Supplementary Software

STAR RNA-seq Data Alignment 3
Knight ADRC and DIAN 3
Spike-In Alignment 6
MSBB 8
Calculating Transcript Integrity Number 12
CircRNA Calling 13
DCC and circRNA_finder calling in Knight ADRC and DIAN 13
DCC Calling in Spike-In 17
DCC Calling in MSBB 17
Salmon Calling of Linear RNAs 18
Knight ADRC 18
MSBB 19
R Libraries 21
Dataset Processing and Differential Expression Analyses 22
Spike-In Processing 22
Knight ADRC 37
DIAN 44
MSBB 52
Regression-Based circRNA Association Independence Analyses 59
Independence from AD-associated cognate linear mRNA changes 59
Independence from AD-associated cell-type proportion changes 64
Differential Expression Meta-Analyses 69
Pre-symptomatic Bootstrap Correlation Distribution Analyses 78
Proportion of Variation Explained Analyses 86
ROCC and AUC Analyses 98
Co-expression Network Analyses 105
Processing Code 105
Template files 131
R Script 131
Bash Script 133
Cluster Processing Script 134
MicroRNA Binding Site Prediction Analyses 134

Overlap and other calculations 137

STAR RNA-seq Data Alignment

Knight ADRC and DIAN

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: STAR Alignment of RNA-seq Data

#####Generate STAR Reference files based on Gencode release v26 - GRCh38
```{bash}
#!/bin/bash

# Knight ADRC Parietal

# Create Star Indices - via Star Manual and Gencode
# Reference files
# GRCh38 annotation file - wget
ftp://ftp.sanger.ac.uk/pub/gencode/Gencode_human/release_26/gencode.v26.primary_assembly.annotation.gtf.gz
# GRCh38 fasta file - wget
ftp://ftp.sanger.ac.uk/pub/gencode/Gencode_human/release_26/GRCh38.primary_assembly.genome.fa.gz

#Knight ADRC Parietal Star code 150n reads

#read length for Knight ADRC Parietal RNASeq as per FastQC is 151, so sjdbOverhang = 150

mkdir GrCh38_150n

/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 22 \
--runMode genomeGenerate \
--genomeDir "$(pwd)/GrCh38_150n" \
--genomeFastaFiles /40/AD/Expression/KnightADRCParietal/06.-References/GRCh38.primary_assembly.genome.fa \
--sjdbGTFfile /40/AD/Expression/KnightADRCParietal/06.-References/gencode.v26.primary_assembly.annotation.gtf \
--sjdbOverhang 150 \
--outFileNamePrefix "$(pwd)/GrCh38_150n"


```
```

#####Align 3 times: paired, R1, R2
```{bash}
#!/bin/bash

JAVA="/usr/java/jre1.8.0_91/bin/java"
JAVAOPTS="-Xms4g -Xmx8g -XX:ParallelGCThreads=1 -Djava.io.tmpdir=/data/tmp"
PICARD="/usr/local/genome/picard-2.10.3/picard.jar"

#Example code to run:

##LOAD GENOME FIRST##
#/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --genomeDir
/40/AD/Expression/KnightADRCParietal/06.-References/GrCh38_150n --runThreadN 22 --genomeLoad
Remove
#/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --genomeDir
/40/AD/Expression/KnightADRCParietal/06.-References/GrCh38_150n --runThreadN 22 --genomeLoad
LoadAndExit

#parallel -j 5 --load 50 ./Alignment_KnightADRCParietal.sh {} :::
/40/AD/Expression/KnightADRCParietal/02.-Processed_Data/01.HISeq/02.-merge_Bam/*.bam


bam=${1##*/}

participant=`echo $bam | rev | cut -d. -f2- | rev`

logfile=log.${participant}.txt

mate1=${participant}_R1.fastq                    #adding _R1.fastq to name file
mate2=${participant}_R2.fastq                    #adding _R2.fastq to name file

star_out_dir=/home/dubeu/Synapse/align_individuals/align1 #set it the output directory

#collecting the adapter sequencing for each individual

R1=`grep -w $participant /40/AD/Expression/KnightADRCParietal/02.-Processed_Data/01.HISeq/03.-
STAR_align/HiSeq_adapter_post_swap.txt | awk '{print $2}'`
R2=`grep -w $participant /40/AD/Expression/KnightADRCParietal/02.-Processed_Data/01.HISeq/03.-
STAR_align/HiSeq_adapter_post_swap.txt | awk '{print $3}'`

echo "Convert $file to fastq format"
echo "$participant -> ($mate1, $mate2)"
```

```
java -jar $PICARD RevertSam VALIDATION_STRINGENCY=LENIENT QUIET=true COMPRESSION_LEVEL=0 I="${1}" SORT_ORDER=queryname OUTPUT=/dev/stdout | java -jar $PICARD SamToFastq I=/dev/stdin FASTQ=$mate1 SECOND_END_FASTQ=$mate2

# run STAR both mates
/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 10 --genomeDir /40/AD/Expression/KnightADRCParietal/06.-References/GrCh38_150n --readFilesIn $mate1 $mate2 --outSAMtype BAM SortedByCoordinate --outFileNamePrefix $star_out_dir/${participant}_circRNA_Align1_ --outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --alignSJDBoverhangMin 15 --seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1 --outFilterMatchNmin 1 --outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --chimScoreSeparation 10 --chimJunctionOverhangMin 15 --clip3pAdapterSeq $R1 $R2 --genomeLoad LoadAndKeep --limitBAMsortRAM 50000000000


 # run STAR R1
/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 10 --genomeDir /40/AD/Expression/KnightADRCParietal/06.-References/GrCh38_150n --readFilesIn $mate1 --outSAMtype None --outFileNamePrefix $star_out_dir/${participant}_${mate1}_circRNA_Align1_ --outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --alignSJDBoverhangMin 15 --seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1 --outFilterMatchNmin 1 --outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --chimScoreSeparation 10 --chimJunctionOverhangMin 15 --clip3pAdapterSeq $R1 $R2 --genomeLoad LoadAndKeep --limitBAMsortRAM 50000000000

 # run STAR R2
/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 10 --genomeDir /40/AD/Expression/KnightADRCParietal/06.-References/GrCh38_150n --readFilesIn $mate2 --outSAMtype None --outFileNamePrefix $star_out_dir/${participant}_${mate2}_circRNA_Align1_ --outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --alignSJDBoverhangMin 15 --seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1 --outFilterMatchNmin 1 --outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --chimScoreSeparation 10 --chimJunctionOverhangMin 15 --clip3pAdapterSeq $R1 $R2 --genomeLoad LoadAndKeep --limitBAMsortRAM 50000000000

${JAVA} ${JAVAOPTS} -jar ${PICARD} BuildBamIndex I=$star_out_dir/${participant}_circRNA_Align1_Aligned.sortedByCoord.out.bam

rm $mate1 $mate2

exit

##Remove Genome afterwards##
```

```
#/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --genomeDir
/40/AD/Expression/KnightADRCParietal/06.-References/GrCh38_150n --runThreadN 22 --genomeLoad
Remove

```

Spike-In Alignment

```
JAVA="/scratch/budde/GATK_pipeline/Programs/jre1.8.0_91/bin/java"
JAVAOPTS="-Xms4g -Xmx8g -XX:ParallelGCThreads=4 -Djava.io.tmpdir=/data/tmp"
PICARD="/scratch/dubeu/SPIKEIN/REF/picard-2.10.3/picard.jar"

##Generate STAR INDEX
#/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 22 --runMode
genomeGenerate --genomeDir "$(pwd)/SPIKEIN_150n" --genomeFastaFiles
/40/AD/Expression/2017_03_MendelianVsSporadics/06.-References/spike-in/ERCC92b.fa --sjdbGTFfile
/40/AD/Expression/2017_03_MendelianVsSporadics/06.-References/spike-in/ERCC92b.gtf --
sjdbOverhang 150 --outFileNamePrefix "$(pwd)/SPIKEIN_150n"

#Example code to run

cd /scratch/dubeu/SPIKEIN/

if [[ ! -f ${BAMFILE##*/} ]] ; then
   exit
fi

bam=${BAMFILE##*/}

subject=`echo $bam | rev | cut -d. -f2- | rev`

logfile=log.${subject}.txt                                    #add log.txt

mate1=${subject}_R1.fastq                                     #adding _R1.fastq to name file:
mate2=${subject}_R2.fastq                                     #adding _R2.fastq to name file:
star_out_dir=/scratch/dubeu/SPIKEIN/align_individuals #set it the output directory

#collecting the adapter sequencing for each individual

R1=`grep -w $subject /scratch/dubeu/SPIKEIN/REF/HiSeq_adapter.txt | awk '{print $2}'`
R2=`grep -w $subject /scratch/dubeu/SPIKEIN/REF/HiSeq_adapter.txt | awk '{print $3}'`

echo "Convert $file to fastq format"
echo "$subject -> ($mate1, $mate2)"

if [[ -f $mate1 ]] ; then
```

```
   exit
fi

if [[ -f $mate2 ]] ; then
   exit
fi

${JAVA} ${JAVAOPTS} -jar $PICARD RevertSam VALIDATION_STRINGENCY=LENIENT QUIET=true
COMPRESSION_LEVEL=0 I="${bam}" SORT_ORDER=queryname OUTPUT=/dev/stdout | $JAVA -jar
$PICARD SamToFastq I=/dev/stdin FASTQ=$mate1 SECOND_END_FASTQ=$mate2


rm ${BAMFILE}

# run STAR
/scratch/dubeu/SPIKEIN/REF/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 4 --genomeDir
/scratch/dubeu/SPIKEIN/REF/SPIKEIN_150n --readFilesIn $mate1 $mate2 --outSAMtype BAM
SortedByCoordinate --outFileNamePrefix $star_out_dir/${subject}_circRNA_Align1_SPIKEIN_ --
outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --alignSJDBoverhangMin 15 --
seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1 --outFilterMatchNmin 1 --
outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --chimScoreSeparation 10 --
chimJunctionOverhangMin 15 --clip3pAdapterSeq $R1 $R2 --genomeLoad LoadAndKeep --
limitBAMsortRAM 50000000000 --alignTranscriptsPerReadNmax 100000

/scratch/dubeu/SPIKEIN/REF/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 4 --genomeDir
/scratch/dubeu/SPIKEIN/REF/SPIKEIN_150n --readFilesIn $mate1 --outSAMtype None --
outFileNamePrefix $star_out_dir/${subject}_${mate1}_circRNA_Align1_SPIKEIN_ --
outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --alignSJDBoverhangMin 15 --
seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1 --outFilterMatchNmin 1 --
outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --chimScoreSeparation 10 --
chimJunctionOverhangMin 15 --clip3pAdapterSeq $R1 $R2 --genomeLoad LoadAndKeep --
limitBAMsortRAM 50000000000 --alignTranscriptsPerReadNmax 100000

rm $mate1

/scratch/dubeu/SPIKEIN/REF/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 4 --genomeDir
/scratch/dubeu/SPIKEIN/REF/SPIKEIN_150n --readFilesIn $mate2 --outSAMtype None --
outFileNamePrefix $star_out_dir/${subject}_${mate2}_circRNA_Align1_SPIKEIN_ --
outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --alignSJDBoverhangMin 15 --
seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1 --outFilterMatchNmin 1 --
outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --chimScoreSeparation 10 --
chimJunctionOverhangMin 15 --clip3pAdapterSeq $R1 $R2 --genomeLoad LoadAndKeep --
limitBAMsortRAM 50000000000 --alignTranscriptsPerReadNmax 100000
```

rm $mate2

cd /scratch/dubeu/SPIKEIN/align_individuals

${JAVA} ${JAVAOPTS} -jar ${PICARD} BuildBamIndex
I=$star_out_dir/${subject}_circRNA_Align1_SPIKEIN_Aligned.sortedByCoord.out.bam

exit

MSBB

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: STAR Alignment of RNA-seq Data

#####Generate STAR Reference files based on Gencode release v26 - GRCh38
```{bash}
#!/bin/bash

# MSBB

# Create Star Indices - via Star Manual and Gencode
# Reference files
# GRCh38 annotation file - wget
ftp://ftp.sanger.ac.uk/pub/gencode/Gencode_human/release_26/gencode.v26.primary_assembly.annotation.gtf.gz
# GRCh38 fasta file - wget
ftp://ftp.sanger.ac.uk/pub/gencode/Gencode_human/release_26/GRCh38.primary_assembly.genome.fa.gz

#read length for MSBB RNASeq is 101, so sjdbOverhang = 100

mkdir GrCh38_100n

/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 22 \
--runMode genomeGenerate \
--genomeDir "$(pwd)/GrCh38_100n" \
--genomeFastaFiles /40/AD/Expression/AMP_AD/06.-References/GRCh38.primary_assembly.genome.fa \
```
```

```
--sjdbGTFfile /40/AD/Expression/AMP_AD/06.-
References/gencode.v26.primary_assembly.annotation.gtf \
--sjdbOverhang 100 \
--outFileNamePrefix "$(pwd)/GrCh38_100n"
```

#####Unalign, merge, and align MSBB

```{bash}
#!/bin/bash

JAVA="/usr/java/jre1.8.0_91/bin/java"
JAVAOPTS="-Xms4g -Xmx8g -XX:ParallelGCThreads=1 -Djava.io.tmpdir=/data/tmp"
PICARD="/usr/local/genome/picard-2.10.3/picard.jar"

#Prior to runs, navigate to directory and load the genome index using STAR#
#Check to see if it is already loaded -> /home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --
genomeDir /40/AD/Expression/AMP_AD/06.-References/GrCh38_100n --runThreadN 22 --genomeLoad
Remove
#Verify it has loaded into memory - ipcs
#/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --genomeDir
/40/AD/Expression/AMP_AD/06.-References/GrCh38_100n --runThreadN 22 --genomeLoad
LoadAndExit

#Example code to run
#parallel -j 50 --delay 120 --load 50 ./ReAlignAMP_AD_STAR1Align.sh {} :::
/30/AD/Expression/Public_Datasets/AMP/Mount_Sinai_RNA-
seq/2016.12.release/sequences/BM44/*.bam

bam=${1##*/}

path="$(dirname "${1}")"

partial=${bam##*/}                                    #file is named as an example
BM_10_755.unmapped.fq

subject=`echo $partial | rev | cut -d. -f5- | rev`   #BM_10_755

bam_final="$path/$bam"

fastq_final="$(ls "${path}/${subject}.unmapped"*)"

mkdir /home/dubeu/Synapse/AMP_AD/Revert/${subject}

#cp "${fastq_temp}" /home/dubeu/Synapse/AMP_AD/Revert/${subject}/
```

```
#mv "/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}.unmapped"*
/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}.unmapped.fq.gz

#fastq_final="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}.unmapped.fq.gz"

#Extract from @RG    ID:BM_22_53    PL:ILLUMINA    PU:HiSeq2500    LB:BM_22_53    DS:rnaseq
SM:BM_22_53    CN:MSSM

ID="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=ID:)[^\t]*')"
PU="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=PU:)[^\t]*')"
PL="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=PL:)[^\t]*')"
LB="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=LB:)[^\t]*')"
DS="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=DS:)[^\t]*')"
SM="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=SM:)[^\t]*')"
CN="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=CN:)[^\t]*')"
ID="$(samtools view -H "$bam_final" | grep '@RG' | grep -oP '(?<=ID:)[^\t]*')"

# Following GATK tutorial - http://gatkforums.broadinstitute.org/firecloud/discussion/6484/

#Convert unmapped fastq to unaligned bam

java -Xmx8G -jar ${PICARD} FastqToSam \
        FASTQ="$fastq_final" \
        OUTPUT="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}_revert_unmapped.ba
m" \
        READ_GROUP_NAME=${ID} \
        PLATFORM=${PL} \
        PLATFORM_UNIT=${PU} \
        LIBRARY_NAME=${LB} \
        DESCRIPTION=${DS} \
        SAMPLE_NAME=${SM} \
        SEQUENCING_CENTER=${CN}


#Convert mapped bam to unaligned bam
# To determine what attributes to clear, use "samtools view BM_22_53.accepted_hits.sort.coord.bam |
cut -f 12- | tr '\t' '\n' | cut -d ':' -f 1 | awk '{ if(!x[$1]++) { print }}'"


java -Xmx8G -jar ${PICARD} RevertSam \
  I="$bam_final" \
  O="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}_revert_mapped.bam" \
  SANITIZE=true \
  MAX_DISCARD_FRACTION=0.005 \
```

```
   ATTRIBUTE_TO_CLEAR=NH \
   ATTRIBUTE_TO_CLEAR=HI \
   ATTRIBUTE_TO_CLEAR=nM \
   ATTRIBUTE_TO_CLEAR=AS \
   ATTRIBUTE_TO_CLEAR=XS \
   SORT_ORDER=queryname \
   RESTORE_ORIGINAL_QUALITIES=true \
   REMOVE_DUPLICATE_INFORMATION=true \
   REMOVE_ALIGNMENT_INFORMATION=true

#Merge bam files together

java -Xmx8G -jar ${PICARD} MergeSamFiles \
        I="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}_revert_unmapped.bam" \
        I="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}_revert_mapped.bam" \
        O="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}_revert_final.bam" \
        SORT_ORDER=queryname

rm /home/dubeu/Synapse/AMP_AD/Revert/${subject}/*_revert_unmapped.bam
rm /home/dubeu/Synapse/AMP_AD/Revert/${subject}/*_revert_mapped.bam

exit

#Align with Star while converting BAM on the fly
# via - http://gatkforums.broadinstitute.org/gatk/discussion/6483

#Set Star Output directory

star_out_dir=/home/dubeu/Synapse/AMP_AD/align_individuals/BM_10 #set it the output directory

## Will use generic Illumina adapter sequence for trimming from
(https://support.illumina.com/bulletins/2016/12/what-sequences-do-i-use-for-adapter-trimming.html)

R1="AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"

#conversion to fastq

# run STAR Align 1
#convert to fastq on the fly
/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --runThreadN 10 --genomeDir
/40/AD/Expression/AMP_AD/06.-References/GrCh38_100n --readFilesIn <(${JAVA} -
Dsamjdk.buffer_size=131072 \
   -Dsamjdk.compression_level=1 \
   -XX:GCTimeLimit=50 \
```

```
   -XX:GCHeapFreeLimit=10 \
   -Xmx128m \
   -jar ${PICARD} SamToFastq \
   I="/home/dubeu/Synapse/AMP_AD/Revert/${subject}/${subject}_revert_final.bam" \
   FASTQ=/dev/stdout) --outSAMtype BAM SortedByCoordinate --outFileNamePrefix
$star_out_dir/${subject}_ --outSJfilterOverhangMin 15 15 15 15 --alignSJoverhangMin 15 --
alignSJDBoverhangMin 15 --seedSearchStartLmax 30 --outFilterMultimapNmax 20 --outFilterScoreMin 1
--outFilterMatchNmin 1 --outFilterMismatchNmax 2 --chimSegmentMin 15 --chimScoreMin 15 --
chimScoreSeparation 10 --chimJunctionOverhangMin 15 --clip3pAdapterSeq ${R1} --genomeLoad
LoadAndKeep --limitBAMsortRAM 30000000000


${JAVA} ${JAVAOPTS} -jar ${PICARD} BuildBamIndex
I=$star_out_dir/${subject}_Aligned.sortedByCoord.out.bam

##Remove Genome afterwards##
#/home/dubeu/Synapse/STAR-2.5.3a/bin/Linux_x86_64/STAR --genomeDir
/40/AD/Expression/AMP_AD/06.-References/GrCh38_100n --runThreadN 22 --genomeLoad Remove

exit
```

Calculating Transcript Integrity Number

```
#!/bin/bash


#Example code to run


#test - parallel -j 1 ./TIN_process.sh {} ::: "/30/AD/Expression/Public_Datasets/AMP/Mount_Sinai_RNA-
seq/2016.12.release/sequences/BM22/hB_RNA_12171.accepted_hits.sort.coord.bam"
#parallel -j 50 --delay 60 --load 35 ./TIN_Process_Parietal.sh {} :::
/home/dubeu/Synapse/align_individuals/align1/*.cram


#converting bam to fasta

# Converts .bam file to .cram format using the following reference
REF="/40/AD/Expression/ KnightADRCParietal /06.-References/GRCh38.primary_assembly.genome.fa"

bamfile=${1##*/}

partial=${bamfile##*/}                                  #file is named as an example
subject=`echo $partial | rev | cut -d. -f2- | rev`
```

```
samtools view -b -T "${REF}" -@ 20 $1 > $subject.sorted.bam

samtools index $subject.sorted.bam

# perl script from: https://github.com/ExpressionAnalysis/ea-utils/blob/master/clipper/gtf2bed
#./gtf2bed.perl <(grep basic /40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26.primary_assembly.annotation.gtf) >
gencode.v26.primary_assembly.annotation_basiconly.bed

#./gtf2bed.perl <(grep basic /40/AD/Expression/ KnightADRCParietal /06.-
References/gencode.v26.primary_assembly.annotation.gtf | grep protein_coding) >
gencode.v26.primary_assembly.annotation_basiconly_proteincoding.bed

tin.py -i $subject.sorted.bam -r
gencode.v26.primary_assembly.annotation_basiconly_proteincoding.bed

rm $subject.sorted.bam
rm $subject.sorted.bam.bai

exit
```

CircRNA Calling

DCC and circRNA_finder calling in Knight ADRC and DIAN

```
##########################################
## Dube et al., 2019 - circRNAs in AD ##
##########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: CircRNA Calling using DCC - KnightADRCParietal

####Calling circRNAs using DCC software: refer to tutorial - (https://github.com/dieterich-lab/DCC)

#####Prepare filtering files

.gft format, Genome build 38:
        USCC - Repeat masker
        UCSC - Simple Repeats

        Cat the files together

#####Generate input files

```{bash}
```
```

ls /home/dubeu/Synapse/align_individuals/*_Aligned.sortedByCoord.out.bam > bam_files

ls /home/dubeu/Synapse/align_individuals/*R1.fastqChimeric.out.junction > mate1

ls /home/dubeu/Synapse/align_individuals/*R2.fastqChimeric.out.junction > mate2

ls /home/dubeu/Synapse/align_individuals/*_Chimeric.out.junction > samplesheet
```

#####Run DCC

```{bash}
/home/dubeu/.local/bin/DCC @DCC_InputFiles/samplesheet -mt1 @DCC_InputFiles/mate1 -mt2 @DCC_InputFiles/mate2 -T 20 -D -R /40/AD/Expression/KnightADRCParietal/06.-References/GRCh38_Repeats_simpleRepeats_RepeatMasker.gtf -an /40/AD/Expression/KnightADRCParietal/06.-References/gencode.v26.primary_assembly.annotation.gtf -Pi -F -M -Nr 1 1 -fg -G -A /40/AD/Expression/KnightADRCParietal/06.-References/GRCh38.primary_assembly.genome.fa -B @DCC_InputFiles/bam_files
```

####Calling circRNAs using circRNA_finder software
```{bash}
#Run processing command in directory containing STAR alignment output
./postProcessStarAlignment.pl /home/dubeu/Synapse/align_individuals/align1/ /home/dubeu/Synapse/circRNA_finder/
```

####Check for Correlation between DCC and circRNA_finder software

```{r}

############################################################
############################################################
##### Filter the circRNAs based on linear ratio ########
############################################################
############################################################

##Adapted these functions from (https://github.com/dieterich-lab/CircTest) to run in parallel


circ_max_perc <- function(circ=circ,linear=linear,Nreplicates=3){
  # convert to vector
  circ = as.numeric(circ)
  linear = as.numeric(linear)

```
  if(length(circ) != length(linear)){
    stop ('Number of samples in circRNA is not equal to Hostgene.')
  }
  Ngroups = length(circ)/Nreplicates
  # calculate percentage
  circ_sum = unname(tapply(circ, (seq_along(1:length(circ))-1) %/% Nreplicates, sum))
  linear_sum = unname(tapply(linear, (seq_along(1:length(linear))-1) %/% Nreplicates, sum))
  perc = max(circ_sum / (circ_sum+linear_sum),na.rm=T)
  return(perc)
}

circFILTER <- function(i, circ, linear, Nreplicates, filter.sample, filter.count, percentage,
circle_description=c(1:3)) {

  del_row=c()

  if (sum(circ[i,-circle_description]>=filter.count)<filter.sample | circ_max_perc(circ[i,-
circle_description], linear[i,-circle_description],Nreplicates=Nreplicates)<percentage) {
          del_row = c(del_row,i)
  }

          del_row

}

library(parallel)

#Generate the DCC files

##Load in circRNA count data generated by DCC##

circCoordinates_parietalCorr <- read.table("/home/dubeu/Synapse/DCC/CircCoordinates", sep="\t",
header=TRUE, stringsAsFactors = FALSE)
CircCount_parietalCorr <- read.table("/home/dubeu/Synapse/DCC/CircRNACount", sep="\t",
header=TRUE, stringsAsFactors = FALSE)
LinearCount_parietalCorr <- read.table("/home/dubeu/Synapse/DCC/LinearCount", sep="\t",
header=TRUE, stringsAsFactors = FALSE)

##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietalCorr)
```

```
circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietalCorr, LinearCount_parietalCorr,
Nreplicates = 1, filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietalCorr <- CircCount_parietalCorr[-circfiltered,]

circCoordinates_parietalCorr_filter <- circCoordinates_parietalCorr[-circfiltered,]


##load parietalCorr phenotype / covariate information
coldata_parietalCorr <- read.csv(file="/40/AD/Expression/KnightADRCParietal/03.-
phenotype/Pheno.csv", header=TRUE, na.strings = "NA")

#Set rownames
rownames(coldata_parietalCorr) <- coldata_parietalCorr$ID_RNAseq


parietalCorr_circCalling_correlation <- data.frame()

for (i in rownames(coldata_parietalCorr)) {
circDCC <- cbind(circCoordinates_parietalCorr_filter, CircCount_parietalCorr)
colnames(circDCC) <- gsub("_circRNA_Align1_Chimeric.out.junction","", colnames(circDCC))
circFinder <- read.table(paste0("/home/dubeu/Synapse/circRNA_finder/",gsub("VY.","VY-
",i),"_circRNA_Align1_filteredJunctions.bed"), header=FALSE)
colnames(circFinder) <- c("Chr","Start","End","Score",i,"Strand")
circFinder$Chr <- as.character(circFinder$Chr)
circFinder$Strand <- as.character(circFinder$Strand)
##Update start position
circFinder$Start <- as.numeric(circFinder$Start) + 1
circDCC <- circDCC[,(colnames(circFinder)[c(1,2,3,6,5)])]
circCalling_correlation <- merge(circDCC,circFinder, by=c("Chr","Start","End","Strand"))

correlation <- cor.test(circCalling_correlation[,5],circCalling_correlation[,7])

parietalCorr_circCalling_correlation <- rbind(correlation$estimate,parietalCorr_circCalling_correlation)

}

rownames(parietalCorr_circCalling_correlation) <- rownames(coldata_parietalCorr)
colnames(parietalCorr_circCalling_correlation) <- "corr"

summary(parietalCorr_circCalling_correlation)
```

DCC Calling in Spike-In

#Note the absence of N (since this is stranded data)
#Note the -Pi for paired end data

/usr/bin/DCC @DCC_InputFiles/samplesheet -mt1 @DCC_InputFiles/mate1 -mt2 @DCC_InputFiles/mate2 -T 20 -D -an /40/AD/Expression/2017_03_MendelianVsSporadics/06.-References/spike-in/ERCC92b.gtf -Pi -F -M -Nr 1 1 -fg -G -A

DCC Calling in MSBB

##
Dube et al., 2019 - circRNAs in AD
##

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: CircRNA Calling using DCC - MSBB

#####Call CircRNAs - DCC
```{bash}
DCC:

#Download gtf files from UCSC, combined using cat to form /40/AD/Expression/AMP_AD/06.-References/GRCh38_Repeats_simpleRepeats_RepeatMasker.gtf

##Repeat below for all 4 regions of MSBB

cd /home/dubeu/Synapse/AMP_AD/DCC
mkdir DCC_InputFiles

ls /home/dubeu/Synapse/AMP_AD/align_individuals/BM10/*_Aligned.sortedByCoord.out.bam > DCC_InputFiles/bam_files
ls /home/dubeu/Synapse/AMP_AD/align_individuals/BM10/*Chimeric.out.junction > DCC_InputFiles/read
ls /home/dubeu/Synapse/AMP_AD/align_individuals/BM10/*_Chimeric.out.junction > DCC_InputFiles/samplesheet

# - N for non-stranded

DCC @DCC_InputFiles/samplesheet -mt1 @DCC_InputFiles/read -T 20 -D -N -R /40/AD/Expression/AMP_AD/06.-References/GRCh38_Repeats_simpleRepeats_RepeatMasker.gtf -an /40/AD/Expression/AMP_AD/06.-References/gencode.v26.primary_assembly.annotation.gtf -F -M -Nr 1 1 -fg -G -A /40/AD/Expression/AMP_AD/06.-References/GRCh38.primary_assembly.genome.fa -B @DCC_InputFiles/bam_files
```

Salmon Calling of Linear RNAs

Knight ADRC

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Linear mRNA Calling using Salmon - Knight ADRC Parietal


####Calling linear genes using Salmon
Followed the manual as provided here: salmon
manual(https://salmon.readthedocs.io/en/latest/salmon.html)
#####Generate the indices
Obtained the transcript fasta file from GENCODE V26 as with the STAR alignment above

######Generate quasi alignment index
```{bash}
/usr/local/genome/salmon-0.8.2/bin/salmon index --gencode -t
/40/AD/Expression/KnightADRCParietal/06.-References/gencode.v26.transcripts.fa -i
gencode.v26_salmon_transcripts_index_quasi --type quasi -k 31
```

```{bash}
#!/bin/bash

JAVA="/usr/java/jre1.8.0_91/bin/java"
JAVAOPTS="-Xms4g -Xmx8g -XX:ParallelGCThreads=1 -Djava.io.tmpdir=/data/tmp"
PICARD="/usr/local/genome/picard-2.10.3/picard.jar"

#Example code to run

#parallel -j 5 --delay 30 --load 50 ./Salmon_Alignment_quasi.sh {} :::
/40/AD/Expression/KnightADRCParietal/02.-Processed_Data/01.HISeq/02.-merge_Bam/*.bam

bam=${1##*/}

subject=`echo $bam | rev | cut -d. -f2- | rev`

logfile=log.${subject}.txt                    #add log.txt

mate1=${subject}_R1.fastq                     #adding _R1.fastq to name file
```
```

```
mate2=${subject}_R2.fastq                                   #adding _R2.fastq to name file

salmon_out_dir=/home/dubeu/Synapse/Salmon/quasi/ #set it the output directory

echo "Convert $file to fastq format"
echo "$subject -> ($mate1, $mate2)"

if [ -d "$salmon_out_dir/${subject}" ]; then
        echo "Already quantified"
        exit
else

        java -jar $PICARD RevertSam VALIDATION_STRINGENCY=LENIENT QUIET=true
COMPRESSION_LEVEL=0 I="${1}" SORT_ORDER=queryname OUTPUT=/dev/stdout | java -jar $PICARD
SamToFastq I=/dev/stdin FASTQ=$mate1 SECOND_END_FASTQ=$mate2

        mkdir $salmon_out_dir/${subject}/

        /usr/local/genome/salmon-0.8.2/bin/salmon quant -p 10 -i
/40/AD/Expression/KnightADRCParietal/06.-References/gencode.v26_salmon_transcripts_index_quasi -
l ISR -1 $mate1 -2 $mate2 -o $salmon_out_dir/${subject}/

        rm $mate1 $mate2

fi

exit
```

```
## MSBB

##########################################
## Dube et al., 2019 - circRNAs in AD ##
##########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Linear mRNA Calling using Salmon - MSBB


####Calling linear genes using Salmon
Followed the manual as provided here: salmon
manual(https://salmon.readthedocs.io/en/latest/salmon.html)
```

#####Generate the indices
Obtained the transcript fasta file from GENCODE V26 as with the STAR alignment above

######Generate quasi alignment index
```{bash}
/usr/local/genome/salmon-0.8.2/bin/salmon index --gencode -t /40/AD/Expression/AMP_AD/06.-References/gencode.v26.transcripts.fa -i gencode.v26_salmon_transcripts_index_quasi --type quasi -k 31
```

```{bash}
#!/bin/bash

JAVA="/usr/java/jre1.8.0_91/bin/java"
JAVAOPTS="-Xms4g -Xmx8g -XX:ParallelGCThreads=1 -Djava.io.tmpdir=/data/tmp"
PICARD="/usr/local/genome/picard-2.10.3/picard.jar"

#Example code to run
#parallel -j 8 --delay 30 --load 40 ./AMP_AD_SalmonQuasi_cram_36.sh {} ::: /30/AD/Expression/Public_Datasets/AMP/Mount_Sinai_RNA-seq/2016.12.release/sequences/BM36/*.bam

REF="/40/AD/Expression/AMP_AD/06.-References/GRCh38.primary_assembly.genome.fa"

bam=${1##*/}

partial=${bam##*/}                                    #file is named as an example BM_10_755.unmapped.fq

subject=`echo $partial | rev | cut -d. -f5- | rev`   #BM_10_755



#Set Salmon Output directory

salmon_out_dir=/home/dubeu/Synapse/AMP_AD/Salmon/quasi/BM36 #set it the output directory

mate1=${subject}_R1.fastq

if [ -d "$salmon_out_dir/${subject}" ]; then
        echo "Already quantified"
        exit
else
```

```
        samtools view -T "${REF}" -b "/30/AD/Expression/Public_Datasets/AMP/Mount_Sinai_RNA-seq/2016.12.release/sequences/Unmapped/BM_36/${subject}_revert_final.cram" | samtools bam2fq - > ${subject}_R1.fastq

        mkdir $salmon_out_dir/${subject}/

        /usr/local/genome/salmon-0.8.2/bin/salmon quant -p 10 -i /40/AD/Expression/AMP_AD/06.-References/gencode.v26_salmon_transcripts_index_quasi -l U -r $mate1 -o $salmon_out_dir/${subject}/

        rm $mate1

fi

exit
```

R Libraries

```
#Parallel Processing
library("BiocParallel")
register(MulticoreParam(50))

#Data processing
library(dplyr)
library(data.table)

#Differential expression analysis
library("DESeq2")

#MetaRNASeq
library(metaRNASeq)

#For ANCOVA

library(car)
library(compute.es)
library(effects)
library(multcomp)
library(pastecs)
library(WRS2)

library(VennDiagram)

#Relative Importance
library(relaimpo)
```

```
#linear Salmon analysis
library("tximport")
library("readr")
library("tximportData")

#Plotting
library(gridExtra)
library(cowplot)
library(ggpubr)
library(ggplot2)
```

Dataset Processing and Differential Expression Analyses

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: KnightADRC Parietal and DIAN Parietal Processing
```

Spike-In Processing

```
library(parallel)


############################################################
############################################################
##### Filter the circRNAs based on linear ratio ########
############################################################
############################################################



circ_max_perc <- function(circ=circ,linear=linear,Nreplicates=3){
  # convert to vector
  circ = as.numeric(circ)
  linear = as.numeric(linear)
  if( length(circ) != length(linear) ){
    stop ('Number of samples in circRNA is not equal to Hostgene.')
  }
  Ngroups = length(circ)/Nreplicates
  # calculate percentage
  circ_sum = unname(tapply(circ, (seq_along(1:length(circ))-1) %/% Nreplicates, sum ))
  linear_sum = unname(tapply(linear, (seq_along(1:length(linear))-1) %/% Nreplicates, sum ))
```

```
  perc = max(circ_sum / (circ_sum+linear_sum),na.rm=T)
  return(perc)
}

circFILTER <- function(i, circ, linear, Nreplicates, filter.sample, filter.count, percentage,
circle_description=c(1:3)) {

  del_row=c()

  if ( sum(circ[i,-circle_description]>=filter.count)<filter.sample | circ_max_perc(circ[i,-
circle_description], linear[i,-circle_description],Nreplicates=Nreplicates)<percentage) {
          del_row = c(del_row,i)
  }

          del_row

}



########################################
########################################
############ Parietal ################
########################################
########################################



circCoordinates_parietal <- read.table("/home/dubeu/Synapse/DCC/CircCoordinates", sep="\t",
header=TRUE, stringsAsFactors = FALSE)
CircCount_parietal <- read.table("/home/dubeu/Synapse/DCC/CircRNACount", sep="\t", header=TRUE,
stringsAsFactors = FALSE)
LinearCount_parietal <- read.table("/home/dubeu/Synapse/DCC/LinearCount", sep="\t", header=TRUE,
stringsAsFactors = FALSE)

nrow(CircCount_parietal)


### 0 percentage



##Filter the circRNAs based on linear ratios
```

```
rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 1, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 2, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 3, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)
```

```
##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 4, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 5, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


### 0.1 percentage

##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 0, filter.count = 3, percentage = 0.1, mc.cores = 50)
```

```
#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 1, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 2, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)
```

```
circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 4, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 5, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)
```

```
###Filter 0.01 percentage

##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 0, filter.count = 3, percentage = 0.01, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 1, filter.count = 3, percentage = 0.01, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 2, filter.count = 3, percentage = 0.01, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063
```

```
circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 3, filter.count = 3, percentage = 0.01, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 4, filter.count = 3, percentage = 0.01, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)
```

```
circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1, filter.sample = 5, filter.count = 3, percentage = 0.01, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal_filter <- CircCount_parietal[-circfiltered,]
nrow(CircCount_parietal_filter)




########################################
########################################
############ Spike In ################
########################################
########################################



circCoordinates_SPIKEIN <-
read.table("/home/dubeu/Synapse/align_individuals/align1_SPIKEIN/CHPC/CircCoordinates", sep="\t", header=TRUE, stringsAsFactors = FALSE)
CircCount_SPIKEIN <-
read.table("/home/dubeu/Synapse/align_individuals/align1_SPIKEIN/CHPC/CircRNACount", sep="\t", header=TRUE, stringsAsFactors = FALSE)
LinearCount_SPIKEIN <-
read.table("/home/dubeu/Synapse/align_individuals/align1_SPIKEIN/CHPC/LinearCount", sep="\t", header=TRUE, stringsAsFactors = FALSE)

nrow(CircCount_SPIKEIN)


### 0 percentage

##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 1, filter.count = 3, percentage = 0, mc.cores = 50)
```

```
#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 2, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 3, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)
```

```
circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 4, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 5, filter.count = 3, percentage = 0, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)



### 0.1 percentage

##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 0, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))
```

```
CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1,
filter.sample = 1, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1,
filter.sample = 2, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1,
filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)
```

```
#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 4, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 5, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


###Filter 0.1 percentage

##Filter the circRNAs based on linear ratios
```

```
rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 0, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 1, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1, filter.sample = 2, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
```

```
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1,
filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1,
filter.sample = 4, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063

circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_SPIKEIN)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_SPIKEIN, LinearCount_SPIKEIN, Nreplicates = 1,
filter.sample = 5, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-
lists/26540063
```

```
circfiltered <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_SPIKEIN_filter <- CircCount_SPIKEIN[-circfiltered,]
nrow(CircCount_SPIKEIN_filter)
```

Knight ADRC

##Knight ADRC Parietal Processing

```{r}

############################################################
############################################################
##### Filter the circRNAs based on linear ratio ########
############################################################
############################################################

##Adapted these functions from (https://github.com/dieterich-lab/CircTest) to run in parallel


circ_max_perc <- function(circ=circ,linear=linear,Nreplicates=3){
  # convert to vector
  circ = as.numeric(circ)
  linear = as.numeric(linear)
  if( length(circ) != length(linear) ){
    stop ('Number of samples in circRNA is not equal to Hostgene.')
  }
  Ngroups = length(circ)/Nreplicates
  # calculate percentage
  circ_sum = unname(tapply(circ, (seq_along(1:length(circ))-1) %/% Nreplicates, sum ))
  linear_sum = unname(tapply(linear, (seq_along(1:length(linear))-1) %/% Nreplicates, sum ))
  perc = max(circ_sum / (circ_sum+linear_sum),na.rm=T)
  return(perc)
}

circFILTER <- function(i, circ, linear, Nreplicates, filter.sample, filter.count, percentage,
circle_description=c(1:3)) {

  del_row=c()

  if ( sum(circ[i,-circle_description]>=filter.count)<filter.sample | circ_max_perc(circ[i,-
circle_description], linear[i,-circle_description],Nreplicates=Nreplicates)<percentage) {
          del_row = c(del_row,i)
  }
```

```
        del_row

}



##########################################
##########################################
## Discovery in Parietal from WUSM ##
##########################################
##########################################


##Load in circRNA count data generated by DCC##

circCoordinates_parietal <- read.table("/home/dubeu/Synapse/DCC/CircCoordinates", sep="\t",
header=TRUE, stringsAsFactors = FALSE)
CircCount_parietal <- read.table("/home/dubeu/Synapse/DCC/CircRNACount", sep="\t", header=TRUE,
stringsAsFactors = FALSE)
LinearCount_parietal <- read.table("/home/dubeu/Synapse/DCC/LinearCount", sep="\t", header=TRUE,
stringsAsFactors = FALSE)

##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietal)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietal, LinearCount_parietal, Nreplicates = 1,
filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)

circfiltered_parietal <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietal <- CircCount_parietal[-circfiltered_parietal,]

circCoordinates_parietal_filter <- circCoordinates_parietal[-circfiltered_parietal,]

#Remove extraneous columns

CircCount_parietal <- CircCount_parietal[,4:ncol(CircCount_parietal)]

##Adjust names
names(CircCount_parietal) <- gsub("_circRNA_Align1_Chimeric.out.junction", "",
names(CircCount_parietal))
```

```
##Collapse circRNA counts onto the gene level, removing those that are not_annotated
CircCount_parietal$Gene <- circCoordinates_parietal[rownames(CircCount_parietal),]$Gene
CircCount_parietal <- subset(CircCount_parietal, Gene != "not_annotated")
mydat.max_parietal <- aggregate(. ~ Gene, data = CircCount_parietal, sum)
rownames(mydat.max_parietal) <- mydat.max_parietal$Gene
mydat.max_parietal$Gene <- NULL
cts_parietal <- mydat.max_parietal
rownames(cts_parietal) <- paste0("circ", rownames(cts_parietal))


##load Parietal phenotype / covariate information
coldata_parietal <- read.csv(file="/40/AD/Expression/KnightADRCParietal/03.-phenotype/Pheno",
header=TRUE, na.strings = "NA")
#Restrict analyses to parietal tissues
coldata_parietal <- subset(coldata_parietal, brain_region == "parietal")


#Rename
coldata_parietal$ID_RNAseq <- gsub("VY-", "VY.", coldata_parietal$ID_RNAseq)

#Temporarily Assigning Ethnicity#
coldata_parietal$Ethnicity <- coldata_parietal$Ethnicity_temp

##Remove Consensus Outliers
coldata_parietal <- subset(coldata_parietal, ConsensusOutlier_RemoveBefore != 1)

#Set rownames
rownames(coldata_parietal) <- coldata_parietal$ID_RNAseq

cts_parietal <- cts_parietal[, rownames(coldata_parietal)]


#Set up factor levels
coldata_parietal$pool <- factor(coldata_parietal$pool, levels=c('1','2'))
coldata_parietal$GENDER <- factor(coldata_parietal$GENDER, levels=c('1','2'))
coldata_parietal$Ethnicity <- factor(coldata_parietal$Ethnicity, levels = c("EA", "AA"))
coldata_parietal$condition <- droplevels(coldata_parietal$condition)




####################
####################
## DE Analyses ##
```

```
###################
##################



##DE based on condition ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietal_condition_filter <- DESeqDataSetFromMatrix(countData = cts_parietal, colData =
coldata_parietal, design = ~ 1)

dds_parietal_condition_filter <- estimateSizeFactors(dds_parietal_condition_filter)

##subset based on covars so that none are missing

dds_parietal_condition_filter <- dds_parietal_condition_filter[
,!is.na(dds_parietal_condition_filter$AGE_at_death)]
dds_parietal_condition_filter <- dds_parietal_condition_filter[
,!is.na(dds_parietal_condition_filter$PMI)]
dds_parietal_condition_filter <- dds_parietal_condition_filter[
,!is.na(dds_parietal_condition_filter$PC1)]
dds_parietal_condition_filter <- dds_parietal_condition_filter[
,!is.na(dds_parietal_condition_filter$PC2)]

#Remove dementedcontrols
dds_parietal_condition_filter <- dds_parietal_condition_filter[,
!(dds_parietal_condition_filter$DementedControl == 1) ]


#Variable under invesitgation is at the end of the design formula
design(dds_parietal_condition_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool + GENDER
+ PC1 + PC2 + condition)

#Run the actual DESeq2 function
dds_parietal_final_condition_filter <- DESeq(dds_parietal_condition_filter, parallel = TRUE, betaPrior =
FALSE)

##Analyze results based on contrast
```

```
res_parietal_condition_filter <- results(dds_parietal_final_condition_filter, contrast=c("condition",
"LOAD", "CO"), alpha = 0.05)
summary(res_parietal_condition_filter)



##DE based on braak ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietal_braak_filter <- DESeqDataSetFromMatrix(countData = cts_parietal, colData =
coldata_parietal, design = ~ 1)

dds_parietal_braak_filter <- estimateSizeFactors(dds_parietal_braak_filter)

##subset based on covars so that none are missing

dds_parietal_braak_filter <- dds_parietal_braak_filter[ ,!is.na(dds_parietal_braak_filter$AGE_at_death)]
dds_parietal_braak_filter <- dds_parietal_braak_filter[ ,!is.na(dds_parietal_braak_filter$PMI)]
dds_parietal_braak_filter <- dds_parietal_braak_filter[ ,!is.na(dds_parietal_braak_filter$braak_final)]
dds_parietal_braak_filter <- dds_parietal_braak_filter[ ,!is.na(dds_parietal_braak_filter$PC1)]
dds_parietal_braak_filter <- dds_parietal_braak_filter[ ,!is.na(dds_parietal_braak_filter$PC2)]

#Remove dementedcontrols
dds_parietal_braak_filter <- dds_parietal_braak_filter[, !(dds_parietal_braak_filter$DementedControl
== 1) ]



##Only include LOAD and controls
dds_parietal_braak_filter <- dds_parietal_braak_filter[ ,dds_parietal_braak_filter$condition != "ADAD" ]
dds_parietal_braak_filter$condition <- droplevels(dds_parietal_braak_filter$condition)


design(dds_parietal_braak_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool + GENDER +
PC1 + PC2 + braak_final)


dds_parietal_final_braak_filter <- DESeq(dds_parietal_braak_filter, parallel = TRUE, betaPrior = FALSE)
```

```
res_parietal_braak_filter <- results(dds_parietal_final_braak_filter, alpha = 0.05)
summary(res_parietal_braak_filter)



##DE based on CDR ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietal_CDR_filter <- DESeqDataSetFromMatrix(countData = cts_parietal, colData =
coldata_parietal, design = ~ 1)

dds_parietal_CDR_filter <- estimateSizeFactors(dds_parietal_CDR_filter)

##subset based on covars so that none are missing

dds_parietal_CDR_filter <- dds_parietal_CDR_filter[ ,!is.na(dds_parietal_CDR_filter$AGE_at_death)]
dds_parietal_CDR_filter <- dds_parietal_CDR_filter[ ,!is.na(dds_parietal_CDR_filter$PMI)]
dds_parietal_CDR_filter <- dds_parietal_CDR_filter[ ,!is.na(dds_parietal_CDR_filter$CDR_e)]
dds_parietal_CDR_filter <- dds_parietal_CDR_filter[ ,!is.na(dds_parietal_CDR_filter$PC1)]
dds_parietal_CDR_filter <- dds_parietal_CDR_filter[ ,!is.na(dds_parietal_CDR_filter$PC2)]

#Remove dementedcontrols
dds_parietal_CDR_filter <- dds_parietal_CDR_filter[, !(dds_parietal_CDR_filter$DementedControl == 1) ]

##Only include LOAD and controls
dds_parietal_CDR_filter <- dds_parietal_CDR_filter[ ,dds_parietal_CDR_filter$condition != "ADAD" ]
dds_parietal_CDR_filter$condition <- droplevels(dds_parietal_CDR_filter$condition)

design(dds_parietal_CDR_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool + GENDER + PC1
+ PC2 + CDR_e)

dds_parietal_final_CDR_filter <- DESeq(dds_parietal_CDR_filter, parallel = TRUE, betaPrior = FALSE)

res_parietal_CDR_filter <- results(dds_parietal_final_CDR_filter, alpha = 0.05)
summary(res_parietal_CDR_filter)

##Test if swapping TIN for RIN impacts the observed effect size
dds_parietal_CDR_filter_RIN <- dds_parietal_CDR_filter
```

```
design(dds_parietal_CDR_filter_RIN) = formula( ~ PMI + QC.RIN + AGE_at_death + pool + GENDER + PC1
+ PC2 + CDR_e)

dds_parietal_final_CDR_filter_RIN <- DESeq(dds_parietal_CDR_filter_RIN, parallel = TRUE, betaPrior =
FALSE)

res_parietal_CDR_filter_RIN <- results(dds_parietal_final_CDR_filter_RIN, alpha = 0.05)
summary(res_parietal_CDR_filter_RIN)

cor.test(res_parietal_CDR_filter[GeneList_parietal_CDR,]$log2FoldChange,
res_parietal_CDR_filter_RIN[GeneList_parietal_CDR,]$log2FoldChange)

cor.test((res_parietal_CDR_filter[rownames(subset(res_parietal_CDR_filter_RIN[GeneList_parietal_CDR,
], pvalue < 0.05)),]$log2FoldChange), (subset(res_parietal_CDR_filter_RIN[GeneList_parietal_CDR,],
pvalue < 0.05)$log2FoldChange))

##########################################
##########################################
## Calculate Parietal Overlap ##
##########################################
##########################################


GeneList_parietal_condition <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_CDR <- rownames(subset(res_parietal_CDR_filter, padj < 0.05))

GeneList_parietal_braak <- rownames(subset(res_parietal_braak_filter, padj < 0.05))


grid.newpage()
draw.triple.venn(area1 = length(GeneList_parietal_condition), area2 = length(GeneList_parietal_CDR),
area3 = length(GeneList_parietal_braak), n12 = length(intersect(GeneList_parietal_condition,
GeneList_parietal_CDR)), n23 = length(intersect(GeneList_parietal_CDR, GeneList_parietal_braak)), n13
= length(intersect(GeneList_parietal_condition, GeneList_parietal_braak)),
   n123 = length(Reduce(intersect, list(GeneList_parietal_condition, GeneList_parietal_CDR,
GeneList_parietal_braak))), category = c("LOAD Case", "CDR", "Braak"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.8,
          cat.fontface = "bold",
          cat.default.pos = "outer",
```

```
  fill = c("skyblue", "pink1", "mediumorchid"))

intersect(GeneList_parietal_condition,GeneList_parietal_CDR,GeneList_parietal_braak)

```
```

DIAN

##DIAN ADAD Parietal Processing
```{r}


##Load in circRNA count data generated by DCC##

circCoordinates_parietalADAD <- read.table("/home/dubeu/Synapse/DCC/CircCoordinates", sep="\t", header=TRUE, stringsAsFactors = FALSE)
CircCount_parietalADAD <- read.table("/home/dubeu/Synapse/DCC/CircRNACount", sep="\t", header=TRUE, stringsAsFactors = FALSE)
LinearCount_parietalADAD <- read.table("/home/dubeu/Synapse/DCC/LinearCount", sep="\t", header=TRUE, stringsAsFactors = FALSE)


##Filter the circRNAs based on linear ratios

rowIndex <- 1:nrow(CircCount_parietalADAD)

circfiltered <- mclapply(rowIndex, circFILTER, CircCount_parietalADAD, LinearCount_parietalADAD, Nreplicates = 1, filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)

#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered_parietalADAD <- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

CircCount_parietalADAD <- CircCount_parietalADAD[-circfiltered_parietalADAD,]

circCoordinates_parietalADAD_filter <- circCoordinates_parietalADAD[-circfiltered_parietalADAD,]

#Remove extraneous columns

CircCount_parietalADAD <- CircCount_parietalADAD[,4:ncol(CircCount_parietalADAD)]

##Adjust names
names(CircCount_parietalADAD) <- gsub("_circRNA_Align1_Chimeric.out.junction", "", names(CircCount_parietalADAD))
```
```

```
##Collapse circRNA counts onto the gene level, removing those that are not_annotated
CircCount_parietalADAD$Gene <-
circCoordinates_parietalADAD[rownames(CircCount_parietalADAD),]$Gene
CircCount_parietalADAD <- subset(CircCount_parietalADAD, Gene != "not_annotated")
mydat.max_parietalADAD <- aggregate(. ~ Gene, data = CircCount_parietalADAD, sum)
rownames(mydat.max_parietalADAD) <- mydat.max_parietalADAD$Gene
mydat.max_parietalADAD$Gene <- NULL
cts_parietalADAD <- mydat.max_parietalADAD
rownames(cts_parietalADAD) <- paste0("circ", rownames(cts_parietalADAD))

##load parietalADAD phenotype / covariate information
coldata_parietalADAD <- read.csv(file="/40/AD/Expression/KnightADRCParietal/03.-phenotype/Pheno",
header=TRUE, na.strings = "NA")
#Restrict analyses to parietal tissues
coldata_parietalADAD <- subset(coldata_parietalADAD, brain_region == "parietal")

#Rename
coldata_parietalADAD$ID_RNAseq <- gsub("VY-", "VY.", coldata_parietalADAD$ID_RNAseq)

#Temporarily Assigning Ethnicity#
coldata_parietalADAD$Ethnicity <- coldata_parietalADAD$Ethnicity_temp

##Remove Consensus Outliers
coldata_parietalADAD <- subset(coldata_parietalADAD, ConsensusOutlier_RemoveBefore != 1)


#Set rownames
rownames(coldata_parietalADAD) <- coldata_parietalADAD$ID_RNAseq

#Restrict counts to match rownames
cts_parietalADAD <- cts_parietalADAD[, rownames(coldata_parietalADAD)]


#Set up factor levels
coldata_parietalADAD$pool <- factor(coldata_parietalADAD$pool, levels=c('1','2'))
coldata_parietalADAD$GENDER <- factor(coldata_parietalADAD$GENDER, levels=c('1','2'))
coldata_parietalADAD$Ethnicity <- factor(coldata_parietalADAD$Ethnicity, levels = c("EA", "AA"))
coldata_parietalADAD$condition <- droplevels(coldata_parietalADAD$condition)
```

```
####################
####################
## DE Analyses ##
####################
####################




##DE based on condition ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietalADAD_condition_filter <- DESeqDataSetFromMatrix(countData = cts_parietalADAD, colData
= coldata_parietalADAD, design = ~ 1)

dds_parietalADAD_condition_filter <- estimateSizeFactors(dds_parietalADAD_condition_filter)

#Remove dementedcontrols
dds_parietalADAD_condition_filter <- dds_parietalADAD_condition_filter[,
!(dds_parietalADAD_condition_filter$DementedControl == 1) ]


dds_parietalADAD_condition_filter <- dds_parietalADAD_condition_filter[
,!is.na(dds_parietalADAD_condition_filter$PMI)]

##Do not filter ADAD by PCs or AOD due to being missing from the dataset
#dds_parietalADAD_condition_filter <- dds_parietalADAD_condition_filter[
,!is.na(dds_parietalADAD_condition_filter$PC1)]
#dds_parietalADAD_condition_filter <- dds_parietalADAD_condition_filter[
,!is.na(dds_parietalADAD_condition_filter$PC2)]
#dds_parietalADAD_condition_filter <- dds_parietalADAD_condition_filter[
,!is.na(dds_parietalADAD_condition_filter$AGE_at_death)]

#Removed AGE_at_death since controls will be older than cases. In addition, replaced PC1 + PC2 with
Ethnicity due to above
design(dds_parietalADAD_condition_filter) = formula( ~ PMI + TIN.median. + pool + GENDER + Ethnicity
+ condition)
```

```
dds_parietalADAD_final_condition_filter <- DESeq(dds_parietalADAD_condition_filter, parallel = TRUE,
betaPrior = FALSE)

##Analyze results based on contrast

res_parietalADAD_condition_filter_ADADvsCO <- results(dds_parietalADAD_final_condition_filter,
contrast=c("condition", "ADAD", "CO"), alpha = 0.05)
summary(res_parietalADAD_condition_filter_ADADvsCO)

res_parietalADAD_condition_filter_ADADvsLOAD <- results(dds_parietalADAD_final_condition_filter,
contrast=c("condition", "ADAD", "LOAD"), alpha = 0.05)
summary(res_parietalADAD_condition_filter_ADADvsLOAD)


##DE based on condition, controlling for Braak Score ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietalADAD_conditionBraak_filter <- DESeqDataSetFromMatrix(countData = cts_parietalADAD,
colData = coldata_parietalADAD, design = ~ 1)

dds_parietalADAD_conditionBraak_filter <-
estimateSizeFactors(dds_parietalADAD_conditionBraak_filter)

#Remove dementedcontrols
dds_parietalADAD_conditionBraak_filter <- dds_parietalADAD_conditionBraak_filter[,
!(dds_parietalADAD_conditionBraak_filter$DementedControl == 1) ]


dds_parietalADAD_conditionBraak_filter <- dds_parietalADAD_conditionBraak_filter[
,!is.na(dds_parietalADAD_conditionBraak_filter$PMI)]

##Do not filter ADAD by PCs or AOD due to being missing from the dataset
#dds_parietalADAD_conditionBraak_filter <- dds_parietalADAD_conditionBraak_filter[
,!is.na(dds_parietalADAD_conditionBraak_filter$PC1)]
#dds_parietalADAD_conditionBraak_filter <- dds_parietalADAD_conditionBraak_filter[
,!is.na(dds_parietalADAD_conditionBraak_filter$PC2)]
#dds_parietalADAD_conditionBraak_filter <- dds_parietalADAD_conditionBraak_filter[
,!is.na(dds_parietalADAD_conditionBraak_filter$AGE_at_death)]
```

```
#Include Braak

dds_parietalADAD_conditionBraak_filter <- dds_parietalADAD_conditionBraak_filter[
,!is.na(dds_parietalADAD_conditionBraak_filter$braak_final)]

dds_parietalADAD_conditionBraak_filter$condition <-
droplevels(dds_parietalADAD_conditionBraak_filter$condition)


#Removed AGE_at_death since controls will be older than cases. In addition, replaced PC1 + PC2 with
Ethnicity due to above
design(dds_parietalADAD_conditionBraak_filter) = formula( ~ PMI + TIN.median. + pool + GENDER +
Ethnicity + braak_final + condition)


dds_parietalADAD_final_conditionBraak_filter <- DESeq(dds_parietalADAD_conditionBraak_filter,
parallel = TRUE, betaPrior = FALSE)

##Analyze results based on contrast

res_parietalADAD_conditionBraak_filter_ADADvsCO <-
results(dds_parietalADAD_final_conditionBraak_filter, contrast=c("condition", "ADAD", "CO"), alpha =
0.05)
summary(res_parietalADAD_conditionBraak_filter_ADADvsCO)

res_parietalADAD_conditionBraak_filter_ADADvsLOAD <-
results(dds_parietalADAD_final_conditionBraak_filter, contrast=c("condition", "ADAD", "LOAD"), alpha =
0.05)
summary(res_parietalADAD_conditionBraak_filter_ADADvsLOAD)


###Condition
##Aggregate / intersect list of genes
#GeneList_parietal_LOADvsCO <- unique(c(rownames(subset(res_parietal_condition_filter, padj <
0.05)), rownames(subset(res_parietal_braak_filter, padj < 0.05)),
rownames(subset(res_parietal_CDR_filter, padj < 0.05))))

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))
```

```
GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.triple.venn(area1 = length(GeneList_parietal_LOADvsCO), area2 =
length(GeneList_parietal_ADADvsLOAD), area3 = length(GeneList_parietal_ADADvsCO), n12 =
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD)), n23 =
length(intersect(GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)), n13 =
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsCO)),
   n123 = length(Reduce(intersect, list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO))), category = c("ADvsCO", "ADADvsAD*", "ADADvsCO"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.5,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))




##########################################
##########################################
## Check of consistency in direction of effect ADAD and LOAD ##
##########################################
##########################################

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


OverlapADAD_CO_ADAD_LOAD <-
cbind(res_parietalADAD_condition_filter_ADADvsCO[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange,
```

```
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange)
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, samesign=
as.data.frame(sign(OverlapADAD_CO_ADAD_LOAD[,1])==sign(OverlapADAD_CO_ADAD_LOAD[,2])))
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, increasingSeverity=
as.data.frame(abs(OverlapADAD_CO_ADAD_LOAD[,1])>abs(OverlapADAD_CO_ADAD_LOAD[,2])))



cbind(res_parietal_condition_filter[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_condition_filter_ADADvsCO[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[Reduce(intersect,
list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO)),]$log2FoldChange)



##########################################
##########################################
## Calculate overlap between ADAD and LOAD ##
##########################################
##########################################

###Condition
##Aggregate / intersect list of genes
#GeneList_parietal_LOADvsCO <- unique(c(rownames(subset(res_parietal_condition_filter, padj <
0.05)), rownames(subset(res_parietal_braak_filter, padj < 0.05)),
rownames(subset(res_parietal_CDR_filter, padj < 0.05))))

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.triple.venn(area1 = length(GeneList_parietal_LOADvsCO), area2 =
length(GeneList_parietal_ADADvsLOAD), area3 = length(GeneList_parietal_ADADvsCO), n12 =
```

```
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD)), n23 =
length(intersect(GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)), n13 =
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsCO)),
   n123 = length(Reduce(intersect, list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO))), category = c("LOADvsCO", "ADADvsLOAD", "ADADvsCO"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.5,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))




##########################################
##########################################
## Check of consistency in direction of effect ADAD and LOAD ##
##########################################
##########################################

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


OverlapADAD_CO_ADAD_LOAD <-
cbind(res_parietalADAD_condition_filter_ADADvsCO[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange,
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange)
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, samesign=
as.data.frame(sign(OverlapADAD_CO_ADAD_LOAD[,1])==sign(OverlapADAD_CO_ADAD_LOAD[,2])))
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, increasingSeverity=
as.data.frame(abs(OverlapADAD_CO_ADAD_LOAD[,1])>abs(OverlapADAD_CO_ADAD_LOAD[,2])))
```

```
cbind(res_parietal_condition_filter[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_condition_filter_ADADvsCO[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[Reduce(intersect,
list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO)),]$log2FoldChange)


```
```

MSBB

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Replication Differential Expression Analyses
```{r}

##########################################
##########################################
## Replication in AMP_AD MSBB ############
##########################################
##########################################

AMP_AD_circDE <- function() {

##Load in circRNA count data generated by DCC##

circCoordinates_BM <<- read.table("/home/dubeu/Synapse/AMP_AD/DCC/BM/CircCoordinates",
sep="\t", header=TRUE)
circCount_BM <<- read.table("/home/dubeu/Synapse/AMP_AD/DCC/BM/CircRNACount", sep="\t",
header=TRUE)
LinearCount_BM <<- read.table("/home/dubeu/Synapse/AMP_AD/DCC/BM/LinearCount", sep="\t",
header=TRUE)

##Filter the circRNAs based on linear ratios

rowIndex <<- 1:nrow(circCount_BM)

circfiltered <<- mclapply(rowIndex, circFILTER, circCount_BM, LinearCount_BM, Nreplicates = 1,
filter.sample = 3, filter.count = 3, percentage = 0.1, mc.cores = 50)
```

```
#remove Null via https://stackoverflow.com/questions/26539441/remove-null-elements-from-list-of-lists/26540063

circfiltered_BM <<- unlist(Filter(Negate(function(x) is.null(unlist(x))), circfiltered))

circCount_BM <<- circCount_BM[-circfiltered_BM,]

circCoordinates_BM_filter <<- circCoordinates_BM[-circfiltered_BM,]

circCoordinates_BM_filter_parietal <<- merge(circCoordinates_BM_filter,
circCoordinates_parietal_filter, by=c("Chr","Start","End"))

circCoordinates_BM_filter_parietal <<-
circCoordinates_BM_filter_parietal[circCoordinates_BM_filter_parietal$Gene.x !=
circCoordinates_BM_filter_parietal$Gene.y,]

circCoordinates_BM_filter_parietal <<-
circCoordinates_BM_filter_parietal[,c("Chr","Start","End","Gene.y")]

circCoordinates_BM_filter$rownum <<- row.names(circCoordinates_BM_filter)

circCoordinates_BM_filter <<- merge(circCoordinates_BM_filter,
circCoordinates_BM_filter_parietal,by=c("Chr","Start","End"), all.x=TRUE)
circCoordinates_BM_filter$Gene <<- as.character(circCoordinates_BM_filter$Gene)

circCoordinates_BM_filter[!is.na(circCoordinates_BM_filter$Gene.y),]$Gene <<-
circCoordinates_BM_filter[!is.na(circCoordinates_BM_filter$Gene.y),]$Gene.y

circCoordinates_BM_filter$Gene <<- as.factor(circCoordinates_BM_filter$Gene)
row.names(circCoordinates_BM_filter) <<- circCoordinates_BM_filter$rownum


#Fix non-filtered

circCoordinates_BM$rownum <<- row.names(circCoordinates_BM)

circCoordinates_BM <<- merge(circCoordinates_BM,
circCoordinates_BM_filter_parietal,by=c("Chr","Start","End"), all.x=TRUE)
circCoordinates_BM$Gene <<- as.character(circCoordinates_BM$Gene)

circCoordinates_BM[!is.na(circCoordinates_BM$Gene.y),]$Gene <<-
circCoordinates_BM[!is.na(circCoordinates_BM$Gene.y),]$Gene.y
```

```
circCoordinates_BM$Gene <<- as.factor(circCoordinates_BM$Gene)
row.names(circCoordinates_BM) <<- circCoordinates_BM$rownum

##Remove extraneous columns
circCount_BM <<- circCount_BM[,4:ncol(circCount_BM)]

#Process names
names(circCount_BM) <<- gsub("_Chimeric.out.junction", "", names(circCount_BM))

#Aggregate on genes names removing the not_annotated
circCount_BM$Gene <<- circCoordinates_BM[rownames(circCount_BM),]$Gene
circCount_BM <<- subset(circCount_BM, Gene != "not_annotated")
mydat.max_BM <<- aggregate(. ~ Gene, data = circCount_BM, sum)
rownames(mydat.max_BM) <<- mydat.max_BM$Gene
mydat.max_BM$Gene <<- NULL
cts_BM <<- mydat.max_BM
rownames(cts_BM) <<- paste0("circ", rownames(cts_BM))

##Load in phenotype data ##
coldata_BM <<- read.csv(file="/home/dubeu/Synapse/AMP_AD/MSBB_Pheno_05_15_2018.csv",
header=TRUE, na.strings = "NA")
coldata_BM <<- subset(coldata_BM, BrodmannArea=="BM")

##Set factor levels

coldata_BM$NP.1 <<- factor(coldata_BM$NP.1, levels=c('1','2','3','4'))
coldata_BM$Apo1 <<- factor(coldata_BM$Apo1, levels=c('0','2','3','4'))
coldata_BM$Apo2 <<- factor(coldata_BM$Apo2, levels=c('0','2','3','4'))
coldata_BM$RACE <<- factor(coldata_BM$RACE, levels=c('A','B','H','W','U'))
coldata_BM$RACE <<- relevel(coldata_BM$RACE, "W")
coldata_BM$SEX <<- factor(coldata_BM$SEX, levels = c("M","F"))
coldata_BM$batch <<- factor(coldata_BM$batch, levels = levels(coldata_BM$batch))

rownames(coldata_BM) <<- coldata_BM$sampleIdentifier

##Relevel NP.1
#levels(coldata_BM$NP.1) <- list("1"=c("1"), "2"=c("2","3","4"))

coldata_BM[(as.character(coldata_BM$NP.1)=="1" & coldata_BM$CDR>=1),]$DementedControl <<- 1

cts_BM <<- cts_BM[, rownames(coldata_BM)]


##DE based on condition ##
```

```
############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
############

dds_BM_condition_filter <<- DESeqDataSetFromMatrix(countData = cts_BM, colData = coldata_BM,
design = ~ 1)

dds_BM_condition_filter <<- estimateSizeFactors(dds_BM_condition_filter)

#Remove individuals with missing data
dds_BM_condition_filter <<- dds_BM_condition_filter[ ,!is.na(dds_BM_condition_filter$batch)]
dds_BM_condition_filter <<- dds_BM_condition_filter[ ,!is.na(dds_BM_condition_filter$PC1)]
dds_BM_condition_filter <<- dds_BM_condition_filter[ ,!is.na(dds_BM_condition_filter$PC2)]

#Remove dementedcontrols
dds_BM_condition_filter <<- dds_BM_condition_filter[, !(dds_BM_condition_filter$DementedControl
== 1)]

dds_BM_condition_filter$batch <<- droplevels(dds_BM_condition_filter$batch)


design(dds_BM_condition_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 +
NP.1)


dds_BM_final_condition_filter <<- DESeq(dds_BM_condition_filter, parallel = TRUE, betaPrior = FALSE)

res_BM_condition_filter <<- results(dds_BM_final_condition_filter, contrast=c("NP.1", "2", "1"), alpha =
0.05)
summary(res_BM_condition_filter)


##DE based on CDR ##

############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
############
```

```
dds_BM_CDR_filter <<- DESeqDataSetFromMatrix(countData = cts_BM, colData = coldata_BM, design =
~ 1)

dds_BM_CDR_filter <<- estimateSizeFactors(dds_BM_CDR_filter)

#Remove individuals with missing data
dds_BM_CDR_filter <<- dds_BM_CDR_filter[ ,!is.na(dds_BM_CDR_filter$batch)]
dds_BM_CDR_filter <<- dds_BM_CDR_filter[ ,!is.na(dds_BM_CDR_filter$PC1)]
dds_BM_CDR_filter <<- dds_BM_CDR_filter[ ,!is.na(dds_BM_CDR_filter$PC2)]

dds_BM_CDR_filter <<- dds_BM_CDR_filter[ ,!is.na(dds_BM_CDR_filter$CDR)]

#Remove dementedcontrols
dds_BM_CDR_filter <<- dds_BM_CDR_filter[, !(dds_BM_CDR_filter$DementedControl == 1)]


dds_BM_CDR_filter$batch <<- droplevels(dds_BM_CDR_filter$batch)


design(dds_BM_CDR_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 + CDR)


dds_BM_final_CDR_filter <<- DESeq(dds_BM_CDR_filter, parallel = TRUE, betaPrior = FALSE)

res_BM_CDR_filter <<- results(dds_BM_final_CDR_filter, alpha = 0.05)
summary(res_BM_CDR_filter)



##DE based on braak ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_BM_braak_filter <<- DESeqDataSetFromMatrix(countData = cts_BM, colData = coldata_BM, design
= ~ 1)

dds_BM_braak_filter <<- estimateSizeFactors(dds_BM_braak_filter)

#Remove individuals with missing data
dds_BM_braak_filter <<- dds_BM_braak_filter[ ,!is.na(dds_BM_braak_filter$batch)]
```

```
dds_BM_braak_filter <<- dds_BM_braak_filter[ ,!is.na(dds_BM_braak_filter$PC1)]
dds_BM_braak_filter <<- dds_BM_braak_filter[ ,!is.na(dds_BM_braak_filter$PC2)]

dds_BM_braak_filter <<- dds_BM_braak_filter[ ,!is.na(dds_BM_braak_filter$bbscore)]

#Remove dementedcontrols
dds_BM_braak_filter <<- dds_BM_braak_filter[, !(dds_BM_braak_filter$DementedControl == 1)]


dds_BM_braak_filter$batch <<- droplevels(dds_BM_braak_filter$batch)


design(dds_BM_braak_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 +
bbscore)


dds_BM_final_braak_filter <<- DESeq(dds_BM_braak_filter, parallel = TRUE, betaPrior = FALSE)

res_BM_braak_filter <<- results(dds_BM_final_braak_filter, alpha = 0.05)
summary(res_BM_braak_filter)



##DE based on PlaqueMean ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_BM_PlaqueMean_filter <<- DESeqDataSetFromMatrix(countData = cts_BM, colData = coldata_BM,
design = ~ 1)

dds_BM_PlaqueMean_filter <<- estimateSizeFactors(dds_BM_PlaqueMean_filter)

#Remove individuals with missing data
dds_BM_PlaqueMean_filter <<- dds_BM_PlaqueMean_filter[
,!is.na(dds_BM_PlaqueMean_filter$batch)]
dds_BM_PlaqueMean_filter <<- dds_BM_PlaqueMean_filter[ ,!is.na(dds_BM_PlaqueMean_filter$PC1)]
dds_BM_PlaqueMean_filter <<- dds_BM_PlaqueMean_filter[ ,!is.na(dds_BM_PlaqueMean_filter$PC2)]

dds_BM_PlaqueMean_filter <<- dds_BM_PlaqueMean_filter[
,!is.na(dds_BM_PlaqueMean_filter$PlaqueMean)]
```

```
#Remove dementedcontrols
dds_BM_PlaqueMean_filter <<- dds_BM_PlaqueMean_filter[,
!(dds_BM_PlaqueMean_filter$DementedControl == 1)]

dds_BM_PlaqueMean_filter$batch <<- droplevels(dds_BM_PlaqueMean_filter$batch)


design(dds_BM_PlaqueMean_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 +
PlaqueMean)


dds_BM_final_PlaqueMean_filter <<- DESeq(dds_BM_PlaqueMean_filter, parallel = TRUE, betaPrior =
FALSE)

res_BM_PlaqueMean_filter <<- results(dds_BM_final_PlaqueMean_filter, alpha = 0.05)
summary(res_BM_PlaqueMean_filter)
}


##BM10
AMP_AD_circDE_BM10 <- gsub("BM","BM10",as.list(body(AMP_AD_circDE)))
#Replace first line to create function
AMP_AD_circDE_BM10[1] <- "AMP_AD_circDE_BM10 <- function() {"
#Add last line to end function
AMP_AD_circDE_BM10[length(AMP_AD_circDE_BM10)+1] <- "}"
write(AMP_AD_circDE_BM10, "/home/dubeu/Synapse/AMP_AD_circDE_BM10.R")


##BM22
AMP_AD_circDE_BM22 <- gsub("BM","BM22",as.list(body(AMP_AD_circDE)))
#Replace first line to create function
AMP_AD_circDE_BM22[1] <- "AMP_AD_circDE_BM22 <- function() {"
#Add last line to end function
AMP_AD_circDE_BM22[length(AMP_AD_circDE_BM22)+1] <- "}"
write(AMP_AD_circDE_BM22, "/home/dubeu/Synapse/AMP_AD_circDE_BM22.R")


##BM36
AMP_AD_circDE_BM36 <- gsub("BM","BM36",as.list(body(AMP_AD_circDE)))
#Replace first line to create function
AMP_AD_circDE_BM36[1] <- "AMP_AD_circDE_BM36 <- function() {"
#Add last line to end function
AMP_AD_circDE_BM36[length(AMP_AD_circDE_BM36)+1] <- "}"
```

```
write(AMP_AD_circDE_BM36, "/home/dubeu/Synapse/AMP_AD_circDE_BM36.R")

##BM44
AMP_AD_circDE_BM44 <- gsub("BM","BM44",as.list(body(AMP_AD_circDE)))
#Replace first line to create function
AMP_AD_circDE_BM44[1] <- "AMP_AD_circDE_BM44 <- function() {"
#Add last line to end function
AMP_AD_circDE_BM44[length(AMP_AD_circDE_BM44)+1] <- "}"
write(AMP_AD_circDE_BM44, "/home/dubeu/Synapse/AMP_AD_circDE_BM44.R")


source('/home/dubeu/Synapse/AMP_AD_circDE_BM10.R')
source('/home/dubeu/Synapse/AMP_AD_circDE_BM22.R')
source('/home/dubeu/Synapse/AMP_AD_circDE_BM36.R')
source('/home/dubeu/Synapse/AMP_AD_circDE_BM44.R')

#Run the actual code
AMP_AD_circDE_BM10()
AMP_AD_circDE_BM22()
AMP_AD_circDE_BM36()
AMP_AD_circDE_BM44()

```
```

Regression-Based circRNA Association Independence Analyses

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Regression Based Independence Analyses
```

Independence from AD-associated cognate linear mRNA changes

```
#####Linear Independence
```
```{r}
sigList_BM44_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_BM44_CDR = sigList_BM44_CDR[order(sigList_BM44_CDR$i_adjpval),]
sigList_BM44_CDR = rownames(sigList_BM44_CDR)
#sigList_BM44_CDR_top <- head(sigList_BM44_CDR, n = 11)
sigList_BM44_CDR_top <- sigList_BM44_CDR

##Have to remove circRNAs for which there are no linear
```
```

```
sigList_BM44_CDR_top <-
paste0("circ",colnames(t(na.omit(norm_BM44_salmon_CDR_counts_regress[gsub("circ","",sigList_BM4
4_CDR_top),]))))

LinearCircTable_BM44 <- data.frame()

for (i in sigList_BM44_CDR_top) {

        print(i)

        print(match(i,sigList_BM44_CDR_top)/length(sigList_BM44_CDR_top))

        Gene_BM44_CDR <- t(norm_BM44_CDR_counts[i,])
        Gene_BM44_CDR_linear <- t(norm_BM44_salmon_CDR_counts[i,])

        regress_BM44_CDR <- cbind(coldata_BM44[rownames(Gene_BM44_CDR),], Gene =
t(norm_BM44_CDR_counts[i,]), Linear = t(norm_BM44_salmon_CDR_counts[gsub("circ","",i),]))

        model_regress_BM44_CDR_circ = lm(CDR ~ regress_BM44_CDR[,i] + PMI + TIN.median. + AOD +
batch + SEX + PC1+PC2, data=regress_BM44_CDR )

        print(summary(model_regress_BM44_CDR_circ))

        model_regress_BM44_CDR_linear = lm(CDR ~ regress_BM44_CDR[,gsub("circ","",i)] + PMI +
TIN.median. + AOD + batch + SEX + PC1+PC2, data=regress_BM44_CDR )

        print(summary(model_regress_BM44_CDR_linear))

        model_regress_BM44_CDR_both = lm(CDR ~ regress_BM44_CDR[,i] +
regress_BM44_CDR[,gsub("circ","",i)] + PMI + TIN.median. + AOD + batch + SEX + PC1+PC2,
data=regress_BM44_CDR )
        model_regress_BM44_CDR_both_correlation <-
cor.test(regress_BM44_CDR[,i],regress_BM44_CDR[,gsub("circ","",i)], method="spearman")
        print(summary(model_regress_BM44_CDR_both))

        library(relaimpo)

        LinearCircTable_BM44 <-
rbind(LinearCircTable_BM44,cbind(as.character(i),summary(model_regress_BM44_CDR_linear)$coeffici
ents[2,1],summary(model_regress_BM44_CDR_linear)$coefficients[2,2],summary(model_regress_BM4
4_CDR_linear)$coefficients[2,4],"",summary(model_regress_BM44_CDR_circ)$coefficients[2,1],summar
y(model_regress_BM44_CDR_circ)$coefficients[2,2],summary(model_regress_BM44_CDR_circ)$coeffici
ents[2,4],"",model_regress_BM44_CDR_both_correlation$estimate,"",summary(model_regress_BM44_
CDR_both)$coefficients[3,1],summary(model_regress_BM44_CDR_both)$coefficients[3,2],summary(mo
```

```
del_regress_BM44_CDR_both)$coefficients[3,4],"",summary(model_regress_BM44_CDR_both)$coefficients[2,1],summary(model_regress_BM44_CDR_both)$coefficients[2,2],summary(model_regress_BM44_CDR_both)$coefficients[2,4],"",(calc.relimp(model_regress_BM44_CDR_both, type = c("lmg"), rela = FALSE))$lmg[2],(calc.relimp(model_regress_BM44_CDR_both, type = c("lmg"), rela = FALSE))$lmg[3]))

}

colnames(LinearCircTable_BM44) <-
c("RNA","LinearEstimate","LinearSE","LinearPvalue","","CircEstimate","CircSE","CircPvalue","","Corr","","BothLinearEstimate","BothLinearSE","BothLinearPvalue","","BothCircEstimate","BothCircSE","BothCircPvalue","","RelaimpoCirc","RelaimpoLinear")

##Change all factors to numeric
LinearCircTable_BM44[] <- lapply(LinearCircTable_BM44, function(x) as.numeric(as.character(x)))
##Recover RNA
LinearCircTable_BM44$RNA <- gsub("circ","",as.character(sigList_BM44_CDR_top))


sigList_parietal_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_parietal_CDR = sigList_parietal_CDR[order(sigList_parietal_CDR$i_adjpval),]
sigList_parietal_CDR = rownames(sigList_parietal_CDR)
sigList_parietal_CDR_top <- sigList_parietal_CDR
sigList_parietal_CDR_top <-
paste0("circ",colnames(t(na.omit(norm_BM44_salmon_CDR_counts_regress[gsub("circ","",sigList_BM44_CDR_top),]))))

LinearCircTable_parietal <- data.frame()
for (i in sigList_parietal_CDR_top) {
        print(i)
        print(match(i,sigList_parietal_CDR_top)/length(sigList_parietal_CDR_top))
        Gene_parietal_CDR <- t(norm_parietal_CDR_counts[i,])
        Gene_parietal_CDR_linear <- t(norm_parietal_salmon_CDR_counts[i,])
        regress_parietal_CDR <- cbind(coldata_parietal[rownames(Gene_parietal_CDR),], Gene =
t(norm_parietal_CDR_counts[i,]), Linear = t(norm_parietal_salmon_CDR_counts[gsub("circ","",i),]))
        Gene_archive_parietal_CDR <- cbind(Gene_archive_parietal_CDR,
as.data.frame(t(norm_parietal_CDR_counts_regress[i,])))
        model_regress_parietal_CDR_circ = lm(CDR_e ~ regress_parietal_CDR[,i] + PMI + TIN.median. +
AGE_at_death + pool + GENDER + PC1+PC2, data=regress_parietal_CDR )
        print(summary(model_regress_parietal_CDR_circ))
        model_regress_parietal_CDR_linear = lm(CDR_e ~ regress_parietal_CDR[,gsub("circ","",i)] + PMI
+ TIN.median. + AGE_at_death + pool + GENDER + PC1+PC2, data=regress_parietal_CDR )
        print(summary(model_regress_parietal_CDR_linear))
```

```
        model_regress_parietal_CDR_both = lm(CDR_e ~ regress_parietal_CDR[,i] +
regress_parietal_CDR[,gsub("circ","",i)] + PMI + TIN.median. + AGE_at_death + pool + GENDER +
PC1+PC2, data=regress_parietal_CDR )
        model_regress_parietal_CDR_both_correlation <-
cor.test(regress_parietal_CDR[,i],regress_parietal_CDR[,gsub("circ","",i)], method="spearman")
        print(summary(model_regress_parietal_CDR_both))
        library(relaimpo)
        LinearCircTable_parietal <-
rbind(LinearCircTable_parietal,cbind(as.character(i),summary(model_regress_parietal_CDR_linear)$coef
ficients[2,1],summary(model_regress_parietal_CDR_linear)$coefficients[2,2],summary(model_regress_
parietal_CDR_linear)$coefficients[2,4],"",summary(model_regress_parietal_CDR_circ)$coefficients[2,1],
summary(model_regress_parietal_CDR_circ)$coefficients[2,2],summary(model_regress_parietal_CDR_c
irc)$coefficients[2,4],"",model_regress_parietal_CDR_both_correlation$estimate,"",summary(model_re
gress_parietal_CDR_both)$coefficients[3,1],summary(model_regress_parietal_CDR_both)$coefficients[
3,2],summary(model_regress_parietal_CDR_both)$coefficients[3,4],"",summary(model_regress_parieta
l_CDR_both)$coefficients[2,1],summary(model_regress_parietal_CDR_both)$coefficients[2,2],summary(
model_regress_parietal_CDR_both)$coefficients[2,4],"",(calc.relimp(model_regress_parietal_CDR_both,
type = c("lmg"), rela = FALSE))$lmg[1],(calc.relimp(model_regress_parietal_CDR_both, type = c("lmg"),
rela = FALSE))$lmg[2]))
}
colnames(LinearCircTable_parietal) <-
c("RNA","LinearEstimate","LinearSE","LinearPvalue","","CircEstimate","CircSE","CircPvalue","","Corr","",
"BothLinearEstimate","BothLinearSE","BothLinearPvalue","","BothCircEstimate","BothCircSE","BothCirc
Pvalue","","RelaimpoCirc","RelaimpoLinear")
##Change all factors to numeric
LinearCircTable_parietal[] <- lapply(LinearCircTable_parietal, function(x) as.numeric(as.character(x)))
##Recover RNA
LinearCircTable_parietal$RNA <- gsub("circ","",as.character(sigList_parietal_CDR_top))
##Generate a table
LinearCircTable_parietal_table <- LinearCircTable_parietal
LinearCircTable_parietal_table$LinearEstimate <-
formatC(LinearCircTable_parietal_table$LinearEstimate, format = "f", digits = 2)
LinearCircTable_parietal_table$LinearPvalue <- formatC(LinearCircTable_parietal_table$LinearPvalue,
format = "e", digits = 2)
LinearCircTable_parietal_table$CircEstimate <- formatC(LinearCircTable_parietal_table$CircEstimate,
format = "f", digits = 2)
LinearCircTable_parietal_table$CircPvalue <- formatC(LinearCircTable_parietal_table$CircPvalue, format
= "e", digits = 2)
LinearCircTable_parietal_table$RelaimpoCirc <- formatC(LinearCircTable_parietal_table$RelaimpoCirc,
format = "e", digits = 2)
LinearCircTable_parietal_table$RelaimpoLinear <-
formatC(LinearCircTable_parietal_table$RelaimpoLinear, format = "e", digits = 2)
LinearCircTable_parietal_table$BothLinearEstimate <-
formatC(LinearCircTable_parietal_table$BothLinearEstimate, format = "f", digits = 2)
```

```
LinearCircTable_parietal_table$BothLinearPvalue <-
formatC(LinearCircTable_parietal_table$BothLinearPvalue, format = "e", digits = 2)
LinearCircTable_parietal_table$BothCircEstimate <-
formatC(LinearCircTable_parietal_table$BothCircEstimate, format = "f", digits = 2)
LinearCircTable_parietal_table$BothCircPvalue <-
formatC(LinearCircTable_parietal_table$BothCircPvalue, format = "e", digits = 2)
LinearCircTable_parietal_table$Corr <- formatC(LinearCircTable_parietal_table$Corr, format = "f", digits
= 2)
LinearCircTable_parietal_table <- as.data.frame(lapply(LinearCircTable_parietal_table, function(x)
gsub("e", " × 10",x) ))
colnames(LinearCircTable_parietal_table) <-
c("RNA","LinearEstimate","LinearPvalue","","CircEstimate","CircPvalue","","Corr","","BothLinearEstimat
e","BothLinearPvalue","","BothCircEstimate","BothCircPvalue","","RelaimpoCirc","RelaimpoLinear")


#FE metafor
library(metafor)
meta_results <- function(x,parietalEst,BM44Est,parietalSE,BM44SE) {
Estimate <- c(parietalEst[x],BM44Est[x])
SE <- c(parietalSE[x],BM44SE[x])
rma(yi=Estimate,sei=SE, method="FE")$pval
}
LinearCircTable_meta <-
LinearCircTable_parietal[,c("RNA","LinearPvalue","CircPvalue","BothLinearPvalue","BothCircPvalue")]
LinearCircTable_meta$LinearPvalue <- as.numeric(unlist(mclapply(1:nrow(LinearCircTable_parietal),
meta_results, mc.cores=5, parietalEst=LinearCircTable_parietal$LinearEstimate,
BM44Est=LinearCircTable_BM44$LinearEstimate,parietalSE=LinearCircTable_parietal$LinearSE,
BM44SE=LinearCircTable_BM44$LinearSE)))
LinearCircTable_meta$CircPvalue <- as.numeric(unlist(mclapply(1:nrow(LinearCircTable_parietal),
meta_results, mc.cores=5, parietalEst=LinearCircTable_parietal$CircEstimate,
BM44Est=LinearCircTable_BM44$CircEstimate,parietalSE=LinearCircTable_parietal$CircSE,
BM44SE=LinearCircTable_BM44$CircSE)))
LinearCircTable_meta$BothLinearPvalue <-
as.numeric(unlist(mclapply(1:nrow(LinearCircTable_parietal), meta_results, mc.cores=5,
parietalEst=LinearCircTable_parietal$LinearEstimate,
BM44Est=LinearCircTable_BM44$LinearEstimate,parietalSE=LinearCircTable_parietal$BothLinearSE,
BM44SE=LinearCircTable_BM44$BothLinearSE)))
LinearCircTable_meta$BothCircPvalue <- as.numeric(unlist(mclapply(1:nrow(LinearCircTable_parietal),
meta_results, mc.cores=5, parietalEst=LinearCircTable_parietal$BothCircEstimate,
BM44Est=LinearCircTable_BM44$BothCircEstimate,parietalSE=LinearCircTable_parietal$BothCircSE,
BM44SE=LinearCircTable_BM44$BothCircSE)))
##Add the average correlation between discovery and replication
LinearCircTable_meta$AvgCorr <-
rowMeans(cbind(LinearCircTable_parietal$Corr,LinearCircTable_BM44$Corr))
```

```
LinearCircTable_meta$AvgRelaimpoCirc <-
rowMeans(cbind(LinearCircTable_parietal$RelaimpoCirc,LinearCircTable_BM44$RelaimpoCirc))
LinearCircTable_meta$AvgRelaimpoLinear <-
rowMeans(cbind(LinearCircTable_parietal$RelaimpoLinear,LinearCircTable_BM44$RelaimpoLinear))

```

Independence from AD-associated cell-type proportion changes

```
#####Cell type proportion change independence
```{r}

sigList_BM44_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_BM44_CDR = sigList_BM44_CDR[order(sigList_BM44_CDR$i_adjpval),]
sigList_BM44_CDR = rownames(sigList_BM44_CDR)
#sigList_BM44_CDR_top <- head(sigList_BM44_CDR, n = 11)
sigList_BM44_CDR_top <- sigList_BM44_CDR

NeuronCircTable_BM44 <- data.frame()

for (i in sigList_BM44_CDR_top) {

	print(i)

	print(match(i,sigList_BM44_CDR_top)/length(sigList_BM44_CDR_top))

	Gene_BM44_CDR <- t(norm_BM44_CDR_counts[i,])
	Gene_BM44_CDR_neuron <- t(norm_BM44_salmon_CDR_counts[i,])

	regress_BM44_CDR <- cbind(coldata_BM44[rownames(Gene_BM44_CDR),], Gene =
t(norm_BM44_CDR_counts[i,]), Linear = t(norm_BM44_salmon_CDR_counts[gsub("circ","",i),]))

	model_regress_BM44_CDR_circ = lm(CDR ~ regress_BM44_CDR[,i] + PMI + TIN.median. + AOD +
batch + SEX + PC1+PC2, data=regress_BM44_CDR )

	print(summary(model_regress_BM44_CDR_circ))

	model_regress_BM44_CDR_neuron = lm(CDR ~ Neuron + Oligodendrocyte + Microglia + PMI +
TIN.median. + AOD + batch + SEX + PC1+PC2, data=regress_BM44_CDR )

	print(summary(model_regress_BM44_CDR_neuron))

	model_regress_BM44_CDR_both = lm(CDR ~ regress_BM44_CDR[,i] + Neuron +
Oligodendrocyte + Microglia + PMI + TIN.median. + AOD + batch + SEX + PC1+PC2,
data=regress_BM44_CDR )
```
```

```
print(summary(model_regress_BM44_CDR_both))

neuron_effect <- summary(model_regress_BM44_CDR_neuron)$coefficients[2,1]
neuron_SE <- summary(model_regress_BM44_CDR_neuron)$coefficients[2,2]
neuron_pval <- summary(model_regress_BM44_CDR_neuron)$coefficients[2,4]
oligo_effect <- summary(model_regress_BM44_CDR_neuron)$coefficients[3,1]
oligo_SE <- summary(model_regress_BM44_CDR_neuron)$coefficients[3,2]
oligo_pval <- summary(model_regress_BM44_CDR_neuron)$coefficients[3,4]
micro_effect <- summary(model_regress_BM44_CDR_neuron)$coefficients[4,1]
micro_SE <- summary(model_regress_BM44_CDR_neuron)$coefficients[4,2]
micro_pval <- summary(model_regress_BM44_CDR_neuron)$coefficients[4,4]

circ_effect <- summary(model_regress_BM44_CDR_circ)$coefficients[2,1]
circ_SE <- summary(model_regress_BM44_CDR_circ)$coefficients[2,2]
circ_pval <- summary(model_regress_BM44_CDR_circ)$coefficients[2,4]

bothneuron_effect <- summary(model_regress_BM44_CDR_both)$coefficients[3,1]
bothneuron_SE <- summary(model_regress_BM44_CDR_both)$coefficients[3,2]
bothneuron_pval <- summary(model_regress_BM44_CDR_both)$coefficients[3,4]
botholigo_effect <- summary(model_regress_BM44_CDR_both)$coefficients[4,1]
botholigo_SE <- summary(model_regress_BM44_CDR_both)$coefficients[4,2]
botholigo_pval <- summary(model_regress_BM44_CDR_both)$coefficients[4,4]
bothmicro_effect <- summary(model_regress_BM44_CDR_both)$coefficients[5,1]
bothmicro_SE <- summary(model_regress_BM44_CDR_both)$coefficients[5,2]
bothmicro_pval <- summary(model_regress_BM44_CDR_both)$coefficients[5,4]

bothcirc_effect <- summary(model_regress_BM44_CDR_both)$coefficients[2,1]
bothcirc_SE <- summary(model_regress_BM44_CDR_both)$coefficients[2,2]
bothcirc_pval <- summary(model_regress_BM44_CDR_both)$coefficients[2,4]


NeuronCircTable_BM44 <-
rbind(NeuronCircTable_BM44,cbind(as.character(i),neuron_effect,neuron_SE,neuron_pval,"",oligo_effe
ct,oligo_SE,oligo_pval,"",micro_effect,micro_SE,micro_pval,"",circ_effect,circ_SE,circ_pval,"",bothneuro
n_effect,bothneuron_SE,bothneuron_pval,"",botholigo_effect,botholigo_SE,botholigo_pval,"",bothmicr
o_effect,bothmicro_SE,bothmicro_pval,"",bothcirc_effect,bothcirc_SE,bothcirc_pval))

}

colnames(NeuronCircTable_BM44) <-
c("RNA","NeuronEstimate","NeuronSE","NeuronPvalue","","OligoEstimate","OligoSE","OligoPvalue","","
MicroEstimate","MicroSE","MicroPvalue","","CircEstimate","CircSE","CircPvalue","","BothNeuronEstima
te","BothNeuronSE","BothNeuronPvalue","","BothOligoEstimate","BothOligoSE","BothOligoPvalue","","
```

```
BothMicroEstimate","BothMicroSE","BothMicroPvalue","","BothCircEstimate","BothCircSE","BothCircPv
alue")

##Change all factors to numeric
NeuronCircTable_BM44[] <- lapply(NeuronCircTable_BM44, function(x) as.numeric(as.character(x)))

##Recover RNA
NeuronCircTable_BM44$RNA <- gsub("circ","",as.character(sigList_BM44_CDR_top))

#####################################################################
#####################################################################
#####################################################################
#####################################################################

sigList_parietal_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_parietal_CDR = sigList_parietal_CDR[order(sigList_parietal_CDR$i_adjpval),]
sigList_parietal_CDR = rownames(sigList_parietal_CDR)
#sigList_parietal_CDR_top <- head(sigList_parietal_CDR, n = 11)
sigList_parietal_CDR_top <- sigList_parietal_CDR

NeuronCircTable_parietal <- data.frame()

for (i in sigList_BM44_CDR_top) {

        print(i)

        print(match(i,sigList_parietal_CDR_top)/length(sigList_parietal_CDR_top))

        Gene_parietal_CDR <- t(norm_parietal_CDR_counts[i,])
        Gene_parietal_CDR_neuron <- t(norm_parietal_salmon_CDR_counts[i,])

        regress_parietal_CDR <- cbind(coldata_parietal[rownames(Gene_parietal_CDR),], Gene =
t(norm_parietal_CDR_counts[i,]), Linear = t(norm_parietal_salmon_CDR_counts[gsub("circ","",i),]))

        model_regress_parietal_CDR_circ = lm(CDR_e ~ regress_parietal_CDR[,i] + PMI + TIN.median. +
AGE_at_death + pool + GENDER + PC1+PC2, data=regress_parietal_CDR )

        print(summary(model_regress_parietal_CDR_circ))

        model_regress_parietal_CDR_neuron = lm(CDR_e ~ Neuron + Oligodendrocyte + Microglia +
PMI + TIN.median. + AGE_at_death + pool + GENDER + PC1+PC2, data=regress_parietal_CDR )

        print(summary(model_regress_parietal_CDR_neuron))
```

```
        model_regress_parietal_CDR_both = lm(CDR_e ~ regress_parietal_CDR[,i] + Neuron +
Oligodendrocyte + Microglia + PMI + TIN.median. + AGE_at_death + pool + GENDER + PC1+PC2,
data=regress_parietal_CDR )

        print(summary(model_regress_parietal_CDR_both))

        neuron_effect <- summary(model_regress_parietal_CDR_neuron)$coefficients[2,1]
        neuron_SE <- summary(model_regress_parietal_CDR_neuron)$coefficients[2,2]
        neuron_pval <- summary(model_regress_parietal_CDR_neuron)$coefficients[2,4]
        oligo_effect <- summary(model_regress_parietal_CDR_neuron)$coefficients[3,1]
        oligo_SE <- summary(model_regress_parietal_CDR_neuron)$coefficients[3,2]
        oligo_pval <- summary(model_regress_parietal_CDR_neuron)$coefficients[3,4]
        micro_effect <- summary(model_regress_parietal_CDR_neuron)$coefficients[4,1]
        micro_SE <- summary(model_regress_parietal_CDR_neuron)$coefficients[4,2]
        micro_pval <- summary(model_regress_parietal_CDR_neuron)$coefficients[4,4]

        circ_effect <- summary(model_regress_parietal_CDR_circ)$coefficients[2,1]
        circ_SE <- summary(model_regress_parietal_CDR_circ)$coefficients[2,2]
        circ_pval <- summary(model_regress_parietal_CDR_circ)$coefficients[2,4]

        bothneuron_effect <- summary(model_regress_parietal_CDR_both)$coefficients[3,1]
        bothneuron_SE <- summary(model_regress_parietal_CDR_both)$coefficients[3,2]
        bothneuron_pval <- summary(model_regress_parietal_CDR_both)$coefficients[3,4]
        botholigo_effect <- summary(model_regress_parietal_CDR_both)$coefficients[4,1]
        botholigo_SE <- summary(model_regress_parietal_CDR_both)$coefficients[4,2]
        botholigo_pval <- summary(model_regress_parietal_CDR_both)$coefficients[4,4]
        bothmicro_effect <- summary(model_regress_parietal_CDR_both)$coefficients[5,1]
        bothmicro_SE <- summary(model_regress_parietal_CDR_both)$coefficients[5,2]
        bothmicro_pval <- summary(model_regress_parietal_CDR_both)$coefficients[5,4]

        bothcirc_effect <- summary(model_regress_parietal_CDR_both)$coefficients[2,1]
        bothcirc_SE <- summary(model_regress_parietal_CDR_both)$coefficients[2,2]
        bothcirc_pval <- summary(model_regress_parietal_CDR_both)$coefficients[2,4]


        NeuronCircTable_parietal <-
rbind(NeuronCircTable_parietal,cbind(as.character(i),neuron_effect,neuron_SE,neuron_pval,"",oligo_eff
ect,oligo_SE,oligo_pval,"",micro_effect,micro_SE,micro_pval,"",circ_effect,circ_SE,circ_pval,"",bothneur
on_effect,bothneuron_SE,bothneuron_pval,"",botholigo_effect,botholigo_SE,botholigo_pval,"",bothmic
ro_effect,bothmicro_SE,bothmicro_pval,"",bothcirc_effect,bothcirc_SE,bothcirc_pval))

}
```

```
colnames(NeuronCircTable_parietal) <-
c("RNA","NeuronEstimate","NeuronSE","NeuronPvalue","","OligoEstimate","OligoSE","OligoPvalue","","
MicroEstimate","MicroSE","MicroPvalue","","CircEstimate","CircSE","CircPvalue","","BothNeuronEstima
te","BothNeuronSE","BothNeuronPvalue","","BothOligoEstimate","BothOligoSE","BothOligoPvalue","","
BothMicroEstimate","BothMicroSE","BothMicroPvalue","","BothCircEstimate","BothCircSE","BothCircPv
alue")


##Change all factors to numeric
NeuronCircTable_parietal[] <- lapply(NeuronCircTable_parietal, function(x) as.numeric(as.character(x)))

##Recover RNA
NeuronCircTable_parietal$RNA <- gsub("circ","",as.character(sigList_parietal_CDR_top))


#FE metafor
library(metafor)
meta_results <- function(x,parietalEst,BM44Est,parietalSE,BM44SE) {
Estimate <- c(parietalEst[x],BM44Est[x])
SE <- c(parietalSE[x],BM44SE[x])
rma(yi=Estimate,sei=SE, method="FE")$pval
}

NeuronCircTable_meta <-
NeuronCircTable_parietal[,c("RNA","NeuronPvalue","OligoPvalue","MicroPvalue","CircPvalue","BothNe
uronPvalue","BothOligoPvalue","BothMicroPvalue","BothCircPvalue")]

NeuronCircTable_meta$NeuronPvalue <- as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal),
meta_results, mc.cores=5, parietalEst=NeuronCircTable_parietal$NeuronEstimate,
BM44Est=NeuronCircTable_BM44$NeuronEstimate,parietalSE=NeuronCircTable_parietal$NeuronSE,
BM44SE=NeuronCircTable_BM44$NeuronSE)))

NeuronCircTable_meta$OligoPvalue <- as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal),
meta_results, mc.cores=5, parietalEst=NeuronCircTable_parietal$OligoEstimate,
BM44Est=NeuronCircTable_BM44$OligoEstimate,parietalSE=NeuronCircTable_parietal$OligoSE,
BM44SE=NeuronCircTable_BM44$OligoSE)))

NeuronCircTable_meta$MicroPvalue <- as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal),
meta_results, mc.cores=5, parietalEst=NeuronCircTable_parietal$MicroEstimate,
BM44Est=NeuronCircTable_BM44$MicroEstimate,parietalSE=NeuronCircTable_parietal$MicroSE,
BM44SE=NeuronCircTable_BM44$MicroSE)))

NeuronCircTable_meta$CircPvalue <- as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal),
meta_results, mc.cores=5, parietalEst=NeuronCircTable_parietal$CircEstimate,
```

```
BM44Est=NeuronCircTable_BM44$CircEstimate,parietalSE=NeuronCircTable_parietal$CircSE,
BM44SE=NeuronCircTable_BM44$CircSE)))

NeuronCircTable_meta$BothNeuronPvalue <-
as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal), meta_results, mc.cores=5,
parietalEst=NeuronCircTable_parietal$BothNeuronEstimate,
BM44Est=NeuronCircTable_BM44$BothNeuronEstimate,parietalSE=NeuronCircTable_parietal$BothNeu
ronSE, BM44SE=NeuronCircTable_BM44$BothNeuronSE)))

NeuronCircTable_meta$BothOligoPvalue <-
as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal), meta_results, mc.cores=5,
parietalEst=NeuronCircTable_parietal$BothOligoEstimate,
BM44Est=NeuronCircTable_BM44$BothOligoEstimate,parietalSE=NeuronCircTable_parietal$BothOligoS
E, BM44SE=NeuronCircTable_BM44$BothOligoSE)))

NeuronCircTable_meta$BothMicroPvalue <-
as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal), meta_results, mc.cores=5,
parietalEst=NeuronCircTable_parietal$BothMicroEstimate,
BM44Est=NeuronCircTable_BM44$BothMicroEstimate,parietalSE=NeuronCircTable_parietal$BothMicro
SE, BM44SE=NeuronCircTable_BM44$BothMicroSE)))

NeuronCircTable_meta$BothCircPvalue <-
as.numeric(unlist(mclapply(1:nrow(NeuronCircTable_parietal), meta_results, mc.cores=5,
parietalEst=NeuronCircTable_parietal$BothCircEstimate,
BM44Est=NeuronCircTable_BM44$BothCircEstimate,parietalSE=NeuronCircTable_parietal$BothCircSE,
BM44SE=NeuronCircTable_BM44$BothCircSE)))
```

Differential Expression Meta-Analyses

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Meta Analyses
```r

############################################
############################################
## Meta-analysis ##
############################################
############################################
```
```

```
library(metaRNASeq)
library(qqman)
##Update Manhattan code by using trace ##

#trace("manhattan",edit=TRUE)

#labs = topHits$SNP, cex = 0.7, font = 3

ChrPosGene_CDR <- unique(circCoordinates_parietal[,c(1,2,3,4)])
ChrPosGene_CDR <- subset(ChrPosGene_CDR, Gene != "not_annotated")
ChrPosGene_CDR$AveragePos <- round(((ChrPosGene_CDR$Start + ChrPosGene_CDR$End)/2))

ChrPosGene_CDR <- ChrPosGene_CDR[order(ChrPosGene_CDR$Gene, -
abs(ChrPosGene_CDR$AveragePos) ), ] #sort by id and reverse of abs(value)
ChrPosGene_CDR <- ChrPosGene_CDR[ !duplicated(ChrPosGene_CDR$Gene), ]  #keep the first
ChrPosGene_CDR$Gene <- paste0("circ",ChrPosGene_CDR$Gene)


ChrPosGene_PlaqueMean <- ChrPosGene_CDR
ChrPosGene_braak <- ChrPosGene_CDR
ChrPosGene_condition <- ChrPosGene_CDR

########################################
################ Combined Code for Meta-analyses ################
########################################

##Have to update the meta-analysis functions to deal with small pvalues

fishercomb <-
function(indpval,BHth = 0.05)
{
        listres = vector("list", 4)
        logpval=do.call(cbind,lapply(indpval, log))
        statc=apply(logpval,1, FUN=function(x) -2*sum(x,na.rm=TRUE))
        ## Added by Andrea for genes filtered in all samples
        ## (otherwise returns a value of 0)
        na.index <- which(apply(logpval, 1, function(x) sum(is.na(x))) == ncol(logpval))
        statc[na.index] <- NA

        notNA=apply(logpval,1,FUN=function(x) sum(!(is.na(x))))
        ## Update code for very small pvalues
```

```
	## Based on: https://stats.stackexchange.com/questions/194133/why-is-the-probability-of-my-chi-square-statistic-equal-to-0
	## Based on: https://stackoverflow.com/questions/40144267/calculating-miniscule-numbers-for-chi-squared-distribution-in-r-numerical-pre

	rpvalc_gen <- function(x) {
		y <- statc[[x]]
		if (y >= 83) {
			pchisq(y,df=(2*notNA[[x]]),lower.tail=FALSE)
		} else {
			1 - pchisq(y, df=(2*notNA[[x]]))
		}
	}

	rpvalc <- unlist(lapply(1:length(statc), rpvalc_gen))


	res = which(p.adjust(rpvalc, method = "BH") <= BHth)
	listres[[1]] = res
	listres[[2]] = statc
	listres[[3]] = rpvalc
	listres[[4]] = p.adjust(rpvalc, method = "BH")
	names(listres) = c("DEindices", "TestStatistic", "rawpval", "adjpval")
	return(listres)
}



invnorm <-
function(indpval, nrep, BHth = 0.05)
{
  listres = vector("list", 4)
  qnormpval= do.call(cbind,lapply(indpval, FUN=function(x) qnorm(1-x)))
  nrepcorr=t(apply(qnormpval,1,FUN=function(x){
  nrep2=nrep
  nrep2[which(is.na(x))]=0
  nrep2}))
  nreptot=apply(nrepcorr,1,sum)
  weight=sqrt(nrepcorr/nreptot)
  wqnormp=weight*qnormpval
  statc=apply(wqnormp,1, FUN=function(x) sum(x,na.rm=TRUE))
	## Added by Andrea for genes filtered in all samples
	## (otherwise returns a value of 0)
	nan.index <- which(apply(wqnormp, 1, function(x) sum(is.nan(x))) == ncol(wqnormp))
```

```
	statc[nan.index] <- NA

	## Update code for very small pvalues
	## Based on: https://stackoverflow.com/questions/5932276/adding-floating-point-precision-to-qnorm-pnorm-in-r
	rpvalc_gen <- function(x) {
		y <- statc[[x]]
		if (y >= 8) {
			rpvalc = pnorm(y, lower.tail = FALSE)
		} else {
			rpvalc = 1 - pnorm(y)
		}
	}

	rpvalc <- unlist(lapply(1:length(statc), rpvalc_gen))

  res = which(p.adjust(rpvalc, method = "BH") <= BHth)
  listres[[1]] = res
  listres[[2]] = statc
  listres[[3]] = rpvalc
  listres[[4]] = p.adjust(rpvalc, method = "BH")
  names(listres) = c("DEindices", "TestStatistic", "rawpval", "adjpval")
  return(listres)
}


meta_analyses <- function () {

##########
##########
## CDR BM ##
######


res_parietal_CDR_BM_trim <<- as.data.frame(res_parietal_CDR_filter)[
intersect(rownames(as.data.frame(res_parietal_CDR_filter)),
rownames(as.data.frame(res_BM_CDR_filter))), ]
res_BM_CDR_trim <<- as.data.frame(res_BM_CDR_filter)[
intersect(rownames(as.data.frame(res_parietal_CDR_filter)),
rownames(as.data.frame(res_BM_CDR_filter))), ]
rawpval_CDR <<-
list("pval1"=res_parietal_CDR_BM_trim[["pvalue"]],"pval2"=res_BM_CDR_trim[["pvalue"]])
```

```
FC_CDR <<-
list("FC1"=res_parietal_CDR_BM_trim[["log2FoldChange"]],"FC2"=res_BM_CDR_trim[["log2FoldChange"
]])
adjpval_CDR <<-
list("adjpval1"=res_parietal_CDR_BM_trim[["padj"]],"adjpval2"=res_BM_CDR_trim[["padj"]])
DE_CDR <<- mapply(adjpval_CDR, FUN=function(x) ifelse(x <= 0.05, 1, 0))
fishcomb_CDR <<- fishercomb(rawpval_CDR, BHth = 0.05)
invnormcomb_CDR <<-
invnorm(rawpval_CDR,nrep=c(nrow(colData(dds_parietal_CDR_filter)),nrow(colData(dds_BM_CDR_filte
r))), BHth = 0.05)


invnormcomb_CDR$DEindices <<- NULL
fishcomb_CDR$DEindices <<- NULL

fish_CDR <<- as.data.frame(fishcomb_CDR)
inv_CDR <<- as.data.frame(invnormcomb_CDR)

both_CDR_BM <<- cbind(fish_CDR, inv_CDR)

colnames(both_CDR_BM) <<- c("f_testStat", "f_rawpval", "f_adjpval", "i_testStat", "i_rawpval",
"i_adjpval")
rownames(both_CDR_BM) <<- rownames(res_BM_CDR_trim)
both_CDR_BM <<- both_CDR_BM[,c(2,5,3,6)]


both_CDR_BM <<- cbind(both_CDR_BM, Chr = ChrPosGene_CDR[,
"Chr"][match(rownames(both_CDR_BM), ChrPosGene_CDR$Gene)])
both_CDR_BM <<- cbind(both_CDR_BM, AveragePos = ChrPosGene_CDR[,
"AveragePos"][match(rownames(both_CDR_BM), ChrPosGene_CDR$Gene)])
both_CDR_BM$Chr <<- gsub("chr", "", both_CDR_BM$Chr)
both_CDR_BM$Chr <<- gsub("X", "23", both_CDR_BM$Chr)
both_CDR_BM$Chr <<- gsub("Y", "24", both_CDR_BM$Chr)
both_CDR_BM$Chr <<- as.integer(both_CDR_BM$Chr)
both_CDR_BM <<- both_CDR_BM[!is.na(both_CDR_BM$Chr),]
both_CDR_BM$Gene <<- rownames(both_CDR_BM)


#manhattan(both_CDR_BM, chr="Chr", bp = "AveragePos", p="f_rawpval", snp = "Gene",
genomewideline = -log10(0.05/nrow(both_CDR_BM)), suggestiveline = FALSE, annotatePval =
0.05/nrow(both_CDR_BM), annotateTop = FALSE, main="circRNA DE CDR Gene Based Meta Mend BM,
Fisher PVal")
```

```
#manhattan(both_CDR_BM, chr="Chr", bp = "AveragePos", p="i_rawpval", snp = "Gene",
genomewideline = -log10(0.05/nrow(both_CDR_BM)), suggestiveline = FALSE, annotatePval =
0.05/nrow(both_CDR_BM), annotateTop = FALSE, main="circRNA DE CDR Gene Based Meta Mend BM,
Stouffer PVal")




##########
##########
## Braak BM ##
######

res_parietal_braak_BM_trim <<- as.data.frame(res_parietal_braak_filter[
intersect(rownames(as.data.frame(res_parietal_braak_filter)),
rownames(as.data.frame(res_BM_braak_filter))), ])
res_BM_braak_trim <<- as.data.frame(res_BM_braak_filter[
intersect(rownames(as.data.frame(res_parietal_braak_filter)),
rownames(as.data.frame(res_BM_braak_filter))), ])
rawpval_braak <<-
list("pval1"=res_parietal_braak_BM_trim[["pvalue"]],"pval2"=res_BM_braak_trim[["pvalue"]])
FC_braak <<-
list("FC1"=res_parietal_braak_BM_trim[["log2FoldChange"]],"FC2"=res_BM_braak_trim[["log2FoldChan
ge"]])
adjpval_braak <<-
list("adjpval1"=res_parietal_braak_BM_trim[["padj"]],"adjpval2"=res_BM_braak_trim[["padj"]])
DE_braak <<- mapply(adjpval_braak, FUN=function(x) ifelse(x <= 0.05, 1, 0))
fishcomb_braak <<- fishercomb(rawpval_braak, BHth = 0.05)
invnormcomb_braak <<-
invnorm(rawpval_braak,nrep=c(nrow(colData(dds_parietal_braak_filter)),nrow(colData(dds_BM_braak_
filter))), BHth = 0.05)


invnormcomb_braak$DEindices <<- NULL
fishcomb_braak$DEindices <<- NULL

fish_braak <<- as.data.frame(fishcomb_braak)
inv_braak <<- as.data.frame(invnormcomb_braak)

both_braak_BM <<- cbind(fish_braak, inv_braak)

colnames(both_braak_BM) <<- c("f_testStat", "f_rawpval", "f_adjpval", "i_testStat", "i_rawpval",
"i_adjpval")
```

```
rownames(both_braak_BM) <<- rownames(res_BM_braak_trim)
both_braak_BM <<- both_braak_BM[,c(2,5,3,6)]


both_braak_BM <<- cbind(both_braak_BM, Chr = ChrPosGene_braak[,
"Chr"][match(rownames(both_braak_BM), ChrPosGene_braak$Gene)])
both_braak_BM <<- cbind(both_braak_BM, AveragePos = ChrPosGene_braak[,
"AveragePos"][match(rownames(both_braak_BM), ChrPosGene_braak$Gene)])
both_braak_BM$Chr <<- gsub("chr", "", both_braak_BM$Chr)
both_braak_BM$Chr <<- gsub("X", "23", both_braak_BM$Chr)
both_braak_BM$Chr <<- gsub("Y", "24", both_braak_BM$Chr)
both_braak_BM$Chr <<- as.integer(both_braak_BM$Chr)
both_braak_BM <<- both_braak_BM[!is.na(both_braak_BM$Chr),]
both_braak_BM$Gene <<- rownames(both_braak_BM)


#manhattan(both_braak_BM, chr="Chr", bp = "AveragePos", p="f_rawpval", snp = "Gene",
genomewideline = -log10(0.05/nrow(both_braak_BM)), suggestiveline = FALSE, annotatePval =
0.05/nrow(both_braak_BM), annotateTop = FALSE, main="circRNA DE braak Gene Based Meta Mend
BM, Fisher PVal")

#manhattan(both_braak_BM, chr="Chr", bp = "AveragePos", p="i_rawpval", snp = "Gene",
genomewideline = -log10(0.05/nrow(both_braak_BM)), suggestiveline = FALSE, annotatePval =
0.05/nrow(both_braak_BM), annotateTop = FALSE, main="circRNA DE braak Gene Based Meta Mend
BM, Stouffer PVal")


##########
##########
## condition BM ##
######

res_parietal_condition_BM_trim <<- as.data.frame(res_parietal_condition_filter[
intersect(rownames(as.data.frame(res_parietal_condition_filter)),
rownames(as.data.frame(res_BM_condition_filter))), ])
res_BM_condition_trim <<- as.data.frame(res_BM_condition_filter[
intersect(rownames(as.data.frame(res_parietal_condition_filter)),
rownames(as.data.frame(res_BM_condition_filter))), ])
rawpval_condition <<-
list("pval1"=res_parietal_condition_BM_trim[["pvalue"]],"pval2"=res_BM_condition_trim[["pvalue"]])
FC_condition <<-
list("FC1"=res_parietal_condition_BM_trim[["log2FoldChange"]],"FC2"=res_BM_condition_trim[["log2F
oldChange"]])
```

```
adjpval_condition <-
list("adjpval1"=res_parietal_condition_BM_trim[["padj"]],"adjpval2"=res_BM_condition_trim[["padj"]])
DE_condition <- mapply(adjpval_condition, FUN=function(x) ifelse(x <= 0.05, 1, 0))
fishcomb_condition <- fishercomb(rawpval_condition, BHth = 0.05)
invnormcomb_condition <-
invnorm(rawpval_condition,nrep=c(nrow(colData(dds_parietal_condition_filter)),nrow(colData(dds_BM
_condition_filter))), BHth = 0.05)


invnormcomb_condition$DEindices <- NULL
fishcomb_condition$DEindices <- NULL

fish_condition <- as.data.frame(fishcomb_condition)
inv_condition <- as.data.frame(invnormcomb_condition)

both_condition_BM <- cbind(fish_condition, inv_condition)

colnames(both_condition_BM) <- c("f_testStat", "f_rawpval", "f_adjpval", "i_testStat", "i_rawpval",
"i_adjpval")
rownames(both_condition_BM) <- rownames(res_BM_condition_trim)
both_condition_BM <- both_condition_BM[,c(2,5,3,6)]


both_condition_BM <- cbind(both_condition_BM, Chr = ChrPosGene_condition[,
"Chr"][match(rownames(both_condition_BM), ChrPosGene_condition$Gene)])
both_condition_BM <- cbind(both_condition_BM, AveragePos = ChrPosGene_condition[,
"AveragePos"][match(rownames(both_condition_BM), ChrPosGene_condition$Gene)])
both_condition_BM$Chr <- gsub("chr", "", both_condition_BM$Chr)
both_condition_BM$Chr <- gsub("X", "23", both_condition_BM$Chr)
both_condition_BM$Chr <- gsub("Y", "24", both_condition_BM$Chr)
both_condition_BM$Chr <- as.integer(both_condition_BM$Chr)
both_condition_BM <- both_condition_BM[!is.na(both_condition_BM$Chr),]
both_condition_BM$Gene <- rownames(both_condition_BM)


#manhattan(both_condition_BM, chr="Chr", bp = "AveragePos", p="f_rawpval", snp = "Gene",
genomewideline = -log10(0.05/nrow(both_condition_BM)), suggestiveline = FALSE, annotatePval =
0.05/nrow(both_condition_BM), annotateTop = FALSE, main="circRNA DE condition Gene Based Meta
Mend BM, Fisher PVal")

#manhattan(both_condition_BM, chr="Chr", bp = "AveragePos", p="i_rawpval", snp = "Gene",
genomewideline = -log10(0.05/nrow(both_condition_BM)), suggestiveline = FALSE, annotatePval =
0.05/nrow(both_condition_BM), annotateTop = FALSE, main="circRNA DE condition Gene Based Meta
Mend BM, Stouffer PVal")
```

```
}


##BM10
meta_analyses_BM10 <- gsub("BM","BM10",as.list(body(meta_analyses)))
#Replace first line to create function
meta_analyses_BM10[1] <- "meta_analyses_BM10 <- function() {"
#Add last line to end function
meta_analyses_BM10[length(meta_analyses_BM10)+1] <- "}"
write(meta_analyses_BM10, "/home/dubeu/Synapse/meta_analyses_BM10.R")

##BM22
meta_analyses_BM22 <- gsub("BM","BM22",as.list(body(meta_analyses)))
#Replace first line to create function
meta_analyses_BM22[1] <- "meta_analyses_BM22 <- function() {"
#Add last line to end function
meta_analyses_BM22[length(meta_analyses_BM22)+1] <- "}"
write(meta_analyses_BM22, "/home/dubeu/Synapse/meta_analyses_BM22.R")

##BM36
meta_analyses_BM36 <- gsub("BM","BM36",as.list(body(meta_analyses)))
#Replace first line to create function
meta_analyses_BM36[1] <- "meta_analyses_BM36 <- function() {"
#Add last line to end function
meta_analyses_BM36[length(meta_analyses_BM36)+1] <- "}"
write(meta_analyses_BM36, "/home/dubeu/Synapse/meta_analyses_BM36.R")

##BM44
meta_analyses_BM44 <- gsub("BM","BM44",as.list(body(meta_analyses)))
#Replace first line to create function
meta_analyses_BM44[1] <- "meta_analyses_BM44 <- function() {"
#Add last line to end function
meta_analyses_BM44[length(meta_analyses_BM44)+1] <- "}"
write(meta_analyses_BM44, "/home/dubeu/Synapse/meta_analyses_BM44.R")



source('/home/dubeu/Synapse/meta_analyses_BM10.R')
source('/home/dubeu/Synapse/meta_analyses_BM22.R')
source('/home/dubeu/Synapse/meta_analyses_BM36.R')
source('/home/dubeu/Synapse/meta_analyses_BM44.R')
```

```
meta_analyses_BM10()
meta_analyses_BM22()
meta_analyses_BM36()
meta_analyses_BM44()
```

Pre-symptomatic Bootstrap Correlation Distribution Analyses

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: PreSymptomatic Analyses
```{r}


#####################################################
#####################################################
### Presymptomatic Correlation Based Analysis ###
#####################################################
#####################################################

presymp_analyses <- function () {

#####################################
##DE based on BM No PreSymp vs. HCs ##
#####################################


##DE based on conditionNoPre ##

dds_BM_conditionNoPre_filter <<- dds_BM_condition_filter

#Remove PreSymptomatic
dds_BM_conditionNoPre_filter <<- dds_BM_conditionNoPre_filter[,
!(dds_BM_conditionNoPre_filter$PreSymptomatic == 1)]

dds_BM_conditionNoPre_filter$NP.1 <<- droplevels(dds_BM_conditionNoPre_filter$NP.1)

dds_BM_conditionNoPre_filter$batch <<- droplevels(dds_BM_conditionNoPre_filter$batch)

levels(dds_BM_conditionNoPre_filter$NP.1) <<- list(HC=c("1"), AD=c("2", "3", "4"))
```
```

```
design(dds_BM_conditionNoPre_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2
+ NP.1)


dds_BM_final_conditionNoPre_filter <<- DESeq(dds_BM_conditionNoPre_filter, parallel = TRUE,
betaPrior = FALSE)

res_BM_conditionNoPre_filter <<- results(dds_BM_final_conditionNoPre_filter, contrast=c("NP.1",
"AD", "HC"), alpha = 0.05)
summary(res_BM_conditionNoPre_filter)

#res_BM_conditionNoPre_filter <<- results(dds_BM_final_conditionNoPre_filter, contrast=c("NP.1", "2",
"1"), alpha = 0.05)
#summary(res_BM_conditionNoPre_filter)


#####################################
##DE based on BM PreSymp vs. HCs ##
#####################################

dds_BM_PreSymp_filter <<- dds_BM_condition_filter

dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[ ,!is.na(dds_BM_PreSymp_filter$batch)]

dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[ ,!is.na(dds_BM_PreSymp_filter$PC1)]
dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[ ,!is.na(dds_BM_PreSymp_filter$PC2)]

#Only CDR 0.5 or less

dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[ ,!is.na(dds_BM_PreSymp_filter$CDR)]
dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[, !(dds_BM_PreSymp_filter$CDR >= 1) ]

#Only braak 2 or less

#dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[ ,!is.na(dds_BM_PreSymp_filter$bbscore)]
#dds_BM_PreSymp_filter <<- dds_BM_PreSymp_filter[, !(dds_BM_PreSymp_filter$PreSymptomatic == 1
& dds_BM_PreSymp_filter$bbscore <= 2) ]

dds_BM_PreSymp_filter$batch <<- droplevels(dds_BM_PreSymp_filter$batch)
dds_BM_PreSymp_filter$NP.1 <<- droplevels(dds_BM_PreSymp_filter$NP.1)

levels(dds_BM_PreSymp_filter$NP.1) <<- list(HC=c("1"), AD=c("2", "3", "4"))
```

```
design(dds_BM_PreSymp_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 +
NP.1)

dds_BM_final_PreSymp_filter <<- DESeq(dds_BM_PreSymp_filter, parallel = TRUE, betaPrior = FALSE)

res_BM_PreSymp_filter <<- results(dds_BM_final_PreSymp_filter, contrast=c("NP.1", "AD", "HC"), alpha
= 0.05)
#res_BM_PreSymp_filter <<- results(dds_BM_final_PreSymp_filter, contrast=c("NP.1", "2", "1"), alpha =
0.05)
summary(res_BM_PreSymp_filter)


#https://stats.idre.ucla.edu/r/faq/how-can-i-generate-bootstrap-statistics-in-r/
#https://stats.stackexchange.com/questions/246211/bootstrap-to-test-differences-between-
correlation-coefficients


sigList_BM_condition <<- rownames(subset(res_BM_conditionNoPre_filter, padj < 0.05))

Presymptomatic_BM_sig <<- merge(as.data.frame(res_BM_PreSymp_filter[sigList_BM_condition,]),
as.data.frame(res_BM_conditionNoPre_filter), by='row.names')

Presymptomatic_BM_notsig <<- merge(as.data.frame(res_BM_PreSymp_filter),
as.data.frame(res_BM_conditionNoPre_filter), by='row.names')
rownames(Presymptomatic_BM_notsig) <<- Presymptomatic_BM_notsig$Row.names
Presymptomatic_BM_notsig <<- Presymptomatic_BM_notsig[ !(
rownames(Presymptomatic_BM_notsig) %in% sigList_BM_condition),]
Presymptomatic_BM_notsig <<- na.omit(Presymptomatic_BM_notsig)

fc <<- function(d, i){
  d2 <<- d[i,]
  return(cor(d2$log2FoldChange.x, d2$log2FoldChange.y, use = "complete.obs"))
}


bootcorr_presymptom_BM_sig <<- boot(Presymptomatic_BM_sig, fc, R=10000)

bootcorr_presymptom_BM_notsig <<- boot(Presymptomatic_BM_notsig, fc, R=10000)


bootcorr_presymptom_BM_sig_CI <<- boot.ci(boot.out = bootcorr_presymptom_BM_sig, type = "all")

bootcorr_presymptom_BM_notsig_CI <<- boot.ci(boot.out = bootcorr_presymptom_BM_notsig, type =
"all")
```

```
}


##BM10
presymp_analyses_BM10 <- gsub("BM","BM10",as.list(body(presymp_analyses)))
#Replace first line to create function
presymp_analyses_BM10[1] <- "presymp_analyses_BM10 <- function() {"
#Replace first line to end function
presymp_analyses_BM10[length(presymp_analyses_BM10)+1] <- "}"
write(presymp_analyses_BM10, "/home/dubeu/Synapse/presymp_analyses_BM10.R")

##BM22
presymp_analyses_BM22 <- gsub("BM","BM22",as.list(body(presymp_analyses)))
#Replace first line to create function
presymp_analyses_BM22[1] <- "presymp_analyses_BM22 <- function() {"
#Replace first line to end function
presymp_analyses_BM22[length(presymp_analyses_BM22)+1] <- "}"
write(presymp_analyses_BM22, "/home/dubeu/Synapse/presymp_analyses_BM22.R")

##BM36
presymp_analyses_BM36 <- gsub("BM","BM36",as.list(body(presymp_analyses)))
#Replace first line to create function
presymp_analyses_BM36[1] <- "presymp_analyses_BM36 <- function() {"
#Replace first line to end function
presymp_analyses_BM36[length(presymp_analyses_BM36)+1] <- "}"
write(presymp_analyses_BM36, "/home/dubeu/Synapse/presymp_analyses_BM36.R")

##BM44
presymp_analyses_BM44 <- gsub("BM","BM44",as.list(body(presymp_analyses)))
#Replace first line to create function
presymp_analyses_BM44[1] <- "presymp_analyses_BM44 <- function() {"
#Replace first line to end function
presymp_analyses_BM44[length(presymp_analyses_BM44)+1] <- "}"
write(presymp_analyses_BM44, "/home/dubeu/Synapse/presymp_analyses_BM44.R")



source('/home/dubeu/Synapse/presymp_analyses_BM10.R')
source('/home/dubeu/Synapse/presymp_analyses_BM22.R')
source('/home/dubeu/Synapse/presymp_analyses_BM36.R')
source('/home/dubeu/Synapse/presymp_analyses_BM44.R')
```

```
presymp_analyses_BM10()
presymp_analyses_BM22()
presymp_analyses_BM36()
presymp_analyses_BM44()



##DE based on conditionNoPre ##

############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
############

dds_parietal_conditionNoPre_filter <- dds_parietal_condition_filter

#Remove PreSymptomatic
dds_parietal_conditionNoPre_filter <- dds_parietal_conditionNoPre_filter[,
!(dds_parietal_conditionNoPre_filter$PreSymptomatic == 1) ]


#Variable under invesitgation is at the end of the design formula
design(dds_parietal_conditionNoPre_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool +
GENDER + PC1 + PC2 + condition)

#Run the actual DESeq2 function
dds_parietal_final_conditionNoPre_filter <- DESeq(dds_parietal_conditionNoPre_filter, parallel = TRUE,
betaPrior = FALSE)

##Analyze results based on contrast

res_parietal_conditionNoPre_filter <- results(dds_parietal_final_conditionNoPre_filter,
contrast=c("condition", "LOAD", "CO"), alpha = 0.05)
summary(res_parietal_conditionNoPre_filter)



########################################
##DE based on parietal PreSymp vs. HCs ##
########################################

dds_parietal_PreSymp_filter <- dds_parietal_condition_filter
```

```
#Only CDR 0.5 or less

dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[
,!is.na(dds_parietal_PreSymp_filter$CDR_e)]
dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[, !(dds_parietal_PreSymp_filter$CDR_e >=
1) ]

#Only braak 2 or less

#dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[
,!is.na(dds_parietal_PreSymp_filter$braak_final)]
#dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[,
!(dds_parietal_PreSymp_filter$PreSymptomatic == 1 & dds_parietal_PreSymp_filter$braak_final <= 2) ]

dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[
,!is.na(dds_parietal_PreSymp_filter$AGE_at_death)]

dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[ ,!is.na(dds_parietal_PreSymp_filter$PMI)]

dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[ ,!is.na(dds_parietal_PreSymp_filter$PC1)]
dds_parietal_PreSymp_filter <- dds_parietal_PreSymp_filter[ ,!is.na(dds_parietal_PreSymp_filter$PC2)]
dds_parietal_PreSymp_filter$condition <- droplevels(dds_parietal_PreSymp_filter$condition)

design(dds_parietal_PreSymp_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool + GENDER +
PC1 + PC2 + condition)


dds_parietal_final_PreSymp_filter <- DESeq(dds_parietal_PreSymp_filter, parallel = TRUE, betaPrior =
FALSE)

##Analyze results based on contrast

res_parietal_PreSymp_filter <- results(dds_parietal_final_PreSymp_filter, contrast=c("condition",
"LOAD", "CO"), alpha = 0.05)
summary(res_parietal_PreSymp_filter)

#https://stats.idre.ucla.edu/r/faq/how-can-i-generate-bootstrap-statistics-in-r/
#https://stats.stackexchange.com/questions/246211/bootstrap-to-test-differences-between-
correlation-coefficients

sigList_parietal_condition <- rownames(subset(res_parietal_conditionNoPre_filter, padj < 0.05))
```

```
Presymptomatic_parietal_sig <-
merge(as.data.frame(res_parietal_PreSymp_filter[sigList_parietal_condition,]),
as.data.frame(res_parietal_conditionNoPre_filter), by='row.names')

Presymptomatic_parietal_notsig <- merge(as.data.frame(res_parietal_PreSymp_filter),
as.data.frame(res_parietal_conditionNoPre_filter), by='row.names')
rownames(Presymptomatic_parietal_notsig) <- Presymptomatic_parietal_notsig$Row.names
Presymptomatic_parietal_notsig <- Presymptomatic_parietal_notsig[ !(
rownames(Presymptomatic_parietal_notsig) %in% sigList_parietal_condition),]
Presymptomatic_parietal_notsig <- na.omit(Presymptomatic_parietal_notsig)

fc <- function(d, i){
   d2 <- d[i,]
   return(cor(d2$log2FoldChange.x, d2$log2FoldChange.y, use = "complete.obs"))
}


bootcorr_presymptom_parietal_sig <- boot(Presymptomatic_parietal_sig, fc, R=10000)

bootcorr_presymptom_parietal_notsig <- boot(Presymptomatic_parietal_notsig, fc, R=10000)


bootcorr_presymptom_parietal_sig_CI <- boot.ci(boot.out = bootcorr_presymptom_parietal_sig, type =
"all")

bootcorr_presymptom_parietal_notsig_CI <- boot.ci(boot.out = bootcorr_presymptom_parietal_notsig,
type = "all")



##########################################
##########################################
## PreSymp Same Direction between the studies ##
##########################################
##########################################

###BM44

##Is the Presymp Log2FC in the same direction as the full analysis Log2FC

sign_BM44_condition_top <- sign(as.data.frame(subset(res_BM44_conditionNoPre_filter, padj <
0.05))$log2FoldChange)
```

```
sign_BM44_PreSymp_top <-
sign(as.data.frame(res_BM44_PreSymp_filter[rownames(as.data.frame(subset(res_BM44_conditionNoPre_filter, padj < 0.05))),])$log2FoldChange)


sign_BM44_consistent_conditionPresymp <-
rownames(as.data.frame(subset(res_BM44_conditionNoPre_filter, padj <
0.05)))[sign_BM44_condition_top == sign_BM44_PreSymp_top]


sign_parietal_condition_top <- sign(as.data.frame(subset(res_parietal_conditionNoPre_filter, pvalue <
0.05))$log2FoldChange)

sign_parietal_PreSymp_top <-
sign(as.data.frame(res_parietal_PreSymp_filter[rownames(as.data.frame(subset(res_parietal_conditionNoPre_filter, pvalue < 0.05))),])$log2FoldChange)

sign_parietal_consistent_conditionPresymp <-
rownames(as.data.frame(subset(res_parietal_conditionNoPre_filter, pvalue <
0.05)))[sign_parietal_condition_top == sign_parietal_PreSymp_top]

intersect(sign_parietal_consistent_conditionPresymp, sign_BM44_consistent_conditionPresymp)

res_BM44_PreSymp_filter[intersect(sign_parietal_consistent_conditionPresymp,
sign_BM44_consistent_conditionPresymp),]

res_parietal_PreSymp_filter[intersect(sign_parietal_consistent_conditionPresymp,
sign_BM44_consistent_conditionPresymp),]


###BM36

##Is the Presymp Log2FC in the same direction as the full analysis Log2FC

sign_BM36_condition_top <- sign(as.data.frame(subset(res_BM36_conditionNoPre_filter, padj <
0.05))$log2FoldChange)

sign_BM36_PreSymp_top <-
sign(as.data.frame(res_BM36_PreSymp_filter[rownames(as.data.frame(subset(res_BM36_conditionNoPre_filter, padj < 0.05))),])$log2FoldChange)
```

```
sign_BM36_consistent_conditionPresymp <-
rownames(as.data.frame(subset(res_BM36_conditionNoPre_filter, padj <
0.05)))[sign_BM36_condition_top == sign_BM36_PreSymp_top]


sign_parietal_condition_top <- sign(as.data.frame(subset(res_parietal_conditionNoPre_filter, pvalue <
0.05))$log2FoldChange)

sign_parietal_PreSymp_top <-
sign(as.data.frame(res_parietal_PreSymp_filter[rownames(as.data.frame(subset(res_parietal_condition
NoPre_filter, pvalue < 0.05))),])$log2FoldChange)

sign_parietal_consistent_conditionPresymp <-
rownames(as.data.frame(subset(res_parietal_conditionNoPre_filter, pvalue <
0.05)))[sign_parietal_condition_top == sign_parietal_PreSymp_top]

intersect(sign_parietal_consistent_conditionPresymp, sign_BM36_consistent_conditionPresymp)

res_BM36_PreSymp_filter[intersect(sign_parietal_consistent_conditionPresymp,
sign_BM36_consistent_conditionPresymp),]




##############################################
##############################################
## PreSymp Numbers for the Paper ##
##############################################
##############################################

nrow(subset(Presymptomatic_BM44_sig, pvalue.x < 0.05))
nrow(Presymptomatic_BM44_sig)

sign(subset(Presymptomatic_BM44_sig, pvalue.x <
0.05)$log2FoldChange.x)==sign(subset(Presymptomatic_BM44_sig, pvalue.x < 0.05)$log2FoldChange.y)


```
```

Proportion of Variation Explained Analyses

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
```
```

##Contact: udube@wustl.edu

##Code Section: Proportion of Variance Explained Analyses
```{r}

```
##
##
Proportion of Variance Explained AMP_AD
##
##

PropVar_analyses <- function() {

##Estimated proportion of variance for CDR

PropVar_BM_CDR_Neuron_anova <<- data.frame()
PropVar_BM_CDR_Neuron_relaimpo <<- data.frame()
Gene_archive_BM_CDR <<- data.frame(row.names = rownames(t(norm_BM_CDR_counts_regress)))
# this will get the intersection of the row.names for everything in the list
sigList_BM_CDR = subset(as.data.frame(both_CDR_BM), i_adjpval < 0.05)
sigList_BM_CDR = sigList_BM_CDR[order(sigList_BM_CDR$i_adjpval),]
sigList_BM_CDR = rownames(sigList_BM_CDR)
sigList_BM_CDR_top <<- head(sigList_BM_CDR, n = 10)

for (i in sigList_BM_CDR_top) {

	print(i)

	print(match(i,sigList_BM_CDR_top)/length(sigList_BM_CDR_top))

	Gene_BM_CDR <<- t(norm_BM_CDR_counts_regress[i,])
  regress_BM_CDR <<- cbind(coldata_BM[rownames(Gene_BM_CDR),], Gene =
t(norm_BM_CDR_counts_regress[i,]))

  #regress_BM_CDR$RACE <<- relevel(regress_BM_CDR$RACE, "W")

	Gene_archive_BM_CDR <<- cbind(Gene_archive_BM_CDR,
as.data.frame(t(norm_BM_CDR_counts_regress[i,])))

  model_regress_BM_CDR = lm(CDR ~ ApoE4_SNP + Neuron + regress_BM_CDR[,i],
data=regress_BM_CDR)

```
        coefs <<- data.frame(coef(summary(model_regress_BM_CDR)))

        af <<- anova(model_regress_BM_CDR)
        afss <<- af$"Sum Sq"
        anova_BM <<- data.frame()
        anova_BM <<- (cbind(af,PctExp=afss/sum(afss)*100))
        anova_BM <<- anova_BM[(nrow(anova_BM)-1),ncol(anova_BM)]
        coefs <<- cbind(coefs, PropVar = anova_BM)



        PropVar_BM_CDR_Neuron_anova <<- bind_rows(PropVar_BM_CDR_Neuron_anova, i =
coefs[(nrow(coefs)-1),])
        PropVar_BM_CDR_Neuron_relaimpo <<- bind_rows(PropVar_BM_CDR_Neuron_relaimpo,
relaimpo = ((calc.relimp(model_regress_BM_CDR, type = c("lmg"), rela = FALSE))$lmg[3])*100)
}

rownames(PropVar_BM_CDR_Neuron_anova) <<- sigList_BM_CDR_top
rownames(PropVar_BM_CDR_Neuron_relaimpo) <<- sigList_BM_CDR_top

regress_BM_CDR_Final <<- cbind(coldata_BM[rownames(Gene_BM_CDR),], Gene_archive_BM_CDR)

formula_BM_CDR = paste("CDR ~ ApoE4_SNP + Neuron +", paste(paste0('`', sigList_BM_CDR_top, '`'),
collapse =" + "))

fullmodel_BM_CDR <<- lm(formula_BM_CDR, data = regress_BM_CDR_Final)

af <<- anova(fullmodel_BM_CDR)
afss <<- af$"Sum Sq"
PropVar_fullmodel_BM_CDR_Neuron_anova <<- (cbind(af,PctExp=afss/sum(afss)*100))
PropVar_fullmodel_BM_CDR_Neuron_relaimpo <<- as.data.frame(calc.relimp(fullmodel_BM_CDR, type
= c("lmg"), rela = FALSE)$lmg)
colnames(PropVar_fullmodel_BM_CDR_Neuron_relaimpo) <<- "relaimpo_lmg"
PropVar_fullmodel_BM_CDR_Neuron_relaimpo$relaimpo_lmg <<-
PropVar_fullmodel_BM_CDR_Neuron_relaimpo$relaimpo_lmg*100



##Estimated proportion of variance for braak

PropVar_BM_braak_Neuron_anova <<- data.frame()
PropVar_BM_braak_Neuron_relaimpo <<- data.frame()
Gene_archive_BM_braak <<- data.frame(row.names = rownames(t(norm_BM_braak_counts_regress)))
```

```
# this will get the intersection of the row.names for everything in the list
sigList_BM_braak = subset(as.data.frame(both_braak_BM), i_adjpval < 0.05)
sigList_BM_braak = sigList_BM_braak[order(sigList_BM_braak$i_adjpval),]
sigList_BM_braak = rownames(sigList_BM_braak)
sigList_BM_braak_top <<- head(sigList_BM_braak, n = 10)



for (i in sigList_BM_braak_top) {

        print(i)

        print(match(i,sigList_BM_braak_top)/length(sigList_BM_braak_top))

        Gene_BM_braak <<- t(norm_BM_braak_counts_regress[i,])
   regress_BM_braak <<- cbind(coldata_BM[rownames(Gene_BM_braak),], Gene =
t(norm_BM_braak_counts_regress[i,]))

   #regress_BM_braak$RACE <<- relevel(regress_BM_braak$RACE, "W")

        Gene_archive_BM_braak <<- cbind(Gene_archive_BM_braak,
as.data.frame(t(norm_BM_braak_counts_regress[i,])))

   model_regress_BM_braak = lm(bbscore ~ ApoE4_SNP + Neuron + regress_BM_braak[,i],
data=regress_BM_braak )

        coefs <<- data.frame(coef(summary(model_regress_BM_braak)))

        af <<- anova(model_regress_BM_braak)
        afss <<- af$"Sum Sq"
        anova_BM <<- data.frame()
        anova_BM <<- (cbind(af,PctExp=afss/sum(afss)*100))
        anova_BM <<- anova_BM[(nrow(anova_BM)-1),ncol(anova_BM)]
        coefs <<- cbind(coefs, PropVar = anova_BM)



        PropVar_BM_braak_Neuron_anova <<- bind_rows(PropVar_BM_braak_Neuron_anova, i =
coefs[(nrow(coefs)-1),])
        PropVar_BM_braak_Neuron_relaimpo <<- bind_rows(PropVar_BM_braak_Neuron_relaimpo,
relaimpo = ((calc.relimp(model_regress_BM_braak, type = c("lmg"), rela = FALSE))$lmg[3])*100)
}

rownames(PropVar_BM_braak_Neuron_anova) <<- sigList_BM_braak_top
```

```
rownames(PropVar_BM_braak_Neuron_relaimpo) <<- sigList_BM_braak_top

regress_BM_braak_Final <<- cbind(coldata_BM[rownames(Gene_BM_braak),],
Gene_archive_BM_braak)

formula_BM_braak = paste("bbscore ~ ApoE4_SNP + Neuron +", paste(paste0('`', sigList_BM_braak_top,
'`'),  collapse =" + "))

fullmodel_BM_braak <<- lm(formula_BM_braak, data = regress_BM_braak_Final)

af <<- anova(fullmodel_BM_braak)
afss <<- af$"Sum Sq"
PropVar_fullmodel_BM_braak_Neuron_anova <<- (cbind(af,PctExp=afss/sum(afss)*100))
PropVar_fullmodel_BM_braak_Neuron_relaimpo <<- as.data.frame(calc.relimp(fullmodel_BM_braak,
type = c("lmg"), rela = FALSE)$lmg)
colnames(PropVar_fullmodel_BM_braak_Neuron_relaimpo) <<- "relaimpo_lmg"
PropVar_fullmodel_BM_braak_Neuron_relaimpo$relaimpo_lmg <<-
PropVar_fullmodel_BM_braak_Neuron_relaimpo$relaimpo_lmg*100



##Estimated proportion of variance for PlaqueMean

PropVar_BM_PlaqueMean_Neuron_anova <<- data.frame()
PropVar_BM_PlaqueMean_Neuron_relaimpo <<- data.frame()
Gene_archive_BM_PlaqueMean <<- data.frame(row.names =
rownames(t(norm_BM_PlaqueMean_counts_regress)))
# this will get the intersection of the row.names for everything in the list
sigList_BM_PlaqueMean = subset(as.data.frame(res_BM_PlaqueMean_filter), padj < 0.05)
sigList_BM_PlaqueMean = sigList_BM_PlaqueMean[order(sigList_BM_PlaqueMean$padj),]
sigList_BM_PlaqueMean = rownames(sigList_BM_PlaqueMean)
sigList_BM_PlaqueMean_top <<- head(sigList_BM_PlaqueMean, n = 10)



for (i in sigList_BM_PlaqueMean_top) {

        print(i)

        print(match(i,sigList_BM_PlaqueMean_top)/length(sigList_BM_PlaqueMean_top))

        Gene_BM_PlaqueMean <<- t(norm_BM_PlaqueMean_counts_regress[i,])
  regress_BM_PlaqueMean <<- cbind(coldata_BM[rownames(Gene_BM_PlaqueMean),], Gene =
t(norm_BM_PlaqueMean_counts_regress[i,]))
```

```
   #regress_BM_PlaqueMean$RACE <<- relevel(regress_BM_PlaqueMean$RACE, "W")

          Gene_archive_BM_PlaqueMean <<- cbind(Gene_archive_BM_PlaqueMean,
as.data.frame(t(norm_BM_PlaqueMean_counts_regress[i,])))

   model_regress_BM_PlaqueMean = lm(PlaqueMean ~ ApoE4_SNP + Neuron +
regress_BM_PlaqueMean[,i], data=regress_BM_PlaqueMean )

          coefs <<- data.frame(coef(summary(model_regress_BM_PlaqueMean)))

          af <<- anova(model_regress_BM_PlaqueMean)
          afss <<- af$"Sum Sq"
          anova_BM <<- data.frame()
          anova_BM <<- (cbind(af,PctExp=afss/sum(afss)*100))
          anova_BM <<- anova_BM[(nrow(anova_BM)-1),ncol(anova_BM)]
          coefs <<- cbind(coefs, PropVar = anova_BM)



          PropVar_BM_PlaqueMean_Neuron_anova <<-
bind_rows(PropVar_BM_PlaqueMean_Neuron_anova, i = coefs[(nrow(coefs)-1),])
          PropVar_BM_PlaqueMean_Neuron_relaimpo <<-
bind_rows(PropVar_BM_PlaqueMean_Neuron_relaimpo, relaimpo =
((calc.relimp(model_regress_BM_PlaqueMean, type = c("lmg"), rela = FALSE))$lmg[3])*100)
}

rownames(PropVar_BM_PlaqueMean_Neuron_anova) <<- sigList_BM_PlaqueMean_top
rownames(PropVar_BM_PlaqueMean_Neuron_relaimpo) <<- sigList_BM_PlaqueMean_top

regress_BM_PlaqueMean_Final <<- cbind(coldata_BM[rownames(Gene_BM_PlaqueMean),],
Gene_archive_BM_PlaqueMean)

formula_BM_PlaqueMean = paste("PlaqueMean ~ ApoE4_SNP + Neuron +", paste(paste0('`',
sigList_BM_PlaqueMean_top, '`'),  collapse =" + "))

fullmodel_BM_PlaqueMean <<- lm(formula_BM_PlaqueMean, data = regress_BM_PlaqueMean_Final)

af <<- anova(fullmodel_BM_PlaqueMean)
afss <<- af$"Sum Sq"
PropVar_fullmodel_BM_PlaqueMean_Neuron_anova <<- (cbind(af,PctExp=afss/sum(afss)*100))
PropVar_fullmodel_BM_PlaqueMean_Neuron_relaimpo <<-
as.data.frame(calc.relimp(fullmodel_BM_PlaqueMean, type = c("lmg"), rela = FALSE)$lmg)
colnames(PropVar_fullmodel_BM_PlaqueMean_Neuron_relaimpo) <<- "relaimpo_lmg"
```

```
PropVar_fullmodel_BM_PlaqueMean_Neuron_relaimpo$relaimpo_lmg <<-
PropVar_fullmodel_BM_PlaqueMean_Neuron_relaimpo$relaimpo_lmg*100

}


##BM10
PropVar_analyses_BM10 <- gsub("BM","BM10",as.list(body(PropVar_analyses)))
#Replace first line to create function
PropVar_analyses_BM10[1] <- "PropVar_analyses_BM10 <- function() {"
#Add last line to end function
PropVar_analyses_BM10[length(PropVar_analyses_BM10)+1] <- "}"
write(PropVar_analyses_BM10, "/home/dubeu/Synapse/PropVar_analyses_BM10.R")

##BM22
PropVar_analyses_BM22 <- gsub("BM","BM22",as.list(body(PropVar_analyses)))
#Replace first line to create function
PropVar_analyses_BM22[1] <- "PropVar_analyses_BM22 <- function() {"
#Add last line to end function
PropVar_analyses_BM22[length(PropVar_analyses_BM22)+1] <- "}"
write(PropVar_analyses_BM22, "/home/dubeu/Synapse/PropVar_analyses_BM22.R")

##BM36
PropVar_analyses_BM36 <- gsub("BM","BM36",as.list(body(PropVar_analyses)))
#Replace first line to create function
PropVar_analyses_BM36[1] <- "PropVar_analyses_BM36 <- function() {"
#Add last line to end function
PropVar_analyses_BM36[length(PropVar_analyses_BM36)+1] <- "}"
write(PropVar_analyses_BM36, "/home/dubeu/Synapse/PropVar_analyses_BM36.R")

##BM44
PropVar_analyses_BM44 <- gsub("BM","BM44",as.list(body(PropVar_analyses)))
#Replace first line to create function
PropVar_analyses_BM44[1] <- "PropVar_analyses_BM44 <- function() {"
#Add last line to end function
PropVar_analyses_BM44[length(PropVar_analyses_BM44)+1] <- "}"
write(PropVar_analyses_BM44, "/home/dubeu/Synapse/PropVar_analyses_BM44.R")



source('/home/dubeu/Synapse/PropVar_analyses_BM10.R')
source('/home/dubeu/Synapse/PropVar_analyses_BM22.R')
source('/home/dubeu/Synapse/PropVar_analyses_BM36.R')
source('/home/dubeu/Synapse/PropVar_analyses_BM44.R')
```

```
PropVar_analyses_BM10()
PropVar_analyses_BM22()
PropVar_analyses_BM36()
PropVar_analyses_BM44()




##########################################
##########################################
## Proportion of Variance Explained parietal ##
##########################################
##########################################

##circHOMER1 correlation

parietal_circHOMER1_corr_otherTop <- t(norm_parietal_CDR_counts[sigList_parietal_CDR_top,])
cor(parietal_circHOMER1_corr_otherTop)
summary(cor(parietal_circHOMER1_corr_otherTop)[1,2:ncol(parietal_circHOMER1_corr_otherTop)])

##Estimated proportion of variance for CDR

PropVar_parietal_CDR_Neuron_anova <- data.frame()
PropVar_parietal_CDR_Neuron_relaimpo <- data.frame()
Gene_archive_parietal_CDR <- data.frame(row.names =
rownames(t(norm_parietal_CDR_counts_regress)))
# this will get the intersection of the row.names for everything in the list
sigList_parietal_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_parietal_CDR = sigList_parietal_CDR[order(sigList_parietal_CDR$i_adjpval),]
sigList_parietal_CDR = rownames(sigList_parietal_CDR)
sigList_parietal_CDR_top <- head(sigList_parietal_CDR, n = 10)


for (i in sigList_parietal_CDR_top) {

        print(i)

        print(match(i,sigList_parietal_CDR_top)/length(sigList_parietal_CDR_top))

        Gene_parietal_CDR <- t(norm_parietal_CDR_counts_regress[i,])
  regress_parietal_CDR <- cbind(coldata_parietal[rownames(Gene_parietal_CDR),], Gene =
t(norm_parietal_CDR_counts_regress[i,]))
```

```
        Gene_archive_parietal_CDR <- cbind(Gene_archive_parietal_CDR,
as.data.frame(t(norm_parietal_CDR_counts_regress[i,])))

   model_regress_parietal_CDR = lm(CDR_e ~ APOE4_final + Neuron + regress_parietal_CDR[,i],
data=regress_parietal_CDR )

        coefs <- data.frame(coef(summary(model_regress_parietal_CDR)))

        af <- anova(model_regress_parietal_CDR)
        afss <- af$"Sum Sq"
        anova_parietal <- data.frame()
        anova_parietal <- (cbind(af,PctExp=afss/sum(afss)*100))
        anova_parietal <- anova_parietal[(nrow(anova_parietal)-1),ncol(anova_parietal)]
        coefs <- cbind(coefs, PropVar = anova_parietal)

        PropVar_parietal_CDR_Neuron_anova <- bind_rows(PropVar_parietal_CDR_Neuron_anova, i =
coefs[(nrow(coefs)-1),])
   PropVar_parietal_CDR_Neuron_relaimpo <- bind_rows(PropVar_parietal_CDR_Neuron_relaimpo,
relaimpo = ((calc.relimp(model_regress_parietal_CDR, type = c("lmg"), rela = FALSE))$lmg[3])*100)
}

rownames(PropVar_parietal_CDR_Neuron_anova) <- sigList_parietal_CDR_top
rownames(PropVar_parietal_CDR_Neuron_relaimpo) <- sigList_parietal_CDR_top

regress_parietal_CDR_e <- cbind(coldata_parietal[rownames(Gene_parietal_CDR),],
Gene_archive_parietal_CDR)

formula_parietal_CDR = paste("CDR_e ~ APOE4_final + Neuron +", paste(paste0('`',
sigList_parietal_CDR_top, '`'),  collapse =" + "))

fullmodel_parietal_CDR <- lm(formula_parietal_CDR, data = regress_parietal_CDR_e)

af <- anova(fullmodel_parietal_CDR)
afss <- af$"Sum Sq"
PropVar_fullmodel_parietal_CDR_Neuron_anova <- (cbind(af,PctExp=afss/sum(afss)*100))
PropVar_fullmodel_parietal_CDR_Neuron_relaimpo <-
as.data.frame(calc.relimp(fullmodel_parietal_CDR, type = c("lmg"), rela = FALSE)$lmg)
colnames(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo) <- "relaimpo_lmg"
PropVar_fullmodel_parietal_CDR_Neuron_relaimpo$relaimpo_lmg <-
PropVar_fullmodel_parietal_CDR_Neuron_relaimpo$relaimpo_lmg*100



##Estimated proportion of variance for braak
```

```
PropVar_parietal_braak_Neuron_anova <- data.frame()
PropVar_parietal_braak_Neuron_relaimpo <- data.frame()
Gene_archive_parietal_braak <- data.frame(row.names =
rownames(t(norm_parietal_braak_counts_regress)))
# this will get the intersection of the row.names for everything in the list
sigList_parietal_braak = subset(as.data.frame(both_braak_BM44), i_adjpval < 0.05)
sigList_parietal_braak = sigList_parietal_braak[order(sigList_parietal_braak$i_adjpval),]
sigList_parietal_braak = rownames(sigList_parietal_braak)
sigList_parietal_braak_top <- head(sigList_parietal_braak, n = 10)


for (i in sigList_parietal_braak_top) {

        print(i)

        print(match(i,sigList_parietal_braak_top)/length(sigList_parietal_braak_top))

        Gene_parietal_braak <- t(norm_parietal_braak_counts_regress[i,])
  regress_parietal_braak <- cbind(coldata_parietal[rownames(Gene_parietal_braak),], Gene =
t(norm_parietal_braak_counts_regress[i,]))

        Gene_archive_parietal_braak <- cbind(Gene_archive_parietal_braak,
as.data.frame(t(norm_parietal_braak_counts_regress[i,])))

  model_regress_parietal_braak = lm(braak_final ~ APOE4_final + Neuron + regress_parietal_braak[,i],
data=regress_parietal_braak )

        coefs <- data.frame(coef(summary(model_regress_parietal_braak)))

        af <- anova(model_regress_parietal_braak)
        afss <- af$"Sum Sq"
        anova_parietal <- data.frame()
        anova_parietal <- (cbind(af,PctExp=afss/sum(afss)*100))
        anova_parietal <- anova_parietal[(nrow(anova_parietal)-1),ncol(anova_parietal)]
        coefs <- cbind(coefs, PropVar = anova_parietal)

        PropVar_parietal_braak_Neuron_anova <- bind_rows(PropVar_parietal_braak_Neuron_anova, i
= coefs[(nrow(coefs)-1),])
  PropVar_parietal_braak_Neuron_relaimpo <- bind_rows(PropVar_parietal_braak_Neuron_relaimpo,
relaimpo = ((calc.relimp(model_regress_parietal_braak, type = c("lmg"), rela = FALSE))$lmg[3])*100)
}

rownames(PropVar_parietal_braak_Neuron_anova) <- sigList_parietal_braak_top
```

```
rownames(PropVar_parietal_braak_Neuron_relaimpo) <- sigList_parietal_braak_top

regress_parietal_braak_Final <- cbind(coldata_parietal[rownames(Gene_parietal_braak),],
Gene_archive_parietal_braak)

formula_parietal_braak = paste("braak_final ~ APOE4_final + Neuron +", paste(paste0('`',
sigList_parietal_braak_top, '`'),  collapse =" + "))

fullmodel_parietal_braak <- lm(formula_parietal_braak, data = regress_parietal_braak_Final)

af <- anova(fullmodel_parietal_braak)
afss <- af$"Sum Sq"
PropVar_fullmodel_parietal_braak_Neuron_anova <- (cbind(af,PctExp=afss/sum(afss)*100))
PropVar_fullmodel_parietal_braak_Neuron_relaimpo <-
as.data.frame(calc.relimp(fullmodel_parietal_braak, type = c("lmg"), rela = FALSE)$lmg)
colnames(PropVar_fullmodel_parietal_braak_Neuron_relaimpo) <- "relaimpo_lmg"
PropVar_fullmodel_parietal_braak_Neuron_relaimpo$relaimpo_lmg <-
PropVar_fullmodel_parietal_braak_Neuron_relaimpo$relaimpo_lmg*100

####ADAD

##Estimated proportion of variance for braak

PropVar_parietalADAD_braak_Neuron_anova <- data.frame()
PropVar_parietalADAD_braak_Neuron_relaimpo <- data.frame()
Gene_archive_parietalADAD_braak <- data.frame(row.names =
rownames(t(norm_parietalADAD_condition_counts_regress)))
# this will get the intersection of the row.names for everything in the list
#sigList_parietalADAD_braak = subset(as.data.frame(both_braak_BM44), i_adjpval < 0.05)
sigList_parietalADAD_braak =
subset(as.data.frame(res_parietalADAD_conditionBraak_filter_ADADvsLOAD), padj < 0.05)
sigList_parietalADAD_braak = sigList_parietalADAD_braak[order(sigList_parietalADAD_braak$padj),]
sigList_parietalADAD_braak = rownames(sigList_parietalADAD_braak)
sigList_parietalADAD_braak_top <- head(sigList_parietalADAD_braak, n = 10)


for (i in sigList_parietalADAD_braak_top) {

        print(i)

        print(match(i,sigList_parietalADAD_braak_top)/length(sigList_parietalADAD_braak_top))

        Gene_parietalADAD_braak <- t(norm_parietalADAD_condition_counts_regress[i,])
```

```
   regress_parietalADAD_braak <- cbind(coldata_parietalADAD[rownames(Gene_parietalADAD_braak),],
Gene = t(norm_parietalADAD_condition_counts_regress[i,]))

        regress_parietalADAD_braak <- subset(regress_parietalADAD_braak,
as.character(condition)=="ADAD")

        Gene_archive_parietalADAD_braak <- cbind(Gene_archive_parietalADAD_braak,
as.data.frame(t(norm_parietalADAD_condition_counts_regress[i,])))

   model_regress_parietalADAD_braak = lm(braak_final ~ APOE4_final + Neuron +
regress_parietalADAD_braak[,i], data=regress_parietalADAD_braak )

        coefs <- data.frame(coef(summary(model_regress_parietalADAD_braak)))

        af <- anova(model_regress_parietalADAD_braak)
        afss <- af$"Sum Sq"
        anova_parietalADAD <- data.frame()
        anova_parietalADAD <- (cbind(af,PctExp=afss/sum(afss)*100))
        anova_parietalADAD <- anova_parietalADAD[(nrow(anova_parietalADAD)-
1),ncol(anova_parietalADAD)]
        coefs <- cbind(coefs, PropVar = anova_parietalADAD)

        PropVar_parietalADAD_braak_Neuron_anova <-
bind_rows(PropVar_parietalADAD_braak_Neuron_anova, i = coefs[(nrow(coefs)-1),])
   PropVar_parietalADAD_braak_Neuron_relaimpo <-
bind_rows(PropVar_parietalADAD_braak_Neuron_relaimpo, relaimpo =
((calc.relimp(model_regress_parietalADAD_braak, type = c("lmg"), rela = FALSE))$lmg[3])*100)
}

rownames(PropVar_parietalADAD_braak_Neuron_anova) <- sigList_parietalADAD_braak_top
rownames(PropVar_parietalADAD_braak_Neuron_relaimpo) <- sigList_parietalADAD_braak_top

regress_parietalADAD_braak_Final <-
cbind(coldata_parietalADAD[rownames(Gene_parietalADAD_braak),],
Gene_archive_parietalADAD_braak)

regress_parietalADAD_braak_Final <- subset(regress_parietalADAD_braak_Final,
as.character(condition)=="ADAD")

formula_parietalADAD_braak = paste("braak_final ~ APOE4_final + Neuron +", paste(paste0('`',
sigList_parietalADAD_braak_top, '`'),  collapse =" + "))

fullmodel_parietalADAD_braak <- lm(formula_parietalADAD_braak, data =
regress_parietalADAD_braak_Final)
```

```
af <- anova(fullmodel_parietalADAD_braak)
afss <- af$"Sum Sq"
PropVar_fullmodel_parietalADAD_braak_Neuron_anova <- (cbind(af,PctExp=afss/sum(afss)*100))
PropVar_fullmodel_parietalADAD_braak_Neuron_relaimpo <-
as.data.frame(calc.relimp(fullmodel_parietalADAD_braak, type = c("lmg"), rela = FALSE)$lmg)
colnames(PropVar_fullmodel_parietalADAD_braak_Neuron_relaimpo) <- "relaimpo_lmg"
PropVar_fullmodel_parietalADAD_braak_Neuron_relaimpo$relaimpo_lmg <-
PropVar_fullmodel_parietalADAD_braak_Neuron_relaimpo$relaimpo_lmg*100


##########################################
##########################################
## Proportion of Variance Explained Numbers ##
##########################################
##########################################


summary(PropVar_parietal_CDR_Neuron_relaimpo)

sum(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_parietal_CDR_Neu
ron_relaimpo),])
summary(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_parietal_CDR
_Neuron_relaimpo),])

summary(PropVar_BM44_CDR_Neuron_relaimpo)

sum(PropVar_fullmodel_BM44_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_BM44_CDR_Neuron
_relaimpo),])
summary(PropVar_fullmodel_BM44_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_BM44_CDR_Ne
uron_relaimpo),])


summary(PropVar_parietalADAD_braak_Neuron_relaimpo)

sum(PropVar_fullmodel_parietalADAD_braak_Neuron_relaimpo[3:nrow(PropVar_fullmodel_parietalAD
AD_braak_Neuron_relaimpo),])
summary(PropVar_fullmodel_parietalADAD_braak_Neuron_relaimpo[3:nrow(PropVar_fullmodel_pariet
alADAD_braak_Neuron_relaimpo),])

```
```

ROCC and AUC Analyses

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
```
```

```
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Predictive Utility Analyses
```{r}


##########################################
##########################################
##### AUC and ROC curves #################
##########################################
##########################################

#https://gist.github.com/jwaage/6d8f4eb096e4f18a0894ca1ce27af834
ggroc <- function(roc, showAUC = TRUE, interval = 0.2, breaks = seq(0, 1, interval), TITLE = "TITLE"){
 require(pROC)
 if(class(roc) != "roc")
  simpleError("Please provide roc object from pROC package")
 plotx <- rev(roc$specificities)
 ploty <- rev(roc$sensitivities)

 ggplot(NULL, aes(x = plotx, y = ploty)) +
  geom_segment(aes(x = 0, y = 1, xend = 1,yend = 0), alpha = 0.5) +
  geom_step() +
  scale_x_reverse(name = "Specificity",limits = c(1,0), breaks = breaks, expand = c(0.001,0.001)) +
  scale_y_continuous(name = "Sensitivity", limits = c(0,1), breaks = breaks, expand = c(0.001, 0.001)) +
  theme_bw() +
  theme(axis.ticks = element_line(color = "grey80")) +
  coord_equal() +
  annotate("text", x = 0.7, y = 0.6, vjust = 0, label = paste("AUC =",sprintf("%.3f",roc$auc))) +
        ggtitle(TITLE)

}

library(ggplot2)
library(dplyr)
#https://gist.github.com/Teebusch/db0ab76d31fd31a13ccf93afa7d77df5
#https://stackoverflow.com/questions/21887088/generate-a-filled-geom-step
ggroc_multi <- function(roc1, roc2, roc3, showAUC = TRUE, interval = 0.2, breaks = seq(0, 1, interval),
TITLE = "TITLE"){
        require(pROC)
        plotx1 <- rev(roc1$specificities)
```
```

```
        ploty1<- rev(roc1$sensitivities)
        plotx2 <- rev(roc2$specificities)
        ploty2<- rev(roc2$sensitivities)
        plotx3 <- rev(roc3$specificities)
        ploty3<- rev(roc3$sensitivities)

        plotx <- c(plotx1, plotx2, plotx3)
        ploty <- c(ploty1, ploty2, ploty3)
        label <- c(rep("Base",length(ploty1)),rep("Circ",length(ploty2)),rep("Circ+Base",length(ploty3)))

        df1 = as.data.frame(cbind(plotx1, ploty1))
        df_areaStep1 <- bind_rows(old = df1,
                new = df1 %>% mutate(y = lag(ploty1)), .id = "Base") %>% arrange(plotx1, Base)
        df_areaStep1$label="Base"
        colnames(df_areaStep1) <- c("Base","plotx","ploty","y","label")

        df2 = as.data.frame(cbind(plotx2, ploty2))
        df_areaStep2 <- bind_rows(old = df2,
                new = df2 %>% mutate(y = lag(ploty2)), .id = "Circ") %>% arrange(plotx2, Circ)
        df_areaStep2$label="Circ"
        colnames(df_areaStep2) <- c("Base","plotx","ploty","y","label")

        df3 = as.data.frame(cbind(plotx3, ploty3))
        df_areaStep3 <- bind_rows(old = df3,
                new = df3 %>% mutate(y = lag(ploty3)), .id = "Circ+Base") %>% arrange(plotx3,
`Circ+Base`)
        df_areaStep3$label="Circ+Base"
        colnames(df_areaStep3) <- c("Base","plotx","ploty","y","label")

        df_areaStepFinal <- bind_rows(df_areaStep1, df_areaStep2, df_areaStep3)
        df_areaStepFinal$label <- as.factor(df_areaStepFinal$label)

 ggplot(NULL, aes(x = plotx, y = ploty, color=label)) +
  geom_step(size=1.5) +
        geom_ribbon(aes(x = plotx, ymin = 0, ymax = ploty, fill=label),alpha=0.2, color=NA, data =
df_areaStep3) +
        geom_ribbon(aes(x = plotx, ymin = 0, ymax = ploty, fill=label),alpha=0.2, color=NA, data =
df_areaStep2) +
        geom_ribbon(aes(x = plotx, ymin = 0, ymax = ploty, fill=label),alpha=0.2, color=NA, data =
df_areaStep1) +
  scale_x_reverse(name = "Specificity",limits = c(1,0), breaks = breaks, expand = c(0.001,0.001)) +
  scale_y_continuous(name = "Sensitivity", limits = c(0,1), breaks = breaks, expand = c(0.001, 0.001)) +
  theme_bw() +
  theme(axis.ticks = element_line(color = "grey80")) +
```

```
   coord_equal() +          ggtitle(TITLE) + scale_color_manual(values=c("darkblue", "darkred",
"darkmagenta"),name="Model") + scale_fill_manual(values=c("darkblue", "darkred",
"darkmagenta"),name="Model") + theme(plot.title = element_text(hjust = 0.5),
legend.text=element_text(size=12), legend.title=element_text(size=12))

  #ggplot(NULL, aes(x = plotx, y = ploty, color=label)) +
   #geom_step(size=1.5) +
          #geom_ribbon(aes(x = plotx, ymin = 0, ymax = ploty, fill=label),alpha=0.2, color=NA, data =
df_areaStepFinal) +
   #scale_x_reverse(name = "Specificity",limits = c(1,0), breaks = breaks, expand = c(0.001,0.001)) +
   #scale_y_continuous(name = "Sensitivity", limits = c(0,1), breaks = breaks, expand = c(0.001, 0.001)) +
   #theme_bw() +
   #theme(axis.ticks = element_line(color = "grey80")) +
   #coord_equal() + ggtitle(TITLE)

}


MSBB_ROC_AUC <- function() {

##Base model

coldata_BM_normcounts <<- as.data.frame(cbind(colData(dds_BM_final_condition_filter),
t(norm_BM_condition_counts)))
coldata_BM_normcounts <<- subset(coldata_BM_normcounts, as.numeric(NP.1) <= 2)
coldata_BM_normcounts$NP.1 <<- droplevels(coldata_BM_normcounts$NP.1)


mylogit_base_BM <<- glm(NP.1 ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 + ApoE4_SNP,
data = coldata_BM_normcounts, family = "binomial")
summary(mylogit_base_BM)
prob_base_BM=predict(mylogit_base_BM,type=c("response"))
coldata_BM_normcounts$prob_base_BM=prob_base_BM
library(pROC)
g_base_BM <<- roc(NP.1 ~ prob_base_BM, data = coldata_BM_normcounts)
ggroc(g_base_BM, TITLE="BM Base")

##Circ Model

sigList_BM_CDR = subset(as.data.frame(both_CDR_BM), i_adjpval < 0.05)
sigList_BM_CDR = sigList_BM_CDR[order(sigList_BM_CDR$i_adjpval),]
sigList_BM_CDR = rownames(sigList_BM_CDR)
sigList_BM_CDR_top <<- head(sigList_BM_CDR, n = 10)
```

```
mylogit_circ_BM <<- glm(paste("NP.1 ~", paste(paste0('`', gsub("-",".",sigList_BM_CDR_top), '`'),
collapse =" + ")), data = coldata_BM_normcounts, family = "binomial")
summary(mylogit_circ_BM)
prob_circ_BM=predict(mylogit_circ_BM,type=c("response"))
coldata_BM_normcounts$prob_circ_BM=prob_circ_BM
library(pROC)
g_circ_BM <<- roc(NP.1 ~ prob_circ_BM, data = coldata_BM_normcounts)
ggroc(g_circ_BM, TITLE="BM Circ")

##Base + Circ Model

sigList_BM_CDR = subset(as.data.frame(both_CDR_BM), i_adjpval < 0.05)
sigList_BM_CDR = sigList_BM_CDR[order(sigList_BM_CDR$i_adjpval),]
sigList_BM_CDR = rownames(sigList_BM_CDR)
sigList_BM_CDR_top <<- head(sigList_BM_CDR, n = 10)

mylogit_base_circ_BM <<- glm(paste("NP.1 ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 +
ApoE4_SNP +", paste(paste0('`', gsub("-",".",sigList_BM_CDR_top), '`'),  collapse =" + ")), data =
coldata_BM_normcounts, family = "binomial")
summary(mylogit_base_circ_BM)
prob_base_circ_BM=predict(mylogit_base_circ_BM,type=c("response"))
coldata_BM_normcounts$prob_base_circ_BM=prob_base_circ_BM
library(pROC)
g_base_circ_BM <<- roc(NP.1 ~ prob_base_circ_BM, data = coldata_BM_normcounts)



##Return all plots
list(ggroc(g_base_BM, TITLE="BM Base"), ggroc(g_circ_BM, TITLE="BM CIRC"), ggroc(g_base_circ_BM,
TITLE="BM Base+CIRC"), ggroc_multi(g_base_BM, g_circ_BM,g_base_circ_BM, TITLE="Relative
Predictive Ability - AMP-AD: MSBB BM"))

}


##BM10
MSBB_ROC_AUC_BM10 <- gsub("BM","BM10",as.list(body(MSBB_ROC_AUC)))
#Replace first line to create function
MSBB_ROC_AUC_BM10[1] <- "MSBB_ROC_AUC_BM10 <- function() {"
#Replace first line to end function
MSBB_ROC_AUC_BM10[length(MSBB_ROC_AUC_BM10)+1] <- "}"
write(MSBB_ROC_AUC_BM10, "/home/dubeu/Synapse/MSBB_ROC_AUC_BM10.R")

##BM22
```

```
MSBB_ROC_AUC_BM22 <- gsub("BM","BM22",as.list(body(MSBB_ROC_AUC)))
#Replace first line to create function
MSBB_ROC_AUC_BM22[1] <- "MSBB_ROC_AUC_BM22 <- function() {"
#Replace first line to end function
MSBB_ROC_AUC_BM22[length(MSBB_ROC_AUC_BM22)+1] <- "}"
write(MSBB_ROC_AUC_BM22, "/home/dubeu/Synapse/MSBB_ROC_AUC_BM22.R")

##BM36
MSBB_ROC_AUC_BM36 <- gsub("BM","BM36",as.list(body(MSBB_ROC_AUC)))
#Replace first line to create function
MSBB_ROC_AUC_BM36[1] <- "MSBB_ROC_AUC_BM36 <- function() {"
#Replace first line to end function
MSBB_ROC_AUC_BM36[length(MSBB_ROC_AUC_BM36)+1] <- "}"
write(MSBB_ROC_AUC_BM36, "/home/dubeu/Synapse/MSBB_ROC_AUC_BM36.R")

##BM44
MSBB_ROC_AUC_BM44 <- gsub("BM","BM44",as.list(body(MSBB_ROC_AUC)))
#Replace first line to create function
MSBB_ROC_AUC_BM44[1] <- "MSBB_ROC_AUC_BM44 <- function() {"
#Replace first line to end function
MSBB_ROC_AUC_BM44[length(MSBB_ROC_AUC_BM44)+1] <- "}"
write(MSBB_ROC_AUC_BM44, "/home/dubeu/Synapse/MSBB_ROC_AUC_BM44.R")



source('/home/dubeu/Synapse/MSBB_ROC_AUC_BM10.R')
source('/home/dubeu/Synapse/MSBB_ROC_AUC_BM22.R')
source('/home/dubeu/Synapse/MSBB_ROC_AUC_BM36.R')
source('/home/dubeu/Synapse/MSBB_ROC_AUC_BM44.R')

MSBB_ROC_AUC_BM10()
MSBB_ROC_AUC_BM22()
MSBB_ROC_AUC_BM36()
MSBB_ROC_AUC_BM44()


##Parietal


coldata_parietal_normcounts <- as.data.frame(cbind(colData(dds_parietal_final_condition_filter),
t(norm_parietal_condition_counts)))
coldata_parietal_normcounts <- coldata_parietal_normcounts[which(
as.character(coldata_parietal_normcounts$condition) == "CO" |
as.character(coldata_parietal_normcounts$condition) == "LOAD"),]
```

```
coldata_parietal_normcounts$condition <- droplevels(coldata_parietal_normcounts$condition)

#Base

mylogit_base_parietal <- glm(condition ~ PMI + TIN.median. + AGE_at_death + pool + GENDER + PC1 +
PC2 + APOE4_final, data = coldata_parietal_normcounts, family = "binomial")
summary(mylogit_base_parietal)
prob_base_parietal=predict(mylogit_base_parietal,type=c("response"))
coldata_parietal_normcounts$prob_base_parietal=prob_base_parietal
library(pROC)
g_base_parietal <- roc(condition ~ prob_base_parietal, data = coldata_parietal_normcounts)
ggroc(g_base_parietal, TITLE="Parietal Base")


sigList_BM44_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_BM44_CDR = sigList_BM44_CDR[order(sigList_BM44_CDR$i_adjpval),]
sigList_BM44_CDR = rownames(sigList_BM44_CDR)
sigList_BM44_CDR_top <- head(sigList_BM44_CDR, n = 10)

#Circ

mylogit_circ_parietal <- glm(paste("condition ~", paste(paste0('`', gsub("-",".",sigList_BM44_CDR_top),
'`'),  collapse =" + ")), data = coldata_parietal_normcounts, family = "binomial")
summary(mylogit_circ_parietal)
prob_circ_parietal=predict(mylogit_circ_parietal,type=c("response"))
coldata_parietal_normcounts$prob_circ_parietal=prob_circ_parietal
library(pROC)
g_circ_parietal <- roc(condition ~ prob_circ_parietal, data = coldata_parietal_normcounts)
ggroc(g_circ_parietal, TITLE="Parietal Circ")


sigList_BM44_CDR = subset(as.data.frame(both_CDR_BM44), i_adjpval < 0.05)
sigList_BM44_CDR = sigList_BM44_CDR[order(sigList_BM44_CDR$i_adjpval),]
sigList_BM44_CDR = rownames(sigList_BM44_CDR)
sigList_BM44_CDR_top <- head(sigList_BM44_CDR, n = 10)

#Base + Circ

mylogit_base_circ_parietal <- glm(paste("condition ~ PMI + TIN.median. + AGE_at_death + pool +
GENDER + PC1 + PC2 + APOE4_final +", paste(paste0('`', gsub("-",".",sigList_BM44_CDR_top), '`'),
collapse =" + ")), data = coldata_parietal_normcounts, family = "binomial")
summary(mylogit_base_circ_parietal)
prob_base_circ_parietal=predict(mylogit_base_circ_parietal,type=c("response"))
coldata_parietal_normcounts$prob_base_circ_parietal=prob_base_circ_parietal
```

```
library(pROC)
g_base_circ_parietal <- roc(condition ~ prob_base_circ_parietal, data = coldata_parietal_normcounts)
ggroc(g_base_circ_parietal, TITLE="Parietal Base+Circ")

#Final plot
ggroc_multi(g_base_parietal, g_circ_parietal,g_base_circ_parietal, TITLE="Relative Predictive Ability - Knight-ADRC: Parietal")


```

Co-expression Network Analyses

Processing Code

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Co-Expression Network Analyses
```{r}



##############################################
##############################################
### MEGENA Generation Code - Parietal ####
##############################################
##############################################


##load Parietal phenotype / covariate information
coldata_parietal_salmon <- read.csv(file="/40/AD/Expression/KnightADRCParietal/03.-phenotype/Pheno.csv", header=TRUE, na.strings = "NA")

#Restrict analyses to parietal tissues
coldata_parietal_salmon <- subset(coldata_parietal_salmon, brain_region == "parietal")

#Rename
coldata_parietal_salmon$ID_RNAseq <- gsub("VY.", "VY-", coldata_parietal_salmon$ID_RNAseq)

#Temporarily Assigning Ethnicity#
coldata_parietal_salmon$Ethnicity <- coldata_parietal_salmon$Ethnicity_temp
```
```

```
##Remove Consensus Outliers
coldata_parietal_salmon <- subset(coldata_parietal_salmon, ConsensusOutlier_RemoveBefore != 1)

#Set up factor levels
coldata_parietal_salmon$pool <- factor(coldata_parietal_salmon$pool, levels=c('1','2'))
coldata_parietal_salmon$GENDER <- factor(coldata_parietal_salmon$GENDER, levels=c('1','2'))
coldata_parietal_salmon$Ethnicity <- factor(coldata_parietal_salmon$Ethnicity, levels = c("EA", "AA"))
coldata_parietal_salmon$condition <- droplevels(coldata_parietal_salmon$condition)



############################
############################
##
############################
############################



#Quasi
dir <- "/home/dubeu/Synapse/Salmon/quasi"

#fmd
#dir <- "/home/dubeu/Synapse/Salmon/fmd/"


#Only Protein Coding
#tx2gene <- read.table("/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26_transcriptID_to_geneID_final_proteincoding")

#All
#tx2gene <- read.table("/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26_transcriptID_to_geneID_final")

#Quasi
tx2gene <- read.table("/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26.metadata.HGNC")


files <- file.path(dir, coldata_parietal_salmon$ID_RNAseq, "quant.sf")
names(files) <- coldata_parietal_salmon$ID_RNAseq

txi <- tximport(files, type="salmon", tx2gene=tx2gene, dropInfReps=TRUE)
```

```
#coldata_parietal_salmon$ID_RNAseq <- gsub("VY-", "VY.", coldata_parietal_salmon$ID_RNAseq)
rownames(coldata_parietal_salmon) <- coldata_parietal_salmon$ID_RNAseq


####################
####################
## DE Analyses ##
####################
####################


##DE based on condition ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietal_salmon_condition_filter <- DESeqDataSetFromTximport(txi, colData =
coldata_parietal_salmon, design = ~ 1)

dds_parietal_salmon_condition_filter <- estimateSizeFactors(dds_parietal_salmon_condition_filter)

##subset based on covars so that none are missing

#Remove dementedcontrols
dds_parietal_salmon_condition_filter <- dds_parietal_salmon_condition_filter[,
!(dds_parietal_salmon_condition_filter$DementedControl == 1)]

dds_parietal_salmon_condition_filter <- dds_parietal_salmon_condition_filter[
,!is.na(dds_parietal_salmon_condition_filter$AGE_at_death)]
dds_parietal_salmon_condition_filter <- dds_parietal_salmon_condition_filter[
,!is.na(dds_parietal_salmon_condition_filter$PMI)]
dds_parietal_salmon_condition_filter <- dds_parietal_salmon_condition_filter[
,!is.na(dds_parietal_salmon_condition_filter$PC1)]
dds_parietal_salmon_condition_filter <- dds_parietal_salmon_condition_filter[
,!is.na(dds_parietal_salmon_condition_filter$PC2)]

design(dds_parietal_salmon_condition_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool +
GENDER + PC1 + PC2 + condition)
```

```
dds_parietal_salmon_final_condition_filter <- DESeq(dds_parietal_salmon_condition_filter, parallel =
TRUE, betaPrior = FALSE)

##Analyze results based on contrast

res_parietal_salmon_condition_filter <- results(dds_parietal_salmon_final_condition_filter,
contrast=c("condition", "LOAD", "CO"), alpha = 0.05)
summary(res_parietal_salmon_condition_filter)




##DE based on braak ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietal_salmon_braak_filter <- DESeqDataSetFromTximport(txi, colData =
coldata_parietal_salmon, design = ~ 1)

dds_parietal_salmon_braak_filter <- estimateSizeFactors(dds_parietal_salmon_braak_filter)

#Remove dementedcontrols
dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[,
!(dds_parietal_salmon_braak_filter$DementedControl == 1)]

dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[
,!is.na(dds_parietal_salmon_braak_filter$AGE_at_death)]

dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[
,!is.na(dds_parietal_salmon_braak_filter$PMI)]

dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[
,!is.na(dds_parietal_salmon_braak_filter$braak_final)]

dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[
,!is.na(dds_parietal_salmon_braak_filter$PC1)]
dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[
,!is.na(dds_parietal_salmon_braak_filter$PC2)]
```

```
##Only include LOAD and controls
dds_parietal_salmon_braak_filter <- dds_parietal_salmon_braak_filter[
,dds_parietal_salmon_braak_filter$condition != "ADAD" ]
dds_parietal_salmon_braak_filter$condition <- droplevels(dds_parietal_salmon_braak_filter$condition)


design(dds_parietal_salmon_braak_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool +
GENDER + PC1 + PC2 + braak_final)


dds_parietal_salmon_final_braak_filter <- DESeq(dds_parietal_salmon_braak_filter, parallel = TRUE,
betaPrior = FALSE)

res_parietal_salmon_braak_filter <- results(dds_parietal_salmon_final_braak_filter, alpha = 0.05)
summary(res_parietal_salmon_braak_filter)



##DE based on CDR ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_parietal_salmon_CDR_filter <- DESeqDataSetFromTximport(txi, colData = coldata_parietal_salmon,
design = ~ 1)

dds_parietal_salmon_CDR_filter <- estimateSizeFactors(dds_parietal_salmon_CDR_filter)


#Remove dementedcontrols
dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[,
!(dds_parietal_salmon_CDR_filter$DementedControl == 1)]


dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[
,!is.na(dds_parietal_salmon_CDR_filter$AGE_at_death)]
dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[
,!is.na(dds_parietal_salmon_CDR_filter$PMI)]
dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[
,!is.na(dds_parietal_salmon_CDR_filter$CDR_e)]
```

```
dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[
,!is.na(dds_parietal_salmon_CDR_filter$PC1)]
dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[
,!is.na(dds_parietal_salmon_CDR_filter$PC2)]

##Only include LOAD and controls
dds_parietal_salmon_CDR_filter <- dds_parietal_salmon_CDR_filter[
,dds_parietal_salmon_CDR_filter$condition != "ADAD" ]
dds_parietal_salmon_CDR_filter$condition <- droplevels(dds_parietal_salmon_CDR_filter$condition)


design(dds_parietal_salmon_CDR_filter) = formula( ~ PMI + TIN.median. + AGE_at_death + pool +
GENDER + PC1 + PC2 + CDR_e)

dds_parietal_salmon_final_CDR_filter <- DESeq(dds_parietal_salmon_CDR_filter, parallel = TRUE,
betaPrior = FALSE)

##subset based on contrast (categorical) or continuous


res_parietal_salmon_CDR_filter <- results(dds_parietal_salmon_final_CDR_filter, alpha = 0.05)
summary(res_parietal_salmon_CDR_filter)



##For Condition

#vsd_parietal_salmon_condition_linear <-
varianceStabilizingTransformation(dds_parietal_salmon_condition_filter, blind = FALSE)
#norm_parietal_salmon_condition_counts <-
as.data.frame(assay(vsd_parietal_salmon_condition_linear))

##For CDR

vsd_parietal_salmon_CDR_linear <- varianceStabilizingTransformation(dds_parietal_salmon_CDR_filter,
blind = FALSE)
norm_parietal_salmon_CDR_counts <- as.data.frame(assay(vsd_parietal_salmon_CDR_linear))

##For braak

#vsd_parietal_salmon_braak_linear <-
varianceStabilizingTransformation(dds_parietal_salmon_braak_filter, blind = FALSE)
#norm_parietal_salmon_braak_counts <- as.data.frame(assay(vsd_parietal_salmon_braak_linear))
```

```
##########################################
##########################################
######## Regressing Out Cofactors ########
##########################################
##########################################

###Regress out cofactors for parietal_salmon dataset

##CDR
mmage_parietal_salmon_CDR=model.matrix(~-
1+dds_parietal_salmon_CDR_filter$TIN.median.+dds_parietal_salmon_CDR_filter$AGE_at_death+dds_
parietal_salmon_CDR_filter$GENDER+dds_parietal_salmon_CDR_filter$PMI+dds_parietal_salmon_CDR
_filter$PC1+dds_parietal_salmon_CDR_filter$PC2+dds_parietal_salmon_CDR_filter$pool)
norm_parietal_salmon_CDR_counts_regress=as.data.frame(t(apply(t(norm_parietal_salmon_CDR_coun
ts),2,function(x){residuals(lm(x~mmage_parietal_salmon_CDR))})))


##braak
mmage_parietal_salmon_braak=model.matrix(~-
1+dds_parietal_salmon_braak_filter$TIN.median.+dds_parietal_salmon_braak_filter$AGE_at_death+dd
s_parietal_salmon_braak_filter$GENDER+dds_parietal_salmon_braak_filter$PMI+dds_parietal_salmon_
braak_filter$PC1+dds_parietal_salmon_braak_filter$PC2+dds_parietal_salmon_braak_filter$pool)
norm_parietal_salmon_braak_counts_regress=as.data.frame(t(apply(t(norm_parietal_salmon_braak_co
unts),2,function(x){residuals(lm(x~mmage_parietal_salmon_braak))})))

##condition
mmage_parietal_salmon_condition=model.matrix(~-
1+dds_parietal_salmon_condition_filter$TIN.median.+dds_parietal_salmon_condition_filter$AGE_at_d
eath+dds_parietal_salmon_condition_filter$GENDER+dds_parietal_salmon_condition_filter$PMI+dds_p
arietal_salmon_condition_filter$PC1+dds_parietal_salmon_condition_filter$PC2+dds_parietal_salmon_
condition_filter$pool)
norm_parietal_salmon_condition_counts_regress=as.data.frame(t(apply(t(norm_parietal_salmon_condi
tion_counts),2,function(x){residuals(lm(x~mmage_parietal_salmon_condition))})))



colnames(norm_parietal_salmon_CDR_counts_regress) <- gsub("VY-", "VY.",
colnames(norm_parietal_salmon_CDR_counts_regress))

colnames(norm_parietal_salmon_braak_counts_regress) <- gsub("VY-", "VY.",
colnames(norm_parietal_salmon_braak_counts_regress))
```

```
colnames(norm_parietal_salmon_condition_counts_regress) <- gsub("VY-", "VY.",
colnames(norm_parietal_salmon_condition_counts_regress))


system("mkdir -p /home/dubeu/Synapse/NetworkAnalysis/parietal")

write.table(norm_parietal_salmon_CDR_counts_regress,
"/home/dubeu/Synapse/NetworkAnalysis/parietal/norm_parietal_salmon_CDR_counts_regress.txt",
sep = "\t", quote=F)

write.table(norm_parietal_CDR_counts_regress,
"/home/dubeu/Synapse/NetworkAnalysis/parietal/norm_parietal_CDR_counts_regress.txt", sep = "\t",
quote=F)

coldata_parietal_MEs <- as.data.frame(colData(dds_parietal_CDR_filter))
save(coldata_parietal_MEs,file="/home/dubeu/Synapse/NetworkAnalysis/parietal/coldata_parietal.Rda
")

###
#Only include protein coding genes and perform correlation with spearman

#make directory structure on CHPC
system('ssh dubeu@login.chpc.wustl.edu "mkdir -p
/scratch/dubeu/Synapse/NetworkAnalysis/parietal/top10kspearman"')

#prepare batch script for 10k also modify to extend the walltime
system("cat /home/dubeu/Synapse/NetworkAnalysis/Dev/top10kspearman.batch | sed
's/AREA/parietal/g' | sed 's/walltime=24:00:00/walltime=36:00:00/g' >
/home/dubeu/Synapse/NetworkAnalysis/parietal/parietal_top10kspearman.batch")

#prepare R script for 10k
system("cat /home/dubeu/Synapse/NetworkAnalysis/Dev/top10kspearman.R | sed 's/AREA/parietal/g'
> /home/dubeu/Synapse/NetworkAnalysis/parietal/parietal_top10kspearman.R")

#Upload files

#counts
system("scp -r
/home/dubeu/Synapse/NetworkAnalysis/parietal/norm_parietal_CDR_counts_regress.txt
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/parietal")
system("scp -r
/home/dubeu/Synapse/NetworkAnalysis/parietal/norm_parietal_salmon_CDR_counts_regress.txt
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/parietal")
```

```
#top10k
system("scp -r /home/dubeu/Synapse/NetworkAnalysis/parietal/parietal_top10kspearman.R
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/parietal/top10kspearman")
system("scp -r /home/dubeu/Synapse/NetworkAnalysis/parietal/parietal_top10kspearman.batch
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/parietal/top10kspearman")

#protein coding list
system("scp -r /40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26_transcriptID_to_geneID_final_proteincoding_uniqlist
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/")

##Send qsub command top10k
system('ssh dubeu@login.chpc.wustl.edu "qsub
/scratch/dubeu/Synapse/NetworkAnalysis/parietal/top10kspearman/parietal_top10kspearman.batch"')




##########################################
##########################################
### MEGENA Generation Code - Including salmon processing  ####
##########################################
##########################################

MEGENA_complete <- function() {

##Linear Transcriptome differential expression ##

##Load in phenotype data ##
coldata_salmon_BM <-
read.csv(file="/home/dubeu/Synapse/AMP_AD/MSBB_Pheno_05_15_2018.csv", header=TRUE,
na.strings = "NA")
coldata_salmon_BM <- subset(coldata_salmon_BM, BrodmannArea=="BM")

##Set factor levels

coldata_salmon_BM$NP.1 <- factor(coldata_salmon_BM$NP.1, levels=c('1','2','3','4'))
coldata_salmon_BM$Apo1 <- factor(coldata_salmon_BM$Apo1, levels=c('0','2','3','4'))
coldata_salmon_BM$Apo2 <- factor(coldata_salmon_BM$Apo2, levels=c('0','2','3','4'))
coldata_salmon_BM$RACE <- factor(coldata_salmon_BM$RACE, levels=c('A','B','H','W','U'))
coldata_salmon_BM$RACE <- relevel(coldata_salmon_BM$RACE, "W")
coldata_salmon_BM$SEX <- factor(coldata_salmon_BM$SEX, levels = c("M","F"))
```

```
coldata_salmon_BM$batch <<- factor(coldata_salmon_BM$batch, levels =
levels(coldata_salmon_BM$batch))

rownames(coldata_salmon_BM) <<- coldata_salmon_BM$sampleIdentifier

##Relevel NP.1
#levels(coldata_BM$NP.1) <- list("1"=c("1"), "2"=c("2","3","4"))

coldata_salmon_BM[(as.character(coldata_salmon_BM$NP.1)=="1" &
coldata_salmon_BM$CDR>=1),]$DementedControl <<- 1

########
##########Salmon Processing#########
#######

#Quasi
dir <<- "/home/dubeu/Synapse/AMP_AD/Salmon/quasi/BM"

#fmd
#dir <<- "/home/dubeu/Synapse/Salmon/fmd/"


#Only Protein Coding
#tx2gene <<- read.table("/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26_transcriptID_to_geneID_final_proteincoding")

#All
#tx2gene <<- read.table("/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26_transcriptID_to_geneID_final")

#Quasi
tx2gene <<- read.table("/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26.metadata.HGNC")


files <<- file.path(dir, rownames(coldata_salmon_BM), "quant.sf")
names(files) <<- rownames(coldata_salmon_BM)

txi_BM <<- tximport(files, type="salmon", tx2gene=tx2gene, dropInfReps=TRUE)



##DE based on condition ##
```

```
#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

#dds_BM_salmon_condition_filter <<- DESeqDataSetFromTximport(txi_BM, colData =
coldata_salmon_BM, design = ~ 1)

#dds_BM_salmon_condition_filter <<- estimateSizeFactors(dds_BM_salmon_condition_filter)

#dds_BM_salmon_condition_filter <<- dds_BM_salmon_condition_filter[
,!is.na(dds_BM_salmon_condition_filter$batch)]
#dds_BM_salmon_condition_filter <<- dds_BM_salmon_condition_filter[
,!is.na(dds_BM_salmon_condition_filter$PC1)]
#dds_BM_salmon_condition_filter <<- dds_BM_salmon_condition_filter[
,!is.na(dds_BM_salmon_condition_filter$PC2)]


#Remove dementedcontrols
#dds_BM_salmon_condition_filter <<- dds_BM_salmon_condition_filter[,
!(dds_BM_salmon_condition_filter$DementedControl == 1)]


#dds_BM_salmon_condition_filter$batch <<- droplevels(dds_BM_salmon_condition_filter$batch)


#design(dds_BM_salmon_condition_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 +
PC2 + NP.1)


#dds_BM_salmon_final_condition_filter <<- DESeq(dds_BM_salmon_condition_filter, parallel = TRUE,
betaPrior = FALSE)

#res_BM_salmon_condition_filter <<- results(dds_BM_salmon_final_condition_filter, contrast=c("NP.1",
"2", "1"), alpha = 0.05)
#summary(res_BM_salmon_condition_filter)


##DE based on CDR ##

#############
## Filtering QC ##
```

```
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############

dds_BM_salmon_CDR_filter <<- DESeqDataSetFromTximport(txi_BM, colData = coldata_salmon_BM,
design = ~ 1)

dds_BM_salmon_CDR_filter <<- estimateSizeFactors(dds_BM_salmon_CDR_filter)

dds_BM_salmon_CDR_filter <<- dds_BM_salmon_CDR_filter[
,!is.na(dds_BM_salmon_CDR_filter$batch)]
dds_BM_salmon_CDR_filter <<- dds_BM_salmon_CDR_filter[ ,!is.na(dds_BM_salmon_CDR_filter$PC1)]
dds_BM_salmon_CDR_filter <<- dds_BM_salmon_CDR_filter[ ,!is.na(dds_BM_salmon_CDR_filter$PC2)]

dds_BM_salmon_CDR_filter <<- dds_BM_salmon_CDR_filter[ ,!is.na(dds_BM_salmon_CDR_filter$CDR)]


#Remove dementedcontrols
dds_BM_salmon_CDR_filter <<- dds_BM_salmon_CDR_filter[,
!(dds_BM_salmon_CDR_filter$DementedControl == 1)]


dds_BM_salmon_CDR_filter$batch <<- droplevels(dds_BM_salmon_CDR_filter$batch)


design(dds_BM_salmon_CDR_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2 +
CDR)


dds_BM_salmon_final_CDR_filter <<- DESeq(dds_BM_salmon_CDR_filter, parallel = TRUE, betaPrior =
FALSE)

res_BM_salmon_CDR_filter <<- results(dds_BM_salmon_final_CDR_filter, alpha = 0.05)
summary(res_BM_salmon_CDR_filter)



##DE based on braak ##

#############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
#############
```

```
#dds_BM_salmon_braak_filter <<- DESeqDataSetFromTximport(txi_BM, colData = coldata_salmon_BM,
design = ~ 1)

#dds_BM_salmon_braak_filter <<- estimateSizeFactors(dds_BM_salmon_braak_filter)

#dds_BM_salmon_braak_filter <<- dds_BM_salmon_braak_filter[
,!is.na(dds_BM_salmon_braak_filter$batch)]
#dds_BM_salmon_braak_filter <<- dds_BM_salmon_braak_filter[
,!is.na(dds_BM_salmon_braak_filter$PC1)]
#dds_BM_salmon_braak_filter <<- dds_BM_salmon_braak_filter[
,!is.na(dds_BM_salmon_braak_filter$PC2)]

#dds_BM_salmon_braak_filter <<- dds_BM_salmon_braak_filter[
,!is.na(dds_BM_salmon_braak_filter$bbscore)]


#Remove dementedcontrols
#dds_BM_salmon_braak_filter <<- dds_BM_salmon_braak_filter[,
!(dds_BM_salmon_braak_filter$DementedControl == 1)]

#dds_BM_salmon_braak_filter$batch <<- droplevels(dds_BM_salmon_braak_filter$batch)

#design(dds_BM_salmon_braak_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1 + PC2
+ bbscore)

#dds_BM_salmon_final_braak_filter <<- DESeq(dds_BM_salmon_braak_filter, parallel = TRUE, betaPrior
= FALSE)

#res_BM_salmon_braak_filter <<- results(dds_BM_salmon_final_braak_filter, alpha = 0.05)
#summary(res_BM_salmon_braak_filter)


##DE based on PlaqueMean ##

############
## Filtering QC ##
## via: https://support.bioconductor.org/p/65256/ ##
## triplicate by lowest number: https://support.bioconductor.org/p/49710/ ##
############

#dds_BM_salmon_PlaqueMean_filter <<- DESeqDataSetFromTximport(txi_BM, colData =
coldata_salmon_BM, design = ~ 1)
```

```
#dds_BM_salmon_PlaqueMean_filter <<- estimateSizeFactors(dds_BM_salmon_PlaqueMean_filter)

#dds_BM_salmon_PlaqueMean_filter <<- dds_BM_salmon_PlaqueMean_filter[
,!is.na(dds_BM_salmon_PlaqueMean_filter$batch)]

#dds_BM_salmon_PlaqueMean_filter <<- dds_BM_salmon_PlaqueMean_filter[
,!is.na(dds_BM_salmon_PlaqueMean_filter$PC1)]
#dds_BM_salmon_PlaqueMean_filter <<- dds_BM_salmon_PlaqueMean_filter[
,!is.na(dds_BM_salmon_PlaqueMean_filter$PC2)]

#dds_BM_salmon_PlaqueMean_filter <<- dds_BM_salmon_PlaqueMean_filter[
,!is.na(dds_BM_salmon_PlaqueMean_filter$bbscore)]

#Remove dementedcontrols
#dds_BM_salmon_PlaqueMean_filter <<- dds_BM_salmon_PlaqueMean_filter[,
!(dds_BM_salmon_PlaqueMean_filter$DementedControl == 1)]

#dds_BM_salmon_PlaqueMean_filter$batch <<- droplevels(dds_BM_salmon_PlaqueMean_filter$batch)


#design(dds_BM_salmon_PlaqueMean_filter) = formula( ~ PMI + TIN.median. + AOD + batch + SEX + PC1
+ PC2 + bbscore)

#dds_BM_salmon_final_PlaqueMean_filter <<- DESeq(dds_BM_salmon_PlaqueMean_filter, parallel =
TRUE, betaPrior = FALSE)

#res_BM_salmon_PlaqueMean_filter <<- results(dds_BM_salmon_final_PlaqueMean_filter, alpha =
0.05)
#summary(res_BM_salmon_PlaqueMean_filter)




##########################################
##########################################
## VSD normalization ##
##########################################
##########################################

##For Condition

#vsd_BM_salmon_condition_circ <<-
varianceStabilizingTransformation(dds_BM_salmon_condition_filter, blind = FALSE)
#norm_BM_salmon_condition_counts <<- as.data.frame(assay(vsd_BM_salmon_condition_circ))
```

```
##For CDR

vsd_BM_salmon_CDR_circ <<- varianceStabilizingTransformation(dds_BM_salmon_CDR_filter, blind =
FALSE)
norm_BM_salmon_CDR_counts <<- as.data.frame(assay(vsd_BM_salmon_CDR_circ))


##For braak

#vsd_BM_salmon_braak_circ <<- varianceStabilizingTransformation(dds_BM_salmon_braak_filter, blind
= FALSE)
#norm_BM_salmon_braak_counts <<- as.data.frame(assay(vsd_BM_salmon_braak_circ))


##For PlaqueMean

#vsd_BM_salmon_PlaqueMean_circ <<-
varianceStabilizingTransformation(dds_BM_salmon_PlaqueMean_filter, blind = FALSE)
#norm_BM_salmon_PlaqueMean_counts <<- as.data.frame(assay(vsd_BM_salmon_PlaqueMean_circ))


##########################################
##########################################
## Regressing Out Cofactors ##
##########################################
##########################################


###Regress out cofactors from counts for PlaqueMean

#mmage_BM_salmon_PlaqueMean=model.matrix(~-
1+dds_BM_salmon_PlaqueMean_filter$TIN.median.+dds_BM_salmon_PlaqueMean_filter$AOD+dds_B
M_salmon_PlaqueMean_filter$SEX+dds_BM_salmon_PlaqueMean_filter$PMI+dds_BM_salmon_Plaque
Mean_filter$PC1+dds_BM_salmon_PlaqueMean_filter$PC2+dds_BM_salmon_PlaqueMean_filter$batch
)
#norm_BM_salmon_PlaqueMean_counts_regress=as.data.frame(t(apply(t(norm_BM_salmon_PlaqueM
ean_counts),2,function(x){residuals(lm(x~mmage_BM_salmon_PlaqueMean))})))


###Regress out cofactors from counts for CDR
```

```
mmage_BM_salmon_CDR <<- model.matrix(~-
1+dds_BM_salmon_CDR_filter$TIN.median.+dds_BM_salmon_CDR_filter$AOD+dds_BM_salmon_CDR_f
ilter$SEX+dds_BM_salmon_CDR_filter$PMI+dds_BM_salmon_CDR_filter$PC1+dds_BM_salmon_CDR_fil
ter$PC2+dds_BM_salmon_CDR_filter$batch)
norm_BM_salmon_CDR_counts_regress <-
as.data.frame(t(apply(t(norm_BM_salmon_CDR_counts),2,function(x){residuals(lm(x~mmage_BM_salm
on_CDR))})))


###Regress out cofactors from counts for braak

#mmage_BM_salmon_braak=model.matrix(~-
1+dds_BM_salmon_braak_filter$TIN.median.+dds_BM_salmon_braak_filter$AOD+dds_BM_salmon_bra
ak_filter$SEX+dds_BM_salmon_braak_filter$PMI+dds_BM_salmon_braak_filter$PC1+dds_BM_salmon_
braak_filter$PC2+dds_BM_salmon_braak_filter$batch)
#norm_BM_salmon_braak_counts_regress=as.data.frame(t(apply(t(norm_BM_salmon_braak_counts),2
,function(x){residuals(lm(x~mmage_BM_salmon_braak))})))

##Output files for MEGENA analysis

system("mkdir -p /home/dubeu/Synapse/NetworkAnalysis/BM")

write.table(norm_BM_salmon_CDR_counts_regress,
"/home/dubeu/Synapse/NetworkAnalysis/BM/norm_BM_salmon_CDR_counts_regress.txt", sep = "\t",
quote=F)

write.table(norm_BM_CDR_counts_regress,
"/home/dubeu/Synapse/NetworkAnalysis/BM/norm_BM_CDR_counts_regress.txt", sep = "\t", quote=F)

coldata_BM_MEs <- as.data.frame(colData(dds_BM_CDR_filter))
save(coldata_BM_MEs,file="/home/dubeu/Synapse/NetworkAnalysis/BM/coldata_BM.Rda")


###
#Only include protein coding genes and perform correlation with spearman

#make directory structure on CHPC
system('ssh dubeu@login.chpc.wustl.edu "mkdir -p
/scratch/dubeu/Synapse/NetworkAnalysis/BM/top10kspearman"')

#prepare batch script for 10k
system("cat /home/dubeu/Synapse/NetworkAnalysis/Dev/top10kspearman.batch | sed 's/AREA/BM/g'
> /home/dubeu/Synapse/NetworkAnalysis/BM/BM_top10kspearman.batch")
```

```
#prepare R script for 10k
system("cat /home/dubeu/Synapse/NetworkAnalysis/Dev/top10kspearman.R | sed 's/AREA/BM/g' >
/home/dubeu/Synapse/NetworkAnalysis/BM/BM_top10kspearman.R")

#Upload files

#counts
system("scp -r /home/dubeu/Synapse/NetworkAnalysis/BM/norm_BM_CDR_counts_regress.txt
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/BM")
system("scp -r
/home/dubeu/Synapse/NetworkAnalysis/BM/norm_BM_salmon_CDR_counts_regress.txt
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/BM")

#top10k
system("scp -r /home/dubeu/Synapse/NetworkAnalysis/BM/BM_top10kspearman.R
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/BM/top10kspearman")
system("scp -r /home/dubeu/Synapse/NetworkAnalysis/BM/BM_top10kspearman.batch
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/BM/top10kspearman")

#protein coding list
system("scp -r /40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26_transcriptID_to_geneID_final_proteincoding_uniqlist
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/")

##Send qsub command top10k
system('ssh dubeu@login.chpc.wustl.edu "qsub
/scratch/dubeu/Synapse/NetworkAnalysis/BM/top10kspearman/BM_top10kspearman.batch"')


}


##BM10
MEGENA_complete_BM10 <- gsub("BM","BM10",as.list(body(MEGENA_complete)))
#Replace first line to create function
MEGENA_complete_BM10[1] <- "MEGENA_complete_BM10 <- function() {"
#Replace first line to end function
MEGENA_complete_BM10[length(MEGENA_complete_BM10)+1] <- "}"
write(MEGENA_complete_BM10, "/home/dubeu/Synapse/MEGENA_complete_BM10.R")

##BM22
MEGENA_complete_BM22 <- gsub("BM","BM22",as.list(body(MEGENA_complete)))
#Replace first line to create function
MEGENA_complete_BM22[1] <- "MEGENA_complete_BM22 <- function() {"
```

```
#Replace first line to end function
MEGENA_complete_BM22[length(MEGENA_complete_BM22)+1] <- "}"
write(MEGENA_complete_BM22, "/home/dubeu/Synapse/MEGENA_complete_BM22.R")

##BM36
MEGENA_complete_BM36 <- gsub("BM","BM36",as.list(body(MEGENA_complete)))
#Replace first line to create function
MEGENA_complete_BM36[1] <- "MEGENA_complete_BM36 <- function() {"
#Replace first line to end function
MEGENA_complete_BM36[length(MEGENA_complete_BM36)+1] <- "}"
write(MEGENA_complete_BM36, "/home/dubeu/Synapse/MEGENA_complete_BM36.R")

##BM44
MEGENA_complete_BM44 <- gsub("BM","BM44",as.list(body(MEGENA_complete)))
#Replace first line to create function
MEGENA_complete_BM44[1] <- "MEGENA_complete_BM44 <- function() {"
#Replace first line to end function
MEGENA_complete_BM44[length(MEGENA_complete_BM44)+1] <- "}"
write(MEGENA_complete_BM44, "/home/dubeu/Synapse/MEGENA_complete_BM44.R")



source('/home/dubeu/Synapse/MEGENA_complete_BM10.R')
source('/home/dubeu/Synapse/MEGENA_complete_BM22.R')
source('/home/dubeu/Synapse/MEGENA_complete_BM36.R')
source('/home/dubeu/Synapse/MEGENA_complete_BM44.R')

MEGENA_complete_BM10()
MEGENA_complete_BM22()
MEGENA_complete_BM36()
MEGENA_complete_BM44()

#Download MEGENA results
#scp -r dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis .



##########################################
##########################################
##### MEGENA Results Postprocessing AMP_AD ###############
##########################################
##########################################
```

```
## cd /home/dubeu/Synapse/NetworkAnalysis/
##scp -r dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis .

library(WGCNA)

MEGENA_PostProcessing <- function() {

setwd("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/BM/top10kspearman")

load("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/BM/top10kspearman/BM_top10kspearman.RData")


### Using WGCNA to create Module Eigengenes

#Bind out dataframe with gene and module membership information to expr dataframe

expr_ME <<- as.data.frame(expr)
expr_ME$Gene <<- rownames(expr)

out_ME <<- merge(out, expr_ME, by = "Gene")

out_ME$Gene <<- paste(out_ME$Gene, "_", out_ME$Module, sep="")

out_ME_modules <<- out_ME$Module

out_ME_expr <<- as.data.frame(out_ME)
rownames(out_ME_expr) <<- out_ME_expr$Gene
out_ME_expr$Module <<- NULL
out_ME_expr$Gene <<- NULL
out_ME_expr <<- t(as.matrix(out_ME_expr))


# Calculate eigengenes
MEList <<- moduleEigengenes(out_ME_expr, colors = out_ME_modules)

MEs <<- MEList$eigengenes
rownames(MEs) <<- rownames(out_ME_expr)

##Load in phenotype data ##
#coldata_MEGENA_BM <<- read.csv(file="/40/AD/Expression/Public_Datasets/AMP/Mount_Sinai_RNA-seq/MSBB_Unified_Phenotype/MSBB_Pheno_05_15_2018.csv", header=TRUE, na.strings = "NA")
#coldata_MEGENA_BM <<- subset(coldata_MEGENA_BM, BrodmannArea=="BM")
```

```
##Set factor levels

#coldata_MEGENA_BM$NP.1 <<- factor(coldata_MEGENA_BM$NP.1, levels=c('1','2','3','4'))
#coldata_MEGENA_BM$Apo1 <<- factor(coldata_MEGENA_BM$Apo1, levels=c('0','2','3','4'))
#coldata_MEGENA_BM$Apo2 <<- factor(coldata_MEGENA_BM$Apo2, levels=c('0','2','3','4'))
#coldata_MEGENA_BM$RACE <<- factor(coldata_MEGENA_BM$RACE, levels=c('A','B','H','W','U'))
#coldata_MEGENA_BM$RACE <<- relevel(coldata_MEGENA_BM$RACE, "W")
#coldata_MEGENA_BM$SEX <<- factor(coldata_MEGENA_BM$SEX, levels = c("M","F"))
#coldata_MEGENA_BM$batch <<- factor(coldata_MEGENA_BM$batch, levels =
levels(coldata_MEGENA_BM$batch))

#rownames(coldata_MEGENA_BM) <<- coldata_MEGENA_BM$sampleIdentifier

coldata_MEGENA_BM <<- coldata_BM
rownames(coldata_MEGENA_BM) <<- coldata_MEGENA_BM$sampleIdentifier

#### CDR
#Merge into coldata

coldata_MEGENA_BM_MEs <<- merge(coldata_MEGENA_BM, MEs, by=0)

modules <<- as.list(colnames(MEs))

my_lms <<- lapply(modules, function(x) lm(as.formula(paste0("CDR ~ TIN.median. + PMI + PC1 + PC2 +
AOD + SEX + batch + ",x)), data=coldata_MEGENA_BM_MEs))

#coef <<- lapply(my_lms, function(x) summary(x)$coefficients[,4] )

#Just pvalues
coef <- lapply(my_lms, function(x) summary(x)$coefficients[(nrow(summary(x)$coefficients)),4] )

coef_df <<- as.data.frame(unlist(coef))
modules_df <<- as.data.frame(unlist(modules))

module_pvalue <<- cbind(modules_df, coef_df)

colnames(module_pvalue) <<- c("Module", "pvalue")

module_pvalue$Module <<- gsub("ME", "", module_pvalue$Module)

module_pvalue$padj <<- p.adjust(module_pvalue$pvalue, method = "BH")

gene_module_pval <<- merge(out, module_pvalue, by = "Module")
```

```
gene_module_pval_sigonly <<- subset(gene_module_pval, padj < 0.05)

#Number of modules
length(unique(gene_module_pval$Module))

#Number of significant modules
length(unique(gene_module_pval_sigonly$Module))

#Number of significant modules with at least one circRNA
length(unique(gene_module_pval_sigonly[grepl("circ", gene_module_pval_sigonly$Gene),]$Module))

gene_module_pval_sigonly_linearonly <<- gene_module_pval_sigonly[!grepl("circ",
gene_module_pval_sigonly$Gene),]

##Loop to produce module file list for FUMA analysis

modulelist <<- unique(gene_module_pval_sigonly$Module)
main.dir = getwd()

dir.create(file.path(main.dir, "modules_lineargenes"), showWarnings = FALSE)

setwd(file.path(main.dir, "modules_lineargenes"))

for (module in modulelist) {

linearlist <<- gene_module_pval_sigonly_linearonly[grepl((paste0("^",module,"$")),
gene_module_pval_sigonly_linearonly$Module),]$Gene

write.table(linearlist, file=module, sep= "\t", quote=F, row.names=F, col.names=F)

}

setwd(main.dir)

write.table(module_pvalue, file = "module_pval_CDR", sep = "\t", quote=F, row.names=FALSE)

#See which circRNAs are significant in meta analysis

MEGENA_BMCDR <<-
read.table("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/BM/top10kspearman/multiscale
_significant.modules.reformat.txt" , header=TRUE)
```

```
MEGENA_BMCDR_ME <<-
read.table("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/BM/top10kspearman/module_p
val_CDR", header = TRUE)
colnames(MEGENA_BMCDR_ME) <<- c("Module","ModulePval")
MEGENA_BMCDR_ME$Module <<- gsub("ME","", MEGENA_BMCDR_ME$Module)

META_MEGENA_BMCDR <<- merge(both_CDR_BM, MEGENA_BMCDR, by = "Gene")
META_MEGENA_BMCDR <<- merge(META_MEGENA_BMCDR, MEGENA_BMCDR_ME, by = "Module")

META_MEGENA_BMCDR_sigOnly <<- subset(META_MEGENA_BMCDR, i_adjpval < 0.05)


}

##BM10
MEGENA_PostProcessing_BM10 <- gsub("BM","BM10",as.list(body(MEGENA_PostProcessing)))
#Replace first line to create function
MEGENA_PostProcessing_BM10[1] <- "MEGENA_PostProcessing_BM10 <- function() {"
#Replace first line to end function
MEGENA_PostProcessing_BM10[length(MEGENA_PostProcessing_BM10)+1] <- "}"
write(MEGENA_PostProcessing_BM10, "/home/dubeu/Synapse/MEGENA_PostProcessing_BM10.R")

##BM22
MEGENA_PostProcessing_BM22 <- gsub("BM","BM22",as.list(body(MEGENA_PostProcessing)))
#Replace first line to create function
MEGENA_PostProcessing_BM22[1] <- "MEGENA_PostProcessing_BM22 <- function() {"
#Replace first line to end function
MEGENA_PostProcessing_BM22[length(MEGENA_PostProcessing_BM22)+1] <- "}"
write(MEGENA_PostProcessing_BM22, "/home/dubeu/Synapse/MEGENA_PostProcessing_BM22.R")

##BM36
MEGENA_PostProcessing_BM36 <- gsub("BM","BM36",as.list(body(MEGENA_PostProcessing)))
#Replace first line to create function
MEGENA_PostProcessing_BM36[1] <- "MEGENA_PostProcessing_BM36 <- function() {"
#Replace first line to end function
MEGENA_PostProcessing_BM36[length(MEGENA_PostProcessing_BM36)+1] <- "}"
write(MEGENA_PostProcessing_BM36, "/home/dubeu/Synapse/MEGENA_PostProcessing_BM36.R")

##BM44
MEGENA_PostProcessing_BM44 <- gsub("BM","BM44",as.list(body(MEGENA_PostProcessing)))
#Replace first line to create function
MEGENA_PostProcessing_BM44[1] <- "MEGENA_PostProcessing_BM44 <- function() {"
#Replace first line to end function
MEGENA_PostProcessing_BM44[length(MEGENA_PostProcessing_BM44)+1] <- "}"
```

```
write(MEGENA_PostProcessing_BM44, "/home/dubeu/Synapse/MEGENA_PostProcessing_BM44.R")



source('/home/dubeu/Synapse/MEGENA_PostProcessing_BM10.R')
source('/home/dubeu/Synapse/MEGENA_PostProcessing_BM22.R')
source('/home/dubeu/Synapse/MEGENA_PostProcessing_BM36.R')
source('/home/dubeu/Synapse/MEGENA_PostProcessing_BM44.R')

MEGENA_PostProcessing_BM10()
MEGENA_PostProcessing_BM22()
MEGENA_PostProcessing_BM36()
MEGENA_PostProcessing_BM44()



##########################################
##########################################
##### MEGENA Results EigenGene Plot ###############
##########################################
##########################################

##Generate plot based on eigengene

library(ggpubr)

ME_BM44_condition <- function(module, y_space=20){
coldata_BM44_MEs_forplot <- coldata_MEGENA_BM44_MEs[,c("CDR","NP.1","bbscore",module)]
colnames(coldata_BM44_MEs_forplot) <- c("CDR","NP.1","bbscore","ME")
coldata_BM44_MEs_forplot$CDR <- as.factor(coldata_BM44_MEs_forplot$CDR)
coldata_BM44_MEs_forplot$bbscore <- as.factor(coldata_BM44_MEs_forplot$bbscore)
coldata_BM44_MEs_forplot <- subset(coldata_BM44_MEs_forplot, as.numeric(NP.1) <= 2)
coldata_BM44_MEs_forplot$condition <- "Control"
coldata_BM44_MEs_forplot[as.numeric(coldata_BM44_MEs_forplot$NP.1)==2,]$condition <- "AD"
p <- ggplot(coldata_BM44_MEs_forplot, aes(condition, ME)) + geom_boxplot(aes(colour = condition),
show.legend=F) + ggtitle("Module Enrichment") + labs(y = "A.U.", x = "") + theme_light() +
stat_compare_means(comparisons=list(c("Control", "AD")),label="p.signif") + theme(plot.title =
element_text(hjust = 0.5, size=27),axis.text=element_text(size=25, face="bold")) +
scale_color_manual(values=c("blue", "black"))
#via: https://stackoverflow.com/questions/48550525/ggpubr-change-font-size-of-stat-compare-means-
kruskal-wallis-p-values
p$layers[[2]]$aes_params$textsize <- 20
p$layers[[2]]$aes_params$vjust <- 0.7
```

```
p
}

##########################################
##########################################
##### MEGENA Results parietal ###############
##########################################
##########################################

setwd("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/parietal/top10kspearman")

load("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/parietal/top10kspearman/parietal_top
10kspearman.RData")

##Post MEGENA processing code, after load MEGENA run RData file
library(WGCNA)

### Using WGCNA to create Module Eigengenes

#Bind out dataframe with gene and module membership information to expr dataframe

expr_ME <- as.data.frame(expr)
expr_ME$Gene <- rownames(expr)

out_ME <- merge(out, expr_ME, by = "Gene")

out_ME$Gene <- paste(out_ME$Gene, "_", out_ME$Module, sep="")

out_ME_modules <- out_ME$Module

out_ME_expr <- as.data.frame(out_ME)
rownames(out_ME_expr) <- out_ME_expr$Gene
out_ME_expr$Module <- NULL
out_ME_expr$Gene <- NULL
out_ME_expr <- t(as.matrix(out_ME_expr))


# Calculate eigengenes
MEList = moduleEigengenes(out_ME_expr, colors = out_ME_modules)

MEs = MEList$eigengenes
rownames(MEs) <- rownames(out_ME_expr)
```

```
##load Parietal phenotype / covariate information
coldata_MEGENA_parietal <- read.csv(file="/40/AD/Expression/KnightADRCParietal/03.-
phenotype/Pheno", header=TRUE, na.strings = "NA")
#Restrict analyses to parietal tissues
coldata_MEGENA_parietal <- subset(coldata_MEGENA_parietal, brain_region == "parietal")


#Rename
coldata_MEGENA_parietal$ID_RNAseq <- gsub("VY-", "VY.", coldata_MEGENA_parietal$ID_RNAseq)

#Temporarily Assigning Ethnicity#
coldata_MEGENA_parietal$Ethnicity <- coldata_MEGENA_parietal$Ethnicity_temp



##Remove Consensus Outliers
coldata_MEGENA_parietal <- subset(coldata_MEGENA_parietal, ConsensusOutlier_RemoveBefore != 1)

#Set rownames
rownames(coldata_MEGENA_parietal) <- coldata_MEGENA_parietal$ID_RNAseq

cts_parietal <- cts_parietal[, rownames(coldata_MEGENA_parietal)]


#Set up factor levels
coldata_MEGENA_parietal$pool <- factor(coldata_MEGENA_parietal$pool, levels=c('1','2'))
coldata_MEGENA_parietal$GENDER <- factor(coldata_MEGENA_parietal$GENDER, levels=c('1','2'))
coldata_MEGENA_parietal$Ethnicity <- factor(coldata_MEGENA_parietal$Ethnicity, levels = c("EA",
"AA"))
coldata_MEGENA_parietal$condition <- droplevels(coldata_MEGENA_parietal$condition)

rownames(coldata_MEGENA_parietal) <- coldata_MEGENA_parietal$ID_RNAseq

#### CDR
#Merge into coldata

coldata_MEGENA_parietal_MEs <- merge(coldata_MEGENA_parietal, MEs, by=0)

modules <- as.list(colnames(MEs))

my_lms <- lapply(modules, function(x) lm(as.formula(paste0("CDR_e ~ TIN.median. + PMI + PC1 + PC2 +
AGE_at_death + GENDER + pool + ",x)), data=coldata_MEGENA_parietal_MEs))

coef <- lapply(my_lms, function(x) summary(x)$coefficients[,4] )
```

```
#Just pvalues
coef <- lapply(my_lms, function(x) summary(x)$coefficients[(nrow(summary(x)$coefficients)),4] )

coef_df <- as.data.frame(unlist(coef))
modules_df <- as.data.frame(unlist(modules))

module_pvalue <- cbind(modules_df, coef_df)

colnames(module_pvalue) <- c("Module", "pvalue")

module_pvalue$Module <- gsub("ME", "", module_pvalue$Module)

module_pvalue$padj <- p.adjust(module_pvalue$pvalue, method = "BH")

gene_module_pval <- merge(out, module_pvalue, by = "Module")

gene_module_pval_sigonly <- subset(gene_module_pval, padj < 0.05)

#Number of modules
length(unique(gene_module_pval$Module))

#Number of significant modules
length(unique(gene_module_pval_sigonly$Module))

#Number of significant modules with at least one circRNA
length(unique(gene_module_pval_sigonly[grepl("circ", gene_module_pval_sigonly$Gene),]$Module))

gene_module_pval_sigonly_linearonly <- gene_module_pval_sigonly[!grepl("circ",
gene_module_pval_sigonly$Gene),]

##Loop to produce module file list for FUMA analysis

modulelist <- unique(gene_module_pval_sigonly$Module)
main.dir = getwd()

dir.create(file.path(main.dir, "modules_lineargenes"), showWarnings = FALSE)

setwd(file.path(main.dir, "modules_lineargenes"))

for (module in modulelist) {

linearlist <- gene_module_pval_sigonly_linearonly[grepl((paste0("^",module,"$")),
gene_module_pval_sigonly_linearonly$Module),]$Gene
```

```
write.table(linearlist, file=module, sep= "\t", quote=F, row.names=F, col.names=F)

}

setwd(main.dir)

write.table(module_pvalue, file = "module_pval_CDR", sep = "\t", quote=F, row.names=FALSE)

#See which circRNAs are significant in meta analysis

MEGENA_parietalCDR <-
read.table("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/parietal/top10kspearman/multiscale_significant.modules.reformat.txt" , header=TRUE)

MEGENA_parietalCDR_ME <-
read.table("/home/dubeu/Synapse/NetworkAnalysis/NetworkAnalysis/parietal/top10kspearman/module_pval_CDR", header = TRUE)
colnames(MEGENA_parietalCDR_ME) <- c("Module","ModulePval")
MEGENA_parietalCDR_ME$Module <- gsub("ME","", MEGENA_parietalCDR_ME$Module)

META_MEGENA_parietalCDR <- merge(both_CDR_BM44, MEGENA_parietalCDR, by = "Gene")
META_MEGENA_parietalCDR <- merge(META_MEGENA_parietalCDR, MEGENA_parietalCDR_ME, by = "Module")

META_MEGENA_parietalCDR_sigOnly <- subset(META_MEGENA_parietalCDR, i_adjpval < 0.05)



```
Template files
R Script
```
library("MEGENA")


expr_circ <-
as.matrix(read.table("/scratch/dubeu/Synapse/NetworkAnalysis/AREA/norm_AREA_CDR_counts_regress.txt", header=TRUE))

expr_linear <-
as.matrix(read.table("/scratch/dubeu/Synapse/NetworkAnalysis/AREA/norm_AREA_salmon_CDR_counts_regress.txt", header=TRUE))
```

```
ProteinCoding <-
(read.table("/scratch/dubeu/Synapse/NetworkAnalysis/gencode.v26_transcriptID_to_geneID_final_prot
eincoding_uniqlist", header=FALSE))

expr_linear <- expr_linear[intersect(rownames(expr_linear), as.character(ProteinCoding[,1])),]

max.genes=10000
if(nrow(expr_linear) >= max.genes) expr_linear=expr_linear[rank(-apply(expr_linear,1,var,na.rm=TRUE))
<= max.genes,]

##No max gene filter
#expr=expr[rank(-apply(expr,1,var,na.rm=TRUE)),]

expr <- rbind(expr_circ, expr_linear)

setwd("/scratch/dubeu/Synapse/NetworkAnalysis/AREA/top10kspearman")

n.cor.perm=10
FDR.cutoff = 0.05
n.cores=4
max.module.size=NULL
min.module.size = 10
n.hub.perm=100
module.pval = 0.05
hub.pval = 0.05
doPar = TRUE
remove.unsig = TRUE
#


#### register multiple cores if needed: note that set.parallel.backend() is deprecated.
if (doPar & getDoParWorkers() == 1)
{
   cl <- parallel::makeCluster(n.cores)
   registerDoParallel(cl)
   # check how many workers are there
   cat(paste("number of cores to use:",getDoParWorkers(),"\n",sep = ""))
}

# calculate correlation
ijw <- calculate.correlation(datExpr = expr, doPerm = n.cor.perm, method = 'spearman', FDR.cutoff =
FDR.cutoff,output.permFDR=FALSE,output.corTable=FALSE,num.cores=n.cores)
```

```
ijw=ijw[,1:3]
ijw[,3]=abs(ijw[,3])
ijw=ijw[order(ijw[,3],decreasing=TRUE),]

# calculate PFN
el <- calculate.PFN(ijw, doPar = doPar, num.cores = n.cores)

g <- graph.data.frame(el,directed = FALSE)

if(is.null(max.module.size)) max.module.size = min(5000,ceiling(vcount(g)/3))

MEGENA.output <- do.MEGENA(g, mod.pval = module.pval, hub.pval = hub.pval, remove.unsig = TRUE,
min.size = 10, max.size = max.module.size, doPar = doPar, num.cores = n.cores, n.perm = n.hub.perm,
save.output = FALSE)


###### unregister cores as these are not needed anymore.
if (getDoParWorkers() > 1)
{
  env <- foreach:::.foreachGlobals
  rm(list=ls(name=env), pos=env)
}


summary.output <- MEGENA.ModuleSummary(MEGENA.output,
                            mod.pvalue = module.pval, hub.pvalue = hub.pval,
                            min.size = 10,max.size = max.module.size,
                            output.sig = TRUE)

geneSetf="multiscale_significant.modules.txt"

geneSetf="multiscale_significant.modules.txt"
output.geneSet.file(summary.output$modules,geneSetf)
#reformat
geneSet=readLines(geneSetf)
out=do.call(rbind,lapply(geneSet,function(x) {a=strsplit(x,'\\t')[[1]];data.frame(Gene=a[-
1],Module=a[1],stringsAsFactors=FALSE)}))
write.table(out,file='multiscale_significant.modules.reformat.txt',row.names=FALSE,col.names=TRUE,se
p='\t',quote=FALSE)

save.image("AREA_top10kspearman.RData")
```

Bash Script

```
#!/bin/bash
```

Script to submit the file

type qsub

qsub /scratch/dubeu/Synapse/NetworkAnalysis/AREA/top10kspearman/AREA_top10kspearman.batch

Cluster Processing Script

#!/bin/bash

Give the job a name to help keep track of running jobs (optional)
#PBS -N AREA_top10kspearman

Specify the resources needed
#PBS -l nodes=1:ppn=4,walltime=24:00:00

#Command to Run
#qsub /scratch/dubeu/Synapse/NetworkAnalysis/AREA/top10kspearman/AREA_top10kspearman.batch

cd /scratch/dubeu/Synapse/NetworkAnalysis/AREA/top10kspearman

Load the environmental variables necessary for running R
module load R-3.4.2

Finally run the R benchmark with the command
Rscript AREA_top10kspearman.R

#Download files at the end
#scp -r
dubeu@dtn01.chpc.wustl.edu:/scratch/dubeu/Synapse/NetworkAnalysis/AREA/top10kspearman
/home/dubeu/Synapse/NetworkAnalysis/AREA/

MicroRNA Binding Site Prediction Analyses

##
Dube et al., 2019 - circRNAs in AD
##

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: MicroRNA Binding Site Prediction

#####Get the TargetScan70 software
```{bash}
```

```
wget http://www.targetscan.org/vert_70/vert_70_data_download/targetscan_70.zip
wget http://www.targetscan.org/vert_70/vert_70_data_download/miR_Family_Info.txt.zip

#Convert miRNA file as per readme
sed '1,1d' /home/dubeu/Synapse/miR_Family_info.txt | cut -f1,2,3 | sort -u > miR_Family_info_all.txt
```

#####Get circRNA sequences and Process for TargetScan

```{bash}
##Download the extract script from DCC
wget https://raw.githubusercontent.com/dieterich-lab/DCC/master/scripts/getcircfasta
mv getcircfasta getcircfasta.py

##Remove the header
tail -n+2 <(sed 's/Chr/Chr\tChr/g' /home/dubeu/Synapse/circCoordinates_parietal | cut -f2-9) >
circs.bed

cat /40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26.primary_assembly.annotation.gtf | awk 'OFS="\t" {if ($3=="exon") {print $1,$4-
1,$5,$10"_exon_"$22,$16,$7}}' | tr -d '";' > /40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26.primary_assembly.annotation_exons.bed

##Run the script

python getcircfasta.py -f /40/AD/Expression/KnightADRCParietal/06.-
References/GRCh38.primary_assembly.genome.fa -c circs.bed -e
/40/AD/Expression/KnightADRCParietal/06.-
References/gencode.v26.primary_assembly.annotation_exons.bed -o circCoordinates_parietal_hg38.fa

##Process the fasta file for TargetScan analysis

awk 'NR%2{printf "%s ",$0;next;}1' /home/dubeu/Synapse/circCoordinates_parietal_hg38.fa |
spaces_to_one_tab.sh | awk '{print $4"::"$1":"$2"-"$3,"9606",$9}' | sed 's/>//g' | awk
'BEGIN{OFS="\t";} {gsub("T","U",$3);print}' > circCoordinates_parietal_hg38
```

####Run TargetScan Analysis

```{bash}
```

```
/home/dubeu/Synapse/microRNA_analysis/targetscan_70/targetscan_70.pl
/home/dubeu/Synapse/microRNA_analysis/miR_Family_info_all.txt circCoordinates_parietal_hg38
circCoordinates_parietal_hg38_prediction

##Create a list of genes

cut -d: -f1 circCoordinates_parietal_hg38 | sort -u > genelist

ProcessScript.sh

        #!/bin/bash

        grep -w $1 circCoordinates_parietal_hg38 > circCoordinates_parietal_hg38_$1

        /home/dubeu/Synapse/microRNA_analysis/targetscan_70/targetscan_70.pl
/home/dubeu/Synapse/microRNA_analysis/miR_Family_info_all.txt circCoordinates_parietal_hg38_$1
targetscan_70_circCoordinates_parietal_hg38_$1.txt

        rm circCoordinates_parietal_hg38_$1

        exit

parallel -j 50 --load 50 -a genelist ./ProcessScript.sh


```

###Processing output in R
```{bash}


## miRNA Processing


##Read into R following the processing
require(data.table)
PredictedmiRNAs <-
fread("/home/dubeu/Synapse/microRNA_analysis/circCoordinates_parietal_hg38_prediction",
header=TRUE)

require(dplyr)
PredictedmiRNAs <- PredictedmiRNAs %>% group_by(a_Gene_ID, miRNA_family_ID) %>% mutate(count
= n())
```

```
PredictedmiRNAs <- unique(PredictedmiRNAs[,c("a_Gene_ID","miRNA_family_ID","count")])
PredictedmiRNAs$gene <- paste0("circ",gsub("::*.*","",PredictedmiRNAs$a_Gene_ID))

PredictedmiRNAs_sigList_BM44_CDR <- PredictedmiRNAs[(PredictedmiRNAs$gene %in%
sigList_BM44_CDR),]

##Collapse to the gene level and choose the greatest number
PredictedmiRNAs_sigList_BM44_CDR <-
PredictedmiRNAs_sigList_BM44_CDR[,c("gene","miRNA_family_ID","count")]

PredictedmiRNAs_sigList_BM44_CDR <- PredictedmiRNAs_sigList_BM44_CDR %>% group_by(gene,
miRNA_family_ID) %>% mutate(max = max(count))

PredictedmiRNAs_sigList_BM44_CDR <-
unique(PredictedmiRNAs_sigList_BM44_CDR[,c("gene","miRNA_family_ID","max")])

colnames(PredictedmiRNAs_sigList_BM44_CDR) <- c("gene","miRNA_family_ID","count")

##Summary table per gene

PredictedmiRNAs_sigList_BM44_CDR_summary <-
unique(PredictedmiRNAs_sigList_BM44_CDR[,c("miRNA_family_ID","count","gene")])

PredictedmiRNAs_sigList_BM44_CDR_summary <- PredictedmiRNAs_sigList_BM44_CDR_summary %>%
group_by(gene) %>% mutate(Total = n(), Min = min(count), Max = max(count), Total1 =
length(gene[count > 1]))

PredictedmiRNAs_sigList_BM44_CDR_summary <-
unique(PredictedmiRNAs_sigList_BM44_CDR_summary[,c("gene","Total","Total1","Min","Max")])


```

Overlap and other calculations

```
########################################
## Dube et al., 2019 - circRNAs in AD ##
########################################

##Author: Umber Dube
##Contact: udube@wustl.edu

##Code Section: Overlap Calculations and Numbers for Paper
```{r}
```
```

```
##########################################
##########################################
## Calculate Parietal Overlap ##
##########################################
##########################################


GeneList_parietal_condition <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_CDR <- rownames(subset(res_parietal_CDR_filter, padj < 0.05))

GeneList_parietal_braak <- rownames(subset(res_parietal_braak_filter, padj < 0.05))


grid.newpage()
draw.triple.venn(area1 = length(GeneList_parietal_condition), area2 = length(GeneList_parietal_CDR),
area3 = length(GeneList_parietal_braak), n12 = length(intersect(GeneList_parietal_condition,
GeneList_parietal_CDR)), n23 = length(intersect(GeneList_parietal_CDR, GeneList_parietal_braak)), n13
= length(intersect(GeneList_parietal_condition, GeneList_parietal_braak)),
   n123 = length(Reduce(intersect, list(GeneList_parietal_condition, GeneList_parietal_CDR,
GeneList_parietal_braak))), category = c("LOAD Case", "CDR", "Braak"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.8,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))

intersect(GeneList_parietal_condition,GeneList_parietal_CDR,GeneList_parietal_braak)


###Condition
##Aggregate / intersect list of genes
#GeneList_parietal_LOADvsCO <- unique(c(rownames(subset(res_parietal_condition_filter, padj <
0.05)), rownames(subset(res_parietal_braak_filter, padj < 0.05)),
rownames(subset(res_parietal_CDR_filter, padj < 0.05))))

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))
```

```
GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.triple.venn(area1 = length(GeneList_parietal_LOADvsCO), area2 =
length(GeneList_parietal_ADADvsLOAD), area3 = length(GeneList_parietal_ADADvsCO), n12 =
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD)), n23 =
length(intersect(GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)), n13 =
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsCO)),
   n123 = length(Reduce(intersect, list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO))), category = c("ADvsCO", "ADADvsAD*", "ADADvsCO"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.5,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))




##########################################
##########################################
## Check of consistency in direction of effect ADAD and LOAD ##
##########################################
##########################################

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


OverlapADAD_CO_ADAD_LOAD <-
cbind(res_parietalADAD_condition_filter_ADADvsCO[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange,
```

```
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange)
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, samesign=
as.data.frame(sign(OverlapADAD_CO_ADAD_LOAD[,1])==sign(OverlapADAD_CO_ADAD_LOAD[,2])))
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, increasingSeverity=
as.data.frame(abs(OverlapADAD_CO_ADAD_LOAD[,1])>abs(OverlapADAD_CO_ADAD_LOAD[,2])))



cbind(res_parietal_condition_filter[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_condition_filter_ADADvsCO[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[Reduce(intersect,
list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO)),]$log2FoldChange)



##########################################
##########################################
## Calculate overlap between ADAD and LOAD ##
##########################################
##########################################

###Condition
##Aggregate / intersect list of genes
#GeneList_parietal_LOADvsCO <- unique(c(rownames(subset(res_parietal_condition_filter, padj <
0.05)), rownames(subset(res_parietal_braak_filter, padj < 0.05)),
rownames(subset(res_parietal_CDR_filter, padj < 0.05))))

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.triple.venn(area1 = length(GeneList_parietal_LOADvsCO), area2 =
length(GeneList_parietal_ADADvsLOAD), area3 = length(GeneList_parietal_ADADvsCO), n12 =
```

```
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD)), n23 =
length(intersect(GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)), n13 =
length(intersect(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsCO)),
   n123 = length(Reduce(intersect, list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO))), category = c("LOADvsCO", "ADADvsLOAD", "ADADvsCO"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.5,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))




##########################################
##########################################
## Check of consistency in direction of effect ADAD and LOAD ##
##########################################
##########################################

GeneList_parietal_LOADvsCO <- rownames(subset(res_parietal_condition_filter, padj < 0.05))

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


OverlapADAD_CO_ADAD_LOAD <-
cbind(res_parietalADAD_condition_filter_ADADvsCO[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange,
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[intersect(GeneList_parietal_ADADvsCO,
GeneList_parietal_ADADvsLOAD),]$log2FoldChange)
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, samesign=
as.data.frame(sign(OverlapADAD_CO_ADAD_LOAD[,1])==sign(OverlapADAD_CO_ADAD_LOAD[,2])))
OverlapADAD_CO_ADAD_LOAD <- cbind(OverlapADAD_CO_ADAD_LOAD, increasingSeverity=
as.data.frame(abs(OverlapADAD_CO_ADAD_LOAD[,1])>abs(OverlapADAD_CO_ADAD_LOAD[,2])))
```

```
cbind(res_parietal_condition_filter[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_condition_filter_ADADvsCO[Reduce(intersect, list(GeneList_parietal_LOADvsCO,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)),]$log2FoldChange,
res_parietalADAD_conditionBraak_filter_ADADvsLOAD[Reduce(intersect,
list(GeneList_parietal_LOADvsCO, GeneList_parietal_ADADvsLOAD,
GeneList_parietal_ADADvsCO)),]$log2FoldChange)



##########################################
##########################################
## Calculate BM44 Overlap ##
##########################################
##########################################


GeneList_BM44_condition <- rownames(subset(res_BM44_condition_filter, padj < 0.05))

GeneList_BM44_CDR <- rownames(subset(res_BM44_CDR_filter, padj < 0.05))

GeneList_BM44_braak <- rownames(subset(res_BM44_braak_filter, padj < 0.05))

GeneList_BM44_PlaqueMean <- rownames(subset(res_BM44_PlaqueMean_filter, padj < 0.05))


grid.newpage()
draw.quad.venn(area1 = length(GeneList_BM44_condition), area2 = length(GeneList_BM44_CDR), area3
= length(GeneList_BM44_braak), area4 = length(GeneList_BM44_PlaqueMean), n12 =
length(intersect(GeneList_BM44_condition, GeneList_BM44_CDR)), n13 =
length(intersect(GeneList_BM44_condition, GeneList_BM44_braak)), n14 =
length(intersect(GeneList_BM44_condition, GeneList_BM44_PlaqueMean)), n23 =
length(intersect(GeneList_BM44_CDR, GeneList_BM44_braak)), n24 =
length(intersect(GeneList_BM44_CDR, GeneList_BM44_PlaqueMean)), n34 =
length(intersect(GeneList_BM44_braak, GeneList_BM44_PlaqueMean)),
   n123 = length(Reduce(intersect, list(GeneList_BM44_condition, GeneList_BM44_CDR,
GeneList_BM44_braak))), n124 = length(Reduce(intersect, list(GeneList_BM44_condition,
GeneList_BM44_CDR, GeneList_BM44_PlaqueMean))), n134 = length(Reduce(intersect,
list(GeneList_BM44_condition, GeneList_BM44_braak, GeneList_BM44_PlaqueMean))), n234 =
length(Reduce(intersect, list(GeneList_BM44_CDR, GeneList_BM44_braak,
GeneList_BM44_PlaqueMean))), n1234 = length(Reduce(intersect, list(GeneList_BM44_condition,
GeneList_BM44_CDR, GeneList_BM44_braak, GeneList_BM44_PlaqueMean))), category = c("LOAD
Case", "CDR", "Braak", "PlaqueMean"), lty = "blank",
          cex = 1.5,
```

```
        fontface = "bold",
        fontfamily = "sans",
        cat.cex = 1.2,
        cat.fontface = "bold",
        cat.default.pos = "outer",
  fill = c("skyblue", "pink1", "mediumorchid", "green"))




##########################################
##########################################
## Calculate Parietal and BM44 Overlap Overall ##
##########################################
##########################################


GeneList_BM44_ALL <- unique(c(GeneList_BM44_condition, GeneList_BM44_CDR,
GeneList_BM44_braak, GeneList_BM44_PlaqueMean))
#GeneList_BM44_ALL <- Reduce(intersect, list(GeneList_BM44_condition, GeneList_BM44_CDR,
GeneList_BM44_braak, GeneList_BM44_PlaqueMean))

GeneList_parietal_ALL <- unique(c(GeneList_parietal_condition, GeneList_parietal_CDR,
GeneList_parietal_braak))

#All
grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.pairwise.venn(area1 = length(GeneList_parietal_ALL), area2 = length(GeneList_BM44_ALL),
cross.area = length(intersect(GeneList_parietal_ALL, GeneList_BM44_ALL)), category = c("Parietal \n
Discovery", "BM44 \n Replication"), lty = "blank", cex = 1.5, fontface = "bold", fontfamily = "sans",
cat.cex = 1.2, cat.fontface = "bold", cat.default.pos = "outer", fill = c("skyblue", "pink1"))




##########################################
##########################################
## Calculate Parietal and BM44 Overlap CDR ##
##########################################
##########################################
```

```
grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.pairwise.venn(area1 = length(GeneList_parietal_CDR), area2 = length(GeneList_BM44_CDR),
cross.area = length(intersect(GeneList_parietal_CDR, GeneList_BM44_CDR)), category = c("Parietal \n
Discovery", "BM44 \n Replication"), lty = "blank", cex = 1.5, fontface = "bold", fontfamily = "sans",
cat.cex = 1.2, cat.fontface = "bold", cat.default.pos = "outer", fill = c("skyblue", "pink1"))

#Same direction of effect?

sum(sign(res_parietal_CDR_filter[rownames(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,],
pvalue < 0.05)),]$log2FoldChange) == sign(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,], pvalue
< 0.05)$log2FoldChange))

##########################################
##########################################
## Calculate Parietal and BM44 Overlap braak ##
##########################################
##########################################

grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.pairwise.venn(area1 = length(GeneList_parietal_braak), area2 = length(GeneList_BM44_braak),
cross.area = length(intersect(GeneList_parietal_braak, GeneList_BM44_braak)), category = c("Parietal \n
Discovery", "BM44 \n Replication"), lty = "blank", cex = 1.5, fontface = "bold", fontfamily = "sans",
cat.cex = 1.2, cat.fontface = "bold", cat.default.pos = "outer", fill = c("skyblue", "pink1"))

#Same direction of effect?

sum(sign(res_parietal_braak_filter[rownames(subset(res_BM44_braak_filter[GeneList_parietal_braak,],
pvalue < 0.05)),]$log2FoldChange) == sign(subset(res_BM44_braak_filter[GeneList_parietal_braak,],
pvalue < 0.05)$log2FoldChange))


##########################################
##########################################
## Calculate Parietal and BM44 Overlap condition ##
##########################################
##########################################

grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.pairwise.venn(area1 = length(GeneList_parietal_condition), area2 =
length(GeneList_BM44_condition), cross.area = length(intersect(GeneList_parietal_condition,
GeneList_BM44_condition)), category = c("Parietal \n Discovery", "BM44 \n Replication"), lty = "blank",
```

```
cex = 1.5, fontface = "bold", fontfamily = "sans", cat.cex = 1.2, cat.fontface = "bold", cat.default.pos =
"outer", fill = c("skyblue", "pink1"))

#Same direction of effect?

sum(sign(res_parietal_condition_filter[rownames(subset(res_BM44_condition_filter[GeneList_parietal_
condition,], pvalue < 0.05)),]$log2FoldChange) ==
sign(subset(res_BM44_condition_filter[GeneList_parietal_condition,], pvalue < 0.05)$log2FoldChange))



##########################################
##########################################
## Calculate overlap between PlaqueMean ##
##########################################
##########################################



GeneList_BM10_PlaqueMean <- rownames(subset(res_BM10_PlaqueMean_filter, padj < 0.05))

GeneList_BM22_PlaqueMean <- rownames(subset(res_BM22_PlaqueMean_filter, padj < 0.05))

GeneList_BM36_PlaqueMean <- rownames(subset(res_BM36_PlaqueMean_filter, padj < 0.05))

GeneList_BM44_PlaqueMean <- rownames(subset(res_BM44_PlaqueMean_filter, padj < 0.05))


grid.newpage()
draw.quad.venn(area1 = length(GeneList_BM10_PlaqueMean), area2 =
length(GeneList_BM22_PlaqueMean), area3 = length(GeneList_BM36_PlaqueMean), area4 =
length(GeneList_BM44_PlaqueMean), n12 = length(intersect(GeneList_BM10_PlaqueMean,
GeneList_BM22_PlaqueMean)), n13 = length(intersect(GeneList_BM10_PlaqueMean,
GeneList_BM36_PlaqueMean)), n14 = length(intersect(GeneList_BM10_PlaqueMean,
GeneList_BM44_PlaqueMean)), n23 = length(intersect(GeneList_BM22_PlaqueMean,
GeneList_BM36_PlaqueMean)), n24 = length(intersect(GeneList_BM22_PlaqueMean,
GeneList_BM44_PlaqueMean)), n34 = length(intersect(GeneList_BM36_PlaqueMean,
GeneList_BM44_PlaqueMean)),
   n123 = length(Reduce(intersect, list(GeneList_BM10_PlaqueMean, GeneList_BM22_PlaqueMean,
GeneList_BM36_PlaqueMean))), n124 = length(Reduce(intersect, list(GeneList_BM10_PlaqueMean,
GeneList_BM22_PlaqueMean, GeneList_BM44_PlaqueMean))), n134 = length(Reduce(intersect,
list(GeneList_BM10_PlaqueMean, GeneList_BM36_PlaqueMean, GeneList_BM44_PlaqueMean))), n234
= length(Reduce(intersect, list(GeneList_BM22_PlaqueMean, GeneList_BM36_PlaqueMean,
GeneList_BM44_PlaqueMean))), n1234 = length(Reduce(intersect, list(GeneList_BM10_PlaqueMean,
```

```
GeneList_BM22_PlaqueMean, GeneList_BM36_PlaqueMean, GeneList_BM44_PlaqueMean))), category
= c("BM10", "BM22", "BM36", "BM44"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.2,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid", "green"))



##########################################################



##########################################
##########################################
## Calculate overlap between Meta Analyses ##
##########################################
##########################################

###CDR

##Aggregate / intersect list of genes
GeneListMeta_CDR_BM10 <- subset(both_CDR_BM10, i_adjpval < 0.05)$Gene
GeneListMeta_CDR_BM22 <- subset(both_CDR_BM22, i_adjpval < 0.05)$Gene
GeneListMeta_CDR_BM36 <- subset(both_CDR_BM36, i_adjpval < 0.05)$Gene
GeneListMeta_CDR_BM44 <- subset(both_CDR_BM44, i_adjpval < 0.05)$Gene

grid.newpage()
#pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
venn_meta_CDR <- draw.quad.venn(area1 = length(GeneListMeta_CDR_BM10), area2 =
length(GeneListMeta_CDR_BM22), area3 = length(GeneListMeta_CDR_BM36), area4 =
length(GeneListMeta_CDR_BM44), n12 = length(intersect(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM22)), n13 = length(intersect(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM36)), n14 = length(intersect(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM44)), n23 = length(intersect(GeneListMeta_CDR_BM22,
GeneListMeta_CDR_BM36)), n24 = length(intersect(GeneListMeta_CDR_BM22,
GeneListMeta_CDR_BM44)), n34 = length(intersect(GeneListMeta_CDR_BM36,
GeneListMeta_CDR_BM44)),
   n123 = length(Reduce(intersect, list(GeneListMeta_CDR_BM10, GeneListMeta_CDR_BM22,
GeneListMeta_CDR_BM36))), n124 = length(Reduce(intersect, list(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM22, GeneListMeta_CDR_BM44))), n134 = length(Reduce(intersect,
```

```
list(GeneListMeta_CDR_BM10, GeneListMeta_CDR_BM36, GeneListMeta_CDR_BM44))), n234 =
length(Reduce(intersect, list(GeneListMeta_CDR_BM22, GeneListMeta_CDR_BM36,
GeneListMeta_CDR_BM44))), n1234 = length(Reduce(intersect, list(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM22, GeneListMeta_CDR_BM36, GeneListMeta_CDR_BM44))), category =
c("Meta\nPCtx-BM10", "Meta\nPCtx-BM22", "Meta\nPCtx-BM36", "Meta\nPCtx-BM44"), lty = "blank",
        cex = 1,
        fontface = "bold",
        fontfamily = "sans",
        cat.cex = 0.8,
        cat.pos=0,
        cat.fontface = "bold",
        cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid", "green"))


###braak

##Aggregate / intersect list of genes
GeneListMeta_braak_BM10 <- subset(both_braak_BM10, i_adjpval < 0.05)$Gene
GeneListMeta_braak_BM22 <- subset(both_braak_BM22, i_adjpval < 0.05)$Gene
GeneListMeta_braak_BM36 <- subset(both_braak_BM36, i_adjpval < 0.05)$Gene
GeneListMeta_braak_BM44 <- subset(both_braak_BM44, i_adjpval < 0.05)$Gene

grid.newpage()
#pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
venn_meta_braak <- draw.quad.venn(area1 = length(GeneListMeta_braak_BM10), area2 =
length(GeneListMeta_braak_BM22), area3 = length(GeneListMeta_braak_BM36), area4 =
length(GeneListMeta_braak_BM44), n12 = length(intersect(GeneListMeta_braak_BM10,
GeneListMeta_braak_BM22)), n13 = length(intersect(GeneListMeta_braak_BM10,
GeneListMeta_braak_BM36)), n14 = length(intersect(GeneListMeta_braak_BM10,
GeneListMeta_braak_BM44)), n23 = length(intersect(GeneListMeta_braak_BM22,
GeneListMeta_braak_BM36)), n24 = length(intersect(GeneListMeta_braak_BM22,
GeneListMeta_braak_BM44)), n34 = length(intersect(GeneListMeta_braak_BM36,
GeneListMeta_braak_BM44)),
   n123 = length(Reduce(intersect, list(GeneListMeta_braak_BM10, GeneListMeta_braak_BM22,
GeneListMeta_braak_BM36))), n124 = length(Reduce(intersect, list(GeneListMeta_braak_BM10,
GeneListMeta_braak_BM22, GeneListMeta_braak_BM44))), n134 = length(Reduce(intersect,
list(GeneListMeta_braak_BM10, GeneListMeta_braak_BM36, GeneListMeta_braak_BM44))), n234 =
length(Reduce(intersect, list(GeneListMeta_braak_BM22, GeneListMeta_braak_BM36,
GeneListMeta_braak_BM44))), n1234 = length(Reduce(intersect, list(GeneListMeta_braak_BM10,
GeneListMeta_braak_BM22, GeneListMeta_braak_BM36, GeneListMeta_braak_BM44))), category =
c("Meta\nPCtx-BM10", "Meta\nPCtx-BM22", "Meta\nPCtx-BM36", "Meta\nPCtx-BM44"), lty = "blank",
        cex = 1,
        fontface = "bold",
```

```
        fontfamily = "sans",
        cat.cex = 0.8,
        cat.pos=0,
        cat.fontface = "bold",
        cat.default.pos = "outer",
  fill = c("skyblue", "pink1", "mediumorchid", "green"))


###condition

##Aggregate / intersect list of genes
GeneListMeta_condition_BM10 <- subset(both_condition_BM10, i_adjpval < 0.05)$Gene
GeneListMeta_condition_BM22 <- subset(both_condition_BM22, i_adjpval < 0.05)$Gene
GeneListMeta_condition_BM36 <- subset(both_condition_BM36, i_adjpval < 0.05)$Gene
GeneListMeta_condition_BM44 <- subset(both_condition_BM44, i_adjpval < 0.05)$Gene

grid.newpage()
#pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
venn_meta_condition <- draw.quad.venn(area1 = length(GeneListMeta_condition_BM10), area2 =
length(GeneListMeta_condition_BM22), area3 = length(GeneListMeta_condition_BM36), area4 =
length(GeneListMeta_condition_BM44), n12 = length(intersect(GeneListMeta_condition_BM10,
GeneListMeta_condition_BM22)), n13 = length(intersect(GeneListMeta_condition_BM10,
GeneListMeta_condition_BM36)), n14 = length(intersect(GeneListMeta_condition_BM10,
GeneListMeta_condition_BM44)), n23 = length(intersect(GeneListMeta_condition_BM22,
GeneListMeta_condition_BM36)), n24 = length(intersect(GeneListMeta_condition_BM22,
GeneListMeta_condition_BM44)), n34 = length(intersect(GeneListMeta_condition_BM36,
GeneListMeta_condition_BM44)),
  n123 = length(Reduce(intersect, list(GeneListMeta_condition_BM10, GeneListMeta_condition_BM22,
GeneListMeta_condition_BM36))), n124 = length(Reduce(intersect, list(GeneListMeta_condition_BM10,
GeneListMeta_condition_BM22, GeneListMeta_condition_BM44))), n134 = length(Reduce(intersect,
list(GeneListMeta_condition_BM10, GeneListMeta_condition_BM36, GeneListMeta_condition_BM44))),
n234 = length(Reduce(intersect, list(GeneListMeta_condition_BM22, GeneListMeta_condition_BM36,
GeneListMeta_condition_BM44))), n1234 = length(Reduce(intersect,
list(GeneListMeta_condition_BM10, GeneListMeta_condition_BM22, GeneListMeta_condition_BM36,
GeneListMeta_condition_BM44))), category = c("Meta\nPCtx-BM10", "Meta\nPCtx-BM22",
"Meta\nPCtx-BM36", "Meta\nPCtx-BM44"), lty = "blank",
        cex = 1,
        fontface = "bold",
        fontfamily = "sans",
        cat.cex = 0.8,
        cat.pos=0,
        cat.fontface = "bold",
        cat.default.pos = "outer",
  fill = c("skyblue", "pink1", "mediumorchid", "green"))
```

```
require(gridExtra)
venn_meta_condition_title <- grid.arrange(gTree(children=venn_meta_condition), ncol=1,
top=textGrob("AD Case", gp=gpar(fontface="bold")))
venn_meta_CDR_title <- grid.arrange(gTree(children=venn_meta_CDR), ncol=1, top=textGrob("CDR",
gp=gpar(fontface="bold")))
venn_meta_braak_title <- grid.arrange(gTree(children=venn_meta_braak), ncol=1, top=textGrob("Braak
Score", gp=gpar(fontface="bold")))

require(cowplot)
plot_grid(venn_meta_CDR_title,venn_meta_braak_title,venn_meta_condition_title, align = "v", nrow
=3, labels=c('A','B','C'))


require(cowplot)
plot_grid(gTree(children=venn_meta_condition),gTree(children=venn_meta_CDR),gTree(children=venn
_meta_braak), align = "v", nrow =3)


##BM44 Meta analyses
#Join the meta-analysis results together into a set

meta_condition_overlap <- list(GeneListMeta_condition_BM44, GeneListMeta_CDR_BM44,
GeneListMeta_braak_BM44)

length.gene.sets=sapply(meta_condition_overlap,length)

total=nrow(both_condition_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_condition_overlap[[1]], meta_condition_overlap[[2]],
meta_condition_overlap[[3]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)
```

```
###Test for significance of overlap ##

library("SuperExactTest")

##CDR
#Join the meta-analysis results together into a set

meta_CDR_overlap <- list(GeneListMeta_CDR_BM10, GeneListMeta_CDR_BM22,
GeneListMeta_CDR_BM36, GeneListMeta_CDR_BM44)

length.gene.sets=sapply(meta_CDR_overlap,length)

total=nrow(both_CDR_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_CDR_overlap[[1]], meta_CDR_overlap[[2]], meta_CDR_overlap[[3]],
 meta_CDR_overlap[[4]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


##braak
#Join the meta-analysis results together into a set

meta_braak_overlap <- list(GeneListMeta_braak_BM10, GeneListMeta_braak_BM22,
GeneListMeta_braak_BM36, GeneListMeta_braak_BM44)

length.gene.sets=sapply(meta_braak_overlap,length)

total=nrow(both_braak_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_braak_overlap[[1]], meta_braak_overlap[[2]], meta_braak_overlap[[3]],
 meta_braak_overlap[[4]])
```

```
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


##condition
#Join the meta-analysis results together into a set

meta_condition_overlap <- list(GeneListMeta_condition_BM10, GeneListMeta_condition_BM22,
GeneListMeta_condition_BM36, GeneListMeta_condition_BM44)

length.gene.sets=sapply(meta_condition_overlap,length)

total=nrow(both_condition_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_condition_overlap[[1]], meta_condition_overlap[[2]],
meta_condition_overlap[[3]],
 meta_condition_overlap[[4]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


###Calculate number with various levels of significance##


#sum(both_CDR_BM44[rownames(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,], pvalue <
0.05)),]$i_rawpval < 5e-8)

sum(both_CDR_BM44[rownames(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,], pvalue <
0.05)),]$i_rawpval < 5e-6)


##Same sign for meta analysis overlap?

#Same direction of effect?
```

```
GeneListMeta_CDR_MSBB <- Reduce(intersect, list(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM22, GeneListMeta_CDR_BM36, GeneListMeta_CDR_BM44))


sign(res_BM22_CDR_filter[GeneListMeta_CDR_MSBB,]$log2FoldChange)
sign(res_BM36_CDR_filter[GeneListMeta_CDR_MSBB,]$log2FoldChange)
sign(res_BM44_CDR_filter[GeneListMeta_CDR_MSBB,]$log2FoldChange)


sum(sign(res_parietal_CDR_filter[rownames(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,],
pvalue < 0.05)),]$log2FoldChange) == sign(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,], pvalue
< 0.05)$log2FoldChange))



##########################################
##########################################
## Calculate overlap between ADAD and Meta ##
##########################################
##########################################

###Condition
##Aggregate / intersect list of genes
#GeneList_parietal_LOADvsCO_meta <- unique(c(rownames(subset(res_parietal_condition_filter, padj <
0.05)), rownames(subset(res_parietal_braak_filter, padj < 0.05)),
rownames(subset(res_parietal_CDR_filter, padj < 0.05))))

GeneList_parietal_LOADvsCO_meta <- GeneListMeta_condition_BM44

GeneList_parietal_ADADvsCO <- rownames(subset(res_parietalADAD_condition_filter_ADADvsCO, padj
< 0.05))

GeneList_parietal_ADADvsLOAD <-
rownames(subset(res_parietalADAD_conditionBraak_filter_ADADvsLOAD, padj < 0.05))


grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.triple.venn(area1 = length(GeneList_parietal_LOADvsCO_meta), area2 =
length(GeneList_parietal_ADADvsLOAD), area3 = length(GeneList_parietal_ADADvsCO), n12 =
length(intersect(GeneList_parietal_LOADvsCO_meta, GeneList_parietal_ADADvsLOAD)), n23 =
length(intersect(GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO)), n13 =
length(intersect(GeneList_parietal_LOADvsCO_meta, GeneList_parietal_ADADvsCO)),
```

```
   n123 = length(Reduce(intersect, list(GeneList_parietal_LOADvsCO_meta,
GeneList_parietal_ADADvsLOAD, GeneList_parietal_ADADvsCO))), category = c("LOADvsCO",
"ADADvsLOAD", "ADADvsCO"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.5,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))


##########################################
##########################################
## Calculate overlap between BM44 Meta ##
##########################################
##########################################



GeneListMeta_condition_BM44_sigOnly <-
rownames(subset(both_condition_BM44[GeneListMeta_condition_BM44,], i_rawpval < 5e-6))
GeneListMeta_CDR_BM44_sigOnly <- rownames(subset(both_CDR_BM44[GeneListMeta_CDR_BM44,],
i_rawpval < 5e-6))
GeneListMeta_braak_BM44_sigOnly <-
rownames(subset(both_braak_BM44[GeneListMeta_braak_BM44,], i_rawpval < 5e-6))

length(Reduce(intersect, list(GeneListMeta_braak_BM44_sigOnly, GeneListMeta_CDR_BM44_sigOnly,
GeneListMeta_condition_BM44_sigOnly)))

length(Reduce(intersect, list(GeneListMeta_braak_BM44, GeneListMeta_CDR_BM44,
GeneListMeta_condition_BM44)))

#Total number:
unique(c(GeneListMeta_CDR_BM44, GeneListMeta_braak_BM44, GeneListMeta_condition_BM44))

grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.triple.venn(area1 = length(GeneListMeta_CDR_BM44), area2 =
length(GeneListMeta_braak_BM44), area3 = length(GeneListMeta_condition_BM44), n12 =
length(intersect(GeneListMeta_CDR_BM44, GeneListMeta_braak_BM44)), n23 =
length(intersect(GeneListMeta_braak_BM44, GeneListMeta_condition_BM44)), n13 =
length(intersect(GeneListMeta_CDR_BM44, GeneListMeta_condition_BM44)),
```

```
   n123 = length(Reduce(intersect, list(GeneListMeta_CDR_BM44, GeneListMeta_braak_BM44,
GeneListMeta_condition_BM44))), category = c("CDR", "Braak", "Case"), lty = "blank",
          cex = 1.5,
          fontface = "bold",
          fontfamily = "sans",
          cat.cex = 1.5,
          cat.fontface = "bold",
          cat.default.pos = "outer",
   fill = c("skyblue", "pink1", "mediumorchid"))


##BM44 Meta analyses
#Join the meta-analysis results together into a set

meta_condition_overlap <- list(GeneListMeta_condition_BM44, GeneListMeta_CDR_BM44,
GeneListMeta_braak_BM44)

length.gene.sets=sapply(meta_condition_overlap,length)

total=nrow(both_condition_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_condition_overlap[[1]], meta_condition_overlap[[2]],
meta_condition_overlap[[3]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


##########################################
##########################################
## Numbers for the paper ##
##########################################
##########################################

summary(res_parietal_CDR_filter)
summary(res_parietal_braak_filter)
summary(res_parietal_condition_filter)
```

```
res_parietal_CDR_filter["circHOMER1",]
res_parietal_CDR_filter["circCDR1-AS",]
res_parietal_condition_filter["circHOMER1",]
res_parietal_CDR_filter["circHOMER1",]

summary(res_BM44_CDR_filter)
summary(res_BM44_braak_filter)
summary(res_BM44_condition_filter)
summary(res_BM44_PlaqueMean_filter)

res_BM44_CDR_filter["circHOMER1",]
res_BM44_braak_filter["circHOMER1",]
res_BM44_condition_filter["circHOMER1",]
res_BM44_PlaqueMean_filter["circHOMER1",]


nrow(subset(both_CDR_BM44, i_adjpval < 0.05))
nrow(subset(both_CDR_BM44, i_rawpval < 5e-6))
nrow(subset(both_CDR_BM44, i_rawpval < 5e-8))


nrow(subset(both_braak_BM44, i_adjpval < 0.05))
nrow(subset(both_braak_BM44, i_rawpval < 5e-6))
nrow(subset(both_braak_BM44, i_rawpval < 5e-8))

nrow(subset(both_condition_BM44, i_adjpval < 0.05))
nrow(subset(both_condition_BM44, i_rawpval < 5e-6))
nrow(subset(both_condition_BM44, i_rawpval < 5e-8))


##########################################################


##########################################
##########################################
## Calculate Parietal and MSBB ALL Overlap Overall ##
##########################################
##########################################

##CDR

GeneList_MSBB_CDR <- Reduce(intersect, list(rownames(subset(res_BM10_CDR_filter, padj < 0.05)),
rownames(subset(res_BM22_CDR_filter, padj < 0.05)), rownames(subset(res_BM36_CDR_filter, padj <
0.05)), rownames(subset(res_BM44_CDR_filter, padj < 0.05))))
```

```
grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.pairwise.venn(area1 = length(GeneList_parietal_CDR), area2 = length(GeneList_MSBB_CDR),
cross.area = length(intersect(GeneList_parietal_CDR, GeneList_MSBB_CDR)), category = c("Parietal \n
Discovery", "All MSBB \n Replication"), lty = "blank", cex = 1.5, fontface = "bold", fontfamily = "sans",
cat.cex = 1.2, cat.fontface = "bold", cat.default.pos = "outer", fill = c("skyblue", "pink1"))

##braak

GeneList_MSBB_braak <- Reduce(intersect, list(rownames(subset(res_BM10_braak_filter, padj < 0.05)),
rownames(subset(res_BM22_braak_filter, padj < 0.05)), rownames(subset(res_BM36_braak_filter, padj
< 0.05)), rownames(subset(res_BM44_braak_filter, padj < 0.05))))

grid.newpage()
pushViewport(viewport(width=unit(0.8, "npc"), height = unit(0.8, "npc")))
draw.pairwise.venn(area1 = length(GeneList_parietal_braak), area2 = length(GeneList_MSBB_braak),
cross.area = length(intersect(GeneList_parietal_braak, GeneList_MSBB_braak)), category = c("Parietal \n
Discovery", "All MSBB \n Replication"), lty = "blank", cex = 1.5, fontface = "bold", fontfamily = "sans",
cat.cex = 1.2, cat.fontface = "bold", cat.default.pos = "outer", fill = c("skyblue", "pink1"))



##########################################
##########################################
## Phenotype Correlations ##
##########################################
##########################################

library(corrplot)

#Parietal Covar Processing
dds_parietal_covar_filter <- dds_parietal_final_CDR_filter
dds_parietal_covar_filter <- dds_parietal_covar_filter[ ,!is.na(dds_parietal_covar_filter$braak_final) ]

#BM10 Covar Processing
dds_BM10_covar_filter <- dds_BM10_final_condition_filter
dds_BM10_covar_filter <- dds_BM10_covar_filter[ ,!is.na(dds_BM10_covar_filter$bbscore) ]
dds_BM10_covar_filter <- dds_BM10_covar_filter[ ,!is.na(dds_BM10_covar_filter$CDR) ]
dds_BM10_covar_filter <- dds_BM10_covar_filter[ ,dds_BM10_covar_filter$NP.1 != "3" ]
dds_BM10_covar_filter <- dds_BM10_covar_filter[ ,dds_BM10_covar_filter$NP.1 != "4" ]
dds_BM10_covar_filter$NP.1 <- droplevels(dds_BM10_covar_filter$NP.1)

#BM22 Covar Processing
```

```
dds_BM22_covar_filter <- dds_BM22_final_condition_filter
dds_BM22_covar_filter <- dds_BM22_covar_filter[ ,!is.na(dds_BM22_covar_filter$bbscore) ]
dds_BM22_covar_filter <- dds_BM22_covar_filter[ ,!is.na(dds_BM22_covar_filter$CDR) ]
dds_BM22_covar_filter <- dds_BM22_covar_filter[ ,dds_BM22_covar_filter$NP.1 != "3" ]
dds_BM22_covar_filter <- dds_BM22_covar_filter[ ,dds_BM22_covar_filter$NP.1 != "4" ]
dds_BM22_covar_filter$NP.1 <- droplevels(dds_BM22_covar_filter$NP.1)

#BM36 Covar Processing
dds_BM36_covar_filter <- dds_BM36_final_condition_filter
dds_BM36_covar_filter <- dds_BM36_covar_filter[ ,!is.na(dds_BM36_covar_filter$bbscore) ]
dds_BM36_covar_filter <- dds_BM36_covar_filter[ ,!is.na(dds_BM36_covar_filter$CDR) ]
dds_BM36_covar_filter <- dds_BM36_covar_filter[ ,dds_BM36_covar_filter$NP.1 != "3" ]
dds_BM36_covar_filter <- dds_BM36_covar_filter[ ,dds_BM36_covar_filter$NP.1 != "4" ]
dds_BM36_covar_filter$NP.1 <- droplevels(dds_BM36_covar_filter$NP.1)

#BM44 Covar Processing
dds_BM44_covar_filter <- dds_BM44_final_condition_filter
dds_BM44_covar_filter <- dds_BM44_covar_filter[ ,!is.na(dds_BM44_covar_filter$bbscore) ]
dds_BM44_covar_filter <- dds_BM44_covar_filter[ ,!is.na(dds_BM44_covar_filter$CDR) ]
dds_BM44_covar_filter <- dds_BM44_covar_filter[ ,dds_BM44_covar_filter$NP.1 != "3" ]
dds_BM44_covar_filter <- dds_BM44_covar_filter[ ,dds_BM44_covar_filter$NP.1 != "4" ]
dds_BM44_covar_filter$NP.1 <- droplevels(dds_BM44_covar_filter$NP.1)


library(corrplot)

par(mfrow=c(2,3))
corr_BM22 <- cbind(CDR = dds_BM22_covar_filter$CDR, Braak = dds_BM22_covar_filter$bbscore,
AD_Case = as.integer(dds_BM22_covar_filter$NP.1))
corrplot::corrplot(cor(corr_BM22), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2, tl.cex = 1.2, cl.lim = c(0.6,1), cl.pos="n")
mtext("BM22", at=2.0, line=-0.5, cex=1.3)
corr_BM10 <- cbind(CDR = dds_BM10_covar_filter$CDR, Braak = dds_BM10_covar_filter$bbscore,
AD_Case = as.integer(dds_BM10_covar_filter$NP.1))
corrplot::corrplot(cor(corr_BM10), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2, tl.cex = 1.2, cl.lim = c(0.6,1), cl.pos="n")
mtext("BM10", at=2.0, line=-0.5, cex=1.3)
corr_BM36 <- cbind(CDR = dds_BM36_covar_filter$CDR, Braak = dds_BM36_covar_filter$bbscore,
AD_Case = as.integer(dds_BM36_covar_filter$NP.1))
corrplot::corrplot(cor(corr_BM36), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2, tl.cex = 1.2, cl.lim = c(0.6,1), cl.pos="n")
mtext("BM36", at=2.0, line=-0.5, cex=1.3)
corr_BM44 <- cbind(CDR = dds_BM44_covar_filter$CDR, Braak = dds_BM44_covar_filter$bbscore,
AD_Case = as.integer(dds_BM44_covar_filter$NP.1))
```

```
corrplot::corrplot(cor(corr_BM44), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2, tl.cex = 1.2, cl.lim = c(0.6,1), cl.pos="n")
mtext("BM44", at=2.0, line=-0.5, cex=1.3)
corr_parietal <- cbind(CDR = dds_parietal_covar_filter$CDR_e, Braak =
dds_parietal_covar_filter$braak_final, AD_Case = as.integer(dds_parietal_covar_filter$condition))
corrplot::corrplot(cor(corr_parietal), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2, tl.cex = 1.2, cl.lim = c(0.6,1), cl.pos="n")
mtext("PCtx", at=2.0, line=-0.5, cex=1.3)
par(mfrow=c(1,1))

##Corr Plots with Astrocyte and Neuron percentages

par(mfrow=c(2,3))
corr_BM22 <- cbind(CDR = dds_BM22_covar_filter$CDR, Braak = dds_BM22_covar_filter$bbscore,
Condition = as.integer(dds_BM22_covar_filter$NP.1), Neuron = dds_BM22_covar_filter$Neuron ,
Astrocyte = dds_BM22_covar_filter$Astrocyte, PlaqueMean = dds_BM22_covar_filter$PlaqueMean)
corrplot::corrplot(cor(corr_BM22), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2)
mtext("BM22", at=2.0, line=-0.5, cex=1)
corr_BM10 <- cbind(CDR = dds_BM10_covar_filter$CDR, Braak = dds_BM10_covar_filter$bbscore,
Condition = as.integer(dds_BM10_covar_filter$NP.1), Neuron = dds_BM10_covar_filter$Neuron ,
Astrocyte = dds_BM10_covar_filter$Astrocyte, PlaqueMean = dds_BM10_covar_filter$PlaqueMean)
corrplot::corrplot(cor(corr_BM10), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2)
mtext("BM10", at=2.0, line=-0.5, cex=1)
corr_BM36 <- cbind(CDR = dds_BM36_covar_filter$CDR, Braak = dds_BM36_covar_filter$bbscore,
Condition = as.integer(dds_BM36_covar_filter$NP.1), Neuron = dds_BM36_covar_filter$Neuron ,
Astrocyte = dds_BM36_covar_filter$Astrocyte, PlaqueMean = dds_BM36_covar_filter$PlaqueMean)
corrplot::corrplot(cor(corr_BM36), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2)
mtext("BM36", at=2.0, line=-0.5, cex=1)
corr_BM44 <- cbind(CDR = dds_BM44_covar_filter$CDR, Braak = dds_BM44_covar_filter$bbscore,
Condition = as.integer(dds_BM44_covar_filter$NP.1), Neuron = dds_BM44_covar_filter$Neuron ,
Astrocyte = dds_BM44_covar_filter$Astrocyte, PlaqueMean = dds_BM44_covar_filter$PlaqueMean)
corrplot::corrplot(cor(corr_BM44), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2)
mtext("BM44", at=2.0, line=-0.5, cex=1)
corr_parietal <- cbind(CDR = dds_parietal_covar_filter$CDR_e, Braak =
dds_parietal_covar_filter$braak_final, Condition = as.integer(dds_parietal_covar_filter$condition),
Neuron = dds_parietal_covar_filter$Neuron , Astrocyte = dds_parietal_covar_filter$Astrocyte)
corrplot::corrplot(cor(corr_parietal), method = "number", type = "lower", mar=c(0,0,2,0), number.cex =
1.2)
mtext("Parietal", at=2.0, line=-0.5, cex=1)
par(mfrow=c(1,1))
```

```
#############################################################
#############Final numbers for the paper#################
#############################################################

#parietal dataset participants
table(dds_parietal_condition_filter$condition)

#BM44 dataset participants
table(dds_BM44_condition_filter$condition)

#CDR
sum(table(dds_BM44_CDR_filter$condition))

#bbscore
sum(table(dds_BM44_braak_filter$condition))

#BM10 dataset participants
table(dds_BM10_condition_filter$condition)

#CDR
sum(table(dds_BM10_CDR_filter$condition))

#bbscore
sum(table(dds_BM10_braak_filter$condition))

#BM22 dataset participants
table(dds_BM22_condition_filter$condition)

#CDR
sum(table(dds_BM22_CDR_filter$condition))

#bbscore
sum(table(dds_BM22_braak_filter$condition))

#BM36 dataset participants
table(dds_BM36_condition_filter$condition)

#CDR
sum(table(dds_BM36_CDR_filter$condition))

#bbscore
```

```
sum(table(dds_BM36_braak_filter$condition))



#parietal - CDR
summary(res_parietal_CDR_filter)
res_parietal_CDR_filter["circHOMER1",]
res_parietal_CDR_filter["circCDR1-AS",]

#parietal - braak
summary(res_parietal_braak_filter)
res_parietal_braak_filter["circHOMER1",]


#parietal - condition
summary(res_parietal_condition_filter)
res_parietal_condition_filter["circHOMER1",]

#parietal - how many pass FDR all analyses
intersect(GeneList_parietal_condition,GeneList_parietal_CDR,GeneList_parietal_braak)

#parietal how many unique associations
unique(c(GeneList_parietal_condition,GeneList_parietal_CDR,GeneList_parietal_braak))

#Replication

#CDR

#Same direction of effect?

sum(sign(res_parietal_CDR_filter[rownames(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,],
pvalue < 0.05)),]$log2FoldChange) == sign(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,], pvalue
< 0.05)$log2FoldChange))

##Correlation of effect size
cor.test((res_parietal_CDR_filter[rownames(subset(res_BM44_CDR_filter[GeneList_parietal_CDR,],
pvalue < 0.05)),]$log2FoldChange), (subset(res_BM44_CDR_filter[GeneList_parietal_CDR,], pvalue <
0.05)$log2FoldChange))

#circHOMER1 example
res_BM44_CDR_filter["circHOMER1",]


##How many meta analysis
```

```
nrow(subset(both_CDR_BM44, i_adjpval < 0.05))

nrow(subset(both_CDR_BM44, i_rawpval < 5e-6))

both_CDR_BM44["circHOMER1",]
both_CDR_BM44["circCDR1-AS",]


#braak

#Same direction of effect?

sum(sign(res_parietal_braak_filter[rownames(subset(res_BM44_braak_filter[GeneList_parietal_braak,],
pvalue < 0.05)),]$log2FoldChange) == sign(subset(res_BM44_braak_filter[GeneList_parietal_braak,],
pvalue < 0.05)$log2FoldChange))

##Correlation of effect size
cor.test((res_parietal_braak_filter[rownames(subset(res_BM44_braak_filter[GeneList_parietal_braak,],
pvalue < 0.05)),]$log2FoldChange), (subset(res_BM44_braak_filter[GeneList_parietal_braak,], pvalue <
0.05)$log2FoldChange))


##How many meta analysis
nrow(subset(both_braak_BM44, i_adjpval < 0.05))

nrow(subset(both_braak_BM44, i_rawpval < 5e-6))


#condition

#Same direction of effect?

sum(sign(res_parietal_condition_filter[rownames(subset(res_BM44_condition_filter[GeneList_parietal_
condition,], pvalue < 0.05)),]$log2FoldChange) ==
sign(subset(res_BM44_condition_filter[GeneList_parietal_condition,], pvalue < 0.05)$log2FoldChange))

##Correlation of effect size
cor.test((res_parietal_condition_filter[rownames(subset(res_BM44_condition_filter[GeneList_parietal_c
ondition,], pvalue < 0.05)),]$log2FoldChange),
(subset(res_BM44_condition_filter[GeneList_parietal_condition,], pvalue < 0.05)$log2FoldChange))

#circHOMER1 example
res_BM44_condition_filter["circHOMER1",]
```

```
##How many meta analysis
nrow(subset(both_condition_BM44, i_adjpval < 0.05))

nrow(subset(both_condition_BM44, i_rawpval < 5e-6))

##Overall meta
length(unique(c(GeneListMeta_CDR_BM44,GeneListMeta_braak_BM44,GeneListMeta_condition_BM44
)))

##How many in all of them
length(Reduce(intersect,
list(GeneListMeta_CDR_BM44,GeneListMeta_braak_BM44,GeneListMeta_condition_BM44)))

Reduce(intersect,
list(GeneListMeta_CDR_BM44,GeneListMeta_braak_BM44,GeneListMeta_condition_BM44))

#How many in all of them are sig
length(Reduce(intersect,
list(GeneListMeta_CDR_BM44_sigOnly,GeneListMeta_braak_BM44_sigOnly,GeneListMeta_condition_B
M44_sigOnly)))

Reduce(intersect,
list(GeneListMeta_CDR_BM44_sigOnly,GeneListMeta_braak_BM44_sigOnly,GeneListMeta_condition_B
M44_sigOnly))



###Meta-analysis overlap

#CDR
length(Reduce(intersect,
list(GeneListMeta_CDR_BM44,GeneListMeta_CDR_BM10,GeneListMeta_CDR_BM22,GeneListMeta_CDR
_BM36)))

#Join the meta-analysis results together into a set

meta_CDR_overlap <- list(GeneListMeta_CDR_BM10, GeneListMeta_CDR_BM22,
GeneListMeta_CDR_BM36, GeneListMeta_CDR_BM44)

length.gene.sets=sapply(meta_CDR_overlap,length)

total=nrow(both_CDR_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))
```

```
(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_CDR_overlap[[1]], meta_CDR_overlap[[2]], meta_CDR_overlap[[3]],
 meta_CDR_overlap[[4]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


#braak
length(Reduce(intersect,
list(GeneListMeta_braak_BM44,GeneListMeta_braak_BM10,GeneListMeta_braak_BM22,GeneListMeta
_braak_BM36)))

#Join the meta-analysis results together into a set

meta_braak_overlap <- list(GeneListMeta_braak_BM10, GeneListMeta_braak_BM22,
GeneListMeta_braak_BM36, GeneListMeta_braak_BM44)

length.gene.sets=sapply(meta_braak_overlap,length)

total=nrow(both_braak_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_braak_overlap[[1]], meta_braak_overlap[[2]], meta_braak_overlap[[3]],
 meta_braak_overlap[[4]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


#condition
length(Reduce(intersect,
list(GeneListMeta_condition_BM44,GeneListMeta_condition_BM10,GeneListMeta_condition_BM22,Ge
neListMeta_condition_BM36)))
```

```
#Join the meta-analysis results together into a set

meta_condition_overlap <- list(GeneListMeta_condition_BM10, GeneListMeta_condition_BM22,
GeneListMeta_condition_BM36, GeneListMeta_condition_BM44)

length.gene.sets=sapply(meta_condition_overlap,length)

total=nrow(both_condition_BM44)

(num.expcted.overlap=total*do.call(prod,as.list(length.gene.sets/total)))

(p=sapply(0:min(length.gene.sets),function(i) dpsets(i, length.gene.sets, n=total)))

common.genes=intersect(meta_condition_overlap[[1]], meta_condition_overlap[[2]],
meta_condition_overlap[[3]],
 meta_condition_overlap[[4]])
(num.observed.overlap=length(common.genes))

(FE=num.observed.overlap/num.expcted.overlap)

dpsets(num.observed.overlap, length.gene.sets, n=total)


###Cross-meta-analysis

GeneListMeta_CDR_MSBB <- Reduce(intersect, list(GeneListMeta_CDR_BM10,
GeneListMeta_CDR_BM22, GeneListMeta_CDR_BM36, GeneListMeta_CDR_BM44))

GeneListMeta_braak_MSBB <- Reduce(intersect, list(GeneListMeta_braak_BM10,
GeneListMeta_braak_BM22, GeneListMeta_braak_BM36, GeneListMeta_braak_BM44))

GeneListMeta_condition_MSBB <- Reduce(intersect, list(GeneListMeta_condition_BM10,
GeneListMeta_condition_BM22, GeneListMeta_condition_BM36, GeneListMeta_condition_BM44))

#all four meta-analyses, all three phenotypes
Reduce(intersect,list(GeneListMeta_CDR_MSBB,GeneListMeta_braak_MSBB,GeneListMeta_condition_
MSBB))

#all four meta-analyses, two quantitative
Reduce(intersect,list(GeneListMeta_CDR_MSBB,GeneListMeta_braak_MSBB))

##Pre-symptomatic
View(bootcorr_table_final)
```

```
##Number of presympAD - parietal
table(colData(dds_parietal_PreSymp_filter)$condition)

##Number of presympAD - BM44
table(colData(dds_BM44_PreSymp_filter)$NP.1)

##Number of nopresympAD - BM44
table(colData(dds_BM44_conditionNoPre_filter)$NP.1)

##Number of nopresympAD - parietal
table(colData(dds_parietal_conditionNoPre_filter)$condition)

##Number of background circs - parietal
nrow(Presymptomatic_parietal_notsig)

##Number of sig circs - parietal
nrow(Presymptomatic_parietal_sig)


ks.test(bootcorr_presymptom_parietal_notsig$t,bootcorr_presymptom_parietal_sig$t, alternative =
"greater")$p.value
ks.test(bootcorr_presymptom_BM44_notsig$t,bootcorr_presymptom_BM44_sig$t, alternative =
"greater")$p.value



##ADAD
summary(res_parietalADAD_condition_filter_ADADvsCO)

summary(summary(res_parietalADAD_condition_filter_braak6_ADADvsLOAD)

#Overlap
#same direction?
sum(OverlapADAD_CO_ADAD_LOAD[,3])

#Greater effect
sum(OverlapADAD_CO_ADAD_LOAD[,4])

##PropVar


summary(PropVar_parietal_CDR_Neuron_relaimpo)

##circHOMER1 correlation
```

```
parietal_circHOMER1_corr_otherTop <- t(norm_parietal_CDR_counts[sigList_parietal_CDR_top,])
cor(parietal_circHOMER1_corr_otherTop)
summary(cor(parietal_circHOMER1_corr_otherTop)[1,2:ncol(parietal_circHOMER1_corr_otherTop)])

sum(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo),])
summary(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_parietal_CDR_Neuron_relaimpo),])

summary(PropVar_BM44_CDR_Neuron_relaimpo)

sum(PropVar_fullmodel_BM44_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_BM44_CDR_Neuron_relaimpo),])
summary(PropVar_fullmodel_BM44_CDR_Neuron_relaimpo[3:nrow(PropVar_fullmodel_BM44_CDR_Neuron_relaimpo),])

#MeanPlaque

BM44
length(GeneList_BM44_PlaqueMean)
res_BM44_PlaqueMean_filter["circHOMER1",]

length(colData(dds_BM44_PlaqueMean_filter)$NP.1)

##Sample size in all plaque mean analyses
length(colData(dds_BM10_PlaqueMean_filter)$NP.1)
length(colData(dds_BM22_PlaqueMean_filter)$NP.1)
length(colData(dds_BM36_PlaqueMean_filter)$NP.1)
length(colData(dds_BM44_PlaqueMean_filter)$NP.1)


#How many in all 4 analyses

length(Reduce(intersect,list(GeneList_BM10_PlaqueMean,GeneList_BM22_PlaqueMean,GeneList_BM36_PlaqueMean,GeneList_BM44_PlaqueMean)))

##Number of individuals in the ROC analyses - parietal
table(coldata_parietal_normcounts$condition)

##Number of individuals in the ROC analyses - BM10
table(coldata_BM10_normcounts$condition)

##Number of individuals in the ROC analyses - BM22
```

```
table(coldata_BM22_normcounts$condition)

##Number of individuals in the ROC analyses - BM36
table(coldata_BM36_normcounts$condition)

##Number of individuals in the ROC analyses - BM44
table(coldata_BM44_normcounts$condition)

##AUC Values - parietal
ggroc(g_base_parietal, TITLE="Parietal Base")
ggroc(g_circ_parietal, TITLE="Parietal Circ")
ggroc(g_base_circ_parietal, TITLE="Parietal Base+Circ")

##AUC Values - BM44
ggroc(g_base_BM44, TITLE="BM44 Base")
ggroc(g_circ_BM44, TITLE="BM44 Circ")
ggroc(g_base_circ_BM44, TITLE="BM44 Base+Circ")



##MEGENA Numbers
length(unique(as.character(META_MEGENA_parietalCDR_sigOnly$Module)))

write.table(GeneListMeta_CDR_BM44,
"/home/dubeu/Synapse/NetworkAnalysis/GeneListMeta_CDR_BM44.txt")
```