Group 11

Gene Expression Analysis to Detect Alzheimers

GROUP MEMBERS

ABHISHEK BANSAL (2022021)

KARTIKEYA (2022241)

ANISH (2022075)

MOHD. MASOOD (2022299)

DEBJIT BANERJI (2022146)

VIJVAL EKBOTE (2022569)

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION

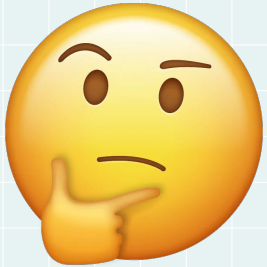
- Alzheimer's disease is a ***progressive disease*** that gets worse over time. Begins with Mild Memory loss and then leading to the ***loss of ability to respond*** to the environment.
- In early stages, it includes ***forgetting recent events*** or conversations and then leading to serious memory problems like ***loss of ability to perform daily activities*** and much more.
- People ***suffering*** with Alzheimer's disease may face ***problems*** such as :
  - *Repeat statements and ask same thing again*
  - *Forget recent conversations, events, appointments*
  - *Misplace items*
  - *Get lost in place which is well known to them*
  - *And, eventually forgetting the names of their family members*

INTRODUCTION

INTRODUCTION

But, How Long it lasts ?

It may be as little as *three or four years* if the person is **older than 80** when diagnosed, to as long as *10 or more years* if the person is **younger.**



And, What's the cure ?

Currently, there is **no cure** for Alzheimer's disease, though there has been **significant progress in recent years** in developing and testing new treatments. Several medicines have been approved by the **U.S. Food and Drug Administration** to treat people with Alzheimer's.

01

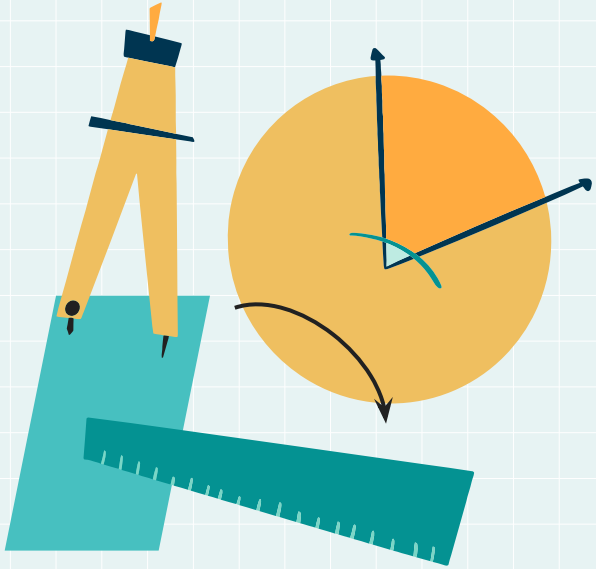
PROBLEM STATEMENT

PROBLEM STATEMENT

- We will be performing **GENE EXPRESSION ANALYSIS** to identify Normal genes and Alzheimer's genes.

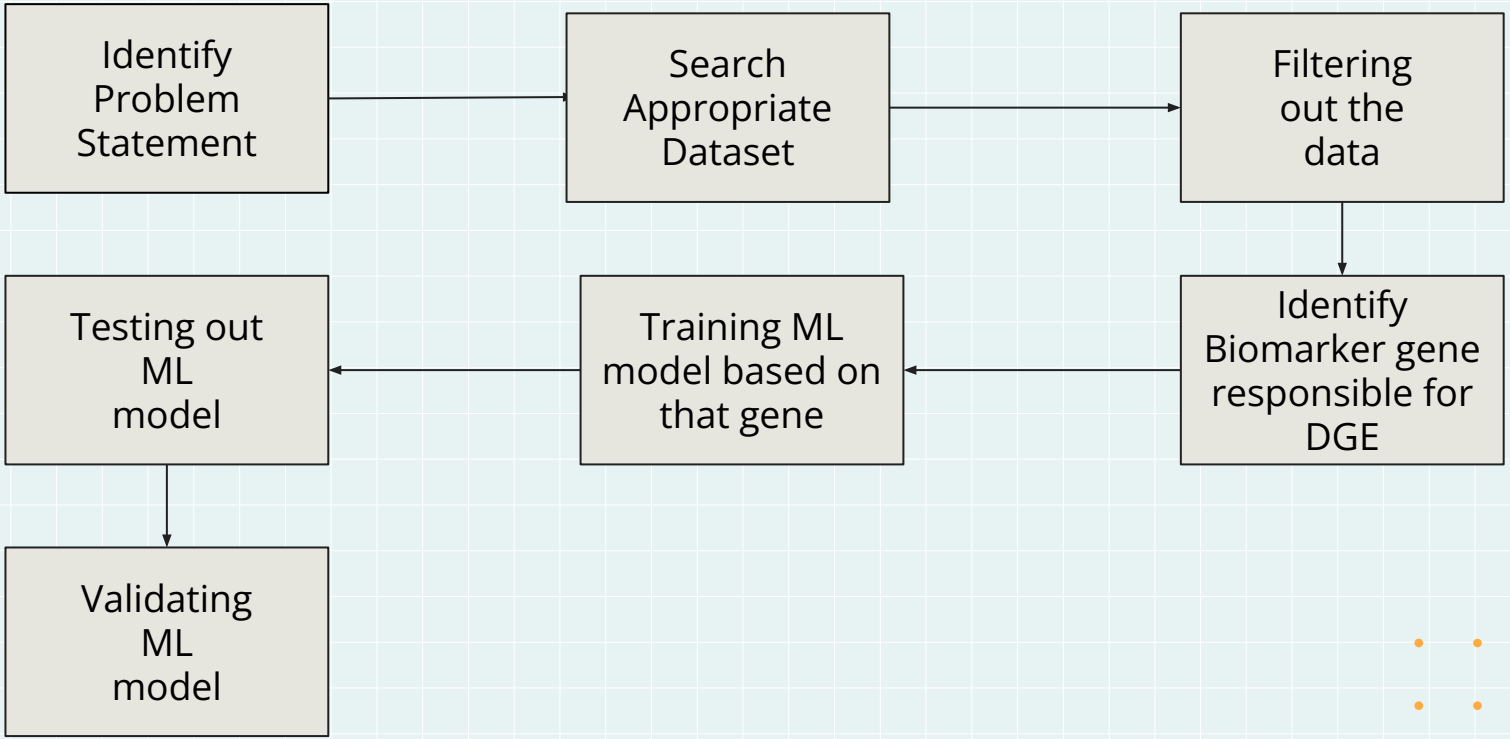
- We will achieve this, using **NCBI GEO2R** tool, which will help us to identify which gene is diseased and which's not, using **DIFFERENTIAL GENE ANALYSIS**, with the help of plots such as **Volcano** plot, **clustered** plot, Statistical Test Analysis etc.

- And, finally, we will be training our model which, given the input of gene sample of a person, will be able to determine whether a person has AD or not.



FLOW OF THE PROJECT

FLOWCHART


Identify Problem Statement
Search Appropriate Dataset
Filtering out the data
Identify Biomarker gene responsible for DGE
Training ML model based on that gene
Testing out ML model
Validating ML model

02

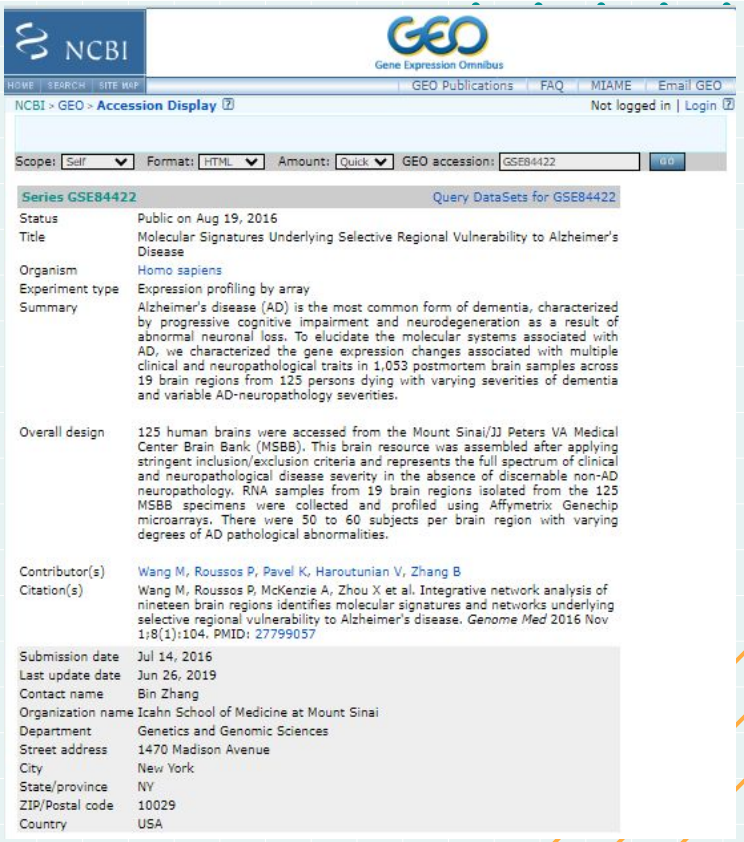
DATA COLLECTION

DATA COLLECTION

- **SOURCE :** Geo2R (link)

- **DATA:** RNA samples from *19* regions isolated from *125* MSBB specimen were collected and profiled using Affymetrix Genechip microarrays. So Total *1053* postmortem brain samples. Each sample has the expression levels of *14160* genes.

- **WHY WE HAVE CHOSEN THIS DATASETS OUT MANY OTHERS PRESENT**: The data is already grouped into classes like Definite Alzheimer and Normal because of which it saves a lot of our preprocessing procedure



NCBI | GEO Gene Expression Omnibus

HOME | SEARCH | SITE MAP | GEO Publications | FAQ | MIAME | Email GEO

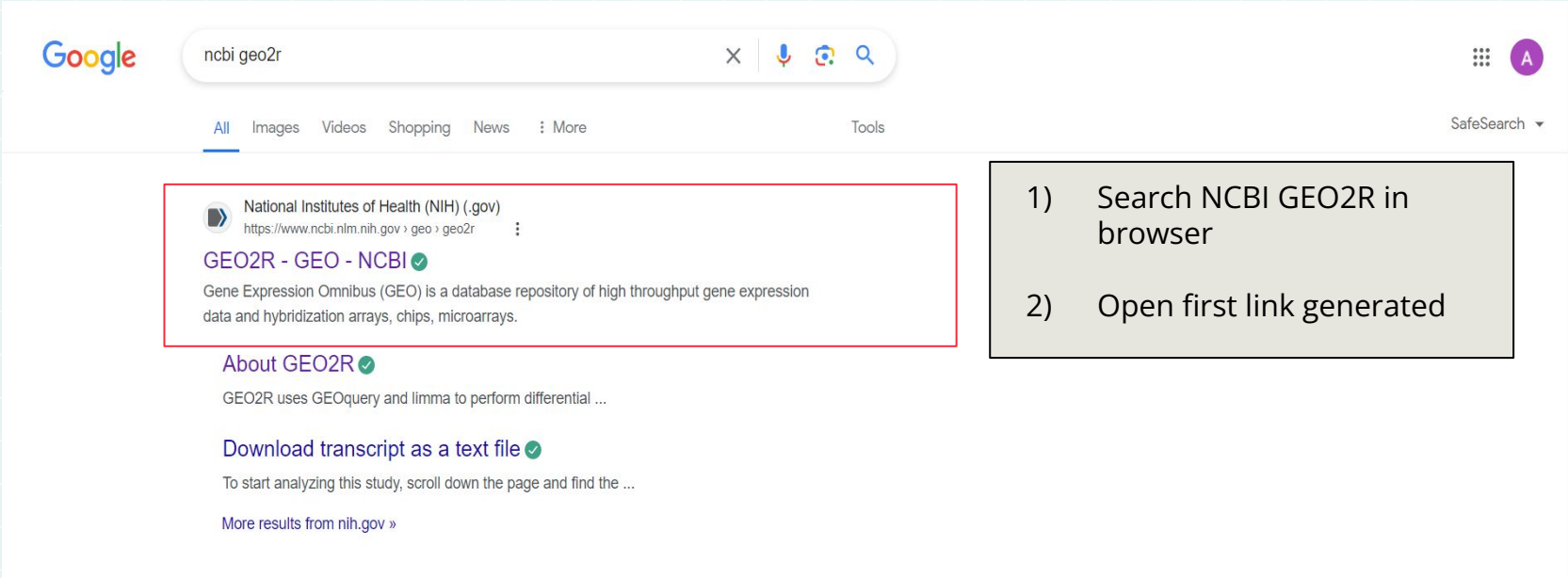
NCBI > GEO > Accession Display | Not logged in | Login

Scope: Self | Format: HTML | Amount: Quick | GEO accession: GSE84422 | GO

Series GSE84422 | Query DataSets for GSE84422

Status	Public on Aug 19, 2016
Title	Molecular Signatures Underlying Selective Regional Vulnerability to Alzheimer's Disease
Organism	Homo sapiens
Experiment type	Expression profiling by array
Summary	Alzheimer's disease (AD) is the most common form of dementia, characterized by progressive cognitive impairment and neurodegeneration as a result of abnormal neuronal loss. To elucidate the molecular systems associated with AD, we characterized the gene expression changes associated with multiple clinical and neuropathological traits in 1,053 postmortem brain samples across 19 brain regions from 125 persons dying with varying severities of dementia and variable AD-neuropathology severities.
Overall design	125 human brains were accessed from the Mount Sinai/JJ Peters VA Medical Center Brain Bank (MSBB). This brain resource was assembled after applying stringent inclusion/exclusion criteria and represents the full spectrum of clinical and neuropathological disease severity in the absence of discernable non-AD neuropathology. RNA samples from 19 brain regions isolated from the 125 MSBB specimens were collected and profiled using Affymetrix Genechip microarrays. There were 50 to 60 subjects per brain region with varying degrees of AD pathological abnormalities.
Contributor(s)	Wang M, Roussos P, Pavel K, Haroutunian V, Zhang B
Citation(s)	Wang M, Roussos P, McKenzie A, Zhou X et al. Integrative network analysis of nineteen brain regions identifies molecular signatures and networks underlying selective regional vulnerability to Alzheimer's disease. *Genome Med* 2016 Nov 1;8(1):104. PMID: 27799057
Submission date	Jul 14, 2016
Last update date	Jun 26, 2019
Contact name	Bin Zhang
Organization name	Icahn School of Medicine at Mount Sinai
Department	Genetics and Genomic Sciences
Street address	1470 Madison Avenue
City	New York
State/province	NY
ZIP/Postal code	10029
Country	USA

DATA COLLECTION PROCEDURE



1) Search NCBI GEO2R in browser

2) Open first link generated

NCBI

GEO Gene Expression Omnibus

GEO Publications | FAQ | MIAME | Email GEO

NCBI » GEO » GEO2R

Login

Use GEO2R to compare two or more groups of Samples in order to identify genes that are differentially expressed across experimental conditions. Results are presented as a table of genes ordered by significance. Full instructions YouTube

Write GSE84422 as Acccession number in the box and click on SET

GEO accession Set

GEO2R | Options | Profile graph | R script

Quick start

- Specify a GEO Series accession and a Platform if prompted.
- Click 'Define groups' and enter names for the groups of Samples you plan to compare, e.g., test and control.
- Assign Samples to each group. Highlight Sample rows then click the group name to assign those Samples to the group. Use the Sample metadata (title, source and characteristics) columns to help determine which Samples belong to which group.
- Click 'Analyze' to perform the calculation with default settings.
- You may change settings in the Options tab.

How to use

Analyze

| NLM | NIH | Email GEO | Disclaimer | Accessibility |

NCBI

GEO Gene Expression Omnibus

GEO Publications | FAQ | MIAME | Email GEO

NCBI » GEO » GEO2R » GSE84422 Login

Use GEO2R to compare two or more groups of Samples in order to identify genes that are differentially expressed across experimental conditions. Results are presented as a table of genes ordered by significance. Full instructions YouTube

GEO accession GSE84422 Set Molecular Signatures Underlying Selective Regional Vulnerability to Alzheimer's Disease

Platform GPL97

Select a platform
GPL96
GPL97
GPL570

Select platform as GPL97 → Scroll down and click on ANALYZE button

Samples Define groups Selected 0 out of 931 samples

Columns Set

Group	Accession	Title	Source name	Subject id	Age	Sex	Race	Postmortem interval minutes	Ph	Clinical dementia rating	Braak neurofibrilla tangle scor
-	GSM2234572	Subject 1006, region Frontal Pole	Frontal Pole, 98 yr old female	1006	98	female	black	300	6.5	3	4
-	GSM2234573	Subject 1018, region Frontal Pole	Frontal Pole, 60 yr old male	1018	60	male	white	1725	6.76	0	0
-	GSM2234574	Subject 1023, region Frontal Pole	Frontal Pole, 92 yr old female	1023	92	female	white	115	6.7	3	6
-	GSM2234575	Subject 1027, region Frontal Pole	Frontal Pole, 80 yr old male	1027	80	male	white	120	6.37	3	6
-	GSM2234576	Subject 1039, region Frontal Pole	Frontal Pole, 85 yr old male	1039	85	male	white	315	6.9	0	2
-	GSM2234577	Subject 111, region Frontal Pole	Frontal Pole, 93 yr old female	111	93	female	white	160	6.1	5	4
-	GSM2234578	Subject 184, region Frontal Pole	Frontal Pole, 86 yr old female	184	86	female	white	126	6.4	2	3
-	GSM2234579	Subject 2010, region Frontal Pole	Frontal Pole, 70 yr old male	2010	70	male	white	1080	6.9	0	1

03

DATA PROCESSING

GEO2R - GEO - NCBI | Farewell Aa Raha Hai | Wagle Ki Duni | New Tab | Claude

ncbi.nlm.nih.gov/geo/geo2r/?acc=GSE84422&platform=GPL96

classroom | Master C Program...

NCBI

GEO Gene Expression Omnibus

GEO Publications | FAQ | MIAME | Email GEO

NCBI » GEO » GEO2R » GSE84422 Login

Use GEO2R to compare two or more groups of Samples in order to identify genes that are differentially expressed across experimental conditions. Results are presented as a table of genes ordered by significance. Full instructions

GEO accession GSE84422 Set Molecular Signatures Underlying Selective Regional Vulnerability to Alzheimer's Disease

Platform GPL96

Samples Define groups Selected **0** out of **951** samples

1) Creating groups

Enter a group name: List

Cancel selection

Normal

Definite AD

Columns Set

ne	Sut		Sex	Race	Postmortem interval minutes	Ph	Clinical dementia rating	Braak neurofibrillary tangle score	Neuropathological category	Average neuritic plaque density	Sum of cerad rating scores in multiple brain regions	Sum of neurofibrillary tangles density multiple brain regions	Brain reg
e, 98 yr old female	1006	98	female	black	300	6.5	3	4	Possible AD	2.48	3	10	Frontal F
e, 60 yr old male	1018	60	male	white	1725	6.76	0	0	Normal	0	0	0	Frontal F
e, 92 yr old female	1023	92	female	white	115	6.7	3	6	definite AD	6.4	9	18	Frontal F
e, 80 yr old male	1027	80	male	white	120	6.37	3	6	Probable AD	7.12	15	22	Frontal F
e, 85 yr old male	1039	85	male	white	315	6.9	0	2	Normal	0	0	4	Frontal F
e, 93 yr old female	111	93	female	white	160	6.1	5	4	Probable AD	8.51	19	10	Frontal F
e, 86 yr old female	184	86	female	white	126	6.4	2	3	Probable AD	6.8	8	6	Frontal F
e, 70 yr old male	2010	70	male	white	1080	6.9	0	1	Normal	0	0	0	Frontal F

Search ENG IN 19:15 10-05-2024

2) Characterising samples into suitable groups

Platform GPL96

Samples

Define groups

Enter a group name: List

enter group class

Cancel selection

Normal (38 samples)

Definite AD (30 samples)

Selected **68** out of **951** samples

Columns Set

ı, 96 yr old female			female	white	195	6.7	0	2	Normal	0	0	3	Putan
ı, 69 yr old male			male	black	255	6.3	0.5	1	Normal	0	0	1	Putan
ı, 82 yr old female			female	white	340	6.1	0	3	Normal	7.16	13	4	Putan
ı, 92 yr old female			female	white	210	6.2	4	4	Normal	5.36	11	6	Putan
ı, 85 yr old female	551	85	female	white	260	6.3	0	2	Normal	4.08	4	2	Putan
ı, 102 yr old female	625	102	female	white	423	6.5	0	6	Normal	1.12	5	10	Putan
ı, 89 yr old female	800	89	female	white	140	6.7	0	2	Normal	1.28	1	4	Putan
ı, 66 yr old male	843	66	male	white	454	6.58	1	2	Normal	0	0	7	Putan
Pole, 92 yr old female	1023	92	female	white	115	6.7	3	6	definite AD	6.4	9	18	Fronta
Pole, 91 yr old female	215	91	female	black	230	7	2	6	definite AD	29.76	29	24	Fronta
Pole, 99 yr old female	256	99	female	white	330	6.2	3	3	definite AD	8.48	20	9	Fronta
Pole, 92 yr old male	273	92	male	white	120	6.1	2	3	definite AD	20.96	25	9	Fronta
Pole, 64 yr old male	316	64	male	white	240	6.4	5	6	definite AD	35.28	31	35	Fronta
Pole, 94 yr old female	332	94	female	white	165	6	0.5	1	definite AD	12.16	29	2	Fronta
Pole, 84 yr old female	347	84	female	white	165	6.6	5	2	definite AD	14.32	33	2	Fronta
Pole, 84 yr old female	364	84	female	white	120	5.9	5	6	definite AD	20.4	35	26	Fronta

1, 70 yr old male	2010	70	male	white	1080	6.9	0	1	Normal	0	0	0	Putan
1, 96 yr old female	230	96	female	white	195	6.7	0	2	Normal	0	0	3	Putan
1, 69 yr old male	495	69	male	black	255	6.3	0.5	1	Normal	0	0	1	Putan
1, 82 yr old female	523	82	female	white	340	6.1	0	3	Norma				

3) Select appropriate parameters like p-value and fold change threshold

GEO2R | Options | Profile graph | R script

Apply adjustment to the P-values. More...

- (●) Benjamini & Hochberg (False discovery rate)
- ( ) Benjamini & Yekutieli
- ( ) Bonferroni
- ( ) Holm

1) Select options button

Apply log transformation to the data. More...

- (●) Auto-detect
- ( ) Yes
- ( ) No

Apply limma precision weights (vooma). More...

- ( ) Yes
- (●) No

Force normalization. More...

- (●) Yes
- ( ) No

2) Apply Normalisation

Category of Platform annotation to display on results.

- ( ) Submitter supplied
- (●) NCBI generated

Plot displays. More...

Significance level cut-off (enter number between 0 and 1) [0.05]

Log 2 fold change threshold [0]

Volcano and Mean-difference plot contrasts (select up to 5)
1 selected (clear)

- [x] Normal vs Definite AD

4) Select Volcano Plots required

If you edit *Options* after performing an analysis, click *Reanalyze* to apply the edits:

Reanalyze

5) Analyze

GEO 2R ANALYZER

Visualizer Tab

List of top differentially expressed genes

GEO2R - GEO - NCBI

ncbi.nlm.nih.gov/geo/geo2r/?acc=GSE84422&platform=GPL96

Platform GPL96

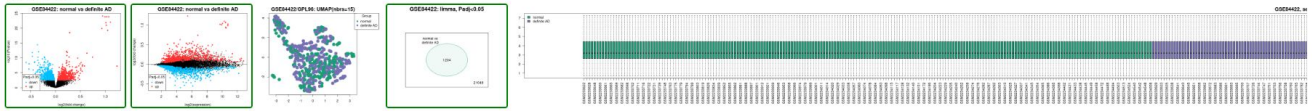
Samples Define groups

Selected 539 out of 951 samples

GEO2R Options Profile graph R script

Reanalyze if you changed any options.

Visualization



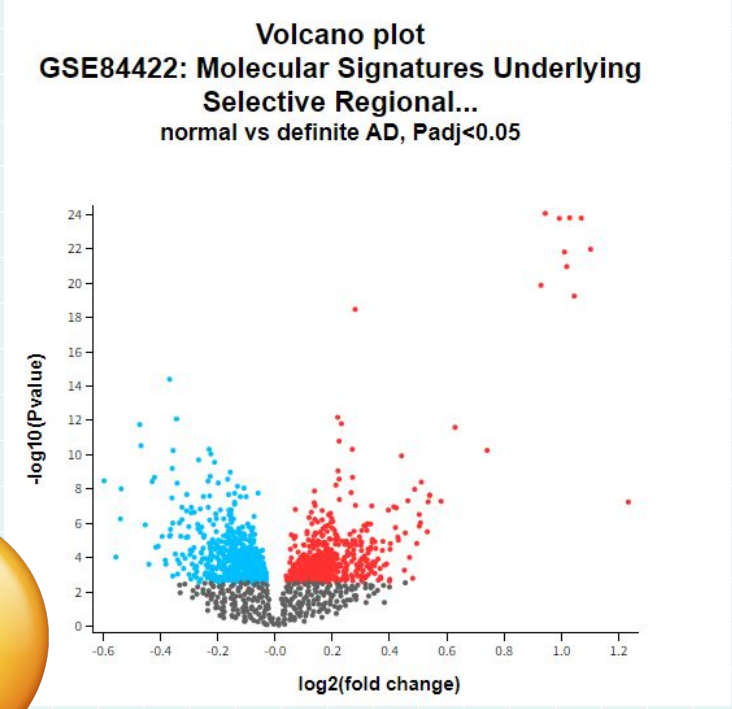
Top differentially expressed genes

Download full table Select columns

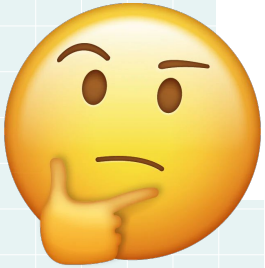
ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
214414_x_at	9.15e-21	8.26e-25	10.82	44.84	0.9437	HBA2///HBA1	hemoglobin subunit alpha 2///hemoglob...
204018_x_at	9.15e-21	1.52e-24	10.75	44.26	1.0293	HBA2///HBA1	hemoglobin subunit alpha 2///hemoglob...
209458_x_at	9.15e-21	1.59e-24	10.74	44.22	1.0695	HBA2///HBA1	hemoglobin subunit alpha 2///hemoglob...
211745_x_at	9.15e-21	1.64e-24	10.74	44.18	0.9928	HBA2///HBA1	hemoglobin subunit alpha 2///hemoglob...
209116_x_at	4.63e-19	1.04e-22	10.27	40.23	1.1015	HBB	hemoglobin subunit beta
217414_x_at	5.55e-19	1.50e-22	10.23	39.88	1.0111	HBA2///HBA1	hemoglobin subunit alpha 2///hemoglob...
211699_x_at	3.38e-18	1.06e-21	10	38.01	1.0195	HBA2///HBA1	hemoglobin subunit alpha 2///hemoglob...
211696_x_at	3.65e-17	1.31e-20	9.7	35.61	0.9291	HBB	hemoglobin subunit beta
217232_x_at	1.38e-16	5.59e-20	9.52	34.22	1.0453	HBB	hemoglobin subunit beta
215483_at	7.45e-16	3.35e-19	9.3	32.52	0.2804	AKAP9	A-kinase anchoring protein 9
205151_s_at	8.14e-12	4.02e-15	-8.09	23.57	-0.3679	TRIL	TLR4 interactor with leucine rich repeats
213396_s_at	1.22e-09	6.59e-13	7.37	18.72	0.2187	AKAP10	A-kinase anchoring protein 10
210473_s_at	1.42e-09	8.30e-13	-7.33	18.5	-0.3435	ADGRA3	adhesion G protein-coupled receptor A3
213515_x_at	2.50e-09	1.57e-12	7.24	17.89	0.2324	HBG2///HBG1	hemoglobin subunit gamma 2///hemogl...
205150_s_at	2.62e-09	1.76e-12	-7.22	17.78	-0.4721	TRIL	TLR4 interactor with leucine rich repeats
209183_s_at	3.54e-09	2.54e-12	7.17	17.43	0.6288	C10orf10	chromosome 10 open reading frame 10

VOLCANO PLOT

So finally we got the volcano plot.....



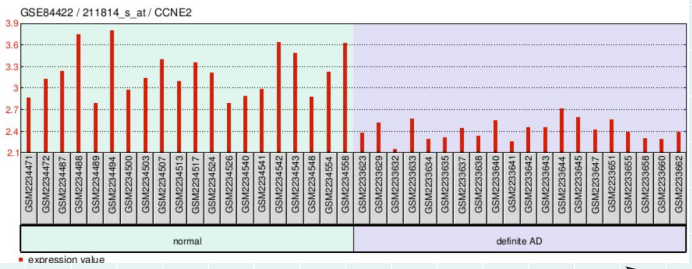
But what does this plot actually mean?

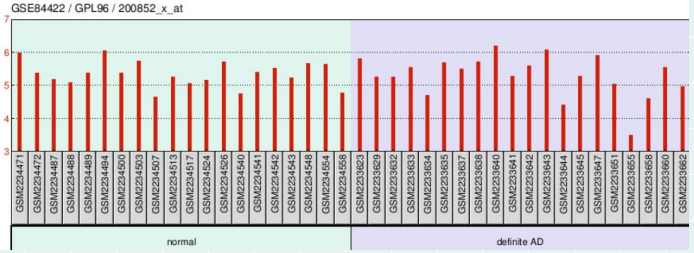


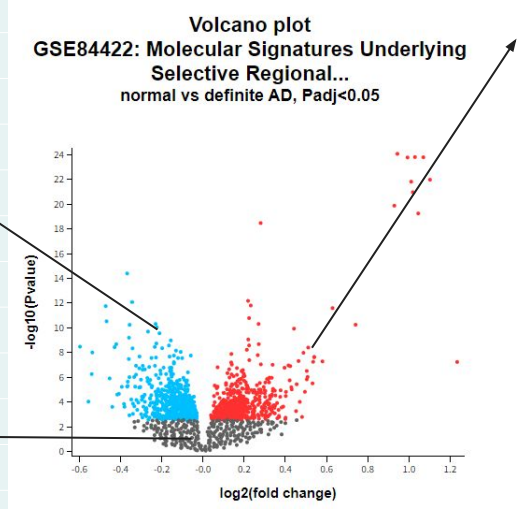
Interpretation Volcano Plot

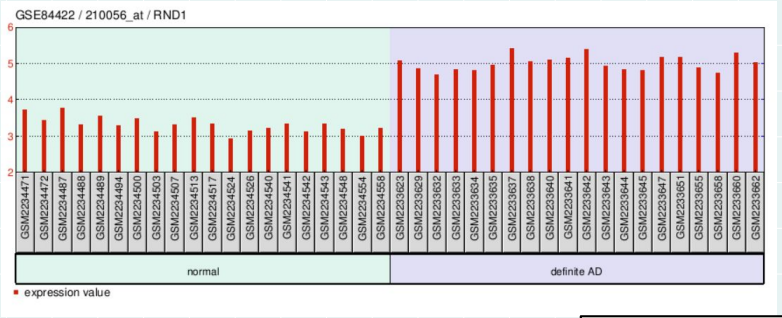
BLUE Points in the volcano plot are down regulated differentially expressed genes

GREY points in the volcano plot are similarly expressed genes and show no significant difference









RED Points in the volcano plot are upregulated differentially expressed genes

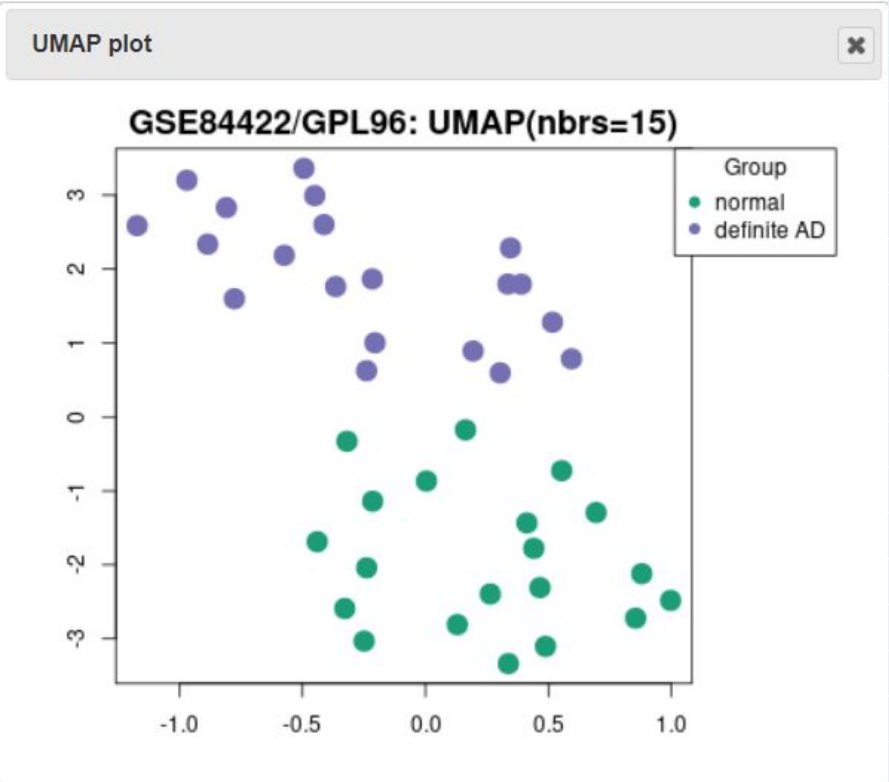
UMAP CLUSTER PLOT

INTERPRETATION

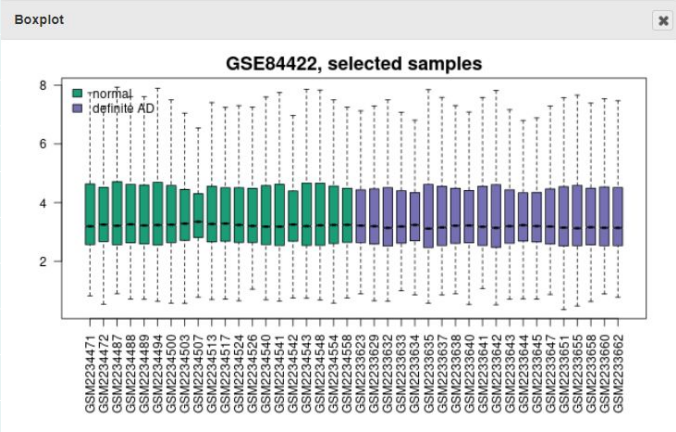
The purple coloured dots denote Alzheimer infected sample whereas green coloured dots represent normal sample

OBSERVATION

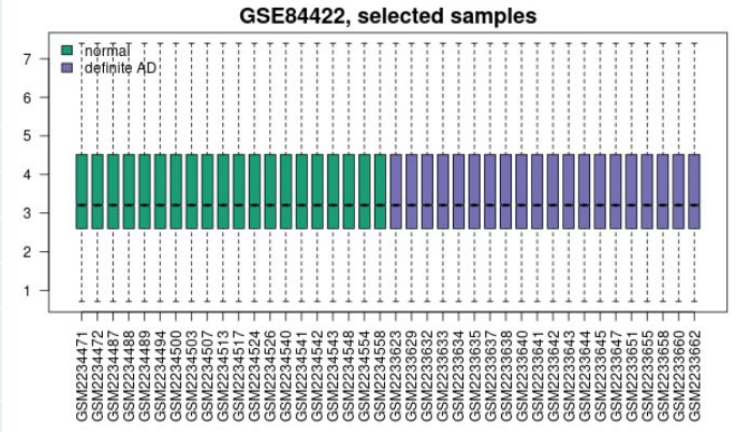
Observe that the purple coloured dots and green coloured dots form collective groups



BOX PLOTS



BEFORE NORMALISATION



AFTER NORMALISATION

TOOL USED :
Geo2R Analyzer
Normalization tool

OBSERVATION :
Centering of medians
Scaling by standard deviation

04

FORMATTING GEO2R DATA

STEPS

Step 1

Reading the metadata obtained from GEO2R into variables.

```python
df = pd.read_csv("96(1).csv")
df.drop('Unnamed: 0',axis = 1,inplace=True)
df.set_index('id_ref',inplace=True)
df.head()
```

✓ 0.3s Python

	subject_id	age	sex	race	postmortem_interval	ph	rating	bnt	cat	anpd	crssum	ntdsum	region
id_ref													
GSM2233621	1006	98	female	black	300	6.50	3.0	4	Possible	2.48	3	10	FrontalPole
GSM2233622	1018	60	male	white	1725	6.76	0.0	0	Normal	0.00	0	0	FrontalPole
GSM2233623	1023	92	female	white	115	6.70	3.0	6	definite	6.40	9	18	FrontalPole
GSM2233624	1027	80	male	white	120	6.37	3.0	6	Probable	7.12	15	22	FrontalPole
GSM2233625	1039	85	male	white	315	6.90	0.0	2	Normal	0.00	0	4	FrontalPole

STEPS

Step 2

Reading the expression levels obtained from GEO2R into variables.

```python
print(len(df.index.unique()))
df2 = pd.read_csv("96(2).csv")
df2.drop('Unnamed: 0',axis = 1,inplace=True)
df2.index.name = 'id_ref'
df2.head()
```

✓ 15.8s Python

951

	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	1405_i_at	1431_at	...	217004_s_at	217005_at	217006_x_at	217007
id_ref															
0	6.244141	2.904342	2.964129	3.237673	1.830522	2.569345	3.270412	2.511328	2.872239	2.541925	...	3.355867	2.613956	2.997592	2.72
1	5.653405	3.080484	3.056430	3.149935	2.013946	2.539322	3.353614	2.598044	3.195277	2.565015	...	3.178337	2.612671	2.616394	2.80
2	6.437320	2.779001	3.377981	3.233940	2.013279	2.747362	3.311479	2.651189	3.320735	2.750777	...	3.223486	2.581647	2.576565	2.41
3	6.422853	2.807162	2.845889	3.047864	1.958638	2.492102	3.323807	2.434233	2.679098	2.407532	...	3.238530	2.634674	2.629828	2.45
4	6.164218	2.833527	2.992403	3.067565	2.041814	2.579532	3.228061	2.576368	2.975454	2.942743	...	3.255730	2.614901	2.718517	2.84

5 rows × 16383 columns

STEPS

Step 3

Cleaning and adjusting the data

```python
l = df.index
ll = [i for i in l]

df2.index = ll
df2.index.name = 'id_ref'
df2.head()
```

✓ 0.0s Python

id_ref	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	1405_i_at	1431_at	...	217004_s_at	217005_at	217006_x_at
GSM2233621	6.244141	2.904342	2.964129	3.237673	1.830522	2.569345	3.270412	2.511328	2.872239	2.541925	...	3.355867	2.613956	2.997592
GSM2233622	5.653405	3.080484	3.056430	3.149935	2.013946	2.539322	3.353614	2.598044	3.195277	2.565015	...	3.178337	2.612671	2.616394
GSM2233623	6.437320	2.779001	3.377981	3.233940	2.013279	2.747362	3.311479	2.651189	3.320735	2.750777	...	3.223486	2.581647	2.576565
GSM2233624	6.422853	2.807162	2.845889	3.047864	1.958638	2.492102	3.323807	2.434233	2.679098	2.407532	...	3.238530	2.634674	2.629828
GSM2233625	6.164218	2.833527	2.992403	3.067565	2.041814	2.579532	3.228061	2.576368	2.975454	2.942743	...	3.255730	2.614901	2.718517

5 rows × 16383 columns

STEPS

Step 4

Merging the metadata and the expression levels DataFrames into a single DataFrame for further processing.

```python
df3 = pd.merge(df,df2,on = 'id_ref')
df3.shape
df3.head()
```

✓ 0.2s Python

	subject_id	age	sex	race	postmortem_interval	ph	rating	bnt	cat	anpd	...	217004_s_at	217005_at	217006_x_at	217007_s_at	2
id_ref																
GSM2233621	1006	98	female	black	300	6.50	3.0	4	Possible	2.48	...	3.355867	2.613956	2.997592	2.725131	
GSM2233622	1018	60	male	white	1725	6.76	0.0	0	Normal	0.00	...	3.178337	2.612671	2.616394	2.806919	
GSM2233623	1023	92	female	white	115	6.70	3.0	6	definite	6.40	...	3.223486	2.581647	2.576565	2.415222	
GSM2233624	1027	80	male	white	120	6.37	3.0	6	Probable	7.12	...	3.238530	2.634674	2.629828	2.450357	
GSM2233625	1039	85	male	white	315	6.90	0.0	2	Normal	0.00	...	3.255730	2.614901	2.718517	2.844469	

5 rows × 16396 columns

STEPS

Step 5

Assigning numeric values to different categories of patient samples for further processing.

```python
df3.cat.value_counts()

l = {'definite':3,'Possible':1,'Normal':0,'Probable':2}

df3.cat = [l[i] for i in df3.cat]
df3.head()
```

✓ 0.0s Python

	subject_id	age	sex	race	postmortem_interval	ph	rating	bnt	cat	anpd	...	217004_s_at	217005_at	217006_x_at	217007_s_at	21700
id_ref																
GSM2233621	1006	98	female	black	300	6.50	3.0	4	1	2.48	...	3.355867	2.613956	2.997592	2.725131	3.1
GSM2233622	1018	60	male	white	1725	6.76	0.0	0	0	0.00	...	3.178337	2.612671	2.616394	2.806919	3.5
GSM2233623	1023	92	female	white	115	6.70	3.0	6	3	6.40	...	3.223486	2.581647	2.576565	2.415222	3.6
GSM2233624	1027	80	male	white	120	6.37	3.0	6	2	7.12	...	3.238530	2.634674	2.629828	2.450357	3.3
GSM2233625	1039	85	male	white	315	6.90	0.0	2	0	0.00	...	3.255730	2.614901	2.718517	2.844469	3.5

5 rows × 16396 columns

05

PERFORMING GSEA

STEPS

Step 1

Reading and manipulating the data obtained after GEO2R Analysis.

```python
geo = pd.read_csv(r"GSE84422.top.table (2).tsv",sep='\t')   # READING DATA OBTAINED FROM GEO2R ANALYSIS
```

✓ 0.7s Python

```python
geo['rank'] = -(np.log10(geo['P.Value']))*geo['logFC']
geo['logp'] = -(np.log(geo['P.Value']))
geo.sort_values(by = "rank",inplace=True,ascending=False)
geo.head()
```

✓ 0.0s Python

	ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title	rank	logp
14	205150_s_at	2.050000e-09	1.380000e-12	7.26	17.97754	0.481	TRIL	TLR4 interactor with leucine rich repeats	5.704718	27.308938
10	205151_s_at	9.810000e-13	4.840000e-16	8.37	25.51129	0.372	TRIL	TLR4 interactor with leucine rich repeats	5.697238	35.264447
38	208151_x_at	1.810000e-06	3.170000e-09	6.02	10.67753	0.581	DDX17	DEAD-box helicase 17	4.937885	19.569534
20	205923_at	5.850000e-08	5.520000e-11	6.69	14.49161	0.464	RELN	reelin	4.759740	23.620058
11	210473_s_at	9.000000e-10	4.970000e-13	7.41	18.94058	0.357	ADGRA3	adhesion G protein-coupled receptor A3	4.392401	28.330186

STEPS

Step 2

Creating a new variable “findata” to store the final data that will be used as the ranked data for GSEA.

```
findata = geo[['Gene.symbol','rank']]
findata.head()
```

✓ 0.0s

	Gene.symbol	rank
14	TRIL	5.704718
10	TRIL	5.697238
38	DDX17	4.937885
20	RELN	4.759740
11	ADGRA3	4.392401

Step 3

Importing “gseapy” in Python and querying about all the libraries available, for the next steps

```
import gseapy as gp

gp.get_library_name()
```

✓ 2.8s

STEPS

Step 4

Performing GSEA using the “GO_Biological_Process_2023” library and printing the results

```python
rnk = gp.prerank(rnk = findata,gene_sets='GO_Biological_Process_2023',seed = 43)
```

```python
rnk.results
```

✓ 32.9s Python


```
{'Regulation Of Cytokine-Mediated Signaling Pathway (GO:0001959)': {'name': 'prerank',
  'es': 0.6597184948499459,
  'nes': 1.2317378117969253,
  'pval': 0.2119460500963391,
  'fdr': 0.48833589908066716,
  'fwerp': 1.0,
  'tag %': '9/33',
  'gene %': '11.47%',
  'lead_genes': 'CD24;CD74;IKBKB;AXL;SIGIRR;CYLD;MAPKAPK2;SYK;PTPN11',
  'matched_genes': 'CD24;CD74;IKBKB;AXL;SIGIRR;CYLD;MAPKAPK2;SYK;PTPN11;IRAK1;HIPK1;SPPL2B;TRIM44;SHARPIN;IKBKE;IL36RN;TNFAIP3;GAS6;VRK2;CASP8;SPATA2;PTPN2
  'hits': [490,
   688,
   1058,
```

STEPS

Step 5

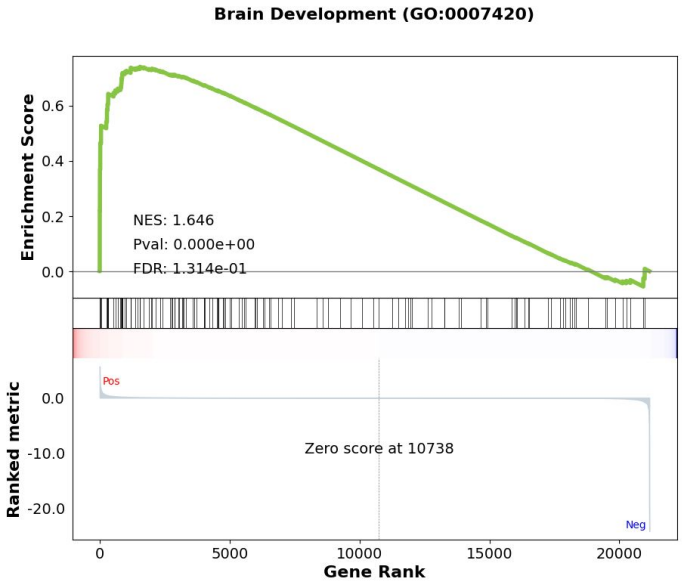
Sorting the results using the p-values and removing pathways having p-values >=0.05

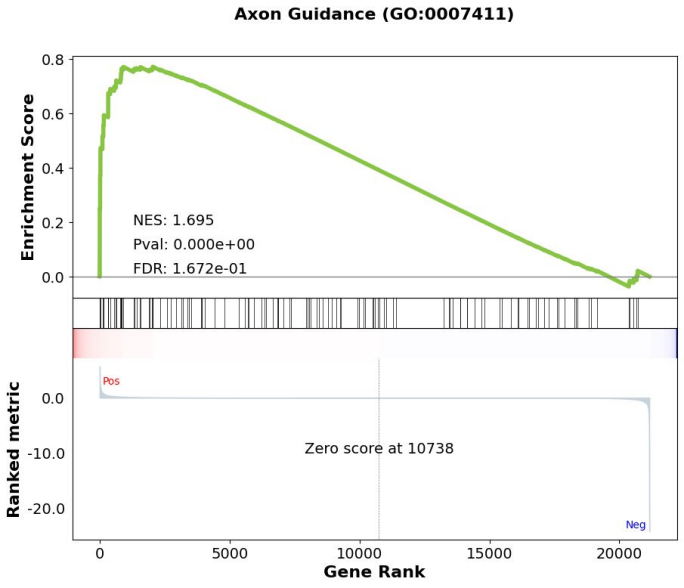
```
from gseapy.plot import gseaplot as gp

out =[]
for term in rnk.results:
    out.append([term,rnk.results[term]['pval'],rnk.results[term]['es'],rnk.results[term]['nes']])

out_df = pd.DataFrame(out, columns = ['Term','pval','es','nes']).sort_values('pval').reset_index(drop=True)

out_df = out_df[out_df.pval < 0.05]

for i in ["Brain Development (GO:0007420)","Axon Guidance (GO:0007411)","Central Nervous System Development (GO:0007417)",
          "Neuron Projection Guidance (GO:0097485)"]:
    axs = rnk.plot(terms = i, figsize = (10,8))
```

STEPS

Step 6

Plotting the GSEA results for some relevant pathways

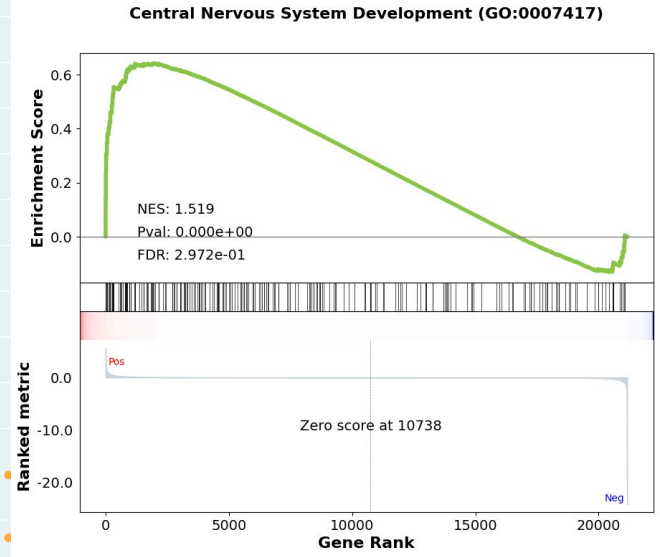


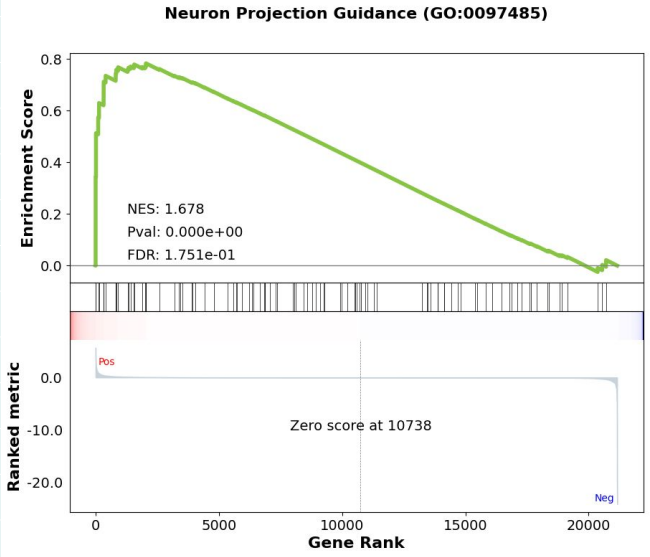


STEPS

Step 6

Plotting the GSEA results for some relevant pathways





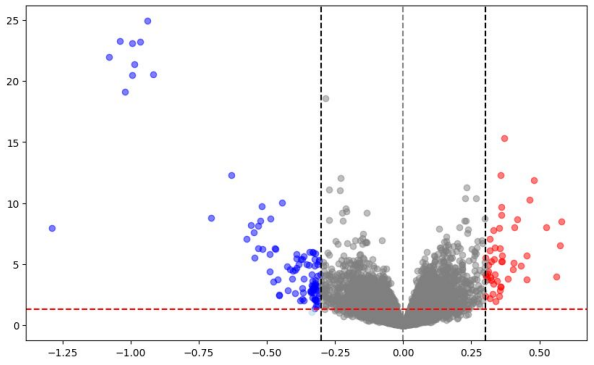
06

IDENTIFYING DEGs

IDENTIFYING DEGs

- We set a ***p value threshold of 0.05***, and a ***fold change of threshold of 2*** in order to narrow down the genes we had
- After the required calculations, we ended up with *121 genes* in total
- We then performed ***PCA*** in order to reduce the number of components, and finally, we ended up with *50 principal components* (features)
- Since our ***sample size*** is *542*, we felt this would a decent enough number, one which would ***not*** lead to ***too much overfitting.***

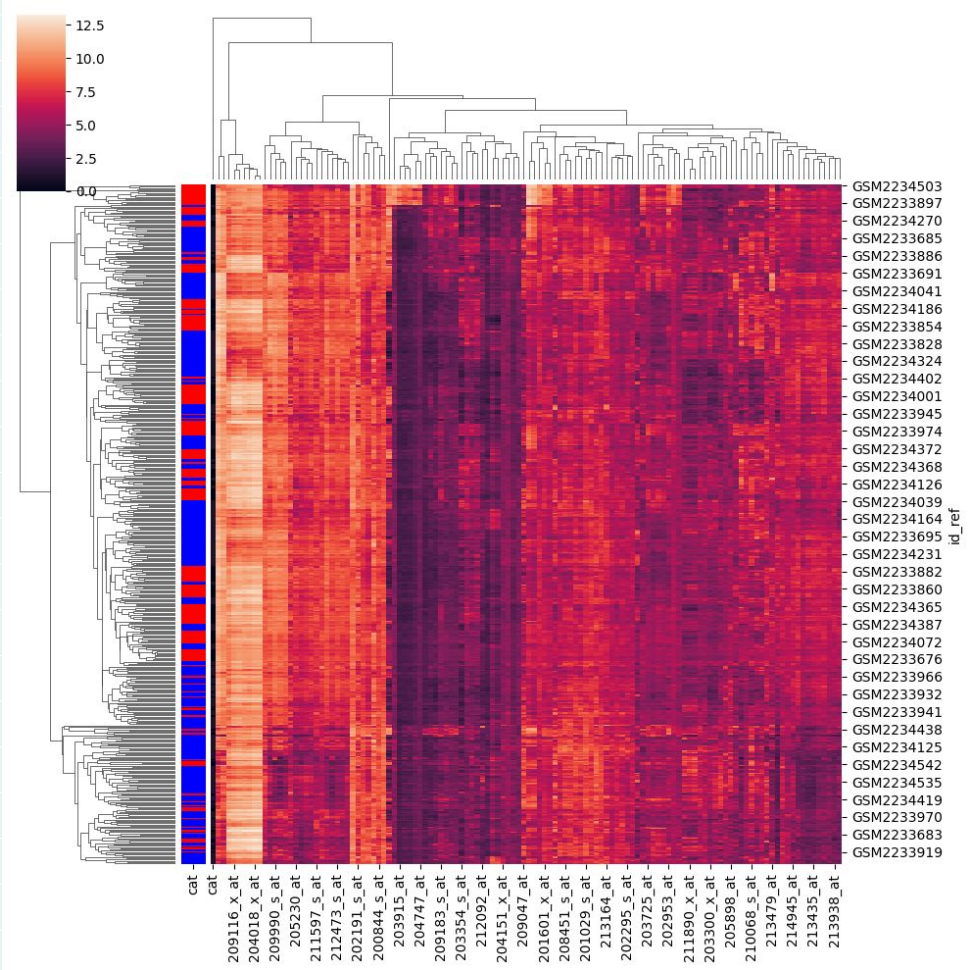
```python
# Finding upregulated and downregulated genes


geo.rename({'P.Value':'p-value'},axis = 1,inplace=True)
geo.rename({'logFC':'log2_fold_change'},axis = 1,inplace=True)
geo['rank'] = -(np.log10(geo['p-value']))*geo['log2_fold_change']

geo['logp'] = -(np.log(geo['p-value']))
geo.head()

fold_change_threshold, p_value_threshold = np.log10(2), 0.05

# Identify differentially expressed genes
upregulated = geo[(geo['log2_fold_change'] > fold_change_threshold) & (geo['p-value'] < p_value_threshold)]
downregulated = geo[(geo['log2_fold_change'] < -fold_change_threshold) & (geo['p-value'] < p_value_threshold)]
no_change = geo[abs(geo['log2_fold_change']) <= fold_change_threshold]

# Create volcano plot
plt.figure(figsize=(10, 6))

# Plot genes with significant upregulation
plt.scatter(upregulated['log2_fold_change'], -np.log10(upregulated['p-value']), color='red', label='Upregulated (significant)', alpha=0.5)

# Plot genes with significant downregulation
plt.scatter(downregulated['log2_fold_change'], -np.log10(downregulated['p-value']), color='blue', label='Downregulated (significant)', alpha=0.5)

# Plot genes with no significant change
plt.scatter(no_change['log2_fold_change'], -np.log10(no_change['p-value']), color='gray', label='No change', alpha=0.5)

# Plot genes with upregulation but not significant
plt.scatter(geo[(geo['log2_fold_change'] > fold_change_threshold) & (geo['p-value'] >= p_value_threshold)]['log2_fold_change'],
            -np.log10(geo[(geo['log2_fold_change'] > fold_change_threshold) & (geo['p-value'] >= p_value_threshold)]['p-value']),
            color='salmon', label='Upregulated (not significant)', alpha=0.5)

# Plot genes with downregulation but not significant
plt.scatter(geo[(geo['log2_fold_change'] < -fold_change_threshold) & (geo['p-value'] >= p_value_threshold)]['log2_fold_change'],
            -np.log10(geo[(geo['log2_fold_change'] < -fold_change_threshold) & (geo['p-value'] >= p_value_threshold)]['p-value']),
            color='lightblue', label='Downregulated (not significant)', alpha=0.5)

# sns.scatterplot(x = 'logFC',y = 'logp',data=geo)

plt.axhline(y=-np.log10(0.05), color='red', linestyle='--')
plt.axvline(x = np.log10(2), color='black', linestyle='--')
plt.axvline(x = -np.log10(2), color='black', linestyle='--')
plt.axvline(x=0, color='grey', linestyle='--')
```



CLUSTER PLOT

- Here, red represents normal, and blue represents definite AD, on the left.

```
lut = dict(zip(y.unique(), "rbg"))
row_colors = y.map(lut)
print(row_colors)
sns.clustermap(X2,row_colors=row_colors)
```



07

TRAINING ML MODEL

PRE-PROCESSING

Step 1

Importing the necessary modules

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.decomposition import PCA
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.mixture import GaussianMixture
from sklearn.metrics import accuracy_score,f1_score,recall_score,precision_score
from sklearn.preprocessing import StandardScaler
```

TRAINING ML MODEL

- We used *70%* of the data to ***train the model***, and *30%* of the data to ***test the model***.
- Standard Scaling was also applied as a preprocessing step
- We trained a variety of ML models, such as ***Naive Bayes, KNN, Logistic Regression*** etc..

```
gnb = GaussianNB()
knn = KNeighborsClassifier()
logreg = LogisticRegression(solver='saga',random_state=41)
rf = RandomForestClassifier(random_state=41,n_estimators=100,max_features=400)
abc = AdaBoostClassifier(random_state=41, n_estimators=250)
gbc = GradientBoostingClassifier(random_state=41,max_features=60)
svc = SVC(random_state=41)

models = [gnb,knn,logreg,rf,abc,svc]

d={}

ss = StandardScaler()

ss.fit_transform(X,y)


pca = PCA(n_components=50)

X = pca.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=41)
```

08

VALIDATION

VALIDATION

- After training all the models, we ***evaluated the accuracy*** of each of the models and finally, chose ***SVC Model*** as the appropriate ML model for the purpose of ***detecting AD in people***, given their expression levels of the required genes

- We have even verified the accuracy of our chosen SVC Model.

```
for i in models:
    print(i,": ",end=" ")
    i.fit(X_train,y_train)
    preds = i.predict(X_test)

    acc = accuracy_score(y_test,preds)
    prec = precision_score(y_test,preds)
    rec = recall_score(y_test,preds)
    f1 = f1_score(y_test,preds)

    results.append([i.__class__.__name__,acc,prec,rec,f1])


    print("Accuracy: ",acc)

results = pd.DataFrame(columns = ['Model','Accuracy','Precision','Recall','F1 Score'],data=results)
```

```
from sklearn.model_selection import cross_val_score

resu = cross_val_score(estimator=svc, X = X, y = y)
resu
```

```
array([0.95412844, 0.90825688, 0.90740741, 0.93518519, 0.82407407])
```

Some performance metrics

Sensitivity/ Recall/ True Positive rate - Used to evaluate how well a model can correctly identify positive instances out of all the actual positive instances present in the dataset.

Sensitivity = true positive / (true positives + false negatives)

Accuracy - Measures the proportion of correct predictions out of all the predictions.

Precision - It evaluates the accuracy of the positive predictions made by a model.

Precision = true positive / (true positives + false positives)

F-1 score - It combines both precision and recall into a single value. It's particularly useful when there's an imbalance between the classes.

F1 score = 2 * Precision * Recall / (Precision + Recall)

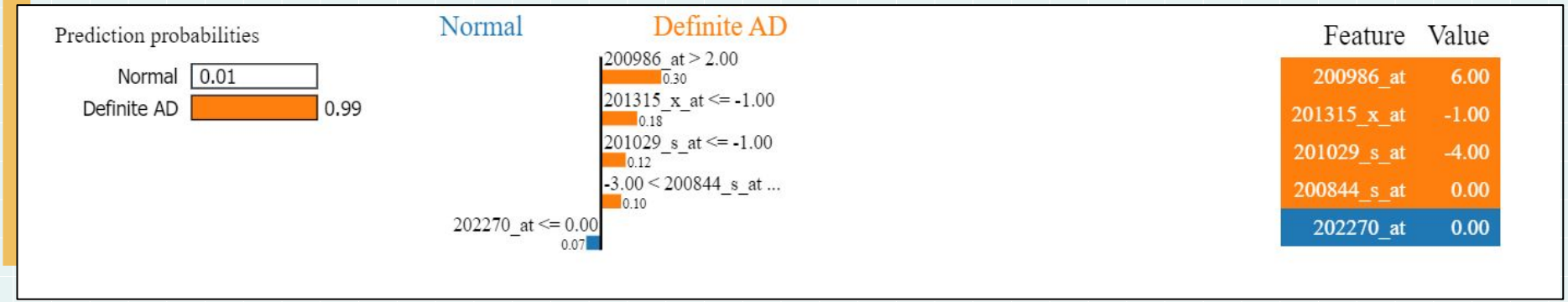
RESULTS

Model	Accuracy	Precision	Recall	F1 Score
GaussianNB	0.840490798	0.825242718	0.913978495	0.867346939
KNeighborsClassifier	0.926380368	0.935483871	0.935483871	0.935483871
LogisticRegression	0.889570552	0.878787879	0.935483871	0.90625
RandomForestClassifier	0.846625767	0.833333333	0.913978495	0.871794872
AdaBoostClassifier	0.877300613	0.87628866	0.913978495	0.894736842
SVC	0.932515337	0.901960784	0.989247312	0.943589744

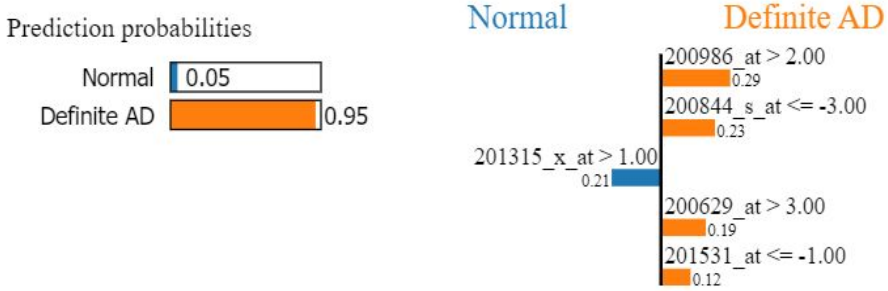
Using LIME for Explainability

We used ***LIME(Locally Interpretable Model-Agnostic Explanations)*** on the same test sample for both KNN and SVC in order to see which features influenced their predictions the most.

KNN


Prediction probabilities
Normal
0.01
Definite AD
0.99
Normal
Definite AD
200986_at > 2.00
0.30
201315_x_at <= -1.00
0.18
201029_s_at <= -1.00
0.12
-3.00 < 200844_s_at ...
0.10
202270_at <= 0.00
0.07
Feature
Value
200986_at
6.00
201315_x_at
-1.00
201029_s_at
-4.00
200844_s_at
0.00
202270_at
0.00

SVC



Feature	Value
200986_at	6.00
200844_s_at	-8.00
201315_x_at	6.00
200629_at	6.00
201531_at	-1.00

CONCLUSION

In conclusion, our approach integrates advanced bioinformatics tools like ***NCBI GEO2R*** for identifying differentially expressed genes associated with Alzheimer's disease. Utilizing techniques such as ***Volcano plots, clustered plots, and statistical tests***, we try to figure out differences between normal and diseased gene expression profiles. This rigorous analysis forms the foundation for training a ***predictive model*** capable of distinguishing Alzheimer's disease from normal gene expression patterns.

THANK YOU

Group - 11

~ ***ML model and GSEA -*** *Debjit & Vijval*

~ ***Statistical Testing (Using Python) -*** *Abhishek & Kartikeya*

~ ***GEO2R Analysis -*** *Masood*

~ ***Data collection and Cleansing -*** *Anish*

~ ***Presentation Quality -*** *Collective*