nopaco: a non-parametric concordance coefficient

Rowan Kuiper and Remco Hogenboezem

June 26, 2019

1 Introduction

Nopaco is a non-parametric concordance coefficient. It can be applied to determine the concordance between two or more repeated measurements in at least two subjects. These values have to be continuous in nature. Not all subjects need to have similar number of repeated measurements (i.e. unbalanced designs are allowed).

In this vignette we will explain the use of this package by tackling the problem depicted in Figure 1. The gene expression profiles for 100 subjects are determined by running microarrays in duplicate for each subject. Two gene models (model A and model B) that allow estimation of risk (e.g. survival), are applied to these profiles. For both models the concordance between risk scores are determined. In addition, the difference between both concordances will be assessed.

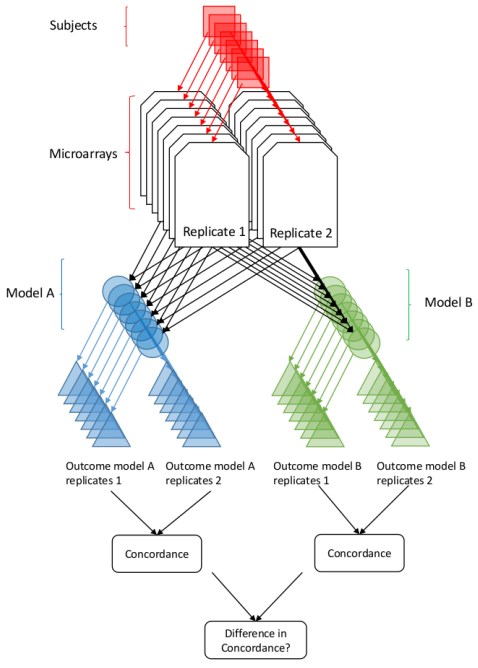

Figure 1: A hypothetical problem in order to illustrate the use of the package.

2 Running nopaco

```
> library(nopaco)
```

The data for the above hypothetical situation can be obtained by:

```
> data(scores)
```

The data is plotted in Figure 2.

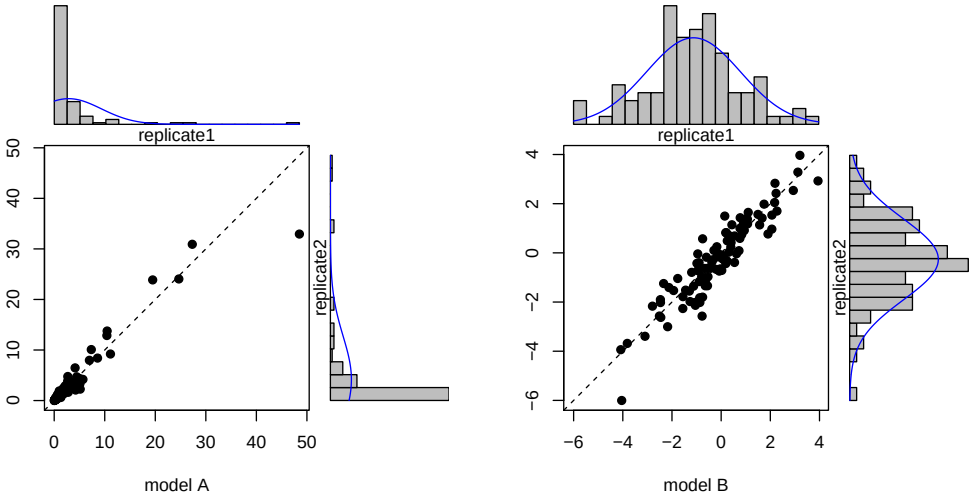

Figure 2: The scores for replicates 1 (horizontally) versus replicates 2 (vertically) as obtained by model A (left) and model B (right). The marginal distributions are given by histograms.

We can obtain the non-parametric concordance coeffcients for the scores obtained by model scores A or B:

```
> getPsi( scores[["modelA"]] )

[1] 0.9457576

> getPsi( scores[["modelB"]] )

[1] 0.9190909
```

These scores reflect the probability that any randomly drawn value out of the data matrix will not fit inbetween any random paired value (i.e. fit inbetween the replicate measurements). Higher probabiltities reflect better concordances. We can test whether or not these concordance are better than random:

```
> concordance.test( scores[["modelA"]] )

One sided concordance test
Alternative hypothesis: psi > 2/3
Estimates are obtained by  pvalue: exact and confidence interval: bootstrap
   psi lower 95%CI upper 95%CI        p
 0.946        0.93           1 2.91e-56

> concordance.test( scores[["modelB"]] )

One sided concordance test
Alternative hypothesis: psi > 2/3
Estimates are obtained by  pvalue: exact and confidence interval: bootstrap
   psi lower 95%CI upper 95%CI        p
 0.919       0.898           1 3.83e-41
```

Here model A seems to have a higher concordance than model B. As these concordances are based on the same underlying set of subjects, we can test for a difference in concordance between model A and model B:

```
> diffTest<-concordance.test(scores[["modelA"]], scores[["modelB"]])
> diffTest
```

```
Two sided concordance test for difference between two concordances
Alternative hypothesis: |psi1 - psi2| > 0
Estimates are obtained by  pvalue: simulation and confidence interval: bootstrap
  psi1  psi2   delta lower 95%CI upper 95%CI       p
 0.946 0.919 -0.0267     -0.0464    -0.00758 0.00704
```

The test shows that the two models have significantly different concordances. In order to extract the results we can use the following commands:

```
> coef(diffTest)

      psi1      psi2       delta lower 95%CI  upper 95%CI           p
 0.9457576 0.9190909 -0.02666667 -0.04636364 -0.007575758 0.007039517

> diffTest$pvalue

[1] 0.007039517

> diffTest$psi

[1] 0.9457576 0.9190909

> diffTest$ci

 lower 95%CI  upper 95%CI
-0.046363636 -0.007575758

> diffTest$alpha

[1] 0.05
```

3 Appendix

3.1 Generating hypothetical data

To create the example given above, let's assume a matrix $\boldsymbol{X}_0$ for 100 subjects in the rows and 3 genes in the columns. Matrix $\boldsymbol{X}_0$ contains the true expressions that are drawn from a multivariate normal distribution $\boldsymbol{X}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ in which the expected values and variances are described by $\boldsymbol{\mu}_0 = [0, 0, 0]$ and

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.4 \\ 0.4 & 0.4 & 1 \end{bmatrix}.$$

.

```
> library(MASS) ##Needed for rmvnorm
> set.seed(1);
> #A covariance matrix
> S <- matrix(0.4,ncol=3,nrow=3); diag(S)<-1
> #Underlying true expressions
> X0 <- mvrnorm(n=100,mu=c(0,0,0),Sigma=S)
```

However, in addition to the true expressions, a microarray experiment will generate a certain amount of noise which is reflected by matrix Z_k for replicate $k = (1..2)$. I.e. matrix $\boldsymbol{Z}_k$ contains replicate specific errors, which are drawn from $\boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{\mu}_\epsilon \boldsymbol{\Sigma}_\epsilon)$ with bias and variance $\boldsymbol{\mu}_\epsilon$ and $\boldsymbol{\Sigma}_\epsilon$. For both replicate $k = 1$ we assume no added bias (i.e $\boldsymbol{\mu}_\epsilon = [0, 0, 0]$). Also we assume no added variance for replicate $k = 1$ while replicate $k = 2$ has

$$\boldsymbol{\Sigma}_\epsilon = \begin{bmatrix} 0.0 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix},$$

resulting in measurement errors in the second replicate that are expected to increase for genes 1 to 3.

```
> #Measurement errors replicate 1
> Z1 <- mvrnorm(n=100,mu=c(0,0,0),Sigma=diag(c(0,0,0)))
> #Measurement errors replicate 2
> Z2 <- mvrnorm(n=100,mu=c(0,0,0),Sigma=diag(c(0,0.1,0.2)))
```

Then the observed expression profiles are defined as the true expressions plus measurement error $\boldsymbol{X}_k = \boldsymbol{X}_0 + \boldsymbol{Z}_k$ for replicate $k = (1..2)$

```
> #Replicate matrix 1: True expressions plus error
> X1 <- X0 + Z1
> #Replicate matrix 2: True expressions plus error
> X2 <- X0 + Z2
```

The next things to define are the two hypothetical models. Let the score of model A be $\boldsymbol{a} = e^{gene1+gene2}$ and the score of model B be $\boldsymbol{b} = gene2 + gene3$.

```
> a<-data.frame(
+ "replicate1"=exp(rowSums(X1[,1:2])),
+ "replicate2"=exp(rowSums(X2[,1:2])),
+ row.names=paste("patient",c(1:nrow(X1)),sep="")
+ )
> b<-data.frame(
+ "replicate1"=rowSums(X1[,2:3]),
+ "replicate2"=rowSums(X2[,2:3]),
+ row.names=paste("patient",c(1:nrow(X1)),sep="")
+ )
> scores = list(modelA=a,modelB=b)
>
>
>
```

3.2 Session info

```
> sessionInfo()

R version 3.6.0 (2019-04-26)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.6 LTS

Matrix products: default
BLAS:   /home/rowan/R/R3.6/R-3.6.0/lib/libRblas.so
LAPACK: /home/rowan/R/R3.6/R-3.6.0/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=nl_NL.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=nl_NL.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=nl_NL.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=nl_NL.UTF-8 LC_IDENTIFICATION=C

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

other attached packages:
[1] MASS_7.3-51.4 nopaco_1.0.4
```

```
loaded via a namespace (and not attached):
[1] compiler_3.6.0  Matrix_1.2-17   parallel_3.6.0  tools_3.6.0
[5] grid_3.6.0      lattice_0.20-38
```