FixelArray:
An R package for fixel-wise statistics

Chenying Zhao

September 13, 2021

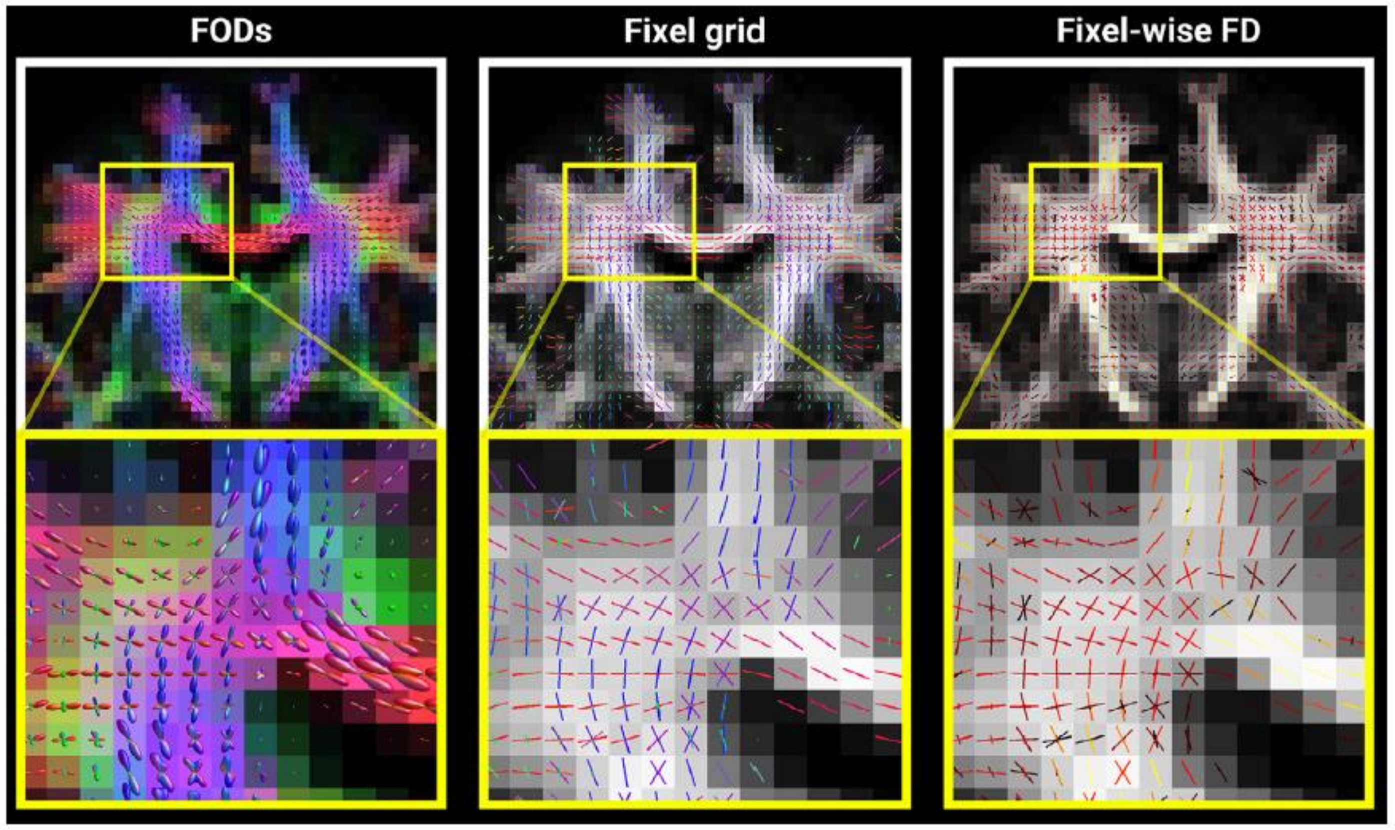
Outline

- Fixel-based analysis: background
- Why building a better software for fixel-wise statistics?
- Methods
- Demo of FixelArray package
- Memory profiling results
- Summary

Outline

- Fixel-based analysis: background
- Why building a better software for fixel-wise statistics?
- Methods
- Demo of FixelArray package
- Memory profiling results
- Summary

Fixel-based analysis: from voxel-wise to fixel-wise analysis



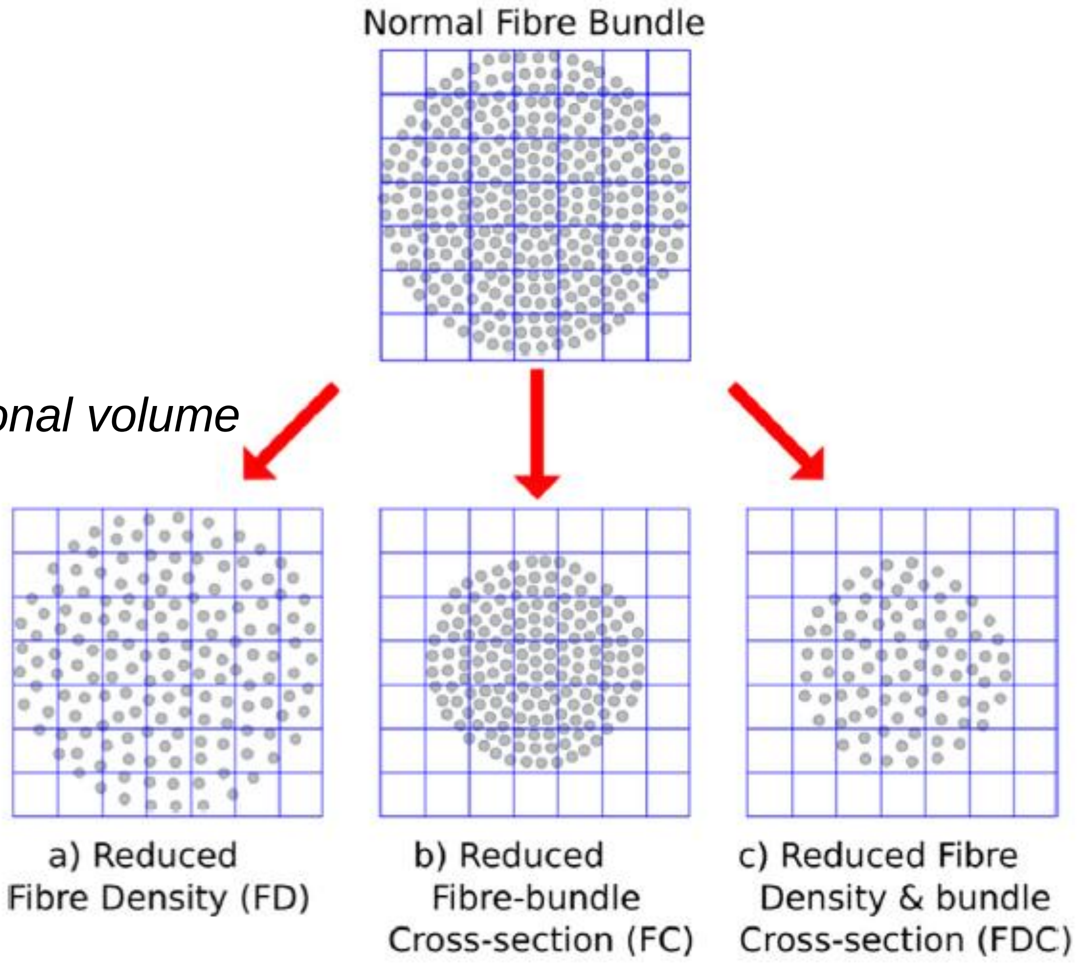
FOD, fiber orientation distribution
FD, fiber density

- "Fixel": an individual **<u>fi</u>**ber population within a vo**<u>xel</u>**

Dhollander et al., NeuroImage 2021
Raffelt et al., NeuroImage 2015

Fixel-wise metrics

Change to intra-axonal volume



Normal Fibre Bundle
a) Reduced Fibre Density (FD)
b) Reduced Fibre-bundle Cross-section (FC)
c) Reduced Fibre Density & bundle Cross-section (FDC)

Raffelt et al., NeuroImage 2017

Outline

- Fixel-based analysis: background
- Why building a better software for fixel-wise statistics?
- Methods
- Demo of FixelArray package
- Memory profiling results
- Summary

Target features: #1: Straightforward to use

MRtrix3

R

Linear regression

$$\begin{array}{ccc} \text{intercept} & \text{sex} & \text{age} \\ 1 & 1 & 21.75 \\ 1 & 1 & 20.50 \\ 1 & 0 & 20.50 \\ 1 & 1 & 20.92 \\ 1 & 0 & 20.92 \\ 1 & 0 & 21.50 \\ \ldots & & \end{array}$$

Design matrix

```
0  0  1
```

Contrast matrix

```
myModel <- lm(data, FD ~ age)
```

Generalized additive model (GAM)

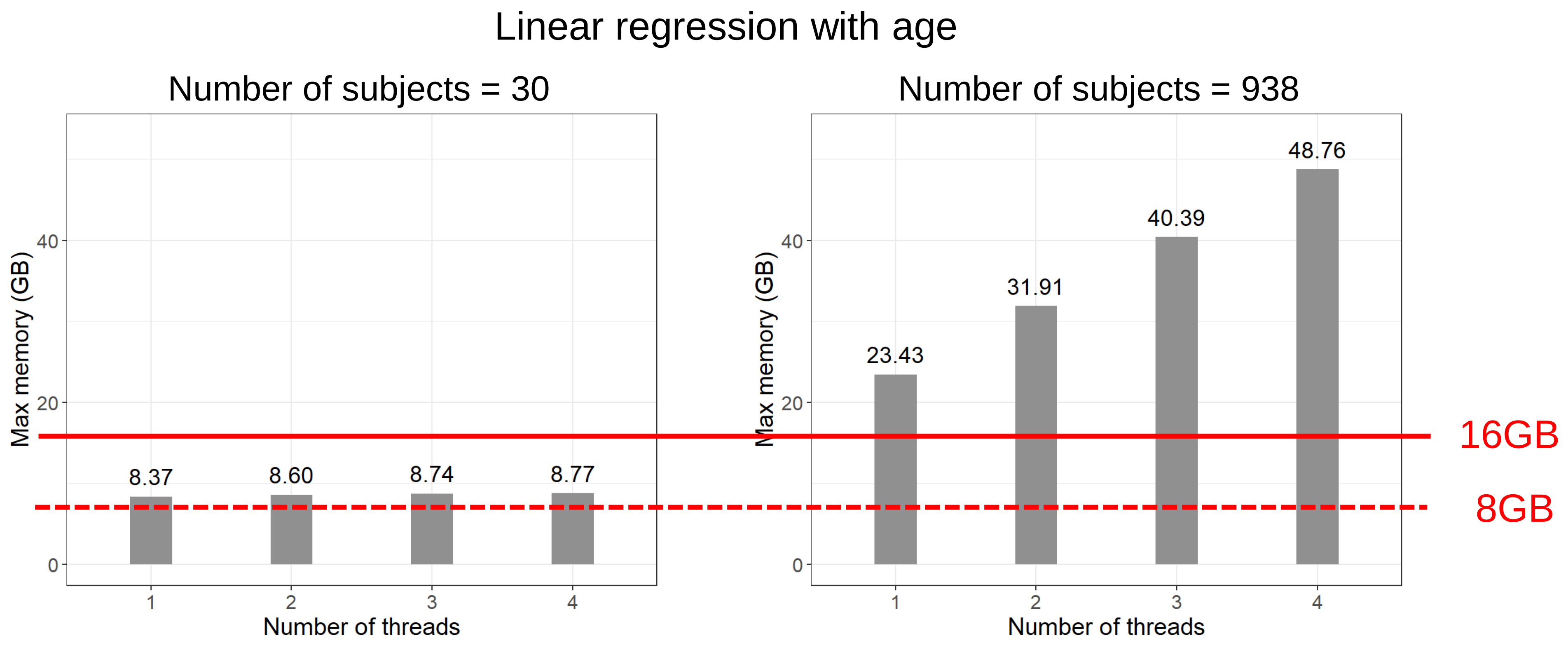
????

```
myModel <- gam(data, FD ~ s(age, k=3))
```

```
myModel <- gam(data, FD ~ s(age, k=3) + sex + motion)
```

Target features: #2: Low memory requirement

How much memory does MRtrix use?



How much memory does your laptop have?

Outline

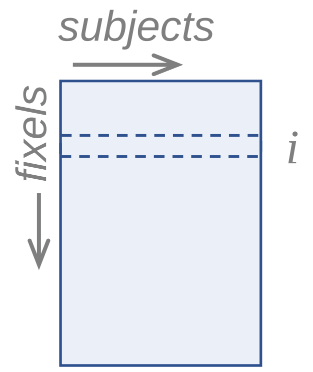
- Fixel-based analysis: background
- Why building a better software for fixel-wise statistics?
- Methods
- Demo of FixelArray package
- Memory profiling results
- Summary

HDF5 file (.h5) and DelayedArray

- The Hierarchical Data Format version 5 (**HDF5**), is an open source file format that supports large, complex, heterogeneous data.

An example .h5 file:

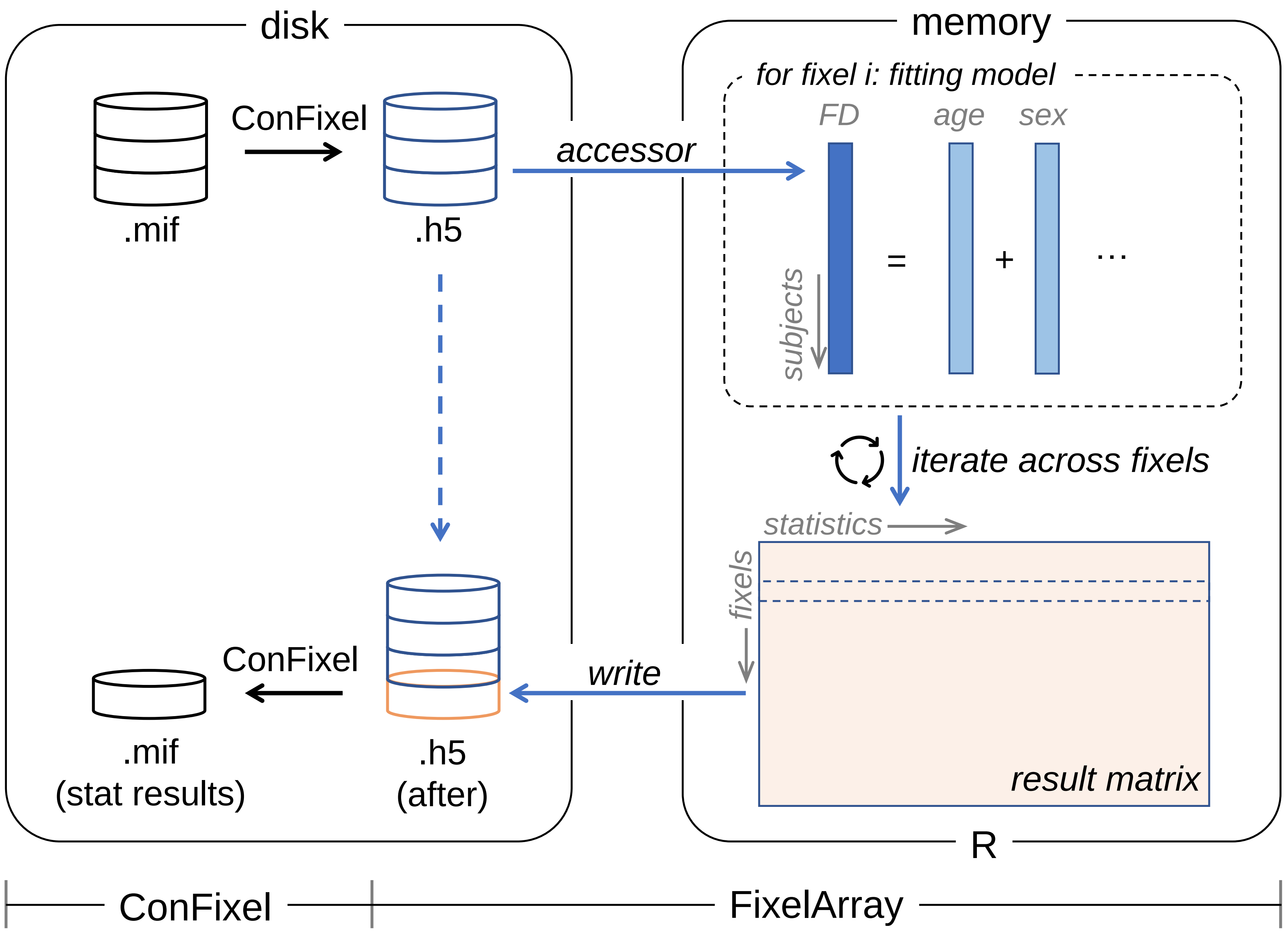
```
/fixels                          602229 x 5
/voxels                          433909 x 4
/scalars
  /FDC
     /ids                        1 x 938
     /values                     602229 x 938
/results
  /lm
    /results_matrix              602229 x 13
```



- **DelayedArray** is an R package that: wraps an array-like object (typically an on-disk object) in a DelayedArray object, thus allows one to perform common array operations on it without loading the object in memory.
    - The point is “***Delayed***”: Data is ***on disk***; Not to load it into (or realize it in) memory until you ***really*** need it

https://bioconductor.org/packages/release/bioc/html/DelayedArray.html
https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5

Workflows



disk
.mif
ConFixel
.h5
accessor
memory
for fixel i: fitting model
FD
age
sex
=
+
…
subjects
iterate across fixels
statistics
fixels
result matrix
write
ConFixel
.mif
(stat results)
.h5
(after)
R
ConFixel
FixelArray

Outline

- Fixel-based analysis: background
- Why building a better software for fixel-wise statistics?
- Methods
- Demo of FixelArray package
- Memory profiling results
- Summary

Demo: loading data as a FixelArray

```
> library(FixelArray)

> filename <- “ltn_FDC_n938.h5”
> fixelarray <- FixelArray(filename, scalar_types = “FDC”)
> fixelarray
```

```
FixelArray located at ltn_FDC_n938.h5

  Fixel data:        602229 fixels
  Voxel data:        433909 voxels
  Subjects:          938
  Scalars:           FDC
  Analyses:
```

Dataset:

- *PNC, Low threshold normal (LTN)*
- *N = 938 (excluded 3 subjects due to missing cognitive data)*
- *Age range: 8-23 years*

```
> object.size(fixelarray)
```

```
15584 bytes
```

On the disk:

ltn_FDC_n938.h5 2.5 GB

Demo

```
> scalars(fixelarray)[[“FDC”]]
```

```
<602229 x 938> matrix of class HDF5Matrix and type "double":
                [,1]       [,2]       [,3] ...     [,937]     [,938]
     [1,] 0.03784793 0.04200817 0.04807322   . 0.12814076 0.19082069
     [2,] 0.65316314 0.21953718 0.20505361   . 0.08036269 0.38789043
     [3,] 0.64559317 0.21095504 0.26770923   . 0.26575619 0.48313230
     [4,] 0.68784863 0.24132447 0.17531958   . 0.16427319 0.39082053
     [5,] 0.55170876 0.28086352 0.19687985   . 0.08512717 0.13990782
      ...          .          .          .   .          .          .
[602225,] 0.34975991 0.22181121 0.12327279   . 0.27788433 0.16905732
[602226,] 0.38397413 0.35014871 0.07027728   . 0.24905141 0.48857829
[602227,] 0.42077079 0.16245778 0.25115448   . 0.12555204 0.14311361
[602228,] 0.29952115 0.00000000 0.00000000   . 0.00000000 0.00000000
[602229,] 0.61790061 0.13066383 0.15461987   . 0.08716168 0.11360963
```

Demo

```
> phenotypes <- read.csv(“ltn_n938.csv”)
> head(phenotypes)
```

```
 subjID      Age
1 subj1 21.75000
2 subj2 20.50000
3 subj3 20.50000
4 subj4 20.91667
5 subj5 20.91667
6 subj6 21.50000
```

Demo: calling FixelArray.lm()

```
> lm.outputs <- FixelArray.lm (FDC~Age, fixelarray, phenotypes, “FDC”,
                               fixel.subset = 1:602229,
                               pbar = TRUE, n_cores = 4)
```

```
subset: default
weights: default
na.action: default
method: default
model: default
x: default
y: default
qr: default
singular.ok: default
contrasts: default
offset: default
Fitting fixel-wise linear models for FDC
initiating....
looping across fixels....
  |==================================                                | 60%, ETA 36:00
```

Options in R’s lm()

Demo: output of FixelArray.lm() and writing it to .h5 file

```
> head(lm.outputs)
```

```
  fixel_id Intercept.estimate Age.estimate Intercept.statistic Age.statistic Intercept.p.value
1        0         0.08920253  0.007580290            3.578825     4.7729306      3.628574e-04
2        1         0.22588093  0.002721890           10.438315     1.9740462      3.326395e-24
3        2         0.26025164  0.003233899           10.590140     2.0652405      7.878374e-25
4        3         0.22657694  0.003489831           10.294376     2.4884261      1.283977e-23
5        4         0.25808970  0.000586250           11.087017     0.3952418      6.325773e-27
6        5         0.27514479  0.001266609           10.506548     0.7590620      1.744655e-24
   Age.p.value model.adj.r.squared model.p.value
1 2.105749e-06        0.0227172518  2.105749e-06
2 4.867018e-02        0.0030821023  4.867018e-02
3 3.917511e-02        0.0034726567  3.917511e-02
4 1.300392e-02        0.0055108330  1.300392e-02
5 6.927545e-01       -0.0009013281  6.927545e-01
6 4.480065e-01       -0.0004525257  4.480065e-01
```

Above are default statistics

Demo: output of FixelArray.lm() – choose the output statistics you want

- Statistics for input variable:
  - Coefficient
  - Estimation’s standard error
  - t statistic
  - p-value
    - Raw, FDR, Bonferroni, etc
- Statistics for the model:
  - R^2
  - Adjusted R^2
  - F statistic
  - p-value
    - Raw, FDR, Bonferroni, etc
  - Degrees of freedom
  - Number of observations used
  - AIC, BIC
  - …

```
var.terms = c("estimate", "statistic", "p.value")
```

```
correct.p.value.terms = “fdr"
```

```
var.model = c("adj.r.squared", "p.value")
```

```
correct.p.value.model = c(“fdr“,”bonferroni”)
```

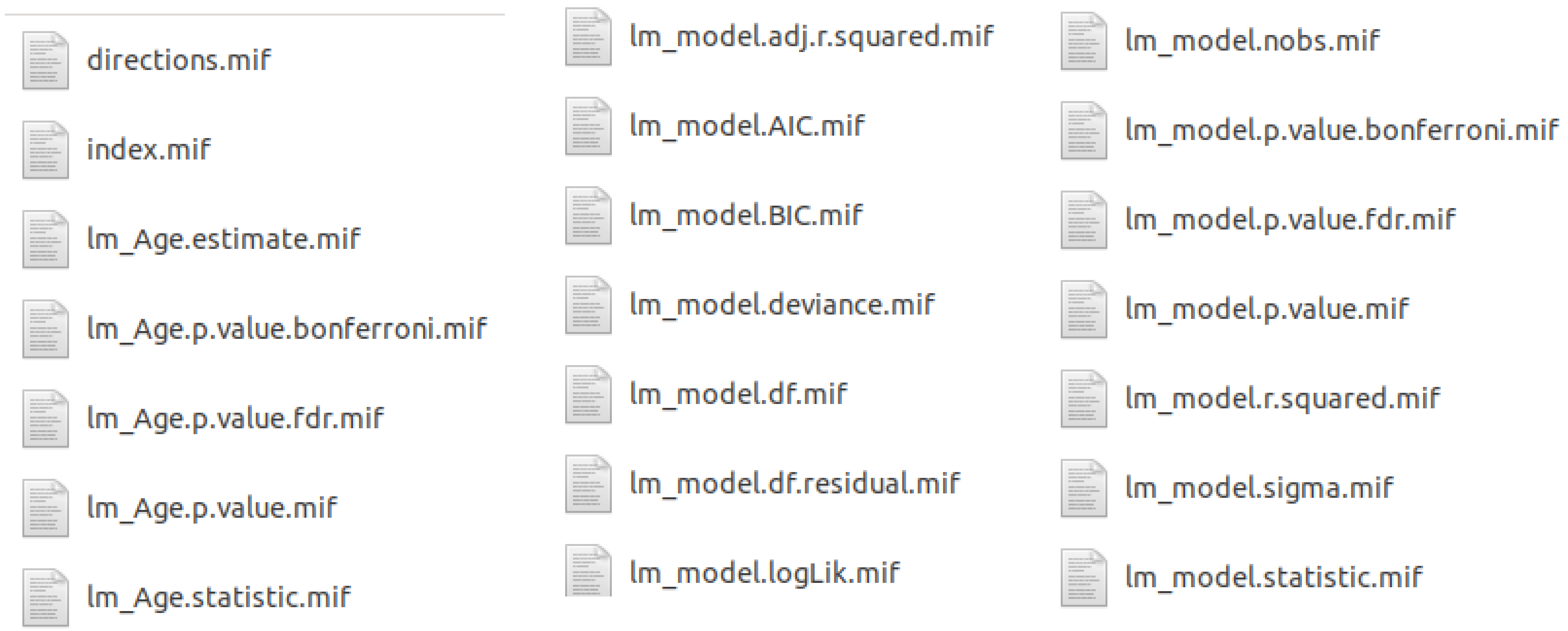
```
full.outputs = TRUE   # I want all outputs!
```

Demo: writing it to .h5 file

```
> writeResults(filename, df.output = lm.outputs, analysis_name=“lm”)
```

No more than 10 lines of code to get fixel-wise statistics done!

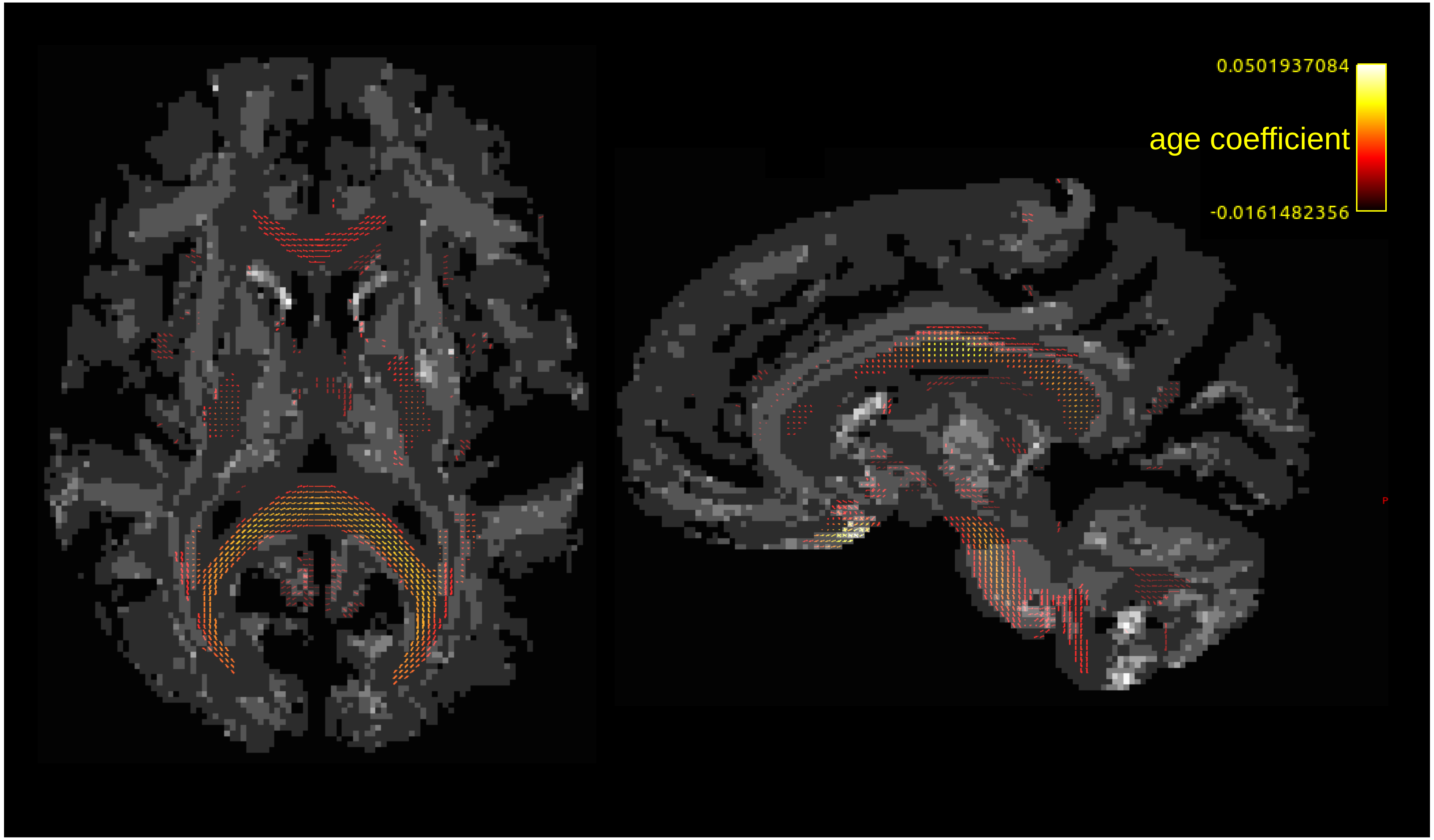
Next step is to use ConFixel to convert to .mif format and view in mrview!



```
directions.mif                 lm_model.adj.r.squared.mif   lm_model.nobs.mif
index.mif                      lm_model.AIC.mif             lm_model.p.value.bonferroni.mif
lm_Age.estimate.mif            lm_model.BIC.mif             lm_model.p.value.fdr.mif
lm_Age.p.value.bonferroni.mif  lm_model.deviance.mif        lm_model.p.value.mif
lm_Age.p.value.fdr.mif         lm_model.df.mif              lm_model.r.squared.mif
lm_Age.p.value.mif             lm_model.df.residual.mif     lm_model.sigma.mif
lm_Age.statistic.mif           lm_model.logLik.mif          lm_model.statistic.mif
```

and more …
(if requested)

Demo: result images shown in mrview



FixelArray.lm() result: FDC ~ Age;
Number of subject = 938; thresholded by model’s p-value (Bonferroni) < 0.05

Outline

- Fixel-based analysis: background
- Why building a better software for fixel-wise statistics?
- Methods
- Demo of FixelArray package
- **Memory profiling results**
- Summary

Multi-core processing

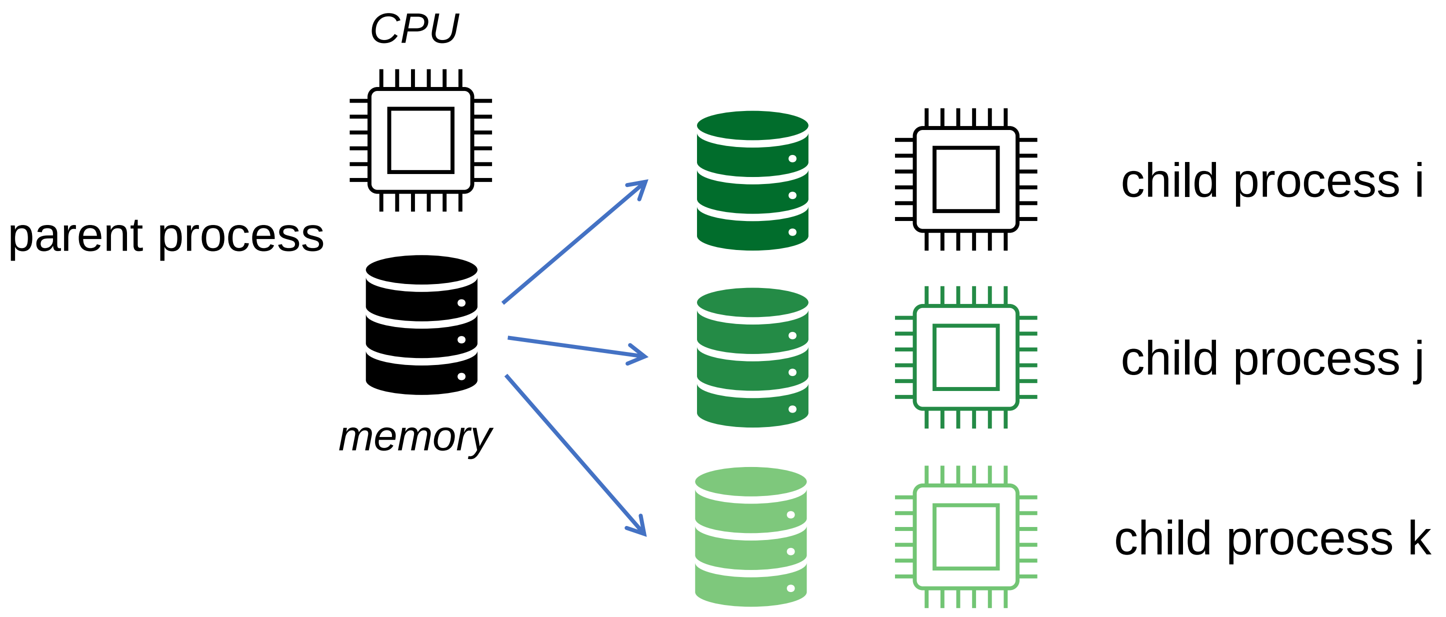
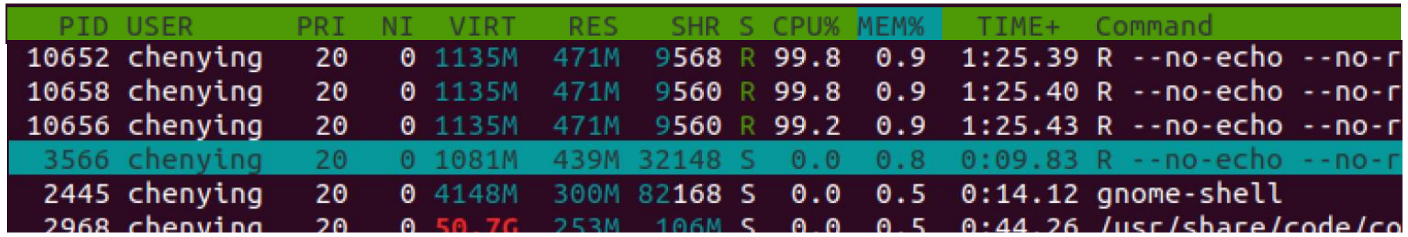


Figure adapted from : https://www.blasbenito.com/post/02_parallelizing_loops_with_r/

```
$ htop
```

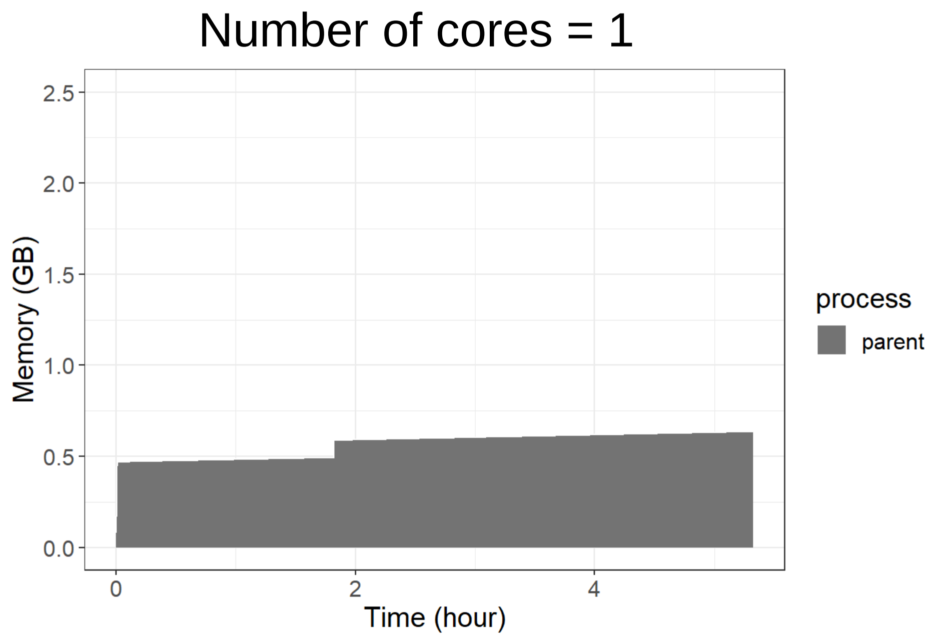


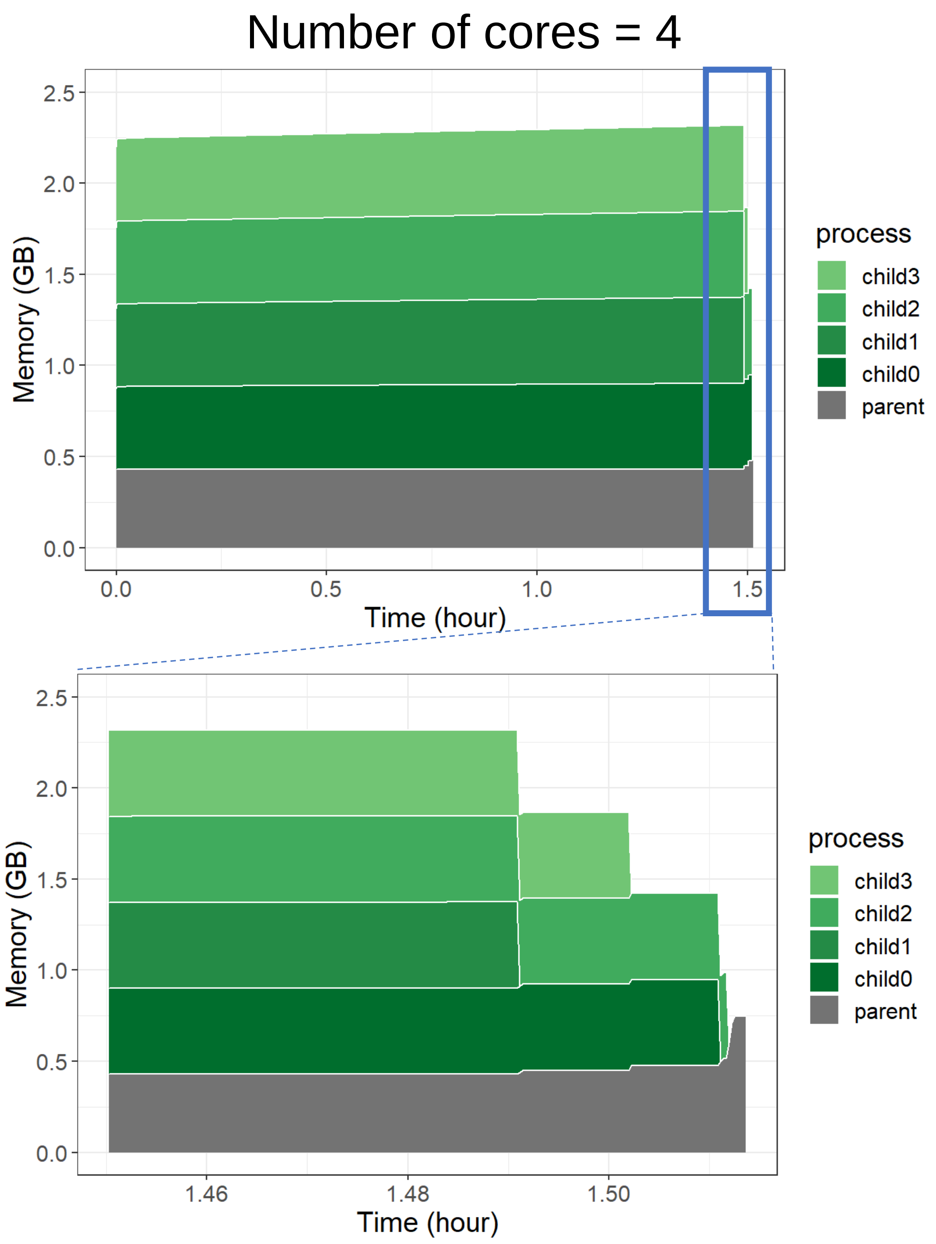
```
  PID USER       PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
10652 chenying    20   0 1135M  471M  9568 R 99.8  0.9  1:25.39 R --no-echo --no-r
10658 chenying    20   0 1135M  471M  9560 R 99.8  0.9  1:25.40 R --no-echo --no-r
10656 chenying    20   0 1135M  471M  9560 R 99.2  0.9  1:25.43 R --no-echo --no-r
 3566 chenying    20   0 1081M  439M 32148 S  0.0  0.8  0:09.83 R --no-echo --no-r
 2445 chenying    20   0 4148M  300M 82168 S  0.0  0.5  0:14.12 gnome-shell
 2968 chenying    20   0 50.7G  253M  106M S  0.0  0.5  0:44.26 /usr/share/code/co
```

child processes {

parent process →

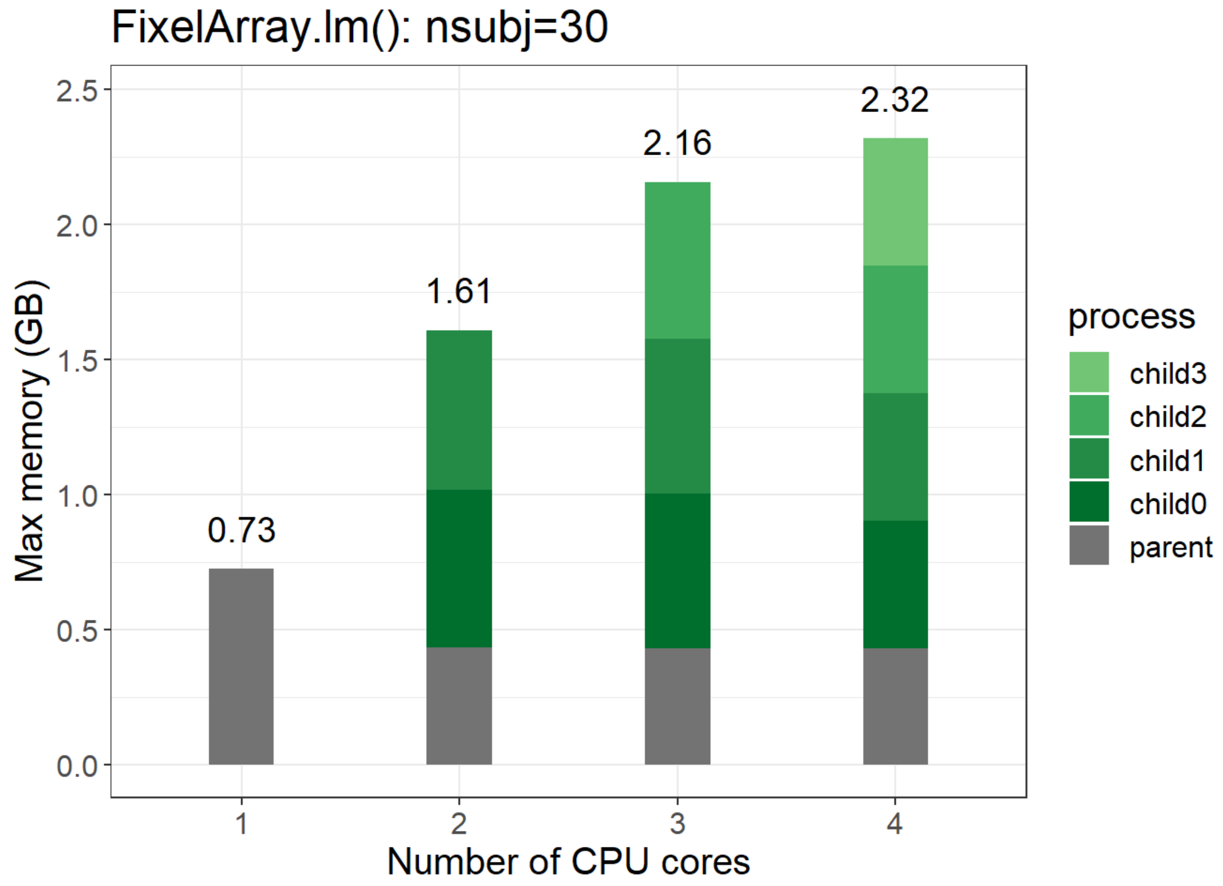
How does FixelArray’s memory change across time?

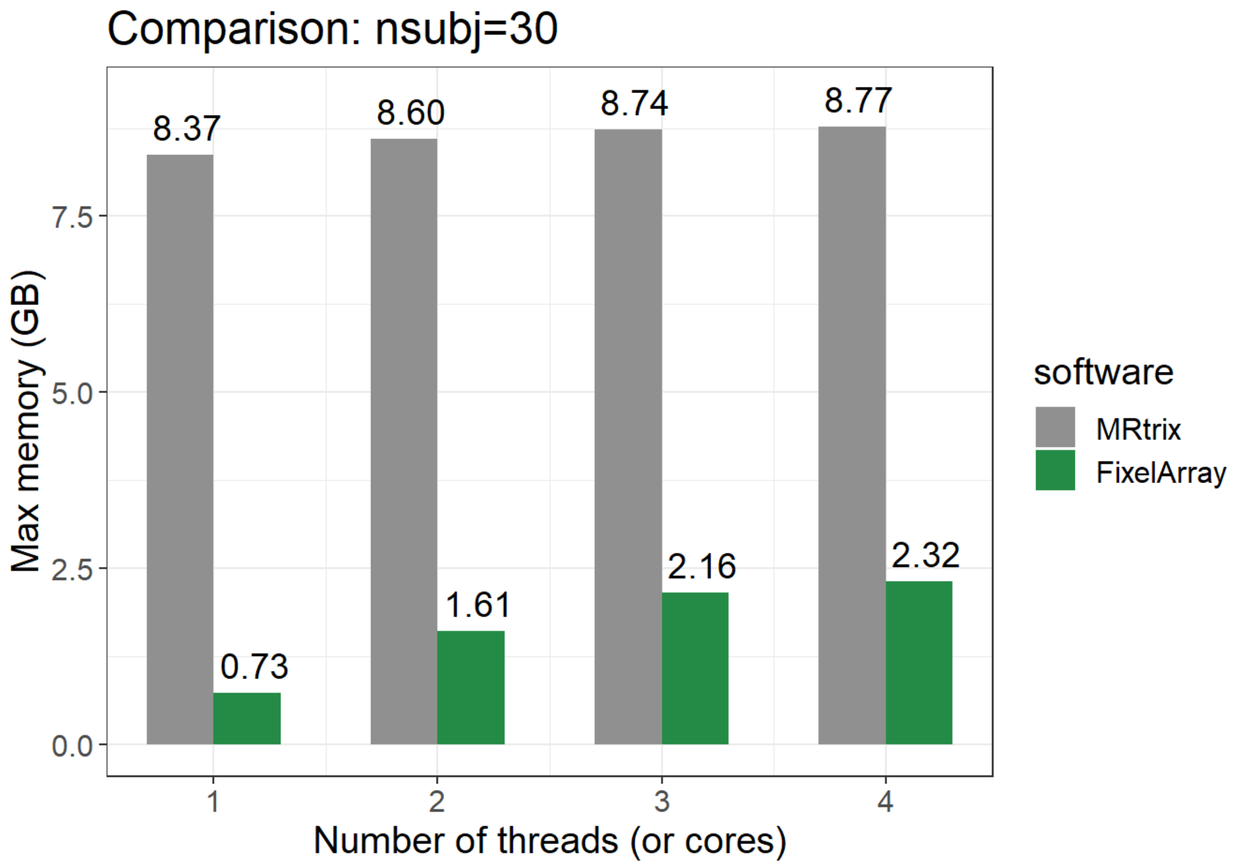




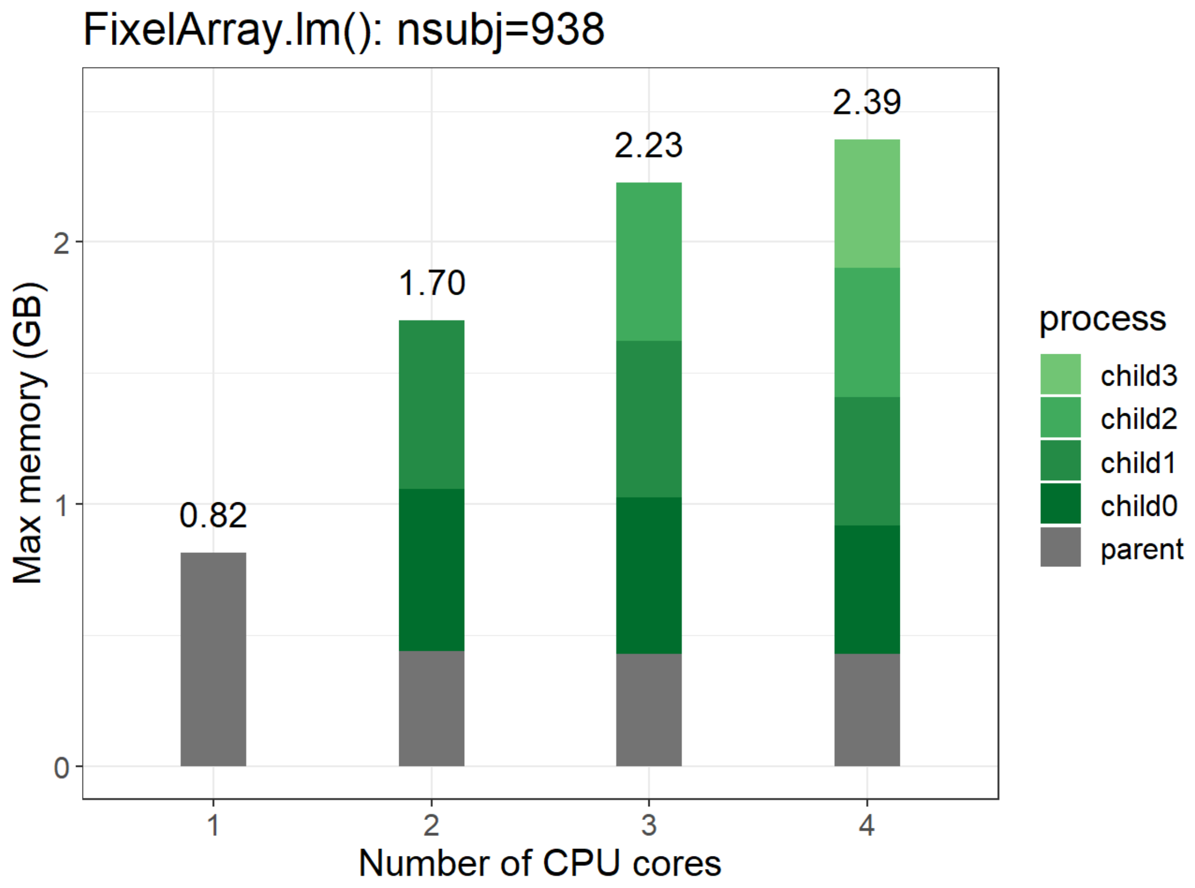
FixelArray.lm()
Number of subjects = 30

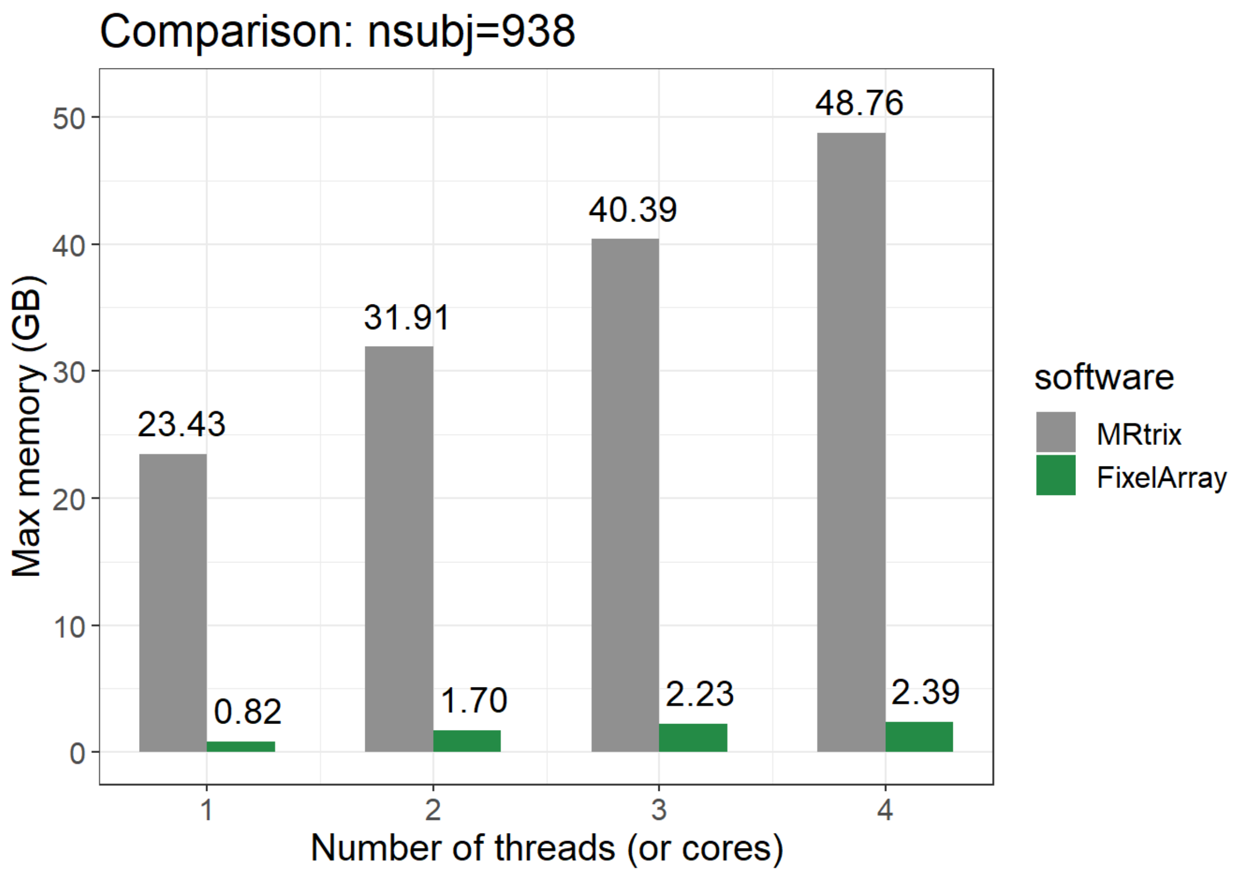
Different number of cores (threads)



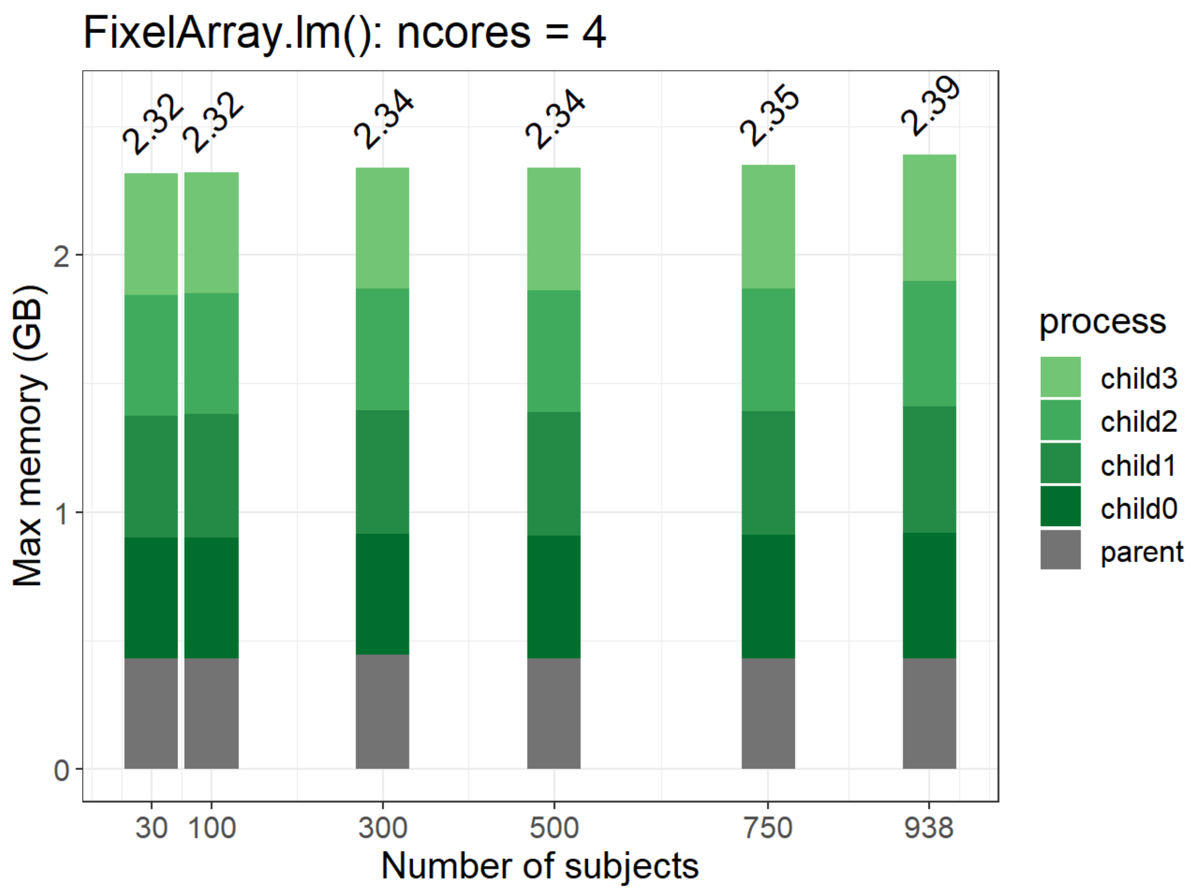


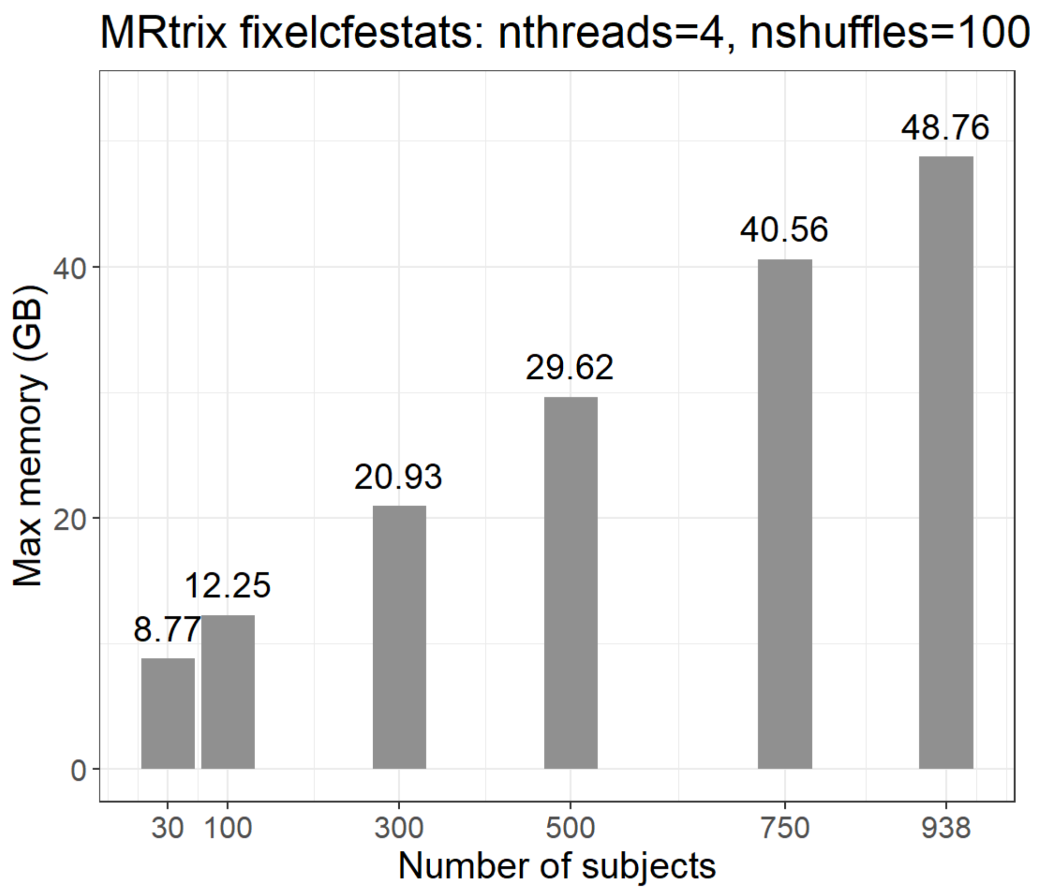
Different number of cores (threads)



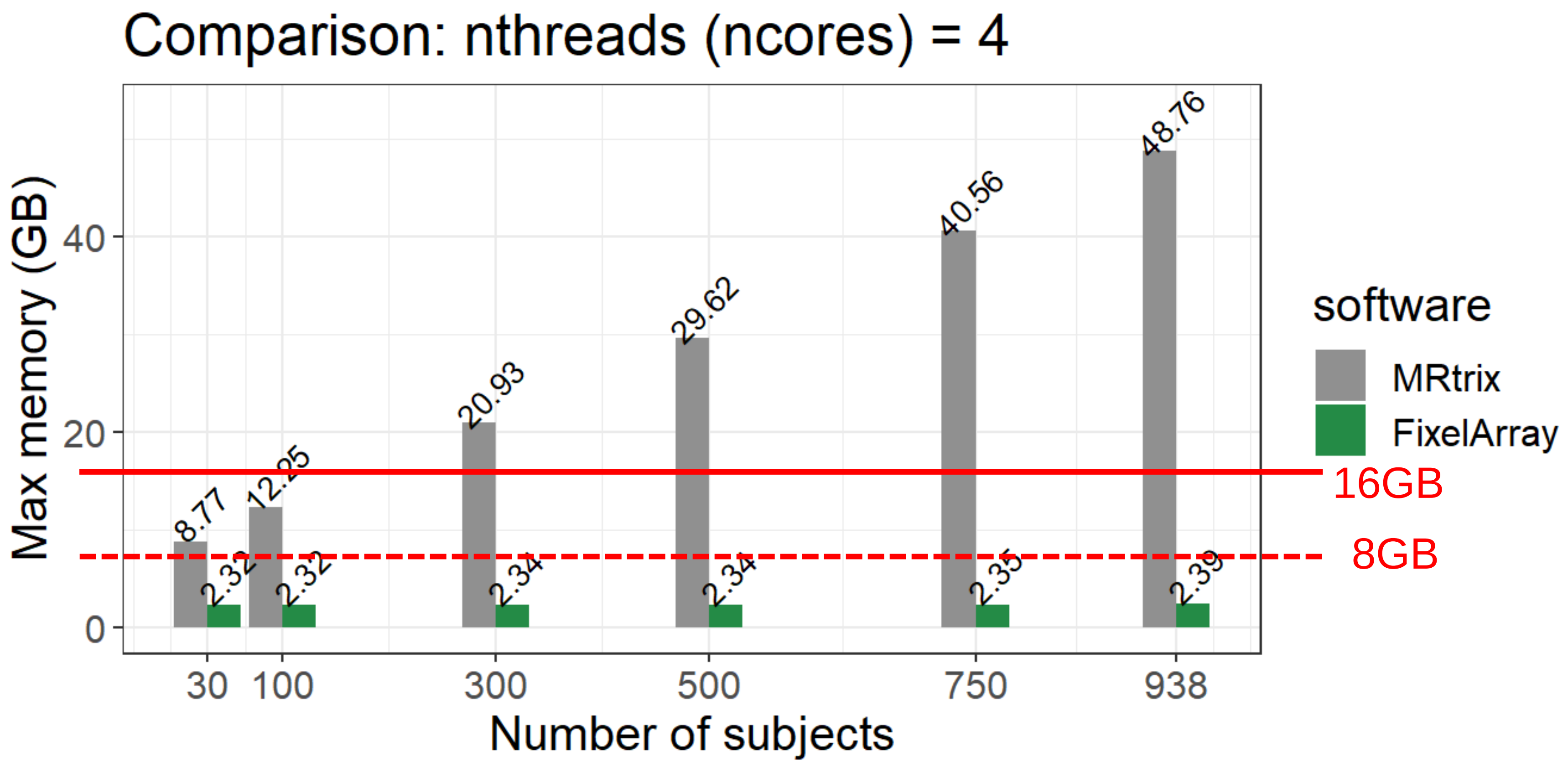


Different number of subjects



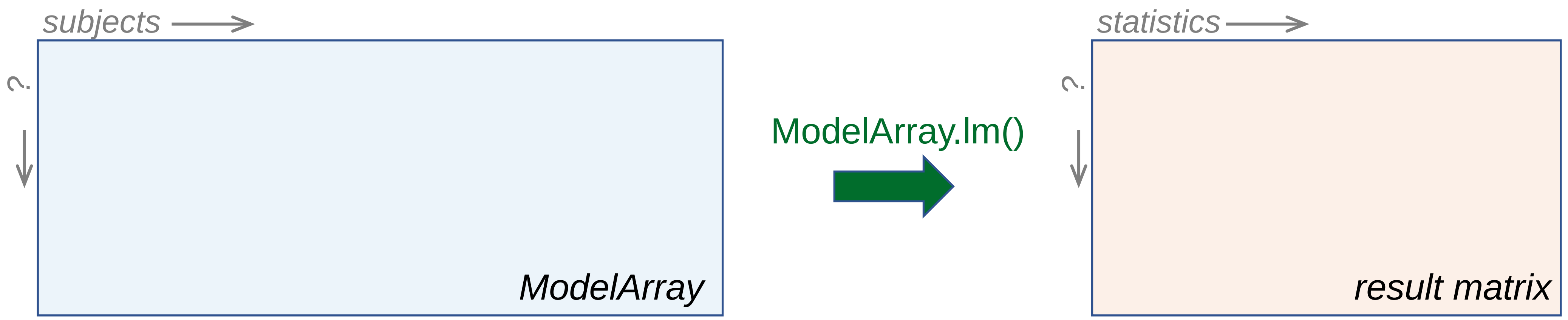


Different number of subjects



What's next?

- FixelArray.gam()
- A generic version? "ModelArray"
    - For any matrix of scalars: voxel-wise, surface vertex-wise, etc



Time required?

4-core (or thread) computing:

- Number of subjects = 30:
  - FixelArray: <2h
  - MRtrix: ~half day (5000 shuffles)

- Number of subjects = 938:
  - FixelArray: <2h
  - MRtrix: several days (5000 shuffles)

On a laptop of Intel Xeon CPU, base speed = 2.81 GHz, best battery life (slowest mode)

Summary

- FixelArray is an R package for fixel-based statistics, that is
    - straightforward to use
    - memory efficient
- GAM and generic “ModelArray” are on the way….

Acknowledgement

- Tinashe Tapera, Matt Cieslak, the informatic team !!
- Josiane Bourque, Valerie Sydnor
- Ted Satterthwaite !!!

Thank you !!!