Sulcal Phenotype Network Analysis

Will Snyder

Reading in data

First, read in the data for 25 randomly sampled adult subjects. Each variable here stores data with each subject as a unique row, with columns representing the measurement at each of 40 sulci. Average depth, depth variability, and longest branch are in mm. Branch span is unitless and ranges 0-1. FD is unitless and ranges ~1-2.

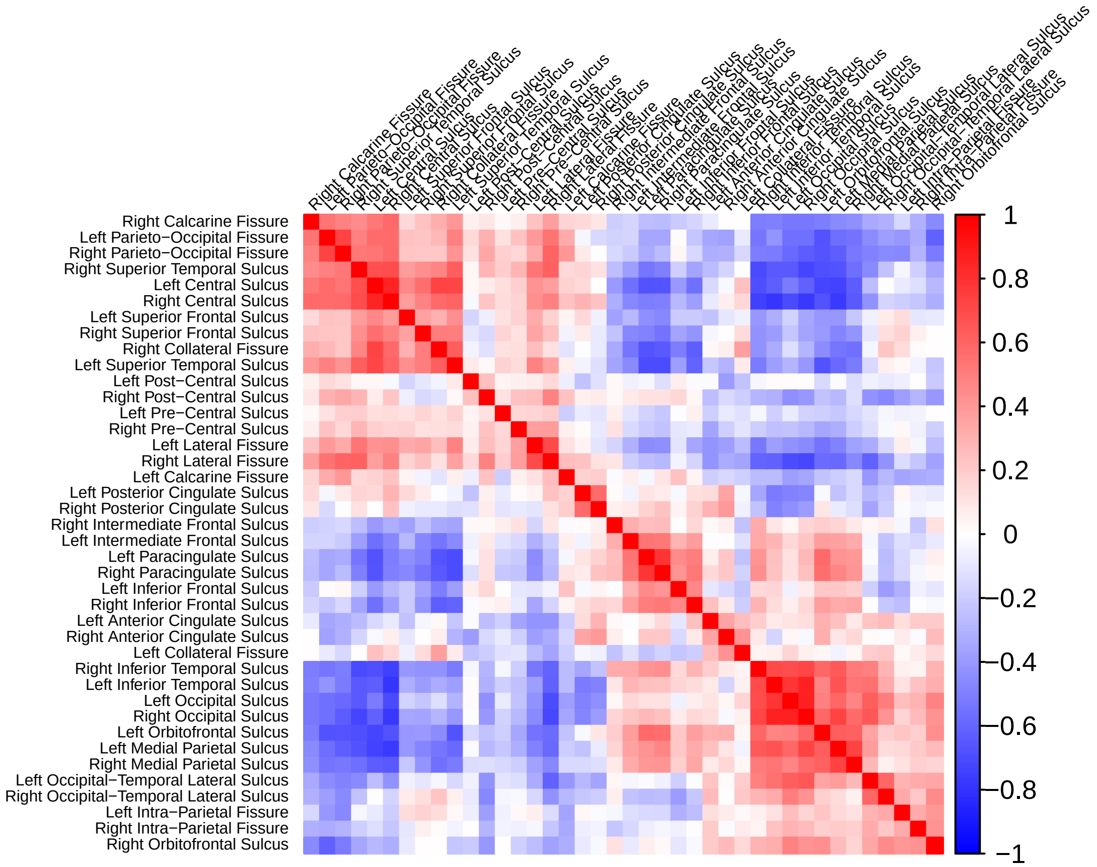
```
average_depth <- readRDS("~/Downloads/spn_analysis/average_depth.RDS")
depth_variability <- readRDS("~/Downloads/spn_analysis/depth_variability.RDS")
longest_branch <- readRDS("~/Downloads/spn_analysis/longest_branch.RDS")
branch_span <- readRDS("~/Downloads/spn_analysis/branch_span.rds")
fd <- readRDS("~/Downloads/spn_analysis/fd.rds")

# in example
head(average_depth[, c(1:4)])  #4 example sulci
```

```
##   Left Lateral Fissure Right Lateral Fissure Left Anterior Cingulate Sulcus
## 1                13.00                 13.10                           7.58
## 2                11.68                 16.10                           7.84
## 3                16.21                 14.50                           9.58
## 4                13.90                 11.71                          10.00
## 5                10.26                 15.64                           8.32
## 6                12.18                 13.84                          10.84
##   Right Anterior Cingulate Sulcus
## 1                            9.05
## 2                            9.53
## 3                            8.84
## 4                            9.19
## 5                            9.42
## 6                            9.16
```

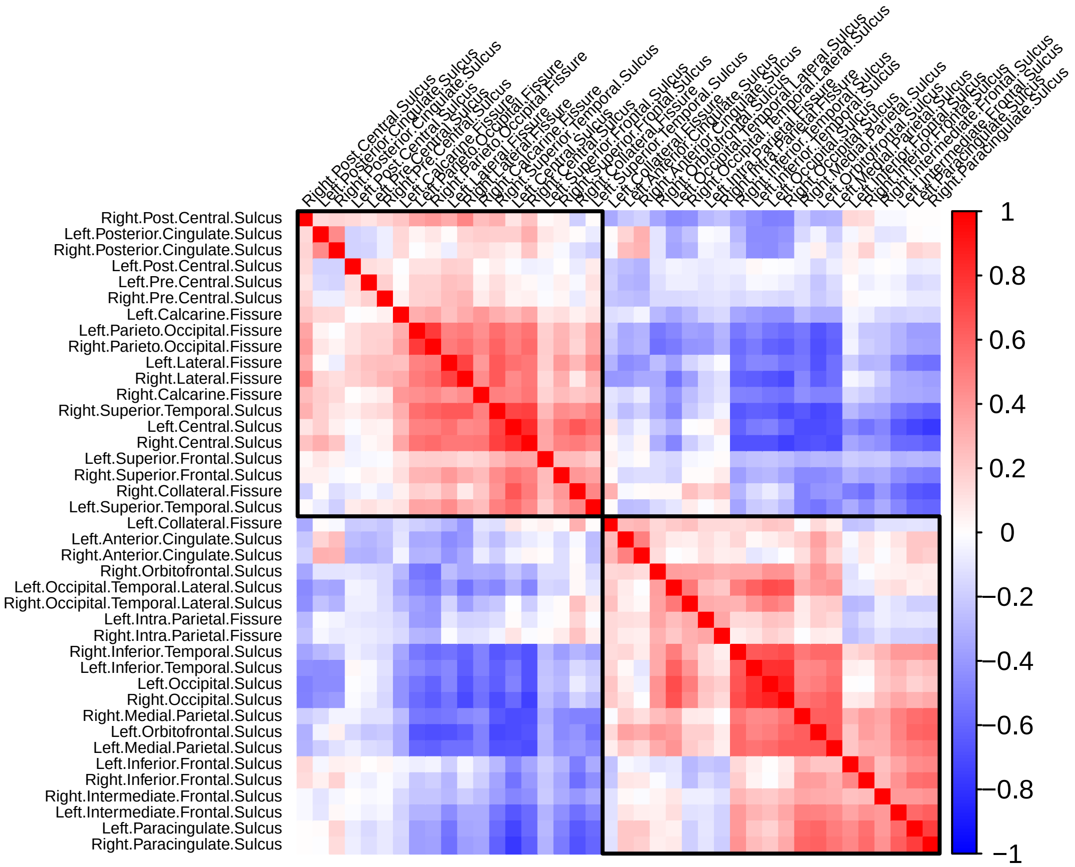
Generate SPNs and create mean SPN

Next, we create sulcal phenotype networks (SPNs) by first z-scoring sulcal phenotypes within each subject across sulci to get relative / normalized estimates of how extreme each sulcus is in its phenotype compared to other sulci in that brain. Then, we correlate the 1x5 vectors for each sulcus within each brain to create a matrix of cross-sulcal correlations that define the subject's SPN.

```
# create function to z-score a vector
z_score <- function(dataset_row) {
    z_vals <- (as.numeric(dataset_row) - mean(as.numeric(dataset_row),
        na.rm = TRUE))/sd(as.numeric(dataset_row), na.rm = TRUE)
    return(z_vals)
}
# apply the z-score function along each row (each subject)
# from dataframes
z.average_depth <- as.data.frame(t(apply(average_depth, 1, z_score)))
z.depth_variability <- as.data.frame(t(apply(depth_variability,
    1, z_score)))
z.longest_branch <- as.data.frame(t(apply(longest_branch, 1,
    z_score)))
z.branch_span <- as.data.frame(t(apply(branch_span, 1, z_score)))
z.fd <- as.data.frame(t(apply(fd, 1, z_score)))

# combine all z-scored values into one dataframe, where
# each row is a subject
z_scored_sulcal_phenotypes <- cbind(z.average_depth, z.depth_variability,
    z.longest_branch, z.branch_span, z.fd)

# create a function that reformats a subject's row of data
# into a matrix that can readily be cross-correlated,
# creating the SPN
create_subject_spn <- function(subject_row) {
    subject_feature_matrix <- matrix(as.numeric(subject_row),
        nrow = 40, ncol = 5)
    sub_correlation_matrix <- cor(t(subject_feature_matrix))
    return(sub_correlation_matrix)
}

# apply the SPN generation function to all subjects
subject_spns <- array(apply(z_scored_sulcal_phenotypes, 1, create_subject_spn),
    c(40, 40, nrow(z.average_depth)))

# get the group mean SPN from our sample dataset, using
# edge-wise averaging of SPN matrices
sample_average_spn <- rowMeans(subject_spns, dims = 2)
colnames(sample_average_spn) <- colnames(average_depth)
rownames(sample_average_spn) <- colnames(average_depth)

# plot the group mean SPN from our sample
corrplot(as.matrix(sample_average_spn), type = "full", tl.col = "black",
    order = "hclust", col = colorRampPalette(c("blue", "white",
        "red"))(200), method = "color", diag = TRUE, tl.cex = 0.5,
    tl.srt = 45)
```



```
# alternatively, load in group mean SPN from ~35,000
# subjects' data N.B. row orders change by hierarchical
# clustering for input to corrplot Notice the split of
# sulci in the 2-cluster solution
full_mean_SPN <- data.frame(read_xlsx(path = "~/Downloads/spn_analysis/Snyder_Neuron_Supplementary_Tabl
    sheet = "Table 2, SPN Mean Edge Weights")[, c(2:41)])
rownames(full_mean_SPN) <- colnames(full_mean_SPN)
corrplot(as.matrix(full_mean_SPN), type = "full", tl.col = "black",
    order = "hclust", col = colorRampPalette(c("blue", "white",
        "red"))(200), method = "color", diag = TRUE, tl.cex = 0.5,
    tl.srt = 45, addrect = 2)
```



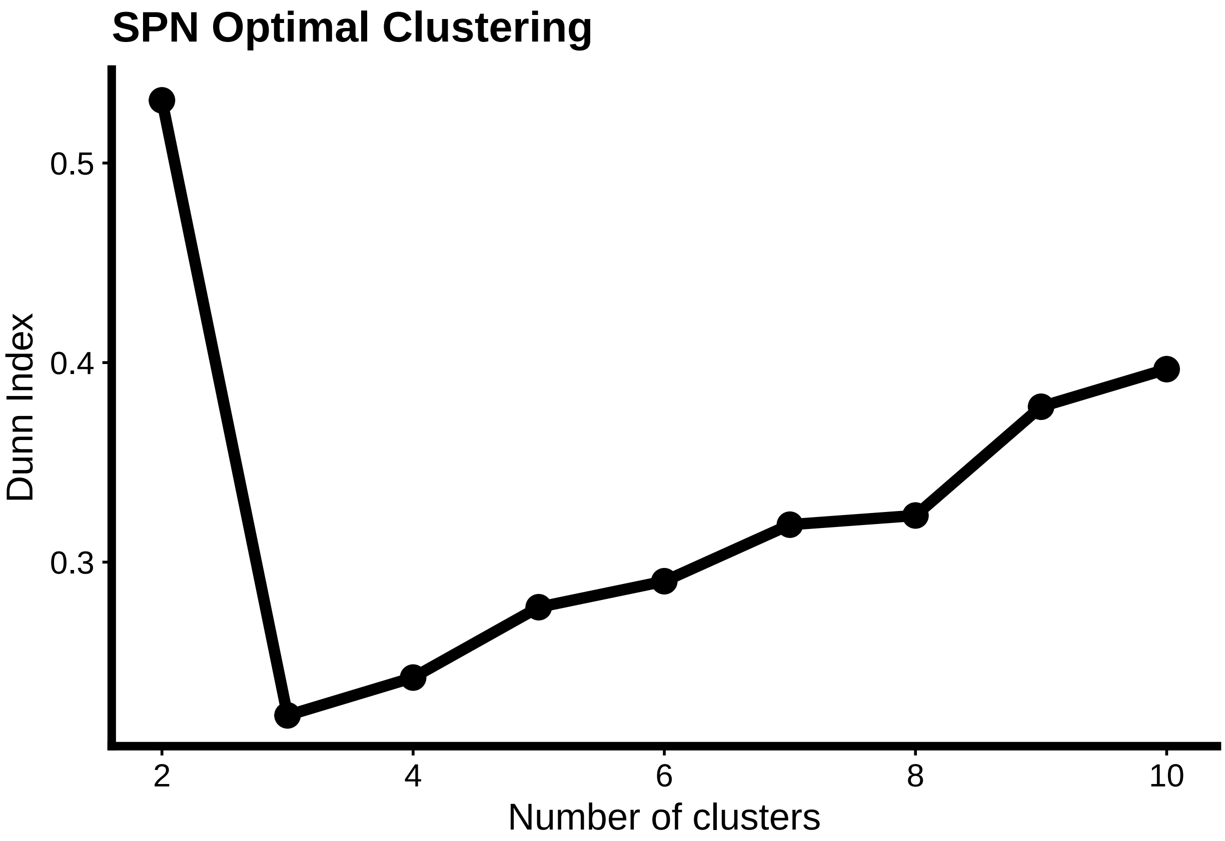
```r
# Note that the sample mean SPN and full (35k subjects')
# mean SPN are highly correlated
upper_vals_full_SPN <- as.matrix(full_mean_SPN)[upper.tri(as.matrix(full_mean_SPN))]  #vectorized upper
upper_vals_sample_SPN <- as.matrix(sample_average_spn)[upper.tri(as.matrix(sample_average_spn))]  #vect
cor(upper_vals_full_SPN, upper_vals_sample_SPN)
```

```
## [1] 0.9676144
```

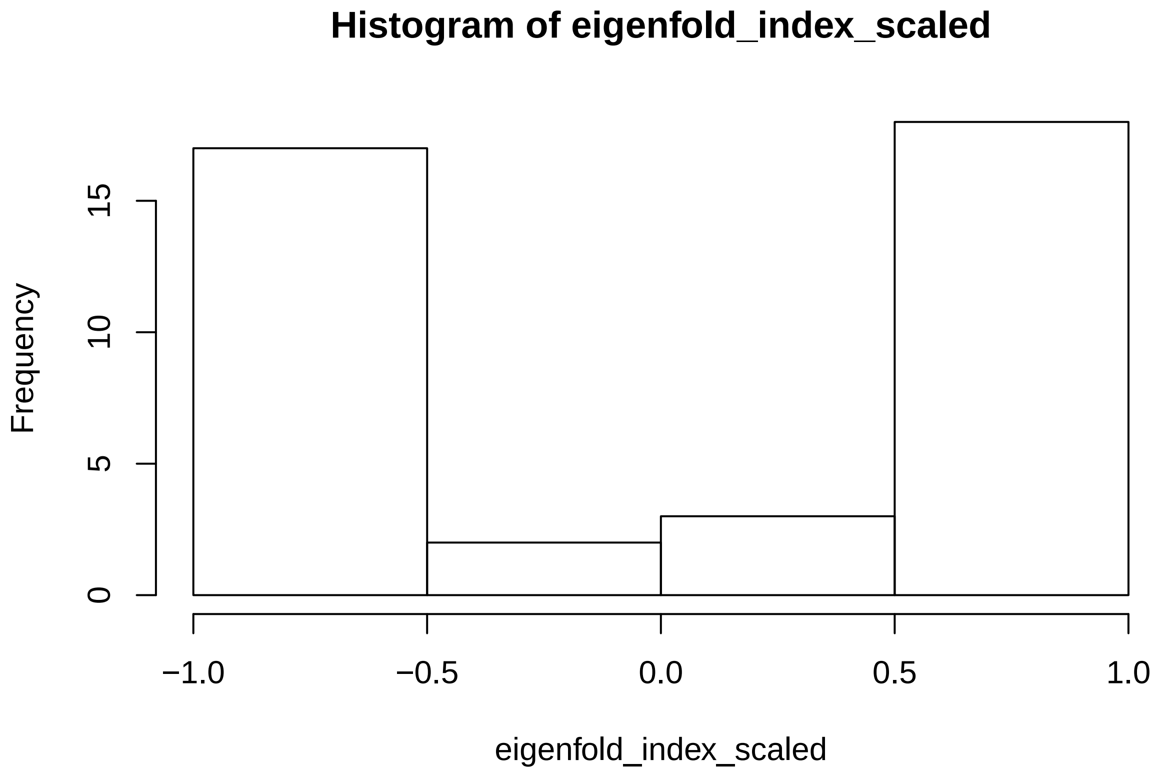
Categorical and dimensional representations of SPNs

We will go through both the clustering and dimensional representation of the group mean SPN from all 35k subjects. Dunn index evaluation is in Figure S5A, including further measures for how robust it is. EFI evaluation is in Figure S6A, including further rationale to only focus on the first principal component. Additionally, subject-level estimation of EFI is shown.

```r
# Dunn index evaluation

# loop over 2 through 10 reasonable clusters from
# heirarchical clustering dendrogram
dunn_index_values <- c()
mean_distance_matrix <- dist(full_mean_SPN)
mean_hclust <- hclust(mean_distance_matrix)
for (i in 2:10) {
    # cut for given number of cluster
    cluster <- cutree(mean_hclust, i)
    dunn_index_values <- c(dunn_index_values, dunn(mean_distance_matrix,
        cluster))
}

# The optimal (maximized) Dunn index is seen at a two
# cluster solution
ggplot() + geom_line(aes(x = 2:10, y = dunn_index_values), size = 2) +
    geom_point(aes(x = 2:10, y = dunn_index_values), size = 4) +
    theme_cowplot() + theme(axis.line = element_line(size = 1.5)) +
    ylab("Dunn Index") + xlab("Number of clusters") + ggtitle("SPN Optimal Clustering")
```



```
# To investigate dimensional representation of the SPN, use
# principal component analysis (PCA)

decomposition <- prcomp(full_mean_SPN, scale. = TRUE, center = TRUE)
# eigenfold index is the first eigenvector
eigenfold_index <- decomposition$rotation[, 1]
# scale for interpretability, but original values are used
# for, e.g. comparison with fetal data
eigenfold_index_scaled <- 2 * ((eigenfold_index - min(eigenfold_index))/(max(eigenfold_index) -
    min(eigenfold_index))) - 1
# Note the bipolar distribution of eigenfold index (EFI)
# values
hist(eigenfold_index_scaled)
```



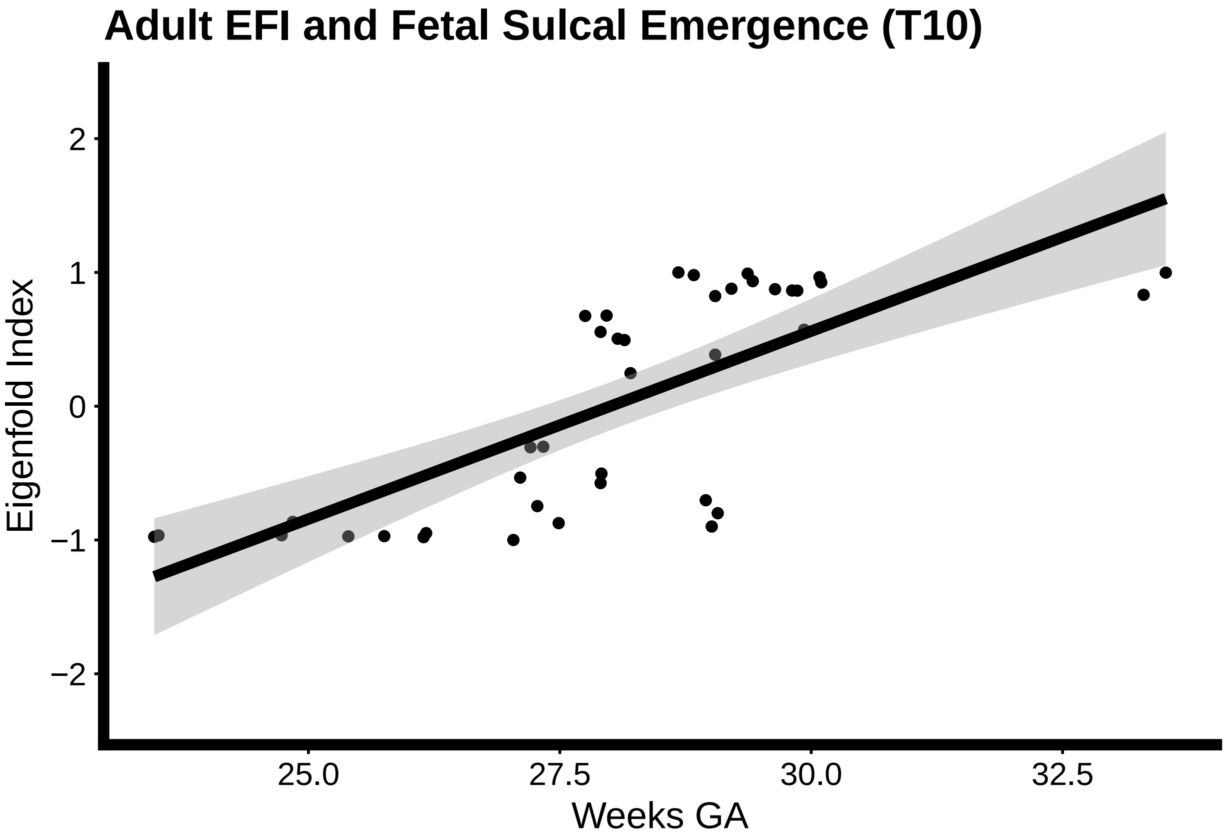
```
# Subject-level EFI is computed as coherence with
# group-level EFI by correlating group-level EFI with
# subject SPN rows
subject_efi <- data.frame(array(0, c(dim(subject_spns)[3], 40)))  #subjects X sulci
colnames(subject_efi) <- colnames(average_depth)
# loop through subjects and sulci
for (i in 1:dim(subject_spns)[3]) {
    for (j in 1:40) {
        subject_efi[i, j] <- cor(subject_spns[j, , i], eigenfold_index)
    }
}

# in example
head(subject_efi[, c(1:4)])  #4 example sulci
```

```
##   Left Lateral Fissure Right Lateral Fissure Left Anterior Cingulate Sulcus
## 1           -0.7374892            -0.7524390                      0.7688223
## 2           -0.6977873            -0.7885214                      0.8168705
## 3           -0.8830229            -0.8921808                     -0.5229639
## 4           -0.8005859            -0.7420807                      0.5307673
## 5           -0.4762979            -0.8576573                      0.3909434
## 6           -0.6981363            -0.7375250                     -0.7093351
##   Right Anterior Cingulate Sulcus
## 1                      0.30354694
## 2                      0.22562846
## 3                      0.63794560
## 4                    -0.01755823
## 5                     0.66312915
## 6                     0.68297469
```

Compare to fetal sulcation milestones

Next we will compare group-level EFI with fetal sulcation milestones, namely when sulci begin to visibly emerge (T10) and when they are half folded (T50). Additionally, we will compare inter-individual variability of subject-level EFI with T50.

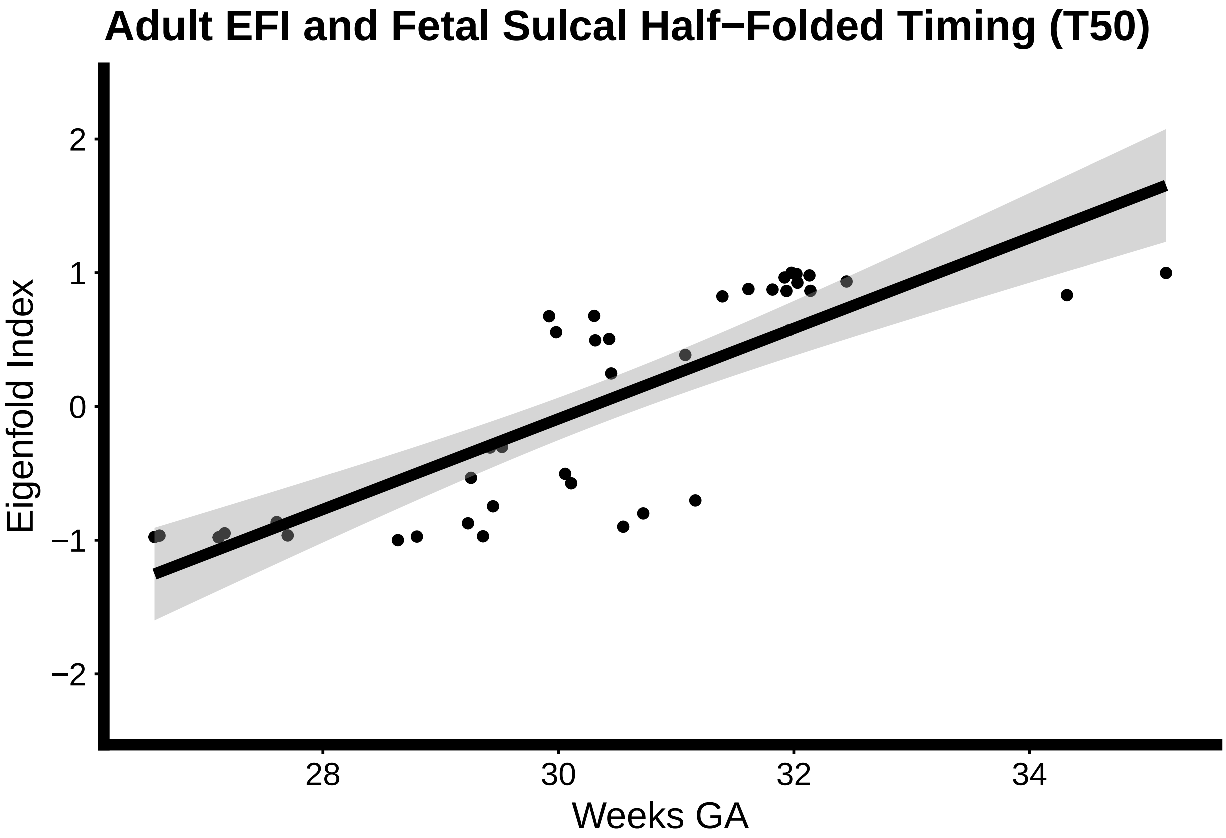
```
t10 <- data.frame(read_xlsx(path = "~/Downloads/spn_analysis/Snyder_Neuron_Supplementary_Tables.xlsx",
    sheet = "Table 5, Fetal Sulcation"))[, 2]
t50 <- data.frame(read_xlsx(path = "~/Downloads/spn_analysis/Snyder_Neuron_Supplementary_Tables.xlsx",
    sheet = "Table 5, Fetal Sulcation"))[, 3]
efi_variability <- data.frame(read_xlsx(path = "~/Downloads/spn_analysis/Snyder_Neuron_Supplementary_Ta
    sheet = "Table 4, Clusters, EFI, EFI Var"))[, 4]

# first show for when sulci first emerge
ggplot(data.frame(x = t10, y = eigenfold_index_scaled), aes(x = x,
    y = y)) + geom_point() + stat_smooth(method = "lm", size = 2,
    color = "black") + theme_cowplot() + theme(axis.line = element_line(size = 2)) +
    xlab("Weeks GA") + ylab("Eigenfold Index") + ylim(-2.3, 2.3) +
    ggtitle("Adult EFI and Fetal Sulcal Emergence (T10)")
```



```
cor(eigenfold_index, t10)
```

```
## [1] 0.7321574
```

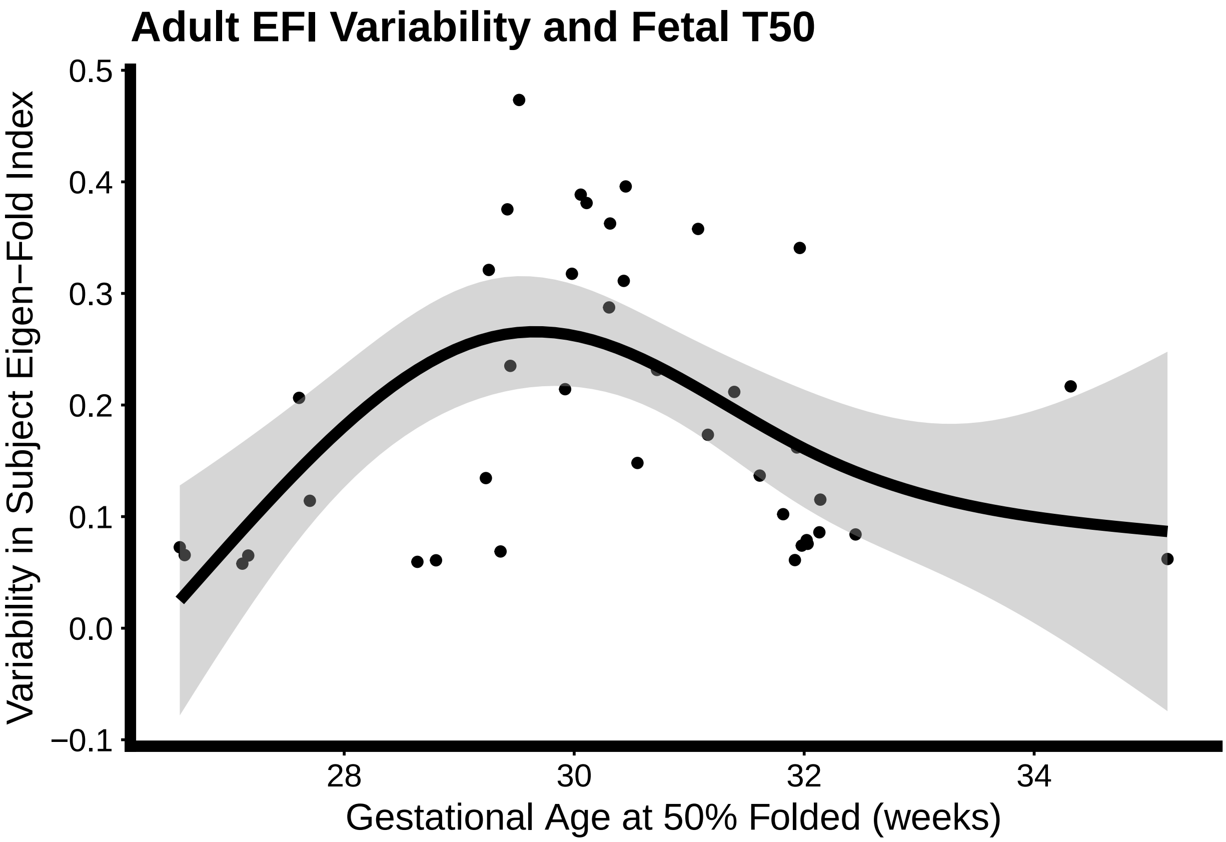
```
# and now a slightly stronger correlation for when sulci
# reach 50% of their final absolute curvature / folding
ggplot(data.frame(x = t50, y = eigenfold_index_scaled), aes(x = x,
    y = y)) + geom_point() + stat_smooth(method = "lm", size = 2,
    color = "black") + theme_cowplot() + theme(axis.line = element_line(size = 2)) +
    xlab("Weeks GA") + ylab("Eigenfold Index") + ylim(-2.3, 2.3) +
    ggtitle("Adult EFI and Fetal Sulcal Half-Folded Timing (T50)")
```



```
cor(eigenfold_index, t50)
```

```
## [1] 0.8061969
```

```
# Finally, a Generalized Addiditve Model (balanced for
# over/underfitting of number of knot points) relating
# adult interindividual variability in sulcal EFI with
# fetal T50
fetal_variability.df <- data.frame(x = t50, y = efi_variability)
ggplot(fetal_variability.df, aes(x = x, y = y)) + geom_point() +
    geom_smooth(method = "gam", formula = y ~ s(x, k = 4), color = "black",
        size = 2) + theme_cowplot() + xlab("Gestational Age at 50% Folded (weeks)") +
    ylab("Variability in Subject Eigen-Fold Index") + theme(axis.line = element_line(size = 2)) +
    ggtitle("Adult EFI Variability and Fetal T50")
```



```
# evaluate significance
gam_model <- gam(formula = y ~ s(x, k = 4), data = fetal_variability.df)
summary(gam_model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(x, k = 4)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.19215    0.01635   11.75 6.56e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##        edf Ref.df     F p-value
## s(x) 2.818  2.976 7.138  0.0015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.316   Deviance explained = 36.5%
## GCV = 0.011826  Scale est. = 0.010697  n = 40
```