

PCA Interpretation

2023-04-30

```
library(ggplot2)
library(knitr)
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## vforcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr    1.0.1     v tidyr    1.3.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
## 
##      lift
library(pracma)

##
## Attaching package: 'pracma'
##
## The following object is masked from 'package:purrr':
## 
##      cross
library(oro.nifti)

## oro.nifti 0.11.4
##
## Attaching package: 'oro.nifti'
##
## The following object is masked from 'package:pracma':
## 
##      magic
##
## The following object is masked from 'package:dplyr':
## 
##      slice
```

```

library(MASS)

##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
library(pracma)

load(file = file.path(root_dir, "R_data", "X_wm.Rda"))
load(file = file.path(root_dir, "R_data", "pca_wm.Rda"))

load(file = file.path(root_dir, "R_data", "X_gm.Rda"))
load(file = file.path(root_dir, "R_data", "pca_gm.Rda"))

load(file = file.path(root_dir, "R_data", "X_cb.Rda"))
load(file = file.path(root_dir, "R_data", "pca_cb.Rda"))

```

PCA is a dimensionality reduction technique used to reduce the dimensionality of large datasets of image pixel intensities. The main idea is to represent the images using a smaller set of features or variables that capture most of the variance in the data. These features are the principal components, which are linear combinations of the original pixel intensities. For example, if we have values of 0.1, 0.5, 0.2 as intensities, then $x_1 * 0.1 + x_2 * 0.5 + x_3 * 0.2$ is a linear combination of the image intensities. If we want to find out information about the image, specifically which sections show the most variance, we can look at the PCA loading vectors

The loading vectors or coefficients of the principal components indicate the contribution of each original pixel intensity to the respective principal component. A higher loading value (positive or negative) indicates a more significant contribution of the corresponding pixel intensity to the respective principal component.

To calculate the loading vectors or coefficients, we need to invert the PCA transformation by multiplying the pseudo-inverse of the original pixel intensity matrix with the principal component matrix. In this case, the A_pinv matrix is the pseudo-inverse of the original pixel intensity matrix, and the b matrix contains the principal components. The resulting x matrix contains the loading vectors or coefficients.

The singular values or PC weights (pca_gm\$sdev) are multiplied by the loading vectors or coefficients to get the final normalized coefficients. The normalization step scales the coefficients to values between 0 and 1, which can be used to generate a new image that highlights the regions of the brain that contribute the most to the principal components.

The resulting image should show brighter values in the regions of the brain that are most important for distinguishing between the different classes of patients (CN, MCI, and AD) in the original dataset.

```

coef_vec_wm <- calc_loading_vec(pca_wm, n_components=160)
coef_mat_wm <- find_contributing_regions(X_wm$nzv, coef_vec_wm, slice=121)

## Warning in img_vec[-which(img_vec == 0)] <- loading_vec: number of items to
## replace is not a multiple of replacement length
saveRDS(coef_mat_wm, file.path(root_dir, "R_data", "coef_mat_wm.rds"))

coef_vec_gm <- calc_loading_vec(pca_gm, n_components=144)
coef_mat_gm <- find_contributing_regions(X_gm$nzv, coef_vec_gm, slice=121)

## Warning in img_vec[-which(img_vec == 0)] <- loading_vec: number of items to
## replace is not a multiple of replacement length

```

```

saveRDS(coef_mat_gm, file.path(root_dir, "R_data", "coef_mat_gm.rds"))

coef_vec_cb <- calc_loading_vec(pca_cb, n_components=157)
coef_mat_cb <- find_contributing_regions(X_cb$nzv, coef_vec_cb, slice=121)

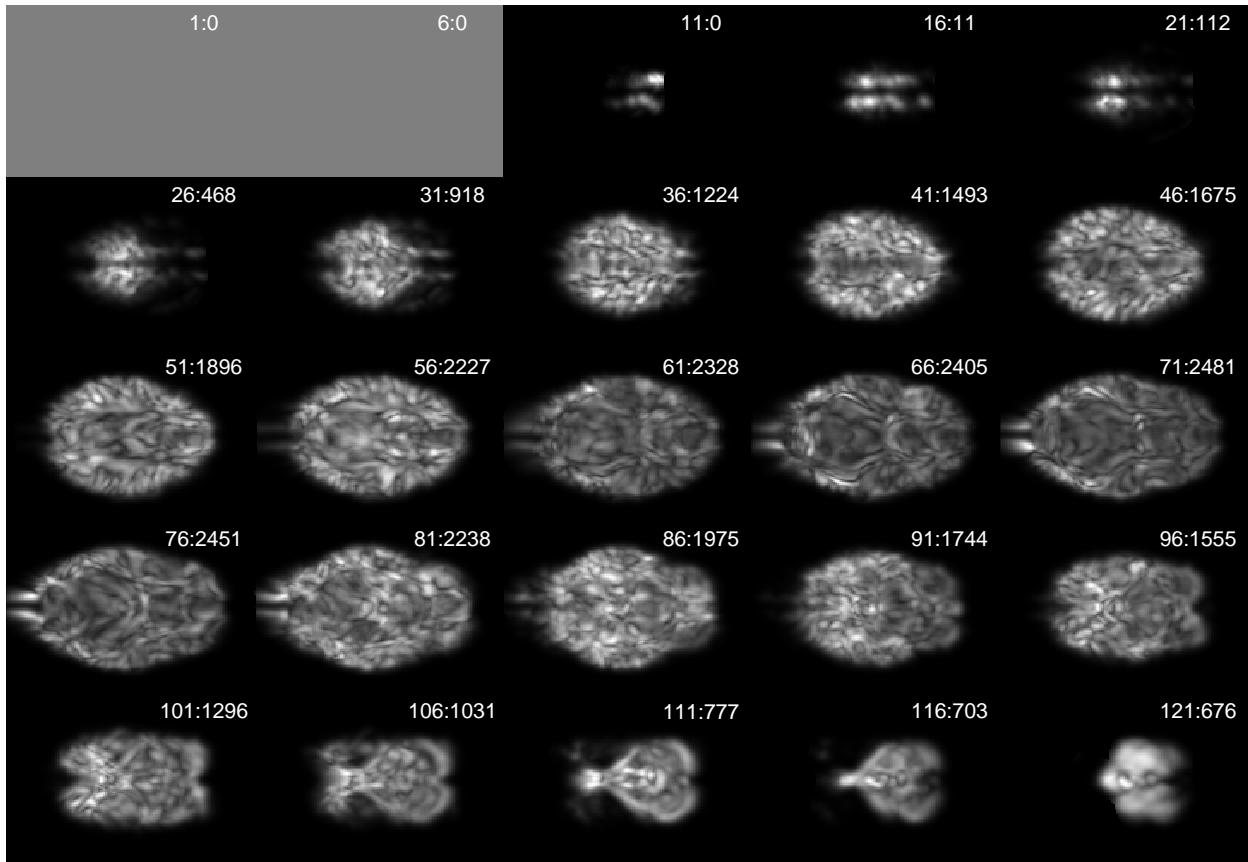
## Warning in img_vec[-which(img_vec == 0)] <- loading_vec: number of items to
## replace is not a multiple of replacement length
saveRDS(coef_mat_cb, file.path(root_dir, "R_data", "coef_mat_cb.rds"))

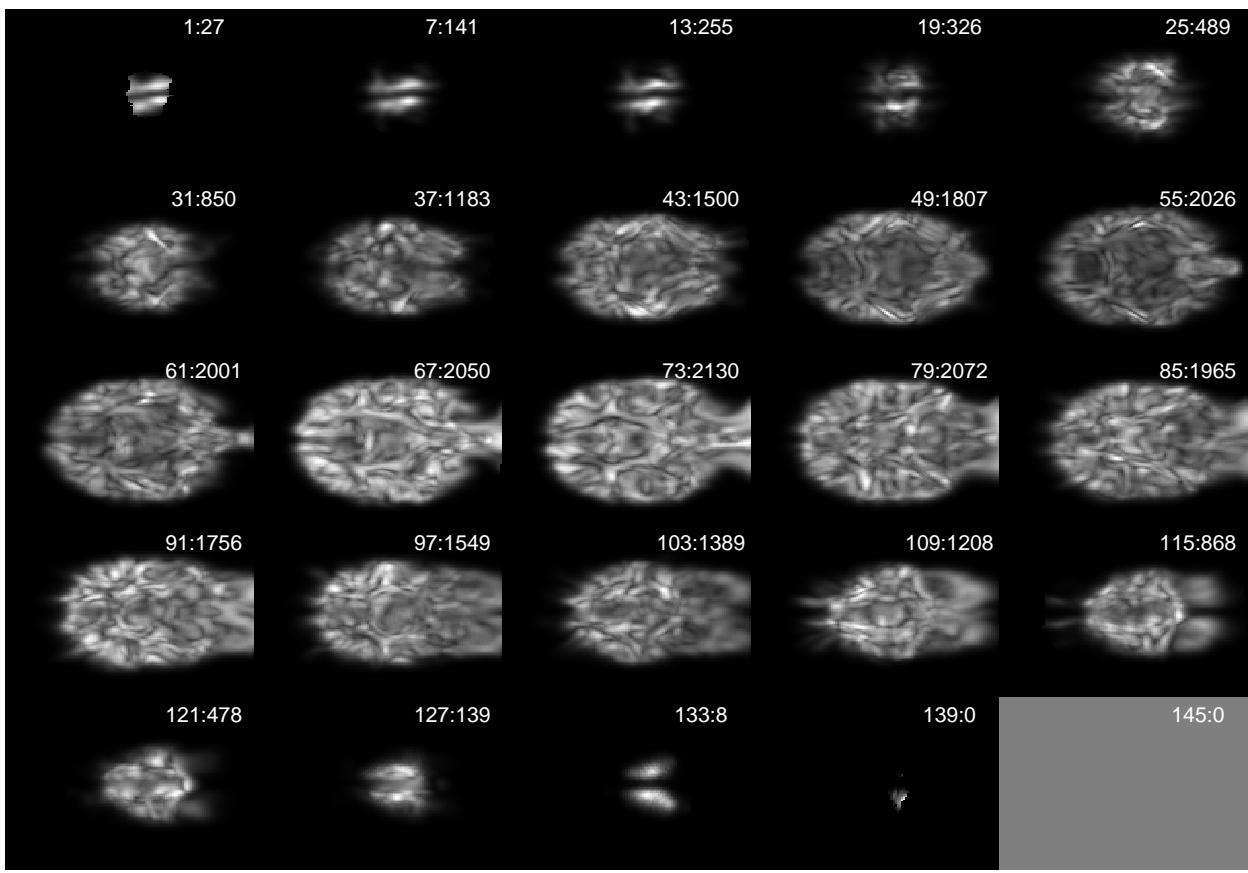
plot_lvs <- function(rds_file) {
  root_dir = '/scratch/users/neuroimage/conda/data'
  path <- file.path(root_dir, "R_data", rds_file)
  loading_vec <- readRDS(path)
  loading_mat <- array(loading_vec, dim=c(121,145,121))

  plot_axial_lv(loading_mat)
  plot_coronal_lv(loading_mat)
}

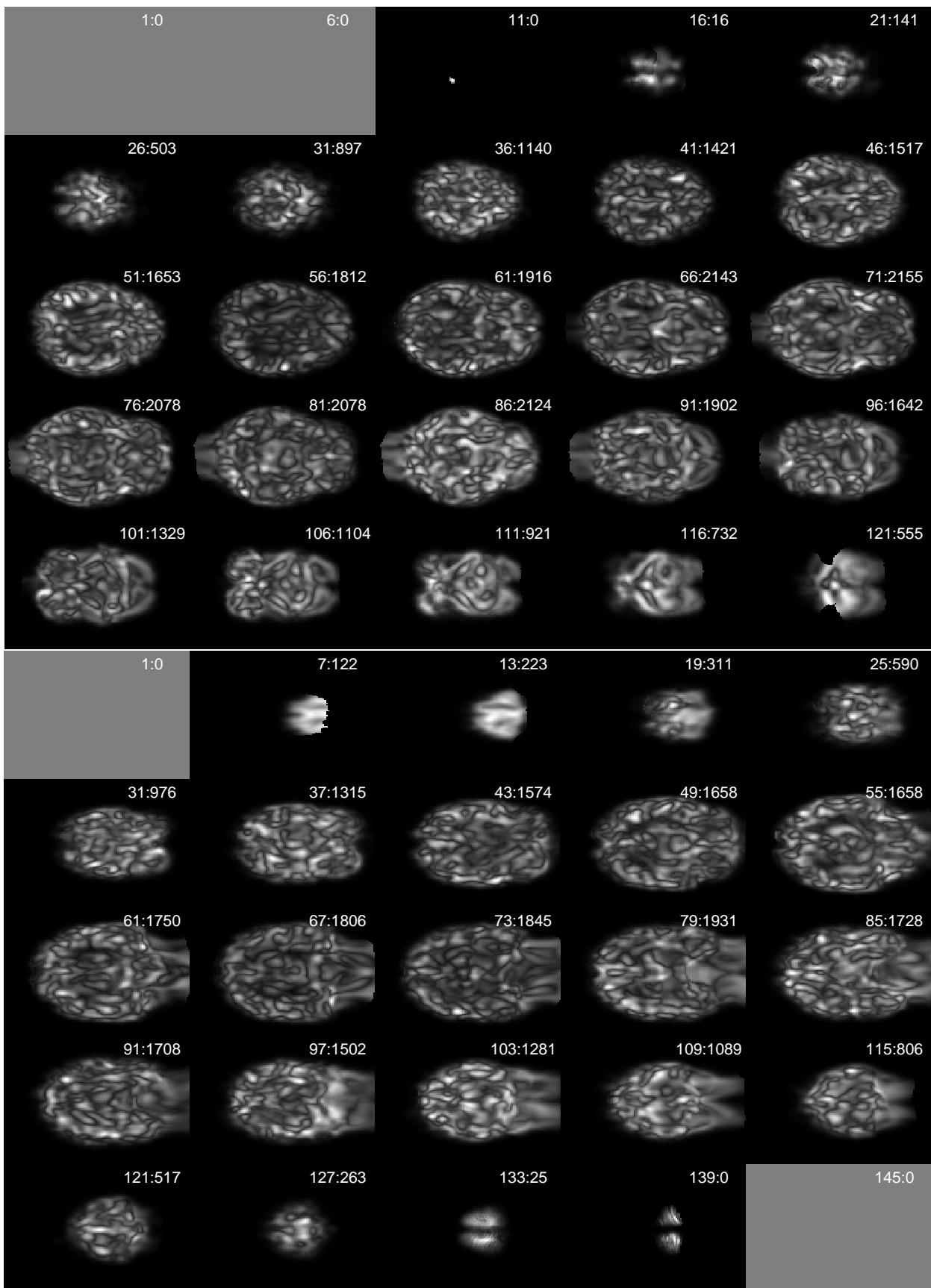
plot_lvs('coef_mat_wm.rds')

```





```
plot_lvs('coef_mat_gm.rds')
```



```
plot_lvs('coef_mat_cb.rds')
```

