

Imaging Packages in R

Neuroconductor and Imaging Packages in R

Some packages we will use

- `oro.nifti`
 - `nifti`
 -
- `neurobase` `oro.nifti`

```
library(oro.nifti)
library(neurobase)
```

Reading in NIfTI images: assignment

readnii

neurobase

nifti

R

t1

```
t1 = readnii("training01_01_t1.nii.gz")
```

t1

```
class(t1)
```

```
[1] "nifti"  
attr(,"package")  
[1] "oro.nifti"
```

nifti images

```
print(t1)
```

```
t1
```

```
NIfTI-1 format
```

```
Type           : nifti
Data Type       : 4 (INT16)
Bits per Pixel  : 16
Slice Code      : 0 (Unknown)
Intent Code     : 0 (None)
Qform Code      : 2 (Aligned Anat)
Sform Code      : 1 (Scanner Anat)
Dimension       : 408 x 512 x 152
Pixel Dimension : 0.43 x 0.43 x 0.82
Voxel Units     : mm
Time Units      : Unknown
```

Operations with `nifti` objects

`img + 2`

`img1 + img2`

• `> >= < <= == !=`

• `! & |`

• `+ - * / ^`

• `log abs sqrt`

```
t1 + t1 + 2 # still a nifti
```

NIfTI-1 format

Type	: nifti
Data Type	: 4 (INT16)
Bits per Pixel	: 16
Slice Code	: 0 (Unknown)
Intent Code	: 0 (None)
Qform Code	: 2 (Aligned_Anat)
Sform Code	: 1 (Scanner_Anat)
Dimension	: 408 x 512 x 152
Pixel Dimension	: 0.43 x 0.43 x 0.82
Voxel Units	: mm
Time Units	: Unknown

Working with **nifti** objects

```
class(t1 > 400) # still a nifti
```

```
[1] "nifti"  
attr(,"package")  
[1] "oro.nifti"
```

```
head(t1 > 400) # values are now logical vs. numeric
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
t1 > 400
```

Subsetting with `nifti` objects: like arrays

`t1`

```
t1[5, 4, 3]
```

```
[1] 0
```

```
t1[5, 4, ] # returns a vector of numbers (1-d)
```

```
t1[, 4, ] # returns a 2-d matrix
```

```
t1[1, , ] # returns a 2-d matrix
```

-
- `t1` `head`

```
head(t1[ t1 > 400 ]) # produces a vector of numbers
```

```
[1] 402 412 435 448 453 430
```


which with nifti objects

which

arr.ind = TRUE

```
head(which(t1 > 400, arr.ind = TRUE))
```

	dim1	dim2	dim3
[1,]	180	258	1
[2,]	175	259	1
[3,]	176	259	1
[4,]	177	259	1
[5,]	178	259	1
[6,]	179	259	1

```
head(which(t1 > 400, arr.ind = FALSE))
```

```
[1] 105036 105439 105440 105441 105442 105443
```

Working with `nifti` objects: reassignment

```
t1_copy = t1  
t1_copy[ t1_copy > 400 ] = 400 # replaced these values!  
max(t1_copy) # should be 400
```

```
[1] 400
```

```
max(t1)
```

```
[1] 1691
```

	t1_copy	t1
t1_copy	t1	

Writing Images out

t1_copy

```
writenii(nim = t1_copy,  
        filename = "training01_t1_under400.nii.gz")  
file.exists("training01_t1_under400.nii.gz")
```

```
[1] TRUE
```

```
file.exists      TRUE
```

```
•           all all(file.exists(VECTOR_OF_FILES))
```

Vectorizing a `nifti`

`nifti`

`vector`

`c()`

```
vals = c(t1)
class(vals)
```

```
[1] "numeric"
```

```
array(c(t1), dim = dim(t1))
```

`t1`

`data.frame`

```
df = data.frame(t1 = c(t1), mask = c(t1 > 400)); head(df)
```

```
   t1  mask
1   0 FALSE
2   0 FALSE
3   0 FALSE
4   0 FALSE
5   0 FALSE
6   0 FALSE
```

File helpers - for constructing filenames

paste

paste0

```
file.path(directory, filename)
/
```

```
directory    filename
```

```
c(paste("img", ".nii.gz"), paste0("img", ".nii.gz"))
```

```
[1] "img .nii.gz" "img.nii.gz"
```

```
x = file.path("output_directory", paste0("img", ".nii.gz")); print(x)
```

```
[1] "output_directory/img.nii.gz"
```

nii.stub

bn = TRUE

```
c(nii.stub(x), nii.stub(x, bn = TRUE))
```

```
[1] "output_directory/img" "img"
```

Main Packages we will use

- `oro.nifti`
- `neurobase` `oro.nifti`
- `fslr`
 -
- `ANTsR`
 -
- `extrantsr` `ANTsR` `oro.nifti`
- `ms.lesion`
 -

Conclusions

-
- nifti
 -
 -
- readnii writenii nifti
- -

Website
