iCARE (Individualized Coherent Absolute Risk Estimation) Package

December 10, 2018

Load the iCARE library

```
> library(iCARE)
```

Load the breast cancer data and set the seed.

```
> data("bc_data", package="iCARE")
> set.seed(50)
```

Example 1: SNP-only model

In this example, we will estimate the risk of breast cancer in ages 50-80.A SNP-only model is fit, with no specific genotypes supplied for estimation. The population disease rates are from SEER.

```
> res_snps_miss = computeAbsoluteRisk(model.snp.info = bc_15_snps,
+                     model.disease.incidence.rates = bc_inc,
+                     model.competing.incidence.rates = mort_inc,
+                     apply.age.start = 50, apply.age.interval.length = 30,
+                     return.refs.risk = TRUE)

Note: You did not provide apply.snp.profile.  Will impute SNPs for 10000 people.
If require more, please provide apply.snp.profile input.
[1] "Note: As specified, the model does not adjust SNP imputations for family history."
   user  system elapsed
 15.740   0.016  19.823
```

Compute a summary of the risks.

```
> summary(res_snps_miss$refs.risk)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.07652 0.09198 0.09575 0.09583 0.09950 0.11966
```

Next, suppose we want to predict risk for three specific women whom we have genotyped; we can then call:

```
> res_snps_dat = computeAbsoluteRisk(model.snp.info = bc_15_snps,
+                      model.disease.incidence.rates = bc_inc,
+                      model.competing.incidence.rates = mort_inc,
+                      apply.age.start = 50, apply.age.interval.length = 30,
+                      apply.snp.profile = new_snp_prof,
+                      return.refs.risk = TRUE)

[1] "Note: As specified, the model does not adjust SNP imputations for family history."
   user  system elapsed
  0.332   0.016   0.349

> names(res_snps_dat)

[1] "risk"      "details"   "beta.used" "refs.risk"
```

These results allow us to create a useful plot showing the distribution of risks
in our reference dataset and to add the risks of the three women to see where
they fall on the population distribution.

```
> plot(density(res_snps_dat$refs.risk),
+              xlim = c(0.07,0.14), xlab = "Absolute Risk of Breast Cancer",
+              main = "Referent SNP-only Risk Distribution: Ages 50-80 years")
> abline(v = res_snps_dat$risk, col = "red")
> legend("topright", legend = "New profiles", col = "red", lwd = 1)
```

Referent SNP−only Risk Distribution: Ages 50−80 years
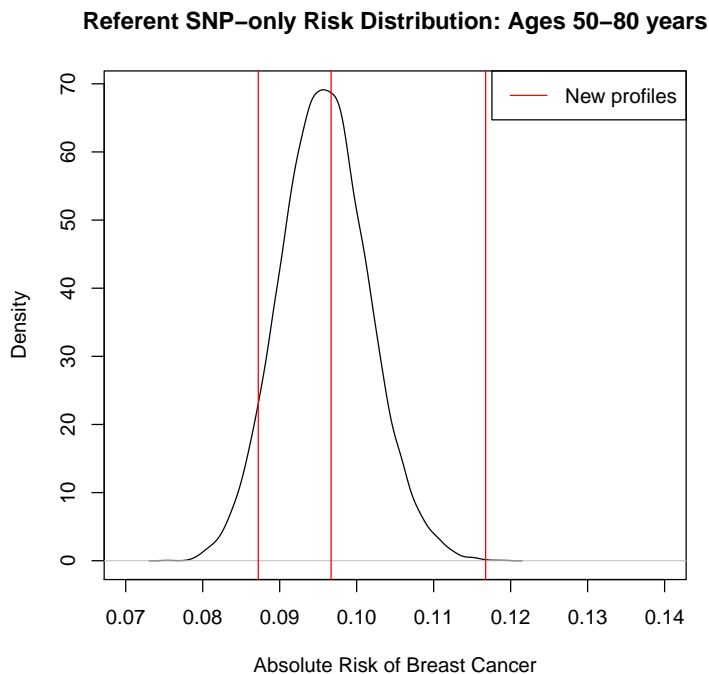
Example 2: Breast cancer risk model with risk-factors and SNPs

In this example, we will estimate the risk of breast cancer in ages 50-80 by fitting a model with two risk factors and 15 SNPs, with three specific covariate profiles supplied for estimation (with some missing data). First, the model.cov.info argument is created.

```
> v1 = list(); v1$name = "famhist"; v1$type = "continuous"
> v2 = list(); v2$name = "parity"; v2$type = "factor";
>                                  v2$levels = c(0,1,2,3,4)
> bc_model_cov_info = list(v1,v2)
> bc_model_formula = observed.outcome ~ famhist + as.factor(parity)
> res_covs_snps = computeAbsoluteRisk(model.formula = bc_model_formula,
+                                 model.cov.info = bc_model_cov_info,
+                                 model.snp.info = bc_15_snps,
+                                 model.log.RR = bc_model_log_or,
+                                 model.ref.dataset = ref_cov_dat,
+                                 model.disease.incidence.rates = bc_inc,
+                                 model.competing.incidence.rates = mort_inc,
+                                 model.bin.fh.name = "famhist",
+                                 apply.age.start = 50,
+                                 apply.age.interval.length = 30,
+                                 apply.cov.profile = new_cov_prof,
+                                 apply.snp.profile = new_snp_prof,
+                                 return.refs.risk = TRUE)

   user  system elapsed
  0.068   0.012   0.084
```

In addition to summarizing and plotting the risk estimates, iCARE includes an option to view more detailed output, by calling:

```
> print(res_covs_snps$details)
```

	Int_Start	Int_End	Risk_Estimate	rs12405132	rs12048493	rs72755295
193502	50	80	0.08077312	NA	NA	0
126252	50	80	0.07709955	0	0	0
15756	50	80	0.12955390	0	1	0

	rs6796502	rs13162653	rs2012709	rs7707921	rs9257408	rs4593472	rs13365225
193502	0	1	1	0	1	1	1
126252	0	2	1	0	1	1	0
15756	1	0	1	0	1	1	0

	rs13267382	rs11627032	rs146699004	rs745570	rs6507583	famhist	parity
193502	0	0	1	2	0	0	2
126252	1	1	0	1	0	0	4
15756	1	0	0	0	0	1	4

Illustration of the validation component

We want to validate a model for predicting absolute risk of disease based on a combined model of two epidemiologic risk factors family history and parity and

15 SNPs using the nested case-control dataset.

The first step is to compute sampling weights. We fit a logistic regression model of inclusion depending on the case/control status, age of study entry and observed followup using the R function **glm**, as shown below:

```
> validation.cohort.data$inclusion = 0
> subjects_included = intersect(validation.cohort.data$id,
+                                 validation.nested.case.control.data$id)
> validation.cohort.data$inclusion[subjects_included] = 1
> validation.cohort.data$observed.followup =
+                         validation.cohort.data$study.exit.age -
+                         validation.cohort.data$study.entry.age
> selection.model = glm(inclusion ~ observed.outcome
+                         * (study.entry.age + observed.followup),
+                             data = validation.cohort.data,
+                             family = binomial(link = "logit"))
> validation.nested.case.control.data$sampling.weights =
+         selection.model$fitted.values[validation.cohort.data$inclusion == 1]
```

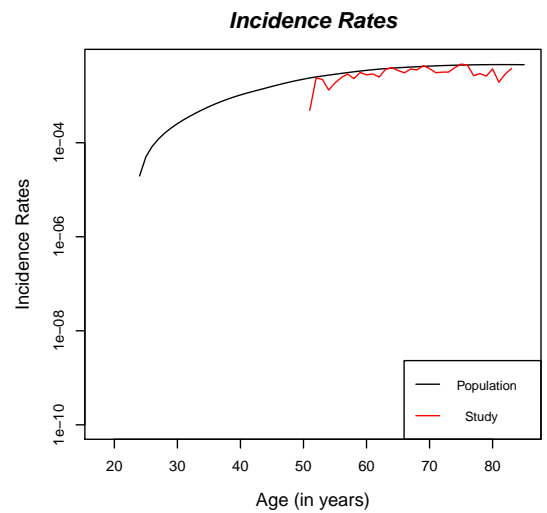The next step is to call the **ModelValidation** function to implement the validation analysis.
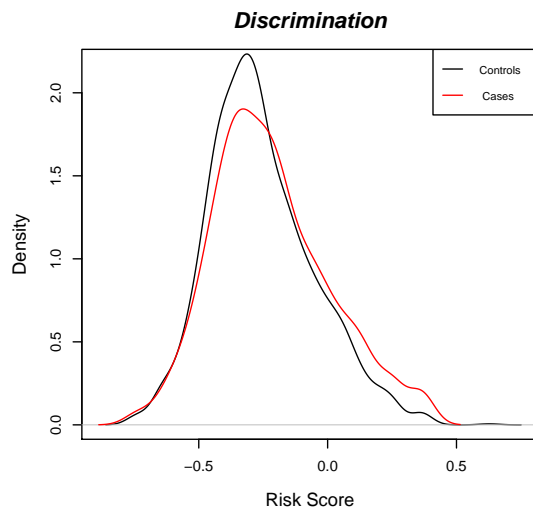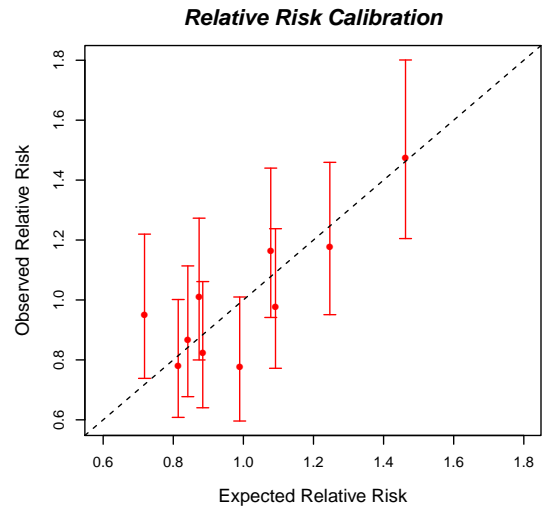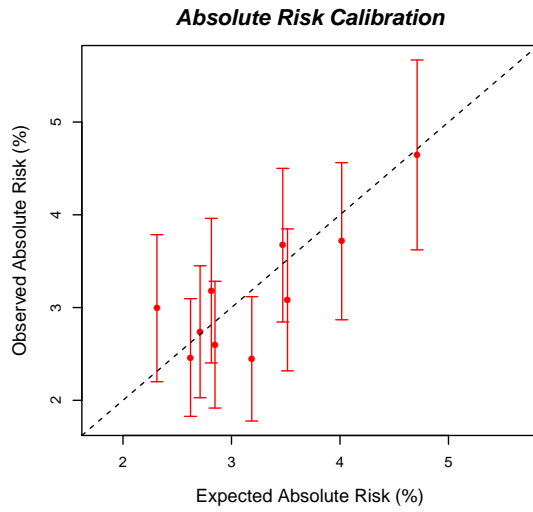
```
> bc_model_formula = observed.outcome ~ famhist + as.factor(parity)
> data = validation.nested.case.control.data
> risk.model = list(model.formula = bc_model_formula,
+                   model.cov.info = bc_model_cov_info,
+                   model.snp.info = bc_15_snps,
+                   model.log.RR = bc_model_log_or,
+                   model.ref.dataset = ref_cov_dat,
+                   model.ref.dataset.weights = NULL,
+                   model.disease.incidence.rates = bc_inc,
+                   model.competing.incidence.rates = mort_inc,
+                   model.bin.fh.name = "famhist",
+                   apply.cov.profile = data[,all.vars(bc_model_formula)[-1]],
+                   apply.snp.profile = data[,bc_15_snps$snp.name],
+                   n.imp = 5, use.c.code = 1, return.lp = TRUE,
+                   return.refs.risk = TRUE)
> output = ModelValidation(study.data = data,
+                           total.followup.validation = TRUE,
+                           predicted.risk.interval = NULL,
+                           iCARE.model.object = risk.model,
+                           number.of.percentiles = 10)

   user  system elapsed
  3.948   0.000   3.957
```

We can also produce a set of useful plots showing the results of the validation analysis.

```
> plotModelValidation(study.data = data,validation.results = output)

NULL
```

Absolute Risk Calibration

Expected Absolute Risk (%) vs Observed Absolute Risk (%)

Relative Risk Calibration

Expected Relative Risk vs Observed Relative Risk

Discrimination

Risk Score vs Density (Controls, Cases)

Incidence Rates

Age (in years) vs Incidence Rates (Population, Study)

Dataset: Example Dataset

Model Name: Example Model

Risk Prediction Interval: Observed Followup

Number of subjects (cases): 2694 (1112)

Follow−up time (years) [mean,range]: [9.782 , (5 , 13)]

Baseline age (years) [mean,range]: [62.721 , (50 , 72)]

E/O [Estimate, 95% CI]: [1.024 , (0.948 , 1.107)]

Absolute Risk Calibration	**Relative Risk Calibration**	**Model Discrimination**
HL Test, df: 11.086 , 10	Test, df: 10.685 , 9	AUC est: 0.54
p−value: 3.50876e−01	p−value: 2.979384e−01	95% CI: (0.517 , 0.563)

Session Information

> sessionInfo()

```
R version 3.5.1 Patched (2018-07-12 r74967)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.5 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.8-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.8-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [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

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

other attached packages:
[1] iCARE_1.10.3   Hmisc_4.1-1       ggplot2_3.1.0  Formula_1.2-3
[5] survival_2.43-3 lattice_0.20-38 gtools_3.8.1    plotrix_3.7-4

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.0          pillar_1.3.0      compiler_3.5.1
 [4] RColorBrewer_1.1-2  plyr_1.8.4        bindr_0.1.1
 [7] base64enc_0.1-3     tools_3.5.1       digest_0.6.18
[10] rpart_4.1-13        checkmate_1.8.5   htmlTable_1.12
[13] tibble_1.4.2        gtable_0.2.0      pkgconfig_2.0.2
[16] rlang_0.3.0.1       Matrix_1.2-15     rstudioapi_0.8
[19] bindrcpp_0.2.2      gridExtra_2.3     stringr_1.3.1
[22] knitr_1.20          withr_2.1.2       dplyr_0.7.8
[25] cluster_2.0.7-1     htmlwidgets_1.3   grid_3.5.1
[28] nnet_7.3-12         tidyselect_0.2.5  data.table_1.11.8
[31] glue_1.3.0          R6_2.3.0          foreign_0.8-71
[34] latticeExtra_0.6-28 purrr_0.2.5       magrittr_1.5
[37] htmltools_0.3.6     backports_1.1.2   scales_1.0.0
[40] splines_3.5.1       assertthat_0.2.0  colorspace_1.3-2
[43] stringi_1.2.4       acepack_1.4.1     lazyeval_0.2.1
[46] munsell_0.5.0       crayon_1.3.4
```