

Applying K-means (or flowPeaks) and Support vector machines to the sample classification problem using the flow cytometry data

Yongchao Ge

Mount Sinai School of Medicine

FlowCAP II summit, Sept 22-23

Outline

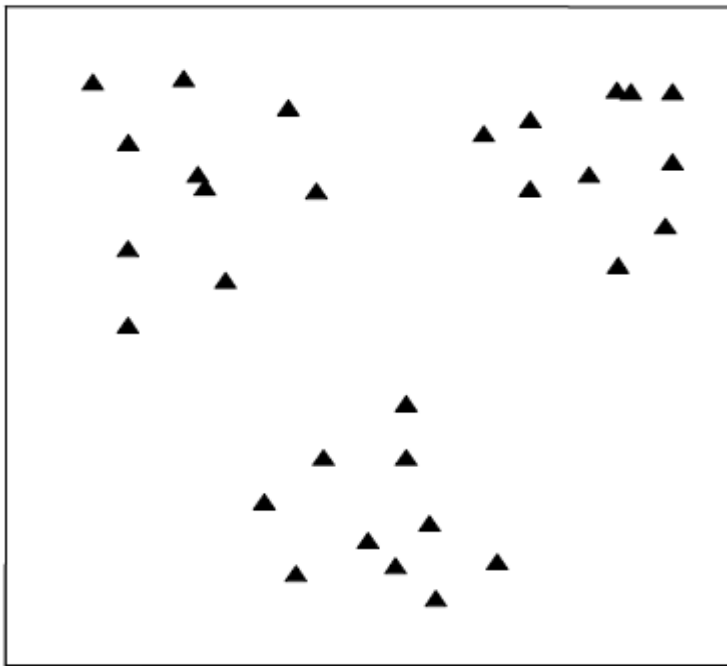
- Objective
- Kmeans and flowPeaks
- Support vector machine
- Results and discussion

Objective

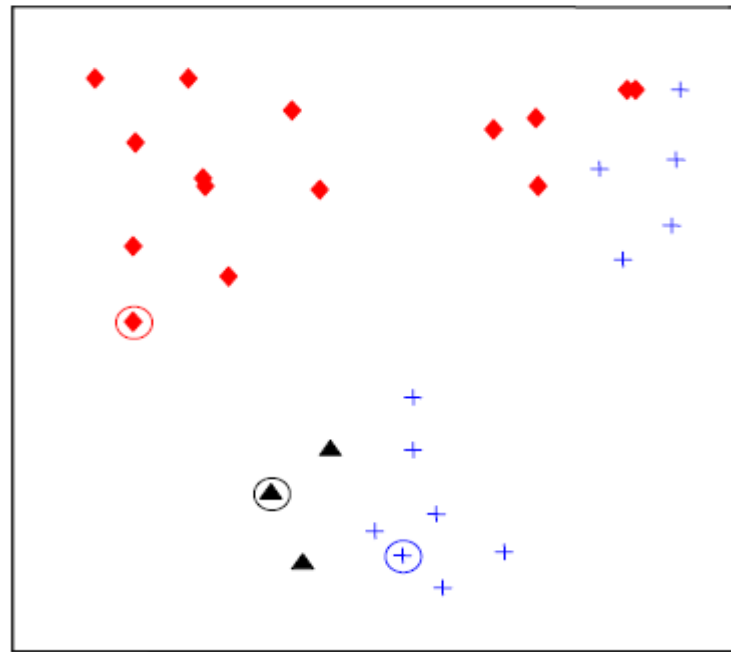
- To build a good classifier for the flow cytometry data
- Step 1: data reduction --- kmeans, flowPeaks
- Step 2: machine learning --- support vector machines
- Using cross-validations to assess the algorithm performance

Kmeans algorithm

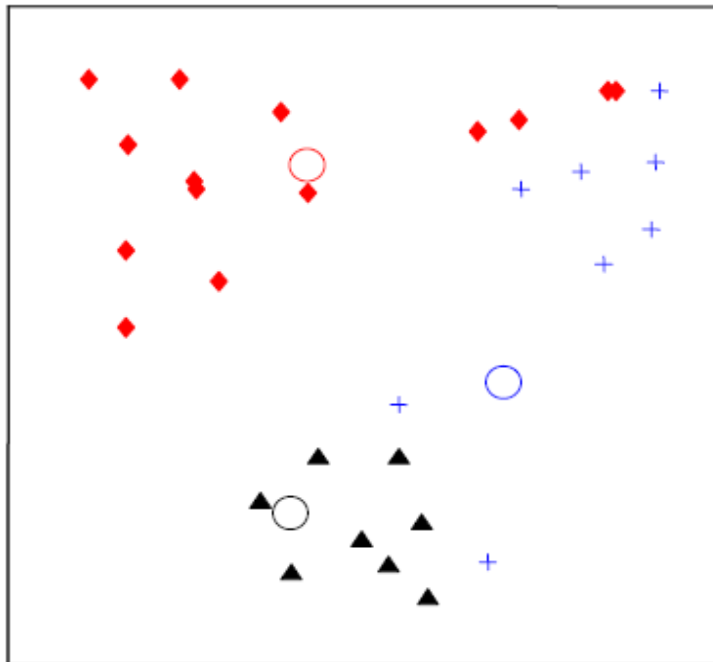
- MacQueen (1967) used the name Kmeans, the idea goes back to Steinhaus (1957), Lloyd (1982) algorithm was proposed in 1957 (wikipedia).
- An iterative algorithm with the two steps:
 - Cluster assignment
 - Center update
- Critical parameters:
 - Initial seeds
 - The number of clusters K



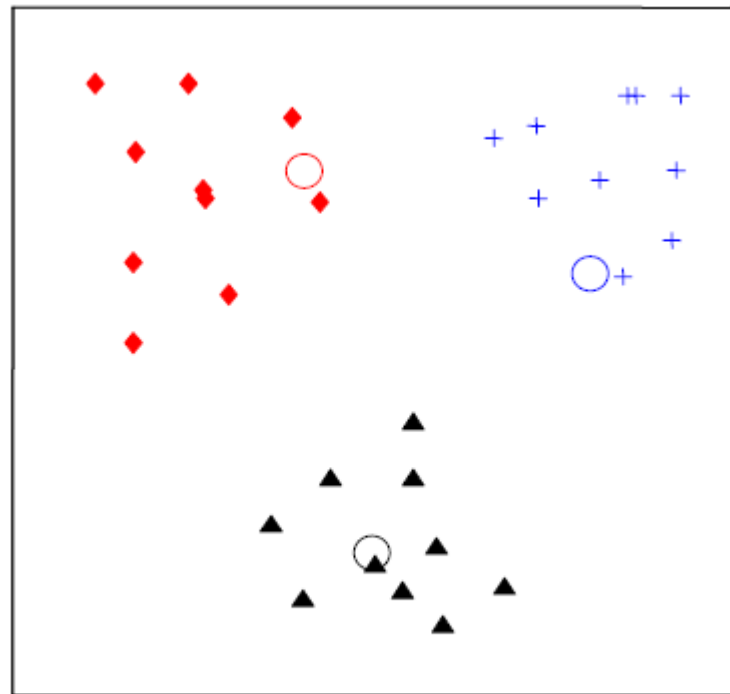
(a) Input data



(b) Seed point selection



(c) Iteration 2



(d) Iteration 3

Kmeans implementation details

- The initial seeds are generated by k-means++, which tries to generate the seeds that are well separated
- The data are organized by k-d tree to increase the computation speed
- The determination of K . Roughly \sqrt{n} , we fix it to be 300 for flowCAP challenges.
 1. to keep as many features as possible
 2. to make sure the estimate of the proportion has a smaller variance

flowPeaks (manuscript in progress)

- It uses the initial Kmeans clustering to build the density function and compute all of the peaks of the density function
- The data are reclassified by the local peaks of the density function
- A web interface will be up on the lab site soon.

Support vector machines

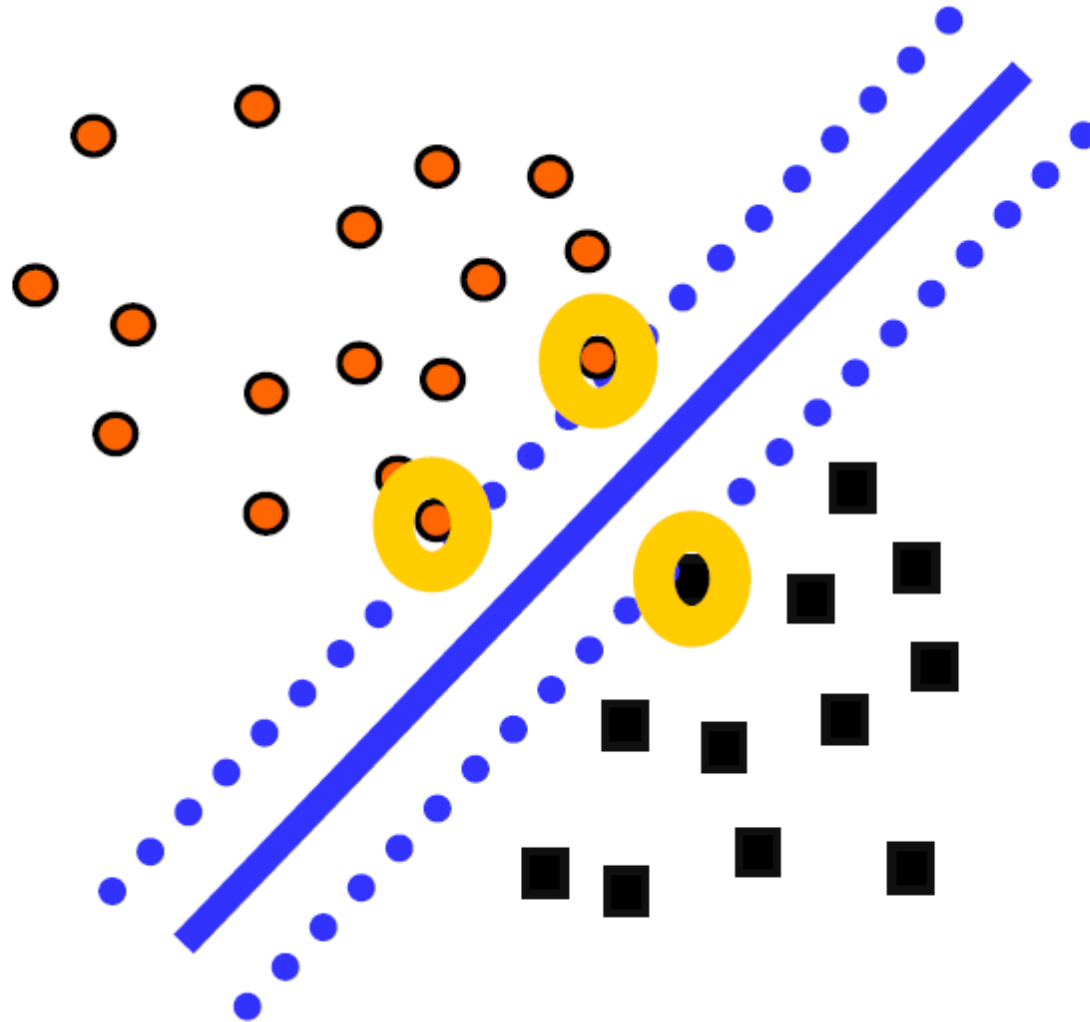
For a two-class linear separable problem, the goal is to find the a hyperplane that is furthest from both classes (or maximize the margins)

The problem setup

$$\begin{aligned} & \min_{w,b} \|w\|^2 / 2 \\ \text{s.t.} \quad & w \cdot x_i - b \geq 1 \quad y_i \in \text{class } 1 \\ & w \cdot x_i - b \leq -1 \quad y_i \in \text{class } -1 \end{aligned}$$

Where x_i is the data and y_i is the class label.

An example



Adapted from Bennett and Campbell, 2000

What if the space not linear separable

$$\min_{w,b} \left\| w \right\|^2 / 2 + C \cdot \sum_{i=1}^n z_i$$

$$\begin{aligned} \text{s.t.} \quad & w \cdot x_i - b \geq 1 - z_i && y_i \in \text{class 1} \\ & w \cdot x_i - b \leq -1 + z_i && y_i \in \text{class -1} \end{aligned}$$

Where relaxed variables z_i and the penalty cost C are non-negative.

SVM

Support vector machine is a very powerful machine learning algorithm

1. It does not have the over-fitting problem for the high dimensional data.
2. It can accommodate nonlinear classification by applying suitable kernels
3. The computation is fast
4. It extends to more than two classes, regression, and novelty detection.

Software choice

There are a lot of implementations available for SVM. libsvm, R package e1071, weka, SVM^{light}

Our choice is WEKA as it offers different machine learning algorithms and different SVM implementations. We fixed the optimization by SMO.

We used linear kernel with degree of 1 with the penalty cost $C=1$.

Variable normalization and filtering

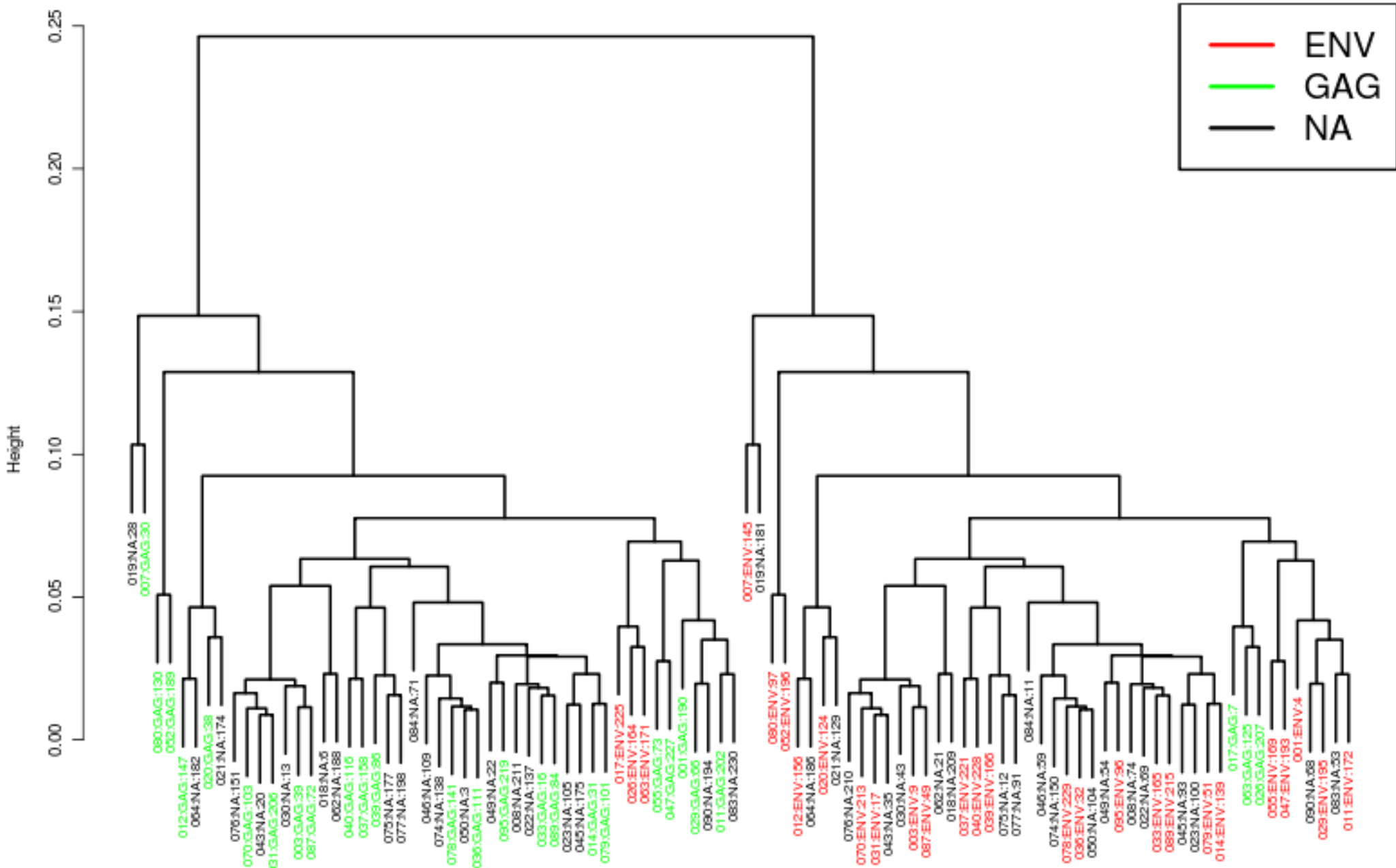
A variable is filtered out if its mean proportion between the two groups is less than 0.1% or the pval from the t-test is greater than 0.5

The remaining variables are normalized with mean 0 and std 1.

Challenge 3A

1. Randomly take 5000 cells from each file, to combine into a big file and compute the K-means with $K=300$
2. Apply the 300 centers to each file to compute the prop. of cells belonging to each center
3. Normalize the prop vector by subtracting the average of the two antigens
4. Apply the SVM to the normalized proportions.

Cluster Dendrogram



A naïve hierarchical clustering can give a classifier with 6 errors out of 54.

For 100 trials of three fold random cross-validations, 99 trials give zero errors out of 54, and 1 trial give two errors.

Using negctrls to normalize the data

- Alternatively, we compute the K-means include the data from the negctrls and NAs, in addition to the two antigens
- Using the average of the prop vectors from the negctrls to normalize the data
- The hierarchical clustering is not that clean
- The SVM gives identical prediction as before
- 100 trials of cross-validation of the training data set gives more errors

Cross-validation comparisons

For 100 trials of 3 fold cross-validations.

		# of errors out of 27						
		0	1	2	3	4	5	6
Normalize with antigens	ENV	99	1					
	GAG	99	1					
Normalize with negctrls	ENV	8	29	33	17	11	1	1
	GAG	89	11					

Results for Challenge 2

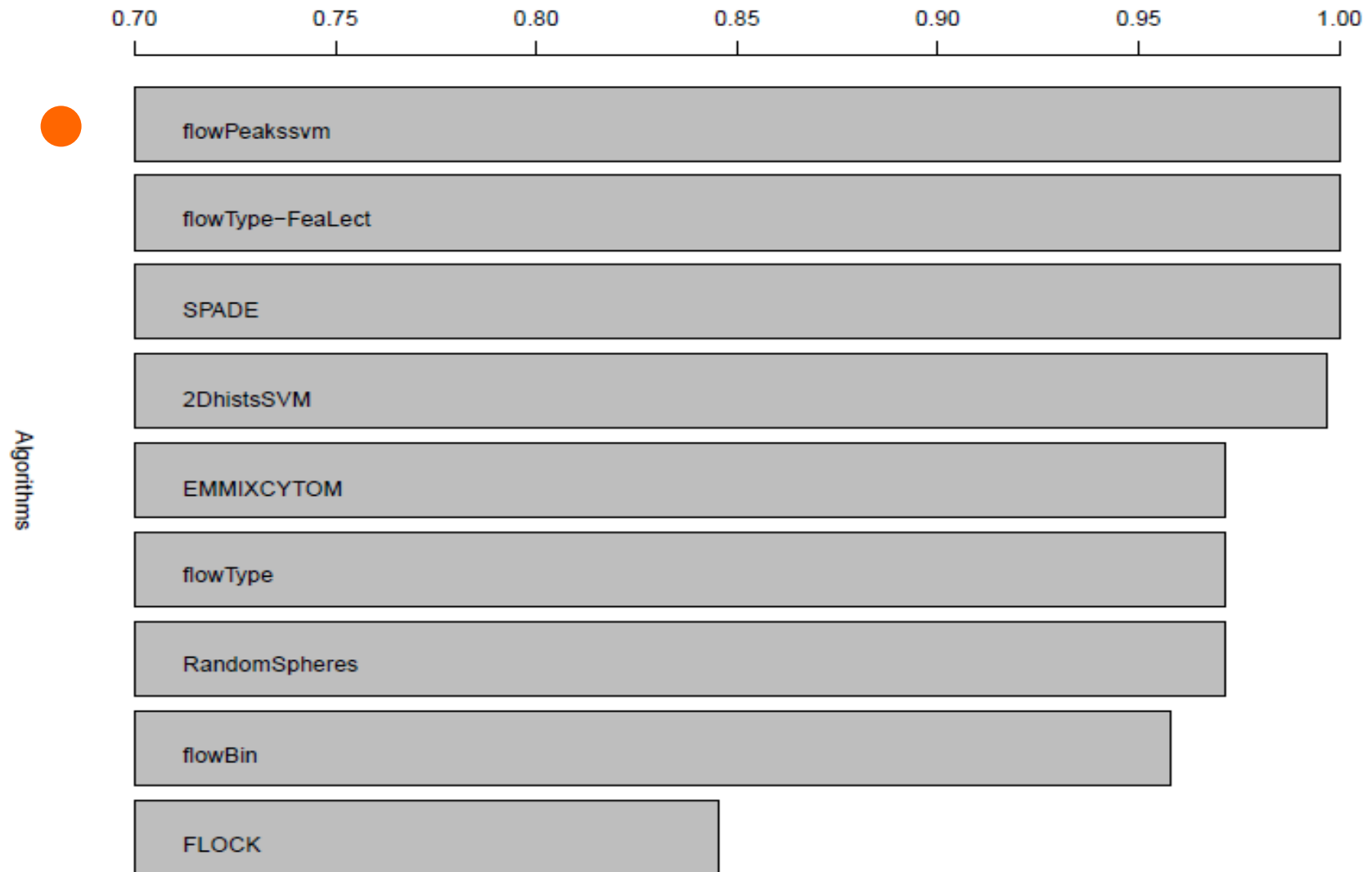
- Compute the flowPeaks for each tubes and each group of patients (aml and normal) independently
- For each patient, compute the 16 prop vectors (8 tubes x 2 conditions) using the results from flowPeaks
- Using the 16 prop. Vectors as the input to the SVM.

Cross-validation comparisons

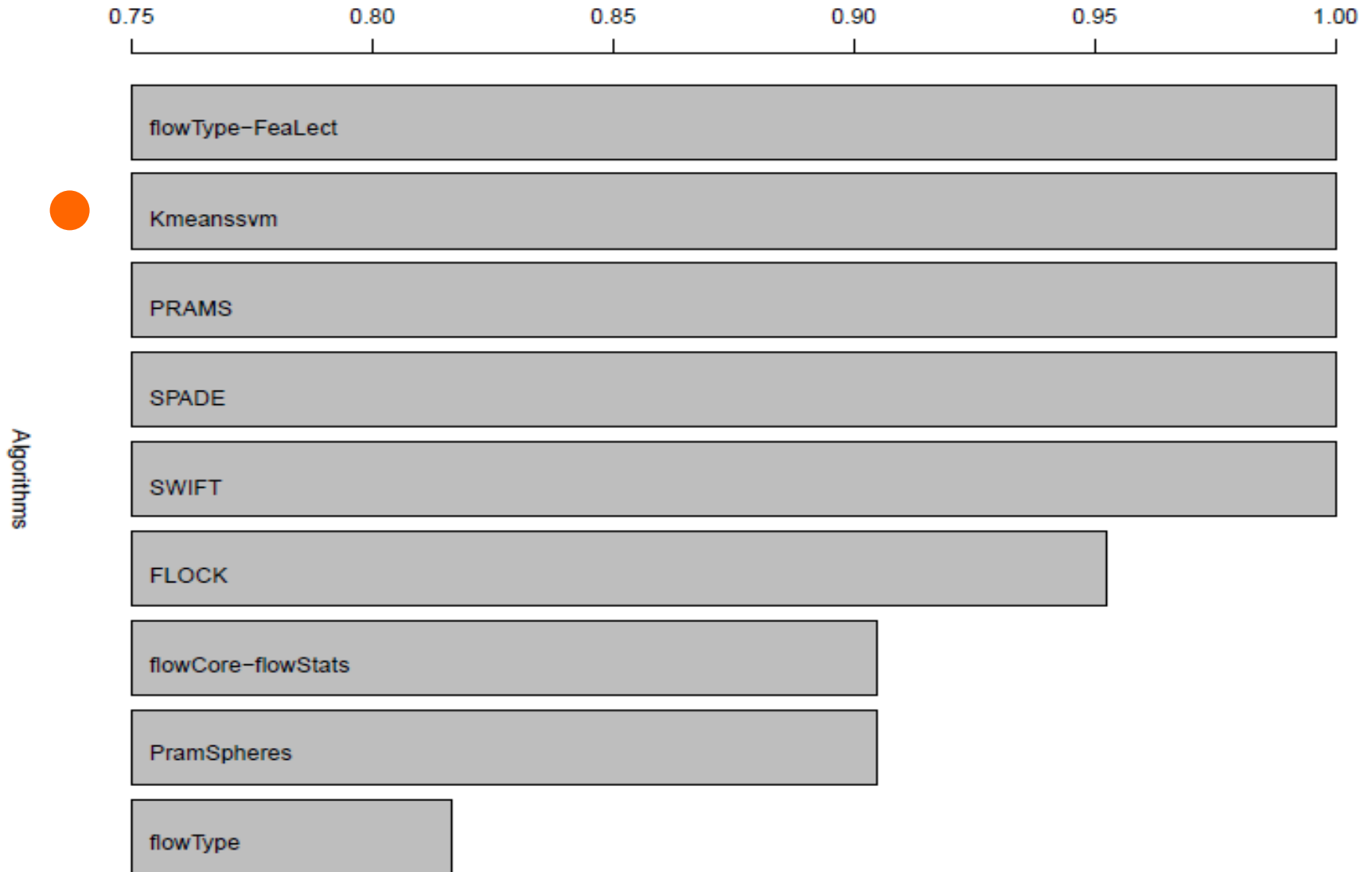
For 100 trials of 3 fold cross-validations.

		# of errors						
		0	1	2	3	4	5	>5
flowPeaks	Aml (23)		8	25	28	21	13	5
	Normal (156)	44	30	19	6	1		
Kmeans	Aml				14	28	31	27
	Normal	16	58	23	3			

Challenge 2: flowPeakssvm



Challenge 3A: Kmeanssvm



Compare the cross- validations and independent test set

The cross-validations give higher error rates. The ideal cross-validations should include the building of the k-means of flowPeaks, which may raise the error rates slightly

Possible explanations?

- The “training data” is the cross-validation is smaller
- The independent test data is less challenging than the training data set

Discussions and Caveats

- Different normalization gives different strengths
- The machine learning can be improved with a better selection of the parameters to train the SVM.

Acknowledgement

- Stuart Sealfon
- Fernand Hayot
- Istvan Sugar
- Boris Hartman
- Maria Chikina
- Grant support: NIH/NIAID HHSN272201000054C