An introduction to mass univariate analysis of three-dimensional phenotypes

Carlo Biffi

2017-04-25

The R tools provided by `mutools3d` package enable to derive association maps between clinical and genetic variables and three-dimensional (3D) phenotypes defined on a triangular mesh. As an examplar application, we provide data for the study of the association between a synthetic clinical variable and wall thickness defined on a 3D left ventricular mesh (atlas). In this tutorial, we are going to illustrate:

1. how to derive the regression coefficient at each vertex of the atlas representing the association between WT and the clinical variable under study (mass univariate regression).
2. how to boost belief in extended areas of signal by using threshold-free cluster enhacement (TFCE) together with an appropriate permutation strategy (Freedman-Lane procedure) to derive new p-values.

Mass Univariate Regression

Mass univariate regression consists of applying a general linear model $Y = X\beta + \epsilon$ at each atlas vertex. The package provides functions `mur` and `murHC4m` to perform mass univariate regression, with the second function also applying HC4m heteroscedascity consistent estimators to correct for violation of homoscedasticity linear regression assumption. Input are a matrix `X` containig the clinical variables to study for each subject and a matrix `Y` containing at different columns the values at different vertices of a three-dimensional phenotype - in this case wall thickness.

To load the dataset included in the package, please run:

```
library(mutools3D)
data(Xtest)
data(Ytest)
```

`X` in this case is a [50x9] matrix containing the synthetic clinical variable to study together with other covariates to adjust the model and the intercept term, which must always be defined in the first column. Categorical variables must be always coded using “dummy” coding. For example, in this case the variable ethnicity which had four levels has been coded by picking the level 1 as a reference level and by creating three dichotomous variables, where each variable contrasts with level 1. `Y` is a [50x27623] matrix containing the values of a three-dimensional phenotype at all the vertices under study.

The output of the mass univariate functions is a matrix storing the regression coefficient β, the t-statistic and the p-values computed at each vertex and for each variable specified.

```
#extract results for synthetic variable
extract = 6

#it is also possible to study more than one variable at the same time
#extract = c(5,6)

result = mur(X,Y, extract)
#or
result = murHC4m(X,Y, extract)
```

Threshold-free cluster enhancement

Threshold-free cluster enhancement (TFCE) is an image-enhancement technique that computes at each vertex of the triangular mesh a score based on the extent and magnitude of the area of coherent signal that surrounds it (Smith and Nichols 2009). Given a statistical map on a 3D mesh, the TFCE function computes the corresponding TFCE map. In the code, the 3D mesh is represented as a computational graph using a two-columns matrix containing the mesh edges definitions: the first column contains the ID of one vertex, the second column contains the ID of the second vertex. To load an example of this matrix together with a V-dimensional vector (V = number of vertices in the mesh) storing the area associated with each vertex in the mesh, please run:

```
#load data for TFCE
data(NNmatrix)
data(areas)
```

TFCE can be then executed by using following code

```
#run TFCE on the t-statistic map previously obtained
TFCEresults = TFCE(result[,2], A, NNmatrix)
```

TFCE-derived pvalues can be computed by performing permutation testing on the input data. We developed a specific function to do so and that implements the Freedman-Lane procedure for permutation testing of the general linear model (Winkler and others 2014).

```
#compute TFCE-derived pvalues using Freedman-Lane procedure.

nPermutations = 1000
HC4m = TRUE
parallel = TRUE
nCores = 8
verbOutput = 0

TFCEresults = permFL(X, Y, extract, A, NNmatrix, nPermutations, HC4m, parallel, nCores, verbOutput)
```

If `verbOutput` is set to 0 the output is a matrix containing in its rows the pvalues computed at each vertex while the number of columns referes to the variables specified in extract. If `verbOutput` is set to 1 the output is a list where the `pval` field contains the pvalues computed at each vertex, `TFCEmatrix` field contains a V x nPermutations matrix containing the TFCE scores computed for each permutation and the tfceScores field is a V-dimensional vector containing the TFCE scores of the non-permuted data.

Last but not least, the derived pvalues using either TFCE or the standard mass univariate approach need to be corrected for multiple testing as we are performing statistical hypothesis testing at each atlas vertex. For doing so, we suggest to use multtest package.

REFERENCES

Smith, Stephen M, and Thomas E Nichols. 2009. “Threshold-Free Cluster Enhancement: Addressing Problems of Smoothing, Threshold Dependence and Localisation in Cluster Inference.” *NeuroImage* 44 (1). Elsevier: 83–98.

Winkler, Anderson M, and others. 2014. “Permutation Inference for the General Linear Model.” *NeuroImage* 92. Elsevier: 381–97.