Published online 19 April 2013

Nucleic Acids Research, 2013, Vol. 41, No. 11 ***e118***
doi:10.1093/nar/gkt254

Meander: visually exploring the structural variome using space-filling curves

Georgios A. Pavlopoulos[1,2,3,*], Parveen Kumar[4], Alejandro Sifrim[1,2], Ryo Sakai[1,2], Meng Lay Lin[5], Thierry Voet[4,5], Yves Moreau[1,2] and Jan Aerts[1,2,*]

[1]Department of Electrical Engineering (ESAT/SCD), University of Leuven, Kasteelpark Arenberg 10, Box 2446, 3001 Leuven, Belgium, [2]iMinds Future Health Department, University of Leuven, Kasteelpark Arenberg 10, Box 2446, 3001 Leuven, Belgium, [3]Division of Basic Sciences, University of Crete, Medical School, Heraklion, 71110 Crete, Greece, [4]Laboratory of Reproductive Genomics, Department of Human Genetics, University of Leuven, Herestraat 49, 3000 Leuven, Belgium and [5]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton - Cambridge, CB10 1SA, UK

Received November 26, 2012; Revised and Accepted March 19, 2013

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

ABSTRACT

The introduction of next generation sequencing methods in genome studies has made it possible to shift research from a gene-centric approach to a genome wide view. Although methods and tools to detect single nucleotide polymorphisms are becoming more mature, methods to identify and visualize structural variation (SV) are still in their infancy. Most genome browsers can only compare a given sequence to a reference genome; therefore, direct comparison of multiple individuals still remains a challenge. Therefore, the implementation of efficient approaches to explore and visualize SVs and directly compare two or more individuals is desirable. In this article, we present a visualization approach that uses space-filling Hilbert curves to explore SVs based on both read-depth and pair-end information. An interactive open-source Java application, called *Meander*, implements the proposed methodology, and its functionality is demonstrated using two cases. With *Meander*, users can explore variations at different levels of resolution and simultaneously compare up to four different individuals against a common reference. The application was developed using Java version 1.6 and Processing.org and can be run on any platform. It can be found at http://homes.esat.kuleuven.be/~bioiuser/meander.

INTRODUCTION

Recent advances of next generation sequencing technologies (1,2) allow the identification of both balanced (inversions, translocations) and unbalanced (deletions, duplications) structural variations (SVs) in the genome. The identification and characterization of such variations is of high importance in current genomic research, as it has been shown that many of them play a significant role in various disorders such as cancer (3). Currently, there are several possible ways to identify and discover SVs in the genome using different types of genomic data (4). First, *read-depth* or *depth-of-coverage* can be used to infer the relative copy number of genomic regions when compared with a reference sample. Second, the relative mapped position of read-pair members, known as *paired-end mapping*, can be used to find deletions, tandem duplications, inversions and intra-chromosomal signatures. Finally, reads that span a DNA breakpoint in the sample appear as split reads when mapped to the reference genome. Several variant callers based on read-depth, pair-end or their combination already exist and are extensively reviewed by Alkan and colleagues (5). Such callers store the results along with the genomic information in flat files that are difficult to process and interpret. Because such genomic data sets range in scale from thousands to millions of data points covering multiple gigabases of sequence, visualization approaches need to cope with such a high complexity and play a key role in revealing patterns of variation and relationships between experimental data sets.

Although most of the current visualization tools focus on interpreting and annotating genomic data, only few of them are designed for data exploration to generate new knowledge and new hypotheses. Genome browsers, such as the *Ensembl* (6), *UCSC* (7), *GBrowse* (8), *Integrative Genomics Viewer* (9) and *Integrated Genome Browser* (10), have been developed to support the visualization of genomic contexts and plot data in a linear form along with annotations, genomic features, scores and positions. Other tools such as *Circos* (11), *Gviz* (12), *GenomeGraphs* (13),

*To whom correspondence should be addressed. Tel: +30 2810 394518; Fax: +32 16 321960; Email: g.pavlopoulos@med.uoc.gr
Correspondence may also be addressed to Jan Aerts. Tel: +32 16 321053; Fax: +32 16 321960; Email: jan.aerts@esat.kuleuven.be

© The Author(s) 2013. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

ggbio (14), *Apollo* (15), *HilbertVis* (16), *GenomeComp* (17), *Seevolution* (18), *Spark* (19), *Gremlin* (20) and *In-GAVsv* (21) are developed to tackle more targeted questions, such as visualization of genomic data using 2D plots, graph-based layouts and most often linear representations. *HilbertVis* (16) and *DHPC* (22) are the tools most closely related to our work, as they implement space-filling Hilbert curves to show genomic data at higher resolutions. Although these tools come with significant advantages, many of them suffer from lack of interactivity, ability to explore data at different resolutions or multi-sample comparison; yet, they have many library dependencies.

In this work, we present *Meander*, an application that combines two different types of visualization to capture inter/intra-chromosomal SVs based both on read-depth and pair-end data. In the first type of visualization, a single chromosome is presented linearly at low resolution as a horizontal line like in most current genome browsers. In the second type of representation, a Hilbert space-filling curve is used to visualize a chromosome in a 2D panel at a much higher level of detail using a folded (snake-like) continuous vector of $512 \times 512 = 262\,144$ pixels. This high resolution allows visual detection of much smaller SV. *Meander* can also simultaneously compare up to four samples against one common reference genome. It comes with a variety of interactive filters that make interactive data exploration easier and the extraction of patterns more targeted and efficient. In addition, *Meander* highlights variations that are supported by double evidence of read-depth and pair-end signals to make unknown variation easier to detect. Although the concept of the space-filling curves can be used to highlight various genomic characteristics like for example in (23) where a Hilbert curve is used to illustrate chromatin organization features in Drosophila, the main aim of *Meander* is the identification and annotation of SVs.

MATERIALS AND METHODS

The Hilbert curve

The theory of space-filling curves was first developed by the mathematician Peano in 1890 (24). A space-filling curve is a continuous mapping from a lower-dimensional space into a higher-dimensional one (two-dimensions in the case of *Meander*). A useful property of a space-filling curve is that it visits all points in a region once it has entered that region and points that are close together in the original curve will be close together in the plane. Although the inverse is not true, points that are close to each other in the plane tend to be close to each other in the original curve. One of the most used curves was proposed by Hilbert in 1891 (25), who gave the first geometrical interpretation. The Hilbert space-filling fractal curve visits every point in a square grid with a size of $2^N \times 2^N$ ($N > 0$). Therefore, the points that belong to the Hilbert curve are always 2^{2N} in number, where N denotes the fold level. The curve, owing to its fractal geometry, always splits an area into quarters, a procedure that can iteratively continue to infinity. Figure 1A shows the folding of the curve across eight iterations for a plane of $2^9 \times 2^9 = 512 \times 512 = 262\,144$ pixels. Notably, for fold-level $N = 9$, every single pixel of the plane is covered. Although a Hilbert curve can be generated for any number of dimensions, in this article, we use a 2D Hilbert curve to represent one chromosome at a time (9).

Implementation of the Hilbert curve

Let $L = \{t|0 \leq t \leq 512\}$ denote the unit interval and $Q = \{(x,y)|0 \leq x \leq 512, 0 \leq y \leq 512\}$ denote the unit square. For each positive integer n, the interval L is partitioned into 4^n subintervals of length 4^{-n} and the square Q into 4^n subsquares of side 2^{-n}. The procedure is calculated recursively, and a one-to-one correspondence between the subintervals of L and the subsquares of Q is constructed so that adjacent subintervals correspond to adjacent subsquares. If the subinterval L_{nk} corresponds to a subsquare Q_{nk} at the n-th partition, then the four subintervals of L_{nk} must correspond to the four subsquares of Q_{nk} at the $(n+1)$-st partition. The implementation of the algorithm in Processing.org is shown in Figure 1B.

Application of the Hilbert curve to genomic data

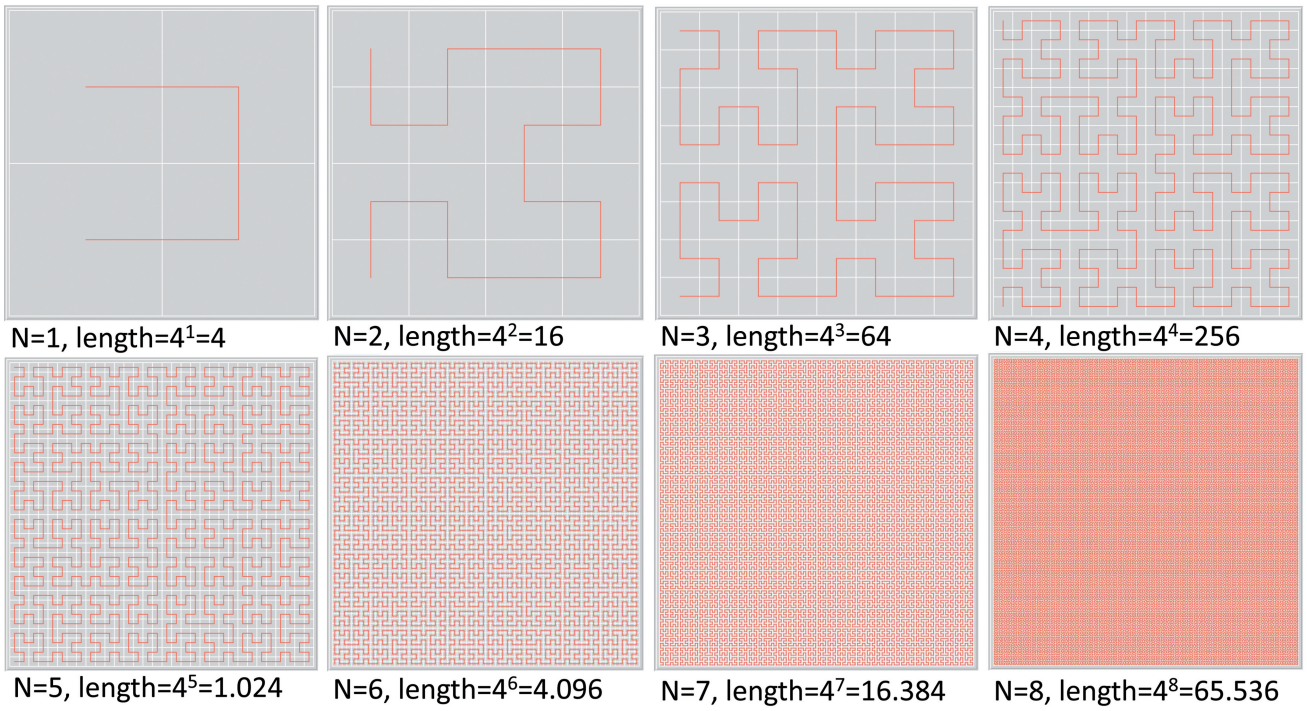
Use of the Hilbert curve

Although many different visualization approaches to represent a genome have been proposed (26,27), in this article, we use the continuous fractal Hilbert space-filling curve to visualize a vector with millions of elements such as human chromosome 1 (~249 000 000 bases) mainly for two reasons. First, from a visual encoding point of view, two loci that are close to each other on the chromosome will be displayed close to each other in the plane, an important characteristic of the Hilbert curve that does not break the linear properties of a vector. The second and most important reason is the resolution gain to browse data at a higher level of detail. For example, given such a vector, a linear plot on a screen of, say, 1000–1200 pixels wide can only depict a whole chromosome at very low resolution. On the contrary, as a Hilbert curve can fold on a 2D plane of $512 \times 512 = 262\,144$ pixels in size, we achieve a gain of ~250 times in resolution (Figure 2A). Areas with high signal values, which may be overlooked in the linear representation owing to the low resolution appear as bigger dense blocks of many pixels in the Hilbert curve.

Bucketing and colour mapping

Given a linear vector of millions of values such as the read-depth signal for a chromosome, we first split this vector into bins of equal size according to the number of pixels available. In a second step, an average signal value of the coverage signal is calculated for each of these segments and assigned to the corresponding bin. For example, in a linear representation (i.e. 1200 pixels), the human chromosome 1 (~249 000 000 bases) will be split into 1200 bins and plotted at a ~243 165 bases/pixel resolution, whereas in a Hilbert view (512×512 pixels), the chromosome will be split into 262 144 bins and plotted at a much higher resolution of 950 bases/pixel (Figure 2A), effectively gaining a ~256-fold increase in resolution. Although in a linear plot, the height of the

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016




```
ArrayList <PVector> points = new ArrayList();
void setup(){
 size(512,512); //create a frame of 512x512 pixels
}
void draw(){
 int fold_level=2; //Fold level 9 covers every pixel of the plane
 hilbert(0,0,512,0,0,512,fold_level); //calculating Hilbert Curve recursively
 for (int i=1;i<points.size();i++)
  line(points.get(i-1).x, points.get(i-1).y, points.get(i).x, points.get(i).y);
}
// xx and yy are the coordinates of the bottom left corner
// xi & xj are the i & j components of the unit x vector of the frame
void hilbert(int xx, int yy, int xi, int xj, int yi, int yj, int n)
{ if (n <= 0)
    points.add(new PVector( xx + (xi + yi)/2, yy + (xj + yj)/2 ));
  else
   {  hilbert(xx,           yy,           yi/2, yj/2,  xi/2,  xj/2, n-1);
      hilbert(xx+xi/2,      yy+xj/2 ,     xi/2, xj/2,  yi/2,  yj/2, n-1);
      hilbert(xx+xi/2+yi/2, yy+xj/2+yj/2, xi/2, xj/2,  yi/2,  yj/2, n-1);
      hilbert(xx+xi/2+yi,   yy+xj/2+yj,  -yi/2,-yj/2, -xi/2, -xj/2, n-1);
   }
}
```

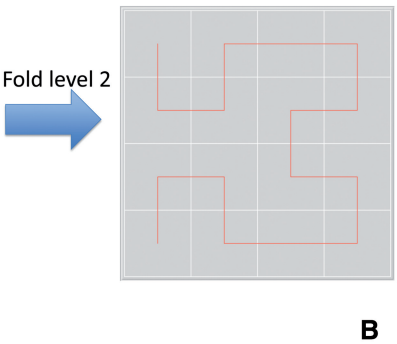


Figure 1. Folding levels of a Hilbert curve. The number of the edges of the Hilbert curve is 4^N, where N denotes the fold level. For a canvas of $2^9 \times 2^9 = 512 \times 512 = 262\,144$ pixel dimension, the fold level $N = 9$ covers every pixel of the plane.

bar represents the strength of the bucket signal, in the Hilbert representation, each pixel is assigned a colour with scaled transparency according to the coverage (Figure 2B). Thus, the darker the pixel colour appears, the higher the coverage is and vice versa. Notably, white areas indicate zeros as coverage or absence of data as the length of a chromosome does not follow the required 2^{2N} length for a Hilbert curve. In the case of sequencing gaps where the coverage value is zero, we assign a white RGB (255 255 255) colour to the corresponding pixels. For example, such coverage gaps are often observed in chromosomal regions such as the centromere, where, often, no DNA sequence is defined. In the second case of absence of data, a white colour is assigned to the pixels of the Hilbert curve, which do not hold any information and do not correspond to any of the chromosome parts. Notably, these pixels always appear in the bottom left corner of the panel where the Hilbert curve finishes. Such behaviour is expected, as a Hilbert curve's length does not correspond to the physical chromosome length. A Hilbert curve should always have a defined length of 4^N, $N > 0$, whereas a physical chromosome does not obey this mathematical rule.

Comparing two plots

As it is difficult to visually observe signal losses or gains when comparing two different Hilbert plots of two samples (Figure 2C), the $log_2(sample/reference)$ ratio between the two individuals is used for a direct comparison. When the signal of a sample is higher than the signal

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

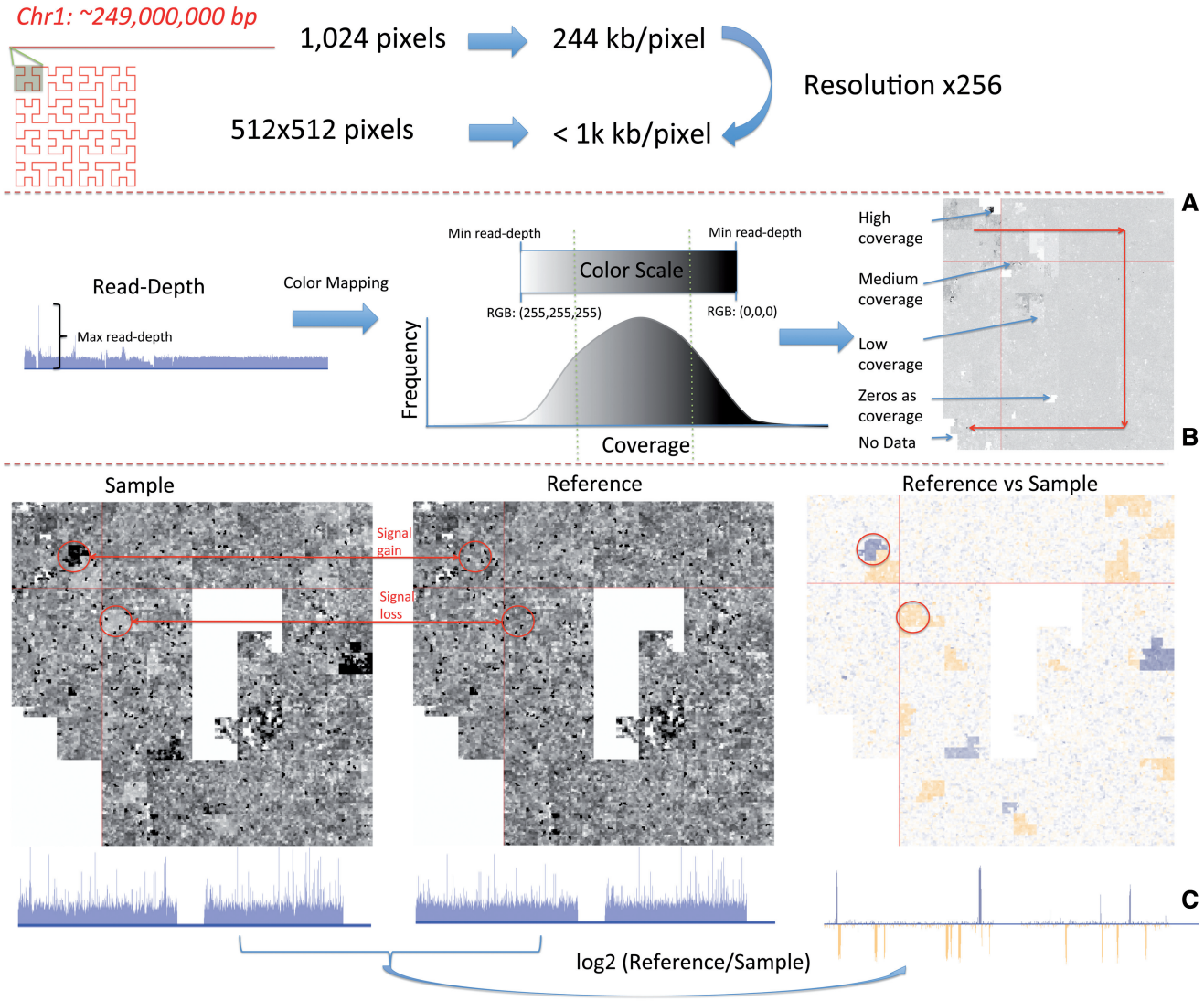


Figure 2. Space filling curves in genomic data. (**A**) Resolution gain comparing the linear with the *Hilbert* representation. (**B**) Colour mapping: The transparency of the colour is adjusted according to the signal value. Dark areas indicate high coverage; light grey areas lower coverage. White areas indicate zero coverage or absence of data. The red arrows show the coordinate system of the system curve. (**C**) Comparison of a sample against a reference: *Left*: The sample and reference human chromosome 1 in both a *Hilbert* and a linear representation. *Right*: The log2 ratio between the reference and the signal. Blue signals indicate possible tandem duplications as reference < sample, and yellow blocks indicate possible deletions as reference > sample.

of the reference, a blue colour is assigned by default, and the transparency of the colour is adjusted according to the ratio's absolute value. Similarly, when the signal of the sample is lower than the signal of the reference, a yellow colour with adjusted transparency is assigned to each pixel. In both cases, the higher the absolute value of the ratio, the darker the colour (Figure 2C).

Meander application

Read-pair and pair-end data

Meander supports visualization of SVs based both on read-pair and pair-end data. In the linear representation, the bar height indicates the value of the $log_2(sample/reference)$ ratio. Negative ratios (red pixels) indicate possible deletions in the sample, whereas positive ratios (blue pixels) indicate possible duplications. Aberrantly, mapped pair-end data can indicate the presence of balanced as well as unbalanced variations. *Meander*, therefore, also links these together, both in the Hilbert and the linear views. Because the Hilbert curve only displays a single chromosome at a time, these links cannot be shown in cases where the partners of a paired-end lie on different chromosomes. To solve this issue, the whole genome split in chromosomes is schematically represented as a rectangle, wrapped around the main Hilbert plot, (see left part of Figure 3B), to allow direct linking between the position of the one paired end that corresponds to the loaded chromosome and the other paired end that corresponds to another chromosome of the same organism.

The graphical user interface

Meander uses four smaller panels to hold the read-depth and pair-end information for up to four different samples.

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

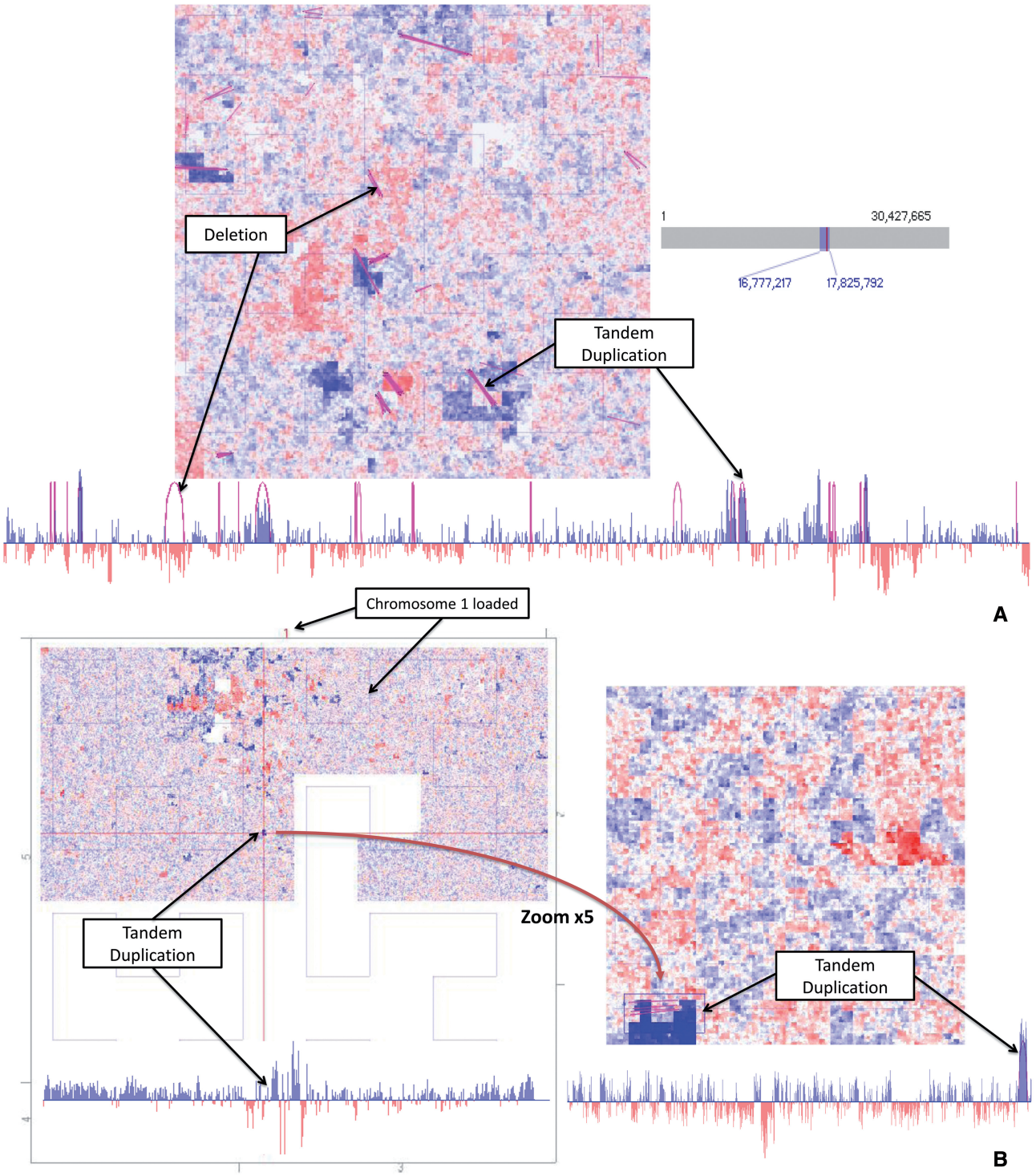


Figure 3. Comparison of chromosome 1 between strain ICE153 from central Asia and strain ICE97 from southern Italy. (**A**) An example of a deletion and a tandem duplication supported by both pair-end and read-depth information. (**B**) The advantage of the Hilbert representation. *Left*: A tandem duplication that is not visible in the linear representation (1 pixel length) but very clear in the Hilbert representation as a bigger block. *Right*: The same tandem duplication at zoom level 5 supported both by read-depth and pair-end evidence.

Any of these panels can be selected to be the focus and displayed at higher resolution. Although the smaller panels always represent information at the lowest zoom level, users have the ability to zoom-in at five different zoom levels to visualize the sample that is loaded on the main panel. Indicators highlight the zoomed areas in a whole chromosome view. Different views in the application are linked so that the position of the cursor in one view is reflected in the others. Finally, one can call the USCS genome browser at any time to see the relevant locus-specific information for a certain position.

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

Dynamic filtering

The *Meander* application comes with various dynamic filters. One can for example select variations by type, keep only the pair-ends with respective mapping distance within a given interval, hide any coverage below a certain threshold or hide log_2 ratios outside a selected interval. In addition, a Bezier curve in the linear plot or a straight line in the Hilbert plot might indicate more than one overlapping pair-ends that cluster together. One can filter these pair-ends according to the number of the pair-ends that form a cluster. For the read-depth information, often one cannot distinguish between the actual signal and the background noise. Therefore, a dynamic filter can adjust the brightness and the contrast of the image of the main panel. As a result, regions with lower intensities are hidden owing to the contrast and more dense regions with high intensities, often indicating an SV, remain highlighted. Finally, one of the strong features of *Meander* is its ability to highlight the regions that are supported by double evidence both from read-depth and pair-end information, providing more confidence about a variation. In cases where a variation is supported only by a pair-end signal or only by a read-depth signal, this might not be sufficient.

Input files

Meander accepts simple tab delimited text files, holding information about the read-depth signals and the pair-end information. Before launching *Meander*, read-depth pileup files, which can vary in size from less than one to several gigabytes depending on the chromosome length, cannot be handled directly and should be pre-processed to compute the relevant coverage information at the five different zoom levels. Therefore, sufficient disk space must be available. Each file that holds information about the average coverage samples per bucket for a Hilbert quarter at any zoom level consists of $512 \times 512 = 262\,144$ lines. Such a file has an average size of 15 MB. Three hundred forty-one of such files are required, as 1 file is necessary for zoom level one, 4 files for zoom level two, 16 files for zoom level three, 64 files for zoom level four and 256 files for zoom level five. This will require on average $15 \times 341 \approx 5$ GB of extra disk space. In addition, raw data files often contain gaps and do not provide any coverage information about every single position of the chromosome. Therefore, *Meander* initially creates an intermediate file containing the coverage signal for every single position of the chromosome to fill these gaps. Chromosomal positions of no coverage are assigned to zero as coverage. This intermediate file can often be double in size compared with the initial file, depending on how prominent the gaps are. This could substantially increase the disk requirements if one wants to pre-process a whole genome, chromosome by chromosome. The pre-processing step is often time expensive depending on the chromosome length but needs to be done only once. On average, 20 min are required to pre-process human chromosome 20 on a single CPU, ~1 h for human chromosome 1 or ~18 h for the whole human genome. Pre-processing is done by running *Meander* application separately in command line, and pre-processed files are available for download on the web site. In terms of memory requirements, *Meander* requires ~1G of RAM to run, as dynamic data structures like hash tables, array lists and interval trees continuously synchronize mouse coordinates with the Hilbert and linear views.

RESULTS

Case study 1: *Arabidopsis thaliana* strains

To demonstrate the functionality of *Meander* and its depiction of combined pair-end and read-depth information as double evidence of possible SVs, we compare two *A. thaliana* strains (ICE97 and ICE153) from the 1001 Genomes Project (28). *Strain ICE153* was collected from Central Asia and sequenced to a depth of 21X; s*train ICE97* was collected from Southern Italy and sequenced to coverage of 19X. We aligned both to the TAIR10 version of the *A. thaliana* reference genome using BWA (29) at default settings and converted file formats using Samtools (30). We then extracted read-depth information from the resulting pileup files and pairing information from the BAM files using a custom bash script. The pair-ends presented here are at least 20 bp in length.

Figure 3A shows an example of a tandem duplication and deletion supported both by strong read-depth signals and pair-end information. Pair-ends are visualized as straight lines in the Hilbert representation and as Bezier curves in the linear representation.

The tandem duplication in Figure 3B shows the advantage of the Hilbert representation over the linear plot. Although the specific tandem duplication is not visible in the linear plot owing to the low resolution (only 1 pixel in length), it pops up as a bigger highlighted block in the Hilbert representation. On the right, both read-pair and pair-end evidence about the specific duplication are presented at a higher zoom level. This indicates that investigation of read-depth and paired-end data through Hilbert curves has a distinct added value to only using linear representations, especially when concerning small SVs.

Case study 2: Breast cancer in human

Acquisition of mutations plays a key role in the origin and progression of cancer (31,32). Large-scale sequencing of whole cancer genomes is revealing an unexpectedly diverse array of mutational profiles, hinting at considerable underlying complexity in somatic mutation processes. High-throughput sequencing generates a huge amount of data, which is difficult to manage and visualize at genome-wide level. To understand the chromosomal instability and genetic changes that are acquired during cell expansion, we compare four single-cell derived subclones of the human breast cancer cell line HCC38.

The *Meander* tool can compare multiple samples simultaneously, at high resolution. Using *Meander*, we can demonstrate the *de novo* changes occurring in the cell, by simultaneously comparing the four subclones against the PD4198b reference genome and detect variations that are unique in any sample or variations that are common in all cells. Figure 4 shows a unique tandem duplication and deletion present in subclone B8FF4C and not present in

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

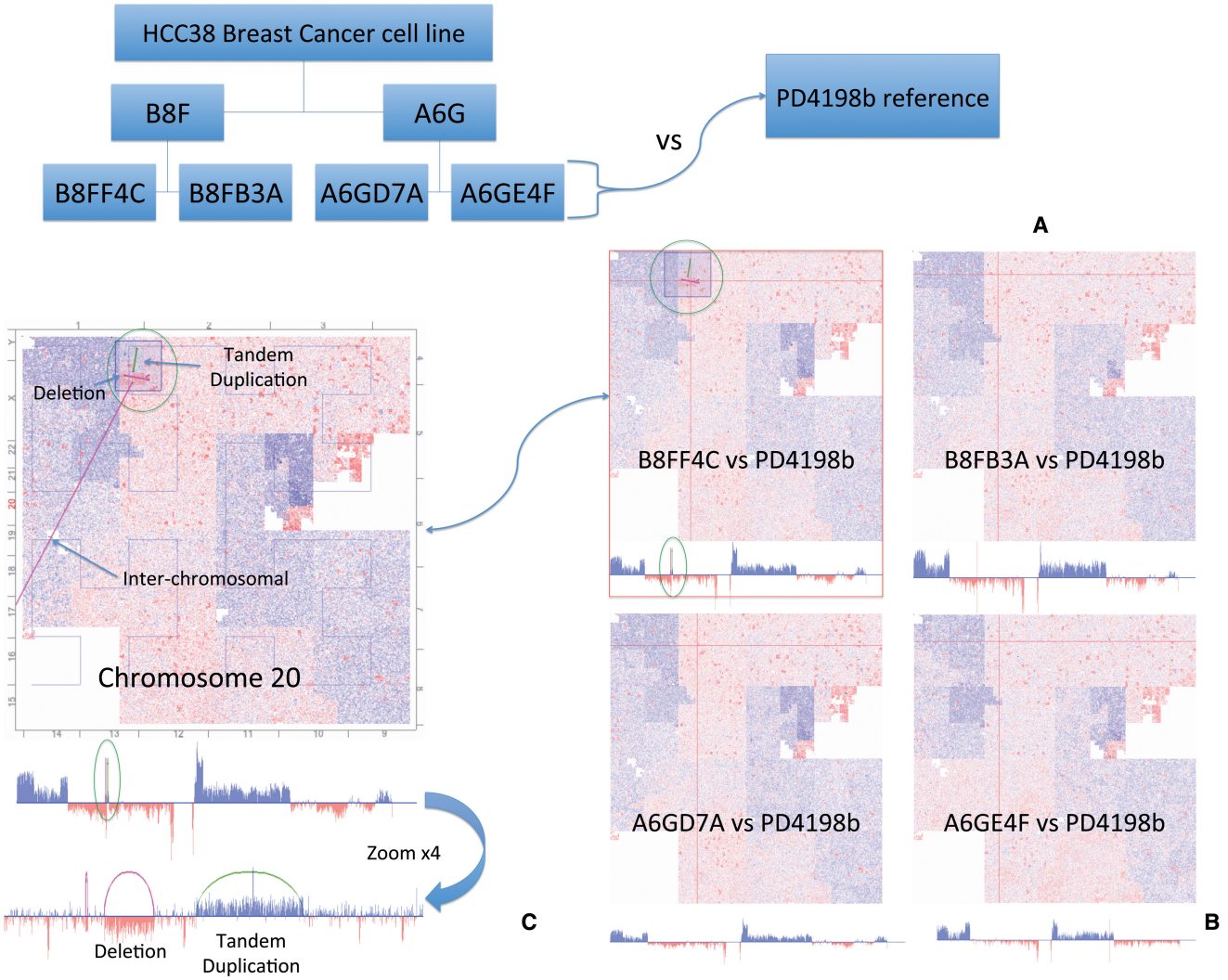


Figure 4. The unstable nature of HCC38. (**A**) Hierarchy of the single-cell derived subclones and comparison with the PD4198b reference genome. (**B**) Comparison between the four subclones against the PD4198b reference genome. Subclone B8FF4C demonstrates a *de novo* tandem duplication and flanking deletion not present in the other subclones. (**C**) Visualization of an inter-chromosomal variation (linked to the q-arm of chromosome 17), a unique deletion and tandem duplication around position 15 200 000 for chromosome 20 not present in the other subclones.

all the other subclones. Evidence about the specific variation is supported both by pair-end and read-depth information.

All four subclones were subjected to low coverage paired-end sequencing where sequencing libraries were prepared according to standard protocol (33,34), and both the ends of DNA-fragments were sequenced on Illumina GAII. Reads were aligned using BWA to the human reference genome GRCh37. The aberrantly mapping read-pairs which can map to alternative locations as a proper-pair were removed. Furthermore, aberrantly mapping read-pairs were sifted against (a) mitochondrial sequence, (b) repeats, (c) known BWA read-pileup regions, and (d) putative germline variants.

DISCUSSION

The field of data visualization covers a wide range of applications, ranging from interactive exploratory visualizations that aid in hypothesis generation to explanatory ones, where a clear message has to be communicated. *Meander* is located on the explanation side of this axis, allowing the researcher to visualize raw data to assess the performance of automated SV calling algorithms or to identify unexpected patterns. It supports visualization of both read-pair and pair-end data and shows genomic signals and signatures at a high resolution. It is highly interactive and comes with many dynamic filters to make the exploration of data easier. Every chromosomal position is linked to the UCSC genome browser and a dynamic navigation system is implemented to help the user orient himself. *Meander* currently supports cross-sample comparisons of up to four samples against a common reference and is a very strong tool for the exploration of *de novo* variation, as variation that is supported by both read-pair and pair-end information can be automatically highlighted.

Although the representation of genomic data with the use of Hilbert curves has already been demonstrated

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

(15,22), the application implements a visualization approach to explore SVs based on read-depth and pair-end data. Although *HilbertVis* is developed for ChIP-chip analysis and *DHPC* for representing chromatin rearrangements, both of them could potentially be used for the detection and exploration of SV, albeit based on read-depth only. In contrast, the interactive *Meander* application shows variations based both on pair-end and read-depth information. In addition, *Meander* goes one step further by enabling simultaneous comparison on four different samples against a reference. It also comes with dynamic filters for read-depth signal, pair-end distance and ratio intervals. Finally, overlay of predicted variations from external variant callers, switching between different views (sample, reference, ratio, read-depth, pair-ends or combination of those) and visualizations (Hilbert and linear) are strong features currently not supported by other applications.

In terms of further development, we plan to extend *Meander's* functionality to include a whole-genome view and allow multiple zooming, as one could be interested in looking at several different regions of a genome simultaneously. In addition, methodologies will be developed and improved on to speed up pre-processing of the data and to internalize SV calling algorithms.

Overall, we believe that *Meander* can stand as a powerful tool in the field of comparative genomics, as well as in aiding in evaluating the quality of predicted SVs towards personalized medicine and in discovering new ones that might be causative for genetic disorders.

ACKNOWLEDGEMENTS

The authors would like to thank Jun Cao, Joffrey Fitz, Karl J Schmid and Detlef Weigel for the *A. thaliana* plant data.

FUNDING

iMinds [SBO 2012]; University of Leuven Research Council [SymBioSys PFV/10/016, GOA/10/009]; and European Union Framework Programme 7 [HEALTH-F2-2008-223040 'CHeartED']. G.A.P. was financially supported by the European Commission FP7 programme 'Translational Potential' (TransPOT; EC contract number 285948). A.S. was supported by IWT [IWT-SB/093289]. Funding for open access charge: KU Leuven.

Conflict of interest statement. None declared.

REFERENCES

1. Metzker,M.L. (2010) Sequencing technologies - the next generation. *Nat. Rev. Genet.*, **11**, 31–46.
2. Cullum,R., Alder,O. and Hoodless,P.A. (2011) The next generation: using new sequencing technologies to analyse gene regulation. *Respirology*, **16**, 210–222.
3. Stankiewicz,P. and Lupski,J.R. (2010) Structural variation in the human genome and its role in disease. *Annu. Rev. Med.*, **61**, 437–455.
4. Medvedev,P., Stanciu,M. and Brudno,M. (2009) Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods*, **6**, S13–S20.
5. Alkan,C., Coe,B.P. and Eichler,E.E. (2011) Genome structural variation discovery and genotyping. *Nat. Rev. Genet.*, **12**, 363–376.
6. Hubbard,T.J., Aken,B.L., Ayling,S., Ballester,B., Beal,K., Bragin,E., Brent,S., Chen,Y., Clapham,P., Clarke,L. *et al.* (2009) Ensembl 2009. *Nucleic Acids Res.*, **37**, D690–D697.
7. Fujita,P.A., Rhead,B., Zweig,A.S., Hinrichs,A.S., Karolchik,D., Cline,M.S., Goldman,M., Barber,G.P., Clawson,H., Coelho,A. *et al.* (2011) The UCSC genome browser database: update 2011. *Nucleic Acids Res.*, **39**, D876–D882.
8. Stein,L.D., Mungall,C., Shu,S., Caudy,M., Mangone,M., Day,A., Nickerson,E., Stajich,J.E., Harris,T.W., Arva,A. *et al.* (2002) The generic genome browser: a building block for a model organism system database. *Genome Res.*, **12**, 1599–1610.
9. Robinson,J.T., Thorvaldsdottir,H., Winckler,W., Guttman,M., Lander,E.S., Getz,G. and Mesirov,J.P. (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.
10. Nicol,J.W., Helt,G.A., Blanchard,S.G. Jr, Raja,A. and Loraine,A.E. (2009) The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics*, **25**, 2730–2731.
11. Krzywinski,M., Schein,J., Birol,I., Connors,J., Gascoyne,R., Horsman,D., Jones,S.J. and Marra,M.A. (2009) Circos: an information aesthetic for comparative genomics. *Genome Res.*, **19**, 1639–1645.
12. Helaers,R., Bareke,E., De Meulder,B., Pierre,M., Depiereux,S., Habra,N. and Depiereux,E. (2011) gViz, a novel tool for the visualization of co-expression networks. *BMC Res. Notes*, **4**, 452.
13. Durinck,S., Bullard,J., Spellman,P.T. and Dudoit,S. (2009) GenomeGraphs: integrated genomic data visualization with R. *BMC Bioinformatics*, **10**, 2.
14. Yin,T., Cook,D. and Lawrence,M. (2012) ggbio: an R package for extending the grammar of graphics for genomic data. *Genome Biol.*, **13**, R77.
15. Lewis,S.E., Searle,S.M., Harris,N., Gibson,M., Lyer,V., Richter,J., Wiel,C., Bayraktaroglir,L., Birney,E., Crosby,M.A. *et al.* (2002) Apollo: a sequence annotation editor. *Genome Biol.*, **3**, RESEARCH0082.
16. Anders,S. (2009) Visualization of genomic data with the Hilbert curve. *Bioinformatics*, **25**, 1231–1235.
17. Yang,J., Wang,J., Yao,Z.J., Jin,Q., Shen,Y. and Chen,R. (2003) GenomeComp: a visualization tool for microbial genome comparison. *J. Microbiol. Methods*, **54**, 423–426.
18. Esteban-Marcos,A., Darling,A.E. and Ragan,M.A. (2009) Seevolution: visualizing chromosome evolution. *Bioinformatics*, **25**, 960–961.
19. Nielsen,C.B., Younesy,H., O'Geen,H., Xu,X., Jackson,A.R., Milosavljevic,A., Wang,T., Costello,J.F., Hirst,M., Farnham,P.J. *et al.* (2012) Spark: a navigational paradigm for genomic data exploration. *Genome Res.*, **22**, 2262–2269.
20. O'Brien,T.M., Ritz,A.M., Raphael,B.J. and Laidlaw,D.H. (2010) Gremlin: an interactive visualization model for analyzing genomic rearrangements. *IEEE Trans. Vis. Comput. Graph*, **16**, 918–926.
21. Qi,J. and Zhao,F. (2011) inGAP-sv: a novel scheme to identify and visualize structural variation from paired end mapping data. *Nucleic Acids Res.*, **39**, W567–W575.
22. Deng,X., Rayner,S., Liu,X., Zhang,Q., Yang,Y. and Li,N. (2008) DHPC: a new tool to express genome structural features. *Genomics*, **91**, 476–483.
23. Kharchenko,P.V., Alekseyenko,A.A., Schwartz,Y.B., Minoda,A., Riddle,N.C., Ernst,J., Sabo,P.J., Larschan,E., Gorchakov,A.A., Gu,T. *et al.* (2011) Comprehensive analysis of the chromatin landscape in Drosophila melanogaster. *Nature*, **471**, 480–485.
24. Peano,G. (1980) Sur une courbe, qui remplit toute une aire plane. *Math. Ann.*, **36**, 157–160.
25. Hilbert,D. (1891) Über die stetige abbildung einer linie auf ein flächenstück. *Math. Ann.*, **38**, 459–460.
26. Nielsen,C. and Wong,B. (2012) Points of view: representing genomic structural variation. *Nat. Methods*, **9**, 631.

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

27. Nielsen,C. and Wong,B. (2012) Points of view: representing the genome. *Nat. Methods*, **9**, 423.
28. Cao,J., Schneeberger,K., Ossowski,S., Gunther,T., Bender,S., Fitz,J., Koenig,D., Lanz,C., Stegle,O., Lippert,C. *et al.* (2011) Whole-genome sequencing of multiple Arabidopsis thaliana populations. *Nat. Genet.*, **43**, 956–963.
29. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.
30. Li,H., Handsaker,B., Wysoker,A., Fennell,T., Ruan,J., Homer,N., Marth,G., Abecasis,G. and Durbin,R. (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.
31. Hanahan,D. and Weinberg,R.A. (2011) Hallmarks of cancer: the next generation. *Cell*, **144**, 646–674.
32. Stratton,M.R., Campbell,P.J. and Futreal,P.A. (2009) The cancer genome. *Nature*, **458**, 719–724.
33. Stephens,P.J., McBride,D.J., Lin,M.L., Varela,I., Pleasance,E.D., Simpson,J.T., Stebbings,L.A., Leroy,C., Edkins,S., Mudie,L.J. *et al.* (2009) Complex landscapes of somatic rearrangement in human breast cancer genomes. *Nature*, **462**, 1005–1010.
34. Quail,M.A., Kozarewa,I., Smith,F., Scally,A., Stephens,P.J., Durbin,R., Swerdlow,H. and Turner,D.J. (2008) A large genome center's improvements to the Illumina sequencing system. *Nat. Methods*, **5**, 1005–1010.

Downloaded from http://nar.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016