Machine Learning in R

Lecture 9 - Clustering and classification of high-dimensional data in a genomic context

Adam Olshen – Epidemiology and Biostatistics and Helen Diller Family Comprehensive Cancer Center

adam.olshen@ucsf.edu

Goal of lecture

1. To give an understanding of how genomic data is analyzed in order
   - to evaluate a genomic analysis in a manuscript
   - to get started on one's own analysis
2. To give real data examples of some techniques already seen in this class such as unsupervised learning and cross-validation
3. To teach some new, simple analysis techniques

Genomic analyses are special because:

- *p* (variables) >> *n* (samples), with *p* in the tens of thousands or more and *n* in the tens
- Design is often suboptimal for financial reasons, lack of available material, or other reasons
- There are often systematic biases in the data due to “batch” effects
- Scientific validation is possible through lab work, model systems or other means

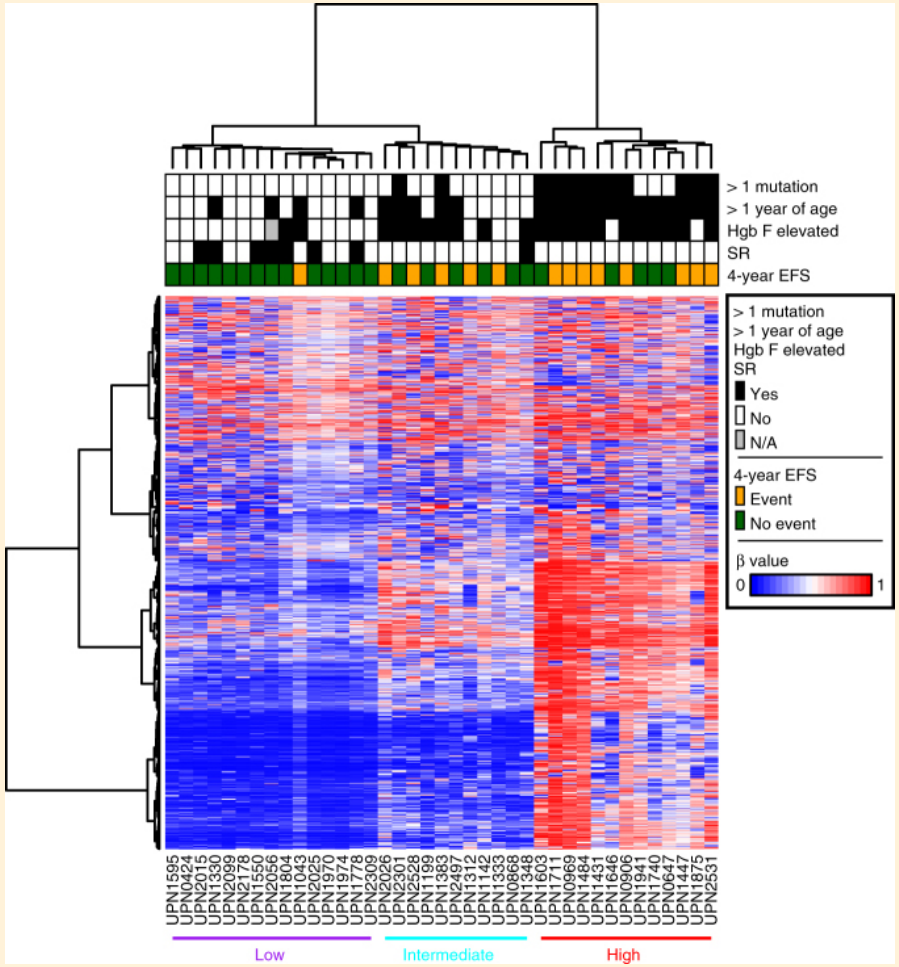
Genomic analyses are special because:

- Interpretability of analysis is usually required
- Pure black box solutions are discouraged
- An attempt is often made to discover the biology behind the results

Outline

- Study design, batch adjustment and normalization
- Visualization/unsupervised learning
- Issues related to classification and a couple simple classification methods
- Biological interpretation and gene theme analysis

Genomics data is often displayed via heatmaps



Breast cancer blood biomarkers: A true story

- Goal: Develop a diagnostic for breast cancer from blood samples
- Blood samples were collected from hundreds of breast cancer cases and hundreds of controls
- Samples were analyzed in batches of 24
- Early results showed consistent gene expression differences between cases and controls!

An optimally poor design

- The analyst (me) asked to see which samples were run in which batches
- Early batches were all controls and later batches were all cases!
- This is problematic for two reasons:
  - Cannot separate batch and case/control effects
  - Cannot determine if shift over time
- I suggested running future batches mixing cases and controls and differences disappeared

An optimally poor design (the lesson)

- One should not apply machine learning without understanding the data generation mechanism
- Common sense reasoning can be used to evaluate the quality of an experiment
  - Have the correct samples been collected?
  - Are there enough samples to have power to detect differences?
- Search for batch effects in data and correct for them if possible

Normalization

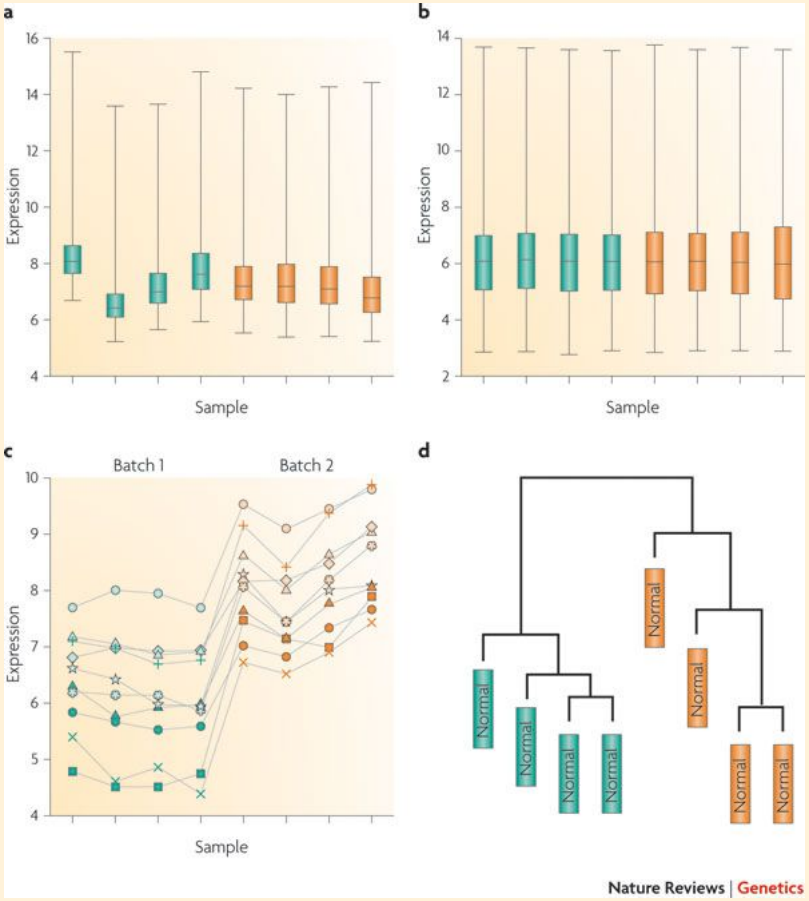
- Normalization is the process of making comparable data from different samples
- Sample preparation or measuring devices may lead to shifts in data, outliers or other changes
- Normalization is very specific to technology, but a couple types are:
  - Light – adjust data to make average values the same
  - Severe – adjust data to make whole distributions the same

Batch effects

- Batch effects are systematic biases in data that can result from collection or processing of samples occurring in groups
- They can be corrected using regression methods like ComBat (Johnson et al.) and SVA (Leek)
- They are often identified using principal components

Normalization does not prevent batch effects

- 8 samples are shown in Figure (a)
- They are normalized in Figure (b)
- Figure (c) shows 10 genes vulnerable to batch effects
- The samples cluster by batch in Figure (d)



Taken from Leek et al. Nature Reviews Genetics volume 11, pages 733–739 (2010)

Example data

nature COMMUNICATIONS

ARTICLE

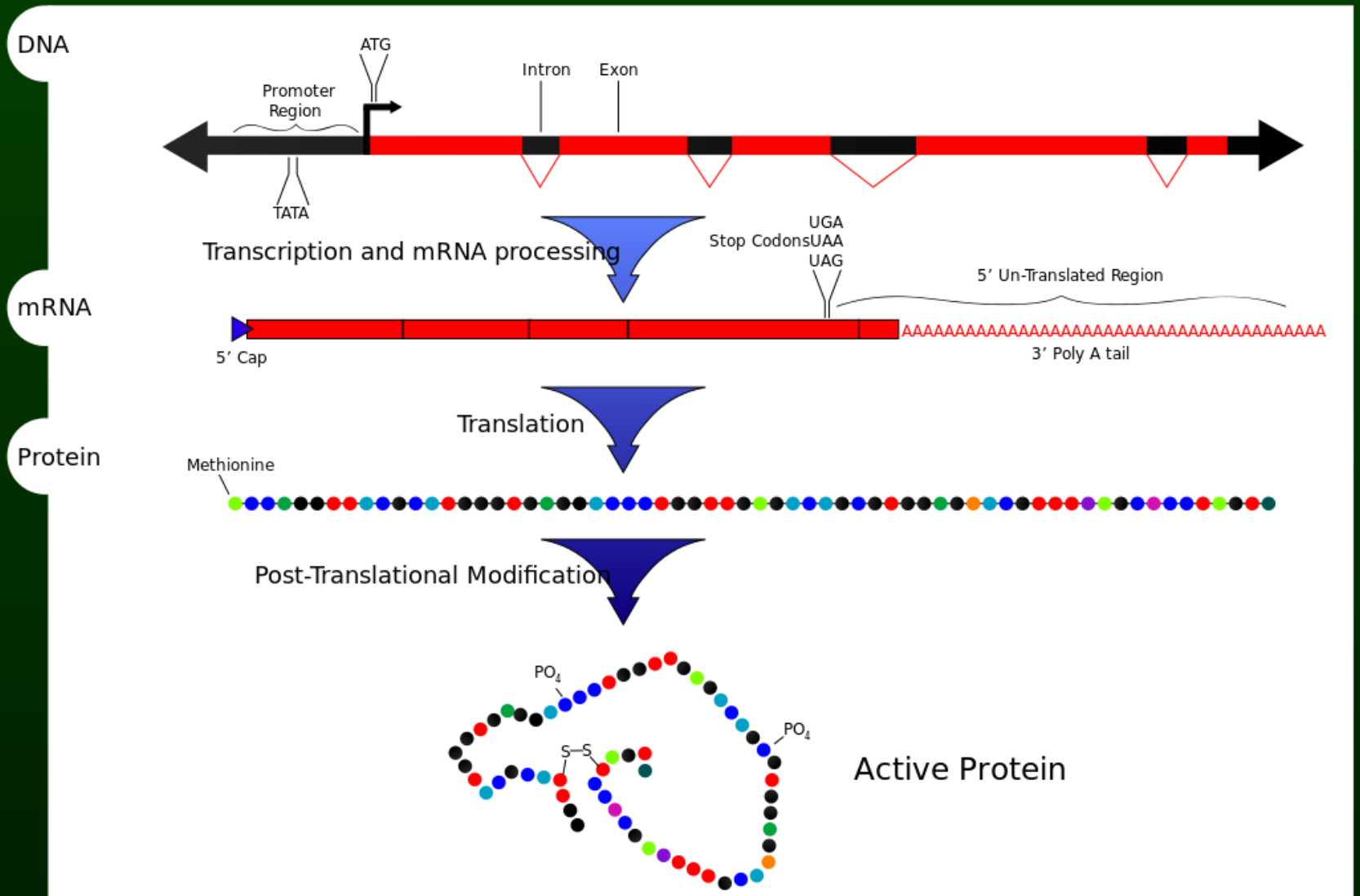
DOI: 10.1038/s41467-017-02178-9 OPEN

Genome-wide DNA methylation is predictive of outcome in juvenile myelomonocytic leukemia

Elliot Stieglitz[1,2], Tali Mazor[3], Adam B. Olshen[2,4], Huimin Geng[5], Laura C. Gelston[1], Jon Akutagawa[1], Daniel B. Lipka[6,7,8], Christoph Plass[6,9], Christian Flotho[9,10], Farid F. Chehab[11], Benjamin S. Braun[1,2], Joseph F. Costello[3] & Mignon L. Loh[1,2]

Central dogma of molecular biology

Figure taken from wikipedia: Original work by Mike Jones



Methylation blocks gene expression

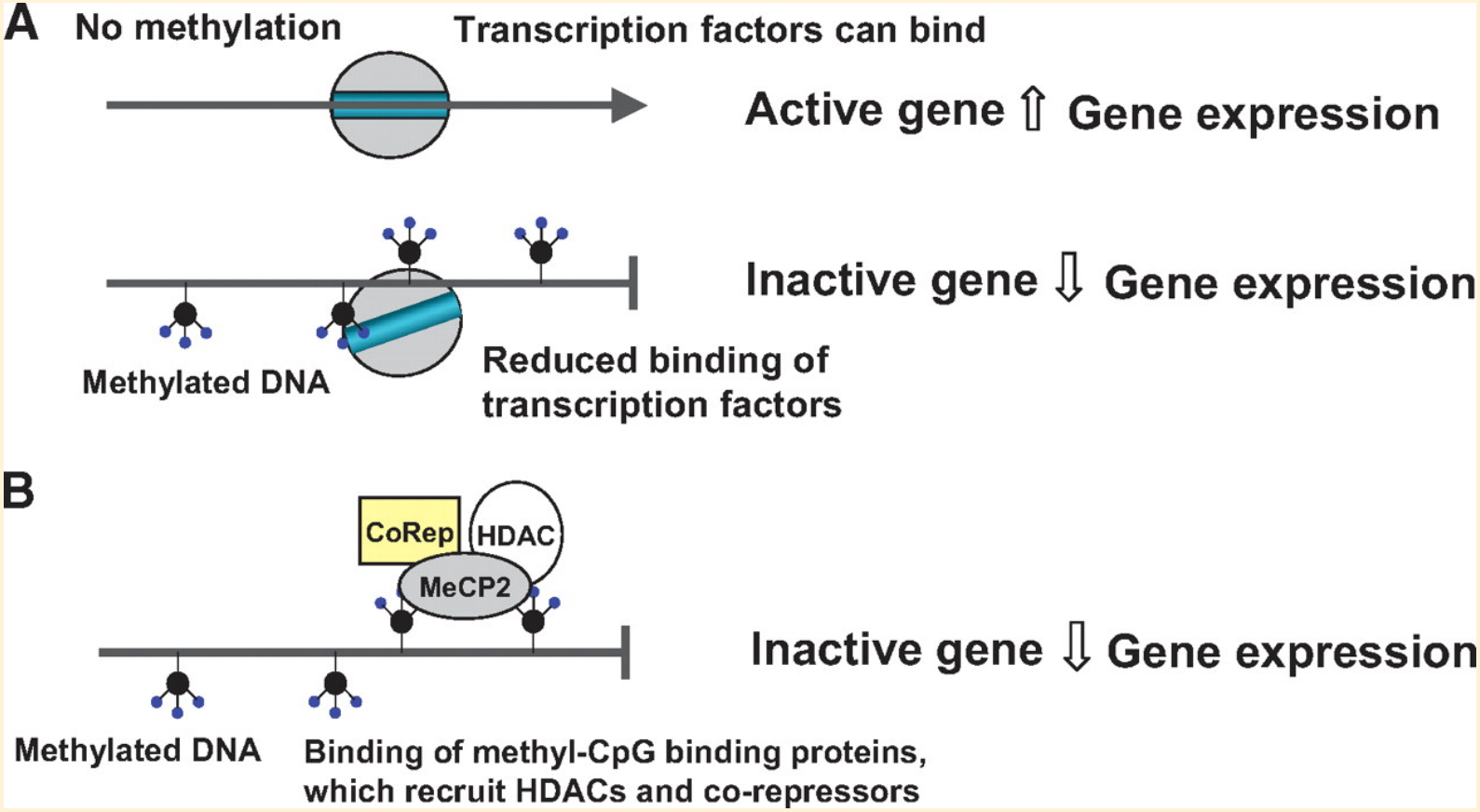


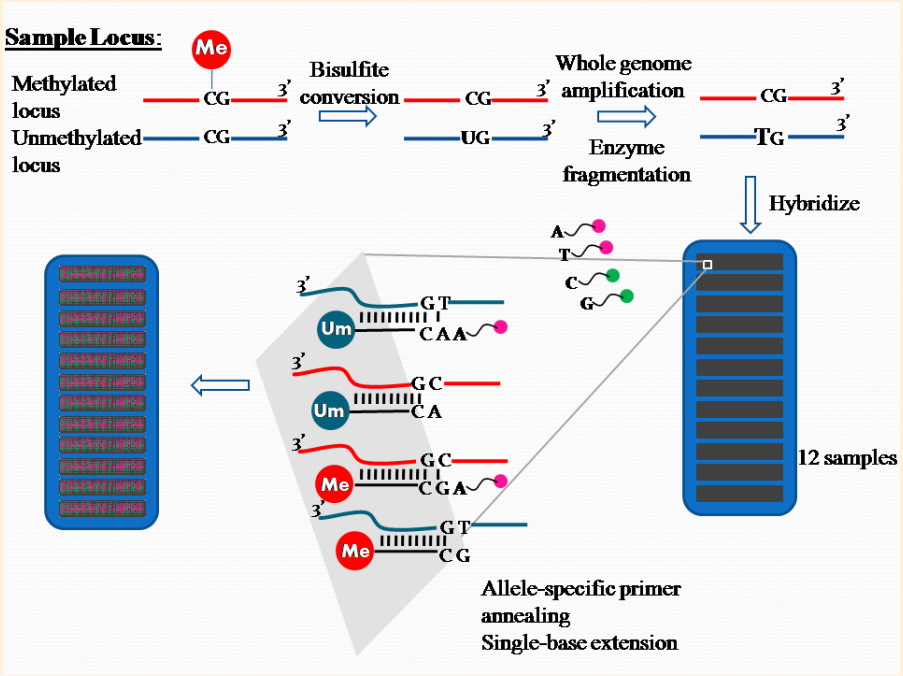
Figure taken from wikipedia: Original work by Mike Jones

Example data

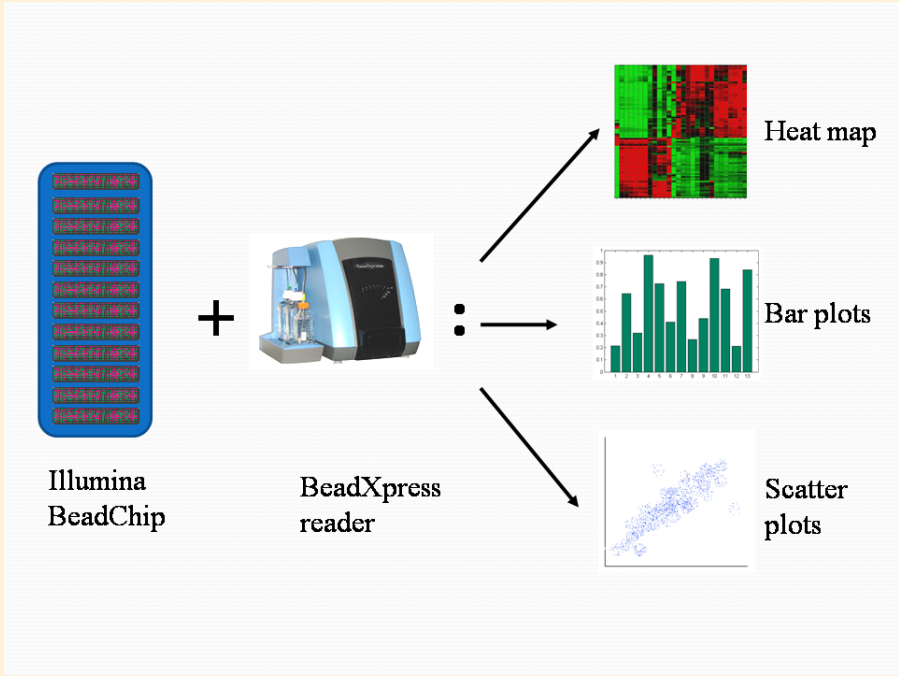
- Juvenile myelomonocytic leukemia (JMML) is a rare and serious form of childhood leukemia
- Outcomes in JMML vary from spontaneous resolution to rapid relapse after hematopoietic stem cell transplantation
- The authors believe that DNA methylation patterns would help predict disease outcome
- The data are genome-wide DNA methylation
  - 39 patient training set (from UCSF)
  - 40 patient test set (from Germany)

Methylation measured using microarrays

JMML samples taken at diagnosis



Step 1



Step 2

Markers, genes and variables

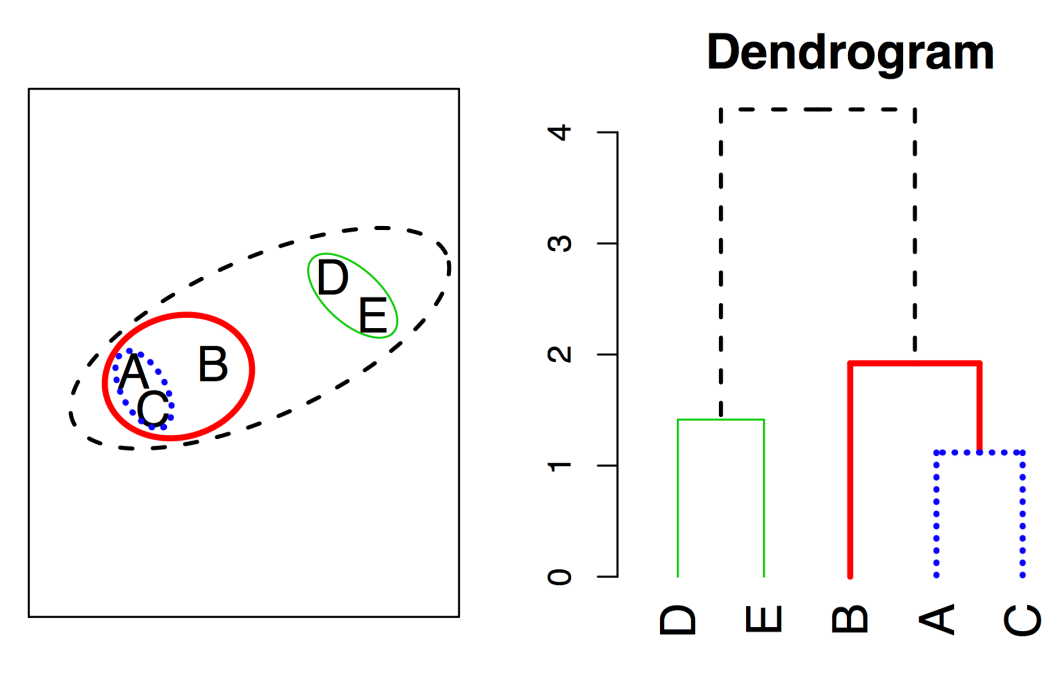
- I will refer to the methylation measurements as *markers*
- Later in the talks I will refer to measurements as *genes*
- I may also use the word *variables* to refer to either

Visualization

- After normalization and adjusting for batch effects my next step is to look at the data
- A common method for data visualization is agglomerative hierarchical clustering

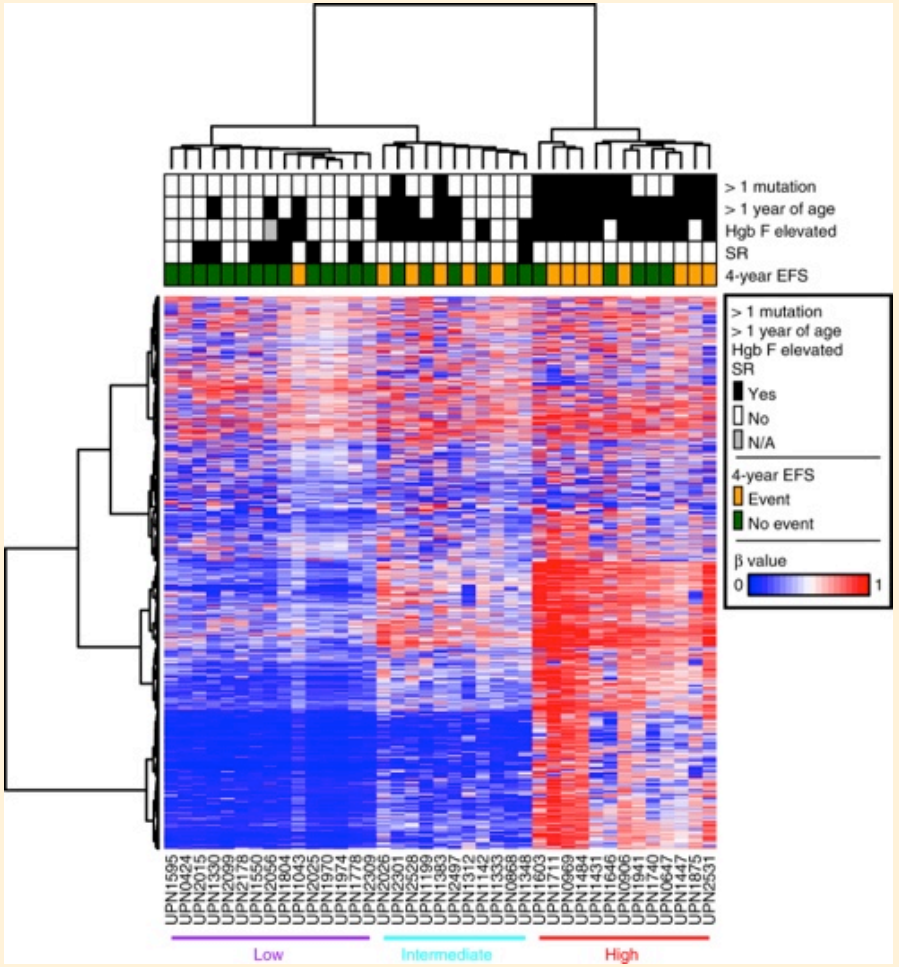
Hierarchical Clustering Algorithm

- Start with each point/object in its own cluster
- Identify the closest two clusters and merge them
- Repeat
- Ends when all points are in a single cluster
- Decide how many clusters there are



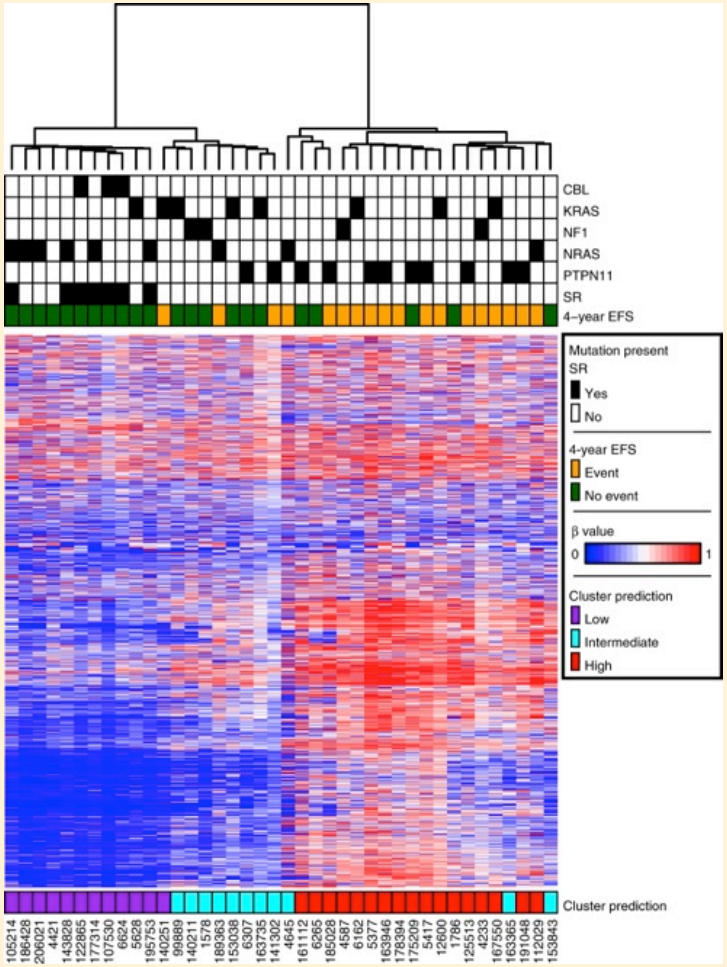
Dendogram of methylation training in both directions with heatmap has massive amounts of information

- 39 samples, 1500 markers
- Lower methylation more blue, higher methylation more red
- Samples naturally split into low, intermediate and high methylation groups
- Sample groups appear related to survival and other factors
- Markers split into four or so groups



Clustering of methylation test

- 40 samples, same 1500 markers
- Samples again cluster into low, intermediate and high clusters
- Prediction based on nearest training set centroid
- Relationship to survival repeated
- Such luck is not often the case!



PCA details

- The *first principal component* of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

that has the largest variance. By *normalized*, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

- We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the loadings of the first principal component; together, the loadings make up the principal component loading vector, $\phi_1 = (\phi_{11}\ \phi_{21}\ \ldots\ \phi_{p1})^T$.
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

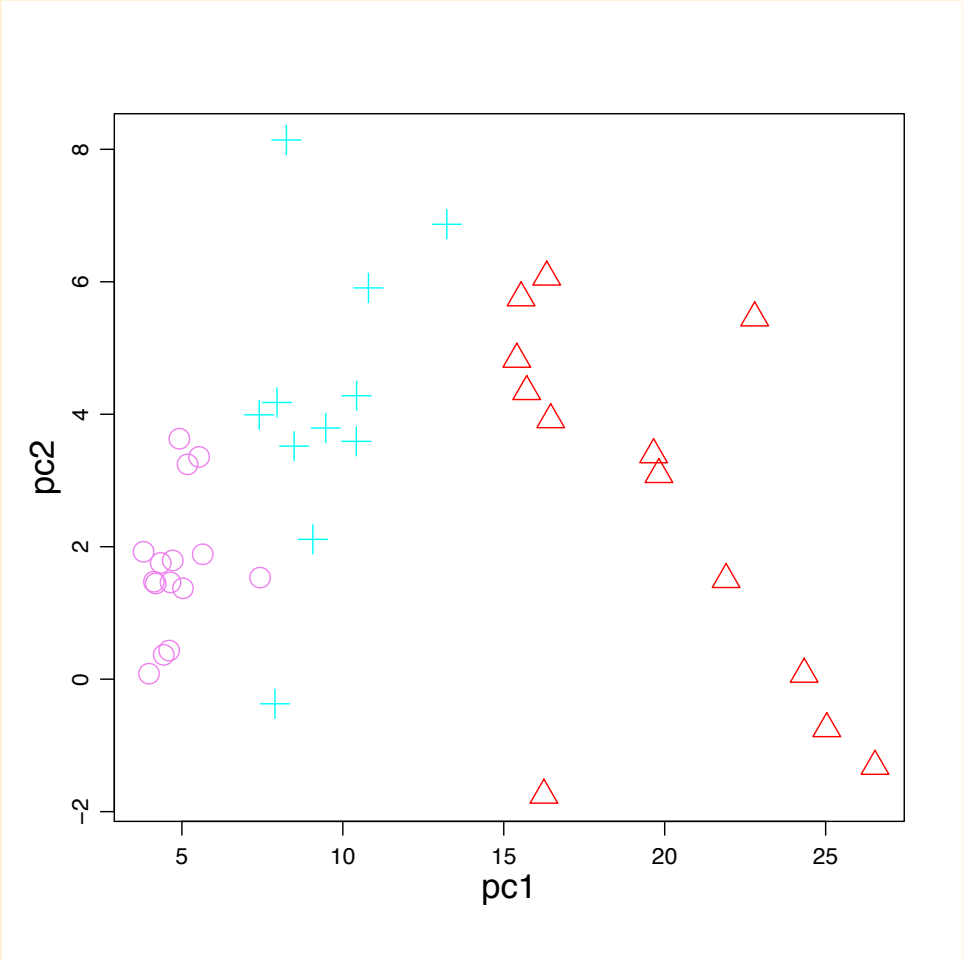
Additional PC's

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are *uncorrelated* with Z_1.
- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$z_{i2} = \phi_{12} x_{i1} + \phi_{22} x_{i2} + \ldots + \phi_{p2} x_{ip},$$

where ϕ_2 is the second principal component loading vector, with elements $\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$.

PCA of methylation training data leads to similar groupings to AHC



○ for low methylation

\+ for intermediate methylation

△ for high methylation

Clustering vs. PCA

- Clustering optimizes for homogeneous subgroups among the observations
- PCA optimizes for a low-dimensional representation of the observations that explains a good fraction of the variance
- PCA solutions can appear like clusters
- PCA useful for finding batch effects

Predictive Modeling/Classification

- Suppose one wants to predict some categorical feature
  - Long or short survivors
  - Responds to therapy or not
  - Cluster membership
- Many methods exits for such predictions
  - Tree-based such as CART/Random Forests
  - Support vector machines
  - “Deep learning”

Predictive Modeling/Classification

I will focus on a few simple methods because:

- Small sample sizes lead to inability to distinguish
- Noisy data makes different methods give similar answers
- Interpretability is often a key consideration

p (markers) >>n (samples), why n small?

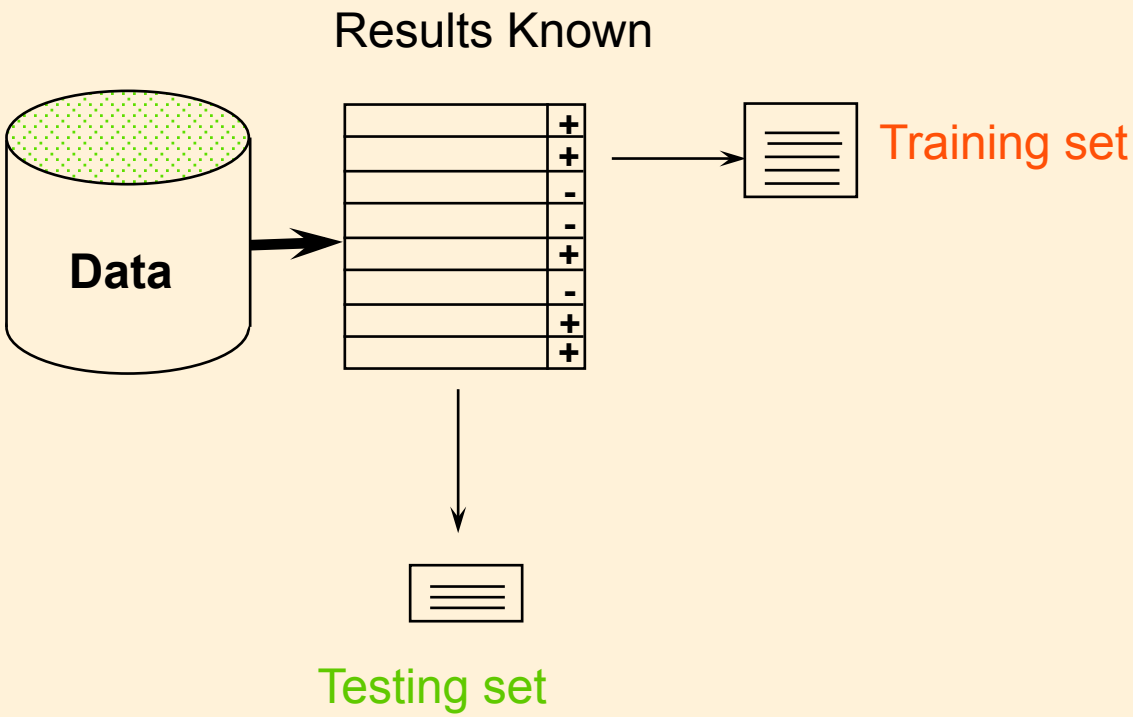
There are many reasons why n is usually small:

- The cost of the assay is hundreds or thousands of dollars each
- For a respective study there are few tissues in a tissue bank
- For a perspective study few samples can be accrued per month
- There is a mistaken belief among investigators that because p is large n can be small

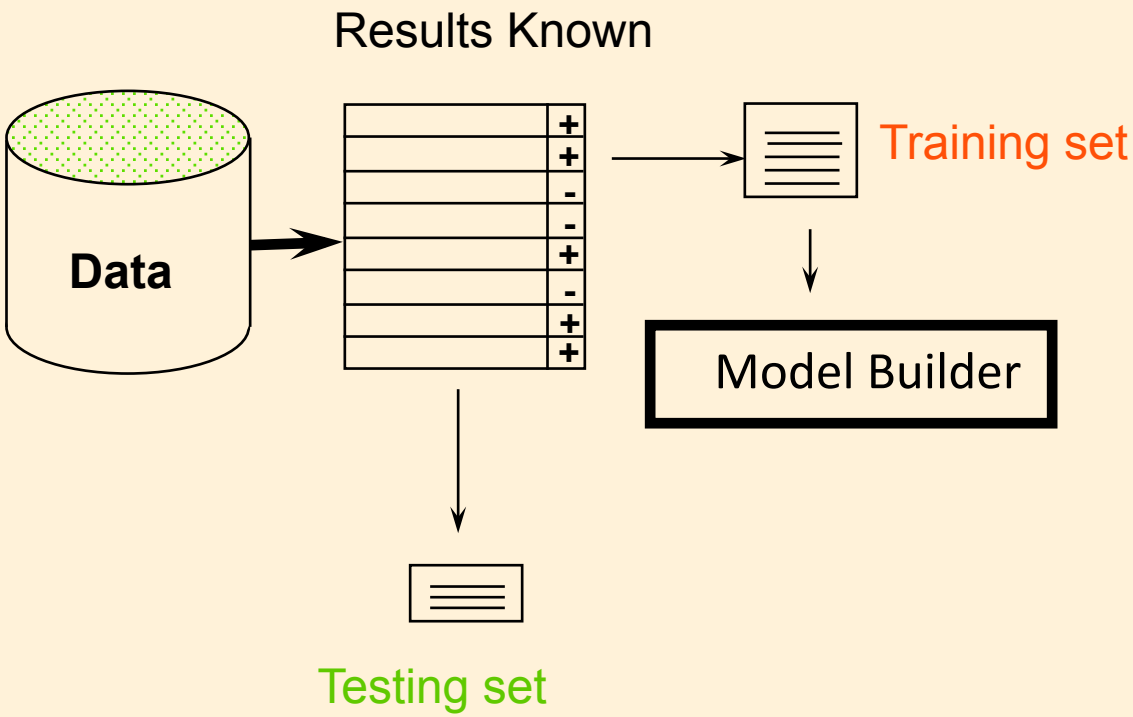
Model building for n small

- Ideally models are built on a training set and evaluated on a test/ validation set
- In genomics models are often evaluated using cross-validation
- In the JMML methylation example the data being taken from two sites led to a natural grouping of training and test despite small size

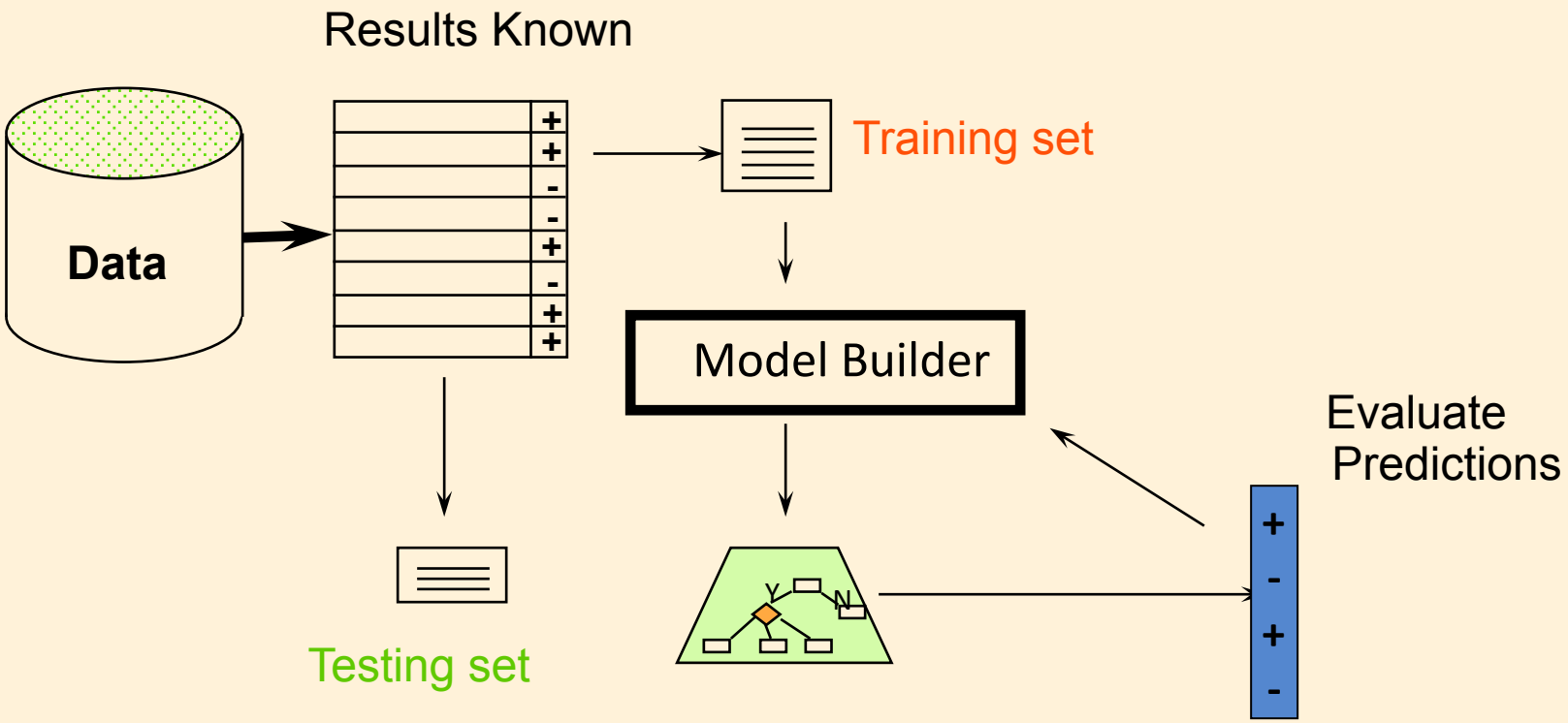
Training/test



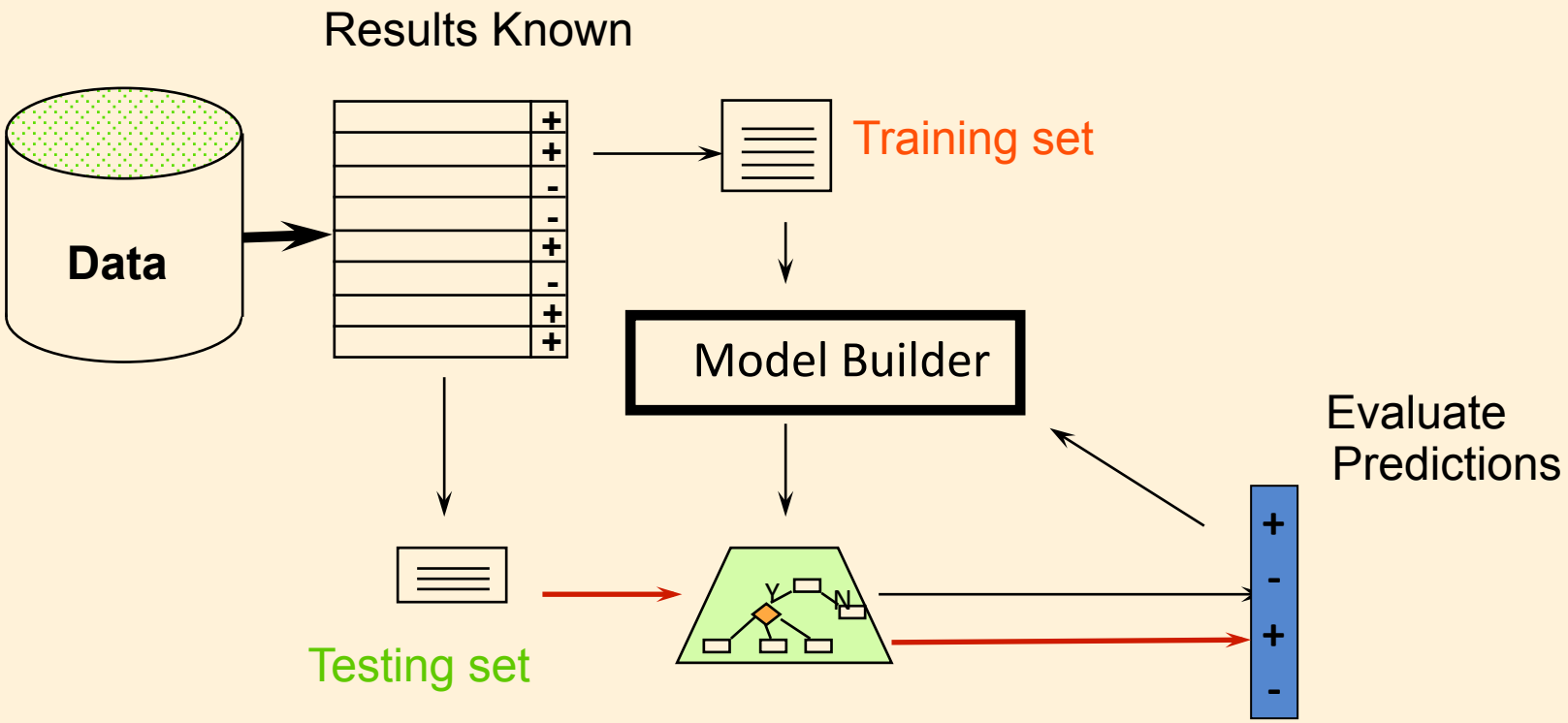
Training/test



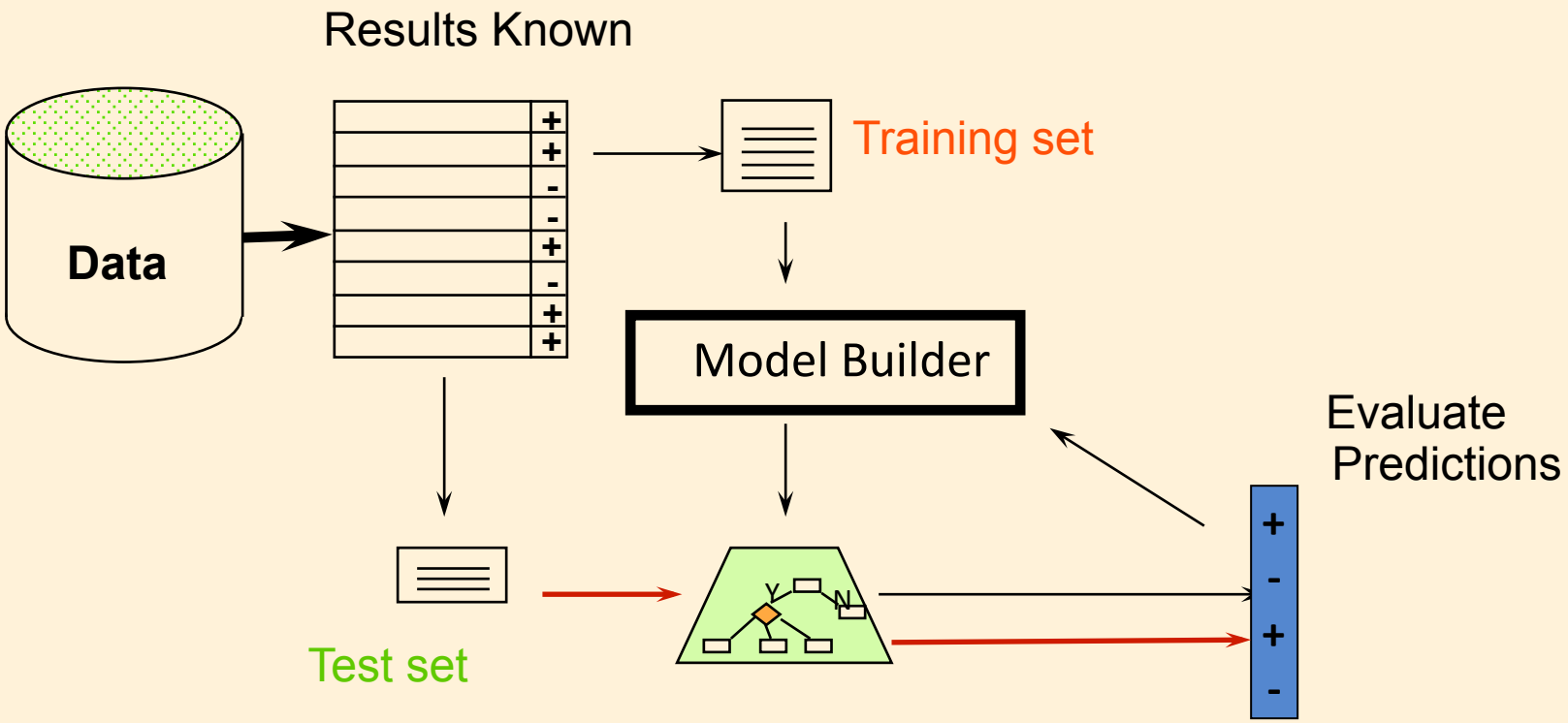
Training/test



Training/test



Training/test



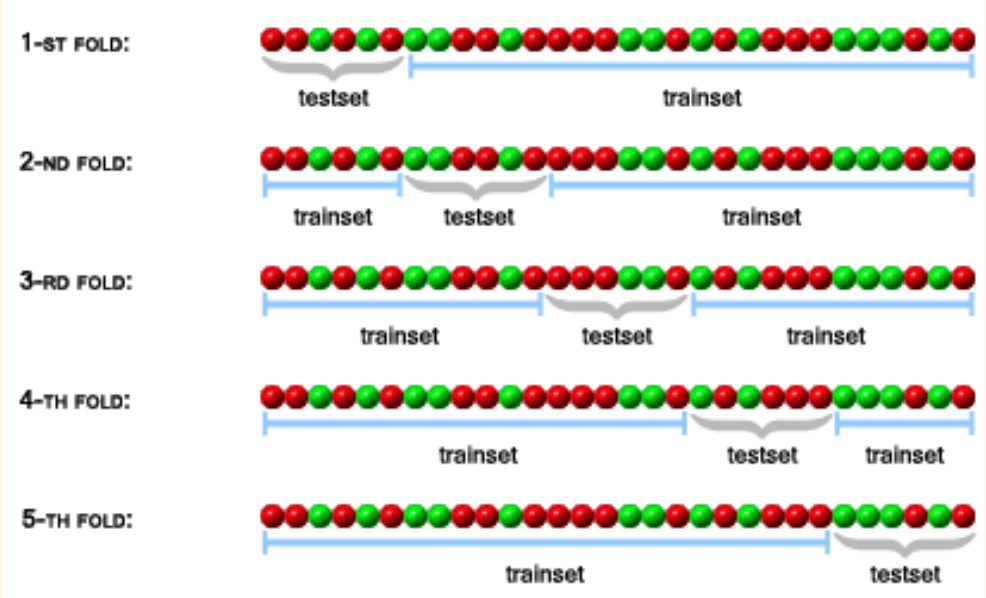
This is great if you have a lot of data to be able to reserve a testing set, but what if you don't?

Cross-validation for training/test

- *Cross-validation*
  - First step: data is split into *k* subsets of equal size
  - Second step: each subset is used in turn for validation and the remainder for training
- This is called *k-fold cross-validation* (10-fold is common choice, also 5-fold for smaller datasets approx. <200 and even *N-fold* for very small datasets, called leave-one-out or loocv)
- Often the subsets are stratified before the cross-validation is performed − e.g. same proportion of each diagnostic class in each fold
- The error estimates are averaged across folds to yield an overall error estimate

E.g. 5-fold cross-validation

- Break up data into 5 groups of the same size
- Hold aside one group for testing and use the rest to build model
- Repeat



- When every group has had a turn at being the test set then you can estimate the accuracy (MSE or classification) based on the combined Test data (i.e. the 5 test-sets)

p>>>n, what happens with p large?

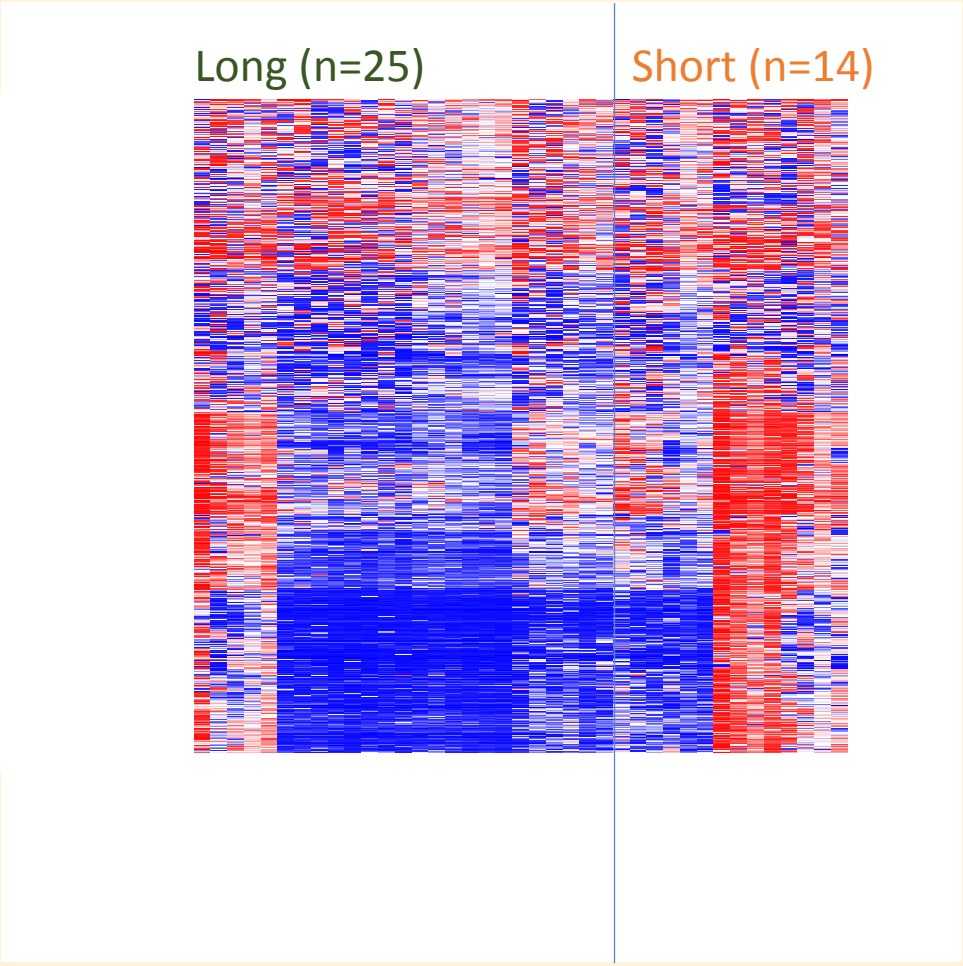
- p is large because there are 20k genes, millions of CpG sites, and human DNA spans 3 billion bases
- Standard methods such as linear discriminant analysis and logistic regression are literally not possible for p>n
- Even for methods that are possible we prefer a reduction in noisy probes to improve performance

How do we handle large p?

1. Shrinkage such as the lasso method
2. Dimensionality reduction such as PCA
3. Selecting the best subset by a statistical measure

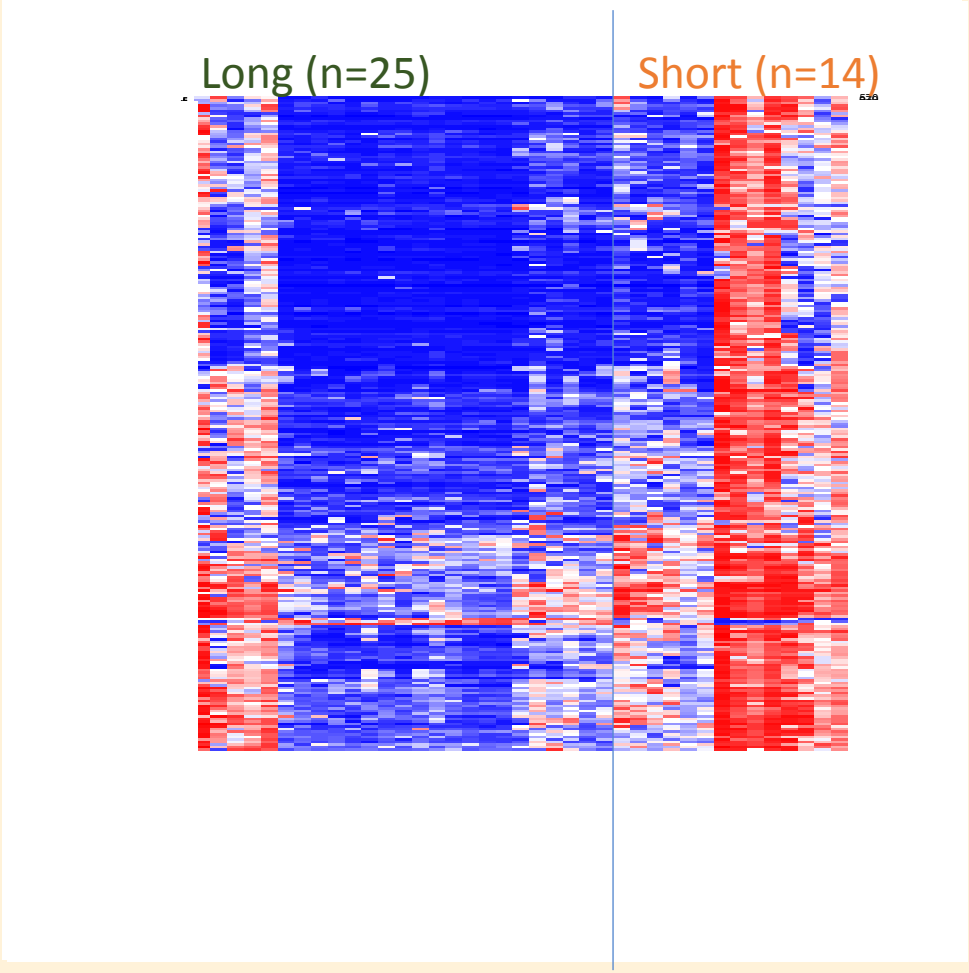
Selecting the best subset of markers by a statistical measure

- Suppose the problem is to distinguishing long and short survivors in methylation data
- Algorithm:
  - Choose best subset with t-test, Wilcoxon test, or other measure
  - Use tree, SVM, or other classifier with subset



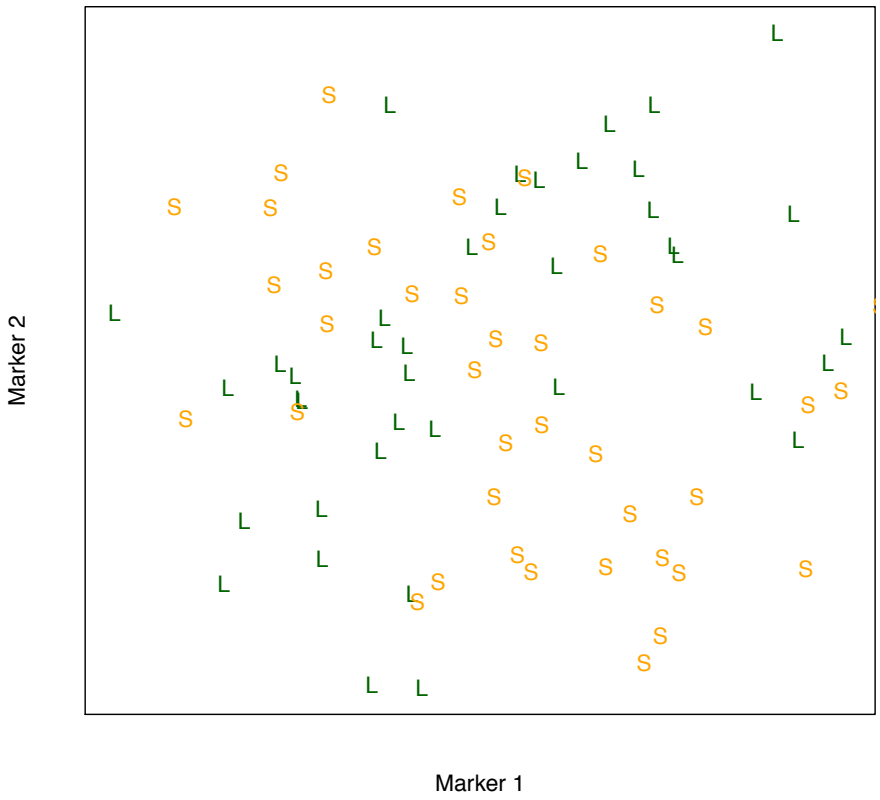
There are the 255 best markers by the Wilcoxon rank sum test

- Suppose the problem is to distinguish long and short survivors in methylation data
- Algorithm:
  - Choose the best subset with t-test, Wilcoxon test, or other measure
  - Use tree, SVM, or other classifier with subset



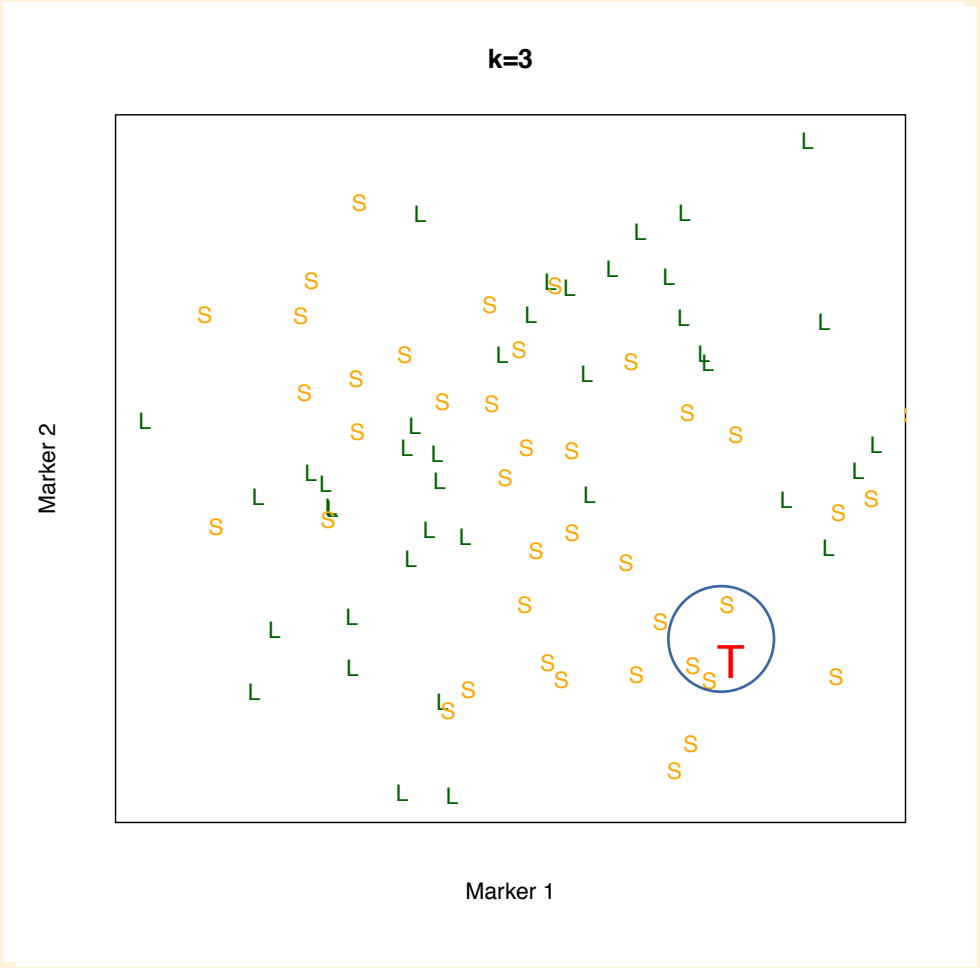
The k-nearest-neighbor rule for classification works well in many contexts

- Compute the distance between the unknown sample and all samples in the training set
- Identify the k nearest neighbors via distance
- Classify the unknown sample based on the majority class of the k nearest neighbors
- k should be odd and at least 3



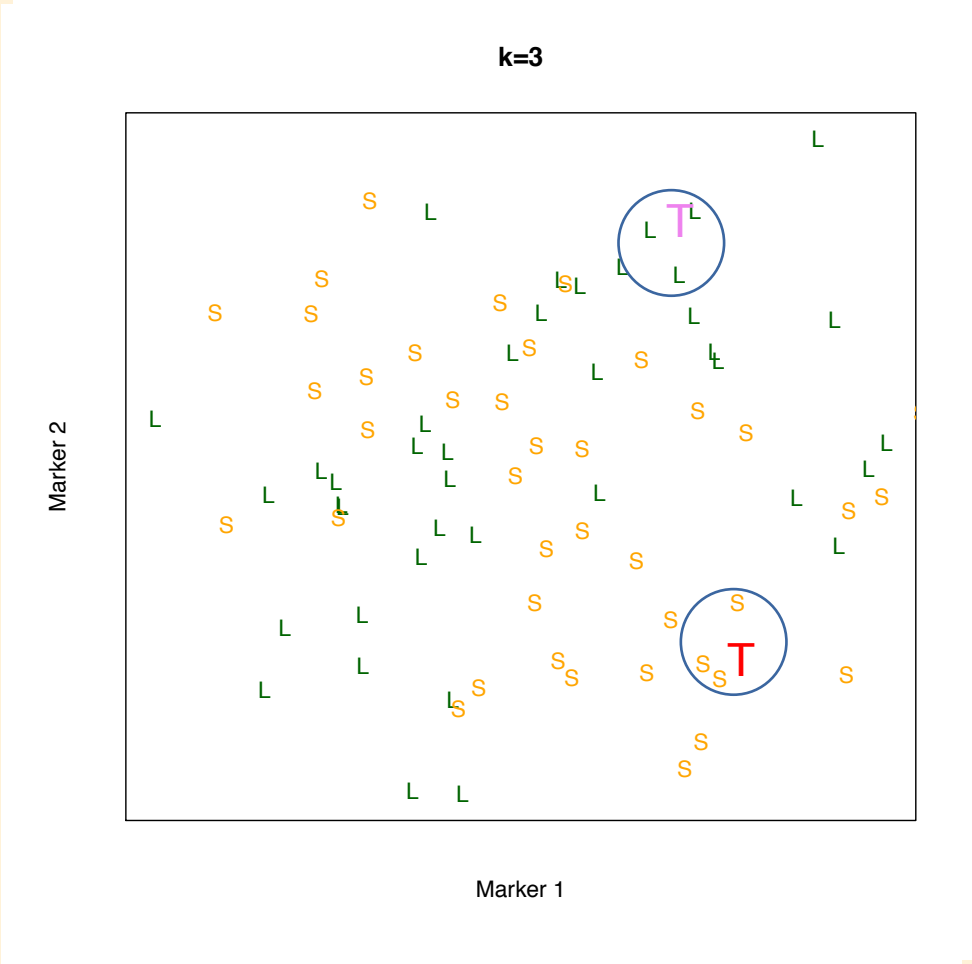
The k-nearest-neighbor rule

- Compute the distance between the unknown sample and all samples in the training set
- Identify the k nearest neighbors via distance
- Classify the unknown sample based on the majority class of the k nearest neighbors
- k should be odd and at least 3



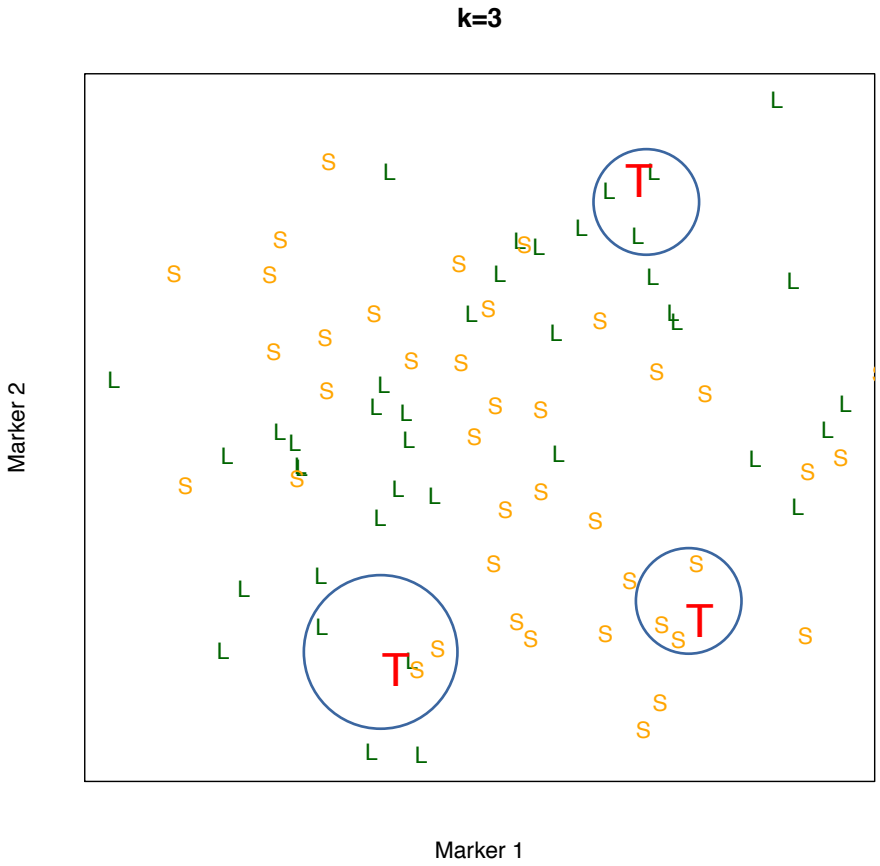
The k-nearest-neighbor rule

- Compute the distance between the unknown sample and all samples in the training set
- Identify the k nearest neighbors via distance
- Classify the unknown sample based on the majority class of the k nearest neighbors
- k should be odd and at least 3



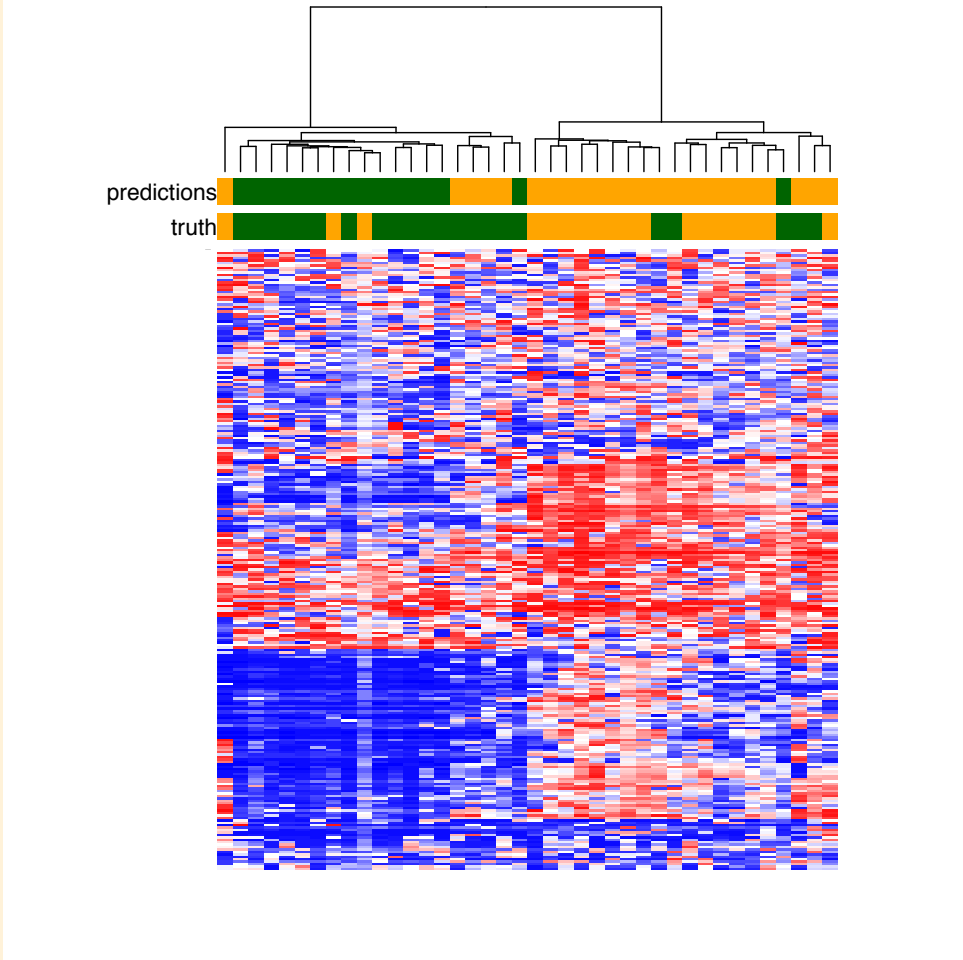
The k-nearest-neighbor rule

- Compute the distance between the unknown sample and all samples in the training set
- Identify the k nearest neighbors via distance
- Classify the unknown sample based on the majority class of the k nearest neighbors
- k should be odd and at least 3



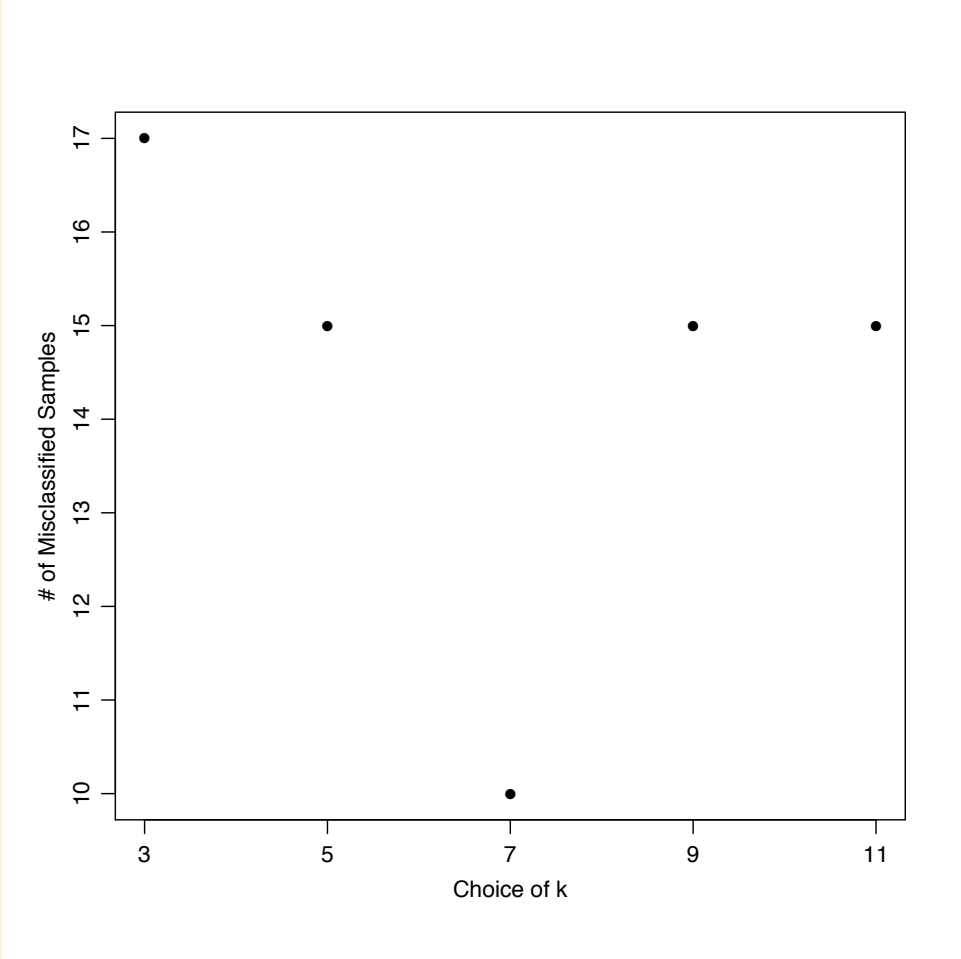
The k-nearest-neighbor rule on methylation test data

- Using k=7
- 16 out of 18 short survivors accurately called
- 14 out of 22 long survivors accurately called



Errors in prediction as k varies

- The success of the classifier varies substantially with k
- Choosing the best k could lead to a biased success estimate
- Two choices to deal with this issue:
  1. Choose k by cross-validation of training set and live with results
  2. Choose many values of k and have predictions vote and classify by the majority vote



Developing a scoring system

- The k-nearest-neighbor method and other methods assign new samples to a class
- Suppose one wants to estimate to what degree a sample belongs to a class
- Some idea of degree could be estimated by varying k and calculating proportion a sample is classified correctly
- Alternatively, a score could be developed to represent degree directly

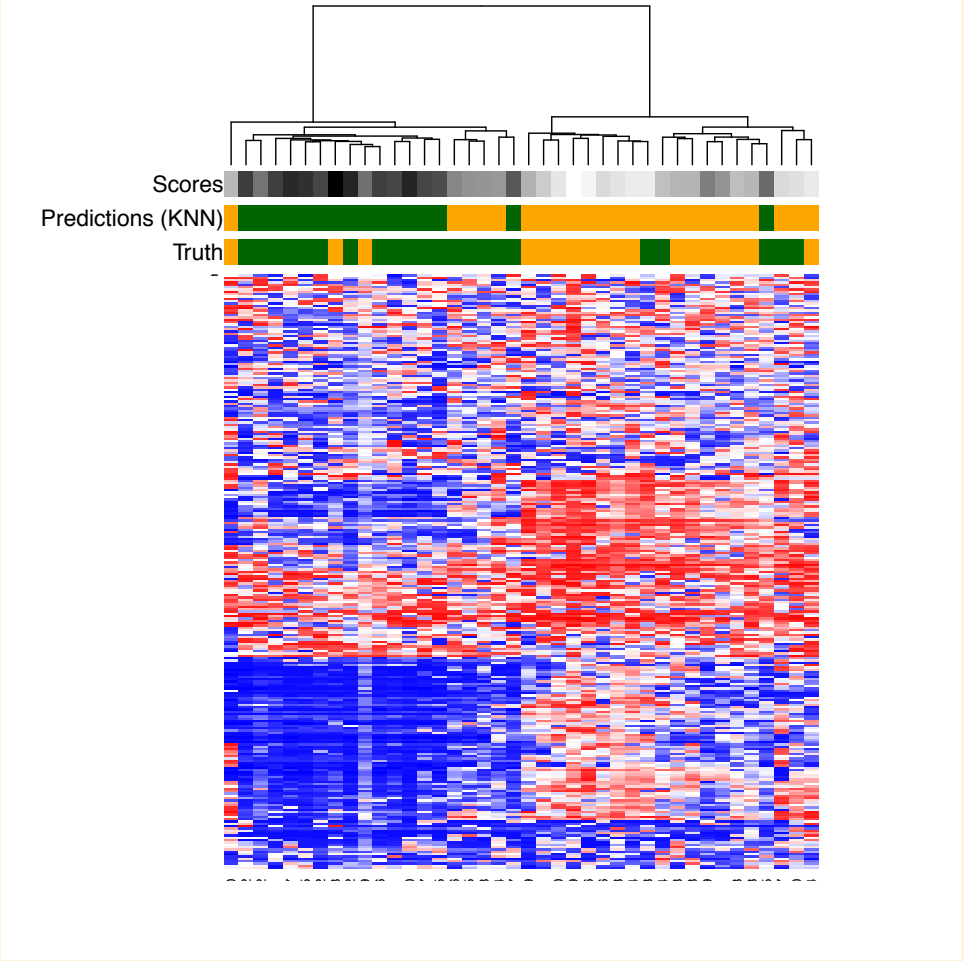
Scores depend on weighted top markers

An algorithm:

1. Choose the top *S* markers by the p-values of an appropriate test. Call these marker values for the *i*th sample $M_{i1},\ldots,M_{is}$
2. Weight these markers using t-statistics, principal components, or another weighting scheme. Call these weights $W_1,\ldots,W_s$ and use for all samples
3. Call the score for a sample $S_i^{initial}=W_1M_{i1}+..+W_sM_{is}$
4. $S_i^{final} = 100 \times (S_i^{initial} - \min(S_i^{initial}))/(\max(S_i^{initial})-\min(S_i^{initial}))$, with min and max across samples to put on 0 to 100 scale.

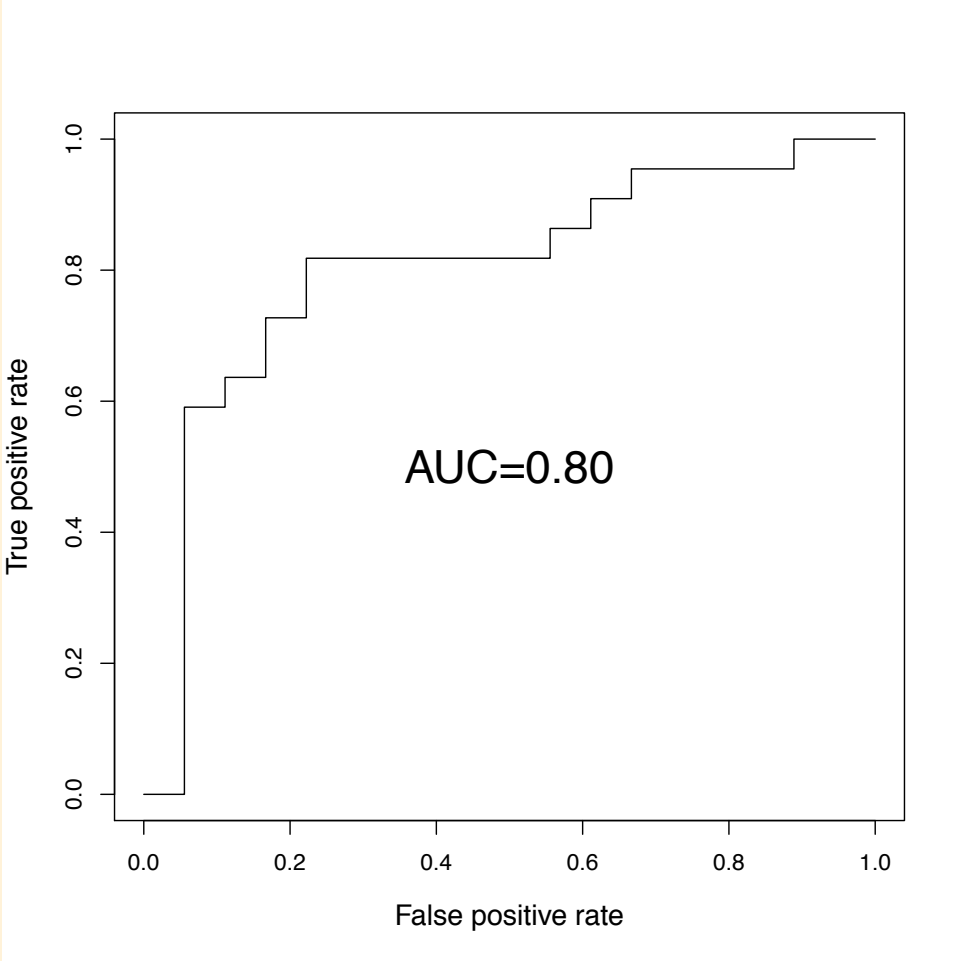
Scoring methylation test data

- Scoring method has been added to KNN predictions
- The lighter the box the more likely to have an event (orange)
- Note the knn and scoring methods are not the same so the predictions may vary



ROC Curve based on scores

- True positive rate = sensitivity is *proportion of samples with events predicted to have events*
- False positive rate = 1- specificity is *proportion of samples without events predicted to have events*
- ROC curve traces the tradeoff between these two quantities
- Area under the ROC curve (AUC) is a measure of the quality of the rule, with 0.5 random and 1 best



Summary of JMML methylation research

- The three methylation subtypes also found by investigators in Germany and Japan
- Each site has been developing assays simpler than microarrays to classify new patients into subtypes
- The hope is to have a subtype classifier in the clinic in the new few years to help guide treatment

Downstream analysis

- For the rest of the lecture we leave what I think of as machine learning and discuss biological interpretation
- The two methods discussed involve statistical analysis of machine learning results

Biological analysis of results

- Start with a list of markers that are in a cluster or have found to be differentially expressed
- If the marker list is related to a biological process it could mean that process is important to the findings
- For example if a group of related genes has higher expression in cancer patients with poorer outcomes the corresponding biological process may be part of the cause of the poorer outcome

Over-representation analysis

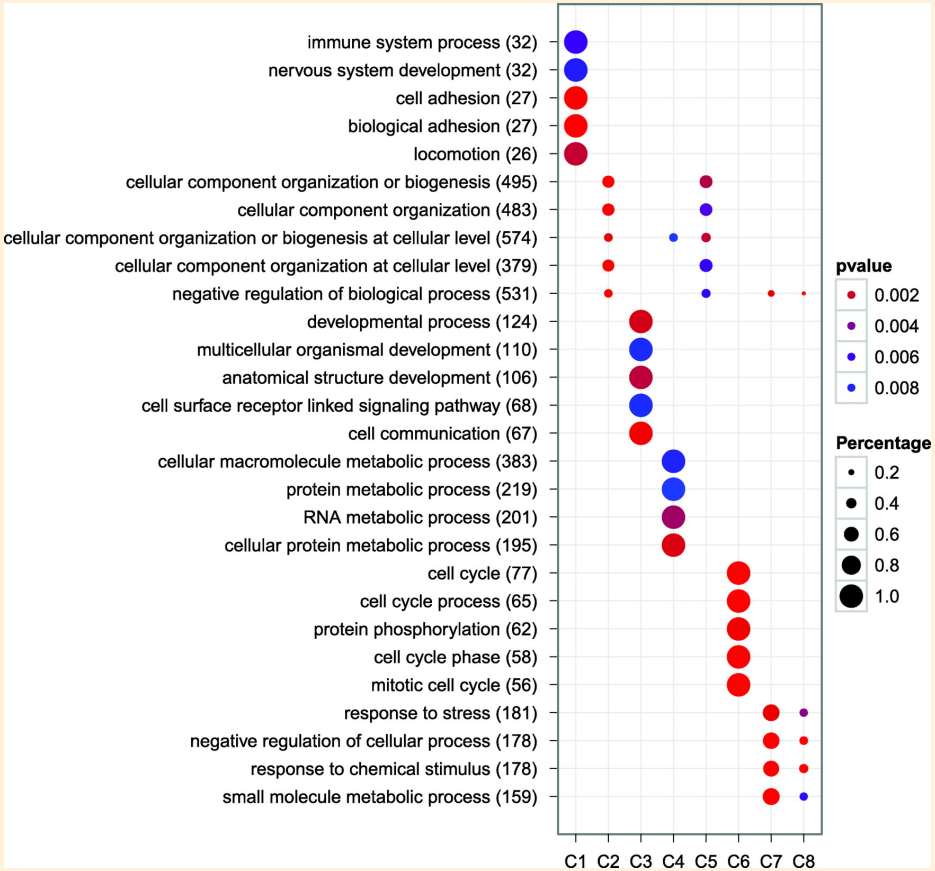
- Over-representation analysis identifies biological processes that are over-represented in a gene list
- Suppose we have this setup:
  - n11 be genes in a gene list and in a biological process list
  - n12 be genes in a gene list and not in a biological process list
  - n21 be genes not in a gene list and in a biological process list
  - n22 be genes not in a gene list and not in a biological process list

		Biological process +	Biological process -
Gene list	+	n11	n12
	-	n21	n22

2 by 2 table assessed by Fisher exact test
Is n11 big given other counts?

Over-representation analysis

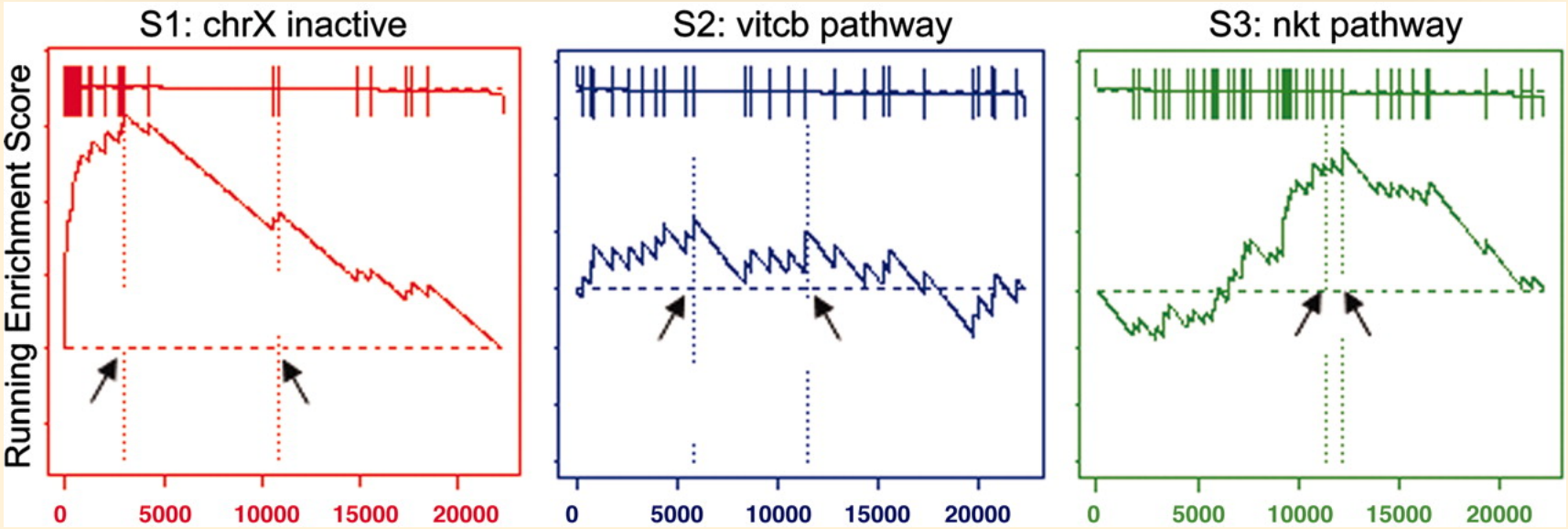
- Cluster on x-axis
- Biological process on the y-axis
- Only significant processes plotted
- Taken from clusterprofiler manuscript (Yu et al. OMICS: A Journal of Integrative Biology, 16(5), 284-287.)



Gene set enrichment analysis

- Gene set enrichment analysis (GSEA) is another common method for assessing a list of differentially expressed genes
- Here one orders genes by p-value or other method
- Then one can assess whether genes from a biological process are earlier in the ordering than expected

Gene set enrichment analysis
(Subramanian et al. PNAS October 25, 2005 102 (43) 15545-15550)



- GSEA analysis of three pathways
- Running enrichment score compares ranks of genes to expected ranks
- Arrows correspond to maximum deviation from expected ranks
- Analysis based on a weighted Kolmogorov-Smirnov test

Concluding remarks

- I introduced pre-processing steps including normalization and batch effect adjustment
- Through an extended data example we saw
  - Unsupervised methods like AHC and PCA
  - Simple supervised methods like KNN and scoring
- We saw how to biologically evaluate machine learning results
- You should be more prepared to evaluate genomic manuscripts and maybe approach your own analyses

R libraries and functions

- For clustering heatmap.2() from **gplots** package
- For principal components prcomp() from **core** package
- For k-nearest-neighbor knn() from **class** package
- For biological interpretation **clusterProfiler** package