Principles of Machine Learning

Day 1

Dennis Wylie, UT Bioinformatics Consulting Group

May 28, 2019

Outline

1 Introduction

2 Gene Expression Data

3 Data Wrangling

4 Normalization

5 Unsupervised Learning: Clustering

What is machine learning?

Perhaps better thought of as "algorithms for learning."

Such algorithms may also be referred to as modeling strategies

$$M$$

which, when provided training data

$$D_{\text{train}}$$

from some particular experiment, "learn" parameters

$$\boldsymbol{\theta}$$

such that the pair

$$(M, \boldsymbol{\theta})$$

can be used to predict likely observations

$$D_{\text{other}}$$

from similar experiments.

Taxonomy of machine learning

Often subdivided into three categories:

Supervised $D = (\mathbf{x}, y)$ consists of inputs $\mathbf{x}$ and outcomes y, with focus on predicting y given $\mathbf{x}$.

Unsupervised $D = \mathbf{x}$ with no particular outcome identified; focus instead on identifying common patterns in $\mathbf{x}$ alone.

Reinforcement $D = (a, \mathbf{x}, y)$ in which the outcome y is also influenced by actions a over which the modeler has control and the focus is on identifying those a most likely to generate desirable y.

Reinforcement learning is not currently very highly studied in the context of gene expression data.

Probabilities

Machine learning can be described probabilistically

- using random variables $\mathbf{X}$ and/or Y and
- defining predictions of fit model $(M, \boldsymbol{\theta})$ as

$$\begin{aligned} \mathbb{P}(\mathbf{X} \mid M, \boldsymbol{\theta}) &\quad \text{(Unsupervised)} \\ \mathbb{P}(\mathbf{X}, Y \mid M, \boldsymbol{\theta}) &\quad \text{(Supervised, Generative)} \\ \mathbb{P}(Y \mid \mathbf{X}, M, \boldsymbol{\theta}) &\quad \text{(Supervised, Discriminative)} \end{aligned}$$

Probabilities

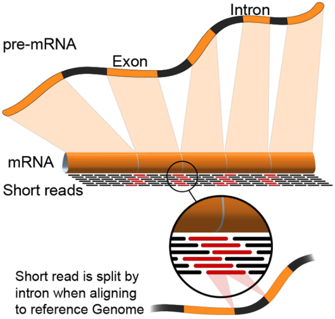
Machine learning can be described probabilistically

- using random variables $\mathbf{X}$ and/or Y and
- defining predictions of fit model $(M, \boldsymbol{\theta})$ as

$$\begin{aligned} \mathbb{P}(\mathbf{X} \mid M, \boldsymbol{\theta}) & \quad \text{(Unsupervised)} \\ \mathbb{P}(\mathbf{X}, Y \mid M, \boldsymbol{\theta}) & \quad \text{(Supervised, Generative)} \\ \mathbb{P}(Y \mid \mathbf{X}, M, \boldsymbol{\theta}) & \quad \text{(Supervised, Discriminative)} \end{aligned}$$

Discriminative algorithms fit only conditional $\mathbb{P}(Y \mid \mathbf{X}, M, \boldsymbol{\theta})$, thereby remaining agnostic about the distribution of $\mathbf{X}$.

RNA-seq


Intron
pre-mRNA
Exon
mRNA
Short reads
Short read is split by intron when aligning to reference Genome

- Most detailed picture of gene expression
- Can detect novel transcripts, alternative splicing, SNVs
- Analysis can be done at exon, transcript, or gene level

RNA-seq set: Shen *et al.* (2012)

Data set obtained from
`http://chromosome.sdsc.edu/mouse/download.html`

> 19 tissues and primary cell types were examined using ChIP-Seq, **RNA-Seq**. Additionally we performed HiC experiments in mouse cortex.
>
> . . . functional sequences in the mouse genome are still poorly annotated a decade after its initial sequencing. We report here a map of nearly 300,000 cis-regulatory sequences in the mouse genome, representing active promoters, enhancers and CTCF binding sites in a diverse set of 19 tissues and cell types. . .

We're only going to look at the RNA-seq data.

RNA-seq set: Patel *et al.* (2014): GSE57872

Data set obtained from Gene Expression Omnibus (GEO) using `GEOquery` (Davis & Meltzer (2007)).

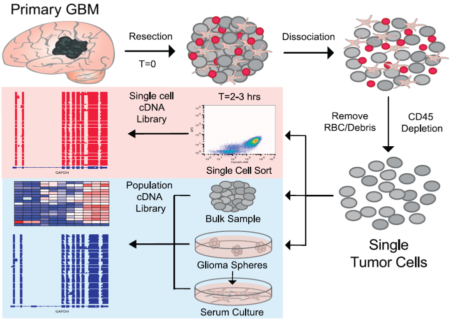
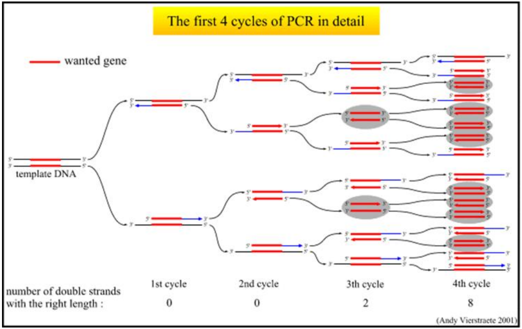


Fig. 1. Intratumoral glioblastoma heterogeneity quantified by single-cell RNA-seq. (A) Workflow depicts rapid dissociation and isolation of glioblastoma cells from primary tumors for generating single-cell and bulk qRNA-seq profiles and deriving glioblastoma culture models.

RT-qPCR

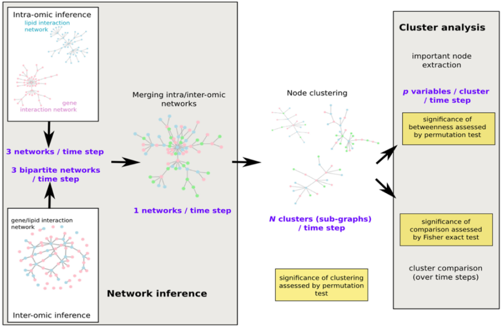

The first 4 cycles of PCR in detail
wanted gene
template DNA
number of double strands with the right length :
1st cycle
0
2nd cycle
0
3th cycle
2
4th cycle
8

(Andy Vierstraete 2001)

- Count number of cycles (Ct) required for fluorescence signal to surpass threshold
  - $\text{Ct} \propto 2^{-(\text{copy number})}$
- Analysis simpler than for RNA-seq
- Need primer pair for gene of interest
- May be cheaper/easier than RNA-seq for measurement of small number of genes

RT-qPCR set: Montastier *et al.* (2015): GSE60946

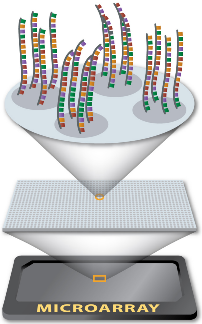
Obtained from GEO using `GEOquery` (Davis & Meltzer (2007)).


Intra-omic inference
lipid interaction network
gene interaction network
3 networks / time step
3 bipartite networks / time step
gene/lipid interaction network
Inter-omic inference
Merging intra/inter-omic networks
1 networks / time step
Network inference
Node clustering
N clusters (sub-graphs) / time step
significance of clustering assessed by permutation test
Cluster analysis
important node extraction
p variables / cluster / time step
significance of betweenness assessed by permutation test
significance of comparison assessed by Fisher exact test
cluster comparison (over time steps)

AT fatty acids and mRNA levels were quantified in 135 obese women at baseline, after an 8-week low calorie diet (LCD) and after 6 months of ad libitum weight maintenance diet (WMD) . . .

A 3 steps approach . . . consisted in inferring intra-omic networks with sparse partial correlations and inter-omic networks with regularized canonical correlation analysis and finally combining the obtained omic-specific network in a single global model.

Microarray



- Analysis simpler than for RNA-seq
- May be cheaper than RNA-seq
- Throughput intermediate between RT-qPCR and RNA-seq
- Lower sensitivity, dynamic range than RNA-seq

Microarray set: Hess *et al.* (2006)

Data set downloaded from `http://bioinformatics.mdanderson.org/pubdata.html`.

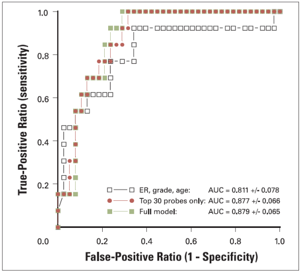


Fig 3. Receiver operating characteristic curves of three distinct pathologic complete response prediction models. The performance of the Diagonal Linear Discriminant Analysis–30 predictor and a predictor based on clinical variables and a combined clinical + pharmacogenomic prediction model are shown in the validation set (n = 51). ER, estrogen receptor; AUC, area under the curve.

We developed a multigene predictor of pathologic complete response (pCR) to preoperative weekly paclitaxel and fluorouracil-doxorubicin-cyclophosphamide (T/FAC) chemotherapy and assessed its predictive accuracy on independent cases.

Loading tabular data

For this class, data provided in tab-delimited text files with header in first column and index in first row.

```
# R:

df = read.table(file, header=TRUE, row.names=1, sep='\t')
```

```
# Python:

import pandas
df = pandas.read_csv(file, header=0, index_col=0, sep='\t')
```

I will use the "=" assignment operator in R in order to minimize differences between R and Python.

The pandas library (McKinney (2012)) for Python provides a DataFrame similar (and in some ways superior) to R's data.frame.

Accessing data — individual elements

Assuming column names are capital letters and row names lower-case:

```
# R:

df[1, 2]
df['a', 'B']
df[1, 'B']
df$B[1]
```

```
# Python:

df.iloc[0, 1]
df.loc['a', 'B']
df.ix[0, 'B']
df['B'][0]
df.B[0]
```

Accessing data — whole Rows or Columns

```
# R:

df[1, ]              ## returns row as data.frame
df['a', ]            ##  same
df[ , 2, drop=FALSE] ## returns column as data.frame
df[ , 2]             ## returns column as vector
df[ , 'B']           ##  same
df$B                 ##  same
```

```
# Python:

df.iloc[0]           ## returns row as pandas.Series
df.loc['a']          ##  same
df.iloc[ [0] ]       ## returns row as pandas.DataFrame
df[ df.columns[1] ]  ## returns column as pandas.Series
df['B']              ##  same
df.B                 ##  same
df[ ['B'] ]          ## returns column as pandas.DataFrame
```

Accessing data — subframes

In both R and Python, asking for *R* rows and *C* columns simultaneously returns *R* × *C* `[dD]ata\.?[fF]rame`.

```
# R:

df[1:3, 1:3]
df[c('a', 'b', 'c'), c('A', 'B', 'C')]
```

```
# Python:

df.iloc[0:3, 0:3]
df.loc[ ['a', 'b', 'c'], ['A', 'B', 'C'] ]
```

Accessing data — where. . .

In both `R` and `Python`, can also select rows or columns of `[dD]ata\.?[fF]rame` using boolean vectors (or matrices).

```
# R:

df[df$B > 0, ]           ## all rows where df$B > 0
df[df$B > 0, 'C']        ## col C vals where df$B > 0
df[df$B > 0, 'B'] = 0    ## now all df$B <= 0
df[ , df[1, ] > 0]       ## all cols where first row > 0
```

```
# Python:

df.loc[df['B'] > 0]        ## all rows where df.B > 0
df.loc[df['B'] > 0, 'C']   ## col C vals where df.B > 0
df[df.B > 0, 'B'] = 0      ## now all df['B'] <= 0
df.loc[:, df.iloc[0] > 0] ## all cols where first row > 0
```

Normalization — RNA-seq

Basic measurement unit: count of reads mapped to a given marker (gene, exon, etc.).

Besides biological expression levels, many technical factors influence counts as well, e.g.:

1. differences in library size (sequencing depth)
2. length of gene

Normalization — RNA-seq

Basic measurement unit: count of reads mapped to a given marker (gene, exon, etc.).

Besides biological expression levels, many technical factors influence counts as well, e.g.:

1. differences in library size (sequencing depth)
2. length of gene

Simplest normalization schemes account for these influences by

1. dividing the total library size (and multiplying by 10^6) to obtain CPM or
2. further dividing by gene length (and multiplying by 10^3) to obtain RPKM

Normalization for gene length may not be necessary in studies which do not attempt to compare expression levels between different genes.

Normalization — RNA-seq: better

Some studies have found that RPKM normalization may not appropriately control for association between gene length and read counts (Dillies *et al.* (2013)).

Further, both CPM and RPKM may overweight influence of few very highly expressed genes which may actually be differentially expressed across samples.

Normalization — RNA-seq: better

Some studies have found that RPKM normalization may not appropriately control for association between gene length and read counts (Dillies *et al.* (2013)).

Further, both CPM and RPKM may overweight influence of few very highly expressed genes which may actually be differentially expressed across samples.

A bit more complicated: "**R**elative **L**og **E**xpression":

- start with read counts r_{ig}
- calculate mean log expression $\frac{1}{n}\sum_j \log r_{jg}$ for gene g
- normalization (size) factor τ_i for sample i:

$$\tau_i = \operatorname*{median}_g \left\{ \frac{r_{ig}}{\exp\left(\frac{1}{n}\sum_j \log r_{jg}\right)} \right\}$$

- normalized expression matrix defined by $\frac{r_{ig}}{\tau_i}$

Normalization — RNA-seq: RLE (DESeq)

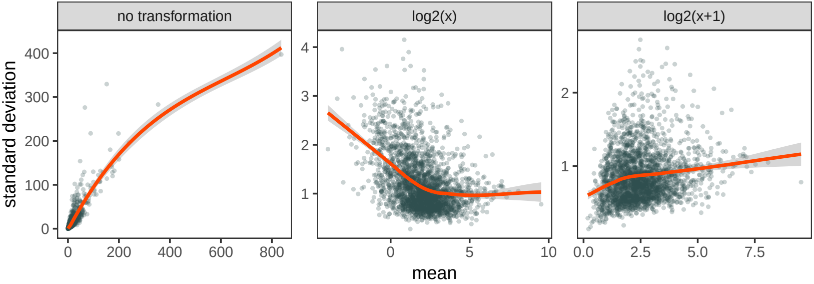
```
# R:

rleSizeFactors = function(x) {
    require(matrixStats)
    xno0 = x[ , colMins(x) > 0]
    geoMeans = exp(colMeans(log(xno0)))
    sizeFactors = rowMedians(sweep(xno0, 2, geoMeans, `/`))
    names(sizeFactors) = rownames(x)
    return(sizeFactors)
}
```

```
# Python:

def rleSizeFactors(x):
    xno0 = x.loc[:, x.min(axis=0) > 0]
    geoMeans = np.exp(np.log(xno0).mean(axis=0))
    sizeFactors = xno0.divide(geoMeans, axis=1).median(axis=1)
    return sizeFactors
```

Variance stabilization



Many statistical methods assume *homoskedasticity*

- i.e., standard deviation independent of mean

This is not true for either counts or RLE-normalized counts!

Adding a small number and then logging is approximate variance-stabilization transformation

$$x_{ig} = f\left(\frac{r_{ig}}{\tau_i}\right)$$

Normalization — RT-qPCR

Basic measurement of RT-qPCR: Ct for given gene (primer pair).

Once again, technical factors such as quantity or quality of nucleic acid in sample may influence measured Ct values.

Since Ct values are already in log-copy number space, simple sample-mean-centering approach can work well. . .

$$\Delta x_{ig} = x_{ig} - \frac{1}{p} \sum_{h=1}^{p} x_{ih}$$

Normalization — RT-qPCR

Basic measurement of RT-qPCR: Ct for given gene (primer pair).

Once again, technical factors such as quantity or quality of nucleic acid in sample may influence measured Ct values.

Since Ct values are already in log-copy number space, simple sample-mean-centering approach can work well. . .

$$\Delta x_{ig} = x_{ig} - \frac{1}{p} \sum_{h=1}^{p} x_{ih}$$

. . . if many genes are measured with expectation that most are not differentially expressed and . . .

Normalization — RT-qPCR

Basic measurement of RT-qPCR: Ct for given gene (primer pair).

Once again, technical factors such as quantity or quality of nucleic acid in sample may influence measured Ct values.

Since Ct values are already in log-copy number space, simple sample-mean-centering approach can work well. . .

$$\Delta x_{ig} = x_{ig} - \frac{1}{p} \sum_{h=1}^{p} x_{ih}$$

. . . if many genes are measured with expectation that most are not differentially expressed and . . .

. . . if none of the Ct values x_{ig} are missing/undefined.

Normalization — RT-qPCR: Mean-Centering

```
# R:

meanCenter = function(x, MARGIN=1) {
    geneHasNAs = apply(x, 3-MARGIN, function(z) {any(is.na(z))})
    means = apply(x, MARGIN, function(z) {mean(z[!geneHasNAs])})
    return(sweep(x, MARGIN, means, `-`))
}
```

```
# Python:

def meanCenter(x, axis=0):
    geneHasNans = (numpy.isnan(x).sum(axis=axis) > 0)
    if axis == 0:
        xnonans = x[ x.columns[~geneHasNans] ]
    elif axis == 1:
        xnonans = x.loc[~geneHasNans]
    means = xnonans.mean(axis=1-axis)
    return x.add(-means, axis=axis)
```

Normalization — RT-qPCR: Normalizers

Conceptually more difficult to deal with RT-qPCR data normalization when most measured genes are differentially expressed.

Usual answer in this case is to include a few “stably expressed” **normalizer** genes in panel.

How does one know what genes are stably expressed?

Normalization — RT-qPCR: Normalizers

Conceptually more difficult to deal with RT-qPCR data normalization when most measured genes are differentially expressed.

Usual answer in this case is to include a few “stably expressed” **normalizer** genes in panel.

How does one know what genes are stably expressed?

1. Use genes other people have declared stable in literature, or

Normalization — RT-qPCR: Normalizers

Conceptually more difficult to deal with RT-qPCR data normalization when most measured genes are differentially expressed.

Usual answer in this case is to include a few “stably expressed” **normalizer** genes in panel.

How does one know what genes are stably expressed?

1. Use genes other people have declared stable in literature, or
2. First apply algorithm to identify normalizers (e.g., Vandesompele *et al.* (2002); Andersen *et al.* (2004); Wylie *et al.* (2011)) to large panel where most genes are not expected to be differentially expressed.

Unsupervised learning

$D = \mathbf{x}$ with no particular outcome identified; focus on identifying common patterns in $\mathbf{x}$ alone.

What do we mean by “patterns?”

- clusters (subgroupings of “similar” samples or genes)
- relationships between variables (gene expression levels or other covariates)
    - strong relationships may lead to identification of hidden/latent factors simultaneously influencing many variables
    - useful for **dimensionality reduction**

While most approaches *can* be represented as probabilistic model

$$\mathbb{P}(\mathbf{X} \mid M, \boldsymbol{\theta})$$

some may be more simply presented without the extra theoretical baggage.

Clustering

Want to find groups of samples *i* or genes *g* such that:

- high similarity of objects within same group
- low similarity between objects in different groups;
- often want clusters to be *disjoint*.

Useful to check data quality/confirm expectations (or spot unexpected structure in data).

Clustering

Want to find groups of samples *i* or genes *g* such that:

- high similarity of objects within same group
- low similarity between objects in different groups;
- often want clusters to be *disjoint*.

Useful to check data quality/confirm expectations (or spot unexpected structure in data).

- if replicates are present, do they cluster together?

Clustering

Want to find groups of samples *i* or genes *g* such that:

- high similarity of objects within same group
- low similarity between objects in different groups;
- often want clusters to be *disjoint*.

Useful to check data quality/confirm expectations (or spot unexpected structure in data).

- if replicates are present, do they cluster together?
- do samples taken from similar tissues, conditions, time points, etc. cluster together?

Clustering

Want to find groups of samples i or genes g such that:

- high similarity of objects within same group
- low similarity between objects in different groups;
- often want clusters to be *disjoint*.

Useful to check data quality/confirm expectations (or spot unexpected structure in data).

- if replicates are present, do they cluster together?
- do samples taken from similar tissues, conditions, time points, etc. cluster together?
- do samples cluster by processing batch or order?

Similarity, dissimilarity, and distance

Usually work with *dissimilarity* measures (often distance metrics).

Common dissimilarity metrics:

1. **Euclidean distance** $d(\mathsf{x}_1, \mathsf{x}_2) = \|\mathsf{x}_1 - \mathsf{x}_2\|_2$
2. **Pearson correlation dissimilarity**

$$d(\mathsf{x}_1, \mathsf{x}_2) = 1 - \frac{\Delta \mathsf{x}_1 \cdot \Delta \mathsf{x}_2}{\|\Delta \mathsf{x}_1\| \|\Delta \mathsf{x}_2\|}$$

where $\Delta \mathsf{x} = \mathsf{x} - \frac{1}{p} \sum_{g=1}^{p} x_g$.

3. **Spearman correlation dissimilarity**

$$d(\mathsf{x}_1, \mathsf{x}_2) = 1 - \frac{\Delta \mathrm{rank}(\mathsf{x}_1) \cdot \Delta \mathrm{rank}(\mathsf{x}_2)}{\|\Delta \mathrm{rank}(\mathsf{x}_1)\| \|\Delta \mathrm{rank}(\mathsf{x}_2)\|}$$

k-Means clustering (MacQueen (1967))

Algorithm:

1. Initialize k "centroids" $\mathbf{c}_a$.
2. Assign each datum $\mathbf{x}_i$ to nearest cluster:

$$\text{clust}(\mathbf{x}_i) = \arg\min_a \|\mathbf{x}_i - \mathbf{c}_a\|$$

3. Reset centroids to mean of associated data:

$$\mathbf{c}_a = \frac{1}{|S_a|} \sum_{i \in S_a} \mathbf{x}_i$$

   where the set $S_a = \{i \mid \text{clust}(\mathbf{x}_i) = a\}$.

4. Repeat steps 2-3 until convergence.

k-Means clustering (MacQueen (1967))

Algorithm:

1. Initialize k "centroids" $\mathbf{c}_a$.
2. Assign each datum $\mathbf{x}_i$ to nearest cluster:

$$\mathrm{clust}(\mathbf{x}_i) = \arg\min_a \|\mathbf{x}_i - \mathbf{c}_a\|$$

3. Reset centroids to mean of associated data:

$$\mathbf{c}_a = \frac{1}{|S_a|} \sum_{i \in S_a} \mathbf{x}_i$$

where the set $S_a = \{i \mid \mathrm{clust}(\mathbf{x}_i) = a\}$.

4. Repeat steps 2-3 until convergence.

Locally minimizes $\sum_{a=1}^{k} \sum_{i \in S_a} (\mathbf{x}_i - \mathbf{c}_a)^2$.

k-Means clustering

k-means clustering is fast and intuitive . . .

. . . but tends to produce (hyper)spherical, equal-sized clusters

- whether they are appropriate or not.

k-Means clustering

k-means clustering is fast and intuitive ...

... but tends to produce (hyper)spherical, equal-sized clusters

- whether they are appropriate or not.

Can be derived from small σ limit of

- probabilistic mixture-of-Gaussians model M
- with parameters $\boldsymbol{\theta} = (\mathbf{c}, \sigma)$ (Ghahramani (2004)):

$$\mathbb{P}(\mathbf{X} = \mathbf{x} \mid M, \mathbf{c}, \sigma) = \sum_{a=1}^{k} \frac{1}{k\sqrt{(2\pi\sigma^2)^p}} \exp\left[\frac{(\mathbf{x} - \mathbf{c}_a)^2}{2\sigma^2}\right]$$

where each Gaussian in the mixture has

- its own centroid vector $\mathbf{c}_a$ but share
- common spherical covariance matrix $\sigma^2 I$.

Hierarchical clustering

Also known as agglomerative (bottom-up) clustering (Mary-Huard *et al.* (2006); Hastie *et al.* (2009)).

Requires extension of (dis)similarity metric from pairs of data $d(\mathsf{x}_i, \mathsf{x}_j)$ to pairs of *clusters*:

$$d(S_a, S_b) = ???$$

For example, so-called "average linkage" defines

$$d(S_a, S_b) = \sum_{i \in S_a} \sum_{j \in S_b} \frac{d(\mathsf{x}_i, \mathsf{x}_j)}{|S_a||S_b|}$$

. . . but there are other possible aggregation criteria as well.

Hierarchical clustering

Algorithm:

1. Initialize each datum to own cluster, $S_i = \{i\}$, define initial set of active clusters $A_0 = \{1, 2, \ldots, n\}$.
2. For iteration t, select two most similar active clusters and merge:

$$\begin{aligned} (a_t, b_t) &= \operatorname*{arg\,min}_{(a,b) \in A_{t-1} \times A_{t-1} \,|\, a<b} d(S_a, S_b) \\ S_{n+t} &= S_{a_t} \cup S_{b_t} \\ A_t &= (A_{t-1} \setminus \{a_t, b_t\}) \cup \{n+t\} \end{aligned}$$

3. If $t < (n-1)$, increment t and repeat step 2. (Note: if you know you want exactly k clusters, stop when $t = n - k$.)

Hierarchical clustering

Algorithm:

1. Initialize each datum to own cluster, $S_i = \{i\}$, define initial set of active clusters $A_0 = \{1, 2, \ldots, n\}$.
2. For iteration t, select two most similar active clusters and merge:

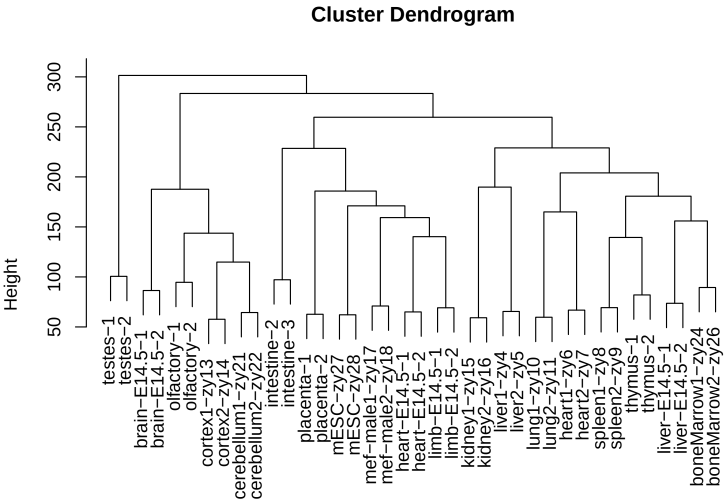
$$\begin{aligned} (a_t, b_t) &= \operatorname*{arg\,min}_{(a,b) \in A_{t-1} \times A_{t-1} \,|\, a<b} d(S_a, S_b) \\ S_{n+t} &= S_{a_t} \cup S_{b_t} \\ A_t &= (A_{t-1} \setminus \{a_t, b_t\}) \cup \{n+t\} \end{aligned}$$

3. If $t < (n-1)$, increment t and repeat step 2. (Note: if you know you want exactly k clusters, stop when $t = n - k$.)

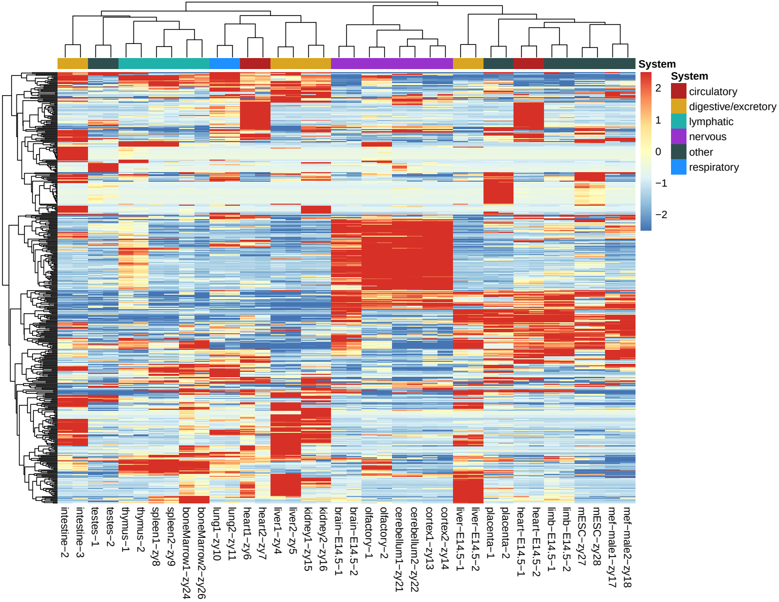
Dendrogram obtained from this process by connecting

- merged clusters a_t and b_t to the new merged cluster $(n+t)$
- sequentially for each iteration t.

Hierarchical clustering (Shen samples)



Hierarchical clustering (high variance genes)



Hierarchical clustering

Some commonly used aggregation criteria:

Average Linkage

$$d(S_a, S_b) = \sum_{i \in S_a} \sum_{j \in S_b} \frac{d(\mathbf{x}_i, \mathbf{x}_j)}{|S_a||S_b|}$$

Single Linkage

$$d(S_a, S_b) = \min_{i \in S_a, j \in S_b} d(\mathbf{x}_i, \mathbf{x}_j)$$

Complete Linkage

$$d(S_a, S_b) = \max_{i \in S_a, j \in S_b} d(\mathbf{x}_i, \mathbf{x}_j)$$

Centroid (where $\mathbf{c}_a$ is centroid of cluster a)

$$d(S_a, S_b) = d(\mathbf{c}_a, \mathbf{c}_b)$$

Ward

$$d^2(S_a, S_b) = \frac{|S_a||S_b|}{|S_a| + |S_b|} d^2(\mathbf{c}_a, \mathbf{c}_b)$$

References I

Andersen, Claus Lindbjerg, Jensen, Jens Ledet, & Ørntoft, Torben Falck. 2004. Normalization of real-time quantitative reverse transcription-PCR data: a model-based variance estimation approach to identify genes suited for normalization, applied to bladder and colon cancer data sets. *Cancer Research*, **64**(15), 5245–5250.

Davis, Sean, & Meltzer, Paul S. 2007. GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. *Bioinformatics*, **23**(14), 1846–1847.

Dillies, Marie-Agnès, Rau, Andrea, Aubert, Julie, Hennequet-Antier, Christelle, Jeanmougin, Marine, Servant, Nicolas, Keime, Céline, Marot, Guillemette, Castel, David, Estelle, Jordi, *et al.* . 2013. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*, **14**(6), 671–683.

Ghahramani, Zoubin. 2004. Unsupervised Learning. *Pages 72–112 of: Advanced Lectures on Machine Learning*. Springer.

Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome. 2009. *The Elements of Statistical Learning*. Springer.

Hess, Kenneth R, Anderson, Keith, Symmans, W Fraser, Valero, Vicente, Ibrahim, Nuhad, Mejia, Jaime A, Booser, Daniel, Theriault, Richard L, Buzdar, Aman U, Dempsey, Peter J, *et al.* . 2006. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology*, **24**(26), 4236–4244.

MacQueen, James. 1967. Some methods for classification and analysis of multivariate observations. *Pages 281–297 of: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. University of California Press.

Mary-Huard, Tristan, Picard, Franck, & Robin, Stéphane. 2006. Introduction to statistical methods for microarray data analysis. *Mathematical and Computational Methods in Biology. Paris: Hermann.*

McKinney, Wes. 2012. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, Inc.

References II

Montastier, Emilie, Villa-Vialaneix, Nathalie, Caspar-Bauguil, Sylvie, Hlavaty, Petr, Tvrzicka, Eva, Gonzalez, Ignacio, Saris, Wim HM, Langin, Dominique, Kunesova, Marie, & Viguerie, Nathalie. 2015. System Model Network for Adipose Tissue Signatures Related to Weight Changes in Response to Calorie Restriction and Subsequent Weight Maintenance. *PLoS Computational Biology*, **11**(1).

Patel, Anoop P, Tirosh, Itay, Trombetta, John J, Shalek, Alex K, Gillespie, Shawn M, Wakimoto, Hiroaki, Cahill, Daniel P, Nahed, Brian V, Curry, William T, Martuza, Robert L, *et al.* . 2014. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, **344**(6190), 1396–1401.

Shen, Yin, Yue, Feng, McCleary, David F, Ye, Zhen, Edsall, Lee, Kuan, Samantha, Wagner, Ulrich, Dixon, Jesse, Lee, Leonard, Lobanenkov, Victor V, *et al.* . 2012. A map of the cis-regulatory sequences in the mouse genome. *Nature*, **488**(7409), 116.

Vandesompele, Jo, De Preter, Katleen, Pattyn, Filip, Poppe, Bruce, Van Roy, Nadine, De Paepe, Anne, & Speleman, Frank. 2002. Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. *Genome Biology*, **3**(7), research0034.

Wylie, Dennis, Shelton, Jeffrey, Choudhary, Ashish, & Adai, Alex T. 2011. A novel mean-centering method for normalizing microRNA expression from high-throughput RT-qPCR data. *BMC Research Notes*, **4**(1), 555.