PCIT: Quick Start Guide

Watson-Haigh, N.S., Kadarmideen, H.N. & Reverter, A., 2010. PCIT: an R package for weighted gene co-expression networks based on partial correlation and information theory approaches. *Bioinformatics*, 26(3), 411-413.[DOI: 10.1093/bioinformatics/btp674]

Firstly, install the latest version of the R programming environment and then install the `PCIT` package:

```
> install.packages("PCIT")
```

If you are using an MPI parallel programming environment, ensure you install and configure the `Rmpi` package:

```
> install.packages("Rmpi")
```

If you are using MS Windows on a system with multiple processing cores and wish to utilise them for parallel processing, we advise you to follow instructions provided by the Department of Statistical & Actuarial Sciences at the University of Western Ontario: http://www.stats.uwo.ca/faculty/yu/Rmpi/ for running `Rmpi` under DeinoMPI as this is the simplest for utilising standalone dual and quad-core MS Windows machines under MPI.

Once R is started you may view documentation, run demos and load example data using standard R functions:

```
# View information about the PCIT package
> help(package="PCIT")
# View help for the pcit() function
> ?pcit
# View/run demos from the PCIT package
> demo(package="PCIT")
> demo(PCIT)
# View/load data sets from the PCIT package
> data(package="PCIT")
> data(PCIT)
```

Since the PCIT algorithm is not restricted to GCNs but can be applied to any correlation-based network, we have not provided functions for handling microarray data. Instead we expect the user to appropriately handle, normalise and filter data, whatever its origin, and provide the correlation matrix for use in PCIT. However, we provide here a brief example for handling microarray data in the form of an `ExpressionSet` object from Bioconductor (Gentleman et al. 2004):

```
# Install the Biobase package from Bioconductor
> source("http://bioconductor.org/biocLite.R")
> biocLite(c("Biobase"))
# Load the Biobase package
> library(Biobase)
# Get/create an ExpressionSet object from
# somewhere: we assume eSet is this object
> dims(eSet)
         exprs
Features  1000
Samples     15
# Calculate a correlation matrix of the
# transposed expression data
> c <- cor(t(exprs(eSet)))
# The amount of RAM required for running PCIT
# on a correlation matrix of a specified size
# can be approximated
> pcitMemoryRequirement(nrow(c), units="MB")
$RAM
[1] 22.88818
$units
[1] "MB"
# The maximum correlation matrix size (number of
# rows and columns) which can be handled with a
# specified amount of RAM can be approximated
> maxMatrixSize(ram=1, units="GB")
[1] 6688
```

The function of applying the PCIT algorithm is `pcit()`. In its simplest form, it only requires a correlation matrix as input and returns a result from which one can obtain the correlation matrix indices for those correlations deemed to be significant.

```
# Perform PCIT on the correlation matrix
> result <- pcit(c)
# Get indices for the meaningful correlations
> signif <- idx(result)
# Plot the distribution of meaningful
# correlations superimposed on all correlations
> plotCorCoeff(c, list("PCIT Meaningful" =
+   signif), col=c("red"))
# Get the indices for the non-meaningful
# correlations
> nonsignif <- idxInvert(nrows(c), signif)
# Set non-meaningful correlations to zero
> c[nonsignif] <- 0
# Create an adjacency matrix from c e.g. by
# using the absolute correlation values
> adj <- abs(c)
```

The adjacency matrix, `adj`, contains only those edges deemed to be meaningful by the PCIT algorithm and can be used in subsequent network analyses.

REFERENCES

Gentleman, R.C. et al., 2004. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10), R80.