Universitat Politècnica de Catalunya
Departament de Llenguatges i Sistemes Informàtics
Màster en Computació

Tesi de Màster

Genome Map: a web application for annotated genome visualization

Estudiant: Bernat Gel
Director: Xavier Messeguer

Data: 25 de Juny de 2007

Contents

List of Figures

Chapter 1

Introduction

As usually happens with new Google applications, their geographical information tool Google Maps (`http://maps.google.es/maps`) has revolutionized the way users expect to interact with maps and location info. Their visual and hihgly responsive interface brings ease of use to geospatial data.

Just like all those geolocalized data is referened to the underlying map genomic features like gens are referenced to the genome sequence. It is our goal to build up a genome browsing tool as easy and intuitive to use as Google Maps and at the same time as powerful and customizable as user may need.

In this chapter we will see a small introduction to what gens and genomes are. We will also see what annotate a genome means, why is this important and wich amount of data it is. We will take a fast glance at how this data is stored and accessed and how there are some programs one can use in orther to access it.

1.1 Genomes, a huge amount of information

In the nucleus of every cell of every living organism there is what is called the genetic code[1]. This code contains all information necessary for the organism to be developed. The genetic code in each cell of an organism is the same and it is called the genome$^{\nabla}$[1].

The genome is a sequence of only four different nitrogenated bases$^{\nabla}$, identified by A, T, C and G, with a length ranging from few thousands of bases for simple viruses to thousands of millions of bases for complex organisms as humans. In larger

[1]This symbol $^{\nabla}$ means that this term is defined in the glossary at the end ot the thesis. It will appear only next to the first appearance of a new term.

genomes the sequence is organized in different chromosomes$^{\nabla}$.

Genes$^{\nabla}$ are the smallest meaningful parts of the genome[2] and they are translated into proteins$^{\nabla}$ wich in turn are the functional base of an organism. They are spread all along the genome and not tightly packet. In fact, genes only represent a small part of all the sequence and the remaining part is basically unimportant in the sense it doesn't codify any protein. Genes themselves are not completely codifying neither but have large chunks of non codifying DNA$^{\nabla}$, the introns$^{\nabla}$, which will be removed before the actual protein translation takes place. Till recent time those big pieces of non codifying DNA where considred junk DNA, but now they are known to codify other special features, like microRNAs$^{\nabla}$, very important for gene expression regulation.

Anyway, genes are not the only interesting features of genomes. Repeated regions, single nucleotide polymorphisms (SNPs$^{\nabla}$), gene promoters$^{\nabla}$, microRNA codifying regions, etc. are all important and good information about their existence, exact location and properties is fundamental for successuful research in genetics. The process of generating this information and addig it to the genome is called annotation. Genome annotation is a joint effort of the research comunity and the natural next step after the basic genomic sequence has been discovered. Many research groups all arround the world are working on genomic feature finding and annotation.

Therefore, it is not possible to assign only one feature type or function to a part of a genome but its meaning depends on what we are studying. For example, a given region can be part of an oncogene$^{\nabla}$ and at the same codify a microRNA, have an SNP and be part of an STS$^{\nabla}$. It is clear, then, that tools used for genome browsing should be genome based, that is, having the genome sequence as the main and always visible part and being able to show or hide different subsets of annotation data uppon it.

1.2 Storing and browsing that information

NCBI (National Center for Biotechnology Information) (`ncbi.nlm.nih.gov/`) is a United States Center with very useful web site which works as a central repository for sequenced genomes. This web provides some interesting tools to manage genome information and allow the user to download, search and explore all the genomes con-

[2]Actually this is not completely true. Genes are the smallest (and only) protein coding sequences, so the only producing "functional parts", but there are smaller parts containing non protein coding sequences with gene expression regulation functions. Good example are microRNA, as tiny as only twenty base pairs but one of the most complex and powerful regulation mechanisms in cells.

tained on the repository. Further more, NCBI contains some databases specialized in information of any type of annotation. The web gives the possibility of doing specialized search over all those databases, and even offers tools for performing integrated analysis of most of the information they have. But NCBI have only a small fraction of all feature information available in their servers and despite some linkings with other public databases it is still not possible to access to all the information in a centralized manner.

EBI-EMBL (`http://www.ebi.ac.uk/`, `http://www.embl.org/`) is its european counterpart. They have also many databases available and have developed many powerful tool to acces/analyze all their data and actually they offer one of the, in my opinion, best genome browsers available today.

As said before, many research groups are working on feature discovery and annotation and many of them use their own databases to store their findings and offer them to the public. This situation makes rapid comunication of findings possible and encourages the development of new and original ways of searching and browsing that data. Unfortunately, it also means a lot of information fragmentation and makes the use of data from different databases in a research project challenging.

Thus, access to those huge amounts of information is not easy for many of its final users, namely biologists, genetists and doctors, and this can be a steep problem to overcome when they are designing new research lines and experiments.

As an example, there are some interesting questions which are almost impossible to answer without using ad-hoc software or browsing big listings by hand in a boring and error prone process:

- Are there any SNPs in the promoter of a given gene?
- Are there any microRNAs codified in a given gene?
- How many SNPs are in the exonic part of a gene?
- In which way two alternative splicings of the same gene are different?
- Is there anything interesting in a given region? And near it?
- Are the genes involved in a given pathway neraby?
- Where are located the genes with a given function?

All those questions are important and deserve being answered, but despite we have all the information we could need, it is too difficult for the researchers to answer them beacuse most small research groups cannot afford developing specialized software needed here.

In this Master Thesis we try to solve this problem developing an application that integrates information from diverse sources and shows it through an easy to use interface allowing non technically skilled researchers to answer questions as the ones above with few mouse clicks.

Chapter 2

State of the Art

This chapter will tell us about some general information about genome browsers today and then, it will show us some of the most used genome browsers, both online and offline. We will also see their *pros* and *cons* as well as screenshots.

There is a huge amount of genomic information stored in public databases accessible from the internet. Many of these databases are involved in big projects concerning data acquisition, analysis and access. There are specialized databases as miRBase [2] or tarBase [3] which have information about a single kind of information (microRNAs and their targets) for a broad range of organisms or as Flybase (http://www.flybase.org/) or Wormbase (http://www.wormbase.org) devoted to the genetic information about a family of organisms.

Some institutions offer tools and services to navigate their own databases and offers links to other related databases. NCBI, for example, is ofering a very powerful and exhautive text base access. It allows users to get integrated information from some of their databases. As an example, in a gene extended information page, there are references to papers about that gene from PubMed, information about the proteins codified by that gene from their protein database and information about functional annotation with links to the Gene Ontology along to the gene's own data. Those access methods are designed to easily access to extensive information when the user knows what he is looking for but are not suitable for any kind of bird-eye browsing nor exposing relations between elements in the database.

There is no standard way of accessing those databases and the usual interface is a web form and report generation. Some of the biggest projects (such as EBI-EMBL and NCBI) are offering means of programatically accessing their information retrieval systems and most projects offer dumps of the databases to download via ftp. Anyway, those dumps are not in a standard format (they usually are plain text tab separated files) and extracting information from them usually requires some

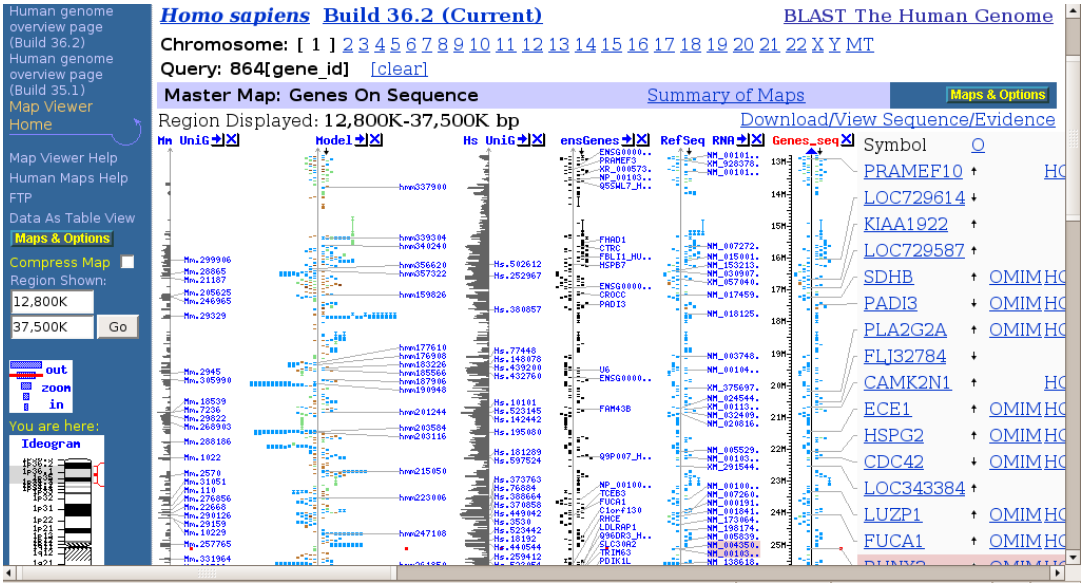

Figure 2.1: *Screenshot of NCBI Map Viewer taken in firefox in ubuntu linux.*

additional knowledge. As mentioned before, different base numbering schemes are in use so crossing information from different databases is complicated.

To overcome these problems, some tools have been developed to help the user to browse the genomic information in a visual way. These applications, ususally known as *genome browsers* or *genome visualizers* have existed for quite a long time but they have some drawbacks in some areas as interactivity or user-friendliness.

Let's see a brief description of the most important ones:

- **NCBI Map Viewer** (`www.ncbi.nlm.nih.gov/mapview`) is one of the most used genome visualizers since this web based application is tightly integrated into the suite of tools provided by the NCBI. Figure 2.1 shows a screenshot of NCBI Map Viewer.

  Positive Aspects

  - It is powerful and is able to show a lot of information at once.
  - It has five predefined zoom levels and allows the user to select the exact region to show.
  - It includes a general view of the current chromosome .

  Negative Aspects

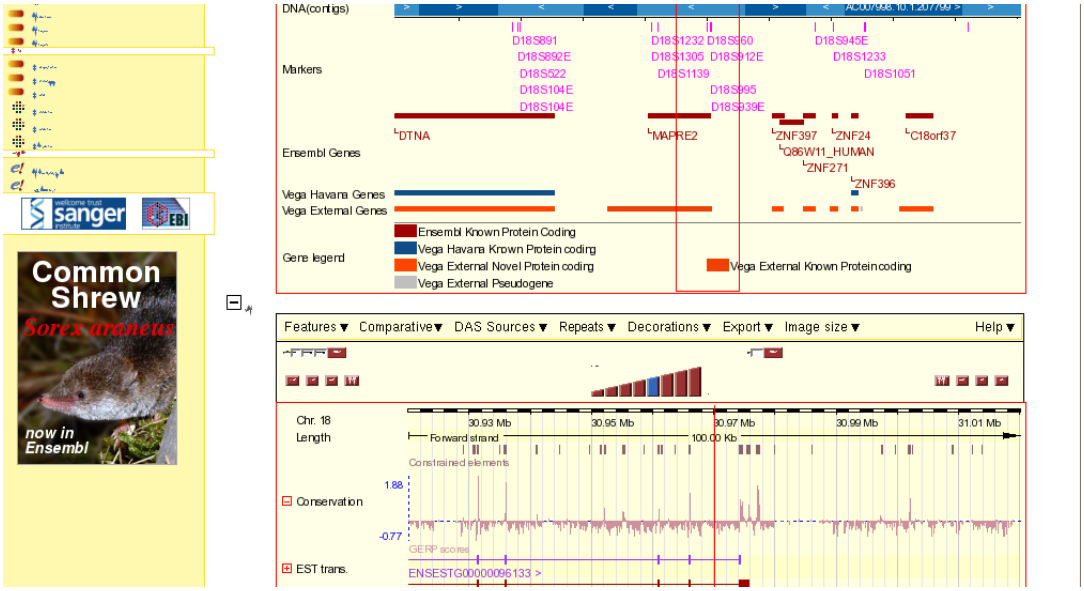

Figure 2.2: *Screenshot of e!Ensembl Contig View.*

- It seems that there is too much information packet in a way that it is not easy to use or understand.
- In my Firefox under Linux the general view of the current chromosome partially falls out of the window and it's unreachable.
- The tool is not interactive in the sense that it doesn't allow direct interaction with the genome (such as moving it).
- Inclusion of new kinds of features from other databases is not possible.
- The interface has, from my point of view, too much text in the image field.

- **e!Ensembl Contig View** (`http://www.ensembl.org/index.html`). This one is a very powerful online tool from the EMBL Project (fig 2.2).

  Positive Aspects

  - There are four views with different detail levels opened at the same time in the main screen: chromosome, overview, detailed and basepair.
  - It integrates many information from many different sources.
  - Its user interface is clean and easy to understand.
  - It includes some analisys tools.
  - It provides different export formats.

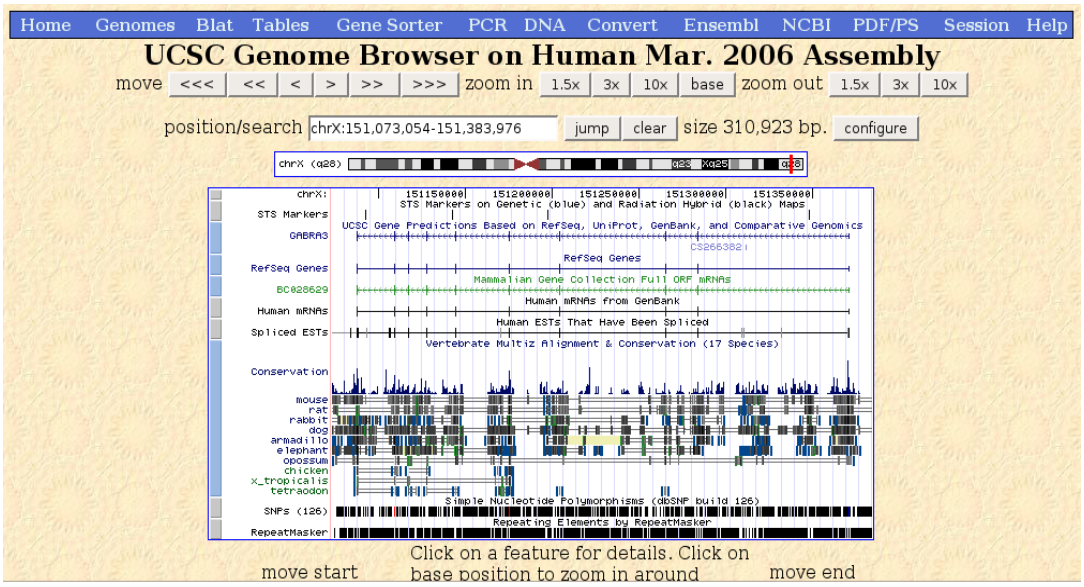

Figure 2.3: *Screenshot of UCSC Genome Browser.*

Negative Aspects

- It does not have direct manipulation. It means that, for instance, it is not possible to move the genome representation with mouse dragging.

- **UCSC Genome Browser**. This one has some similarities with the two previous tools: all of them are online tools able to generate static images and to manage sequences as long as one chromosome. (fig 2.3).

  Positive Aspects

  - It has a cleaner and more customizable interface than other two.
  - It supports a broad range of features from diverse databases and different ways of showing them.
  - It can show different aligned genomes to compare between.
  - It has a useful general view to easily jump to different places on the chromosome.
  - It offers a configuration page with multiple parameters to adjust.

  Negative Aspects

  - It does not have direct manipulation.

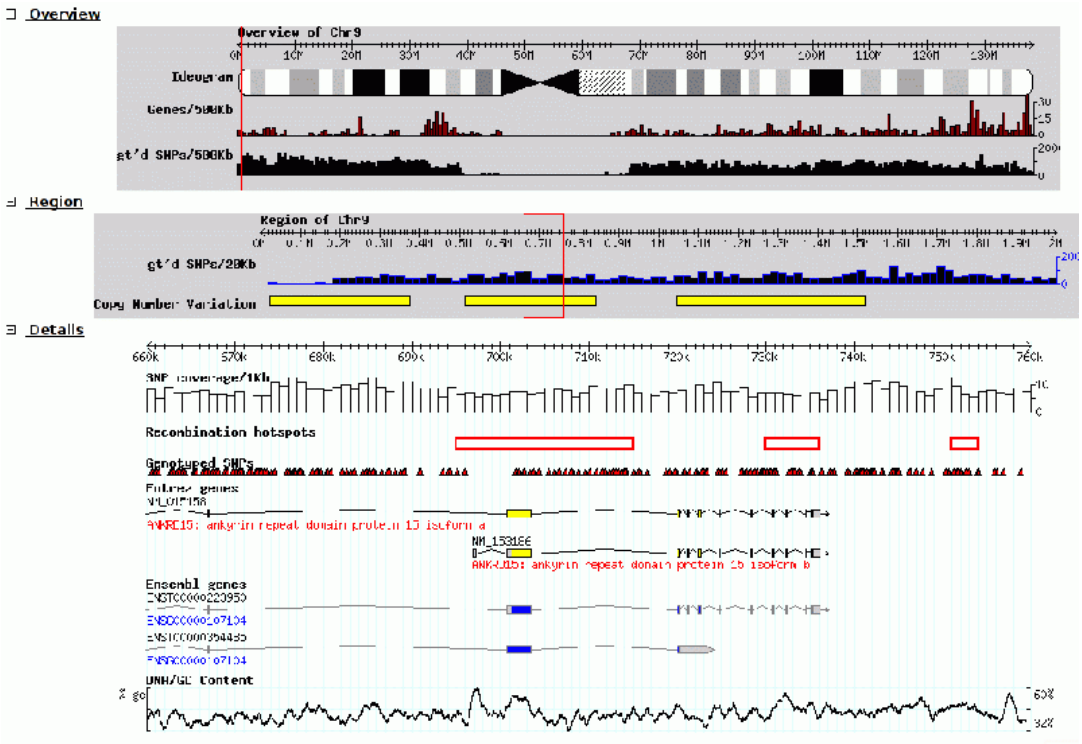

Figure 2.4: *Screenshot of GBrowse.*

- **GBrowse** (`www.gmod.org/wiki/index.php/GBrowse`) [4] is the last web based tool listed here. It works generating static images like the other ones mentioned above (fig 2.4).

  Positive Aspects

  - It has a detalied zoomable view as well as a bird eye view for faster navigation.
  - It includes different interesting options such as modules to talk to databases and other plugins.

  Negative Aspects

  - This tool is not ready to use. You need to dowload, intall and configure it in your own server.[1]
  - It does not have direct manipulation.

[1] Actually, there are two examples of servers using it, but neither of them is working on the human genome.

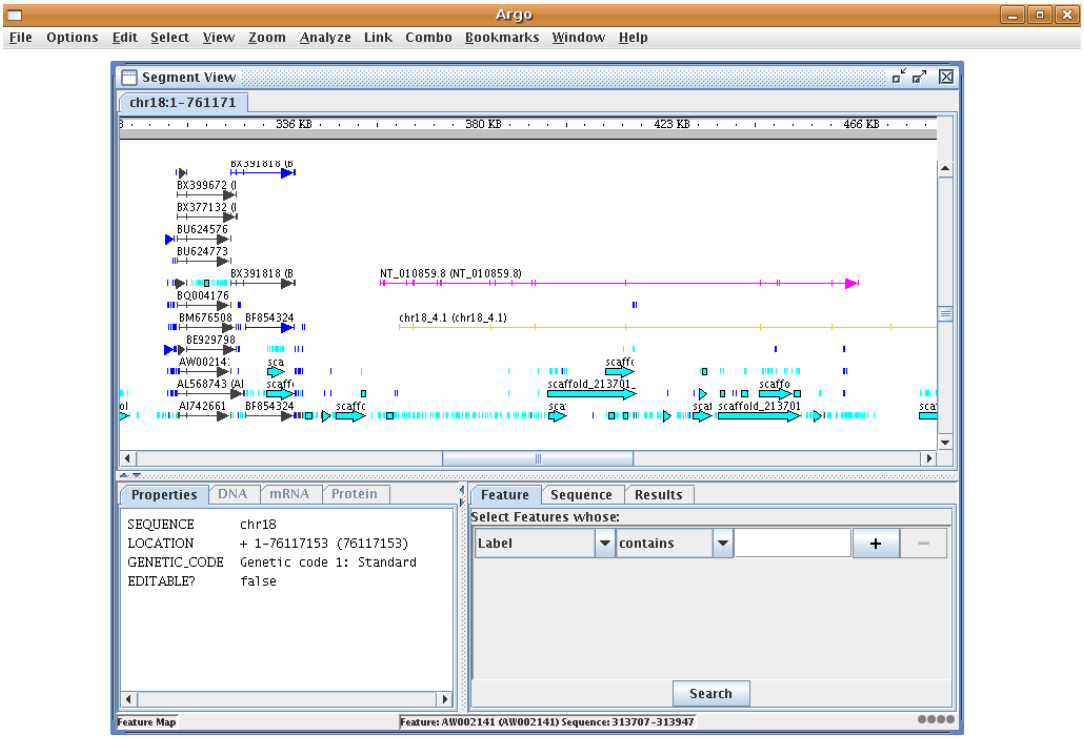

Figure 2.5: *Screenshot of Argo.*

- **Argo** (`http://www.broad.mit.edu/annotation/argo/`) It is a JAVA desktop application from the Broad Institute, a research collaboration of MIT, Harvard and Whitehead Institute. (fig 2.5).

  Positive Aspects

  - It allows feature editing.
  - It allows quasi-direct interaction, since the genome can be moved, but not by dragging it but via an scroll bar.
  - It can automatically read sequences from various sites and has quite a rich but fixed set of features.
  - The data layout is really clear.
  - An applet version with minimal functionality exists.
  - It has a tool called ComBo [5] which allows users to compare different aligned sequences.

  Negative Aspects

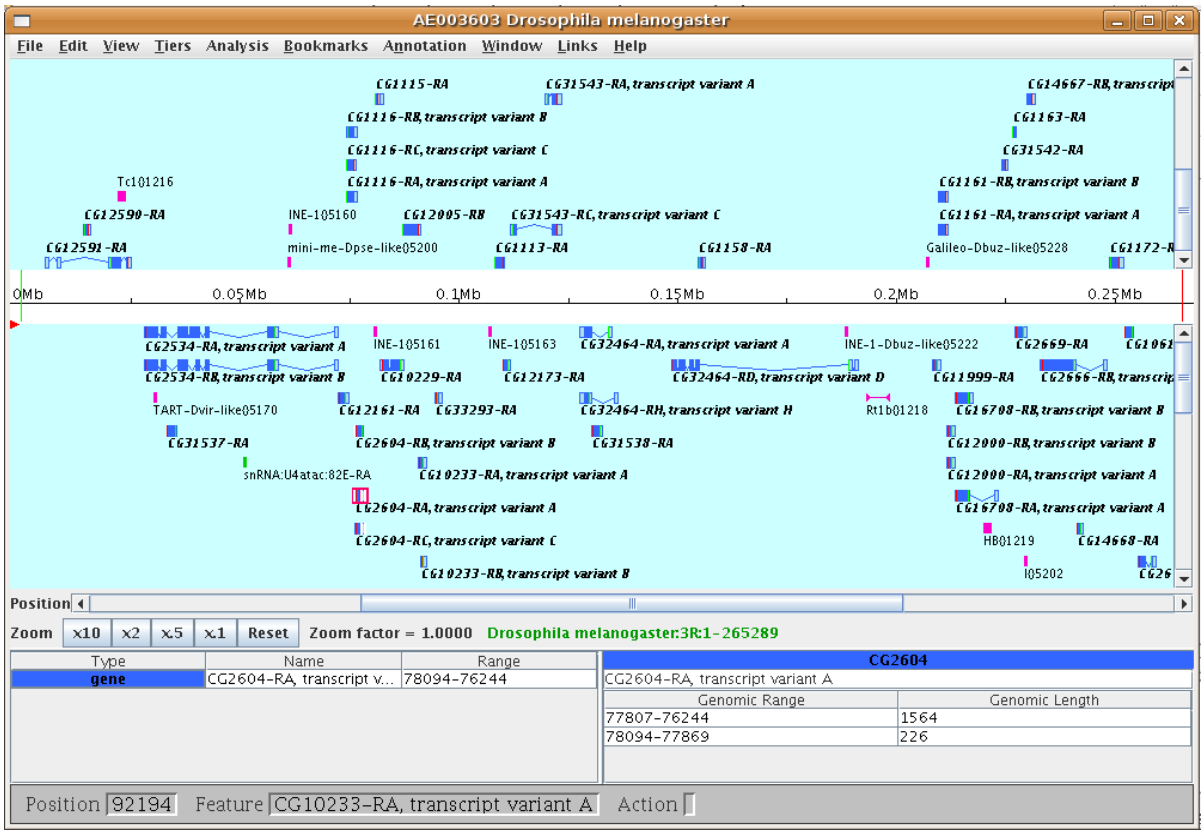

Figure 2.6: *Screenshot of Apollo.*

  - It is somewhat slow when working with medium-large sequences (they say from 500kb above) but working with a sequence of 400kb and adding new features renders it unresponsible for more than thirty seconds.
  - The information given about each feature is a little bit poor.
  - I have been unable to access the applet version. It returns a server error.

- **Apollo**(`http://fruitfly.org/annot/apollo/`) [6] It is also a JAVA desktop application and it is part of GMOD like GBrowse is (fig 2.6).

  Positive Aspects

  - It can also load sequences from the Internet like Argo, but it seems to do it faster.
  - The graphical representation of the features is quite good.
  - It is intended to help in curating annotation data and thus provides means to add and edit annotation

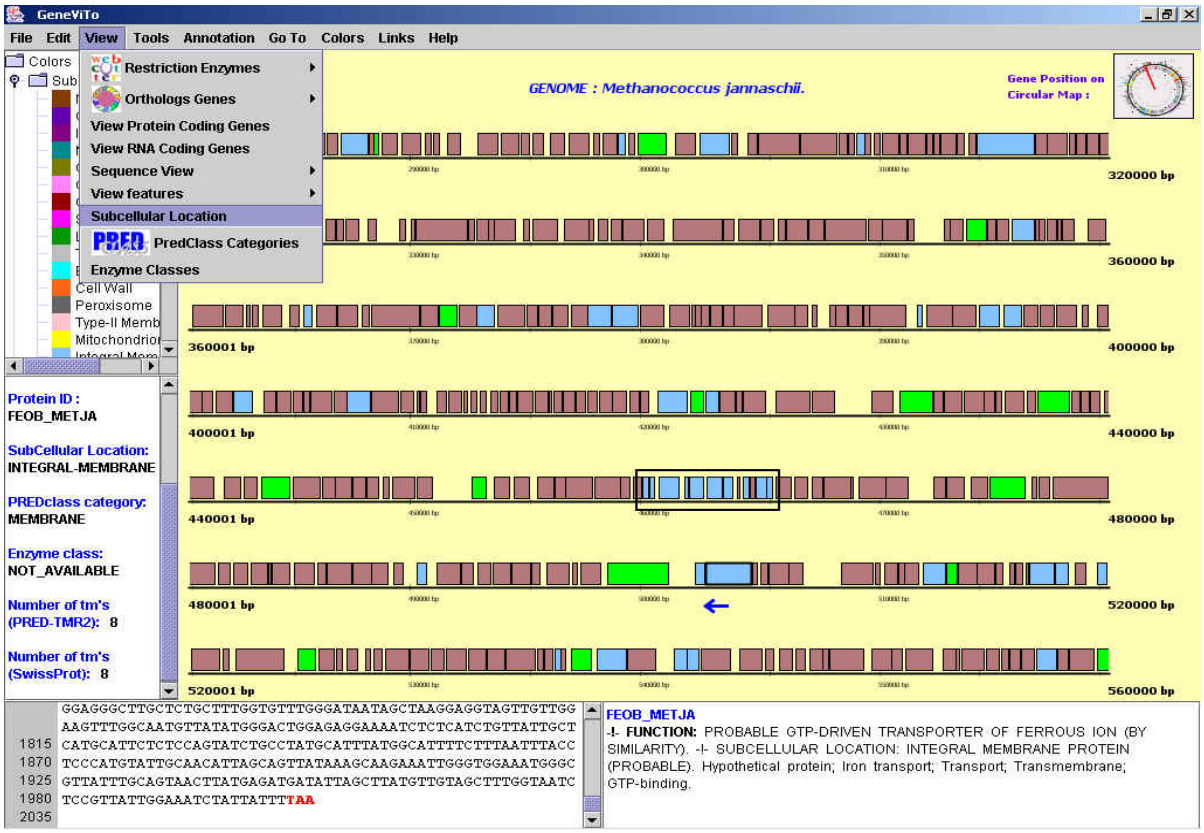

Figure 2.7: *Screenshot of GeneViTo.*

– It has been in real use for some time

Negative Aspects

– It has a limited set of features to show.

- **GeneViTo (Genome Visualization Tool)** (`http://athina.biol.uoa.gr/bioinformatics/GENEVITO/`) is a somewhat old JAVA based desktop application made in 2003 in the University of Athens. It is not possible to download it without requesting it from the authors, so I haven't tried it by myself. The main problem it has is that genomes have to be loaded in a special format and only two circular bacterial genomes are provided in the webpage. It doesn't give any information on wich features it shows or if it's possible to add new ones from other sources (fig 2.7).

We have viewed the most important genome browsers in use today. On one hand, we had the online web tools, all of them very similar. They had a lot of information available to show (altough some of them were poorer than the others in that aspect). They all had quite good graphical static output and various degrees of user-friendliness. Anyway, non of them provided direct manipulation of the genome but were all limited to moving by searching or button clicking, doing it way less intiuitive. On the other hand there were the stand-alone JAVA applications. Those applications had various degrees of usability (the basic option were not always evident) and required installation in our machine. They all collected the sequence information from the internet databases, taking quite long time in some cases, and were not able to treat very large chromosomes. This class of programs had, in most cases, direct interaction with the genome and offered a wider range of analysis options.

Chapter 3

GenomeMap: the proposed solution

In this chapter we will analyze which are the desired properties of our new genome browser and we will sketch a possible solution trying to satisfy all of them.

3.1 Main Goals

Our desired genome browser is intended to be used by biologists and doctors working on genetics. Those users are usually experts on their own fields but they are not always technic enougth to use some of the tools available today. We think our new tool should help those users and not interfere in their work and thus, user-friendliness will be one of our most important goals.

Then, the proposed solution should be

- **Easy to use**
  The intended audience is basically biologists and doctors working on genetics. It should be easy for them to get the information they need: select the correct genome, select features to show and navigate to the selected zone. A method for getting extended information and exporting it in a suitable format would be desirable too.
- **Informative**
  In spite of its simplicity, the application have to be informative enough to be useful. Its user interface should provide as much information as possible while not being bloated. The use of visual clues is preferred over large amounts of text.

- **Interactive**
  Direct interaction of the user with the genome being explored is also wanted. Direct interaction means being able to move over the genome, zooming in and out at the desired places. This movement should be smooth and fast. Other kinds of interaction are also interesting (moving to a gene or position, etc...) but these are not a substitute for direct interaction.
- **Ready to Use**
  The application should not require any kind of installation process nor data collection. For many of the intended users, genomic browsing is only a small part of their experiment planning process sp that should not be difficult for them nor take a lot of time. It means that all pre-work such as downloading data and transforming it to a suitable format has been done for them and preferably they do not even need to know that.
- **Easily Extensible**
  That application have to be easily extensible by adding new kinds of features, genomes and data sources as well as new output formats with little effort. An easy and well defined way for other applications (such as gene predictors) to interact with the tool would be desirable too.
- **Customizable**
  Customization of the application appearance (feature representation, etc...) and content (which features to show and where) is an interesting feature too. In addition, users should be able to show custom features based on their own data, maybe using a specific file format.

3.2 The Solution

We think the best way of producing a ready-to-use application nowadays is using web technologies. Web browsers are installed on almost every computer and they have evolved into an environment where quite powerful applications can run. Therefore our application will be a web application running in the browser.

The main part of the window will be a sliding panel containing an image representing the selected genome. Genes and features will be displayed here in an graphical way allowing several of them to be used at the same time. The genome will be shown as a thin horizontal line with numbers next to it marking the base position and at appropiate zoom levels the actual base sequence will be shown. Genes will be represented in some way below the genome while features will be drawn as marks on the upper side. Different kind of features will have different colors, sizes and z-index and the user will be able to freely change those settings.

There will be a toolbar/menu providing the most usual options for navigation (navigate to a gene, to an exact base or to a feature, show an exact range of bases...), feature displaying (show/hide STS, SNPs, microRNAs...) and searching facilities.

An extended information pane will also be included. It will be used to show extended and precise information about features the mouse moves over. Extended information for genes will include symbol and name, geneId, other identifiers, exact position in genome, description, GO identifiers associated with them, accession numbers for coded proteins and links to databases with more information (at least to the NCBI, HUGO, KEGG and GO if available). Feature info will depend on the type of feature but at least identifiers and exact position information will be given. The extended information will be generated on the fly and it will be easy to add to it new fields if they are required.

There will be an additional configuration page where users will select personal adjustments for feature styles, etc, and an additional window where users will be able to select for showing not so common features and ideally features coming from other softwares like a microRNA target prediction. Subseting of features will also be available here, to, for example, show only genes involved in cancer pathways, or only those SNPs with a wild-type frequency less than 0.5.

To feed the web application with actual content there will be some server side applications and a database to store the information. The base genomic information (mainly the base sequence and the genes) will be the last release of the reference genome at the NCBI. There will be programs (parsers, batch queriers...) to extract the information from its source, filter it and save it to the database. Going to the original source every time the information is needed would give more up-to-date data but it would be absolutely impractical due to the amount of time needed. Furthermore, parsing in advance, give us the possibility to perform exact searches on the reference sequence and to find the exact location of each feature.

There will be some scripts to actually interface with the database and return the needed info and another one able to create the genome images on the fly to serve them to the web application.

Chapter 4

Implementation

In this chapter we will see the main design and technological decisions made. Some technologies used will be presented and partially evaluated. A prototype implementation will also be shown and the viability of the project will be stated.

4.1 Why a prototype?

One of the aspects in which `Genome Map` differentiates from other similar projects is direct interaction with the genome, that provides a more natural browsing experience, but at the same time, this is one of the most challenging. Browsers were not initially designed to act as application containers and so they offer a limited API and are not very optimized for code execution. Furthermore, JavaScript$^{\triangledown}$, the language used in web applications programming, is known to be slow and lacks any kind of drawing capabilities more than those offered by HTML itself.

All those factors make it difficult to implement a web application able to respond to mouse events and move the genome representation smoothly enough to not negatively incide in the user experience. Then, one of the main goals of this first implementation was to demonstrate that such application is possible if coded carefully and that satisfying user experience can be achieved when the right ideas are applied at the right places.

A basic prototype implementation with the basic client side features is working and it is able to move the genome representation smoothly. It mainly lacks the feature integration since at the moment it is showing only some sample data instead of real features but, as a technology test, it has achieved quite good results.

4.2 General decisions

The first non-technic decision to make is which genome do we want to use as the base reference. We decided to use the NCBI reference assembly as the base genome, concretely the human reference genome version 36.2 (14th September 2006). We dowloaded the genebank files since they contain the sequences as well as gene and STS information. For the prototype implementation we will work only on chromosome 18, wich has 766 genes in 77,753,510 bases. Of those bases, 74,534,531 have been successfully determined.

There is one genebank file for each chromosome (24 different in humans) and inside of each file there are different contig sequences along with their annotation. A contig sequence is the maximum length contig they have been able to assemble from the experimental data. It means that contig content and contig order is known but the sequence between contigs is unknown and of an unknown lenght. Thus, referencing the base positions from the beginning of the sequence can be potentially dangerous and misleading so we decided to reference each base counting from the beginning of its contig, i.e. to uniquely identify a single base a pair (contig, number of base) is needed.

Different contigs of the same chromosome will be drawn one next to the other clearly marking their divisions but different genomes will not be drawn in the same genome window at the same time. This works the same way in all other genomic browsers we tried.

4.3 Technology Choices

Once decided that the application will run as a web based tool there are not many technologies to choose from. We used XHTML$^{\nabla}$ as the base rendering technology, CSS$^{\nabla}$ to define the styling and JavaScript to add functionality. This combination is usually known as DHTML$^{\nabla}$ (Dynamic HTML).

During the program execution some data may be requested from the server (such as extended gene information or new features). In those cases we will use AJAX$^{\nabla}$ technology to get the information without triggering a whole page reload. In fact, JSON$^{\nabla}$ (`http:\\www.json.org`) will be used for data serialization instead of XML due to its much less verbose nature and its great integration with JavaScript processing (maybe in this case AJAJ would be a more correct name but AJAX is the one in use.).

Apart of eficiency, cross browser compatibility is one of the greatest challenges in development of web applications. JavaScript was initially developed by

the browsers themselves without any kind of official language specification and with many propriety extensions. Nowadays there is a standard specification by the ECMA (European Computer Manufacturers Association) `http://www.ecma-international.org/publications/standards/Ecma-262.htm`▽ and browsers are slowly converging onto it, but there are still many differences and those differences are specially important in the way events are handled, a very important feature for our application. To help us in the cross browser programming, we used Prototype `http://www.prototypejs.org/`, a JavaScript library extending the base language with useful functions and wrapping arround browser incompatibilities. The tooltips apperaring when browsing over gens and features, are generated using a specific library called Wz_Tooltip (`http://www.walterzorn.com`).

The server is an Apache 1.3 running on Solaris. To build up the dynamically generated pages (for example the information about available zoom levels is injected in the main page at loading time to speed up things a bit) we use PHP. This la nguage is also used to write the scripts answering the AJAX calls. The code generating the genome images on the fly is written in perl and run as a cgi script by Apache itself. Perl is used also, in conjunction with C++, in the parsing of information files and will be used to write the scripts gathering information from other sources. MySQL is the database used to store all needed information except for the sequences that are stored as plain text files in fasta format.

4.4 The implementation

4.4.1 Client Side

The prototype page has been designed following the previous seen ideas. It has a central sliding genome map que user can move by click-and-drag with the mouse and zoom in and out using the right buttons (2 in fig 4.1). It also has a menu on the left with the main options written (1 in fig 4.1) and an information pane on the rigth side of the screen with some basic info about the feature under the mouse pointer, a gene in the figure (3 in fig 4.1).

The most important part of the page is the central sliding window. It is the reason for the application to exist and it is also the most complex part. It is actually a second page contained into an iframe. Other configurations has been tried but this one has proven to be the fastest setup. The page inside the iframe tag contains all the complexity of the application and the main page acts only as a user interface providing access to the functions of the inner web page.

The genome images are absolutely positioned with respect to the inner page

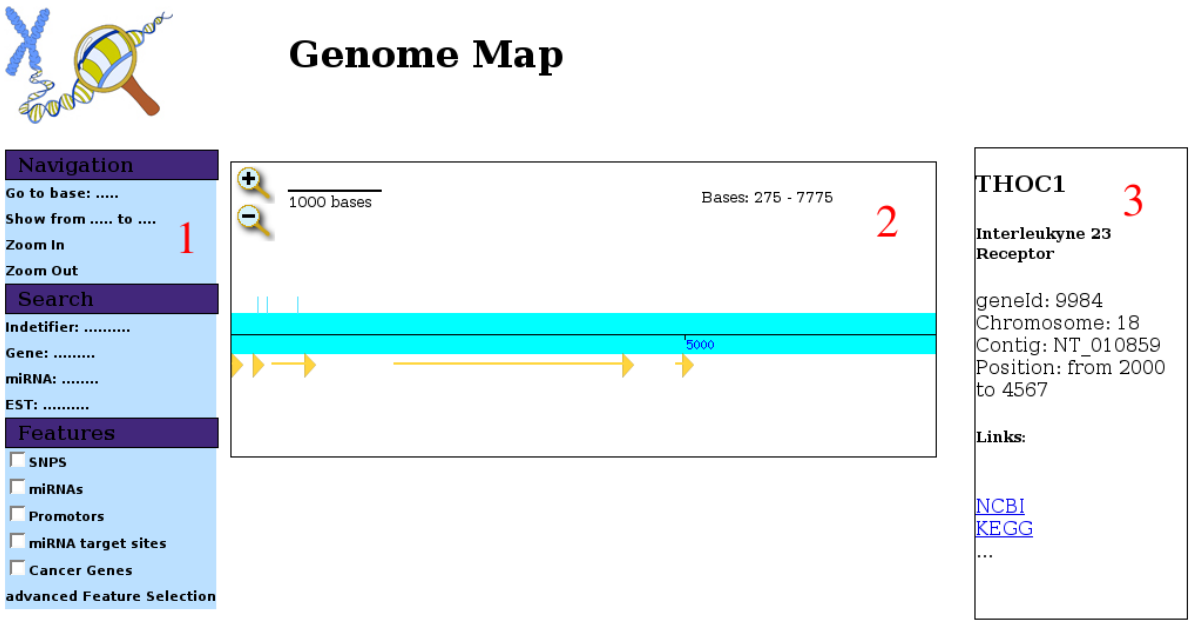

Figure 4.1: *Screenshot of the prototype implementation of Genome Map. Note this is only a first implementation and the appearance will be changed.*

and they are not displayed when they overflow over the iframe margins (the images are, in fact, longer than the margin itself and there are five of them in a row). Features are absolutely positioned with respect to a sliding frame contained in the inner page and so, when dragging the genome, it is only necessary to move the images and the sliding frame.

When the margin at one side of the sliding frame is too small, an image is added on that side and removed from the other and the features are updated accordingly. To save CPU time, this margin check is done only on mouse up events. This little trick has been proven to be very useful as the incrase in speed has been very good.

New features the user might ask for are feched via AJAX calls, to ensure the page doesn't reload entirely. AJAX is used also to retrieve the extended information about every feature the mouse hovers.

4.4.2 Server Side

There is a C++program to parse the genebank files. It extracts the contig sequences and stores them in files and extracts. It also extracts gene info (symbol, name, position, strand, GOids...) and stores it in the MySQL database.

A perl CGI script generates the images using GD library (`http://www.libgd.org/Main_Page`). But why generating the images on the fly? Our first thougt was pregenerating all images in a preprocessing, but in some zoom levels where only 100 bases are placed in each image that would be as much as about 10 000 000 of images for only one zoom level. This would be too much stress for the file system and absolutely not scalable to different genomes. The second option was storing the images as blobs in the database but the images were being served too slow to be useful. And so, the third approach was chosen and images are now generated on the fly.

Chapter 5

Conclusions and Future Work

Visual genome inspection is an important tool specially for biologists and doctors with not much technical skills. Inparticular, it is very useful for easily and fast finding and understanding relations between genomic features.

For the last few years some genome browsers have been build, both online and stand-alone. Anyway, web based ones lack intuitive genome manipulation and some of them have a little clumsy interface but they are ready to use and have a pretty big set of features to choose on. Stand alone applications, on the other hand, have a more intuitive interaction and are generally more powerful, some of them with great genome and features images but they lack readiness since the user have to download and install them and all data (very large data files in some cases) have to be downloaded prior to use.

Our aim is to build a new genome browser trying to get the best of both worlds, taking the rich user interface to the web. At the moment, we've got a running prototype that has been a good technology test. We have shown it is possible to have an interactive web base genome browser if right technologies are used.

5.0.3 Future Work

Since we have only developed a technological prototype a lot of work has still to be done. Here is a list of some of the most promising ideas and planned additions (not in any special order).

- Add real features to the system. Build up a quite complete list of features available from diverse datasources. Create parsers for them and add them to the system.

- Define an interchange based (most probably based on XML) to allow feature request to and from other applications. As a first step to it, make the program interface with Promo, the promoter predictor.
- Build the extended feature selection and definition page. Allow selecting only subsets of features (based on functional information, or pathways, etc...). This could be very interesting for example in cancer research.
- Build a preferences page where user can change their visual preferences like feature marks or styles. Preferences should be exportable and importable.
- Add exporting options to the application. It might be interesting to export rhe current sequences and features in various formats. Allow fine grain selection of what is going to be exported.
- Add support for analysis tools. Tools designed to answer some predefined questions like features sharing sequences, or to find features combinations.
- Allow users to have more than one genome image at the same time. Different genome images would use different zoom levels or eventually different sequences. If they were using the same base sequence, they would be be linked or no. If linked, moving one of them moves the others and visual clues are provided to show how they all relate.
- Allow users to add their own annotations from files. They could represent multiple things, like regions of interest, already studied dequences, etc...
- Add links to the companies selling lab products. For example, when browsing over an SNP, a link to the Applied Biosystems web page where the RT-PCR primer for that SNP is sold. This would save much time to researchers.

Glossary

Genome Genome refers to the whole genetic code of an organism, all the DNA sequences contained in its cells. When talking about the genome of an specie we are actually refering to the genome of one individual of such specie, the one wich have been selected to be sequenced.

Nitrogenated bases These are the tiny pieces making the code in the genomic sequence. They are four different molecules wich can be linked in any order to make a long chain. The order in which thos bases are linked is what is called the genome sequence. They can be A (Adenine), T (Thymine), C (Cytosine) and G (Guanine).

Chromosome On complex organisms, the DNA sequence is usually organized in parts called chromosomes. They are very long, continuous pieces of DNA, which contains many genes, regulatory elements, etc..

Gene A gene is a part of of the genome wich can be translated into a protein. A gene can codify more than one protein (via a process called alterative plicing). They are one of the most important genomic features.

Protein A protein is a macromolecule made of a chain of aminoacids. Protein can have very different roles in organisms, from structural function to enzymatic activity. The sequence of aminoacids confoming a protein is codified in a gene.

DNA The Deoxyribonucleic acid is the molecule which constitutes the genomic code. It acts as a long term memory and stores all the information concerning the developement and fucntioning of a living organism.

microRNA This class of tiny molecules (18-22bp) play a fundamental role in genomic regulation. They form a complex network of inteactions controlling where and how much genes are expresed.

SNP (Single Nucleotide Polymorphism) SNPs are mutations in only one nucleotide. They are thought to be the base of the differences between different

individuals of the same specie. A polymorphism in a gene, can reduce or enhace the protein activity or even render it completely useless.

Promoter The first step in creating a protein from a gene is transcription, that is, copying the sequence of DNA in a RNA molecule. This process is controlled and initied by promoters, regions of the genomic sequences that are bindings for involved molecules.

intron An intron is a part of a sequence of a gene that althought is transcribed is removed from the sequence before the final translation to protein. Therefore these are non coding sequences inside the genes.

exon Exons are the contrary of introns, it is, the protein coding parts of genes.

oncogene An oncogene is a gene involved in carcinogenesys processes.

STS (Sequence Tagged Site) Since reference sequences evolve as the genome is more deeply studied, it is important to have fixed references all across the genome for the researchers to work. STS are small unique sequences used as references.

JavaScript JavaScript is an implementation of ECMAScript standard but its usually used to reference ECMAScript itself. It is the language used inside web browsers to provide functionality to webpages.

XHTML Is an extension of the HTML format accepting somr more tags and functionality. It is also XML compliant.

CSS (Cascading Style Sheets) CSS is an standar used to modify the style of HTML elements.

DHTML (Dynamic HTML) This usually refers to the combined use of XHTML, CSS and JavaScript to build and run a web page.

AJAX (Asynchronous JavaScript and XML) AJAX is a technology based on already existing bits and pieces to provide a new and more interactive functionality to web pages.

JSON (JavaScript Object Notation) This is a very lightweigth text based format for object serialization. It is based in JavaScript and thus integrates very good with it. It is way less verbose than XML.

Bibliography

[1] Benjamin Lewin. *Genes, VIII*. Pearson Prentice Hall, 2004.

[2] S Griffiths-Jones. The microrna registry. *Nucleic Acid Research*, 32 (Database issue):D109–D111, 2004.

[3] P Sethupathy, B Corda, and AG Hatzigeorgiou. Tarbase: a comprehensive database of experimentally supported animal microrna targets. *RNA*, 12:192–197, 2006.

[4] Lincoln D. Stein, Christopher Mungall, ShengQiang Shu, Michael Caudy, Marco Mangone, Allen Day, Elizabeth Nickerson, Jason E. Stajich, Todd W. Harris, Adrian Arva, and Suzanna Lewis. The Generic Genome Browser: A Building Block for a Model Organism System Database. *Genome Res.*, 12(10):1599–1610, 2002.

[5] Reinhard Engels, Tamara Yu, Chris Burge, Jill P. Mesirov, David DeCaprio, and James E. Galagan. Combo: a whole genome comparative browser. *Bioinformatics*, 22(14):1782–1783, 2006.

[6] SE Lewis, SMJ Searle, N Harris, et al. Apollo: A sequence annotation editor. *Genome Biology*, 3(12), 2002.

Index