High-Throughput Methods: Microarrays

• one common HT study type: measuring RNA abundances

• what is varied: individuals, strains, cell types, environmental conditions, disease states, etc.

• what is measured: RNA quantities for thousands of genes, exons or other transcribed sequences
Expression Profiles

- for most of these methods, we can think of the output as a 2D matrix

- rows represent genes

- columns represent different experimental conditions (e.g., people with different types of leukemia)

- each row/column represents an expression profile
Microarrays

- a.k.a. DNA chips, gene chips, DNA arrays etc.

- two general types that are popular
  - spotted arrays (pioneered by Pat Brown @ Stanford)
  - oligonucleotide arrays (pioneered by Affymetrix Inc.)

- both based on the same basic principles
  - anchoring pieces of DNA to glass/nylon/silicon slides
  - complementary hybridization
Microarrays

• spotted array animation
  – http://www.bio.davidson.edu/courses/genomics/chip/chip.swf

• main differences with oligonucleotide arrays:
  – instead of putting entire genes on an array, put sets of DNA oligonucleotides (fixed length sequences, typically 25-60 nucleotides in length)
  – mRNA samples are processed separately on individual arrays, instead of in pairs
Computational Tasks

- **differential expression**
  - which genes have different expression levels across two groups?

- **clustering**
  - which genes seem to be regulated together?
  - which treatments/individuals have similar expression profiles?

- **classification**
  - to which functional class does a given gene belong?
  - to which class does a given sample belong? (e.g., does this patient have ALL or AML?)
Clustering Gene Expression Profiles

• given:
  – expression profiles for a set of genes or experiments/individuals/time points (whatever columns represent)

• do:
  – organize profiles into clusters
  – instances in the same cluster are highly similar to each other
  – instances from different clusters have low similarity to each other
Motivation for Clustering

• exploratory data analysis
  – understanding general characteristics of data
  – visualizing data

• generalization
  – infer something about an instance (e.g. a gene) based on how it relates to other instances

• everyone else is doing it
The Clustering Landscape

- there are many different clustering algorithms
- they differ along several dimensions
  - hierarchical vs. partitional (flat)
  - hard (no uncertainty about which instances belong to a cluster) vs. soft clusters
  - disjunctive (an instance can belong to multiple clusters) vs. conjunctive/non-disjunctive
  - deterministic (same clusters produced every time for a given data set) vs. stochastic (some randomness)
  - distance/similarity measure used
Distance/Similarity Measures

- many clustering methods employ a distance (or similarity) measure to assess the distance between
  - a pair of instances
  - a cluster and an instance
  - a pair of clusters
- given a distance value, it is straightforward to convert it into a similarity value
  \[
  \text{sim}(x, y) = \frac{1}{1 + \text{dist}(x, y)}
  \]
- not necessarily straightforward to go the other way
- we’ll describe our algorithms in terms of distances
Distance Metrics

• properties of metrics
  \( \text{dist}(x_i, x_j) \geq 0 \)
  \( \text{dist}(x_i, x_i) = 0 \)
  \( \text{dist}(x_i, x_j) = \text{dist}(x_j, x_i) \)
  \( \text{dist}(x_i, x_j) \leq \text{dist}(x_i, x_k) + \text{dist}(x_k, x_j) \)

• some distance metrics
  
  **Manhattan**
  \[
  \text{dist}(x_i, x_j) = \sum_e \left| x_{i,e} - x_{j,e} \right|
  \]

  **Euclidean**
  \[
  \text{dist}(x_i, x_j) = \sqrt{\sum_e \left( x_{i,e} - x_{j,e} \right)^2}
  \]

  \( e \) ranges over the individual measurements for \( x_i \) and \( x_j \)
Hierarchical Clustering: A Dendrogram

leaves represent instances (e.g. genes)

height of bar indicates degree of distance within cluster
Hierarchical Clustering of Expression Data
Hierarchical Clustering

• can do top-down (divisive) or bottom-up (agglomerative)
• in either case, we maintain a matrix of distance (or similarity) scores for all pairs of
  – instances
  – clusters (formed so far)
  – instances and clusters
Bottom-Up Hierarchical Clustering

given: a set \( X = \{x_1,...,x_n\} \) of instances

for \( i := 1 \) to \( n \) do

\[ c_i := \{x_i\} \]  \hspace{1cm} // each object is initially its own cluster, and a leaf in tree

\[ C := \{c_1,...,c_n\} \]

\( j := n \)

while \( |C| > 1 \)

\( j := j + 1 \)

\[ (c_a, c_b) := \arg\min_{(c_u,c_v)} \text{dist}(c_u,c_v) \]  \hspace{1cm} // find least distant pair in \( C \)

\[ c_j = c_a \cup c_b \]  \hspace{1cm} // create a new cluster for pair

add a new node \( j \) to the tree joining \( a \) and \( b \)

\[ C := C - \{c_a, c_b\} \cup \{c_j\} \]

return tree with root node \( j \)
Hierarchical Clustering: Example
Distance Between Two Clusters

- the distance between two clusters can be determined in several ways
  - *single link*: distance of two most similar instances
    \[
    \text{dist}(c_u, c_v) = \min \left\{ \text{dist}(a, b) \mid a \in c_u, b \in c_v \right\}
    \]
  - *complete link*: distance of two least similar instances
    \[
    \text{dist}(c_u, c_v) = \max \left\{ \text{dist}(a, b) \mid a \in c_u, b \in c_v \right\}
    \]
  - *average link*: average distance between instances
    \[
    \text{dist}(c_u, c_v) = \frac{1}{n} \sum_{a \in c_u} \sum_{b \in c_v} \text{dist}(a, b)
    \]
Cluster Distances Illustrated

- **single link**
- **complete link**
- **average link**
Partitional Clustering

• divide instances into disjoint clusters
  – flat groupings vs. tree structure

• key issues
  – how many clusters should there be?
  – how should clusters be represented?
Partitional Clustering Example

Mouse malic enzyme isomerase
Mouse anticoagulant protein C gene
Human EGF-33 protein
Rat MCG57
Mouse NADH-ubiquinone oxidoreductase 13 kDa IP subunit (ub13)
Mouse caseinase
Hamster integral membrane protein CII-3
Mouse alpha-catenin
Mouse polyclonal globulin receptor (plgR)
Mouse phenylethanol form sulfotransferase (M-STPL) gene
Mouse caseinase
Mouse alcohol dehydrogenase 1 (ADH-1)
Mouse antithrombin III
Mouse LIM-kinesin 1 (Limk1) gene
Human unknown clone
Rat beta-alanine synthase
Human UDP-glucosyltransferase 2 family, polypeptide 510 (UGT2B10)
Human mitochondrial H-protein (HMPH)
Mouse 18 subunit of coagulation factor VIII
Mouse glutathione S-transferase, GST-T1 (thromboplast)
Mouse short-chain dehydrogenase ERAD2
Hamster Mkp17
Mouse major histocompatibility locus class III region
Human c-myc inducible gene 21 (mig-2)

Mouse 17-beta-hydroxysteroid dehydrogenase type II
Mouse ketone
Mouse CYP1A3
Mouse NADPH-cytochrome P450 oxidoreductase
Rat ub10 ubiquitin-conjugating enzyme (E2) 76K
Mouse UDP-glucose dehydrogenase (LydH)
Mouse CYP1A1
Mouse major histocompatibility locus
Human unknown clone
Rat unknown protein E37
Mouse CYP1A2
Rat liver fructose-1,6-bisphosphatase
Mouse catechohol-O-methyltransferase
Mouse CYP1A2
Mouse CYP1A2
Mouse HPD-0 gene
Mouse CYP1A2
Rat riboflavin II
Mouse CYP1A2
Partitional Clustering from a Hierarchical Clustering

• we can always generate a partitional clustering from a hierarchical clustering by “cutting” the tree at some level
$K$-Means Clustering

- assume our instances are represented by vectors of real values
- put $k$ cluster centers in same space as instances
- each cluster is represented by a vector $\vec{f}_j$
- consider an example in which our vectors have 2 dimensions
K-Means Clustering

- each iteration involves two steps
  - assignment of instances to clusters
  - re-computation of the means
$K$-Means Clustering: Updating the Means

• for a set of instances that have been assigned to a cluster $c_j$, we re-compute the mean of the cluster as follows:

$$\mu(c_j) = \frac{\sum_{\tilde{x}_i \in c_j} \tilde{x}_i}{|c_j|}$$
**K-Means Clustering**

given: a set $X = \{\vec{x}_1 ... \vec{x}_n\}$ of instances

select $k$ initial cluster centers $\vec{f}_1 ... \vec{f}_k$

while stopping criterion not true do
  for all clusters $c_j$ do
    
    // determine which instances are assigned to this cluster
    
    $$c_j = \left\{ \vec{x}_i \mid \forall f_l \text{ dist}(\vec{x}_i, \vec{f}_j) < \text{dist}(\vec{x}_i, \vec{f}_l) \right\}$$

    for all means $\vec{f}_j$ do
      
      // update the cluster center
      
      $$\vec{f}_j = \mu(c_j)$$
$K$-means Clustering Example

Given the following 4 instances and 2 clusters initialized as shown. Assume the distance function is $\text{dist}(x_i, x_j) = \sum_{e} |x_{i,e} - x_{j,e}|$

\[
\begin{align*}
\text{dist}(x_1, f_1) &= 2, \quad \text{dist}(x_1, f_2) = 5 \\
\text{dist}(x_2, f_1) &= 2, \quad \text{dist}(x_2, f_2) = 3 \\
\text{dist}(x_3, f_1) &= 3, \quad \text{dist}(x_3, f_2) = 2 \\
\text{dist}(x_4, f_1) &= 11, \quad \text{dist}(x_4, f_2) = 6
\end{align*}
\]

\[
\begin{align*}
f_1 &= \left\langle \frac{4+4}{2}, \frac{1+3}{2} \right\rangle = \langle 4,2 \rangle \\
f_2 &= \left\langle \frac{6+8}{2}, \frac{2+8}{2} \right\rangle = \langle 7,5 \rangle
\end{align*}
\]

\[
\begin{align*}
\text{dist}(x_1, f_1) &= 1, \quad \text{dist}(x_1, f_2) = 7 \\
\text{dist}(x_2, f_1) &= 1, \quad \text{dist}(x_2, f_2) = 5 \\
\text{dist}(x_3, f_1) &= 2, \quad \text{dist}(x_3, f_2) = 4 \\
\text{dist}(x_4, f_1) &= 10, \quad \text{dist}(x_4, f_2) = 4
\end{align*}
\]
assignments remain the same, so the procedure has converged

\[ f_1 = \left( \frac{4 + 4 + 6}{3}, \frac{1 + 3 + 2}{3} \right) = \langle 4.67, 2 \rangle \]

\[ f_2 = \left( \frac{8}{1}, \frac{8}{1} \right) = \langle 8, 8 \rangle \]
EM Clustering

• in $k$-means as just described, instances are assigned to one and only one cluster
• we can do “soft” $k$-means clustering via an expectation maximization (EM) algorithm
  – each cluster represented by a distribution (e.g. a Gaussian)
  – $E$-step: determine how likely is it that each cluster “generated” each instance
  – $M$-step: adjust cluster parameters to maximize likelihood of instances
Representation of Clusters

• in the EM approach, we’ll represent each cluster using an \( m \)-dimensional multivariate Gaussian

\[
N_j(\tilde{x}_i) = \frac{1}{\sqrt{(2\pi)^m |\Sigma_j|}} \exp\left[-\frac{1}{2}(\tilde{x}_i - \tilde{\mu}_j)^T \Sigma_j^{-1} (\tilde{x}_i - \tilde{\mu}_j)\right]
\]

where

\( \tilde{\mu}_j \) is the mean of the Gaussian

\( \Sigma_j \) is the covariance matrix

this is a representation of a Gaussian in a 2-D space
EM Clustering

- the EM algorithm will try to set the parameters of the Gaussians, \( \Theta \), to maximize the log likelihood of the data, \( X \)

\[
\text{log likelihood}(X \mid \Theta) = \log \prod_{i=1}^{n} \Pr(\vec{x}_i)
\]

\[
= \log \prod_{i=1}^{n} \sum_{j=1}^{k} N_j(\vec{x}_i)
\]

\[
= \sum_{i=1}^{n} \log \sum_{j=1}^{k} N_j(\vec{x}_i)
\]
EM Clustering

• the parameters of the model, \( \Theta \), include the means, the covariance matrix and sometimes prior weights for each Gaussian

• here, we’ll assume that the covariance matrix and the prior weights are fixed; we’ll focus just on setting the means
EM Clustering: Hidden Variables

• on each iteration of \textit{k-means} clustering, we had to assign each instance to a cluster

• in the EM approach, we’ll use \textit{hidden variables} to represent this idea
  – similar idea to the “hidden” state in HMMs

• for each instance $\vec{x}_i$, we have a set of hidden variables $z_{i1}, \ldots, z_{ik}$

• we can think of $z_{ij}$ as being 1 if $\vec{x}_i$ is a member of cluster $j$ and 0 otherwise
EM Clustering: the E-step

• recall that $z_{ij}$ is a hidden variable which is 1 if $N_j$ generated $\vec{x}_i$ and 0 otherwise

• in the E-step, we compute $h_{ij}$, the expected value of this hidden variable

$$h_{ij} = E(z_{ij} | \vec{x}_i) = \frac{N_j(\vec{x}_i)}{\sum_{l=1}^{k} N_l(\vec{x}_i)}$$
EM Clustering: the M-step

- given the expected values $h_{ij}$, we re-estimate the means of the Gaussians

$$\mu_j = \frac{\sum_{i=1}^{i} h_{ij} \bar{x}}{\sum_{i=1}^{i} h_{ij}}$$

- can also re-estimate the covariance matrix and prior weights, if we’re varying them
EM Clustering Example

Consider a one-dimensional clustering problem in which the data given are:

\[ x_1 = -4 \]
\[ x_2 = -3 \]
\[ x_3 = -1 \]
\[ x_4 = 3 \]
\[ x_5 = 5 \]

The initial mean of the first Gaussian is 0 and the initial mean of the second is 2. The Gaussians have fixed width; their density function is:

\[
f(x, \mu) = \frac{1}{\sqrt{8\pi}} e^{-\frac{1}{2} \left( \frac{x-\mu}{2} \right)^2}
\]

where \( \mu \) denotes the mean (center) of the Gaussian.
EM Clustering Example

\[ \begin{align*}
\mu_1 &= 0 \\
\mu_2 &= 2 \\
\mu_1 &= -1.94 \\
\mu_2 &= 3.39
\end{align*} \]

- continue the E-steps and M-steps until convergence
EM and $K$-Means Clustering

- both are sensitive to initial positions (means) of clusters
  - one solution: run the algorithm many times with different initial positions, keep the clustering that looks best

- have to choose value of $k$ for both
  - one solution: run the algorithm with many different values of $k$ and select the $k$ that looks best
  - more specifically, use cross-validation
Cross Validation to Select $k$

- with EM clustering, we can run the method with different values of $k$, use CV to evaluate each clustering

\[
\sum_i \log \text{Pr}(x_i)
\]

to evaluate clustering

- then run method on all data once we’ve picked $k$
Evaluating Clustering Results

• given random data without any “structure”, clustering algorithms will still return clusters

• the gold standard: do clusters correspond to natural categories?

• do clusters correspond to categories we care about? (there are lots of ways to partition the world)
Evaluating Clustering Results

• external validation
  – e.g. do genes clustered together have some common function?

• internal validation
  – how well does clustering optimize intra-cluster similarity and inter-cluster dissimilarity?

• relative validation
  – how does it compare to other clusterings using these criteria?
  – e.g. with a probabilistic method (such as EM) we can ask: how probable does held-aside data look as we vary the number of clusters.
Comments on Clustering

• there many different ways to do clustering; we’ve discussed just a few methods

• hierarchical clusters may be more informative, but they’re more expensive to compute

• clusterings are hard to evaluate in many cases
Computational Tasks

• **differential expression**
  – which genes have different expression levels across two groups?

• **clustering**
  – which genes seem to be regulated together?
  – which treatments/individuals have similar expression profiles?

• **classification**
  – to which functional class does a given gene belong?
  – to which class does a given sample belong? (e.g., does this patient have ALL or AML?)
Classifying Gene Expression Profiles

• **given:**
  – expression profiles for a set of genes or experiments/individuals/time points (whatever columns represent)
  – a “class label” (e.g. disease vs. healthy) for each profile

• **do:**
  – learn a model that can accurately predict labels for new gene expression profiles
Breast Cancer Outcomes Prediction

- Nevins et al., *Human Molecular Genetics*, 2003

- microarray and clinical data from 86 lymph-node positive breast cancer patients
  - 12,625 genes measured using Affymetrix arrays

- goal is to distinguish between high risk (recurrence within 5 years) and low risk (recurrence-free for 5 years)
Calculating “Metagenes”

- the features used in their model are not mRNA measurements from individual genes
- instead they compute “metagenes”, which consist of linear combinations of gene measurements
- procedure
  - ran $k$-means clustering (with $k=500$) on original microarray data set
  - computed first principal component of each cluster (i.e. dominant expression profile characteristics)
  - each of these principal components becomes a metagene
A Decision Tree Classifier

- low risk/high risk cases in training set that reach this node
- smoothed probability estimate of high risk
- outcome of test at internal node above
Decision Tree Classifiers

- tree-based classifiers partition the data using axis-parallel splits
Inducing Tree-Based Classifiers

• there are many decision-tree learning methods
• two most common are
  – C4.5 (Quinlan)
  – CART (Breiman, Friedman, Olshen, Stone)
• Nevins et al. use their own method
• all DT learning methods have the same basic algorithm structure: recursively grow a tree top-down
Breast Cancer Outcomes Prediction

- predictive accuracy estimated by cross-validation
  - 85-90% correct predictions (low-risk, high-risk)

- *bonus*: decision trees give us insight into which types of genes (as characterized by metagenes) play a key role in breast cancer, and how they might interact
MM vs. MGUS Classification

- Hardin et al., *Statistical Applications in Genetics and Mol. Bio.*, 2004

- Standard lab classification of MM (multiple myeloma) and MGUS (benign skin tumors) is quite accurate… but biologically uninformative

- Can this classification be done using expression profiles?

- Learned models might
  - Enable molecular diagnosis
  - Lend insight into disease progression

- ~12,625 genes measured on Affymetrix (oligonucleotide) microarrays
Results: Cancer vs. Normal

ROC for MM vs. Normal

sensitivity (% MM classified as MM)

1-specificity (% Normal classified as MM)

logistic-best
logistic-for
SVM
EOV
Naive Bayes
NSC
Results: Benign vs. Normal

ROC for MGUS vs. Normal

sensitivity (% MGUS classified as MGUS)

1-specificity (% Normal classified as MGUS)
Results: Cancer vs. Benign

ROC for MM vs. MGUS

sensitivity (% MM classified as MM)

1-specificity (% MGUS classified as MM)

logistic-best
logistic-for
SVM
EOV
Naive Bayes
NSC
we discussed two computational tasks
  – clustering: when you don’t know the categories of interest
  – classification: when you do know the categories of interest

_class discovery_ is an interesting task that falls between classification and clustering
  – identify classes of profiles that don’t seem to fit into any of the modeled categories
  – e.g. new subtypes of cancer, new types of toxic substances

we’ve discussed methods in the context of _microarray_ data, but they can also be applied to a wide variety of high-throughput biological data
Other High-Throughput Methods

- varied: individuals, strains, cell types, environmental conditions, disease states, etc.
- measured: RNA quantities
- technology: microarrays

- varied: same as above
- measured: protein or small molecule quantities
- technology: 2D gel electrophoresis + mass spec

- varied: individuals
- measured: variation at specific genome locations
- technology: SNP chips, etc.
Single-Nucleotide Polymorphisms

- SNPs: individual positions in an organism’s DNA where variation is common
  - these variations often constitute our phenotypic differences, e.g., eye or hair color
- roughly 2 million known SNPs in humans
- new Affymetrix chips can scan whole-genomes for 500,000 of these!
- easier, faster, cheaper, and more effective to measure SNPs than to completely sequence genomes for everyone
Genome-Wide Association Studies

• gather a population of individuals
  – perhaps some with a disease, some without

• genotype each individual at polymorphic markers
  – such as SNPs

• test for statistical correlation between markers and a variable of interest
  – e.g. how well does this marker “predict” the disease
Wellcome Trust GWAS

~2000 individuals each for 7 diseases

shared set of ~3000 controls (i.e. healthy)

Affymetrix SNP chips
Next Time…

• basic molecular biology
• sequence alignment
• probabilistic sequence models
• gene expression analysis
• protein structure prediction
  – by Ameet Soni