

Software

Open Access

GenomeGraphs: integrated genomic data visualization with R

Steffen Durinck*^{†1,2}, James Bullard*^{†2}, Paul T Spellman¹ and Sandrine Dudoit^{2,3}

Address: ¹Life Sciences Department, Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, USA, ²Division of Biostatistics, School of Public Health, UC Berkeley, 101 Haviland Hall, Berkeley, CA 94720-7358, USA and ³Department of Statistics, UC Berkeley, 367 Evans Hall, Berkeley, CA 94720-3860, USA

Email: Steffen Durinck* - steffen@stat.berkeley.edu; James Bullard* - bullard@berkeley.edu; Paul T Spellman - PTspellman@lbl.gov; Sandrine Dudoit - sandrine@stat.berkeley.edu

* Corresponding authors †Equal contributors

Published: 6 January 2009

Received: 14 April 2008

BMC Bioinformatics 2009, 10:2 doi:10.1186/1471-2105-10-2

Accepted: 6 January 2009

This article is available from: <http://www.biomedcentral.com/1471-2105/10/2>

© 2009 Durinck et al; licensee BioMed Central Ltd.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Background: Biological studies involve a growing number of distinct high-throughput experiments to characterize samples of interest. There is a lack of methods to visualize these different genomic datasets in a versatile manner. In addition, genomic data analysis requires integrated visualization of experimental data along with constantly changing genomic annotation and statistical analyses.

Results: We developed *GenomeGraphs*, as an add-on software package for the statistical programming environment R, to facilitate integrated visualization of genomic datasets. *GenomeGraphs* uses the *biomaRt* package to perform on-line annotation queries to Ensembl and translates these to gene/transcript structures in viewports of the *grid* graphics package. This allows genomic annotation to be plotted together with experimental data. *GenomeGraphs* can also be used to plot custom annotation tracks in combination with different experimental data types together in one plot using the same genomic coordinate system.

Conclusion: *GenomeGraphs* is a flexible and extensible software package which can be used to visualize a multitude of genomic datasets within the statistical programming environment R.

Background

Computational biologists are dealing with a growing range of genomic datasets, including microarray (e.g., mRNA, ChIP, SNP, CGH, and tiling-Chip) and ultra high-throughput sequencing (e.g., mRNA-Seq and ChIP-Seq) data. An increasing number of biological studies involve multiple, distinct, and high-throughput assays to characterize samples of interest. Novel and flexible visualization methods are needed to integrate these various data sources and combine them with annotation data from biological databases such as Ensembl [1].

Genome browsers such as the Ensembl Genome Browser [1], NCBI Entrez Map Viewer [2], and UCSC's Golden Path Genome Browser [3] enable upload and visualization of experimental data but have limited plotting options, do not provide data analysis capabilities of the displayed data, and are too far removed from the environment used to conduct statistical analysis. Other tools linking genome annotation to experimental data are mostly limited to a specific data type or rely on the Genome Browser's viewers for visualization. Statistical Viewer [4] for example facilitates interpretation of linkage and asso-

ciation data by providing a plug-in for data upload to the Ensembl Genome Browser.

The X:Map [5] genome annotation database and its companion software package *exonmap* enable integrated visualization of experimental data and genome annotation but it is specific to exon arrays and requires a local installation of the Ensembl database. It does not currently support visualization of multiple datasets and does not represent alternative splicing structures.

The main drawback of the tools described above is that they are not programmatically accessible and cannot be integrated into an analysis pipeline requiring batch processing. In addition, the required data upload step does not scale well for large and complex datasets.

The statistical programming environment R <http://www.r-project.org> along with the Bioconductor Project <http://www.bioconductor.org> provide a plethora of methods and tools to analyze and visualize data. The software package described in this paper, *GenomeGraphs*, builds on this functionality by providing an integrated API for direct visualization of data from a variety of sources. *GenomeGraphs* allows complex customization to facilitate a more complete integration and representation of genomic datasets.

Implementation

Graphic composition

Genomic dataset objects

We developed *GenomeGraphs* as an add-on package for the statistical programming environment R [6]. It utilizes the S4 class system and represents each genomic data type as a specific class. The root class *gdObject* provides basic functionality for display of data that can be mapped onto the genome (see Table 1). All data-type specific classes extend *gdObject* and corresponding display functionalities are built on top of this class. An example is the *GenericArray* class which represents gene expression microarray and arrayCGH data. This class takes a matrix of intensities as input which can easily be extracted from *ExpresionSet* objects as produced by the Bioconductor *affy* package. Another example is the *GeneRegion* class which represents strand-specific genes in a given genomic region. Quantitative genomic data, such as data from arrayCGH and tiling array experiments, frequently have associated segmented data. Segmented data are represented by the *Segmentation* class. Additional classes exist that represent ideograms, genomic axes, and legends. Regions of interest can be highlighted on the plot by using objects of the *RectangleOverlay* class. Once *gdObjects* are created, they can be visualized in one plot using the main plotting function, *gdPlot*.

New technological developments to characterize cellular states may need novel representations. Classes represent-

Table 1: Overview of classes representing drawable genomic datasets

Class	Description
<i>gdObject</i>	the root class of the system, never directly instantiated
<i>DisplayPars</i>	class managing various plotting parameters
<i>Gene</i>	class representing a gene
<i>GeneRegion</i>	class defining a region of a chromosome, generally a set of genetic elements (genes)
<i>Transcript</i>	class defining a transcript
<i>TranscriptRegion</i>	class defining a region of a chromosome, generally a set of genetic elements (transcripts)
<i>Ideogram</i>	class representing an ideogram
<i>Title</i>	class to draw a title
<i>Legend</i>	class to draw a legend
<i>GenomeAxis</i>	class to draw an axis
<i>AnnotationTrack</i>	class used to represent custom annotation
<i>Overlay</i>	root class for overlays, never directly instantiated
<i>RectangleOverlay</i>	class to represent rectangular regions of interest
<i>TextOverlay</i>	class to draw text on plots
<i>Segmentation</i>	class to draw horizontal lines in various sets of data
<i>GenericArray</i>	class to draw data from microarrays.
<i>ExonArray</i>	class to draw data from exon microarrays.
<i>GeneModel</i>	class to draw custom gene models (intron-exon structures)
<i>BaseTrack</i>	class to draw arbitrary data at a given base
<i>MappedRead</i>	class to plot sequencing reads that are mapped to the genome

The main class, of which all other drawable classes are subclasses of, is the *gdObject* class. The *Gene*, *Transcript*, and *GeneRegion* classes are subclasses of *gdObject* and represent annotation data that are retrieved on-line from Ensembl upon creation of these objects. The third set of classes represent experimental data such as microarray and copy number data which can both be modeled using the *GenericArray* class. Probe-level exon array data can be represented by the *ExonArray* class and base-specific values such as nucleotide conservation scores among different species are represented by the *BaseTrack* class.

ing these new data types can be easily added to *GenomeGraphs* and if the corresponding drawing methods are defined, the new data structures can be plotted using *gdPlot* along with data from existing classes.

Genome annotation retrieval from Ensembl using biomaRt
GenomeGraphs relies on the *biomaRt* package [7] to retrieve genomic annotation information on-line from Ensembl using BioMart web services [8]. The annotation information retrievable through *biomaRt* ranges from gene annotation, transcript isoforms to SNP data. This information can be retrieved from the most current release of Ensembl or from archived releases. By using *biomaRt*, there is no need for a local database installation of Ensembl, greatly facilitating the software installation procedure.

Custom genome annotation tracks

Ensembl contains annotation of a limited number of eukaryotic genomes. Any custom genome annotation can be visualized in *GenomeGraphs* by constructing instances of the *AnnotationTrack* class. For instance, genomic annotation encoded in GFF files can be easily used to create a custom *AnnotationTrack* object for visualization. To use the *AnnotationTrack* class, region start and end positions need to be given, as well as how these regions are to be grouped.

Mapping of user data to genomic coordinates

GenomeGraphs is a visualization tool and as such does not provide mappings of user supplied data to the genome. Instances of the class *gdObject* take as input genomic coordinates provided by the user who is responsible for ensuring that these coordinates match the relevant genome annotation. To get the chromosomal coordinates of the data, users can either rely on the annotation provided by the platform which generated the data or on independently created mappings to the genome.

Results

Example 1: arrayCGH and exon array data

In this first example, we illustrate how different genomic datasets can be visualized together in an integrated *GenomeGraphs* graphic. We use arrayCGH and Affymetrix exon array data and plot these together with genomic annotation from Ensembl.

We first load the *GenomeGraphs* package and one of its example datasets. This dataset contains copy number data and segmented copy number data, as well as exon array data for a small genomic region. Once the data are loaded, a *gdObject* is created for each data type, namely a *Segmentation* object containing the copy number segments, a *GenericArray* object containing the raw copy number data, an *Ideogram* object representing the relevant chromosome we are plotting, a *GenericArray* object containing the exon

array data, and a *GenomeAxis* object for the genomic coordinate axis.

```
> library(GenomeGraphs)
> data('exampleData', package='GenomeGraphs')
> seg = makeSegmentation(value = segments,
  start = segStart, end = segEnd, dp = DisplayPars(color =
  'dodgerblue2', lwd = 2, lty = 'dashed'))
> copyNumber = makeGenericArray(intensity = cn,
  probeStart = probestart,
  segmentation = seg, dp = DisplayPars(size = 3, color =
  'seagreen', type="dot"))
> ideogram = makeIdeogram(chromosome = 3)
> expression = makeGenericArray(intensity = intensity,
  probeStart = exonProbePos,
  dp = DisplayPars(color='darkred', type='point'))
> genomeAxis = makeGenomeAxis(add53 = TRUE, add35
  = TRUE)
```

In a next step, genomic annotation information is retrieved on-line from Ensembl using the *biomaRt* package. We first connect to the Ensembl BioMart database and select the human (*hsapiens*) dataset. Then, we retrieve gene structures on the forward and reverse strands of the region we want to visualize.

```
> minbase = 180292097
> maxbase = 180492096
> mart = useMart('ensembl', data-
  set='hsapiens_gene_ensembl')
> genesplus = makeGeneRegion(start = minbase, end =
  maxbase, strand = '+', chromosome = '3', biomart = mart)
> genesmin = makeGeneRegion(start = minbase, end =
  maxbase, strand = '-', chromosome = '3', biomart = mart)
```

In a last step, the *gdPlot* function is called to plot instances of *gdObject* that were created above. The objects are given to *gdPlot* as a list and the order in the list will determine the plotting order from top to bottom. A minimum and maximum base position are also given as arguments to restrict the visualization to this particular genomic region. The plot produced from this example is shown in Figure 1.

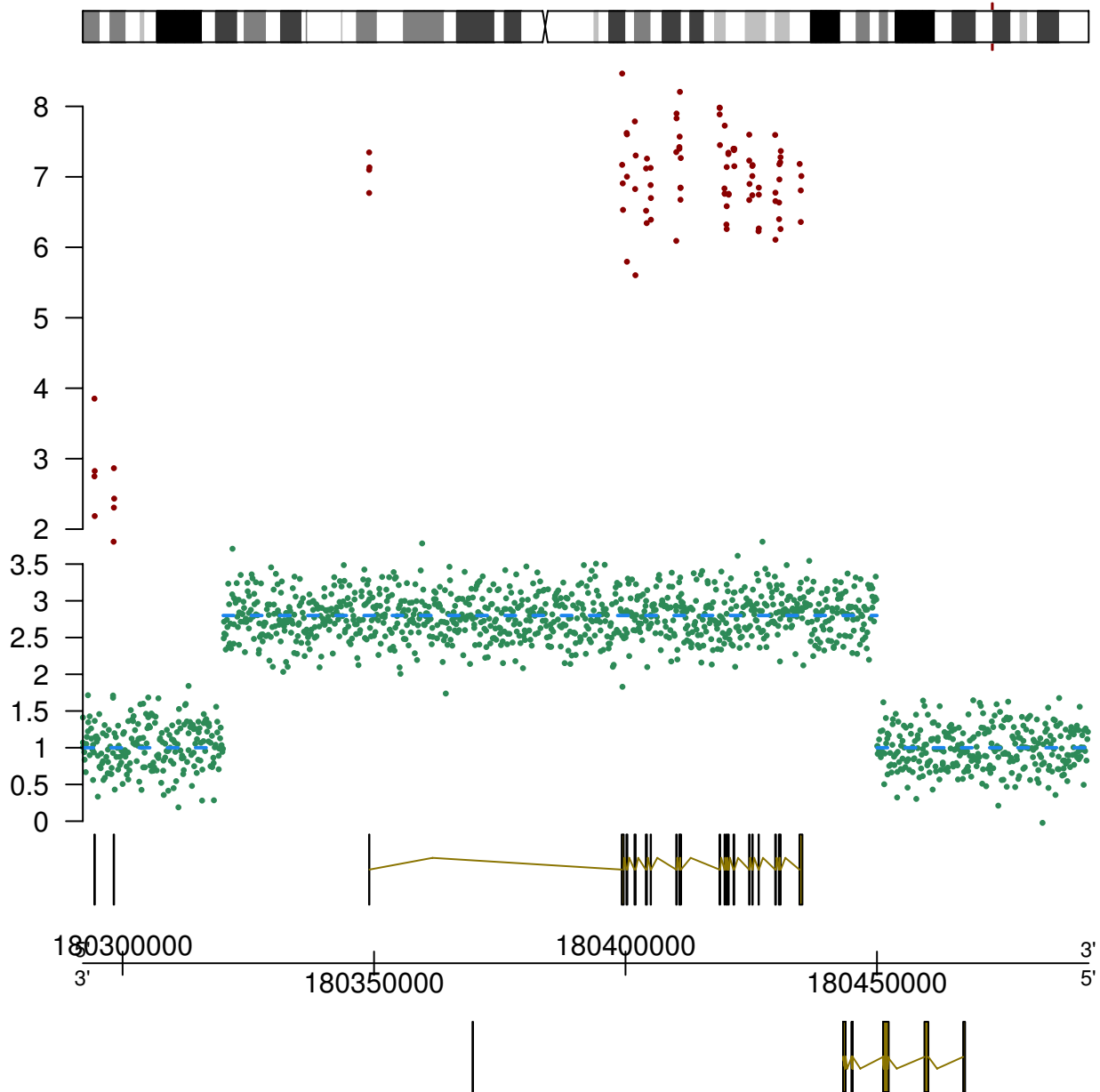


Figure 1
ArrayCGH and exon array data. The first track in this figure shows an ideogram of the human chromosome 3. The red marker highlights the plotted genomic region. The second track shows exon array data, where each data point corresponds to a probe measuring the expression level of an exon. The third track displays copy number data in green and segmented copy number data with dashed blue lines. Note the amplification which can be seen in both the copy number and exon array tracks, suggesting that the amplification event results in higher expression levels of the gene in this region. The bottom track shows the gene annotation data from Ensembl.

```
> gdPlot(list(ideogram, expression, copyNumber, gene-
plus, genomeAxis, genesmin), minBase = minbase, max-
Base = maxbase)
```

Example II: Transcript isoforms and exon array data
 In a second example, we show how probe-level exon array data from the Affymetrix GeneChip® Human Exon 1.0 ST

platform (data available from <http://www.affymetrix.com>), can be plotted along with gene models from Affymetrix as well as gene and transcript annotation from Ensembl. The data of the exon array are not plotted at the exact chromosomal location of the probes in order to clearly visualize alternative splicing events. Most of the exons are represented on the Human Exon 1.0 ST platform by four probes. The location of these four probes are equally spaced in the data plots. Each exon is separated by a vertical line and the exons are linked to their genomic location by connecting lines. This visualization makes it easy to relate alternative exon usage, as observed in the exon array data, to known alternative transcript isoforms in Ensembl (Figure 2). The region highlighted in the plot shows the exon that is not expressed in the samples. To generate this plot, we first create the different subclasses of *gdObject*, namely: *Title*, *ExonArray*, *Gene*, *Transcript*, and *Legend* objects. In addition, we make a custom annotation track using the *AnnotationTrack* class.

```
> data('unrData', package='GenomeGraphs')

> title = makeTitle(text = 'ENSG00000009307', color = 'darkred')

> col = colorRampPalette(c('firebrick2','dodgerblue2'))(length(unrData[1,]))

> exon = makeExonArray(intensity = unrData, probeStart = unrPositions[,3], probeEnd = unrPositions[,4],

  probelD = as.character(unrPositions[,1]), nProbes = unrNProbes,

  dp = DisplayPars(color = col, mapColor = 'dodgerblue2'), displayProbesets = FALSE)

> affyModel <- makeAnnotationTrack(start = unrPositions[,3], end = unrPositions[,4],

  feature = "gene_model", group = "ENSG00000009307",

  dp = DisplayPars(gene_model = "darkblue"))

> gene = makeGene(id = 'ENSG00000009307', biomart = mart)

> transcript = makeTranscript(id = 'ENSG00000009307', biomart = mart)

> legend = makeLegend(text = c('affyModel','Ensembl Gene', 'Ensembl Transcript'),
```

```
  fill = c('darkgreen','orange','cornflowerblue'), cex = 0.5)
```

In a second step, we use the *RectangleOverlay* class to create a highlighted region followed by the *gdPlot* function to produce the integrated plot.

```
> rOverlay = makeRectangleOverlay(start = 115085100, end = 115086500, region = c(3,5),

  dp = DisplayPars(alpha = .2, fill = "olivedrab1"))

> gdPlot(list(title, exon, affyModel, gene, transcript, legend), minBase = 115061061, maxBase = 115102147, overlay = rOverlay)
```

The plot generated in this second example is shown in Figure 2.

Example III: Short read sequencing and tiling array data

In the final example, we show how complex and diverse sets of data can be integrated to facilitate joint analysis and draw biological conclusions by presenting data from various published datasets on yeast. First, we construct a list where each *gdObject* represents either annotation or a publicly available dataset. We have plotted data from Ensembl, an Illumina sequencing dataset [9], Affymetrix tiling array data [10], nucleosome position data [11], and conservation data across 7 related species [12].

```
> data("seqDataEx", package = "GenomeGraphs")

> str = seqDataEx$dauid[, "strand"] == 1

> biomart = useMart("ensembl", "scerevisiae_gene_ensembl")

> pList = list("-" = makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,

  strand = "-", biomart = biomart,

  dp = DisplayPars(plotId = TRUE, idRotation = 0, cex = .5)),

  "+" = makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,

  strand = "+", biomart = biomart,

  dp = DisplayPars(plotId = TRUE, idRotation = 0, cex = .5)),

  makeGenomeAxis(dp = DisplayPars(byValue = 1e3, size = 3)),

  "+" = makeGeneRegion(chromosome = "IV", start = 1300000, end = 1310000,

  strand = "+", biomart = biomart,

  dp = DisplayPars(plotId = TRUE, idRotation = 0, cex = .5)),
```

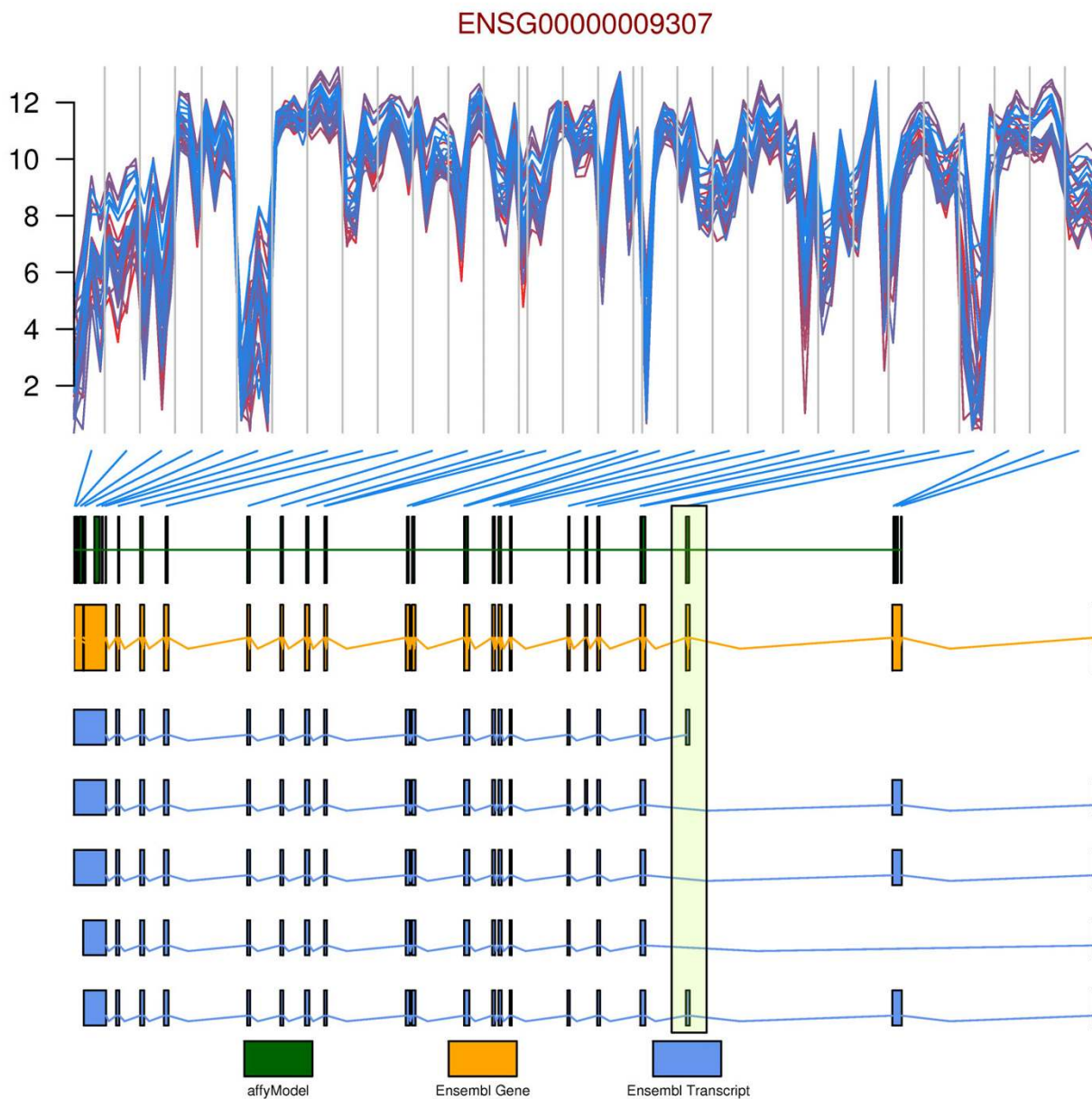


Figure 2
Transcript isoforms and exon array data. Probe-level exon array data is plotted in the top graphic. The data of the exon array is intentionally not plotted on the exact chromosomal location of the probes in order to clearly visualize alternative splicing events. Each line in the top track represents a different sample. Usually, there are four probes per exon on the Affymetrix GeneChip® Human Exon 1.0 ST platform, vertical gray lines group these four probes belonging to the same exon together. The blue connecting lines map these exons to gene models as defined by Affymetrix (green) and Ensembl (orange). Transcript isoforms known for this gene are plotted in dark blue. The region highlighted in the plot by a *RectangleOverlay* object shows the exon that is not expressed in the samples. One can see that this is a known alternatively spliced exon as annotated by Ensembl.

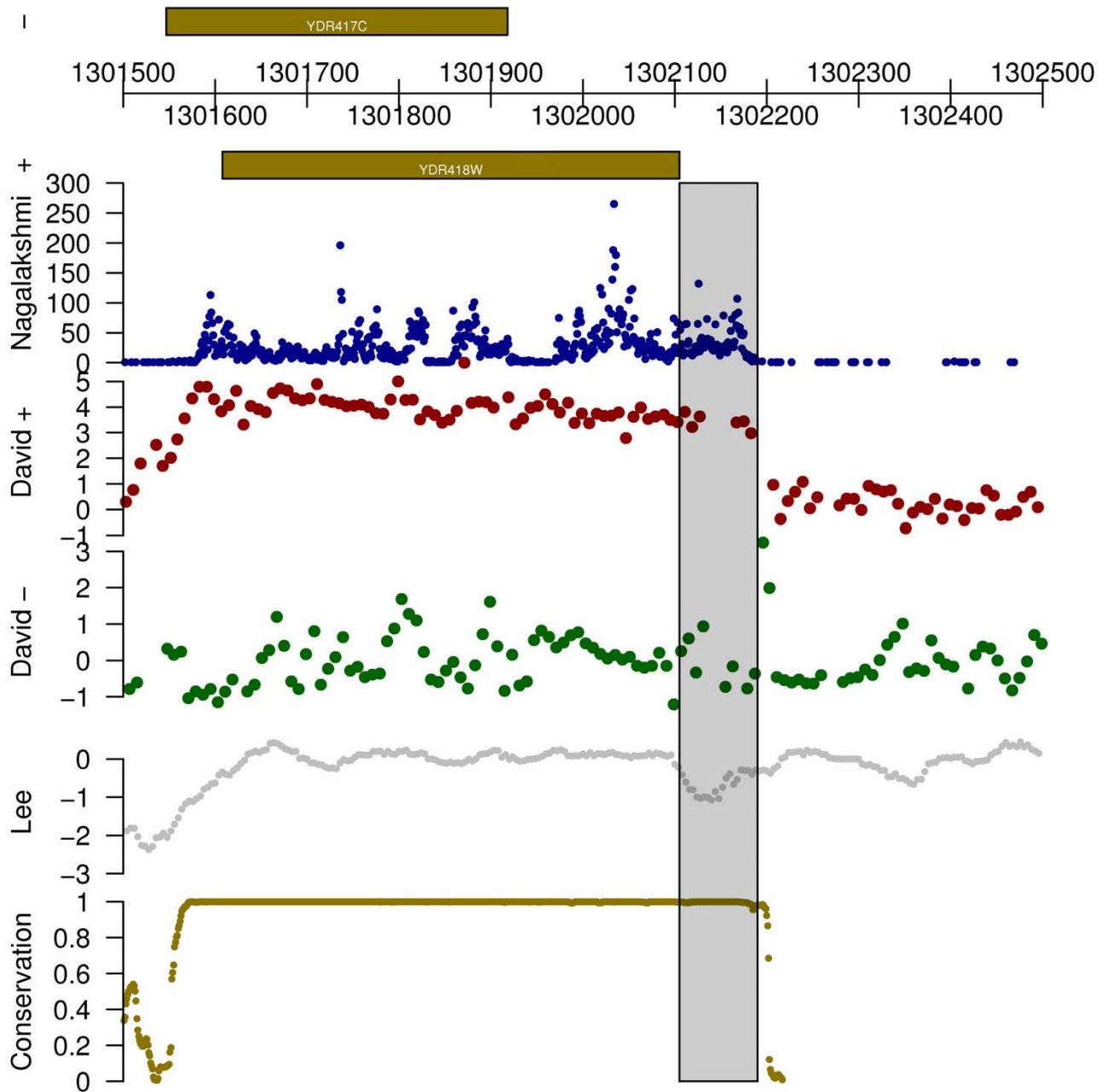


Figure 3
Short read sequencing and tiling array data. Data plotted are Illumina sequencing data from Nagalakshmi et al. [9], tiling array data from David et al. [10], nucleosome data from Lee et al. [11], and conservation track data from Siepel et al. [12]. The semi-transparent box highlights a possible annotation error in SGD, as suggested by the occurrence of the transcript in multiple separate datasets. In addition, the conservation track data demonstrate corroborating evidence for the possibility of a longer gene.

```
"Nagalakshmi" = makeBaseTrack(base = seqDataEx$snyder [, "location"], value = seqDataEx$snyder [, "counts"],
```

```
dp = DisplayPars(lwd = .3, color = "darkblue", ylim = c(0,300))),
```

```
"David +" = makeGenericArray(probeStart = seqDataEx$david [str, "location"],
```

```
intensity = seqDataEx$david [str, "expr", drop = FALSE],
```

```
dp = DisplayPars(pointSize = .5)),
```

```
"David -" = makeGenericArray(probeStart = seqDataEx$david [!str, "location"],
```

```
intensity = seqDataEx$david [!str, "expr", drop = FALSE],
```

```
dp = DisplayPars(color = "darkgreen", pointSize = .5)),
```

```
"Lee" = makeBaseTrack(base = seqDataEx$nislow [, "location"],
```

```
value = seqDataEx$nislow [, "value"], dp = DisplayPars(color="grey", lwd = .25)),
```

```
"Conservation" = makeBaseTrack(base = seqDataEx$conservation [, "location"],
```

```
value = seqDataEx$conservation [, "score"],
```

```
dp = DisplayPars(color="gold4", lwd = .25)))
```

Having constructed the list of elements we wish to plot, we now set up an overlay, using the *RectangleOverlay* class, to highlight a region of interest. Finally, we plot the result using *gdPlot*. Although configuring and designing the initial plot may seem laborious, once we have this basic structure we can easily produce plots for all regions of interest.

```
> rOverlay = makeRectangleOverlay(start = 1302105, end = 1302190, region = c(4,8), dp = DisplayPars(alpha = .2))
```

```
> gdPlot(pList, minBase = 1301500, maxBase = 1302500, overlay = rOverlay)
```

The plot produced in this third example is shown in Figure 3.

Conclusion

GenomeGraphs is a versatile and extensible visualization package in R, which is well suited to create integrated displays of diverse experimental datasets and genomic annotation information. By using the *biomaRt* package, annotation information is retrieved directly from Ensembl and there is no need to install and maintain annotation databases locally. Custom annotation tracks can also be created by using the *AnnotationTrack* class. Finally, *GenomeGraphs* provides the user with tight integration into R providing immediate access to a wealth of statistical methods.

The software package comes with a vignette which is an executable document that demonstrates how to use the package. The examples described in this paper are also included in the vignette and can be executed after installation of the package. More complex features are also demonstrated in the vignette. Future versions of the package will include more flexibility in terms of plotting parameters and plotting novel features such as visualizing SNP information as annotated by Ensembl and stacked sequencing read representations.

Availability and requirements

GenomeGraphs is an open source software package under the *Artistic-2.0* license and has been contributed to the Bioconductor Project. The software and source code are available for download from <http://www.bioconductor.org>. This document was produced using R-2.8.0 and *GenomeGraphs* version 1.2.0 available at the following URL: <http://bioconductor.org/packages/2.3/bioc/html/GenomeGraphs.html>. The package has been tested and run on OS X, Windows, and a variety of Linux systems. *GenomeGraphs* depends on the following software packages *XML*, *RCurl*, and *biomaRt*, which can be downloaded from Bioconductor or installed from R using the <http://www.bioconductor.org/biocLite.R> script. The versatility of *GenomeGraphs* visualization relies on the powerful R plotting package *grid* [13]. Each *gdObject* is plotted in an individual *viewPort* from the *grid* package. *Grid* is typically installed together with the base installation of R.

Authors' contributions

SD and JB developed the software package. PS and SD provided scientific advice and the resources to develop the software.

Acknowledgements

We would like to acknowledge Elizabeth Purdom and Mark Robinson for beta testing early versions of the software and contributing the ExonArray example dataset. We thank the anonymous reviewers for their comments and suggestions to improve this work. Funding was provided by the U54 CA 112970 grant of the TCGA project <http://cancergenome.nih.gov/>.

References

- Hubbard T, Aken B, Ayling S, Ballester B, Beal K, Bragin E, Brent S, Chen Y, Clapham P, Clarke L, Coates G, Fairley S, Fitzgerald S, Fernandez-Banet J, Gordon L, Graf S, Haider S, Hammond M, Holland R, Howe K, Jenkinson A, Johnson N, Kahari A, Keefe D, Keenan S, Kissella R, Kokocinski F, Kulesha E, Lawson D, Longden I, et al: **Ensembl 2009**. *Nucleic Acids Res* 2009:D690-697.
- Wheeler D, Barrett T, Benson D, Bryant S, Canese K, Chetvernin V, Church D, Dicuccio M, Edgar R, Federhen S, Feolo M, Geer L, Helmberg W, Kapustin Y, Khovayko O, Landsman D, Lipman D, Madden T, Maglott D, Miller V, Ostell J, Pruitt K, Schuler G, Shumway M, Sequiera E, Sherry S, Sirotkin K, Souvorov A, Starchenko G, Tatusov R, et al: **Database resources of the National Center for Biotechnology Information**. *Nucleic Acids Research* 2008:D780-D786.
- Karolchik D, Kuhn R, Baertsch R, Barber G, Clawson H, Diekhans M, Giardine B, Harte R, Hinrichs A, Hsu F, Kober K, Miller W, Pedersen J, Pohl A, Raney B, Rhead B, Rosenbloom K, Smith K, Stanke M, Thakkapallayil A, Trumbower H, Wang T, Zweig A, Haussler D, Kent W: **The UCSC Genome Browser Database: 2008 update**. *Nucleic Acids Research* 2008:D773-D779.
- Stenger J, Xu H, Haynes C, Hauser E, Pericak-Vance M, Goldschmidt-Clermont P, Vance J: **Statistical Viewer: a tool to upload and integrate linkage and association data as plots displayed within the Ensembl genome browser**. *BMC Bioinformatics* 2005, **6**:95.
- Yates T, Okoniewski M, Miller C: **X:Map: annotation and visualization of genome structure for Affymetrix exon array analysis**. *Nucleic Acids Research* 2008:D780-D786.
- R Development Core Team: *R: A Language and Environment for Statistical Computing* 2008 [<http://www.R-project.org>]. R Foundation for Statistical Computing, Vienna, Austria
- Durinck S, Moreau Y, Kasprzyk A, Davis S, Moor BD, Brazma A, Huber W: **BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis**. *Bioinformatics* 2005, **21**(16):3439-3440.
- Kasprzyk A, Keefe D, Smedley D, London D, Spooner W, Melsopp C, Hammond M, Rocca-Serra P, Cox T, Birney E: **EnsMart: a generic system for fast and flexible access to biological data**. *Genome Res* 2004, **14**(1):160-169.
- Nagalakshmi U, Wang Z, Waern K, Shou C, Raha D, Gerstein M, Snyder M: **The transcriptional landscape of the yeast genome defined by RNA sequencing**. *Science* 2008, **320**(5881):1344-1349.
- David L, Huber W, Granovskaia M, Toedling J, Palm CJ, Bofkin L, Jones T, Davis RW, Steinmetz LM: **A high-resolution map of transcription in the yeast genome**. *Proc Natl Acad Sci USA* 2006, **103**(14):5320-5325.
- Lee W, Tillo D, Bray N, Morse RH, Davis RW, Hughes TR, Nislow C: **A high-resolution atlas of nucleosome occupancy in yeast**. *Nat Genet* 2007, **39**(10):1235-1244.
- Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, Rosenbloom K, Clawson H, Spieth J, Hillier LW, Richards S, Weinstock GM, Wilson RK, Gibbs RA, Kent WJ, Miller W, Haussler D: **Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes**. *Genome Res* 2005, **15**(8):1034-1050.
- Murrell P: *R Graphics* Boca Raton: CRC Press; 2005.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
http://www.biomedcentral.com/info/publishing_adv.asp

