Project

Plan

Data

- How many samples of each data type? (i.e., 23andMe, vcfs, etc.)
- What annotated phenotype data do we have?
    - Balanced?
- What can we compare against?
- What kind of preprocessing do we need to do?
    - Getting same set of SNPs for each sample
        - Imputation? Subsetting?
    - MAF normalization
    - z-score
- Input and target format
    - X is a n x m matrix
    - Y is...

Model

- Baseline architecture
    - How many hidden layers? How wide?
    - Activation functions
    - Loss function (balance?)

```
class Feedforward(torch.nn.Module):
        def __init__(self, input_size, hidden_size):
            super(Feedforward, self).__init__()
            self.input_size = input_size
```

```
            self.hidden_size = hidden_size
            self.fc1 = torch.nn.Linear(self.input_size, self.hidden_size)
            self.relu = torch.nn.ReLU()
            self.fc2 = torch.nn.Linear(self.hidden_size, 1)
            self.sigmoid = torch.nn.Sigmoid()
        def forward(self, x):
            hidden = self.fc1(x)
            relu = self.relu(hidden)
            output = self.fc2(relu)
            output = self.sigmoid(output)
            return output
```

Experimentation

- Architectures
- Use only GWAS SNPs?
    - p-value thresholds or effect size cutoffs
- Compare to logistic regression from sklearn
- Additional features?*
- Completely different feature sets?*

*If time

Interpretation

- Visualizations of weights
- Which weights go to 0?

Implementation

- Host on Github: Repo name?
    - bin — scripts and such
    - data — house data
    - doc — write-up and ppt
    - results — figs, datafiles generated
    - models — saved models

    - config? — yml files, requirements.txt
- tensorboard?
- runnable .py files for each step
    - download_data.py or .sh - probably just need to include the wget command or something like that in README.md → where to access data
    - preprocess.py - probably will be the hardest step, script to preprocess all data into matrix (maybe a notebook)
    - train.py — training script
    - model.py — model definition
    - test.py — run against test set, compare to other models
    - others?

Questions?

- LD, is this taken care of by network? Or do we need to prune?