Gene Expression Analysis

Selected Topics in Computer Intelligence - 2015

Bioinformatics Programming

Computer Engineering, Chiang Mai University

Gene Expression Microarrays

- Commonly called the Gene Chip
- Make it possible to simultaneously measure the rate at which a cell or tissue is expressing each of its genes
  - There are thousands of genes in a single cell or a tissue at a single time point
- Microarrays can be used to …
  - Snapshot the biological activity to infer regulatory pathways
  - Identify novel targets for drug design
  - Improve the diagnosis, prognosis, and treatment planning
  - Help analyzing novel gene functions which cannot be strongly identify from sequence comparison

Background

- Almost every cell in the body has the same DNA
  - Genes are portions of the DNA that code for proteins
- A gene is expressed through a two-step process
  - Aka. Central dogma
  - First, a gene is transcribed into RNA
  - RNA is then translated into the corresponding protein
- Gene-expression microarrays allows us to monitor the DNA-to-RNA portion of this biological process
  - Ability to measure the transcription of all the genes in an organism at once – overwhelming data
  - A dataset can consist of roughly 100 samples, each contains about expression of 10,000 genes
  - Multidimensional data

Background

- Finding some combinations of genes whose expression levels can distinguish groups of data is heavy task
  - e.g., groups of patients who DO and DO NOT have disease
- There are many tasks that require analyzing microarray data and many ways to apply machine learning

Background

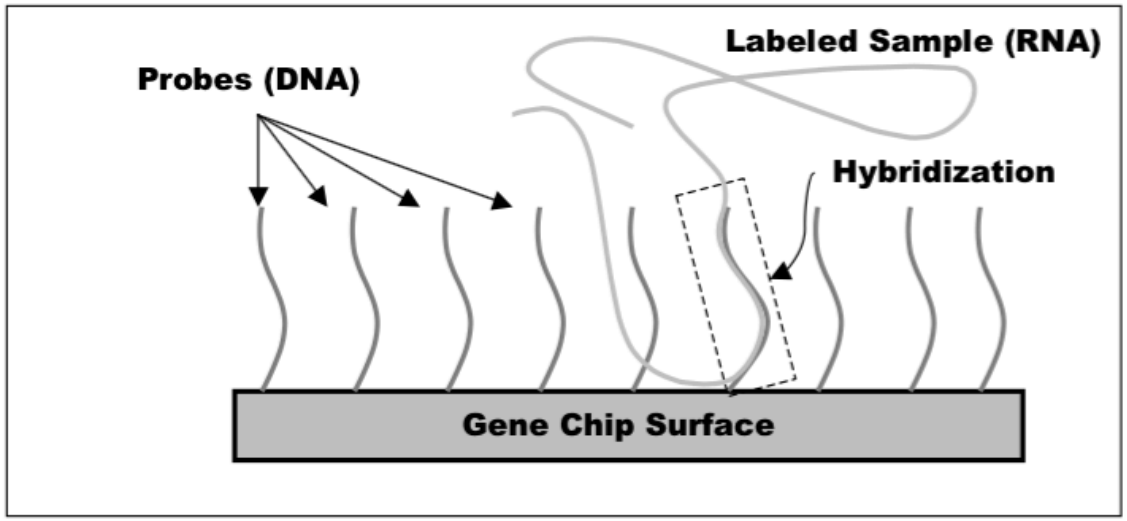
- Complementarity is central to the double-stranded structure of DNA and the process of DNA replication
- Biologists have taken advantage of this to detect specific sequences of base within strands of DNA & RNA
  - First synthesizing a probe
    - a piece of DNA that is the reverse complement of a sequence one wants to detect – put them on microarray
  - Introducing this probe to a solution containing DNA or RNA to be search – sample
  - The probe will bind to the sample if it finds its complement in the sample – binding sites

Background

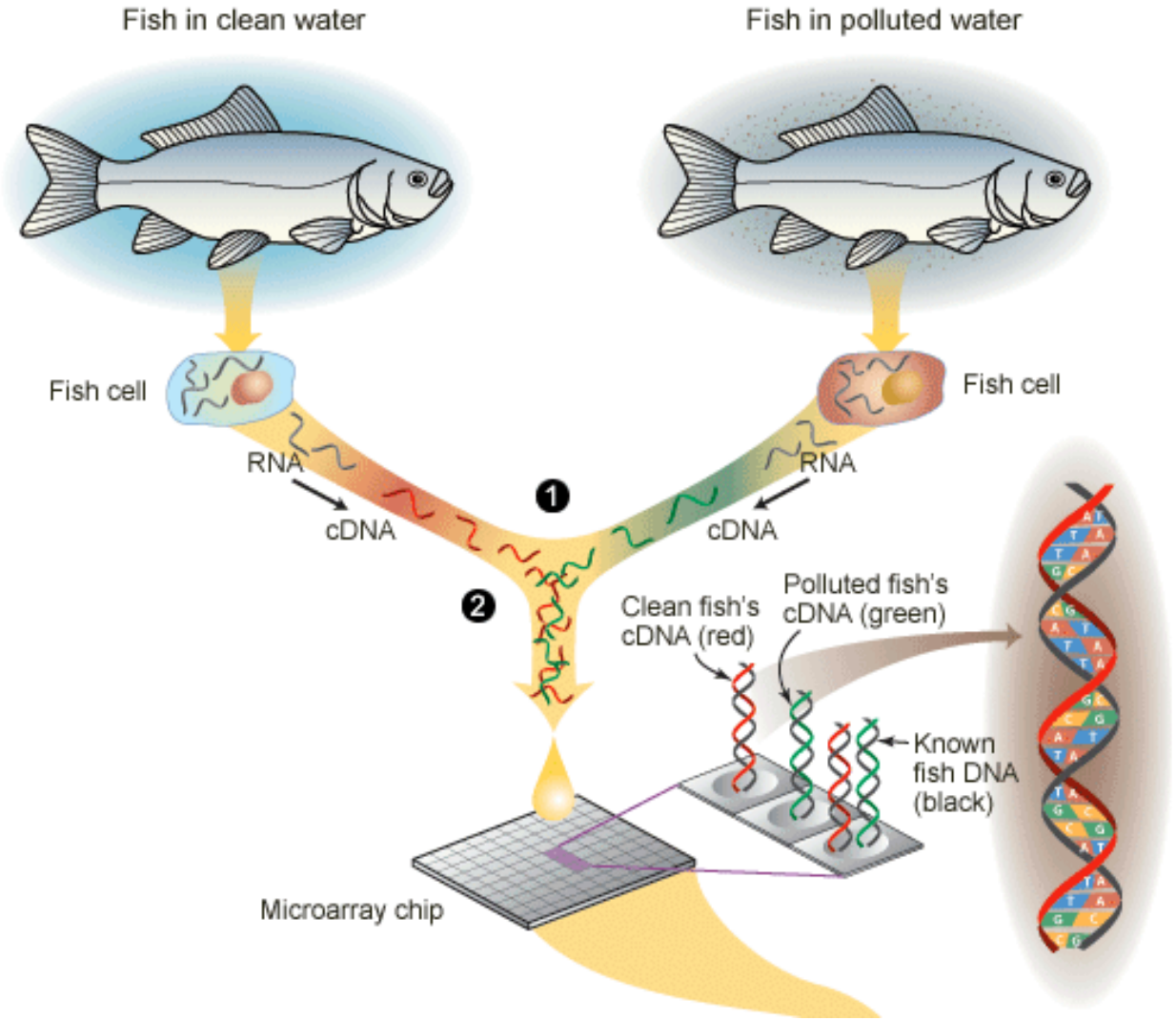
- Binding does not always happen in practice
- The act of binding between probe and sample is called hybridization
  - We can labels the probes using a fluorescent tag
  - After the hybridization experiment, we can determine the presence or absence of the sequence-of-interest in the sample

What are Gene Chips?

- DNA probe technology has been adapted for detection tens of thousands sequences simultaneously
  - Synthesizing a large number of different probes (~25-bases)
    - **Oglionucleotide**: short DNA or RNA molecules
  - Placing each probe at a specific position on some surface
  - RNA samples are about 10 times as long as the probe

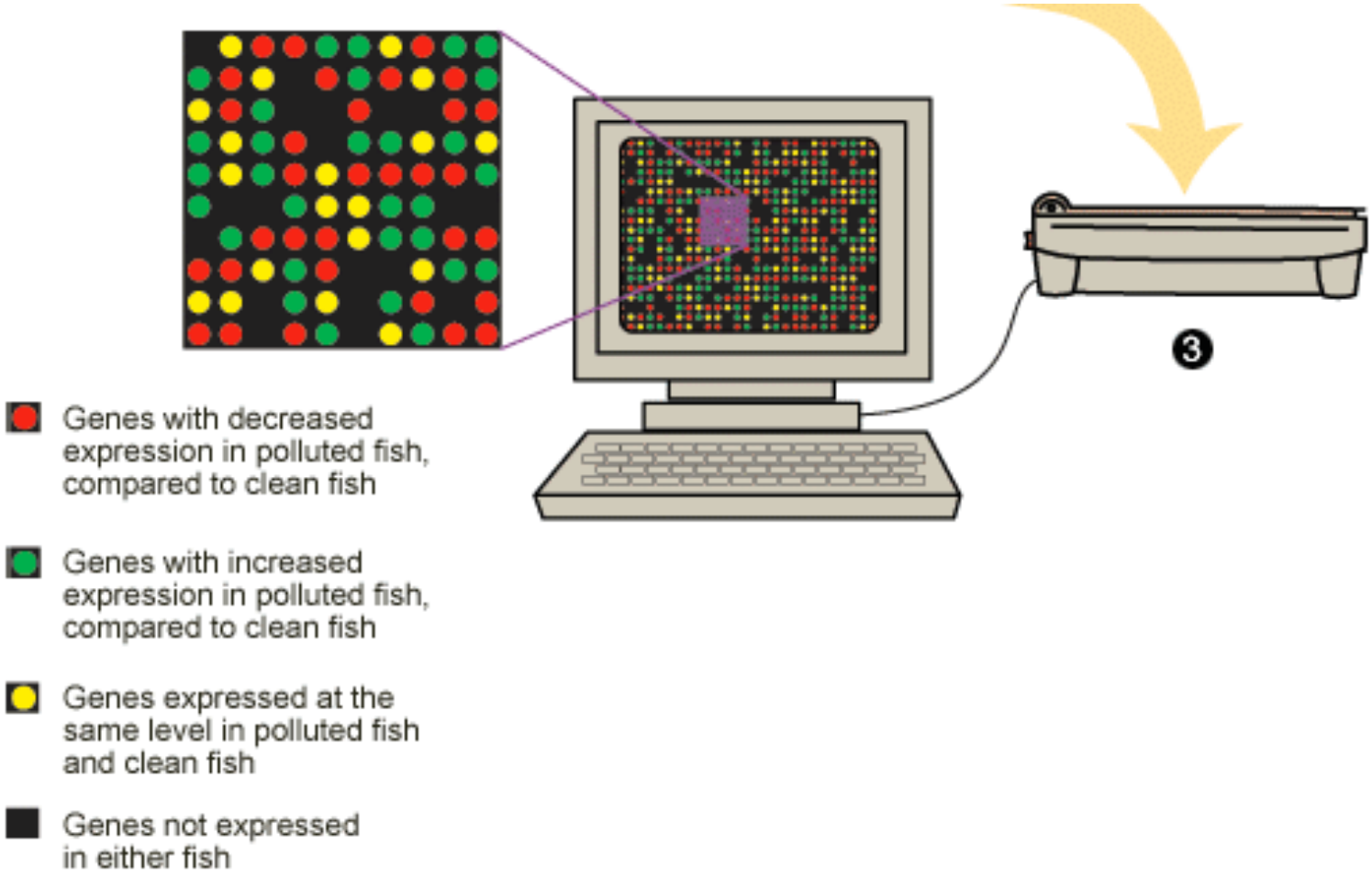


How gene chips work?



http://www.whoi.edu/services/communications/oceanusmag.050826/v43n2/hahn.html

How gene chips work?

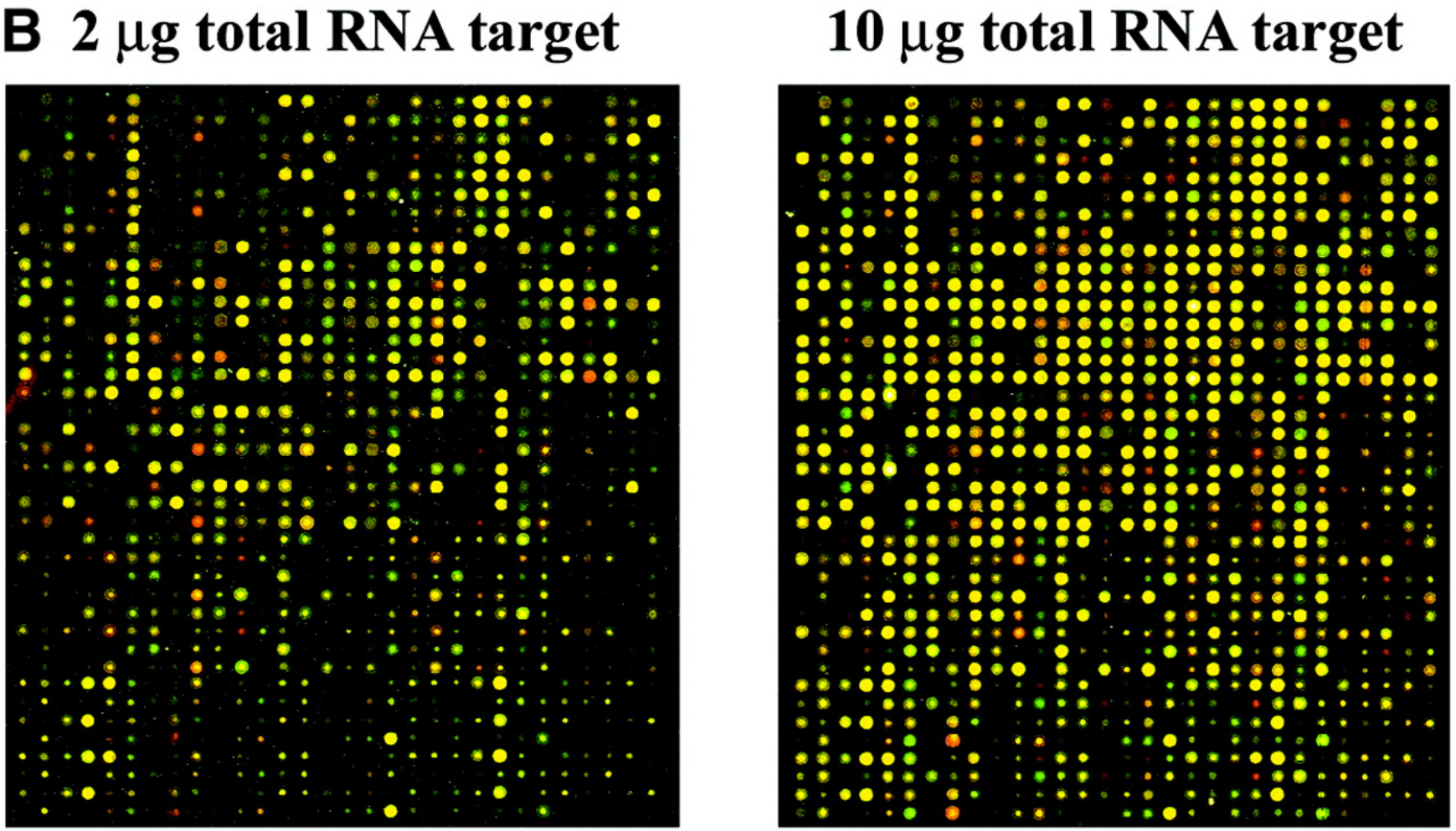


http://www.whoi.edu/services/communications/oceanusmag.050826/v43n2/hahn.html

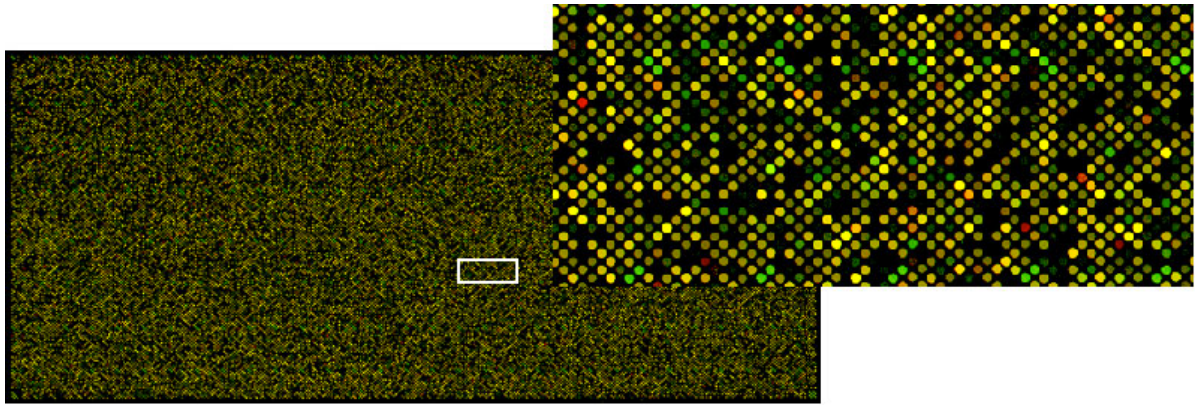
Data Collection and Preprocessing

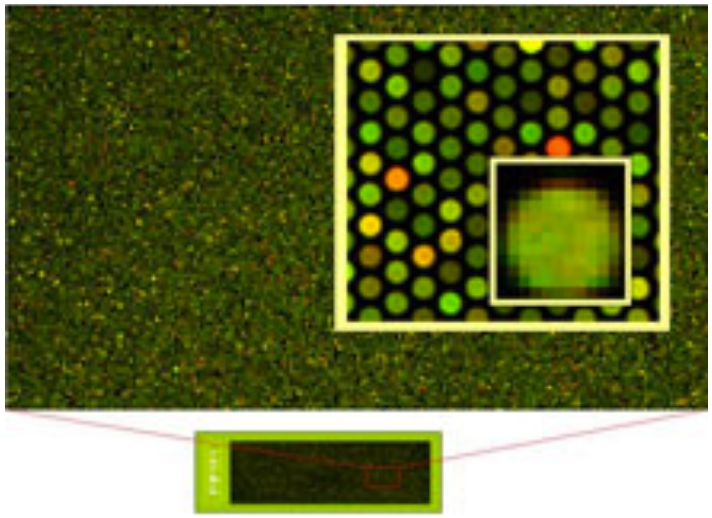
- An optical scanner is used to records the fluorescence intensity values at each spot on the gene chip
- In case of gene-expression arrays
  - There will be many experiments measuring the same set of genes under various circumstances:
    - **Various Conditions**: when cell is heated up or cool down, when some drug is added, …
    - **Various time points**: 5, 10, … after adding an antibiotic

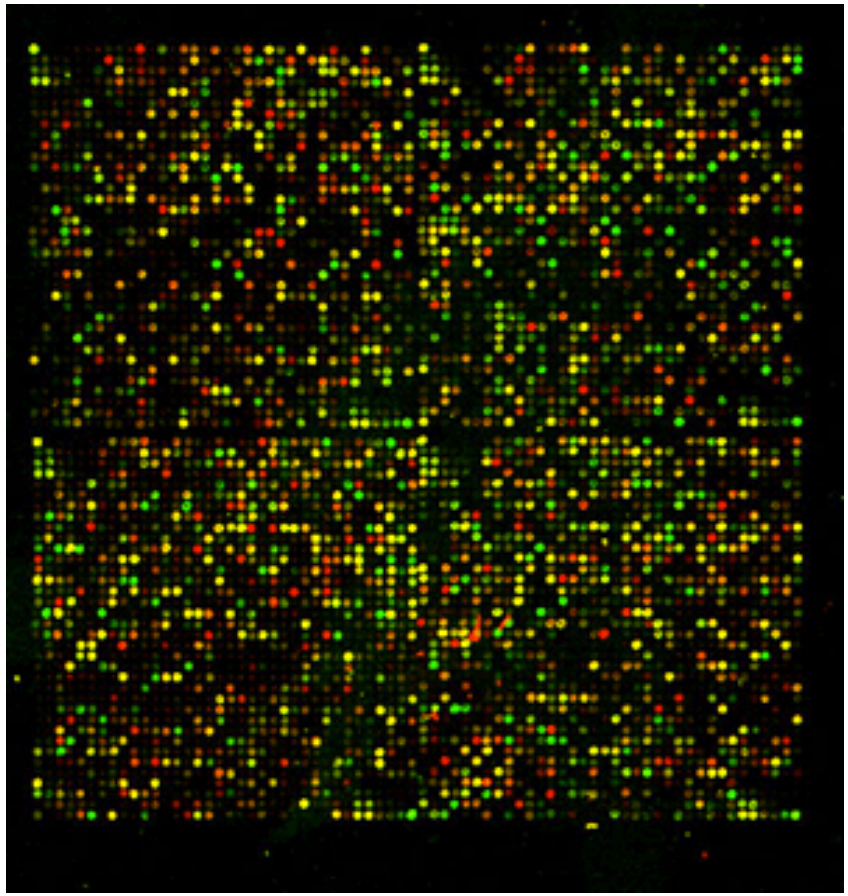
Data Collection and Preprocessing



Data Collection and Preprocessing







Data Collection and Preprocessing

- Gene view
  - Expression levels under different conditions – features

Features →

←Examples	Experiment 1	Experiment 2	…	Experiment *N*
Gene 1	1083	1464	…	1115
Gene 2	1585	398	…	511
…	…	…	…	…
Gene *M*	170	302	…	751

  - Examples can be labeled according to some category of interest: normal cells and cancerous cells

Data Collection and Preprocessing

- Experiment view
  - The features are the expression values for all the genes

Features →

←Examples	Gene 1	Gene 2	…	Gene *M*
Experiment 1	1083	1585	…	170
Experiment 2	1464	398	…	302
…	…	…	…	…
Experiment *N*	1115	511	…	751

Data Collection and Preprocessing

- Genes are on the order of a 1000 bases long
- Probes on gene chips are typically on the order of 25 bases long
  - Most probes do not hybridize to their sample as we would like
  - Partially hybridize to other sample is possible
  - The sample might fold up and hybridize to itself
- Microarrays typically use about a dozen of probes for each gene
  - An algorithm combines the measured fluorescence levels for each probe in this set
  - then estimate the expression level for the associated gene

Data Collection and Preprocessing

- The raw signal values typically contain a lot of noise
  - Synthesis of probes
  - Creation and labeling of samples
  - Reading of the fluorescent signals
- Replication of each experiment is often required but in a very small number of times (~100 USD for each chip)

Design of Microarrays

- If we have a better way of picking good probes:
  - We can use fewer probes per gene, thereby more genes can be tested per microarray
  - We can get more accurate results
  - Machine learning have been used to address the task
- Create training set for machine learning system
  - Place all possible probes for a given set of genes on a microarray
  - Which probes produce strong fluorescence levels when the gene's RNA is applied to the gene chip
    - If the probes all hybridized equally, there would be a uniformly high signal across the entire chip – NOT the case

Gene Expression Analysis

- Sequence comparison often helps to discover the function of a newly sequenced gene - similarity
- For many genes:
  - Sequence similarity of genes in the same functional family is weak
  - Genes with the same function sometimes have no sequence similarity at all
- The functions of more than 40% of the genes in sequenced genomes are still unknown

Gene Expression Analysis

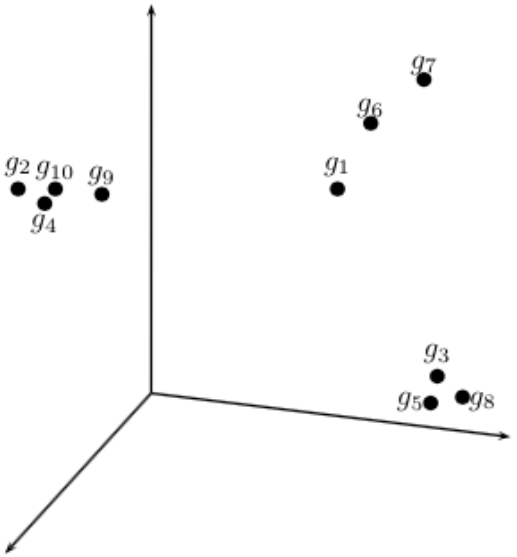
- The outcome from microarrays are usually in the form of
  - n x m *expression matrix* I
    - n-rows corresponding to genes
    - m-columns corresponding to different conditions and time points
  - $\mathrm{I}_{i,j}$ – the expression level of gene i in experiment j
  - i^{th} row is called the expression pattern of gene i
    - Pairs of genes with similar expression pattern have similar rows
- If the expression pattern of two genes are similar
  - There is a good chance that these genes are somehow related
  - i.e., perform similar function or involved in the same process

Gene Expression Analysis

- Clustering algorithms
  - Group genes with similar expression pattern into clusters
  - These clusters correspond to groups of functionally related genes
- To cluster the expression data, the expression matrix is transformed into an n x n distance matrix - d
  - $d_{i,j}$ – how similar the expression patterns of genes i and j are
- Goal of clustering – clusters should satisfy two conditions
  - Homogeneity : high intra-cluster (behavior) similarity,
    $d_{i,j}$ should be small if i and j belong to the same cluster
  - Separation : how inter-cluster (behavior) similarity
    $d_{i,j}$ should be large if i and j belong to different cluster

Gene Expression Analysis

- Example



Time	1 hr	2 hr	3 hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

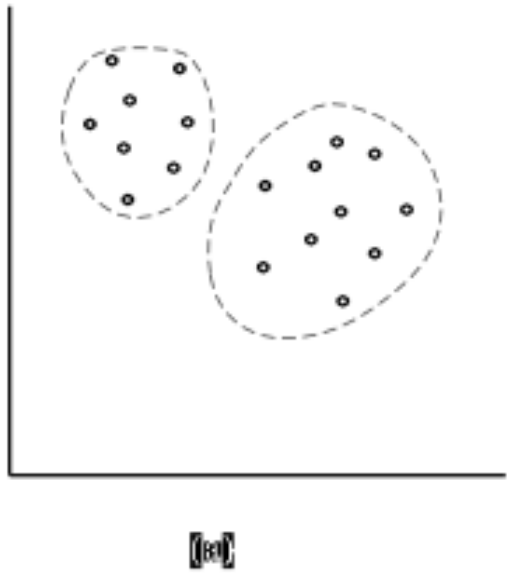
(a) Intensity matrix, $\mathbf{I}$

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

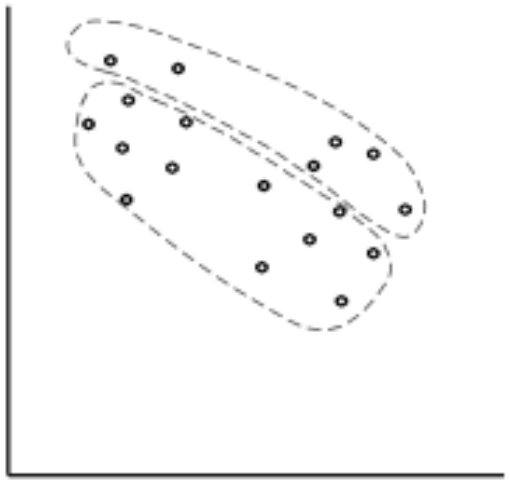
(b) Distance matrix, $\mathbf{d}$

Gene Expression Analysis

- A good clustering is the one that stick to the goals
  - Better clustering of genes gives rise to a better grouping of genes on a functional level
  - There is no such algorithm that performs well on every dataset



(a)

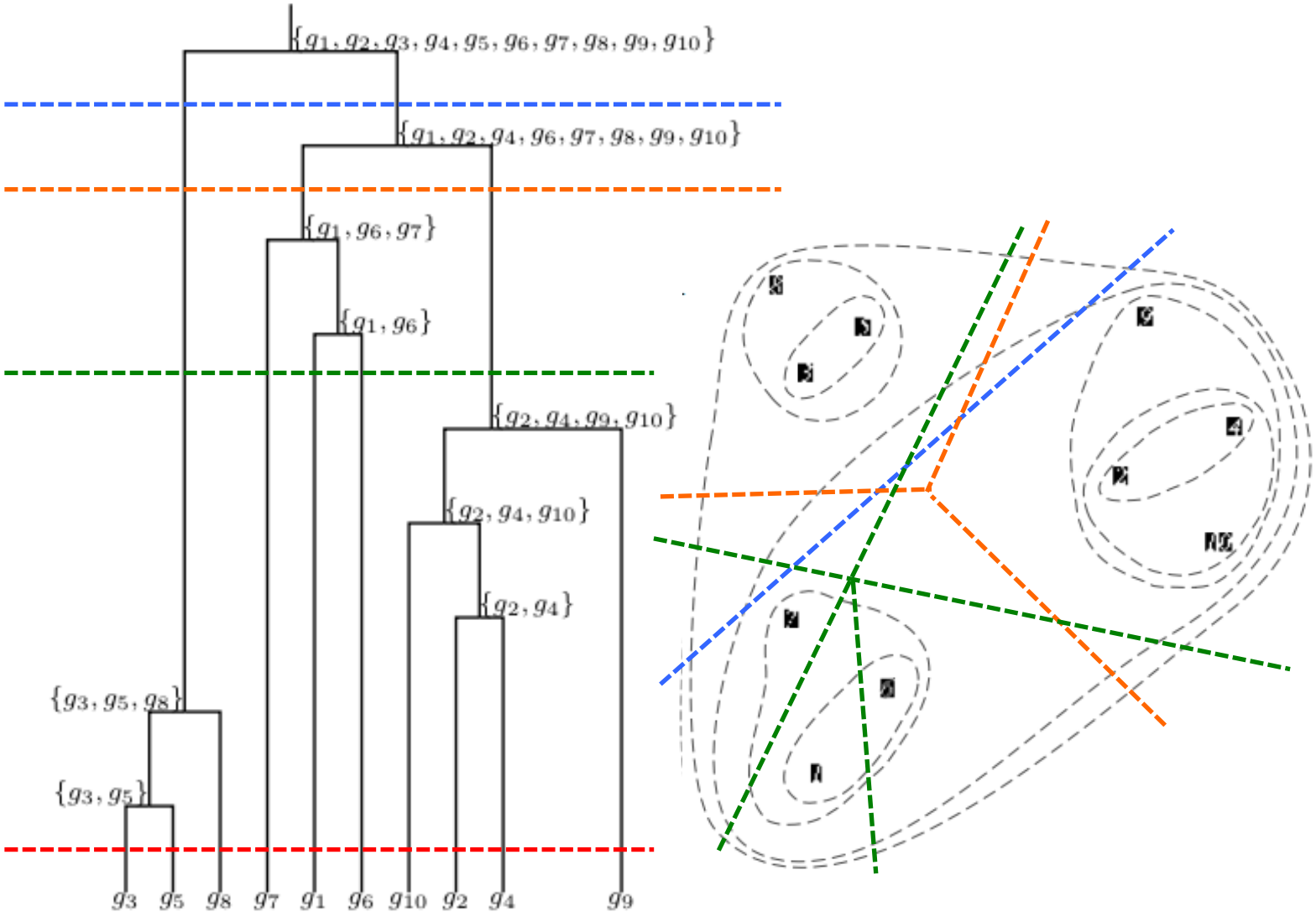


(b)

Hierarchical Clustering

- In many cases clusters have subclusters, and so on…
- This technique organizes elements into a tree
  - Genes are represented as the leaves of a tree
  - Edges of the trees are assigned lengths and distances between to leaves – *correlate with entries in the distance matrix*
- The tree actually describes a family of different partitions
  - Each with a different number of clusters: from 1 to n
  - We can see them by drawing a horizontal line through the tree
    - Each line crosses the tree at i point $(1 \leq i \leq k) \Rightarrow i$ clusters

Hierarchical Clustering



Hierarchical Clustering

- HIERARCHICALCLUSTERING Algorithm

```
HIERARCHICALCLUSTERING(d, n)
 1  Form n clusters, each with 1 element
 2  Construct a graph T by assigning an isolated vertex to each cluster
 3  while there is more than 1 cluster
 4      Find the two closest clusters C1 and C2
 5      Merge C1 and C2 into new cluster C with |C1| + |C2| elements
 6      Compute distance from C to all other clusters
 7      Add a new vertex C to T and connect to vertices C1 and C2
 8      Remove rows and columns of d corresponding to C1 and C2
 9      Add a row and column to d for the new cluster C
10  return T
```

- The largest partition has ***n*** single-element clusters
  - Every element forming its own cluster – n clusters
- The 2nd largest partition
  - Combines the two closest cluster from the largest partition
  - n-1 clusters

Hierarchical Clustering

- HIERARCHICALCLUSTERING Algorithm
  - How to compute distance from the new cluster, C, to all other clusters
    - Clustering algorithms compute these distances differently
    - Yield different answers from the same hierarchical clustering algorithm, for examples, …
    - Smallest distance between any pair of their elements

$$d_{min}(C^*, C) = \min_{x \in C^*, y \in C} d(x, y)$$

    - The average distance between their elements

$$d_{avg}(C^*, C) = \frac{1}{|C^*||C|} \sum_{x \in C^*, y \in C} d(x, y).$$

k-Means Clustering

- One of the most popular clustering methods for points in multidimensional spaces
- n x m expression matrix can be view as …
  - A set of n points in m dimensional space
  - and … partition them into k subsets (k is know in advanced)
- **Minimize** the squared error distortion for a set of n points $\mathcal{V}=\{v_1, \ldots v_n\}$ and a set of k centers $X=\{x_1, \ldots x_k\}$ is …

$$d(\mathcal{V}, \mathcal{X}) = \frac{\sum_{i=1}^{n} d(v_i, \mathcal{X})^2}{n}$$

k-Means Clustering

k-Means Clustering Problem:
Given n data points, find k center points minimizing the squared error distortion.

Input: A set, $\mathcal{V}$, consisting of n points and a parameter k.

Output: A set $\mathcal{X}$ consisting of k points (called centers) that minimizes $d(\mathcal{V}, \mathcal{X})$ over all possible choices of $\mathcal{X}$.

- After knowing k centers
  - We can simply assigning each points to its closest center, x_i
- One of the most popular clustering heuristics that often generates good solutions in GXP analysis is **Lloyd** algorithm

k-Means Clustering

- We can choose arbitrary k points as "cluster representatives"
- The algorithm iteratively performs the following two steps until … either it ***converges*** or until the ***change is very small***
  - Assign each data point to cluster C_i corresponding to the closest x_i
  - After assigning all ***n*** data points, compute new cluster representatives
    - Using the Center of Gravity (CG) of each cluster
- The **Lloyd** algorithm Often converge to local minimum
  - The clustering cost emphasize the homogeneity and ignore the separation condition
- There is a more conservative approach to move only one element between clusters in each iteration

k-Means Clustering

- Assuming every partition P has clustering cost – $cost(P)$
  - Measures the quality of the partition
  - The *smaller* the cost, the *better* that clustering is
  - Choice for $cost(P)$: The squared error distortion
- Assuming each center point is **CG** of its cluster
- $P_{i \to C}$ denotes the partition obtained from P by moving element i from its cluster to C
- $\Delta(i \to C) =$ improved clustering cost

$$= cost(P) - cost(P_{i \to C}) > 0$$

k-Means Clustering

- PROGRESSIVEGREEDYK-MEANS(k)

```
PROGRESSIVEGREEDYK-MEANS(k)
 1  Select an arbitrary partition P into k clusters.
 2  while forever
 3      bestChange ← 0
 4      for every cluster C
 5          for every element i ∉ C
 6              if moving i to cluster C reduces the clustering cost
 7                  if Δ(i → C) > bestChange
 8                      bestChange ← Δ(i → C)
 9                      i* ← i
10                      C* ← C
11      if bestChange > 0
12          change partition P by moving i* to C*
13      else
14          return P
```