

CGEN(Case-control.GENetics) Package

July 21, 2026

```
> 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
0  68 137 155 218 169
1  31 163 205 240 193

```

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	-0.075567125	0.13031659	-0.57987342	5.620000e-01
oral.years	-0.055859074	0.02585978	-2.16007523	3.076685e-02
n.children	-0.040560606	0.02957534	-1.37143331	1.702399e-01
age.group_1	-0.861804761	0.24071782	-3.58014520	3.434033e-04
age.group_2	0.100146409	0.15130732	0.66187420	5.080518e-01
age.group_3	0.202837978	0.14268782	1.42155077	1.551567e-01
age.group_5	-0.005926216	0.14186752	-0.04177289	9.666797e-01
BRCA.status	3.337052065	0.32525743	10.25972601	1.070099e-24
BRCA.status:oral.years	0.082378446	0.03077064	2.67717694	7.424542e-03
BRCA.status:n.children	-0.083520091	0.04956159	-1.68517780	9.195427e-02

\$EB

	Estimate	Std.Error	Z.value	Pvalue
Intercept	-0.07487117	0.13037548	-0.5742734	5.657828e-01
oral.years	-0.05545995	0.02666606	-2.0797956	3.754429e-02

n.children	-0.04045957	0.02965025	-1.3645609	1.723911e-01
age.group_1	-0.84125102	0.24363153	-3.4529644	5.544621e-04
age.group_2	0.04736410	0.16101655	0.2941567	7.686382e-01
age.group_3	0.11436147	0.15558658	0.7350342	4.623186e-01
age.group_5	0.01151712	0.14485344	0.0795088	9.366279e-01
BRCA.status	3.38995511	0.41948315	8.0812665	6.409754e-16
BRCA.status:oral.years	0.08016092	0.03787611	2.1163976	3.431100e-02
BRCA.status:n.children	-0.10879888	0.12741046	-0.8539242	3.931470e-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] 122.7287
```

```
$CML$df
```

```
[1] 3
```

```
$CML$pvalue
```

```
[1] 1.993932e-26
```

```
$EB
```

```
$EB$test
```

```
[1] 120.381
```

```
$EB$df
```

```
[1] 3
```

```
$EB$pvalue
```

```
[1] 6.388153e-26
```

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 = TRUE)
```

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] 355 29 73 72 272 272 30 42 30 42
```

```
> mx$tblCC
```

```
      strat
      1  2  3
[1,] 86 12 32
[2,]  0  0  0
[3,]  0  0 23
[4,] 254 91  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	355	86
2	0	0	0	29	48
3	0	0	0	73	48
4	1	0	0	72	48
5	0	0	0	272	77

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"))
```

```
$CLR
              Estimate Std. Error  Z.value      Pvalue
BRCA.status      4.16822318 0.78192505  5.3307196 9.782438e-08
oral.years       -0.05347220 0.02682722 -1.9932071 4.623878e-02
n.children        -0.04837260 0.03255916 -1.4856831 1.373630e-01
BRCA.status:oral.years 0.01770353 0.10022474  0.1766383 8.597925e-01
BRCA.status:n.children -0.25123219 0.24434755 -1.0281756 3.038672e-01
```

```
$CCL
              Estimate Std. Error  Z.value      Pvalue
BRCA.status      3.50438133 0.39501310  8.8715573 7.213002e-19
oral.years       -0.05745608 0.02668810 -2.1528726 3.132870e-02
n.children        -0.04862828 0.03290015 -1.4780565 1.393927e-01
BRCA.status:oral.years 0.12175915 0.04049286  3.0069293 2.639012e-03
BRCA.status:n.children -0.03125679 0.06495237 -0.4812263 6.303557e-01
```

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] 119.3774
```

```
$df
[1] 3
```

```
$pvalue
[1] 1.050813e-25
```

Session Information

```
> sessionInfo()
```

```
R version 4.6.1 (2026-06-24)
```

```
Platform: x86_64-pc-linux-gnu
```

```
Running under: Ubuntu 26.04 LTS
```

```
Matrix products: default
```

```
BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-p0.3.32.so; LAPACK version
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8
```

```
LC_NUMERIC=C
```

```
[3] LC_TIME=en_US.UTF-8
```

```
LC_COLLATE=en_US.UTF-8
```

```
[5] LC_MONETARY=en_US.UTF-8
```

```
LC_MESSAGES=en_US.UTF-8
```

```
[7] LC_PAPER=en_US.UTF-8
```

```
LC_NAME=C
```

```
[9] LC_ADDRESS=C
```

```
LC_TELEPHONE=C
```

```
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
time zone: Etc/UTC
```

```
tzcode source: system (glibc)
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] cluster_2.1.8.2 CGEN_3.48.0      mvtnorm_1.4-2    survival_3.8-9
```

```
loaded via a namespace (and not attached):
```

```
[1] xfun_0.60      Matrix_1.7-5    lattice_0.22-9   splines_4.6.1
```

```
[5] maketools_1.3.2 knitr_1.51      buildtools_1.0.0 cli_3.6.6
```

```
[9] grid_4.6.1     compiler_4.6.1  sys_3.4.3        tools_4.6.1
```

```
[13] evaluate_1.0.5 rlang_1.3.0
```