

# iCARE (Individualized Coherent Absolute Risk Estimation) Package

April 26, 2022

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_72_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
11.764  0.120  11.884
```

Compute a summary of the risks.

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

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.05745 0.08666 0.09494 0.09600 0.10422 0.15882
```

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_72_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
      0.350  0.008  0.358

```

```

> names(res_snps_dat)

```

```

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

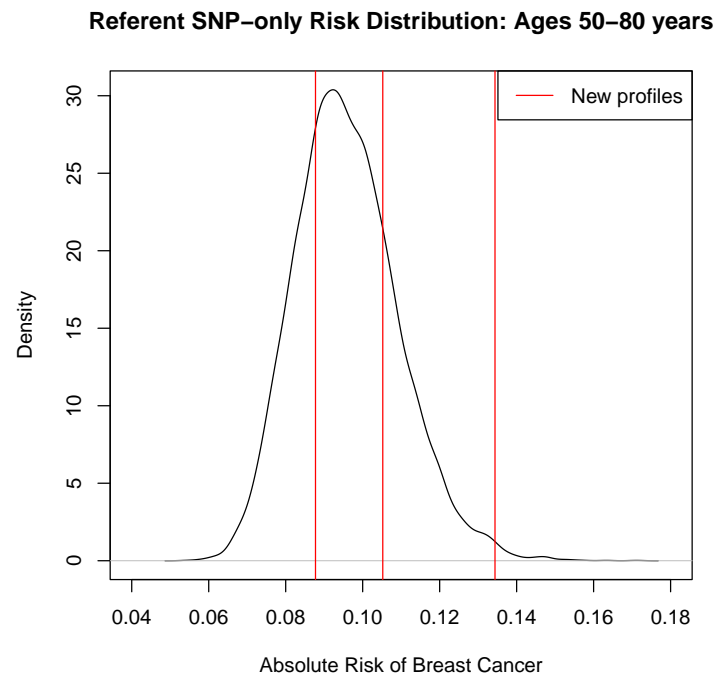
```

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$refs.risk),
+       xlim = c(0.04,0.18), xlab = "Absolute Risk of Breast Cancer",
+       main = "Referent SNP-only Risk Distribution: Ages 50-80 years")
> abline(v = res_snps_dat$risk, col = "red")
> legend("topright", legend = "New profiles", col = "red", lwd = 1)

```



## 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 classical risk factors and 72 SNPs, with three specific covariate profiles supplied for estimation (with some missing data). More details on risk factors are available in the manual.

```
> res_covs_snps = computeAbsoluteRisk(model.formula = bc_model_formula,
+                                     model.cov.info = bc_model_cov_info,
+                                     model.snp.info = bc_72_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
1.232  0.224  1.456
```

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	rs616488	rs11552449	rs11249433	rs12405132
1	50	80	0.10240752	NA	NA	NA	NA
2	50	80	0.08994616	2	0	NA	NA
3	50	80	0.16910925	2	0	1	1
	rs12048493	rs6678914	rs4245739	rs72755295	rs12710696	rs4849887	rs2016394
1	NA	0	0	0	0	0	0
2	NA	NA	NA	NA	1	1	0
3	1	1	1	0	2	0	0
	rs1550623	rs16857609	rs6762644	rs4973768	rs12493607	rs6796502	rs9790517
1	0	0	0	1	1	0	1
2	0	2	1	1	1	1	2
3	0	0	0	2	1	0	1
	rs6828523	rs10069690	rs13162653	rs2012709	rs10941679	rs10472076	rs1353747
1	0	1	2	0	0	2	0
2	0	0	1	0	0	1	1
3	0	0	1	0	0	0	1
	rs7707921	rs1432679	rs11242675	rs204247	rs9257408	rs4593472	rs720475
1	0	1	2	0	0	1	1
2	0	0	1	2	1	1	0
3	1	2	1	2	1	1	0
	rs9693444	rs13365225	rs6472903	rs2943559	rs13267382	rs11780156	rs1011970

1	1	1	1	0	0	0	0	
2	0	0	1	0	2	1	1	
3	1	1	0	0	1	0	0	
rs10759243 rs2380205 rs7072776 rs11814448 rs7904519 rs11199914 rs554219								
1	0	2	2	0	0	1	1	
2	1	0	0	0	0	0	0	
3	1	1	1	0	2	0	1	
rs75915166 rs11820646 rs12422552 rs17356907 rs1292011 rs11571833 rs2236007								
1	0	1	1	0	1	0	1	
2	0	0	0	0	0	0	0	
3	0	1	1	0	2	0	0	
rs2588809 rs999737 rs941764 rs11627032 rs17817449 rs11075995 rs13329835								
1	0	0	1	0	1	1	1	
2	1	0	0	1	1	1	0	
3	0	0	1	0	0	1	1	
rs146699004 rs745570 rs527616 rs1436904 rs6507583 rs4808801 rs3760982								
1	0	0	0	0	0	1	0	
2	1	2	0	0	0	1	1	
3	1	2	1	1	0	1	1	
rs2284378 rs2823093 rs17879961 rs132390 rs6001930 famhist menarche_dec parity								
1	1	1	0	0	0	0	8	0
2	1	0	0	0	0	0	10	0
3	0	0	0	0	0	0	1	0
birth_dec agemeno_dec height_dec bmi_dec rd_menohrt rd2_everhrt_e								
1	2	2	6	10	1	0		
2	2	1	6	4	1	0		
3	1	7	1	10	1	0		
rd2_everhrt_c rd2_currhrt alcoholweek_dec ever_smoke								
1	0	0	1	1				
2	0	0	6	0				
3	0	0	1	1				

## Illustration of the validation component

We want to validate a model for predicting absolute risk of disease based on a combined model of classical risk factors and 72 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.

```

> 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_72_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_72_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
147.535  0.559 152.327

```

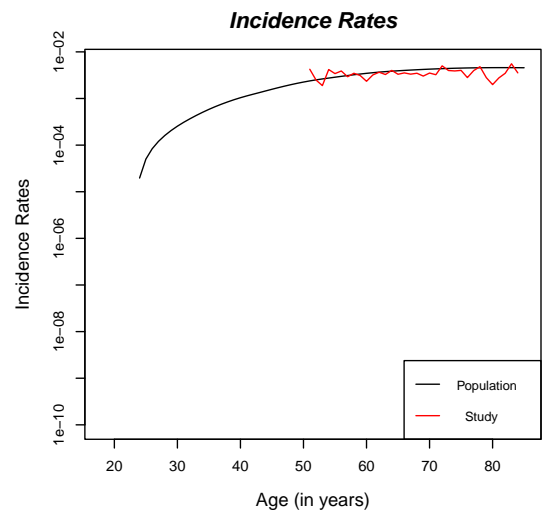
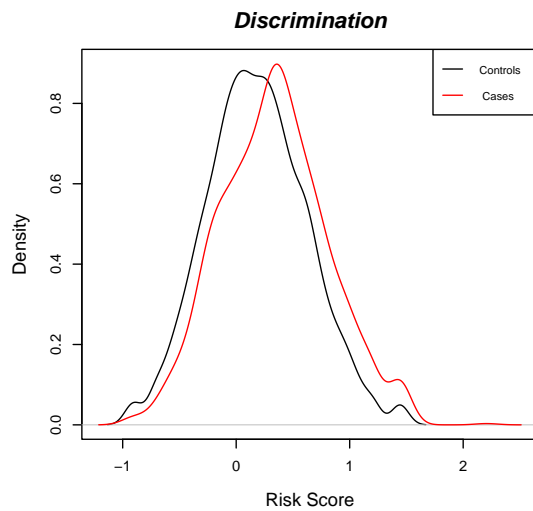
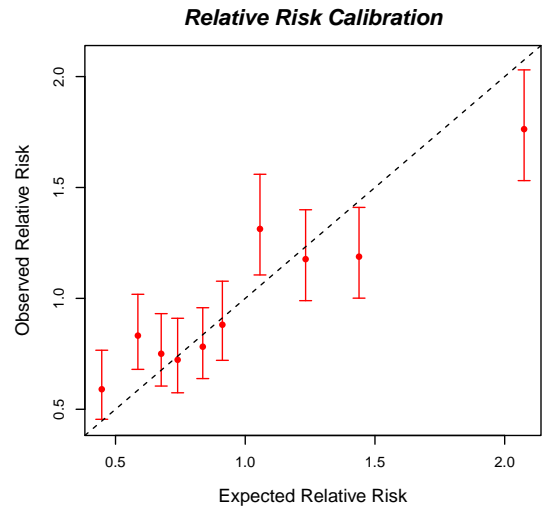
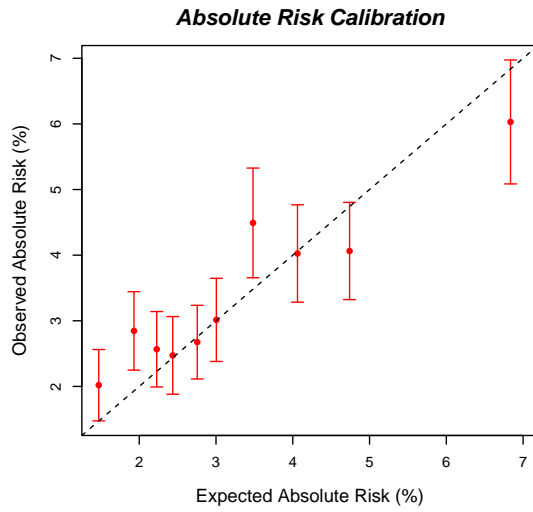
We can also produce a set of useful plots showing the results of the validation analysis.

```

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

NULL

```



Dataset: Example Dataset

Model Name: Example Model

Risk Prediction Interval: Observed Followup

Number of subjects (cases): 5285 ( 1251 )

Follow-up time (years) [mean,range]: [ 9.706 , ( 5 , 13 ) ]

Baseline age (years) [mean,range]: [ 62.556 , ( 50 , 72 ) ]

E/O [Estimate, 95% CI]: [ 0.967 , ( 0.908 , 1.03 ) ]

**Absolute Risk Calibration**

HL Test, df: 25.925 , 10

p-value: 3.842949e-03

**Relative Risk Calibration**

Test, df: 35.528 , 9

p-value: 4.807e-05

**Model Discrimination**

AUC est: 0.587

95% CI: ( 0.568 , 0.605 )

## Session Information

```
> sessionInfo()
```

```
R version 4.2.0 RC (2022-04-19 r82224)
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
Running under: Ubuntu 20.04.4 LTS
```

```
Matrix products: default
```

```
BLAS: /home/biocbuild/bbs-3.15-bioc/R/lib/libRblas.so
```

```
LAPACK: /home/biocbuild/bbs-3.15-bioc/R/lib/libRlapack.so
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=en_GB             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_1.24.0   Hmisc_4.7-0    ggplot2_3.3.5  Formula_1.2-4
[5] survival_3.3-1 lattice_0.20-45 gtools_3.9.2   plotrix_3.8-2
```

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

```
[1] tidyselect_1.1.2    xfun_0.30      purrr_0.3.4
[4] splines_4.2.0      colorspace_2.0-3 vctrs_0.4.1
[7] generics_0.1.2     htmltools_0.5.2 base64enc_0.1-3
[10] utf8_1.2.2         rlang_1.0.2    pillar_1.7.0
[13] foreign_0.8-82     glue_1.6.2     withr_2.5.0
[16] DBI_1.1.2          RColorBrewer_1.1-3 jpeg_0.1-9
[19] lifecycle_1.0.1    stringr_1.4.0  munsell_0.5.0
[22] gtable_0.3.0       htmlwidgets_1.5.4 latticeExtra_0.6-29
[25] knitr_1.38         fastmap_1.1.0  fansi_1.0.3
[28] htmlTable_2.4.0    scales_1.2.0   backports_1.4.1
[31] checkmate_2.1.0    gridExtra_2.3  png_0.1-7
[34] digest_0.6.29     stringi_1.7.6  dplyr_1.0.8
[37] grid_4.2.0         cli_3.3.0      tools_4.2.0
[40] magrittr_2.0.3     tibble_3.1.6   cluster_2.1.3
[43] crayon_1.5.1       pkgconfig_2.0.3 ellipsis_0.3.2
[46] Matrix_1.4-1       data.table_1.14.2 assertthat_0.2.1
[49] rstudioapi_0.13    R6_2.5.1       rpart_4.1.16
[52] nnet_7.3-17        compiler_4.2.0
```