LookSeq: A browser-based viewer for deep sequencing data

Heinrich Magnus Manske and Dominic P. Kwiatkowski

Genome Res. 2009 19: 2125-2132 originally published online August 13, 2009
Access the most recent version at doi:10.1101/gr.093443.109

LookSeq: A browser-based viewer for deep sequencing data

Heinrich Magnus Manske[1] and Dominic P. Kwiatkowski

Wellcome Trust Sanger Institute, Hinxton, Cambridge CB10 1SA, United Kingdom

Sequencing a genome to great depth can be highly informative about heterogeneity within an individual or a population. Here we address the problem of how to visualize the multiple layers of information contained in deep sequencing data. We propose an interactive AJAX-based web viewer for browsing large data sets of aligned sequence reads. By enabling seamless browsing and fast zooming, the LookSeq program assists the user to assimilate information at different levels of resolution, from an overview of a genomic region to fine details such as heterogeneity within the sample. A specific problem, particularly if the sample is heterogeneous, is how to depict information about structural variation. LookSeq provides a simple graphical representation of paired sequence reads that is more revealing about potential insertions and deletions than are conventional methods.

[Supplemental material is available online at http://www.genome.org. LookSeq is freely available at http://lookseq.sourceforge.net.]

New technologies for massively parallel DNA sequencing allow a genome to be sequenced to great depth, i.e., with many sequence reads covering each nucleotide position (Margulies et al. 2005; Shendure et al. 2005; Bentley et al. 2008; Branton et al. 2008; Campbell et al. 2008a; Harris et al. 2008; Hillier et al. 2008; Shendure and Ji 2008). Deep sequencing data can be valuable for many different purposes. It has been estimated that approximately 30-fold depth of paired 35-base reads is needed to discover 99% of true variants in non-repeat regions of the genome of a diploid individual (Bentley et al. 2008; H Li et al. 2008). In excess of 100-fold coverage may be needed for assembling short sequence reads in highly variable regions of the genome. And the possibility of sequencing to a much greater depth, e.g., more than 10^4-fold, creates unprecedented opportunities to investigate the genetic basis of heterogeneity within individual biological samples. This has many potential clinical and biological applications, e.g., to analyze viral mutation rates (Harris et al. 2008) or to investigate the genetic driving forces that determine the evolution of a cancer cell population within an individual patient (Campbell et al. 2008b).

The first stage in analysis of deep sequencing data is to align sequence reads against a reference genome with sufficient confidence to identify true variants. This can be an exceptionally complex analytical problem, particularly when attempting to align short sequence reads of imperfect quality to a highly variable or repetitive genome. A good example of how this problem may be addressed is the popular MAQ software (H Li et al. 2008), which assigns to each genotype call an error probability that is based on a number of factors, including raw sequence quality and mapping quality, a measure of the confidence that individual reads have been mapped to the correct location in the genome (H Li et al. 2008). The development and optimization of alignment algorithms for short sequence reads is currently a very active area of research (Bentley et al. 2008; Hillier et al. 2008; H Li et al. 2008; R Li et al. 2008; Lin et al. 2008; Schatz et al. 2007b; Smith et al. 2008; Langmead et al. 2009). Many algorithms allow the user to define a number of parameters that effectively alter the stringency of alignment or filter the output to achieve the optimal trade-off between false positive and false negative results. Thus a common problem confronting an investigator is to visually inspect the data to compare the alignments and variant calls made by different algorithms or by a specific algorithm at different parameter settings. If sequencing has been performed to great depth, and particularly if the biological sample is heterogeneous, then the process of data visualization is nontrivial.

Many of the available tools for visualizing sequence read alignments derive from pioneering work on genome sequence assembly based on capillary sequencing data (Dear and Staden 1991; Bonfield et al. 1995; Gordon et al. 1998; Schatz et al. 2007a). This approach has been usefully extended for next-generation sequencing, e.g., in EagleView, a client-installed software that can handle a large volume of sequence reads and that allows assemblies to be constructed from multiple technology platforms (Huang and Marth 2008).

Here, we explore a different approach to the visualization of deep sequencing data. We address a general problem inherent in very large data sets, i.e., that too detailed a view tends to drown the user in data, whereas too condensed a view may lose important details in the abundance of data. Specifically, we address the problem that a simple pile up of sequence alignments is a useful way of displaying the detail of a conventional genome assembly, but it is impractical for deep sequencing data as only a proportion of reads can be viewed at a time; whereas on the other hand, a collapsed view that summarizes information across all reads might obscure potentially important details such as heterogeneity, outliers, and haplotypic relationships. Our proposed solution aims to make it as easy as possible for an investigator to browse across a large genomic region and to zoom in to inspect interesting features at any desired level of resolution.

Results

Viewing deep sequencing data

Our goal was to enable the user to browse seamlessly along the genome and to zoom effortlessly in and out of a very large set of sequence reads, thus assimilating fine details while maintaining

[1] **Corresponding author.**
E-mail mm6@sanger.ac.uk; fax 44-1223-4919.

Article published online before print. Article and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.093443.109. Freely available online through the *Genome Research* Open Access option.

a global perspective. To achieve this, we developed an interactive web viewer with similar intuitive features to Google Maps (http://maps.google.com), albeit on a much simpler, one-dimensional scale. The display is managed by an AJAX-based web page, while the graphics are rendered server-side, by on-the-fly calculations from an alignment database. Starting from the level of a whole chromosome (Fig. 1A), a user can zoom into a region of interest (Fig. 1B) and proceed to a detailed view of sequence reads at the level of individual bases (Fig. 1C). This is done by clicking the mouse or by selecting a predefined level of zoom. Once the desired level of magnification has been reached, the user can navigate around the region by simply dragging the display. Alternatively, the user can navigate directly to a point of interest by entering the genomic coordinates or searching the sequence annotation.

LookSeq has the potential to be tailored for different user requirements, some of which have been implemented in the demonstration version (Supplemental figures; http://lookseq.sf.net). These include switches to show or hide perfect matches with the reference genome or known polymorphisms. Novel ways of viewing read pair data are described in more detail below. There are options to view genome coverage, GC content, and annotations to the reference sequence, which can include relevant external links such as PubMed (Supplemental Fig. 1). Double-clicking on a sequence read at maximum zoom can take the user to the corresponding coordinate on a reference genome database such as Ensembl or PlasmoDB. This behavior can be adapted to other web services through a JavaScript function.

Using multiple approaches to capture genome variation

For a highly polymorphic genome, there are considerable statistical challenges in constructing the genome sequence of an individual by sequencing random fragments and mapping these onto a reference genome. These challenges increase if the sequenced fragments are short, as is the case for the first generation of new sequencing technologies, e.g., about 35–70 bp for an Illumina Genome Analyzer compared with more than 500 bp for conventional Sanger sequencing. With short read lengths, it is difficult to align highly variable sequences to the reference genome, particularly if there are many repetitive and nonunique sequences or if there is biological heterogeneity within an individual sample.

LookSeq provides a graphical way of comparing putative genome variations identified by different statistical methods on the same set of sequence reads. For example, the MAQ program aligns short sequence reads to the reference genome, allowing for a small number of mismatches, assigning a mapping quality score that indicates the probability that the true alignment is not the one found by MAQ. By varying the mapping quality threshold, the user attempts to optimize the discovery rate of authentic variants while minimizing the false-positive rate. Using LookSeq, the user can compare the results produced by different settings of the MAQ program, e.g., with different mapping quality thresholds, or can compare the outputs of a range of different algorithms (Supplemental Fig. 2a). In this example, the different fragment size ranges used by the algorithms, and the tolerance of MAQ for mismatches, can be observed. Such comparisons of different alignment algorithms run on the same set of sequencing data may alert the user to systematic errors and biases with a particular method and may help to consolidate the evidence for a true variant, e.g., one that is supported by different algorithms.

To facilitate comparisons between different alignment algorithms, sequencing technologies, and samples, LookSeq can display a secondary data track. This track is a duplication of the primary display and will zoom, pan, and change display mode accordingly. However, when changing the data source for the primary track, the secondary track will keep the data source it was created with. This way, the user can directly compare two different data sources in the same display. Supplemental Figure 2a uses this feature. In this mode, the auxiliary coverage track will show a comparison of the coverage of the two samples (Supplemental Fig. 2b).

LookSeq allows the user to configure screen displays that compare or combine different sets of sequencing data, either on the same sample or on different samples (Supplemental Fig. 3). Graphical comparisons between data sets can highlight both true biological variations and errors in the sequencing process, while combining data sets can highlight subtle or complex variations by increasing coverage. In many cases, true variation can be distinguished from errors by the consistency of the variation in question within the sample. Supplemental Figure 4 shows an example of a likely "true" single nucleotide polymorphism (SNP) variant (not experimentally verified) and a putative sequencing error. The user can immediately ascertain that the visible region is well and evenly covered and is generally conserved in the sample. In this context, at location a, the variation (reference mismatch, in red) is supported by the vast majority of the reads, making it a believable SNP, whereas at location b, only two reads show the variation, while the majority of the reads displays the reference allele, indicating a sequencing error.

It can also be helpful to compare short-read sequencing data with long-read capillary sequencing data on the same samples, as the short-read data provide great depth of coverage while the long-read data provide supportive evidence for alignment accuracy (Supplemental Fig. 5). Two-base data are not natively supported in the current version of LookSeq but can be viewed when converted to normal DNA sequence.

To enable these on-screen comparisons and combinations, LookSeq uses a universal database to store alignments from different sequencing technologies and algorithms. It lists all available data sets next to the display, and allows switching between them. When a user is viewing a particular chromosomal region for one data set and then switches to another data set, all the viewing options remain constant, allowing the user to make a direct comparison of the same chromosomal region on different samples or with different alignment algorithms.

Alternatively, LookSeq may be switched to a multisource mode in which multiple data sources are brought together on screen. This process is independent of the origin of the data so the outputs of multiple sequencing runs, different alignment algorithms, and different sequencing technologies can be viewed, navigated, and zoomed in a single display.

Using information from read pairs

Read pair data provide a rich source of information that is only partially captured by conventional sequence alignment viewers. A read pair, or paired end read, refers to two sequence reads obtained at opposite ends of a single fragment of DNA. This is an optional feature of several sequencing technologies. Here we show read pairs generated by shearing DNA into fragments of approximately the same length (e.g., 200 ± 50 nucleotides) and then sequencing about 35 nucleotides at each end, i.e., a total of 70 nucleotides per

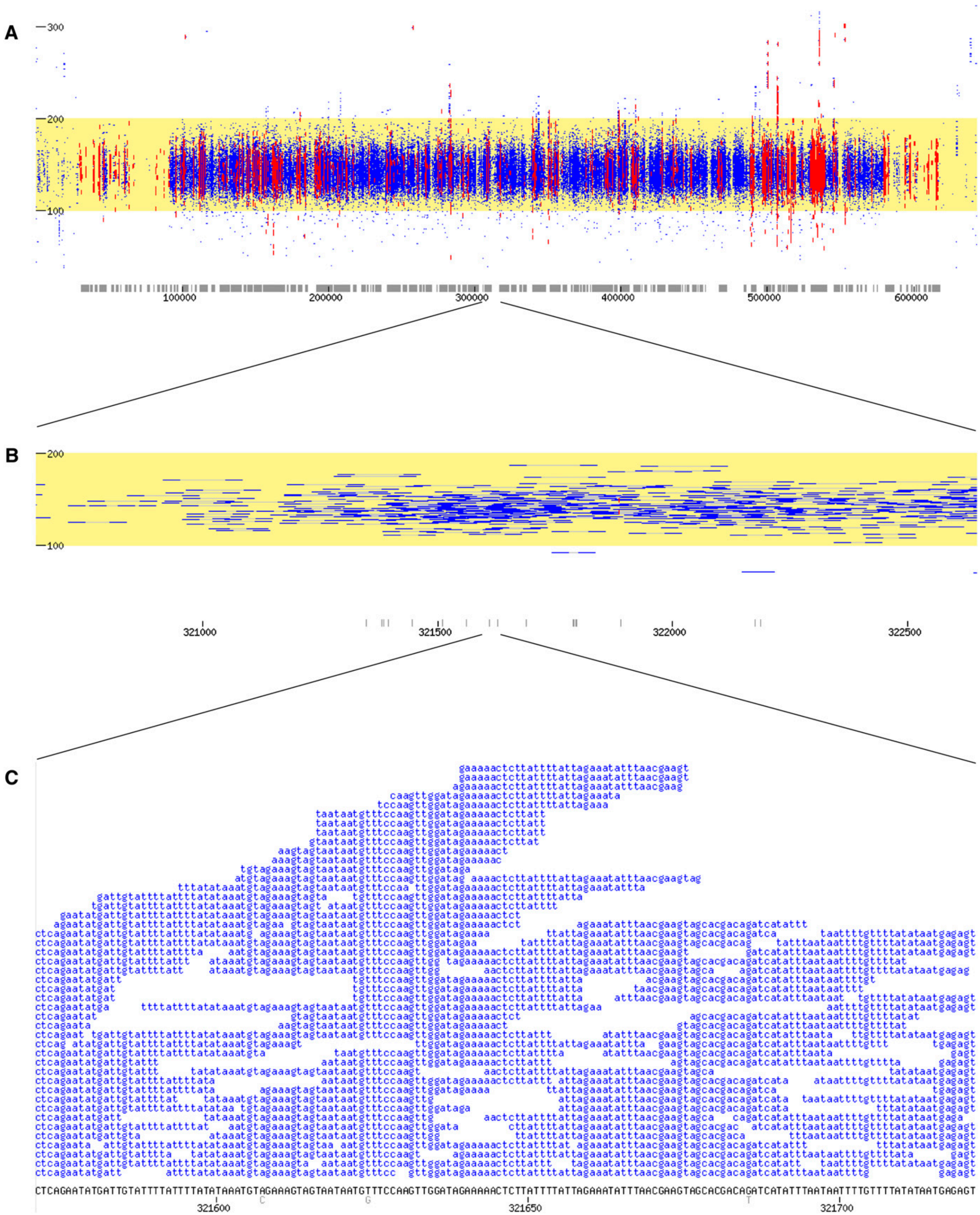

Figure 1. Different zooms/views. (*A*) InDel view, whole chromosome. (*B*) InDel view, 2-kbp region. (*C*) Pile-up view, single-base resolution. Perfect matching bases are in blue; mismatches (potential SNPs) are in red, and range of expected fragment size is in yellow.

fragment, using an Illumina Genome Analyzer. Read pair data can be useful in several ways. Genomic regions with sequence inversions may be identified by discrepancies in the orientation of the two paired ends. Genomic regions with insertions or deletions may be identified by discrepancies between the expected length of the DNA fragment and its apparent length when the read pairs are mapped onto the reference genome.

LookSeq allows the user to view read pair data in both a conventional and a novel way. The conventional approach organizes the display of read pairs based solely on their alignment to the reference genome. The novel approach provides extra information by plotting read pair data in two dimensions (Fig. 2), using the mapping distance of the read pairs on one axis to detect variation. The horizontal axis shows the position where a read pair maps onto the reference genome. The vertical axis shows the apparent size of the DNA fragment based on where the two ends of the read pair map onto the reference genome. Consider read pair data for a test sample that has some deletions and some insertions with respect to a reference genome. If a read pair spans a deletion, then the two ends will map onto the reference genome further apart than expected; i.e., the DNA fragment size will appear to be larger than it really is (Supplemental Fig. 6a). Conversely if a read pair spans an insertion, then when the two ends are mapped onto the reference genome, the DNA fragment size will appear shorter than it really is (Supplemental Fig. 6b).

The LookSeq read pair view provides a rapid way of assimilating different lines of evidence for a structural variant. The features of a simple deletion are illustrated in Figure 3. There is a discrete gap in coverage when the test sample is mapped onto the reference genome. Spanning the gap is a stack of paired reads whose apparent fragment size is greater than for the sample as a whole. The apparent increase in fragment size corresponds to the size of the deletion; i.e., it should be equal to the gap in coverage. Figure 4 illustrates a more complex situation where the test sample is heterogeneous (this is a cultured line of malaria parasites with a small genomic deletion in some individuals but not others). The physical location of the deletion is shown by a distinct gap in coverage with respect to the reference genome, but some read pairs extend into the gap. This distribution of fragment lengths is consistent with two subpopulations of parasites, one with the deletion and one without. The deletion site appears to be slightly different in the two technologies. However, zooming in on the deletion site and applying pileup view, it becomes clear that the site in question is a repeat. We suspect the capillary alignment algorithm had a choice between different chunks of the repeat to call “deleted” and chose the wrong one.

Read pair data are also useful for investigating major sequence differences between the test sample and the reference genome, e.g., large insertions or regions of intense polymorphism. To highlight the boundaries of such regions, LookSeq provides a specific view of lone matching reads, where one half of a read pair maps onto the reference genome but the other half does not (Supplemental Fig. 7). Clusters of half-mapping reads indicate complex variation within the fragment size of the read pairs. These clusters should ideally appear on either side of the variation.

Performance

In this version of LookSeq, server-side image generation requires less than a second, and just a few seconds for rendering of megabase-sized regions with hundreds of thousands of reads. Rendering time will increase roughly linearly when merging multiple data

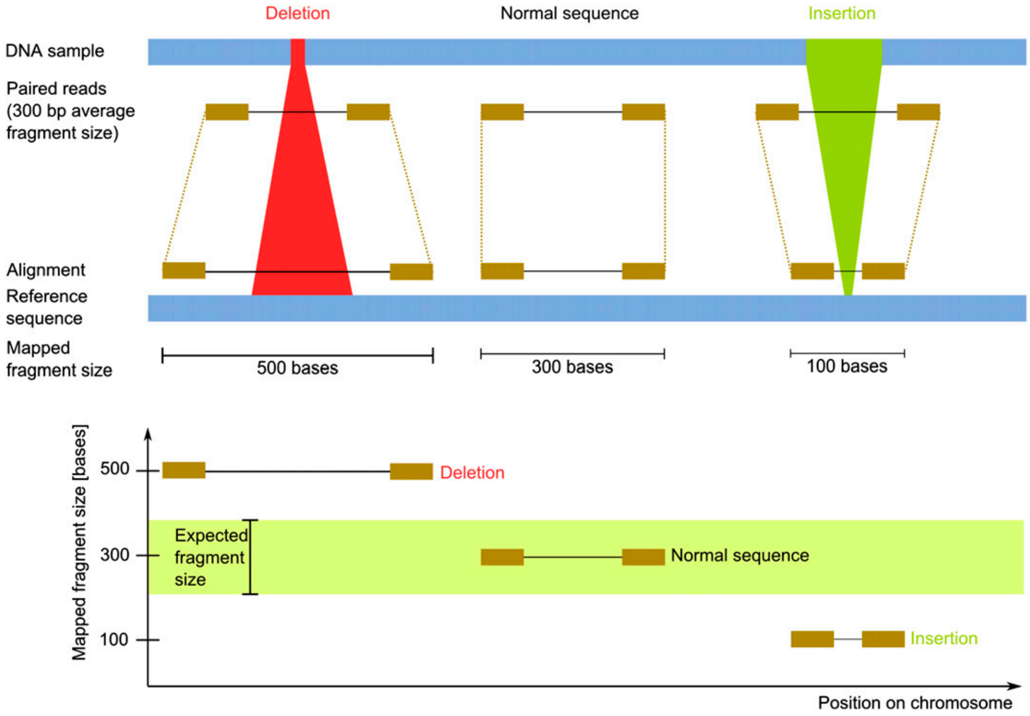

Figure 2. A DNA sequence containing a deletion and an insertion is sequenced using Illumina paired reads with 300 base fragments. The read pairs covering the deletion will map farther apart than expected, whereas the read pair covering the insertion maps closer together. Both variants will show up in the mapped fragment size plot, whereas the unaltered read pair falls into the expected fragment size range.

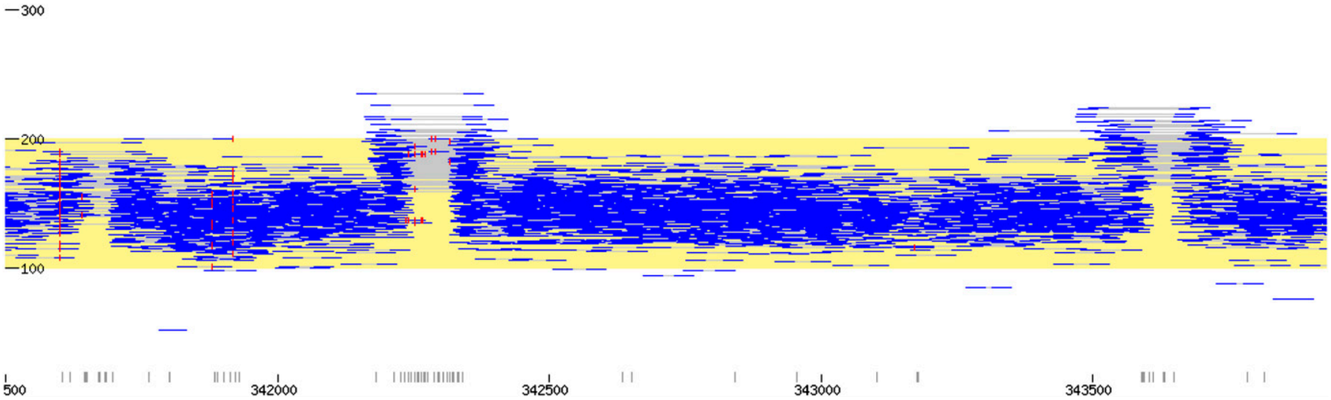

Figure 3. Read pair view of an ~2-kbp region. Chromosomal position on horizontal axis; pair alignment distance on vertical axis. Reads are in blue; read pairs are connected by gray lines. Expected fragment size is highlighted in yellow. Mismatches to the reference (possible SNPs) are marked in red.

sources into one display; however, even a seven-lane display usually renders in less than 3 sec. The current version of LookSeq uses Perl on the server side, for portability between server installations as well as compatibility with a range of other bioinformatics solutions. The reads to be displayed are read into memory from the respective data sources, grouped by type (perfect match, containing mismatches, single reads, inversions, etc.), and rendered in a specific order, so that e.g., mismatches (SNPs) are always rendered last, as they contain important information and should not be "overpainted" by other, less significant data. Possible performance improvements and reduced memory consumption could be achieved by rendering reads on the fly to separate image layers, which are subsequently combined to form a single image. This approach is being explored at the moment. Future versions of the rendering algorithm could also be augmented or replaced for performance reasons using other languages like Java or C/C++.

LookSeq response times appear to compare favorably with those of traditional web-based genome browsers, despite the additional amount of sequence alignment data. Taking the PlasmoDB site as an example of a gbrowse-based installation (Stein et al. 2002), we found that page reload typically took in the region of 4 sec when querying random displays of genome annotation for a 30-kb region. Browsing similarly sized regions in LookSeq, showing paired read alignments in addition to genome annotation

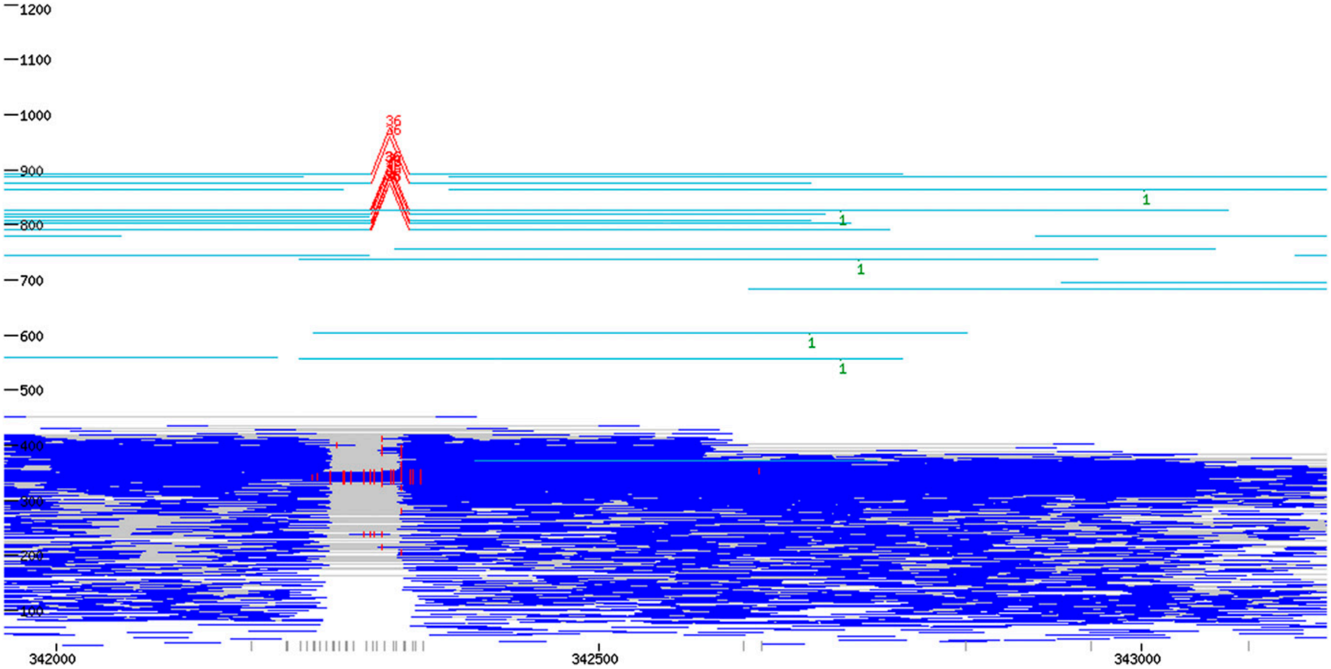

Figure 4. A deletion in a non-clonal sample. Illumina reads are in blue, read pairs are connected by gray lines. Reference mismatches are shown as red dots. Capillary reads (from CIGAR data) of the same sample are in cyan, with deletion as red triangles. Some of the capillary sequences do not show the deletion, and some of the Illumina reads extend into the gap created by the deletion; the presence of both variants within the sample is thus indicated by both technologies. The sample is from *P. falciparum*, all data generated at the Sanger Institute.

features, took in the region of 1 sec. Clearly, such comparisons of web installations might be influenced by many factors other than the software. Although stand-alone client-installed software might be configured to have faster response times, for many users it is impractical to undertake the process of installation and often compilation (e.g., Schatz et al. 2007a), as well as download and eventual conversion of entire sets of alignment data, that this requires.

While the LookSeq database format has been optimized for both size and speed, both vary with the database engine. In our test cases, the size of the SQLite databases averaged around 50 MB per 1 million Illumina reads. In terms of speed, SAM/BAM format appears to be slightly slower than SQLite and MySQL but is more versatile and, as an emerging standard format for alignments, does not require conversion.

Discussion

In LookSeq, we have attempted to develop a simple visual tool for users to assimilate large amounts of deep sequencing data. The only software installation required by the user is a standard web browser, and calculations are performed server-side allowing distributed analysis of large data sets. The use of AJAX technology, which enables seamless browsing and rapid zooming, makes it relatively effortless for a user to explore large and complex data sets. LookSeq provides a graphical representation of read pair fragment length, which is not featured in conventional alignment viewers, that allows the user to visualize on a single graph different lines of evidence for a structural variant.

In this initial version of LookSeq, we have not attempted to develop the full range of features that might be required by a broad base of users. Many of these additional features depend on integration with other applications and data formats that are still under development or rapidly evolving. By making the source code available open-source, we hope that useful features of LookSeq will be adapted and extended by other developers to serve the needs of particular user groups. Below we highlight some key considerations for future development of LookSeq and other deep sequencing alignment viewers.

The first issue is with the organization of deep sequencing alignment data for large genomes and how this is retrieved by a web application. In a conventional genome browser, each time that a user wishes a new view of the data, a request is submitted to the server and the new view is downloaded in its entirety, which can be time-consuming. In contrast, an AJAX-based viewer such as LookSeq downloads more information than the user immediately requires, so that new views can be rapidly generated without returning to the server. This works well for bacteria because current AJAX technologies can easily accommodate the amount of data required for a user to browse seamlessly and rapidly across the whole genome. It also works well for lower eukaryotes, such as *Plasmodium*, whose individual chromosomes can easily be viewed in their entirety. However the current version of LookSeq could not easily manage, e.g., a whole human chromosome, so it is necessary to organize the data in such a way that LookSeq can make appropriate requests to the server to retrieve part of a chromosome, say 2 Mb at a time, and to move easily from one part to the next.

The second issue revolves around the standardization of sequence alignment data formats. Alignment data contains the reads and their mapping to a reference, as generated by an alignment algorithm. CIGAR (http://www.ensembl.org/info/website/glossary.html), an alignment format often used for capillary data, is of limited use to new sequencing technologies, for which no alignment format is prevalent at the time of writing. As an initial working foundation, we have created a generic SQL database schema, which can store both CIGAR and new sequencing technology alignment data and can be populated from several algorithms, including MAQ, Bowtie (Langmead et al. 2009), and SSAHA2 (Ning et al. 2001). LookSeq can access such a database as both SQLite and MySQL and should be compatible with any database compatible to basic SQL standards. LookSeq can also directly use the SAM/BAM format of SAM Tools (Li et al. 2009), an emerging candidate for a sequence alignment standard. LookSeq can be expanded to accommodate more input formats in the future.

The third concerns the basic functionality of LookSeq and its extensibility. At time of writing, LookSeq can visualize read alignments and some basic properties, such as sequence annotation, coverage, and GC contents, as separate "tracks." This information is taken from the aforementioned alignment databases, as well as some auxiliary files (e.g., reference sequence, annotation database) stored alongside the alignment data. While this appears limiting at first glance, LookSeq, acting as a framework, will therefore enable extensibility in terms of further contextualized displays/tracks. These can either be generated by LookSeq itself by accessing local or remote databases, by "plug-ins" that can be installed in conjunction with LookSeq on the same server or by image/map-generating software hosted on a third-party server, essentially forming a so-called "mesh-up" composed of multiple tracks based on various processes, data sources, servers, and generation methods. Examples for such tracks could be automated predictions of large-scale structural variation, such as deletions, insertions, and copy number variations (CNVs); however, the generic nature of this concept would also allow for side-by-side display of read alignments from several samples for comparison, linkage disequilibrium (LD) maps, or properties unconceivable by us at this point. LookSeq will therefore offer virtually unlimited extensibility in terms of data volume, type, source, processing algorithm, and location.

Methods

Alignments of sequencing reads to a reference sequence can be calculated using a multitude of algorithms, which tend to have different output formats. It was our aim to allow for the visualization of the results of many of these algorithms. To achieve that, we designed a database schema that is versatile enough to accommodate common alignment data, while reducing storage requirements and access times to a minimum (Supplemental Fig. 8). We also created converter scripts for the output formats of MAQ for short reads and CIGAR alignments; support for further formats is planned for the future. Alternatively, LookSeq can use alignment data stored in the SAM/BAM format. SAM files are then used transparently instead of a database.

Each sample alignment, which may consist of several merged data sets of the same biological sample, is stored in its own database; auxiliary files and databases hold common, organism-specific data such as reference sequence and lists of potential SNPs.

In order to display alignments stored in a database, we created a series of Perl scripts, which query both the sample database and the organism-specific databases and generate either text-based query results or images for visualization. These scripts run on the same systems that hold the databases. The scripts can be invoked through the internet via their respective URL.

The user can access these displays through a web browser. For maximum flexibility, we created an interface featuring panning and zooming features using AJAX (Asynchronous JavaScript and XML) technology to interact with the user and translating intuitive commands such as dragging an image into queries to the aforementioned server-side Perl scripts, integrating both data and images generated by these scripts into a consistent browsing experience.

Database

Conversion from one of the alignment input formats yields a series of SQL commands, which are then used to create the actual database. On our demo site, we use SQLite (http://SQLite.org/) as our database engine. SQLite does not require a separate database server and stores each database in a separate file. MySQL can be used alternatively, which can be beneficial for large sample sets, but requires further server setup. Other SQL-based server software should work as well. ANSI standard SQL is used, so the database schema can be exported to other compliant SQL-based databases.

We found that for alignments based on short reads (e.g., Illumina), individual reads, or read pairs fall into one of four categories, i.e., perfectly mapping read pairs, read pairs with mismatches compared with the reference, read pairs indicating inversions, and single reads. The latter type of reads can result either from single-ended Illumina runs or from a read pair where only one end maps to the reference.

Consequently, we created four different table types, optimized for each read category. Perfectly mapping read pairs do not need to store the read sequence, as we know it to be identical to the reference at the alignment position. Similarly, where tables with read pairs hold two position values (one for each read "half"), the single read table only needs to hold one.

Reads containing mismatches are stored as a string of lowercase characters, except for the mismatches, which are uppercase. This allows the highlighting of mismatches in the visualization without the need to compare each read to the reference.

Within the visualization, only a single chromosome is accessed at any time. To reduce both the required amount of storage and the query speed of the database, the four tables are generated for each reference sequence (e.g., a chromosome) of the organism separately. This removes the need to store sequence information with each read and reduces data lookup to the reads within a single sequence, instead of all reads in the sample. Display of these read groups can be toggled in the application, so the image rendering script does not have to load and process read data for sequences or read types that are not required.

Capillary- and 454 Life Sciences-based (Roche) CIGAR alignments are stored in their own tables, as the number of reads of these types is typically much lower. Chromosome/contig data such as name and length, as well as meta-data such as read length, also reside in their own tables.

While our database schema performs well today, it is not entirely generic. Therefore, LookSeq also supports SAM format, an emerging standard for alignment formats. We are in the process of extending LookSeq to take advantage of special SAM features, including but not limited to display of small InDels, quality scores, read counts, and read paired with reverse orientation.

Code repository

The source code for LookSeq is available at the open-source repository SourceForge. It can be downloaded via the Subversion software as described on http://sourceforge.net/svn/?group_id=240451.

About the sequencing data shown in this paper

We illustrate the method with whole-genome sequencing data from the malaria parasite *Plasmodium falciparum*, whose genome size is 23 Mb. Sequencing was performed with an Illumina Genome Analyzer. *P. falciparum* DNA was sheared into fragments of about 200 bp, and each fragment was sequenced for about 35 bp at both ends, i.e., a total of about 70 bp per fragment: This is known as a paired end read. Each lane generated about 7 million paired end reads, i.e., a total of about 500 mB sequence data, which in theory would give about 20× depth of coverage if distributed evenly across the *P. falciparum* genome. By analyzing the same sample on multiple lanes, it is relatively straightforward to view more than 100× coverage for a single *P. falciparum* isolate.

Acknowledgments

We thank Yali Xue at Sanger for the human chromosome Y screenshot (Supplementary Fig. 9). We thank our colleagues in the malaria, pathogen, and sequencing teams at the Wellcome Trust Sanger Institute, as well as colleagues who have contributed *P. falciparum* samples for sequencing, for many helpful suggestions and for allowing use of anonymized samples for demonstration purposes. We thank Heng Li for helping us with optimizing MAQ settings. This work was funded by the Wellcome Trust, the Bill and Melinda Gates Foundation, and the Medical Research Council.

References

Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, Hall KP, Evers DJ, Barnes CL, Bignell HR, et al. 2008. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456:** 53–59.

Bonfield JK, Smith K, Staden R. 1995. A new DNA sequence assembly program. *Nucleic Acids Res* **23:** 4992–4999.

Branton D, Deamer DW, Marziali A, Bayley H, Benner SA, Butler T, Ventra MD, Garaj S, Hibbs A, Huang X, et al. 2008. The potential and challenges of nanopore sequencing. *Nat Biotechnol* **26:** 1146–1153.

Campbell PJ, Pleasance ED, Stephens PJ, Dicks E, Rance R, Goodhead I, Follows GA, Green AR, Futreal PA, Stratton MR. 2008a. Subclonal phylogenetic structures in cancer revealed by ultra-deep sequencing. *Proc Natl Acad Sci* **105:** 13081–13086.

Campbell PJ, Stephens PJ, Pleasance ED, O'Meara S, Li H, Santarius T, Stebbings LA, Leroy C, Edkins S, Hardy C, et al. 2008b. Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nat Genet* **40:** 722–729.

Dear S, Staden R. 1991. A sequence assembly and editing program for efficient management of large projects. *Nucleic Acids Res* **19:** 3907–3911.

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

Harris TD, Buzby PR, Babcock H, Beer E, Bowers J, Braslavsky I, Causey M, Colonell J, Dimeo J, Efcavitch JW, et al. 2008. Single-molecule DNA sequencing of a viral genome. *Science* **320:** 106–109.

Hillier LW, Marth GT, Quinlan AR, Dooling D, Fewell G, Barnett D, Fox P, Glasscock JI, Hickenbotham M, Huang W, et al. 2008. Whole-genome sequencing and variant discovery in C. elegans. *Nat Methods* **5:** 183–188.

Huang W, Marth G. 2008. EagleView: A genome assembly viewer for next-generation sequencing technologies. *Genome Res* **18:** 1538–1543.

Langmead B, Trapnell C, Pop M, Salzberg SL. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* **10:** R25. doi: 10.1186/gb-2009-10-3-r25.

Li H, Ruan J, Durbin R. 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res* **18:** 1851–1858.

Li R, Li Y, Kristiansen K, Wang J. 2008. SOAP: Short oligonucleotide alignment program. *Bioinformatics* **24:** 713–714.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. 1000 Genome Project Data Processing Subgroup. 2009. The sequence alignment/map (SAM) format and SAMtools. *Bioinformatics* **25:** 2078–2079.

Lin H, Zhang Z, Zhang MQ, Ma B, Li M. 2008. ZOOM! Zillions of oligos mapped. *Bioinformatics* **24:** 2431–2437.

Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen Y-J, Chen Z, et al. 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437:** 376–380.

Ning Z, Cox AJ, Mullikin JC. 2001. SSAHA: A fast search method for large DNA databases. *Genome Res* **11:** 1725–1729.

Schatz MC, Phillippy AM, Shneiderman B, Salzberg SL. 2007a. Hawkeye: An interactive visual analytics tool for genome assemblies. *Genome Biol* **8:** R34. doi: 10.1186/gb-2007-8-3-r34.

Schatz MC, Trapnell C, Delcher AL, Varshney A. 2007b. High-throughput sequence alignment using Graphics Processing Units. *BMC Bioinformatics* **8:** 474. doi: 10.1186/1471-2105-8-474.

Shendure J, Ji H. 2008. Next-generation DNA sequencing. *Nat Biotechnol* **26:** 1135–1145.

Shendure J, Porreca GJ, Reppas NB, Lin X, McCutcheon JP, Rosenbaum AM, Wang MD, Zhang K, Mitra RD, Church GM. 2005. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* **309:** 1728–1732.

Smith AD, Xuan Z, Zhang MQ. 2008. Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC Bioinformatics* **9:** 128. doi: 10.1186/1471-2105-9-128.

Stein LD, Mungall C, Shu S, Caudy M, Mangone M, Day A, Nickerson E, Stajich JE, Harris TW, Arva A, et al. 2002. The generic genome browser: A building block for a model organism system database. *Genome Res* **12:** 1599–1610.

Received March 5, 2009; accepted in revised form July 30, 2009.