DEVELOPMENT OF A PROTOTYPE VISUALIZATION APPROACH FOR NEXT-GENERATION SEQUENCING TECHNOLOGIES USING GENOME NAVIGATOR FRAMEWORK

by

TIANGE CUI

A thesis submitted to the

Graduate School-Camden

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of Master of Science

Graduate Program in Biology

written under the direction of

Andrey Grigoriev

and approved by

_____

Andrey Grigoriev

_____

Daniel H. Shain

_____

Kwangwon Lee

Camden, New Jersey. May 2012

ABSTRACT OF THE THESIS

DEVELOPMENT OF A PROTOTYPE VISUALIZATION APPROACH FOR NEXT-GENERATION SEQUENCING TECHNOLOGIES USING GENOME NAVIGATOR FRAMEWORK

By TIANGE CUI

Thesis Director:

Andrey Grigoriev

The advent of next-generation sequencing (NGS) technologies has greatly accelerated the development of the genome analysis. Sequencing efficiency has improved significantly in last decade after the sequencing of the human genome was completed. A large number of NGS platforms have been developed with incredibly lower cost and higher throughputs. Terabyte of data are generated every day along with the NGS workstations, but the pace of analyzing these data is still not satisfactory. The lag between the available data and the interpretation tools will ultimately slow the progress of NGS. In this work, I aimed to add functionalities to the display of NGS data in an existing genomic visualization tool - Genome Navigator (GN). By adding new features especially for the display of NGS data, the prototype GN can be used in various sequencing projects

and help researchers visualize and interpret sequencing data, while exploring interested genomic regions, and validating their hypotheses. After updates, the prototype GN is able to handle large amounts of sequencing data and display thousands of reads at the same time. It supports several display modes such as stacked view, condensed view, and complex view. At different zoom levels, different amounts of details are shown to the users. Moreover, several utility tools have been developed to support GN when dealing with different input formats. Several new server interface features are designed for interaction with the user. I used the real sequencing data, tested the functionalities of the improvements, and had several positive results. I expect that GN will also be useful in other NGS applications and play an important role in today's NGS research.

ACKNOWLEDGEMENT

I would like to thank Dr. Andrey Grigoriev for being not only a wonderful advisor but also a respected mentor to me. He greatly inspires my interests in next-generation sequencing technologies and my desires to explore the genetics world. Under his guidance, I was able to find the directions for my project and the best solutions to my problems. I am very grateful to have such an opportunity to work with Dr. Grigoriev. I would also like to thank Kevin Abbey who set up our testing server, Sulbha Choudhari for the constructing of some sequencing data, and Sean Smith for the critical reading and comments on the manuscript. In addition, I like to thank Dr. Kwangwon Lee and Joseph Kawash for providing sequencing data of *Neurospora crassa* that greatly facilitated my designing process.

I would also like to thank Dr. Daniel H. Shain for recruiting me to the program, Dr. Nir Yakoby for the enlightening instructions and memorable *Drosophila* lab experiences, and all the members in the Biology department who gave me all kinds of help when I needed it. It has been a great journey, and I am thankful for my family and all my friends for their incredible support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Since Nobel laureates Frederick Sanger and Walter Gilbert independently developed a rapid determination method for DNA sequencing (Sanger *et al.* 1977; Maxam and Gilbert 1977), many alternative sequencing methods that are creative refinements of Sanger's chain-termination method, aspiring to reduce costs and time, have been developed. In 2005, 454 Life Sciences (Roche) first developed a 454 pyrosequencing method and sequenced the full genome of *Mycoplasma genitalium* (Margulies *et al.* 2005). In the same year, George Church's lab also developed a multiplex polony sequencing protocol using the same strategy (Shendure *et al.* 2005). Since then, implementations of high-throughput NGS technologies have been continually optimized and applied to various commercial platforms (Table 1). In 2012, two major sequencing companies Life Technologies and Illumina announced their newest NGS systems, Ion Proton (Life Technologies 2012) and MiSeq 2500, which enable researchers and clinicians to sequence whole human genome in approximately 24 hours with the cost of as low as $1,000 (Ilumina 2012). With such developments on cost, read length, and throughput per run, unprecedented amounts of sequencing data are generated every day. As a result, one of the most pressing problems to researchers is how to correctly and efficiently interpret these data and understand the biological meanings and applications behind them.

There are several categories of NGS research, such as *de novo* assembly for target gene of interests (Miller *et al.* 2010) and sequence alignment (Li *et al.* 2009) for identifying variations. Many areas benefit from these NGS technologies including but not limited to *de novo* sequencing, ChIP-sequencing, genomic resequencing of human genomes for identifying variations among the populations (Altshuler 2010), and RNA sequencing for transcriptome analysis (Daines *et al.* 2010). Ideally, visualization tools can not only show the reference sequences, sequence contigs, and all the sequencing reads together, but also give the user a clue on what are the characteristics of the sequencing data and what are the discrepancies between reference sequence and sequencing data. Because the NGS data are mostly huge in size and written in plain text, direct attempts will only cause informatics challenges. The ability to view and manually interact with the raw data in a straightforward manner is an indispensible part in data analysis. In order to do these, the visualization tools first need to be able to deal with large sequencing data, which means finding the best way to pack them into the limited computer resources; also the tools should be able to provide users a broader view about the interested region so that hypotheses can be made with the help of the software. Therefore an easily customized, highly compatible, simply installed visualization tool is useful for any researchers and developers before the analysis of genomic data. There are quite a number of visualization tools trying to help users achieve the aforementioned goals. Viewers for Sanger-type sequencing data such as Consed (Gordon *et al.* 1998) and

BugView (Leader 2004) are not suitable for NGS data anymore while other available tools for visualizing NGS data have their own merits and drawbacks. Some of the disadvantages such as limited operation system support, installation limitation, stand-alone designing, lacking of analytical assistance, etc. need to be improved for more effective investigations.

Here, we introduce the GN, a visualization tool that employs a Java applet, DerBrowser, as its display tool together with CGI scripts to serve data to the applet (Grigoriev 1997, 1998). GN is designed by Dr. Andrey Grigoriev and recently redeveloped to display NGS data and serve as an interactive World-Wide-Web GUI to any data sources containing NGS data on mapped objects. Designed in 1996, the Java applet DerBrowser no longer meets the needs of NGS's increasing magnitude of data. So we decided to update this applet by adding new features (such as massive data handling ability, right-click behavior, variations indication, informative statistic graph, etc.) to make it suitable for future research and usage. Bearing this in mind, a whole set of WWW server interface was redesigned with a series of supporting utility tools that I developed. Several stripes have been refactored for better extensibility and display accuracy. The Graphical user interface (GUI) of DerBorwser is shown in Fig. 1.

MATERIALS AND METHODS

Materials

a. NGS Data

In this work, Illumina and SOLiD sequencing data of filamentous fungi *Neurospora crassa* (Strain FGSC 2223 and FGSC 4825), which is recognized as a eukaryotic model organism from Dr. Kwangwon Lee's lab are used to assist validating the functionalities of GN. Some Sequence Alignment/Map (SAM) files of *N. crassa* supercontigs (named supercont10.1 to supercont10.7, using Broad Institute assembly 10 represent 7 chromosomes) are constructed by Dr. Andrey Grigoriev and Sulbha Choudhari. The reference sequences used in this project are *N. crassa* OR74A (NC10) from Broad Institute.

b. Server

The specifications of the server are as following:

Address:

http://ccib-bsb-164.rutgers.edu

Software:

Operating system: CentOS (Version release 6.2), Web server: Apache (Version 2.2.15), Java JRE: Oracle Java (Version 1.7.0_02).

Hardware:

Dell PowerEdge R415, 2x 3.0 GHz AMD Opteron Processor 4284, 32 GB DDR3 1600MHz ECC Memory, Raid 1 (mirror) 2x 500 GB SATA 7200 rpm.

c. Development Software

Eclipse IDE for Java Developers (Version: Indigo Release), Eclipse SDK (Version: 3.7.0), ActivePerl (Version: 5.14.2 Build 1402 (64-bit)), SAMtools (Version 1.4), and SSH Secure Shell for Workstations (Version 3.2).

Development workflow

The java applet DerBrowser is programmed using Eclipse IDE for Java Developers on a local machine (Windows 7 Home Premium SP1 64bit / Ubuntu 11.10 for Intel x86, Intel Core2 Quad Q9000 2.0GHz, 4GB DDR3 1333MHz). The workbench of Eclipse is shown in Fig. 2. DerBrowser is a jar (Java ARchive) file which aggregates many Java class files and other resources. In order to update the applet, the jar file must be first imported into the Eclipse. Then by understanding the functionalities and connections between each class, users could revise the code and add new features to the applet. Fig. 3 shows an example code of PLOT class for drawing plot graphs. After finishing the updates, all the Java classes will be exported into a new jar file as the new DerBrowser.

The applet was then uploaded and tested on the server. The inputs for the GN can come from multiple sources, such as the raw data from sequencing companies, local research data generated by researchers, available data from online databases, or genes with certain information (Fig. 4). Then data will be processed and unified by various converting tools depending on the data types. These data are first processed by BWA (Li and Durbin 2009) and SAMtools (Li *et al.* 2009) into indexed SAM files. These SAM files are then processed by a Java utility tool samParse into the format that DerBrowser can interpret. After receiving the input data from the user at the web interface, CGI scripts will extract the desired region from these parsed SAM data, prepare the data stream for the applet, and pass it to the DerBrowser for display.

Development of the utility tools

With various data formats for NGS data, they need to be normalized before being passed to DerBrowser for parsing. And only a certain part of data is actually needed to be extracted from the NGS data to generate the display. Therefore, data processing tools are developed at the same time for such purposes. Tools are written in different languages such as Perl and Java for different cases. The choice of developing language is based on the capability and efficiency (such as Perl is good at text handling while Java is convenient for group data with common attributes) of that language towards the specific task we want to achieve.

RESULTS

Features of GN

a. Enhanced visualization effect and massive data handling ability

By optimizing the calculation for arranging display objects, DerBrowser can accurately display data streams at both low and high zooms levels. In SEQUENCE stripe, more than 6,000 sequencing reads can be shown in one request at the same time (Fig. 5). No misbehaviors and distortions can be seen even under such a compacted area.

Biological meaning/application: This feature enables GN to be used in genome *de novo* sequencing and resequencing projects for handling genome assembly of thousands of reads.

b. A space efficiency solution- condensed view of SEQUENCE-type stripe.

In order to view more objects in one display, a condensed view mode is introduced to the DerBrowser. All objects are defined 2 pixels in height as a constant during semantic zooming (which means by changing the zoom, objects not only change their size, but in additional they can change shape, details or their very presence in the display (Boulos 2003)). Base pair discrepancies are visible with highlighted colors. Note that Fig. 6A uses exactly the same data as Fig. 5, but stacked

view in Fig. 5 can only show partial reads. Under condensed view, DerBrowser shows all of the 6,118 reads in one window (**Fig. 6A**).

Biological meaning/application: This feature gives users an overall impression about the sequencing quality (random errors vs. large amount of erroneously mapped reads) and characteristics (the distribution of repeats, SNPs, deletions or insertions region).

c. Visualization of genomic discrepancies in SEQUENCE-type stripe with semantic zooming.

Once any sequence is defined as the reference sequence, other sequencing data can be compared to this sequence and all the differences will be visualized in highlighted colors (Fig. 7). More specifically, by default all of the consensus base pairs are shown in light gray. For the all variations, deletions are shown in white with a "-" sign on it. Non-coded nucleotides, which are a skipped region from the reference, are also colored white, with a letter "N" on them. Insertions are shown in magenta with a "+" sign on it. The details about the insertions could be read from the adjacent reads at the same position, which is also colored in magenta. Hard clippings which are clipped sequences not present in reference sequence are colored black.

Biological meaning/application: This feature enables users to locate regions with sequencing/assembly errors and inspect the whole region in detail. Combining with

other SNP detection tools or data from other sequencing platforms, GN is also able to help users validate nucleotide polymorphisms. Furthermore, it will help assembly tools to improve their alignment algorithms.

d. Dual-mode for PLOT graphs in relative colors: Histogram and Line graph.

Depending on the value of the "Plot display type" tag, the plot stripe is now able to switch views between two modes. The four nucleobases are predefined in four different colors so that plot graph shows the certain base pair's density in relative colors in a position-dependent manner (Fig. 8).

Biological meaning/application: This feature enables GN not only to display the positional distribution of certain types of data, but also to show their respective density at certain position.

e. User-based dynamic input system.

A set of web server with complementary CGI scripts is designed for users to easily pass all of the parameters for the applets. With CGI scripts, users can customize their own data such as picking a desired region of interest without loading information of the whole chromosome, assigning certain sequence as the reference sequence, or aligning different data all together (Fig. 9).

Biological meaning/application: This feature enables users to investigate

comparative genomics in a customized way. Different genomes of interest can be loaded together and displayed in the same window.

Example Usages

a. SEQUENCE stripe - Validation of potential polymorphisms

With the consensus errors in highlighted color, users are able to find the SNP candidates manually by scrolling along the chosen region. The highlighted colors can tell user whether the variations are random errors or potential SNP locations (Fig. 10). If a variation occurs in most of the sequencing reads at the same position, it may be worth scrutinizing and checking with available database.

b. PLOT stripe - Chromosome-wide view of read coverage

To have a brief impression of the reads' distribution and mapping characteristics, a good usage for GN is to generate a histogram graph showing the total frequency of objects at all positions (Fig. 11). In this way, users are able to evaluate the sequencing quality and the characteristics of a certain chromosome.

c. LOCI stripe – Display the distribution of data with similar features

By defining objects of same features, such as repeats, restriction sites, etc., the LOCI stripe is not only able to show locus on a gene map, but also able to display the

distribution of these data for multiple analysis purposes. Any group of data defined in the LOCI stripe can be shown in a locus-style in the display (Fig. 12).

Supporting utility tools

Three utility tools are developed for the use of format converting: The fasFileSplitor which is written in Perl is able to split a multiple sequences FASTA file into separate ones by certain criteria, such as chromosome name. The samParser which is written in Java is able to interpret a CIGAR string and convert an original SAM file into a normalized SAM file. The plotGenerator which is written in Perl can calculate the distribution of reads from a normalized SAM file within a customized interval.

DISCUSSION

Generality and flexibility of GN

There are several reasons why we deem GN as a useful solution for visualizing NGS data. The first one is its convenient configurations for multiple purposes. As a general genomic visualization tool, GN is not limited to certain sequencing platforms that provide NGS data. Sequencing outputs from two different sequencing platforms, Illumina and SOLiD have been tested. As a common format for reference sequence, FASTA format is the first type of data format that needed to be normalized since it cannot be parsed by DerBrowser directly. Starting with a ">" symbol, the first line is the descriptions, followed by the detailed sequence in one line, until the next “>” appears. A multiple sequences FASTA file needs to be separated before using as a reference sequence, which can be accomplished by fasFileSplitor. A SAM file also needs to be parsed before use. There are 11 mandatory fields in each alignment line, some of which are not currently used by DerBrowser such as bitwise FLAG, observed template length and ASCII of base QUALity plus 33, etc. More importantly, the genetic variation information stored in a SAM file CIGAR string need to be interpreted first. A set of color schemes are added to DerBrowser as described in Fig. 7. With the help of samParser, a SAM file is able to be converted into a .txt file with the interpretation of base pair discrepancies. Integrating utility tools, including the developing ones, into the GN system

in the future will give GN more general power in data handling.

Furthermore, since Java is supported by most web browsers using Java Virtual Machine (JVM), GN is a cross-platform visualization tool. The size of the current version of DerBrowser is only 56kB, which can be instantly downloaded while loading the web page. Because the applet is executed on the user's machine, the running speed is not restricted by the network bandwidth. These are great advantages compared with some of the popular existing genome viewers that either support limited platform(s) (such as Consed (Gordon *et al.* 1998) which is not support Windows), or need to download and install the software on the local machine (such as Genomorama (Gans and Wolinsky 2007) which needed to be downloaded before use), or install certain prerequisite software(s)/package(s)/module(s) before going through a possibly complicated setup process (such as GBrowse (Stein *et al.* 2002) which need to install activePerl and Apache web server together with some Perl modules before use).

There are many other visualization tools written in Java platform with relative merits and disadvantages. Despite the limitations such as system requirements and lacking certain functions towards NGS data, they all have certain unique features. Some tools such as Bluejay (Turinsky *et al.* 2005), CGView (Stothard and Wishart 2005), and GeneViTo (Vernikos *et al.* 2003) have a circular view which can show gene positions on a circular map. Some tools such as Apollo (Lewis *et al.* 2002), Argo (Engels *et al.* 2006), Artemis (Rutherford *et al.* 2000), and SeqVISTA (Hu et al. 2003) can edit and create

annotations. Most of these tools plus GATA (Nix and Eisen 2005), Mauve (Darling *et al.* 2004) and Sockeye (Montgomery *et al.* 2004) are able to save the display as a graphic output. Adding these features will further improve the development of GN. One of the most popular genomic viewers among them which shares similar designing strategies with GN is Integrative Genomics Viewer (IGV) (Robinson *et al.* 2011) developed by Broad Institute. It has similar options for the aligned reads as GN, such as highlighted variations, detailed view when at nucleotide resolution, whole chromosome coverage plot, etc. But as suggested by its name, IGV is much more integrated than the current version of GN. For example, when loading a reference to IGV, it automatically splits the multiple sequences file into separate files in a new folder, which severs the same functions as the fasFileSplitor. Another feature “group tracks” in IGV is an idea similar to the CGI scripts used at GN’s data input interface. With the future development of the integration of utility tools, GN will be greatly improved.

Another advantage of GN is that all the data can be uploaded to the server which any user with permission is able to view the display. Compared to stand-alone genome viewers, users can share their findings by simply sending the hyperlinks of the corresponding regions. Users are even able to generate same displays on different machines at the same time.

Using GN to interpret sequencing data and generate hypotheses

Previous results have shown some features and examples which GN can accomplish. Here, some findings using sequencing data of *N. crassa* are shown with the demonstrations of the role of GN.

Sequencing data from Illumina technologies and SOLiD technologies are compared by aligning them adjacent to each other (Fig. 13; Fig. 14). There are several obvious sequencing discrepancies between these two sequencing technologies. The first one is the different algorithm for dealing with the terminal regions. Illumina technologies tend to have longer reads (80bp) and both ends of reads are sequenced even sometimes there are sequencing errors or alignment errors that are not very informative (Fig.13A). SOLiD technologies have shorter reads (50bp) and lots of clipped sequence present on both ends. This may be due to the remaining primers or the low mapping quality at the end of reads (Fig. 13B). Based on the data examined, Illumina technologies are likely to have more misaligned reads/sequencing errors. SOLiD technologies generate more reads and better coverage, but due to the clipped region, some positions have a low coverage, and it will be hard to determine if the genetic variation is just a random sequencing/alignment error (Fig. 14). For example, at position 49,447, in Fig. 14A there's a G/A discrepancy in most of the reads, but in Fig. 14B, there is only one read in the screenshot showing the same discrepancy. Most reads have a clipped sequence at this region.

Secondly, SNP genotype variations can be found between two sequencing technologies. Current SNP discovery tools have trouble telling if a SNP discrepancy at the SNP site is a sequencing error or an alignment error. For example, in Fig. 13, at position 2,008, Illumina technologies found a T/C/G SNP, but in SOLiD technologies there's a C/G SNP. When expanding to a broader range, some Illumina reads also show a C/G SNP. Giving the fact that there is a 2-bp insertion at position 2,004 before these T/C/G variants, it is quite possible that these T/C/G SNPs are caused by alignment shift. In Fig. 15, at position 2,423, there's a deletion in Illumina reads, but in SOLiD, the sequence matches the reference sequence at this position. By closely scrutinizing this region, the consensus base pairs at position 2,423 are all connected with a clipped sequence with a CC sequence near this position. Noticing that the sequence after deletion is also CC, the best explanation is that the CC in the SOLiD reads is misaligned due to the clipped end and deletions do exist at position 2,423. Under such circumstances, further inquiry at certain locations may result in some meaningful findings.

Thirdly, the ability to organize display objects gives GN great advantages when applied to comparative genomics' study. To assist researchers in understanding the relationship between genome structures and functions across different strains, NGS data of interest can be compared in GN. For example, by comparing the sequencing data of SOLiD-2223 and SOLiD-4825, potential SNP regions can be visualized in Fig. 16 and Fig. 17. In Fig. 16A, there's a G/A discrepancy at position 5,934, but in Fig. 16B, in the

4825 strain most reads have the same genotype variation at position 5,935. It is shown that in 2223 strain there are three genotype discrepancies, which are T/C at position 5,891, G/A at position 1,899 and T/C at position 5,955, but none of them are shown in 4825 strain. In Fig. 17A, there's a T/A discrepancy, but in 4825 strain, there are three genotype changes present at positions 6,628 (T/C), 6,639 (G/A) and 6,711(T/A). These regions maybe candidate areas that are worth further exploring.

Another finding from the whole-chromosome coverage histogram is that A-T content rich regions and multiple-N regions are corresponding to the gaps of the coverage histogram (Fig. 18). Fig. 18A shows the AT-rich regions. Each region uses the minimum value of the AT ratio in the region as the left coordinates and the maximum value of the AT ratio in this region as the right coordinates. Fig. 18B shows the multiple-N nucleotide region. Since the N stands for unknown nucleic acid residue, multiple-N regions are unmapped regions in the reference sequence. Therefore, they are corresponding to the gaps in the coverage map. The whole length for chromosome 2 is 4,478,603bp. The maximum value at a 5K window is 10,533 reads. As shown in Fig. 18 and discussed before, SOLiD generated more reads than Illumina and has a better coverage.

What's next?

Since the priority of this project is to develop and test the feasibility of the prototype GN for handling NGS data, efficiency and performance optimization are not

weighed too much during the development. But there's a possibility to reduce the loading time by optimizing the algorithm for parsing the data stream. In the future, I would like to optimize the performance of the GN framework with regard to loading and displaying data. Since GN is capable of connecting with useful databases, such as GenBank and dbSNP, by building connections between them will provide more information about the selected object. Also, finding alternative storage method (such as cloud storage services) for NGS data before the cost of storage is less expensive than the cost of sequencing is another urgent issue. The connections between GN and new data sources also need to be considered. More data format converting tools need to be developed to enhance the generality of DerBrowser. After that, integrating the whole set into a Java application will make GN an ideal visualization tool for the field of NGS.

Concluding statements

I have shown the emergence of NGS technologies together with some pressing issues; the comparison between major sequencing companies and their current sequencing platforms; the designing purposes for specific features and functionalities of GN; and some example usages with real sequencing data. Several new features have been added to GN for the needs of NGS data. A set of web interface and utility tools are developed to support the GN. All of which suggest GN is a useful visualization tool for NGS research. Main features of GN are listed in Table 2.

Fig. 1. Example interface of DerBrowser. E. coli cloning vectors pBR322 are used to generate display. **a.** Navigation to previous/next region. **b.** Navigation to up/down region. **c.** Scrollbar for browser data. **d.** STRIPE. **e.** OBJECTS **f.** OBJECT NAME. **g.** DISPLAY. Users can choose which stripes they want to display. **h.** Example popup menu with the stripe options. **i.** ABOUT. This provides more information about selected objects when connected to a database. **j.** NEW MAP. This enables users to select additional stripes and retrieve a new map. **k.** FIND. A search function for display objects (by objects' name). **l.** Zoom in/out. **m.** Semantic zooming bar for changing the scale and display mode of the objects. **n.** SELECTED OBJECT. The window shows the name of selected object. **o.** MOUSE OVER. Window shows the x-coordinates while moving the mouse along the chromosome. If mouse is moving over an object, the name of the object will be shown in this window. **p.** "?"-MENU. Applet customization function. User can select font, set scale unit and choose other misc options. **q.** CLONE-type stripe. **r.** LOCI-type stripe. **s.** PLOT-type stripe. **t.** EXON-INTRON-type stripe. **u.** SEQUENCE stripe. **v.** Genomic Coordinates. This indicates the physical position of the objects.

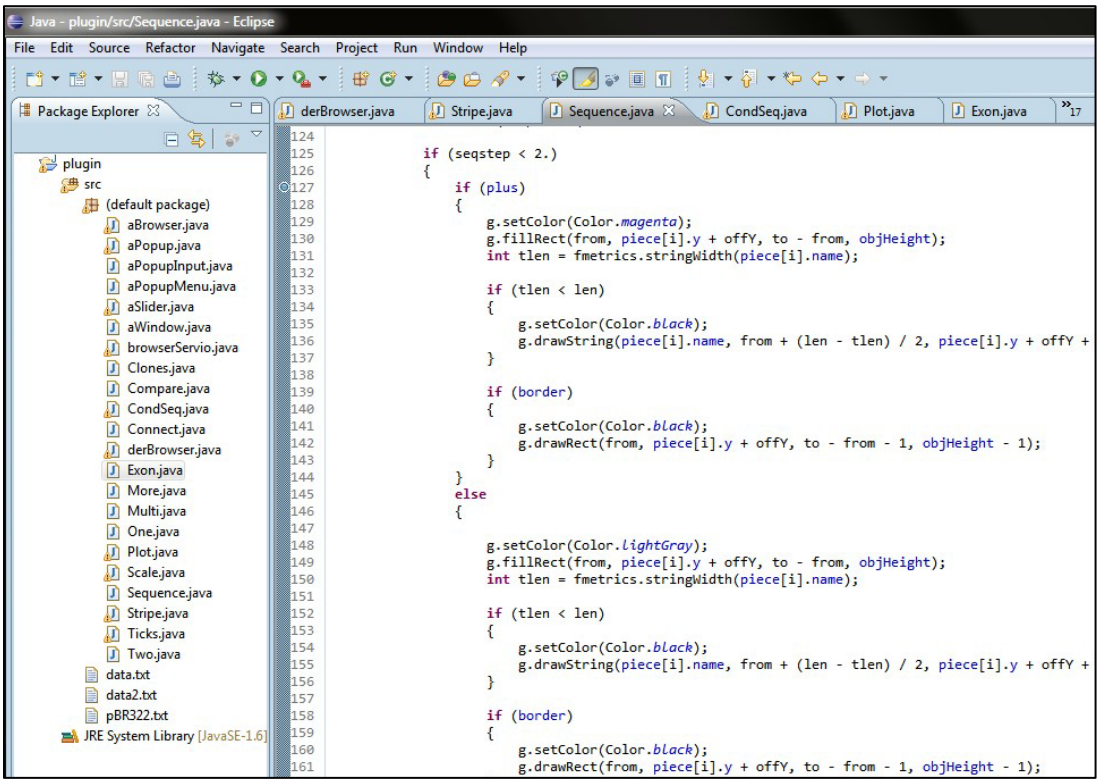

Fig. 2. Screenshot of Eclipse's workbench. On the left is the package explorer which listed all the Java classes for development.

```
public void draw(Graphics g, aWindow win, Font font, One selected)
{
    Graphics2D g2 = (Graphics2D) g;
    int offY = top - win.offY;
    if (offY > win.height - 1 || offY + height < 1)
        return;
    int baseY = height - 2 * objPad + offY;
    for (int i = 2; i < numObj; i++)
    {
        int from = (int) (win.scale * piece[i - 1].from) - win.offX, to = (int) (win.scale * piece[i].from)
                - win.offX;
        if (to <= 0 || from >= win.width)
            continue;
        g.setColor(piece[i - 1].color);
        if (hist)
        {
            if (piece[i - 1].y == piece[i].y)
            {
                int startY = piece[i].to < 0 ? (int) Math.ceil(piece[0].to
                        / scale) : piece[i - 1].y;
                int realHeight = piece[i].to < 0 ? (int) Math.ceil(-1
                        * piece[i].to / scale) : (int) Math
                        .ceil(piece[i].to / scale);
                g.fillRect(from, startY + offY, to - from, realHeight);
            }

            else
            {
                int startY = piece[i].to < 0 ? (int) Math.ceil(piece[0].to
                        / scale) : piece[i - 1].y;
                int realHeight = piece[i].to < 0 ? (int) Math.ceil(-1
                        * piece[i].to / scale) : (int) Math
                        .ceil(piece[i - 1].to / scale);
                g.fillRect(from, startY + offY, to - from, realHeight);
            }
        }
        else
        {
            g.drawLine(from, piece[i - 1].y + offY, to, piece[i].y + offY);
        }
    }
    g.setColor(piece[numObj - 1].color);
    g.fillRect(
            (int) (win.scale * piece[numObj - 1].from) - win.offX,
            (piece[numObj - 1].to < 0 ? (int) Math
                    .ceil(piece[0].to / scale) : piece[numObj - 1].y)
                    + offY,
            1 * (int) win.scale,
            (piece[numObj - 1].to < 0 ? (int) Math.ceil(-1
                    * piece[numObj].to / scale) : (int) Math
                    .ceil(piece[numObj - 1].to / scale)));
    g.setColor(axcol);
    g.drawLine(0, zeroY + offY, win.width, zeroY + offY);
    g.drawLine(0, offY, 0, baseY);
    g.setFont(font);
    FontMetrics fmetrics = g.getFontMetrics(font);
    int fh = 8;
    g.drawString("" + piece[0].to, 2, offY + fh);
    g.drawString("" + piece[0].from, 2, baseY);
}
```

Fig. 3. Code used in PLOT class for drawing plot graphs. Each sequencing read is treated as an individual object and has its own value. The “draw” method draws them one by one in the display window.

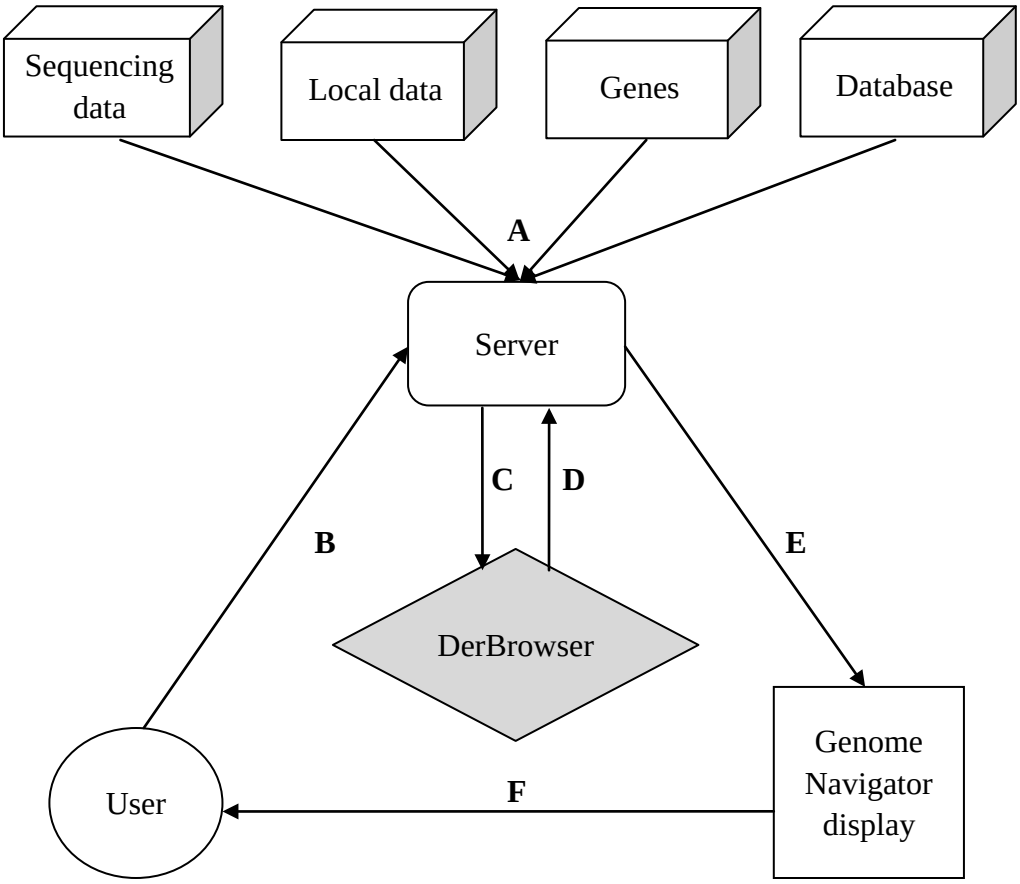

Fig. 4. A flow chart demonstrates a user calling for GN. (A). Data from different sources are sent to the web server and stored for the future use. (B). User accesses web interface and sends an inquiry with data specifications to the web server, which extracts and passes the information by CGI script. (C). CGI script reads stored data with user's input and processes the data stream into the format which can be interpreted by the applet. (D). Applet gets the data streams and informs the web server that it is ready to generate the display. (E). Server gets the input from the applet and generates a HTML page for the Genome Navigator display. (F). Output is visualized by the user in the browser window in HTML, together with the Java interface.

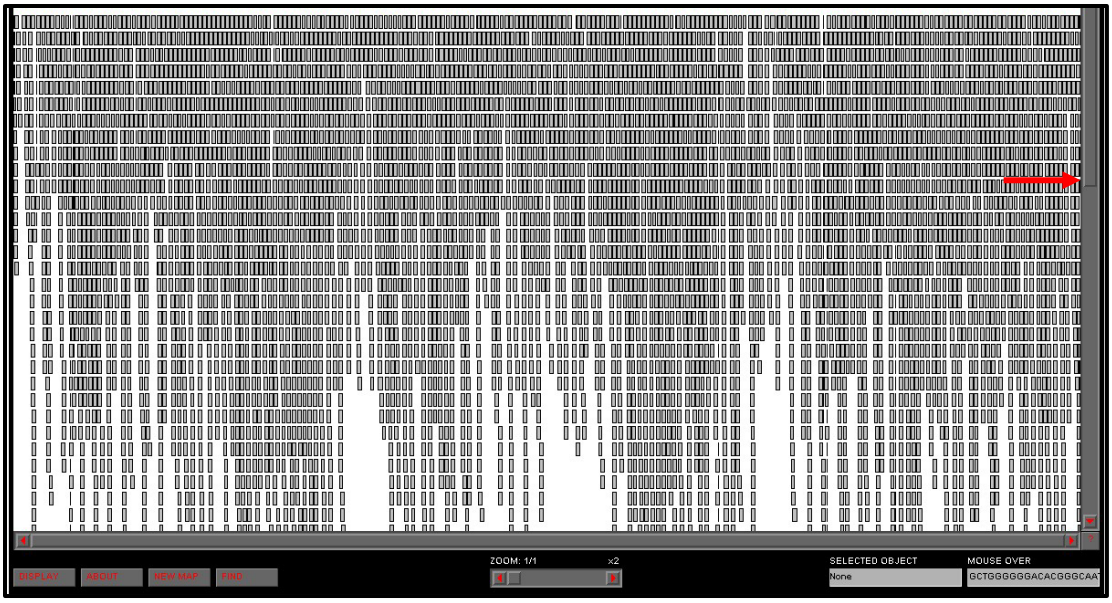

Fig. 5. Sequencing reads display at stacked view. Data are from supercont10.1 of *N. crassa* strain 2223 by Illumina technologies displaying at position [1,000-20,000]. The 19,001bp region contains 6,118 reads. Red arrow indicates the position of the scrollbar which is at about 1/3 of the stacked height.

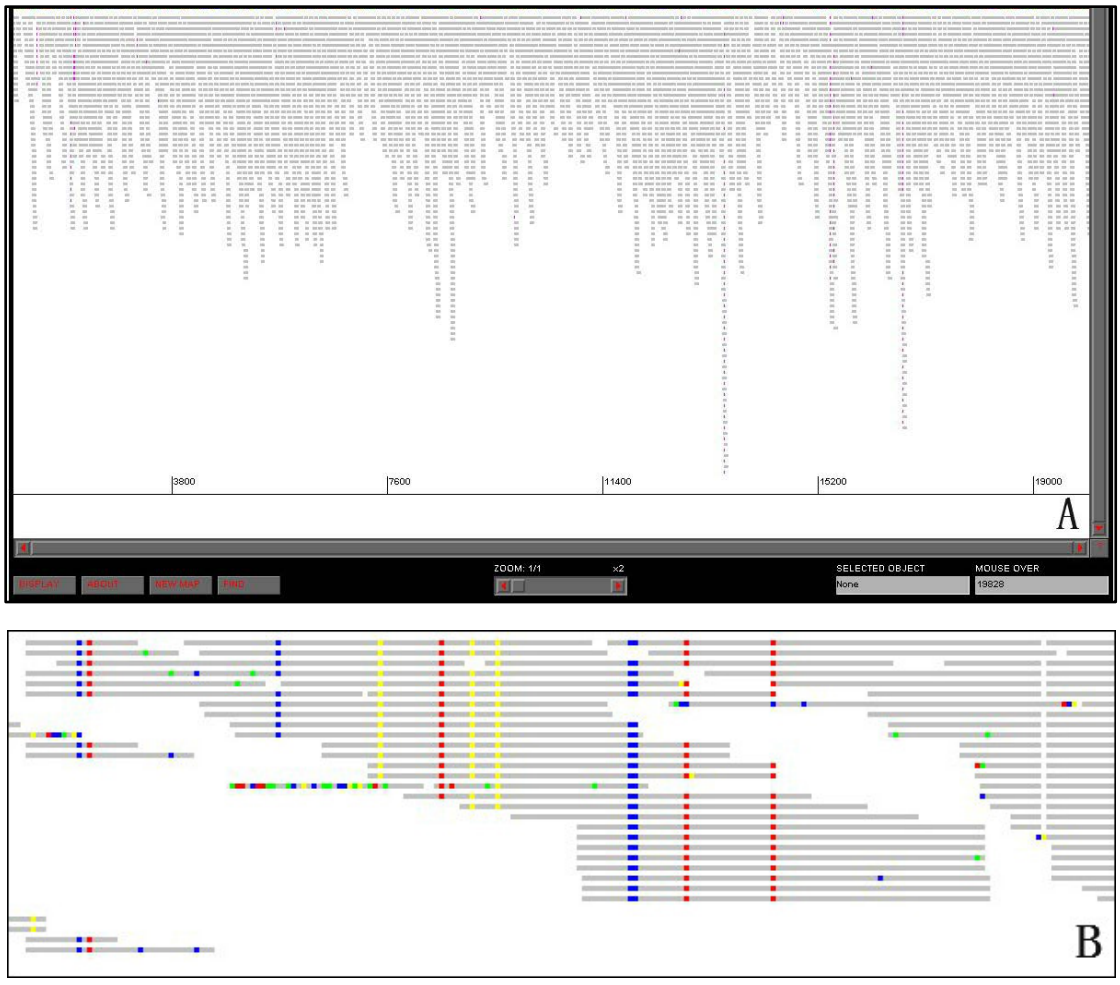

Fig. 6. Sequencing reads display at condensed view. (A) A screenshot of Supercont10.1 of *N. crassa* strain 2223 sequencing data display at position [1,000-20,000] in low zoom condensed view, same data as shown in Fig. 5. (B) High zoom view of the condensed SEQUENCE stripe, the mismatches are easily seen in highlight colors.

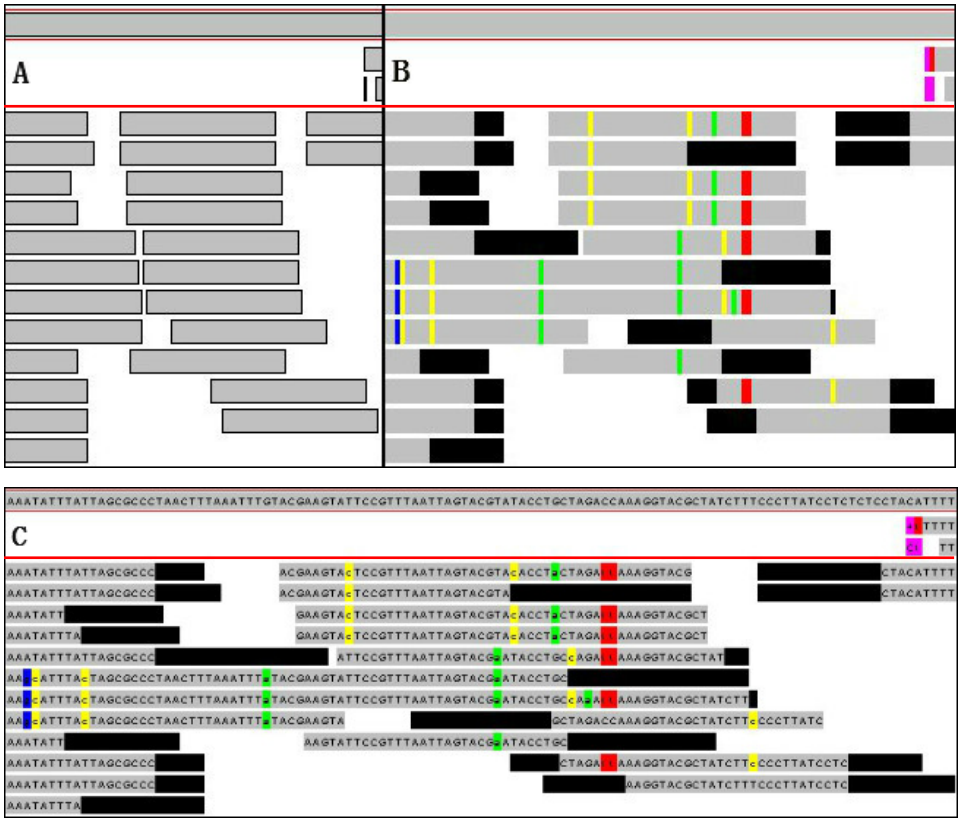

Fig. 7. Genomic changes detection at different zooming levels. Same data are from supercont10.1 of *N. crassa* strain 2223 generated by Illumina (above read line) and SOLiD technologies (below red line) displaying at position [1,050-1,163]. (A). Stacked view. The base pairs are not specified, no highlighted variations can be seen in highlight colors at this zooming level. (B), (C) Complex view. The selected object on top with a red rectangle is the reference sequence of *N. crassa* supercont10.1 strain 2223. All the discrepancies are shown in different colors and base pairs are visible at high zoom level. Color scheme is defined as following: A: Green, T: Red, C: Yellow, G: Blue, Deletion: White with "-", Insertion: Magenta with "+", Detailed insertion: Magenta with letters, Hard clipping: Black.

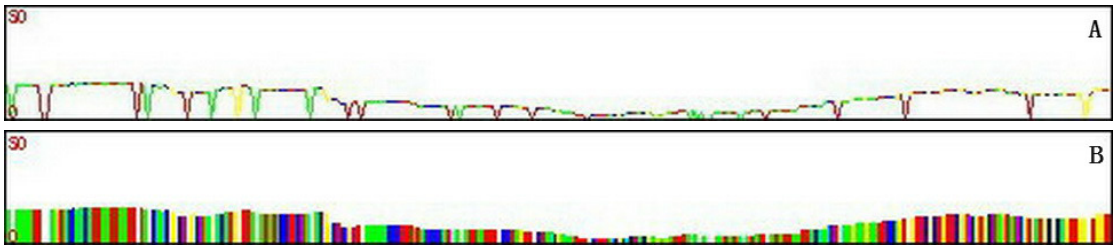

Fig. 8. Dual-mode of PLOT display. Data are from supercont10.1 of *N. crassa* strain 2223 using Illumina technologies. Different colors represent for 4 base pairs (A: Green, T: Red, C: Yellow, G: Blue). The height at certain position is the nucleotide density. (A). Line graph. (B). Histogram

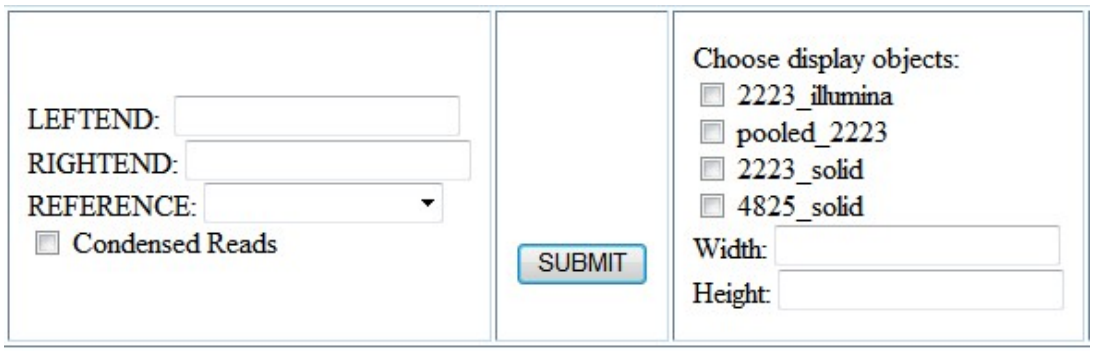

Fig. 9. User input interface. By clicking submit button, the user's inputs will be sent to the server for processing and generates proper data stream for the applet to interpret. Users can define the region of interest, the reference sequence, the combinations of display objects, the size of the applet and the mode of display.

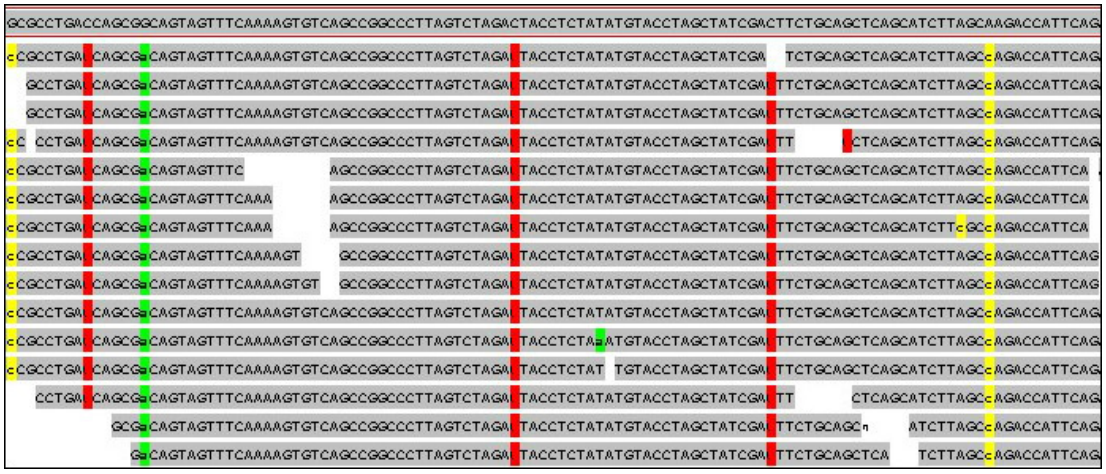

Fig. 10. Example of potential SNP detection. The selected object on top with a red rectangle is the reference sequence of *N. crassa* supercont10.1 strain 2223 at position of [13,734-13,848]. Six positions are showing consistent variations (C/G, T/C, A/G, T/C, T/C, C/A), which maybe potential SNP locations. Several random discrepancies are also shown in the graph, which may due to random sequencing errors.

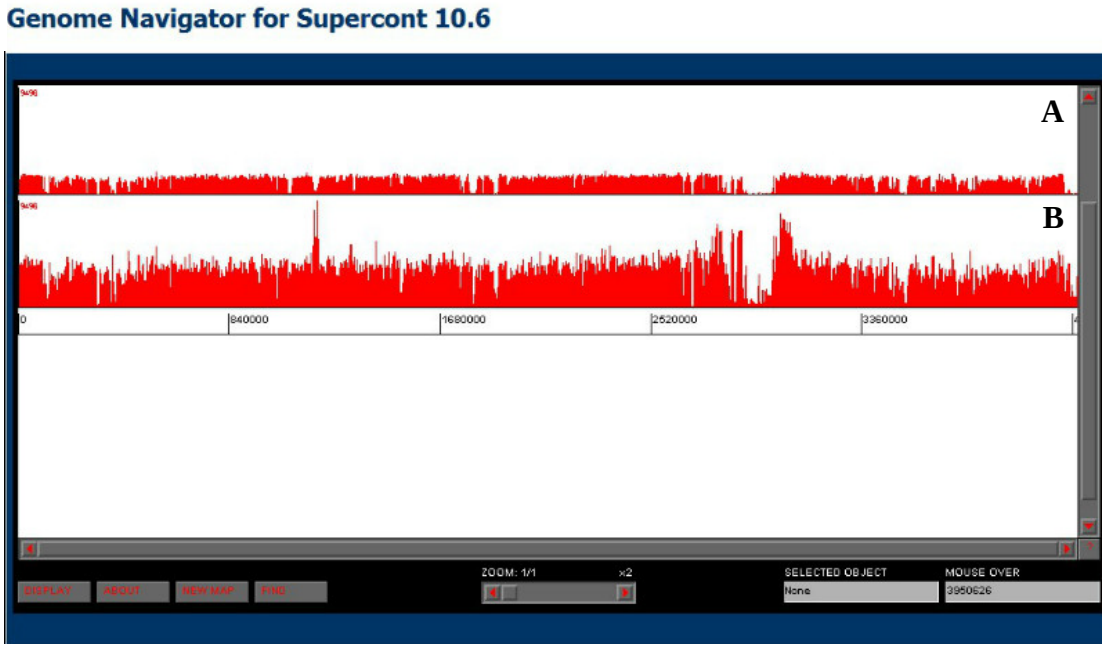

Fig. 11. Example of whole-chromosome coverage histogram. The strain used is *N. crassa* 2223. The graph is generated in a 5K interval. (A) Illumina sequencing method, (B) Solid sequencing method. The whole length for chromosome 6 is 4,218,251bp. The maximum value at a 5K window is 9,496 reads.

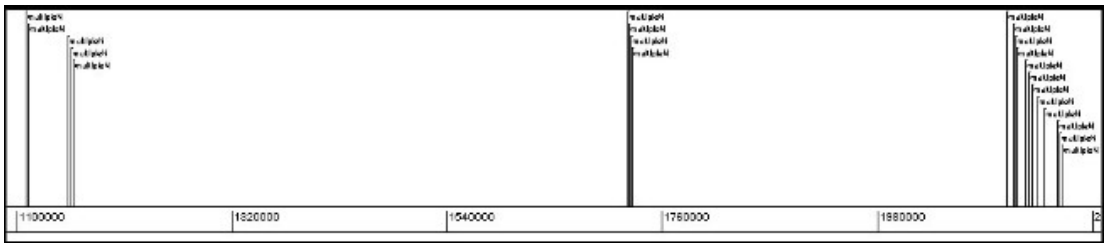

Fig. 12. Example of group data display. The strain used is *N. crassa* 2223. All the regions with consecutive N-nucleotides are picked with the positional information. Multiple-N segments are the unmapped gaps in the reference sequence.

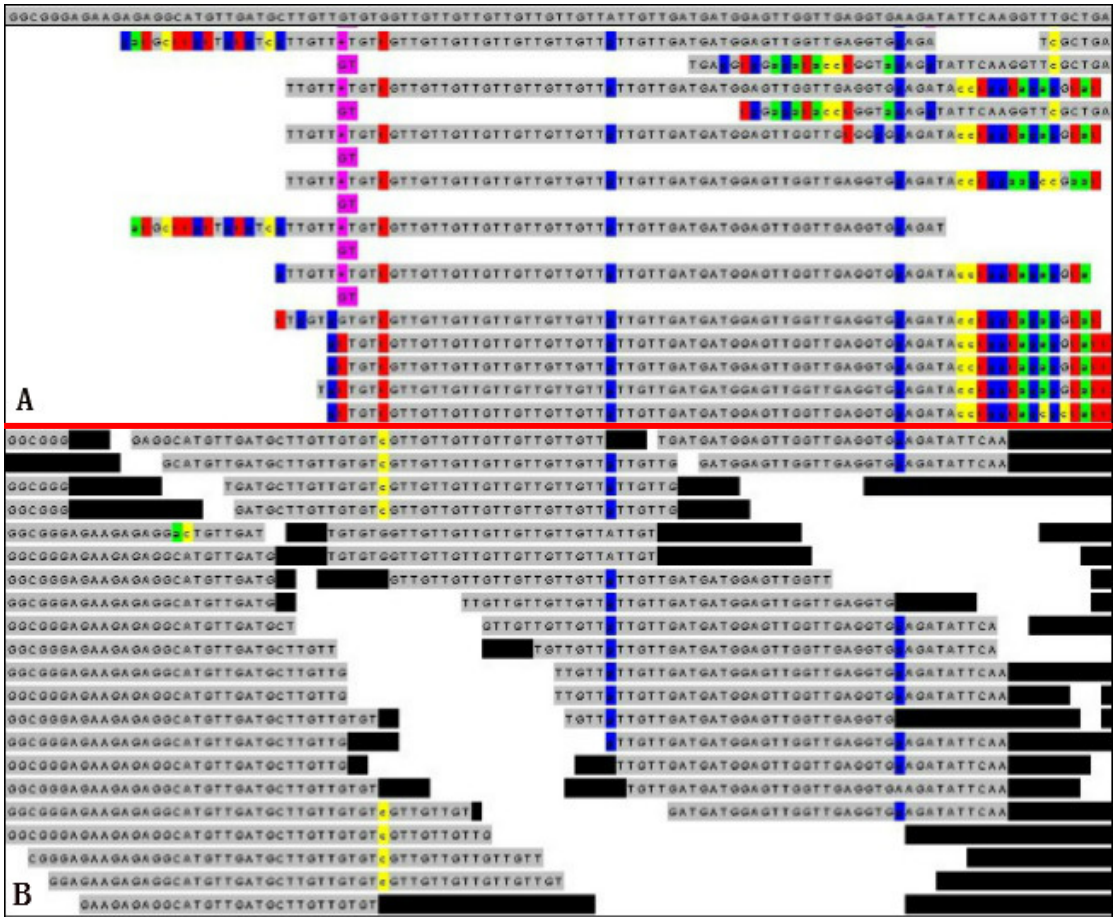

Fig. 13 Examples of sequencing discrepancies. Data are from supercont10.1 of *N. crassa* strain 2223 at region [1,968-2,078]. (A). Data are from Illumina technologies (B). Data are from SOLiD technologies. The first segment on top is the reference sequence. Red line indicates the boundary between two data sets.

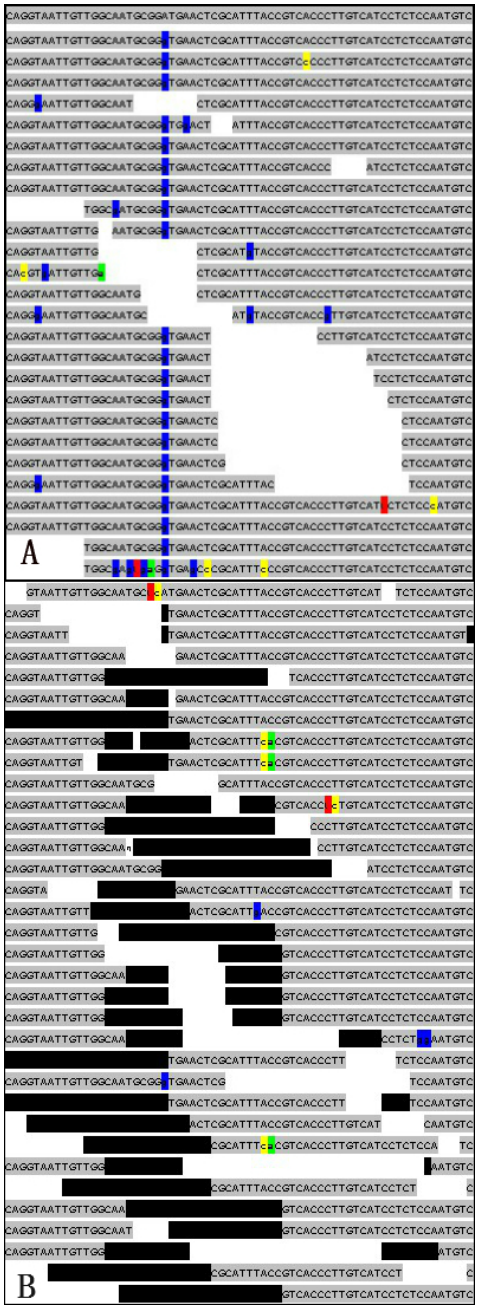

Fig. 14. Examples of ambiguous sequencing discrepancies. Data are from supercont10.1 of *N. crassa* strain 2223 at region [49,425-49,490]. (A). Illumina technologies. (B). SOLiD technologies. The first segment on top is the reference sequence.

Fig. 15. Examples of possible misalignments. Data are from supercont10.1 of *N. crassa* strain 2223 at region [2,374-2,457]. (A). Illumina technologies (B). SOLiD technologies.

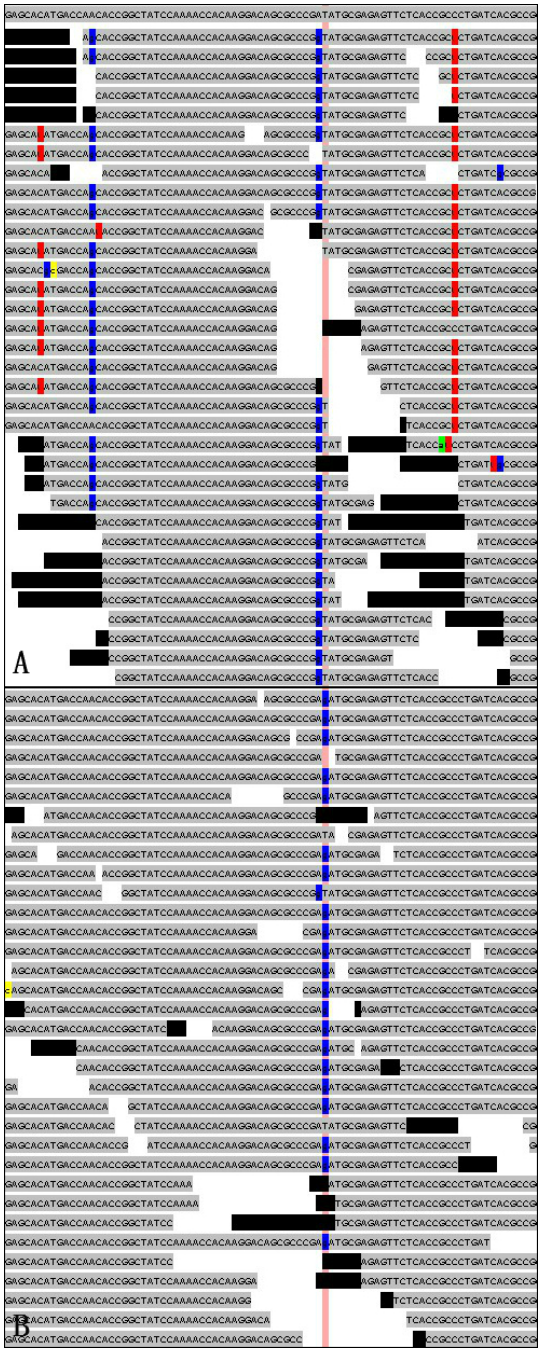

Fig. 16. Example of discrepancies between two *N. crassa* strains. Data are from supercont10.1 of *N. crassa* at region [5,893-5,961] using SOLiD technologies. (A). *N. crassa* strain 2223 (B). *N. crassa* strain 4825. The first segment on top is the reference sequence.

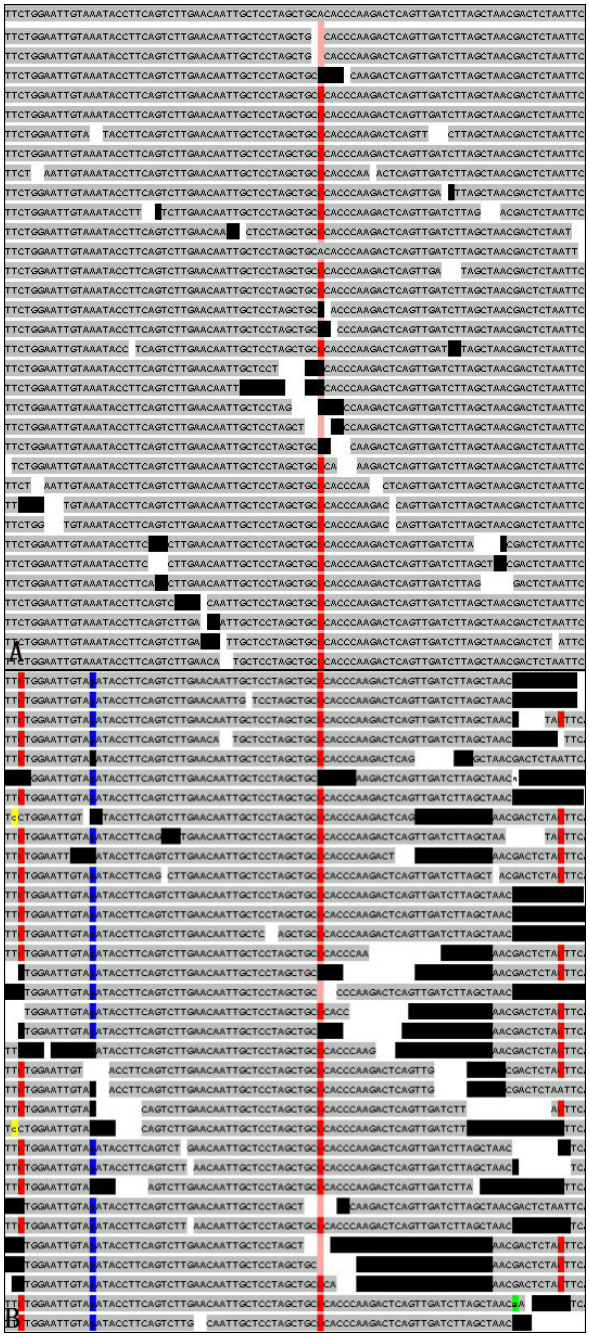

Fig. 17. Example of comparison between two *N. crassa* strains. Data are from supercont10.1 of *N. crassa*t region [6,626-6,714] using SOLiD technologies. (A). *N. crassa* strain 2223 (B). *N. crassa* strain 4825. The first segment on top is the reference sequence.

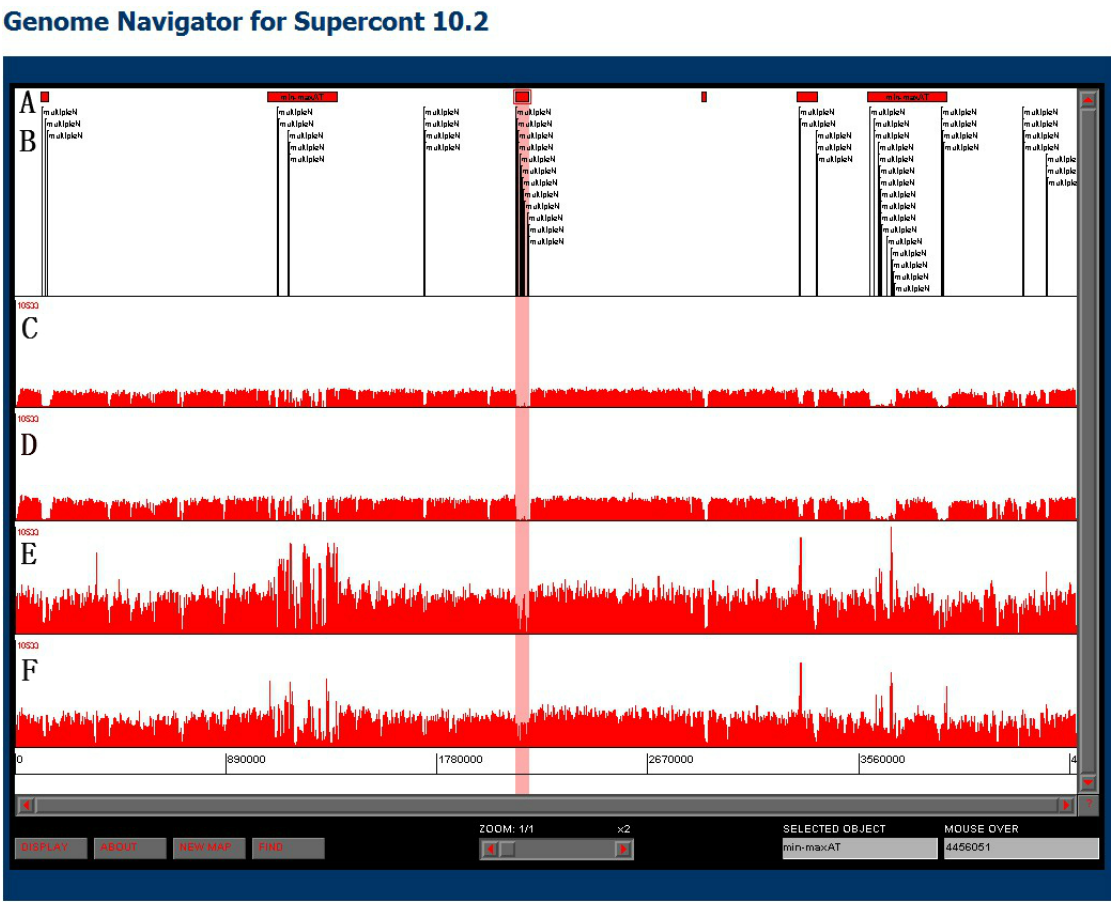

Fig. 18. Example of coverage histogram for Supercont10.2 of *N. crassa*. The graph is generated in a 5K interval. (A). The AT-rich region from supercont10.2 of *N. crassa* strain 2223 using SOLiD technologies. (B). the multiple-N region. (C). whole chromosome coverage of strain 2223 using Illumina technologies. (D). whole chromosome coverage of pooled strain 2223 using Illumina technologies. (E). whole chromosome coverage of strain 2223 using SOLiD technologies. (F). whole chromosome coverage of strain 4825 using SOLiD technologies.

Table 1. Comparison of leading DNA sequencing commercial platforms to date.

Company	Former company	Systems	Maximum read length	Typical Throughput	Run Time
Illumina	Solexa	HiSeq 2500/1500	2 × 100 bp	600 Gb/ 300 Gb[1]	~11 days/ ~8.5 days[1]
		HiSeq 2000/1000	2 × 100 bp	600 Gb/ 300 Gb	~11 days/ ~8.5 days
		Genome Analyzer IIx	2 × 150 bp	85 - 95 Gb	~ 14 days
		MiSeq	2 × 250 bp	6 - 7 Gb	>35 hours
Helicos	N/A	HeliScope	35 bp	21 - 35 Gb per run, 420 - 700 Mb per channel	>1 Gb per hour
Life Technologies	Ion Torrent	Ion PGM	200 bp[2]	~ 1 Gb[2]	4.5 hours[2]
Life Technologies	Applied Biosystems	SOLiD 4	2 x 50 bp[3]	100 GB and 1.4 billion tags per run[3]	8 - 9 days[3]
Pacific Biosciences	N/A	PACBIO RS	860–1100 bp	N/A	0.5–2 hours
Roche	454	GS FLX+	Up to 1,000 bp	700 Mb	23 hours
		GS FLX	Up to 600 bp	450 Mb	10 hours

Note: 1. Under Rapid Run mode, Illumina HiSeq 2500/1500 can generate 120 Gb / 60Gb output in ~27 hours. 2. Results are achieved by using Ion 318 Chip. 3. Results are achieved by using mate-paired libraries (insert sizes from 600 bp to 10 KB).

Table 2. Specifications of GN

Categories	Features
System support and requirement	32-bit and 64-bit versions of Windows, Linux, Mac OS X and any other platform supporting Java Virtual Machine. Java Runtime Environment
Developing Platforms	Implemented in Java
Support input format	FASTAQ*, FASTA*, SAM*, BAM*, TXT
Data sources	NGS data, Local files, Databases
Display mode	Stack view (reads are piled up along Y-axis) Condensed view (reads are fixed to 2-pixel height) Complex view with semantic zooming (reads with variations are shown in highlighted colors)
Plot mode	Line graph, Histogram
Statistics plots	Coverage plot (coverage of certain region of given interval) Distribution plot (base pair density at given region)
Group data distribution	LOCI stripe with pinpoint indication of data with same type.
Directionality indication	EXON-INTRON stripe with directional information
Base pair discrepancy indication	Indication of genetic variations at nucleotide resolution
User interaction	Users are able to customize input and choose the display options
Highlighted colors	Mismatches, chosen stripe, chosen object and the corresponding position along Y-axis, chosen nucleotide and the corresponding position along Y-axis
Right-click behavior	Choose certain object by right-clicking Choose certain nucleotide at high zoom level by right-clicking

Note: * These formats need to be pre-processed by format converting tools before use.

REFERENCES

Altshuler, D., Durbin, R. M., Abecasis, G.R. (2010). A map of human genome variation from population-scale sequencing. *Nature* 467:1061–73.

Boulos, M. N. K. (2003). The use of interactive graphical maps for browsing medical/health Internet information resources. *International Journal of Health Geographics* 2(1).

Daines, B., Wang, H., Wang, L., Li, Y., Han, Y., Emmert, D., Gelbart, W., Wang, X., Li, W., Gibbs, R. and Chen, R. (2011). The *Drosophila melanogaster* transcriptome by paired-end RNA sequencing. *Genome Res.* 21:315–24.

Darling, A., Mau, B., Blattner, F.R. and Perna, N.T. (2004). Mauve: Multiple alignment of conserved genomic sequence with rearrangements. *Genome Res.* 14(7):1394-1403.

Engels, R., Yu, T., Burge, C., Mesirov, J.P., DeCaprio, D. and Galagan, J.E. (2006). Combo: a whole genome comparative browser. *Bioinformatics* 22(4):1782-1783.

Gans, J.D. and Wolinsky, M. (2007). Genomorama: genome visualization and analysis. *BMC Bioinformatics*. 14;8:204.

Gordon, D., Abajian, C., and Green, P. (1998). Consed: A graphical tool for sequence finishing. *Genome Res.* 8: 195–202.

Grigoriev, A. (1998). Reusable graphical interface to genome information resources. *In proc. of PSB conference*.

Grigoriev, A. (1998). Microweb: Genome Navigator. *Trends in Microbiology*, 6, 184.

Grigoriev, A. (1997) Genomes with a view. *Trends in Genetics*, 13, 499.

Hu, Z., Frith, M., Niu, T. and Weng, Z. (2003). SeqVISTA: a graphical tool forsequence feature visualization and comparison. *BMC Bioinformatics*, 4(1).

Illumina, Inc. (2010). *De novo* assembly using illumina reads.

Illumina, Inc. (2012). HiSeq 2500 application note.

Illumina, Inc. (2012). Real science real performance.

Leader, D.P., (2004). BugView: a browser for comparing genomes. *Bioinformatics* 20(1):129-30.

Lewis, S.E., Searle, S.M.J., Harris, N., Gibson, M., Iyer, V., Richter, J., Wiel, C., Bayraktaroglu, L., Birney, E., Crosby, M.A., Kaminker, J.S., Matthews, B.B., Prochnik, S.E., Smithy, C.D., Tupy, J.L., Rubin, G.M., Misra, S., Mungall, C.J. and Clamp ME (2002). Apollo: a sequence annotation editor. *Genome Biology* 3(12):1-14.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R. and 1000 Genome Project Data Processing Subgroup. (2009) The Sequence alignment/map (SAM) format and SAMtools. *Bioinformatics* 25, 2078-9.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 15;25(14):1754-60.

Life Technologies, Inc. (2012). PGM™ for genes. Proton™ for genomes.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., PYu, P., Begley, R. F. and Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 437, 376–380.

Maxam, A.M. and Gilbert, W. (1977). A new method for sequencing DNA. *Proc. Natl. Acad. Sci.* 74, 560–564.

Miller, J. R., Koren, K. and Sutton, G. (2010). Assembly algorithms for next-generation sequencing data. *Genomics* 95, 315–327.

Montgomery, S.B., Astakhova, T., Bilenky, M., Birney, E., Fu, T., Hassel, M., Melsopp, C., Rak, M., Robertson, A.G., Sleumer, M., Siddiqui, A.S. and Jones S.J.M. (2004). Sockeye: A 3D environment for comparative genomics. *Genome Res.* 14(5):956-962.

Nix, D.A. and Eisen, M.B. (2005). GATA: a graphic alignment tool for comparative esequence analysis. *BMC Bioinformatics* 6(9).

Robinson, J. T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E. S., Getz, G. and Mesirov, J. P. (2011) Integrative genomics viewer. *Nature Biotechnology* 29, 24–26.

Rutherford, K., Parkhill, J., Crook, J., Horsnell, T., Rice, P., Rajandream, M.A. and Barrell, B. (2000). Artemis: sequence visualization and annotation. *Bioinformatics* 16(10):944-945.

Sanger, F., Air G. M., Barrell, B. G., Brown, N. L., Coulson, A. R., Fiddes, J. C., Hutchison, C. A., Slocombe, P. M., and Smith, M. (1977). Nucleotide sequence of bacteriophage φX174 DNA. *Nature* 265, 687 – 695.

Shendure, J., M., Porreca, G. J., Reppas, N. B., Lin X., McCutcheon, J. P., Rosenbaum, A. M., Wang, M. D., Zhang, K., Mitra, R. D. and Church, G. M. (2005). Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* 309, 1728–1732.

Stein, L. D., Mungall, C., Shu, S., Caudy, M., Mangone, M., Day, A., Nickerson, E., Stajich, J. E., Harris, T. W., Arva, A. and Lewis, S. (2002). The generic genome browser: a building block for a model organism system database. *Genome Res.* 12(10):1599-610.

Stothard, P. and Wishart, D.S. (2005). Circular genome visualization and exploration using CGView. Bioinformatics 21(4):537-539.

Turinsky, A.L., Ah-Seng, A.C., Gordon, P.M.K., Stromer, J.N., Taschuk, M.L., Xu, E.W. and Sensen, C.W. (2005). Bioinformatics visualization and integration with open standards: The Bluejay genomic browser. *In Silico Biology* 5(2):187-98.

Vernikos, G., Gkogkas, C., Promponas, V., Hamodrakas, S. (2003) GeneViTo: Visualizing gene-product functional and structural features in genomic datasets. *BMC Bioinformatics* 4(1):53.