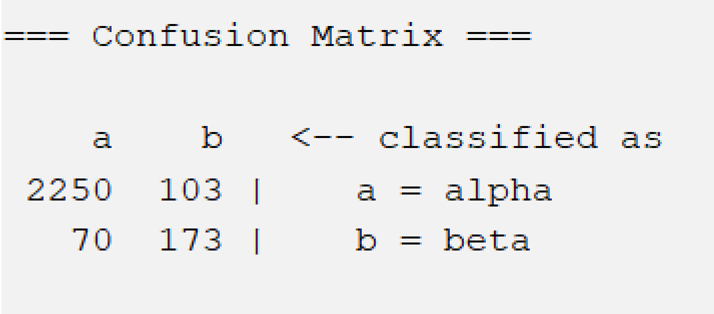
Computational Biology Lab

BE21B037 P2

1. Install Weka (https://waikato.github.io/weka-wiki/downloading_weka/)
2. Prepare the input file in .arff format using amino acid composition (see example)
3. Open the file using Weka
4. Classify them using different machine learning techniques (select any 10 from different classes/sub-classes)
5. Use training set and compute sensitivity, specificity and accuracy from the confusion Matrix.

The 10 ML techniques used are:

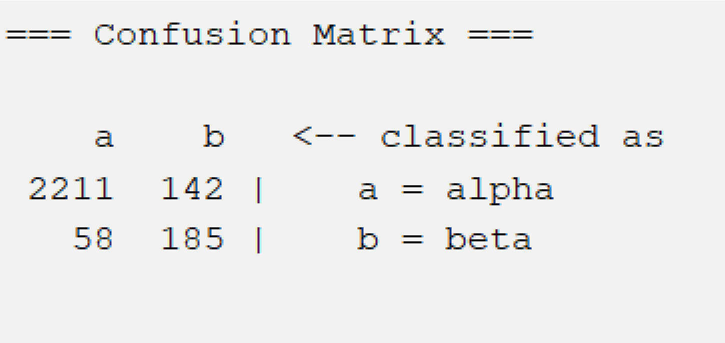
1) bayes.BayesNet



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2250  103 |    a = alpha
   70  173 |    b = beta
```

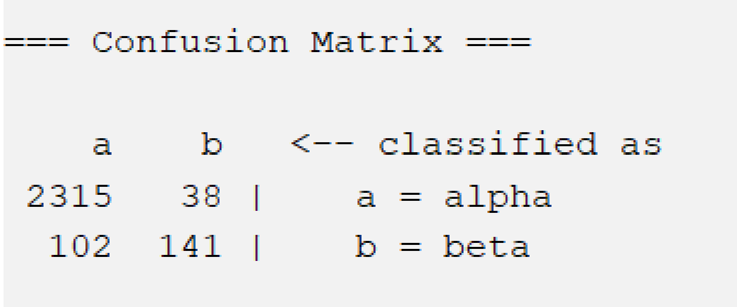
2) bayes.Naive Bayes



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2211  142 |    a = alpha
   58  185 |    b = beta
```

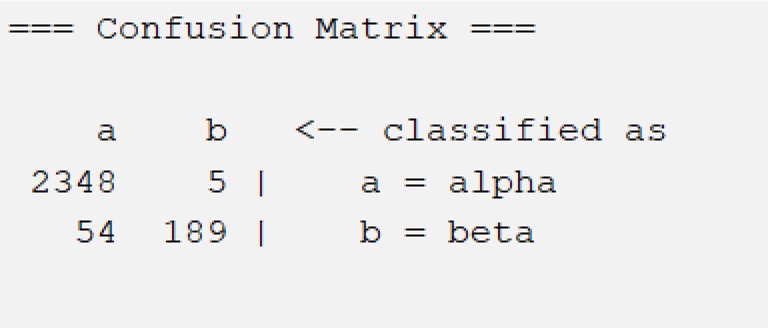
3) functions.Logistic



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2315   38 |    a = alpha
  102  141 |    b = beta
```

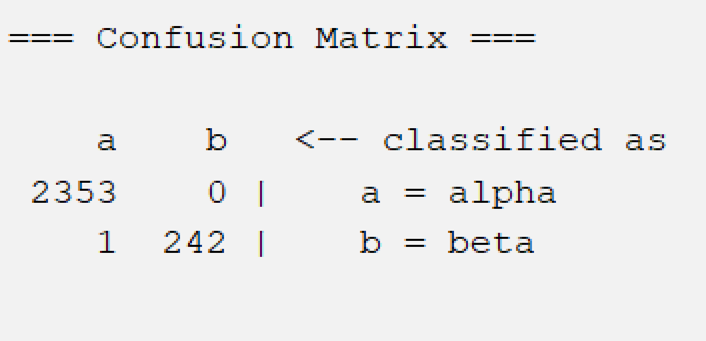
4) functions. Multilayer Perceptron



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2348    5 |    a = alpha
   54  189 |    b = beta
```

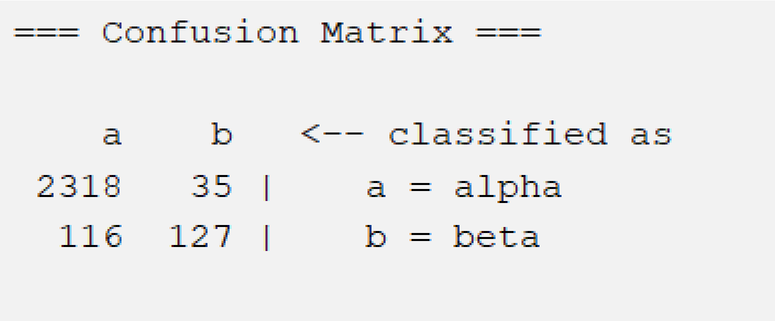
5) lazy.KStar



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2353    0 |    a = alpha
    1  242 |    b = beta
```

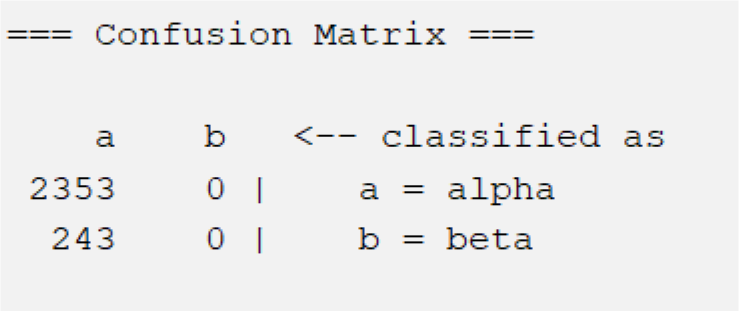
6) rules. DecisionTable



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2318   35 |    a = alpha
  116  127 |    b = beta
```

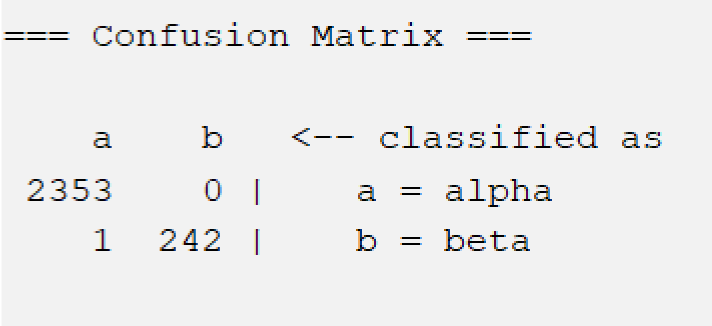
7) trees. DecisionStump



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2353    0 |    a = alpha
  243    0 |    b = beta
```

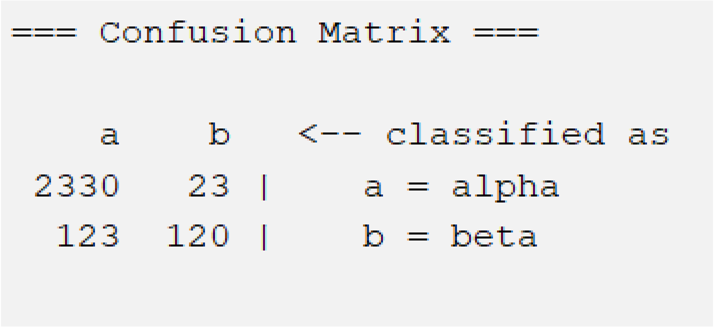
8) trees.Random Forest



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2353    0 |    a = alpha
    1  242 |    b = beta
```

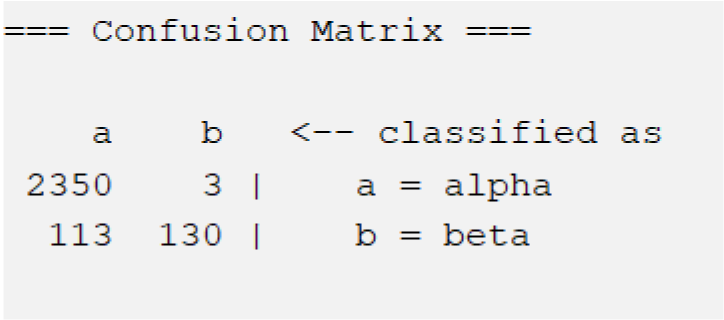
9) meta.LogitBoost



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2330   23 |    a = alpha
  123  120 |    b = beta
```

10) meta.Bagging



```
=== Confusion Matrix ===

    a    b   <-- classified as
 2350    3 |    a = alpha
  113  130 |    b = beta
```

Using the confusion matrix we can calculate the sensitivity, specificity and accuracy which has been tabulated below

Method	TP	TN	FP	FN	sensitivity	specificity	accuracy
BayesNet	2250	173	70	103	0.956	0.712	0.933
Naive Bayes	2211	185	58	142	0.940	0.761	0.923
Logistic	2315	141	102	38	0.984	0.580	0.946
Multilayer Perceptron	2348	189	54	5	0.998	0.778	0.977
KStar	2353	242	1	0	1.000	0.996	1.000
DecisionTable	2318	127	116	35	0.985	0.523	0.942
DecisionStump	2353	0	243	0	1.000	0.000	0.906
Random Forest	2353	242	1	0	1.000	0.996	1.000
LogitBoost	2330	120	123	23	0.990	0.494	0.944
Bagging	2350	130	113	3	0.999	0.535	0.955

Alpha - Positive
Beta - Negative

6. Repeat with 5-fold, 10-fold, 20-fold and 66% split cross-validations. Select the best method from the performance

Method	5-Fold			10-Fold		
	sensitivity	specificity	accuracy	sensitivity	specificity	accuracy
BayesNet	0.949	0.650	0.921	0.951	0.654	0.923
Naive Bayes	0.941	0.733	0.922	0.938	0.737	0.919
Logistic	0.981	0.564	0.942	0.980	0.547	0.940
Multilayer Perceptron	0.968	0.593	0.933	0.972	0.634	0.940
KStar	0.965	0.523	0.924	0.965	0.523	0.924
DecisionTable	0.985	0.399	0.930	0.985	0.383	0.929
DecisionStump	1.000	0.000	0.906	1.000	0.000	0.906
Random Forest	0.995	0.453	0.943	0.993	0.469	0.944
LogitBoost	0.982	0.481	0.935	0.985	0.486	0.938
Bagging	0.989	0.424	0.936	0.991	0.453	0.941

Method	20-Fold			66% split		
	sensitivity	specificity	accuracy	sensitivity	specificity	accuracy
BayesNet	0.950	0.654	0.923	0.955	0.623	0.926
Naive Bayes	0.938	0.745	0.920	0.943	0.753	0.926
Logistic	0.982	0.564	0.943	0.983	0.532	0.943
Multilayer Perceptron	0.966	0.597	0.931	0.981	0.597	0.948
KStar	0.966	0.527	0.924	0.968	0.519	0.929
DecisionTable	0.985	0.366	0.927	0.964	0.403	0.915
DecisionStump	1.000	0.000	0.906	1.000	0.000	0.913
Random Forest	0.993	0.465	0.943	0.995	0.442	0.947
LogitBoost	0.986	0.481	0.939	0.984	0.442	0.937
Bagging	0.992	0.424	0.939	0.988	0.429	0.939

Alpha - Positive
Beta - Negative

We pick Naive Bayes as the algorithm with the best performance, since we need the model to best classify beta values. The sensitivity values are all close to 0.9 but specificity values are lower, so we pick the model with the best specificity reading, which happens to be naive bayes model

7. Keep 70%, 60% and 50% of the data as training and use others as test set to evaluate the performance of the best method.

Method	Percentage	sensitivity	specificity	accuracy
Naive Bayes	70%	0.945	0.753	0.927
	60%	0.930	0.800	0.919
	50%	0.939	0.764	0.922

8. Evaluate the importance of each residue by eliminating each residue (repeat 20 times). See the decrease in performance.

Method	removed aa	sensitivity	specificity	accuracy
Naive Bayes	A	0.938	0.765	0.922
	C	0.937	0.770	0.921
	D	0.941	0.741	0.923
	E	0.941	0.753	0.923
	F	0.947	0.737	0.927
	G	0.928	0.761	0.913
	H	0.940	0.749	0.922
	I	0.928	0.761	0.913
	K	0.935	0.749	0.918
	L	0.946	0.724	0.925
	M	0.944	0.720	0.923
	N	0.935	0.708	0.913
	P	0.933	0.761	0.917
	Q	0.941	0.741	0.922
	R	0.941	0.761	0.924
	S	0.934	0.753	0.917
	T	0.941	0.753	0.923
	V	0.941	0.757	0.924
	W	0.941	0.761	0.924
	Y	0.939	0.765	0.923

We see that the residue that decreases the accuracy the most is amino acids, G, I and N.
Which are glycine, isoleucine, asparagine respectively.

9. Analyze the same using only one residue at the time.

Method	only aa	sensitivity	specificity	accuracy
Naive Bayes	A	1.000	0.000	0.906
	C	1.000	0.000	0.906
	D	0.992	0.008	0.900
	E	1.000	0.000	0.906
	F	1.000	0.000	0.906
	G	0.999	0.000	0.905
	H	1.000	0.000	0.906
	I	1.000	0.000	0.906
	K	1.000	0.000	0.906
	L	1.000	0.000	0.906
	M	1.000	0.000	0.906
	N	0.977	0.140	0.899
	P	1.000	0.000	0.906
	Q	0.997	0.000	0.904
	R	1.000	0.000	0.906
	S	1.000	0.000	0.906
	T	0.994	0.016	0.902
	V	1.000	0.000	0.906
	W	1.000	0.000	0.906
	Y	0.996	0.000	0.903

We see the residues that cause the algorithm to perform the best is N which is asparagine. The specificity is higher than the for other residues while still maintaining the accuracy and sensitivity.

10. Construct a balanced dataset (243 each for alpha and beta) and obtain the results with 5-fold cross validation.

Method (5-fold)	TP	TN	FP	FN	sensitivity	specificity	accuracy
Naive Bayes	185	209	34	58	0.761	0.860	0.811
Naive Bayes with unbalanced dataset					0.941	0.733	0.922

11. Tabulate and discuss the results.

The tabulated results have been attached above for each of the specific questions. When picking the best ML model we picked the one that had high, specificity, sensitivity and accuracy. Since all the ML models had > 90% accuracy, the key factor was specificity, due to the unbalanced dataset which contains more alpha compositions than beta. We see that **all three performance factors decrease**, when the data is balanced.

The files used can be found here(balanced dataset).