

Multi Omics Clustering



Outline

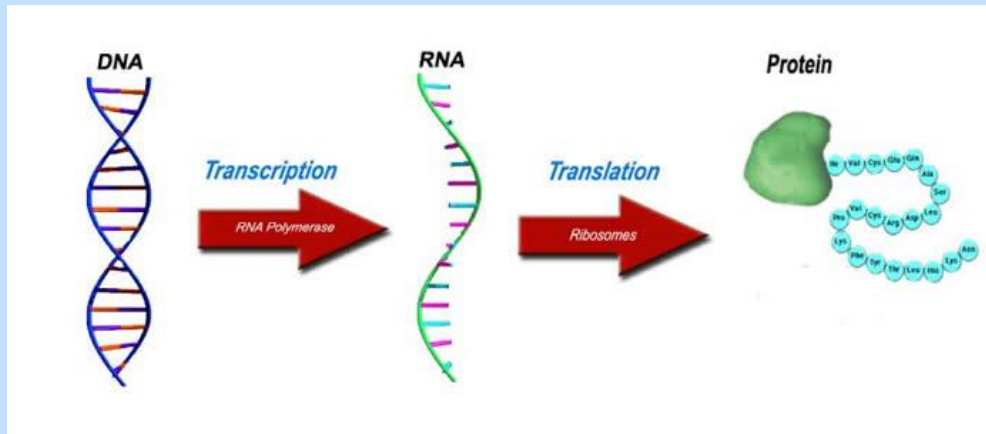
- Introduction
- Cluster of Clusters (COCA)
- iCluster
- Nonnegative Matrix Factorization (NMF)
- Similarity Network Fusion (SNF)
- Multiple Kernel Learning (MKL)

Outline

- Introduction
- Cluster of Clusters (COCA)
- iCluster
- Nonnegative Matrix Factorization (NMF)
- Similarity Network Fusion (SNF)
- Multiple Kernel Learning (MKL)

Omics

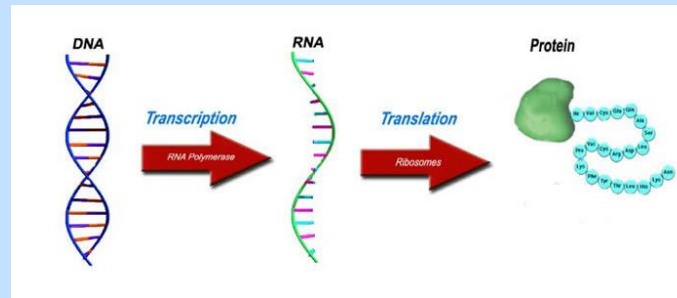
- “Basic dogma of biology”:



- So far in the course - mainly RNA
- Can't we use DNA or protein data?

Omics

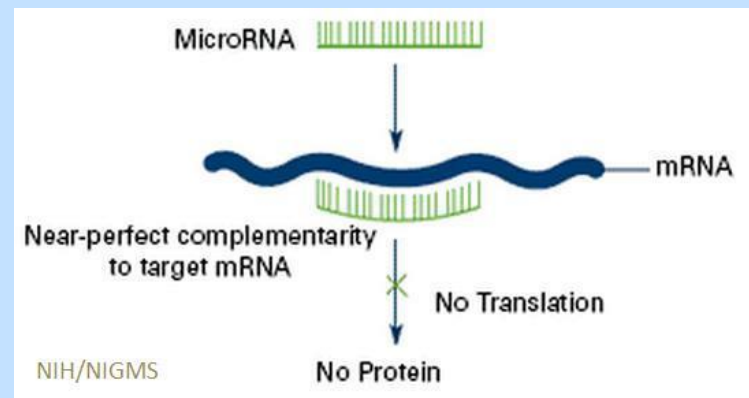
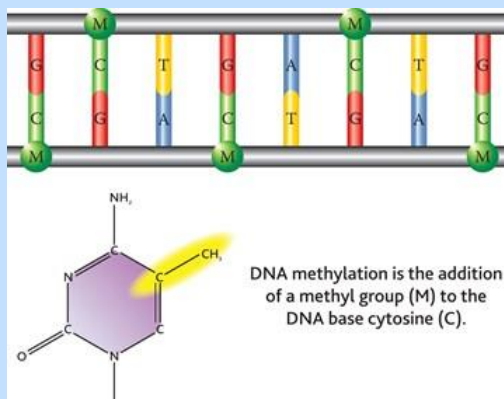
- “Basic dogma of biology”:



- Omics - characterization of specific type of biological data
- Anything that ends with -ome
- Genome, genomic (adjective), genomics (study of genome)
- Genome, transcriptome, proteome

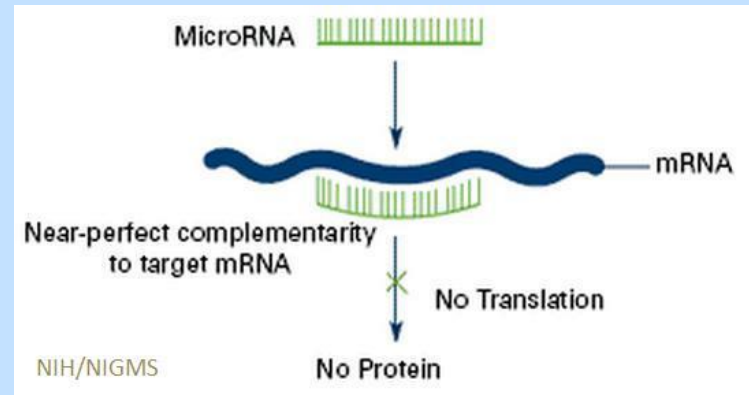
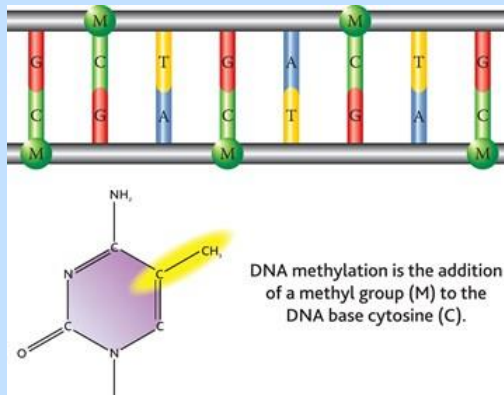
Additional Omics

- All cells in the human body share (approximately) the same DNA
- However, different genes are expressed and in different abundance in different tissues
- Regulation that is present not only in the genome
- Methylation and microRNA



Additional Omics

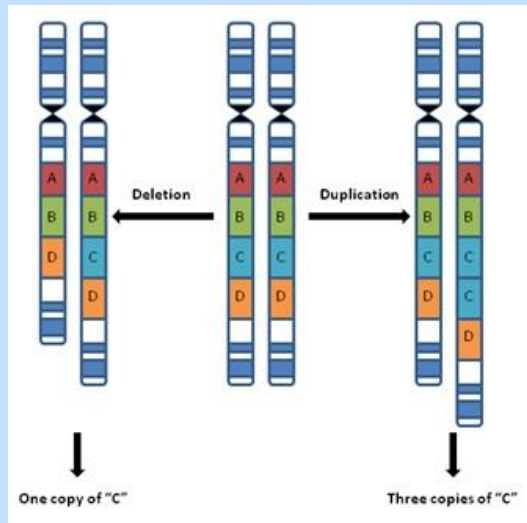
- Methylation and microRNA



- Methylation - "punctuation" for the genetic code
 - Methylation of promoters correlated with decreased expression
- MicroRNA - RNA molecules not coding for protein
 - Can stop RNA from being translated

Additional Omics

- Copy number variations



- Prevalent in cancer

Additional Omics

- Genome
 - Transcriptome (expression)
 - Proteome
 - Methylome
 - MicroRNA
 - Copy number variations
 - (Clinical parameters)
-
- All can be measured in a high throughput manner
 - (Either arrays, sequencing, or mass spectrometry)

Additional Omics

- Genome
- Transcriptome (expression)
- Proteome
- Methylome
- MicroRNA
- Copy number variations
- (Clinical parameters)

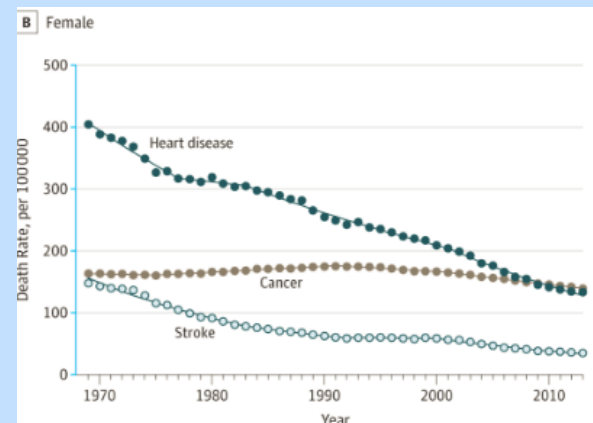
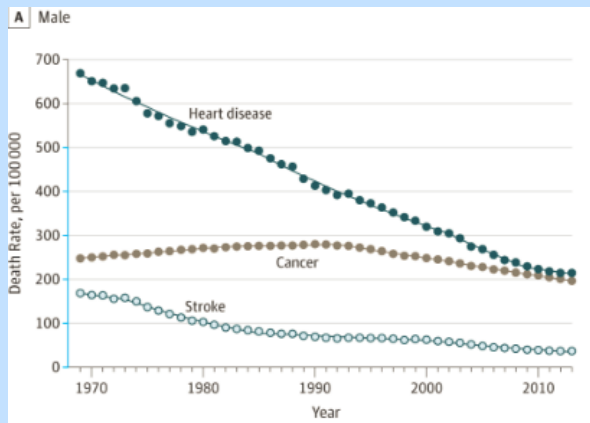
- Can be used to answer different questions
 - Predict phenotype from genotype
 - Predict age from methylation

Multi Omics

- Using several types of omics data
- Multi omics clustering
- Multi omics dimension reduction
- Multi omics predictions
- ...
- This talk: multi omics clustering for cancer subtyping

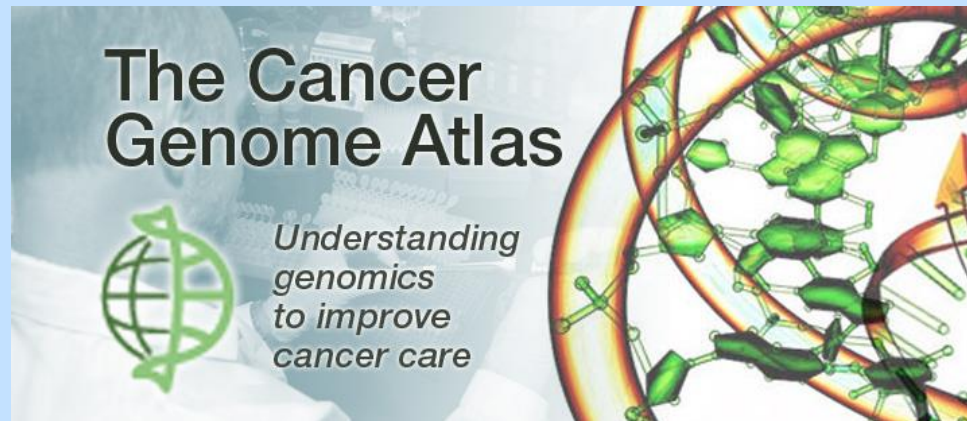
Cancer Subtyping

- Cancers are heterogeneous (even within a tissue)
- Therapeutic decisions based on pathologic parameters and biomarkers
- High throughput expression data used in recent years (PAM50, MammaPrint, Oncotype...)
- Copy number, methylation etc. has a known role in cancer prognosis



TCGA

- The Cancer Genome Atlas
- Collect and analyze data from cancer patients using high throughput technologies
- Samples from 11000 patients, more than 30 tumor types
- (Hundreds of millions of dollars)



Multi Omics Data

- Mutations - binary (or sequence)
- Copy number variations - counts
- Gene expression, micro RNA expression, protein arrays - numerical (hundreds miRNA, 20000 genes)
- DNA methylation - beta value (up to 450K sites)
- Clinical parameters - age, tumor size...

	Gene1 Exp	Gene2 Exp	Gene3 Exp
Patient1	323	643	50
Patient2	356	712	38
Patient3	344	680	58

	CpG 1	CpG 2	CpG 3
Patient1	0.2	0.3	0.12
Patient2	0.25	0.32	0.17
Patient3	0.23	0.35	0.09

Approaches

Early integration

- Concatenate matrices
- Dimensionality
- Data from different distributions

Intermediate integration

- Omics are different "views" of clusters
- Build model using all omics

Late integration

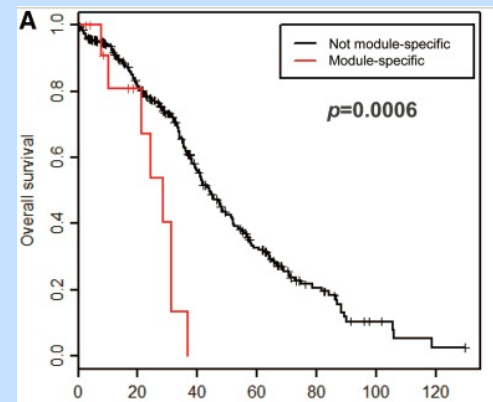
- Consensus clustering
- Dependencies between features from different omics
- Weak but consistent signals

Approaches

- Support for any omic data type
 - General
 - Loses knowledge of the biological role
 - (Continuous vs. discrete)
- Omic specific methods
 - For example - expression is increasing in copy number
- Omic specific feature representation
 - Replace genes with pathways

Comparing Clusterings

- Compare to “gold standard”
 - No gold standard for cancer subtypes
- Create synthetic data
- Compare prognosis or other clinical and genomic features
- Use homogeneity, separation, silhouette score...



Silhouette Score

- $a(i)$ - average distance of i to points within its cluster
- $b(i)$ - average distance of i to points within closest cluster it doesn't belong to
- Silhouette score for i :
- $s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}$
- Between -1 and 1
- Silhouette score for clustering is average silhouette score across samples
- (Requires a definition of distance)

Introduction - Recap

- Omics
- Multi omics and how the datasets look
- Cancer subtyping
- TCGA
- Multi omics clustering approaches
- Comparing clusterings

Outline

- Introduction
- **Cluster of Clusters (COCA)**
- iCluster
- Nonnegative Matrix Factorization (NMF)
- Similarity Network Fusion (SNF)
- Multiple Kernel Learning (MKL)

Cluster of Clusters (COCA)

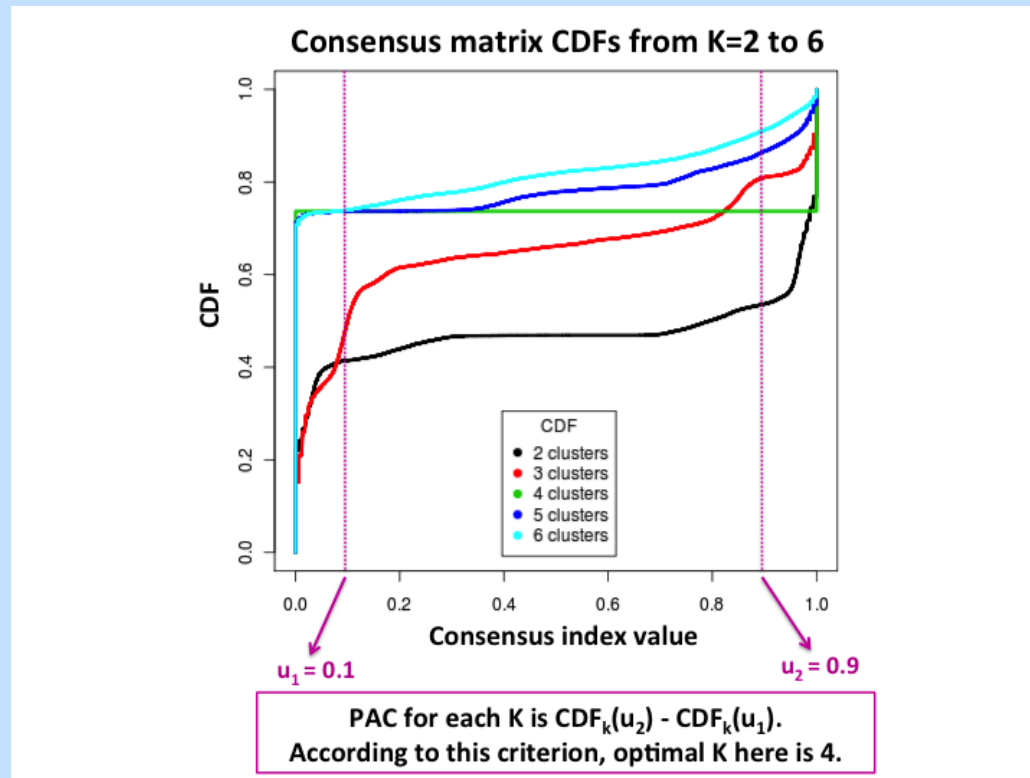
- Hoadley et. al (Cell, 2014)
 - (as part of The Cancer Genome Atlas Research Network)
- Late integration method
- Tissue of origin is heavily used for therapeutic decision making
- Cluster TCGA samples from multiple tissues
- What are the clusters? Do they match the tissue of origin?

Consensus Clustering Reminder

- The data $D = \{e_1, \dots, e_N\}$; e_i : GE profile of sample/patient #i
- Want a partition $\{P_1, \dots, P_k\}$ of the items
- $D^{(h)}$: resampled dataset #h
- $M^{(h)}$: result of clustering $D^{(h)}$
 - $M^{(h)}(i,j) = 1$ if i,j in same cluster, 0 o/w
- $I^{(h)}(i,j) = 1$ if i,j are both included in $D^{(h)}$
- $\mathcal{M}(i,j) = \sum_h M^{(h)}(i,j) / \sum_h I^{(h)}(i,j)$ \mathcal{M} : consensus matrix
- Change to distance: $\mathcal{D}(i,j) = 1 - \mathcal{M}(i,j)$ $\mathcal{M}(i,j)$ consensus index of i,j
- Cluster \mathcal{D} using a distance based method, e.g. HC

Consensus Clustering Reminder

- $CDF(c) = |\{(i,j) \mid i < j, \mathcal{M}(i,j) \leq c\}| / N(N-1)/2$



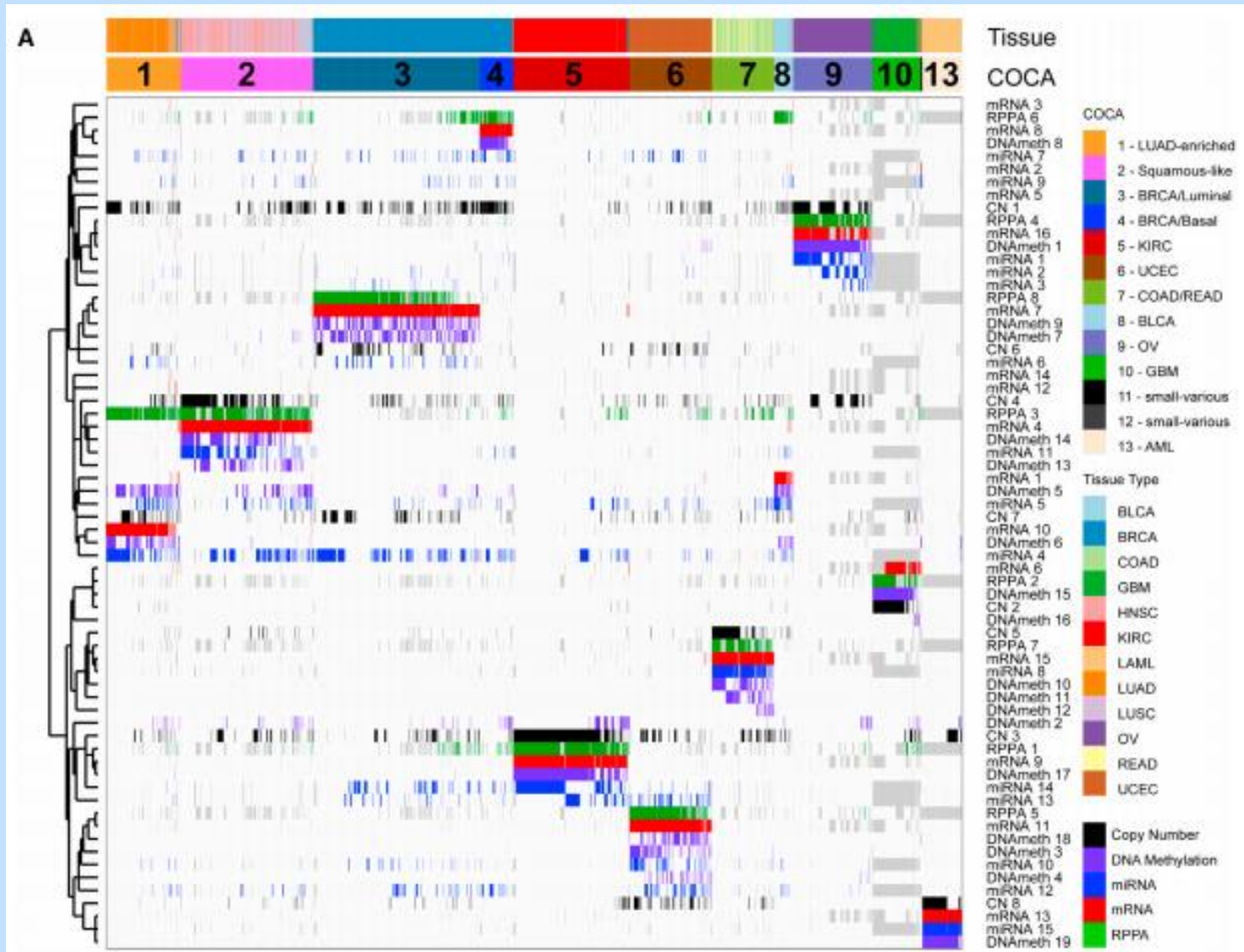
COCA

- Cluster each omic separately
 - Can use any clustering algorithm for each omic
 - different omics can use different k
- Represent each sample by an indicator vector of the single omic cluster memberships:
 - Sample is in cluster 3 out of 5 in omic 1: (0, 0, 1, 0, 0)
 - Sample is in cluster 2 out of 3 in omic 2: (0, 1, 0)
 - Sample representation: (0, 0, 1, 0, 0, 0, 1, 0)
- Run consensus clustering on the new representation (80% sampling, hierarchical clustering algorithm) for the samples and return its output

COCA - Results

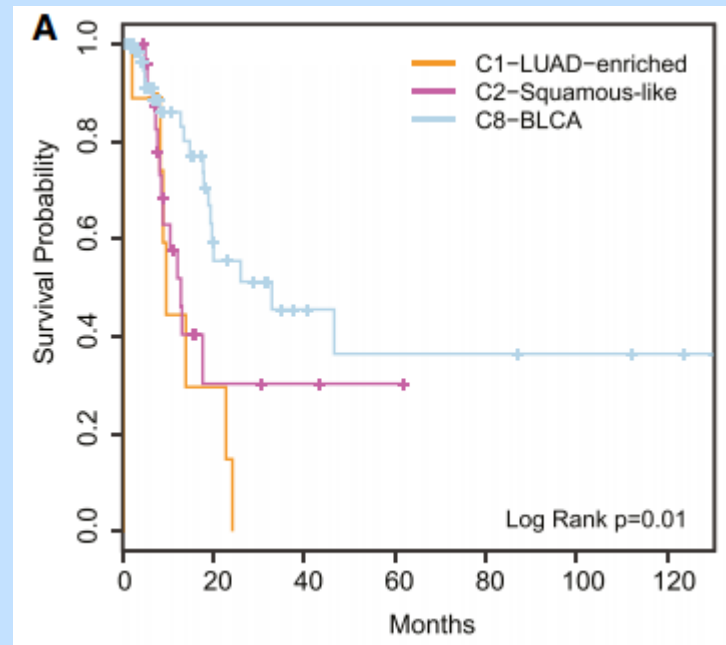
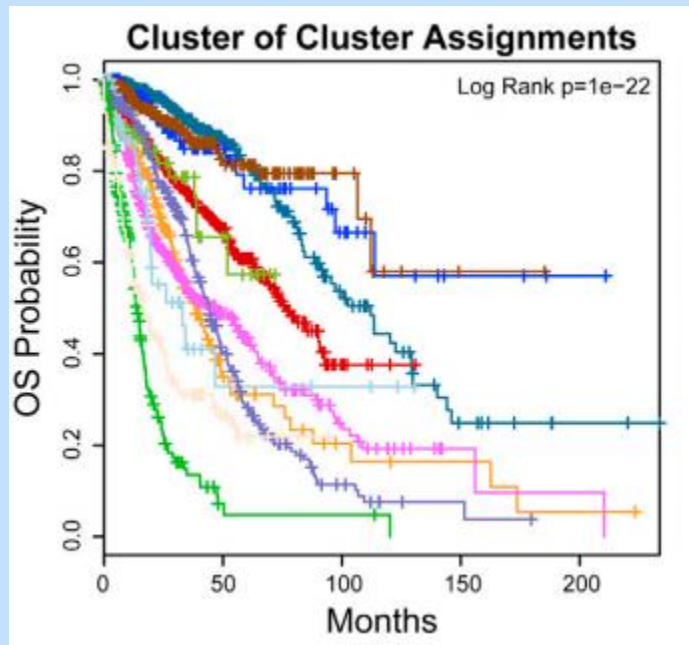
- Run on all (3527) TCGA samples from 12 cancer tissues
- Use expression, methylation, miRNA, copy number, RPPA (protein arrays)
- Each with different clustering scheme - hierarchical, NMF, consensus clustering...
- 11 clusters, 5 nearly identical to tissue of origin
- Lung squamous, head and neck cluster together
- Bladder cancer split into 3 pan-cancer subtypes

COCA - Results



COCA - Results

- Survival analysis of the different clusters
- Survival analysis of bladder cancers within different clusters



COCA - Recap

- Algorithm:
 - Cluster each omic separately
 - Cluster membership indicator representation
 - Consensus clustering on that representation
- Run on TCGA data from all available tissues
- Clusters generally match tissue of origin, with few exceptions (squamous cancers, bladder cancer)

Outline

- Introduction
- Cluster of Clusters (COCA)
- **iCluster**
- Nonnegative Matrix Factorization (NMF)
- Similarity Network Fusion (SNF)
- Multiple Kernel Learning (MKL)

iCluster

- Shen, Olshen, Ladanyi (Bioinformatics, 2009)
 - Memorial Sloan-Kettering Cancer Center, New York
- Dimension reduction
- m different omics, X_i observed matrices of dimension $p_i \times n$
- Z - $k \times n$ cluster membership binary matrix

- | | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |

iCluster

- m different omics, X_i observed matrices of dimension $p_i \times n$
- Z - $k \times n$ cluster membership binary matrix
- $X = WZ + \epsilon$
- ϵ is added per column
- It is normal with zero mean and diagonal covariance (each feature has different independent noise)
- Equal observed values for data coming from same cluster (up to Gaussian noise)
- (PCA with membership as low rank representation. Also, very similar to k-means)

iCluster

- $X = WZ + \epsilon$
- Multi omic version:
- $X_1 = W_1Z + \epsilon_1$
- $X_2 = W_2Z + \epsilon_2$
- ...
- $X_m = W_mZ + \epsilon_m$
- Each ϵ_i is normal with zero mean and diagonal covariance ψ_i (again, added per column)

Bayesian Statistics

- First, Frequentist statistics
- Tossing a coin n times with probability p to heads
- Can estimate p by maximizing the likelihood:
 $\Pr(\text{data} \mid p)$
- Why maximize $\Pr(\text{data} \mid p)$ and not $\Pr(p \mid \text{data})$?
- p is not a random variable, it is a number!

Bayesian Statistics

- In Bayesian statistics, parameters are random variables
- $\Pr(p)$ - prior probability for parameter p (e.g. $\text{uniform}[0,1]$)
- Natural way to incorporate domain knowledge to the model
- $\Pr(p \mid \text{data}) = \Pr(\text{data} \mid p) * \Pr(p) / \Pr(\text{data})$
- Can maximize $\Pr(p \mid \text{data})$ or take $E[p \mid \text{data}]$

iCluster

- $X_1 = W_1 Z + \epsilon_1$
 - $X_2 = W_2 Z + \epsilon_2$
 - ...
 - $X_m = W_m Z + \epsilon_m$
-
- Instead of a discrete Z , use continuous Z^*
 - Assume prior distribution $Z^* \sim N(0, I)$
 - Note that W are numbers, and Z^* random variables
 - (Other formulations use $Z^* Z^{*'} = I$. Using normal Z^* may lose interpretability).

iCluster

- $X_1 = W_1 Z + \epsilon_1$
- $X_2 = W_2 Z + \epsilon_2$
- ...
- $X_m = W_m Z + \epsilon_m$
- $W = (W_1, \dots, W_m)'$
- $X = (X_1, \dots, X_m)' \sim N(0, WW' + \psi)$
- Can write the log likelihood and try to numerically optimize

Expectation Maximization

- Reminder: similar problem in de novo motif discovery
- Observed sequence is either a part of a motif or the background - denote the unknown data Z

Outline of EM algorithm:

- Choose starting θ
- Repeat until convergence of θ :
 - E-step: Re-estimate Z from θ, X
 - M-step: Re-estimate θ , from X, Z
- Repeat all of the above for various starting values $\theta, \lambda \dots$

Expectation Maximization

Outline of EM algorithm:

- Choose starting θ
- Repeat until convergence of θ :
 - E-step: Re-estimate Z from θ, X
 - M-step: Re-estimate θ , from X, Z
- Repeat all of the above for various starting values $\theta, \lambda \dots$
- $X_i = W_i Z + \epsilon_i$
- In our case, $\theta = (W, \psi)$

EM for iCluster

$$l_c(W, \psi, Z) = -\frac{n}{2} \left\{ \sum_{i=1}^M p_i \ln(2\pi) + \ln \det(\Psi) \right\}$$

$$-\frac{1}{2} \left\{ \text{tr}((X - WZ^*)' \Psi^{-1} (X - WZ^*)) + \text{tr}(Z^{*'} Z^*) \right\}.$$

$$\left(\det(2\pi \Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)' \Sigma^{-1}(\mathbf{x}-\mu)} \right)$$

- **Number of parameters = $O(\sum p_i * k) \gg n$**
- Add regularization: encourage the model to use less parameters
- Lasso regularization (Tibshirani, 1996)
- $l(W, \psi) = l_c(W, \psi) - \lambda * \sum_i \sum_j \sum_k |w_{ijk}|$
- λ - tradeoff between likelihood and shrinkage
- Feature selection!

EM for iCluster

- Repeat until convergence of θ :
 - E-step: Re-estimate Z from θ, X
 - M-step: Re-estimate θ , from X, Z
- E-step (expected value of Z given current parameter estimates and data):

$$E[Z^*|X] = W' \Sigma^{-1} X \text{ and} \\ E[Z^*Z^{*'}|X] = I - W' \Sigma^{-1} W + E[Z^*|X]E[Z^*|X]'$$

- M-step:

$$\Psi^{(t+1)} = \frac{1}{n} \text{diag} \left\{ XX' - W^{(t)} E[Z^*|X]X' \right\}$$

$$W_{\text{lasso}}^{(t+1)} = \text{sign}(W^{(t+1)}) \left(|W^{(t+1)}| - \lambda \right)_+,$$

- $W^{(t+1)} = (X E[Z^*|X])' (E[Z^*Z^{*'}|X])^{-1}$
- $\Sigma = W W' + \psi$

EM for iCluster

- E-step (expected value of Z given current parameter estimates and data):

$$E[Z^*|X] = W' \Sigma^{-1} X \text{ and} \\ E[Z^*Z^{*'}|X] = I - W' \Sigma^{-1} W + E[Z^*|X]E[Z^*|X]'$$

- M-step:

$$\Psi^{(t+1)} = \frac{1}{n} \text{diag} \left\{ XX' - W^{(t)} E[Z^*|X]X' \right\} \quad W_{\text{lasso}}^{(t+1)} = \text{sign}(W^{(t+1)}) \left(|W^{(t+1)}| - \lambda \right)_+,$$

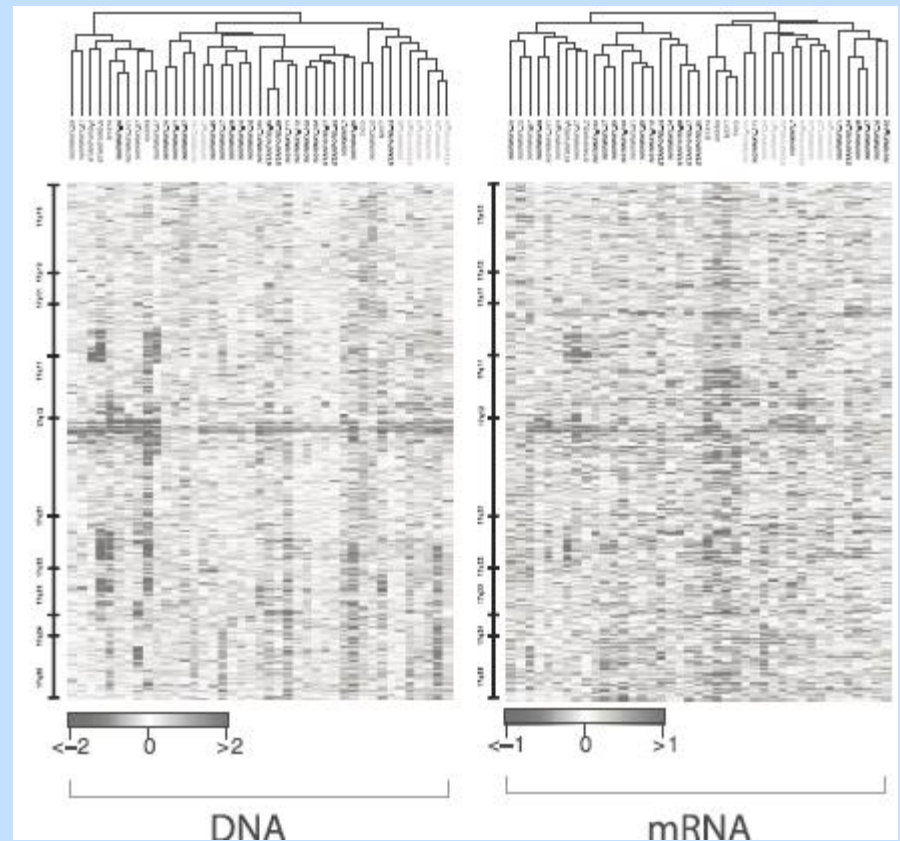
- $W^{(t+1)} = (X E[Z^*|X])' (E[Z^*Z^{*'}|X])^{-1}$
- $\Sigma = WW' + \psi$
- Finally, estimate for Z is given by $E[Z^*|X]$
- Run k-means on $E[Z^*|X]$ to obtain final cluster assignments

iCluster Model Selection

- How to choose k and λ ?
- $E[Z^*|X]' E[Z^*|X]$ is $n \times n$ matrix
- For cluster matrix Z , ordered by cluster membership, $E[Z|X]' E[Z|X]$ is a perfect 1-0 block matrix
- Measure distance of absolute values between observed normalized $E[Z^*|X]' E[Z^*|X]$, and perfect one
- Measures the posterior probability that two samples belong to the same cluster
- Choose k and λ that minimize the distance

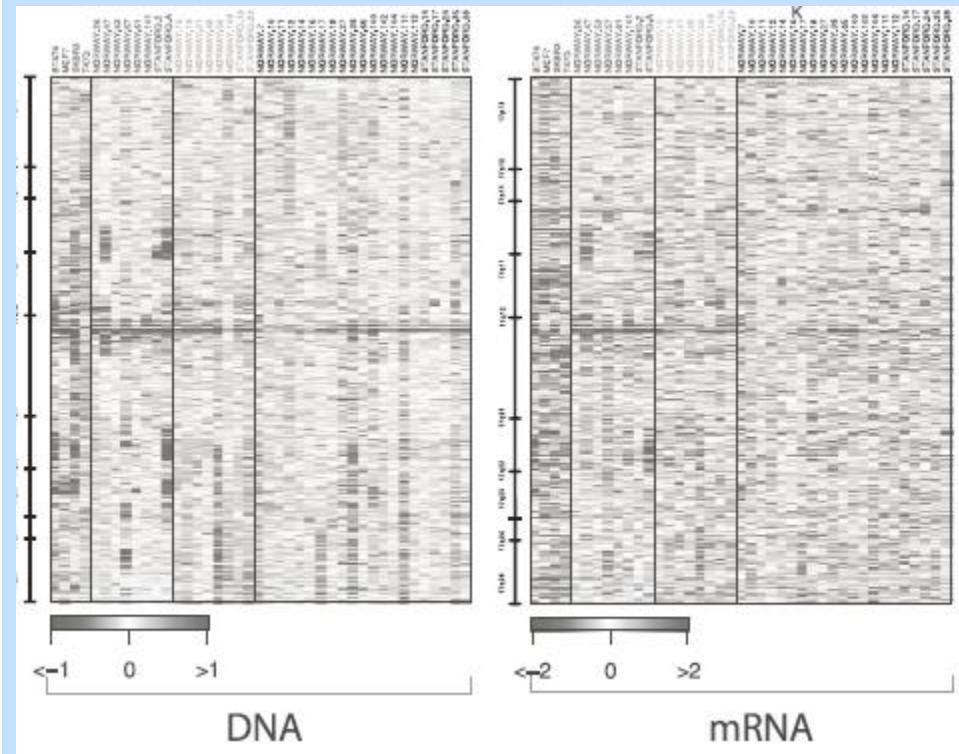
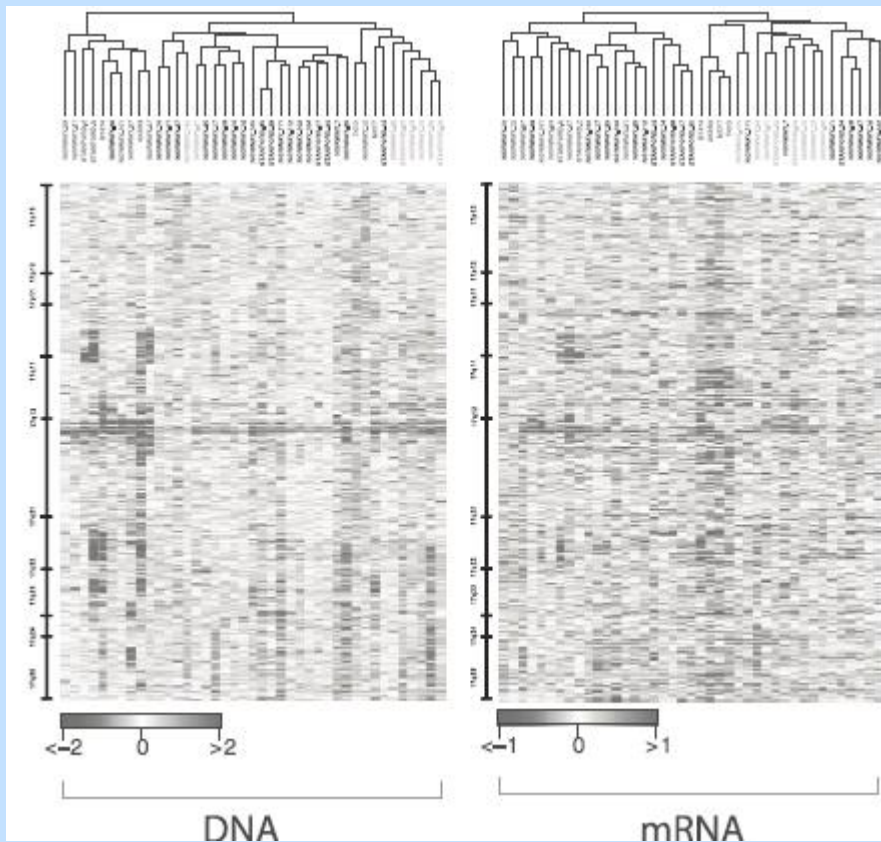
iCluster - Results

- Dataset: gene expression and copy number variation
 - 37 breast cancer + 4 cell lines samples
 - 91 lung adenocarcinoma
- Separate omic hierarchical clustering



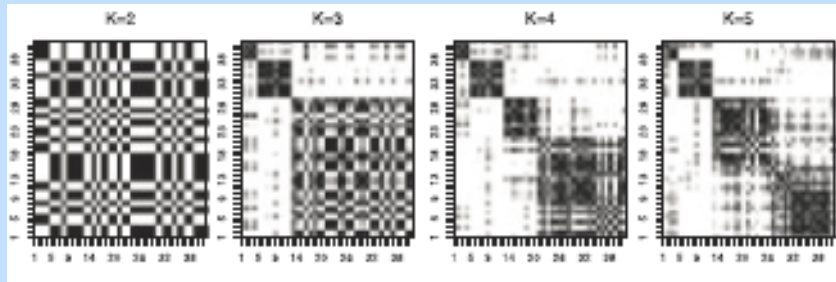
iCluster - Results

- Separate compared to integrative clustering

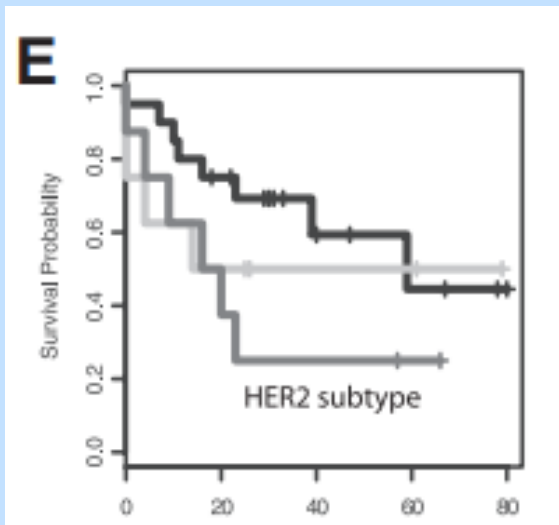


iCluster - Results

- $E[Z^*|X]' E[Z^*|X]$ matrix:

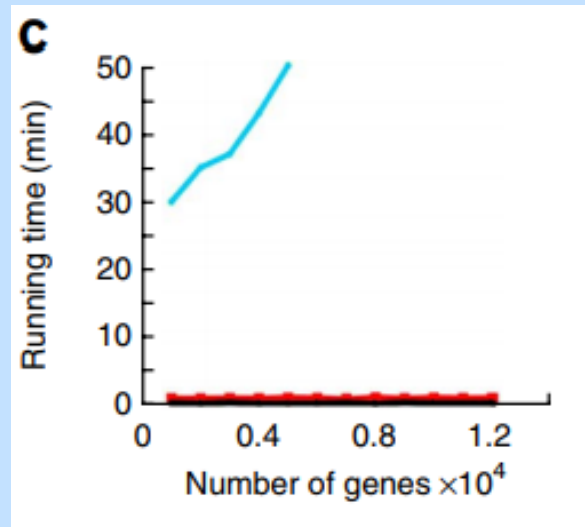


- Survival analysis:



iCluster - Results

- Runtime
- About an hour for ~200 samples, 4000 + 1300 + 500 features, and therefore requires gene preselection



iCluster - Recap

- Low dimension + probabilistic model
- $X_i = W_i Z + \epsilon_i$
- Z and ϵ_i have normal distribution
- Find parameters using EM with regularization for a sparse model
- Use deviation from perfect clustering matrix to determine k and λ
- Run on breast and lung adenocarcinoma samples

Outline

- Introduction
- Cluster of Clusters (COCA)
- iCluster
- Nonnegative Matrix Factorization (NMF)
- Similarity Network Fusion (SNF)
- Multiple Kernel Learning (MKL)

Joint NMF

- Shihua Zhang, ..., Jasmine Zhou (2012, bioinformatics)
 - University of Southern California, now at UCLA
- NMF = Nonnegative Matrix Factorization
- Dimension reduction -basic idea similar to iCluster
- Model can be used for clustering, but in this work the main goal is to find “md-modules”: (possibly overlapping) sets of features from all omics that define the patients' molecular profile

NMF

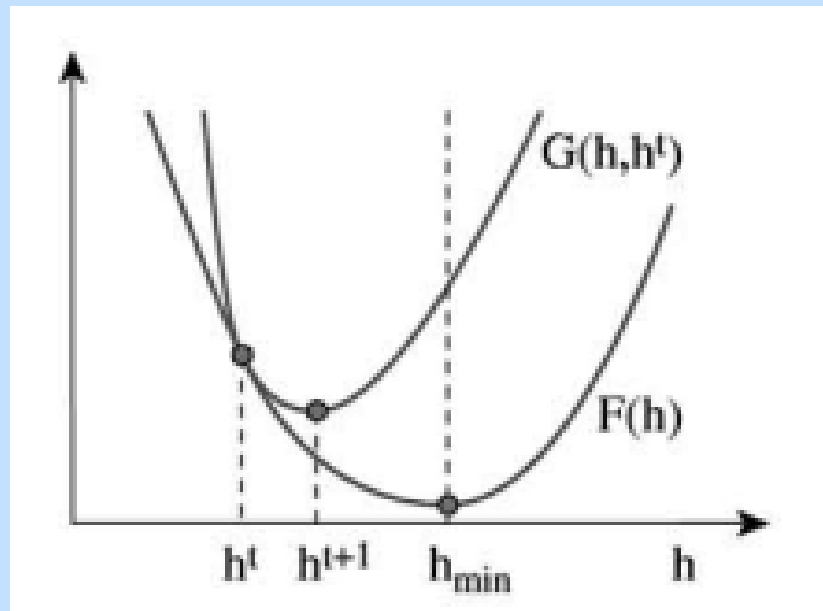
- NMF = Nonnegative Matrix Factorization
- Given a non negative matrix X , factorize it as $X = WH$ s.t $W, H \geq 0$,
- $x_{.j} = \sum_k w_{.k} h_{kj} = Wh_{.j}$
- Higher interpretability, makes sense where data is comprised of several parts
- Minimize $\|X - WH\|_F^2$ ($\|A\|_F = \sqrt{\sum_i \sum_j a_{ij}^2}$)
- Often optimized using multiplicative update rule:
- $H_{ab} = H_{ab} \frac{(W^T X)_{ab}}{(W^T W H)_{ab}}, W_{ab} = W_{ab} \frac{(X H^T)_{ab}}{(X H H^T)_{ab}}$

NMF - Proof Sketch

- Lee and Seung, NIPS 2000
- Minimize $\|X - WH\|_F^2$
- Will show proof sketch for H update rule
- Minimize $F(h) = \|x - Wh\|_F^2$
- Definition: G is an auxiliary function for $F(h)$ if:
 $G(h, h') \geq F(h)$, $G(h, h) = F(h)$
- Lemma: if G is an auxiliary function, then F is non increasing under the update: $h^{t+1} = \operatorname{argmin}_h G(h, h^t)$
- Proof: $F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t)$

NMF - Proof Sketch

- Definition: G is an auxiliary function for $F(h)$ if:
 $G(h, h') \geq F(h)$, $G(h, h) = F(h)$
- Lemma: if G is an auxiliary function, The F is non increasing under the update: $h^{t+1} = \operatorname{argmin}_h G(h, h^t)$



NMF - Proof Sketch

- Lemma (not proved here):
- $K_{ab}(h^t) = \frac{\delta_{ab}(W^T W h^t)_a}{h_a^t}$
- $G(h, h^t) = F(h^t) + (h - h^t)^T \nabla F(h^t) + \frac{1}{2} (h - h^t)^T K(h^t) (h - h^t)$
- Is an auxiliary function for $F(h)$
- (Easy to see that $G(h, h) = F(h)$)
- $h^{t+1} = \operatorname{argmin}_h G(h, h^t)$ gives the update:
- $H_{ab} = H_{ab} \frac{(W^T X)_{ab}}{(W^T W H)_{ab}}$

NMF

- Now in genomic context
- $X = WH$, $x_{.j} = \sum_k w_{.k} h_{kj} = Wh_{.j}$
- X is $M \times N$ matrix, M patients and N features
- W is $M \times k$ matrix, k is the number of modules
- H is $k \times N$ matrix
- W 's columns are basis vectors for the features (e.g genes), H matrix is the coefficients

Joint NMF

- $X = WH, x_{.j} = \sum_k w_{.k} h_{kj} = Wh_{.j}$
- $X_l = WH_l$
- X_l is $M \times N_l$ matrix, M patients and N_l features
- W is $M \times k$ matrix, k is the number of modules
- H_l is $k \times N_l$ matrix
- Basis vectors (W) are identical in all omics, different coefficient matrices

Joint NMF

- $X_l = WH_l$
- Optimization problem is $\min \sum_l ||X_l - WH_l||_F^2, W \geq 0, H_l \geq 0$
- Adapt the single matrix multiplicative update rule (ex):
- $H_{ab} = H_{ab} \frac{(W^T X)_{ab}}{(W^T W H)_{ab}}, W_{ab} = W_{ab} \frac{(X H^T)_{ab}}{(X H H^T)_{ab}}$

$$W_{ia} = W_{ia} \frac{(X_1 H_1^T + X_2 H_2^T + X_3 H_3^T)_{ia}}{(W(H_1 H_1^T + H_2 H_2^T + H_3 H_3^T))_{ia}},$$
$$(H_I)_{a\mu} = (H_I)_{a\mu} \frac{(W^T X_I)_{a\mu}}{(W^T W H_I)_{a\mu}}, \quad I = 1, 2, 3.$$

Joint NMF

- $X_l = WH_l$
- Can cluster W 's rows or H 's columns to get clustering of the samples or of the features
- Here, look for md-modules
- Allow each feature to belong to more than one md-module
- look at each H_l , and for each of its k rows, include features with z score (using feature's mean and std) exceeding some threshold

$$z_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i}$$

Joint NMF

- look at each H_l , and for each of the k vectors, include features with z score (using feature's mean and std) exceeding some threshold

$$z_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i}$$

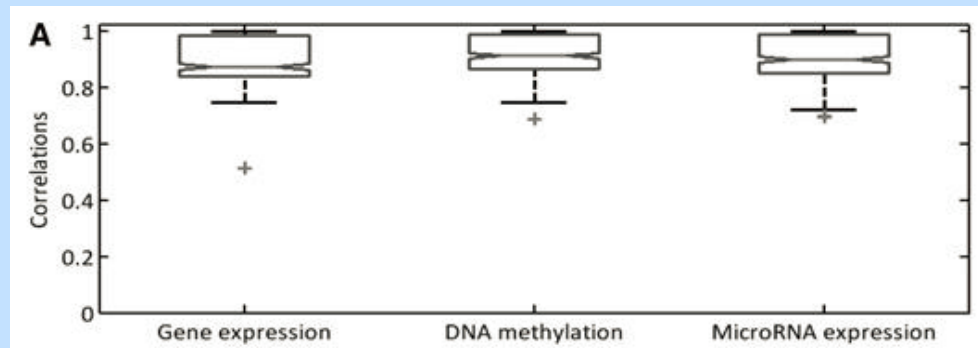
- Similarly, look at Columns of W and associate a patient with an md-module if its z-score exceeds some threshold
- The output is k md-modules, with features from each omic (and samples) associated with them

Joint NMF - Results

- Use ovarian cancer data from TCGA
- 385 samples
- 3 omics: gene expression, methylation and miRNA expression
- Negative values - double all features, one with positive and one with absolute value of negative
- $K = 200$ md-modules
- Cover ~3000 genes, ~2000 methylation sites, 270 miRNAs
- Average module sizes are ~240 genes, ~162 methylation sites and ~14 miRNAs (high overlap)

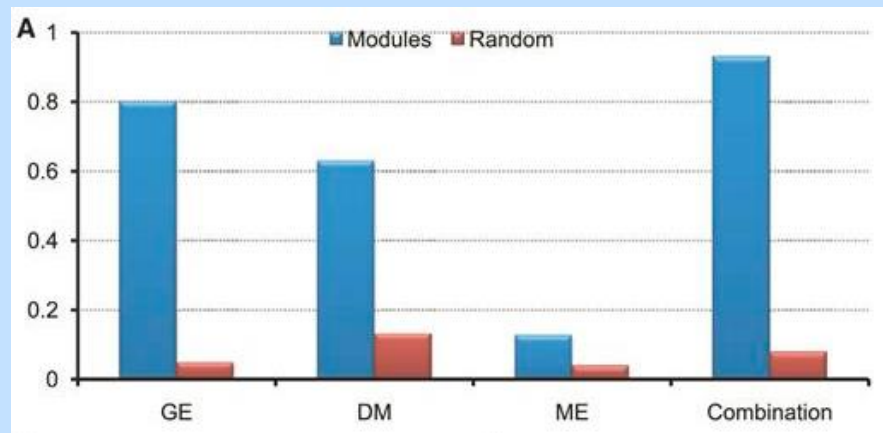
Joint NMF - Results

- Correlations of observed and reconstructed features
- The model doesn't lose "too much" information



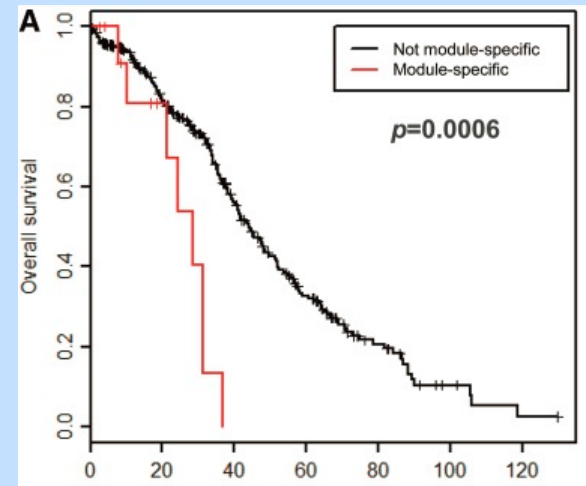
Joint NMF - Results

- For each module, look at:
 - Gene expression features in that module
 - Genes adjacent to methylation sites in the module
 - Genes regulated by miRNAs in the module
- Count md-modules with (FDR corrected) GO enrichment of at least one term
- Compare to random md-modules
- Combining all omics is more biologically meaningful



Joint NMF - Results

- 22 md-modules are enriched in genes with a known role in cancer (with $p\text{-value} < 0.05$). Note we would expect 10 by chance.
- 20 modules contain patients with significantly different age characteristics compared to patients not in the module
- (In plot: survival analysis for patients in module 166 compared to other patients. Didn't mention how many modules have different survival).



Joint NMF - Recap

- Low dimension + non negativity constraint
- $X_l = WH_l$
- Optimized using multiplicative update rules
- Look for md-modules: sets of features from all omics that largely determine the observed data
- Md-modules calculated from the factorization with z-score
- Run on TCGA ovarian cancer data
- Looking at all omics gives higher enrichment in md-modules compared to each omic alone

Outline

- Introduction
- Cluster of Clusters (COCA)
- iCluster
- Nonnegative Matrix Factorization (NMF)
- **Similarity Network Fusion (SNF)**
- Multiple Kernel Learning (MKL)

Similarity Network Fusion

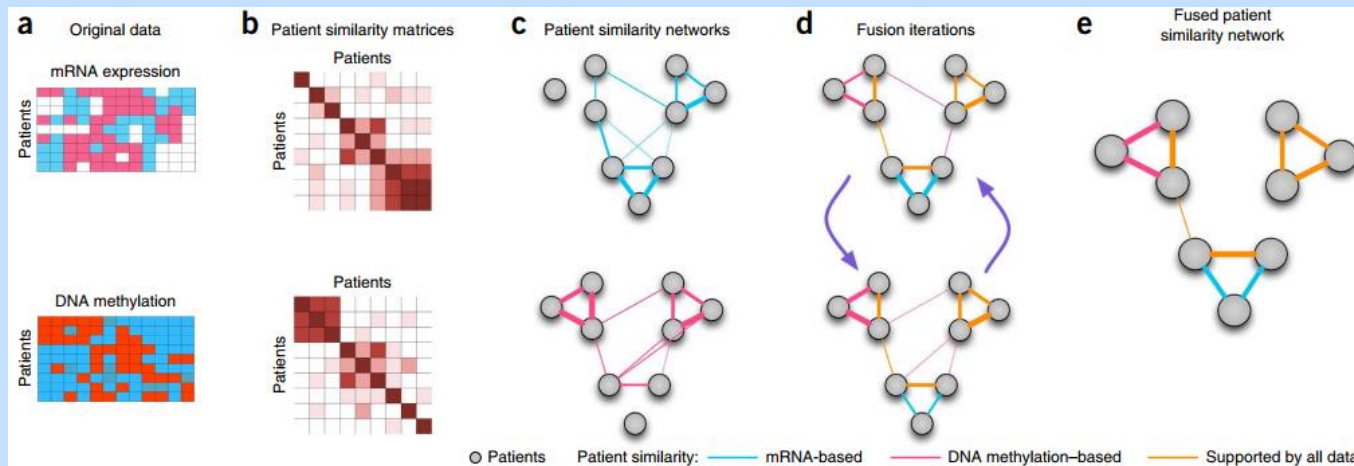
- Bo Wang, ..., Anna Goldenberg (Nature Methods, 2014)
 - University of Toronto
- Number of features \gg number of samples
- Dimension reduction methods' complexity depends on the number of features
- Formulations with non-convex / no closed form solution objective functions, so have to try many different initialization points

Similarity Network Fusion

- Idea: cluster based only on patients' similarity
- Aside from similarity computation, complexity is a function of the number of patients
- Less sensitivity to feature selection in practice
- More difficult to give interpretation to features as part of the model
- (Can still do analysis once we have the clusters, e.g. differentially expressed genes)

Similarity Network Fusion

- Construct similarity network for each omic
- Nodes are patients, edges' weights are similarity of patients in omic
- Iteratively update the weights of the networks, bringing the networks closer
- Obtain fused network



Similarity Network Fusion

- Initialization:

$$W(i, j) = \exp\left(-\frac{\rho^2(x_i, x_j)}{\mu \epsilon_{i, j}}\right)$$

$$P(i, j) = \begin{cases} \frac{W(i, j)}{2 \sum_{k \neq i} W(i, k)}, & j \neq i \\ 1/2, & j = i \end{cases}$$

$$S(i, j) = \begin{cases} \frac{W(i, j)}{\sum_{k \in N_i} W(i, k)}, & j \in N_i \\ 0 & \text{otherwise} \end{cases}$$

- $\epsilon_{i, j}$ measures the average distance of i and j to their neighbors, to correct for density
- N_i - k nearest neighbors of sample i , different k than the cluster number (~ 15 - 20 in practice)
- W - similarity, P - relative similarity, S - relative similarity within nearest neighbors
- P will be updated in each iteration

Similarity Network Fusion

- W - similarity, P - relative similarity, S - relative similarity within nearest neighbors
- (Assume for now we have two omics)
- P is updated in each iteration:

$$\mathbf{P}_{t+1}^{(1)} = \mathbf{S}^{(1)} \times \mathbf{P}_t^{(2)} \times (\mathbf{S}^{(1)})^T$$

$$\mathbf{P}_{t+1}^{(2)} = \mathbf{S}^{(2)} \times \mathbf{P}_t^{(1)} \times (\mathbf{S}^{(2)})^T$$

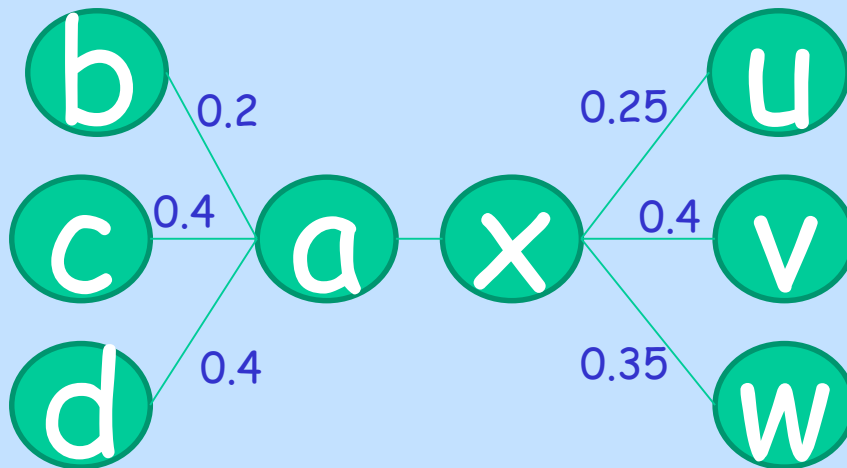
$$\mathbf{P}_{t+1}^{(1)}(i, j) = \sum_{k \in N_i} \sum_{l \in N_j} \mathbf{S}^{(1)}(i, k) \times \mathbf{S}^{(1)}(j, l) \times \mathbf{P}_t^{(2)}(k, l)$$

Similarity Network Fusion

$$\mathbf{P}_{t+1}^{(1)} = \mathbf{S}^{(1)} \times \mathbf{P}_t^{(2)} \times (\mathbf{S}^{(1)})^T$$

$$\mathbf{P}_{t+1}^{(2)} = \mathbf{S}^{(2)} \times \mathbf{P}_t^{(1)} \times (\mathbf{S}^{(2)})^T$$

$$\mathbf{P}_{t+1}^{(1)}(i, j) = \sum_{k \in N_i} \sum_{l \in N_j} \mathbf{S}^{(1)}(i, k) \times \mathbf{S}^{(1)}(j, l) \times \mathbf{P}_t^{(2)}(k, l)$$



$$p_{t+1}^1 = 0.2 * 0.25 * 0.02 + 0.2 * 0.4 * 0.007 + \dots$$

i	j	$p_t^2(i, j)$
b	u	0.02
b	v	0.007
b	w	0.01
c	u	0.09
c	v	0.08
c	w	0.003
d	u	0.05
d	v	0.008
d	w	0.03

Similarity Network Fusion

$$\mathbf{P}_{t+1}^{(1)} = \mathbf{S}^{(1)} \times \mathbf{P}_t^{(2)} \times (\mathbf{S}^{(1)})^T$$

$$\mathbf{P}_{t+1}^{(2)} = \mathbf{S}^{(2)} \times \mathbf{P}_t^{(1)} \times (\mathbf{S}^{(2)})^T$$

$$\mathbf{P}_{t+1}^{(1)}(i, j) = \sum_{k \in N_i} \sum_{l \in N_j} \mathbf{S}^{(1)}(i, k) \times \mathbf{S}^{(1)}(j, l) \times \mathbf{P}_t^{(2)}(k, l)$$

- Intuition: weighted average of neighbor similarities
- Only neighbors - for robustness
- (P normalized and made symmetric at the end of every iteration)
- Converges!
- After a few iterations:
- For more than two omics:
- We now have one network

$$\mathbf{P}^{(c)} = \frac{\mathbf{P}_t^{(1)} + \mathbf{P}_t^{(2)}}{2}$$

$$\mathbf{P}^{(v)} = \mathbf{S}^{(v)} \times \left(\frac{\sum_{k \neq v} \mathbf{P}^{(k)}}{m-1} \right) \times (\mathbf{S}^{(v)})^T, v = 1, 2, \dots, m$$

Spectral Clustering

- Cluster similarity matrix S
- Assume two clusters of \sim equal size
- If s_i, s_j belong to the same cluster, then $S(i, j) \gg 0$
- Otherwise, $S(i, j) \ll 0$
- $\sum_{i,j} (v_i - v_j)^2 S(i, j)$ is maximized when $v_i = \frac{1}{\sqrt{n}}$ for first cluster, $v_i = -\frac{1}{\sqrt{n}}$ for second cluster
- Instead of enforcing $v_i = \pm \frac{1}{\sqrt{n}}$, constraint $\|v\|_2 = 1, \|v\|_1 = 0$

Spectral Clustering

- Instead of enforcing $v_i = \pm \frac{1}{\sqrt{n}}$, constraint $\|v\|_2 = 1, \|v\|_1 = 0$
- $\min \sum_{i,j} (v_i - v_j)^2 S(i,j), s.t. v^T v = 1, \|v\|_1 = 0$
- Define $L = D - S$, where D is the row sum diagonal matrix. L is the graph's Laplacian.
- $v^T L v = \frac{1}{2} \sum_{i,j} (v_i - v_j)^2 S(i,j)$
- (Note the resemblance to PCA optimization problem - $\max v^T X^T X v, v^T v = 1$)
- Solution is second smallest eigenvector of L
- (Second - because $\|v\|_1 = 0$, v orthogonal to $\vec{1}$)

Spectral Clustering

- $\min \sum_{i,j} (v_i - v_j)^2 S(i,j), s.t. v^t v = 1, ||v||_1 = 0$
- Can now use v to cluster the samples, for example positive v values belong to one cluster and negative to the other
- Can be derived as an approximation to:

$$\text{RatioCut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

- For more than two clusters, find $2, \dots, k$ smallest eigenvectors of L
- The problem is solved by clustering V 's rows (using k -means for example)

Similarity Network Fusion

- After a few iteration:

$$\mathbf{P}^{(c)} = \frac{\mathbf{P}_t^{(1)} + \mathbf{P}_t^{(2)}}{2} .$$

- Cluster the network using spectral clustering (slightly different variation)
- Can use the network for other tasks
- Reminder: cox proportional hazards model
- Can use the network for regularization while learning the cox model's parameters such that similar patients will have similar prognoses

SNF - Results

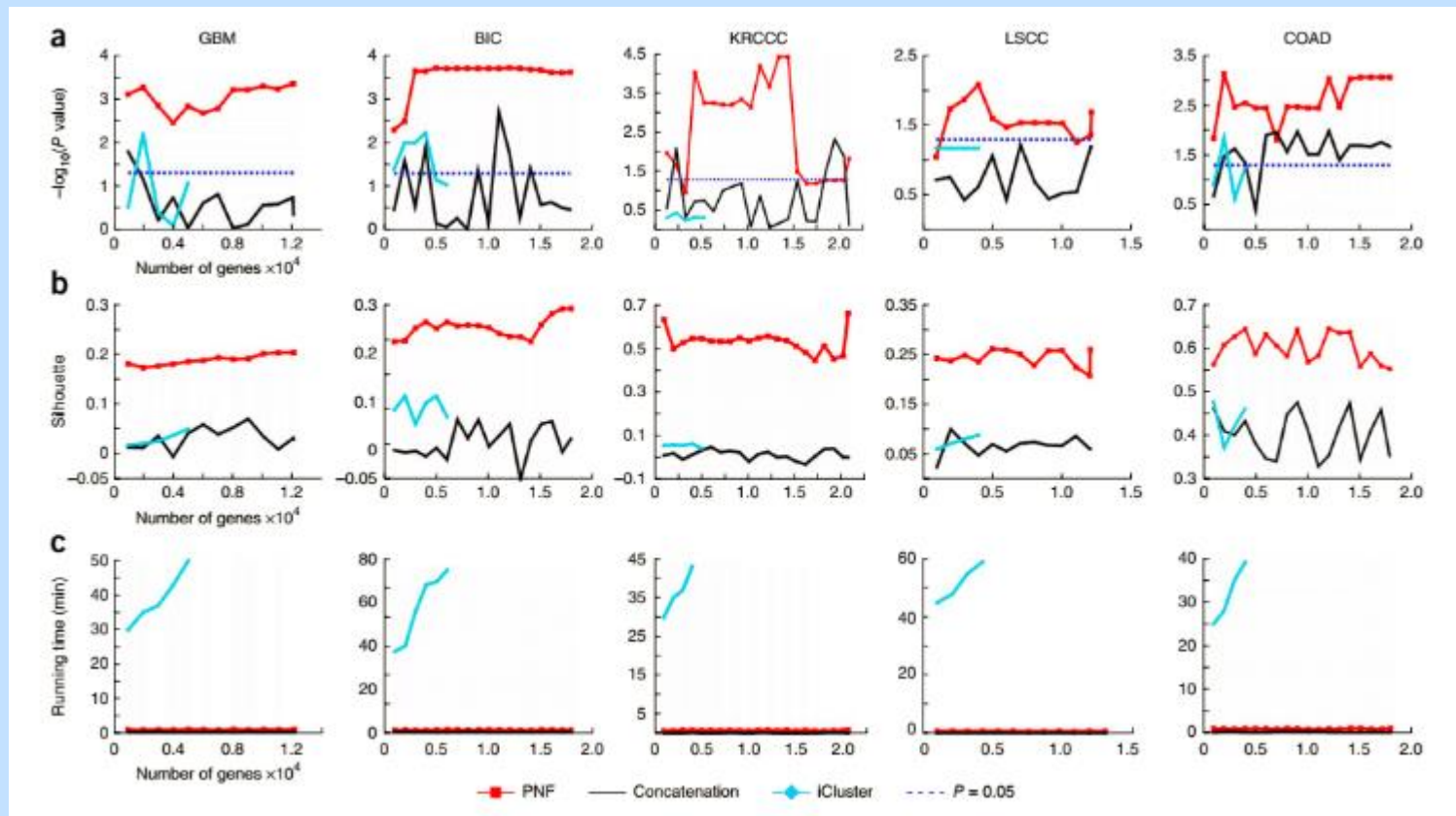
- Use gene expression, methylation and micro RNA
- 5 different cancer types: Glioblastoma Multiforme (aggressive brain cancer), breast, kidney, lung and colon
- Each cancer type has 90-215 patients
- (Different heuristics to choose k , $\mu = 0.5$ empirically)
- Compare prognosis using log rank for > 2 groups

Table 1 | SNF-based analysis versus individual data types

Cancer type	mRNA expression	DNA methylation	miRNA	SNF
GBM (3 clusters)	0.54	0.11	0.21	2.0×10^{-4}
BIC (5 clusters)	0.03	0.05	0.30	1.1×10^{-3}
KRCCC (3 clusters)	0.20	0.61	0.17	2.9×10^{-2}
LSCC (4 clusters)	0.06	0.26	0.46	2.0×10^{-2}
COAD (3 clusters)	0.18	0.04	0.46	8.8×10^{-4}

Analysis using Cox log-rank test P values.

SNF - Results



SNF - Recap

- Similarity based
- Patient similarity network per omic, followed by iteratively bringing the networks close to one another, until we have one network
- Cluster the network with spectral clustering
- Additional usages for the network
- Run on TCGA data from multiple tissues, and compare to iCluster, single omic and concatenation

Outline

- Introduction
- Cluster of Clusters (COCA)
- iCluster
- Nonnegative Matrix Factorization (NMF)
- Similarity Network Fusion (SNF)
- Multiple Kernel Learning (MKL)

Multiple Kernel Learning

- Speicher and Pfeifer (2015, bioinformatics)
 - Max Planck Institute for Informatics
- Similarity based method
- Multiple Kernel Learning - the general idea of using several kernels
- Have been used on single data type, mainly in supervised context but also in unsupervised
- Idea: use different kernels for different omics, together with multiple kernel dimension reduction algorithms

Graph Embedding

- x_i are input vectors, W is input similarity graph, D is diagonal constraint matrix

$$\underset{v}{\text{minimize}} \sum_{i,j=1}^N \left\| v^T x_i - v^T x_j \right\|^2 w_{ij}$$

$$\text{subject to } \sum_{i=1}^N \left\| v^T x_i \right\|^2 d_{ii} = \text{const.}$$

- Look for v that projects x vectors to a line such that similarities are kept
- (D matrix is mainly in order to avoid the trivial solution)
- (Difference from spectral clustering?)

Graph Embedding

$$\underset{v}{\text{minimize}} \quad \sum_{i,j=1}^N \left\| v^T x_i - v^T x_j \right\|^2 w_{ij}$$

$$\text{subject to} \quad \sum_{i=1}^N \left\| v^T x_i \right\|^2 d_{ii} = \text{const.}$$

- Can be shown that optimal v is necessarily in the span of the vectors:

$$v = \sum_{n=1}^N \alpha_n x_n$$

- (Kernel trick reminder: $K(x, y) = \langle \phi(x), \phi(y) \rangle$)
- $$\begin{aligned} v^T x_i - v^T x_j &= \sum_{n=1}^N \alpha_n x_n^T x_i - \sum_{n=1}^N \alpha_n x_n^T x_j \\ &= \sum_{n=1}^N \alpha_n K(n, i) - \sum_{n=1}^N \alpha_n K(n, j) \end{aligned}$$

Multiple Kernel Learning

- $v^t x_i - v^t x_j = \sum_{n=1}^N \alpha_n x_n^t x_i - \sum_{n=1}^N \alpha_n x_n^t x_j$
 $= \sum_{n=1}^N \alpha_n K(n, i) - \sum_{n=1}^N \alpha_n K(n, j)$
- We want different kernels for different omics
- $\sum_m \beta_m K_m$, $\beta_m \geq 0$ is also a kernel (ex)
- $K(n, i) = \sum_m \beta_m K_m(n, i)$
- $\sum_{n=1}^N \alpha_n \sum_m \beta_m K_m(n, i) = \alpha^t K^i \beta$

$$\mathcal{X}^i = \begin{pmatrix} K_1(1, i) & \cdots & K_M(1, i) \\ \vdots & \ddots & \vdots \\ K_1(N, i) & \cdots & K_M(N, i) \end{pmatrix} \in \mathbb{R}^{N \times M}$$

Multiple Kernel Learning

- From:
$$\underset{v}{\text{minimize}} \sum_{i,j=1}^N \left\| v^T x_i - v^T x_j \right\|^2 w_{ij} \quad \text{subject to} \quad \sum_{i=1}^N \left\| v^T x_i \right\|^2 d_{ii} = \text{const.}$$

- To:

$$\underset{\alpha, \beta}{\text{minimize}} \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta - \alpha^T \mathcal{K}^j \beta \right\|^2 w_{ij} \quad \text{subject to} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta \right\|^2 d_{ij} = \text{const.}$$

- With constraints:

$$\beta_m \geq 0, m = 1, 2, \dots, M. \quad \|\beta\|_1 = 1$$

$$\mathcal{K}^i = \begin{pmatrix} K_1(1,i) & \cdots & K_M(1,i) \\ \vdots & \ddots & \vdots \\ K_1(N,i) & \cdots & K_M(N,i) \end{pmatrix} \in \mathbb{R}^{N \times M}$$

Multiple Kernel Learning

$$\underset{\alpha, \beta}{\text{minimize}} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta - \alpha^T \mathcal{K}^j \beta \right\|^2 w_{ij} \quad \text{subject to} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta \right\|^2 d_{ij} = \text{const.}$$

$$\beta_m \geq 0, m = 1, 2, \dots, M. \quad \|\beta\|_1 = 1$$

$$\mathcal{K}^i = \begin{pmatrix} K_1(1,i) & \cdots & K_M(1,i) \\ \vdots & \ddots & \vdots \\ K_1(N,i) & \cdots & K_M(N,i) \end{pmatrix} \in \mathbb{R}^{N \times M}$$

- W and D :

$$w_{ij} = \begin{cases} 1, & \text{if } i \in \mathcal{N}_k(j) \vee j \in \mathcal{N}_k(i) \\ 0, & \text{else} \end{cases}$$

$$d_{ij} = \begin{cases} \sum_{n=1}^N w_{in}, & \text{if } i = j \\ 0, & \text{else.} \end{cases}$$

Multiple Kernel Learning

$$\underset{\alpha, \beta}{\text{minimize}} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta - \alpha^T \mathcal{K}^j \beta \right\|^2 w_{ij}$$

$$\text{subject to} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta \right\|^2 d_{ij} = \text{const.}$$

- α projects points to a single dimension
- Use matrix A instead to project to a different dimension
- Dimension not necessarily equal to the number of clusters

Multiple Kernel Learning

$$\underset{\alpha, \beta}{\text{minimize}} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta - \alpha^T \mathcal{K}^j \beta \right\|^2 w_{ij}$$

$$\text{subject to} \quad \sum_{i,j=1}^N \left\| \alpha^T \mathcal{K}^i \beta \right\|^2 d_{ij} = \text{const.}$$

- Optimize A and β iteratively in an alternating manner
- β is optimized using semidefinite programming

$$\begin{aligned} \min_{x^1, \dots, x^n \in \mathbb{R}^n} \quad & \sum_{i,j \in [n]} c_{i,j} (x^i \cdot x^j) \\ \text{subject to} \quad & \sum_{i,j \in [n]} a_{i,j,k} (x^i \cdot x^j) \leq b_k \quad \forall k. \end{aligned}$$

- A is optimized by solving a generalized eigenvalue problem
- Cluster the data projection $A^t K^i \beta$ using k-means

MKL - Results

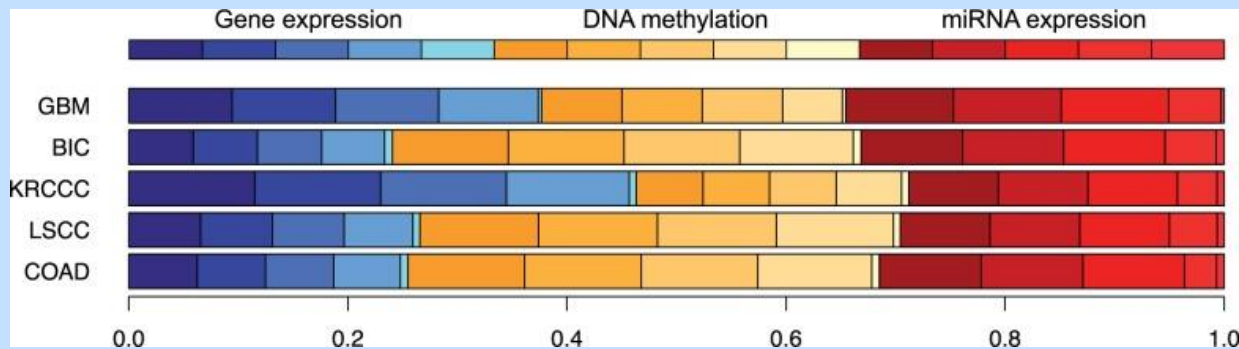
- Run on GBM, breast, lung, kidney and colon cancer (SNF dataset, ~90-215 patients per subtype)
- Use either 1 or 5 kernels per dataset:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

- $\gamma = \frac{1}{2d^2}, \gamma_n = c_n \gamma, c_n \in \{10^{-6}, 10^{-3}, 1, 10^3, 10^6\}$
- Fix the dimension to 5, and choose k using silhouette score
- β values measure the effect of each kernel

MKL - Results

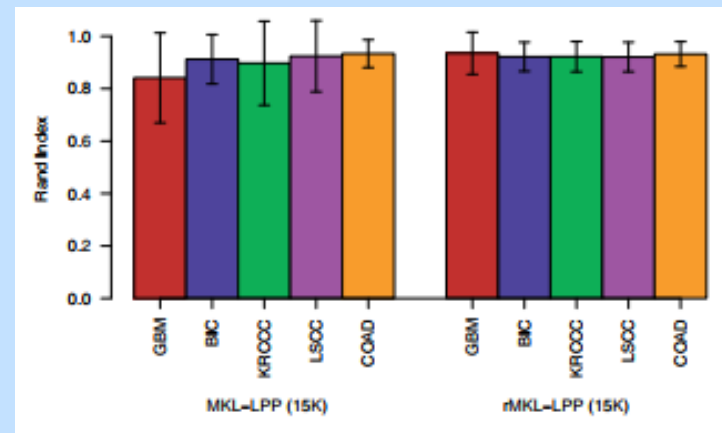
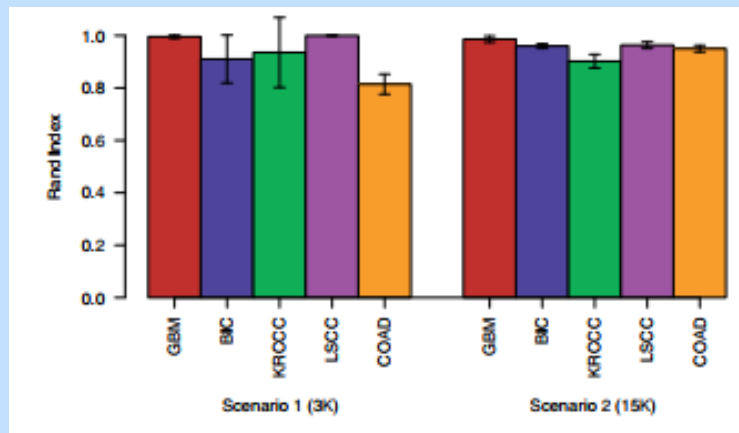
- β values measure the effect of each kernel
- Survival analysis comparing to SNF



Cancer type	SNF	rMKL-LPP	
		3K	15K
GBM	2.0E-4 (3)	4.5E-2 (5)	6.5E-6 (6)
BIC	1.1E-3 (5)	3.0E-4 (6)	3.4E-3 (7)
KRCCC	2.9E-2 (3)	0.23 (6)	4.0E-5 (14)
LSCC	2.0E-2 (4)	2.2E-3 (2)	2.4E-4 (6)
COAD	8.8E-4 (3)	2.8E-2 (2)	2.8E-3 (6)

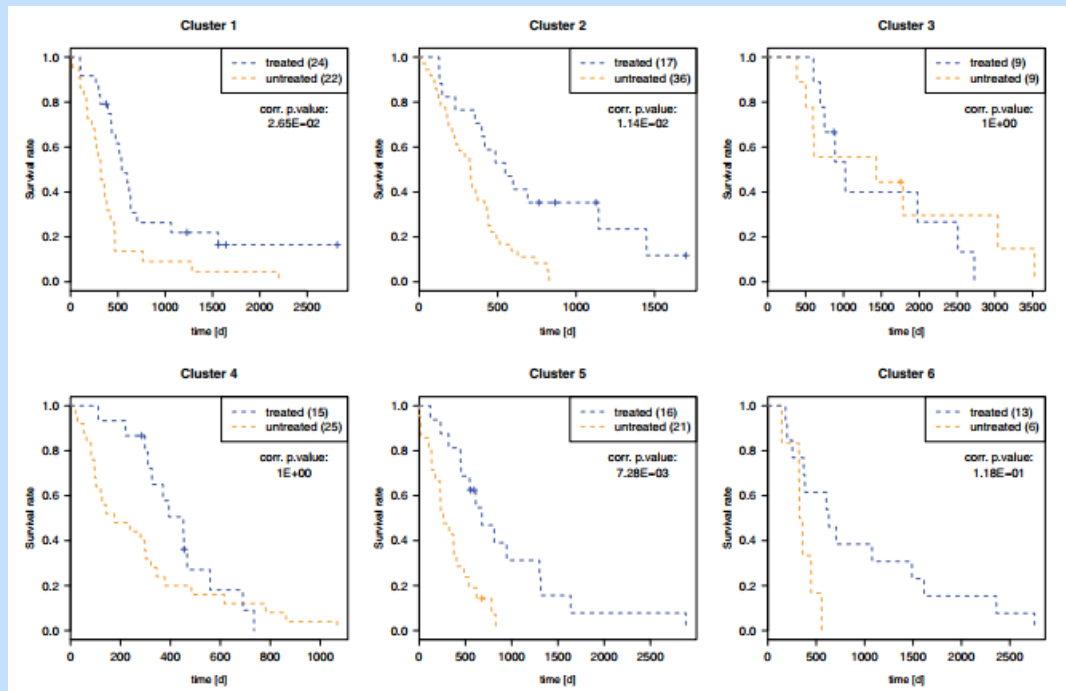
MKL - Results

- Method's robustness
- Leave-one-out clustering: each patient is left out, the algorithm is run, and then the patient is added by projecting it and adding to the cluster with nearest mean
- Left - 3K vs 15k. Right - β sums to 1 constraint



MKL - Results

- Survival analysis does not consider the treatment given
- Response to Temozolomide (chemotherapy drug used for brain cancers) within different clusters



MKL - Recap

- Similarity based
- Graph embedding: dimension reduction such that neighbors in the original dimension remain close in the low dimension
- Use the kernel trick + different kernel(s) for each omic
- Compare prognosis to SNF and show the effect of multiple kernels on robustness

Summary

- Omic
- Multi omics data
- In this talk - methods that apply to numerical omics
- COCA - late integration
- Shared subspace models:
 - iCluster - probabilistic linear model
 - NMF - factorization with non negativity constraints

Summary

- Shared subspace models:
 - iCluster - probabilistic linear model
 - NMF - factorization with non negativity constraints
- Similarity based models:
 - Similarity network fusion - creating a unified similarity network
 - Multiple kernel learning - using different kernels for each omic
- Complexity and non-numerical omics vs. analysis of the feature within the model
- (Do different omics share the same underlying clustering?)

FIN

