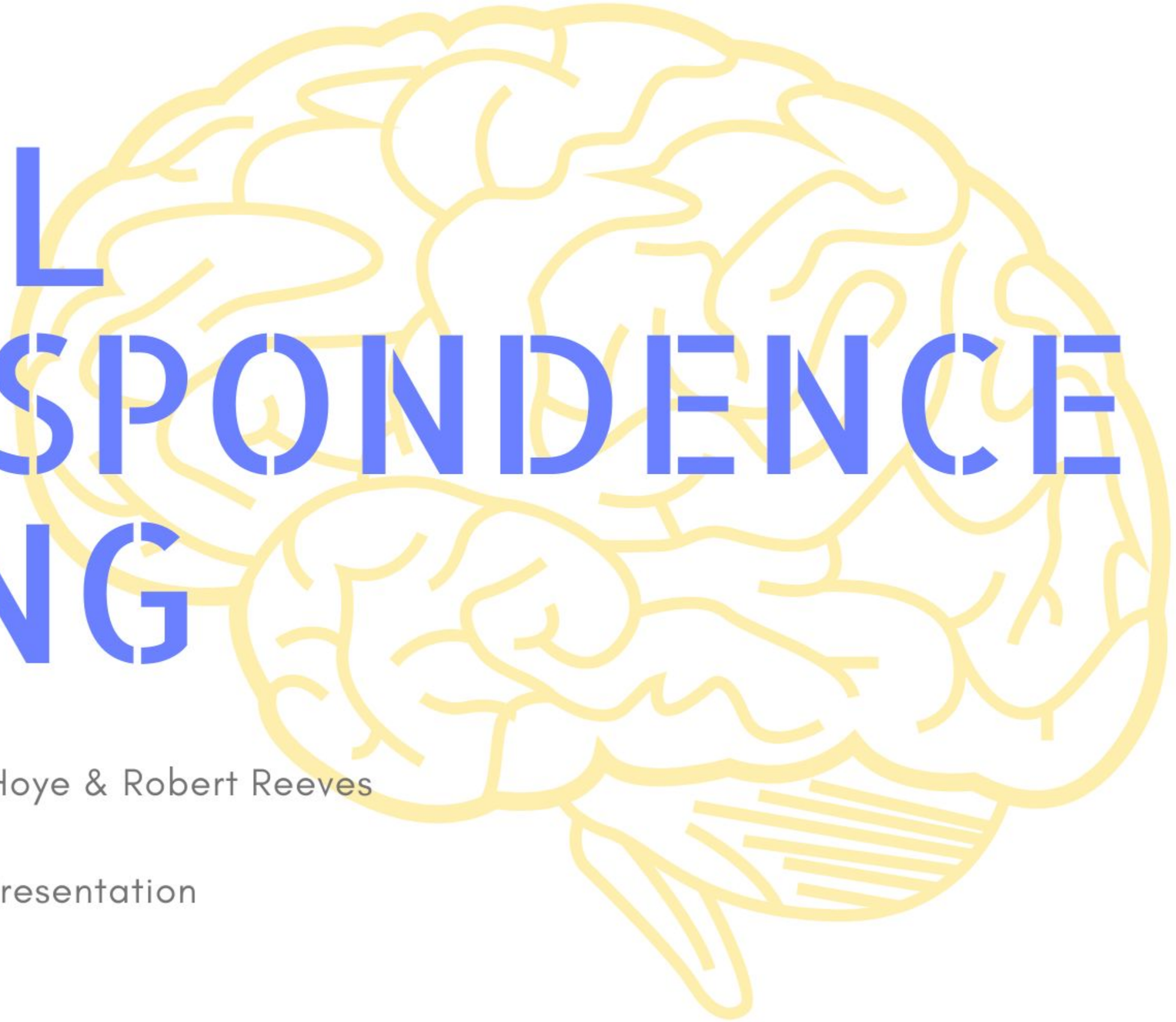
NEURAL CORRESPONDENCE MAPPING



Adita Zeqollari, Arlens Zeqollari, Erik Hoye & Robert Reeves
Advisor: Bradley Voytek
DSE Cohort 5 - Group 4 - Capstone Presentation
https://github.com/voytek/NCM

Today's Presentation

Background & Problem Definition (Adita)

Data Sources & Preprocessing (Adita)

NCM Package Overview (Erik)

NCM Package Demo (Erik)

Modeling (Arlens)

Pipeline Overview (Robert)

Cloud Infrastructure Demo (Robert)

Findings & Conclusion (Arlens)

Background

1990 - "Decade of the Brain"

Declared by George Bush to enhance public awareness of the benefits to be derived from brain research, **leading to lots and lots of research and major advances....**

We still know very little about the brain today

Diverse methods in data processing, sampling & brain atlases between different labs

- Older studies cannot easily be compared to newer studies
- Studies utilize **different coordinate systems**
- Scientific papers contain high level summaries
- No common repository

Facilitate semi-automatic hypothesis generation to speed research discovery by providing a flexible and extendable library for merging disparate neuroscience data into a common coordinate system

PROBLEM DEFINITION

Why is this important?

- **Standardization for Scientific Research**
  - ***"The future of scientific progress will be aided by bridging the gap between the millions of published research articles and modern databases such as the Allen brain atlas (ABA)."***

    - Voytek et al.

- **Possible Use Case**
  - Brain injury/trauma is typically very difficult to treat/rehabilitate due to uncertainty of all functional correlations to specific brain areas. Our library could make it easier for doctors to help patients through tailored therapies.

Data Sources

ALLEN BRAIN ATLAS

Gene expressions

- MRI voxel coordinates
- 20,787 genes with multiple readings for 946 probes x 6 brains

NEUROSYNTH

Terms from Neuroscience published research

- MNI-XYZ and Taliaracs coordinates
- 3,200 terms from 14,371 publications

ECOG

Neural power spectra

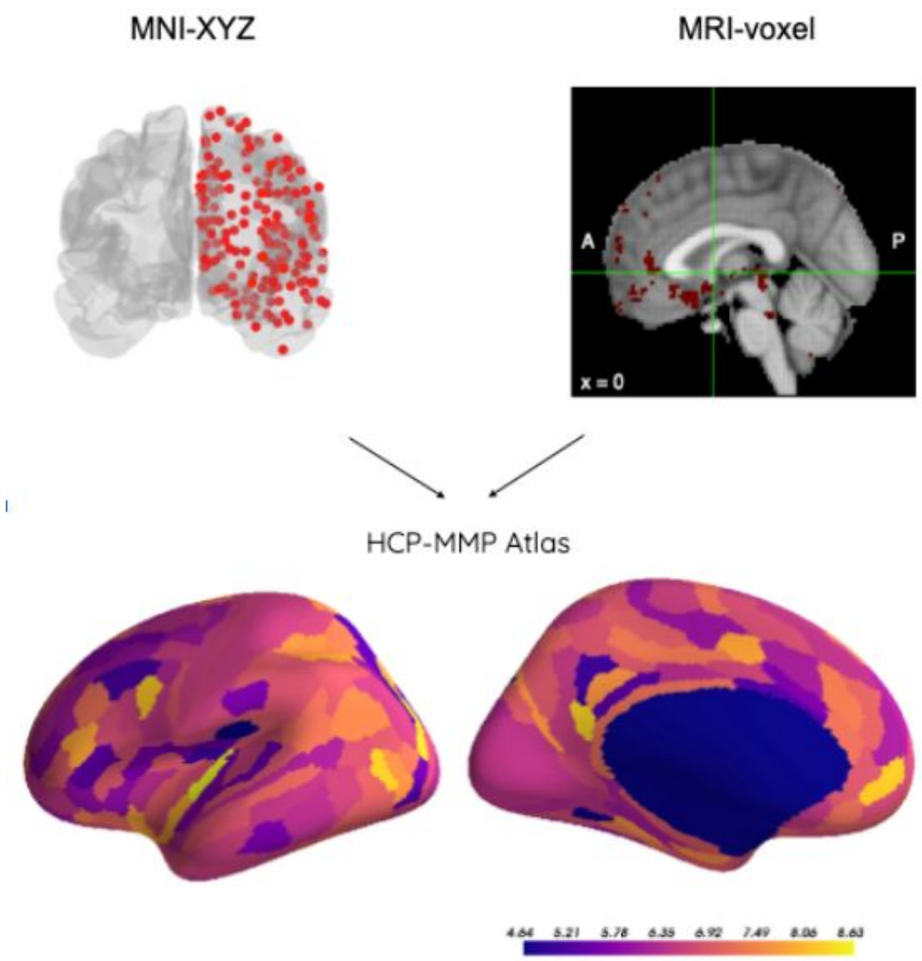
- MNI-XYZ coordinates
- Time series
- 1,723 electrode readings
- 110 subjects

Data Preprocessing

- The datasets we worked with are not required for using the NCM package, but they allowed us to demonstrate the potential of the package.
- All data required some preprocessing/cleaning & formatting in order to be utilized by the package.
- The NCM package is set up to work with preprocessed, clean data that is formatted to align with the package.
- Since our project requires a 2 step preprocessing method (cleaning then mapping with package) we will be focusing on the second step.

Neural Correspondence Mapping Python Package

NCM Python Package



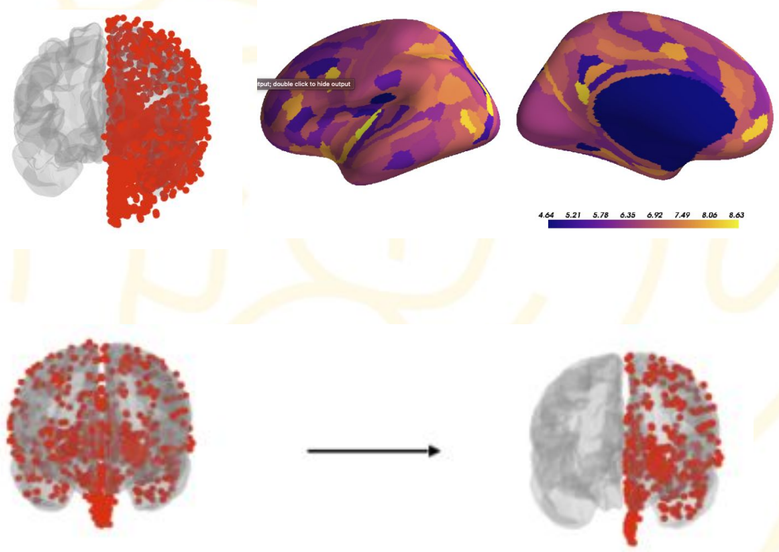
- Modularized class based structure which takes in disparate Neuroscience data and maps it to the same generalized spatial frame to allow for further analysis

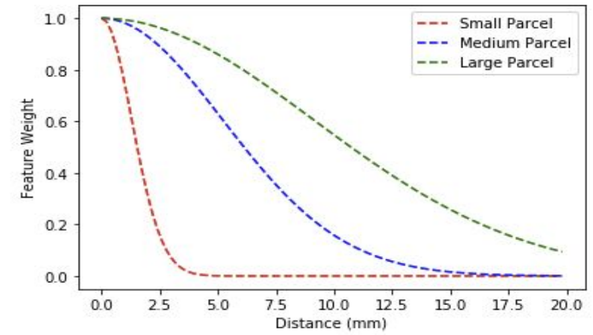
Basic Package Transformation Requirements

1. Ingestion of feature data and required parcellation scheme

2. Identifying the type of transformation desired

3. Deciding the desired mapping technique



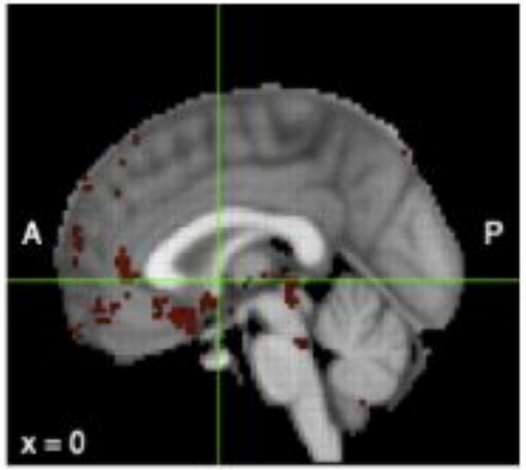


Ingestion of Feature Data & Required Parcellation Scheme

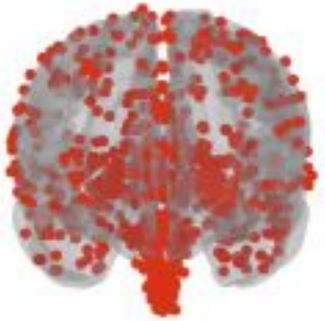
- Data must be in either MNI-XYZ or MRI-voxel coordinates for ingestion

- Atlas defined in Nifti files

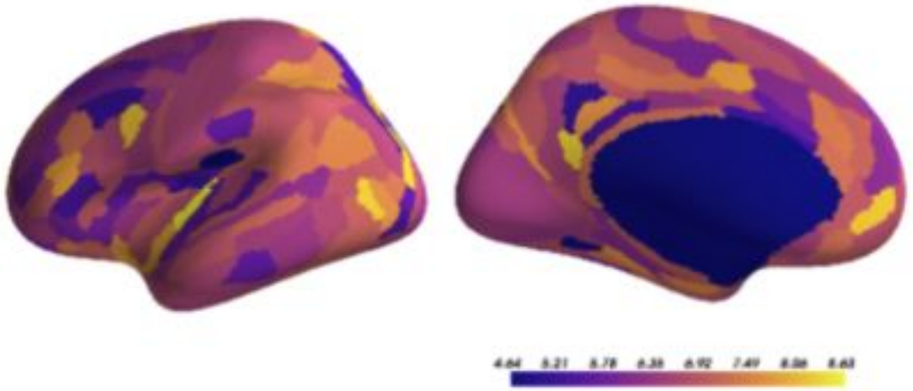
MRI-voxel



MNI-XYZ

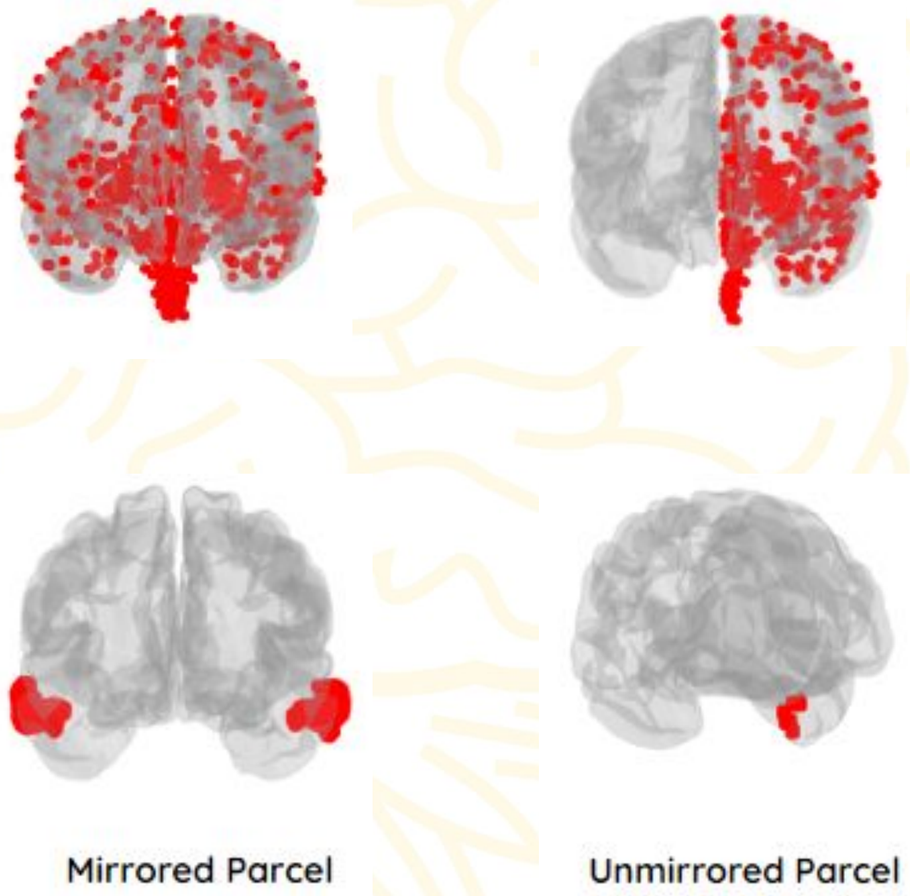


HCP-MMP Atlas



Identifying the Type of Transformation

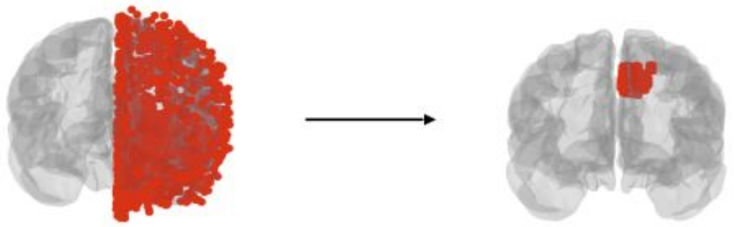
- Full Brain vs. Single Sided
- Mirrored or Unmirrored



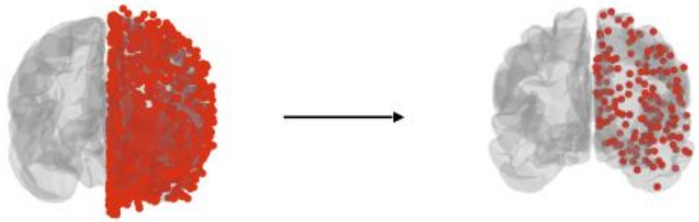
Desired Mapping Technique

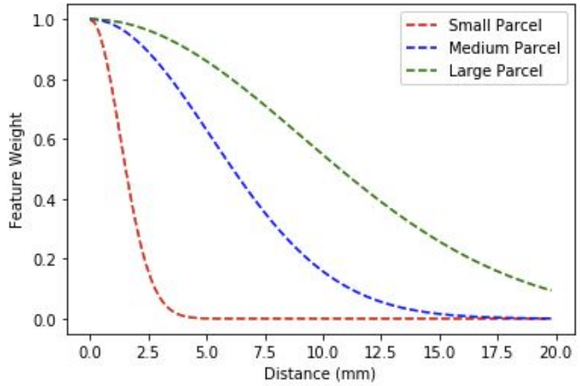
- No “right” answer

- Possible Mapping Techniques:
  - **Method 1**: Project to nearest parcel & average feature representation
  - **Method 2**: Calculate mean parcel location and distance for average weighted feature representation
    - Distance based Gaussian weighting function
    - Varies based on parcel size



Mapping Points to Closest Parcel

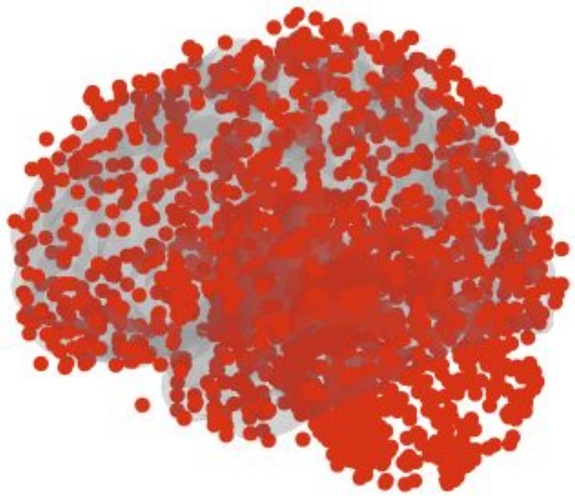


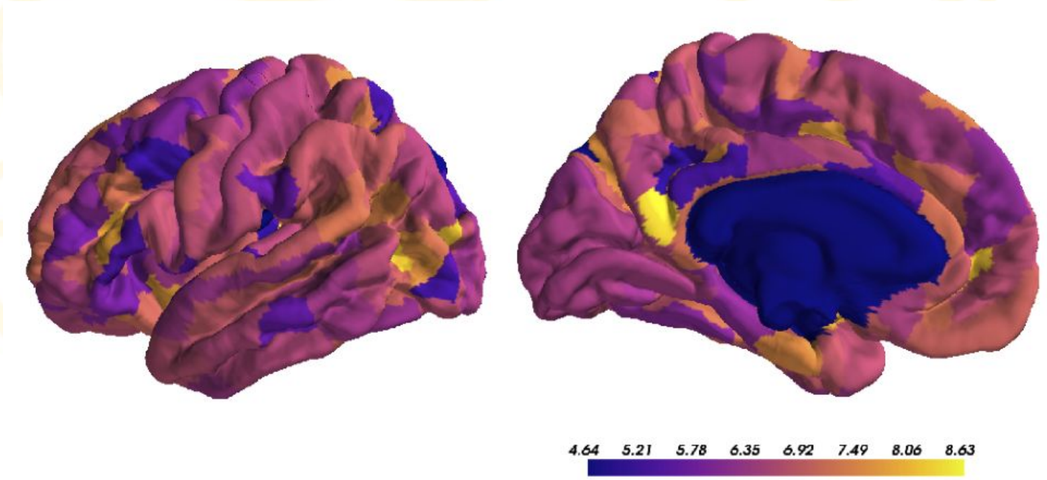


Calculating Parcel Value by Weighting Samples based on Distance

Package Final Result

Disparate precise data mapped to a generalized parcellation schema while preserving accuracy





NCM Demo

	0	1	2	3	4	5	6	7	8	9	...	2654	2655	2656	2657	
729	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	3.972890	4.912563	4.363407	3.870797	4.03
731	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.00
736	7.544136	8.199535	6.806199	7.519986	6.634473	7.860158	7.086888	8.135712	7.068755	8.080848	...	6.532894	7.029888	6.333311	6.450816	6.67
737	10.840025	10.709327	9.837552	9.259373	9.758753	9.029801	10.684216	10.889349	8.101776	8.639627	...	7.968856	7.914093	8.033839	8.262465	8.14
740	7.456360	7.416313	6.223476	6.978941	7.181507	7.370230	6.808955	6.872047	6.070466	6.623760	...	5.167980	6.306512	5.505834	5.877171	6.10

5 rows × 2664 columns

In [5]:
```
#converting dataframe mni-xyz coordinates to numpy array
xyz_coordinates = annotation[['mni_x','mni_y','mni_z']].to_numpy()
features = expression.to_numpy()
# Only testing on first 1000 features for speed purposes
test_features = features[:1000]
print('Coordinate Array Shape: '+str(xyz_coordinates.shape))
print('Feature Array Shape: '+str(test_features.shape))
```

```
Coordinate Array Shape: (2664, 3)
Feature Array Shape: (1000, 2664)
```

Visualization of all ABA probe measurements in MNI-XYZ

Modeling

Modeling

For modeling, the ABA dataset was the best candidate.

Q: Given parcel gene expressions for all other parcels, can we predict all gene expressions for a specific parcel?

ABA Data	X			Y
(21000x180)	parcel1	parcel2	parcel3	parcel3
geneA	0.456	0.456	0.456	0.456
geneB	0.456	0.456	0.456	0.456
geneC	0.456	0.456	0.456	0.456
...	0.456	0.456	0.456	0.456
GeneX	0.456	0.456	0.456	0.456

Modeling

xgBoost

Gradient boosted ensemble method

- gamma: 0.25
- learning_rate: 0.05
- max_depth: 10
- n_estimators: 400
- subsample: 0.75

Adaboost

Gradient boosted ensemble method

- learning_rate: 1.0
- loss: 'linear'
- n_estimators: 100

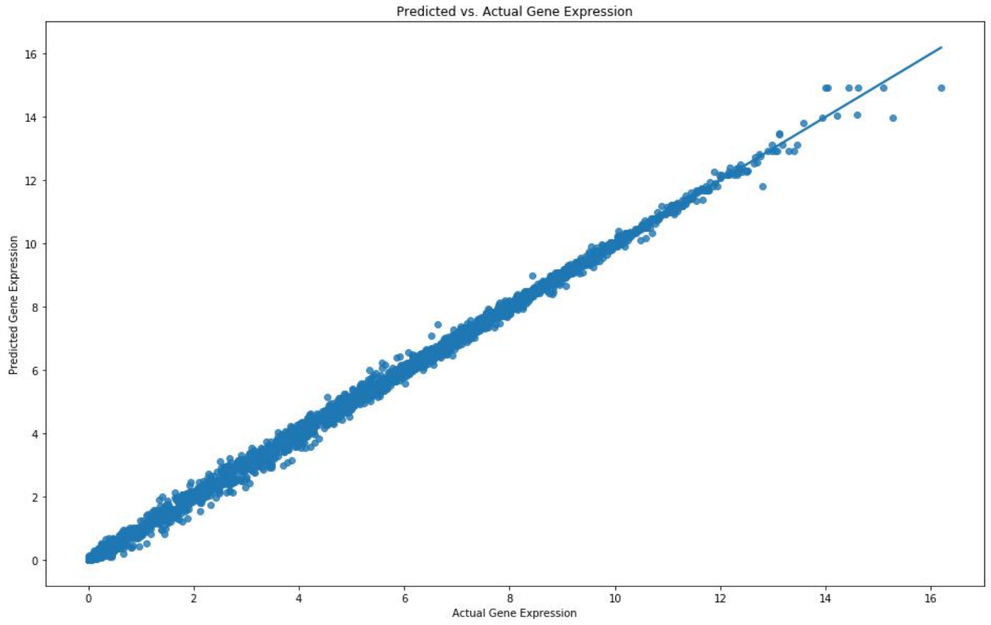
Random Forest

Bagging-based ensemble method

- n_estimators: 1400
- Min_samples_split: 2
- Min_samples_leaf: 2
- max_features: auto
- bootstrap: True

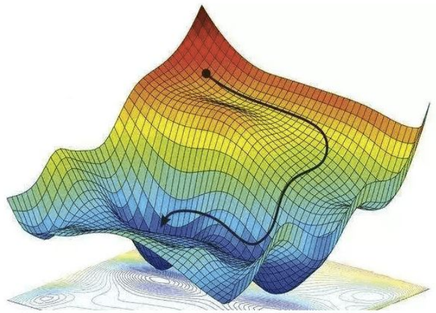
xgBoost

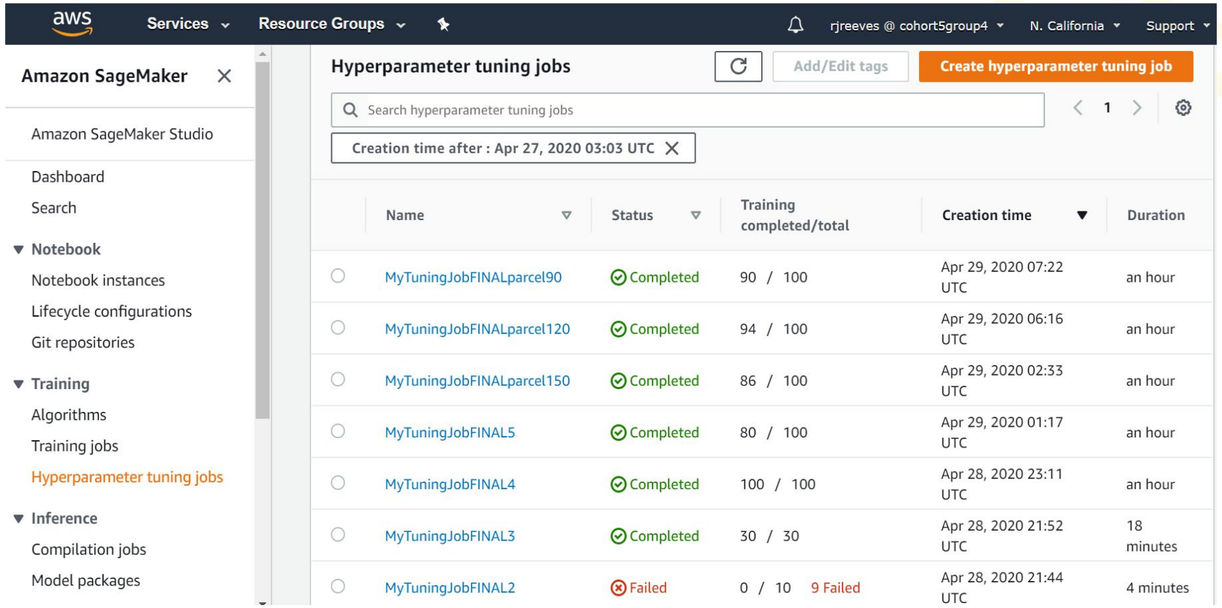
Model (Type)	Best Model Parameters	Cross - Validation	Train / Val / Test (%)	GridsearchCV Parameter Space
xgBoost Regressor (Gradient Boosted Ensemble Method)	**'gamma'**: 0.25 **'learning_rate'**: 0.05 **'max_depth'**: 10 **'n_estimators'**: 400 **'subsample'**: 0.75	3-fold CV	56/24/20	**'N_estimators'**:[50, 100, 400], **'Max_depth'**:[3, 5, 10], **'Learning_rate'**:[0.05, 0.1, 0.5], **'Subsample'**:[0.5,.75, 1], **'gamma'**: [0.25, 0.5, 1, 3]



- Achieved MSE of 0.02 using locally trained xgBoost model.
- These were the best results with respect to accuracy and speed.

Modeling





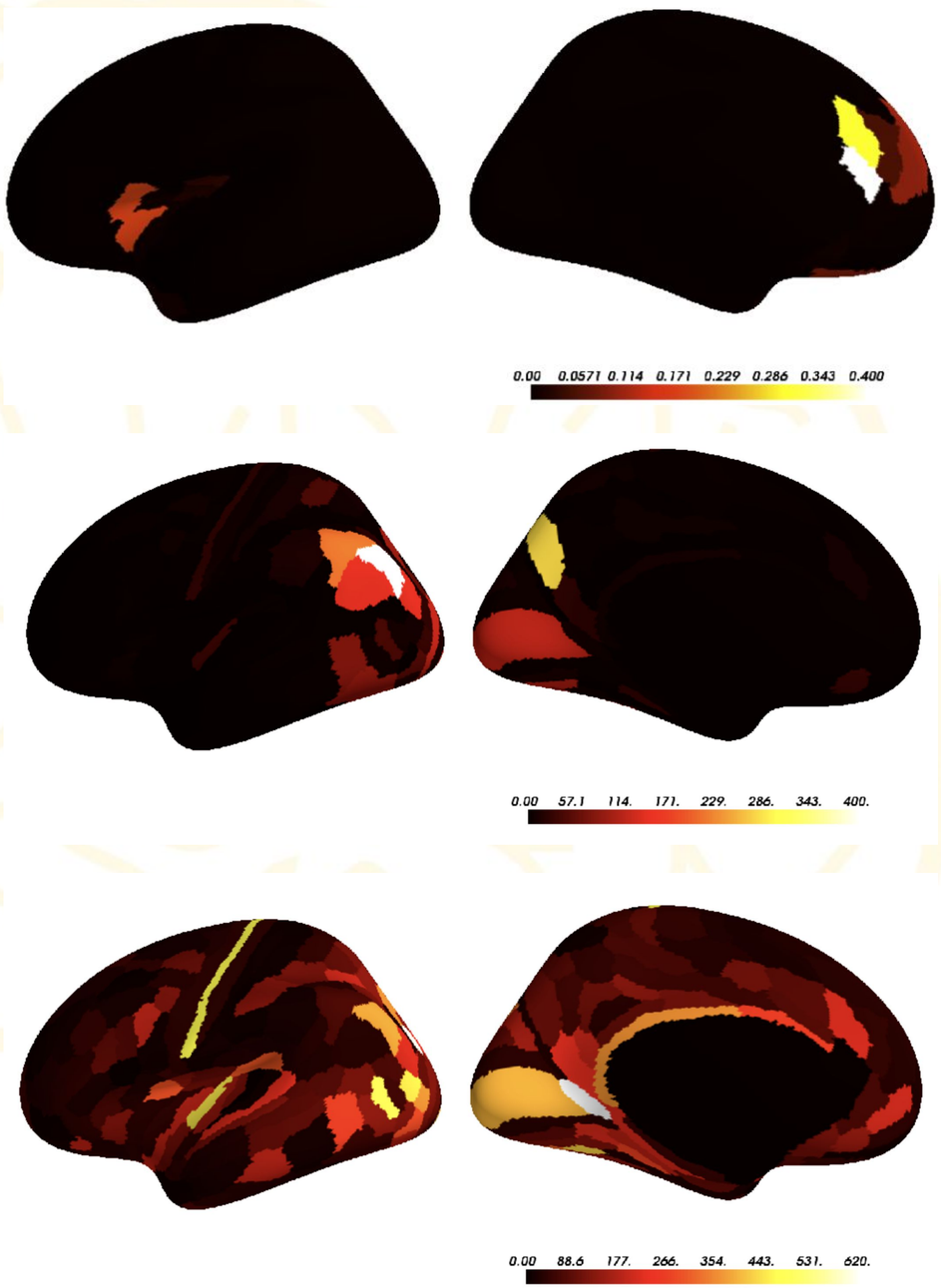
- xgBoost on AWS Sagemaker to leverage powerful compute & integrated storage via S3
- Parallelized Bayesian hyperparameter tuning jobs for improved accuracy
- Model storage and documentation
- Deployment possibilities as a model endpoint

Modeling

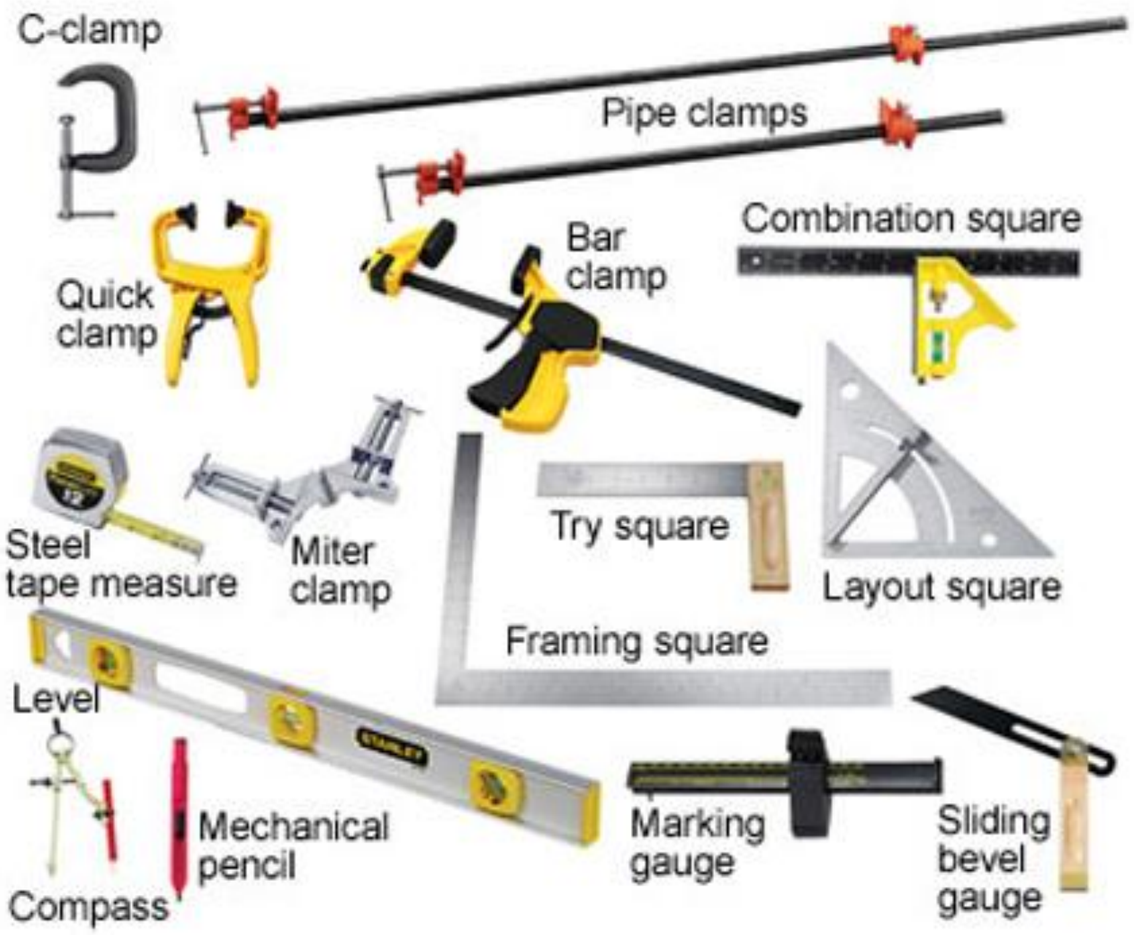
- White parcel is our target and the parcels in varying colors are the parcels with significant importance.

- The black regions signify parcels which had little to no influence on predicting the target value.

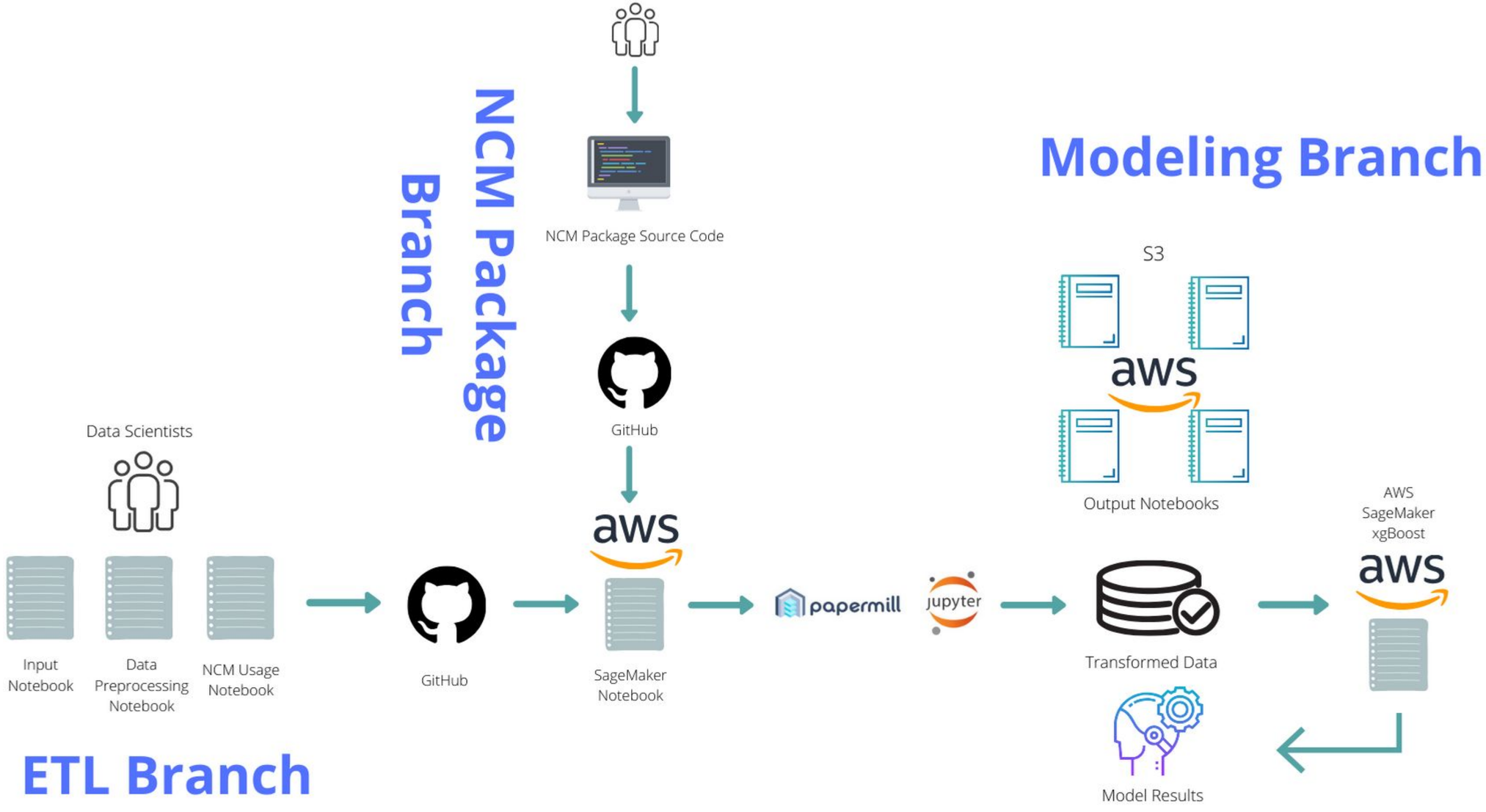
- Accuracy was robust across different parcels and different distributions of gene expression values.



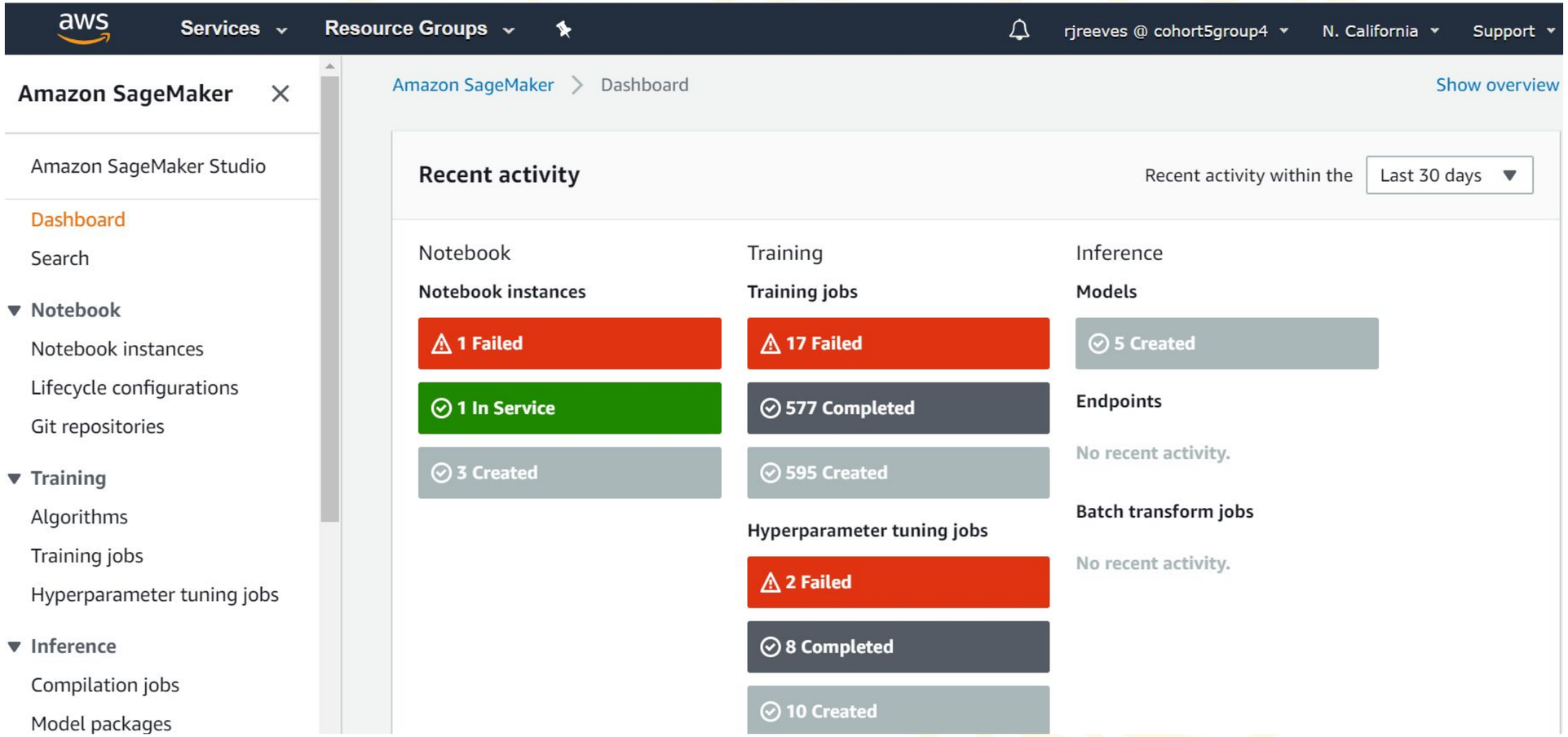
Pipeline Architecture



Pipeline Architecture



AWS Sagemaker



Github

voytek / NCM Private

Watch 4 | Star 2 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 1 | Wiki | Security 0 | Insights

Neural Correspondence Mapping

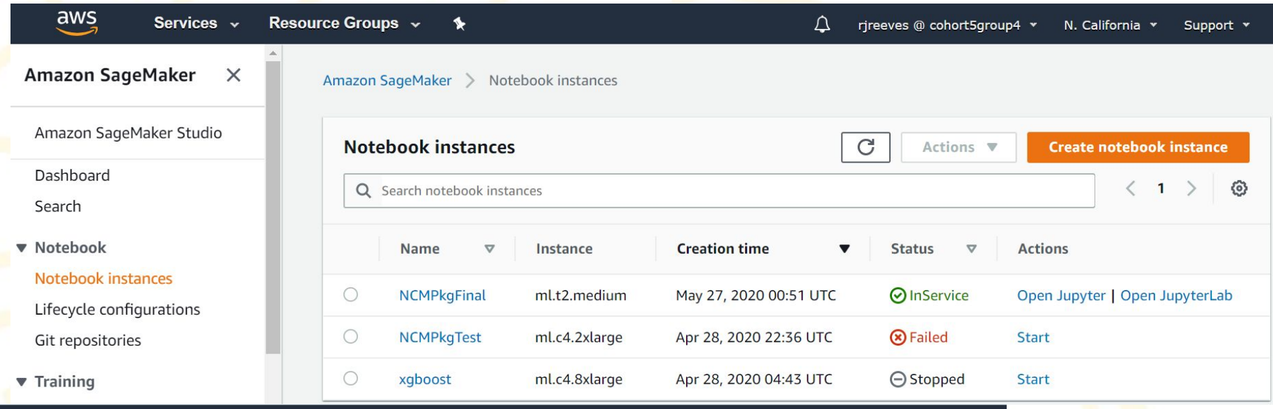
163 commits | 6 branches | 0 packages | 0 releases | 5 contributors | Apache-2.0

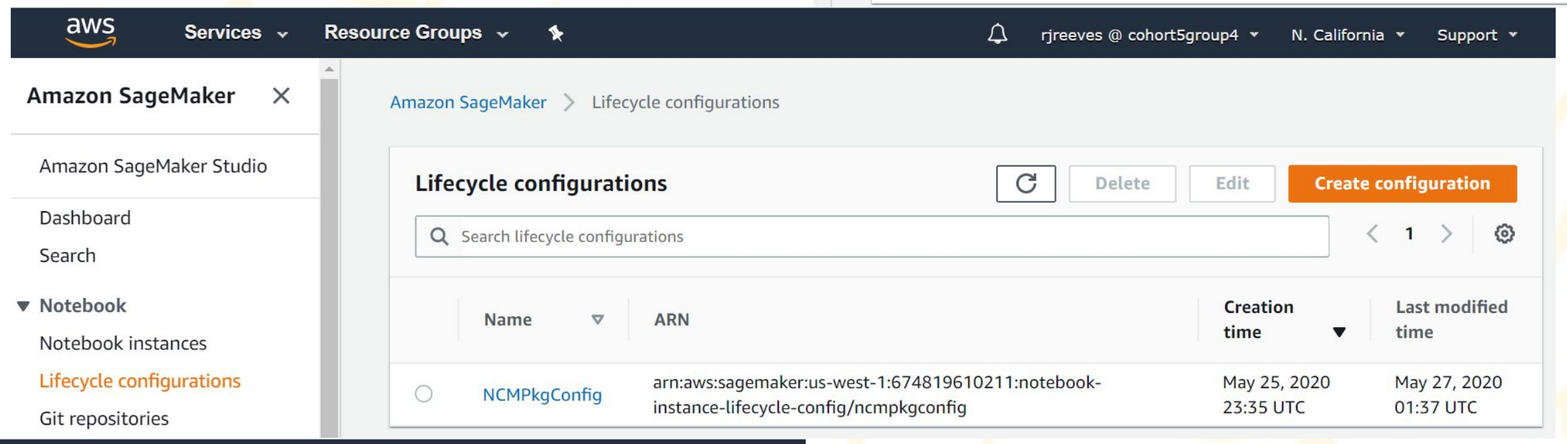
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

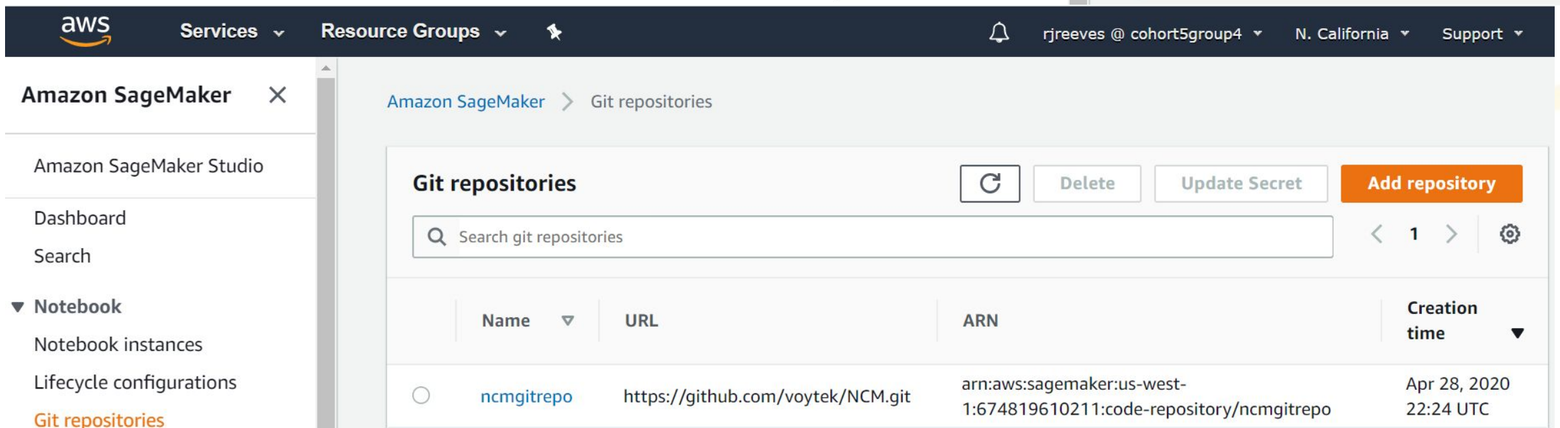
erhoye30 Update README.md — Latest commit 54b9358 28 minutes ago

Examples	relative path changed	5 hours ago
Exploratory_NoteBooks	moved to Preprocessing	2 days ago
Modeling	Update NS notebooks	21 days ago
Preprocessing	Update Read from S3	2 hours ago
Reports	Create Report 8.pdf	2 days ago
Testing	updated to run in AWS based on testing	6 hours ago
ncm	Update transform_data_functions.py	2 days ago
.gitignore	Implemented loading of files from /data assets	2 months ago
LICENSE	Initial commit	4 months ago

AWS Sagemaker







Hyperparameter Tuning Jobs

aws Services Resource Groups rjreeves @ cohort5group4 N. California Support

Amazon SageMaker

- Amazon SageMaker Studio
- Dashboard
- Search
- Notebook
  - Notebook instances
  - Lifecycle configurations
  - Git repositories
- Training
  - Algorithms
  - Training jobs
  - Hyperparameter tuning jobs
- Inference
  - Compilation jobs
  - Model packages

Hyperparameter tuning jobs

Add/Edit tags | Create hyperparameter tuning job

Search hyperparameter tuning jobs

Creation time after : Apr 27, 2020 03:03 UTC

Name	Status	Training completed/total	Creation time	Duration
MyTuningJobFINALparcel90	Completed	90 / 100	Apr 29, 2020 07:22 UTC	an hour
MyTuningJobFINALparcel120	Completed	94 / 100	Apr 29, 2020 06:16 UTC	an hour
MyTuningJobFINALparcel150	Completed	86 / 100	Apr 29, 2020 02:33 UTC	an hour
MyTuningJobFINAL5	Completed	80 / 100	Apr 29, 2020 01:17 UTC	an hour
MyTuningJobFINAL4	Completed	100 / 100	Apr 28, 2020 23:11 UTC	an hour
MyTuningJobFINAL3	Completed	30 / 30	Apr 28, 2020 21:52 UTC	18 minutes
MyTuningJobFINAL2	Failed	0 / 10 9 Failed	Apr 28, 2020 21:44 UTC	4 minutes

aws Services Resource Groups rjreeves @ cohort5group4 N. California Support

Amazon SageMaker

- Amazon SageMaker Studio
- Dashboard
- Search
- Notebook
  - Notebook instances
  - Lifecycle configurations
  - Git repositories
- Training
  - Algorithms
  - Training jobs
  - Hyperparameter tuning jobs
- Inference
  - Compilation jobs
  - Model packages

Amazon SageMaker > Training jobs

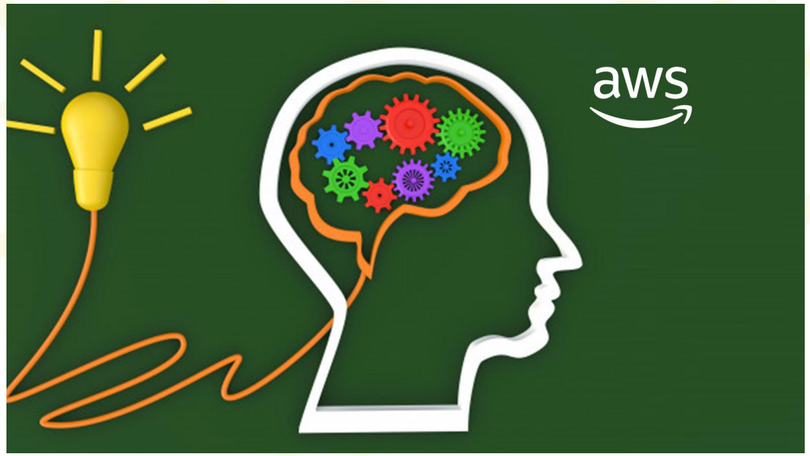
Training jobs

Actions | Create training job

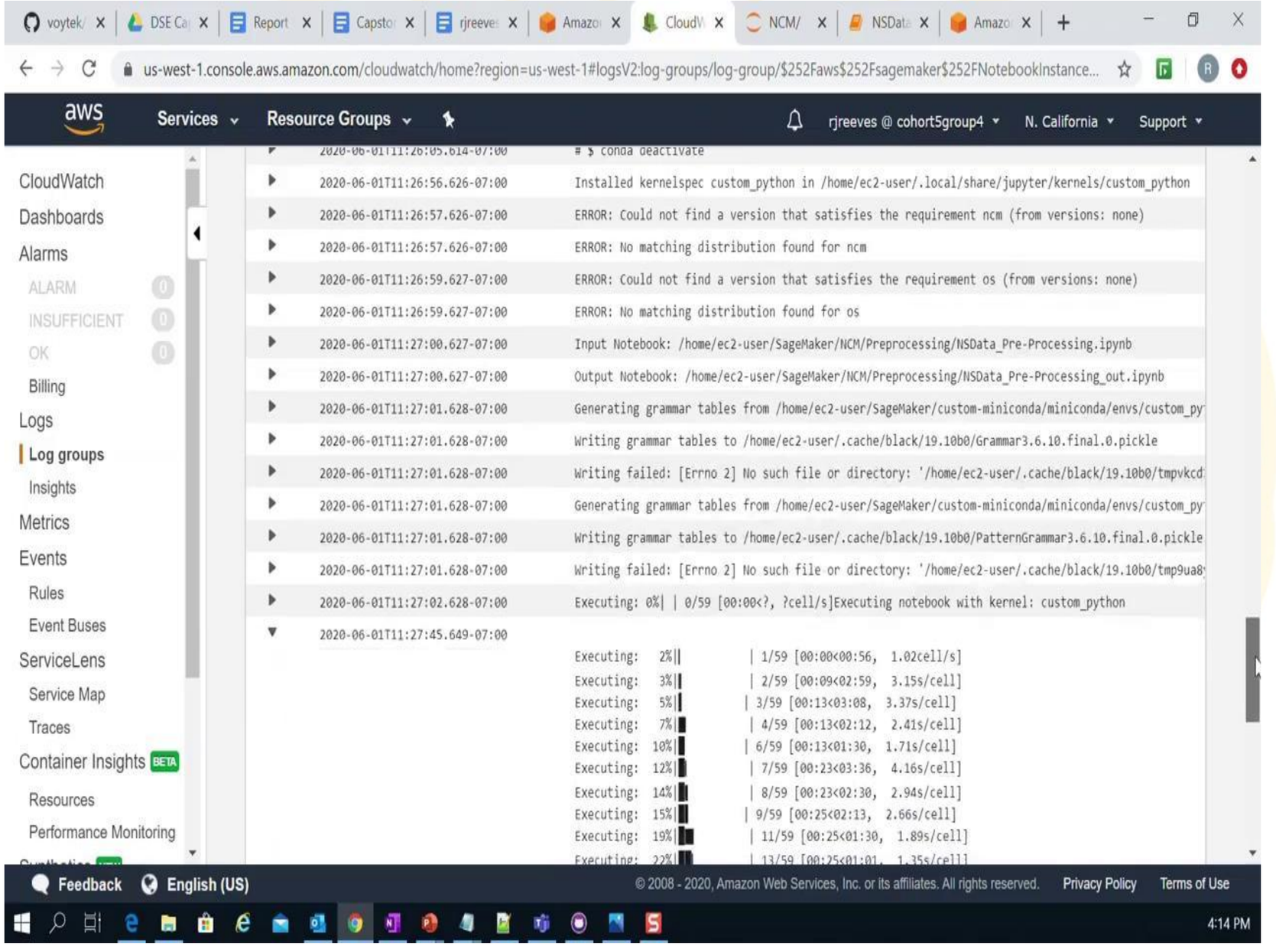
Search training jobs

Status : Failed | Creation time after : Apr 27, 2020 02:55 UTC

Name	Creation time	Duration	Status
MyTuningJobFINAL2-010-02bcf888	Apr 28, 2020 21:44 UTC	3 minutes	Failed
MyTuningJobFINAL2-009-12634254	Apr 28, 2020 21:44 UTC	3 minutes	Failed
MyTuningJobFINAL2-008-cd5b339a	Apr 28, 2020 21:44 UTC	4 minutes	Failed
MyTuningJobFINAL2-007-78119435	Apr 28, 2020 21:44 UTC	3 minutes	Failed
MyTuningJobFINAL2-006-93839b46	Apr 28, 2020 21:44 UTC	3 minutes	Failed
MyTuningJobFINAL2-005-3d7c8a1b	Apr 28, 2020 21:44 UTC	4 minutes	Failed
MyTuningJobFINAL2-004-447dfec0	Apr 28, 2020 21:44 UTC	3 minutes	Failed



AWS Demo


us-west-1.console.aws.amazon.com/cloudwatch/home?region=us-west-1#logsV2:log-groups/log-group/$252Faws$252Fsagemaker$252FNotebookInstance...
aws
Services
Resource Groups
rjreeves @ cohort5group4
N. California
Support
CloudWatch
Dashboards
Alarms
ALARM
INSUFFICIENT
OK
Billing
Logs
Log groups
Insights
Metrics
Events
Rules
Event Buses
ServiceLens
Service Map
Traces
Container Insights BETA
Resources
Performance Monitoring
2020-06-01T11:26:56.626-07:00 Installed kernelspec custom_python in /home/ec2-user/.local/share/jupyter/kernels/custom_python
2020-06-01T11:26:57.626-07:00 ERROR: Could not find a version that satisfies the requirement ncm (from versions: none)
2020-06-01T11:26:57.626-07:00 ERROR: No matching distribution found for ncm
2020-06-01T11:26:59.627-07:00 ERROR: Could not find a version that satisfies the requirement os (from versions: none)
2020-06-01T11:26:59.627-07:00 ERROR: No matching distribution found for os
2020-06-01T11:27:00.627-07:00 Input Notebook: /home/ec2-user/SageMaker/NCM/Preprocessing/NSData_Pre-Processing.ipynb
2020-06-01T11:27:00.627-07:00 Output Notebook: /home/ec2-user/SageMaker/NCM/Preprocessing/NSData_Pre-Processing_out.ipynb
2020-06-01T11:27:01.628-07:00 Generating grammar tables from /home/ec2-user/SageMaker/custom-miniconda/miniconda/envs/custom_py
2020-06-01T11:27:01.628-07:00 Writing grammar tables to /home/ec2-user/.cache/black/19.10b0/Grammar3.6.10.final.0.pickle
2020-06-01T11:27:01.628-07:00 Writing failed: [Errno 2] No such file or directory: '/home/ec2-user/.cache/black/19.10b0/tmpvkcd
2020-06-01T11:27:01.628-07:00 Generating grammar tables from /home/ec2-user/SageMaker/custom-miniconda/miniconda/envs/custom_py
2020-06-01T11:27:01.628-07:00 Writing grammar tables to /home/ec2-user/.cache/black/19.10b0/PatternGrammar3.6.10.final.0.pickle
2020-06-01T11:27:01.628-07:00 Writing failed: [Errno 2] No such file or directory: '/home/ec2-user/.cache/black/19.10b0/tmp9ua8
2020-06-01T11:27:02.628-07:00 Executing: 0%| | 0/59 [00:00<?, ?cell/s]Executing notebook with kernel: custom_python
2020-06-01T11:27:45.649-07:00
Executing: 2%| | 1/59 [00:00<00:56, 1.02cell/s]
Executing: 3%| | 2/59 [00:09<02:59, 3.15s/cell]
Executing: 5%| | 3/59 [00:13<03:08, 3.37s/cell]
Executing: 7%| | 4/59 [00:13<02:12, 2.41s/cell]
Executing: 10%| | 6/59 [00:13<01:30, 1.71s/cell]
Executing: 12%| | 7/59 [00:23<03:36, 4.16s/cell]
Executing: 14%| | 8/59 [00:23<02:30, 2.94s/cell]
Executing: 15%| | 9/59 [00:25<02:13, 2.66s/cell]
Executing: 19%| | 11/59 [00:25<01:30, 1.89s/cell]
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
4:14 PM

Findings

A package could be developed in Python to ingest, transform & visualize disparate neuroscience data

AWS computing could be leveraged to expand the functionality & efficiency of the package

Modularity & scalability of the package supports further enhancements & collaborative development

Interesting and unintuitive associations between parcels for a given gene can generate hypotheses about physical or functional interactions

The package could be utilized to perform modeling of gene expressions showing that gene expressions can be predicted accurately

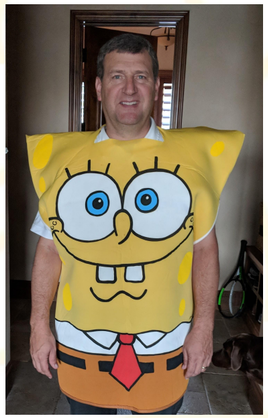
Thank you!

Arlens Zeqollari

Aspiring Random Number Generator

Erik Hoye

BERT Watcher



Robert Reeves

Stadium Announcer

Adita Zeqollari

Top contributor to Netflix's recommendation engine training data

"...the work you've done is really awesome, and I hope my lab can pick it up and run with it."

- Professor Bradley Voytek