The Pediatric Template of Brain Perfusion: Resting state functional mri processing

Brian B. Avants et al.

PICSL

2016-08-12

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Overview

This is a compilable document with source code located here:

https://github.com/stnava/ANTsTutorial

To get this source, do:

```
git clone http://github.com/stnava/ANTsTutorial.git
```

It is expected that you will compile and, after downloading data, run this:

```
rmarkdown::render("src/PTBP_rsfmri.Rmd")
```

from within the cloned `ANTsTutorial` directory. The document needs the complete PTBP subject data discussed below. It depends on *R*, *rmarkdown* and *ANTsR* primarily.

Herein, links are in this color.

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Overview

The Pediatric Template of Brain Perfusion (PTBP) at figshare.

- Free multiple modality MRI data with demographics and psychometrics

Overview

The Pediatric Template of Brain Perfusion (PTBP) at figshare.

- Free multiple modality MRI data with demographics and psychometrics
- Collected to aid as a reference for normal variability in structure and function during adolescence

Overview

The Pediatric Template of Brain Perfusion (PTBP) at figshare.

- Free multiple modality MRI data with demographics and psychometrics
- Collected to aid as a reference for normal variability in structure and function during adolescence
- The data is accompanied by an organized csv file

Overview

The Pediatric Template of Brain Perfusion (PTBP) at figshare.

- Free multiple modality MRI data with demographics and psychometrics
- Collected to aid as a reference for normal variability in structure and function during adolescence
- The data is accompanied by an organized csv file
- The full data is available at figshare

Overview

The Pediatric Template of Brain Perfusion (PTBP) at figshare.

- Free multiple modality MRI data with demographics and psychometrics
- Collected to aid as a reference for normal variability in structure and function during adolescence
- The data is accompanied by an organized csv file
- The full data is available at figshare
- Here we use a single subject from this dataset.

Overview

The Pediatric Template of Brain Perfusion (PTBP) at figshare.

- Free multiple modality MRI data with demographics and psychometrics
- Collected to aid as a reference for normal variability in structure and function during adolescence
- The data is accompanied by an organized csv file
- The full data is available at figshare
- Here we use a single subject from this dataset.
- There is also a template contained in the download.

Download the neurobattery data

From within the ANTsTutorial directory:

```
git clone http://github.com/jeffduda/NeuroBattery.git
```

This will give you both raw and processed output for a single multiple modality subject.

FIXME: need to actually run to get output . . . that would take too long.

We test (occasionally) against this reference output to monitor stability of our processing.

Resting state fMRI

We present *basic* processing strategies here:

- Motion correction

Resting state fMRI

We present *basic* processing strategies here:

- Motion correction
- Mapping to subject-space T1

Resting state fMRI

We present *basic* processing strategies here:

- Motion correction
- Mapping to subject-space T1
- Mapping to a T1 group template

Resting state fMRI

We present *basic* processing strategies here:

- Motion correction
- Mapping to subject-space T1
- Mapping to a T1 group template
- Data-driven nuisance modeling

Resting state fMRI

We present *basic* processing strategies here:

- Motion correction
- Mapping to subject-space T1
- Mapping to a T1 group template
- Data-driven nuisance modeling
- Network metrics and visualization

Resting state fMRI

We present *basic* processing strategies here:

- Motion correction
- Mapping to subject-space T1
- Mapping to a T1 group template
- Data-driven nuisance modeling
- Network metrics and visualization
- **many of these strategies are reused for DWI and ASL**

Motion correction

We do more or less the same thing for any time series modality.

```
fmri = antsImageRead( fmrifn )
amc = antsMotionCalculation(fmri,moreaccurate=0)
```

- Will motion correct with affine map

Motion correction

We do more or less the same thing for any time series modality.

```
fmri = antsImageRead( fmrifn )
amc = antsMotionCalculation(fmri,moreaccurate=0)
```

- Will motion correct with affine map
- Will produce a mask and motion parameters

Motion correction

We do more or less the same thing for any time series modality.

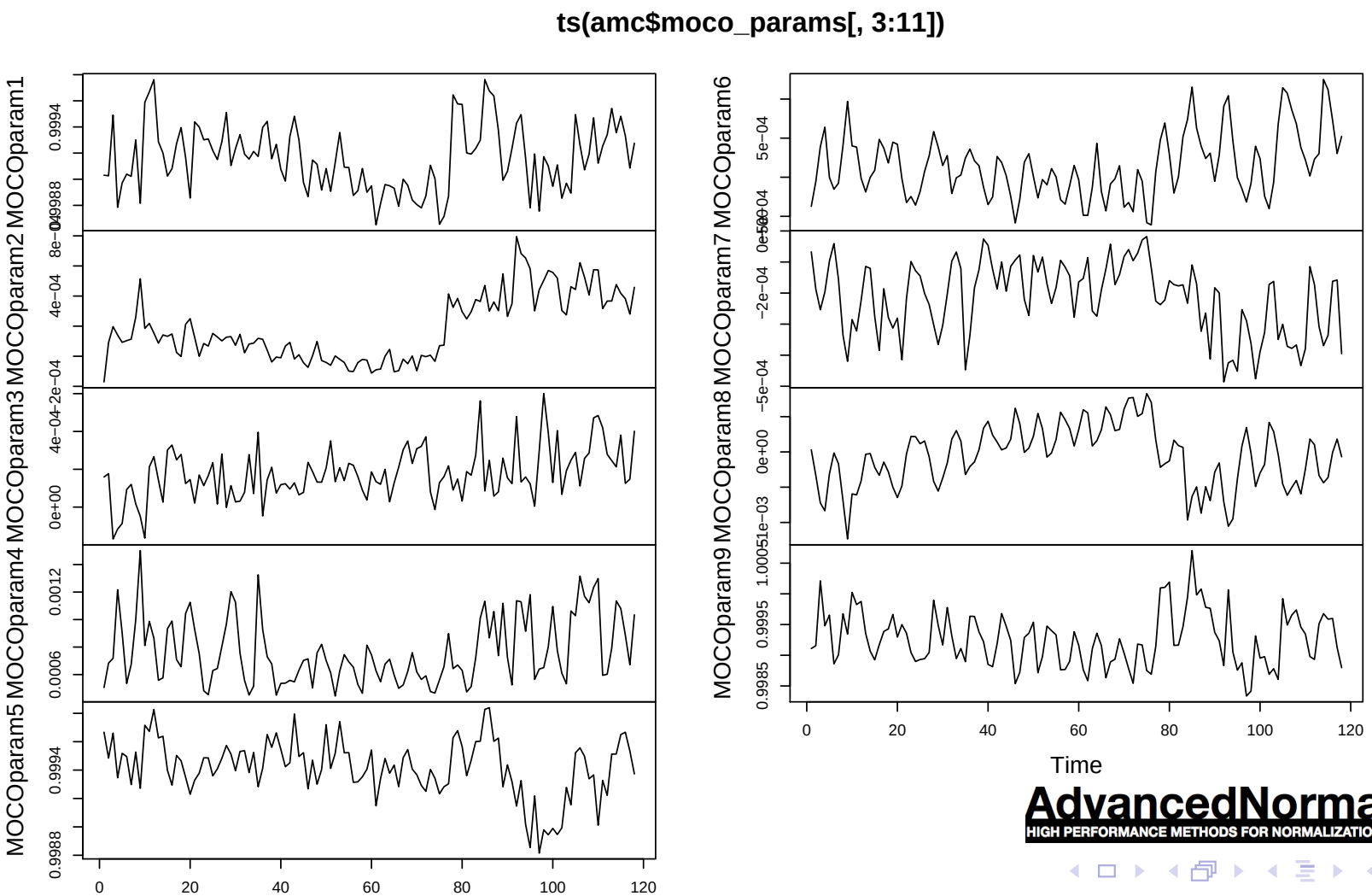
```
fmri = antsImageRead( fmrifn )
amc = antsMotionCalculation(fmri,moreaccurate=0)
```

- Will motion correct with affine map
- Will produce a mask and motion parameters
- “moco_img” “moco_params” “moco_avg_img” “moco_mask” “dvars”

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Visualize motion parameters: Matrix

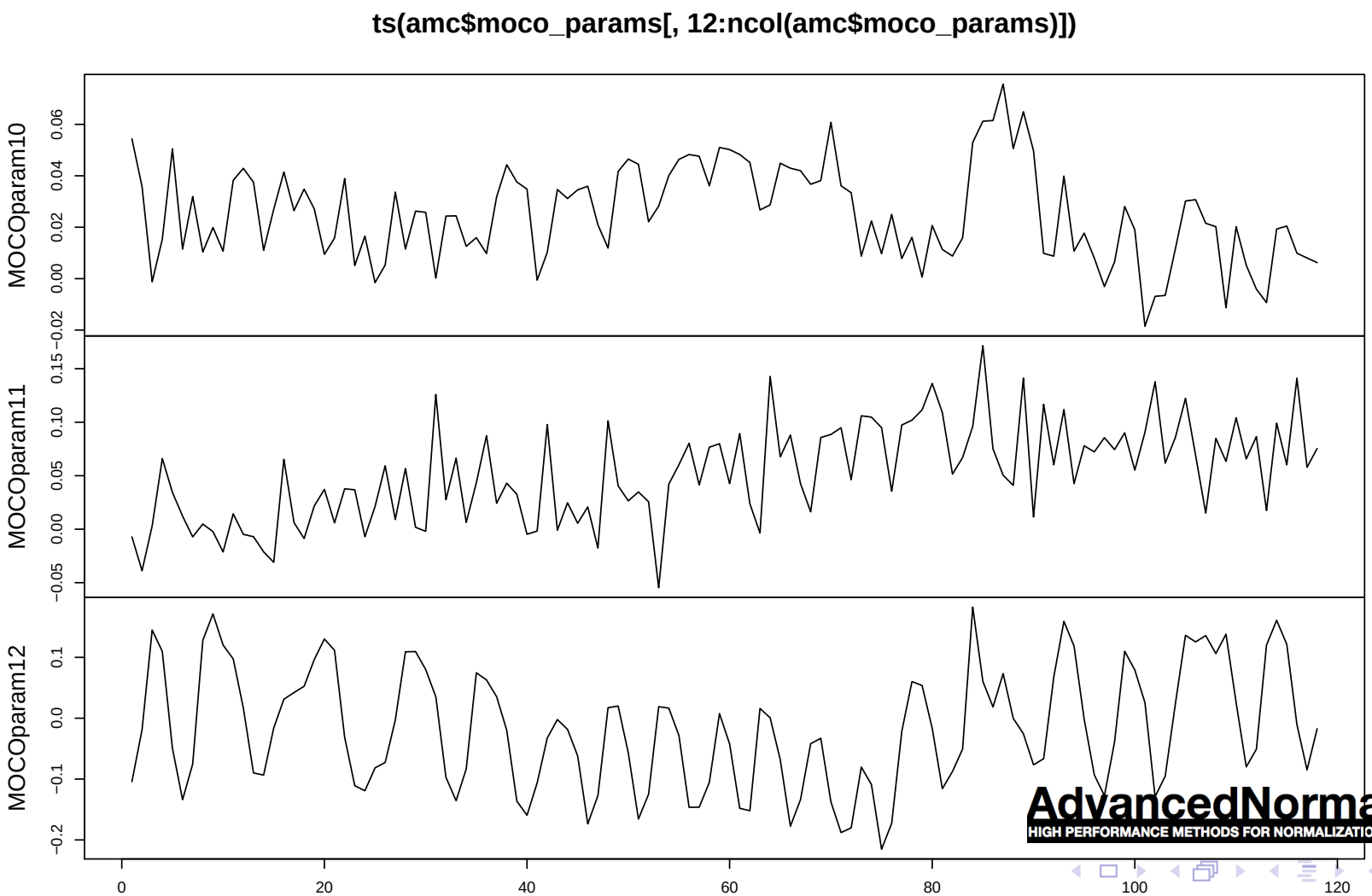
```
plot( ts( amc$moco_params[,3:11] ) )
```



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Visualize motion parameters: Translation

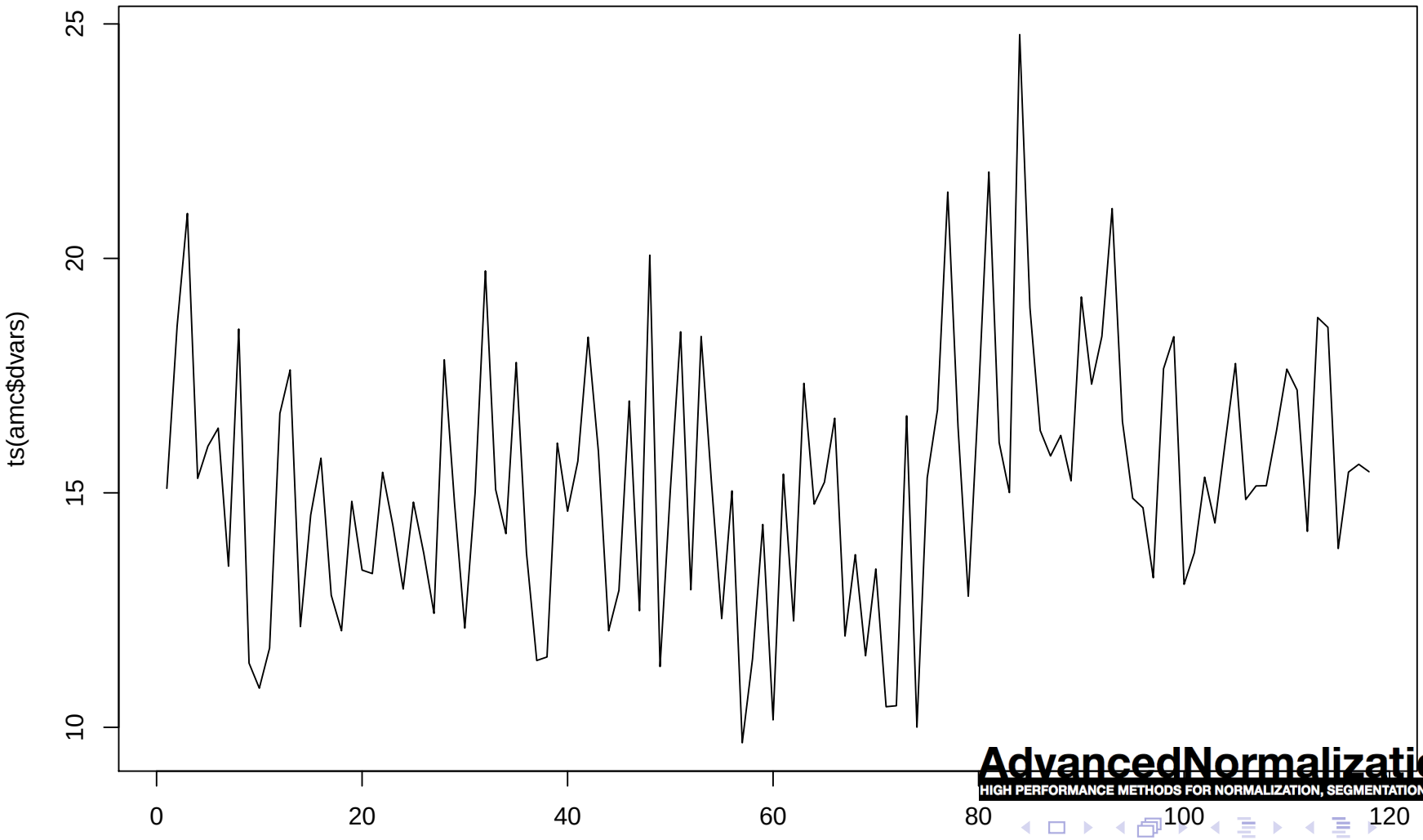
```
plot( ts( amc$moco_params[,12:ncol(amc$moco_params)] ) )
```



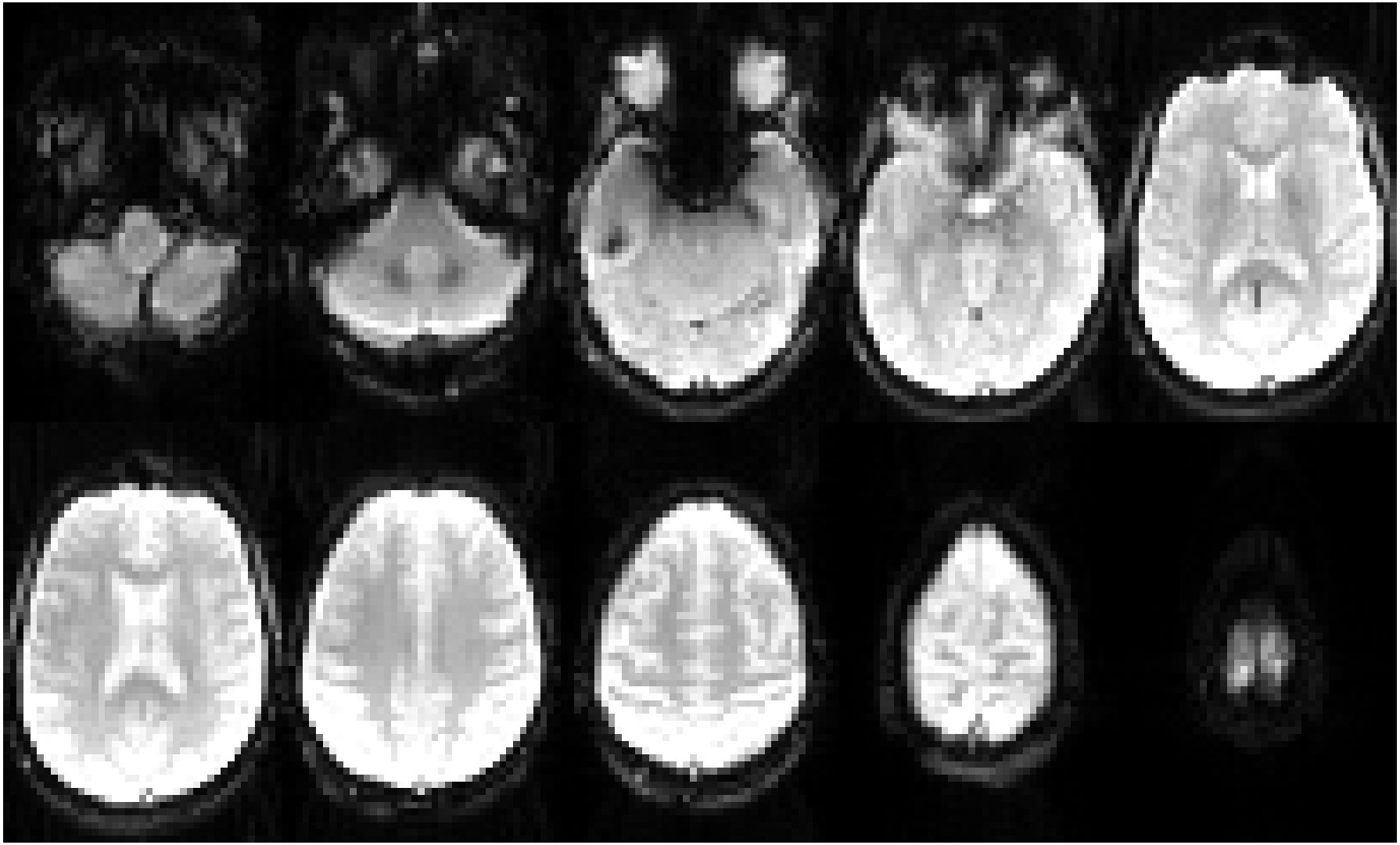
AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Visualize nuisance parameters: DVARS

```
plot( ts( amc$dvars ) )
```



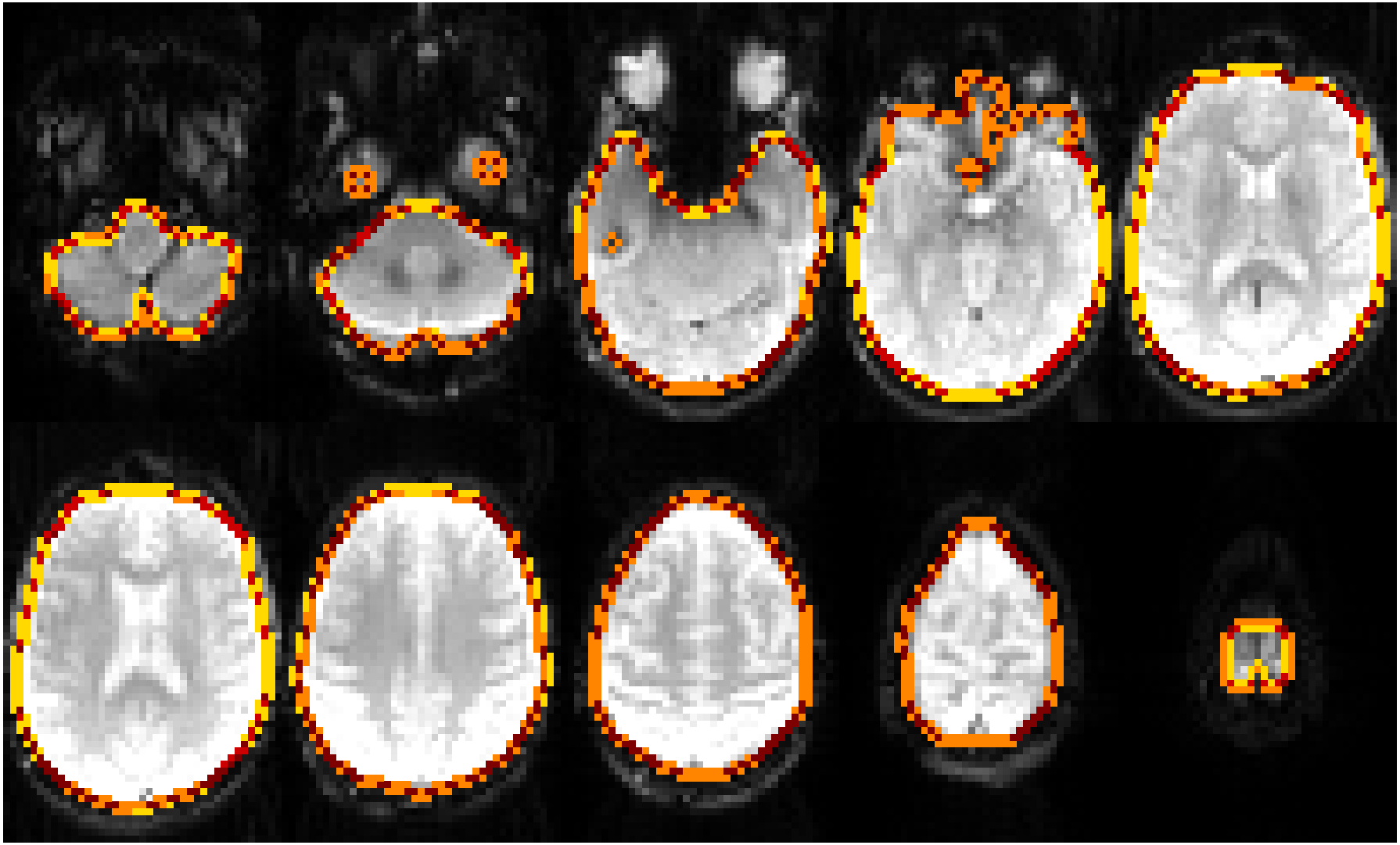
Look at the calculated average



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

NULL

Look at the calculated mask (gradient image)



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

NULL

Mapping to subject-space T1

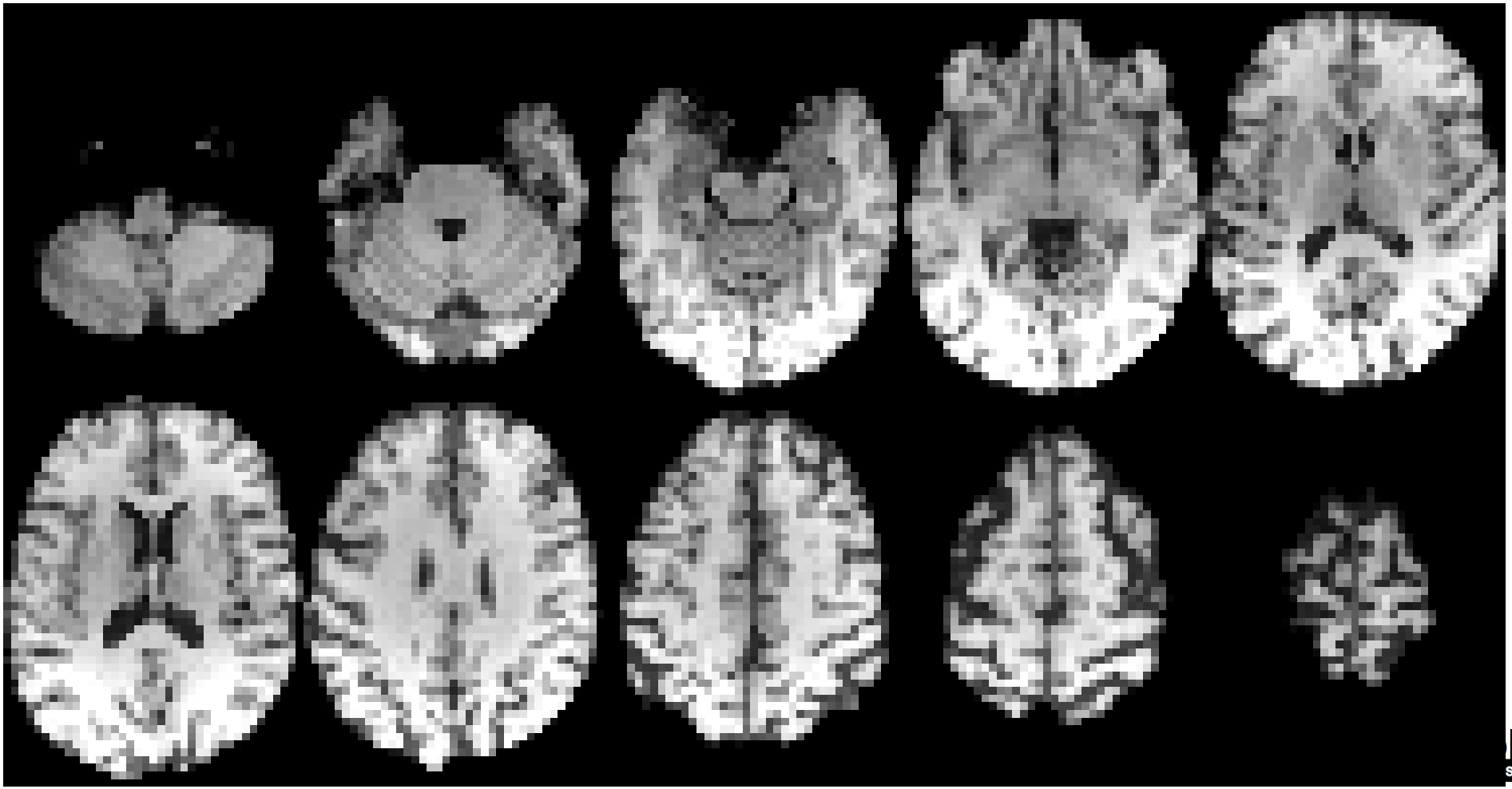
We now have an “anatomical” image . . . the average BOLD.

Let’s quickly map to T1.

```
t1seg=paste(pre,"seg.nii.gz",sep='')
t1n4=paste(pre,"t1.nii.gz",sep='')
if ( file.exists(t1seg)  )
  {
  t1seg=antsImageRead( t1seg )
  t1n4=antsImageRead( t1n4 )
  t1brain=t1n4 * thresholdImage( t1seg, 1, 6 )
  # might modify above depending on coverage
  }
bavgn3=n3BiasFieldCorrection( amc$moco_avg_img, 2 ) * amc$moco
# disco=antsRegistration( bavgn3, t1brain, "SyNBold" ) # probe
disco=antsRegistration( bavgn3, t1brain, "SyN" )
segw=antsApplyTransforms( bavgn3, t1seg,
```

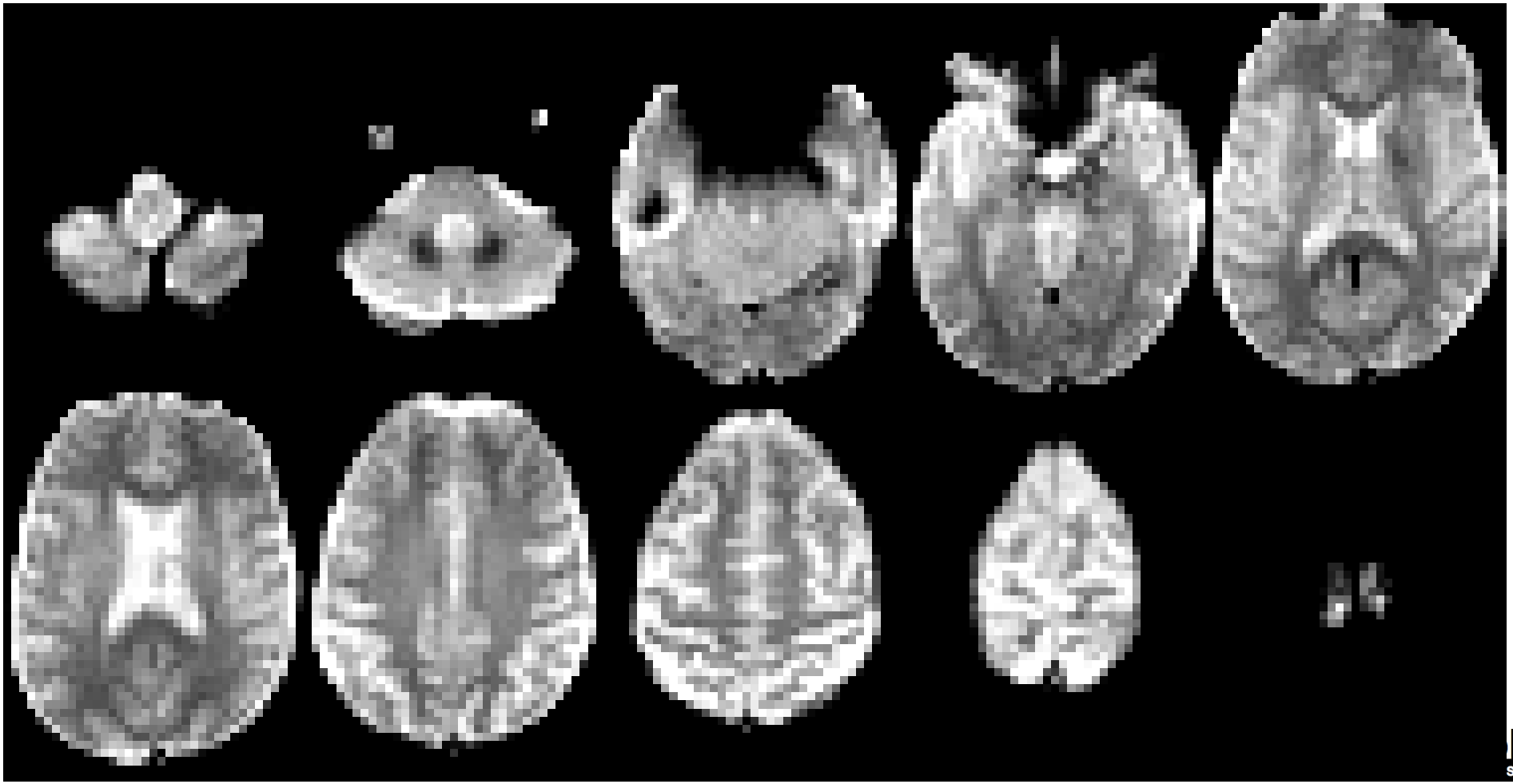
Mapped T1

```
plot( disco$warpedmovout, axis=3 )
```



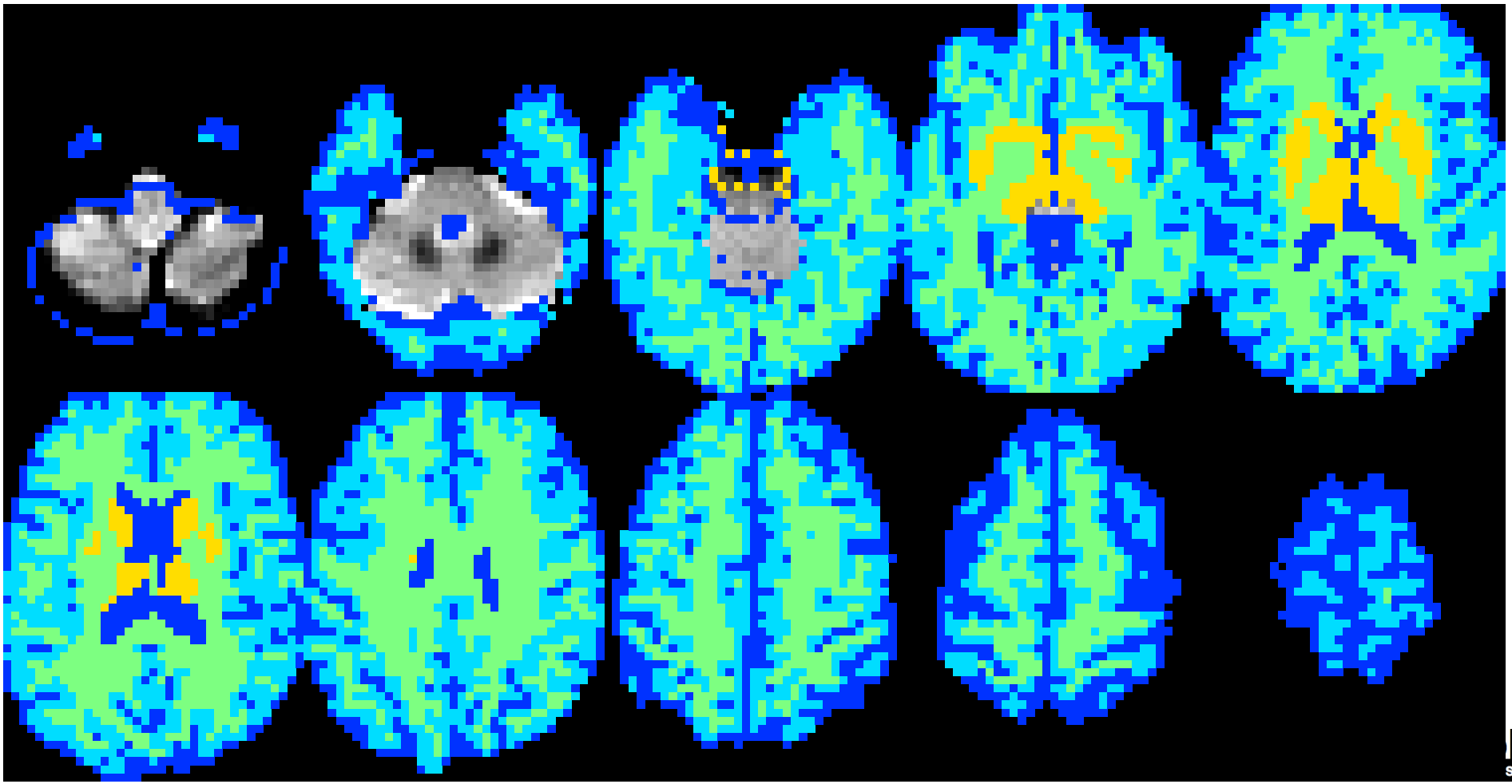
Target image

```
plot(  bavgn3 , axis=3 )
```



Mapped Segmentation

```
plot( bavgn3, segw, window.overlay=c(0,5), axis=3 )
```



Mapping to a T1 group template

We concatenate the distortion correction parameters with the group template mapping.

Then apply to the labels to bring them to the BOLD subject space.

Exercise?

We already did this so let's just read the labels.

```
aalfn=paste(pre,"aal.nii.gz",sep='')
if ( file.exists(aalfn) ) {
  aalimg = antsImageRead( aalfn )
}
```

A mapping exercise: Template to T1 to Bold

```
if ( ! file.exists(aalfn) ) {
  mni = antsImageRead( getANTsRData( "mni" ) )    # download te
  mnia = antsImageRead( getANTsRData( "mnia" ) ) # download te
  areg = antsRegistration( disco$warpedmovout, mni, typeofTran
  aalimg = antsApplyTransforms( disco$warpedmovout, mnia, tran
                                interpolator = 'nearestneighbor'
  plot( bavgn3, aalimg, window.overlay=c(0,max(aalw)), axis=3
  }
```

How can we improve on this approach?

Can we exploit transform composition? Use a better reference?

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

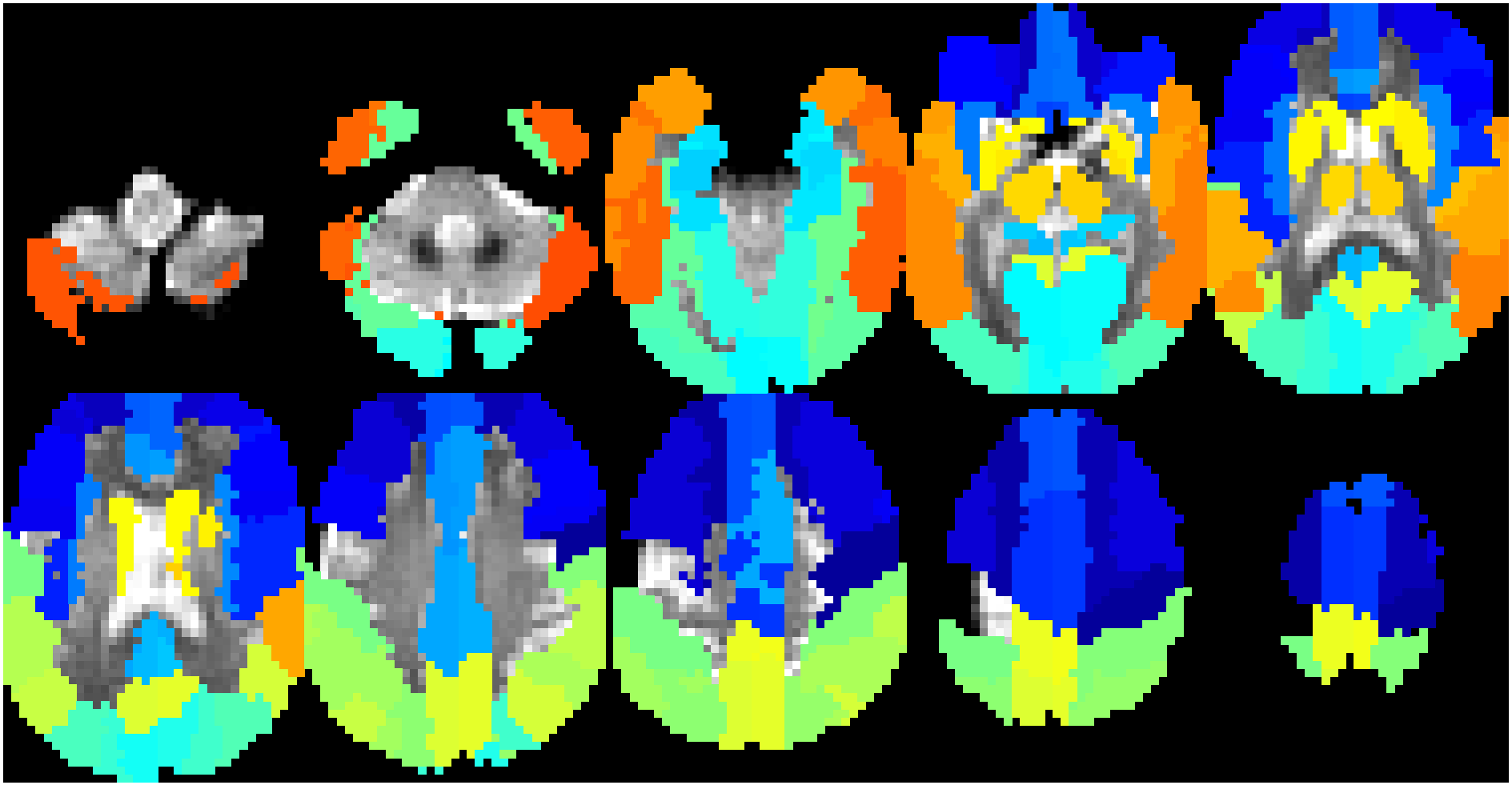
A mapping solution: Template to T1 to Bold

```
mni = antsImageRead( getANTsRData( "mni" ) )    # download temp
mnia = antsImageRead( getANTsRData( "mnia" ) ) # download temp
areg2 = antsRegistration( t1brain, mni, typeofTransform = 'SyN
concatMap = c( disco$fwdtransforms, areg2$fwdtransforms )
aalimg = antsApplyTransforms( disco$warpedmovout, mnia,
    transformlist = concatMap,
    interpolator = 'nearestneighbor' )
```

Why might this be better?

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

View the labels



```
## NULL
```

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Data-driven nuisance modeling

Nick prepackaged a generic processor for this . . .

- We have a few methods but `compcor` is nice.

```
boldpre=preprocessfMRI( fmri,
  numberOfCompCorComponents = 6,
  doMotionCorrection = 0,
  useMotionCorrectedImage = 0,
  spatialSmoothingType='none',
  spatialSmoothingParameters = mean( antsGetSpacing(fmri)[1:3]
  residualizeMatrix = TRUE,
  frequencyLowThreshold=0.01,
  frequencyHighThreshold=0.1
  )
```

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Preprocessor outputs

Nick prepackaged a generic processor for this . . .

- This redoes a few things we did above but now you know a little about what's happening inside.

```
> names(boldpre)
[1] "cleanBoldImage"    "maskImage"          "DVARS"
[4] "DVARSpostCleaning" "FD"                 "globalSignal"
[7] "nuisanceVariables"
```

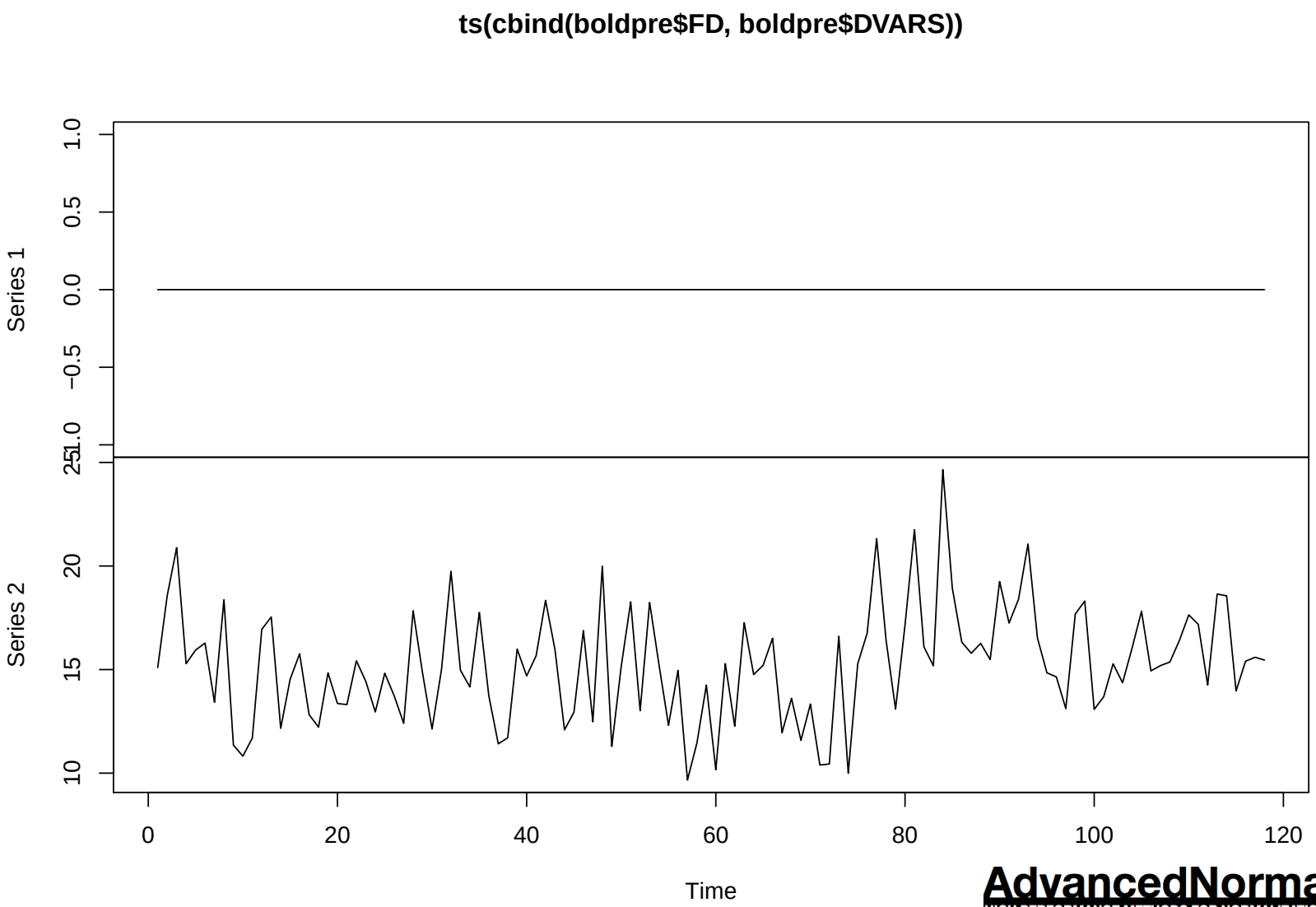
Preprocessor outputs

Nick prepackaged a generic processor for this . . .

- This redoes a few things we did above but now you know a little about what's happening inside.
- Should we smooth?

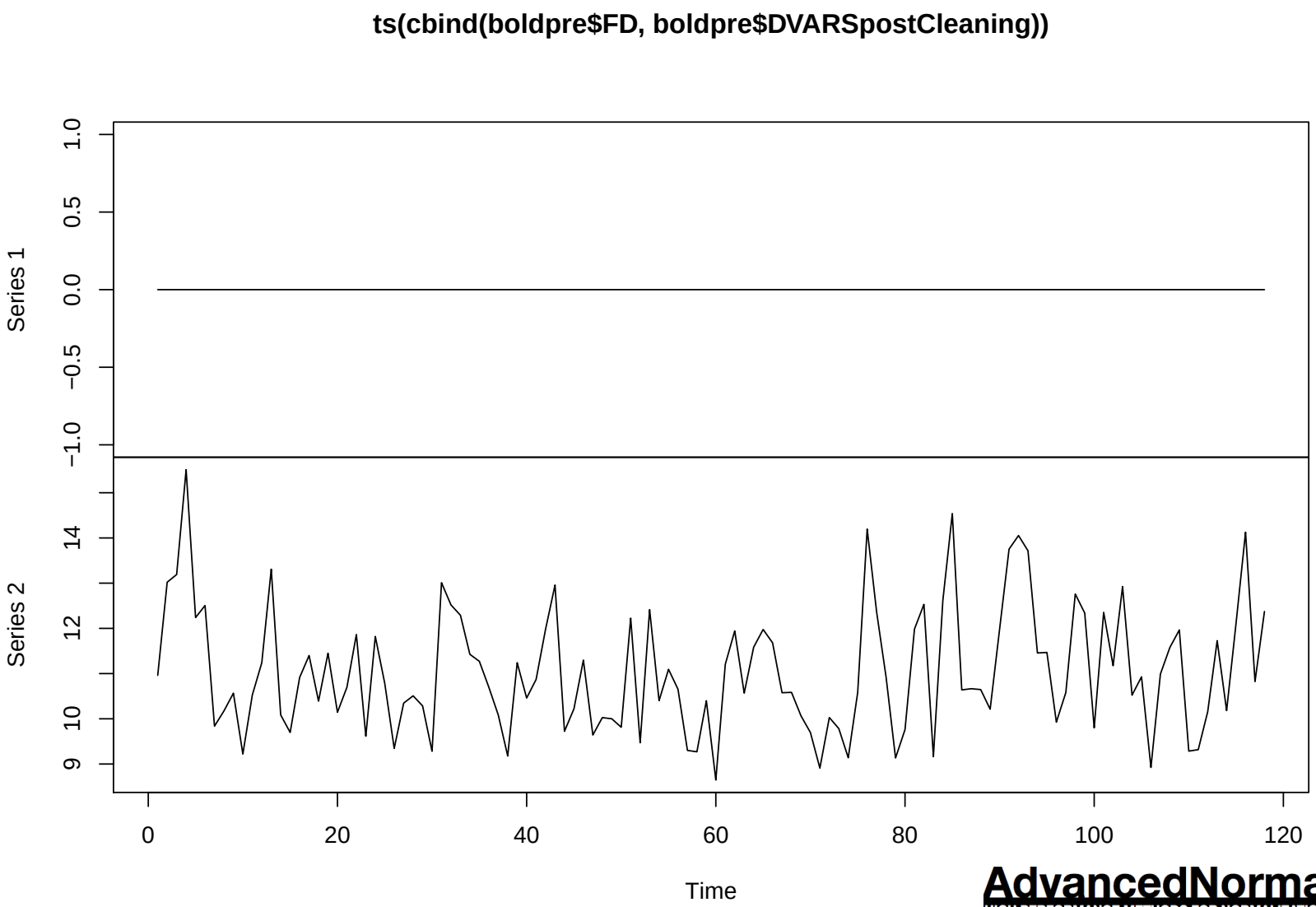
```
> names(boldpre)
[1] "cleanBoldImage"    "maskImage"          "DVARS"
[4] "DVARSpostCleaning" "FD"                 "globalSignal"
[7] "nuisanceVariables"
```

Look at FD and DVARS



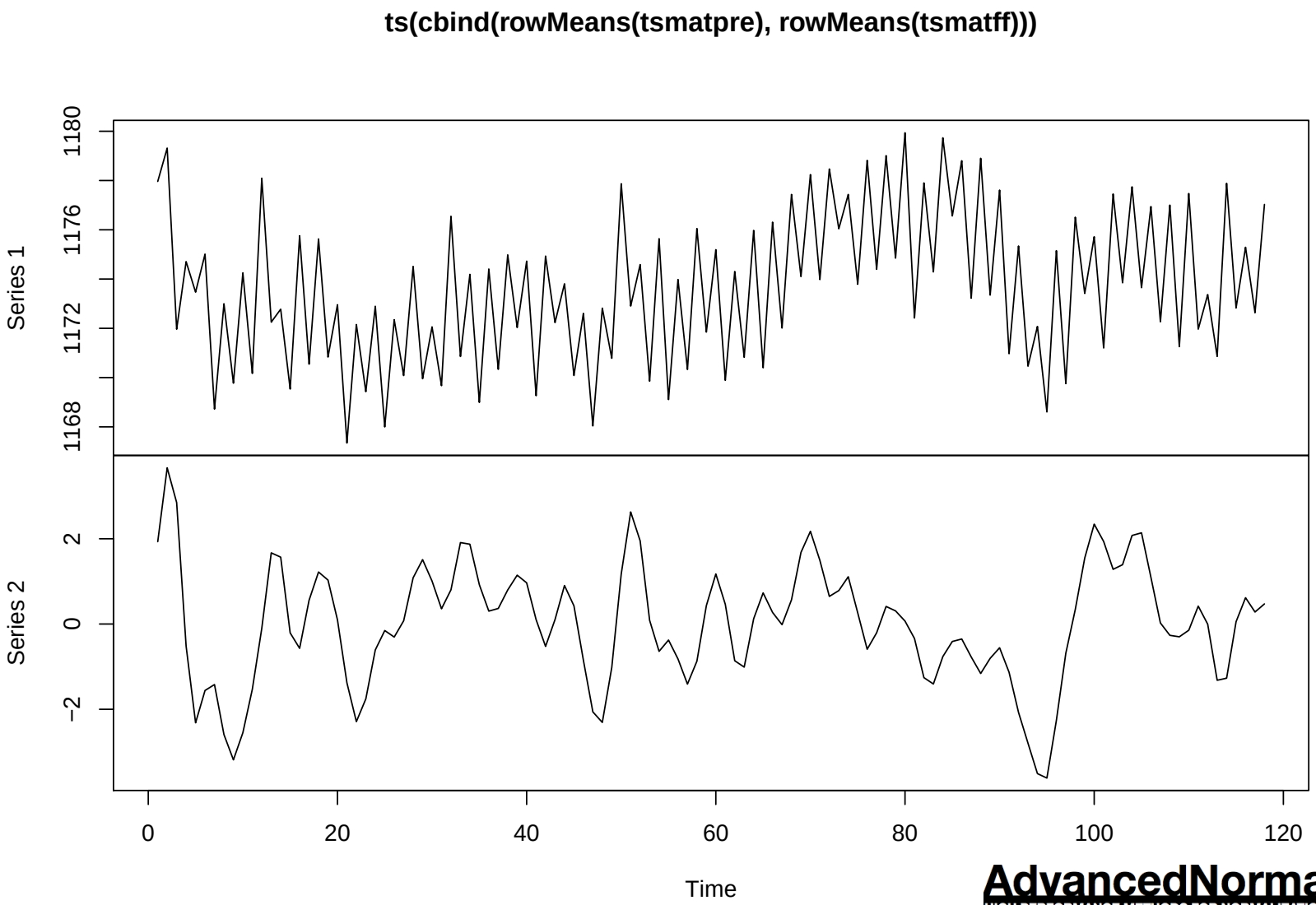
AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Look at FD and DVARS: Post clean



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Global signal



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Nuisance modeling: a little detail . . .

Nuisance variables can take several different forms.

- frequency filtering removes “non-neural signal” (putatively)

Nuisance modeling: a little detail . . .

Nuisance variables can take several different forms.

- frequency filtering removes “non-neural signal” (putatively)
- tissue-specific nuisance variables try to capture non-neural signal in non-neural tissue

Nuisance modeling: a little detail . . .

Nuisance variables can take several different forms.

- frequency filtering removes “non-neural signal” (putatively)
- tissue-specific nuisance variables try to capture non-neural signal in non-neural tissue
- data-driven methods, such as `compcor` or ICA, seek to estimate the nuisance signal from the data

Nuisance modeling: a little detail . . .

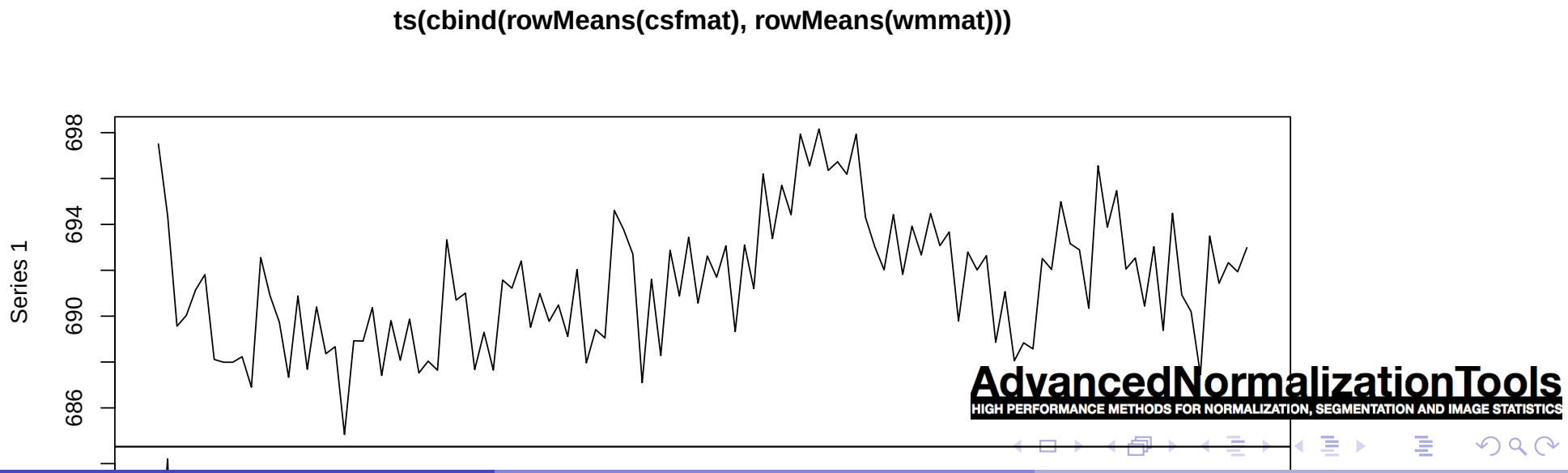
Nuisance variables can take several different forms.

- frequency filtering removes “non-neural signal” (putatively)
- tissue-specific nuisance variables try to capture non-neural signal in non-neural tissue
- data-driven methods, such as `compcor` or ICA, seek to estimate the nuisance signal from the data
- at this time, i prefer `compcor` . . . . why might we prefer it?

Tissue nuisance variables

Get tissue signals.

```
csfmat = timeseries2matrix( amc$moco_img,
      thresholdImage( segw, 1, 1 ) )
wmmat = timeseries2matrix( amc$moco_img,
      thresholdImage( segw, 3, 3 ) )
plot( ts( cbind( rowMeans( csfmat ),
                 rowMeans( wmmat )  ) ) )
```



CompCor

- Find high variance voxels (tend to be in CSF and WM)

```
# ccmat = timeseries2matrix( amc$moco_img, amc$moco_mask )
mycompcor = compcor( tsmatff, 6 )
print( colnames( mycompcor ) )
```

```
## [1] "compcorr1" "compcorr2" "compcorr3" "compcorr4" "compc
```

CompCor

- Find high variance voxels (tend to be in CSF and WM)
- Perform PCA on these voxels

```
# ccmat = timeseries2matrix( amc$moco_img, amc$moco_mask )
mycompcor = compcor( tsmatff, 6 )
print( colnames( mycompcor ) )
```

```
## [1] "compcorr1" "compcorr2" "compcorr3" "compcorr4" "compc
```

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

CompCor

- Find high variance voxels (tend to be in CSF and WM)
- Perform PCA on these voxels
- Use the top k components as covariates of no interest

```
# ccmat = timeseries2matrix( amc$moco_img, amc$moco_mask )
mycompcor = compcor( tsmatff, 6 )
print( colnames( mycompcor ) )
```

```
## [1] "compcorr1" "compcorr2" "compcorr3" "compcorr4" "compc
```

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

CompCor

- Find high variance voxels (tend to be in CSF and WM)
- Perform PCA on these voxels
- Use the top k components as covariates of no interest
- Advantages: automated, fast, principled, validated with physiological measurements

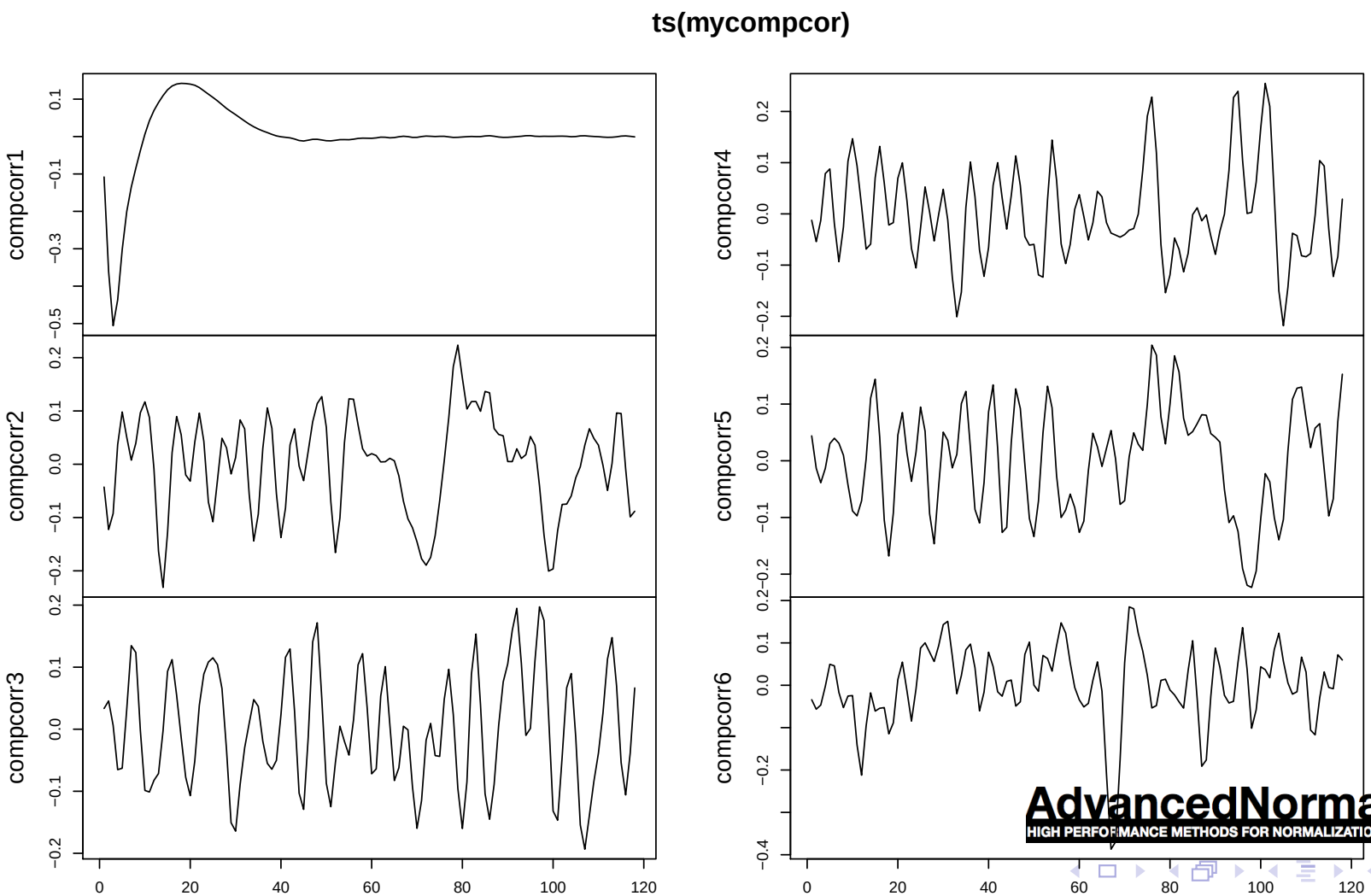
```
# ccmat = timeseries2matrix( amc$moco_img, amc$moco_mask )
mycompcor = compcor( tsmatff, 6 )
print( colnames( mycompcor ) )
```

```
## [1] "compcorr1" "compcorr2" "compcorr3" "compcorr4" "compc
```

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

CompCor: Plot

```
plot( ts( mycompcor ) )
```



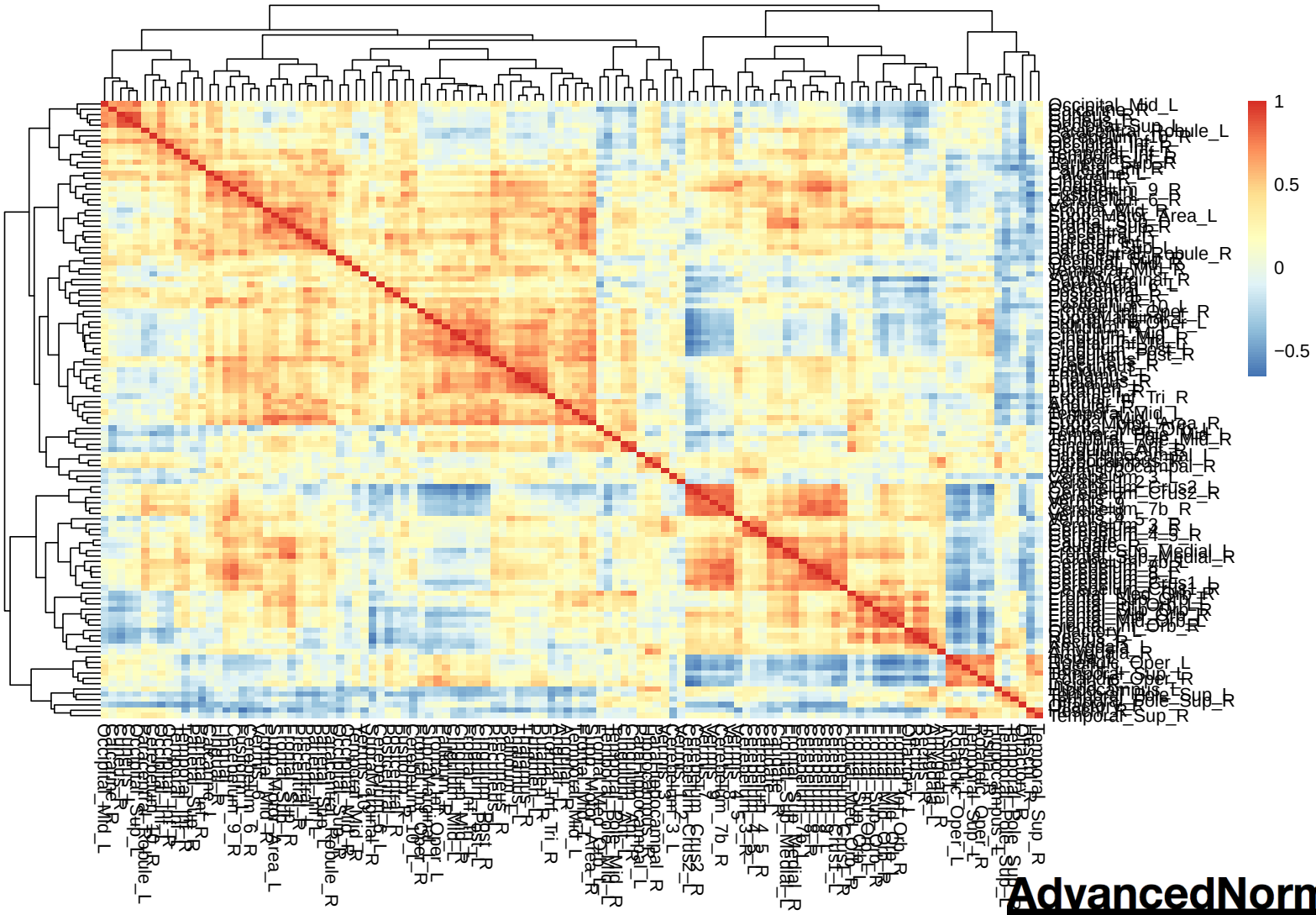
AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Now we can construct time-series averages for each region

Just use matrix multiplication.

```
data("aal")
labmat = labels2matrix( aalimg,  boldpre$maskImage,
                        targetLabels = aal$label_num )
residmat = residuals( lm( tsmatff ~ mycompcor  ) )
tsavg = residmat %*% t(labmat)
tsavgcor = antsrimpute( cor(tsavg) )
rownames( tsavgcor ) = aal$label_name
colnames( tsavgcor ) = aal$label_name
```

Look quickly at the correlations



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Network metrics

Now we can estimate connectivity from the BOLD data.

We'll use some nice *ANTsR* tricks for this.

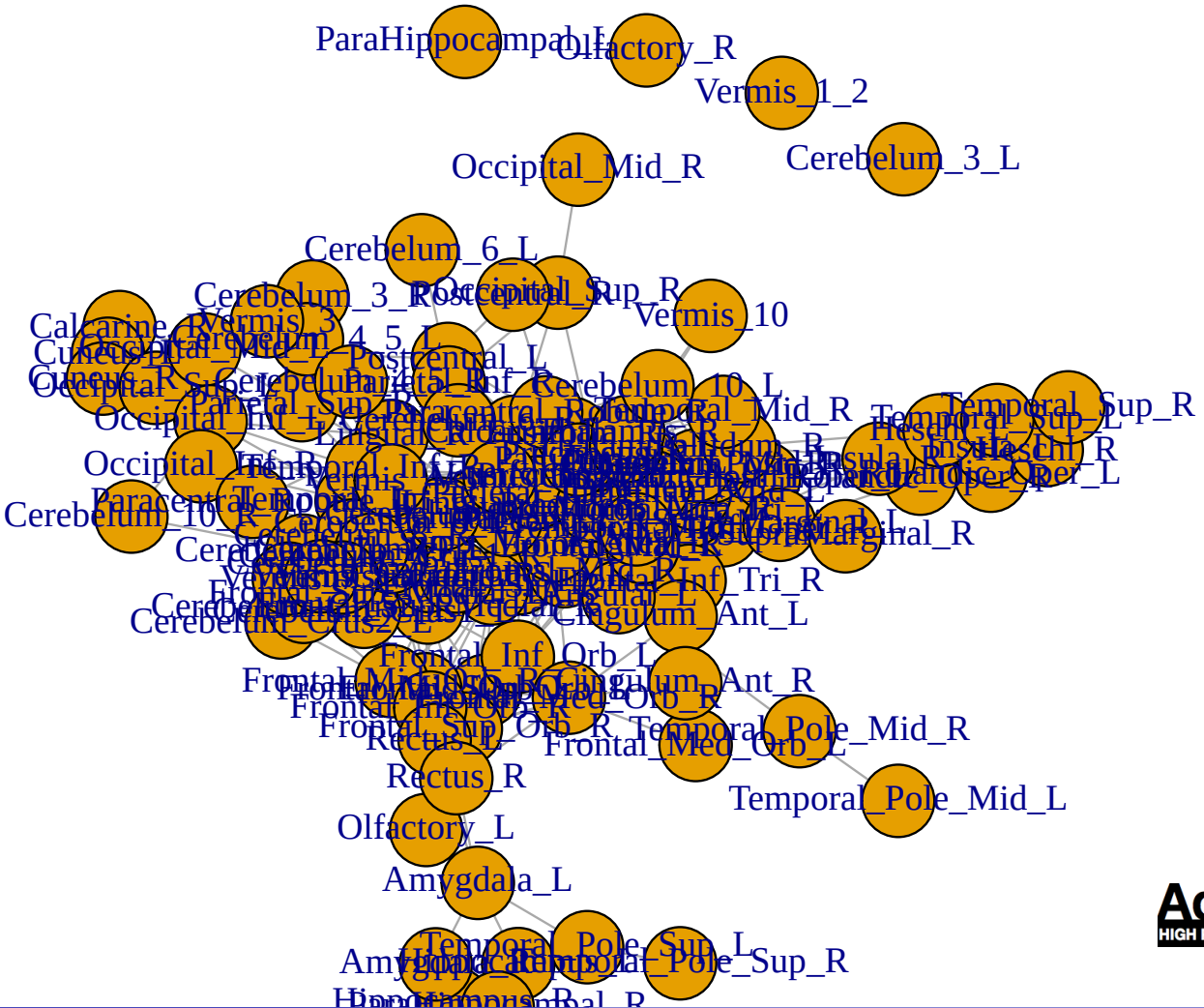
```
gmet <- makeGraph( tsavgcor, graphdensity = 0.1,
                   communityMethod = 'greedy' )
```

Outputs

```
> names(gmet)
 [1] "mygraph"           "centrality"        "closeness"
 [4] "pagerank"          "degree"            "betweeness"
 [7] "localtransitivity" "strength"          "degcent"
[10] "effinv"            "community"         "walktrapcomm"
[13] "adjacencyMatrix"
```

Network visualization with `igraph`

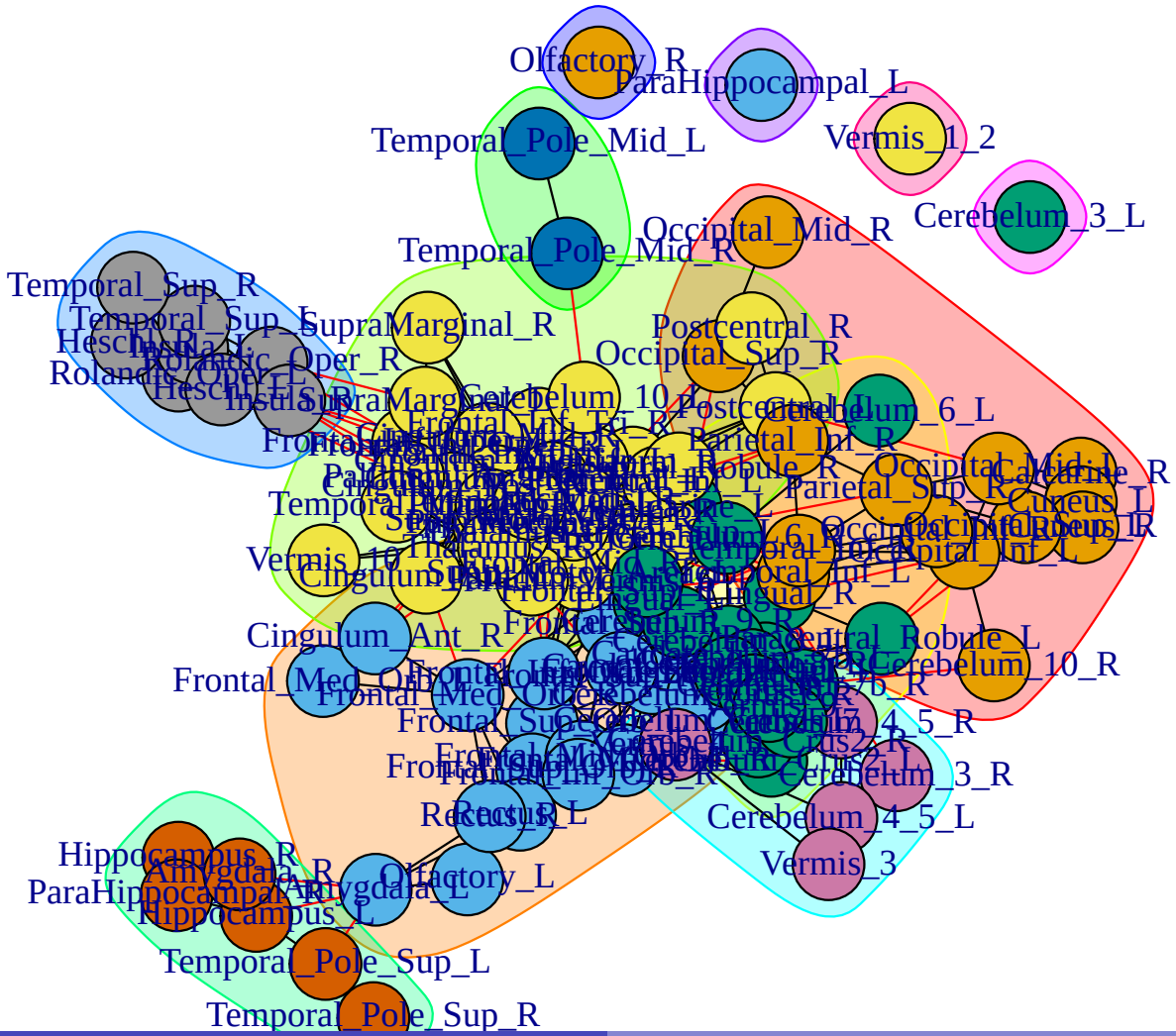
```
plot( gmet$mygraph )
```



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Community visualization with igraph

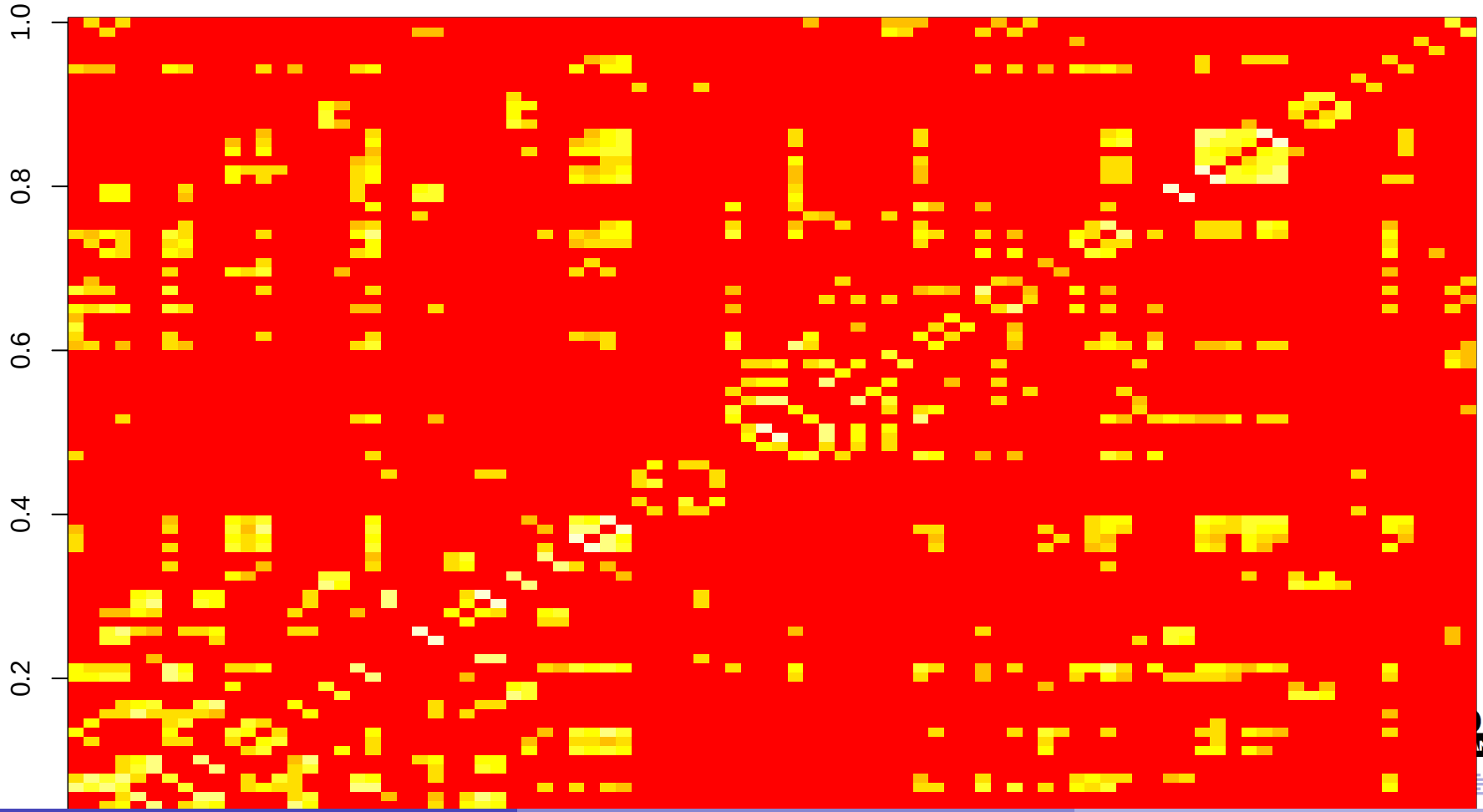
```
plot( gmet$community, gmet$mygraph)
```



AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Look at the connection matrix

```
metweights=gmet$adjacencyMatrix[1:90,1:90]
image(metweights)
```



Network visualization in brain space

This is something we have to run “by hand”

```
cnt<-getCentroids( aalimg, clustparam = 0 )
aalcnt<-cnt[1:90,1:3] # cortex
brain<-renderSurfaceFunction( surfimg=
  list( boldpre$maskImage ) , alphasurf=0.1,
  smoothsval = 1.5 )
metweights[ metweights < 0.01 ] = 0
plotBasicNetwork( centroids = aalcnt, brain, weights=metweigh
```

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.
- Maybe we should have thrown a few frames away ... how? hint: `matrix2timeseries` ...

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.
- Maybe we should have thrown a few frames away . . . how? hint: `matrix2timeseries` ...
- Maybe we should have imputed some data . . . see `antsBOLDNetworkAnalysis`

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.
- Maybe we should have thrown a few frames away ... how? hint: `matrix2timeseries` ...
- Maybe we should have imputed some data ... see `antsBOLDNetworkAnalysis`
- Maybe we should have used the tissue segmentation ....

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.
- Maybe we should have thrown a few frames away ... how? hint: `matrix2timeseries ...`
- Maybe we should have imputed some data ... see `antsBOLDNetworkAnalysis`
- Maybe we should have used the tissue segmentation ....
- How might we do group statistics?

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.
- Maybe we should have thrown a few frames away ... how? hint: `matrix2timeseries` ...
- Maybe we should have imputed some data ... see `antsBOLDNetworkAnalysis`
- Maybe we should have used the tissue segmentation ....
- How might we do group statistics?
- We mostly produce node metrics but edge metrics are good too ...

Discussion

- There is also `antsBOLDNetworkAnalysis` but it makes many assumptions that may not hold. Need to look at the code.
- Maybe we should have thrown a few frames away ... how? hint: `matrix2timeseries` ...
- Maybe we should have imputed some data ... see `antsBOLDNetworkAnalysis`
- Maybe we should have used the tissue segmentation ....
- How might we do group statistics?
- We mostly produce node metrics but edge metrics are good too ...
- Any other thoughts?

AdvancedNormalizationTools
HIGH PERFORMANCE METHODS FOR NORMALIZATION, SEGMENTATION AND IMAGE STATISTICS