Detecting Pre-Symptomatic Cognitive Impairment through Epigenetic Biomarkers and Machine Learning for Early Intervention

William Guo - Northwood High School - Irvine, CA, 926200

Background:

ALZHEIMER'S DISEASE

- Progressive type of dementia that affects **memory** and other **important cognitive functions**.
- While treatment and medication can temporarily help with symptoms, **there is no cure**.

STATISTICS

5.8 MILLION
cases in 2020 (US)

▼

13.8 MILLION
cases by 2050 (US)

1 in 3
Seniors die with Alzheimer's or another form of dementia.

500,000 Americans die every year because of Alzheimer's.

DIAGNOSIS ISSUES

There is no single test that can definitively determine whether a person has Alzheimer's Disease.

▶

Diagnosis for Alzheimer's often comes **too late** and the damage already done is **irreversible.**

▶

Confirmatory diagnosis such as PET scans are **expensive** and CSF procedures are **invasive.**

Background:

RECENT FINDINGS

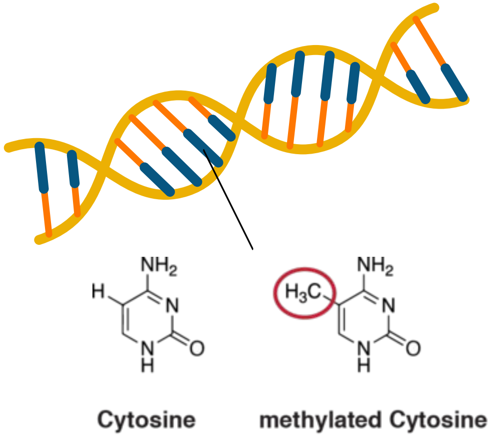
Studies have suggested that epigenetic mechanisms like **DNA methylation** play a role in the **development of Alzheimer's**.

↓

These epigenetic mechanisms may hold **the key** to the potential creation of a pre-clinical **prediction model**.

DNA METHYLATION

DNA methylation is a **vital epigenetic process** by which a methyl group (CH3) is added to DNA, typically affecting **gene expression**.



SPECULATION

Specific CpG (Cytosine - Phosphate - Guanine) sites in the promoter region become **abnormally methylated**

↓

Possibly **leads to the development** of Alzheimer's Disease

Objective:

HYPOTHESIS

Can DNA Methylation be used as a biomarker to diagnose Alzheimer's Disease?

OBJECTIVE

Create a method of detecting pre-symptomatic cognitive impairment that is simple, inexpensive, and minimally invasive

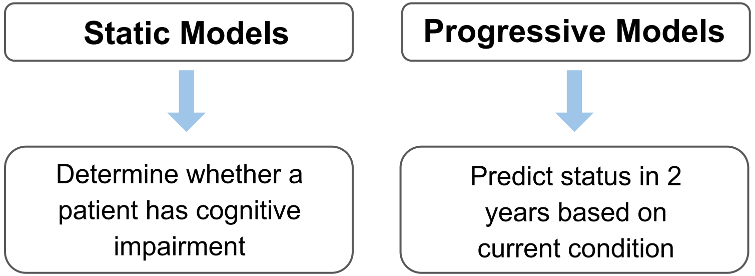
DEFINITION REFRESHER

➢ **Biomarker:** A measurable substance that is indicative of some phenomenon

➢ **DNA Methylation:** A biological process where methyl groups are added to DNA

➢ **Alzheimer's Disease:** A common type of dementia that affects memory

GENERAL DESIGN GOAL



Static Models

Determine whether a patient has cognitive impairment

Progressive Models

Predict status in 2 years based on current condition

Background Objective Materials Procedure Results Application Conclusion Acknowledgements

Materials:

DATASETS

GEO (Gene Expression Omnibus):

- ➢ GSE153712 (Peripheral Blood)
  - ➢ 832,602 CpG Sites
  - ➢ 161 AD (Alzheimer’s Disease)
  - ➢ 94 MCI (Mild Cognitive Impairment)
  - ➢ 471 ND (No Disease)
- ➢ GSE156984 (Brain):
  - ➢ 728,553 CpG Sites
  - ➢ 127 AD (Alzheimer’s Disease)
  - ➢ 117 ND (No Disease)

ADNI (University of Southern California):

- ➢ Dataset (Peripheral Blood)
  - ➢ 865859 CpG Sites
  - ➢ 400 AD (Alzheimer’s Disease)
  - ➢ 895 MCI (Mild Cognitive Impairment)
  - ➢ 610 ND (No Disease)

*Samples (ADNI) followed disease progression

PROGRAMS

Anaconda 3 (Python Language):

- ➢ Built ML Models using scikit-learn (Jupyter Notebook)

RStudio (R Language):

- ➢ Format datasets

Linux (Amazon Web Services):

- ➢ Download and process datasets

WinSCP:

- ➢ Transfer files between Linux and personal laptop

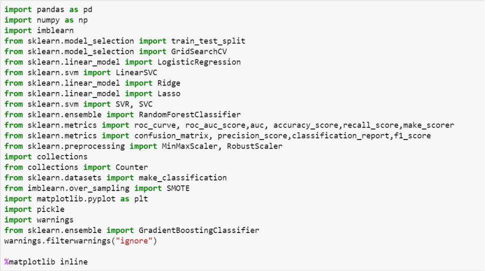
Microsoft Excel:

- ➢ Observe data and perform t_test to get p-value

MACHINE LEARNING

Machine Learning Algorithms:

- ➢ SVC (Support Vector Classifier)
- ➢ LogisticRegression
- ➢ RandomForest
- ➢ GradientBoostingClassifier



```
import pandas as pd
import numpy as np
import imblearn
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.svm import SVR, SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_curve, roc_auc_score,auc, accuracy_score,recall_score,make_scorer
from sklearn.metrics import confusion_matrix, precision_score,classification_report,f1_score
from sklearn.preprocessing import MinMaxScaler, RobustScaler
import collections
from collections import Counter
from sklearn.datasets import make_classification
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
import pickle
import warnings
from sklearn.ensemble import GradientBoostingClassifier
warnings.filterwarnings("ignore")

%matplotlib inline
```

* Import Data Wrangling Libraries and Machine Learning Algorithms

Procedure:

STATIC MODELS

1 Download Datasets and other Important Files

- ➢ Use wget in Linux to download GSE156984 dataset and Series Matrix File
- ➢ Use wget to download GSE153712 Series Matrix File and normalized value file using load() in R

2 Process and Format Data

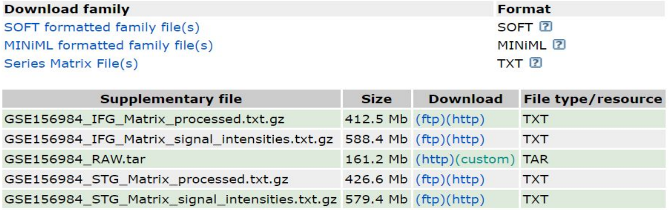
- ➢ Unzip files in Linux and format each file
- ➢ Extract sample ID’s and disease status from Series Matrix file for each dataset
- ➢ Calculate cell composition of samples (GSE153712)

3 Transfer Files to Personal Computer

- ➢ Use WinSCP to transfer all files need from Linux to personal computer
- ➢ Attach .csv to end of the filenames, which allows it to be opened in Microsoft Excel

PICTURES

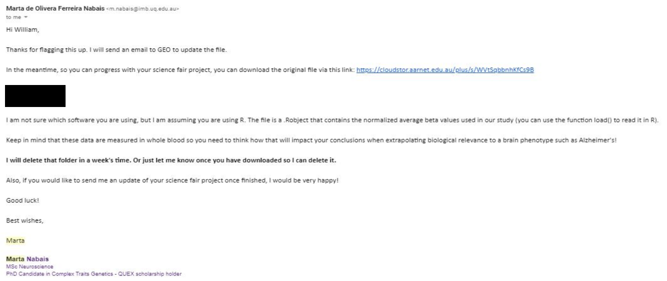
Figure 1: *GSE156984 Downloadable Files*



Download family	Format
SOFT formatted family file(s)	SOFT ?
MINiML formatted family file(s)	MINiML ?
Series Matrix File(s)	TXT ?

Supplementary file	Size	Download	File type/resource
GSE156984_IFG_Matrix_processed.txt.gz	412.5 Mb	(ftp)(http)	TXT
GSE156984_IFG_Matrix_signal_intensities.txt.gz	588.4 Mb	(ftp)(http)	TXT
GSE156984_RAW.tar	161.2 Mb	(http)(custom)	TAR
GSE156984_STG_Matrix_processed.txt.gz	426.6 Mb	(ftp)(http)	TXT
GSE156984_STG_Matrix_signal_intensities.txt.gz	579.4 Mb	(ftp)(http)	TXT

Figure 2: *GSE153712 File Received from Dr. Marta Nabais*



Marta de Olivera Ferreira Nabais <m.nabais@imb.uq.edu.au>
to me

Hi William,

Thanks for flagging this up. I will send an email to GEO to update the file.

In the meantime, so you can progress with your science fair project, you can download the original file via this link: https://cloudstor.aarnet.edu.au/plus/s/WVtSqbbnhKfCs9B

[redacted]

I am not sure which software you are using, but I am assuming you are using R. The file is a .Robject that contains the normalized average beta values used in our study (you can use the function load() to read it in R).

Keep in mind that these data are measured in whole blood so you need to think how that will impact your conclusions when extrapolating biological relevance to a brain phenotype such as Alzheimer's!

I will delete that folder in a week's time. Or just let me know once you have downloaded so I can delete it.

Also, if you would like to send me an update of your science fair project once finished, I would be very happy!

Good luck!

Best wishes,

Marta

Marta Nabais
MSc Neuroscience
PhD Candidate in Complex Traits Genetics - QUEX scholarship holder

Procedure:

STATIC MODELS (cont.)

4

Find Important CpG Sites

- ➢ Attach status and samples to datasets in Excel
- ➢ Sort files by status and perform t_test. For GSE153712, do t_test in terms of different status combinations (AD vs. ND, AD vs. MCI, MCI vs. ND)
- ➢ Record sites with a p-value under 1E-5
- ➢ Find similar sites between GSE156984, AD vs. ND, AD vs. MCI, and MCI vs. ND site list

5

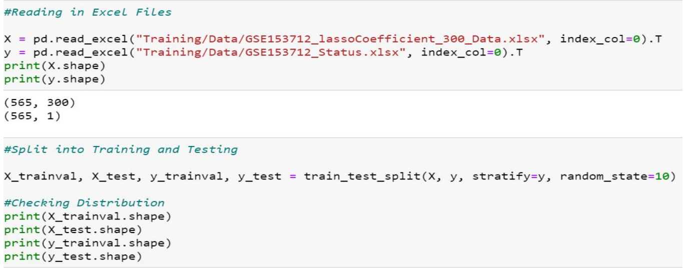
Create Machine Learning Models

- ➢ Create 3 models using 4 machine learning algorithms with only data from MCI and ND
  - ➢ 1) 6 similar sites (1E-5)
  - ➢ 2) Sites with p-value < 1E-5 (594)
  - ➢ 3) Collection of 5 models with different amounts of sites (1E-5) found using lasso

*Cell Composition and Gender were also considered in the ML Models

PICTURES

Figure 3: *Reading in Lasso_300probes*

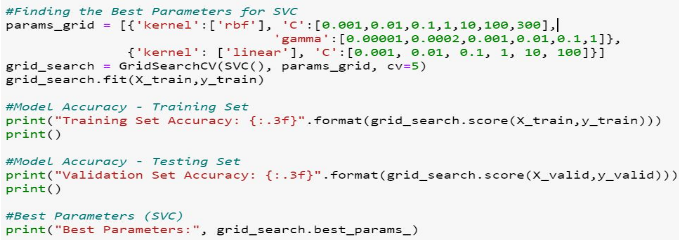


```
#Reading in Excel Files

X = pd.read_excel("Training/Data/GSE153712_lassoCoefficient_300_Data.xlsx", index_col=0).T
y = pd.read_excel("Training/Data/GSE153712_Status.xlsx", index_col=0).T
print(X.shape)
print(y.shape)
```

```
(565, 300)
(565, 1)
```

```
#Split into Training and Testing

X_trainval, X_test, y_trainval, y_test = train_test_split(X, y, stratify=y, random_state=10)

#Checking Distribution
print(X_trainval.shape)
print(X_test.shape)
print(y_trainval.shape)
print(y_test.shape)
```

Figure 4: *Finding Best Parameters for SVC*



```
#Finding the Best Parameters for SVC
params_grid = [{'kernel':['rbf'], 'C':[0.001,0.01,0.1,1,10,100,300],
                                    'gamma':[0.00001,0.0002,0.001,0.01,0.1,1]},
                {'kernel': ['linear'], 'C':[0.001, 0.01, 0.1, 1, 10, 100]}]
grid_search = GridSearchCV(SVC(), params_grid, cv=5)
grid_search.fit(X_train,y_train)

#Model Accuracy - Training Set
print("Training Set Accuracy: {:.3f}".format(grid_search.score(X_train,y_train)))
print()

#Model Accuracy - Testing Set
print("Validation Set Accuracy: {:.3f}".format(grid_search.score(X_valid,y_valid)))
print()

#Best Parameters (SVC)
print("Best Parameters:", grid_search.best_params_)
```

Procedure:

PROGRESSIVE MODELS

1 Download Datasets and other Important Files

- ➢ Use R program minfi to download and convert original .idat file to beta (methylation) file
- ➢ Get annotation and merge Excel file from ADNI

2 Process and Format Data

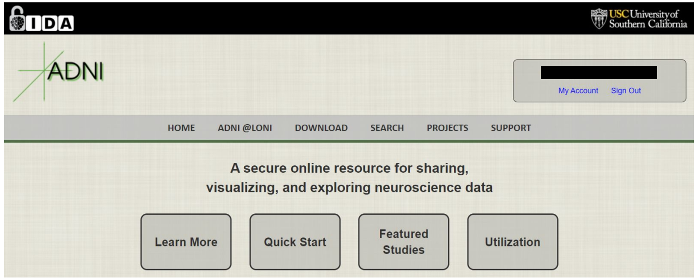
- ➢ Use script to split beta file, format each file through R, and calculate cell composition of samples

3 Format and Evaluate Information Excel File

- ➢ Combine annotation and merge file using a script that matches RID # and sample examination date
- ➢ Find 4 different groups of samples based on status
  - ➢ 1) ND → ND (in 2 years)
  - ➢ 2) ND → MCI (in 2 years)
  - ➢ 3) MCI → MCI (in 2 years)
  - ➢ 4) MCI → AD (in 2 years)

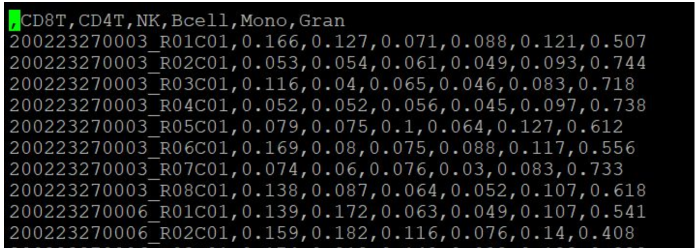
PICTURES

Figure 5: *ADNI Website* *(Access to Data Granted)*



A secure online resource for sharing, visualizing, and exploring neuroscience data

Figure 6: *ADNI Cell Composition File*



```
,CD8T,CD4T,NK,Bcell,Mono,Gran
200223270003_R01C01,0.166,0.127,0.071,0.088,0.121,0.507
200223270003_R02C01,0.053,0.054,0.061,0.049,0.093,0.744
200223270003_R03C01,0.116,0.04,0.065,0.046,0.083,0.718
200223270003_R04C01,0.052,0.052,0.056,0.045,0.097,0.738
200223270003_R05C01,0.079,0.075,0.1,0.064,0.127,0.612
200223270003_R06C01,0.169,0.08,0.075,0.088,0.117,0.556
200223270003_R07C01,0.074,0.06,0.076,0.03,0.083,0.733
200223270003_R08C01,0.138,0.087,0.064,0.052,0.107,0.618
200223270006_R01C01,0.139,0.172,0.063,0.049,0.107,0.541
200223270006_R02C01,0.159,0.182,0.116,0.076,0.14,0.408
```

Procedure:

PROGRESSIVE MODELS (cont.)

4 Discover Significant CpG Sites

- ➢ Find DNA methylation values present in the beta files of each sample in every status group
- ➢ Use the R library matrixTests to perform t_tests
  * (ND → ND vs ND → MCI) and (MCI → MCI vs. MCI → AD)
- ➢ Create list of probes that a p-value under 1E-5 for ND vs MCI and MCI vs AD and find/record the methylation values of these specific probes

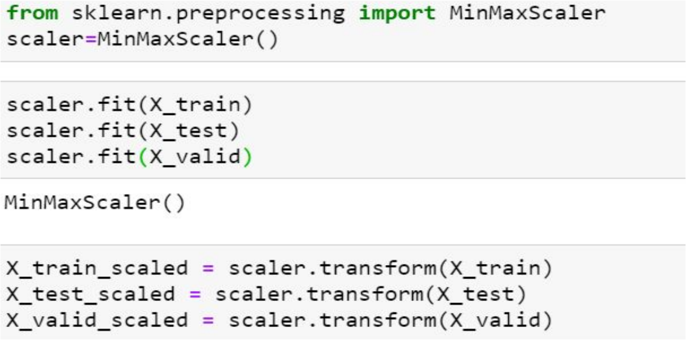
5 Create Machine Learning Models

- ➢ Using 4 machine learning algorithms, create 2 models with one having data from (**ND**→ND vs. **ND**→MCI) and the other model having (**MCI** →MCI vs **MCI**→AD).
  - ➢ 1) List of 79 sites (1E-5: ND→ND vs. MCI→AD)
  - ➢ 2) List of 80 sites (1E-5: MC →MCI vs. MCI→AD)

*Cell Composition and Gender were also considered in the ML Models

PICTURES

Figure 6: *Scaling Data using MinMaxScaler*



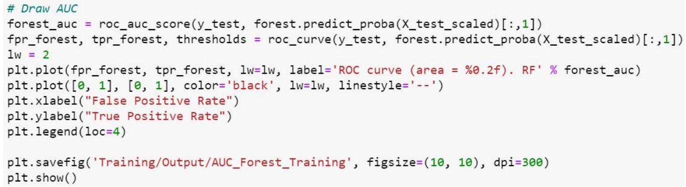
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
```

```
scaler.fit(X_train)
scaler.fit(X_test)
scaler.fit(X_valid)
```

```
MinMaxScaler()
```

```
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
X_valid_scaled = scaler.transform(X_valid)
```

Figure 7: *Code for AUC (RandomForest)*

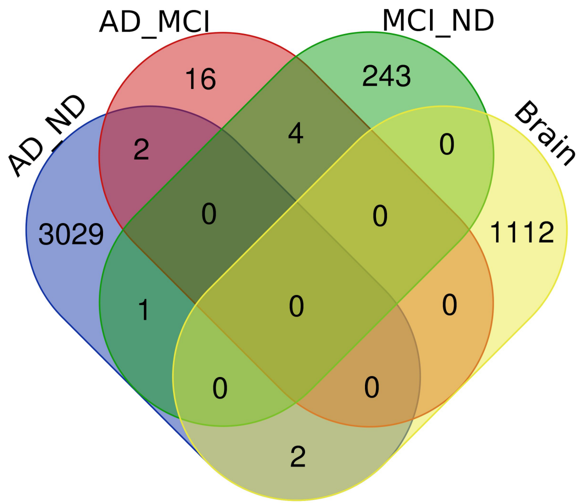


```
# Draw AUC
forest_auc = roc_auc_score(y_test, forest.predict_proba(X_test_scaled)[:,1])
fpr_forest, tpr_forest, thresholds = roc_curve(y_test, forest.predict_proba(X_test_scaled)[:,1])
lw = 2
plt.plot(fpr_forest, tpr_forest, lw=lw, label='ROC curve (area = %0.2f). RF' % forest_auc)
plt.plot([0, 1], [0, 1], color='black', lw=lw, linestyle='--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend(loc=4)

plt.savefig('Training/Output/AUC_Forest_Training', figsize=(10, 10), dpi=300)
plt.show()
```

Results:

SIGNIFICANT SITES (p-value < 1E-5): GSE156984 Brain & GSE153712 Blood

Figure 8: *Venn Diagram of GSE156984 and GSE153712 Probes (1E-5)*



GSE156984 (Brain):

- ➢ 1114 CpG Sites

GSE153712 (MCI vs. ND):

- ➢ 248 CpG Sites

GSE153712 (AD vs. ND):

- ➢ 3034 CpG Sites

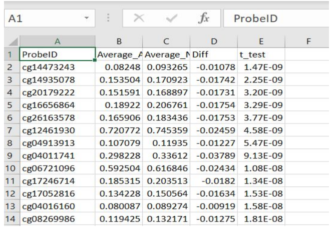
GSE153712 (AD vs. MCI):

- ➢ 22 CpG Sites

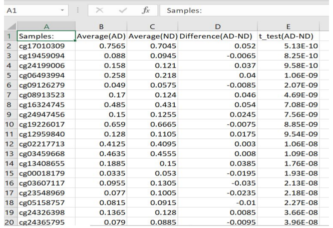
GSE156984 & GSE153712 Comparison:

- ➢ 8 Shared CpG Sites

Figure 9: *GSE156984 & GSE153712 Probe Sites*



	A	B	C	D	E	F
1	ProbeID	Average_A	Average_N	Diff	t_test	
2	cg14473243	0.08248	0.093265	-0.01078	1.47E-09	
3	cg14935078	0.153504	0.170923	-0.01742	2.25E-09	
4	cg20179222	0.151591	0.168897	-0.01731	3.20E-09	
5	cg16656864	0.18922	0.206761	-0.01754	3.29E-09	
6	cg26163578	0.165906	0.183436	-0.01753	3.77E-09	
7	cg12461930	0.720772	0.745359	-0.02459	4.58E-09	
8	cg04913913	0.107079	0.11935	-0.01227	5.47E-09	
9	cg04011741	0.298228	0.33612	-0.03789	9.13E-09	
10	cg06721096	0.592504	0.616846	-0.02434	1.08E-08	
11	cg17246714	0.185315	0.203513	-0.0182	1.34E-08	
12	cg17052816	0.134228	0.150564	-0.01634	1.53E-08	
13	cg04016160	0.080087	0.089274	-0.00919	1.58E-08	
14	cg08269986	0.119425	0.132171	-0.01275	1.81E-08	



	A	B	C	D	E
1	Samples:	Average(AD)	Average(ND)	Difference(AD-ND)	t_test(AD-ND)
2	cg17010309	0.7565	0.7045	0.052	5.13E-10
3	cg19459094	0.088	0.0945	-0.0065	8.25E-10
4	cg24199006	0.158	0.121	0.037	9.58E-10
5	cg06493994	0.258	0.218	0.04	1.06E-09
6	cg09126279	0.049	0.0575	-0.0085	2.07E-09
7	cg08913523	0.17	0.124	0.046	4.69E-09
8	cg16324745	0.485	0.431	0.054	7.08E-09
9	cg24947456	0.15	0.1255	0.0245	7.56E-09
10	cg19226017	0.659	0.6665	-0.0075	8.85E-09
11	cg12959840	0.128	0.1105	0.0175	9.54E-09
12	cg02217713	0.4125	0.4095	0.003	1.06E-08
13	cg03459668	0.4635	0.4555	0.008	1.09E-08
14	cg13408655	0.1885	0.15	0.0385	1.76E-08
15	cg00018179	0.0335	0.053	-0.0195	1.93E-08
16	cg03607117	0.0955	0.1305	-0.035	2.13E-08
17	cg23548969	0.077	0.1005	-0.0235	2.18E-08
18	cg05158757	0.0815	0.0915	-0.01	2.27E-08
19	cg24326398	0.1365	0.128	0.0085	3.66E-08
20	cg24365795	0.079	0.0885	-0.0095	3.96E-08

Results:

6 SIGNIFICANT SITES SELECTED FOR STATIC MODEL #1

19 Sites (8 similar sites, Top 10 MCI-ND, cg07773593):

AD vs. MCI + AD vs. ND = cg07347869, cg05234135

AD vs. ND + MCI vs. ND = **cg04876500**

AD vs. MCI + MCI vs.ND = cg14706655, cg17422516, cg09234764, **cg09044631**

AD vs. ND + Brain: cg09559780, cg06532212

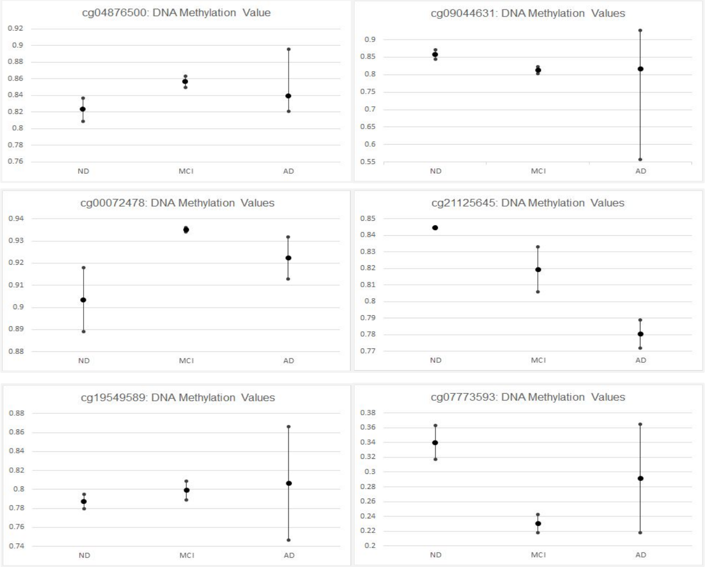
Another Probe (Featured in other studies): **cg07773593**

MCI vs. ND (Top 10 Sites)

- cg07241675, **cg00072478**, **cg21125645**, cg22471641, cg20630239, cg10948751, cg03700990, cg01747278, cg00862028, **cg19549589**

*Selected Sites based on Methylation Change Correlation from ND Disease to Mild Cognitive Impairment and Alzheimer's (Fig. 10)

Figure 10: *Average-Min-Max Graphs of 6 Probes*



Results:

SIGNIFICANT SITES (p-value < 1E-5) SELECTED FOR PROGRESSIVE MODELS

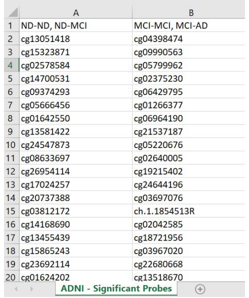
ND → ND vs. ND → MCI:

➢ 79 CpG Sites

MCI → MCI vs. MCI → AD:

➢ 80 CpG Sites

Figure 11: *ADNI Sites*



	A	B
1	ND-ND, ND-MCI	MCI-MCI, MCI-AD
2	cg13051418	cg04398474
3	cg15323871	cg09990563
4	cg02578584	cg05799962
5	cg14700531	cg02375230
6	cg09374293	cg06429795
7	cg05666456	cg01266377
8	cg01642550	cg06964190
9	cg13581422	cg21537187
10	cg24547873	cg05220676
11	cg08633697	cg02640005
12	cg26954114	cg19215402
13	cg17024257	cg24644196
14	cg20737388	cg03697076
15	cg03812172	ch.1.1854513R
16	cg14168690	cg02042585
17	cg13455439	cg18721956
18	cg15865243	cg03967020
19	cg23692114	cg22680668
20	cg01624202	cg13518670

ADNI - Significant Probes

Figure 12: Top 20 Sites for the ND to MCI Conversion in Two Years

The DAVID functional analysis shows association with neurological disease.

Probe ID	Methylation MCI in 2 yrs	Methylation ND in 2 yrs	Difference	P value	Chromo-some	Position	Gene	Biological Pathway	Disease Class
cg13051418	0.143	0.084	0.059	2.25E-07	12	115139149			
cg15323871	0.420	0.350	0.070	5.29E-07	14	60043906	C14orf38		
cg02578584	0.866	0.913	-0.047	1.43E-06	8	8325700			
cg14700531	0.598	0.540	0.059	1.68E-06	8	733412			
cg09374293	0.448	0.558	-0.110	1.92E-06	21	48081242	PRMT2	Transcription regulation	**Neurological**
cg05666456	0.358	0.273	0.085	2.57E-06	4	170214340			
cg01642550	0.336	0.296	0.040	5.6E-06	16	89098327			
cg13581422	0.254	0.332	-0.078	5.97E-06	3	130236522			
cg24547873	0.488	0.541	-0.053	6.32E-06	1	17086558	MST1P9	Macrophage stimulation	
cg26954114	0.573	0.479	0.094	7.55E-06	15	96838120			
cg17024257	0.706	0.653	0.053	7.94E-06	3	171528758	PLD1	Signal transduction	Metabolic
cg20737388	0.533	0.648	-0.115	8E-06	11	73668626	DNAJB13	HSP40 co-chaperone	**Neurological**
cg03812172	0.531	0.707	-0.176	1E-05	7	44184403	GCK	Glycogen biosynthesis	**Neurological.** Immune
cg14168690	0.317	0.219	0.098	1.1E-05	21	32819053	TIAM1	Protein localization	**Neurological.** Metabloic
cg13455439	0.399	0.487	-0.088	1.53E-05	11	69934128	ANO1	Chloride channel	
cg15865243	0.703	0.632	0.072	1.62E-05	12	34496342			
cg23692114	0.447	0.413	0.034	1.69E-05	2	75154873	LINC01291		
cg13649415	0.715	0.755	-0.039	1.74E-05	3	46621737	TDGF1	Cell differentiation	Metabolic
cg00240732	0.432	0.375	0.057	1.76E-05	7	70923396	WBSCR17	Membrane trafficking	**Neurological.** Immune
cg08707819	0.351	0.311	0.039	1.83E-05	14	103059391	RCOR1	Neural cell differentiation	

Results:

BEST STATIC MODEL (Detect Current MCI)

Figure 13: *Lasso_300_probes Model Accuracy*

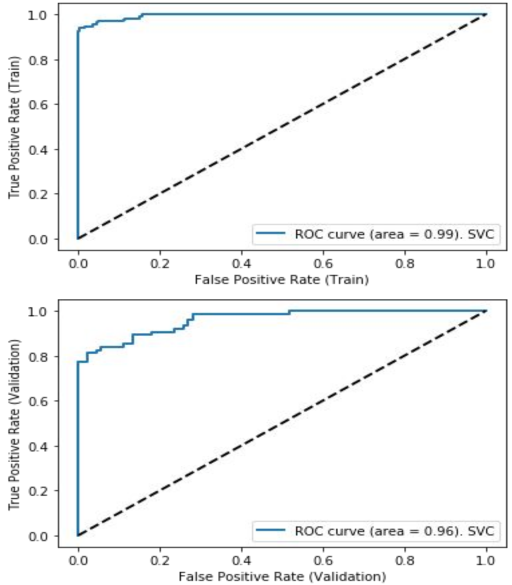
Model (Algorithms):	Training Accuracy:	Validation Accuracy:	Testing Accuracy:
SVC	**95.9%**	**89.6%**	**78.9%**
LogisticRegression	97.8%	82.3%	71.1%
RandomForest	97.5%	84.8%	70.4%
GradientBoosting	96.1%	82.9%	67.6%

Training: Train the model (Sees data and learns from it)

Validation: Evaluate the given model. Helps fine tune the algorithm’s hyperparameters

Testing: Unbiased evaluation of the final model

Figure 14: *Area Under the Curve (SVC)*



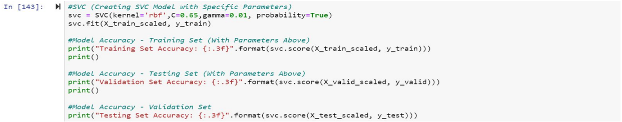
Results:

PROGRESSIVE MODEL (Predict ND to MCI Conversion in 2 Years)

Figure 15: *ND → ND vs. ND → MCI Model Accuracy*

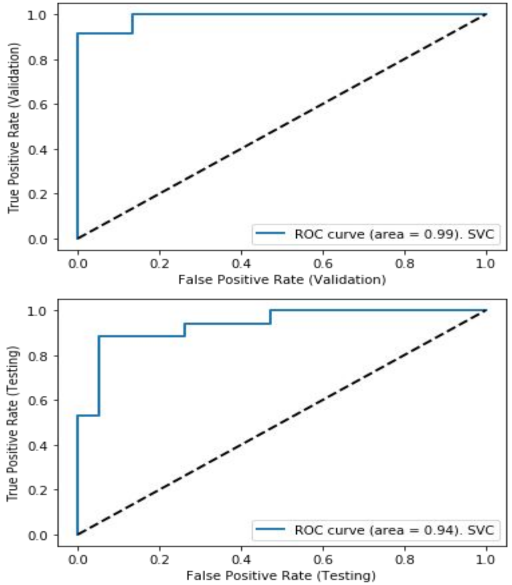
Model (Algorithms):	Training Accuracy:	Validation Accuracy:	Testing Accuracy:
SVC	**98.7%**	**96.3%**	**86.1%**
LogisticRegression	98.7%	96.3%	83.3%
RandomForest	97.5%	81.5%	77.8%
GradientBoosting	94.9%	81.5%	66.7%

Figure 16: *SVC Fitted Model*



```
In [143]:  #SVC (Creating SVC Model with Specific Parameters)
           svc = SVC(kernel='rbf',C=0.65,gamma=0.01, probability=True)
           svc.fit(X_train_scaled, y_train)

           #Model Accuracy - Training Set (With Parameters Above)
           print("Training Set Accuracy: {:.3f}".format(svc.score(X_train_scaled, y_train)))
           print()

           #Model Accuracy - Testing Set (With Parameters Above)
           print("Validation Set Accuracy: {:.3f}".format(svc.score(X_valid_scaled, y_valid)))
           print()

           #Model Accuracy - Validation Set
           print("Testing Set Accuracy: {:.3f}".format(svc.score(X_test_scaled, y_test)))
```

Figure 17 *Area Under the Curve (SVC)*



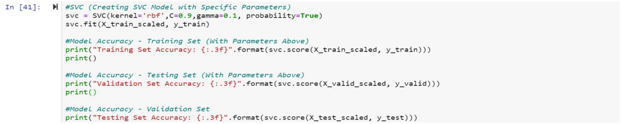
Results:

PROGRESSIVE MODEL (Predict MCI to AD Conversion in 2 Years)

Figure 18: *MCI → MCI vs. MCI → AD Model Accuracy*

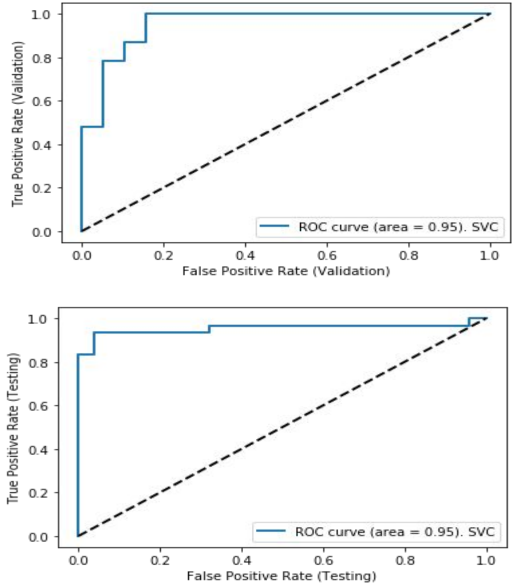
Model (Algorithms):	Training Accuracy:	Validation Accuracy:	Testing Accuracy:
SVC	**97.6%**	**88.1%**	**90.9%**
LogisticRegression	93.5%	88.1%	90.9%
RandomForest	97.6%	76.2%	74.8%
GradientBoosting	97.6%	69.0%	60.0%

Figure 19: *SVC Fitted Model*



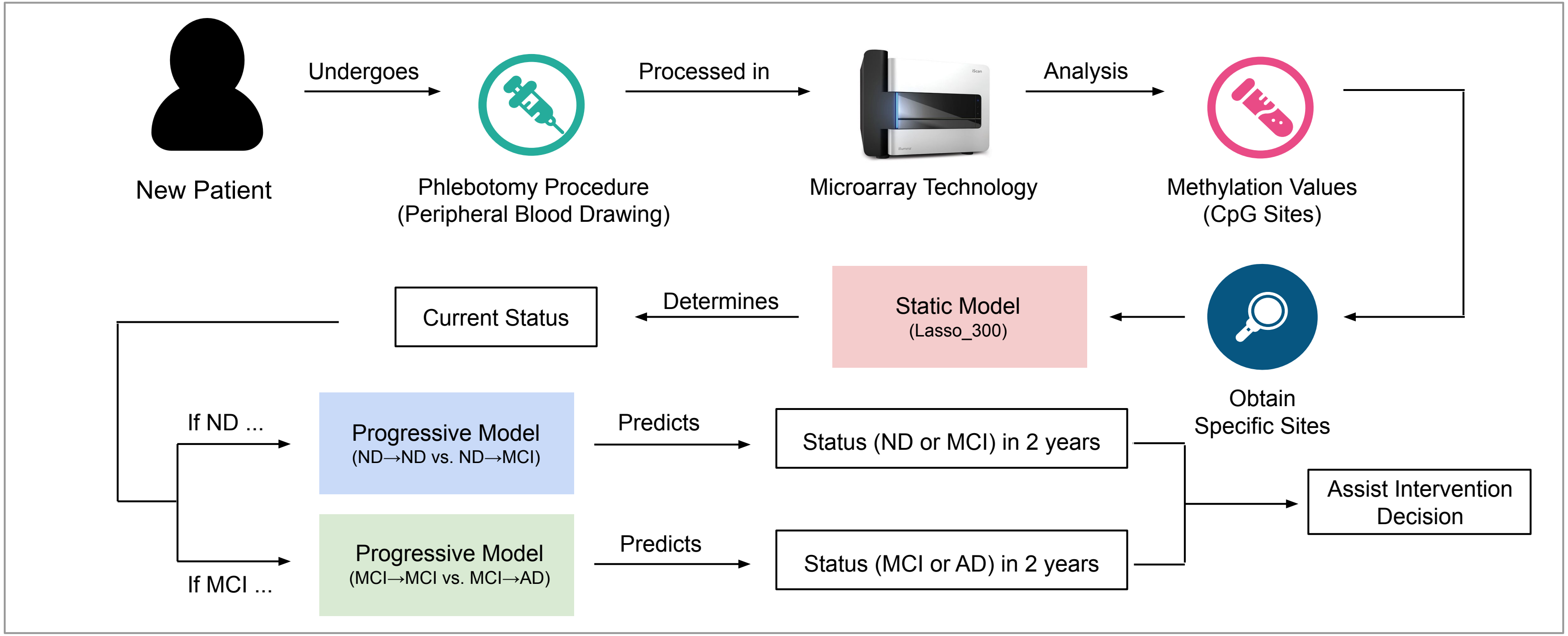
```
In [41]:  #SVC (Creating SVC Model with Specific Parameters)
          svc = SVC(kernel='rbf',C=0.9,gamma=0.1, probability=True)
          svc.fit(X_train_scaled, y_train)

          #Model Accuracy - Training Set (With Parameters Above)
          print("Training Set Accuracy: {:.3f}".format(svc.score(X_train_scaled, y_train)))
          print()

          #Model Accuracy - Testing Set (With Parameters Above)
          print("Validation Set Accuracy: {:.3f}".format(svc.score(X_valid_scaled, y_valid)))
          print()

          #Model Accuracy - Validation Set
          print("Testing Set Accuracy: {:.3f}".format(svc.score(X_test_scaled, y_test)))
```

Figure 20 *Area Under the Curve (SVC)*



Application:

REALISTIC APPLICATION



New Patient
Undergoes
Phlebotomy Procedure (Peripheral Blood Drawing)
Processed in
Microarray Technology
Analysis
Methylation Values (CpG Sites)
Obtain Specific Sites
Static Model (Lasso_300)
Determines
Current Status
If ND ...
Progressive Model (ND→ND vs. ND→MCI)
Predicts
Status (ND or MCI) in 2 years
If MCI ...
Progressive Model (MCI→MCI vs. MCI→AD)
Predicts
Status (MCI or AD) in 2 years
Assist Intervention Decision

Conclusion:

CONCLUSION

1 Discovered Numerous CpG Sites with an Association to Alzheimer’s Disease

- GSE156984 (Brain): 1114 CpG Sites
- GSE153712 (MCI vs. ND): 248 CpG Sites
- Comparison: 8 Shared CpG Sites
- GSE153712 (AD vs. ND): 3034 CpG Sites
- GSE153712 (AD vs. MCI): 22 CpG Sites

2 Created Several Accurate Machine Learning Models

- Best Static Model (Lasso-3oo):
  - Training Accuracy: 95.9%, Validation Accuracy: 89.6%, Testing Accuracy: 78.9%
  - Determines the patient’s current status (No Disease or Mild Cognitive Impairment)
- Progressive Model (ND → ND vs. ND → MCI):
  - Training Accuracy: 98.7%, Validation Accuracy: 96.3%, Testing Accuracy: 86.1%
  - Determines status in 2 years if current status is ND (No Disease or Mild Cognitive Impairment)
- Progressive Model (MCI → MCI vs. MCI → AD):
  - Training Accuracy: 97.6%, Validation Accuracy: 88.1%, Testing Accuracy: 90.9%
  - Determines status in 2 years if current status is MCI (Mild Cognitive Impairment or Alzheimer’s Disease)

Acknowledgements:

Acknowledgements

I would like to express my utmost gratitude to Dr. Marta Nabais (University of Queensland, Australia) and the ADNI Data Sharing and Publications Committee (DPC) of University of Southern California. Dr. Nabais and the DPC provided me with access to the GSE153712 and the ADNI dataset, respectively, and support in spirit. This project wouldn't have existed without their kind support.