CS 5984: Biclustering Algorithms for Gene Expression Analysis

T. M. Murali

February 7, 2006
Problems with Hierarchical Clustering

- It is a global clustering algorithm.
- Considers *all* genes to be equally important for all samples.
Problems with Hierarchical Clustering

- It is a global clustering algorithm.
- Considers all genes to be equally important for all samples.
- What if only a subset of the genes are co-expressed across only a subset of the samples?
- What if different subsets of the genes are co-expressed for different subsets of samples?
Example: Roberts et al. (Science 2000)
Example: Alizadeh et al. (Nature 2000)
Example: Alizadeh et al. (Nature 2000)
Biclustering

- A **bicluster** is a subset of genes and a subset of samples with the property that the selected genes are co-expressed only in the selected samples.
Biclustering

- A *bicluster* is a subset of genes and a subset of samples with the property that the selected genes are co-expressed only in the selected samples.
- By selecting samples *and* genes, a bicluster represents condition-specific patterns of expression.
- Issues in biclustering:
**Biclustering**

- A *bicluster* is a subset of genes and a subset of samples with the property that the selected genes are co-expressed only in the selected samples.
- By selecting samples *and* genes, a bicluster represents condition-specific patterns of expression.
- Issues in biclustering:
  - How do we measure the degree of co-expression of a subset of genes in a subset of samples?
  - How many biclusters should we compute?
  - How do we compare two different sets of biclusters?
History of Biclustering

- **Block clustering**: Hartigan 1972, recursively partition matrix into blocks.
- Biclustering formulated in the context of gene expression data by Cheng and Church, ISMB 2000.
- Since 2000, a number of papers have been published on biclustering.
  - Iterative signature algorithm: Bergmann, Ihmels, and Barkai, Phys Review E 2003
  - Two surveys of biclustering:
Biclustering: Cheng and Church

- Defined the score of a bicluster to be its mean squared residue.
- Developed an iterative algorithm for computing biclusters with residue less than $\delta$ (specified by the user) by addition and deletion of genes and samples.
- To find multiple biclusters, they “erase” the values in the previously-computed biclusters and continue.
Mean Squared Residue

- $A$ = matrix of gene expression values, $a_{ij}$ = value in the $i$th row and $j$th column of $A$.
- $I$ = subset of genes/rows, $J$ = subset of conditions/columns.
- $A_{IJ}$ = submatrix of $A$ containing the rows in $I$ and the columns in $J$.
- The mean squared residue of $A_{IJ}$ is

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2,$$

where

- $a_{iJ}$ = average of values in $A_{IJ}$ along row $i$, $a_{Ij}$ = average of values in $A_{IJ}$ along column $j$ and $a_{IJ}$ = average of all values in $A_{IJ}$.
Examples of Mean Squared Residue

\[ H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{IJ} - a_{Ij})^2 \]

- Constant matrix:

- Single element:

- Matrix with elements chosen randomly from the interval \([a, b]\) has expected mean squared residue \((b-a)^2/12\).
Examples of Mean Squared Residue

\[
H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} \left( a_{ij} - a_{iJ} - a_{Ij} - a_{IJ} \right)^2
\]

- Constant matrix: 0.
Examples of Mean Squared Residue

\[ H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{lJ} - a_{lJ})^2 \]

- Constant matrix: 0.
- Single element:
Examples of Mean Squared Residue

\[ H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{i.J} - a_{.j} + a_{IJ})^2 \]

- Constant matrix: 0.
- Single element: 0.
Examples of Mean Squared Residue

\[ H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{IJ} - a_{IJ})^2 \]

- Constant matrix: 0.
- Single element: 0.
- Matrix with elements chosen randomly from the interval \([a, b]\) has expected mean squared residue \((b - a)^2 / 12\).
Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$. 
### Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$.
- Why find the largest submatrix?
Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$.
- Why find the largest submatrix?
- How do we measure size of a submatrix?

Perimeter: maximise $|I| + |J|$, can be solved in polynomial time but is inappropriate when $\#\text{rows} >> \#\text{columns}$.

Square: maximise $|I| = |J|$, NP-Hard.

Area: maximise $||I|\|J||$, also NP-Hard (proven after the Cheng and Church paper).
Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$.

- Why find the largest submatrix?

- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.
Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$.
- Why find the largest submatrix?
- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.
- How hard is the problem?
Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$.
- Why find the largest submatrix?
- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $||I||J||$.
- How hard is the problem?
  - Perimeter: maximise $|I| + |J|$, can be solved in polynomial time but is inappropriate when $\# \text{rows} \gg \# \text{columns}$.
Computational Problem

- Given a matrix $A$ and a parameter $\delta$ find the largest submatrix of $A$ with mean squared residue less than $\delta$.

- Why find the largest submatrix?

- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.

- How hard is the problem?
  - Perimeter: maximise $|I| + |J|$, can be solved in polynomial time but is inappropriate when $\#rows \gg \#columns$.
  - Square: maximise $|I| = |J|$, NP-Hard.
  - Area: maximise $|I||J|$, also NP-Hard (proven after the Cheng and Church paper).
Algorithms

- Since the problems are computationally intractable, use heuristics to find biclusters of “large” size.
- Basic idea: add/delete a row/column until mean squared residue does not decrease.
  - Delete a row/column if its deletion improves mean squared residue.
  - Add a row/column if its addition improves mean squared residue.
  - Add some tricks to allow deletion/addition of multiple rows/columns so that it is not necessary to recompute mean squared residue after each change.