are sensitive to mouse clicks. Each black region in a band stands for an ORF. Each line connecting two bands represents a similarity relationship between two distinct blocks, on DNA (gray) and protein (black) level. The block size is 500 nucleotides. The DNA/Protein threshold is set to > 300 (about 50% sequence identity). The central horizontal line represents the centromere.

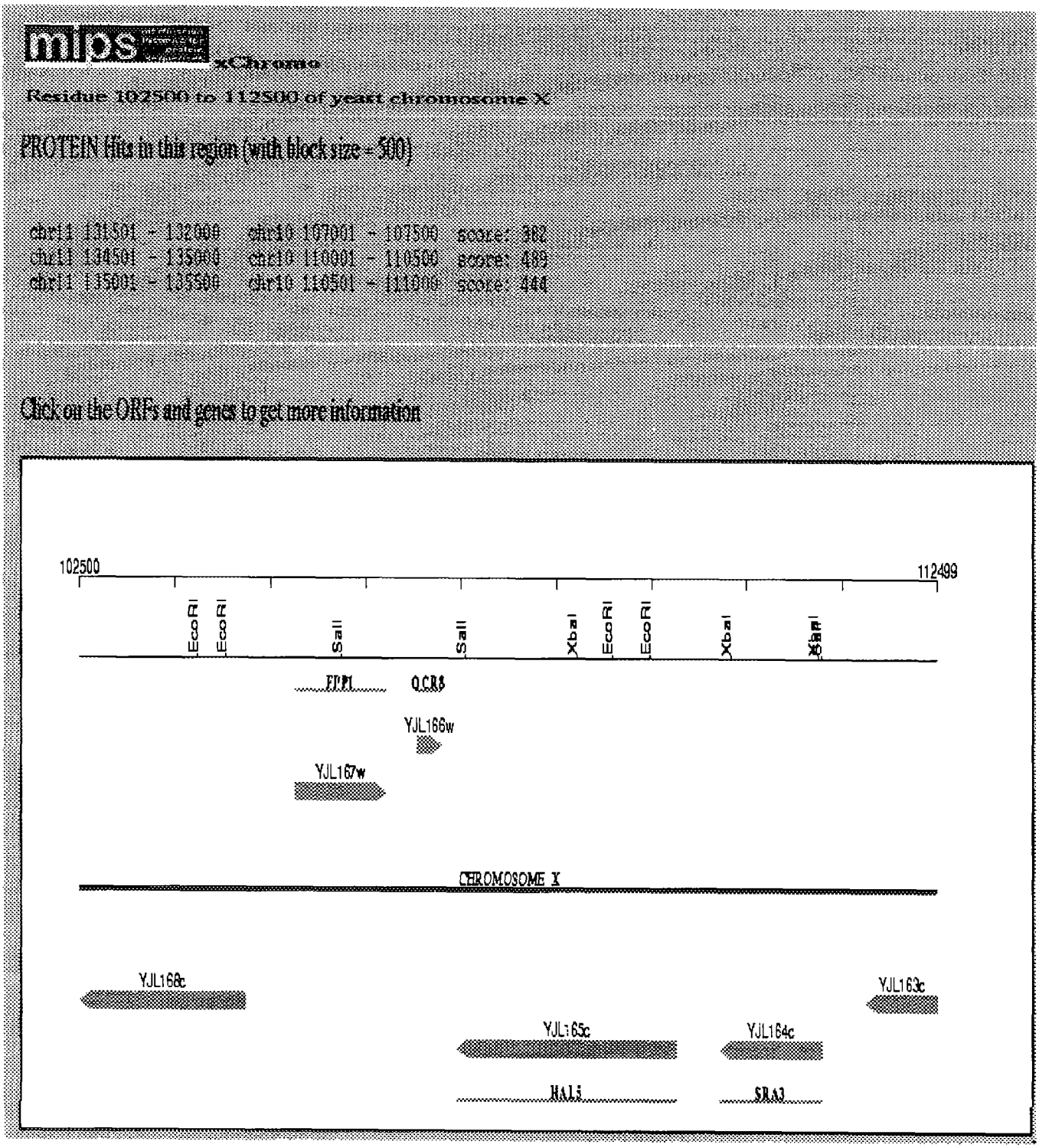

Figure 4: Detailed image of the region (102,500-112,500) of chromosome X selected by mouse click from the view of the yeast genome displayed in figure 3. In the top section a tabular representation of the relationships identified in that region is given. The bottom section provides an XCHROMO view of the region displaying ORFs as pointed rectangles, known genes as thin rectangles and annotated genetic elements (LTRs, deltas, etc.) as arrows.

Service Integration

The interconnection of isolated resources containing multiple data and functions is of major importance in exploring large sets of complex, interdependent biological data. Data and resource integration should be transparent for the user who expects a homogeneous intuitive graphical interface to access services in a uniform way, independent of the platform used and its location. Any genomic sequencing project can be viewed as an application based on multiple services and resources. Up to now, the largest genomic entity to be considered is a single contiguous sequence, treated as an independent isolated unit. However, eukaryotes like fungi, plants and mammals are built on genomes composed of chromosomes, which show common properties in a number of genetic elements (e.g. telomeres, centromeres).

A Study on an integrated services approach to sequence data management has been presented (Heumann et al. 1995). To extend this concept to complete genomes, the genome browser is embedded into this system and allows for an additional level of abstraction. The information related to a genome is made accessible in an interactive way using a set of gateways. A gateway compensates for the constraints imposed by the interface mechanism (e.g. the WWW) without modification of the underlying service. The gateway makes the specific characteristics of services (temporal behavior, statelessness or state dependency and residence on heterogeneous platforms) transparent to the user interface. For this application we require gateways that maintain the program XCHROMO (by S. Liebl, *MIPS*) that provides as a service a detailed graphical image of the genomic information identified in a specific region. For each chromosome we require an independent XCHROMO-gateway. Thus, the genome browser is a broker that links independent resource servers. The browser allows to express specific relationships of interest.

Figure 2 shows a prototype of the WWW form that triggers the filtering mechanism of the genome browser. This form-based user interface hides the specifics of the graph operations and relational algebra from the user. The flexibility of this user interface can be extended easily to allow the user to explore the data on different levels of abstraction. The inspection of genome data is possible by formulating queries about a global property of the genome such as "all ORFs that are duplicated above a certain threshold of similarity". The user can now narrow this selection to any specific class of proteins. Figure 3 gives an example of the view resulting from the GSG specification shown in figure 2.

Figure 3 shows three Tys on chromosome X, one starting near position 197,500 and the paired Tys starting near position 472,500 as regions of dominant duplication on both DNA and protein levels. In addition, the telomeres can be identified as a well known region of frequent duplications. The display represents not only an overview of conserved regions but also provides a mechanism for a more detailed analysis. Below we analyze chromosome X (Galibert et al. 1996) as an example. This chromosome shows several interesting features to be investigated by the browser. Close to position 107,000, a paired duplication to chromosome XI can be identified. Figure 4 gives a detailed XCHROMO display of this region of chromosome X. The user can interactively select this detail by clicking near the corresponding endpoint of the similarity line. The XCHROMO display makes the detailed structure apparent and allows access to annotated textual information by following the dynamic links to other related services. The genome browser is easy to use as a tool to identify clustered duplications of ORFs. Duplication of ORF clusters is a widely distributed feature in yeast. The browser gives direct indications to the location of these clusters. In most cases, a more detailed analysis is required to elucidate the details of ORF duplication, for example between chromosomes X and XI. Analysis of duplication patterns by applying a sliding window technique to the GSG is described hereinafter.

Sliding Window and ORF Duplication

The redundancy of the yeast genome in terms of evolutionarily related ORFs is estimated at 30%. This redundancy can be analyzed in a systematic way. In an other work we have demonstrated that properties of the GSG can be used to characterize genetic elements (Heumann & Mewes 1996c). To narrow the view to investigate weak ORF relationships between pairs of chromosomes, a sliding window of 100,000 nucleotides is chosen to move across chromosome X. For each window duplications that score above 30% identity are displayed. on average, 5% of the blocks of each window are duplicated. However, this value may vary widely for different chromosomes and regions inspected. In general clustered duplication of ORFs is a frequent but specific event. As shown in figure 5, several independent events of duplication can be detected while moving the window along the chromosome.

Figure 5a shows a cluster of collinear duplications of a set of ORFs. Figure 5b and figure 5c shows that parts of the left arm of chromosome X have been duplicated to the right arm of chromosome XI, while figure 5d shows how a cluster of several blocks belonging to a single ORF are duplicated in a collinear way from the left arm of chromosome X to the left arm of chromosome XI. Figure 5d shows how several events overlay in disperse way. Figure 5f shows how a cluster of several blocks belonging to a single ORF are duplicated in a collinear way from the right arm of chromosome X to the left arm of chromosome XI. The distance to the centromere appears to be conserved for some of the duplicated clusters investigated here. This observation needs to be confirmed by the exhaustive analysis of the complete genome sequence.

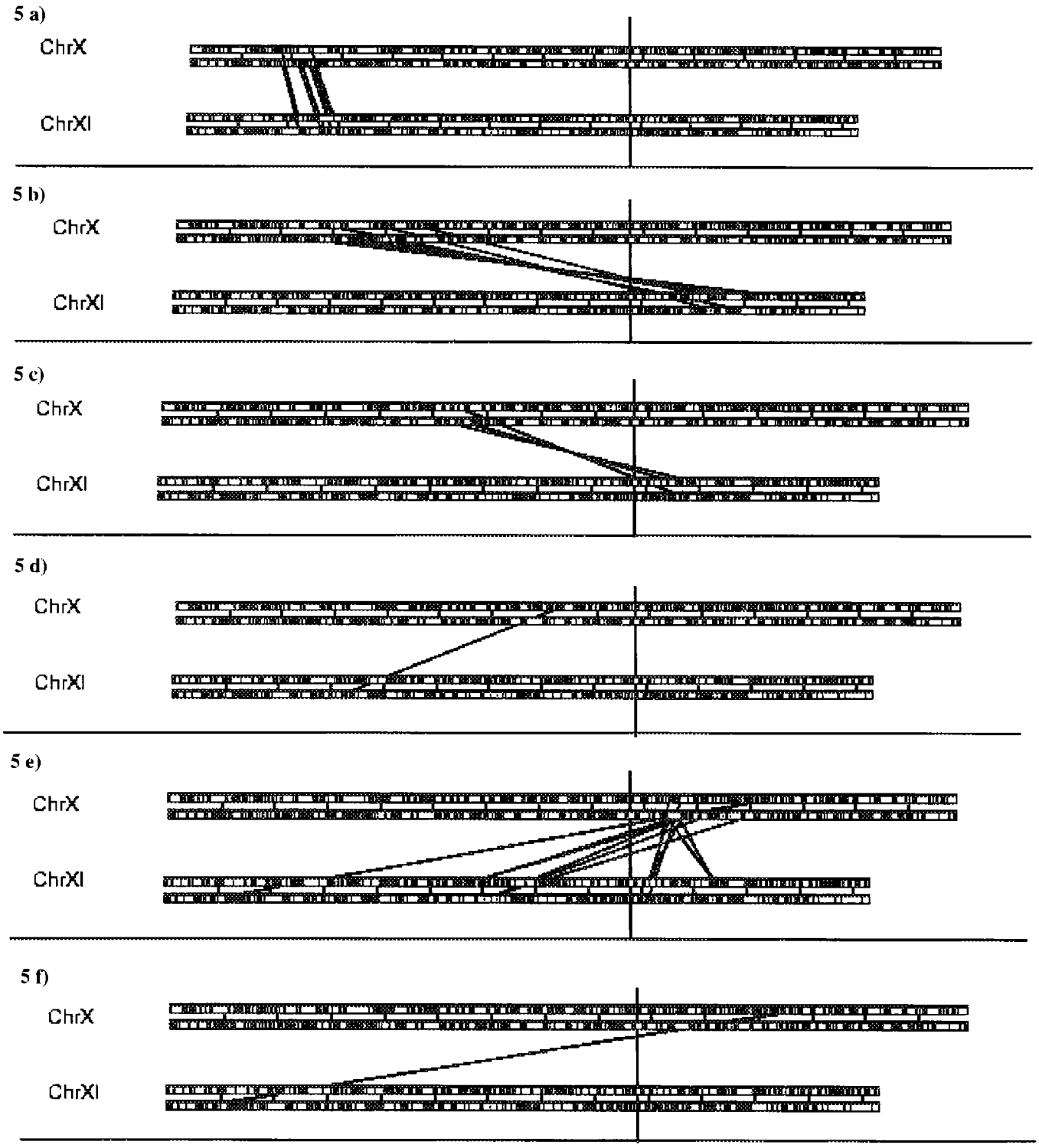

Figure 5: (a-f) Snapshot views of a sliding window (window size 100,000 nucleotides) investigating duplications of coding regions between chromosome X and XI. After every 50,000 nucleotides a black tig mark is set between the bands that represent a chromosome. Each step (a-f) represents a view with a subsequent offset of 100,000 nucleotides, starting at position 50,000-150,000 of chromosome X for figure 5a) and ending at position 550,000-650,000 of chromosome X for figure 5f). The protein threshold is set to > 150 (about 30% sequence identity). Block size is set to 500. The central horizontal line represents the centromere.

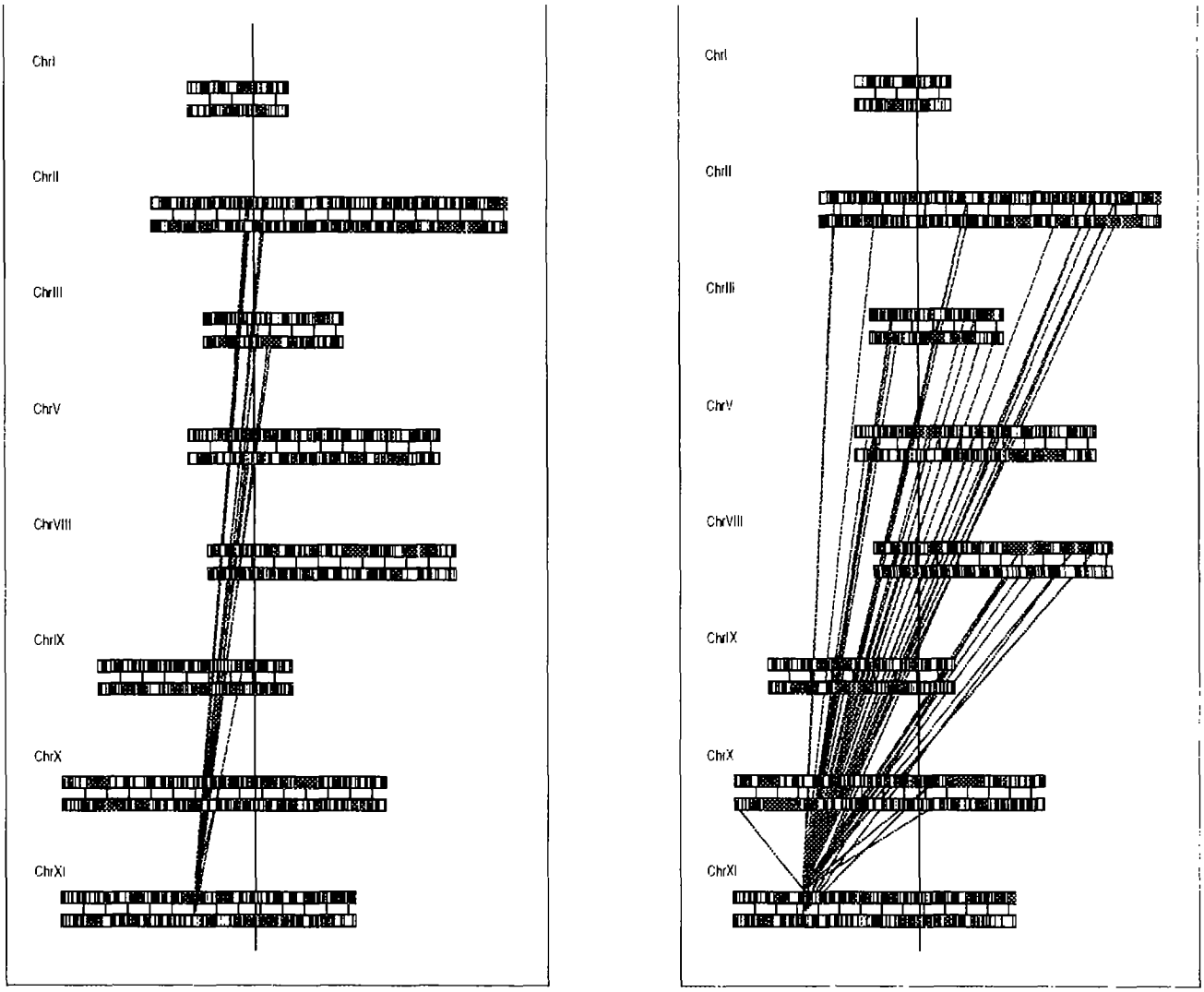

Figure 6: Views of a genome similarity graph to inspect specific intergenic regions (telomeres, ORFs and known genetic elements removed). After every 50,000 nucleotides a black tig mark is set between the bars representing a chromosome. The DNA threshold is set to > 25 (about 50% sequence identity). Block size is set to 50. The central horizontal line represents the centromere. block size set to 50 nucleotides.

a) (LEFT) Block 6041 of chromosome XI (302001-302050).

b) (RIGHT) Block 3298 of chromosome XI (164851-164900).

Intergenic Regions

As mentioned earlier, the HPT can be configured to operate on a lower resolution suitable for the investigation of promoter sites in intergenic regions. For this purpose, a block size of 50 nucleotides is selected. At this resolution we perform an all-against-all comparison of the genome similar to the ORF analysis and represent the result as a GSG. From this GSG we must filter out all coding regions, telomeres and other annotated genetic elements first. Figure 6 shows an example of two 1-to-many block duplications that were selected from the set of all relations identified. We have choosen these two examples by selecting the most frequently duplicated sets of blocks.

Figure 7 a/b gives the corresponding multiple sequence alignment of the pattern found in figure 6. The block set displayed in figure 7a has been identified as part of an LTR (long terminal repeat) that has not yet been correctly annotated. This demonstrates that the method is well suited to detect errors in the annotation and improving the quality of the interpretation of the sequence.

7 a)

```
Chr8_10988 AGGCTATAAT ATTAGATATA CAGAATATAC TAGAAGT.TC TCCTCGAGGA T.......
 Chr8_2679 AGGCTATTAT ATTAGGTATA CACAATATAC TAGAAGT.TC TCCTCGAGGA T.......
Chr10_9672 ........AT ATTAGGTATA CAGAATATAC TAGAAGT.TC TCCTCGAGGA TATAGGAA
 Chr2_5077 ........AT ATTAGGTATA CAGAATATAC TAGAAGT.TC TCCTCGAGGA TATAGGAA
 Chr5_9874 ........AT ATTAGGTATA CAGAATATAC TAGAAGT.TC TCCTCGAGGA TATAGGAA
 Chr5_8892 .......AAT ATTAGGTATA CAGAATATAC TAGAAGT.TC TCCTCGAGGA TATAGGAA
Chr11_6267 ....TATAAT ATT.GGTATA CAGAATATAC TAGAAGT.TC TCCTCGAGGA TATAGG..
 Chr3_5880 .....ATAAT ATTAGGTATA TAGAATATAC TAGAAGT.TC TCCTCGAGGA TATAGG..
Chr11_6041 ....TATAAC ATTAGGTATA CAGAATATAC TAGAAGTGCC TCCTCGAGGA TCTA....
 Chr1_3616 .......... ..TAGGTATA CAGAATATAC TTGAAGGTTC TCCTCGAGGG TCTAGGAA
```

7 b)

```
Chr11_2296 ......ATTC TTACCTTAAG CATCCACTCA TATACATATA TATATATATA TATGTC..
Chr3_1048  ......GTAC TTACCATCTT CTCTCCTTTA TATATATATA TATATATGTA TATTTT..
Chr3_1007  .......... .TA.TATA.. TATATA.T.A TATATATGTA TGTCCATACG .....GGT
Chr5_4467  .........T GTT.TATA.. TATATA.T.A TATATATATA TGTATATACG AACTCGGT
Chr3_4935  ........TT ATG.TATA.. TATATA.T.A TATATATATG CGTAATTATG CA...GAT
Chr10_9116 AC.CAAGTTT ACA.TATA.. TATATA.T.A TATATATATA TATATATATA TATATC..
Chr2_6953  ACACACACAC ACA.CATA.. TATATA.T.A TATATATATA TATATATATA TATAT...
Chr2_13921 ...AATATAT ACA.TATA.. TATATA.T.A TATATATATA TATATATATA TATGTACA
Chr2_13921 .......TGT ACA.TATA.. TATATA.T.A TATATATATA TATATATATA TATGTATA
Chr10_9116 .........G ATA.TATA.. TATATA.T.A TATATATATA TATATATATA TATGTAAA
Chr2_12803 ......ATAT ACA.TACA.. TATATA.T.A TATATATATA TATATATACA TCTTTTGA
Chr9_3550  ...TACATTT ATG.CACA.. TATATA.T.A TATATATATA TGTATATGTA AATGTATA
Chr2_693   ...TTATTAA AAT.AAGT.. GATATA.T.A CATATATATA TATATATATA TATATATA
Chr8_9317  AGAGTATTTT AGG.AATT.. TATATA.T.A TATATATATA TATATATATA TATAT...
Chr2_12803 ....GGTTCA AAA.GATG.. TATATA.T.A TATATATATA TATATATATG TATGTATA
Chr8_6802  ......TTTT GTGCTATA.. TATATA.T.A TATATATATA TATATATATA ACATAGAG
Chr8_6802  ....GACTCT ATGTTATA.. TATATA.T.A TATATATATA TATATATATA GCACAAAA
Chr11_3298 ......ATAT AT...ATA.. TATATA.T.A TATATATATA TATATATGTA ACTTATTT
Chr2_693-2 .......TAT AT...ATA.. TATATA.T.A TATATATATA TGTATATATC ACTTATTT
Chr8_9317  .......... AT...ATA.. TATATA.T.A TATATATATA TATATATATA AATTCCTA
Chr2_6953  ......ATAT ATA.TATA.. TATATA.T.A TATATATATA TATATATGTG TGTGTGTG
Chr5_5599  ......ATAT ATA.TATA.. TATATA.T.A TATATATATA TGTGCGTGCG TGTGTGTG
Chr3_4471  ......TTAT ATA.TATA.. TATATA.T.A TATATGTGTG TTTGTATAC. TCTGTGGG
Chr11_1577 ......ACAT .TA.TATA.. TATATA.T.A TATATATATG TTT......G TGTGTGTA
Chr11_11588.........A GTGCTATG.. AATATATT.A CATATATATA TATATATATG T.TGTGTA
Chr5_3403  .......... AGGCGATAGC GATAAAGA.A GATATATATA TATATATATG TATGAAGA
```

Figure 7: Multiple sequence alignment of the duplicated blocks shown in figure 5 (using the standard program PILEUP).

a) (TOP) Block 6041 chromosome XI: contains a motif of a LTR.

b) (BOTTOM) Block 3298 chromosome XI: resident between ORFs YKL152c and YKL151c conatins a TATA-box.

It is also suitable to generate patterns corresponding to genetic elements. In fact, the multiple alignment is a profile generated without assumptions. Figure 7b gives a typical example of the frequent TATA-box signal found in the intergenic regions. In order to identify novel promoter sites, filtering techniques must be applied to allow for more complex context information to be included. The functional classification of the corresponding ORF has been shown to correlate with certain promotor types (Quandt et al. 1996). The application of HPT as a fast retrieval engine enables the development of a sensitive methodology for promoter detection as outlined. We expect that the integration of these more advanced features of the GSG (sliding window and multiple resolution) to the genome browser will allow for easier data exploration in these fields of high interest.

Summary

A visualization technique for whole genome data based on the HPT has been introduced. It combines a versatile data structure suitable for the computational exploration of the data with adequate visualization techniques. We have given only a few examples to outline the inherent potential of the method. The analysis of genomic sequence data by the top-down approach allows for the systematic investigation of data generated by independent experiments. Further systematic, detailed analysis will provide valuable information for the subsequent experimental analysis, e.g. the disruption of complete clusters or duplication of individual members duplicated. The ease of tuning parameters as scope, window size, and sensitivity allows the user to apply the methodology for genome analysis in a versatile, interactive way to detect yet undiscovered important biological features.

Acknowledgments

This work was supported by funds from the European Commission (BIO2-CT93-0003). MIPS is grateful for the financial support through the Max-Planck-Society and the Forschungszentrum für Umwelt (GSF).

References

Aho, A., Hopcroft, J., and Ullman, J., 1974 *The design and analysis of computer algorithms.* Mass.: Addison-Wessley.

Bieganski, P., 1995. Genetic Sequence Data Retrieval and Manipulation based on Generalized Suffix Trees. Ph.D. diss. Dept. of Computer Science University of Minnesota.

Bork, P.; Ouzounis, C.; Sander, C.; Scharf, M.; Schneider, R.; and Sonnhammer, E. 1992 Comprehensive sequence analysis of the 182 predicted open reading frames of yeast chromosome III. *Protein Science* 1:1677-1690.

Dujon, B.; et al. 1994 The complete sequence of chromosome XI of Saccharomyces Cerevisiae. *Nature* 396:371-378.

Feldmann, H.; et al. 1994 Complete DNA-Sequence of Yeast Chromosome-II. *EMBO JOURNAL* 13: 5795-5809.

Ferragina, P.; and Grossi, R. 1995 A Fully-Dynamic Data Structure for External Substring Search. STOC'95.

Galibert, et. al. 1996 The complete Sequence of Saccharomyces cerevisiae chromosome X. *EMBO Journal.* in press.

Gonnet, G.; Mark, A.; and Benner, S. 1992 Exhaustive Matching of the Entire Protein Sequence Database. *Science* 256:1443-1445.

Heumann, K.; Harris, C.; Kaps, A.; Liebl, S.: Maierl, A.; Pfeiffer, F.; and Mewes, H.W. 1995 An Integrated Services Approach to Biological Sequence Databases. In Bioinformatics (edts.) Schomburg, D.; and Lessel, U.: From Nucleic Acids and Proteins to Cell Metabolism; GBF Monographs. 18:3-16.

Heumann, K.; and Mewes, H.W. 1995 Fragment Assembly Project. *DIMACS Workshop'95 (*1995). Forthcoming.

Heumann, K.; and Mewes, H.W. 1996a The Hashed Position Tree (HPT), String Matching and Local Sequence Similarity for Large Database searching. Forthcoming.

Heumann, K;. and Mewes, H.W. 1996b Matching of the Human EST against the Open Reading Frames of Yeast. Forthcoming.

Heumann, K.; and Mewes, H.W. 1996c Analysis and Visualization of Complete Genomes. Forthcoming.

McCreight, E.M. 1976 A space-economical suffix tree construction algorithm. *J. As soc. Comp. Mach.* 23:262-272.

Mewes, H.W.; and Heumann, K. 1995 Genome Analysis: Pattern Search in Biological Macromolecules. *Lecture Notes in Computer Science.* 937:261-285.

Ozier-Kaleogeropoulos, O.; Richard, G.-F.; Perrin, A.; Fairhead, C.; Gaillon, L.; and Dujon, B. 1995 The Yeast Genome: Structural Redundancy. In Proceedings of the Seventh International Conference on Yeast Genetics and Molecular Biology; Lisboa Portugal.

Quandt, K.; Frech, P.; Karas, H.; Wingender, E.; and Werner, T. 1996 MatInd and MatInspector - New fast sensitive tools for detection of consensus matches in nucleotide sequence data. *NAR.* in press.

Ukkonen, E. 1993 On--line Construction of Suffix Trees. Report A--1993--1, University of Helsinki, Finland, Department of Computer Science.

Weiner, P. 1973 Linear pattern matching algorithms. In Conference Record, IEEE 14th Annual Symposium on Switching and Automata Theory, 1-11.