Creating IGV HTML reports with tracktables

Thomas Carroll[1*]

[1] Bioinformatics Core, MRC Clinical Sciences Centre;

[*]thomas.carroll (at)imperial.ac.uk

May 15, 2016

Contents

1 The tracktables package

Visualising genomics data in genome browsers is a key step in both quality control and the initial interrogation of any hypothesis under investigation.

The organisation of large collections of genomics data (such as from large scale high-thoughput sequencing experiments) alongside their associated sample or experimental metadata allows for the rapid evaluation of patterns across experimental groups.

Broad's Intergrative Genome Viewer (IGV) provides a set of methods to make use of sample metadata when visualising genomics data. As well as identifying sample metadata within the genome browser, this sample information can be used in IGV to group, sort and filter samples.

This use of sample metadata, alongside the ability to control IGV through ports, provides the desired framework to rapidly interrogate large cohorts of genomics data once the appropriate file structure is built.

The Tracktables package provides a set of tools to build IGV session files from data-frames of sample files and their associated metadata as well as produce IGV linked HTML reports for high-throughput visualisation of sample data in IGV.

2 Creating IGV sessions and HTML reports using tracktables

The two main functions within the `tracktables` package are the `MakeIGVSession()` function for creating IGV session XMLs and any associated sample metadata files and the `maketracktable()` function to create HTML pages containing the sample metadata and interval tables used to control IGV.

2.1 Creating input files for tracktables

`tracktables` functions require the user to provide both a data-frame of metadata information and a data-frame of the paths of sample files to be visualised in IGV.

These data-frames must both have one column named "SampleName" which contains unique sample IDs. This column will be used to match samples and only samples within both files will be included in the IGV session.

The remaining metadata SampleSheet columns may be user-defined but must all contain column titles. (See example below)

The FileSheet (with file paths) must contain the columns "SampleName", "bam","bigwig" and "interval". These columns may contain NA values when no relavant file is associated to a sample.

Here we create a small example SampleSheet (containing metadata) and FileSheet (containing file locations) from some example histone mark, RNA polymerase 2 and Ebf ChIP-seq.

```
library(tracktables)

fileLocations <- system.file("extdata",package="tracktables")

bigwigs <- dir(fileLocations,pattern="*.bw",full.names=TRUE)
intervals <- dir(fileLocations,pattern="*.bed",full.names=TRUE)
bigWigMat <- cbind(gsub("_Example.bw","",basename(bigwigs)),
                   bigwigs)
intervalsMat <- cbind(gsub("_Peaks.bed","",basename(intervals)),
                      intervals)

FileSheet <- merge(bigWigMat,intervalsMat,all=TRUE)
FileSheet <- as.matrix(cbind(FileSheet,NA))
colnames(FileSheet) <- c("SampleName","bigwig","interval","bam")
```

```
SampleSheet <- cbind(as.vector(FileSheet[,"SampleName"]),
                     c("EBF","H3K4me3","H3K9ac","RNAPol2"),
                     c("ProB","ProB","ProB","ProB"))
colnames(SampleSheet) <- c("SampleName","Antibody","Species")
```

The SampleSheet contains a small section of metadata for the EBF, RNApol2, H3K4me3 and H3K9ac ChIP. The "SampleName" column contains the unique IDs.

```
head(SampleSheet)
```

```
##      SampleName Antibody  Species
## [1,] "EBF"      "EBF"     "ProB"
## [2,] "H3K4me3"  "H3K4me3" "ProB"
## [3,] "H3K9ac"   "H3K9ac"  "ProB"
## [4,] "RNAPol2"  "RNAPol2" "ProB"
```

The FileSheet contains the "SampleName" column with unique IDs matching those founds in the SampleSheet. The remaining columns are "bam","bigwig" and "interval" and list the full paths of relevant files to be displayed in IGV.

```
head(FileSheet)
```

```
##      SampleName bigwig                       interval                bam
## [1,] "EBF"      "pathTo/EBF_Example.bw"     "pathTo/EBF_Peaks.bed" NA
## [2,] "H3K4me3"  "pathTo/H3K4me3_Example.bw" NA                     NA
## [3,] "H3K9ac"   "pathTo/H3K9ac_Example.bw"  NA                     NA
## [4,] "RNAPol2"  "pathTo/RNAPol2_Example.bw" NA                     NA
```

Note that not all samples have intervals associated to them and ,here, none of these samples have BAM files associated to them. NA values within the FileSheet will be ignored by `tracktables` functions.

2.2 Creating an IGV session XML file

`tracktables` can create an IGV session XML and associated sample information file from this SampleSheet and FileSheet.

In addition to the FileSheet and SampleSheet, the `MakeIGVSession()` function requires the location to write to, the filename for the session XML and the genome to be used in IGV (see IGV for details on supported genomes).

```
MakeIGVSession(SampleSheet,FileSheet,igvdirectory=getwd(),"Example","mm9")
```

This creates two files in the current working directory containing the sample information file for IGV ("SampleMetadata.txt") and the session XML itself to be loaded into IGV ("Example.xml").

2.3 Creating a Tracktable HTML report

As well as producing session XMLs and sample information files, the `tracktables` package can produce HTML reports which contain metadata and methods to control IGV.

The report structure is made of a main **Tracktables Experiment Report** which houses the metadata from the SampleSheet data-frame and links to open a sample's files in IGV (the sample's bigwig, bam or interval files). All sample files are associated with their relevant sample metadata and grouped together by their unique sample name.

When a sample has an interval file associated to it, the Tracktables Experiment Report also contains a link to a further sample specific **Tracktables Interval Report**. This interval report contains a table of interval locations, any metadata associated with intervals and further links to focus IGV on an interval's genomic location.

```
HTMLreport <- maketracktable(fileSheet=FileSheet,
                             SampleSheet=SampleSheet,
                             filename="IGVExample.html",
                             basedirectory=getwd(),
                             genome="mm9")
```

In this example the `maketracktables()` function creates an HTML report "IGVExample.html" (The Tracktable Experiment Report) in the current working directory alongside the relevant sample IGV session XMLs, The Tracktable Experiment Reports (named by SampleName) and the sample information file. The function also returns the XML doc to allow the user to add further customisation to the main report.

Further configuration of the report can be achieved through the use of the `colourBy` arguments and `igvParams` class object passed to the `maketracktables()` function. The `colourBy` argument accepts a character argument corresponding to the metadata column by which samples will be coloured in IGV.

The `igvParams` class defines how files will be displayed in IGV. `igvParams` accepts a range of arguments corresponding to supported IGV display methods (see reference manual for full details).

In this example, all files are colour grouped by their antibody metadata and display ranges sets to be between 1 and 5 for bigwigs files with no autoscale set.

```
igvDisplayParams <- igvParam(bigwig.autoScale = "false",
                             bigwig.minimum = 1,
                             bigwig.maximum = 5)


HTMLreport <- maketracktable(FileSheet,SampleSheet,"IGVex2.html",getwd(),"mm9",
                             colourBy="Antibody",
                             igvParam=igvDisplayParams)
```

3 Note on the use of relative paths

Since `tracktables` uses relative paths to communicate with IGV, in practice the creation of `tracktable`'s reports in a new directory, alongside any files to display, is advised. This allows for the report to be high portable and so delivered to the user as a functional unit to use with IGV.

4 Session Information

Here is the output of `sessionInfo` on the system on which this document was compiled:

```
toLatex(sessionInfo())
```

- R version 3.3.0 (2016-05-03), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: knitr 1.13, tracktables 1.6.2
- Loaded via a namespace (and not attached): BiocGenerics 0.18.0, BiocParallel 1.6.2, BiocStyle 2.0.2, Biostrings 2.40.0, GenomeInfoDb 1.8.2, GenomicRanges 1.24.0, IRanges 2.6.0, RColorBrewer 1.1-2, Rsamtools 1.24.0, S4Vectors 0.10.0, XML 3.98-1.4, XVector 0.12.0, bitops 1.0-6, evaluate 0.9, formatR 1.4, highr 0.6, magrittr 1.5, ore 1.4.0, parallel 3.3.0, reportr 1.2.1, stats4 3.3.0, stringi 1.0-1, stringr 1.0.0, tools 3.3.0, tractor.base 3.0.0, zlibbioc 1.18.0