Fourth Eurographics Workshop on Parallel Graphics and Visualization (2002)
D. Bartz, X. Pueyo, E. Reinhard (Editors)

Mining the Human Genome using Virtual Reality

Bram Stolk, Faizal Abdoelrahman[†], Anton Koning and Paul Wielinga

SARA Computing and Networking Services, Amsterdam, The Netherlands

Jean-Marc Neefs, Andrew Stubbs, An de Bondt, Peter Leemans[‡] and Peter van der Spek

Johnson & Johnson Pharmaceutical Research & Development, Beerse, Belgium

Abstract

The analysis of genomic data and integration of diverse biological data sources has become increasingly difficult for researches in the life sciences. This problem is exacerbated by the speed with which new data is gathered through automated technology like DNA microarrays. We developed a virtual reality application for visualizing hierarchical relationships within a gene family and for visualizing networks of gene expression data. Integration of other information from multiple databases with these visualizations can aid pharmaceutical researchers in selecting target genes or proteins for new drugs. We found the application of virtual reality to the field of genomics to be successfull.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Multimedia Information Systems]: Artificial, Augmented and Virtual Realitues J.3 [Life and Medical Sciences]: Biology and Genetics

1. Introduction

The human genomic code – the genetic blueprint which is contained in every cell – consists at a low conceptual level of over 3 billion elements (nucleotides). These nucleotides are labelled either G A T or C. Genes are sequences of nucleotides that typically span from 100 to 10000 nucleotides. Currently approximately 45000 genes have been identified (either predicted or lab verified) for the human genome. A large number of these genes serve as the templates for the basic building blocks of life known as proteins. Proteins are translated from active subsequences of a gene preceded by a step called transcription. During transcription the genetic code (DNA) in the cell-nucleus is transcribed into messenger RNA (mRNA) outside of the cell-nucleus.

The proteins themselves consist of strings of amino-acids. This flat sequence is also named the primary structure of a protein. There are 20 different aminoacids which are encoded in the genes by a sequence of 3 nucleotides (a codon). Therefore an active gene sequence – from start codon to stop codon – with a length of 3000 nucleotides encodes a protein sequence of length 1000. As there are 64 possible combinations of the 4 nucleotides, some of combinations are redundant and code for the same aminoacid. Proteins form the structural elements of cells and give rise to the concept of phenotypes (eg. the color of one's eyes).

These proteins are the targets for pharmaceutic intervention. More importantly protein-protein interactions form a complex network that make up the signaling and biochemical pathways. These biochemical pathways are the low-level chemical processes that make an organism function. From these pathways biologists can infer which processes are involved when certain abnormal states of the human body (i.e. diseases) are encountered.

Taxonomists categorize the relationship between different species, likewise bioinformaticians organize proteins into gene families based on their sequence and motif (a common pattern within a sequence) similarity. One of the publicly

[†] currently at Sentient Machine Research, Amsterdam, The Netherlands
[‡] CMG-contractor

available genomic databases, Ensembl[5], contains the the sequence, the genomic location, the function and the gene family association of several thousand lab verified and in silico predicted proteins. Ensembl and other databases, including proprietary databases contains the expression information (profiles) of these proteins in multiple tissues which has both normal and abnormal pathology. Integrating, visualising and mining the information in these databases represents a significant challenge.

In this paper we describe two approaches to mine genomic data, one based of the hierarchical relations of proteins in a gene family and the other based on the many to many relations of gene expression profiles.

2. Why using Virtual Reality for mining?

The human visual system is able to process enormous amounts of information in real time, which is why since the early nineties research has been undertaken to visualize abstract data (a.k.a. information) in order to provide insights into the data that would otherwise be impossible to gain.

The main challenge for visualising genomics data was in our case to visualize relations between entities (see next sections). A natural way to visualize relations is by drawing graphs. Drawing graphs in an estethically pleasing manner in 2D is a difficult problem, on which extensive research has already been done. It is frequently used with the purpose of information visualization. Graph drawing in three dimensions has not been subject to similarly extensive research. Note that graph drawing in 2D is not a special case of 3D graph drawing, as notions of 'edge-crossings' (which are minimalized in 2D drawing) have little meaning in the 3D case.

However, drawing graphs in three dimensions have the advantages that are stated by Herman[4] et al:

- The extra dimension would give, literally, more "space", and this would easy the problem of displaying large structures.
- The user can navigate to find a view without occlusions.

For an effective visualization of these 3D graphs, we require the use of virtual reality (VR) technology. This VR technology includes stereo vision (different images for left and right eye to enable depth cues). It also includes motion tracking where hand and head movements are measured. Colin Ware and Glenn Franck have made a quantative analysis of the performance of stereo and motion cues[9] with remarkable results. Test subjects were given the task to interpret 3D nets. The tests showed that the use of stereo vision improved performance by 60% and the use of head tracking improved the performance by 120%. Using both stereo vision and motion cues resulted in a 200% improvement.

The no-occlusion view is especially easy to obtain in the main virtual reality facility in use at SARA: the CAVE^{TM}[3]. Navigation in a CAVE^{TM}environment can be achieved by simple physical movements of the user. For instance, the mere act of crouching in the CAVE^{TM}can provides a view from below on a virtual object.

3. Hierarchical relationships in gene families

The visualisation challenge is to display a large number of hierarchical relations between proteins. The relations are defined by a so called gene family tree as they are based on sequence similarity. The gene family tree is computed with a neighbor joining algorithm using the software package Clustal W[8]. This is a bottom up procedure which groups together similar proteins in subbranches of the constructed binary tree. At the lowest level the algorithm needs a measurement of similarity between two proteins, which is defined on the basis of the amount of difference between two amino-acid sequences. It then groups together sequences which are most similar or groups together formed clusters on the basis of similarity with the average cluster member (some artificial average sequence). The result is a tree in which the leaf nodes represent proteins and the intermediate nodes higher in the tree represent protein clusters. The higher an intermediate node the bigger the cluster it represents, with the root node representing the total group of proteins.

Our work consists for one part of the visualisation of two important groups of proteins, namely the gene protein coupled receptors (GPCRs) and the nuclear receptors. The GPCRs are an important group in that they allow signals from outside the cell to enter the cell. This can be done because a G Protein is coupled to a receptor that is lengthy enough to pass several times through the cell membrane. When a ligand such as a hormone binds to the receptor's ligand binding domain (which is located outside of the cell), the coupled G protein is activated by the receptor's protein activation domain (inside the cell). This in turn initiates a sequence of steps within the cell that ultimately causes the transcription of the target genes. The nuclear hormone receptors are also signal enablers, active in the nucleus of the cel. Nuclear repeceptors bind to the promotor regions of the genes and switch on cascades of downstream genes. Analysis of such groups of genes on DNA microarrays[11] is of biopharmaceutical interest to understand the effects of certain drugs.

For the construction of the spatial layout of the tree, we chose for a simple algorithm that recursively subdivides the 3D space using spherical coordinates. Sphere partitions are assigned to branches of the tree based on the sizes of these branches relative to their sybling branches.

By considering an intermediary or cluster node of a tree, one can display all the sequences which can be reached from this node properly aligned below each other. In this display gaps are introduced in the sequences to have the columns match as closely as possible and to maximize the alignment score. The alignment is computed again using Clustal

W which takes the computed phylogenetic tree as the basis for the multiple sequence alignment. By color coding the amino-acids one can start to characterize clusters by common amino-acid patterns. Alternatively one can first come up with an amino-acid pattern obtained from a functional domain and highlight these patterns on all sequences.

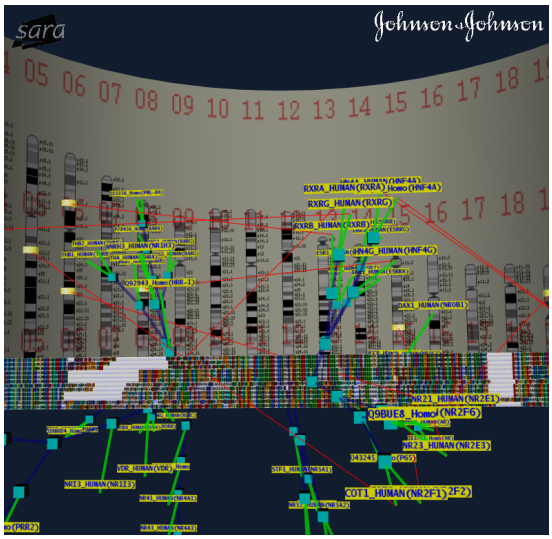

Figure 1: *Tree visualization of nuclear receptors.*

Because proteins are the basic active units and genes their corresponding blueprints we can relate more available information to the proteins in the phylogenetic tree. Important information is gene position on the chromosome. There are 24 different human chromosomes and each chromosome is divided into banding regions. The chromosome number and banding pattern that is the origin of a protein can help a geneticist to derive the function of this protein by knowing the function of closely located proteins. We therefore implemented a chromosome map and a two-way selection mechanism that allows the user to either select proteins in the tree, the position of which is then shown on the map or regions on the chromosome map, after which the proteins corresponding to genes in that region are shown.

As mentioned earlier even more protein interaction data can be brought into view if one takes into account the protein's function, the pathways in which the proteins are involved in or the tissues in which the transcribed gene material is most expressed. Some of this functionality has already been implemented, while other possibilities will be considered later in the project.

4. Many to many relations derived from expression profiles

Our challenge in this case is visualizing a set of genes/proteins with their interrelations as a cyclic graph. A possible measure for a relation between two entities is the so called 'Fischer Exact sampling'[1] which calculates the expression as a so called p-value. When considering a set of N genes/proteins, we can define a maximum of N*(N-1)/2 relations.

The data that we have been working with is not complete, and only a subset of the relations is known. But even this set is so large that we choose to reduce it by keeping only the highest similarity values of our set of known similarities. Also, as the p-values range from 10^{-40} to 10^{-130} we had to apply logarithmic scaling, and map them to spatial distances ranging from 0.0 to 90.0

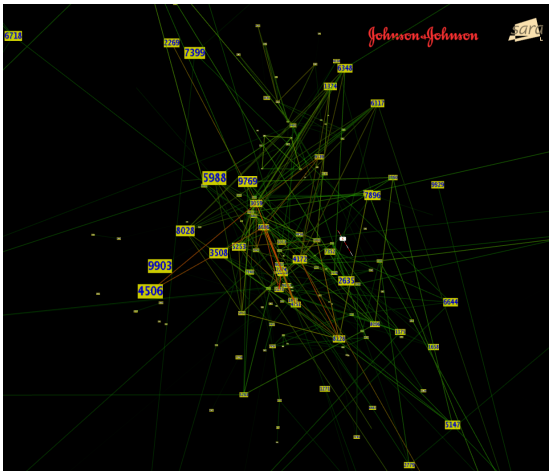

Figure 2: *Many to many relations visualized.*

So we construct graph-layouts in three dimensions, based on the set of similarity values. When constructing the graph, we have to determine a 3D position for each gene in our dataset. We want to choose these positions in such a way, that the distances between connected genes are proportional to the similarity value defined between the genes. Ofcourse, some datasets will have more than one solution, but most datasets will not have a solution at all, so we have to find an optimum. Even a very simply case of a cyclic graph with nodes A,B,C with dist(A,B) = dist(B,C) = 1.0, dist(A,C) = 10.0 has no solution. We found that limiting the number of relations per node will result in beter convergence to a solution with a smaller error. We can either discard the lowest similarity values (highest distances) of nodes with more than a specific number of relations or alternatively, we can have the weights rapidly approach zero as the relation ordinal increases.

One way to obtain the configuration of positions of genes in 3D is by using a technique called Multi Dimensional Scaling (MDS). The computation of the configuration is done by optimizing the weighted STRESS[7] function: $\Sigma_{i=1}^{N}\Sigma_{j=1}^{N} w_{ij}(d_{ij} - D_{ij})^2$, where d_{ij} represents the computed

euclidean distance between points i and j in the configuration and D_{ij} reflects the original similarity value between i and j.

In this function the difference in similarities between points in the computed configuration and in the original dataset is minimized. Finding the global optimal solution is a combinatorial optimization problem and is considered NP-Complete. Therefore a number of methods have been proposed to compute near optimal solutions. Simulated Annealing (SA) has been identified as a very good approximation algorithm in this respect[6]. As opposed to gradient descent algorithms which frequently get stuck in local minima, SA can escape local minima in search for better solutions. This means that as opposed to downhill-moves only, the SA algorithm can occasionally allow uphill moves in its search for a better solution.

If the cost in state n is lower than that of m the move is always accepted, otherwise the move is only accepted with the given probability.

In the context of SA the STRESS function becomes the cost function that is used to check whether moves are accepted. The weighting scheme adopted can have a great impact on the solutions SA generates. For example with a large set of constraints (relations) we could not achieve low energy states. We therefore came up with the following weighting scheme. For this per gene all relations in which a gene participated were sorted by similarity value. Then we applied the following additional weighting for each relation: $w_{ij} = \frac{1}{e^{\alpha(ord(i)+ord(j))/2}}$

Where $ord(i)$ represents the position on the sorted list for gene i where the relation from i to j can be found. This entails an exponential decrease of the effect that relations with low similarity values have on the cost function, making it possible to achieve low enough energy states to be meaningfully discerned in 3D visualizations of the found configurations. See figure 3 for the effect of different alphas on weight.

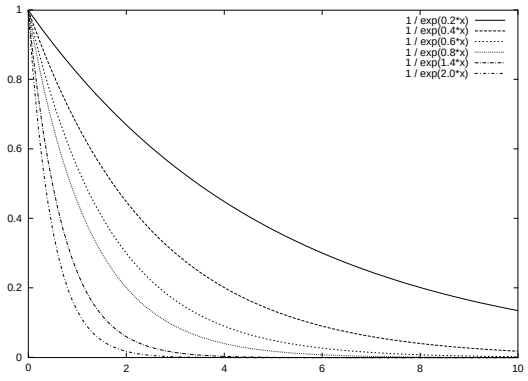

Figure 3: *The effect of alpha on the average ordinal position and weight.*

The parameter α is used to control the influence of less important relations. The following plot illustrates the results obtained when we vary alpha between 0 and 4. (plot shows average over 10 runs per alpha, on a dataset with 548 points and 12000 relations), see figure 4.

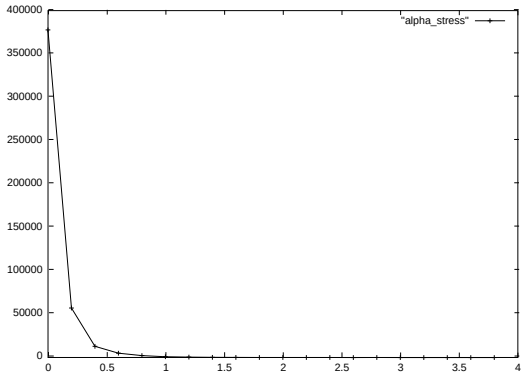

Figure 4: *The effect of alpha on the stress or energy.*

Choosing α too high causes few relations to not be of any effect, but with little stress. Choosing α too low causes many relations to be of influence causing high stress values. The answer lies somewhere in the middle.

5. Implementation

When developing VR applications, SARA uses a modular approach named 'sarasim'. With sarasim, the Python[12] interpreter is used and all application components are in the form of Python modules. Application specific components are typically implemented in C++, and automatically converted for use in Python by the SWIG[13] tool. The use of Python gives us the following advantages:

- Rapid prototyping without re-compilations.
- Convenient coupling of different technologies, as the Python scripting language can be used as the syntactic glue for integrating a heterogenous set of objects.
- Access to a wealth of domain specific functionality. For instance, BioPython[10] gave us instant access to biological databases, simply by importing just another Python module.
- Convenient configuration. Placing the run-time configuration in a Python script is preferable over a plethora of command line options. It is easy to maintain, readable, and allows for more complex expressions in your configuration.

The 3D graphics functionality is provided by a Python module created from SGI's OpenGL|Performer[2] library, which offers scene graph and real-time rendering functionality on top of OpenGL. The VR functionality is provided by a Python module based on VRCO's CAVELib^{TM}[14] library. The CAVELib allows applications to be run on a variety of

VR-systems, ranging from a simple workstation to multiple screen solutions such as RealityCenters™and CAVEs™. The sarasim system is available on both MIPS-Irix and Intel-GNU/Linux operating systems.

6. Conclusions

We have successfully applied virtual reality to problems in the genomic research field. Using saragene –in the early stages of the project– bioinformaticians have already identified new relations between genes which may have eluded them when using conventional approaches only. This raises high expectations for future efforts.

The sarasim programming environment proved to be flexible and allowed us to quickly develop bio-informatics datamining applications, while originally being aimed at CAD-review and simulation.

Acknowledgements

This reseach was sponsored by Janssen Pharmaceutica N.V. and we would like to thank the bio-informatics team of Johnson & Johnson Pharmaceutical Research & Development.

We would also like to thank Michel Rosenberg at SGI Belgium for organizing the promotional event that resulted in this project.

References

1. A. Agresti. A Survey of Exact Inference for Contegency Tables. *Statitical Science*, **7**:131-153, 1992.
2. S. Clay, J. Zhao and C. Insinger. IRIS Performer 2.2: Rendering for High-Performance and Interactive Graphics Applications. *SGI whitepaper* docnr 007-3534-001, electronically available at `http://www.sgi.com/software/performer`
3. C. Cruz-Neira, D.J. Sandin and T.A. DeFanti. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE™. *Proceedings SIGGRAPH '93 Computer Graphics Conference*, ACM SIGGRAPH, pp. 135-142, 1993.
4. I. Herman, G. Melançon and M. Scott Marshall. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, **6**(1):24-43, 2000.
5. T. Hubbard, et al. The Ensembl genome database project. *Nucleic Acids Research*, **30**:38-41, 2002.
6. H. Klock and J.M. Buhmann. Data visualization by multidimensional scaling: A deterministic annealing approach. *Pattern Recognition*, **33**(4):651-669, 1999.
7. J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, **29**:1-27, 1964.
8. J. Thompson, D.G. Higgins and T.J. Gibson. Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, **22**:4673-4680, 1994.
9. C. Ware and G. Franck. Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions *ACM Transactions on Graphics*, **15**(2):121-140, 1996.
10. `http://www.biopython.org`
11. `http://cmgm.stanford.edu/pbrown`
12. `http://www.python.org`
13. `http://www.swig.org`
14. `http://www.vrco.com`

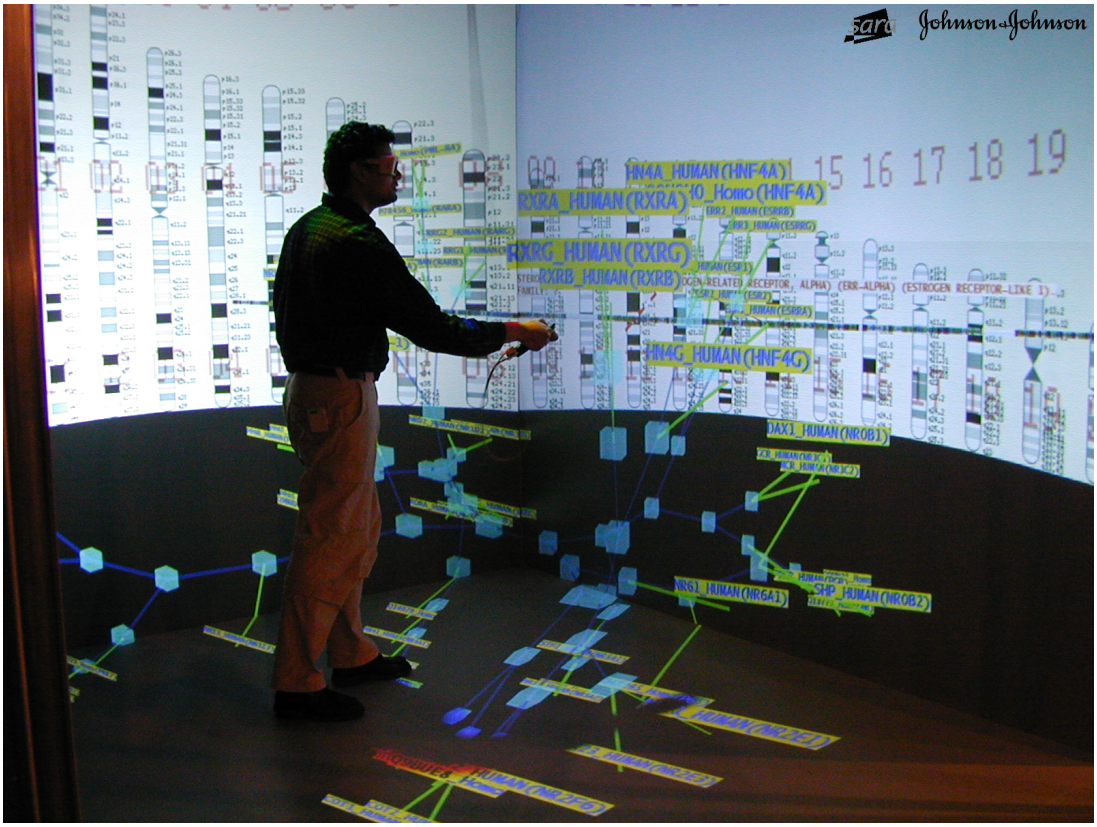

Figure 5: *Bio-Informatics scientist mining the human genome.*

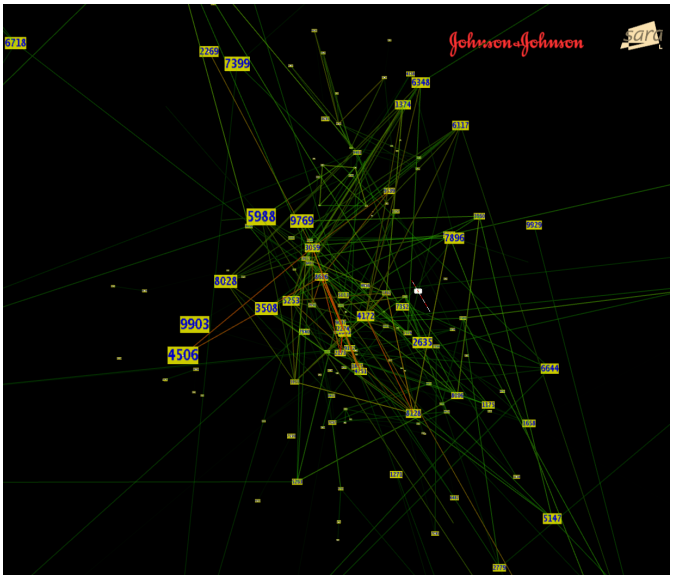

Figure 6: *Many to many relations visualized.*