

Accessing Genome annotations from the UCSC Genome Browser

Marc Carlson

October 28, 2009

1 Introduction

The *rtracklayer* package provides functions and methods that can be used to get the data tables behind the UCSC tracks and import them as data.frames. This vignette will explore some of these and document the capabilities with specific examples.

1.1 Retrieving Exon Boundary information

In general, when you want to get some data from UCSC, you will want to first make a session. The most common thing is that you will want a session with the UCSC genome browser, so this is the default behaviour.

```
> library(rtracklayer)
> session <- browserSession()
```

Once you have done this, you will need to choose which genome you want to work on. To do that, you should use the `ucscGenomes` function to list all the available genomes and then choose one as follows.

```
> head(ucscGenomes())
```

	db	organism	date
1	hg19	Human	Feb. 2009
2	hg18	Human	Mar. 2006
3	hg17	Human	May 2004
4	hg16	Human	July 2003
5	panTro2	Chimp	Mar. 2006
6	panTro1	Chimp	Nov. 2003

Then you can set the value of the chosen genome for your session using the **genome** command. The following command sets it to be human build hg18.

```
> genome(session) <- "hg18"
```

To search for tracks/tables are available you can use the `trackNames` method like this:

```
> head(trackNames(session))
```

Base Position	Chromosome Band	STS Markers	FISH Clones	Recomb Rate
"ruler"	"cytoBand"	"stsMap"	"fishClones"	"recombRate"
Map Contigs				
"ctgPos"				

Finally, you can retrieve the data from UCSC by using the `ucscTableQuery` command. In this case we just want to get the whole table so we will leave out the option of passing in the segment of the genome we would want to retrieve it for. The following example will create a query to retrieve the entire table/track for the `refGene` track from mouse.

```
> query <- ucscTableQuery(session, "refGene")
```

Then we can use the `getTable` method to return the data in the query.

```
> head(getTable(query))
```

[illegible]

```

6                                     58953,
                                     exonEnds id      name2
1 4692,4901,5810,6628,6918,7231,7605,7924,8229,14754,19233, 0      WASH5P
2                                     4692,4901,5810,6631,6918,7231,7502, 0 LOC100288778
3                                     25037,25344,25944, 0      FAM138F
4                                     25037,25344,25944, 0      FAM138A
5                                     25037,25344,25944, 0      FAM138C
6                                     59871, 0      OR4F5
  cdsStartStat cdsEndStat                                     exonFrames
1      unk      unk -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
2      unk      unk                                     -1,-1,-1,-1,-1,-1,-1,
3      unk      unk                                     -1,-1,-1,
4      unk      unk                                     -1,-1,-1,
5      unk      unk                                     -1,-1,-1,
6      cpl      cpl                                     0,

```

1.2 Some other Resources

Several kinds of data are available for access. Here are some tracks from human, that I expect are likely to be popular:

CPG Islands: "cpgIslandExt"

Access to genes known to be associated with disease: "gad", "omimGene"

Nucleosome Occupancy: "uwNucOcc" (this one is causing trouble)

Genomic Segmental Duplications: "genomicSuperDups"

Conserved TFBS: "tfbsConsSites"

1.3 Restricting annotations to a Genomic Region

Sometimes you may also want to restrict the amount of data you retrieve. In these cases you can pass a `GenomicRanges` object in to the `ucscTableSession` so that it will limit the values returned to only the region of interest. This can be especially true when looking at data that occurs in a lot of places in the genome such as SNPs. Below is an example that will return the SNPs on a particular region of Chromosome 12.

```

> query <- ucscTableQuery(session, "snp130",
+                           GenomicRanges(57795963, 57815592, "chr12"))
> head(getTable(query))

```

```

  bin chrom chromStart chromEnd      name score strand refNCBI refUCSC
1 1025 chr12   57796046 57796047 rs12822426     0      +      C      C

```

2	1025	chr12	57796211	57796212	rs1356171	0	-	G	G
3	1025	chr12	57796920	57796920	rs34250032	0	+	-	-
4	1025	chr12	57797200	57797201	rs2120529	0	-	C	C
5	1025	chr12	57797278	57797279	rs2120528	0	-	T	T
6	1025	chr12	57797524	57797525	rs12831695	0	+	T	T
		observed	molType		class				
1		C/T	genomic		single				
2		C/T	genomic		single				
3		-/C	genomic		insertion				
4		A/G	genomic		single				
5		A/T	genomic		single				
6		G/T	genomic		single				
								valid	avHet
1								by-cluster,by-frequency,by-hapmap	0.087777
2								unknown	0.000000
3								unknown	0.000000
4								by-cluster,by-frequency,by-2hit-2allele,by-hapmap,by-1000genomes	0.175690
5								by-cluster,by-frequency,by-2hit-2allele,by-hapmap,by-1000genomes	0.340527
6								unknown	0.000000
		avHetSE	func	locType	weight				
1		0.190220	unknown	exact	1				
2		0.000000	unknown	exact	1				
3		0.000000	unknown	between	1				
4		0.238701	unknown	exact	1				
5		0.233034	unknown	exact	1				
6		0.000000	unknown	exact	1				

1.4 Even More Resources

Here are some additional types of information that are expected to be popular:

Mapped Ests: "est"

Recombination Rates: "recombRate"

Microsatellites: "microsat"

Comparative genomics information: "chainBosTau4" (eg. compare w/bovines)

2 Session Information

The version number of R and packages loaded for generating the vignette were:

R version 2.10.0 Patched (2009-10-27 r50222)

i386-apple-darwin9.8.0

locale:

[1] C/en_US.UTF-8/C/C/C/C

attached base packages:

[1] tools stats graphics grDevices utils datasets methods

[8] base

other attached packages:

[1] rtracklayer_1.6.0 RCurl_1.2-1 bitops_1.0-4.1

loaded via a namespace (and not attached):

[1] BSgenome_1.14.0 Biobase_2.6.0 Biostrings_2.14.0 IRanges_1.4.0

[5] XML_2.6-0