Qlucore Omics Explorer 3.2

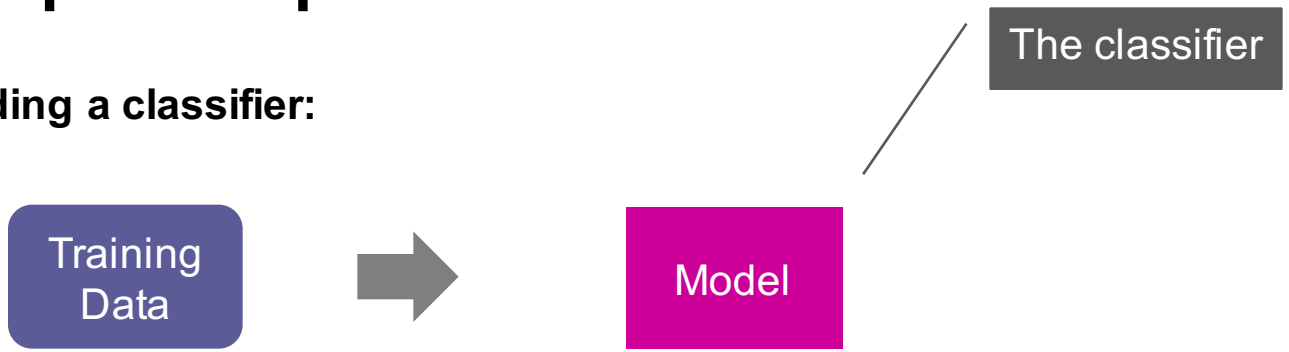
QLUCORE®

Classification – Pattern recognition Supervised Machine Learning

- **Purposes:**
  - Build a classifier
  - Use a classifier to classify “new” samples
- **Methods:**
  - kNN
  - Support Vector Machines (SVM)
  - Random Trees (RT)
- **Use:**
  - Build a classifier (a model of your data) to enable classification of “new” samples. For example, predict/clarify/confirm disease stages for the samples when the stage is not known, or is unclear.

Basic principles

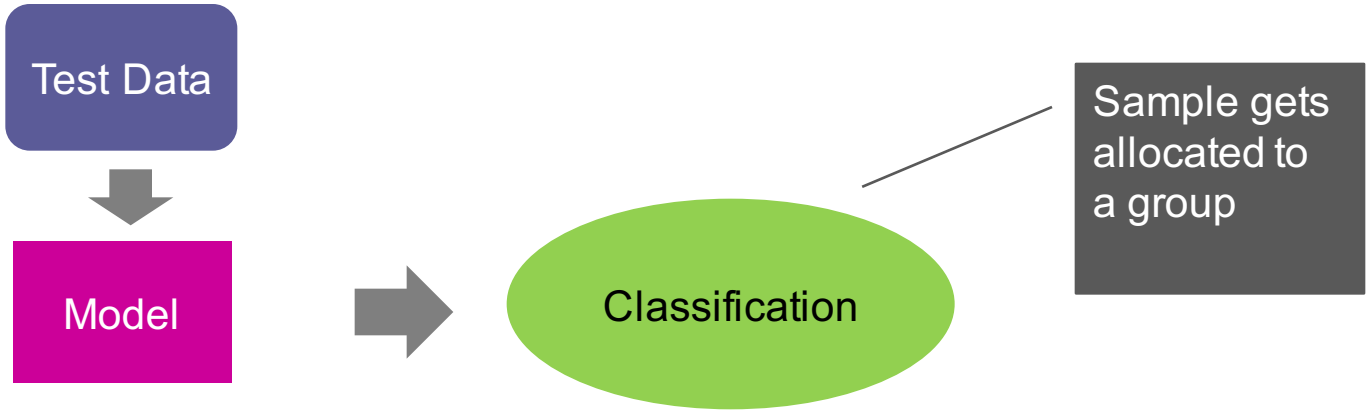
Building a classifier:



KEY

Selecting type of classifier method
Ranking
Optimizing hyper parameters

Using a classifier:



QLUCORE

Benefits

- Visually-guided work flow
- Three powerful methods (kNN, SVM and RT)
- Select method and basic parameters – in a few clicks
- In-built validation (based on cross-validation) in addition to the option to use an external validation data set
- Classify new sample(s) by one click
- Extensive report, and direct quality feedback by ROC/AUC plots (if using external validation dataset with two groups in Key annotation, SVM and Random trees)

Classifier algos: *k*-NN

Classifier – this term refers to the machine learning algorithm:
here K-NN, SVM and RT

1. ***k*-Nearest Neighbors algorithm** (or ***k*-NN**) is a non-parametric, wighted method.

The input consists of the *k* closest training examples in the feature space.

It is an example of lazy learning, where the function is only approximated locally and all computation is deferred until classification.

The *k*-NN algorithm is among the simplest of all machine learning algorithms. For example, a common weighting scheme consists in giving each neighbor a weight of 1/*d*, where *d* is the distance to the neighbor.

Classifier algos: SVM

2. S**upport vector machines** (**SVMs**) are supervised learning models with associated learning algorithms. Linear and Non-parametric. Non-probabilistic model.

An SVM model is a representation of the training set examples as points in space, mapped so that separate groups are divided by a clear gap (that is as wide as possible).

New data to be classified are then mapped into that same space and predicted to belong to a group based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly.

Posthoc interpretation of SVM to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

Classifier algos: Random Trees

3. **Random Trees (RT, the same as Random Decision Forests** are an ensemble learning, which operate by constructing a multitude of decision trees at training time and outputting the class of the individual trees.

Both RT and KNN can be viewed as *weighted neighborhoods schemes*.

RT predictors naturally lead to a dissimilarity measure between the observations. The idea is to construct an RT predictor that distinguishes the "observed" data from suitably generated synthetic data. The observed data are the original unlabeled data and the synthetic data are drawn from a reference distribution.

An RT dissimilarity can handle mixed variable types well, is invariant to monotonic transformations of the input variables, and is robust to outlying observations.

The RT dissimilarity has been used in a variety of applications, e.g. to find groups of patients based on tissue marker data.[21]

Building a classifier - workflow

- During the build process, algorithms test a number of different combinations of variables and classifier parameters in order to find the optimal one, which enables the best classifier performance:
- Training of several classifier candidates using different variable sub-sets and parameters;
- Each candidate is trained using stratified k-fold cross-validation;
- The parameter and variable combination that yields the best result is then used to train a final classifier.

Look at the UI

Normalization

- When a classifier is used, the quality, as well as the pre-processing steps are very important
- **Main rule: pre-processing and normalization have to be exatly the same between the data used to create the classifier as the data that is being classified**
- The normal recommendation is use the Normalization Mean = 0 and Var = 1 (method tab) in all cases when the data to be classified has a reasonable number of samples
- In QOE it possible to automatically apply the normalization parameters from the data set used to create the classifier also to the sample(s) being classified

Other Considerations

- Normalization
- Noise level and nature in training data and data to classify (the closer the better)
- Stringency - depends on your application
- Min acceptable Accuracy – depends on application, signal strenght for the annotation of interest. See the number of mis-classifications in the build report