

Summer 2018 Projects Summary

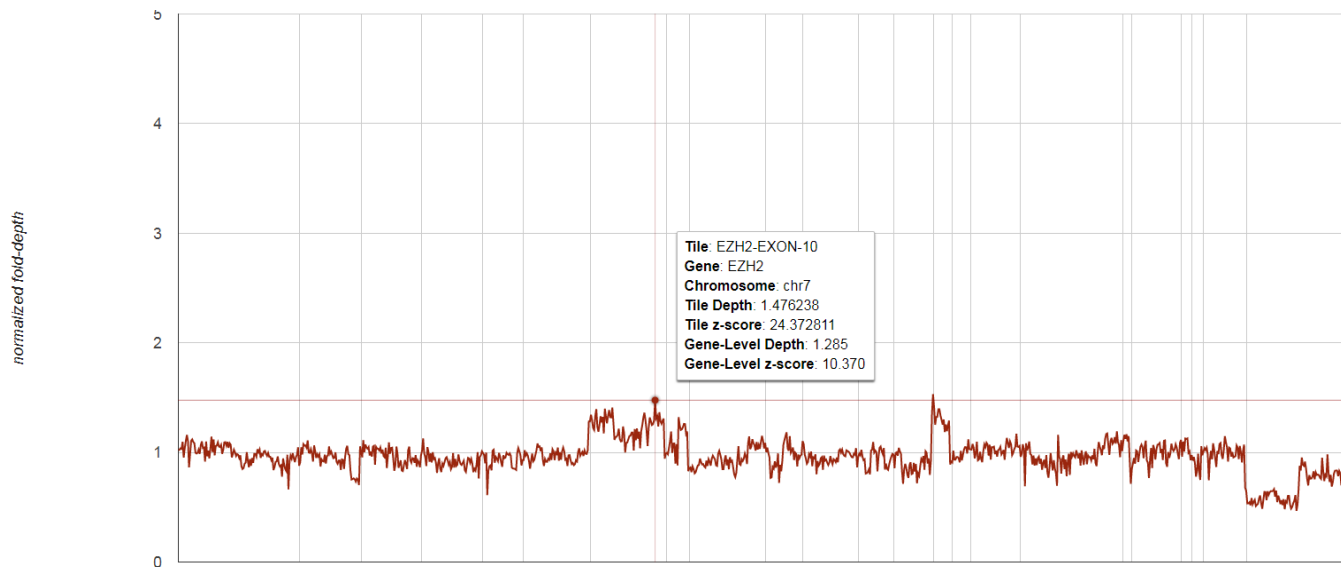
August 24th 2018

Projects

- Copy Number Plots
- Mutalyzer Lookup
- Allele Frequency Noise Plots
- Water Barcode Filter
- Myeloid Data Entry
- MSI Classification

Copy Number Plots

- Plots that show copy number irregularities
- Include a 'tooltip'
 - Gene and chromosome
 - Tile-level depth and z-score
 - Gene-level information
- More easily identify copy number irregularities, on both tile and gene level

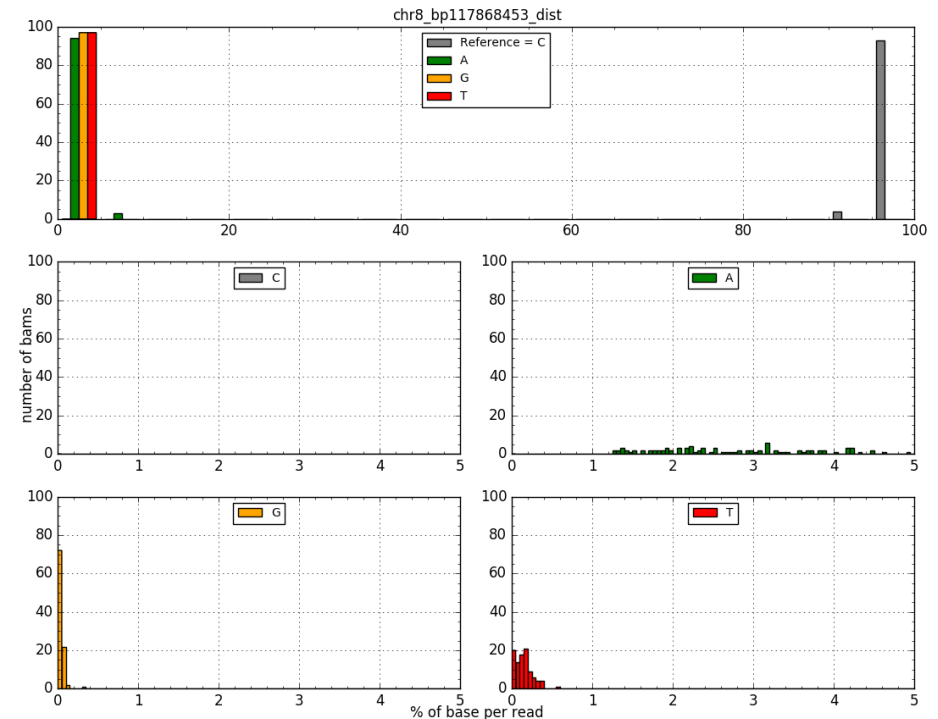


Mutalyzer Lookup

- Query Mutalyzer to include HGVS nomenclature base change and amino acid change in variant report
- Validation script
- Already being added to the pipeline

Allele Frequency Noise Plots

- Quantify the frequency of each base at a user-entered chromosome and coordinate
 - Using 97 myeloid cases
- Visually appreciate noise at certain positions to make more informed variant calls



Water Barcode Filter

- Search for potential contamination of the water barcode
- Script maps water barcode reads to hg19 and Phix genome, filtering out mapped reads
- Returns Fastq file of unmapped reads for BLAST search

Myeloid Data Entry

- Add fields containing CALR, FLT3 and CEBPA results to myeloid data spreadsheet
- More robust dataset for future analysis

Microsatellite Instability Classification by NGS

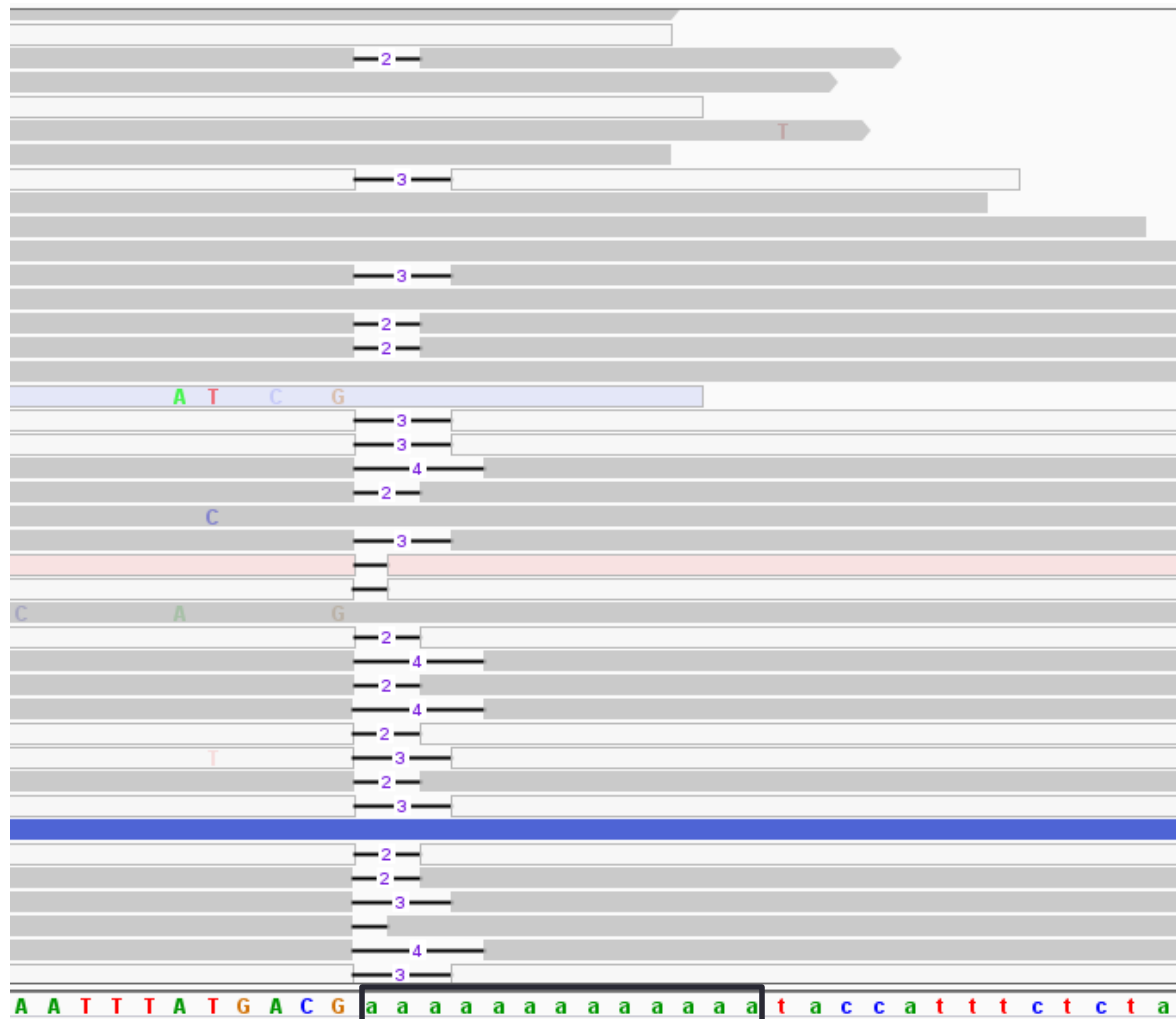
Microsatellite Instability

- Characterized by the “spontaneous loss or gain of nucleotides from repetitive DNA tracts”
 - Caused by defective mismatch repair
- Is a diagnostic phenotype for certain cancer types with clinical implications
 - Actionable marker for immune-checkpoint-blockade therapy (Hause et al. 2016)

Microsatellite Instability

- Develop a model that can accurately detect Microsatellite Instability using NGS data
- Add informative loci to the STAMP selector
 - Clinically relevant and actionable

Repeat Identification



Repeat Identification



- Flanking regions
 - Immediately adjacent to the mononucleotide repeat
 - 7 bases long
 - Long enough to be unique, identifiable in individual reads
 - Short enough to maximize usable reads

Repeat Identification

- Pull all reads containing bases in the region of the locus
 - Psam
- Scan and find flanking regions
 - Ordered search
 - First flank *must* be found before the second
 - Brute-force partial string matching
 - Mismatch allowance = 2
- Validation: filter
 - Baseline = 90%
 - Modified for some loci
 - Those that contain a non-homogenous run

MSI Calling

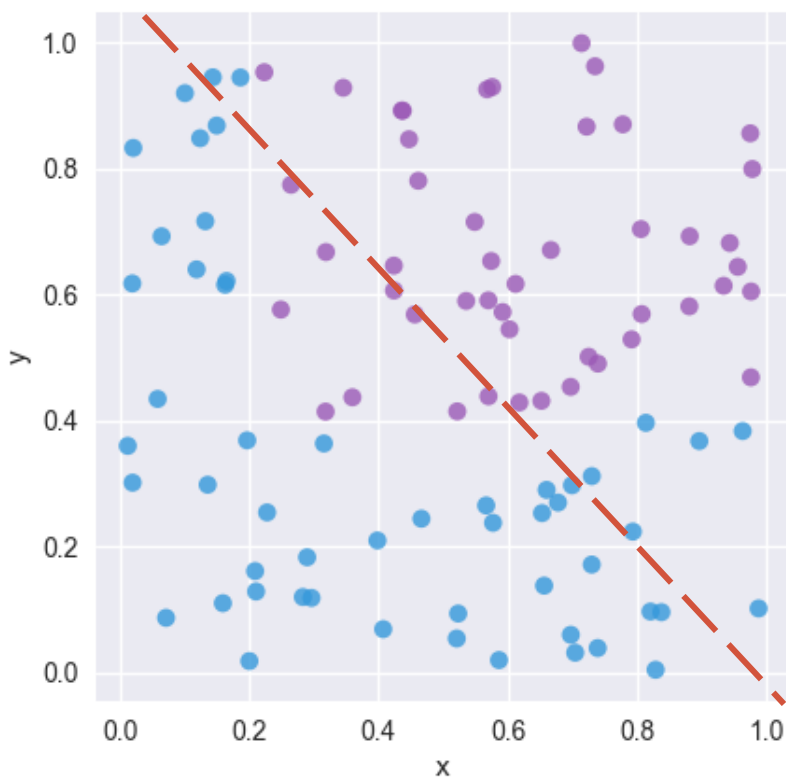
- Binary classifier
 - Simplification of MSI-H and MSI-L
- Optimization
 - Accuracy
 - Sensitivity
 - Specificity

Machine Learning Approach

- TensorFlow
 - Python API developed by Google
- Develop a linear function that can separate the data into 2 categories
 - MSI or MSS
- Apply function to a data point to calculate its score
- Apply sigmoid function to calculate $p(\text{MSI})$
 - Supervised learning
 - Classification problem

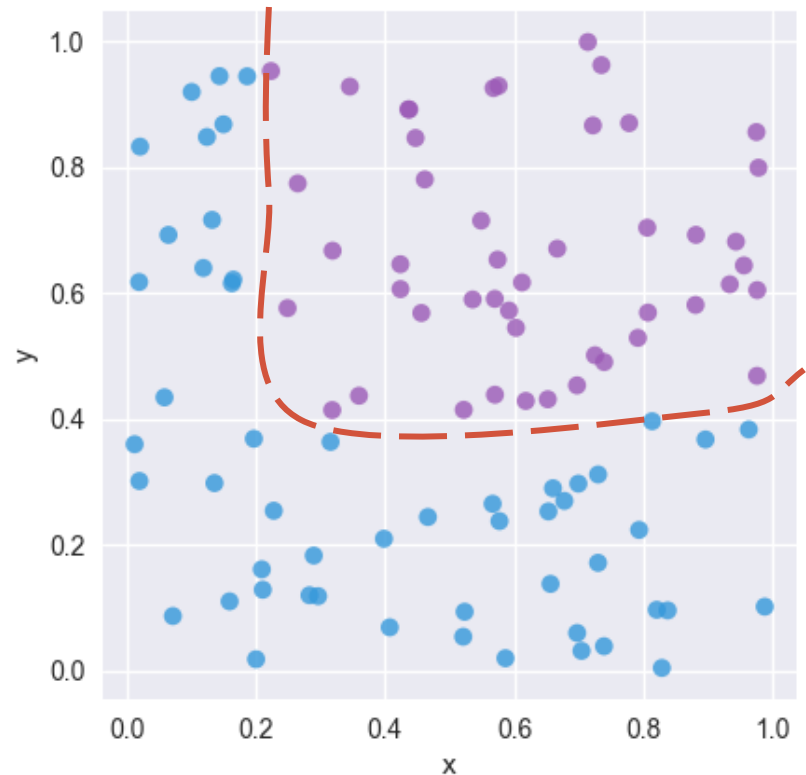
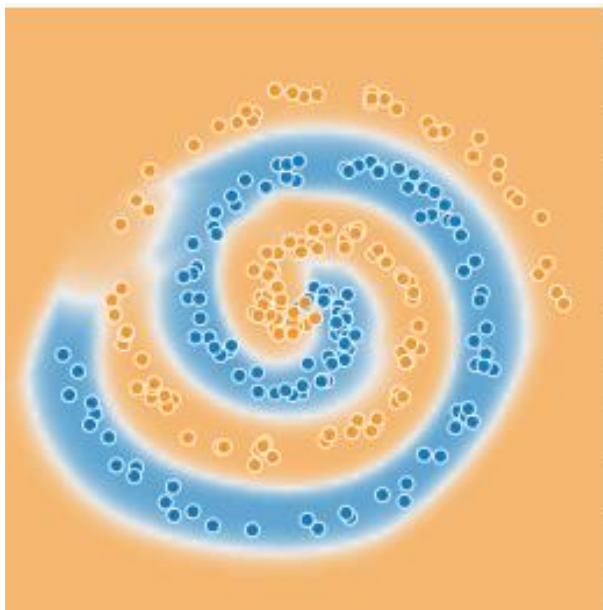
Google's 'ML Crash Course'

- Finding the best 'best-fit line' to distinguish two groups of data
 - Minimizing error



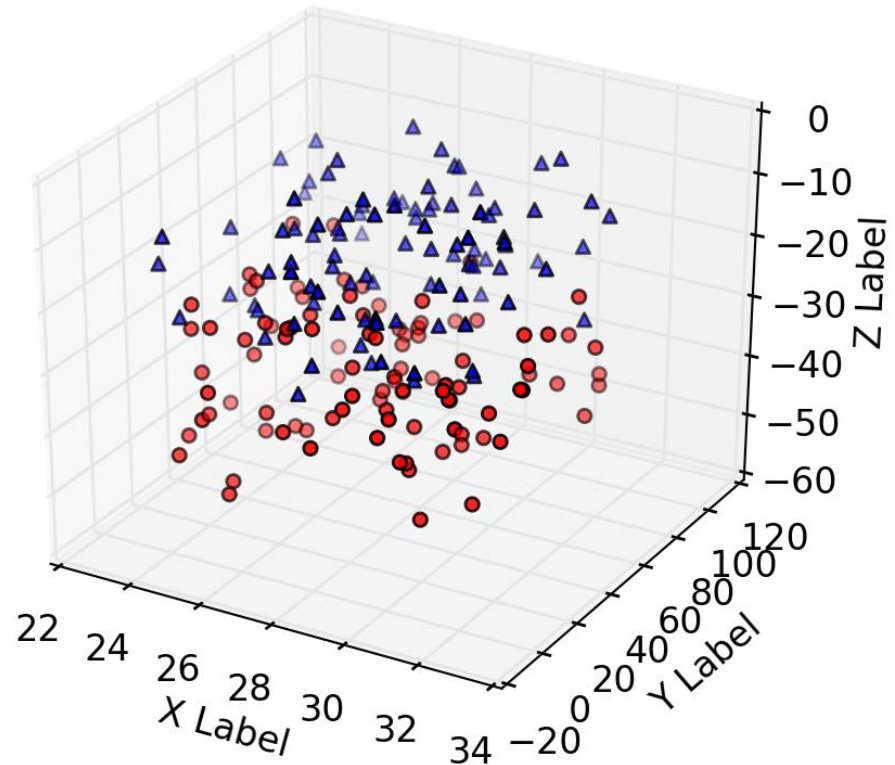
Google's 'ML Crash Course'

- Finding the best 'best-fit line' to distinguish two groups of data
 - Minimizing the error



Google's 'ML Crash Course'

- Finding the best 'best-fit line' to distinguish two groups of data
 - Minimizing the error
- ML allows expansion of this idea to n -dimensions



Machine Learning Approach

- Locus-based calls
 - Develop a linear model using ML for each locus individually
- Overall call based on the proportion of all 27 loci that are unstable
- Simplification
 - 7 loci with high number of BAMs with reads available, good depth at the loci
 - Initial proof of concept

MSI Calling - Features

- Number of lengths
 - $[11, 12, 11, 13, 14] = 4$
- Distance from mode
 - Average distance from MSS sample mode of all reads
- Standard deviation
- Average Length

Datasets

- All files
 - 602 BAM files with MSS, MSI-L or MSI-H annotation
 - 241 COAD-READ, 361 UCEC
 - 212 MSI, 390 MSS
- Mode training set
 - Randomly generated set of 100 BAM files with MSS status to generate mode length
 - 43 COAD-READ, 57 UCEC
 - 0 MSI, 100 MSS

Datasets (continued)

- Training Set
 - 300 BAM files of mixed status to train the model
 - 118 COAD-READ, 182 UCEC
 - 130 MSI, 170 MSS
- Validation Set
 - 100 BAM files of mixed status to validate the model
 - 44 COAD-READ, 56 UCEC
 - 40 MSI, 60 MSS
- Test Set
 - 102 remaining BAM files of mixed status to test the model
 - 36 COAD-READ, 66 UCEC
 - 42 MSI, 60 MSS

Datasets (continued)

- Randomly generated sets
 - Representative of both cancer types
- Low coverage problem
 - Low coverage at certain loci excludes some files from some locus analyses
 - Actual size of datasets vary based on individual BAM coverage at any given locus

Locus-Based Calling

- Generate ML model for each locus individually
 - Weight for each feature, bias
 - Sigmoid activation function to generate $p(\text{MSI})$ in range (0, 1)
- Choose loci with highest AUC
 - Exclude those that have no reads in many BAM files
 - Try different combinations of loci
- Determine MSI status based on number of loci with 'MSI' call

Locus-Based Calling (continued)

- Loci examined
 - BAT-26
 - MSI-07
 - MSI-09
 - H-06
 - MSI-06
 - MSI-04
 - HSPH1-T17
- Threshold: 0.500000
- Min no. loci: 4
- Total files: 93
- Correct predictions: 78
 - True pos: 24
 - True neg: 54
 - False pos: 2
 - False neg: 13
- **Accuracy: 0.838710**
- **Sensitivity: 0.648649**
- **Specificity: 0.964286**

Probability-Based Calling

- Generate ML model for each locus individually
 - As before
- Choose loci with highest AUC
 - As before
- Determine MSI status based on average $p(\text{MSI})$ across loci examined

Probability-Based Calling (example)

- BAM file: TCGA-00-0000

- BAT-26

- $p(\text{MSI}) = 0.998$

- MSI-07

- $p(\text{MSI}) = 0.488$

- MSI-09

- $p(\text{MSI}) = 0.478$

- H-06

- $p(\text{MSI}) = 0.499$

- MSI-06

- $p(\text{MSI}) = 0.898$

- MSI-04

- $p(\text{MSI}) = 0.978$

- HSPH1-T17

- $p(\text{MSI}) = 0.408$

- Locus-based call: MSS

- 3 MSI, 4 MSS

- Probability-based call: MSI

- Avg $p(\text{MSI})$: 0.678

Probability-Based Calling (continued)

- Loci examined
 - BAT-26
 - MSI-07
 - MSI-09
 - H-06
 - MSI-06
 - MSI-04
 - HSPH1-T17
- Threshold: 0.500000
- Total files: 93
- Correct predictions: 78
 - True pos: 23
 - True neg: 55
 - False pos: 1
 - False neg: 14
- **Accuracy: 0.838710**
- **Sensitivity: 0.621622**
- **Specificity: 0.982143**

Probability-Based Calling (continued)

- Problem: some loci may be more indicative of MSI status than others
 - Need weights assigned to each measure at each locus individually

A Bigger ML Problem

- Solution: instead of individually looking at one locus with m features, consider all n loci with all m features at once
 - Machine learning problem in $n \times m$ dimensions
- Exclude loci with too few usable files

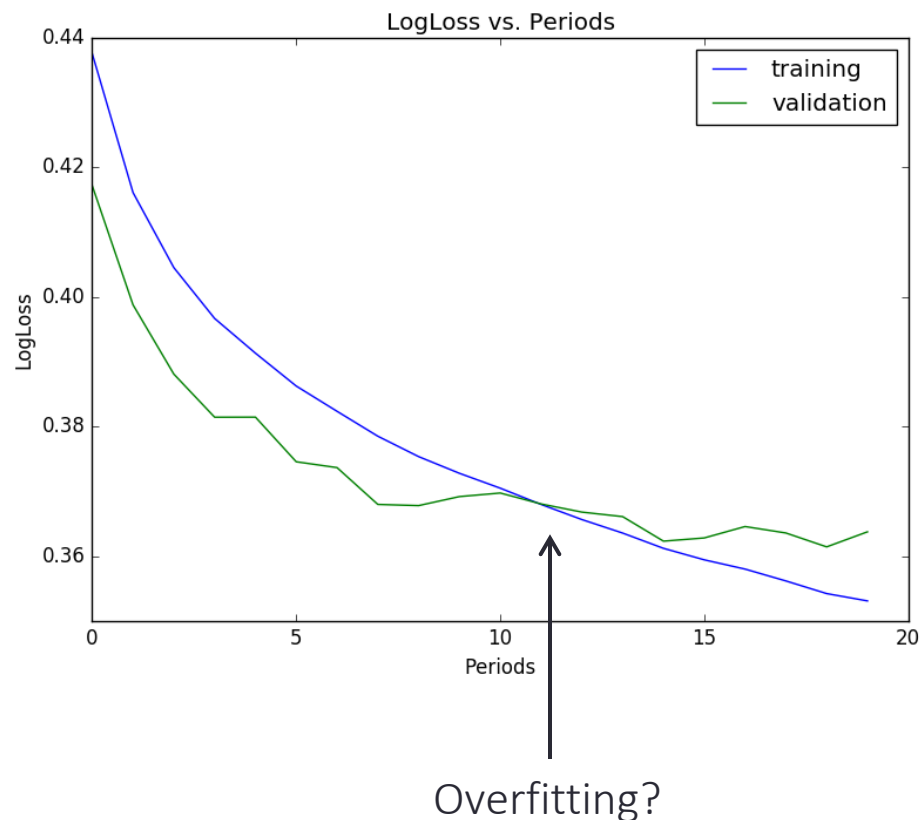
Benefits

- Can report a probability instead of a binary call
- Easy to add new loci
 - Will be considered relative to other loci
- Easy to choose the most relevant loci
 - Defining a selector for STAMP

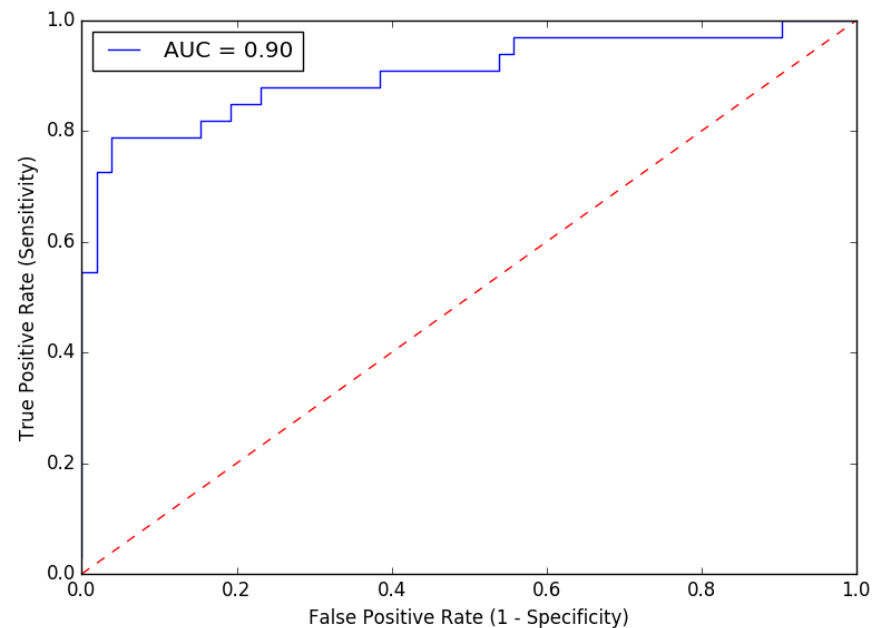
Machine Learning v2 – Setup

- 21 loci x 4 features each
 - 84-dimensional model
- Training Set
 - 266 cases
 - 124 MSI, 142 MSS
- Validation Set
 - 87 cases
 - 33 MSI, 52 MSS
- Test Set
 - 93 cases
 - 38 MSI, 55 MSS

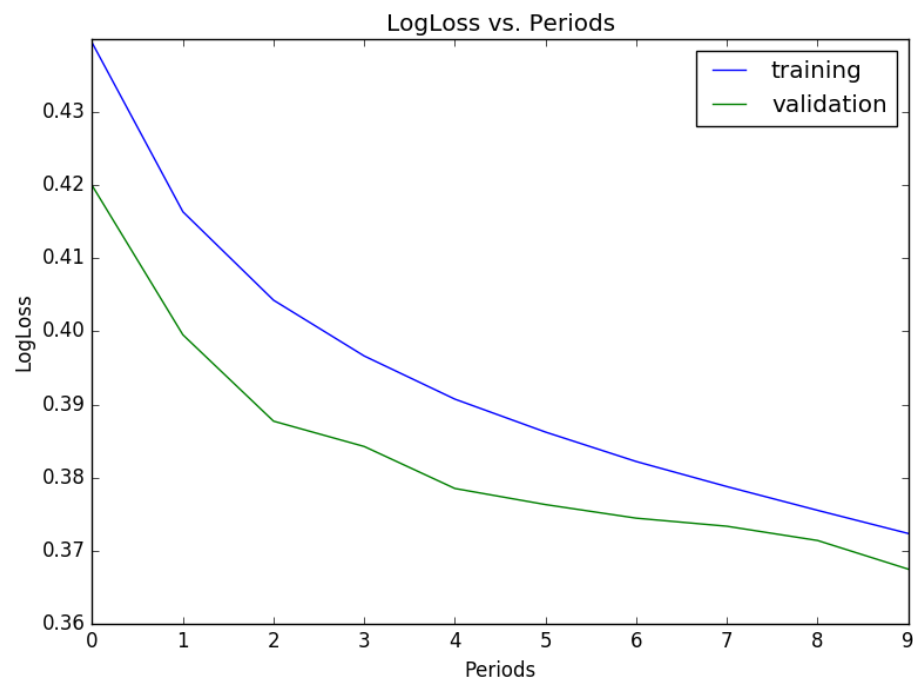
Machine Learning v2 - Results



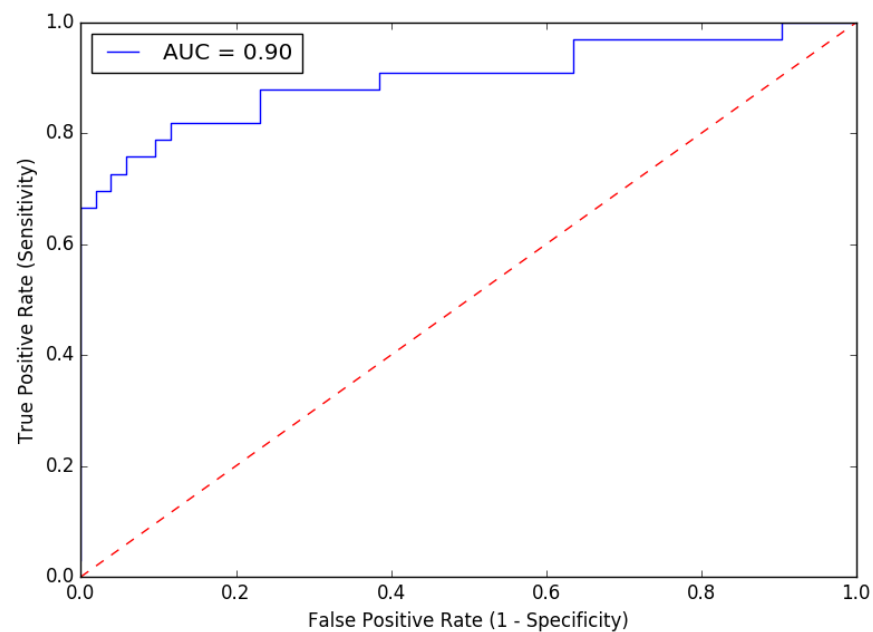
Learning rate = 0.01
Steps = 200,000
Batch size = 10



Machine Learning v2 - Results



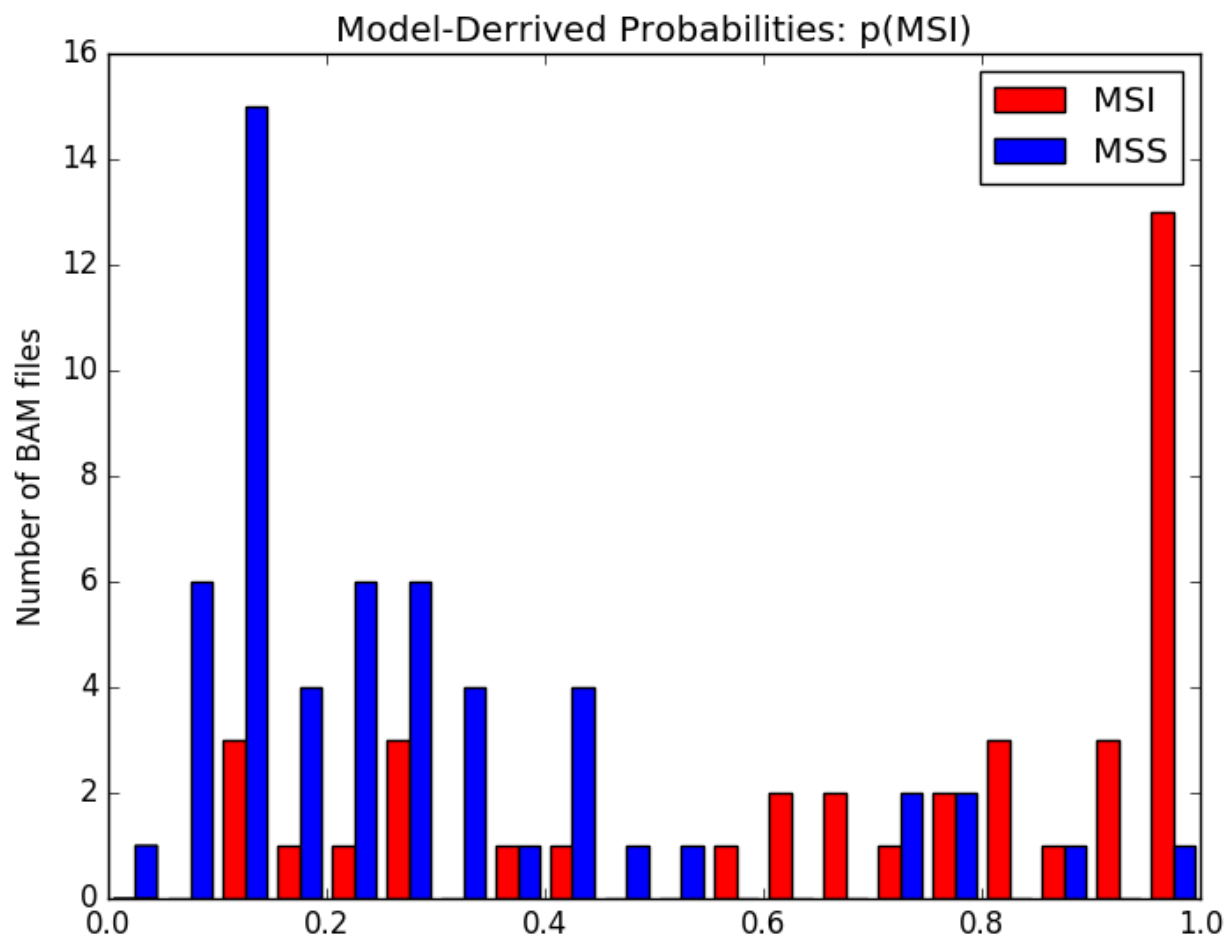
Learning rate = 0.01
Steps = 100,000
Batch size = 10



Machine Learning v2 – Test Set

- Loci examined
 - MSI-11
 - MSI-14
 - H-10
 - HSPH1-T17
 - BAT-26
 - BAT-25
 - MSI-04
 - MSI-06
 - MSI-07
 - MSI-01
 - MSI-03
 - MSI-09
 - H-09
 - H-08
 - H-01
 - H-03
 - H-02
 - H-04
 - H-07
 - H-06
 - H-05
- Threshold: 0.500000
- Total files: 93
- Correct predictions: 76
 - True pos: 28
 - True neg: 48
 - False pos: 7
 - False neg: 10
- Accuracy: 0.817204
- Sensitivity: 0.736842
- Specificity: 0.872727

Machine Learning v2 – Test Set



Summary

- Best AUC of $\approx 90\%$
 - Measures the predictive ability of the model
- Maximum accuracy of $\approx 82\%$ on test set
- Machine learning using all loci and all features is likely the best option
 - More samples with better depth to train on

Further Investigations

Improve Existing Model

- Get more BAM files
 - Training, validation and test sets are stale
 - Get BAM files with better coverage for more accurate results, better prediction
- Research clinical relevance of MSI diagnosis
 - Fine-tune parameters to favor either false positives or false negatives
- Regularization
 - L2 regularization
 - Penalizes any weight that is too large, drives weights asymptotically to 0
 - L1 regularization
 - Penalizes total weight of all features, drives non-informative weights to 0

Further Investigations

Expand Existing Model

- Add features
 - Earth mover distance
- Add a label for 'MSI-L' and 'MSI-H'
- Investigate more loci
 - Hause et al. used 223,082 microsatellite loci to develop a MSI calling method with 96.6% accuracy
- Train using matched normal samples
 - Detect low, baseline level of MSI in tumor samples

Further Investigations

Investigate New Models

- Linear classifier models using different features
 - Feature crosses
- Neural nets
 - Multi-class neural nets
 - Require more training, validation and test data, overfitting concern