DiscoPlot: Discordant read visualisation

Mitchell J. Sullivan, Scott A. Beatson*

*Australian Infectious Diseases Research Centre, and School of Chemistry and Molecular Biosciences, The University of Queensland, Brisbane, QLD 4072, Australia

*To whom correspondence should be addressed. Tel: +61 7 3365 4863; Email: s.beatson@uq.edu.au

Keywords: Bioinformatics, software, structural variation, misalignment, microbial genomics
Abstract

Over the last decade, the emergence of high-throughput sequencing has led to an increase in both the size and scope of genome sequencing projects. Although genome sequencing and analysis has changed dramatically during this time, the way read alignments are visualised has remained largely unchanged. To address the problem of visualising growing sequencing datasets, we have developed DiscoPlot, a tool for visualising read alignments using a two-dimensional scatterplot. DiscoPlot allows the user to quickly identify genomic rearrangements, misassemblies and sequencing artefacts by providing a scalable method for visualising large sections of the genome. It reads single-end or paired read alignments in SAM, BAM or standard BLAST tab format and creates a scatter plot of opaque crosses representing the alignments to a reference. DiscoPlot is freely available (under a GPL license) for download (Mac OS X, Unix and Windows) at https://mjsull.github.io/DiscoPlot.

Introduction

The emergence of high-throughput sequencing has led to an increase in both the size and scope of genome sequencing projects. Assembly and mapping visualisation tools, such as Tablet (Milne et al. 2013), BamView (Carver et al. 2010) and Savant (Fiume et al. 2010) have been developed to be able to handle the computational challenges presented by large volumes of data, however analysing even small genomes can be time consuming. Furthermore, misassemblies and structural rearrangements can be obfuscated by the large volume of data making it difficult to distinguish between discordant reads caused by structural variations and discordant reads caused by spurious mapping or read chimeras. The time consuming nature of visual analysis has led to a task originally accomplished manually, such as identifying structural variations or misassemblies, being automated. Many programs have been developed for identifying structural variations using alignment information such as SVDetect (Zeitouni et al. 2010), ForestSV (Michaelson and Sebat 2012), BreakDancer (Chen et al. 2009) and InGap-SV (Qi and Zhao 2011). Tools have also developed that make use of paired read alignment information to identify misassembled contigs, including REAPR (Hunt et al. 2013) and amosvalidate (Phillippy et al. 2013).
These tools may make assumptions that may not be true for all datasets. For example, high levels of read chimeras, highly variable coverage and bimodal insert sizes can all lead to overcalling or undercalling of structural variations. Problems also arise when the original structure and a structural variant are both present in the same sequencing run. Such sites where multiple structural variants exist are often biological significant, such as genes undergoing DNA invertase mediated prophage tail fibre allele switching (Forde et al. 2014). Thus there is a clear need for a scalable, visual approach that enables the user to quickly identify genomic rearrangements, misassemblies and sequencing artefacts.

A related problem has underpinned visualisation software development for Hi-C data (Servant et al. 2012). Hi-C probes the three-dimensional architecture of genomes by coupling proximity based ligation with high-throughput sequencing (Lieberman-Aiden et al. 2009). The Hi-C methodology produces large volumes of mate-pair sequenced reads with each paired read coming from separate but spatially close regions of the genome. Reads are then mapped back to a reference genome, and regions of the genome that physically interact can be inferred from the concentration of reads that map to a particular region. One way of accomplishing this is by binning reads in a 2-dimensional array whereby the x and y coordinates are determined by the mapping coordinates of the left and right read pair, and visualising the array using a heatmap. This approach provides a method of visualising large volumes of read alignments across an entire genome.

A two-dimensional layout provides a scalable method for visualising read alignments. Heatmaps, while good at visualising “low resolution” information, as in the case of Hi-C that involves interaction between large (>100,000 bp) regions of the genome, have several major drawbacks which make them unsuitable for visualising structural variations using paired-end reads. Bin number is limited by the resolution of the image being created. If the bin size is larger than a genomic rearrangement, a two-dimensional approach will not work. Single significant bins can be difficult to spot at high resolutions. In contrast, if a wide array of values are present in the bins it can be difficult to find a colour gradient to make significantly different values distinguishable from one another. This problem is magnified if colours are also needed to indicate other information, such as read orientation.
To solve these problems DiscoPlot has been developed. DiscoPlot is a Python application that uses matplotlib (Hunter 2007) to display read alignments as a scatter plot. It uses an opaque cross in which the size of the cross is proportional to the number of reads mapping to the bin. This approach has several inherent advantages over heatmaps. Representing counts as sizes allows colour to be used to display additional information such as read orientation. Markers allow single bins with a significant number of reads to easily be identified even when a large number of bins has been used. Anti-aliasing allows the user to use more bins than resolution permits while still retaining some accuracy obtained by using a small bin size. This approach has also been adapted to work with single-end long reads. The script can create figures from paired alignments using a SAM, indexed BAM file and single-end reads using and alignment in standard BLAST tab format. DiscoPlot is an open source project available on Windows, OSX and GNU/Linux.

**Implementation**

DiscoPlot allows genome structural variants, misassemblies, circularisation of the genome and sequencing artefacts such as read chimeras to be quickly identified. A reference genome of width $w$ base pairs is divided into $n$ bins of size $b$ base pairs. The user can either specify the number of bins to use or the size of the bins. Two sparse matrices $M_a$ and $M_b$ of size $n \times n$ are built to count direct and inverted alignments, and two arrays $A_a$ and $A_b$ of length $n$ are also created to count unaligned sequence.

**Paired-end sequencing matrix construction**

Read alignments are processed from a read-alignment file in standard SAM or BAM format (Li et al. 2009). If reads are orientated in opposite directions $M_a[(x/b)][(y/b)]$ is incremented by one, where $x$ is the position of the read aligning to the forward strand and $y$ is the position of its pair (Figure 1). If both ends of the read map to the forward strand $x$ is the position of the mate aligning closest to the 5’ end and $y$ is determined by the position of its pair. Similarly, if both ends of the read map to the reverse strand the $x$ is determined by which mate is closest to the 3’ end of the forward strand and the $y$ by its pair. In both cases $M_a[(x/b)][(y/b)]$ is incremented by one. The positions ($x$) of reads with unmapped mates are also tracked in separate arrays, $A_a[(x/b)]$ is incremented by one for each read that maps to the
forward strand with an unmapped pair. $A_d[x/b]$ is incremented by one for each read that maps to the reverse strand with an unmapped pair.

Figure 1: Creation of a DiscoPlot using mate-pair reads. In the above figure the pink rectangle represents the reference sequence. The linked arrows represent the left (blue) and right (orange)
pairs of mate-paired reads. The square underneath is the DiscoPlot representation of how the reads map. 1) DiscoPlot represents the chromosome in 2D space. 2) This reference is divided up into bins. 3) When a read maps to the reference DrawGrid represents it as a cross on the grid where the x coordinate of the cross is the start of the read aligning the forward strand maps to, and the y coordinate is determined by mapping position of the start of its pair. 4) Multiple reads can be represented in this way. 4) In the x axis each cross aligns with a read mapping to the forward strand. 5) In the y axis each cross aligns with a read aligning to the reverse strand. 6) The size of the cross represents how many read pairs map to that bin. 7) When mate-pair reads align concordantly with the reference sequence crosses occur at position x, y where y is x minus the insert size of the pair. 8) If read data is not concordant with the reference genome crosses occur in unexpected locations. Read pairs that align to the same strand are indicated by red crosses. 9) A representation of how region of the chromosome that excises into a circular product would look when mapped against the chromosomal reference.

**Single-end sequencing matrix construction**

Local alignments of each read to a reference sequence are provided by the user in standard BLAST tab format or generated automatically using NCBI-BLAST (Camacho et al. 2009). To reduce the number of spurious hits, alignments less than the user defined identity or length are ignored (default values 95% and 100 basepairs (bp), respectively). The alignment with the highest Bit-score becomes the primary alignment. If several alignments score equally one is chosen at random to prevent spikes at repetitive sites in the genome. If the entire read does not align in a single High Scoring Pair (HSP), subsequent alignments are iterated through from highest to lowest Bit-score. If the alignment includes a previously unaligned portion of the read it is included in a list of secondary alignments. Each alignment is defined by four positions \( Q_s, Q_e, R_s \) and \( R_e \). \( Q_s \) is the position in the read of the start of the alignment, \( Q_e \) is the position in the read of the end of the alignment, \( R_s \) is always less than \( Q_s \). \( R_s \) is the position in the reference of the start of the alignment, \( R_e \) is the position in the reference of the end of the alignment. If alignment is to the forward strand of the reference \( R_e \) is less than \( R_s \), if the alignment is to the reverse strand \( R_s \) is greater than \( R_e \). If the primary alignment is to the reverse strand of the reference, the primary
and secondary alignments for the read are converted so that they are the equivalent of those that would
be found by aligning the reverse compliment of the read. To reverse these alignments $Q_s$ becomes the $r$
- $Q_e$ and $Q_e$ becomes the $r$ - $Q_s$, where $r$ is the length of the read. $R_s$ and $R_e$ are switched. After these
values have been set an anchor ($a$) is set as $R_s$ from the primary alignment. Then, for each position ($i$)
in the length ($l$) of each alignment a set of $x$ and $y$ values are calculated where $x = \lfloor (Q_s + a + i) / b \rfloor$ and $y = \lfloor (R_s (i/l) + R_e (1-i/l)) / b \rfloor$. If the alignment is the primary alignment or in the same orientation as the primary
alignment each unique combination of $x$ and $y$ is incremented by one in $M_a$. If the alignment is not in
the same orientation as the primary alignment each unique combination of $x$ and $y$ is incremented by
one in $M_b$ (Figure 2). If the 5’ end of read mapping to the forward strand or the 3’ end of a read mapping
to the reverse strand doesn’t align to the reference $A_a[R_s/b]$ is incremented by one. Similarly, if the 3’
end of a read mapping to the forward strand or the 5’ end of a read mapping to the reverse strand
$A_a[R_e/b]$ is incremented by one. This process is repeated for all reads.
Figure 2: Creation of a DiscoPlot using single end reads. The reference genome is represented by a pink rectangle, which is divided into bins. Reads are represented by orange (forward) and blue (reverse) arrows. The square underneath is the DiscoPlot representation of how the reads map. 1) The anchor is set at the 5′ end of the primary alignment in the reference. A cross is drawn at bin X, Y where X and Y are equal to the bin containing the anchor. For each extra bin the read aligns...
to an additional cross is drawn. The X coordinate of the cross is the position in the read relative
to the anchor and is located on the X axis relative to the initial cross. The Y coordinate is the
position that section of the read aligns to in the reference. For concordant reads this creates a
straight diagonal. 2) Additional reads can be added, if two or more reads map to the same bin the
size of the cross is increased. 3) The blue rectangle represents the genome from which the reads
were generated and grey polygons represent how it aligns to the reference. The X position
increments by one for each additional bin the read maps to. The Y position corresponds to the
position that section of the read aligns to in the reference. This read has two alignments to the
reference, as the second alignment is orientated in the opposite direction of the primary alignment
crosses are drawn in blue. 4) Additional reads are added. The resulting DiscoPlot is similar to a
Dotplot representation of the genome from which the reads were generated and the reference.

**Drawing the Graph**

For each bin in $M_a$ an opaque red $\times$ is drawn with $x$ and $y$ coordinates equal to the $x$ and $y$ coordinates
of the bin multiplied by $b$. The width and height of the cross in points equals $\sqrt{\frac{2000c}{b \times m}}$ where $m$ is
the median value of the most populated diagonal in the matrix and $c$ is the count of alignments in that
bin. Bins from $M_b$ are drawn as blue opaque $\times$ symbols. Bins from $A_a$ are drawn along the x axis as
opaque green $+$ symbols and bins from $A_b$ are drawn along the y axis as opaque green $+$ symbols. Size,
thickness and opacity can all be adjusted by the user. Crosses are used because they allow deletions and
other features to become apparent even when visualising large sections of the reference genome. If more
than one reference is present in an alignment, alignments are separated by an opaque green band; this
allows potential integration events to be observed. Discoplot also allows the user to define regions of
the reference to view enabling precise visualisation of genomic rearrangements (Figure 4).
Discoplot uses matplotlib to display read mapping in an interactive window that can be zoomed and
scrolled to identify exact coordinates of markers. A subsection of the plot can be displayed;
alternatively, the plot can be broken up into a grid to highlight multiple subsections. Images can be
created from the GUI or written straight to an image from the command-line (PNG, SVG, PDF
supported on most systems, see matplotlib documentation). Reads associated with markers can be written to a separate BAM or FASTA file using the command-line.

**Case studies**

**Case study 1: simulated Illumina reads using a mock variant genome of uropathogenic *E. coli* UTI89**

A mock variant genome was created using the genome of *Escherichia coli* UTI89. *E. coli* str. UTI89 is an uropathogenic *E. coli* isolated from a patient with an acute urinary tract infection. It contains a 5,065,741 bp chromosome and an 114,230 bp plasmid. A 300bp inversion and a 3000bp inversion were created at bases 50,000..50,300 and 100,000..100,300. A 300bp insertion and a 3000bp insertion were created at bases 150,000 and 200,000. Bases in the ranges 250,000..250,300 and 300,000..303,000 were deleted. Bases 400,000..400,300 were moved to position 350,000 and bases 500,000..503,000 were moved to position 450,000. Paired-end reads with an insert size of 300bp were simulated at 200 times coverage from the mock genome using GemSIM (McElroy et al. 2012) and mapped to UTI89 using BWA (Li 2013). A DiscoPlot was generated from the resulting BAM file using a bin size of 100bp (Figure 3).
Figure 3: DiscoPlot of a mock genome. A mock genome was created by adding genomic rearrangements to the chromosome of *E. coli* str. UTI89. Paired-end reads generated from the mock genome (query) with GemSim and mapped back to UTI89 (reference). The first ~500 Kbp were then visualised using DiscoPlot.

1000bp single-end reads were also generated from the mock genome at 20 times coverage and mapped back to UTI89. Sizes of reads and structural variants were chosen to demonstrate how each structural...
variant is represented in the DiscoPlot when the read length or insert size is less than or equal to the structural variant or greater than the structural variant. A close up of each of the structural variants was also generated (Figure 4).

Figure 4: DiscoPlots of common structural variants. Each box shows a common genomic rearrangement represented by a DiscoPlot. Rows A and B were created using 100 bp long paired-end reads with an insert size of 300bp. Rows C and D were created using single-end reads with an average length of 1000bp. For each box the rearrangement in the sequenced genome is listed,
followed by the scale of the gridlines in brackets. A1, C1: 300 bp deletion (400 bp). A2, C2: 300
bp insertion (400 bp). A3, C3: 300 bp inversion (400 bp). A4, C4: 300 bp sequence translocated
50 Kbp upstream (10 Kbp). B1, D1: 3000 bp deletion (1000 bp). B2, D2: 3000 bp insertion (500
bp). B3, D3: 3000 bp inversion (1000 bp). B4, D4: 3000 bp sequence translocated 50 Kbp
upstream (10 Kbp).

Case study 2: real Illumina reads from sequencing uropathogenic E. coli UTI89

An Illumina HiSeq2000 101 bp paired-end sequencing run of *Escherichia coli* str. UTI89 (ENA acc:
ERR687901) was aligned to the publicly available reference genome (CP000243.1) and its plasmid
(CP000244). The alignment was then examined using DiscoPlot. Low quality reads were trimmed,
where possible, or filtered if the average per-base quality was less than 30 or had one pair trimmed to
less than 50 base pairs. In total, 2,433,934 high quality read pairs with an average insert size of 367.25
base pairs with a standard deviation of 59.1 were aligned using the BWA mem algorithm (Li 2013).
The resulting BAM file was then visualised using DiscoPlot using 100,000 bins revealing several
structural variations (Figure 5).
Figure 5: The dynamic nature of the genome of E. coli str. UTI89. Discoplot of paired-end reads from a clonal culture of UTI89 mapped back to the published reference chromosome and plasmid. Coordinates from 0 to 5,065,741 represent the chromosome of E. coli UTI89, coordinates ≥ 5,066,000 represent the plasmid of E. coli UTI89.

Sites with discordantly mapping reads were identified using DiscoPlot (Figure 6). Reads from these sites were randomly sampled and mapped back to the reference using BLASTn (Camacho et al. 2009) to confirm if they had been correctly aligned. Structural variants were inferred from context of region of the genome to which the reads mapped, mapping distance and orientation of the aligned reads. Local alignments of soft-clipped reads were performed to identify exact boundaries of structural variations.
**Figure 6: Discordant reads in E. coli str. UTI89.** a) Read alignment indicates inversion of bases 919,638..922,323. 12bp inverted repeat present at terminals of region. Start and stop of inverted region occurs in two probable tail fibre proteins. Two additional tail fibre assembly proteins are encoded within the boundaries of this region. Region is immediately downstream of a putative DNA invertase gene. b, f, h, i) Reads are misaligned as they map equally well in a concordant position. c) Read alignment indicates circularisation of bases 1,653,000..1,662,603. 17bp direct repeats present at terminals of this region. Region also encodes five putative phage-related membrane proteins, two putative phage proteins, three phage hypothetical proteins, four
hypothetical proteins and a single putative phage related secreted protein. Size of crosses indicates coverage of this region is higher than average. Only a single read (indicated by the cross, top left) indicates potential excision of this region. d) Read alignments indicate inversion of bases 2,109,690..2,114,003. Region contains ~100bp inverted repeat at terminals which encodes a tRNA. Region contains 3 hypothetical proteins and an additional tRNA identical to the repeats. A P4-phage integrase is present immediately downstream of the inversion. The lack of concordantly mapping reads at prophage boundary indicates that the inverted region has reached fixation in the population. e) Reads indicate inversion of bases 2,906,008..2,906,936. 15bp inverted repeats present at terminals of this region. The 3’ end of a putative tail fibre assembly gene is encoded by this region. g) Read alignments indicate inversion of bases 4,907,424..4,907,737. Regions has 9bp inverted repeat at terminals. It is located in a non-coding region between fimA and fimE which encode the type I fimbriae.

The majority of structural variants cannot be explained by misalignment of the reads or sequencing error and therefore have biological significance. Two sites of variation are potentially caused by tail fibre allele switching (Figure 6a and 6e), similar to that previously shown in the genome of uropathogenic E. coli ST131 strain EC958 (Forde et al. 2014). An inversion event occurred at a site with several tRNA (Figure 6d). At another site, phase variation of a fimbriae loci was occurring (Figure 6g). A circularising region was also identified (Figure 6c), this region was present in the sequencing data at much higher coverage (243.1× coverage compared to 88.3× the average coverage of the chromosome). It is possible that this phage was induced in a small fraction of the bacteria. Additional analysis, such as PCR or long read sequencing, is needed to confirm the exact nature of the identified structural variants.

In addition to the structural variants detected, reads misaligned to several repetitive regions (Figures 6b, 6f, 6h, 6i). Misaligned reads all had insert sizes much smaller than the average, which likely explains why they had been misplaced by BWA mem. (Figure 7).
Figure 7: Misaligned reads and insert size. Blue columns indicate frequency of the mapping distances of all aligned reads. Each vertical red line indicates the corrected mapping distances of a misaligned read from site b (Figure 6).

For each batch, BWA mem estimates the mean and variance of insert size distributions from reliable single-end hits. BWA then uses a normal distribution, determined by these values, to estimate the likelihood of each of the possible pairings of reads with multiple alignments. This value is combined with the alignment scores of the reads to determine the probable alignment of a read pair. If the putative insert size of an alignment varies significantly from the mean BWA considers the read a chimeric pair; BWA considers all chimeric alignments of a read pair equally likely. As insert sizes from our sequencing run were not normally distributed, BWA considers the reads with small insert sizes chimeric and arbitrarily assigns them to one of the instances of a repeat. This resulted in the misalignment of reads with small insert sizes at large exact repeats.
Conclusions

Discoplot provides a simple, scalable method for visualising read alignments to a reference. It allows the user to quickly identify genomic rearrangements, misassemblies and sequencing artefacts. A visual approach is inherently sequence type agnostic, this allows the user to immediately analyse their alignments without calculating the average length or standard deviation of their read library. This is particularly useful when analysing complex libraries such as mate-pair reads that may contain a shadow library of read pairs in the reverse orientation. We have shown how DiscoPlot can be used to quickly identify genome rearrangements and misaligned reads in bacterial genomes. In future, additional colours may be incorporated into to display other metrics such as average number of single nucleotide variants per read in a bin. To make DiscoPlot more accessible, features only available through the use of the command line could be incorporated into a graphical user interface.

Acknowledgments

The authors would also like to thank Professor Mark Schembri for making available the UTI89 Illumina datasets used in this project. We would also like to thank Mitchell Stanton-Cook for helping make DiscoPlot.py available through PyPI.

Funding

SAB is supported by a Career Development Fellowship from the National Health and Medical Research Council Career of Australia [grant number APP1090456].

References


Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC,
Fiume M, Williams V, Brook A, Brudno M. 2010. Savant: genome browser for high-throughput
Forde BM, Ben Zakour NL, Stanton-Cook M, Phan M-D, Totsika M, Peters KM, Chan KG, Schembri
MA, Upton M, Beatson SA. 2014. The Complete Genome Sequence of *Escherichia coli*
EC958: A High Quality Reference Sequence for the Globally Disseminated Multidrug
90-95.
Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. In
The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*
**25**(16): 2078-2079.
Lieberman-Aiden E, van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie
BR, Sabo PJ, Dorschner MO et al. 2009. Comprehensive mapping of long-range interactions
McElroy K, Luciani F, Thomas T. 2012. GemSIM: general, error-model based simulator of next-
Michaelson JJ, Sebat J. 2012. forestSV: structural variant discovery through statistical learning. *Nat
Tablet for visual exploration of second-generation sequencing data. *Briefings in
Phillippy AM, Schatz MC, Pop M. 2008. Genome assembly forensics: finding the elusive mis-

Qi J, Zhao F. 2011. inGAP-sv: a novel scheme to identify and visualize structural variation from

2843-2844.

2010. SVDetect: a tool to identify genomic structural variations from paired-end and mate-