The Journal of Open Source Software

miet: an R package for region of interest analysis from magnetic reasonance images

Benoit Combès[1]

1 Empenn INSERM - Institut National de la Santé et de la Recherche Médicale, Inria Rennes – Bretagne Atlantique , IRISA_D5 - SIGNAUX ET IMAGES NUMÉRIQUES, ROBOTIQUE

DOI: 10.21105/joss.01862

Software

- Review
- Repository
- Archive

Editor: Ariel Rokem

Reviewers:

- @janfreyberg
- @neuroimaginador

Submitted: 29 October 2019
Published: 27 January 2020

Summary

In a medical imaging study, a set of magnetic resonance imaging (MRI) data is generally stored as a hierarchy of folders containing MRI volumes. These volumes may correspond to a variety of e.g. subjects, subject-types, centers, acquisition types that structures the folders naming and hierarchy. Here is an example of such structure:

```
|-- center01
|   |-- healthySubject01
|   |   `-- month0
|   |-- healthySubject02
|   |   `-- month0
|   |-- patient01
|   |   |-- month0
|   |   `-- month12
|   `-- patient02
|       |-- month0
|       `-- month12
`-- center02
    |-- healthySubject01
    |   |-- month0
    |   `-- month12
    |-- healthySubject02
    |   |-- month0
    |   `-- month12
    |-- patient01
    |   |-- month0
    |   `-- month12
    `-- patient02
        |-- month0
        `-- month12
```

Each of the folders at the bottom of the hierarchy (here `month0` or `month12`) contains a set of files with MRI volumes for the given subject (that may be, for example, either a healthy subject or a patient) at the given time of their follow-up, acquired in the given center. These volumes may consist of a set of raw data as well as post-processed data such as segmentation masks, co-registered volumes, etc. that are produced from image processing tools. Obviously, the same dataset could have been structured in many other ways.

Once all this data is produced, the next step generally consists in analyzing them. An ubiquitous analysis in medical imaging is the so-called region-of-interest based analysis that consists in comparing statistics of MR signals over sets of predefined regions (Poldrack, 2007).

Such an analysis needs to iterate over all MR volumes of interest to read them and to extract the relevant statistics. To the best of our knowledge, there is no dedicated framework to ease such tasks, which may result in lengthy technical analysis codes. `miet` - acronym of medical imaging extraction tools - is an R package attempting to fill that gap. More specifically, `miet` is designed to specify and extract tibbles (Müller & Wickham, 2019), which is an improved `data.frame` structure, ready for data analysis from a specified folder hierarchy and a set of extraction formulas.

A short example

`miet` is based on the following three successive steps:

1. Populate: Given a description of how the folders are structured, build a "population" tibble gathering information about data hierarchy and location.
2. Extract: Build a tibble that contains, for each element from your population, values extracted from the associated MRI volumes.
3. Expand: Gather information from a set of generated tibbles into a single tibble.

At the end of these three stages, we end up with a tibble ready for data analysis.

The four next subsections present an example involving a data set of spinal cord MRI volumes and use it to illustrate the three previous operations. For a reproducible example with publicly available (synthetic) data, see the `miet` readme (https://gitlab.inria.fr/miet/miet).

Setting

Our purpose in this short example is to assess the difference of an MRI measurement called the mean magnetization transfer ratio or MTR (Grossman, Gomori, Ramer, Lexa, & Schnall, 1994) from one scanner to another (also refer to as the scanner-effect). More precisely, we want to assess the difference of MTR measurements on two different scanners in a given anatomical region: voxels corresponding to specific vertebral levels (called C3C6) and assess their correlations with subject body mass index. For this purpose, we dispose from 3D volumes containing MTR values and 3D volumes containing vertebral levels for a set of 10 participants scanned on two different scanners (examples of data are given in Figure 1).

We consider that our data are contained in the following file hierarchy:

```
|-- center01
|    |-- 01
|    |-- 02
|    |---....
|
`-- center02
     |-- 01
     |-- 02
     |---....
```

Moreover, we assume that each of these folders contains (among others) the file mtr.nii.gz, the 3D MTR volume and the file vertebralLevel.nii.gz, that contains the volume representing the vertebral label of each voxel. We assume that in a given folder, these two volumes have the same geometry.

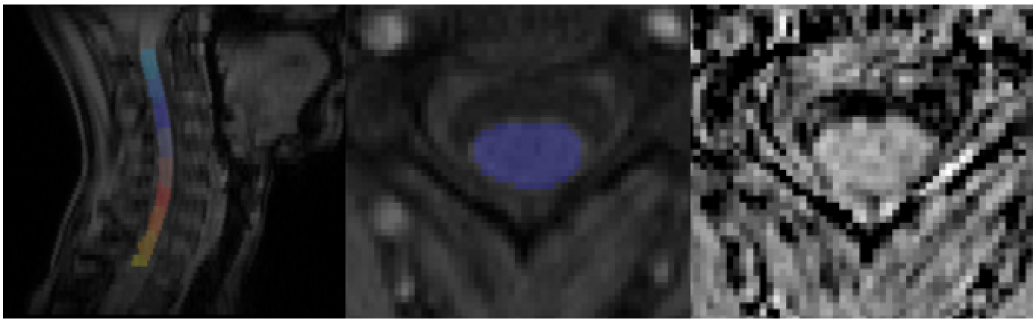

Figure 1: Example of volume data. From left to right: i) Sagital view of vertebral labeling vertebralLevel.nii.gz on the corresponding anatomical volume (0 is the background and is not displayed, values 1 to 9 labels respectively levels C1 to T2 and are color-coded). ii) Axial view (C3 level) of vertebral labeling vertebralLevel.nii.gz of the spinal cord on the corresponding anatomical volume. iii) Corresponding axial view for the magnetization transfer map mtr.nii.gz. Roughly speaking, the darker the pixel the lowest the myelin content.

Populate

First, we are going to parse the overall folder hierarchy to identify each subject and center. This is performed using the function miet_populate. This function uses a description of the folder hierarchy, given as regular expressions (an introduction to regular expression in R can be found in the stringr reference manual https://stringr.tidyverse.org/articles/regular-expressions.html), and extracts all suitable candidates:

```
library(miet)

# this is the folder hierarchy location
ROOT='/home/bcombes/dataExp/'

# this is how the folders are structured
pattern='{{centerId}}/{{subjectId}}'

# this specifies each element of variable 'pattern'
mEnv=new.env()
mEnv$centerId='center\\d{2}' #centerId is coded by "center" plus 2 digits
mEnv$subjectId='\\d{2}' #subjectId is coded by 2 digits

population=miet_populate(ROOT,pattern,mEnv)
print(population)
# A tibble: 20 x 2
#   centerId subjectId
#   <chr>    <chr>
# 1 center01 01
# 2 center01 02
# 3 center01 03
# 4 center01 04
# 5 center01 05
# ...
#[1] "With path variable(s):  {{path}}"
```

The resulting `population` variable is a tibble with one row for each of the final folder and two columns, named `centerId` and `subjectId`.

Extract

Once `population` is defined, for each of its elements, we want to extract the mean of volume `mtr.nii.gz` over the set of voxels of `vertebraeLevel.nii.gz` which equals 3,4,5 or 6. The function `miet_extract` is designed for this purpose:

```
MTRC3C6=miet_extract(population,
                     formulaLabel='[[vertebraeLevel]] %in% seq(3,6)',
                     valueListName='[[mtr]]',
                     statFunction=mean)
```

The function `miet_extract` iterates over the elements of `population`, extracts all the voxels satisfying a given R expression given as a string and computes a given statistic (here `mean`) over them. To specify such expressions, `miet` uses a special notation for MRI volume, `[[X]]` that can read as “the 3D array corresponding to MRI file X.nii.gz in the current acquisition folder”. The previous `miet_extract` call should thus read as:

```
For each acquisitions set in 'population'
    Compute the 'mean' of volume 'mtr.nii.gz' over voxels
    for which 'vertebraeLevel.nii.gz in [3,6]'
```

Looking at the content of the resulting tibble:

```
print(MTRC3C6)

# A tibble: 20 x 3
#  metric   uniqueId     value
#  <chr>    <chr>        <dbl>
#1 mean_mtr center01_01  35.0
#2 mean_mtr center01_02  34.7
#3 mean_mtr center01_03  34.5
#4 mean_mtr center01_04  35.3
#...
```

We obtain a 3-column-tibble. Its first column contains the name of the extracted metric (here it is always `mean_mtr`), the second one is an identifier toward the acquisition session and the last one is the value of the mean MTR on C3C6 for the given acquisition session. Notice that `miet` proposes other more versatile functions (e.g. `miet_apply`) to extract statistics from MRI.

Expand

Now, we want to include the subject body-mass index (that equals subject-weight (in kg)/ subject-size^2 (in meter^2)) in our data. This value is typically extracted from e.g., a csv file containing subject characteristics. In our example,

```
characteristics=read.csv(paste0(ROOT,'/characteristic.csv'),
                         colClasses=c('character','character',
                         'numeric','numeric','numeric'))
print(characteristics)

#   subjectId sex age size weight
#1         01   F  31 1.60     68
#2         02   M  34 1.70     80
#3         03   F  21 1.55     50
#4         04   M  27 1.73     62
#5         05   F  36 1.63     65
#...
```

The next step consists in gathering all the relevant elements (here, mean MTR, subject id and center as well as its corresponding weight and size characteristics) into a single tibble. `miet_expand` is designed for such a purpose and takes as arguments the population and a set of a tibbles to be merged with:

```
expandedDf=miet_expand(population,MTRC3C6,
                       characteristics)
head(expandedDf)

# A tibble: 6 x 9
#  uniqueId centerId subjectId mean_mtr    sex     age  size weight
#  <chr>    <chr>    <chr>           <dbl> <chr> <dbl> <dbl>  <dbl>
#1 center0… center01 01               36.6 F        31  1.6       68
#2 center0… center01 02               34.8 M        34  1.7       80
#3 center0… center01 03               35.7 F        21  1.55      50
#4 center0… center01 04               37.0 M        27  1.73      62
#...
```

This new tibble gathers all the needed information and is now ready to be visualised and analysed with the appropriate R packages.

While this falls out of the immediate scope of `miet`, we demonstrate that here:

```
library(ggplot2)
library(dplyr)

ggplot(expandedDf,aes(x=subjectId,y=mean_mtr))+
        geom_point(aes(color=centerId,size=weight/size^2))
```

The resulting plot is shown in Figure 2.

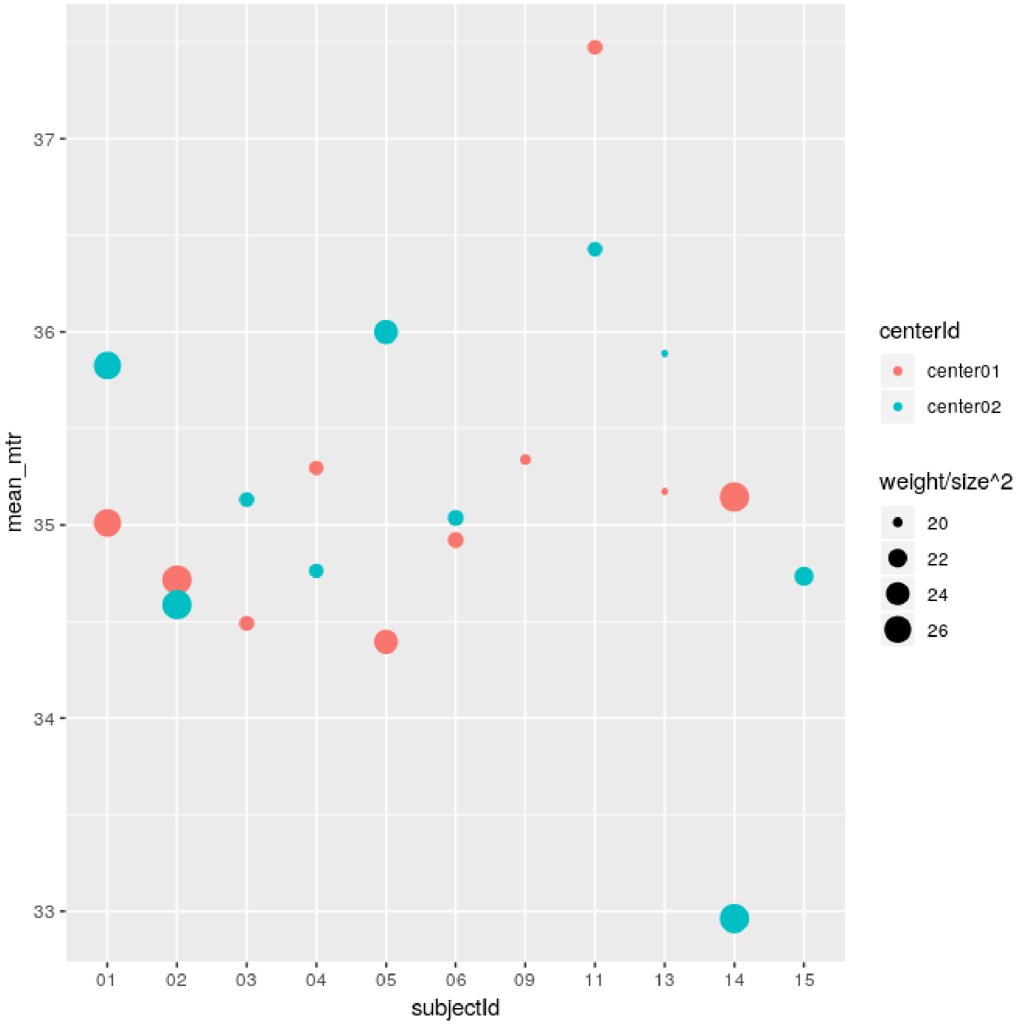

Figure 2: Dispersion of MTR on C3C6

Let's look at the statistics for each center:

```
expandedDf %>% group_by(centerId) %>%
           summarise(mu=mean(mean_mtr),sigma=sd(mean_mtr))

# A tibble: 2 x 3
#  centerId     mu sigma
#  <chr>     <dbl> <dbl>
#1 center01   35.2 0.863
#2 center02   35.1 0.988
```

The difference between the two centers is low, with respect to the data variability. Not surprisingly, there is no evidence against a none-null center effect:

```
summary(lm(data=expandedDf,mean_mtr~centerId+subjectId))

#Call:
#lm(formula = mean_mtr ~ centerId + subjectId, data = expandedDf)
```

```
#
#Residuals:
#    Min      1Q  Median      3Q     Max
#-1.0910 -0.3297  0.0000  0.3297  1.0910
#
#Coefficients:
#                    Estimate Std. Error t value Pr(>|t|)
#(Intercept)        3.542e+01  5.987e-01  59.154 7.41e-12 ***
#centerIdcenter02  6.812e-05  3.787e-01   0.000   0.9999
#...
```

We then choose to pool measurements from center01 and center02 to assess relationship to body mass index.

```
pooledDf = expandedDf %>% group_by(subjectId) %>%
          summarize(mtr=mean(mean_mtr),bmi=first(weight)/first(size)^2)

pooledDf %>% ggplot(aes(x=mtr,y=bmi)) + geom_point() + geom_smooth(method='lm')
```

The resulting plot is given in Figure 3.

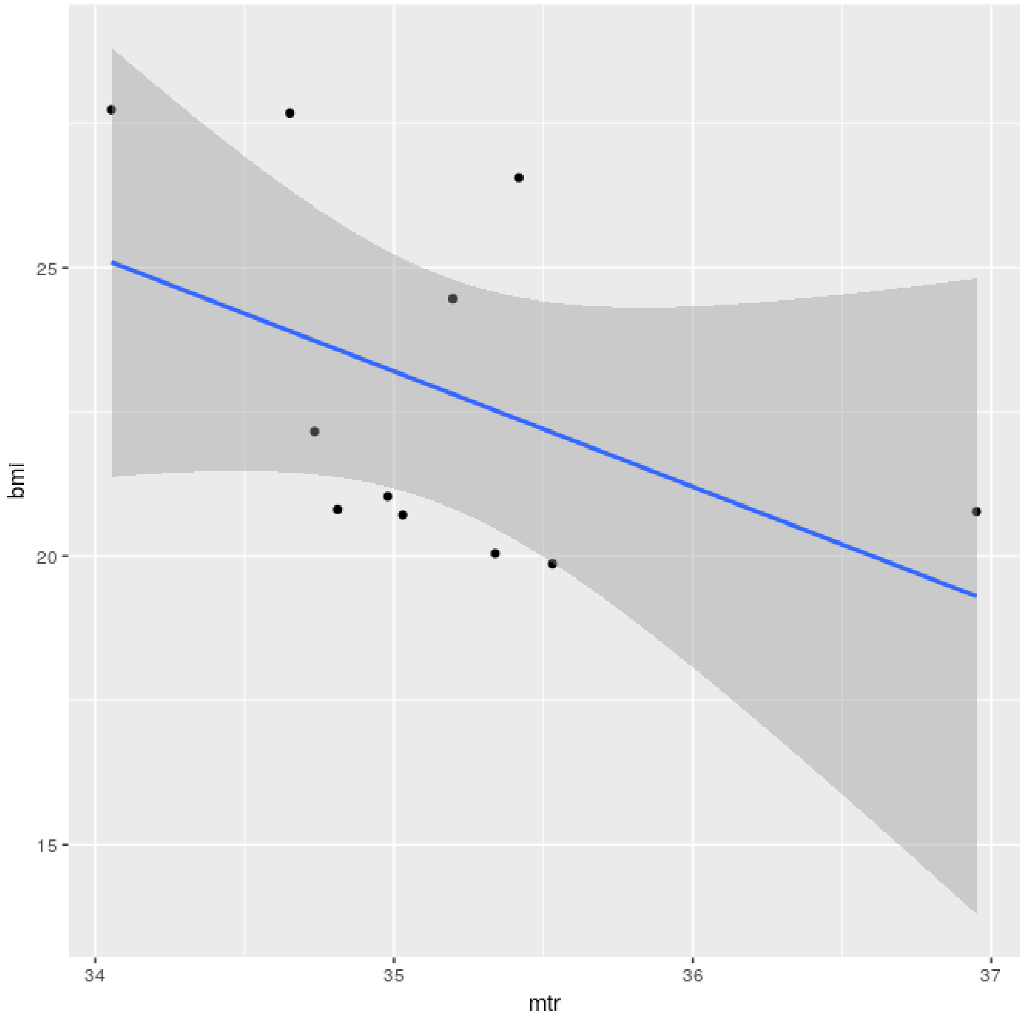

Figure 3: BMI as a function of MTR on C3C6

Given this visualization, it is hard to conclude on the relationship between BMI and MTR measurements on the basis of these data.

To conclude this simple example

To conclude this simple example, we outline that using `miet` avoided us to explicitly:

- write loops over patient/acquisition/…
- deal with MRI files

and thus allows code writing and reviewing to focus on the statistics of interest.

The example developed in this section is intentionally designed to be minimal: all default arguments match well with the situation. `miet` provides ways to deal with a variety of other situations. See the `miet` vignette (available at https://gitlab.inria.fr/miet/miet/blob/master/doc/vignette.pdf) for more details.

Availability

`miet` is an open source software made available under the MIT license. It can be installed from its gitlab repository using the `devtools` package: `devtools::install_git("https://gitlab.inria.fr/miet/miet.git")`.

Acknowledgements

We acknowledge Camille Maumet for sharing their enthusiasm toward the publication of this work, as well as for their numerous relevant comments. We acknowledge Julien Louis for their numerous relevant comments. We acknowledge the reviewers for their comments and suggestions.

References

Grossman, R. I., Gomori, J. M., Ramer, K. N., Lexa, F. J., & Schnall, M. D. (1994). Magnetization transfer: Theory and clinical applications in neuroradiology. *Radiographics*, *14*(2), 279–290. doi:10.1148/radiographics.14.2.8190954

Müller, K., & Wickham, H. (2019). *Tibble: Simple data frames*. Retrieved from https://CRAN.R-project.org/package=tibble

Poldrack, R. A. (2007). Region of interest analysis for fMRI. *Social cognitive and affective neuroscience*, *2*(1), 67–70. doi:10.1093/scan/nsm006