CartoonPlus: A New Scaling Algorithm for Genomics Data

Joanna Jakubowska[a], Ela Hunt[b], and Matthew Chalmers[a]

[a]Department of Computing Science, University of Glasgow, UK
[b]Department of Computer Science, ETH Zurich, Switzerland
`asia@dcs.gla.ac.uk,hunt@inf.ethz.ch,matthew@dcs.gla.ac.uk`

Abstract. We focus on visualisation techniques used in genome browsers and report on a new technique, CartoonPlus, which improves the visual representation of data. We describe our use of smooth zooming and panning, and a new scaling algorithm and *focus on* options. CartoonPlus allows the users to see data not in original size but scaled, depending on the data type which is interactively chosen by the users. In VisGenome we have chosen genes as the basis for scaling. All genes have the same size and all other data is scaled in relationship to genes. Additionally, objects which are smaller than genes, such as micro array probes or markers, are scaled differently to reflect their partitioning into two categories: objects in a gene region and objects positioned between genes. This results in a significant legibility improvement and should enhance the understanding of genome maps.

Key words: Genome Visualisation, Visualisation Techniques, Scaling Algorithm, Large Data Sets.

1 Introduction

Medical researchers find it difficult to locate the correct biological information in the large amount of biological data and put it in the right context. Visualisation techniques are of great help to them, as they support data understanding and analysis. We reported our findings from a survey of visualisation techniques used in genome browsers in [8]. We developed a prototype of a new genome browser, VisGenome, which uses the available techniques. VisGenome [9] was designed in cooperation with medical researchers from a hospital. We found that the majority of genome browsers show only a selection of data for one chromosome. This is obvious, because the amount of available information is so large that it is impossible to show all data in one view. Expressionview [4], for example, shows QTLs [1] and micro array probes and no other data. Some of the tools, such as Ensembl [6], show many types of data but use a number of different

---

[1] A quantitative trait locus (QTL) is a part of a chromosome which is correlated with a physical characteristic, such as height or disease. Micro array probes are used to test gene activity (expression).

data views, which make the users disoriented and lost in the tool and data space. Moreover, Ensembl shows as much information as it is possible in one view, instead of offering a view or a panel with additional information. A large number of genome browsers show only a chromosome and do not allow one to see a comparison of two chromosomes from different species. Exceptions include SyntenyVista [7] and Cinteny [15] which show a comparative view of two genomes but are limited with regard to other data, such as micro array probes. On the other hand, SynView [17] visualises multi-species comparative genome data at a higher level of abstraction.

We aim to find a solution which clearly presents all the available information, including all relevant information the biologists wish to see. We aim to find a solution for data analysis which overcomes both representational and cognitive problems.

Here, we describe single and comparative genome representations, see Figure 1 and 2. A single representation is a view which shows data for one chromosome. A comparative representation illustrates relationships between two or more chromosomes.

Our contribution is a scaling algorithm which we call CartoonPlus. CartoonPlus allows the users to see data more clearly by choosing one kind of data as basis and scaling other data types in relationship to the basis. The solution does not show data in its natural size but allows one to see relationships between different kinds of data more clearly, especially in a comparative representation.

The paper is organised as follows. Section 2 provides the background about visualisation techniques and their usefulness for medical researchers. Section 3 introduces the visualisation techniques we used in VisGenome and provides details of our new algorithm. We discuss our work in Section 4 and the last section concludes.

2   Related Work

This section examines existing visualisation techniques used in genomics data representation and clarifies why a new scaling algorithm is necessary.

A variety of scientific visualisation techniques are available and could be used for genomics. 2D techniques are very common in gene data visualisation and 3D techniques are rarely used [8]. An exception is [13] which uses a 3D model of the data. In the following we discuss the techniques used in 2D applications.

Fisheye [5] shows detail for an element and its neighborhood, but only an overview for the other elements. It is used in a number of graphical applications, for example for photo corrections, but it is hardly used in biology, with the exception of Wu [12] who used fisheye to show tables representing micro array results. Magic lenses [16] allow the user to transform the data and display extra information, see Zomit [14]. The majority of genome browsers offer scrolling and zooming [1] which are both easy to use. Zooming by buttons is well known and used by the medical researchers. Ensembl [6] uses this kind of zooming. BugView [11] also uses zooming by buttons which makes an impression
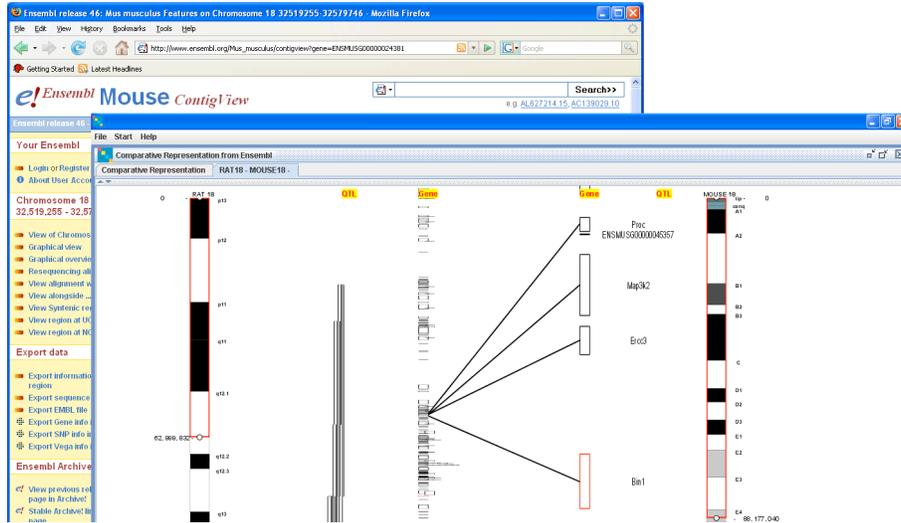
Fig. 1. The comparative representation for the rat chromosome 18 and the mouse chromosome 18. The gene Bin1 in the mouse chromosome 18 is in focus. The background shows additional information from Ensembl for Bin1, activated by clicking on the gene.

of smooth zooming. Cartoon scaling is applied to biological data in [7]. The technique deforms the original data and makes it easier to read. SyntenyVista shows all genes in the same size and this makes it clear which genes share a homology link. A true physical representation of genes causes some of them to overlap and the users often cannot precisely see the genes connected by a homology link. This work motivated us to design an improved algorithm for scaling for different kinds of data, and not only for genes. Our new algorithm, CartoonPlus, makes the display of biological data clearer in both single and comparative representations. It makes it easy to see which genes and QTLs share a homology link in a comparative representation and highlights differences and dependencies between different kinds of data in a single representation. Objects that are larger than a basis object form one category. Another category consists of objects smaller than the basis or lying in between basis objects. Those objects contained within a basis object are treated differently than the objects in between.

3   Visualisation extensions

VisGenome loads QTLs, genes, micro array probes, bands, and markers, and pairs of homologies from Ensembl. It shows single chromosomes or comparisons of two chromosomes from different species. The application uses the visualisation metaphors and algorithms offered by Piccolo [2]. Piccolo puts all zooming and panning functionality and about 140 public methods into one base object class, called PNode. Every node can have a visual characteristic, which makes the

overall number of objects smaller than in other techniques which require two objects, an object and an additional object having a visual representation, as in Jazz [2]. A Jazz node has no visual appearance on the screen, and it needs a special object (visual component), which is attached to a certain node in a scene graph and which defines geometry and color attributes. Piccolo supports the same core feature set as Jazz (except for embedded Swing widgets), but it primarily uses compile-time inheritance to extend functionality and Jazz uses run-time composition instead. Piccolo supports hierarchies, transforms, layers, zooming, internal cameras, and region management which automatically redraws the portion of the screen that corresponds to objects that have changed.

In the continuation of the section, we present a new scaling algorithm, CartoonPlus, and then we outline other known visualisation techniques which we implemented.
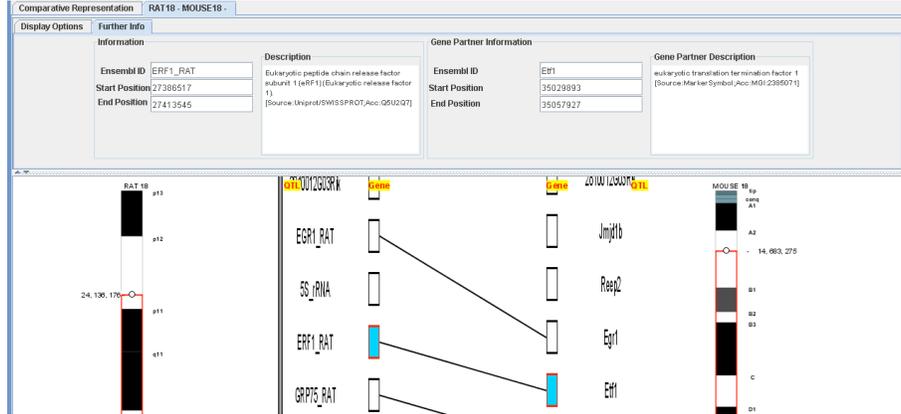

Fig. 2. The comparative representation for the rat chromosome 18 and the mouse chromosome 18. The data is scaled by the scaling algorithm which makes all genes the same size and QTL size depends on genes. Genes ERF1_RAT and Etf1 are linked by a homology line and marked in blue.

Scaling Algorithm: We developed a scaling algorithm for arbitrary genomics data which extends existing solutions. SyntenyVista [7] scaled genes only in a comparative representation. We offer scaling for all data, in both single and comparative representations, see Figure 2 and 3. Previous algorithms were constrained, while the new one scales multiple data types together, with reference to the basis. A user chooses the basis for scaling and then other elements are scaled in relationship to the chosen data type. In the current prototype we chose genes as a basis, so we scale all genes to the same size. An extension of this work is to allow the user to change the basis for scaling interactively. The algorithm looks at other types of data which are smaller or larger than genes, such as markers, micro array probes, or QTLs, and scales them accordingly. We divide

Fig. 3. 1: Single representation for the rat chromosome 18. Three genes (SMAD7_RAT, Gm672, 307259) and one micro array probe set (1384287_at, see arrow) are coloured by different colours, selected by the user interactively. 2: CartoonPlus algorithm (see Figure 4). G2 is a gene which begins at A and ends at B. m1, m2, and m3 are elements smaller than G2 and Q1 is bigger than G2.

all elements smaller than genes into two groups: elements which are in a gene region and elements which are in the region between two genes, see Figure 3.1.A. For each type of data holding items smaller than the basis, we create a column holding elements which are situated within the gene boundaries and a second column containing elements which are situated between two genes. For all elements which are in the gene region, we choose the same size for each element, and the same applies to all elements which are in the area between genes. The size of the elements depends on their number in a gene region. This means that if in an area of a gene there is only one marker, it has the same height as the gene, but if there are 10 markers, they together have the same size as the gene (each marker is set to 1/10th of gene height). When an element is on a gene boundary, it is partially in a gene region and partially between two genes, and we situated it in the gene region. We also scale elements like QTLs which are bigger than genes. We look where a QTL begins and ends and we paint it starting at the gene where it begins and ending at the gene where it finishes. The solution allows us to present clearly a homology between genes in a comparative representation and additionally to show relations between micro array probes, markers, genes, and QTLs in two species.

Figure 4 outlines the scaling algorithm. All genes, markers, micro array probes and QTLs are stored in hashtables. The algorithm iterates over all genes (line 2). First we scale markers and micro array probes which are between genes (the previous gene and the current one), see Figure 3.2 object m1 between G1

```
1 CartoonPlus() {
2    for(gene in GENEs) {
3        ResizeAndPaint(gene)
4        ScaledMarkersBetween = GET_MARKERS_BETWEEN()
5        for (each marker in ScaledMarkersBetween)
6            ResizeAndPaint(marker)
7        ScaledMicroArrayProbesBetween = GET_MICRO_ARRAY_PROBES_BETWEEN
8        for (each micro_array_probe in ScaledMicorArrayProbesBetween)
9            ResizeAndPaint(micro_array_probe)
10        ScaledMarkers = GET_MARKERS_IN()
11        for (each marker in ScaledMarkers)
12            ResizeAndPaint(marker)
13        ScaledMicroArrayProbes = GET_MICRO_ARRAY_PROBES_IN()
14        for (each micro_array_probe in ScaledMicroArrayProbes)
15            ResizeAndPaint(micro_array_probe)
16        ScaledQTLs = GET_QTLS_FOR_GENE()
17        for(each QTL from ScaledQTLs)
18            if (QTL.end>D AND QTL.end<=B)
19                ResizeAndPaint(QTL)
20                delete(QTL from ScaledQTLs)
21    }
22 }
23 GET_MARKERS_BETWEEN() {
24    for(marker in MARKERs)
25        if(marker.start>=D AND marker.end<=A)
26            markers.add(marker)
27    return(markers)
28 }
29 GET_MARKERS_IN(){
30    for(marker in MARKERs)
31        if((marker.start<=A AND marker.end>A) OR (marker.start>A))
32            markers.add(marker)
33    return(markers)
34 }
35 GET_QTLS_FOR_GENE(){
36    for(QTL in QTLs)
37        if(QTL.start>D AND QTL.start<=B)
38            QTLs.add(QTL)
39    return(QTLs)
40 }
```

Fig. 4. CartoonPlus algorithm. Hierarchy of object sizes: chromosome $\geq$ QTL$\geq$ gene $\geq$ marker and micro array probe.

and G2. Then we scale markers and micro array probes with a start coordinate before the gene and end coordinate inside the gene or start coordinate inside the gene region, see Figure 3.2 objects m2 and m3, and Figure 4 lines 4-15. Then we place QTLs which begin inside the gene region or in the region between a previous gene and the current gene, see Figure 3.2 object Q1. For each gene we check as well where the end coordinate of a QTL is, and, depending on this, we paint the element. In the pseudo-code we used function ResizeAndPaint which for basis data gives all elements the same size. For small objects, such as m1, m2, or m3, function ResizeAndPaint calculates how many elements are in the gene area or in the area between genes, and divides the area by the number of elements and then the elements are painted in the calculated size. For large elements, ResizeAndPaint calculates the hight of the elements as the beginning

of the gene where the QTL starts and end of the gene where it ends. If a QTL begins or ends between genes, the function takes the end of the previous gene or start of the next gene as its coordinates.

Navigation: We offer "overview and detail" views which are manipulated by mouse and keyboard interaction. At the beginning the users see an overview of all chromosomes and can choose the one they would like to see in detailed view. When they see all data for a selected chromosome, the tool gives them the possibility to see an overview of all data, but also details for each part of the data. The users can mark a region which is interesting for them and interact only with the selected part. To make the view clear, instead of presenting all information in one view, we use an info panel which shows additional information for the selected elements on mouse-over (Figure 2).

Marking a Region of Interest: The users can choose a chromosome region of interest (via tab 'Display Options', visible in Figure 2), and manipulate the view only inside the region. This functionality, which marks the region on the chromosome with a red box, is offered by both single and comparative representations. The red box can be moved along the chromosome and its boundaries can be adjusted. The main view shows only the data for the marked region and the users manipulate the data in the selected area. This means that when the user zooms or pans in the main view, all or some of the data from the red square is available. Data outside the coordinates marked by the square is not shown. We found the functionality useful, especially for the users who work with a particular part of a chromosome.

Zooming and Panning: We offer smooth zooming which supports the visual exploration of the chromosome space, based on Piccolo [2]. This provides efficient repainting of the screen, bounds management, event handling and dispatch, picking, animation, layout, and other features. The zooming technique allows the users to keep an area of interest in focus during interaction with the data. Zooming is manipulated by the right mouse button by moving it to the right (zoom in) or to the left (zoom out). Panning uses the left mouse button. Both interactions are easy to use and the users quickly become familiar with them, as confirmed by our study [10].

Focus On: Focus on (Figure 1) makes the focal element large enough, so that its name can be read, moves it to the center of the view, and marks its boundaries in red, which allows the user to see a small part of a viewing history until he changes the region of interest. This means that the user can see which elements he focused on during the session. In a single representation, when the user focuses on an element, all neighbouring elements in the view become proportionally larger in all columns. In a comparative representation, only elements in the chromosome containing the chosen element are changed, and all elements on the other chromosome maintain the same size. This allows the users to see an overview of elements from one chromosome and details for the selected element in the second chromosome. If the user wants all elements in the two columns to be of the same size, he chooses focus elements in both. Then we set the size of all elements to be the same.
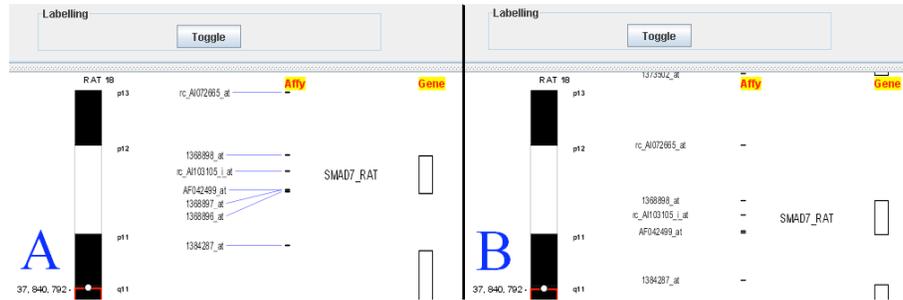
Fig. 5. The single representation for rat chromosome 18. **A** shows all labels and links which connect labels with the elements. **B** shows only a selection of labels shown next to the objects they describe.

Labelling: Because of a large amount of data, there is a problem with labels, especially for elements that have the same location. To solve the problem we allow the users to switch between viewing all labels or only a selection. When all labels are visible, they are connected by blue links to the visible elements. When the user moves the mouse close to the element, a link becomes highlighted, which allows the user to localize the element description faster, see Figure 5 A. In selected label view, Figure 5 B, we display only a small subset of labels. If there is enough room, the element name is displayed. For elements with the same coordinates, it is the first element in alphabetic order. We show as next the label for the next element which has enough room for the label.

Additional Information: Many genome browsers place all data into one view, which makes the data difficult to read. We display additional information in an info panel, see Figure 2. In a comparative representation, we show two types of information. We display Ensembl id, coordinates and a description for each element which is pointed to by a mouse. In a comparative representation when the user points to an element from one chromosome which has a homology with an element from the other chromosome, the additional information is displayed for both genes, see Figure 2. Display Options Tab allows the users some data manipulation, like choosing the range of the chromosome region displayed, changing between view with scaled data and unscaled data, or between views with all labels and selected labels. In our solution we do not have to display all information in the main view and this improves usability.

Colours: We use black and white for most data, however, after marking a region of the chromosome, the user can choose color for each of the elements by clicking on the object while pressing Alt. The default colour choice view is displayed and the user can change the colour of the marked element, see Figure 3.1. Additionally, the object boundaries are marked in red during *focus on* and all bands in the chromosomes are coloured by standard colours.

Supporting Data: Ensembl [6] offers data collected from publications and experiments. To help the user contextualise the data, we provide access to Ensembl

by clicking on a feature of interest, which invokes Ensembl in the browser (Figure 1).

Homologies: To support comparative genome analysis, we show chromosomes which have homologies with other chromosomes. Our solution allows the users to identify all homologous chromosomes quickly. When a user looks at all chromosomes in a number of species, and clicks on one, all the homologous chromosomes in other species are highlighted, and facilitate the choice of homology for visual analysis (not shown).

4 Discussion

We examined the visualisation techniques used in genome browsers, and recognised that a number of tools used in biological research implement well known visualisation techniques, but only a few experiment with new techniques. CartoonPlus adds a novel extension to the array of available solutions. It can be used in single and comparative representations. In a single representation, the users can see all data scaled, depending on a chosen basis, which allows them to see clearly which micro array probes and markers are related to a gene. In a comparative representation, the scaling makes homologies between genes clearer.

Among all genome browsers we studied, only SyntenyVista [7] uses a scaling algorithm, however it was used only in a comparative representation and only for genes. The solution we used is novel and it could be useful not only in genomic data but also in different fields of biology and medicine which use one linear scale for many types of objects. We are testing the new technique in an experiment with biological researchers who now use a combination of data from Ensembl and their own lab experiments. We conducted a *user study*, to identify future improvements and assess the usability of our solution and saw that biologists found it useful, especially for scaling small objects (SNPs) [10]. We will next offer the users interactive choice of the basis for the scaling. We want to improve colouring and give the users the option to add colour to a region and not only to a single element.

Conclusions

We designed and implemented a new scaling algorithm and combined it with some known visualisation techniques. Our new technique presents the data more clearly, especially in a comparative representation where the users want to see homologies. We believe our visualisation extension improves on the existing tools which try to present as much data as possible or only a predefined subset of data. The combination of scaling, labelling and focus techniques we offer is likely to support an improved understanding of data relationships, as required in biomedical research. In the long term we see significant potential for user control over exactly how and where scaling is done, as in magic lenses [3], although we emphasise the need for a user study to validate this.

References

1. Bederson, B. B. et al.: Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. UIST'94, ACM, 17–26 (1994).
2. Bederson, B. B. et al.: Toolkit Design for Interactive Structured Graphics. *IEEE Trans. Soft. Eng.*, **30**(8):535–546 (2004).
3. Bier, E. A. et al.: Toolglass and magic lenses: the see-through interface. SIGGRAPH, 73–80 (1993).
4. Fischer, G. et al.: Expressionview: visualization of quantitative trait loci and gene-expression data in Ensembl. *Genome Biol* **4**:R477 (2003).
5. Furnas, G. W.: Generalized Fisheye Views. *CHI*, 16–23 (1986).
6. Hubbard, T. J. P. et al.: Ensembl 2007. *Nucleic Acids Res.* 35, Database issue: D610-D617 (2007).
7. Hunt, E. et al.: The Visual Language of Synteny. *OMICS*, **8**(4):289–305 (2004).
8. Jakubowska, J. et al.: Granularity of genomics data in genome visualisation. TR-2006-221 (2006).
   http://www.dcs.gla.ac.uk/publications/PAPERS/8212/AsiaElaMatthewCHI06.pdf.
9. Jakubowska, J. et al.: VisGenome: visualisation of single and comparative genome representations. *Bioinformatics* **23**(19):2641–2642 (2007).
10. Jakubowska, J. et al.: Mixed Paradigm User Study. *In preparation.*
11. Leader, D. P.: BugView: a browser for comparing genomes. *Bioinformatics* **20**:129–130 (2004).
12. Min, W. et al.: A fisheye viewer for microarray-based gene expression data. *BMC Bioinformatics* **7**:452 (2006).
13. Montgomery, S. B. et al.: Sockeye: A 3D Environment for Comparative Genomics. *Genome Research* **14**(5):956–62 (2004).
14. Pook, S., Vaysseix, G., and Barillot, E.: Zomit: biological data visualization and browsing. *Bioinformatics*, **14**(9):807–814 (1998).
15. Sinha, A. U. and Meller, J.: Cinteny: flexible analysis and visualization of synteny and genome rearrangement in multiple organisms. *BMC Bioinformatics* **8**:82 (2007).
16. Stone, M. C. et al.: The movable filter as a user interface tool. HCI'94 Human Factors in Computing Systems, ACM Press 306–312 (1994).
17. Wang, H. et al.: SynView: a GBrowse-compatible approach to visualizing comparative genome data. *Bioinformatics* **22**(18):2308–2309 (2006).