CGEN(Case-control.GENetics) Package

October 18, 2010

```
> library(CGEN)
```

Example of snp.logistic

Load the ovarian cancer data and print the first 5 rows.

```
> data(Xdata, package = "CGEN")
> Xdata[1:5, ]
```

```
    id case.control BRCA.status oral.years n.children age.group ethnic.group
1 sub1            0           0          0          1         1            3
2 sub2            1           1          0          2         4            1
3 sub3            0           0          0          2         4            1
4 sub4            1           0          0          3         3            1
5 sub5            1           0          0          3         1            2
  BRCA.history gynSurgery.history family.history
1            0                  0              0
2            0                  0              2
3            0                  0              0
4            0                  0              0
5            0                  0              0
```

For this analysis, the main effects will be "age.group", "n.children", and "oral.years". We will let "age.group" be a categorical variable in the model and we will create dummy variables for it. The dummy variables will be called "age.group_1", "age.group_2", ... "age.group_5".

```
> for (a in unique(Xdata[, "age.group"])) {
+     dummyVar <- paste("age.group_", a, sep = "")
+     Xdata[, dummyVar] <- 0
+     temp <- Xdata[, "age.group"] == a
+     if (any(temp))
+         Xdata[temp, dummyVar] <- 1
+ }
```

To determine the baseline category, and if any categories need to be combined, get the frequency counts for the age.group variable by case-control status.

```
> table(Xdata[, "case.control"], Xdata[, "age.group"], exclude = NULL)
```

```
          1   2   3   4   5 <NA>
0        68 137 155 218 169    0
1        31 163 205 240 193    0
<NA>      0   0   0   0   0    0
```

We will let "age.group_4" will be the reference category, "case.control" be the response variable and "BRCA.status" be the SNP variable. Let the variables "oral.years" and "n.children" also interact with the SNP variable. Also let the stratification variable for the constrained maximum likelihood method (CML) be "ethnic.group".

```
> mainVars <- c("oral.years", "n.children", "age.group_1", "age.group_2",
+     "age.group_3", "age.group_5")
> fit <- snp.logistic(Xdata, "case.control", "BRCA.status", main.vars = mainVars,
+     int.vars = c("oral.years", "n.children"), strata.var = "ethnic.group")
```

Compute a summary table for the models.

```
> getSummary(fit)
```

$UML

	Estimate	Std.Error	Z.value	Pvalue
Intercept	-0.05218922	0.13346145	-0.3910434	6.957651e-01
oral.years	-0.04931037	0.02570532	-1.9182947	5.507366e-02
n.children	-0.03611838	0.02911900	-1.2403715	2.148380e-01
age.group_1	-0.74504465	0.25261026	-2.9493840	3.184081e-03
age.group_2	-0.03278004	0.16379592	-0.2001273	8.413810e-01
age.group_3	0.03711828	0.15484252	0.2397163	8.105502e-01
age.group_5	0.07350744	0.14974476	0.4908849	6.235079e-01
BRCA.status	3.63850273	0.65340234	5.5685487	2.568699e-08
BRCA.status:oral.years	0.05282430	0.10376623	0.5090703	6.107030e-01
BRCA.status:n.children	-0.19683136	0.21145427	-0.9308460	3.519332e-01

$CML

	Estimate	Std.Error	Z.value	Pvalue
Intercept	-0.075537129	0.13031518	-0.57964951	5.621510e-01
oral.years	-0.055886604	0.02586138	-2.16100616	3.069486e-02
n.children	-0.040560944	0.02957507	-1.37145706	1.702325e-01
age.group_1	-0.861617067	0.24071420	-3.57941940	3.443584e-04
age.group_2	0.100160977	0.15130724	0.66197083	5.079899e-01
age.group_3	0.202959708	0.14268825	1.42239955	1.549103e-01
age.group_5	-0.005850914	0.14186749	-0.04124211	9.671029e-01
BRCA.status	3.336345966	0.32522287	10.25864501	1.082145e-24
BRCA.status:oral.years	0.082422493	0.03077170	2.67851571	7.394926e-03
BRCA.status:n.children	-0.083435714	0.04956214	-1.68345671	9.228671e-02

$EB

	Estimate	Std.Error	Z.value	Pvalue
Intercept	-0.07484380	0.13037388	-0.57407049	5.659201e-01
oral.years	-0.05548263	0.02667243	-2.08014868	3.751190e-02
n.children	-0.04045989	0.02964999	-1.36458350	1.723840e-01

```
age.group_1          -0.84115072 0.24361689 -3.45276025 5.548819e-04
age.group_2           0.04736591 0.16101815  0.29416502 7.686318e-01
age.group_3           0.11435760 0.15559408  0.73497398 4.623553e-01
age.group_5           0.01155011 0.14484740  0.07973982 9.364442e-01
BRCA.status           3.38957782 0.42001695  8.07009773 7.024192e-16
BRCA.status:oral.years  0.08019553 0.03790896  2.11547693 3.438932e-02
BRCA.status:n.children -0.10876260 0.12751323 -0.85295146 3.936862e-01
```

Compute Wald tests for the main effect of the SNP and interactions.

```
> getWaldTest(fit, c("BRCA.status", "BRCA.status:oral.years", "BRCA.status:n.children"))

$UML
$UML$test
[1] 110.048

$UML$df
[1] 3

$UML$pvalue
[1] 1.071535e-23


$CML
$CML$test
[1] 122.7265

$CML$df
[1] 3

$CML$pvalue
[1] 1.996149e-26


$EB
$EB$test
[1] 120.3581

$EB$df
[1] 3

$EB$pvalue
[1] 6.461091e-26
```

Example of snp.scan.logistic

The snp.scan.logistic function is used for computing many single SNP analyses. Unlike the snp.logistic function where the data must exist in memory when calling the function, the snp.scan.logistic function requires that the data be in separate external files. One file is for the genotype data and the other

is for the phenotype data. Both files must contain subject ids in in order to link the two together. For this example, the sample genotype data file is delimited by a vertical bar ("|") and the first row contains the subject ids. Starting from the second row is the SNP name followed by the genotypes for each subject. Missing genotypes are denoted by "NA". Get the path to the genotype data and define the snp.list input argument.

```
> g <- system.file("sampleData", "SNPdata.rda", package = "CGEN")
> snp.list <- list(file = g, file.type = 1, delimiter = "|", in.miss = "NA")
```

Define the pheno.list argument. Use the ovarian cancer data.

```
> f <- system.file("sampleData", "Xdata.txt", package = "CGEN")
> pheno.list <- list(file = f, id.var = "id", file.type = 3, delimiter = "\t",
+     response.var = "case.control", strata.var = "ethnic.group",
+     main.vars = c("oral.years", "n.children"), int.vars = "oral.years")
```

Define the output file and options list.

```
> out.file <- "snp.scan.logistic.output.file_h5jb47j.txt"
> op <- list(out.file = out.file)
```

Call the scan function. Since the "cc.var" option was not specified in pheno.list, the minor allele for each SNP will be determined from using all subjects instead of using only the controls, and a note will be printed stating that "cc.var" was not specified in pheno.list.

```
> ret <- snp.scan.logistic(snp.list, pheno.list, op = op)

[1] "snp.list$stream is set to 0 for file.type = 1"
[1] "For the analysis, 1579 observations will be used."
response0
  0   1
747 832
```

Read in the output file and print the first 5 rows.

```
> x <- read.table(out.file, sep = "\t", header = TRUE)
> x[1:5, ]
```

	SNP	Alleles	MAF	CML.omnibus.test	CML.omnibus.pvalue
1	rs11102647	AT	0.4873096	0.8972792	0.6384962
2	rs6695241	GC	0.4971501	0.6640485	0.7174699
3	rs12567796	GC	0.4694561	0.8216977	0.6630871
4	rs2810583	TA	0.4920836	1.9778100	0.3719837
5	rs4654986	TA	0.4967078	1.3230070	0.5160747

	CML.omnibus.df	UML.omnibus.test	UML.omnibus.pvalue	UML.omnibus.df
1	2	0.06016725	0.9703644	2
2	2	1.18504400	0.5529310	2
3	2	0.78915980	0.6739631	2
4	2	1.92497300	0.3819420	2
5	2	0.15666350	0.9246576	2

```
  EB.omnibus.test EB.omnibus.pvalue EB.omnibus.df
```

```
1        0.4779968          0.7874162              2
2        0.6772118          0.7127633              2
3        0.7921235          0.6729651              2
4        1.9493000          0.3773244              2
5        0.4716941          0.7899015              2
```

Create a QQ-plot of the unconstrained p-values.

```
> ret <- QQ.plot(x[, "UML.omnibus.pvalue"])
```

Get the path to the locus map file which contains the chromosome and location of each SNP.

```
> map <- system.file("sampleData", "LocusMapData.txt", package = "CGEN")
```

Display the first 5 lines of this file and define the list for the locus map data.

```
> read.table(map, sep = "\t", header = 1, nrows = 5)
```

```
          SNP CHROMOSOME   LOCATION
1 rs11102647           1 113783261
2  rs6695241           1 172626514
3 rs12567796           1  18262009
4  rs2810583           1  41549436
5  rs4654986           1  21883504
```

```
> lmap.list <- list(file = map, file.type = 3, header = 1, delimiter = "\t",
+     snp.var = "SNP", chrm.var = "CHROMOSOME", loc.var = "LOCATION")
```

Define the vector of p-value variables to be plotted and the options list for the chromosome.plot function.

```
> plot.vars <- c("UML.omnibus.pvalue", "CML.omnibus.pvalue", "EB.omnibus.pvalue")
> op <- list(pch = c(21, 22, 23), alt.colors = 1)
```

Create a chromosome plot of the p-values from all methods.

```
> ret <- chromosome.plot(out.file, plot.vars, lmap.list, op = op)
```

Get the top 10 hits of the scan based on the unconstrained maximum likelihood method.

```
> temp <- sort(x[, "UML.omnibus.pvalue"], index.return = TRUE)$ix
> topHits <- x[temp, ][1:10, ]
> topHits
```

```
          SNP Alleles       MAF CML.omnibus.test CML.omnibus.pvalue
105   rs679218      AT 0.4894870         14.428390       0.0007360638
162  rs1899650      AT 0.4846416         11.969130       0.0025173100
229  rs5928051      AT 0.4946168         11.473950       0.0032245000
83   rs6476055      TA 0.4867004         11.039370       0.0040071180
66   rs7800565      GC 0.4987334          5.656813       0.0591069700
205  rs2828958      CG 0.4962001          6.181401       0.0454700900
131   rs234584      GC 0.4841672         10.408230       0.0054939100
111  rs7299271      GC 0.4840426          8.977490       0.0112347300
```

```
197 rs4254559     GC 0.4851172      11.427770      0.0032998290
55  rs6914547     AT 0.4881557       9.546990      0.0084507910
    CML.omnibus.df UML.omnibus.test UML.omnibus.pvalue UML.omnibus.df
105              2         10.199950          0.006096903              2
162              2          8.490510          0.014332080              2
229              2          8.268689          0.016013160              2
83               2          8.089834          0.017511160              2
66               2          7.506675          0.023439390              2
205              2          7.151767          0.027990680              2
131              2          6.939406          0.031126270              2
111              2          6.670272          0.035609740              2
197              2          6.484036          0.039084950              2
55               2          5.998464          0.049825310              2
    EB.omnibus.test EB.omnibus.pvalue EB.omnibus.df
105       14.612070      0.0006714734              2
162       11.756140      0.0028001890              2
229       11.719030      0.0028526320              2
83        10.919690      0.0042542190              2
66         6.093185      0.0475205600              2
205        5.785618      0.0554203100              2
131        9.596130      0.0082456870              2
111        8.840032      0.0120340400              2
197        9.045870      0.0108571200              2
55         9.641040      0.0080625900              2
```

Example of snp.matched

First let us use "age.group1", "gynSurgery.history" and "BRCA.history" to match the subjects finely into small sets. We will perform the matching only within each ethnic group. We check the case control distribution within ethnic groups.

```
> table(Xdata$case.control, Xdata$ethnic.group)

      1   2   3
  0 509 183  55
  1 593 193  46
```

Thus, allowing matched sets of size 3 should be enough to match all the subjects in each ethnic group. For illustration, let us use maximum matched set size of 4 for ethnic groups 1 and 2 and that of 3 for ethnic group 3. Let us use daisy to compute the distance matrix, which automatically chooses Gower's distance if there are one or more categorical variables.

```
> library("cluster")
> size <- ifelse(Xdata$ethnic.group == 3, 3, 4)
> d <- daisy(Xdata[, c("age.group_1", "gynSurgery.history", "BRCA.history")])
> mx <- getMatchedSets(d, CC = TRUE, NN = TRUE, ccs.var = Xdata$case.control,
+     strata.var = Xdata$ethnic.group, size = size, fixed = FALSE)
```

The return object mx contains vectors corresponding to CC and NN matching as well as corresponding summary matrices tblCC and tblNN. Summaries

can be inspected to see how many matched sets of each size were created (along rows) for each of the ethnic groups (along columns). The strata vectors are then appended to the data.frame, before calling the analysis function snp.matched.

```
> mx$CC[1:10]

 [1]  1  3  3  4  6  6  9  4 10  9

> mx$tblCC

      strat
        1   2  3
  [1,]  78   4  0
  [2,] 506 180 37
  [3,]   0   0  9
  [4,]   3   3  0

> Xdata <- cbind(Xdata, CCStrat = mx$CC, NNStrat = mx$NN)
> Xdata[1:5, ]

    id case.control BRCA.status oral.years n.children age.group ethnic.group
1 sub1            0           0          0          1         1            3
2 sub2            1           1          0          2         4            1
3 sub3            0           0          0          2         4            1
4 sub4            1           0          0          3         3            1
5 sub5            1           0          0          3         1            2
  BRCA.history gynSurgery.history family.history age.group_1 age.group_4
1            0                  0              0           1           0
2            0                  0              2           0           1
3            0                  0              0           0           1
4            0                  0              0           0           0
5            0                  0              0           1           0
  age.group_3 age.group_2 age.group_5 CCStrat NNStrat
1           0           0           0       1       1
2           0           0           0       3       2
3           0           0           0       3       2
4           1           0           0       4       2
5           0           0           0       6       3
```

We will look at the interaction of BRCA.status with oral.years and n.children using formulas.

```
> intVars <- ~oral.years + n.children
> snpVars <- ~BRCA.status
> fit <- snp.matched(Xdata, "case.control", snp.vars = snpVars,
+     main.vars = intVars, int.vars = intVars, cc.var = "CCStrat",
+     nn.var = "NNStrat")
```

Compute a summary table for the fitted CLR and CCL models.

```
> getSummary(fit, method = c("CLR", "CCL"))

list()
```

Compute Wald tests for the omnibus effect of BRCA.status for the HCL method.

```
> getWaldTest(fit$HCL, c("BRCA.status", "BRCA.status:oral.years",
+       "BRCA.status:n.children"))

$test
[1] NA

$df
[1] 0

$pvalue
[1] NA
```

Session Information

```
> sessionInfo()

R version 2.12.0 RC (2010-10-11 r53293)
Platform: i386-pc-mingw32/i386 (32-bit)

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] splines   stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] cluster_1.13.1  CGEN_1.2.0      survival_2.35-8

loaded via a namespace (and not attached):
[1] tools_2.12.0
```