

The Escore Package: Estimating HIV-1 exposure scores in R

Romel D. Mackelprang
International Clinical Research Center
Department of Global Health
University of Washington

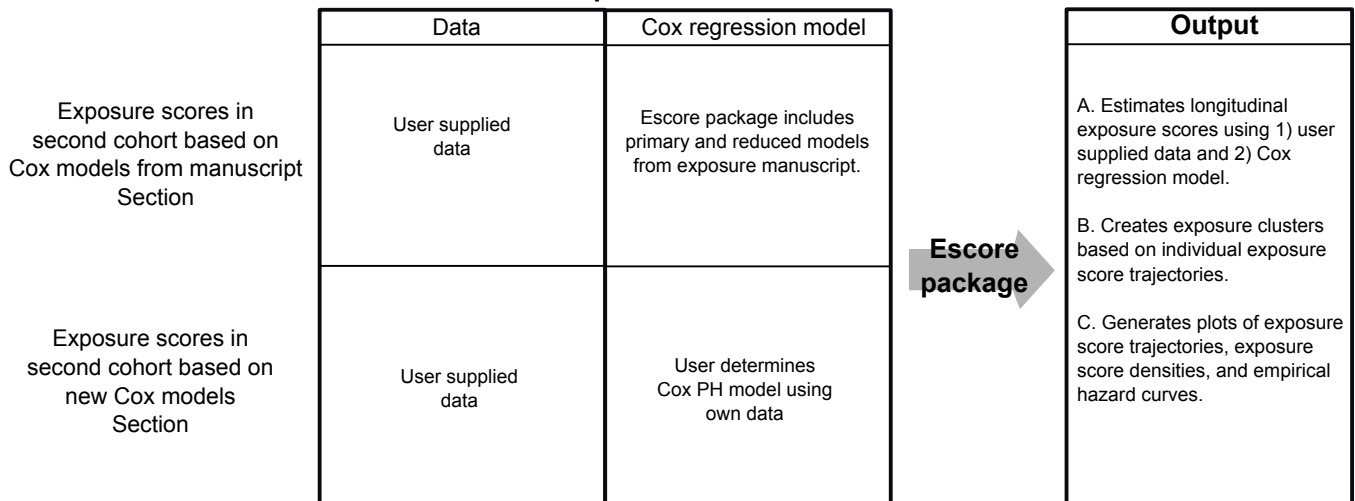
romelm@uw.edu

September 27, 2011

Summary

Studies of host biological factors associated with resistance to HIV-1 acquisition may be aided by appropriately estimating participant HIV-1 exposure levels across multiple study visits. We developed the Escore package in R to provide a platform for researchers to determine longitudinal exposure clusters within their own cohorts using the approach proposed in the primary manuscript. This package allows estimation of exposure scores based on Cox PH regression models developed in the Partners in Prevention HSV/HIV cohort and presented in the paper, and also allows estimation of exposure scores using new Cox PH models. Finally, there is a utility for simulating the effects of ignoring exposure levels when studying potential host correlates of resistance to infection. This vignette provides brief instructions for getting started with R and gives examples for using the Escore package in separate cohorts.

Figure 1: Overview of Escore package
Input



Contents

1	Introduction to R	3
2	The <code>Escore</code> package	3
2.1	Download source file	3
2.2	Install <code>Escore</code> package	3
2.3	Example dataframes	4
3	Estimating exposure scores in a new cohort using models presented in the manuscript	4
3.1	Load and examine Cox PH model objects	4
3.2	Use <code>escore()</code> function to estimate HIV-1 exposure scores for new data	6
3.3	Re-categorize participants by exposure cluster using the <code>cluster.cat</code> function	7
3.4	Additional exposure score plots	8
3.4.1	Exposure score density plots	8
3.4.2	Smoothed hazard plots by exposure category	9
3.4.3	ROC curves	10
3.5	Exporting exposure clusters	11
3.6	Evaluating different numbers of exposure clusters	12
4	Estimating exposure scores using new Cox PH model	13
4.1	Create new Cox PH model	13
4.2	Create new exposure scores	13
4.3	Apply new Cox PH model to secondary dataset and compare discriminatory power of exposure scores in both datasets	14
5	Simulations	15
5.1	Generate exposure score distributions and infection in hypothetical cohorts	15
5.2	Simulate distribution of a continuous host factor	16
5.3	Selection of HESN controls	17
5.3.1	Host factor values among randomly selected controls	18
5.3.2	Host factor values among cases and exposure matched controls	19

1 Introduction to R

The R Project for Statistical Computing is a free software environment for statistical computing and graphics that runs on multiple platforms, including UNIX, Windows, and MacOS (<http://www.r-project.org/>). R provides a wide variety of statistical and graphical techniques, as well as object-oriented programming features, that make it a strong platform for estimating HIV-1 exposure scores.


R can be downloaded from the Comprehensive R Archive Network (CRAN)(<http://cran.r-project.org/>) and installed on most operating systems. The CRAN website also provides several manuals that are extremely useful when first using R, especially “An Introduction to R”, of which new users may wish to start with Appendix A; A Sample Session. Packages in R “provide a mechanism for loading optional code, data, and documentation as needed”. The base distribution includes about 30 packages and more than 3000 additional packages can be downloaded directly from the CRAN website to provide extended functionality.

2 The Escore package

We have developed the Escore package for estimating HIV-1 exposure scores, creating exposure score clusters, and visualizing exposure score data. Specifically, this package utilizes existing R libraries to:

- Determine visit specific exposure scores across study visits based on a participant’s observed covariate values and the regression coefficients from a Cox PH model.
- Cluster participants into homogeneous groups based on longitudinal exposure trajectories.
- Compute statistics for analyzing model performance using receiver operator characteristic (ROC) curves and areas under the ROC curves (AUC).
- Provide graphical utilities for plotting exposure score densities and smoothed hazard curves for exposure clusters.

2.1 Download source file

At this time, the Escore package has not been uploaded to the CRAN website and needs to be installed locally from a source (tar.gz) file. The source file is embedded in this pdf document and needs to be saved to a directory on the user’s computer. This .tar.gz file can be accessed by clicking the following icon () or through the attachments pane in Acrobat (View > Show/hide > Navigation panes)¹ Alternatively, the tar.gz file can be downloaded from the the ICRC website (<http://depts.washington.edu/uwicrc/>).

2.2 Install Escore package

After the source file is saved to a local directory, open R and install the Escore package by running the command `'install.packages(` `path/escore_1.0.tar.gz', repos=NULL, type="source")'`, where “path” is the local directory in which the file is saved.

```
> ###Install escore package
> install.packages("H:/escore_1.0.tar.gz", repos=NULL, type="source")
> ###Load package
> library(escore)
>
```

After the Escore package has been loaded, HTML help files can be viewed by running:

```
> help.start()
>
```

¹Security features in Acrobat prohibit saving file archives (e.g. zip and tar.gz files) that are attached to pdf documents. Therefore, it was necessary to include a .txt extension at the end of the file name to 'fool' Acrobat'. After saving the file to a local directory, the .txt extension should be deleted so the filename ends with .tar.gz

2.3 Example dataframes

For this vignette, we have generated two example dataframes for using the `Escore` package. These dataframes have similar covariate distributions and incidence of infection as the Partners in Prevention HSV/HIV data. However, while the distribution of censoring times for the Cox PH model for the example data was similar to the Partners in Prevention HSV/HIV data, the distribution of infection times was different. Therefore, the examples shown here do not replicate the decreasing hazard curves observed in the actual cohort (Section 3.4.2).

The example dataframes have the same variable names and formats as the data used to create the Cox PH models and exposure scores in the manuscript, and include longitudinal data in which participants have 1-8 observations (1 per quarterly study visit) with time-varying covariates. The variable 'start' represents the time of the previous study visit and 'stop' represents the time of the current visit or the estimated date of infection if an event occurred.

The dataframes can be loaded and explored as follows:

```
> ###Load data included with Escore package
> data(primary)
> data(secondary)
> ###View help file for primary data that includes variable definitions
> ?primary
> ###Show data for ptids 4 and 5, except for the pregnant and gender.circum
> ###variables which take up too much space
> subset(primary, ptid%in%c(4,5), select=-c(pregnant, gender.circum))
```

	ptid	event	start	stop	sex	logvl	index.gud	hsv	age10	gender	visitmonth
25	4	0	1	86	0	4.586081	0	1	2.868957	0	3
26	4	1	87	195	1	4.221537	0	1	2.868957	0	6
33	5	0	1	94	1	3.931206	0	1	3.159264	0	3
34	5	0	95	184	0	3.969620	0	1	3.159264	0	6
35	5	0	185	266	0	4.728799	0	1	3.159264	0	9
36	5	0	267	375	1	3.431338	0	1	3.159264	0	12
37	5	0	376	458	0	4.066267	0	1	3.159264	0	15
38	5	0	459	552	0	4.953510	0	1	3.159264	0	18
39	5	0	553	652	0	4.099625	0	1	3.159264	0	21

```
>
```

Thus, ptid # 4 was uninfected at 86 days but became infected on the 195th day. Ptid #5, on the other hand, was never infected and was censored upon exiting the study at 652 days.

3 Estimating exposure scores in a new cohort using models presented in the manuscript

The `Escore` package can be used to calculate exposure scores for a new dataset using regression coefficients from Cox PH models presented in the manuscript. Specifically, `Escore` includes two models from the manuscript. Model 1 is the Cox PH model that best fit the data when using backwards variable selection based on AIC values. Model 2 is a reduced version of Model 1 that does not include STI data. The variables used in each model are shown in Table 1.

3.1 Load and examine Cox PH model objects

Model 1 and model 2 are loaded as follows:

```
> ###Load data object storing models
> data(paper.models)
> ###Examine coefficients for model 1
> paper.model1 #coefficients for model 2 viewed with command 'paper.model2'
```

```
Cox Proportional Hazards Model
```

```

cph(formula = Surv(time = start , time2 = stop , event = event) ~
sex + logvl + index.gud + hsv + pregnant + gender.circum +
age10 + gender , data = clong)

      Obs      Events Model L.R.      d.f.      P      Score      Score P
19144      84      128.54      8      0      133.31      0
R2
0.099

      coef se(coef)      z      p
sex      1.440      0.232      6.21      5.15e-10
logvl      0.987      0.137      7.20      5.92e-13
index.gud      0.497      0.280      1.77      7.61e-02
hsv      0.773      0.298      2.59      9.50e-03
pregnant      0.710      0.376      1.89      5.89e-02
gender.circum      -0.485      0.302      -1.61      1.08e-01
age10      -0.386      0.140      -2.75      5.99e-03
gender      -0.393      0.287      -1.37      1.71e-01

```

```

>
  Escape documentation describes the variables and parameterization of these models and can be accessed by running
> ###View documentation for Cox PH models from manuscript
> ?paper.model1
>

```

Table 1: Variables included in longitudinal Cox PH regression models from the Partners in Prevention HSV/HIV cohort.

Variable name	Description	Values	Model 1	Model 2
sex	Dichotomous variable indicating unprotected sex at each visit	0=No, 1=Yes	X	X
logvl	Plasma HIV-1 RNA levels of the HIV-infected partner (log ₁₀)	Continuous	X	X
index.gud	Dichotomous variable indicating if HIV-infected partner had GUD at each visit	0=No, 1=Yes	X	
hsv	Dichotomous variable indicating if HESN participant was HSV-2 seropositive at baseline	0=No, 1=Yes	X	
pregnant	Dichotomous variable indicating if female HESN participants were pregnant at each study visit	0=No OR male, 1=Yes	X	
gender	Dichotomous variable indicating gender of HESN participants	0=male, 1=female	X	X
gender.circum	Dichotomous variable indicating male circumcision status of HESN participants	0=Uncircumcised OR female, 1=Circumcised male	X	X
age10	Age of HESN participants divided by 10	Continuous	X	X
start	Survival analysis variable for start time of interval (days)	Integer	X	X
stop	Survival analysis variable for stop time of interval (days)	Integer	X	X
event	Survival analysis variable indicating if event (HIV-infection) occurred during interval.	0=No, 1=Yes	X	X

3.2 Use `escore()` function to estimate HIV-1 exposure scores for new data

The `escore()` function is the main 'workhorse' of the `Escore` package and is used to determine HIV-1 exposure scores, trajectories, and clusters. Specifically, this function

1. Calculates exposure scores based on longitudinal covariates and coefficients from a Cox PH model created using the `cph()` function from the `Design` library. Exposure scores are determined by finding linear predictors that are normalized to the sample average (relative linear predictors). An exposure score of 0 represents the average exposure score in the sample. Relative exposure scores were used because the Cox PH model is a relative risk model and is not an absolute risk model.
2. Generates longitudinal exposure clusters using k-means cluster analysis for longitudinal data (see the `kml()` package).
3. Calculates the mean exposure score across all study visits for each participant.
4. Determines discriminatory power of the mean longitudinal exposure scores in separating participants with and without infection by estimating the area under receiver operator characteristic curves (AUC).

We can calculate exposure scores in the primary dataset using model 1 from the manuscript by running

```
> ###Create escore object
> escore1 <- escore(model=paper.model1, data=primary,
+                 vars=list(id="ptid", visit="visitmonth", event="event",
+                 time="start", time2="stop"), newdata=TRUE, seed=100,
+                 nbClusters=6)
>
```

```
> ###Summary of escore object
> escore1
```

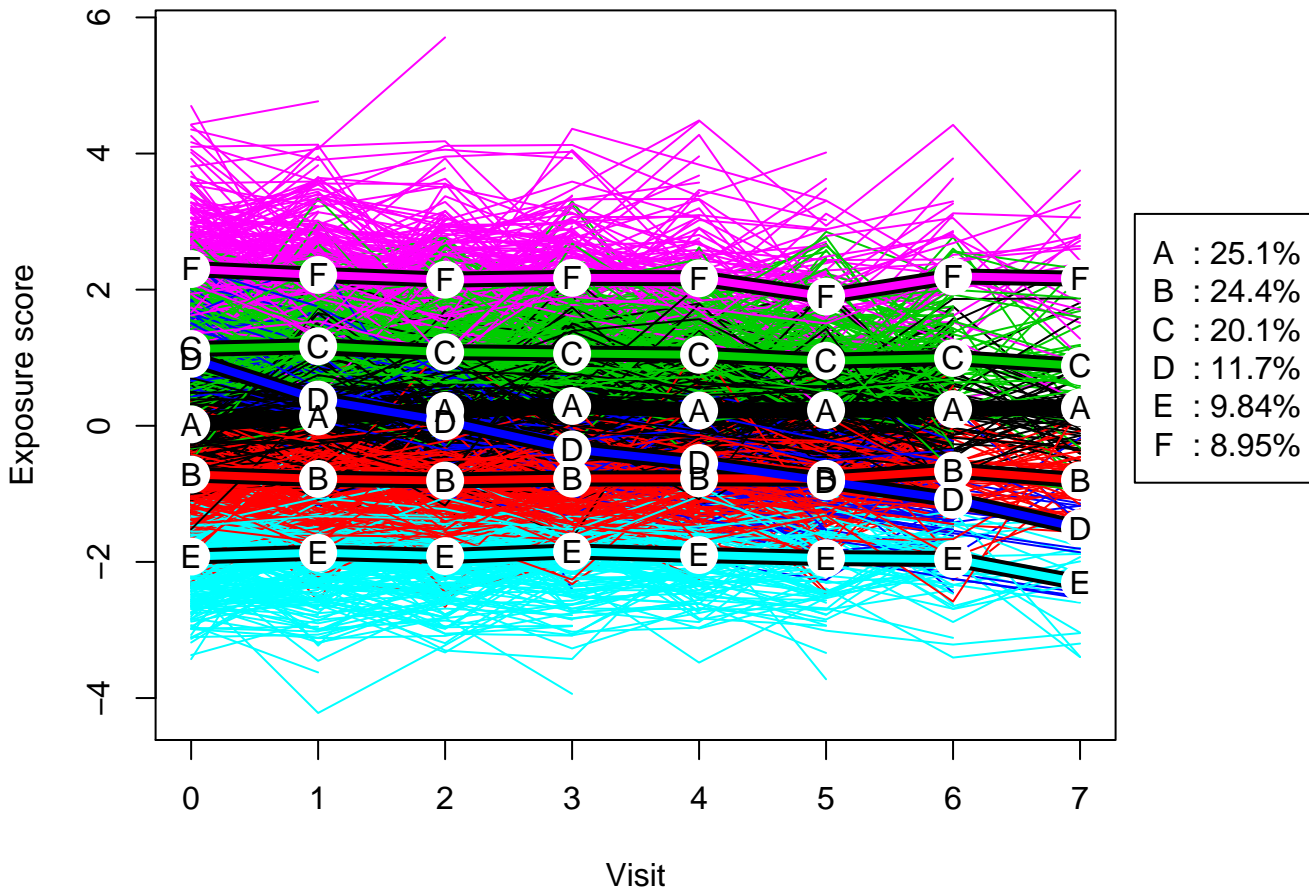
```
***Longitudinal exposure scores***
*Number of participants=3027
*Total number of visits=13508
*Average number of visits per participant=4.46250412950116
*Total number of events=96
*Average exposure score at all study visits=0.1
*AUC for ROC curve=0.763327362674867
```

```
>
```

As output, the `escore()` function saves an S4 object (named `escore1` here) that includes slots for the `cph` model, longitudinal exposure scores, clusterization objects, and cluster dataframe.

Longitudinal exposure score clusters can be further evaluated by plotting individual exposure score trajectories and cluster means at each visit.

```
> plot.cluster(x=escore1, xlab="Visit", ylab="Exposure score")
>
```



This plot demonstrates that using 6 clusters results in one cluster that systematically experienced decreasing exposure scores over follow-up and a “highest” risk cluster that consisted of approximately 9% of the cohort.

3.3 Re-categorize participants by exposure cluster using the `cluster.cat` function

Identifying a small subset of participants with the highest longitudinal exposure scores requires more granularity (a greater number of clusters) than is needed for participants with lower exposure scores. Furthermore, alphanumeric labels for clusters are not intuitive for communicating results. Therefore, it is desirable to re-categorize some clusters into more easily interpretable categories. This can be done using the `cluster.cat()` function by running

```
> ###Create new cluster categories
> escore1 <- cluster.cat(escore1, labels=list(A="Lower", B="Lower", C="Lower",
+ D="Decreasing", E="Lower", F="Highest"))
```

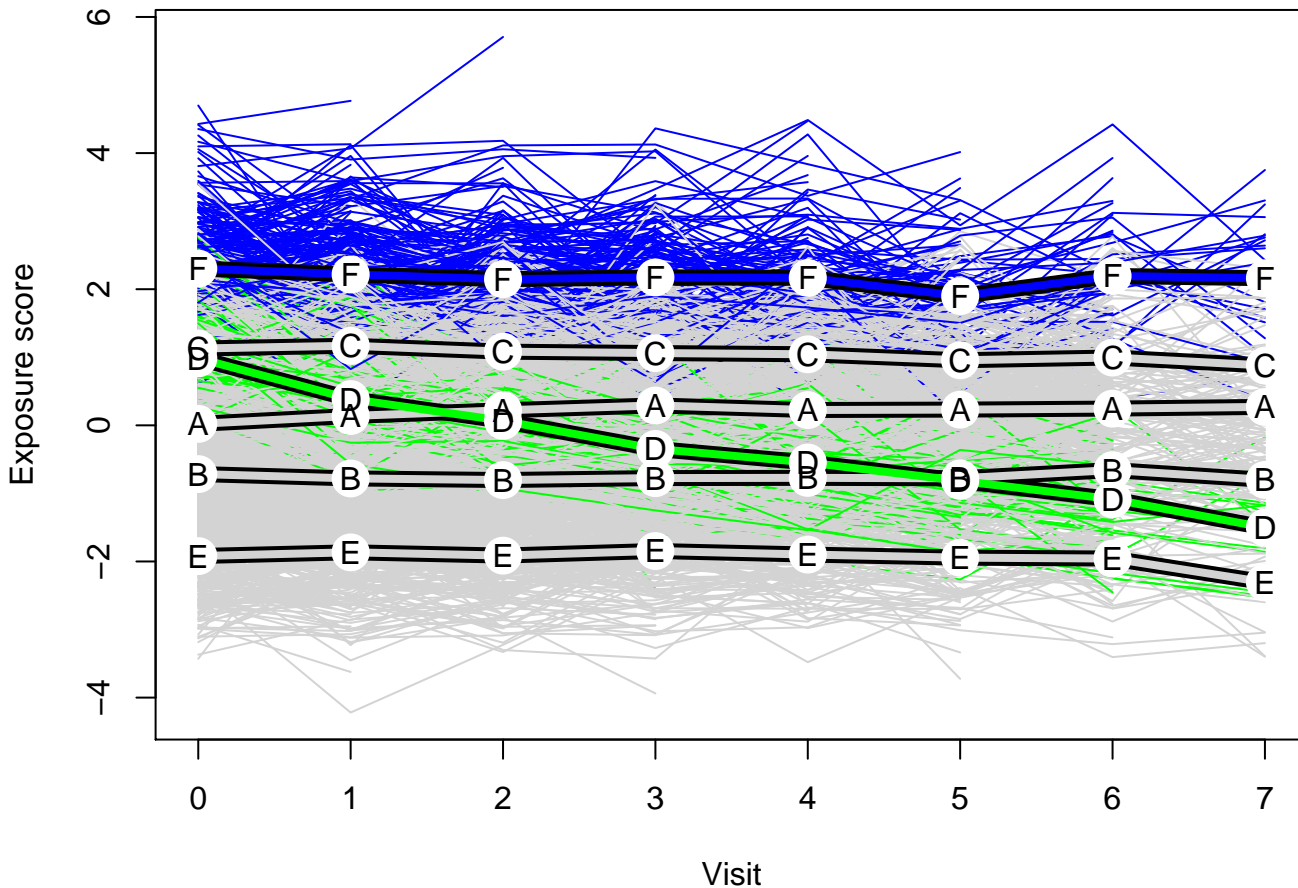
	Decreasing	Highest	Lower
A	0	0	704
B	0	0	683
C	0	0	563
D	327	0	0
E	0	0	276
F	0	251	0

```
>
>
```

where labels is a list of new categories. The `cluster.cat()` function creates a new variable in `escore1` that re-categorizes participants according to the labels option and original clusters are maintained.

The previous plot of clusters/trajectories can now be updated with new color schemes based on the re-categorization by using the `col.group` argument in the `plot.cluster` function.

```
> ###Plot clusters with colors based on new categories
> plot.cluster(x=score1, xlab="Visit", ylab="Exposure score", col.group=TRUE,
+             new.col=c("green", "blue", "lightgrey"), legend=FALSE)
>
```



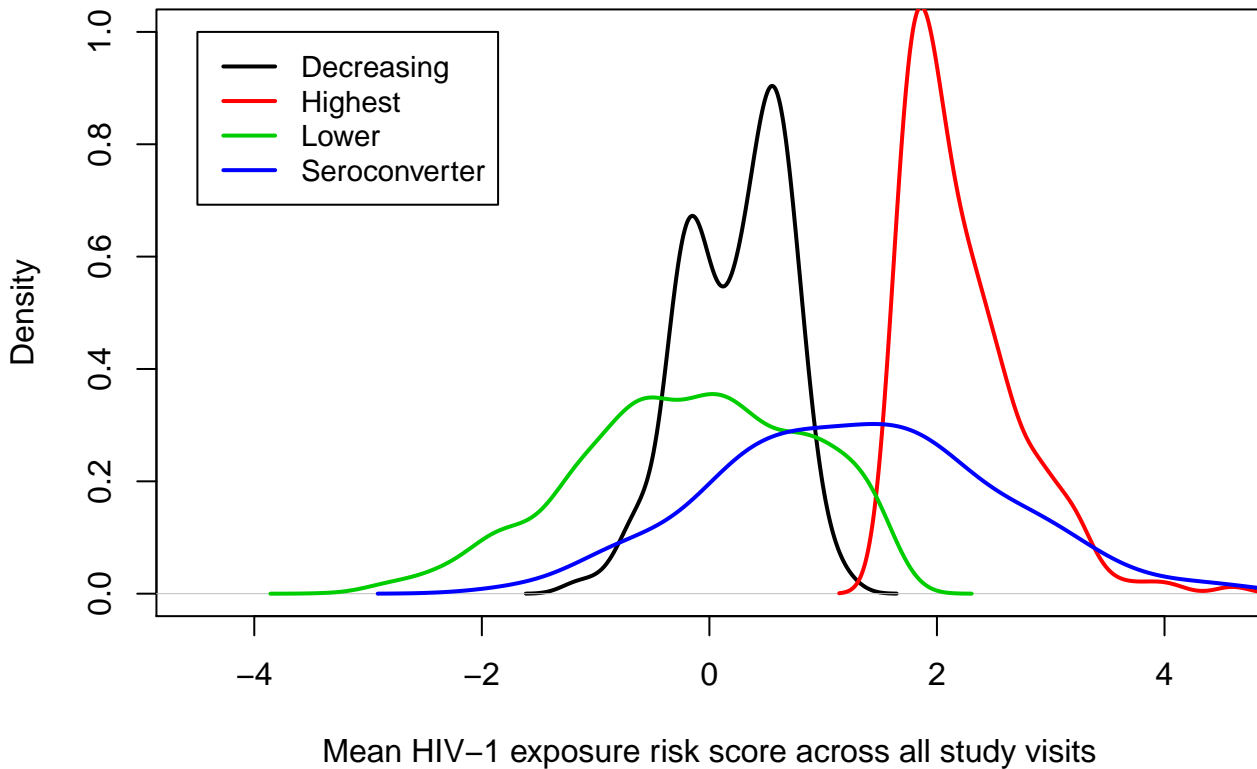
3.4 Additional exposure score plots

In addition to plotting exposure score trajectories and cluster means, the `escore` function also provides functions for plotting exposure score densities, smoothed hazard curves, and ROC curves.

3.4.1 Exposure score density plots

First, we plot the exposure score densities by exposure categories, with an additional category for participants who became infected with HIV-1. Note that the density plots are for a participants mean exposure score across all study visits. Density plots are created using the `plot.escore.density()` function by running

```
> ###Plot exposure score densities
> plot.escore.density(score1, use.cat=TRUE, by.event=TRUE, event.label="Seroconverter")
>
```

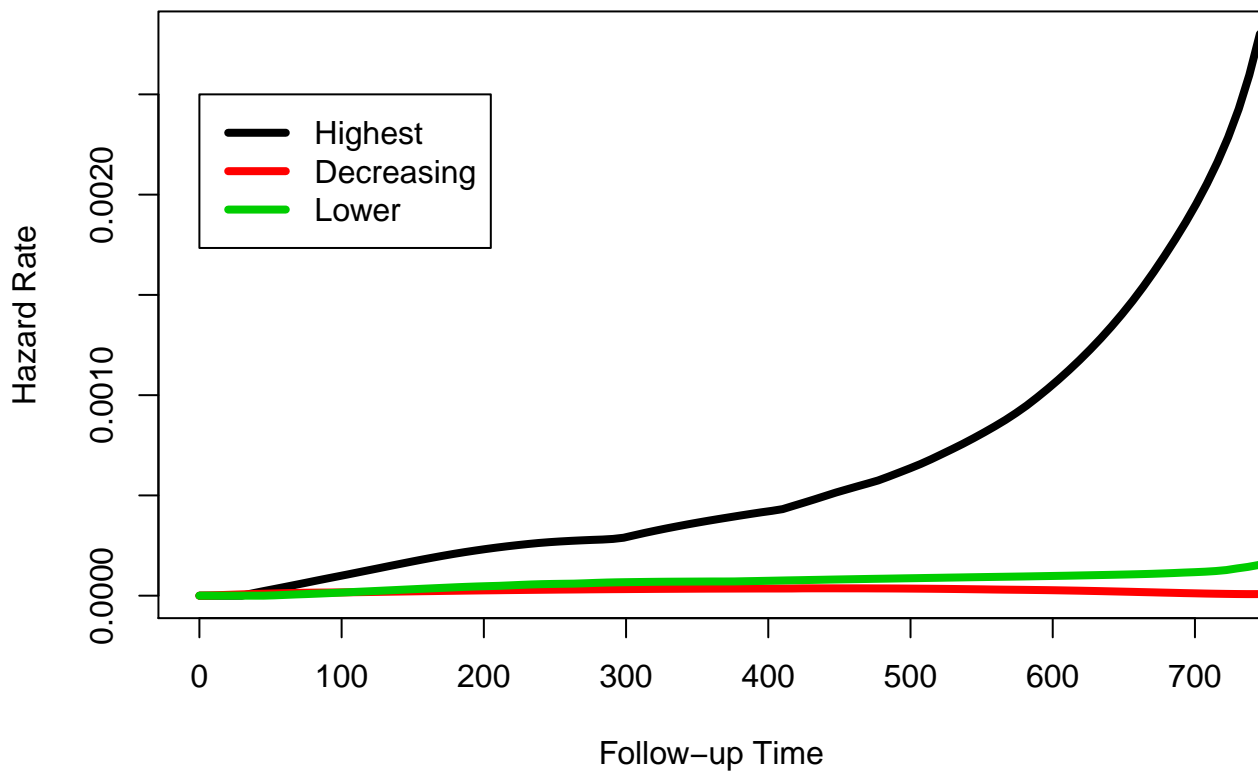



This plot mirrors the plot from the manuscript and demonstrates that the highest longitudinal cluster did in fact have the highest mean exposure scores when compared to HESN participants with low or decreasing exposure scores, as well as seroconverters.

3.4.2 Smoothed hazard plots by exposure category

Second, we can plot smoothed hazard functions for HIV-1 infection by exposure score clusters. The hazard functions represent the instantaneous probability of infection at any time during follow-up. The Escore `plot.hazard()` function uses the the `muhaz` package and is run as follows

```
> ###Plot smoothed hazard curves
> plot.hazard(escore1, use.cat=TRUE, ref="Highest",
+             legend.y=0.0025, legend.x=0)
>
```

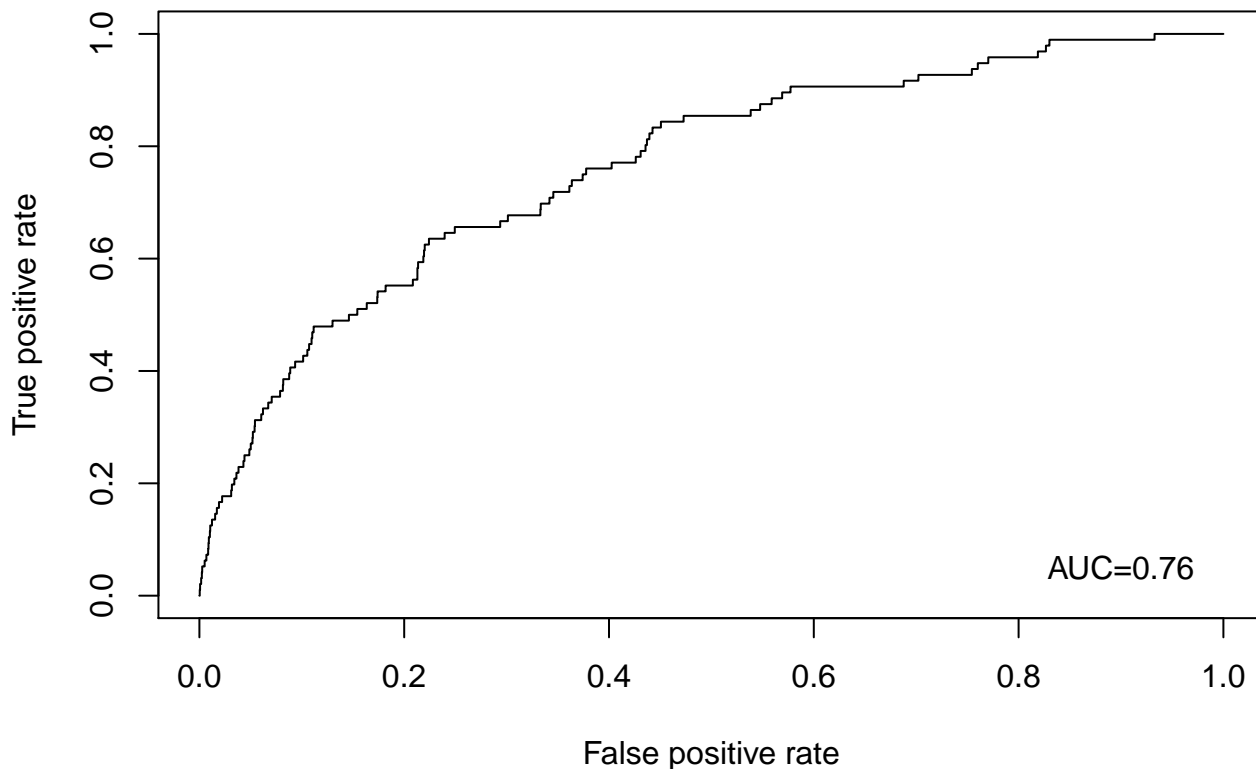


Note: In the actual Partners in Prevention HSV/HIV data, the hazard of infection among the highest risk cluster decreased over follow-up. When creating the example dataframes we did not simulate the non-normal distribution of survival times among this group. Therefore, we do not observe the same empirical hazard function in this example.

3.4.3 ROC curves

Finally, we can plot an ROC curve using the `plot.roc()` function.

```
> ###Plot ROC curve
> plot.roc(score1, col="black", print.auc=TRUE)
>
```



The `print.auc` argument allows the user to toggle on/off printing AUC values on the plot. The hazard of infection is greatest among participants in the highest exposure category. However, the hazard rate does not decrease over the course of follow-up as observed in the actual data from our cohort. This is because we did not make assumptions about hazard curves when generating the hypothetical data.

3.5 Exporting exposure clusters

For generating summaries of exposure scores and identifying the highest risk HESN participants for further study, it is useful to export exposure clusters and mean exposure scores to an external dataframe containing other summary variables. The `merge.escore()` function extracts 1. mean exposure scores, 2. original exposure clusters, and 3. re-categorized exposure clusters for each participant from the `escore` data object and merges those data with a second dataframe.

```
> ###First, create a summary data frame indicating if each participant
> ###ever reported unprotected sex or was infected
> ever.sex <- tapply(primary$sex, primary$ptid, max)
> infected <- tapply(primary$event, primary$ptid, max)
> summary.data <- data.frame(ptid=names(ever.sex), ever.sex=ever.sex,
+                             infected=infected)
> ###Merge exposure clusters with summary.data
> summary.data <- merge.escore(x=escore1, data=summary.data,
+                               name="escore1")
> ###Look at new variable names of summary.data
> names(summary.data)
```

```
[1] "ptid" "escore1.mean.escore" "escore1.cluster"
[4] "escore1.cluster.cat" "ever.sex" "infected"
```

>

Now that `summary.data` includes exposure cluster assignments, we are able to evaluate summary statistics by cluster.

```

> ###First, change the values of ever.sex and infected from numeric dichotomous variables
> ###to factors with interpretable labels.
> summary.data$ever.sex ← factor(summary.data$ever.sex, labels=c("No", "Yes"))
> summary.data$infected ← factor(summary.data$infected, labels=c("No", "Yes"))
> ###Create table showing characteristics of original exposure clusters
> summary(escor1.cluster~infected+ever.sex, data=summary.data,
+ method="reverse")

```

Descriptive Statistics by escor1.cluster							
	N	A (N=704)	B (N=683)	C (N=563)	D (N=327)	E (N=276)	F (N=251)
infected : Yes	3027	3% (19)	1% (7)	4% (25)	1% (4)	0% (1)	11% (28)
ever.sex : Yes	3027	32% (225)	22% (153)	44% (249)	42% (136)	10% (27)	69% (172)

```

> ###Create table showing characteristics of re-categorized exposure clusters
> summary(escor1.cluster.cat~infected+ever.sex, data=summary.data,
+ method="reverse")

```

Descriptive Statistics by escor1.cluster.cat				
	N	Decreasing (N=327)	Highest (N=251)	Lower (N=2226)
infected : Yes	3027	1% (4)	11% (28)	2% (52)
ever.sex : Yes	3027	42% (136)	69% (172)	29% (654)

>
>

3.6 Evaluating different numbers of exposure clusters

The number of clusters participants should be grouped into is a somewhat arbitrary decision that should be based on the researchers needs and exploration of data for trends. For example, we have specified that participants should be grouped into 6 original clusters because doing identifies a small group of participants who have the highest exposure scores but is large enough for further study, and because it identifies a a group with decreasing exposure scores over time. At this time, the `escor()` function can only group participants into a single number of clusters. However, the `kml` package can be used directly to explore differing numbers of clusters. For instance,

```

> ###Extract ClusterizeLongData object from escor object
> cld ← escor1@cld
> ###Create 4-5 clusters
> kml(cld, nbClusters=4:5)
> ###Plot with 4 clusters (not shown in vignette)
> plot(cld, y=4)
>
>

```

4 Estimating exposure scores using new Cox PH model

The `Score` package allows exposure scores to be calculated based on any Cox PH model, not just the models included in the manuscript. Therefore, other researchers are able to develop their own regression models for HIV-1 acquisition risk and use these models to calculate exposure scores and create exposure clusters.

4.1 Create new Cox PH model

For instance, using the hypothetical dataset we can create a new model:

```
> ###Create model
> new.model <- cph(Surv(time=start , time2=stop , event=event)~logvl+sex , data=primary)
> ###View coefficients
> new.model
```

```
Cox Proportional Hazards Model

cph(formula = Surv(time = start , time2 = stop , event = event) ~
     logvl + sex , data = primary)

      Obs      Events Model L.R.      d.f.      P      Score      Score P
13508          96    114.71         2         0    133.88         0
R2
0.085

      coef se(coef)      z      p
logvl 0.815   0.103  7.94 2.11e-15
sex    1.531   0.205  7.48 7.27e-14
```

```
>
```

4.2 Create new exposure scores

We can now use `new.model` to estimate exposure scores for participants in the primary dataframe (same data used to make model) by setting the `newdata` argument of the `escore()` function to `FALSE`.

```
> ###Create escore object using new.model
> escore.new <- escore(model=new.model , data=primary ,
+                      vars=list(id="ptid" , visit="visitmonth" , event="event" ,
+                                time="start" , time2="stop") , newdata=FALSE , seed=100 ,
+                      nbClusters=6)
>
```

```
> ###Summary
> escore.new
```

```
***Longitudinal exposure scores***
*Number of participants=3027
*Total number of visits=13508
*Average number of visits per participant=4.46250412950116
*Total number of events=96
*Average exposure score at all study visits=0
*AUC for ROC curve=0.752057744796995
```

```
>
```

4.3 Apply new Cox PH model to secondary dataset and compare discriminatory power of exposure scores in both datasets

If a new Cox PH model is generated for estimating exposure scores, it is desirable to validate this model on a secondary cohort. One way to do this using the `Escore` package is to compare the ability of individual mean exposure scores to discriminate between participants with and without infection based on ROC curves.

To do this, we must first estimate exposure scores for a secondary dataset using a Cox PH model that was created with data from the primary dataset.

```
> ###Load secondary data
> data(secondary)
> ###Create exposure scores for secondary data
> escore.new2 <- escore(model=new.model, data=secondary,
+                       vars=list(id="ptid", visit="visitmonth", event="event",
+                                 time="start", time2="stop"), newdata=TRUE, seed=100,
+                       nbClusters=6)
>
>
```

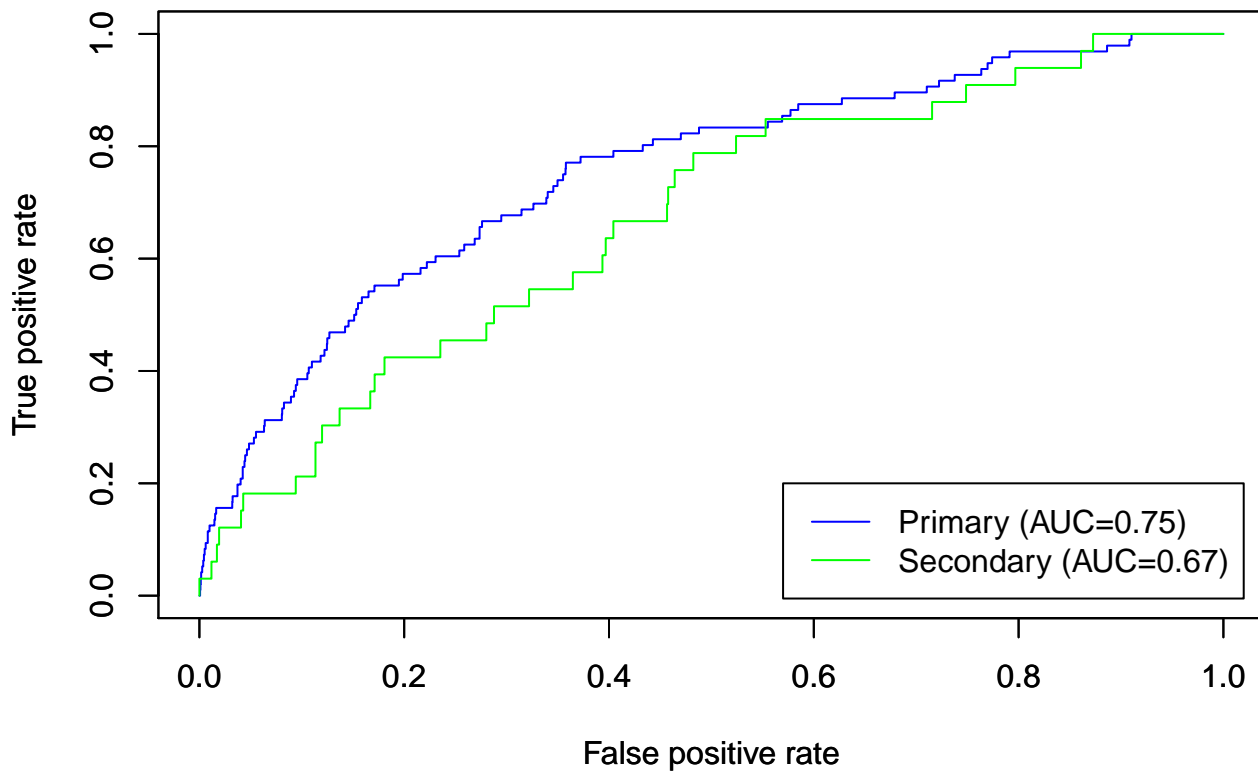
```
> ###Summary
> escore.new2
```

```
*** Longitudinal exposure scores ***
* Number of participants=968
* Total number of visits=4222
* Average number of visits per participant=4.36157024793388
* Total number of events=33
* Average exposure score at all study visits=-0.02
* AUC for ROC curve=0.671366067087996
```

```
>
```

Next, ROC curves can be plotted for both sets of exposures scores:

```
> ###Curve for primary data
> plot.roc(escore.new, col="blue", new.plot=TRUE, print.auc=FALSE)
> ###Curve for secondary data
> plot.roc(escore.new2, col="green", new.plot=FALSE, print.auc=FALSE)
> ###Add legend with AUC values
> legend(x=.57, y=.2,
+        legend=c(paste("Primary (AUC=", round(escore.new@auc, digits=2), ")"), sep=""),
+                 paste("Secondary (AUC=", round(escore.new2@auc, digits=2), ")"), sep=""),
+        col=c("blue", "green"), lty=1)
>
>
>
```



5 Simulations

The `Score` package also contains functions to simulate potential biases arising from not accounting for HIV-1 exposure in a case-control analysis to determine if a continuous variable (host factor) is associated with resistance to infection.

5.1 Generate exposure score distributions and infection in hypothetical cohorts

The `sim.infection()` function allows creation of `n.sim` hypothetical datasets with `n` participants, where `n.sim` indicates the number of simulations. First, exposure scores are created based on a normal distribution by specifying the mean and standard deviation (`sd`). For the manuscript, the mean and `sd` for simulations was derived from the exposure scores observed in the Partners in Prevention HSV/HIV study. Next, infections were simulated by determining probabilities of infection that were based on exposure scores and coefficients from a logistic regression relating exposure scores and infection in the real data.

```
> ###Simulate exposure score distributions and infections.
> sim <- sim.infection(mu=0, sigma=1.2, intercept=-4.96, beta=1.41,
+                     n=3400, n.sim=100)
>
```

creates 100 datasets with 3400 people. Because these simulations were based on statistics from real data, the number of infections should be similar to what was actually observed. We can check this by calling

```
> ###Find average number of infections across all cohorts.
> n.infections <- apply(sim@infections, 2, function(x) table(x)[2])
> n.infections <- summary(n.infections)
> n.infections
```

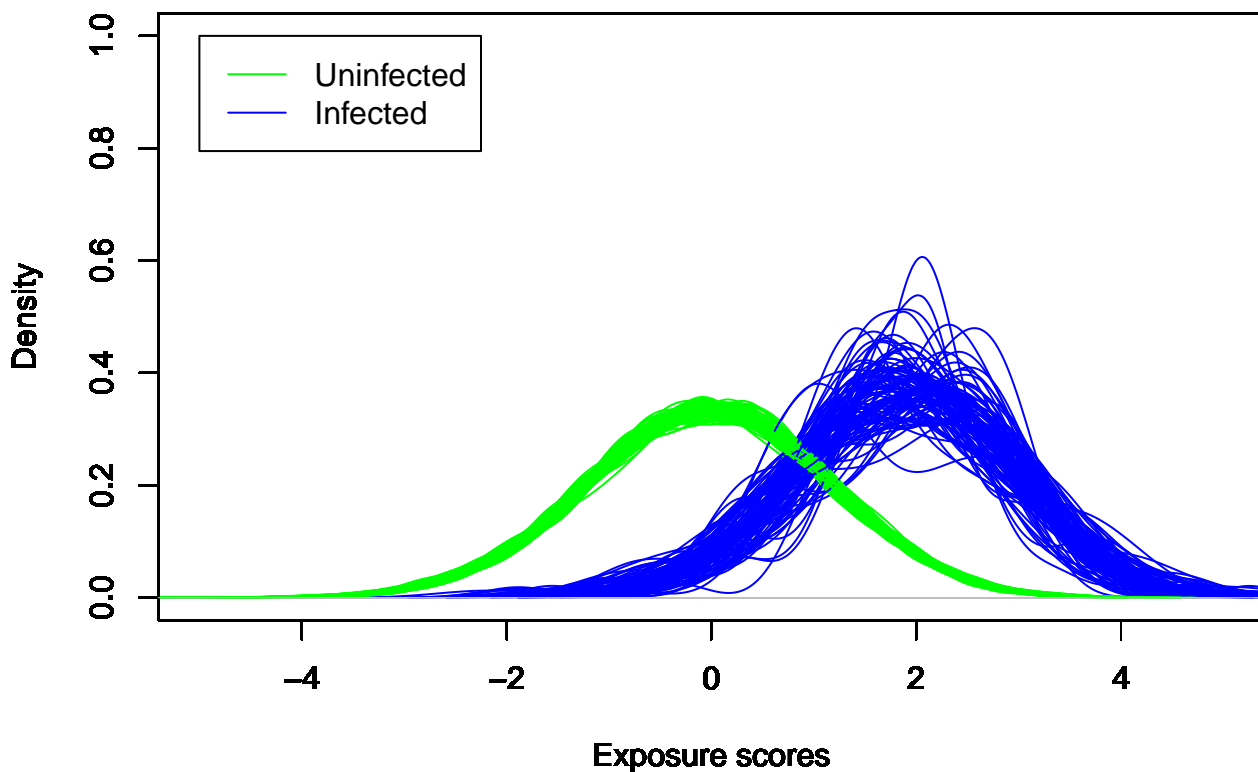
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
53.00	66.00	71.00	71.08	76.00	87.00

>

Next, we can check that higher exposure scores are associated with infection status by plotting exposure score densities:

```
> ###Create plot
> plot.sim.infection(x=sim, main="Simulated exposure score densities by infection status")
>
```

Exposure scores for all obs. by infection.



This plot demonstrates that, on average, exposure scores were higher HIV+ than HIV- persons in the simulated datasets.

5.2 Simulate distribution of a continuous host factor

For this exercise, we are interested in evaluating the ability of a case-control study to identify true differences in a continuous host factor (marker) between cases and controls. Therefore, we simulated values of a marker using an OLS regression model that relates exposure scores and infection to the marker:

$$marker_i = \alpha + \beta_{score}x_{scorei} + \beta_{infection}x_{infectioni} + \epsilon_i \quad (1)$$

The *sim.marker()* function takes α , β_{score} , and $\beta_{infection}$ as arguments. For each individual i , *sim.marker()* estimates ϵ_i by assuming that random errors have a normal distribution with $\mu = 0$ and standard deviation (sigma) specified by the user.

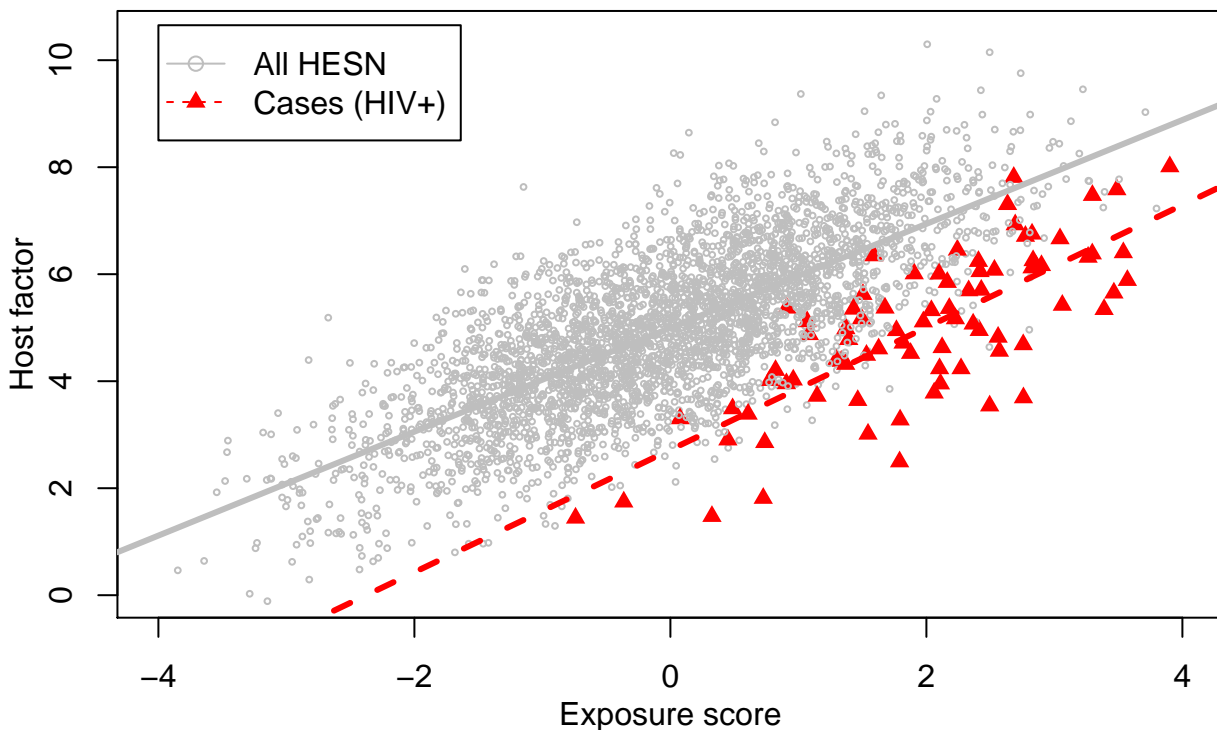
Scenario 2 in the manuscript assumes that 1) a 1-unit increase in exposure score is associated with a 1-unit increase in a continuous hypothetical host factor ($\beta_{\text{exposure}} = 1$), and 2) the average level of the host factor is 2-units lower among those acquiring HIV than among HESNs of the same exposure level ($\beta_{\text{infection}} = 2$). We also assumed that the average host factor level among HESNs with exposure score=0 was 5 ($\alpha = 5$) and that the standard deviation of the random errors was 1 ($\sigma = 1$). Host factor values can be simulated using these assumptions by running

```
> ###Simulate values of a continuous host factor
> sim <- sim.marker(x=sim, intercept=5, beta.exposure=1, beta.infection=-2, true.err.val=1)
>
```

and the associations between exposure scores, infection, and the values of the host factor are plotted by

```
> ###Plot simulated host factor values with exposure scores and infection
> plot.sim.marker(sim, ylab="Host factor", xlim=c(-4,4), ylim=c(0,10.5),
+               points.cex1=0.4, points.cex2=1,
+               main="Host factor associated with HIV-1 exposure\n and HIV-1 acquisition",
+               line=2, col=c("grey", "red"),
+               legend=list(x=-4, y=8.5, legend=c("All HESN", "Cases (HIV+)")))
>
>
```

Host factor associated with HIV-1 exposure and HIV-1 acquisition



5.3 Selection of HESN controls

In the manuscript we evaluated effects of using different strategies for selecting HESN controls on observed associations between the hypothetical host factor and infection. First, we considered the effect of randomly selecting a single control from all HESN for every case and, subsequently, evaluated effects of selecting controls using 1:1 matching based on

exposure score. The `sim.cc()` function selects controls for each simulated dataset using both of these strategies ². Additionally, this function selects a third set of controls by selecting HESNs with the highest exposure scores.

After exposure scores, infection, and values of the hypothetical host factor have been simulated using the `sim.infection()` and `sim.marker()` functions, we can select controls by running

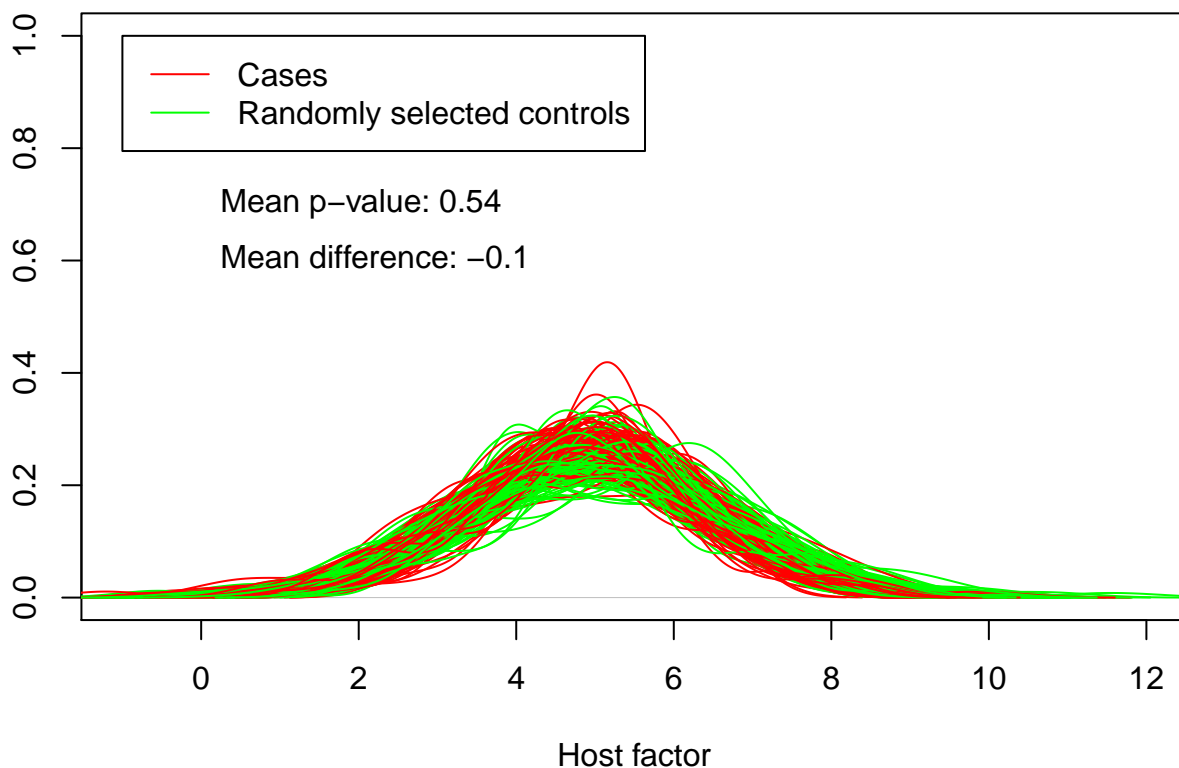
```
> ###Select controls for each simulated dataset
> sim <- sim.cc(x=sim, method="nearest")
>
```

5.3.1 Host factor values among randomly selected controls

Finally, we create density plots for values of the hypothetical host factor among cases and randomly selected controls:

```
> ###Plot for randomly selected controls
> plot.sim.cc(sim, type="random", type.label="Randomly selected controls",
+           main="Host factor densities among cases \n and randomly selected controls",
+           score="marker", lty=1, print.p=c(0,0.7), print.diff=c(0,0.6))
>
```

Host factor densities among cases and randomly selected controls



This plot demonstrates that randomly selecting controls results in cases and controls having very similar host factor values, suggesting that there is not an association between the host factor and infection. We know that this is a *false-negative* association because the call in Section 5.2 specified that the average host factor value was two units higher among cases than HESNs.

In order to understand why we observed a false-negative association, we can look at the distribution of exposure scores between cases and randomly selected controls.

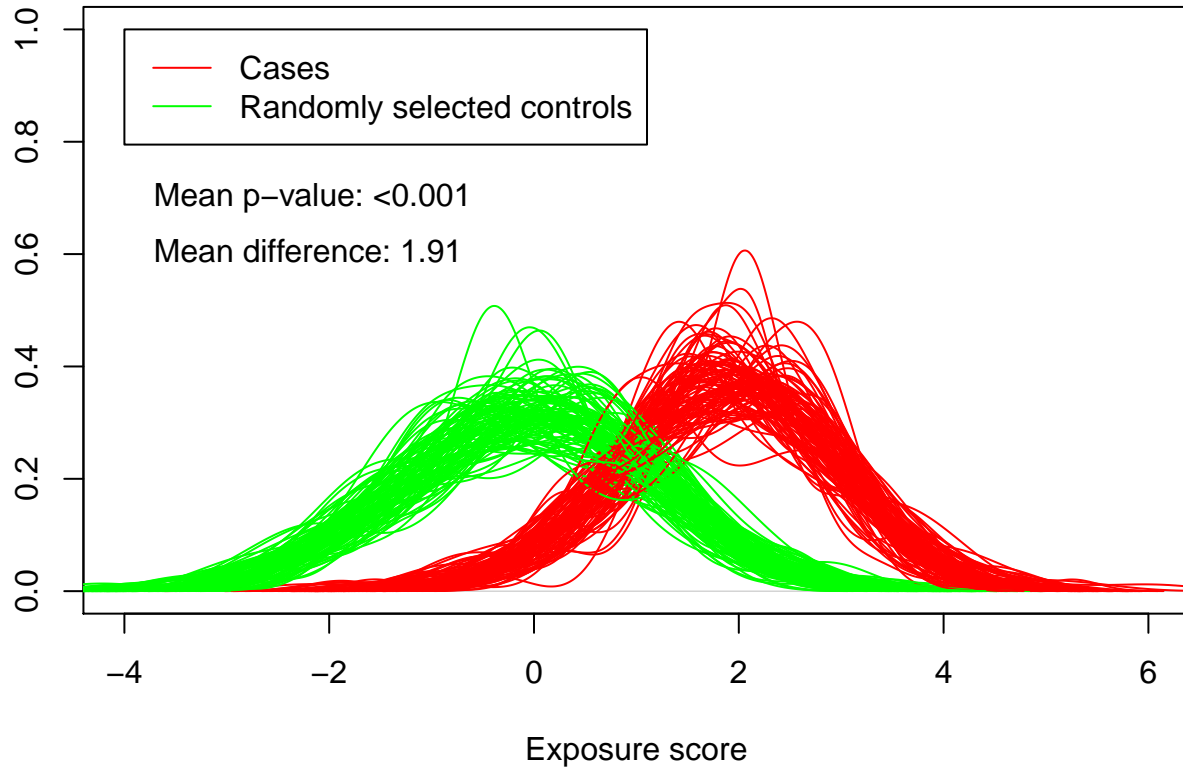
²`sim.cc()` selects matched controls using the `matchit()` function which allows various algorithms for optimal matching. For this example, we matched controls using nearest neighbor matching

```

> plot.sim.cc(sim, type="random", type.label="Randomly selected controls",
+           main="Exposure score densities among cases \n and randomly selected controls",
+           score="escore", lty=1, print.p=c(-3.9, .7), print.diff=c(-3.9, .6))
>
>

```

Exposure score densities among cases and randomly selected controls



This shows that cases had higher exposure scores than randomly selected controls, and in combination with knowledge that exposure scores are associated with the host factor and infection, explains the false-positive result.

5.3.2 Host factor values among cases and exposure matched controls

Next, we evaluated association between the hypothetical host factor and case/control status when 1:1 matching on exposure scores was used to select controls.

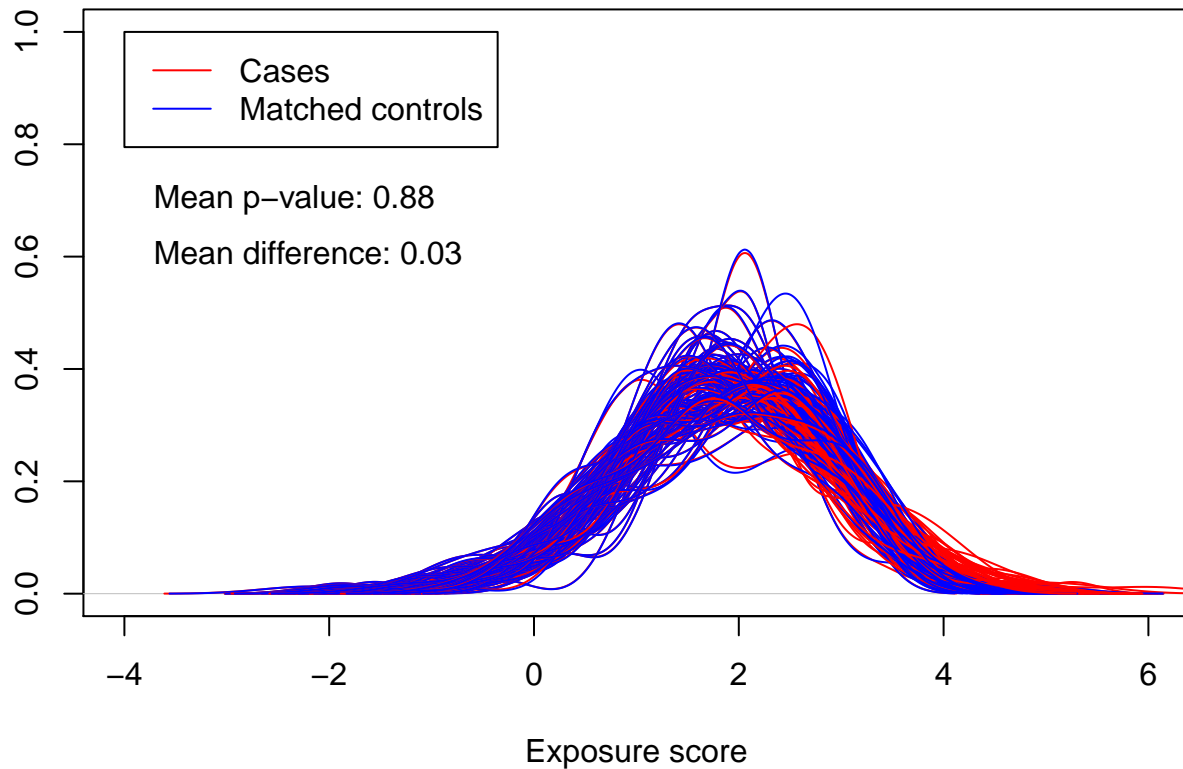
Plotting the exposure score densities demonstrates that, as expected, this results in cases and controls without any systematic differences in exposure.

```

> plot.sim.cc(sim, type="matched", type.label="Matched controls",
+           main="Exposure score densities among cases \n and matched controls",
+           score="escore", lty=1, cols=c("red", "blue"), print.p=c(-3.9, .7),
+           print.diff=c(-3.9, .6))
>
>

```

Exposure score densities among cases and matched controls



Comparing hypothetical host factor values between cases and matched controls, on the other hand, reveals that the average difference is approximately -2, which is what we specified as the association between the host factor and infection status.

```
> ###Plot for matchedcontrols
> plot.sim.cc(sim, type="matched", type.label="Matched controls",
+ main="Host factor densities among cases \n and matched controls",
+ score="marker", lty=1, cols=c("red", "blue"),
+ print.p=c(-.5, .7), print.diff=c(-.5, .6))
>
```

Host factor densities among cases and matched controls

