Upstream Analysis of Single-soma transcriptomics of tangle-bearing neurons in Alzheimer's disease - Inhibitory

Gautam Subedi

2025-01-02

STEPS INCLUDES

1. Loading the required packages and dataset
2. Exploratory Data Analysis and pre-processing: i). mitochondiral percentage cutoffs ii).nFeature_RNA cutoffs iii).nCount_RNA cutoffs 3). Data Normalization 4). Finding Variables Features 5). Data Scaling 6). Linear Dimensionality Reduction : PCA 7). Non-linear Dimensionality Redcution: UMAP 8). Finding Neighbours and Clusters based on condition (Alzheimer's disease Vs Normal) 9). Comparison of cluster with removal of batch effects

Link to dataset: https://cellxgene.cziscience.com/e/9813a1d4-d107-459e-9b2e-7687be935f69.cxg/

##. Loading required library and dataset

```
setwd("/Users/lord/singleCell")
list.files(all.files = TRUE, full.names = TRUE)
```

```
##  [1] "./."                              "./.."
##  [3] "./.Rapp.history"                  "./.RData"
##  [5] "./.RDataTmp"                      "./.Rhistory"
##  [7] "./Alzheimer.rds"                  "./Cell.R"
##  [9] "./DEA_Alzheimer.R"                "./DEGs_results.csv"
## [11] "./Inhibitory_Alzheimer.rds"       "./local.rds"
## [13] "./Mean-variance trend.jpeg"       "./old.rds"
## [15] "./RAPI.R"                         "./Rplot.jpeg"
## [17] "./Rplot.png"                      "./ScSoma_Alzhemeir.rds"
## [19] "./Single_Cell_RNA_seq_Analysis.R" "./singlecell_Analysis.R"
## [21] "./voom.jpeg"                      "./Working_seurat.R"
```

```
library(Seurat)
```

```
## Loading required package: SeuratObject
```

```
## Loading required package: sp
```

```
## 'SeuratObject' was built with package 'Matrix' 1.6.3 but the current
## version is 1.6.5; it is recomended that you reinstall 'SeuratObject' as
## the ABI for 'Matrix' may have changed
```

```
##
## Attaching package: 'SeuratObject'

## The following object is masked from 'package:base':
##
##     intersect
```

```
library(patchwork)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.4.4     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(harmony)
```

```
## Loading required package: Rcpp
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggplot2)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
##
## The following object is masked from 'package:patchwork':
##
##     align_plots
```

```
file <- readRDS("ScSoma_Alzhemeir.rds")
```

Head view of data and adding mitochondiral percentage feature in the seurat dataset for later subsetting data

```
##                         nCount_RNA nFeature_RNA percent.mt SORT Amyloid Age
## C0001_AAACGGGCAGTACACT-1        927          694  1.8338727 MAP2      C3  73
## C0001_AAACGGGGTACCGTAT-1       1251          886  6.4748201 MAP2      C3  73
## C0001_AAACGGGTCCAGAGGA-1       2972         1695  0.2018843 MAP2      C3  73
## C0001_AAACGGGTCCTTCAAT-1       3619         1875  2.4592429 MAP2      C3  73
## C0001_AAAGATGGTCAACATC-1       3931         2247  2.4675655 MAP2      C3  73
## C0001_AAAGATGGTGCTTCTC-1       1523         1001  3.2829941 MAP2      C3  73
## C0001_AAAGTAGAGACCTAGG-1       2089         1360  0.7659167 MAP2      C3  73
## C0001_AAAGTAGGTATAGGTA-1       1214          785 10.7907743 MAP2      C3  73
## C0001_AAATGCCGTCGGGTCT-1        777          599  6.3063063 MAP2      C3  73
## C0001_AAATGCCGTGTCCTCT-1       1155          783  4.9350649 MAP2      C3  73
##                         RIN nCount_SCT nFeature_SCT nCount_Exon nFeature_Exon
## C0001_AAACGGGCAGTACACT-1 5.7       2092          787         344           296
## C0001_AAACGGGGTACCGTAT-1 5.7       2440          915         598           463
## C0001_AAACGGGTCCAGAGGA-1 5.7       2949         1695         968           725
## C0001_AAACGGGTCCTTCAAT-1 5.7       3269         1873        1442           920
## C0001_AAAGATGGTCAACATC-1 5.7       3441         2246        1753          1262
## C0001_AAAGATGGTGCTTCTC-1 5.7       2238         1002         664           494
## C0001_AAAGTAGAGACCTAGG-1 5.7       2343         1359         745           619
## C0001_AAAGTAGGTATAGGTA-1 5.7       2383          821         616           448
## C0001_AAATGCCGTCGGGTCT-1 5.7       1948          746         382           310
## C0001_AAATGCCGTGTCCTCT-1 5.7       2398          813         550           383
##                         PMI Braak Sample.ID                Cell.Types
## C0001_AAACGGGCAGTACACT-1  13    VI AD-MAP2-1           In7_ADARB2-CALB2
## C0001_AAACGGGGTACCGTAT-1  13    VI AD-MAP2-1           In6_ADARB2-LAMP5
## C0001_AAACGGGTCCAGAGGA-1  13    VI AD-MAP2-1             In1_LHX6-PVALB
## C0001_AAACGGGTCCTTCAAT-1  13    VI AD-MAP2-1           In6_ADARB2-LAMP5
## C0001_AAAGATGGTCAACATC-1  13    VI AD-MAP2-1           In6_ADARB2-LAMP5
## C0001_AAAGATGGTGCTTCTC-1  13    VI AD-MAP2-1             In1_LHX6-PVALB
## C0001_AAAGTAGAGACCTAGG-1  13    VI AD-MAP2-1      In5_LHX6-ADARB2-LAMP5
## C0001_AAAGTAGGTATAGGTA-1  13    VI AD-MAP2-1                In3_LHX6-SST
## C0001_AAATGCCGTCGGGTCT-1  13    VI AD-MAP2-1           In7_ADARB2-CALB2
## C0001_AAATGCCGTGTCCTCT-1  13    VI AD-MAP2-1 In2_LHX6-PVALB-Chandelier
```

```
##                           tissue_ontology_term_id assay_ontology_term_id
## C0001_AAACGGGCAGTACACT-1          UBERON:0000451            EFO:0009899
## C0001_AAACGGGGTACCGTAT-1          UBERON:0000451            EFO:0009899
## C0001_AAACGGGTCCAGAGGA-1          UBERON:0000451            EFO:0009899
## C0001_AAACGGGTCCTTCAAT-1          UBERON:0000451            EFO:0009899
## C0001_AAAGATGGTCAACATC-1          UBERON:0000451            EFO:0009899
## C0001_AAAGATGGTGCTTCTC-1          UBERON:0000451            EFO:0009899
## C0001_AAAGTAGAGACCTAGG-1          UBERON:0000451            EFO:0009899
## C0001_AAAGTAGGTATAGGTA-1          UBERON:0000451            EFO:0009899
## C0001_AAATGCCGTCGGGTCT-1          UBERON:0000451            EFO:0009899
## C0001_AAATGCCGTGTCCTCT-1          UBERON:0000451            EFO:0009899
##                           disease_ontology_term_id cell_type_ontology_term_id
## C0001_AAACGGGCAGTACACT-1            MONDO:0004975                 CL:0000498
## C0001_AAACGGGGTACCGTAT-1            MONDO:0004975                 CL:0000498
## C0001_AAACGGGTCCAGAGGA-1            MONDO:0004975                 CL:0000498
## C0001_AAACGGGTCCTTCAAT-1            MONDO:0004975                 CL:0000498
## C0001_AAAGATGGTCAACATC-1            MONDO:0004975                 CL:0000498
## C0001_AAAGATGGTGCTTCTC-1            MONDO:0004975                 CL:0000498
## C0001_AAAGTAGAGACCTAGG-1            MONDO:0004975                 CL:0000498
## C0001_AAAGTAGGTATAGGTA-1            MONDO:0004975                 CL:0000498
## C0001_AAATGCCGTCGGGTCT-1            MONDO:0004975                 CL:0000498
## C0001_AAATGCCGTGTCCTCT-1            MONDO:0004975                 CL:0000498
##                           development_stage_ontology_term_id
## C0001_AAACGGGCAGTACACT-1                      HsapDv:0000167
## C0001_AAACGGGGTACCGTAT-1                      HsapDv:0000167
## C0001_AAACGGGTCCAGAGGA-1                      HsapDv:0000167
## C0001_AAACGGGTCCTTCAAT-1                      HsapDv:0000167
## C0001_AAAGATGGTCAACATC-1                      HsapDv:0000167
## C0001_AAAGATGGTGCTTCTC-1                      HsapDv:0000167
## C0001_AAAGTAGAGACCTAGG-1                      HsapDv:0000167
## C0001_AAAGTAGGTATAGGTA-1                      HsapDv:0000167
## C0001_AAATGCCGTCGGGTCT-1                      HsapDv:0000167
## C0001_AAATGCCGTGTCCTCT-1                      HsapDv:0000167
##                           self_reported_ethnicity_ontology_term_id
## C0001_AAACGGGCAGTACACT-1                                  unknown
## C0001_AAACGGGGTACCGTAT-1                                  unknown
## C0001_AAACGGGTCCAGAGGA-1                                  unknown
## C0001_AAACGGGTCCTTCAAT-1                                  unknown
## C0001_AAAGATGGTCAACATC-1                                  unknown
## C0001_AAAGATGGTGCTTCTC-1                                  unknown
## C0001_AAAGTAGAGACCTAGG-1                                  unknown
## C0001_AAAGTAGGTATAGGTA-1                                  unknown
## C0001_AAATGCCGTCGGGTCT-1                                  unknown
## C0001_AAATGCCGTGTCCTCT-1                                  unknown
##                           sex_ontology_term_id is_primary_data
## C0001_AAACGGGCAGTACACT-1          PATO:0000383            TRUE
## C0001_AAACGGGGTACCGTAT-1          PATO:0000383            TRUE
## C0001_AAACGGGTCCAGAGGA-1          PATO:0000383            TRUE
## C0001_AAACGGGTCCTTCAAT-1          PATO:0000383            TRUE
## C0001_AAAGATGGTCAACATC-1          PATO:0000383            TRUE
## C0001_AAAGATGGTGCTTCTC-1          PATO:0000383            TRUE
## C0001_AAAGTAGAGACCTAGG-1          PATO:0000383            TRUE
## C0001_AAAGTAGGTATAGGTA-1          PATO:0000383            TRUE
## C0001_AAATGCCGTCGGGTCT-1          PATO:0000383            TRUE
```

```
## C0001_AAATGCCGTGTCCTCT-1         PATO:0000383            TRUE
##                           organism_ontology_term_id donor_id suspension_type
## C0001_AAACGGGCAGTACACT-1            NCBITaxon:9606 Subject6            cell
## C0001_AAACGGGGTACCGTAT-1            NCBITaxon:9606 Subject6            cell
## C0001_AAACGGGTCCAGAGGA-1            NCBITaxon:9606 Subject6            cell
## C0001_AAACGGGTCCTTCAAT-1            NCBITaxon:9606 Subject6            cell
## C0001_AAAGATGGTCAACATC-1            NCBITaxon:9606 Subject6            cell
## C0001_AAAGATGGTGCTTCTC-1            NCBITaxon:9606 Subject6            cell
## C0001_AAAGTAGAGACCTAGG-1            NCBITaxon:9606 Subject6            cell
## C0001_AAAGTAGGTATAGGTA-1            NCBITaxon:9606 Subject6            cell
## C0001_AAATGCCGTCGGGTCT-1            NCBITaxon:9606 Subject6            cell
## C0001_AAATGCCGTGTCCTCT-1            NCBITaxon:9606 Subject6            cell
##                           tissue_type              cell_type     assay
## C0001_AAACGGGCAGTACACT-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAACGGGGTACCGTAT-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAACGGGTCCAGAGGA-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAACGGGTCCTTCAAT-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAAGATGGTCAACATC-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAAGATGGTGCTTCTC-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAAGTAGAGACCTAGG-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAAGTAGGTATAGGTA-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAATGCCGTCGGGTCT-1      tissue inhibitory interneuron 10x 3' v2
## C0001_AAATGCCGTGTCCTCT-1      tissue inhibitory interneuron 10x 3' v2
##                                      disease     organism    sex
## C0001_AAACGGGCAGTACACT-1 Alzheimer disease Homo sapiens female
## C0001_AAACGGGGTACCGTAT-1 Alzheimer disease Homo sapiens female
## C0001_AAACGGGTCCAGAGGA-1 Alzheimer disease Homo sapiens female
## C0001_AAACGGGTCCTTCAAT-1 Alzheimer disease Homo sapiens female
## C0001_AAAGATGGTCAACATC-1 Alzheimer disease Homo sapiens female
## C0001_AAAGATGGTGCTTCTC-1 Alzheimer disease Homo sapiens female
## C0001_AAAGTAGAGACCTAGG-1 Alzheimer disease Homo sapiens female
## C0001_AAAGTAGGTATAGGTA-1 Alzheimer disease Homo sapiens female
## C0001_AAATGCCGTCGGGTCT-1 Alzheimer disease Homo sapiens female
## C0001_AAATGCCGTGTCCTCT-1 Alzheimer disease Homo sapiens female
##                                       tissue self_reported_ethnicity
## C0001_AAACGGGCAGTACACT-1 prefrontal cortex                 unknown
## C0001_AAACGGGGTACCGTAT-1 prefrontal cortex                 unknown
## C0001_AAACGGGTCCAGAGGA-1 prefrontal cortex                 unknown
## C0001_AAACGGGTCCTTCAAT-1 prefrontal cortex                 unknown
## C0001_AAAGATGGTCAACATC-1 prefrontal cortex                 unknown
## C0001_AAAGATGGTGCTTCTC-1 prefrontal cortex                 unknown
## C0001_AAAGTAGAGACCTAGG-1 prefrontal cortex                 unknown
## C0001_AAAGTAGGTATAGGTA-1 prefrontal cortex                 unknown
## C0001_AAATGCCGTCGGGTCT-1 prefrontal cortex                 unknown
## C0001_AAATGCCGTGTCCTCT-1 prefrontal cortex                 unknown
##                           development_stage observation_joinid
## C0001_AAACGGGCAGTACACT-1 73-year-old stage         5$tsN89o-C
## C0001_AAACGGGGTACCGTAT-1 73-year-old stage         i+HeMEM)4<
## C0001_AAACGGGTCCAGAGGA-1 73-year-old stage         oi@)SZWo%-
## C0001_AAACGGGTCCTTCAAT-1 73-year-old stage         V#;BPg@9;|
## C0001_AAAGATGGTCAACATC-1 73-year-old stage         dkOgL6!4f@
## C0001_AAAGATGGTGCTTCTC-1 73-year-old stage         ^?T+UY#N82
## C0001_AAAGTAGAGACCTAGG-1 73-year-old stage         (eu`a*%Ov_
## C0001_AAAGTAGGTATAGGTA-1 73-year-old stage         MU())5A+{Z
```

```
## C0001_AAATGCCGTCGGGTCT-1 73-year-old stage          &|OzZf|?o=
## C0001_AAATGCCGTGTCCTCT-1 73-year-old stage          eUFi!dg‘MW
```

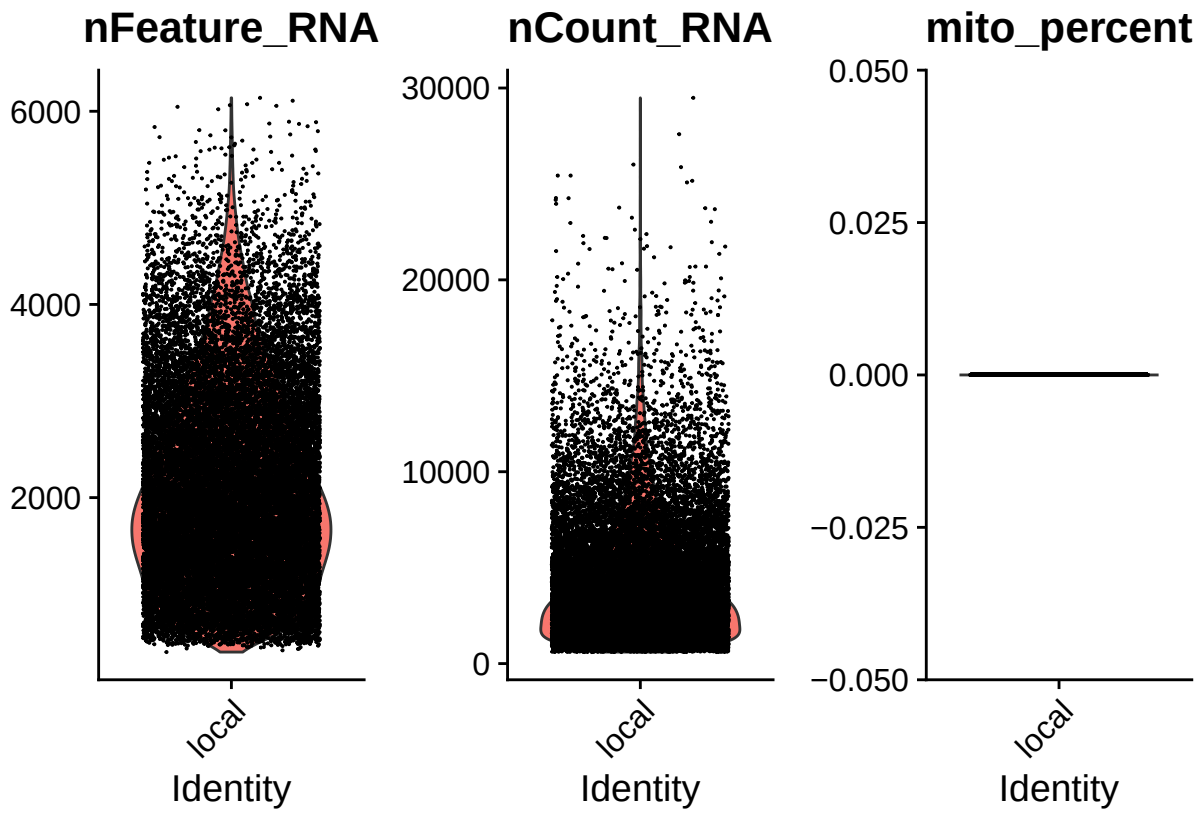
##. Extracting row names into new column called sample and separating sample column into patient and barcode for identifying clusters based on patient

```
file$sample <- rownames(file@meta.data)
file@meta.data <- separate(file@meta.data, col = 'sample', into = c('Patient', 'Barcode'),sep = '_')
```

Violin plot to identfy the variation in the dataset with three features

```
VlnPlot(file, features = c("nFeature_RNA", "nCount_RNA", "mito_percent"), ncol = 3)
```

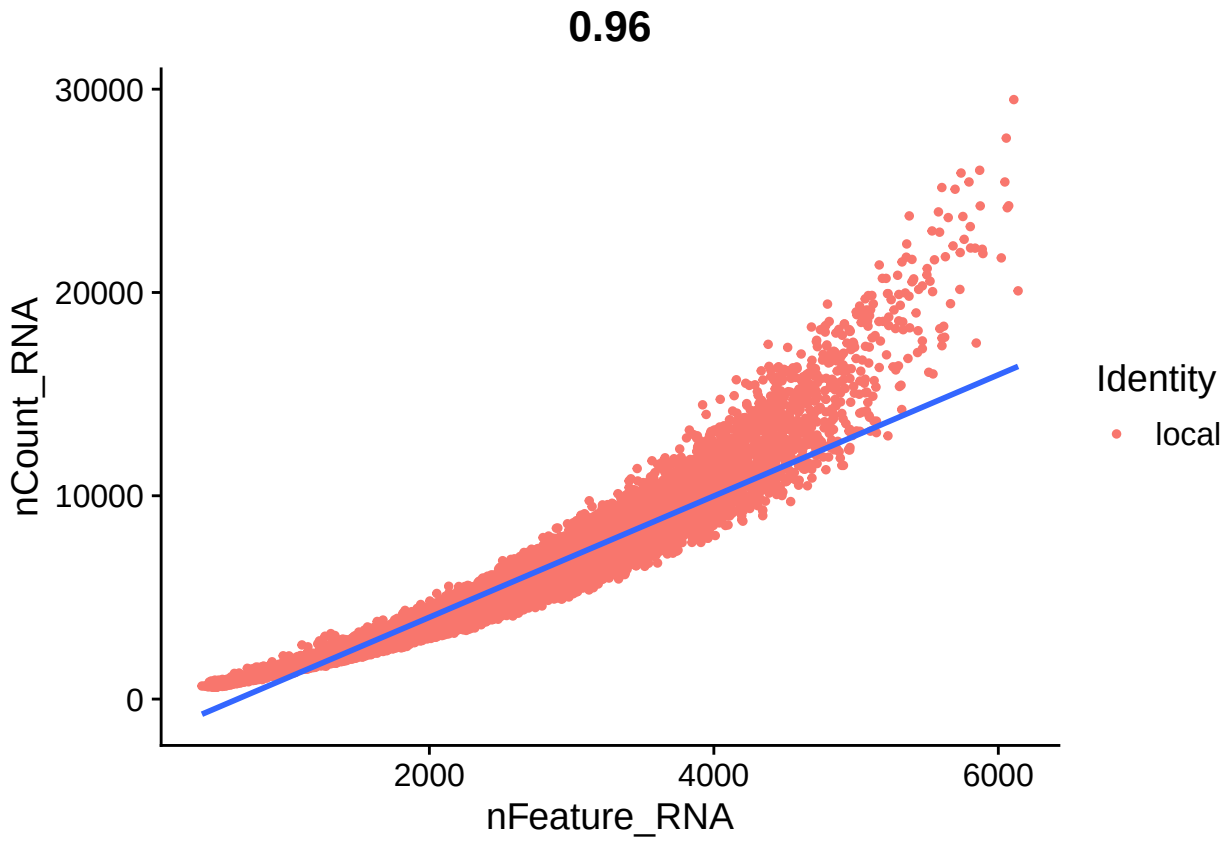
```
## Warning in SingleExIPlot(type = type, data = data[, x, drop = FALSE], idents =
## idents, : All cells have the same value of mito_percent.
```



Scatter plot to visualize and identify the cutoff margins of the feature and counts

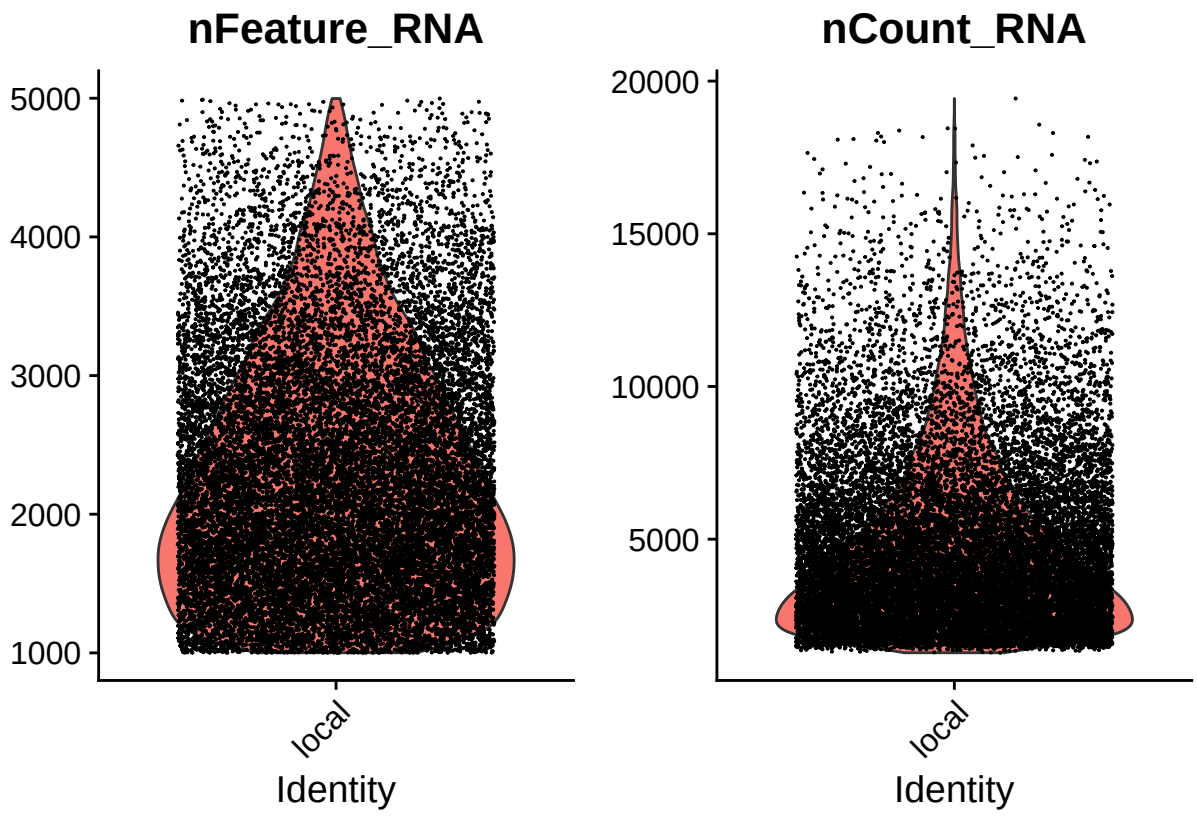
```
FeatureScatter(file, feature1 = "nFeature_RNA" , feature2 = "nCount_RNA") + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



##. Violin plot after data cleaning with cutoffs

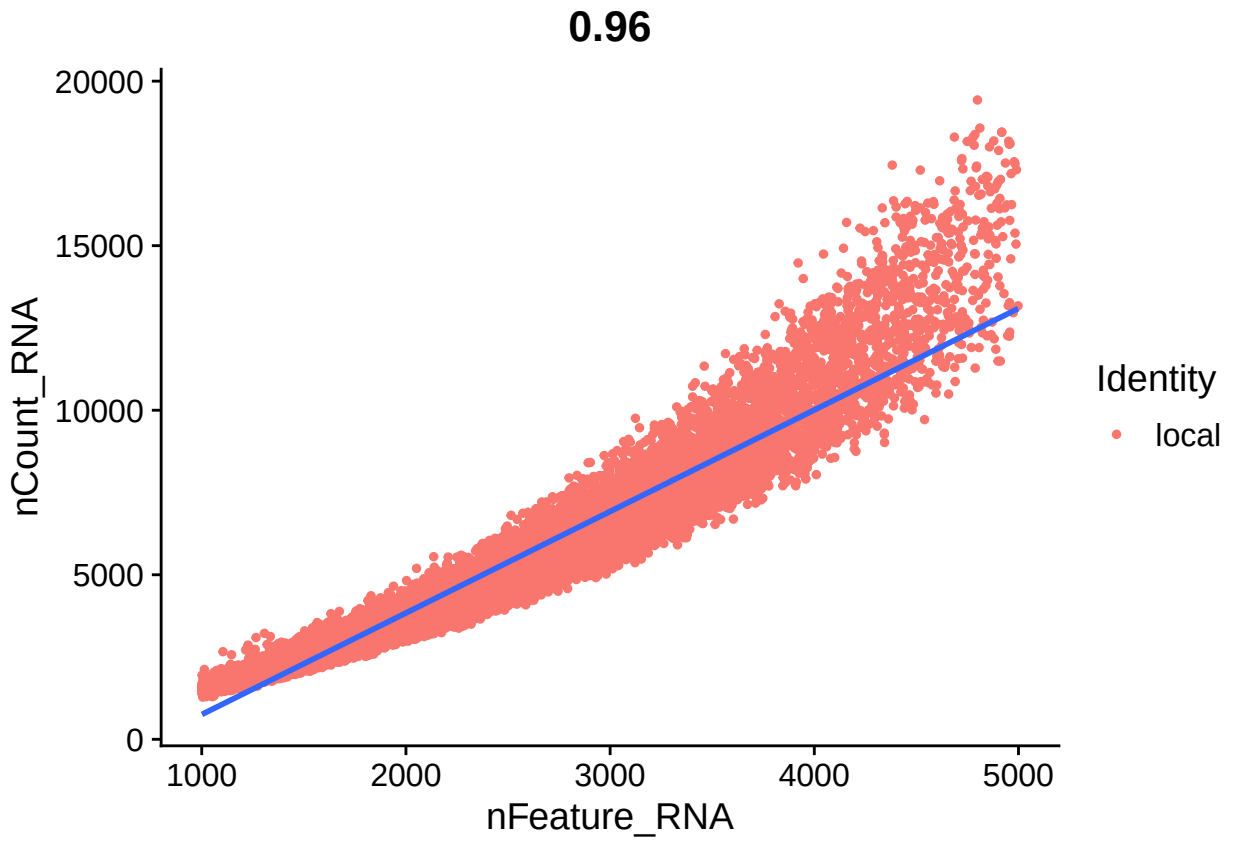
```
file_subset <- subset(file, subset = nFeature_RNA > 1000 & nFeature_RNA <5000 &nCount_RNA > 800)
VlnPlot(file_subset, features = c("nFeature_RNA" , "nCount_RNA"), ncol = 2)
```



##. Scatter Plot after data cleaning to check the fit of the line

```
FeatureScatter(file_subset, feature1 ="nFeature_RNA" , feature2 = "nCount_RNA") + geom_smooth(method =
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



##. Log Normalization

```
file_subset <- NormalizeData(file_subset, normalization.method = "LogNormalize" , scale.factor = 10000)
```

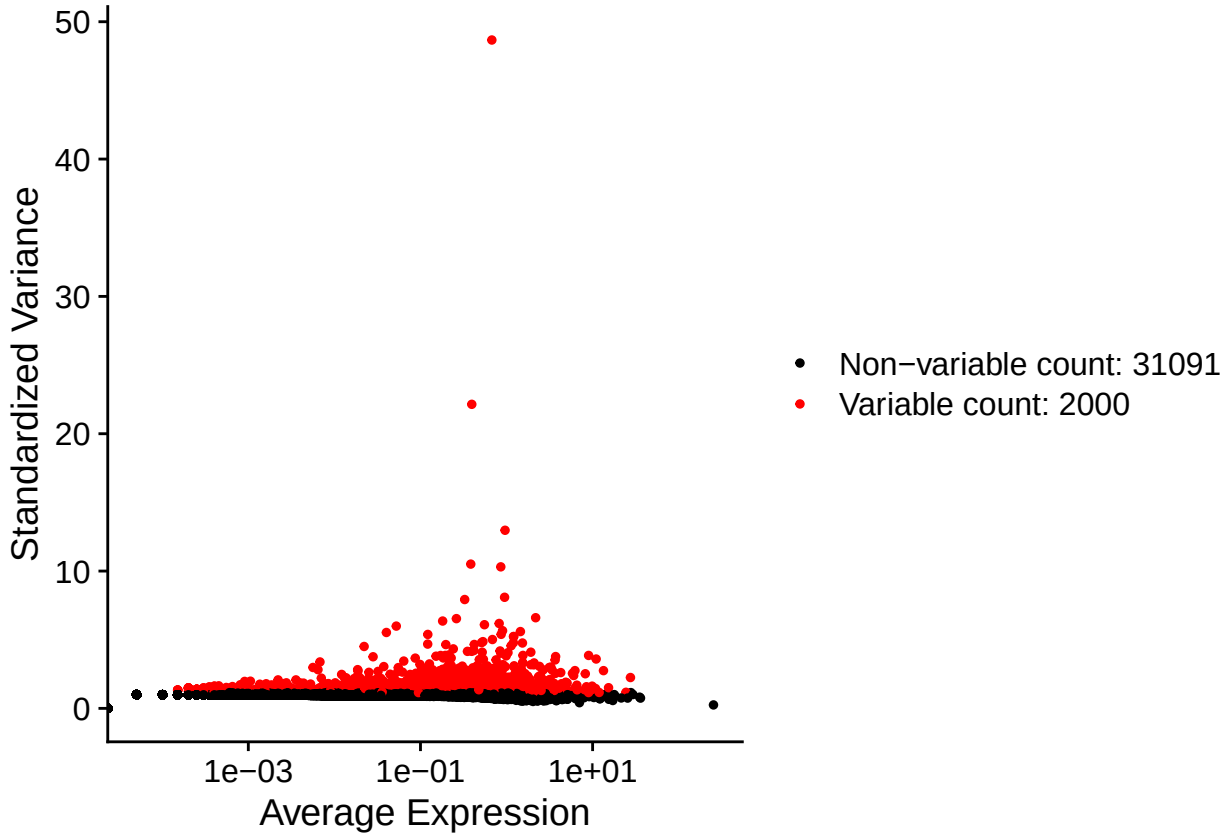
##. Finding varibale features from a normalized data

```
file_subset <- FindVariableFeatures(file_subset, selection.method = "vst", nfeatures = 2000)
```

Identification of top 10 highly vairbale features and variable feature plot

```
top10 <- head(VariableFeatures(file_subset), 10)

# plot variable features with and without labels
plot1 <- VariableFeaturePlot(file_subset)
plot1
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

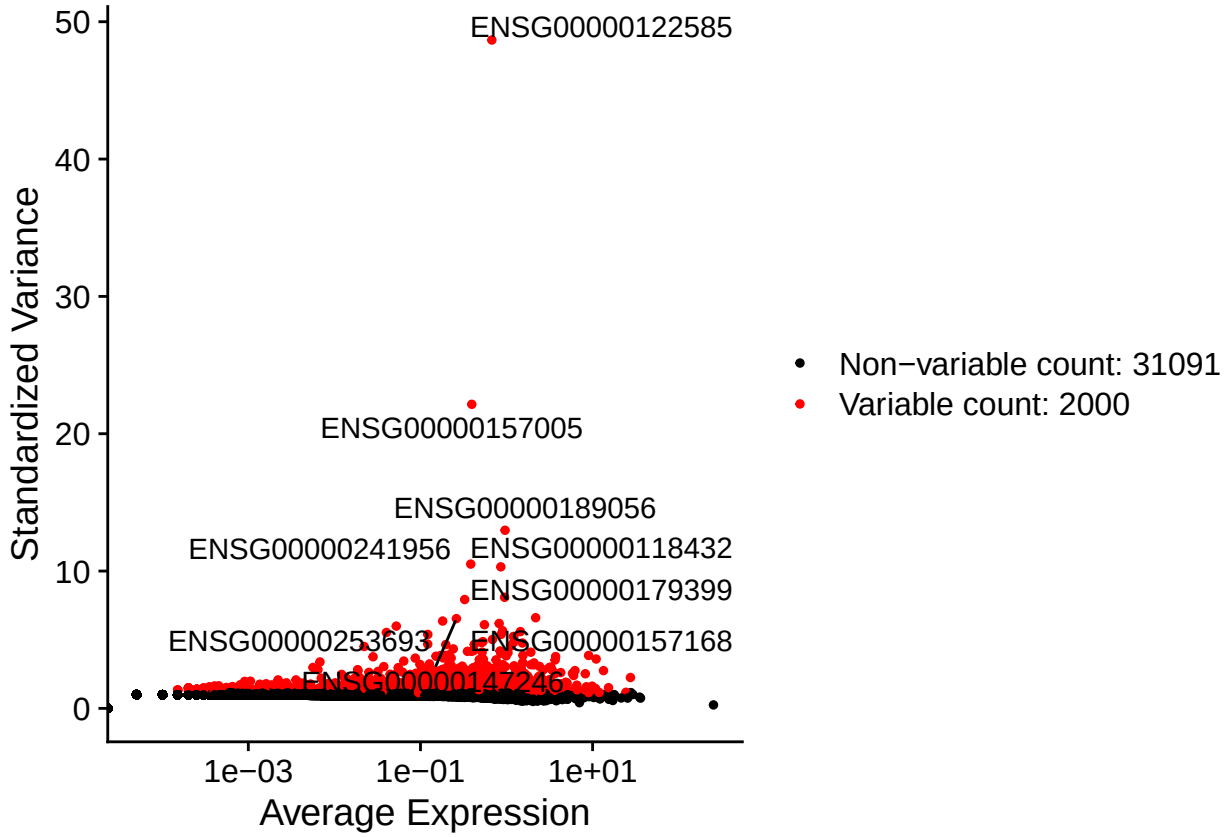


##. Variable feature plot with gene(features) name

```
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE,  xnudge = 0, ynudge = 0)
plot2
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



##. Scaling the dataset to have mean as 0 and standard deviation as 1, so that there will be no effect of highly varibale genes into downstream analysis

```
all.genes <- rownames(file_subset)
file_subset <- ScaleData(file_subset, features = all.genes)
```

```
## Centering and scaling data matrix
```

Linear dimensionality reduction using PCA

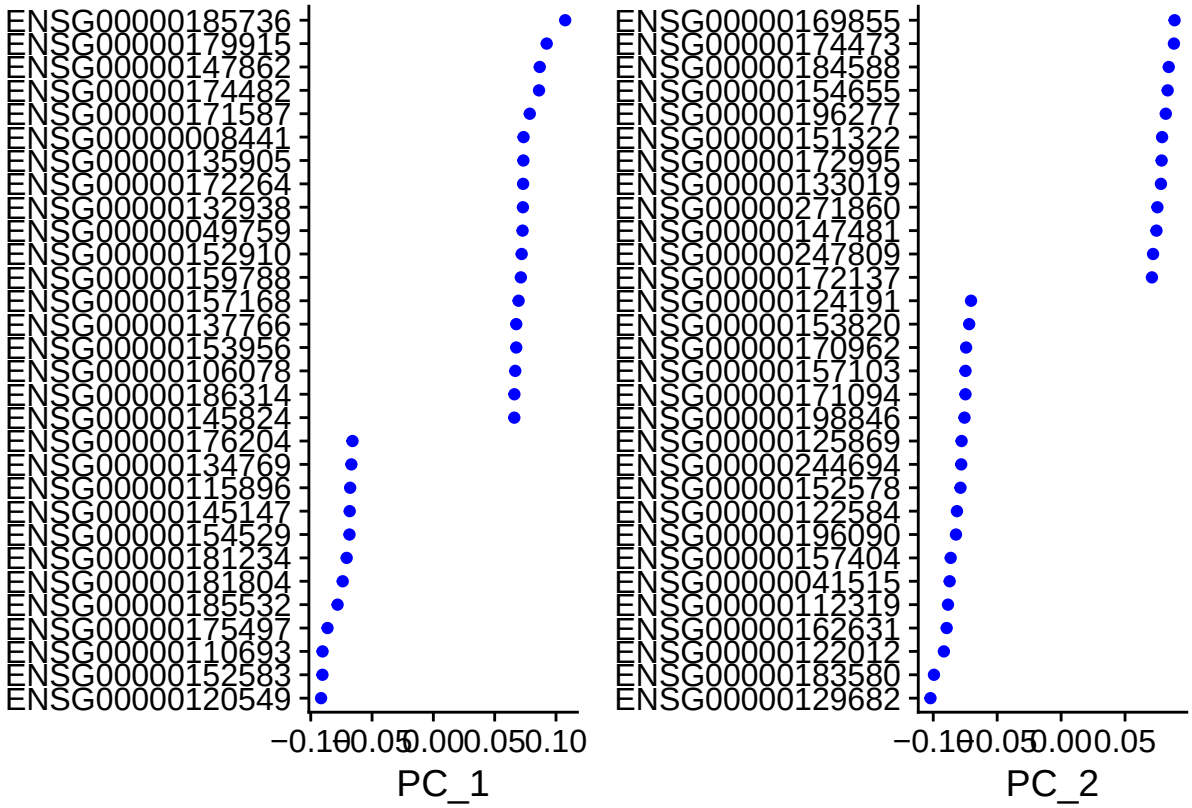
```
file_subset <-RunPCA(file_subset)
```

```
## PC_ 1
## Positive:  ENSG00000185736, ENSG00000179915, ENSG00000147862, ENSG00000174482, ENSG00000171587, ENSG
##     ENSG00000152910, ENSG00000159788, ENSG00000157168, ENSG00000137766, ENSG00000153956, ENSG0000010
##     ENSG00000182132, ENSG00000168843, ENSG00000146648, ENSG00000117707, ENSG00000133019, ENSG0000013
## Negative:  ENSG00000120549, ENSG00000152583, ENSG00000110693, ENSG00000175497, ENSG00000185532, ENSG
##     ENSG00000134769, ENSG00000176204, ENSG00000168993, ENSG00000236107, ENSG00000198673, ENSG0000016
##     ENSG00000106714, ENSG00000136237, ENSG00000164100, ENSG00000122584, ENSG00000009694, ENSG0000019
## PC_ 2
## Positive:  ENSG00000169855, ENSG00000174473, ENSG00000184588, ENSG00000154655, ENSG00000196277, ENSG
##     ENSG00000247809, ENSG00000172137, ENSG00000040731, ENSG00000145416, ENSG00000244128, ENSG0000015
##     ENSG00000163630, ENSG00000185352, ENSG00000118733, ENSG00000159788, ENSG00000132334, ENSG0000013
```

```
## Negative:  ENSG00000129682, ENSG00000183580, ENSG00000122012, ENSG00000162631, ENSG00000112319, ENSG
##     ENSG00000244694, ENSG00000125869, ENSG00000198846, ENSG00000171094, ENSG00000157103, ENSG0000017
##     ENSG00000156687, ENSG00000169282, ENSG00000186314, ENSG00000031081, ENSG00000153956, ENSG0000013
## PC_ 3
## Positive:  ENSG00000231079, ENSG00000206579, ENSG00000280441, ENSG00000177103, ENSG00000186094, ENSG
##     ENSG00000139364, ENSG00000231185, ENSG00000198626, ENSG00000112530, ENSG00000120658, ENSG0000016
##     ENSG00000122584, ENSG00000147724, ENSG00000165246, ENSG00000119771, ENSG00000149256, ENSG0000018
## Negative:  ENSG00000111640, ENSG00000143153, ENSG00000205542, ENSG00000166165, ENSG00000105649, ENSG
##     ENSG00000126267, ENSG00000103363, ENSG00000229117, ENSG00000140319, ENSG00000173267, ENSG0000010
##     ENSG00000130770, ENSG00000125356, ENSG00000189043, ENSG00000111669, ENSG00000137267, ENSG0000009
## PC_ 4
## Positive:  ENSG00000183166, ENSG00000170396, ENSG00000155052, ENSG00000146555, ENSG00000005108, ENSG
##     ENSG00000140470, ENSG00000152578, ENSG00000171444, ENSG00000077092, ENSG00000149571, ENSG0000011
##     ENSG00000109452, ENSG00000069667, ENSG00000041515, ENSG00000185274, ENSG00000181234, ENSG0000023
## Negative:  ENSG00000072657, ENSG00000115252, ENSG00000184672, ENSG00000162670, ENSG00000152822, ENSG
##     ENSG00000166342, ENSG00000157005, ENSG00000114805, ENSG00000157445, ENSG00000183098, ENSG0000017
##     ENSG00000152894, ENSG00000137672, ENSG00000171189, ENSG00000143195, ENSG00000116729, ENSG0000016
## PC_ 5
## Positive:  ENSG00000170011, ENSG00000162947, ENSG00000181722, ENSG00000144619, ENSG00000146555, ENSG
##     ENSG00000126733, ENSG00000132938, ENSG00000134532, ENSG00000233723, ENSG00000134533, ENSG0000011
##     ENSG00000137968, ENSG00000107518, ENSG00000147246, ENSG00000134115, ENSG00000196277, ENSG0000022
## Negative:  ENSG00000185760, ENSG00000118432, ENSG00000182836, ENSG00000130226, ENSG00000152377, ENSG
##     ENSG00000228222, ENSG00000158258, ENSG00000245532, ENSG00000170624, ENSG00000055813, ENSG0000016
##     ENSG00000136205, ENSG00000187416, ENSG00000082438, ENSG00000143153, ENSG00000147571, ENSG0000006
```

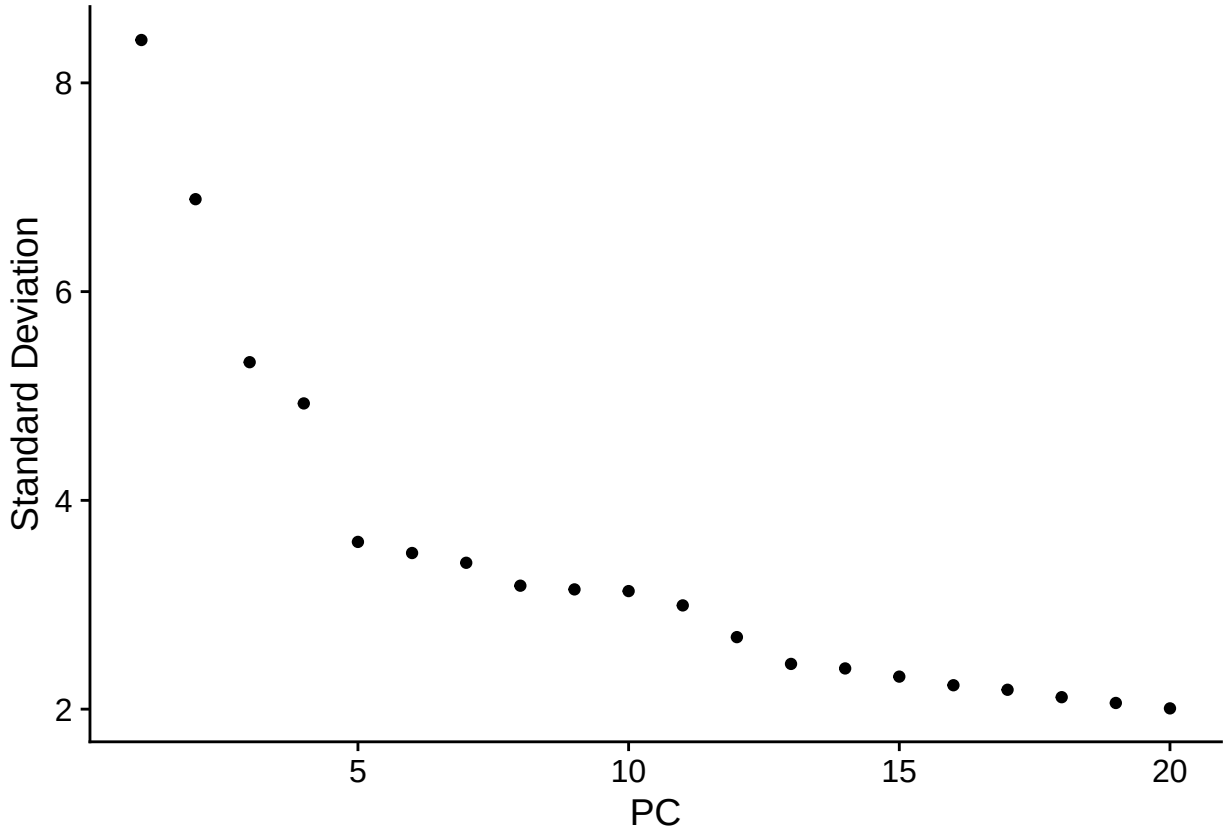
##. Plot visualizing principle componenet 1 and 2

```
VizDimLoadings(file_subset, dims = 1:2, reduction = "pca")
```



##. Elbow plot to determine number of pricinple component required to have data information

```
ElbowPlot(file_subset)
```



##. Finding neighbors and clusters of the features

```
file_subset <- FindNeighbors(file_subset, dims = 1:18, graph.name = "RNA_snn")
```

```
## Computing nearest neighbor graph

## Computing SNN

## Only one graph name supplied, storing nearest-neighbor graph only
```

```
file_subset <- FindClusters(file_subset)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 19858
## Number of edges: 191174
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8882
## Number of communities: 23
## Elapsed time: 0 seconds

## 5 singletons identified. 18 final clusters.
```

```
table(file_subset$seurat_clusters)
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2493 2053 1947 1634 1548 1291 1281 1128 1032  900  786  781  762  657  622  559
##   16   17
##  223  161
```

##. All 20 can be used but decided to use 18 dimensions of PC from the elbow plot

```
file_subset <- RunUMAP(file_subset, dims = 1:18, reduction = 'pca')
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
```

```
## 17:18:44 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 17:18:44 Read 19858 rows and found 18 numeric columns
```

```
## 17:18:44 Using Annoy for neighbor search, n_neighbors = 30
```

```
## 17:18:44 Building Annoy index with metric = cosine, n_trees = 50
```

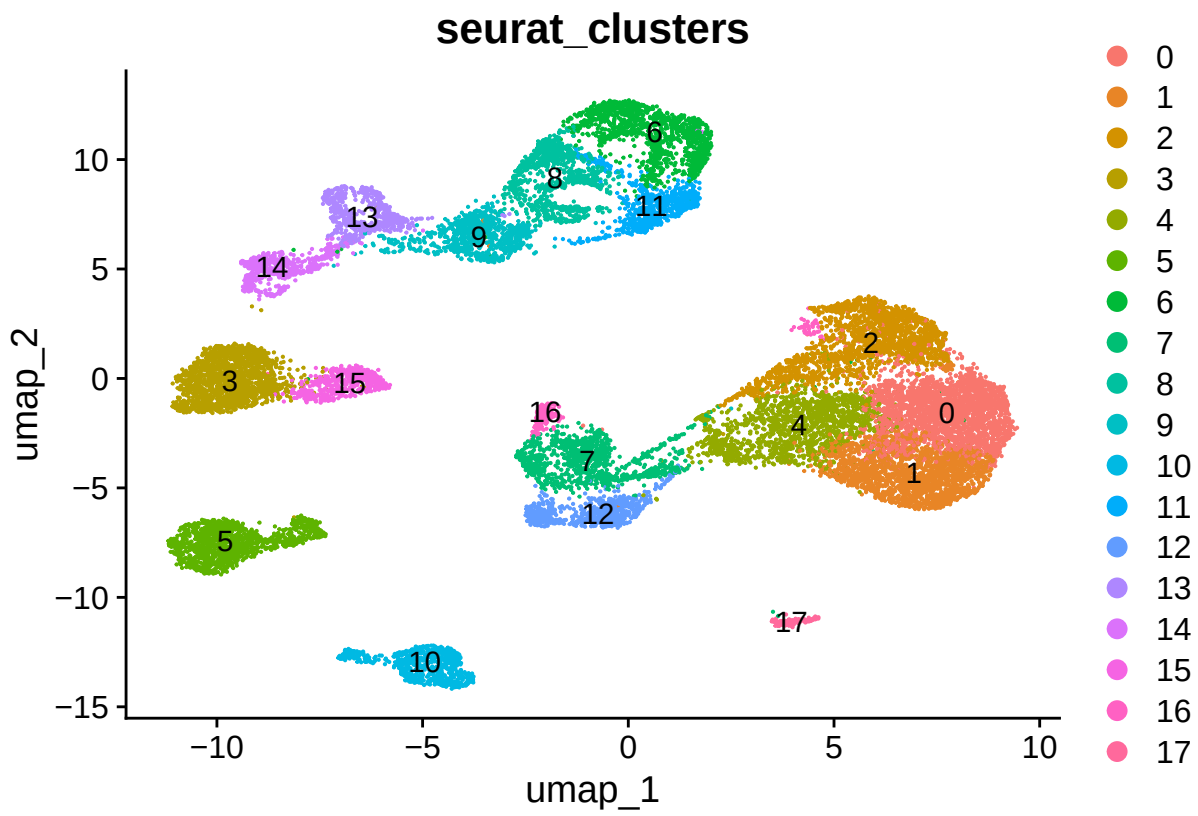
```
## 0%   10   20   30   40   50   60   70   80   90   100%
```

```
## [----|----|----|----|----|----|----|----|----|----|
```

```
## **************************************************|
## 17:18:45 Writing NN index file to temp file /var/folders/9l/1hw3qq5x4vs6zhdryctxzm4c0000gn/T//RtmpZg
## 17:18:45 Searching Annoy index using 1 thread, search_k = 3000
## 17:18:48 Annoy recall = 100%
## 17:18:49 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 17:18:49 Initializing from normalized Laplacian + noise (using RSpectra)
## 17:18:50 Commencing optimization for 200 epochs, with 798312 positive edges
## 17:18:56 Optimization finished
```

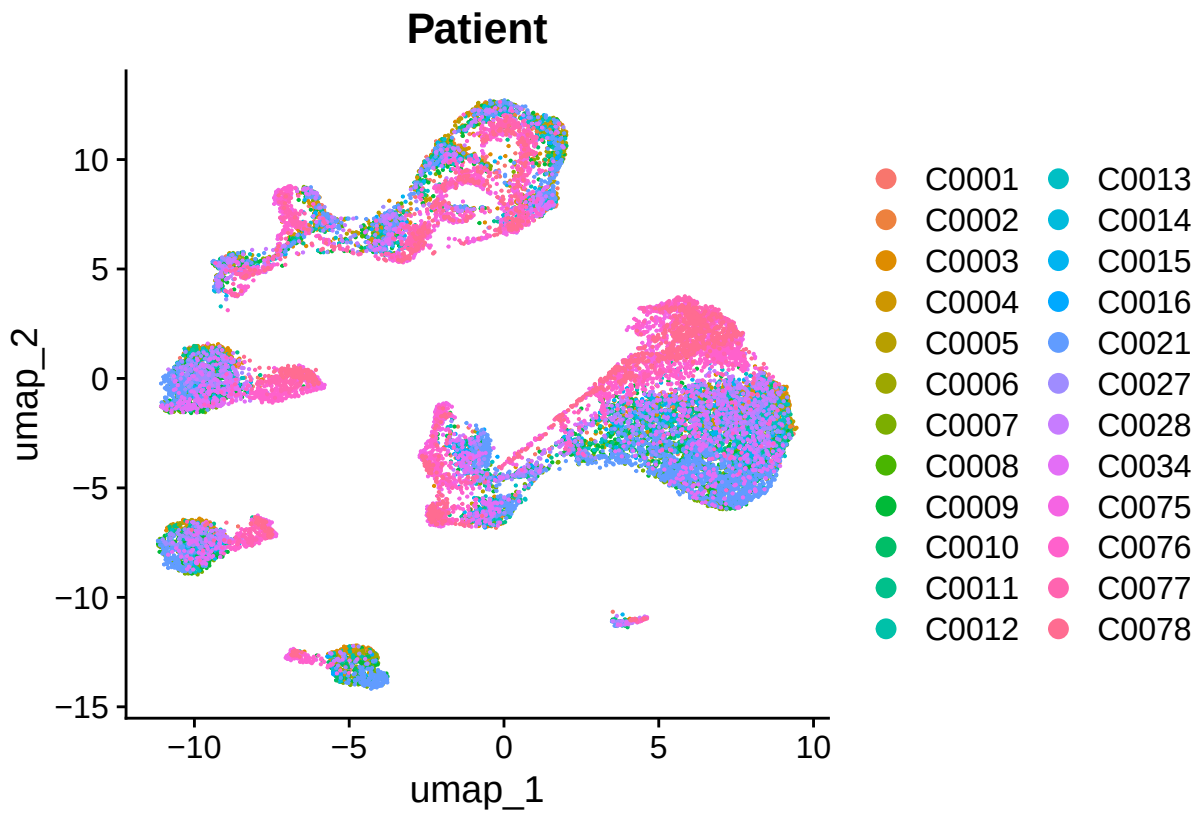
##. Dimplot to visualize seurat cluster, cluster based on condition(Disease and patient)

```
seurat_cluster <- DimPlot(file_subset, reduction = "umap", group.by = "seurat_clusters", label = TRUE)
Patient_before_harmony <- DimPlot(file_subset, reduction = "umap", group.by = "Patient")
disease_before_harmony <- DimPlot(file_subset, reduction = "umap", group.by = "disease")
seurat_cluster
```



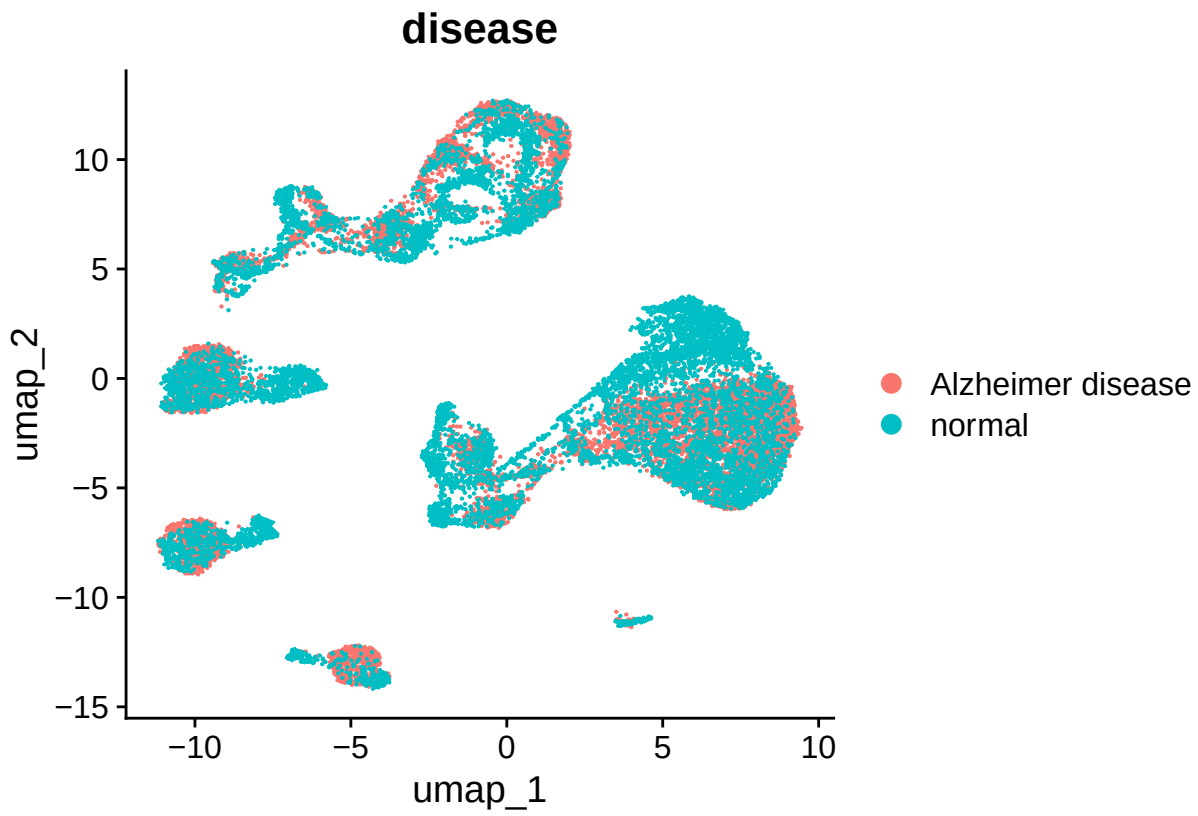
Visualization of a patient VS disease cluster

```
Patient_before_harmony
```



##. Visualization od disease clusters

```
disease_before_harmony
```



##. Harmony to remove batch effect as the Dimplot of disease conditions is slightly misleading cluster 2 and 1, it will remove the batch effects

```
# Apply Harmony to remove batch effect based on the 'disease' column
harmony_subset <- RunHarmony(
  object = file_subset,
  group.by.vars = "disease",
  dims.use = 1:18  # Use the same dimensions you selected in FindNeighbors
)
```

```
## Transposing data matrix

## Initializing state using k-means centroids initialization

## Harmony 1/10

## Harmony 2/10

## Harmony 3/10

## Harmony converged after 3 iterations
```

##. Running UMAP, FindingNeighbors, Clusters again but using redution as harmony this time

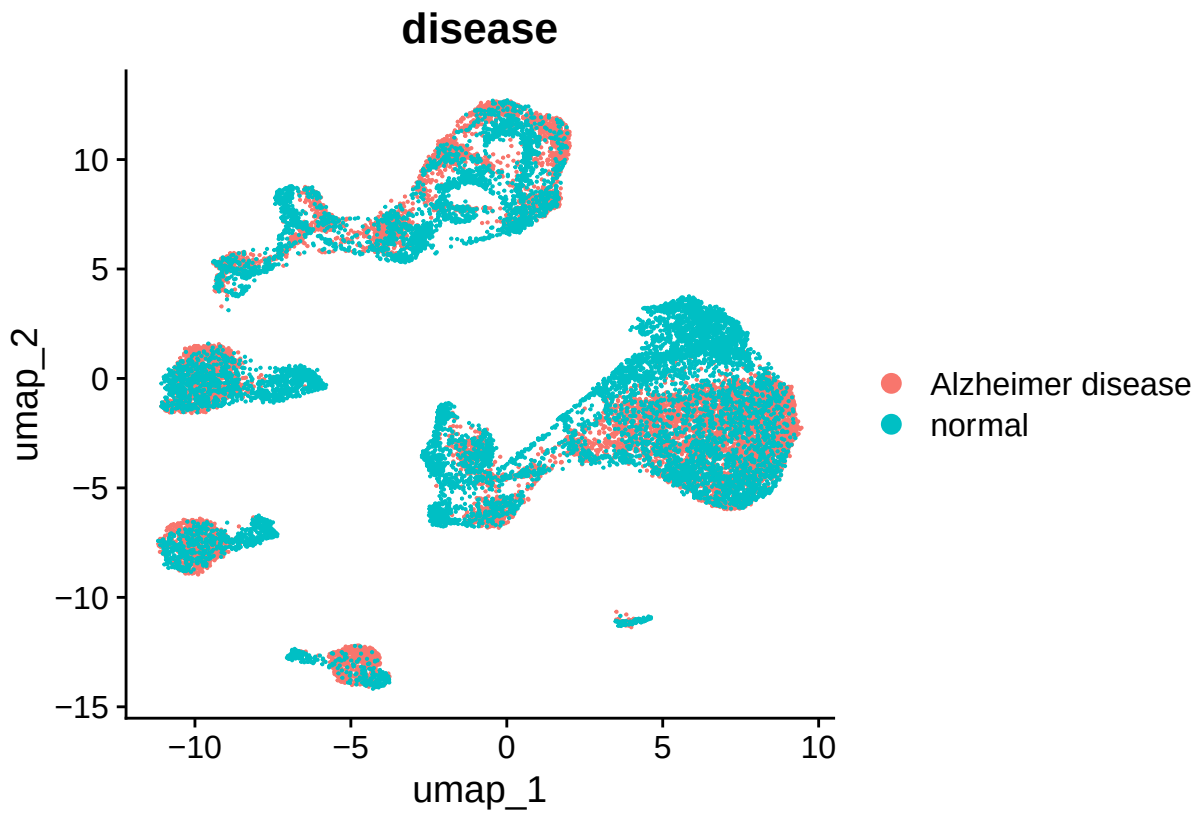
```r
harmony_subset <- harmony_subset%>%
  RunUMAP(reduction = 'harmony',dims = 1:18) %>%
  FindNeighbors(reduction = 'harmony' , dims = 1:18) %>%
  FindClusters(resolution = 0.5)
```

```
## 17:19:09 UMAP embedding parameters a = 0.9922 b = 1.112

## 17:19:09 Read 19858 rows and found 18 numeric columns

## 17:19:09 Using Annoy for neighbor search, n_neighbors = 30

## 17:19:09 Building Annoy index with metric = cosine, n_trees = 50

## 0%   10   20   30   40   50   60   70   80   90   100%

## [----|----|----|----|----|----|----|----|----|----|

## **************************************************|
## 17:19:10 Writing NN index file to temp file /var/folders/9l/1hw3qq5x4vs6zhdryctxzm4c0000gn/T//RtmpZg
## 17:19:10 Searching Annoy index using 1 thread, search_k = 3000
## 17:19:14 Annoy recall = 100%
## 17:19:14 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 17:19:15 Initializing from normalized Laplacian + noise (using RSpectra)
## 17:19:17 Commencing optimization for 200 epochs, with 804202 positive edges
## 17:19:23 Optimization finished
## Computing nearest neighbor graph
## Computing SNN

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 19858
## Number of edges: 677325
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9234
## Number of communities: 15
## Elapsed time: 2 seconds
```

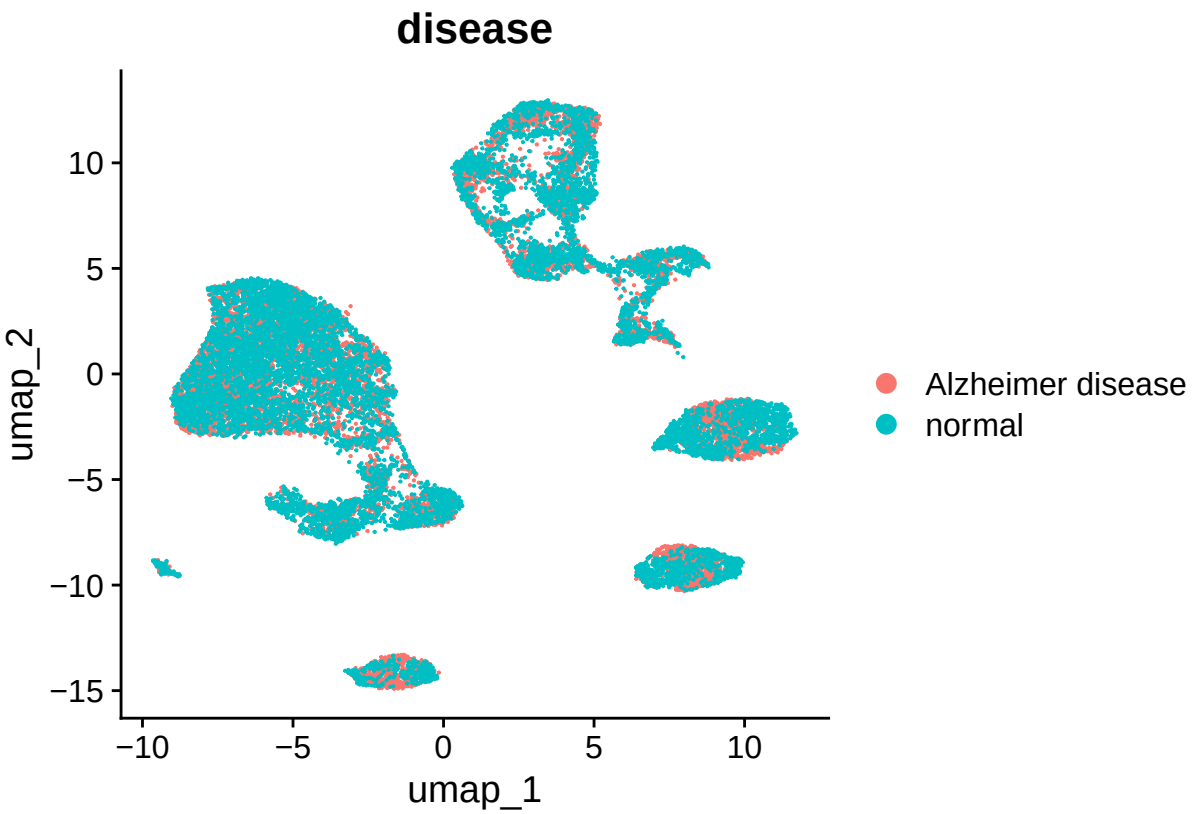
```r
disease_After_harmony <- DimPlot(harmony_subset,reduction = 'umap', group.by = 'disease')
Seurat_cluster_After_harmony <- DimPlot(harmony_subset,reduction = 'umap', group.by = "seurat_clusters"
```

##. Dimplot of before VS After of condition(disease) after batch correction

```r
disease_before_harmony
```




```
disease_After_harmony
```


disease
umap_2
umap_1
10
5
0
−5
−10
−15
−10
−5
0
5
10
Alzheimer disease
normal