

HowTo: Build and use chromosomal information

Jeff Gentry

March 4, 2011

1 Overview

The `annotate` package provides a class that can be used to model chromosomal information about a species, using one of the metadata packages provided by Bioconductor. This class contains information about the organism and its chromosomes and provides a standardized interface to the information in the metadata packages for other software to quickly extract necessary chromosomal information. An example of using *chromLocation* objects in other software can be found with the `alongChrom` function of the *geneplotter* package in Bioconductor.

2 The `chromLocation` class

The *chromLocation* class is used to provide a structure for chromosomal data of a particular organism. In this section, we will discuss the various slots of the class and the methods for interacting with them. Before this though, we will create an object of class *chromLocation* for demonstration purposes later. The helper function `buildChromLocation` is used, and it takes as an argument the name of a Bioconductor metadata package, which is itself used to extract the data. For this vignette, we will be using the *hgu95av2.db* package.

```
> library("annotate")
> z <- buildChromLocation("hgu95av2")
> z
```

Instance of a `chromLocation` class with the following fields:

```
Organism: Homo sapiens
Data source: hgu95av2
Number of chromosomes for this organism: 93
Chromosomes of this organism and their lengths in base pairs:
  1 : 249250621
  2 : 243199373
  3 : 198022430
  4 : 191154276
```

5 : 180915260
6 : 171115067
7 : 159138663
X : 155270560
8 : 146364022
9 : 141213431
10 : 135534747
11 : 135006516
12 : 133851895
13 : 115169878
14 : 107349540
15 : 102531392
16 : 90354753
17 : 81195210
18 : 78077248
20 : 63025520
Y : 59373566
19 : 59128983
22 : 51304566
21 : 48129895
6_ssto_hap7 : 4928567
6_mcf_hap5 : 4833398
6_cox_hap2 : 4795371
6_mann_hap4 : 4683263
6_apd_hap1 : 4622290
6_qbl_hap6 : 4611984
6_dbb_hap3 : 4610396
17_ctg5_hap1 : 1680828
4_ctg9_hap1 : 590426
1_gl000192_random : 547496
Un_gl000225 : 211173
4_gl000194_random : 191469
4_gl000193_random : 189789
9_gl000200_random : 187035
Un_gl000222 : 186861
Un_gl000212 : 186858
7_gl000195_random : 182896
Un_gl000223 : 180455
Un_gl000224 : 179693
Un_gl000219 : 179198
17_gl000205_random : 174588
Un_gl000215 : 172545
Un_gl000216 : 172294
Un_gl000217 : 172149
9_gl000199_random : 169874
Un_gl000211 : 166566

```

Un_gl000213 : 164239
Un_gl000220 : 161802
Un_gl000218 : 161147
19_gl000209_random : 159169
Un_gl000221 : 155397
Un_gl000214 : 137718
Un_gl000228 : 129120
Un_gl000227 : 128374
1_gl000191_random : 106433
19_gl000208_random : 92689
9_gl000198_random : 90085
17_gl000204_random : 81310
Un_gl000233 : 45941
Un_gl000237 : 45867
Un_gl000230 : 43691
Un_gl000242 : 43523
Un_gl000243 : 43341
Un_gl000241 : 42152
Un_gl000236 : 41934
Un_gl000240 : 41933
17_gl000206_random : 41001
Un_gl000232 : 40652
Un_gl000234 : 40531
11_gl000202_random : 40103
Un_gl000238 : 39939
Un_gl000244 : 39929
Un_gl000248 : 39786
8_gl000196_random : 38914
Un_gl000249 : 38502
Un_gl000246 : 38154
17_gl000203_random : 37498
8_gl000197_random : 37175
Un_gl000245 : 36651
Un_gl000247 : 36422
9_gl000201_random : 36148
Un_gl000235 : 34474
Un_gl000239 : 33824
21_gl000210_random : 27682
Un_gl000231 : 27386
Un_gl000229 : 19913
M : 16571
Un_gl000226 : 15008
18_gl000207_random : 4262

```

Once we have an object of the *chromLocation* class, we can now access its various slots to get the information contained within it. There are six slots in

this class:

```
organism:      This lists the organism that this object is describing.
dataSource:    Where this data was acquired from.
chromLocs:     A list with an element for every unique chromosome
               name, where each element contains a named vector where
               the names are probe IDs and the values describe the
               location of that probe on the chromosome. Negative
               values indicate that the location is on the antisense
               strand.
probesToChrom: A hash table which will translate a probe ID to the
               chromosome it belongs to.
chromInfo:     A numerical vector representing each chromosome, where
               the names are the names of the chromosomes and the
               values are the lengths of those chromosomes.
geneSymbols:   An environment that maps a probe ID to the appropriate
               gene symbol.
```

There is a basic 'get' type method for each of these slots, all with the same name as the respective slot. In the following example, we will demonstrate these basic methods. For the `probesToChrom` and `geneSymbols` methods, the return value is an environment which maps a probe ID to other values, we will be using the probe ID '32972_at', which was selected at random for these examples. We are showing only part of the `chromLocs` method's output as it is quite long in its entirety.

```
> organism(z)
```

```
[1] "Homo sapiens"
```

```
> dataSource(z)
```

```
[1] "hgu95av2"
```

```
> names(chromLocs(z))
```

[1] "1"	"10"	"11"
[4] "12"	"13"	"14"
[7] "15"	"16"	"17"
[10] "18"	"19"	"2"
[13] "20"	"21"	"22"
[16] "3"	"4"	"5"
[19] "6"	"7"	"8"
[22] "9"	"X"	"Y"
[25] "17_ctg5_hap1"	"6_cox_hap2"	"4_ctg9_hap1"
[28] "6_ssto_hap7"	"6_qbl_hap6"	"6_dbb_hap3"
[31] "6_mcf_hap5"	"1_g1000191_random"	"Un_g1000223"
[34] "6_apd_hap1"	"6_mann_hap4"	

```

> chromLocs(z)[["Y"]]

266_s_at 31911_at 32864_at 32991_f_at 35885_at 36321_at 37583_at
-21152525 15815446 -2654895 -6733959 14813159 14774297 -21867302
38182_at 40030_at 40097_at 41214_at 1185_at 31534_at 31534_at
21758441 7142012 22737610 2709622 1405508 2803111 2803517
34753_at 38355_at 38355_at 40435_at 40436_g_at 41138_at 629_at
59213948 15016018 15016698 -1455044 -1455044 2559227 59330251
938_at 31411_at 31411_at 31411_at 34477_at 34477_at 34477_at
59330251 -27177051 25130409 26764150 -15434914 -15409389 -15360262
32930_f_at 32930_f_at 32930_f_at 32930_f_at 34172_s_at 34172_s_at 34215_at
16634487 16635625 16636453 16733900 1660485 1660485 1660485
34215_at 35073_at 35073_at 36553_at 36553_at 36554_at 36554_at
1660485 535078 535078 -1472031 -1472031 -1472031 -1472031
39168_at 39168_at 33665_s_at 33665_s_at 33665_s_at 35447_s_at 35447_s_at
-2354454 -2354454 1337692 1337692 1351570 1664347 1683940
35447_s_at
1684025

> get("32972_at", probesToChrom(z))

[1] "X"

> chromInfo(z)

1 2 3 4
249250621 243199373 198022430 191154276
5 6 7 X
180915260 171115067 159138663 155270560
8 9 10 11
146364022 141213431 135534747 135006516
12 13 14 15
133851895 115169878 107349540 102531392
16 17 18 20
90354753 81195210 78077248 63025520
Y 19 22 21
59373566 59128983 51304566 48129895
6_ssto_hap7 6_mcf_hap5 6_cox_hap2 6_mann_hap4
4928567 4833398 4795371 4683263
6_apd_hap1 6_qbl_hap6 6_dbb_hap3 17_ctg5_hap1
4622290 4611984 4610396 1680828
4_ctg9_hap1 1_g1000192_random Un_g1000225 4_g1000194_random
590426 547496 211173 191469
4_g1000193_random 9_g1000200_random Un_g1000222 Un_g1000212
189789 187035 186861 186858
7_g1000195_random Un_g1000223 Un_g1000224 Un_g1000219
182896 180455 179693 179198

```

17_gl000205_random	Un_gl000215	Un_gl000216	Un_gl000217
174588	172545	172294	172149
9_gl000199_random	Un_gl000211	Un_gl000213	Un_gl000220
169874	166566	164239	161802
Un_gl000218	19_gl000209_random	Un_gl000221	Un_gl000214
161147	159169	155397	137718
Un_gl000228	Un_gl000227	1_gl000191_random	19_gl000208_random
129120	128374	106433	92689
9_gl000198_random	17_gl000204_random	Un_gl000233	Un_gl000237
90085	81310	45941	45867
Un_gl000230	Un_gl000242	Un_gl000243	Un_gl000241
43691	43523	43341	42152
Un_gl000236	Un_gl000240	17_gl000206_random	Un_gl000232
41934	41933	41001	40652
Un_gl000234	11_gl000202_random	Un_gl000238	Un_gl000244
40531	40103	39939	39929
Un_gl000248	8_gl000196_random	Un_gl000249	Un_gl000246
39786	38914	38502	38154
17_gl000203_random	8_gl000197_random	Un_gl000245	Un_gl000247
37498	37175	36651	36422
9_gl000201_random	Un_gl000235	Un_gl000239	21_gl000210_random
36148	34474	33824	27682
Un_gl000231	Un_gl000229	M	Un_gl000226
27386	19913	16571	15008
18_gl000207_random			
4262			

```
> get("32972_at", geneSymbols(z))
```

```
[1] "NOX1"
```

Another method which can be used to access information about the particular *chromLocation* object is the `nChrom` method, which will list how many chromosomes this organism has:

```
> nChrom(z)
```

```
[1] 93
```

3 Summary

The *chromLocation* class has a simple design, but can be powerful if one wants to store the chromosomal data contained in a Bioconductor package into a single object. These objects can be created once and then passed around to multiple functions, which can cut down on computation time to access the desired information from the package. These objects allow access to basic but also important information, and provide a standard interface for writers of other software to access this information.