Externalized expression plus genotype data structures

VJ Carey

August 25, 2011

1 Introduction

The smlSet class of *GGBase* has served for some time as a holistic representation of an expression genetics experiment. As SNP panels have grown to 8 million plus, it is impractical to employ unified containers for such experiments.

This document describes the first phase of externalizing large genotype data resources for expression genetics data structures. Given an smlSet instance, the `externalize()` function will create an R source package with chromosome-specific files of genotype data in `snpStats::SnpMatrix` instances. When this package is installed and loaded, the `getSS` function will collect information necessary to construct an smlSet with a specified number of chromosomes.

2 HapMap Phase II for expression genetics

The Bioconductor GGdata package has represented the Phase II HapMap data for CEPH/CEU cohort as a single smlSet instance. Loading the data for use on an 8GB macbook pro takes

```
> library(GGdata)
Loading required package: illuminaHumanv1.db
> unix.time(data(hmceuB36))
   user  system elapsed
 40.309   0.729  41.581
```

At this point we can access 4 million SNP for each subject very easily, but we seldom need to do this atomically; we would be just as happy to step across chromosomes and so the slow load and large memory footprint

```
> gc() # after above
           used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  7166465 382.8   10591793 565.7  7167437 382.8
Vcells 73027380 557.2   80745864 616.1 73028403 557.2
```

are inessential results of a holistic representation approach that is not effective with very large SNP panels.

The externalize() function can be called

```
> externalize(hmceuB36, "ggdPack")
now install ggdPack
NULL
```

After installation we can use

```
library(ggdPack)
```

which prints the message

```
To get a tailored smlSet, use getSS("ggdPack", [chrvec])
available chromosomes are named  1 10 ... X Y
```

We use `getSS`

```
> unix.time(gg1 <- getSS("ggdPack", "1"))
   user  system elapsed
  3.514   0.090   3.605
> gg1
SnpMatrix-based genotype set:
number of samples:  90
number of chromosomes present:  1
annotation: illuminaHumanv1.db
Expression data dims: 47293 x 90
Phenodata: An object of class "AnnotatedDataFrame"
  sampleNames: NA06985 NA06991 ... NA12892 (90 total)
  varLabels: famid persid ... male (7 total)
  varMetadata: labelDescription
> gc()
          used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  3504115 187.2    4953636 264.6  3885278 207.5
Vcells 10444243  79.7   12242489  93.5 11388329  86.9
```

and see rapid access and a relatively small footprint.

We can easily visit all chromosomes, as this naive code illustrates:

```
> allc = as.character(1:22)
> unix.time(ll <- lapply(allc, function(x) getSS("ggdPack", x)))
   user  system elapsed
 68.533   1.493  70.131
> gc()
            used   (Mb) gc trigger   (Mb)  max used   (Mb)
Ncells   7053675  376.8   10591793  565.7   8189107  437.4
Vcells 163291667 1245.9  173709474 1325.3 165150995 1260.1
```

The memory footprint of growing the entire list is unpleasant. Also, there are excessive copies of the expression data (one for each chromosome).

Instead, if we process and then discard:

```
> unix.time(ll <- lapply(allc, function(x) {tmp <- getSS("ggdPack", x); NULL}))
   user  system elapsed
 44.805   0.826  45.743
> gc()
          used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 3148343 168.2    5241317 280.0  4169255 222.7
Vcells 1828070  14.0    9260197  70.7 11572531  88.3
```

all is well.