Session 1: Statistical Machine Learning

Prof Jean Yang, Dr Dario Strbenac, Dr Ellis Patrick, Dr Shila Ghazanfar

29 June 2018

THE UNIVERSITY OF SYDNEY

Roadmap

- Part 1: Introduction to statistical machine learning
    - Using R code to build classification models with RNA-seq or microarray data and basic performance assessment: 90 minutes.
- Afternoon tea: 30 minutes.

Roadmap

- Part 1: Introduction to statistical machine learning
    - Using R code to build classification models with RNA-seq or microarray data and basic performance assessment: 90 minutes.
- Afternoon tea: 30 minutes.
- Part 2: Performance assessment with cross-validation
    - Understanding the ClassifyR package and using cross-validation to assess an existing classifier: 80 minutes.
- Final wrap up - overview of the latest methods on biologically guided machine learning approaches: 10 minutes.

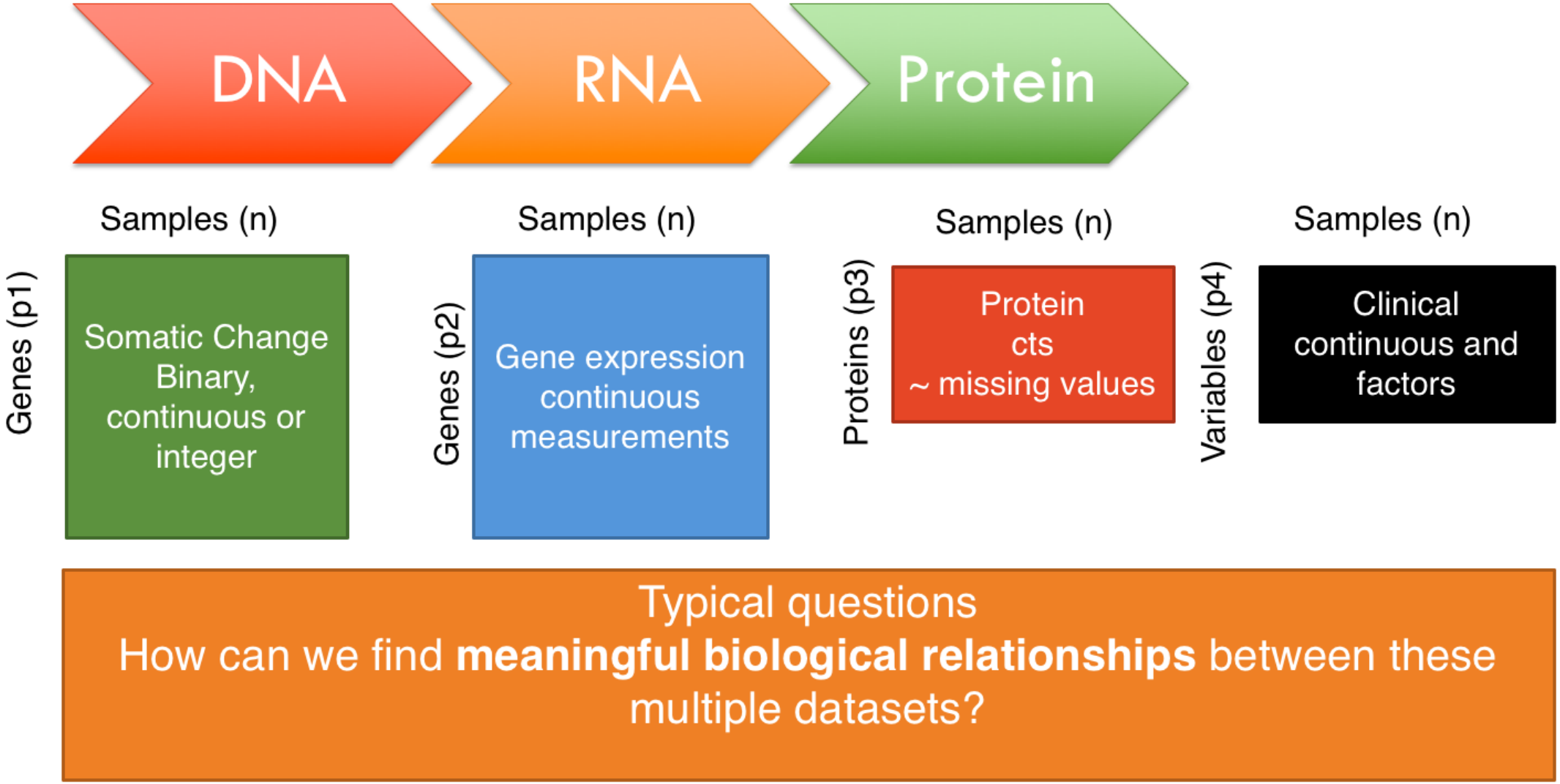
Terminology

- Statistical machine learning
  - Unsupervised: classes unknown, want to discover them from the data (cluster analysis)
  - Supervised: classes are predefined, want to use a (training or learning) set of labeled objects to form a classifier for classification of future observations
- Alternative terminology
  - Computer science: unsupervised and supervised learning.
  - Bioinformatics literature: class discovery and class prediction.
  - Statistics: Clustering and classification or discriminant analysis.

Finding omics data online

- Short Read Archive http://www.ncbi.nlm.nih.gov/Traces/sra
- Gene Expression Omnibus https://www.ncbi.nlm.nih.gov/geo/
- The Cancer Genome Atlas https://cancergenome.nih.gov/
- Synapse https://www.synapse.org
- Recount2 https://jhubiostatistics.shinyapps.io/recount/

What does biomedical data look like?



Typical questions

How can we find **meaningful biological relationships** between these multiple datasets?

Possible input ?

Samples (n)

Genes (p1)

Somatic Change
Binary, continuous
or integer

Samples (n)

Genes (p2)

Gene expression
continuous
measurements

Samples (n)

Proteins (p3)

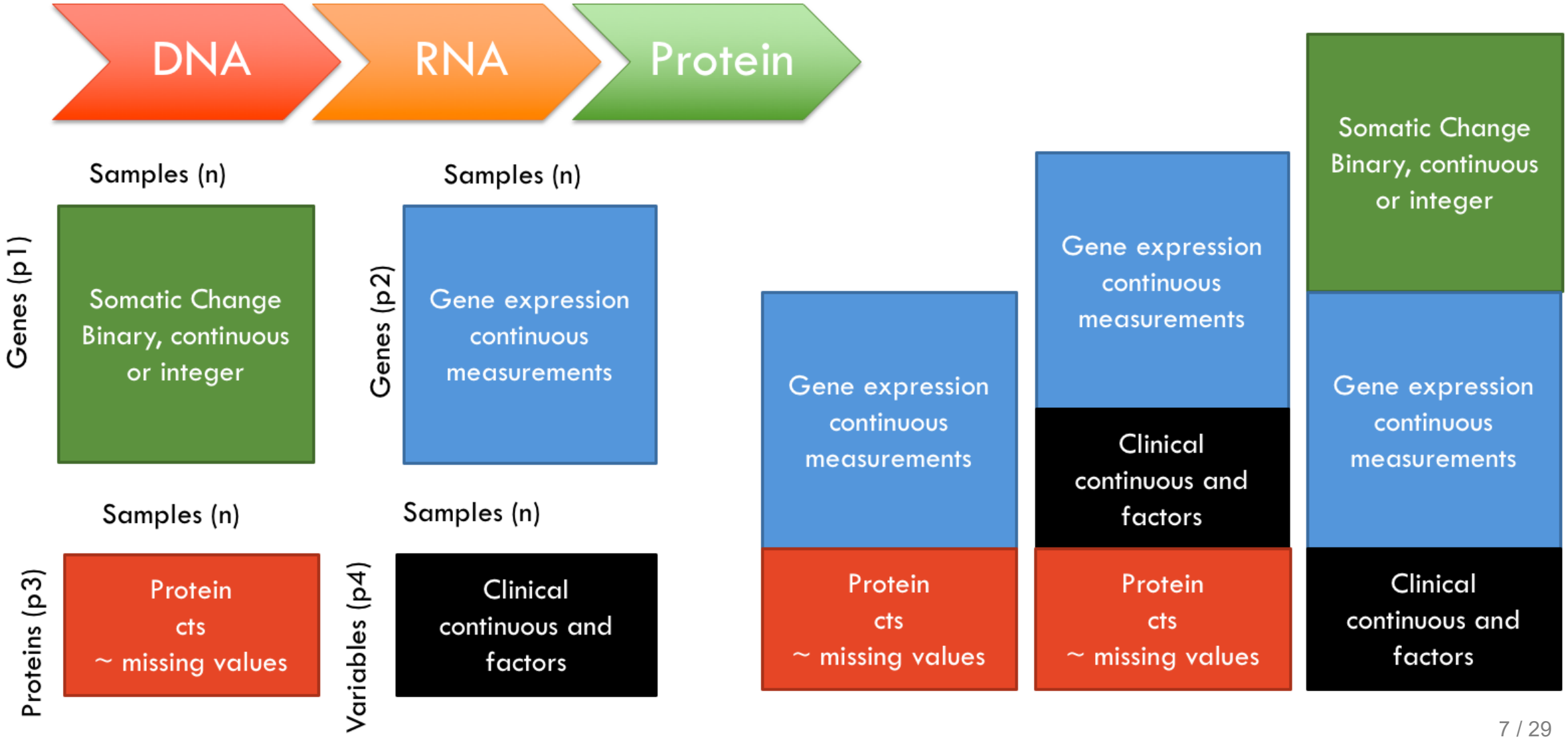
Protein
cts
~ missing values

Samples (n)

Variables (p4)

Clinical
continuous and
factors

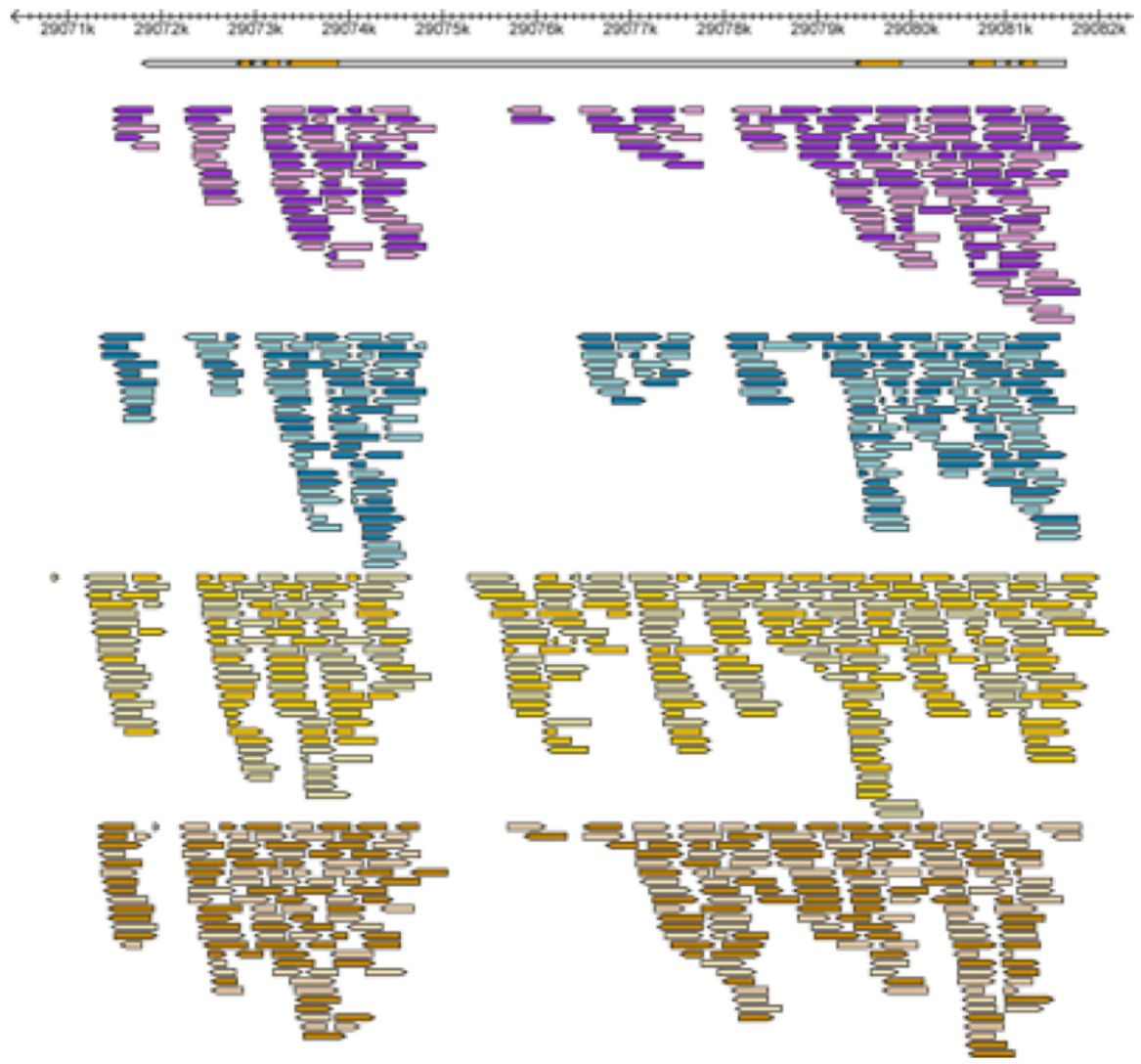
Possible input ?

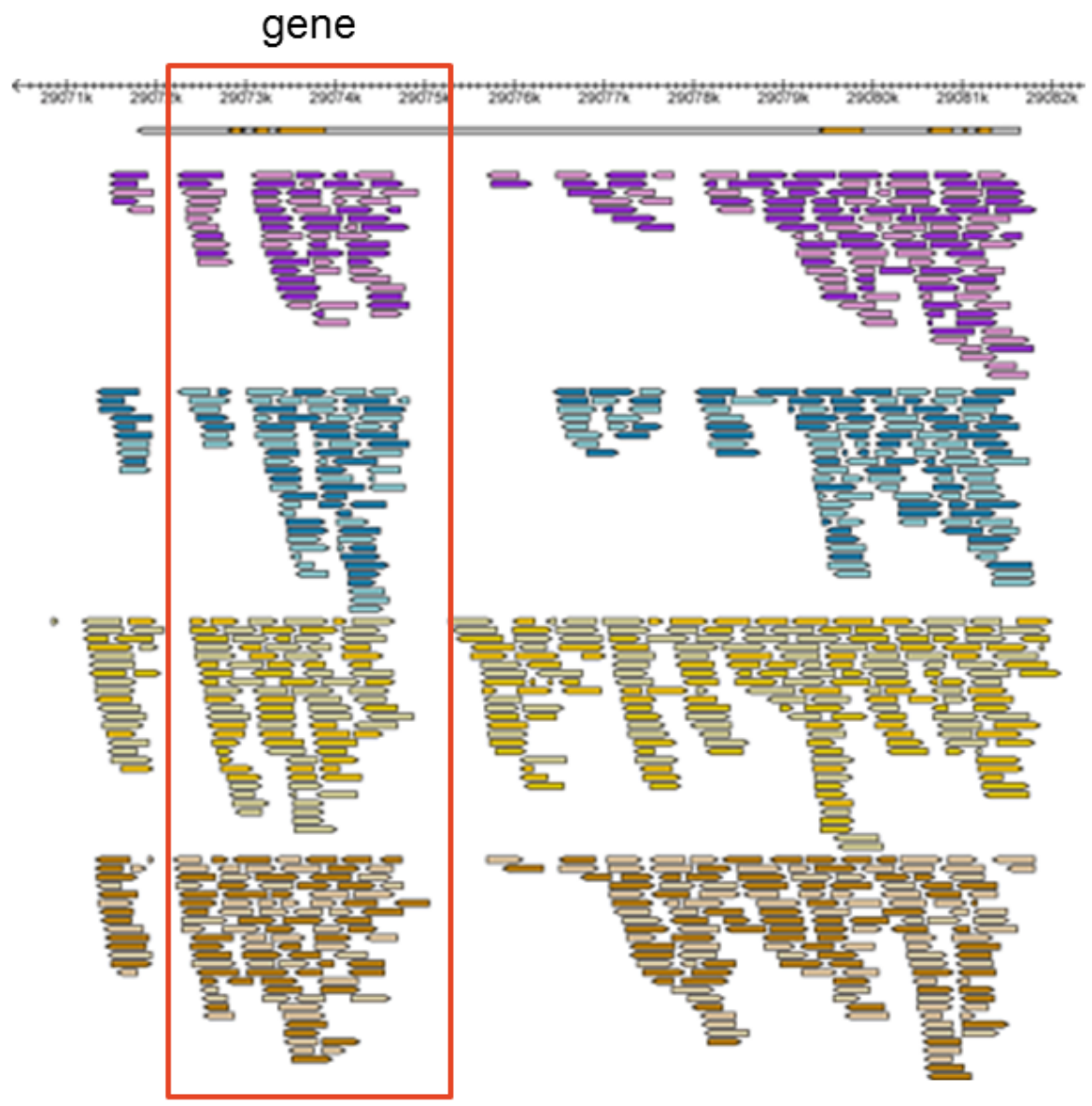


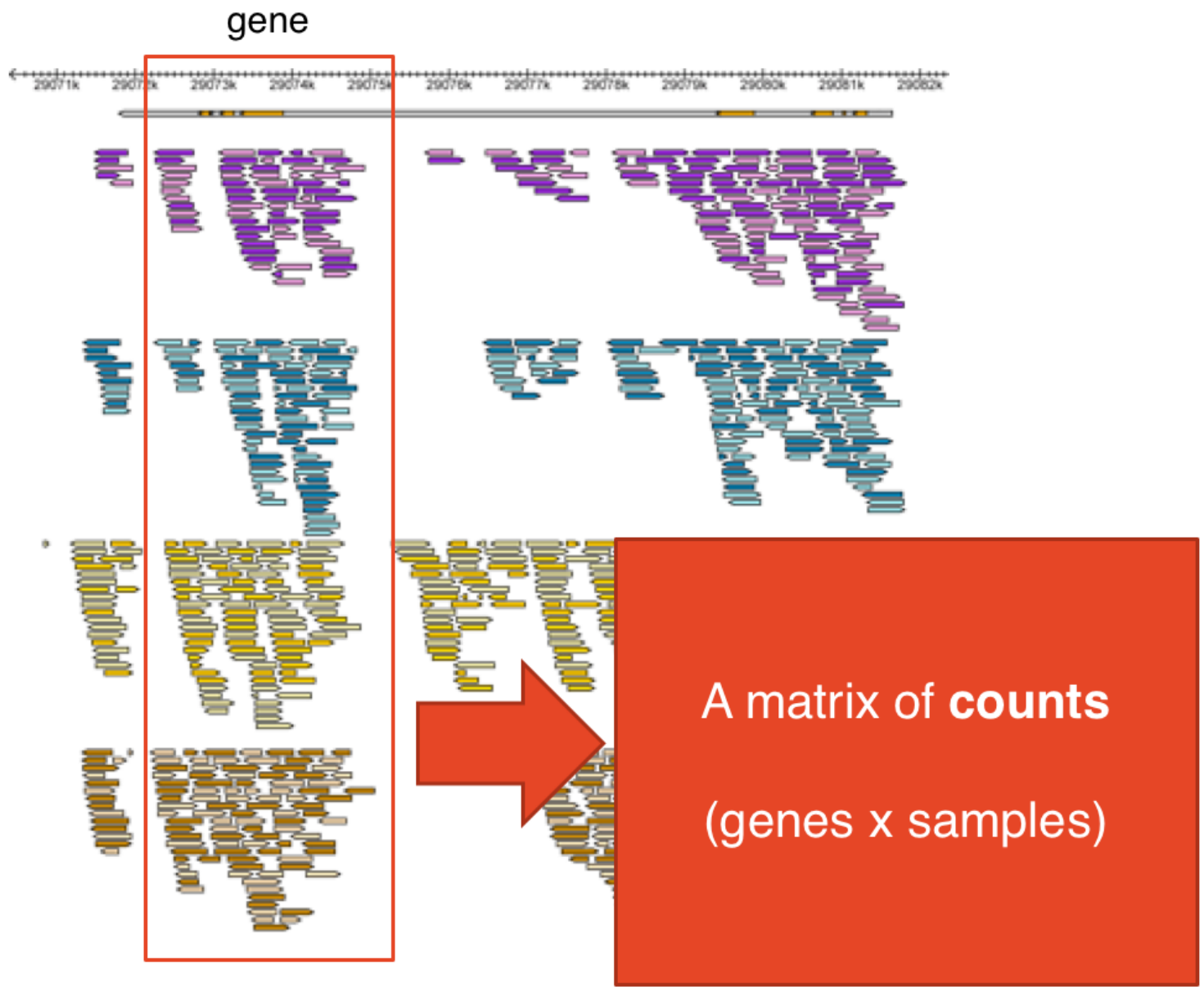
Steps in preparing an RNA-seq library

1. Purify RNA
2. Bind polyA fraction (mRNA)
3. Fragment RNA (200 bp)
4. Convert to cDNA by random priming
5. Apply adaptors and sequence
6. Analyze millions of 25 bp reads

geospiza
FROM SAMPLE TO RESULTS™


29071k
29072k
29073k
29074k
29075k
29076k
29077k
29078k
29079k
29080k
29081k
29082k


gene

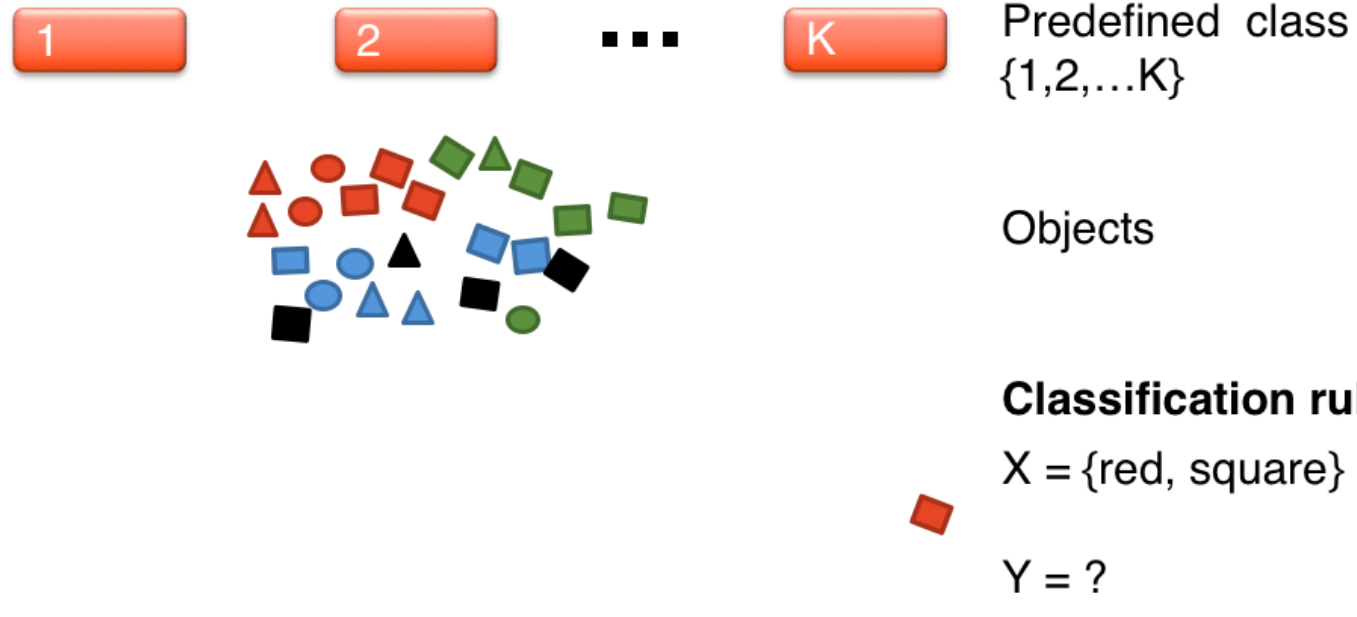

gene
A matrix of **counts**
(genes x samples)

Basic principles of discrimination

Each object associated with

- a class label (or response) $Y \in \{1, 2, \ldots, K\}$ and
- a feature vector of P measurements: $\mathbf{X} = (X_1, \ldots, X_P)$

Aim: predict Y from **X**.



Y = Class Label = 2

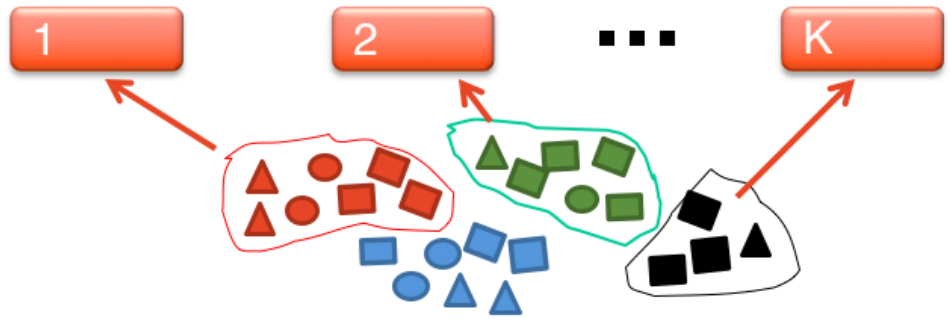
X = Feature vector
{colour, shape}

Basic principles of discrimination

Each object associated with

- a class label (or response) $Y \in \{1, 2, \ldots, K\}$ and
- a feature vector of P measurements: $\mathbf{X} = (X_1, \ldots, X_P)$

Aim: predict Y from **X**.



Predefined class {1,2,…K}

Objects

Y = Class Label = 2

X = Feature vector
{colour, shape}

Classification rule ?

X = {red, square}

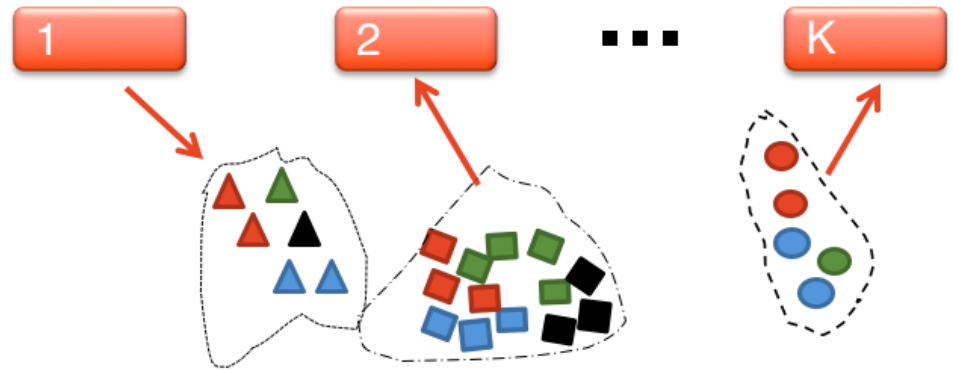
Y = ?

Basic principles of discrimination

Each object associated with

- a class label (or response) $Y \in \{1, 2, \ldots, K\}$ and
- a feature vector of P measurements: $\mathbf{X} = (X_1, \ldots, X_P)$

Aim: predict Y from **X**.



Predefined class {1,2,…K}

Objects

Y = Class Label = 2

X = Feature vector
{colour, shape}

Classification rule ?

X = {red, square}

Y = ?

Feature selection

Why?

- Lead to better classification performance by removing variables that are noisy with respect to the outcome.
- May provide useful insights into etiology of a disease.
- Can eventually lead to the diagnostic tests (e.g. "lympho chip")

Feature selection

Why?

- Lead to better classification performance by removing variables that are noisy with respect to the outcome.
- May provide useful insights into etiology of a disease.
- Can eventually lead to the diagnostic tests (e.g. "lympho chip")

Approaches?

Methods fall into three basic categories:

- Filter methods
- Wrapper methods
- Embedded methods

The simplest and most frequently used methods are the filter methods.

Feature selection: filter method

- Typical RNA-Seq experiments measure tens of thousands of genes
- In reality, only a subset of these are related to the outcome
- Typically aim to select features that appear differentially expressed in the training dataset

Note: The feature selection should be done within cross-validation for correctness.

Case study: Acute Myeloid Leukaemia (AML) Treatment Resistance

Case study: Acute Myeloid Leukaemia (AML) Treatment Resistance

- Primary therapy resistance is a major problem in acute myeloid leukemia treatment. Approximately 20-30% of younger adult patients with AML and as many as 50% of older adults are refractory to induction treatment.
- Research findings are published in *Haematologica* in 2018.
- The data is available from GEO Browser as 250 `txt.gz` files of gene-level read counts or a Microsoft Excel file where the gene expression values were standardised to have a mean of 0 and variance of 1.

Case study: Acute Myeloid Leukaemia (AML) Treatment Resistance

- Primary therapy resistance is a major problem in acute myeloid leukemia treatment. Approximately 20-30% of younger adult patients with AML and as many as 50% of older adults are refractory to induction treatment.
- Research findings are published in *Haematologica* in 2018.
- The data is available from GEO Browser as 250 `txt.gz` files of gene-level read counts or a Microsoft Excel file where the gene expression values were standardised to have a mean of 0 and variance of 1.
- Data input and processing will be covered in more detail in hands-on part

Clinical Data

The clinical data provides information about seven different characteristics of the patients.

- The phenotype of interest here is **Response**

```
head(sampleInfo)
```

```
##       ID Gender Age  Response Survival Time Status RUNX1-RUNX1T1 Fusion RUNX1 Mutation
## 1 GEO-13   Male  70 Resistant           126   Dead                   No             Yes
## 2 GEO-14   Male  61 Resistant           226   Dead                   No             Yes
## 3 GEO-15 Female  61 Resistant           103   Dead                   No             Yes
## 4 GEO-16   Male  60 Resistant           118   Dead                   No             Yes
## 5 GEO-17   Male  56 Resistant           296   Dead                   No              No
## 6 GEO-18   Male  25 Resistant           230  Alive                   No             Yes
```

```
nrow(sampleInfo)
```

```
## [1] 250
```

Clinical Data

Observe the number of samples which are resistant to treatment and which are not.

```
table(sampleInfo[, "Response"], useNA="always")
```

```
##
## Resistant Sensitive      <NA>
##        71       164        15
```

Clinical Data

Observe the number of samples which are resistant to treatment and which are not.

```
table(sampleInfo[, "Response"], useNA="always")
```

```
##
## Resistant Sensitive      <NA>
##        71       164        15
```

71 patients are Resistant, 164 Sensitive and 15 patients have no resistance information.

Gene expression data: Read counts

- This will be explored further in the hands-on part.

```
dim(readCounts)
```

```
## [1] 23367   250
```

There are 23367 genes in the counts table and 250 AML patients.

Gene expression data: Read counts

- This will be explored further in the hands-on part.

```
dim(readCounts)
```

```
## [1] 23367   250
```

There are 23367 genes in the counts table and 250 AML patients.

```
readCounts[1:6, 1:6]
```

```
##          GEO-13 GEO-14 GEO-15 GEO-16 GEO-17 GEO-18
## A1BG        270     84     22    245    380    581
## A1BG-AS1     32     59      7     60    126     88
## A1CF          0      0      0      0      0      0
## A2LD1        14      4     18     19     35      1
## A2M          38      8     58    185      0     20
## A2ML1         0      0      1      0      0      0
```

Discovery and Validation Data

- For this workshop, we will be splitting our data into equal discovery and validation sets.
- We will use the validation set to assess our simple 2-gene classifiers - you will look at other feature selection methods in the hands-on part

```
dim(readsCPM_train)
```

```
## [1] 23367   118
```

```
table(sampleInfoComplete[trainingIndex,"Response"])
```

```
##
## Resistant Sensitive
##        44        74
```

```
dim(readsCPM_test)
```

```
## [1] 23367   117
```

DLDA: A Linear Boundary Classifier

Diagonal Linear Discriminant Analysis (DLDA) is special version of LDA which assumes no covariance between the features which enables it to be used when the number of genes exceeds the number of samples - an almost universal occurrence in omics. If there were only two genes used, the separation line would be a straight line. In higher dimensions, it is a plane.

DLDA: A Linear Boundary Classifier

DLDA from sparsediscrim for this case study has no tuning parameters to specify. Like many classical statistical classifiers, it requires the genes to be the columns of the matrix and the samples to be the rows of the matrix.

```
library(sparsediscrim)
twoGenes_train = readsCPM_train[c("TMUB2","PHF8"),]
trained_dlda = dlda(t(twoGenes_train), sampleInfoComplete[trainingIndex,"Response"])

predicted_dlda = predict(trained_dlda, t(twoGenes_test))[["class"]]
table(sampleInfoComplete[testingIndex,"Response"], predicted_dlda)
```

```
##             predicted_dlda
##              Resistant Sensitive
##   Resistant         21         6
##   Sensitive         40        50
```

- This is a confusion matrix showing the **known** response as well as what we predicted
- In the hands-on session you will use these types of output to evaluate the performance of classifiers

Random Forest: A Non-Linear Boundary Classifier

A random forest is a set of decision trees which each guess the class of each sample. The class which has the most common guesses for a particular sample is predicted as that sample's class. The decision boundary is often non-linear. For example, IF value < 5 AND IF value > 11 THEN Poor.

Random forests in R are provided by the package *randomForest*. Random forests have a large number of options. Here, the default values are used. Like DLDA, it requires the **genes** to be the **columns** of the matrix and the **samples** to be the **rows** of the matrix.

Random Forest: A Non-Linear Boundary Classifier

Unlike DLDA, randomForest takes the training data and test data all as inputs to one function. Also, the predicted classes are stored in the list element named "predicted". Each programmer has his/her own convention and classification code written for one classifer won't work for another.

```
library(randomForest)
```

```
trained_rf = randomForest(t(twoGenes_train),
                          sampleInfoComplete[trainingIndex,"Response"])

predicted_rf = predict(trained_rf, t(twoGenes_test))
table(sampleInfoComplete[testingIndex,"Response"], predicted_rf)
```

```
##             predicted_rf
##              Resistant Sensitive
##   Resistant         13        14
##   Sensitive         25        65
```

Support Vector Machine: Non-Linear Boundary

An SVM classifier may either use a linear or non-linear boundary, depending on the user's choice of *kernel*. Understanding how it works requires an undersanding of convex optimisation, so that is not covered in this course.

The default kernel used is the radial basis function kernel which calculates a non-linear boundary. Like many classical statistical classifiers, it requires the **genes** to be the **columns** of the matrix and the **samples** to be the **rows** of the matrix.

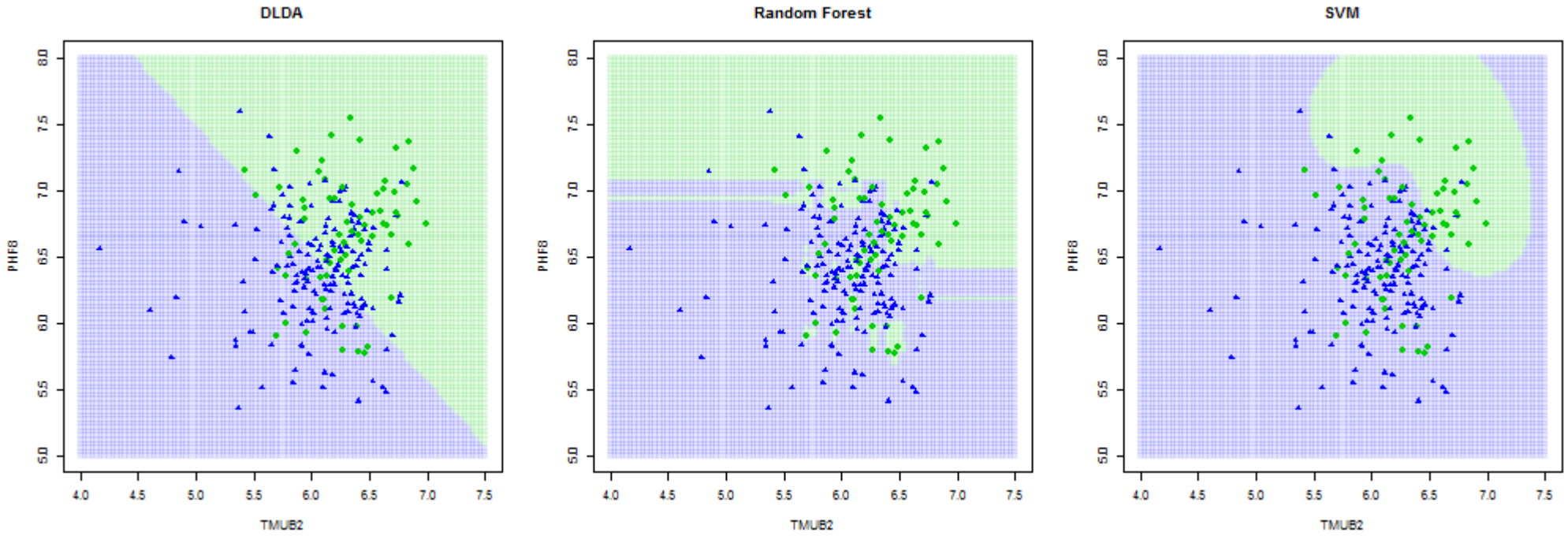
```
library(e1071)
trained_svm = svm(t(twoGenes_train), sampleInfoComplete[trainingIndex,"Response"])

predicted_svm = predict(trained_svm, t(twoGenes_test))
table(sampleInfoComplete[testingIndex,"Response"], predicted_svm)
```

```
##             predicted_svm
##              Resistant Sensitive
##   Resistant          9        18
##   Sensitive          3        87
```

Summary of classifiers

- We can look at the fitted models to better understand how these classifiers behave.
- This is only for two genes, things become more difficult to visualise in higher dimensions.
- Different classifiers have different strengths and may be more suitable in different circumstances



Later today: Performance assessment

- Evaluation of overall error, sample-specific error, precision, recall.
- Cross-validation to evaluate classifier performance
- Comparison of the DLDA, Random forest and SVM classifiers.

Now: Hands-on session

- Activity Overview
  - Loading RNA-seq data set: Acute Myeloid Leukaemia treatment resistance
  - Cleaning and transformation
  - Fitting widely used classifiers to a simple partition of the data set