Fall 2010

# Finding Duplication Events Using GenomeVectorizer

Elena Kochetkova
*San Jose State University*

# FINDING DUPLICATION EVENTS USING GENOMEVECTORIZER

A Thesis

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Elena Kochetkova

December 2010

The Designated Thesis Committee Approves the Thesis Titled

FINDING DUPLICATION EVENTS USING GENOMEVECTORIZER

by

Elena Kochetkova

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

December 2010

| | |
|---|---|
| Dr. Tsau Young Lin | Department of Computer Science |
| Dr. Johnny Martin | Department of Computer Science |
| Eric Louie | IBM |

ABSTRACT

FINDING DUPLICATION EVENTS USING GENOMEVECTORIZER

by Elena Kochetkova

GenomeVectorizer is a software application designed to extend the functionality of GenomePixelizer, a genome-visualization tool that was developed for the Department of Plant Pathology at University of California, Davis, in 2002. GenomeVectorizer was written using XML, XSLT, and SVG technologies combined with JavaScript scripting to provide the level of flexibility, dynamism, and interactivity not supported by the TCL/TK written application (GenomePixelizer). This new visualization tool was tested with available data from the *Arabidopsis* NBS-LRR study, and its output was compared to the output of GenomePixelizer. The relationships drawn at the same identity value were identical.

GenomeVectorizer was successfully applied to study NBS-LRR genes and duplication events in *Glycine max* (soybean). The images of NBS-LRR genes were generated at 50, 60, and 70 percent identity. Images also showed the relationships between the duplication events. At a glance, it was easy to determine that duplication regions include almost half of the genome.

Currently, the user can generate an image at a specified percent identity, highlight gene relationships by clicking on the identity value inside the identity matrix, visit a gene's database entry, and drag chromosomes away from each other.

# ACKNOWLEDGMENTS

I would like to extend a huge thank you to Dr. Johnny Martin, a teacher whose passion for the subject of the class was contagious and who led me to fall in love with open-source, not-quite-bug-free technology that has a lot of potential. A huge thank you also goes to my supervisor at UC Davis, Dr. Alex Kozik, who not only gave me the opportunity to participate in exciting projects (one of which led to the conception of this thesis), but also provided me with a steady source of employment for a number of years.

My deep appreciation goes to my thesis advisor, Dr. Tsau Young Lin, for his patience, great advice, and encouragement.

Finally, I would like to thank my sister Anna for always being available for help and for always being an example of scholastic excellence, my husband, Joe Ryan, for his patience, compassion, and moral support, and my newborn daughter for giving me an extra boost of motivation to finish my degree before her birth.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# I. BIOLOGICAL BACKGROUND

## A. *Homology*

Finding homology is important for tracing the evolution of living organisms. Homology means similarity in structure due to common ancestry. Genes related by homology are called homologs, and homologs are divided into two sub-categories: orthologs – genes related due to a speciation event, and paralogs – genes related due to a duplication event [1].

## B. *Genome Duplication Events*

Gene duplication is believed to play a major role in evolution [2]. As Hurles [3] points out, this role is evidenced through "the widespread existence of gene families." The gene duplication process creates a new copy of a gene that is not subject to selective pressure. This paralog can mutate without negative consequences for the organism and can potentially boost genetic resistance to disease or code for a new function [4].

Duplication events in plants are studied very extensively, since plants are “the most prolific genome duplicators” [4]. *Arabidopsis thaliana* has experienced at least two rounds of genome duplication, the recent one occurring about 24-40 million years ago [5].

## C. *Resistance Genes*

In plant genomes, resistance genes (R genes) are responsible for plant disease resistance against pathogens [6]. Michelmore and Meyers [7], in their review of a "birth-and-death process" model for R gene evolution, postulated that "the defense system of

plants may be ancient and predate the evolution of the immune system." Similarities have been identified between proteins coded by R genes in different plant species [8]. There were also findings of similar genes in mammals [7, 9].

R genes encode a number of protein motifs; the most prevalent class contains NBS-LRR protein motifs [7]. The NBS (nucleotide binding site), a common protein motif in all organisms, is thought to be important for ATP or GTP binding [10, 11]. LRR (leucine rich repeats) proteins appear to be involved in protein-protein interactions (important for signal transduction, cell adhesion, DNA repair, recombination, transcription, RNA processing, disease resistance, and ice nucleation) [12, 13].

NBS-LRR proteins also contain the TIR (toll interleukin 1 receptor) domain, which is thought to "play a signaling role during resistance responses mediated by TIR-containing R proteins" [14].

### D. *Overview of Homology Finding Algorithms*

In genetics, sequence alignment is used to align two or more DNA (or protein) sequences that are suspected to be homologous and to find the regions of conservation. Any difference in the produced alignment is due to mutation during evolution (insertion or deletion of nucleotides from the sequence). A DNA (or protein) sequence with unknown structure and function could also be aligned or searched against a sequence with known structure and function. If the two sequences produce a high-quality match, the protein structure and function of the unknown sequence are assumed to be those of the known sequence.

Depending on whether a pair of sequences or multiple sequences need to be aligned, pairwise or multiple sequence alignment techniques are used. "Percent identity" is the degree of similarity of two or more sequences. If sequences have high percent identity, they are likely homologous.

A pairwise sequence alignment is a comparison of two sequences. There are two types of computational techniques used for alignment: local sequence alignment and global sequence alignment. Local sequence alignment is used for finding repeating regions within the same sequence or regions of similarity within dissimilar sequences. The purpose of global sequence alignment is to produce the best match over the entire length of two relatively similar sequences. Dynamic programming techniques are used for pairwise sequence alignment. Figure 1 shows an example of pairwise sequence alignment for two human zinc finger proteins, identified on the left by the GenBank accession number [15].

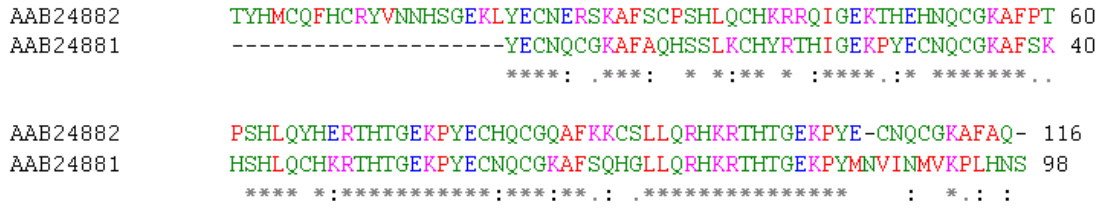


Figure 1. A sequence produced by ClustalW [15].

Single letters: amino acids,
Red: small, hydrophobic, aromatic, not Y,
Blue: acidic,
Magenta: basic,
Green: hydroxyl, amine, amide, basic,
Gray: others,
"*": identical,
":": conserved substitutions (same color group),
".": semi-conserved substitution (similar shapes). [24

Aligned regions of sequences AAB24881 and AAB24882 have 83.7% identity, which could be an indication that the two sequences are homologous. The large gap in the alignment (positions 1-20 in AAB24882) is an indication that there was a large insertion/deletion event produced by evolution.

A multiple sequence alignment is used to align more than two sequences that are hypothesized to be evolutionarily related. Some of the goals of a multiple sequence alignment are to determine phylogenetic relationships and trace evolution, to determine conserved regions, and to determine the overall structure of a protein. ClustalW [16] is one of the popular tools used for multiple sequence alignment. ClustalW operates in three steps:

1. performs pairwise sequence alignments (dynamic programming techniques are used at this step and the identity matrix is created),
2. creates guide tree (phylogenetic tree) based on the identity matrix ("distance" matrix) values, and
3. builds progressive alignment (series of multiple pairwise alignments are progressively aligned following the branching order of a phylogenetic tree).

# II. INTRODUCTION

GenomeVectorizer was introduced for the first time in 2009 at SVG Open Conference as an example of the application of XML, XSLT, SVG, and Javascript technologies to the creation of visual solutions for biological research. It was presented under the name "GenomePixelizer SVG-fied," named so after its prototype, GenomePixelizer. The name GenomeVectorizer was suggested by a co-creator of the GenomePixelizer tool, Alex Kozik, and better reflects the nature of the generated graphics: vector images as oppose to pixel mapping.

There are many GenomeVisualization tools available either as stand-alone programs or as Web applications, including Argo Genome Browser [17], Circos [18], and Alfresco [19].

While all of the above-mentioned tools have strengths, they all have stale interfaces; all interactivity is implemented by means of links and pop-up windows. The user cannot drag clusters of genes away from chromosomes or drag chromosomes in order to better view the relationships between the genes.

GenomeVectorizer is lightweight, dynamic, and interactive. It lets the user mouse over a gene or chromosome in order to view the name, click on the gene in order to visit the database entry related to it, and drag a chromosome up or out in order to better view the relationships between the genes.

## III. GENOMEPIXELIZER – THE PROTOTYPE

The prototype tool, GenomePixelizer, was released in 2002 for the Department of Plant Pathology at UC Davis, led by R. W. Michelmore. This tool was developed specifically for studying the evolution of NBS-LRR encoding genes in *Arabidopsis* [20], the plant whose genome was completely sequenced by the end of 2000 [21] and that serves as a model for the study of plant genetics. NBS-LRR genes represent the major class of disease-resistance genes in flowering plants [22]. Their evolution in relation to other genome duplication events is of great importance and interest to the scientists studying plant genetics.

GenomePixelizer was designed to visualize "the relationships between duplicated genes in genome(s) and to follow relationships between members of gene clusters" [20]. This tool "generates custom images of genomes out of the given set of genes. Each element on the picture has a physical address defined by coordinates (pixels), hence the name 'GenomePixelizer'" [20].

Figure 2 shows the main interface of GenomePixelizer.

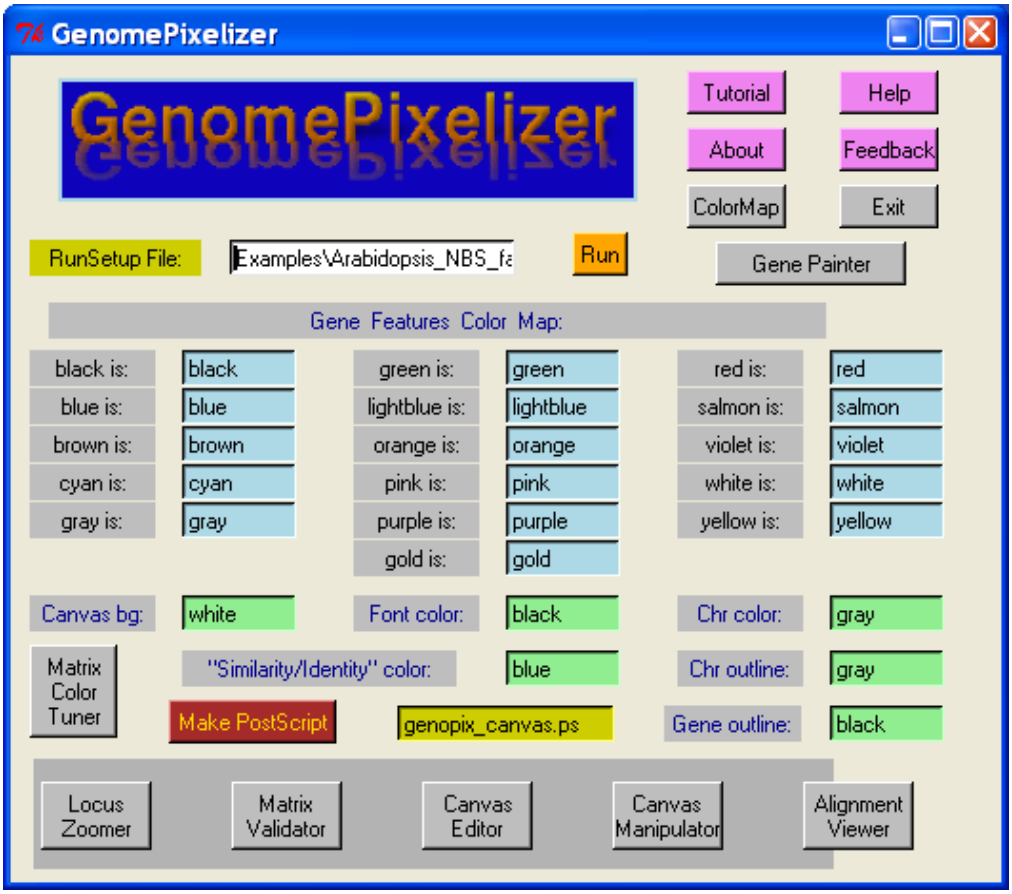


Figure 2. GenomePixelizer main interface.

The main interface of GenomePixelizer serves a dual purpose:

- a dialog where the user provides RunSetup file information – the file that contains all the necessary input information in order to run the tool (names of the files containing gene locations and distance matrix information, number of chromosomes and their sizes, percent identity, and other information), and
- an editing tool allowing the user to edit the colors of chromosomes and genes, background colors, and font color.

Figures 3 and 4 show examples of the visualizations produced by GenomePixelizer.

Figure 3 represents NBS, P450, PK-LRR clustering in the *Arabidopsis* genome, with NBS genes colored in orange, P450 genes colored in green, and PK-LRR genes colored in purple; the genes' relationship is shown at 75% identity.

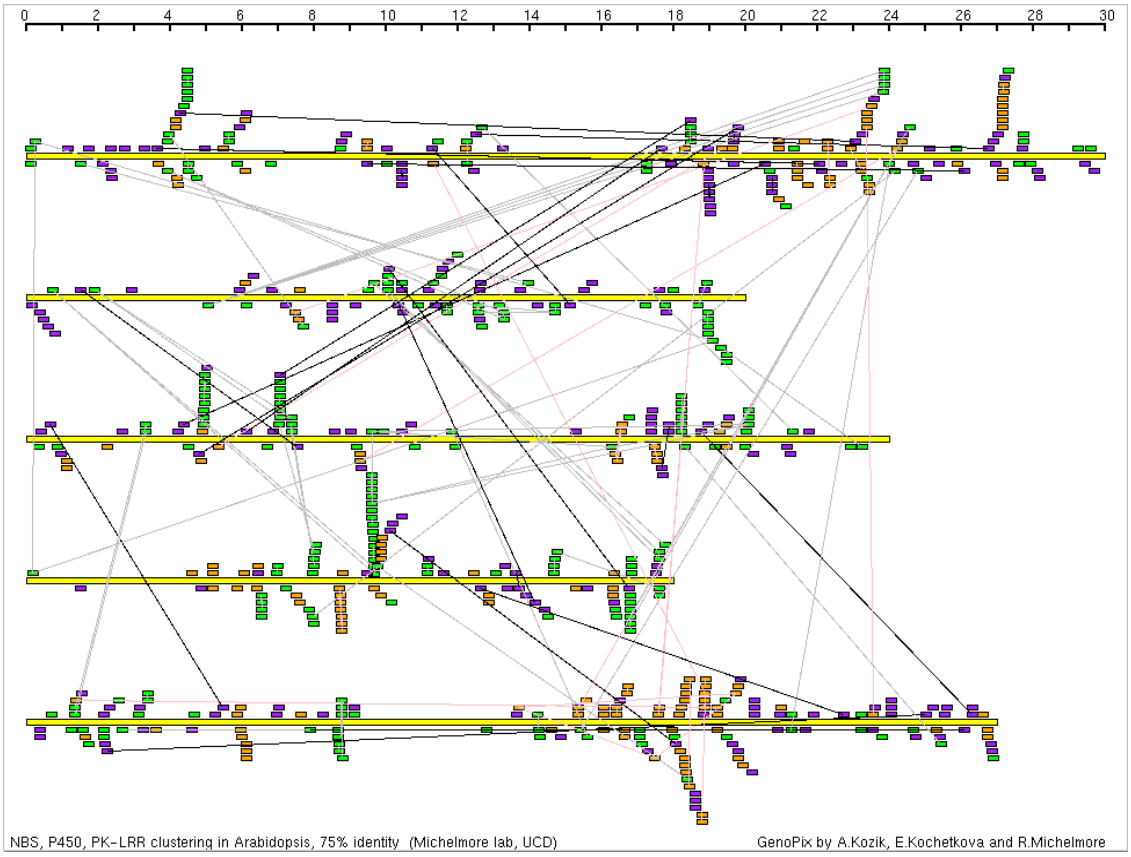


Figure 3. Output produced by the GenomePixelizer tool [20].

Figure 4 shows the zoom-in feature of GenomePixelizer.

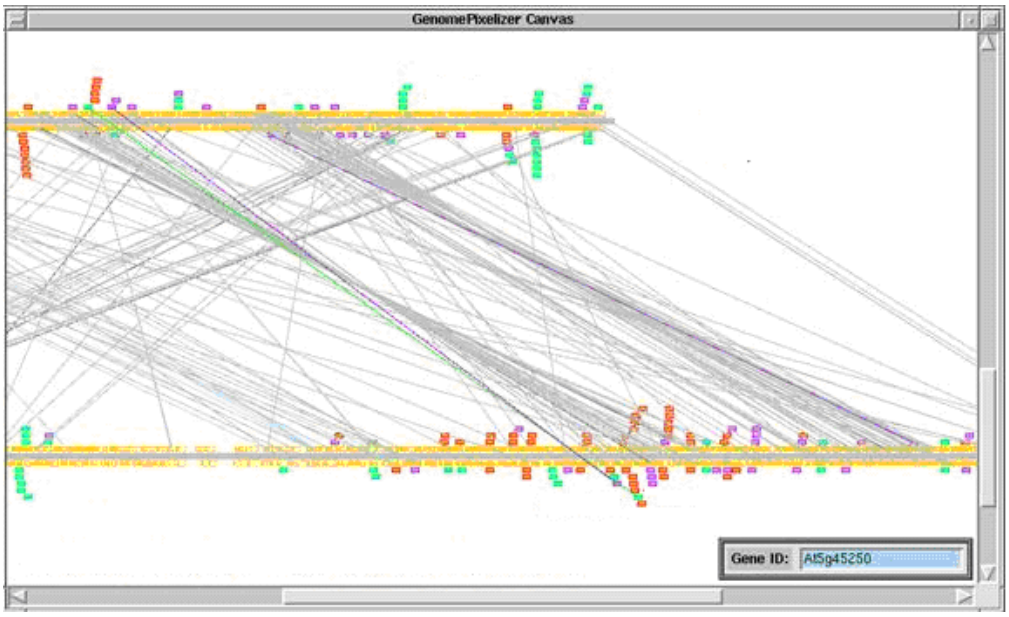


Figure 4. Output produced by GenomePixelizer – *Arabidopsis thaliana*, segmental duplications of chr IV and chr V [20].

Figure 5 shows the interface for zoom-in functionality, which is quite complicated and may appear confusing at first sight.

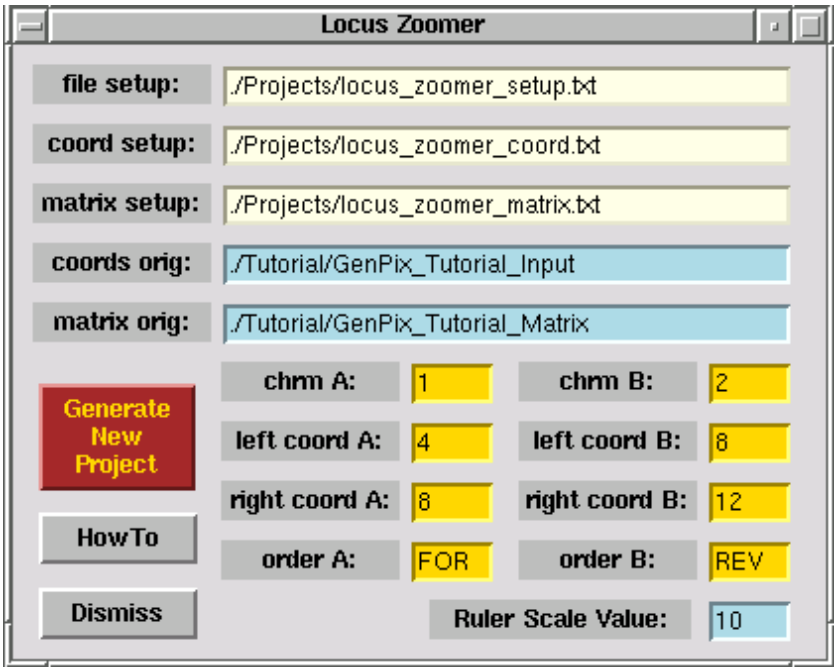


Figure 5. GenomePixelizer tool – zoom-in functionality interface.

## IV. GENOMEPIXELIZER VS. GENOMEVECTORIZER

Table 1 summarizes the main functionalities of both visualization tools and highlights similarities and differences between the tools.

TABLE 1. DIFFERENCES BETWEEN GENOMEPIXELIZER AND GENOMEVECTORIZER

| | **GenomePixelizer** | **GenomeVectorizer** |
|---|---|---|
| **Platform-independent** | **Yes** | **Yes** |
| **Browser-interpreted** | **N/A** | **Yes** |
| **Need to download and install language environment** | **No – Windows (latest release comes in a form of an executable file)**<br>**Yes – other systems** | **No** |
| **Allows for a quick view of a whole genome** | **Yes** | **Yes** |
| **Zoom-in** | **Zoom-in functionality is coded and is available through separate interface** | **Zoom-in functionality is activated by pressing "Ctrl" and "+" in a browser** |
| **Regions with high gene density can be drawn using automatic or manual correction.** | **Yes** | **There is no provision for manual correction yet** |
| **Allows the viewing of relationships between different sets of genes based** | **Yes** | **Yes + allows viewing of relationships between genes located on the** |

| | | |
|---|---|---|
| **on a distance matrix file.** | | **same chromosome** |
| **The source of sequences is not restricted to a single organism and it is possible to view relationships between different genomes.** | **Yes** | **Yes** |
| **Can be used to generate images of genetic maps with a given set of genetic markers.** | **Yes** | **Not implemented** |
| **Generated images can be captured by any screenshot program and incorporated into Web pages. The generated image can also be saved as a PostScript file.** | **Yes** | **Not implemented** |
| **Can generate HTML ImageMap tags. This feature can be used to create "clickable" images for Web pages or online presentations.** | **Yes** | **Not implemented** |

Genome Pixelizer is a TCL/TK written stand-alone application that runs on any computer platform (Unix/Linux, Windows, Mac) that supports the TCL/TK toolkit [20]. GenomeVectorizer is written using XML, XSLT, and SVG technologies combined with JavaScript scripting. All these technologies are browser interpreted and do not require download and installation of a language environment. Some browsers may require the

download of an SVG plug-in [23].

Like GenomePixelizer, GenomeVectorizer provides a zoomed-out view of the whole genome [23].

GenomePixelizer takes in three input files. Batch information contained there can easily be manipulated in spreadsheet applications, such as MS Excel or StarOffice [20]. In GenomeVectorizer, a single XML input file is required and can easily be manipulated [23].

In GenomePixelizer, zoom-in functionality is a semi-automated process in which the user must specify the coordinates of the desired region. Zoom-in functionality for GenomeVectorizer is built into the browser and is activated by pressing the "Ctrl" and "+" keys simultaneously (Figure 8). No extra coding is required.

In GenomePixelizer, regions with high gene density can be drawn using automatic or manual correction; however, manual correction is rather time consuming for large sets of genes [11]. GenomeVectorizer does not allow for manual correction [23].

Like GenomePixelizer, GenomeVectorizer allows for viewing the relationships between different genomes [23].

# V. GENOMEVECTORIZER

## *A. Implementation*

### *1) Data Sources*

The input to the program is a single XML file: Input.xml. The data is represented there in two parts:

1. *Information about chromosomes*: chromosome ID and size (in Mb) and information about each gene located on this chromosome: gene name, location, Watson/Crick orientation, as well as color assigned to it (black is designated to show gene duplication regions between chromosomes). Location may be provided as:
   - averaged location between a gene's start and end positions, or
   - gene-region start position wrapped in <gne_loc_start> </gne_loc_start> tags and gene-region end position wrapped in <gne_loc_end></gne_loc_end>, or
   - averaged position combined with the beginning and the end of the gene region.

   Below is the example of the entry:

```
<chromosome id="1" size="20000000">

        <gene color="orange">
                <gne_name>Gene_K</gne_name>
                <gne_location>6200000</gne_location>
                  <gne_loc_start></gne_loc_start>
                <gne_loc_end></gne_loc_end>
                <gne_orientation>C</gne_orientation>
        </gene>
```

```
<gene>
        <gne_name color="orange">Gene_W</gne_name>
        <gne_location>6400000</gne_location>
        <gne_orientation>C</gne_orientation>
        <gne_loc_start></gne_loc_start>
        <gne_loc_end></gne_loc_end>

</gene>
…
```

2. *Distance matrix:* distance information between two genes. Distance data are obtained by means of the applications that perform multiple-sequence alignment, like ClustalW.

```
<matrix>

    <row><gene_a>Gene_A</gene_a><gene_b>Gene_E</gene_b><dist>0.9857</dist></row>
    <row><gene_a>Gene_A</gene_a><gene_b>Gene_U</gene_b><dist>0.9286</dist></row>
    <row><gene_a>Gene_A</gene_a><gene_b>Gene_Y</gene_b><dist>0.8429</dist></row>
                                                    …
```

Currently, these input data are populated manually. The single XML file that GenomeVectorizer uses as input replaces three input files that the original TCL/TK-based GenomePixelizer uses: *Setup File*, providing information about the widget's window size, number of chromosomes, size of chromosomes, cutoff values, etc., *Input File*, containing chromosome number, gene name, gene's location on the chromosome, orientation, and color and *Distance Matrix File*, containing pairs of genes and their distance ("similarity") [23].

### 2) *Code Design*

The graphical portion of GenomeVectorizer is written using XPATH, XSLT, and SVG. Interactivity is provided through JavaScript methods. The code is contained in four

files: parser.xsl, drawingtools.xsl[1], show_gene_tip.js[2] and loadxmldoc.js[3].

- parser.xsl – creates SVG viewbox element, parses out information about chromosome and genes using XPATH queries and sends it to drawingtools.xsl for drawing objects on canvas. It parses out information about distance matrix and creates a table with distance values, allowing for interactivity between the table and the SVG canvas.
- drawingtools.xsl - contains XSL templates and SVG code for drawing grid, chromosomes, and genes and for displaying synteny between genes.
- show_gene_tip.js – displays information based on the mouse-over events, according to Table 2. This file also contains code that provides the ability to drag chromosomes (along with genes that are grouped with them) about the canvas.
- loadxmldoc.js - loads XML document into DOM structure.

TABLE 2. INTERACTIVITY FEATURES CORRESPONDING TO THE MOUSE-OVER EVENTS

| **Mouse-Over Event** | **Action** |
|---|---|
| Genes | Display gene names. |
| Chromosomes | Display chromosome number. |
| Synteny (connecting lines) | Display names of the connected (similar) genes. |

[1] Layout of these files taken from dinosaurs' bar graph example, found in http://surguy.net/articles/client-side-svg.xml.

[2] Tool Tip code is taken from http://svg-whiz.com/svg/Tooltip2.svg.

[3] loadxmldoc.js is taken from http://www.w3schools.com/DOM/dom_loadxmldoc.

## 3) *Main Algorithm*

```
xsl:stylesheet

xsl:template match="genome"

  create svg viewbox area

  xsl:call-template name="graphStyles"
  xsl:call-template name="graphFilters"
  xsl:call-template name="drawLines"

  xsl:for-each select="//chromosome

    select chromosome id and size

    svg:g id="{$chrom_id}"
          xsl:call-template name="drawChromosome”

          xsl:for-each select="//chromosome[$chrom_id]/gene"
               xsl:call-template name="drawGenes"
           end  xsl:for-each
    end svg:g
  end xsl:for-each

  svg:g id='ToolTip'
    rectangle with text in it containg tipTitle and tipDesc elements
  end svg:g

end xsl:template

xsl:template match="/"
  <html>
    <head>
         var xmlDoc=loadXMLDoc("Soybean_NBS_LRR.xml");

         for (var j=0;j&lt;matrix.length;j++)

             fill up two dimentional  matrix with distance values

             if (dist &gt;= percent)
                   draw_synteny(gene_a_name, gene_b_name);

         end for

    </head>

    <body onload="Init();">

       Init() activates ToolTip and dragability

           <form id="form" name="id">

    create button" onClick="window.location.reload()"
                  create button   onClick="show_dist_matrix()”


           </form>
    </body>
  </html>

</xsl:template>

</xsl:stylesheet>
```

## *B. Visualization*

### *1) Quick Overview*

The resulting visual is an SVG graph that plots chromosomes, places genes over chromosomes according to their specified locations, and draws lines connecting genes with a "similarity" value that is higher than the cutoff value.

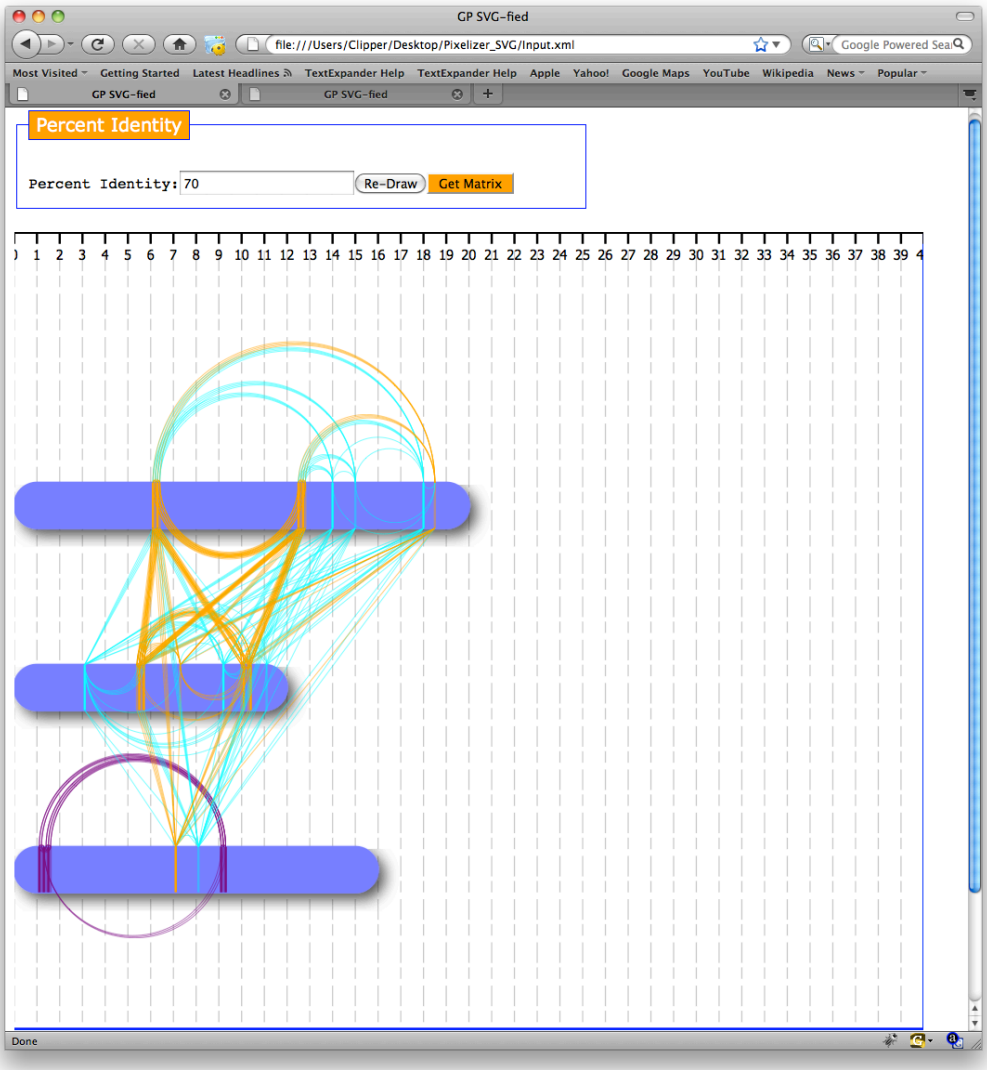


Figure 6. Sample output produced by GenomeVectorizer.

*2) Description*

The chromosomes are drawn according to their sizes in megabases (Mb). One grid interval represents 1 Mb. In Figure 6, there are three chromosomes of sizes 20, 12, and 16 Mb. Genes are placed inside chromosomes according to their averaged locations ( (gene start position – gene end position) / 2 ). The opacity of the genes indicates Watson/Crick orientation. Genes with Watson (forward) orientation are represented with solid colors, and genes with reverse orientation are represented with colors that are 40 percent opaque. The "similarity" of the genes is represented by means of lines and arcs: straight lines if genes are "similar" to genes on different chromosomes, and arcs if genes are "similar" to genes on the same chromosome. Similarity cutoff value (percent identity) is provided by the user.

Figure 7 shows the output produced by GenomeVectorizer when run on the *Arabidopsis* NBS genes dataset. Here we can see five chromosomes of sizes 30, 20, 24, 18, and 27 Mb. The cyan color represents TIR, NBS, and LRR-positive genes; the green color, TIR and NBS-positive genes; the orange color, NBS-positive genes; and the pink color, NBS and LRR-positive genes. Forward-oriented genes are shown in solid colors, and reverse-oriented genes are shown with 40 percent opaque colors. The identity cutoff value is 70 percent.

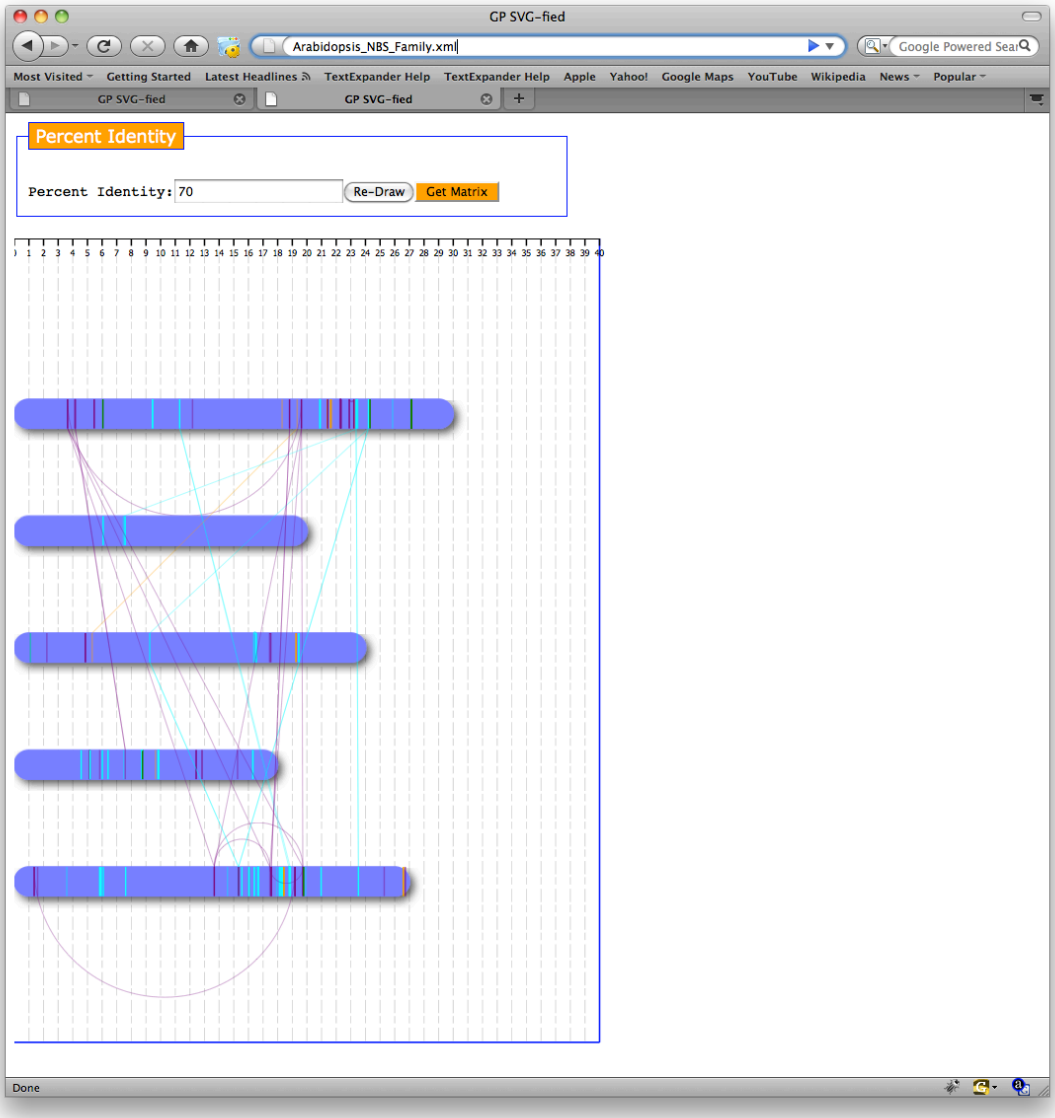


Figure 7. Output produced by GenomeVectorizer.

*3) Scalability*

Users can zoom into the area of interest by pressing “Ctrl” and “+” buttons simultaneously (Figure 8).

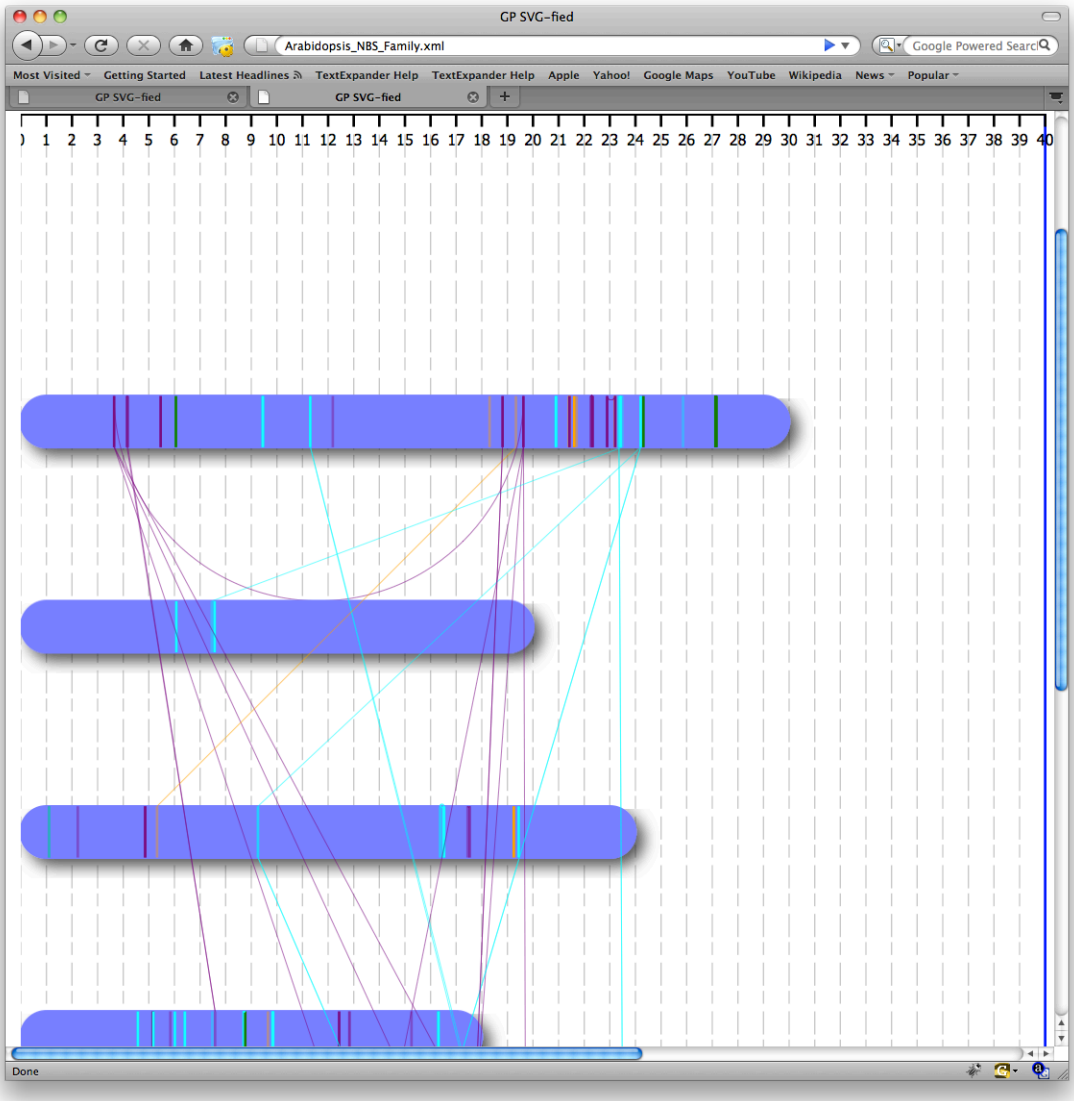


Figure 8. Output produced by GenomeVectorizer after user zooms in.

*C. Interactivity*

Once the user enters the percent identity and clicks the "Retrieve" button, SVG representation of XML data is displayed and a new browser window pops up displaying the identity matrix (Figure 8). Identity values greater than the user-specified percent identity are shown in black, and identity values that are lower than the percent identity are grayed out.

Once the user clicks on the "Get Matrix" button, the matrix containing gene distance values opens. The values below the cutoff value are grayed out, and the values above the cutoff value are visible and clickable.

The user can click on any identity value inside the matrix; the color of the cell containing that value turns red and the line or arc connecting two genes in the graph is highlighted in red and becomes bolder (Figures 9 and 10). Once the user moves a mouse over the chromosome or over the line within the chromosome representing the gene, the chromosome or gene name is displayed; the same thing happens when the user moves a mouse over a line connecting two genes: a popup displays, showing which two genes are connected. Once the user clicks on a gene, he or she will land on a database entry (NCBI Genbank, TAIR, or other) related to this particular gene (Figure 11).

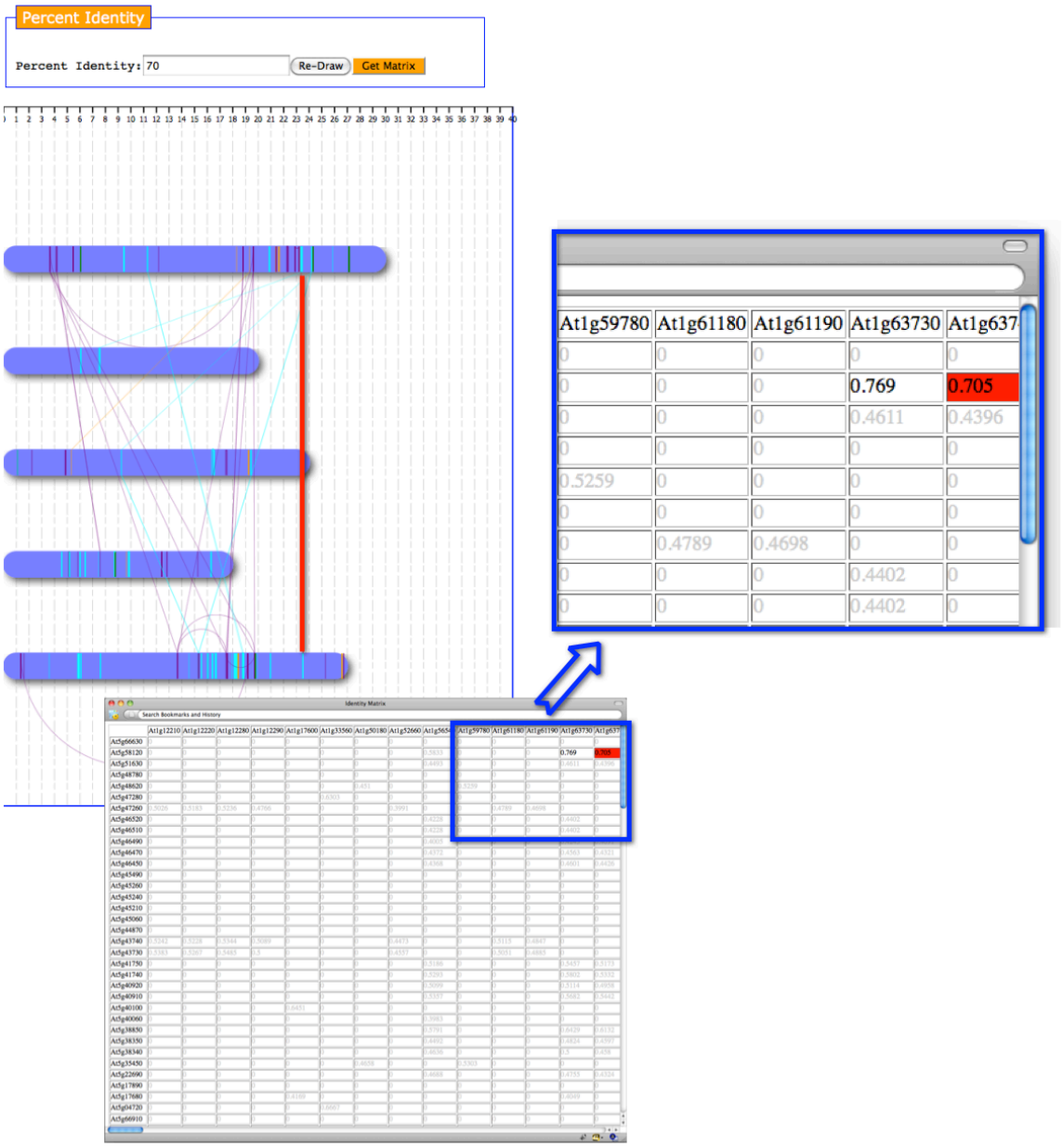


Figure 9. GenomeVectorizer – Interactivity.

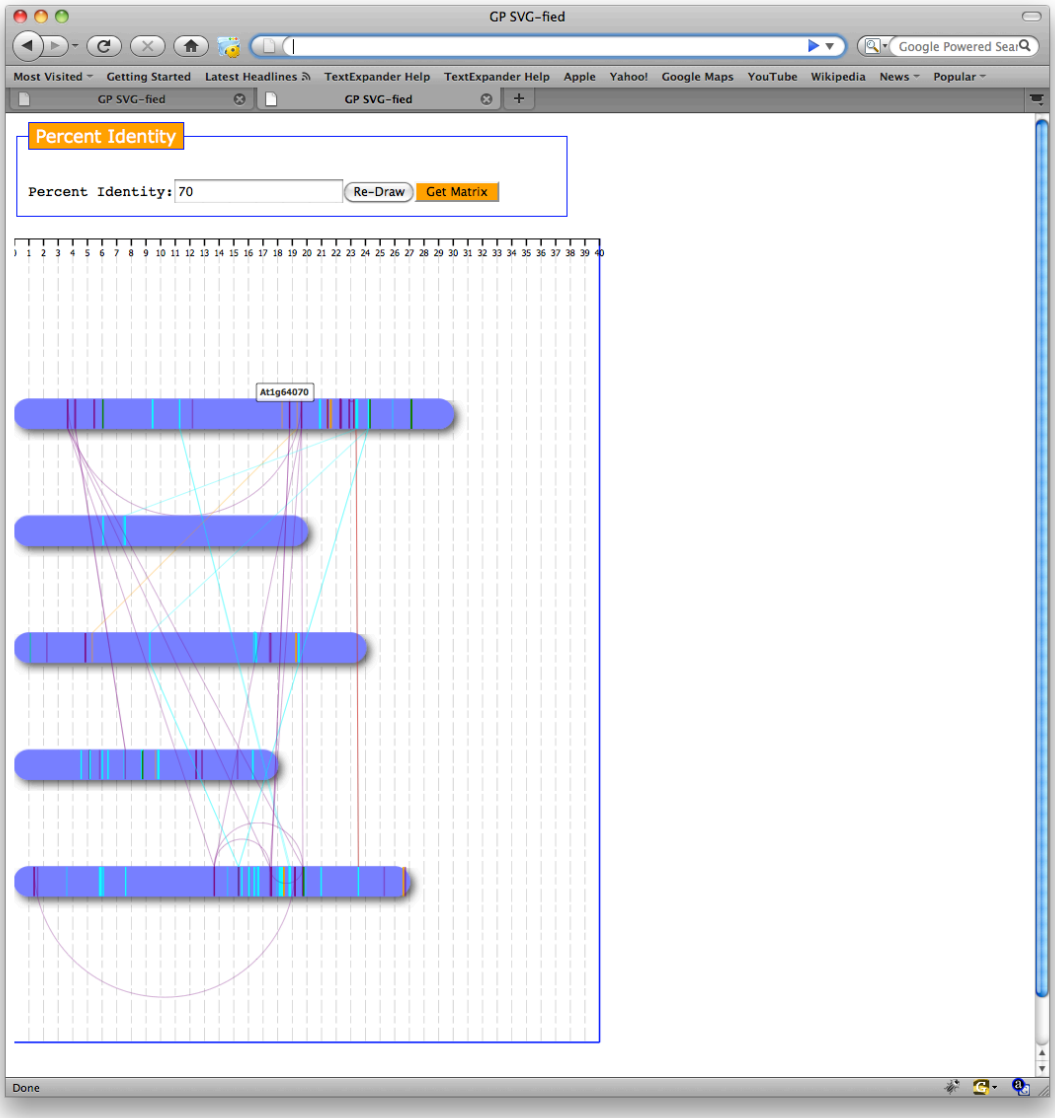


Figure 10. GenomeVectorizer – Interactivity (continued).

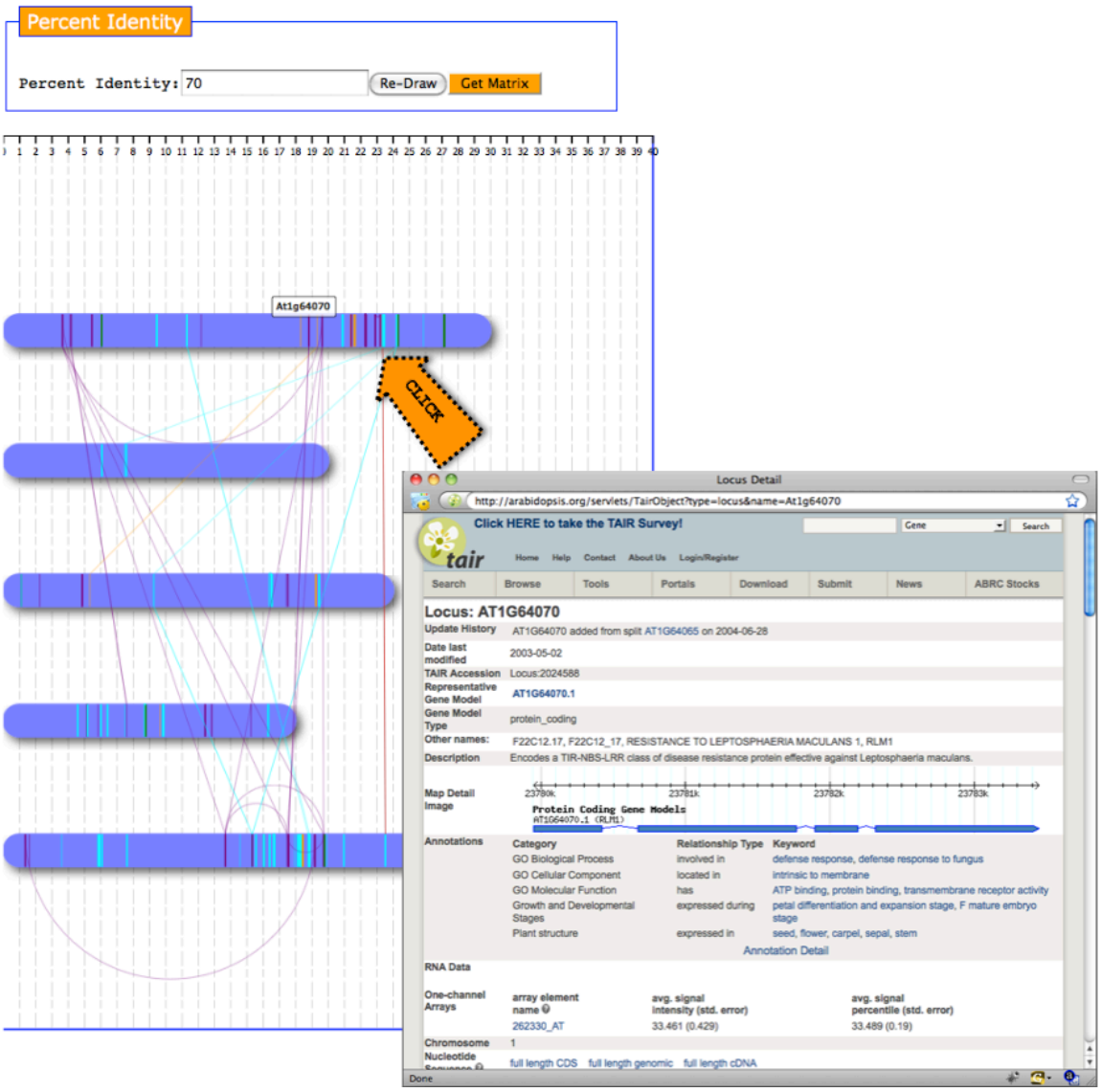


Figure 11. Database information about particular gene.

One downside of GenomeVectorizer at this point is that overlapping genes are not visible. The author has tried to represent genes depending on the gene lengths (heights) in order to distinguish overlapping genes (Figure 20); however, gene heights relative to the overall size of the chromosome are very insignificant. The chromosome sizes would need

to be five times larger for easy viewing of variation in gene heights and tracking of the relationships. In the future, the stacked gene design implemented in GenomePixelizer (Figure 3) will be considered. Also, implementing genes that have 'C' (reverse) orientation with lower opacity does not make them stand out. The author is still exploring different ways of representing genes that have reverse orientation.

The dragging functionality is partially implemented and allows for dragging chromosomes around the canvas in order to see gene relationships better.

# VI. APPLICATION OF GENOMEVECTORIZER

## A. *Arabidopsis thaliana NBS Family*

GenomePixelizer was designed in 2002 for gene analysis in *Arabidopsis thaliana* (NBS-LRR-encoding genes in particular [22]), a plant the whole genome of which was sequenced in December of 2000 [21].

The *Arabidopsis thaliana* NBS family dataset was downloaded alongside GenomePixelizer executable, released October 1, 2003 [20], and used to test GenomeVectorizer.

Figure 12 shows the visual produced by means of GenomeVectorizer. Figure 13 shows the output of GenomePixelizer, both at 70% identity. Gene relationships, represented by lines and arcs in GenomeVectorizer and by lines in GenomePixelizer, are identified in identical way in the figures.

As previously mentioned, the main drawback of GenomeVectorizer, which will be addressed in the second release of the tool, is the inability to visualize the overlapping regions.

A big advantage of GenomeVectorizer is that the "similarity" relationships between the genes located on the same chromosome are instantly obvious.

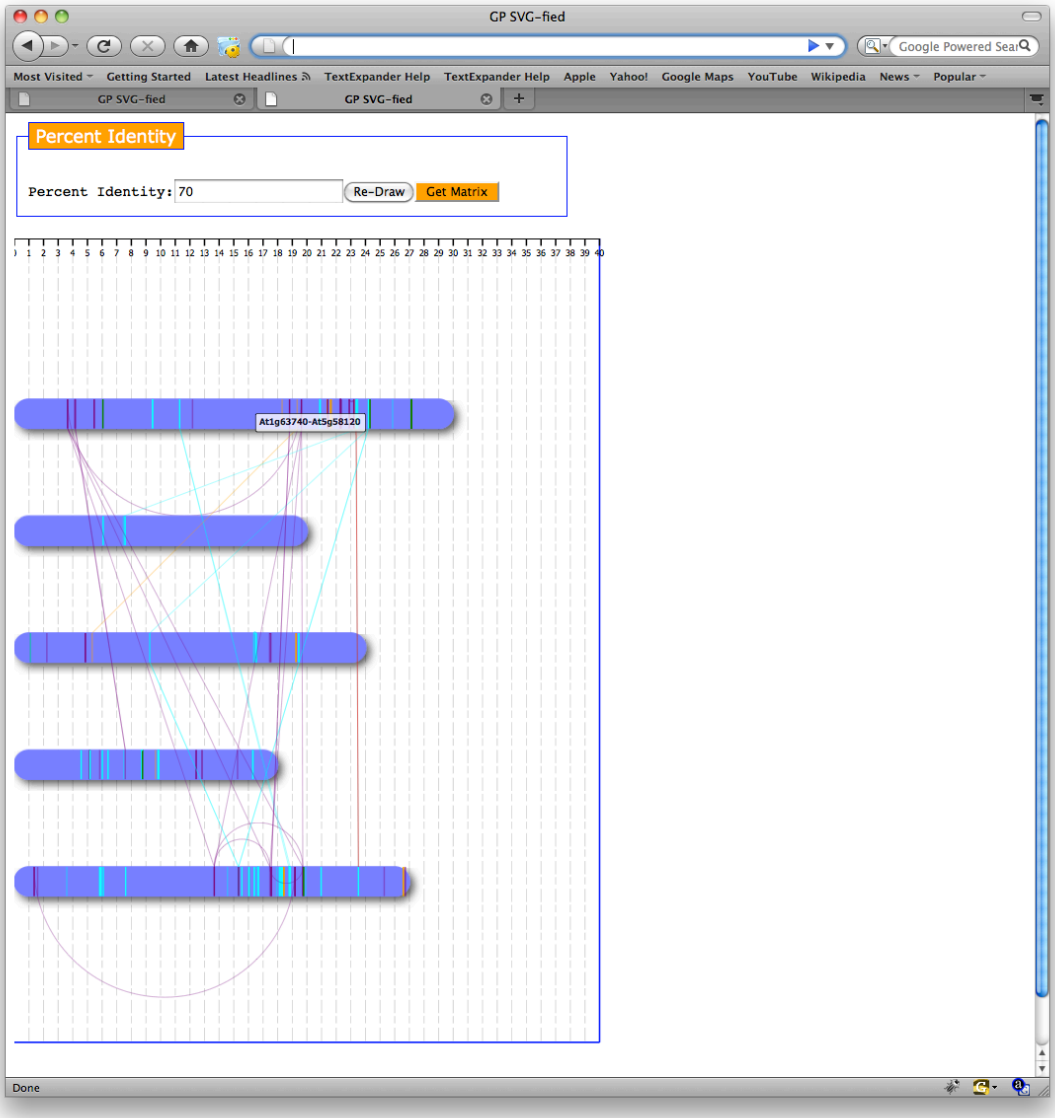


Figure 12. GenomeVectorizer - *Arabidopsis thaliana* NBS family. 70% identity.

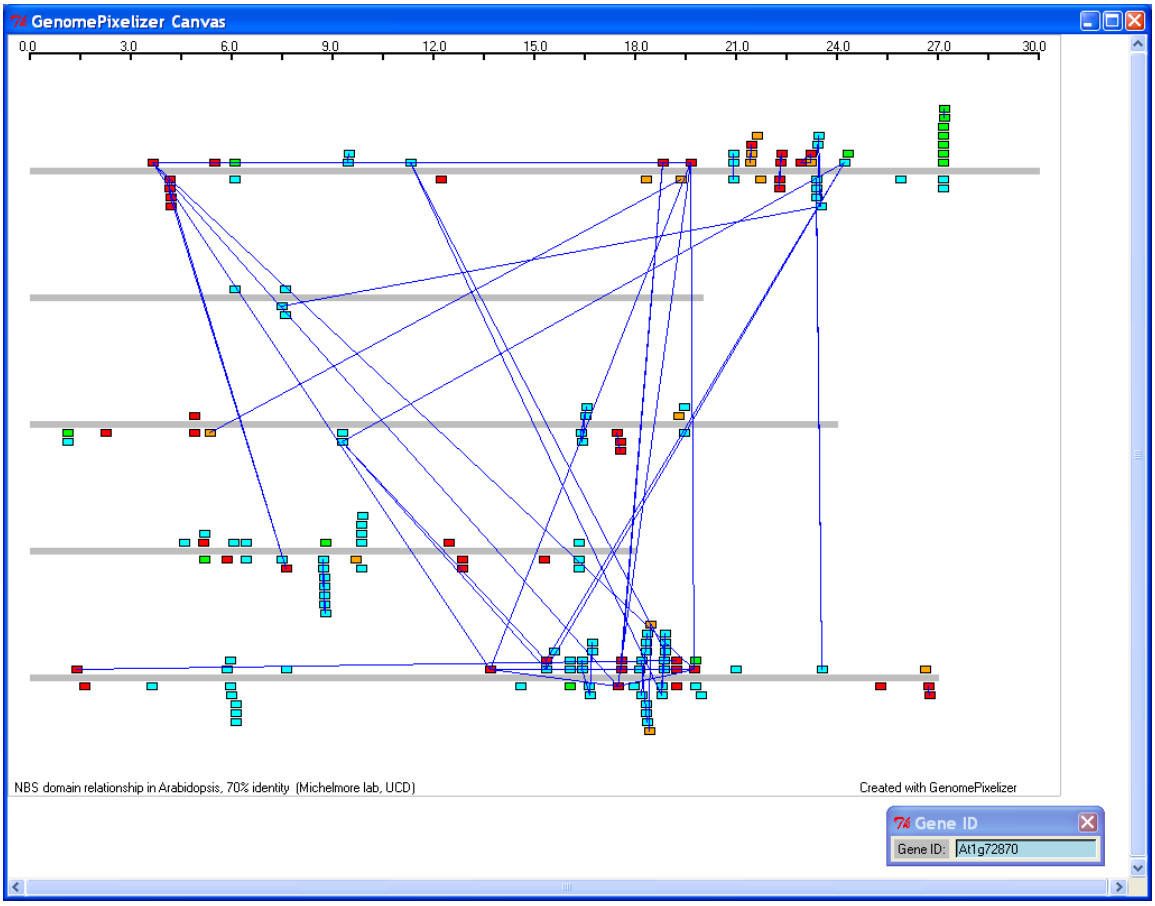


Figure 13. GenomePixelizer - *Arabidopsis thaliana* NBS family. 70% identity.

### B. *Soybean NBS-LRR Family*

GenomeVectorizer was successfully used for analysis of soybean NBS-LRR genes. The dataset was provided by Leah McHale, Ohio State University. The first ten chromosomes (out of twenty total) of soybean are shown in Figures 14, 15 and 16, with NBS-LRR genes displayed at 50, 60 and 70% identity, respectively. In this sample display, dark yellow color is chosen for TIR genes, orange represents TIR pseudogenes, while non-TIR genes are not shown.

This visualization also builds on improved technique of visualizing gene duplication

regions. We observe black opaque squares, which, as mentioned in section V.A.1, represent duplicated regions, as well as the black opaque polygon lines connecting them, which represent 100% identity between each pair of duplication regions. As we can see, the duplicated regions include almost half of the genome.

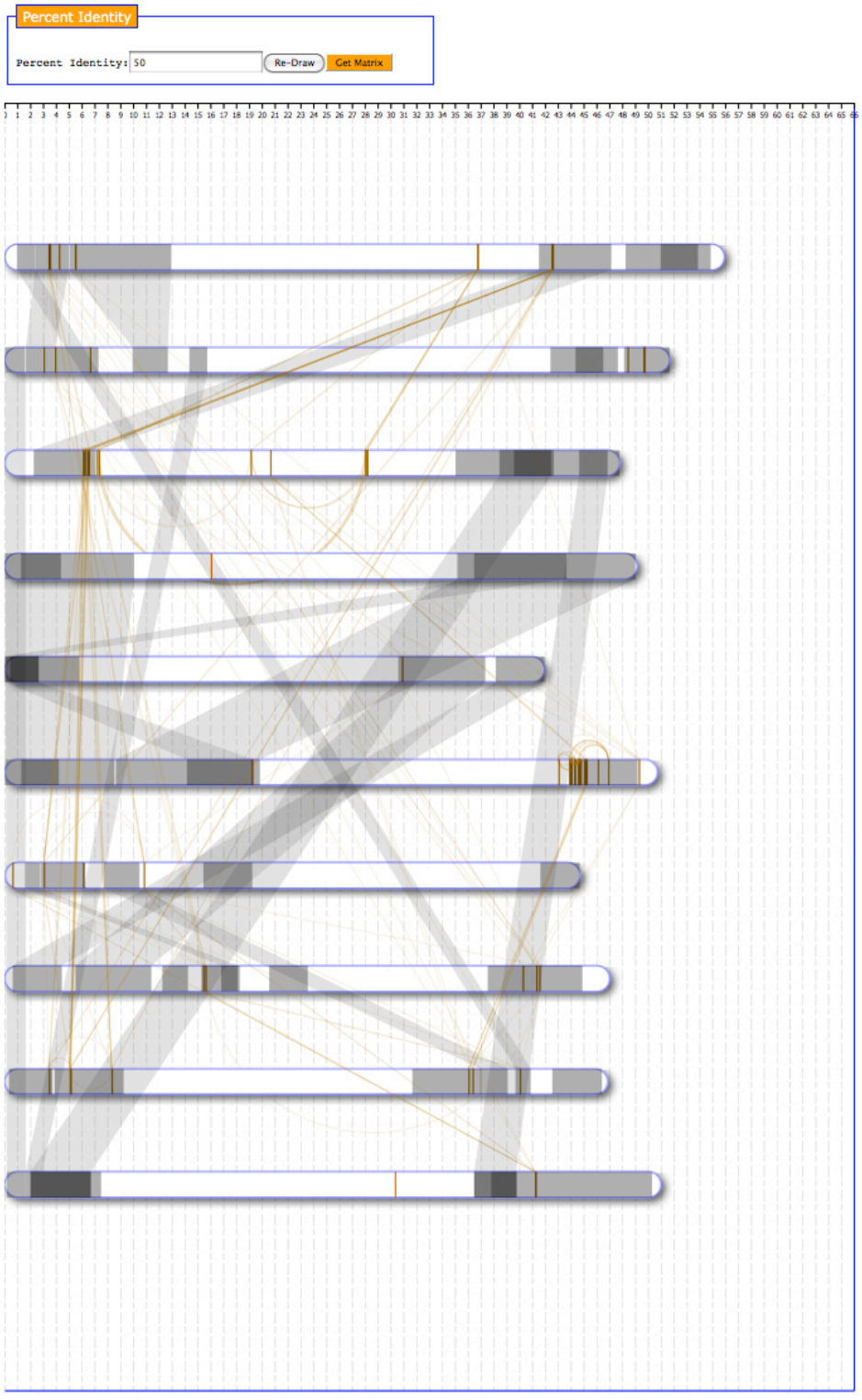


Figure 14. Soybean NBS-LRR genes visualization. 50% identity. TIR genes only.

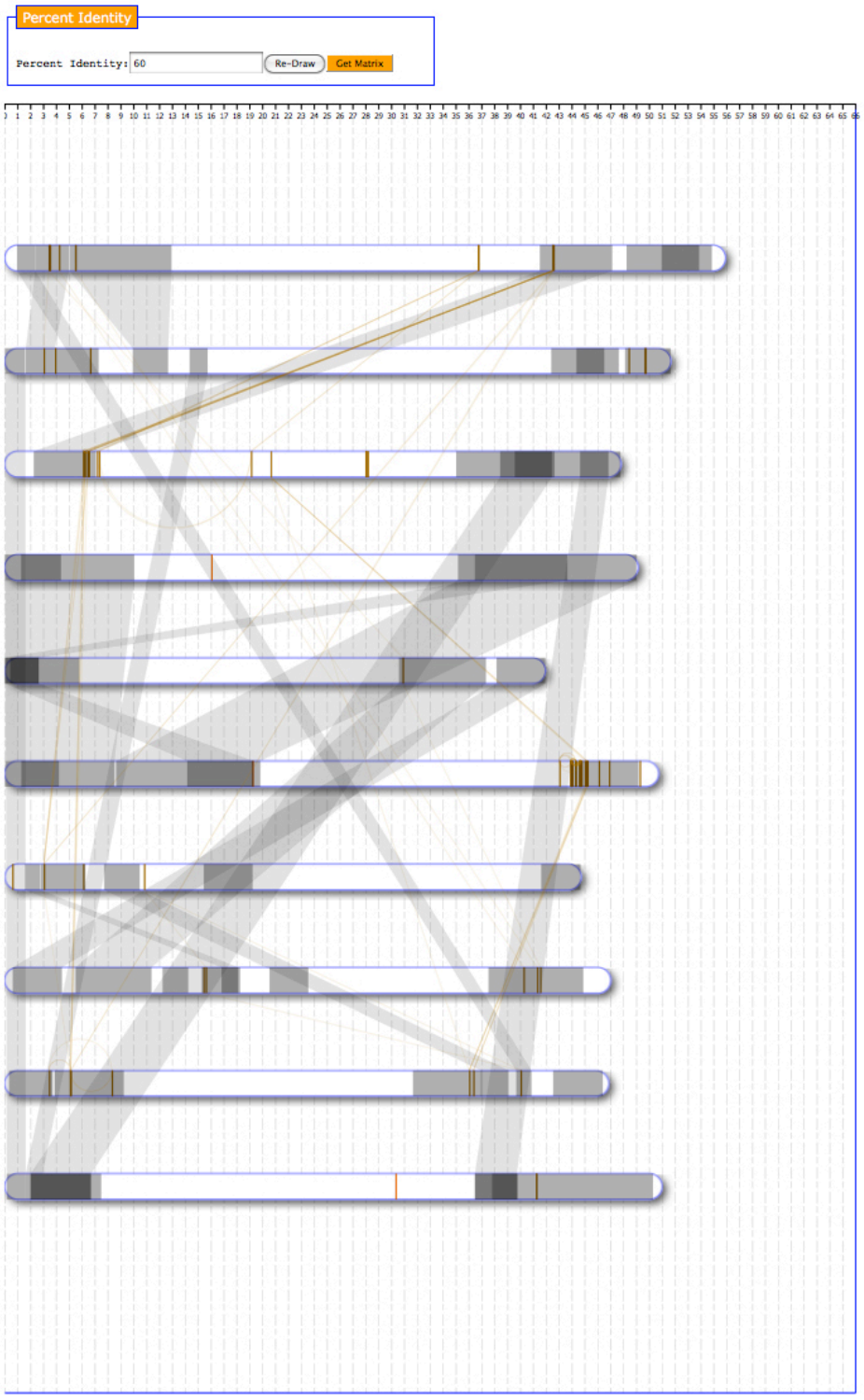


Figure 15. Soybean NBS-LRR genes visualization. 60% identity. TIR genes only.

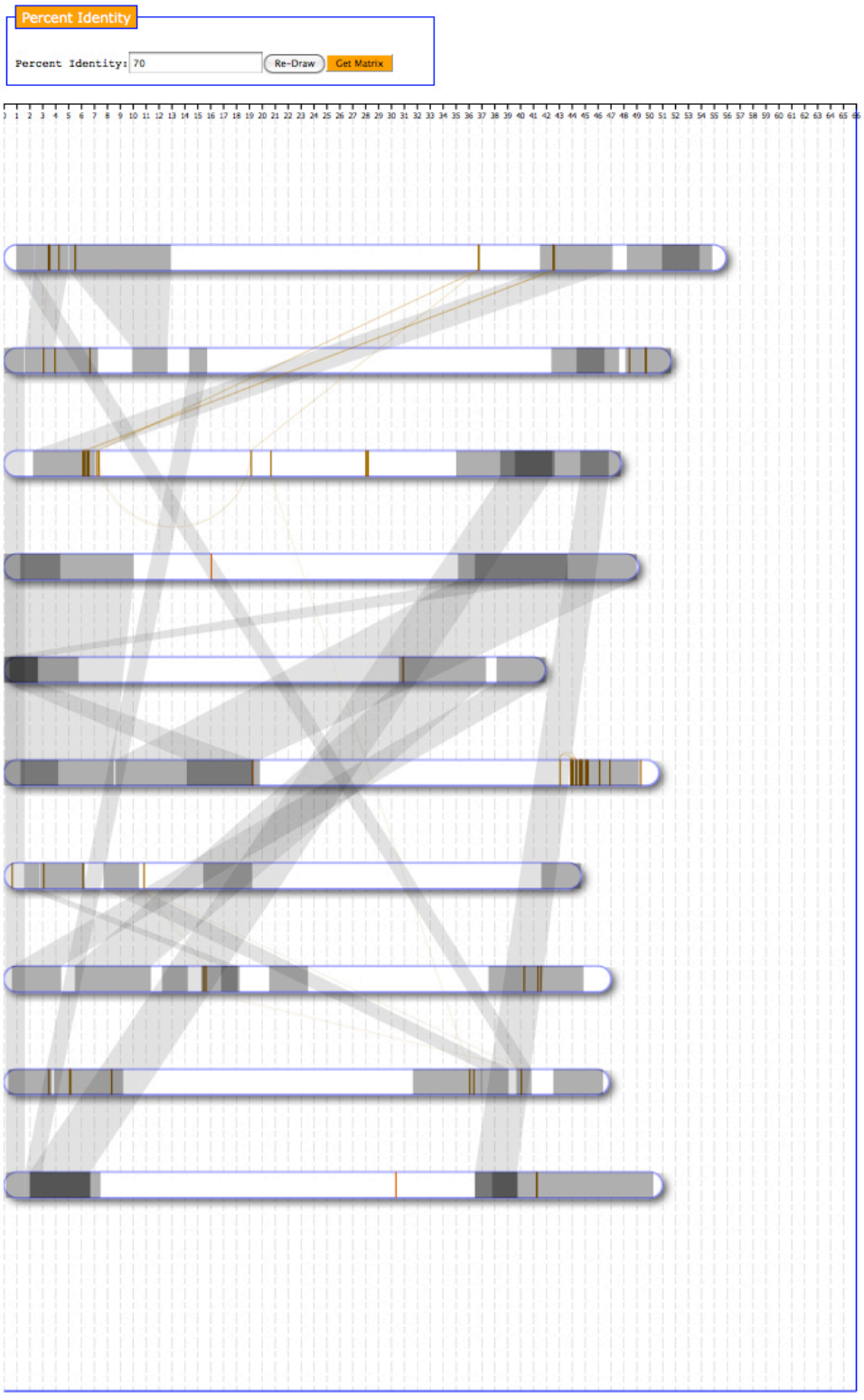


Figure 16. Soybean NBS-LRR genes visualization. 70% identity. TIR genes only.

Figures 17 and 18 display soybean NBS-LRR genes located on the first ten chromosomes in a twelve-color scheme. Since GenomeVectorizer is not yet capable of clearly displaying overlapping genes, GenomeVectorizer was run separately with a dataset containing only forward-oriented genes (W orientation – Figure 18) and separately with a dataset containing only reverse-oriented genes (C orientation – Figure 17).

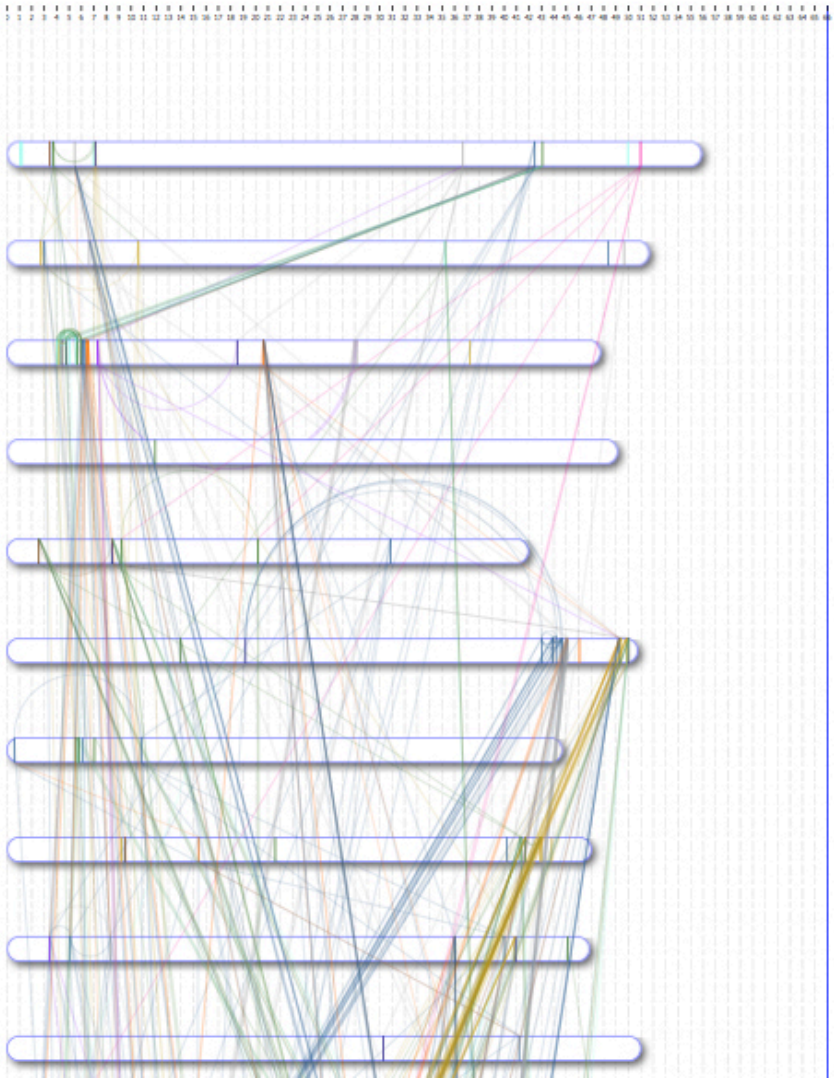

Figure 17. GenomeVectorizer. C Orientation. 50% identity.

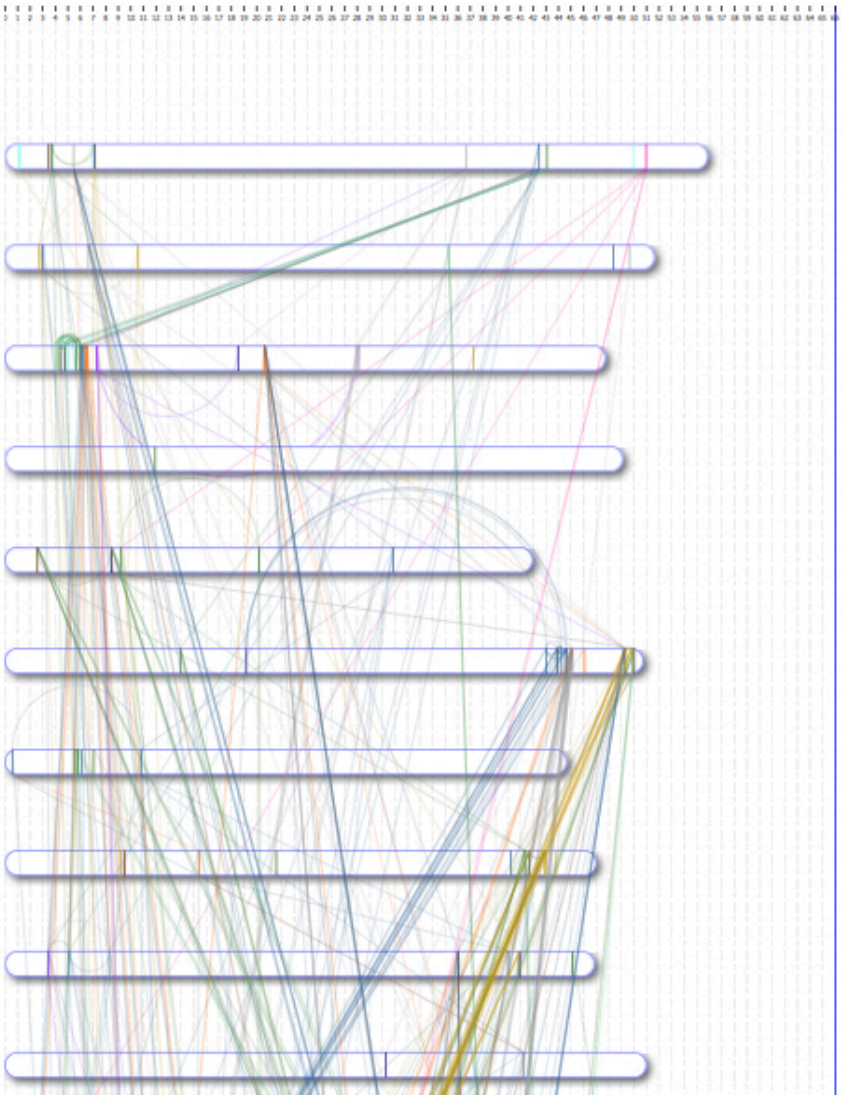

Figure 18. GenomeVectorizer. W Orientation. 50% identity.

For ease of viewing, one-to-one duplication relationships were separated from the rest of the "identity" relationships (Figure 19). It is easy to see that duplication regions include almost half of the genome.

Figure 20 contains a partial snapshot of gene lengths (heights) relative to 5% of overall chromosome size. This was another attempt to better see the overlapping genes.

Appendix A shows the code modifications that were made in order to produce the graph in Figure 20.

So far it seems that gene stacking similar to that of GenomePixelizer (Figure 13) might be the most optimal way to represent gene overlapping.

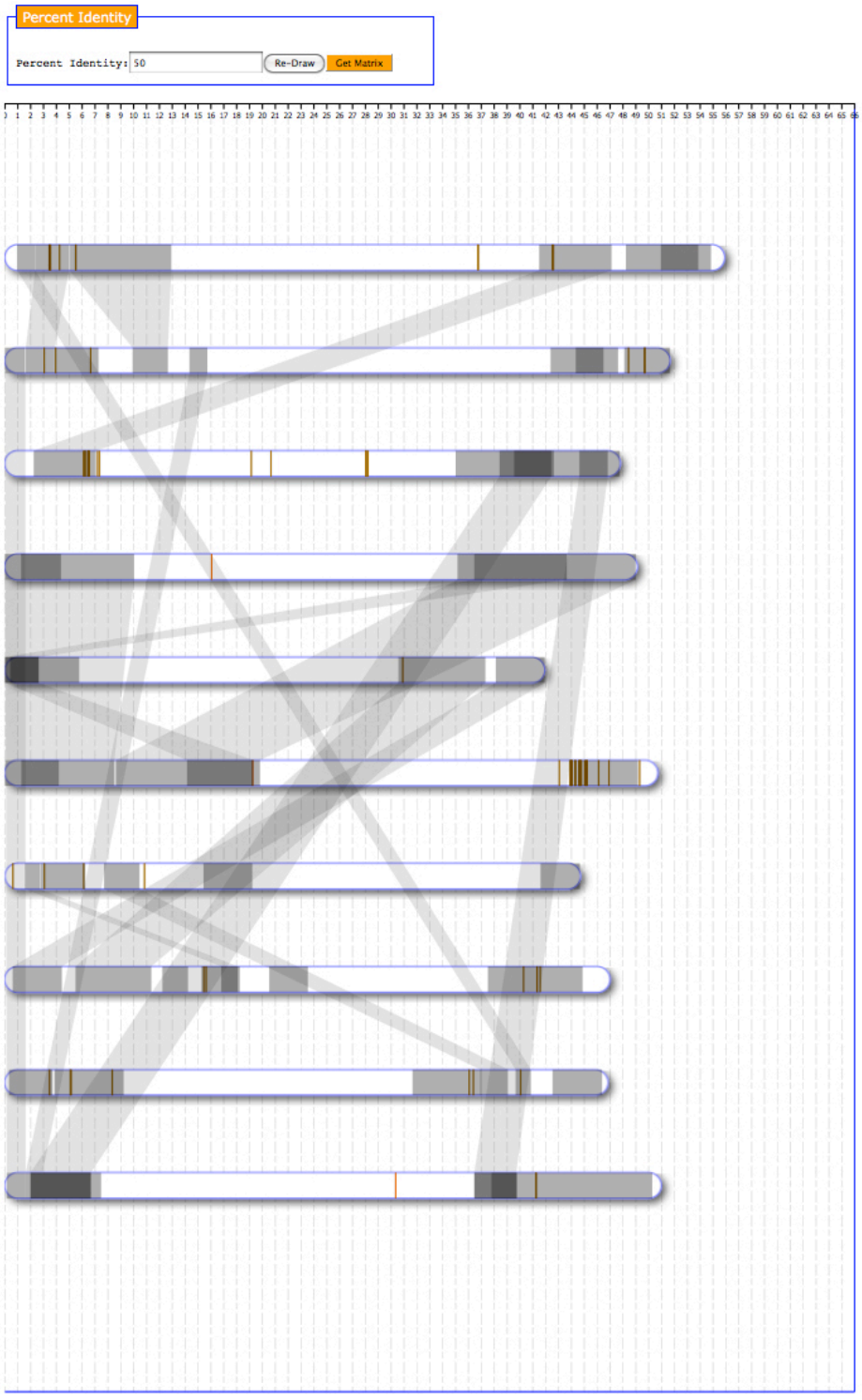


Figure 19. GenomeVectorizer, soybean NBS-LRR genome's duplication regions.

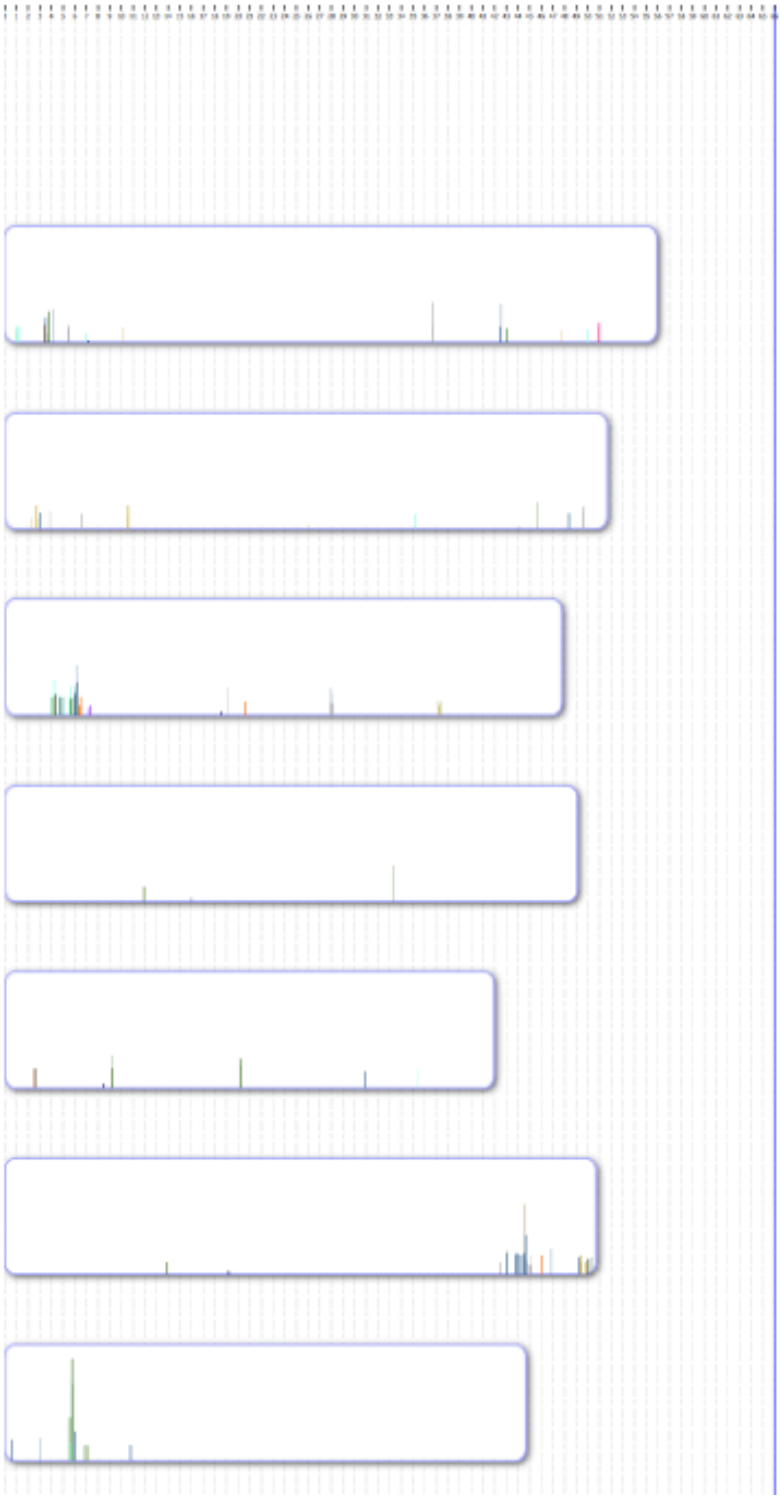

Figure 20. GenomeVectorizer, soybean NBS-LRR genome lengths (heights) relative to 5% of overall chromosome size.

# VII. FUTURE WORK

Currently, GenomeVectorizer has a significantly different way of presenting visual information than its predecessor, GenomePixelizer (compare the graphical outputs in Figure 3 and Figure 6). The author is still exploring efficient ways to represent this information. Representing reverse gene orientation by lowering the opacity of the color does not seem to be a visually optimal solution.

GenomePixelizer performs gene stacking in order to represent overlapping genes. This idea will need to be incorporated in GenomeVectorizer's future release.

The capability to rearrange objects on the canvas by dragging them sets GenomeVectorizer apart from currently available genome visualization applications.

Currently, another functionality of the tool is being designed [24]: the ability to isolate a gene cluster for analysis. The gene of interest would be dragged away from the chromosome along with the genes that connect (with identity value above the cutoff value) to the gene of interest.

An ambitious goal on the part of the author would be to make GenomeVectorizer a sort of "Genetic Editor," like Inkscape (http://www.inkscape.org/) for creating SVG graphics. This "Genetic Editor" would permit the user to scale the chromosomes to the optimal viewing size, change chromosomes' and genes' colors, stretch the relationships (connecting lines between genes) and much more.

Minor modifications could be added to improve the tool:

- The ruler that is located at the top of the canvas needs to be created as a separate object that is always located on top of the page no matter how far the

user scrolls. This ruler object should be movable so that the user can move it to the chromosome of interest and measure it.

- GeneTip Tool, the pop-up containing the name that appears once the user mouses over a chromosome, a gene, or the identity relationship (line connecting two genes), is buggy and needs to be improved.
- The following buttons could be added: " W Only" –would allow for viewing only the genes containing "W" (forward) orientation and their relationships; "C Only" – would allow for viewing only the genes containing "C" (reverse) orientation and their relationships, "Dupl Only" – would allow for viewing of the relationships between the duplicated regions only.

Major modifications to be considered for GenomeVectorizer:

- Creating the capability to expand/condense views of overlapping gene regions.
- Designing an algorithm that randomly assigns visually distinct colors to the color categories (like color1, color2, etc.) specified by users in the XML file.

# REFERENCES


[1] E. Koonin, "Orthologs, paralogs, and evolutionary genomics," Annu. Rev. Genet., 2005, 39:309-338.

[2] E. F. Vanin, "Processed pseudogenes: characteristics and evolution," Annu. Rev. Genet., 1985, 19:253–272.

[3] M. Hurles, "Gene duplication: the genomic trade in spare parts," PLoS Biol., 2004, 2 (7), e206.

[4] *Wikipedia, The Free Encyclopedia*, s.v. "Gene duplication," http://en.wikipedia.org/wiki/Gene_duplication (accessed April 30, 2010).

[5] G. Blanc, K. Hokamp, and K. H. Wolfe, "A recent polyploidy superimposed on older large-scale duplications in the *Arabidopsis* genome," Genome Res., Feb. 2003, 13(2):137-44.

[6] *Wikipedia, The Free Encyclopedia*, s.v. "R gene," http://en.wikipedia.org/wiki/R_gene (accessed April 30, 2010).

[7] R. W. Michelmore and B. C. Meyers, "Clusters of resistance genes in plants evolve by divergent selection and a birth-and-death process," Genome Res., 1998, 8(11):1113-1130.

[8] K. E. Hammond-Kosack and J. D. Jones, "Plant disease resistance genes". Annu. Rev. Plant Physiol. Plant Mol. Biol., June1997, 48:575–607.

[9] E. A. Van der Biezen and J. D. Jones, "The NB-ARC domain: A novel signaling motif shared by plant resistance gene products and regulators of cell death in animals," Curr. Biol., 1998, 8:226–227.

[10] M. Saraste, P. R. Sibbad and A. Wittinghofer, "The P-loop-a common motif in ATP- and GTP-binding proteins," Trends Biochem. Sci., 1990, 15(11):430-434.

[11] J. E. Walker, M. Saraste, M. J. Runswick, and N. J. Gay, "Distantly related genes in the alpha and beta subunits of ATP synthetase, myosin, kinases and other ATP-

requiring enzymes and a common nucleotide-binding fold," EMBO Journal, 1982, 1(8):945-951.

[12] B. Kobe and J. Deisenhofer, "Proteins with leucine-rich repeats," Curr. Opin. Struct. Biol., 1995, 5:409–416.

[13] B. Kobe and J. Deisenhofer, "The leucine-rich repeat: a versatile binding motif," Trends Biochem. Sci., 1994, 19:415–421.

[14] M. R. Swiderski, D. Birker, and J. D. Jones, "The TIR domain of TIR-NB-LRR resistance proteins is a signaling domain involved in cell death induction," Mol. Plant Microbe Interact., 2009, 22(2):157-165.

[15] *Wikipedia, The Free Encyclopedia*, s.v. "Homology (biology)," http://en.wikipedia.org/wiki/Homology_%28biology%29 (accessed August 30, 2009).

[16] EMBL-EBI, *ClustalW*, http://www.ebi.ac.uk/Tools/clustalw2/index.html (accessed August 31, 2009)

[17] BROAD Institute, *Argo Genome Browser*, http://www.broad.mit.edu/annotation/argo/ (accessed May 22, 2009)

[18] Genome Sciences Center, *Circos*, http://mkweb.bcgsc.ca/circos/ (accessed May 22, 2009)

[19] Sanger Institute, *Alfresco*, http://www.sanger.ac.uk/Software/Alfresco/ (accessed May 22, 2009)

[20] University of California, Davis, *GenomePixelizer - Genome Visualization Tool*, http://atgc.org/GenomePixelizer/ (August 31, 2009).

[21] The *Arabidopsis* Genome Initiative, "Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*," Nature, 2000, 408(6814):796-815.

[22] B. C. Meyers, A. Kozik, A. Griego, H. Kuang, and R. W. Michelmore, "Genome-wide analysis of NBS-LRR-encoding genes in *Arabidopsis*," Plant Cell, 2003, 15(4): 809-834.

[23] E. Kochetkova, "GenomePixelizer SVG-fied," SVG Open 2009, http://www.svgopen.org/2009/papers/51-GenomePixelizer_SVGfied/.

[24] M. N. Katrumane and E. Kochetkova, “Gene Cluster Analysis with GenomeVectorizer,” unpublished.

[25] J. Cheung, X. Estivill, R. Khaja, J. R. MacDonald, K. Lau, L. C. Tsui, et al., “Genome-wide detection of segmental duplications and potential assembly errors in the human genome sequence,” Genome Biol., 2003, 4:R25.

[26] A. Kozik, E. Kochetkova, and R. Michelmore, “GenomePixelizer - a visualization program for comparative genomics within and between species,” Bioinformatics, Feb. 2002, 18(2):335-336.

[27] D. Leister, “Tandem and segmental gene duplication and recombination in the evolution of plant disease resistance gene,” Trends Genet., 2004, 20(3):116-122.

[28] E. Lyons and M. Freeling, “How to usefully compare homologous plant genes and chromosomes as DNA sequences,” Plant J., 2008, 53:661–673.

[29] L. McHale, “Genome-wide identification of NBS-LRR encoding genes in *Glycine max*,” in press.

# APPENDIX A: CODE MODIFICATIONS TO CREATE OUTPUT BASED ON GENE LENGTHS (HEIGHTS)

## A. *drawingtool.xsl*

```
<xsl:template name="drawGenes">
…
<xsl:choose>
   <xsl:when test="$gene_orientation = 'W'">
          <svg:line id="{$gene_name}" x1="{$gene_position*10}" y1="{ $yy }"
                                    x2="{$gene_position*10}" y2="{ $yy - $gene_height }"
                                    style="stroke:{$gene_color};stroke-width:1;stroke-opacity:1"
                                    onmousedown='getGeneInfo("{$gene_name};")'>
                 <svg:title><xsl:value-of select="$gene_name"/></svg:title>
           </svg:line>
    </xsl:when>
    <xsl:otherwise>
          <svg:line id="{$gene_name}" x1="{$gene_position*10}" y1="{ $yy - $gene_height}"
                                                x2="{$gene_position*10}" y2="{ $yy}"
                                                style="stroke:{$gene_color};stroke-width:1;stroke-opacity:0.4"
                                                onmousedown='getGeneInfo("{$gene_name};")'>
                  <svg:title><xsl:value-of select="$gene_name"/></svg:title>
          </svg:line>
     </xsl:otherwise>
</xsl:choose>
…

</xsl:template>


<xsl:template name="drawChromosome">
    …
    <xsl:param name="bar_height" >100</xsl:param>
    …
</xsl:template>
```

## B. *parser.xsl*

```
…
if (dist &gt;= percent)
 //    draw_synteny(gene_a_name, gene_b_name);
…


<xsl:variable name="geneA_height">
    <xsl:call-template name="abs">
          <xsl:with-param name="n"
                    select="(($geneA_end_pos - $geneA_start_pos) div $chrom_mb_size) * 100 * 1000 * 2" />
    </xsl:call-template>
</xsl:variable>

…

<xsl:template name="abs">
<xsl:param name="n" />
<xsl:choose>
     <xsl:when test="$n &gt;= 0">
          <xsl:value-of select="$n" />
     </xsl:when>
     <xsl:otherwise>
          <xsl:value-of select="0 - $n" />
     </xsl:otherwise>
</xsl:choose>
</xsl:template>
```

## APPENDIX B: GENOMEVECTORIZER AND GENOMEPIXELIZER CITATIONS

### *A. GenomeVectorizer Publications*

E. Kochetkova, "GenomePixelizer SVG-fied," SVG Open 2009, http://www.svgopen.org/2009/papers/51-GenomePixelizer_SVGfied/.

E. Kochetkova, "Finding Duplication Events Using GenomeVectorizer," GrC 2010, in press.

### *B. GenomeVectorizer Citations*

N. M. Katrumane and E. Kochetkova, "Gene Cluster Analysis with GenomeVectorizer," unpublished.

### *C. GenomePixelizer Citations*

*2007*

S. Yang, K. Jiang, H. Araki, J. Ding, Y. H. Yang, and D. Tian, "A molecular isolation mechanism associated with high intra-specific diversity in rice," Gene. 2007 Jun 1; 394(1-2):87-95. Epub 2007 Feb 24.

C. Dardick, J. Chen, T. Richter, S. Ouyang, and P. Ronald, "The rice kinase database. A phylogenomic database for the rice kinome**,"** Plant Physiol. 2007 Feb; 143(2):579-586. Epub 2006 Dec 15.

*2006*

M. Romanov, M. Koriabine, M. Nefedov, P. de Jong, and O. Ryder, "Construction of a California condor BAC library and first-generation chicken–condor comparative physical map as an endangered species conservation genomics resource," Genomics 2006 Jun; 88 (2006) 711–718.

L. Timms, R. Jimenez, M. Chase, D. Lavelle, L. McHale, A. Kozik, et al., "Analyses of synteny between *Arabidopsis thaliana* and species in the *Asteraceae* reveal a complex network of small syntenic," Genetics. 2006 Aug; 173(4):2227-2235.

C. Dardick and P. Ronald, “Plant and animal pathogen recognition receptors signal through non-RD kinases,” PLoS Pathog. 2006 Jan; 2(1):e2.

*2005*

L. Feuk, J. R. MacDonald, T. Tang, A. R. Carson, M. Li, G. Rao, et al., “Discovery of human inversion polymorphisms by comparative analysis of human and chimpanzee DNA sequence assemblies,” PLoS Genet. 2005 Oct; 1(4):e56.

R. Guyot, X. Cheng, Y. Su, Z. Cheng, E. Schlagenhauf, B. Keller, et al., “Complex organization and evolution of the tomato pericentromeric region at the FER gene locus”, Plant Physiol. 2005 Jul; 138(3):1205-1215.

L. K. Fritz-Laylin, N. Krishnamurthy, M. Tor, K. V. Sjolander, and J. D. Jones, “Phylogenomic analysis of the receptor-like proteins of rice and *Arabidopsis*,” Plant Physiol. 2005 Jun; 138(2):611-623.

R. J. Wisser, Q. Sun, S. H. Hulbert, S. Kresovich, and R. J. Nelson, “Identification and characterization of regions of the rice genome associated with broad-spectrum, quantitative disease resistance,” Genetics. 2005 Apr; 169(4):2277-2293.

*2004*

R. Guyot and B. Keller, “Ancestral genome duplication in rice,” Genome. 2004 Jun; 47(3):610-614.

T. Zhou, Y. Wang, J. Q. Chen, H. Araki, Z. Jing, K. Jiang, and J. Shen, “Genome-wide identification of NBS genes in japonica rice reveals significant expansion of divergent non-TIR NBS-LRR genes,” Mol. Genet. Genomics. 2004 May; 271(4):402-415

*2003*

S. D. Marshall, J. J. Putterill, K. M. Plummer, and R. D. Newcomb, “The carboxylesterase gene family from *Arabidopsis thaliana*,” J. Mol. Evol. 2003 Nov; 57(5):487-500.

K. E. Hammond-Kosack and J. E. Parker, “Deciphering plant-pathogen communication: fresh perspectives for molecular resistance breeding,”
Curr. Opin. Biotechnol. 2003 Apr; 14(2):177-193.

A. Ureta-Vidal, L. Ettwiller, and E. Birney, “Comparative genomics: genome-wide analysis in metazoan eukaryotes,” Nat. Rev. Genet. 2003 Apr; 4(4):251-262.

S. W. Scherer, J. Cheung, J. R. MacDonald, L. R. Osborne LR, et.al., "Human chromosome 7: DNA sequence and biology," Science. 2003 May 2; 300(5620):767-772.

J. Cheung, M. D. Wilson, J. Zhang, R. Khaja, J. R. MacDonald, H. H. Heng, et al., "Recent segmental and gene duplications in the mouse genome," Genome Biol. 2003; 4(8):R47.

J. Cheung, X. Estivill, R. Khaja, J. R. MacDonald, K. Lau, L. C. Tsui, et al., "Genome-wide detection of segmental duplications and potential assembly errors in the human genome sequence," Genome Biol. 2003; 4(4):R25.

B. C. Meyers, A. Kozik, A. Griego, H. Kuang, and R. W. Michelmore, "Genome-wide analysis of NBS-LRR-encoding genes in *Arabidopsis*," Plant Cell. 2003 Apr; 15(4):809-834.

G. Gimelli, M. A. Pujana, M. G. Patricelli, S. Russo, D. Giardino, L. Larizza, et al., "Genomic inversions of human chromosome 15q11-q13 in mothers of Angelman syndrome patients with class II (BP2/3) deletions," Hum. Mol. Genet. 2003 Apr 15; 12(8):849-858.

T. K. Mitchell, M. R. Thon, J. S. Jeong, D. Brown, J. Deng, and R. A. Dean, "The rice blast pathosystem as a case study for the development of new tools and raw materials for genome analysis of fungal plant pathogens," New Phytologist. 2003 July; 159(1):53-61.

*2002*

X. Estivill, J. Cheung, M. A. Pujana, K. Nakabayashi, S. W. Scherer, and L. C. Tsui, "Chromosomal regions containing high-density and ambiguously mapped putative single nucleotide polymorphisms (SNPs) correlate with segmental duplications in the human genome," Hum. Mol. Genet. 2002 Aug 15; 11(17):1987-1995.

J. M. Gagne, B. P. Downes, S. H. Shiu, A. M. Durski, and R. D. Vierstra, "The F-box subunit of the SCF E3 complex is encoded by a diverse superfamily of genes in *Arabidopsis*," Proc. Natl. Acad. Sci. U S A. 2002 Aug 20; 99(17):11519-11524.