

# Imputing missing values using the `pcaMethods` package

Wolfram Stacklies and Henning Redestig  
CAS-MPG Partner Institute for Computational Biology (PICB)  
and  
Max Planck Institute for Molecular Plant Physiology  
and  
Potsdam, Germany  
<http://bioinformatics.mpimp-golm.mpg.de/>

February 23, 2007

## 1 Missing value imputation

One application for missing value robust principal component analysis is that it effectively can be used to impute the missing values and thus obtain an estimated complete data set. The `pcaMethods` package was partly written with this application in mind.

PCA is a way of creating a model of a matrix,  $X$ , by defining two parameter matrices, the scores,  $T$ , and the loadings,  $P$ , which together have less values than the original matrix but when multiplied with each other will reconstruct the original matrix. I.e.:

$$X = 1 \times \bar{x} + T'P + E$$

where  $E$  is the error matrix and  $1 \times \bar{x}$  denotes the original variable averages. Now if  $X$  contains missing values but we still are able to get complete estimates of  $P$  and  $T$  then we can use:

$$\hat{X} = 1 \times \bar{x} + T'P$$

as an estimate for  $x_{i,j}$  if  $x_{i,j}$  is missing.

This can be done as the following example illustrates. First we attach the metabolite data set with missing values.

```
> library(pcaMethods)
```

```
Loading required package: MASS
```

```
Loading required package: ellipse
```

```
Loading required package: pls
```

```
Attaching package: 'pls'
```

The following object(s) are masked from package:stats :

loadings

```
> data(metaboliteData)
> mD <- metaboliteData
> sum(is.na(mD))
```

```
[1] 311
```

Now we get the estimated data set by using PPCA and three principal components.

```
> imputed <- pca(mD, nPcs = 3, method = "ppca")@completeObs
```

If we compare with the original values we see that the error is rather low.

```
> data(metaboliteDataComplete)
> mdComp <- metaboliteDataComplete
> sum((mdComp[is.na(mD)] - imputed[is.na(mD)])^2)/sum(mdComp[is.na(mD)]^2)
```

```
[1] 0.2194447
```

If the data we are interested in was gene expression set of class 'exprSet' we could simply do

```
> library(Biobase)
```

Welcome to Bioconductor

Vignettes contain introductory material.

To view, simply type 'openVignette()' or start with 'help(Biobase)'.

For details on reading vignettes, see the openVignette help page.

```
> data(exSet)
> exSetNa <- exSet
> exSetNa@exprs[sample(13000, 200)] <- NA
> lost <- is.na(exSetNa@exprs)
> impExSet <- pca(exSetNa, nPcs = 2, method = "ppca")[[2]]
> sum((exSet@exprs[lost] - impExSet@exprs[lost])^2)/sum(exSet@exprs[lost]^2)
```

```
[1] 0.05928953
```

Different results will be obtained with different PCA algorithms. Which one to use depends on the general structure of the data set and the imputation performance can be estimated by cross-validation. Please see the 'introduction' vignette on further details on how to use the cross-validation capabilities of this package.