BRIEFINGS IN BIOINFORMATICS. VOL 14. NO 2. 193–202
Advance Access published on 24 March 2012
doi:10.1093/bib/bbs012

Using Tablet for visual exploration of second-generation sequencing data

Iain Milne, Gordon Stephen, Micha Bayer, Peter J.A. Cock, Leighton Pritchard, Linda Cardle, Paul D. Shaw and David Marshall

Submitted: 2nd July 2011; Received (in revised form): 24th February 2012

Abstract

The advent of second-generation sequencing (2GS) has provided a range of significant new challenges for the visualization of sequence assemblies. These include the large volume of data being generated, short-read lengths and different data types and data formats associated with the diversity of new sequencing technologies. This article illustrates how Tablet—a high-performance graphical viewer for visualization of 2GS assemblies and read mappings—plays an important role in the analysis of these data. We present Tablet, and through a selection of use cases, demonstrate its value in quality assurance and scientific discovery, through features such as whole-reference coverage overviews, variant highlighting, paired-end read mark-up, GFF3-based feature tracks and protein translations. We discuss the computing and visualization techniques utilized to provide a rich and responsive graphical environment that enables users to view a range of file formats with ease. Tablet installers can be freely downloaded from http://bioinf.hutton.ac.uk/tablet in 32 or 64-bit versions for Windows, OS X, Linux or Solaris. For further details on the Tablet, contact tablet@hutton.ac.uk.

Keywords: *visualization; second-generation sequencing; assembly validation*

INTRODUCTION

Visualization of DNA sequence assemblies and read mappings enables scientific discovery by providing new insights that may not otherwise be obvious, for example, elucidation of gene structure and visualization of alternative mRNA splice products. It also forms a critical part of the quality assurance process, for example, in visual confirmation of polymorphisms, and supplements existing assessment methods such as the computation of the N_{50} statistic [1] and testing against known sequences with similarity search algorithms (e.g. BLAST [2]).

However, the emergence of second-generation sequencing (2GS) technologies (e.g. Illumina (http://www.illumina.com), Roche 454 [3], ABI Solid [4]), along with their fast pace of change, and a combination of short read lengths and vast read numbers have rendered most pre-2GS tools for sequence assembly and visualization practically unusable, stimulating the development of a new generation of applications.

Additionally, the reduced cost per nucleotide has meant that new uses for 2GS sequencing have evolved, such as genotyping and quantitative

Corresponding author. Iain Milne, Information & Computational Sciences, The James Hutton Institute (JHI), Dundee DD2 5DA, Scotland, UK. Tel: +44 (0) 844 928 5428; Fax: +44 (0) 844 928 5429; Email: iain.milne@hutton.ac.uk

Iain Milne is a postdoctoral computer scientist at James Hutton Institute (JHI), and is the principal developer of several popular visualization-focused bioinformatics applications.

Gordon Stephen is a computer scientist at JHI and works mainly on bioinformatics programming.

Micha Bayer is a bioinformatician at JHI. The main focus of his work is the processing and analysis of second-generation sequencing data.

Peter Cock not only works on plant pathogen genomics at JHI, but also contributes to open source science projects such as the Galaxy workflow system and Biopython.

Leighton Pritchard is a computational biologist at JHI and works mostly on microbial plant pathogens.

Linda Cardle is a bioinformatician at JHI whose main focus is the assembly, analysis and annotation of second-generation sequence data.

Paul Shaw is a bioinformatician at JHI and works primarily in biological database development and pedigree visualization.

David Marshall heads the Information and Computational Sciences Group at JHI. His principal interest is the analysis of genotype and sequence diversity in crop plants.

expression analysis. This has led to a larger uptake of sequencing among biologists, and it is now more important than ever before to provide the community with software that is available for most computing platforms and is both easy to install and use.

In our previous publication [5] (Figure 1), we described Tablet, an assembly viewer designed from the outset to efficiently handle 2GS datasets. It provides biologists and bioinformaticians with fast, interactive visualizations packaged up in an intuitive interface that can be installed and run across a number of operating systems without any prerequisites and with minimal system requirements.

Since Tablet's original publication at the beginning of 2010, we have released a further 17 public updates for it. These have made almost 100 new features available, including many important additions to its core functionality, such as support for the SAM and BAM [6] alignment formats, paired-end read visualization, new colour schemes and visualization enhancements, import and plotting of annotation and restriction enzyme data and many visual and interactive aids for locating and inspecting regions of biological interest. A full listing of all new functionality is available on the Tablet website.

In this article, we take a detailed look at Tablet's features and cover its new capabilities and enhancements before moving on to discuss how its visualizations support 2GS analysis. We then provide a technical overview describing some of the more important aspects of its implementation. Finally, we will look at some of the functionality and features planned for Tablet's further development.

FEATURES

Tablet supports both 2GS and Sanger [7] data in the following commonly used file formats: ACE, AFG, MAQ, SOAP, SAM or BAM (Table 1). Read and reference/consensus data are included in ACE and

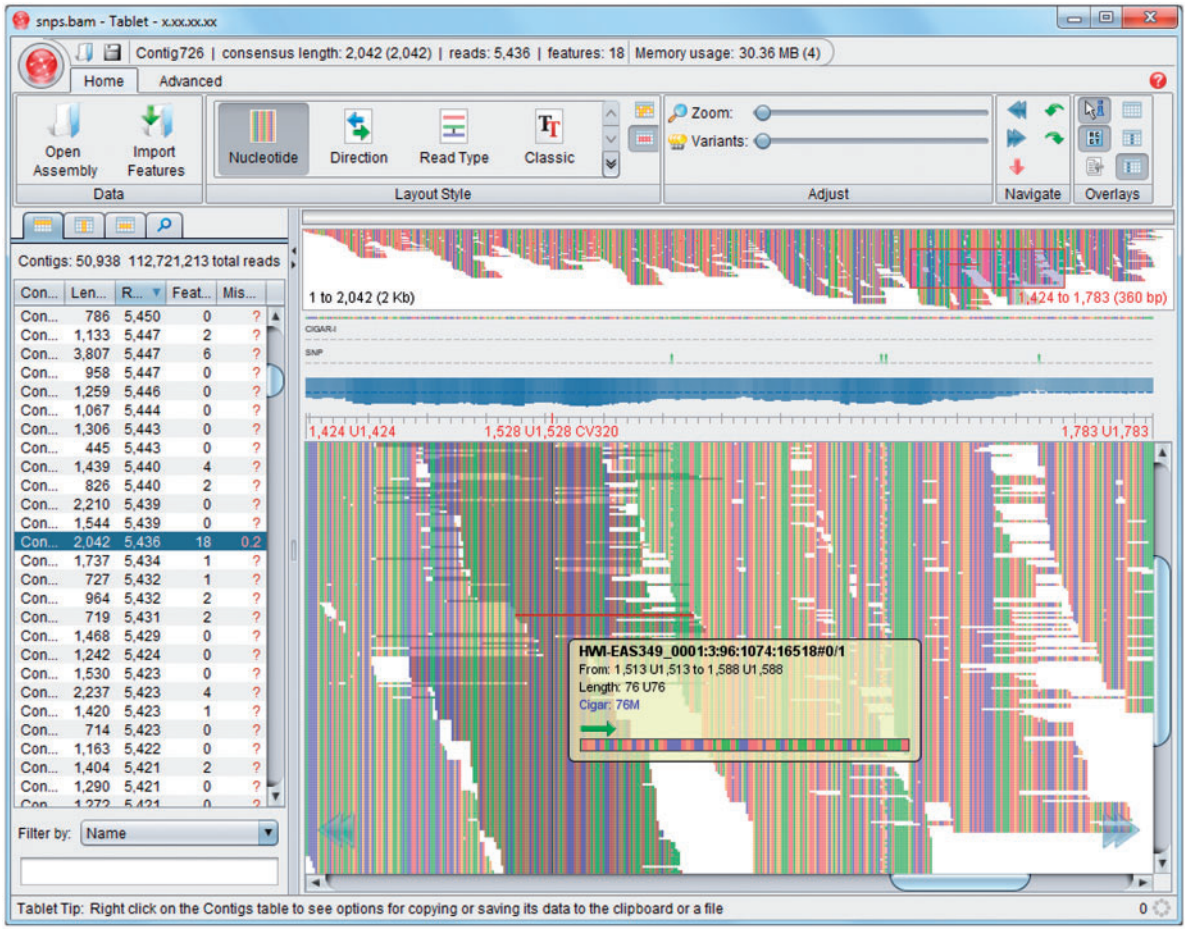

Figure 1: A small contig from an Illumina BAM file as seen in Tablet. The darker shading over the reads is tracking all reads at the same nucleotide position as the user's cursor. The read directly under the mouse is highlighted and summarized in the tooltip, along with a graphical representation of its sequence.

AFG files, unlike the other formats that contain only the read data. Reference sequences are held separately in these cases and can be loaded from either FASTA or FASTQ input files. These are optional however, and Tablet can still display a mapping without them.

Once loaded, an assembly/mapping is presented in a live-filterable table that lists its sequence contigs (or reference sequences), along with other supplementary information such as their lengths and read counts (N.B. the term 'contig' is used as a catch-all phrase for any single unit of visualization, be it a genuine contig or scaffold from a *de novo* assembly, or a mapping of reads against a reference sequence). This provides a useful overview when working on a finished or nearly finished genome with a small number of chromosomes (e.g. the human genome). However, the driver behind this design is to cope with draft genomes where hundreds of thousands of contigs is not uncommon, and a simple drop-down list is not practical.

Tablet visualizes a selected contig by rendering both reference/consensus and read data as sequences of colour-coded bases running from left-to-right across the screen. Although this style of display is relatively common, it presents challenges in terms of optimizing for performance and visual impact. We have built upon our experiences in developing TOPALi [8] and Flapjack [9] to ensure the end result renders fluidly in a visually appealing manner and draws attention to informative regions of the visualization. First, low saturation colours are chosen, making the display easier on the eye, especially with a very busy screen full of thousands of reads. Second, each nucleotide base uses a gradient–paint technique that helps to maintain visual structure within large areas of what would otherwise be identical colour, for example, blocks of aligned reads with a region of homopolymer DNA. The gradient allows the user to perceive individual columns, rather than a single contiguous run of colour. Finally, Tablet only displays the actual nucleotide values when there is the resolution to do so, meaning that below a certain zoom level (when the text would become too small to maintain readability) this overlay is disabled, resulting in a cleaner visualization when zoomed further out. The textual value for each base is also anti-aliased which improves clarity against the graded background.

A range of colour schemes are available, with live-preview functionality to assist in spotting patterns in the data that the different schemes may present. The basic nucleotide scheme renders bases by letter, drawing attention to sequence-composition patterns such as microsatellites, poly-A tails, mononucleotide runs and GC rich areas. Other schemes employ colours which indicate the direction reads were sequenced in, read type, sample name (from SAM/BAM read group headers) or paired status. All schemes support functionality to dynamically highlight variant bases (where a read differs from the reference) which can provide visual notification of single nucleotide polymorphisms (SNPs) or sequencing errors.

The reads can be displayed on screen at varying zoom levels, and in one of two primary layout modes: *packed*—where Tablet attempts to keep every read as close to the top of the screen as possible, only moving reads further down when they overlap another read; and *stacked*—where each read is displayed on its own row. Variations on these layout schemes are also available when viewing

Table 1: Input file formats supported by Tablet

Input file	Format	Type	Paired-end[a]	Random access[b]
ACE	Text	Read + Reference/Consensus	No (but planned)	No
AFG	Text	Read + Reference/Consensus	No (but planned)	No
MAQ	Binary/text[c]	Mapping	No	No
SOAP	Text	Mapping	No	No
SAM	Text	Mapping	Yes	No
BAM	Binary	Mapping	Yes	Yes
FASTA	Text	Reference/Consensus	N/A	No
FASTQ	Text	Reference/Consensus	N/A	No
GFF3	Text	Feature Annotations	N/A	No

[a]Paired-end visualization support in Tablet, [b]Random access to specific data within the file without the requirement to load all of the file, [c]Binary output requires conversion to MAQ's text format before viewing in Tablet.

paired-end data, which forces Tablet to try and keep paired reads on the same row at all times, in addition to visually linking the two ends of a pair.

Designed for responsive and intuitive navigation, Tablet provides smooth scrolling and panning of the data at all zoom levels. Navigation is further aided by overview visualizations that display either the read coverage across a contig or a scaled-to-fit summary of all of its data, enabling simultaneous local and global views of the data. The current position within a contig is displayed graphically and clicking or dragging with the mouse will instantly relocate the main view to that position. For very large contigs, it is possible to generate subsets of these overviews to provide more detailed summaries of specific regions.

Tablet can plot features on one or more tracks running alongside the data. Any type of biological (or custom) feature is supported by this, provided that the data can be accommodated by the GFF3 format (http://www.sequenceontology.org/gff3.shtml). The minimum requirement for this is for a feature to have a start and end position (although these can be the same), and an association with a contig. Examples of features used in our own work include SNPs, indels, restriction enzyme cleavage sites and gene-associated features such as start/stop codons, exons and introns.

To draw attention to particular facets of the visualization, many components support contextual highlighting, where the main display will instantly navigate to and refocus on a region of interest, while dimming the rest of the display for further clarity. Examples include selecting features or reads from their respective tables (the latter providing an at-a-glance summary of all reads on screen), or from results generated by Tablet's advanced search facility where the user can apply regular expression-based searches to locate reads by name (within or across contigs), or to search for subsequences of nucleotide data across the data set. Some highlighting features also function in real-time, such as drawing attention to all reads intersecting the mouse's location, visually linking mate pairs, or marking the position of a highlighted read on the overview.

Detailed information on reads is displayed using a visually rich tooltip, which lists a read's name, location, length and orientation, and also renders a graphic showing an overview of the read's structure. The tooltip can adapt its display to the assembly format in use (Figure 2), providing additional information if the reads contain CIGAR [10] entries, or in the case of paired data where mate position, inferred insert size and a twin tooltip containing the mate's information can then be shown.

The read data are exportable using right-click menu options. Output is formatted as FASTA and can be copied to the clipboard or saved to a file. This can include just a single read or reference/consensus sequence, all reads at a given position, all reads on screen or all reads within a contig. Export options also extend to saving coverage summaries for an assembly/mapping, or to output screen captures of Tablet's visualizations. Other right-click options exist to aid navigation. Clicking on a read presents options to focus the view on its starting or ending location. With paired-end data, refocusing can be applied to a read's mate, regardless of whether it is located within the same contig or another.

To aid integration with analysis workflows or automated pipelines, Tablet supports a number of startup parameters, such as being able to specify not only the input files to load, but also the contig and nucleotide position to be initially displayed. Both local (command line switches) and remote (Java Web Start) options are available, and all input files can be hosted remotely which is particularly efficient for very large BAM files. As an example, results from a SNP detection workflow could be presented to the user with the option to view any SNP in Tablet, to allow manual inspection of the aligned reads.

APPLICATIONS

A common application of second-generation sequencing is whole-genome shotgun sequencing. This involves visual validation of the draft *de novo* assemblies at all stages of the process, for example, looking for variations in coverage level. Regions of high coverage may be indicative of a collapsed repeat, while regions of very low coverage could indicate misassembly as a result of extending the assembled region beyond its true extent, to join two or more regions that are not in fact contiguous. Figure 3 demonstrates good, even read coverage from reads from two different sequencing technologies (454/Illumina) in a hybrid assembly.

To add a greater degree of confidence to an assembly, it may be practical to use more than one assembler, or more than one parameter set per assembler. This exploits the relative advantages of different assemblers to resolve difficult regions, and

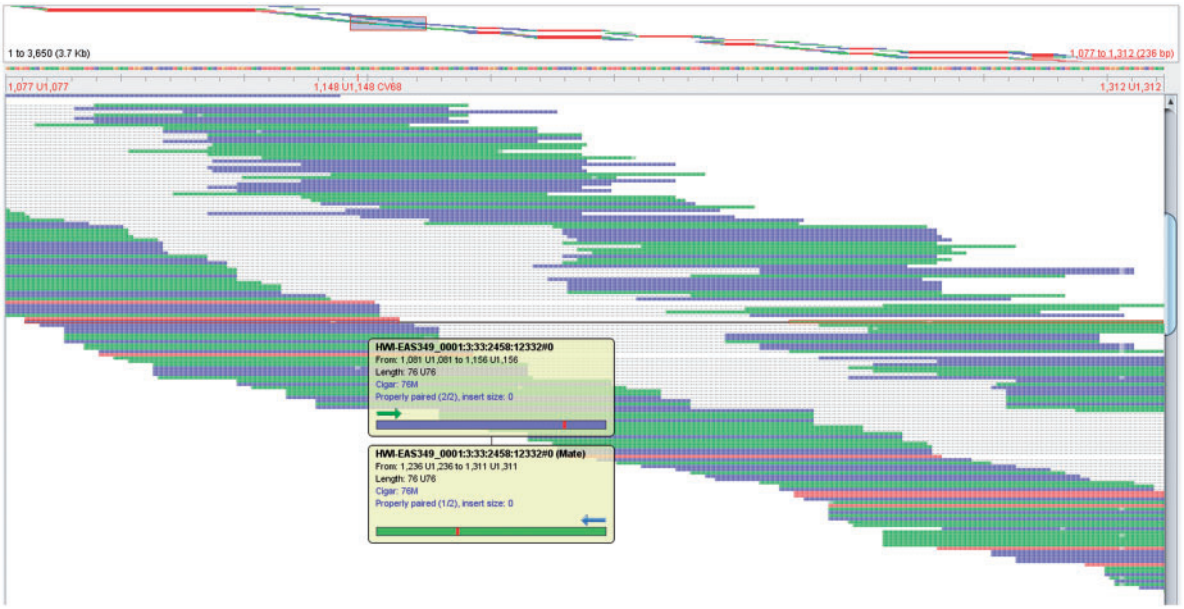

Figure 2: Paired-end data from an Illumina RNA-Seq read mapping. Here, the data are stacked and the read type colour scheme is in use, displaying mate status in green (first in pair) and blue (second in pair). Orphaned reads are in red. Both reads within a pair are simultaneously rendered by the tooltip. Also of note are the variant bases—marked up in red—visible within each read's graphical summary.

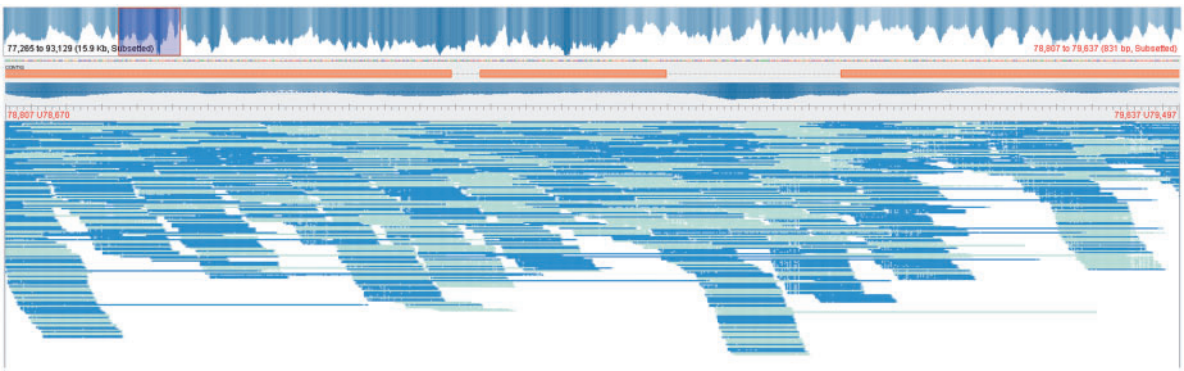

Figure 3: Visualization of a hybrid 454/Illumina bacterial *de novo* assembly in Tablet, with assembled contigs from a NEWBLER (Roche) assembly of the same 454 data shown on the feature track (orange blocks). The longer 454 reads are clearly visible among the shorter Illumina reads. Consistently high coverage levels (blue histograms) across this block indicate that the additional Illumina data support the assembly of three NEWBLER contigs into a single contig.

allows for the choice of an optimal assembly from amongst several options. The outputs of different assemblers or choices of parameter values may be compared visually in Tablet to assess their effect on, and confidence in the assemblies, and to help resolve differences between them (Figure 4). Similarly, visualization of the 'meta-assembly' of two or more draft assemblies may also help resolve misassemblies, or apparent repeats that originate from assembler stringency in the meta-assembly process.

Tablet is also useful for visual inspection of transcripts from *de novo* RNA sequence assemblies. This is illustrated in Figure 5, showing an initial MIRA 454 assembly [11] of a nematode transcriptome (unpublished data). The coverage plot immediately draws attention to the possibility of a misassembly between the short high coverage region on the right, and the larger left-hand portion with at most half the coverage. Running BLASTX [12] on the contig sequence against the NR database gives

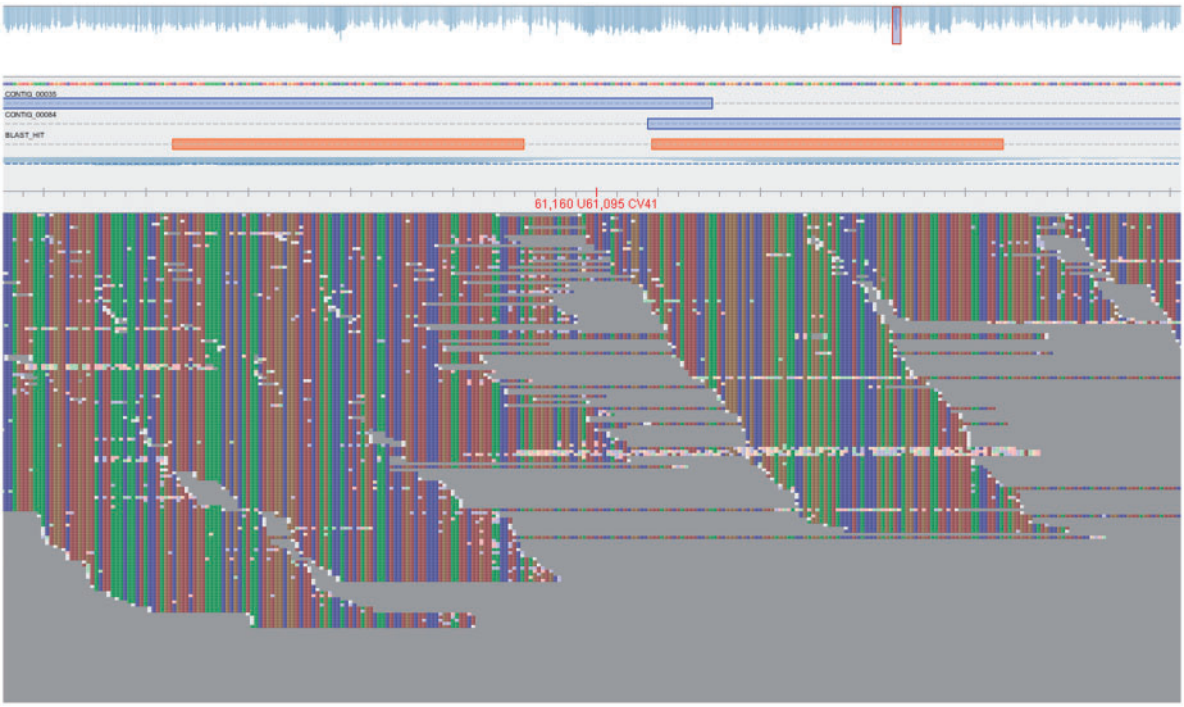

Figure 4: Visualization of a hybrid 454/Illumina bacterial *de novo* assembly, with contigs from a NEWBLER assembly of the same 454 data shown on the feature track (blue blocks above the canvas), and aligned tRNA features from a related isolate (orange blocks). Two contigs from the NEWBLER assembly overlap in the hybrid assembly, and two tRNAs from a Sanger-sequenced, related isolate align to this region. The two NEWBLER contigs each correspond to a single tRNA sequence. In the sequenced isolate, this region is annotated and contains an additional, third tRNA gene. This suggests possible misassembly using the hybrid approach, due to the similarity of tRNA sequences and short Illumina read size. By inspection using Tablet, sequence coverage was found to be low in the region flanking the second tRNA showing, and the number and pattern of mismatches (light bases) corresponds to the missing tRNA, consistent with a misassembly in which one of the three tRNA genes is erroneously excluded.

excellent and consistent matches to animal proteins for 81% of the contig. Checking the high-coverage region reveals this to be a SMART primer sequence used in the sample preparation, which had not been removed prior to assembly. Thus, inspecting the assembly in Tablet quickly highlighted a simple (and easily rectified) error in the construction of this assembly.

Once a satisfactory assembly has been reached, typically the next step is annotation with gene finding software such as Augustus [13] or Glimmer [14] to elucidate gene structure. Importing these annotations into Tablet allows biologists to view gene models alongside the contigs produced, for example, enabling visual inspection of reads supporting apparent frame shifts or premature stop codons.

Another key use of loading gene annotation into Tablet is for transcriptome data mapped to a reference genome. The coverage of RNA-Seq reads can visually support (or contradict) predicted gene models, particularly intron/exon boundaries in eukaryotes. Figure 6 shows a contig from a draft genome *de novo* assembly produced using all of the reads from a full Illumina HiSeq run. The CLC Bio Assembly Cell (http://www.clcbio.com/index.php?id=1331) *de novo* assembler was used to generate the contigs.

The original Illumina RNA-Seq reads were mapped onto this contig sequence using the Bowtie-Tophat pipeline. Bowtie [15] is used to map all reads onto the reference that can be matched along their full length, but retains all reads that cannot be mapped in this way. A significant proportion of the unmapped reads span splice junctions, but normally these would be discarded along with reads that fail to match criteria in other ways. Tophat [16] maps the retained reads across the splice junctions, splitting them into two sections as appropriate, annotating the gap with a CIGAR markup. In the absence of alternatively spliced transcripts, the

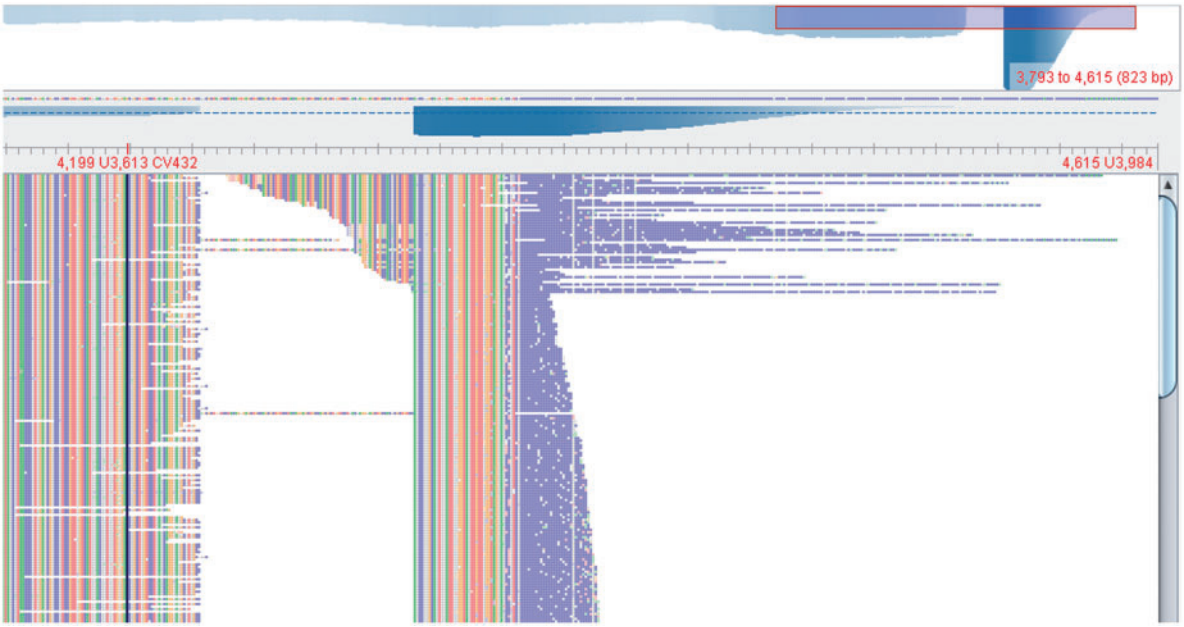

Figure 5: Visualizing a MIRA 454 *de novo* assembly of a nematode transcriptome (unpublished data). The black vertical bar was added via the context menu to highlight ungapped column 3613, the right most edge of BLASTX matches to the reverse complement of a conserved animal protein. This region likely marks the true start of the protein, with its poly-A tail at the start of the contig (off screen to the left). The high coverage region at the rightmost end of the contig (on screen) is from the SMART primer sequence used in preparation of the library, erroneously not removed before this assembly.

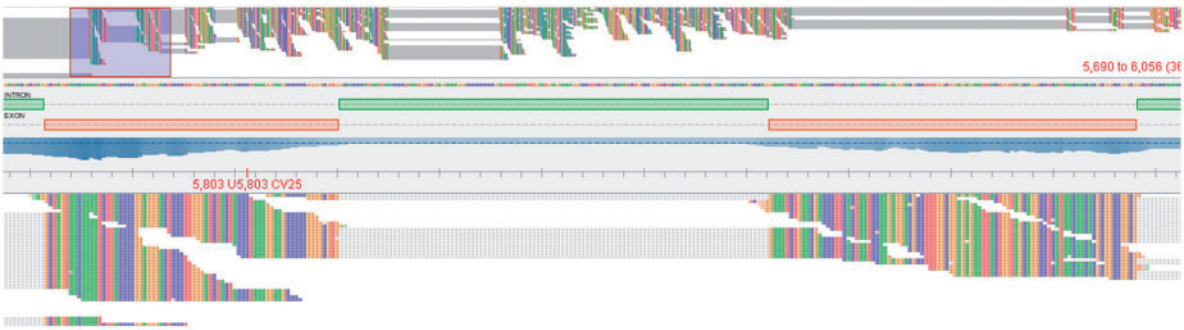

Figure 6: A draft genome contig, with an imported GFF3 feature track of gene models generated with Augustus. The blocks of RNA-Seq reads (mapped withTopHat) are generally in very good agreement with the gene prediction visible on the feature tracks running above the mapped reads. The upper track shows non-coding regions in green, while the lower shows coding regions in red.

RNA-seq reads will form clearly delineated blocks separated by intronic regions where no reads have been mapped. The results of this analysis allow Tablet to provide an insightful visualization for gene model validation.

Another important application of Tablet is variant detection and validation. In Figure 7, Illumina RNASeq reads from a number of plant samples were mapped onto an existing set of Sanger EST-derived unigenes. SNP discovery was then carried out using the GigaBayes [17] variant detection software, with the resultant GFF3 file containing the SNP locations loaded into Tablet.

TECHNOLOGIES

Tablet enables fast, visually rich exploration of second-generation sequence data by utilizing

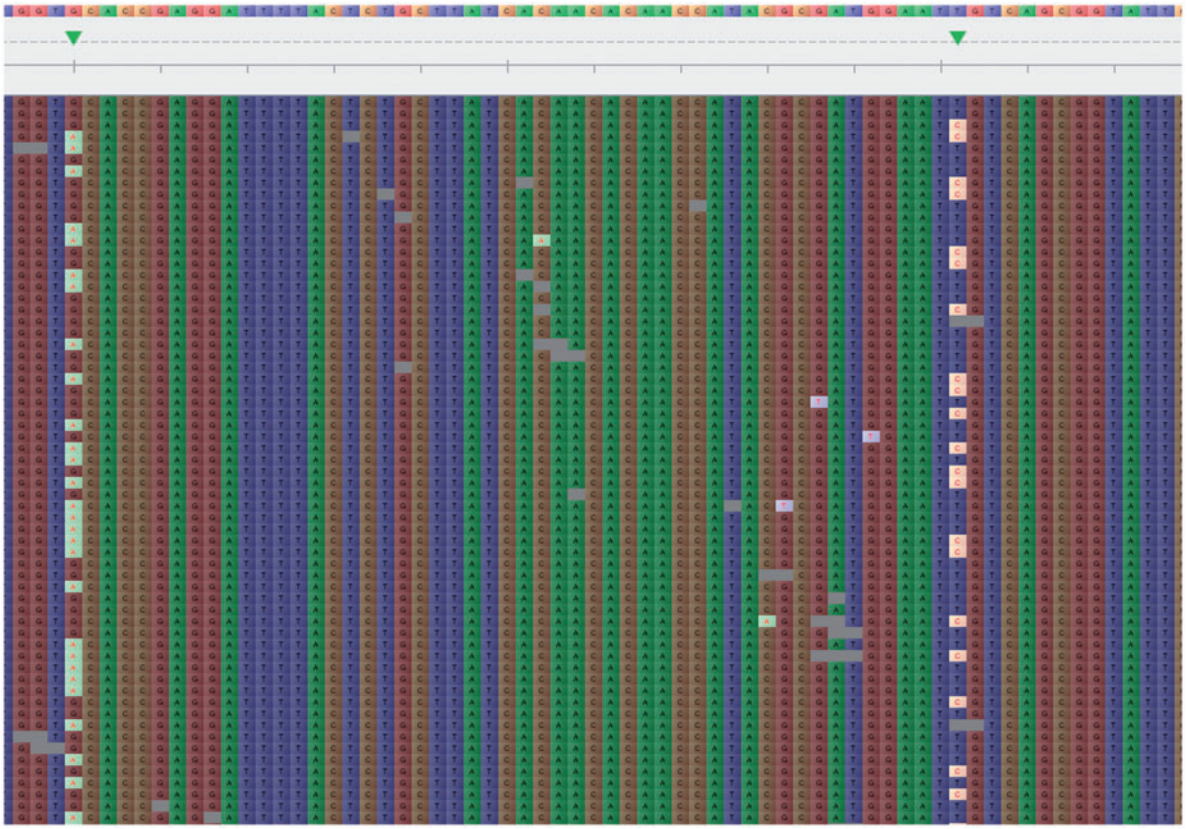

Figure 7: Visual validation of SNPs is facilitated by Tablet's variant highlighting feature, here set high, aiding the validation of several visible SNPs by making them stand out from the rest of the data. Random read errors distributed across the mapping are also visible as lighter points on the display.

a combination of programming techniques—and optimizations—that allow it to be functional yet lightweight.

In Ref. [5], we describe memory-based and disk-based strategies for viewers handling 2GS datasets. The former is often applicable to text-based formats such as ACE and AFG, whereas the latter is particularly suited to the binary BAM format, designed as it is to allow small, location-based snapshots of data to be loaded from large multi-gigabyte files. Indeed, its indexing makes it practical for Tablet (via the Picard library [6]) and other assembly viewers to display BAM files hosted on remote servers.

In an attempt to utilize the strengths of both of these approaches, and due to many commonly used 2GS analysis tools only supporting non-BAM formats, Tablet was conceived as a hybrid viewer. Memory use is kept to a minimum by holding only a 'skeleton' of the data that can be used to determine where—in a Tablet layout—a read should be displayed. For each read in this skeleton, we simply store an ID, its position against the consensus sequence and its length. All other information is kept on disk. This approach allows many millions of reads to be handled without a significant memory overhead but also enables features such as entire-contig overviews, coverage plots and instantaneous navigation from one point to another. Knowing the location of all reads also eliminates the erratic visual repacking of data that occurs as reads come into or out of view and the local scale changes.

For actual visualization of the assembly/mapping, each read must be populated with its DNA sequence data which is expressible as a simple string of letters and kept on disk until required. Storing these sequences is inefficient, requiring two bytes of memory for every (Unicode) character. By defining our own internal alphabet that requires less than 16 characters in total, Tablet can store two bases per byte of memory (half a byte can store any value between 0 and 15), thereby achieving better compression (4×) and faster I/O. This is essential to maintain a smooth frame-rate while scrolling. The ID for each

read is used to locate and access its sequence on disk via a buffered random access reader optimized for Tablet's renderer.

A second cache implementation utilizes an embedded SQLite (http://www.sqlite.org) database for supplementary read information that is not required for real-time rendering. For example, locating a paired read's mate requires information (the mate's name, its contig, etc.) that is only required temporarily, and would add too much overhead if kept in memory. Additionally, this cache holds the names of every read in the assembly (in itself potentially GBs of data) and is optimized to allow Tablet to search through them quickly.

Another important factor in the speed of display concerns the actual rendering techniques used for nucleotide bases, which contain multiple gradient, transparency and anti-aliased effects. Requiring significant computational time to draw, we alleviate this problem by pre-creating just a single copy of each possible type. These images can be bit-block transferred to the drawing canvas thousands of times faster than recreating the graphic at each location would take. A further speed increase is achieved by pre-processing and storing variant bases as separate alphabet entries. This allows them to be visually distinct from normal bases with no increase in render time. Various double and triple buffering techniques are used to reduce queries to the caches and ensure that screen updates remain fast while allowing graphically rich overlays, highlights and basic animation to be run in response to user interactions.

Tablet employs multi-core processors when available. Various background tasks, such as the creation of the overview and coverage diagrams or the translations of large runs of DNA sequence into protein will run multi-threaded. More significantly, a novel interleaving technique allows for the simultaneous rendering of as many rows of read data at once as there are processors available, giving the renderer a near-linear speed increase per core.

Tablet is written using Java 1.6 but is packaged for distribution using install4j (http://www.ej-technologies.com/products/install4j/overview.html) that allows it to be easily installed and updated on all supported systems, including those without an existing Java runtime. Tablet is freely available from http://bioinf.hutton.ac.uk/tablet along with documentation and sample files.

FUTURE DIRECTIONS

We are currently concentrating on providing additional methods for drawing attention to read data, such as via additional colour schemes based on read lengths, insert sizes for paired-end data and extending the current read group scheme to fully support visualization of all available flags (such as sequencing technology and library). Further extensions to this will allow for (re)sorting of the display based on read or sample names. Purposely highlighting regions of extreme coverage is also planned, while a more general mechanism to show other numerical data such as calculated GC content/skew is under consideration.

We plan to enhance our GFF3 support so that visualization of hierarchical data structures is possible. Beyond GFF3, we intend to include support for automatic restriction site tagging and highlighting using data provided via REBASE [18]. The current search functionality will be enhanced to provide searching through features too.

Paired-end visualization support within Tablet is limited to assemblies in SAM or BAM formats, and we may investigate extending this to other formats that can encapsulate this data, such as AFG.

Tablet is under continual development, and versions with new features and performance enhancements are released frequently. The email address listed earlier can be used to contact us with comments, bug reports or requests for new features or functionality.

Key Points

- 2GS presents formidable challenges for visualization software.
- Tablet is a lightweight, high-performance graphical viewer for second-generation assemblies and read mappings.
- Visual inspection of 2GS data with Tablet aids scientific discovery and quality assurance.

Acknowledgements

We thank Joanne Russell, Arnis Druka, Robbie Waugh, Ian Toth and John Jones for providing the data used in some of the examples. We would also like to thank colleagues within the Cell & Molecular Sciences and BioSS Programmes at The James Hutton Institute for their input to this project.

FUNDING

The Scottish Government (RERAD, Programme 1); the Scottish Funding Council; Scottish Enterprise through the Scottish Bioinformatics Research Network (SBRN) project.

References

1. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. *Genomics* 2010;**95**: 315–27.
2. Altschul SF, Gish W, Miller W, *et al*. Basic local alignment search tool. *J Mol Biol* 1990;**215**:403–10.
3. Margulies M, Egholm M, Altman W, *et al*. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 2005;**437**:376–80.
4. Pandey V, Nutter RC, Prediger E. Applied Biosystems SOLiD™ System: ligation-based sequencing. In: Janitz M, (ed). *Next Generation Genome Sequencing: Towards Personalized Medicine*. Weinheim, Germany: Wiley-VCH, 2008.
5. Milne I, Bayer M, Cardle L, *et al*. Tablet—next generation sequence assembly visualization. *Bioinformatics* 2010;**26**: 401–2.
6. Li H, Handsaker B, Wysoker A, *et al*. The sequence alignment/map format and SAMtools. *Bioinformatics* 2009;**15**: 2078–9.
7. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci USA* 1977; **74**:5463–67.
8. Milne I, Lindner D, Bayer M, *et al*. TOPALi v2: a rich graphical interface for evolutionary analyses of multiple alignments on HPC clusters and multi-core desktops. *Bioinformatics* 2009;**25**:126–7.
9. Milne I, Shaw P, Stephen G, *et al*. Flapjack—graphical genotype visualization. *Bioinformatics* 2010;**26**:3133–4.
10. Slater GSC, Birney E. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* 2005;**6**:31.
11. Chevreux B, Wetter T, Suhai S. Genome sequence assembly using trace signals and additional sequence information. *Comput Sci Biol: Proc Ger Conf Bioinformatics* 1999;**99**: 45–56.
12. Camacho C, Coulouris G, Avagyan V, *et al*. Blast+: architecture and applications. *BMC Bioinformatics* 2009;**10**:421.
13. Stanke M, Waack S. Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics* 2009;**19(Suppl. 2)**:215.
14. Salzberg S, Delcher A, Kasif S, *et al*. Microbial gene identification using interpolated Markov models. *Nucleic Acids Res* 1998;**26**:544–8.
15. Langmead B, Trapnell C, Pop M, *et al*. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 2009;**10**:R25.
16. Trapnell C, Pachter L, Salzberg S. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 2009;**25**: 1105–1111.
17. Marth GT, Korf I, Yandell MD, *et al*. A general approach to single-nucleotide polymorphism discovery. *Nat Genet* 1999; **23**:452–6.
18. Roberts RJ, Vincze T, Posfai J, Macelis D. REBASE—a database for DNA restriction and modification: enzymes, genes and genomes. *Nucleic Acids Res* 2010;**38**:234–6.