Visualisation tools for next-generation sequencing

Simon Anders

EMBL-EBI
Outline

- Exploring and checking alignment with alignment viewers
- Using genome browsers
- Getting an overview over the whole data with Hilbert curve visualization
- Displaying peaks alongside feature annotation with the “GenomeGraph” package
Alignment viewers: MaqView

Jue Ruan (Beijing Genomics Institute) et al.
Alignment Viewer: MapView

Hua Bao (Sun Yat-Sen University, Guangzhou) et al.
### SAMtools pileup format

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>Position</th>
<th>Reference Base</th>
<th>Consensus Base</th>
<th>Consensus Quality</th>
<th>SNP Quality</th>
<th>Maximum Mapping Quality</th>
<th>Coverage</th>
<th>Base Pile-up</th>
<th>Base Quality Pile-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>25514</td>
<td>G</td>
<td>G</td>
<td>42</td>
<td>0</td>
<td>25</td>
<td>5</td>
<td>....^::</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25515</td>
<td>T</td>
<td>T</td>
<td>42</td>
<td>0</td>
<td>25</td>
<td>5</td>
<td>....</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25516</td>
<td>A</td>
<td>G</td>
<td>48</td>
<td>48</td>
<td>25</td>
<td>7</td>
<td>GGGGG^:G^:g</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25517</td>
<td>G</td>
<td>G</td>
<td>51</td>
<td>0</td>
<td>25</td>
<td>8</td>
<td>........^::</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25518</td>
<td>T</td>
<td>T</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>11</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25519</td>
<td>T</td>
<td>T</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>11</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25520</td>
<td>G</td>
<td>G</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>11</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25521</td>
<td>T</td>
<td>T</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>11</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25522</td>
<td>A</td>
<td>A</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>11</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25523</td>
<td>A</td>
<td>A</td>
<td>72</td>
<td>0</td>
<td>25</td>
<td>15</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25524</td>
<td>C</td>
<td>C</td>
<td>72</td>
<td>0</td>
<td>25</td>
<td>15</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25525</td>
<td>C</td>
<td>C</td>
<td>56</td>
<td>0</td>
<td>24</td>
<td>18</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25526</td>
<td>A</td>
<td>A</td>
<td>81</td>
<td>0</td>
<td>24</td>
<td>18</td>
<td>........:</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>25527</td>
<td>A</td>
<td>A</td>
<td>56</td>
<td>0</td>
<td>24</td>
<td>18</td>
<td>........:</td>
<td></td>
</tr>
</tbody>
</table>

**Fields:** chromosome, position, reference base, consensus base, consensus quality, SNP quality, maximum mapping quality, coverage, base pile-up, base quality pile-up
Coverage vectors

Figure taken from Zhang et al., PLoS Comp. Biol. 2008
Coverage vectors

A coverage vector (or "pile-up" vector) is an integer vector with one element per base pair in a chromosome, tallying the number of reads (or fragments) mapping onto each base pair.

It is the essential intermediate data type in assays like ChIP-Seq or RNA-Seq.

Visualising coverage vectors is non-trivial, but essential for

- quality control
- hypothesis forming
- etc.
Barski et al. (Cell, 2007) have studied histone modification in the human genome with ChIP-Seq.

I use their data for H3K4me1 and H3K4me3 as example data.

(Each data set is from two or three Solexa lanes)
Coverage vector for a full chromosome (chr10)
Zoom in

H3K4me1

H3K4me3
Genome browser tracks

Tracks may contain

- Features (intervals with name)
  - without score
  - with score
- vectors (continuously varying score)

Standard formats for genome browser tracks

- BED
- GFF
- **Wiggle fixedStep** and **variableStep**
Displaying tracks alongside annotation

- Either, upload your track file to a web-base browser
  - UCSC genome browse
  - Ensembl genome browser
- or use a stand-alone browser on your desktop computer
  - Integrated Genome Browser (IGB) [Genoviz]
  - Argo Genome Browser [Broad Institute]
  - Artemis [Sanger Institute]

Displaying large amounts of data requires patience and lots of RAM. Not all tools handle it well.
rtracklayer: Bioconductor package by M. Lawrence (FHCRC)

- import and export BED, Wiggle, and GFF files
- manipulate track data and get sub-views
- directly interact with a genome browser (UCSC or Argo) to drive displaying of track data
Difference between the track formats

- **Formats for feature-by-feature data:**
  - BED
  - GFF

- **Formats for base-by-base scores**
  - Wiggle
  - BedGraph

- **Wiggle has three sub-types:**
  - [BED-like]
  - variableStep
  - fixedStep
Wiggle format: variableStep and fixedStep

browser position chr19:59304200-59310700
browser hide all
track type=wiggle_0 name="varStepTrack" description="varStep example"
   visibility=full autoScale=off viewLimits=0.0:25.0 color=50,150,255
   yLineMark=11.76 yLineOnOff=on priority=10
variableStep chrom=chr19 span=150
59304701 10.0
59304901 12.5
59305401 15.0
59305601 17.5
59305901 20.0
59306081 17.5
59306301 15.0
59307871 10.0
track type=wiggle_0 name="fixedStepTrack" description="fixedStep example"
fixedStep chrom=chr19 start=59307401 step=300 span=200
1000
900
800
700
600
500
400
300
200
100

All coordinates 1-based!
track type=bedGraph name="BedGraph Track"
chr19 59302000 59302300 -1.0
chr19 59302300 59302600 -0.75
chr19 59302600 59302900 -0.50
chr19 59302900 59303200 -0.25
chr19 59303200 59303500 0.0
chr19 59303500 59303800 0.25
chr19 59303800 59304100 0.50
chr19 59304100 59304400 0.75
chr19 59304400 59304700 1.00

All coordinates 0-based, half-open!

Specs: See UCSC Genome Browser web site
Back to the bird’s eyes view
We need a way to get a general overview on the data without either not seeing any details not getting lost in them.

A possible solution: Hilbert curve visualisation

The Hilbert curve
What is hidden in here?
Hilbert plot of the constructed example vector
Construction of the Hilbert curve: Level 1
Construction of the Hilbert curve: Level 2
Construction of the Hilbert curve: Level 3
Construction of the Hilbert curve: Level 4
Hilbert curve: Approaching the limit
Coverage vector for a full chromosome (chr10)

H3K4me1

H3K4me3

chrom. 10
Hilbert plot of the coverage vectors

H3K4me1 (mono-methylation)

H3K4me3 (tri-methylation)
HilbertVis

Hilbert curve data display

Displayed data

Previous: 71157_ratio.gff: chr12
Load
Unload
Save Img
About
Quit

Bin under mouse cursor
Position: 71,664,833
Value: xxx

Full sequence

0
114,279,756
57,139,878
85,709,817

Displayed part of sequence

Color key
Darker
Lighter

Effect of left mouse button
Zoom in 4x
Zoom in 64x
Linear plot

Zoom out
Zoom out 4x
Zoom out 64x

Pixel size
Coarser
Finer

Linear plot - HilbertVis

10
71,633,446
71,689,246

-10
71157_ratio.gff: chr12

EMBL-EBI
HilbertVis

- stand-alone tool to display GFF, Wiggle, Maq map
  
  http://www.ebi.ac.uk/huber-srv/hilbert/
  
  (or Google for “hilbertvis”)

- R package to display any long R vector
  - either via commands for batch processing
    
    Bioconductor package “HilbertVis”
  - or via GUI for exploring
    
    Bioconductor package “HilbertVisGUI”
Three-colour Hilbert plot

Overlay of the previous plots and exon density

**red:** mono-methylation

**green:** tri-methylation

**blue:** exons
Log fold-changes between two *Arabidopsis* eco-types, chromosome 2.

[Data courtesy of M. Seiffert, IPK Gatersleben]
Other uses: Conservation scores

Human chromosome 10: Comparing phastCons conservation scores with exon density
GenomeGraphs: Bioconductor package by S. Durrinck, UCB

- Load gene models from Ensembl via BiomaRt and plots them, alongside experimental data
library(GenomeGraphs)
library(HilbertVis)

mart <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")

start <- 57000000
end <- 59000000

plusStrand <- makeGeneRegion( chromosome = 10,
    start = start, end = end, strand = "+", biomart = mart )

minusStrand <- makeGeneRegion( chromosome = 10,
    start = start, end = end, strand = "-", biomart = mart )

genomeAxis <- makeGenomeAxis( )
track.ctcf <- makeBaseTrack(
  base = seq( start, end, length.out = 10000 ),
  value = shrinkVector(
    as.vector( cov.ctcf$chr10[start:end] ), 10000 ),
  dp = DisplayPars( lwd = 0.5, color="red", ylim=c(0, 50) )
)

track.gfp <- makeBaseTrack(
  base = seq( start, end, length.out = 10000 ),
  value = shrinkVector(
    as.vector( cov.gfp$chr10[start:end] ), 10000 ),
  dp = DisplayPars( lwd = 0.5, color="blue", ylim=c(0, 50) )
)

gdPlot( list( `plus`=plusStrand, `CTCF`=track.ctcf,
  genomeAxis, `GFP`=track.gfp, `minus`=minusStrand ) )