Structural MRI statistics using RMINC v. 0.4

Author:
Jason Lerch

February 3, 2008

Preface

This mini-book attempts to provide a general introduction to the statistics side of structural brain imaging, with a heavy emphasis on practical worked examples. It also introduces a particular toolkit, RMINC, designed to make running these types of statistical analyses easier. It is targeted at the general user, who may or may not have some statistical background, but does have some data they want analysed in a straightforward way. It is not meant to be a complete handbook on statistics, but hopefully will provide enough of a primer to get by, at least for a little while.

This mini-book exists for a number of reasons. I have over the years been asked multiple questions relating to structural brain imaging and statistics, and have had the chance to learn answers to those questions from countless people. This book thus exists as an attempt to put some of those answers down on paper. Secondly, writing this book is part and parcel of the development of RMINC; it is easier to write code useable by others if one documents it first, and then writes code to fit the documentation.

The book will likely be an incomplete work in progress for a long while yet. The LaTeXsource for this book are packaged along with RMINC itself, and contributors are most welcome!

Contents

Chapter 1

Introduction

The process of analysing brain imaging data is typically comprised of a series of stages. The study is designed with various choices made about the biology question that is to be addressed and the data necessary to answer the questions thus posed. Then the data is then acquired, and once that is completed, the images are processed in various automatic, semi-automatic, or manual ways and then analysed.

This book deals mainly with the final part, the data analysis, though there will be several side-tracks into the other topics. A single example will be used throughout: a mouse brain imaging study comparing two genotypes. The methods described herein should be easily transferable to any other structural imaging study which looks at brain shape, tissue classification, or signal intensities.

1.1 Installing the tools

All the analyses will be performed using RMINC, which is a library designed to handle MINC volumes inside the R statistical environment. All the tools needed are freely available, and should run on just about any computer/operating system. Installation and setup is described in some more detail below.

1.1.1 R

Quick background about R.
Installing R.
Where to find further reading.

1.1.2 MINC

Quick background about MINC.
Installing MINC.
Where to find more information.

1.1.3 RMINC

Quick background about RMINC.

Installing RMINC.

1.2 Overview of the analysis process

The data analysis process usually proceeds in the following way. First the input images are assessed for correctness; any obvious processing errors are removed from any subsequent analyses. The question of what constitutes an outlier is often a tricky one. In order to avoid the temptation to manipulate the data in a biased way it is best if the person who reviews the input data is blind about the categorization of each particular dataset.

Once all the acceptable datasets are in place a series of descriptive statistics can be generated, usually consisting of means and standard deviations of all images in the study as well as of all the subgroupings. This is followed by generating statistical maps of the main variables of interest. These are then thresholded for significance while taking multiple comparisons into account. There is then often a series of steps in which new statistical models are analyzed and thresholded until the results become more understandable. This usually involves lots of plotting of individual datapoints.

Chapter 2

Preparing the data

This chapter will briefly discuss how to generate structural imaging data useable for the statistical analyses described in the rest of the book.

2.1 Types of datasets

Describe what can be done.

2.1.1 Voxel based morphometry

Some more detail on VBM.

2.1.2 Deformation based morphometry

Some more detail on DBM.

2.1.3 Other

Mention cortical thickness, manual segmentation, etc.

2.2 Input data

Preparing the spreadsheet.

Reading study information into R.

Once the files have been processed, the easiest way to proceed is by settting up a text file containing all the necessary information about each scan. This file should be comma or space separated, have one row per scan, with each column containing info about each scan. One of the columns should contain the filename pointing to the MINC volumes to be processed. An example might look like the following:

```
file,Genotype,scale
filename1.mnc,+,0.98
filename2.mnc,-,0.91
filename3.mnc,-,0.92
```

Notice how the first row contains a header. This is optional, but makes later access to the data easier and is therefore recommended.

The next step is to actually load this filename into R. The steps are given below:

```
> library(RMINC)
> gf <- read.csv("control-file.glim")
```

The two library commands load the RMINC and xtable libraries into R. xtable is only necessary for display purposes in this manual (which, by the way, is being written using Sweave, a tool for combining R with latex). The variable `glim.file` is then assigned the location of the input data file, which in turn is read into the `gf` variable.

```
> xtable(gf[c(1, 2, 3, 22, 23, 24), c(2, 4, 5, 6)], caption = "GLIM File")
```

	Cage	Genotype	Internal.ID	Weight..g.
1	1.00	-	475738.00	51.33
2	1.00	+	475740.00	39.54
3	2.00	-	475755.00	36.54
22	13.00	+	475959.00	39.36
23	13.00	-	475963.00	48.60
NA				

Table 2.1: GLIM File

The little code fragment and table above just shows what a subset of the gf variable looks like. To get a feel for the data and some of the RMINC functions, the first file is then read and a histogram produced (for the curious, the files used in this example actually consist of log Jacobians of deformation fields, not voxel density maps usually used in VBM).

Chapter 3

Descriptive statistics

Introduction to what descriptive statistics are and what they are good for.

Group means and standard deviations.

```
> library(RMINC)
> overall.mean <- mincMean(gf$jacobians)
```

```
Method: mean
Number of volumes: 23
Volume sizes: 141 274 210
N GROUPS: 1.000000
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done
```

```
> overall.mean
```

```
Multidimensional MINC volume
Columns:
[1] "/projects/mice/jlerch/frankland/tgf-beta/2007/tgf-beta_processed//img_23nov07.0.novembe
```

Writing to file.

```
> mincWriteVolume(overall.mean, "overall-mean.mnc")
```

```
Writing column 1 to file overall-mean.mnc
Sizes: 141 274 210
Range: 0.062284 -0.210842
```

Most often we are more interested in how the means break down by the grouping in this dataset. This can be done by adding another variable to the mincMean call:

```
> group.means <- mincMean(gf$jacobians, gf$Genotype)
```

```
Method: mean
Number of volumes: 23
Volume sizes: 141 274 210
N GROUPS: 2.000000
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done
```

```
> group.means
```

```
Multidimensional MINC volume
Columns:       - +
[1] "/projects/mice/jlerch/frankland/tgf-beta/2007/tgf-beta_processed//img_23nov07.0.novembe
```

The *Genotype* variable has two levels in it: + and -. So it will take the mean for all subjects in each group. These can then be written to file by specifiying the column.

```
> mincWriteVolume(group.means, "group1.mnc", "+")
```

```
Writing column + to file group1.mnc
Sizes: 141 274 210
Range: 0.672118 -0.429721
```

```
> mincWriteVolume(group.means, "group2.mnc", "-")
```

```
Writing column - to file group2.mnc
Sizes: 141 274 210
Range: 0.351076 -1.169893
```

If the difference between the two columns is of interest, one can just subtract the two data columns:

```
> difference <- group.means[, "+"] - group.means[, "-"]
> mean(difference)
```

```
[1] -0.002525745
```

```
> mincWriteVolume(difference, "diff.mnc", gf$jacobians[1])
```

```
Sizes: 141 274 210
Range: 1.838180 -0.780371
```

Notice how *mincWriteVolume* now needs a third argument: the name of a minc-file which has the same dimensions as the data. By default commands such as *mincMean* will store that information; after the subtraction above, however, the result is just a series of numbers with all metadata removed, so it has to be specified when writing the data to file.

Of course means are not the only items of interest. Also computable are the standard-deviations, variances, and sums, as illustrated below. Just like *mincMean* a column of filenames is required and a grouping variable is optional.

```
> v <- mincVar(gf$jacobians, gf$Genotype)

Method: var
Number of volumes: 23
Volume sizes: 141 274 210
N GROUPS: 2.000000
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done

> s <- mincSd(gf$jacobians)

Method: var
Number of volumes: 23
Volume sizes: 141 274 210
N GROUPS: 1.000000
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done

> s2 <- mincSum(gf$jacobians, gf$Genotype)

Method: sum
Number of volumes: 23
Volume sizes: 141 274 210
N GROUPS: 2.000000
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done
```

Chapter 4

Linear Models

4.1 First linear model

```
> vs <- mincLm(jacobians ~ Genotype, gf)

Method: lm
Number of volumes: 23
Volume sizes: 141 274 210
N: 23 P: 2
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done

> vs

Multidimensional MINC volume
Columns:       F-statistic (Intercept) Genotype+
[1] "/projects/mice/jlerch/frankland/tgf-beta/2007/tgf-beta_processed//img_23nov07.0.novembe

> mincWriteVolume(vs, "simple-lm.mnc", "Genotype+")

Writing column Genotype+ to file simple-lm.mnc
Sizes: 141 274 210
Range: 19.596058 -18.206707
```

mincLm is the command to run linear models in RMINC. Its basic use is to provide a formula (same syntax as the R `lm` command) with the left side containing the filenames, the right side the variables to be regressed. The output of mincLm depends on the formula. There will always be a column of F-statistics, representing the significance of the entire model. Then there is one column for each of the terms in the mode. The above linear model, relating the Jacobian determinant to Genotype, will thus have three columns:

F-statistic representing the significance of the entire model.

(Intercept) the intercept term - this term is rarely interesting, as it tests for whether the intercept is 0. There's no reason to believe it should be in most cases, so this value will be highly significant but meaningless.

Genotype+ the term testing whether the "+" level of the Genotype factor is significant. In this case this term is the most interesting and therefore the one written to file.

The output is placed into a variable that can be written to file in the same way as described in the descriptive statistics section.

4.2 Plotting voxels

```
> options(show.signif.stars = FALSE)
> voxel <- mincGetVoxel(gf$jacobians, 0, 0, 0)
> summary(lm(voxel ~ Genotype, gf))

Call:
lm(formula = voxel ~ Genotype, data = gf)

Residuals:
       Min         1Q     Median         3Q        Max
-2.422e-05 -7.890e-06  4.447e-06  6.224e-06  1.351e-05

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.035e-06  3.728e-06   2.423   0.0245
Genotype+   -7.288e-06  5.162e-06  -1.412   0.1726

Residual standard error: 1.237e-05 on 21 degrees of freedom
Multiple R-Squared: 0.0867,        Adjusted R-squared: 0.04321
F-statistic: 1.994 on 1 and 21 DF,  p-value: 0.1726

> vs[1, ]

F-statistic (Intercept)   Genotype+
   1.993651    2.423246   -1.411967
```

The code above does the following: it gets the voxel from coordinates 0,0,0 for all subjects, then computes a linear model relating that voxel to Genotype using standard R functions. Lastly it prints the results from that same voxel as computed by `mincLm`. This helps illustrate what the output of `mincLm` stores: the F-statistic is the same as can be found in the last line of the summary command, and the t-statistics for the Intercept and Genotype column can be found under "t-value" when using standard R functions.

`mincGetVoxel` needs three coordinates, given in voxel space in the same order as stored in the file. Just printing the voxel will show the corresponding world coordinates:

```
> voxel

 [1]  1.525891e-05  1.268326e-05  1.525891e-05 -1.518142e-05  1.525891e-05
 [6]  1.525891e-05  1.332761e-05  1.109743e-05 -1.106965e-05 -1.242381e-05
[11]  5.798346e-06  1.525891e-05 -7.485501e-06  1.525891e-05  1.525891e-05
[16]  1.288196e-05  6.193524e-06 -1.648016e-05  1.525891e-05  1.525891e-05
[21] -5.299584e-07 -2.187095e-05  6.070320e-06

Voxel Coordinates: 0 0 0
World Coordinates: -6.27 -8.19 -4.2
```

If the coordinates are specified in world coordinates then `mincGetWorldVoxel` is what you want - it also takes three coordinates, this time in world space in xspace,yspace,zspace order:

```
> world.voxel <- mincGetWorldVoxel(gf$jacobians, -6.27, -8.19,
+     -4.2)
> world.voxel

 [1]  1.525891e-05  1.268326e-05  1.525891e-05 -1.518142e-05  1.525891e-05
 [6]  1.525891e-05  1.332761e-05  1.109743e-05 -1.106965e-05 -1.242381e-05
[11]  5.798346e-06  1.525891e-05 -7.485501e-06  1.525891e-05  1.525891e-05
[16]  1.288196e-05  6.193524e-06 -1.648016e-05  1.525891e-05  1.525891e-05
[21] -5.299584e-07 -2.187095e-05  6.070320e-06

Voxel Coordinates: 0 0 0
World Coordinates: -6.27 -8.19 -4.2
```

4.3 Using subsets

It is quite common to want to run a linear model on only a subset of the data. This can be quite easily accomplished in `mincLm` using an extra subsetting specification:

```
> vs <- mincLm(jacobians ~ Genotype, gf, Sex == "M")

Method: lm
Number of volumes: 14
Volume sizes: 141 274 210
N: 14 P: 2
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done

> vs
```

```
Multidimensional MINC volume
Columns:       F-statistic (Intercept) Genotype+
[1] "/projects/mice/jlerch/frankland/tgf-beta/2007/tgf-beta_processed//img_23nov07.0.novembe
```

This is the same linear model command as executed above, but this time using only the male mice for the computation.

4.4 Multiple Comparisons

The example below illustrates the entire process involved in going running a linear model and correcting for multiple comparisons using the False Discovery Rate:

```
> vs <- mincLm(jacobians ~ Sex * Genotype, gf)

Method: lm
Number of volumes: 23
Volume sizes: 141 274 210
N: 23 P: 4
In slice
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  2
Done

> qvals <- mincFDR(vs, mask = "mask.mnc")

Sizes: 141 274 210
Start: 0 0 0
Count: 141 274 210

Computing FDR threshold for all columns
  Computing threshold for  F-statistic
  Computing threshold for  (Intercept)
  Computing threshold for  SexM
  Computing threshold for  Genotype+
  Computing threshold for  SexM:Genotype+

> qvals

Multidimensional MINC volume
Columns:       F-statistic (Intercept) SexM Genotype+ SexM:Genotype+
[1] "/projects/mice/jlerch/frankland/tgf-beta/2007/tgf-beta_processed//img_23nov07.0.novembe
Degrees of Freedom: 3 19
FDR Thresholds:
     F-statistic (Intercept)     SexM Genotype+ SexM:Genotype+
0.01   4.0360423    3.773553 5.809281  3.193820            NaN
0.05   1.8909837    2.474292 4.637493  2.138480       6.575371
0.1    1.0476866    1.945903 3.980688  1.622814       5.902329
```

```
0.15   0.5447665    1.608717 3.502085  1.282796       5.615089
0.2    0.1244980    1.337898 3.089758  1.010200       5.381285

> mincWriteVolume(qvals, "Genotype-FDR.mnc", "Genotype+")

Writing column Genotype+ to file Genotype-FDR.mnc
Sizes: 141 274 210
Range: 1.000000 0.000008
```

The first command computes an ANOCA using `mincLm`. The results are then passed on to `mincFDR`, which computes the False Discovery Rate threshold separately for each of the terms in the linear model. Only results from within the mask specified as an optional argument to `mincFDR` are considered. The thresholds detected at different levels (0.01, 0.05, 0.10, 0.15, and 0.20) are then printed out. The "Genotype+" column is then written to file.

Chapter 5

Bits and Pieces

5.1 Correlations

5.2 non-parametric statistics

5.3 mixed effects models

5.4 Cleaning up