Title: Patient stratification of cancer via integrated genomic and transcriptomic profiles coupling with network-aware clustering

Name: Yang Pan UID: 504567911

Background and Significance

For decades, stratifying patients based on physiological and molecular measurements is one of the key biomedical question. With the concept of Precise Medicine (PM) becomes popular, Stratified Medicine (SM) has been recently discussed by Willis and Lord (1) in a review paper in 2015. Cancer, as a complex (in mechanism) and heterogeneous (among patients) disease, can be better understood and cured by gaining insights of the hidden samples structures driven by shared underlying mechanisms. Patient stratification can be considered and solved as an unsupervised clustering problem.

In reviewing of stratification methodology development papers and recent cancer studies, we found many efforts have been made on patient stratification in cancer studies based on individual mutations (2), combination of mutated genes (3), expression profiles (4), or the combination of several data types (5, 6). In terms of clustering approach, hierarchical clustering (HC) is commonly seen in many cancer studies(7) like TCGA, it is simple to perform but insufficient to capture local similarities among a subset of subjects as it based on global similarities. Other advanced unsupervised methods including probabilities models (6, 8), factor analysis (9), SVD (10) and self-organizing map (SOM) (11) are also employed to this problem, however they suffer various limitations like instable result, single data source and so on.

The consensus from those papers is that a desirable stratification should be comprised of 1) an efficient clustering method, 2) a comprehensive integration of different level of molecular data and 3) a proper use of biological networks for interactions among molecular features. However, there is no work combining three of those important aspects, including the most recent or state-of-art developments: A recent deep learning-based work developed a deep belief network (DBN) (12) to stack multiple data types, however it ignores biological networks when constructing abstract features. Another popular method, called iCluster (9), also provides good data integration but ignores the feature correlations like gene-gene network. A successful work based on non-negative matrix factorization (NMF) named NBS (13) provides a network-aware clustering. However, it only uses mutation data which limits the comprehensiveness of stratifying patients.

To our best knowledge, there is no existing model that provides both integration of mutation and expression data and leveraging network of molecular feature interaction for patient stratification problem. Hence, here we proposed a graph-regularized multiple NMF-based model (m-gNMF), performing patient stratification based on patterns from combination of mutation profile and gene expression profile (could be other data types as well) coupling with known protein-protein network. On the top of that, technically we evaluated if adding a smoothing matrix to gNMF (ns-gNMF) could help to create sparseness and allowing a compact clustering between molecular signatures with patients, which might help us to increase the stableness and accuracy. We decided to evaluate our m-gNMF and ns-gNMF model separately for simplicity and clarity.

In the study, we first generated simulated data to evaluate our proposed methods, comparing them with existing state-of-art NMF-based methods. After sufficient simulation tests, we found that by integrating two input datasets, our method m-gNMF outperforms single input gNMF (NBS) by a large margin under the condition that two input matrices support similar patient structure. In another experiments, we also realized that our ns-gNMF had no superiority over gNMF in terms of accuracy, thus we did not add it to later experiments. Finally, we analyzed real cancer somatic mutation and expression dataset from TCGA with our m-gNMF model.

Methods

Data collection

To test the performance of the proposed algorithms using real data, we downloaded cancer somatic mutation and expression dataset from ovarian cancer (OV) study of TCGA. Only non-synonymous mutation is considered. Normalized RNA-seq expression data is used. Along with the molecular datasets, clinical data of each patient are also retrieved. Filtering patients with very few mutations (less than 10) resulted a dataset with 356 patients.

A STRING database derived protein-protein interaction (PPI) network is directly downloaded from NBS package's dataset. In this way we can keep network parameters and properties consistent with NBS paper, which helps this study focusing on clustering methods comparison.

Data simulation

To test the performance of the proposed algorithms via simulation data, we generated simulated patient-gene mutation matrix based on real biological data and known PPI network. This simulation is based on NBS simulation pipeline(13), which is driven by the assumption that patients with same cancer type can be stratified into small numbers of subtypes driven by a set of genes in one or just few pathways/modules.

For this study, we simulated binary mutation matrix with driver mutation frequency from 2.5%, 5%, 7.5%, 10.0% and 12.5% with a single network module assigned to each subtype. The driver frequency is decided according to the evidence that the real frequency is around 4% (2-8 mutations) in cancer data (14). Other parameters related to generating the network are set to default as suggested by previous studies(13). The simulated datasets in this paper are all with 200 patients, 8000 genes with 0 or 1 mutation measurements, 4 clusters where each cluster associated with a single network module of size 30-50 genes.

For testing multiple input NMF, we used the pipeline to simulate two patient-gene matrixes with same patient stratification (Figure 3, Top box) and two patient-gene matrixes with different patient subgroup assignments (Figure 3, Bottom box). In real world, the level of discordance of the two inputs are varied. We simulate the two extreme cases to know the performance of proposed method in the best and worst case scenarios.

Mutation matrix smoothing

Both real and simulated mutation cohort data can be considered as a binary matrix where 0 represents no mutation on a specific gene while 1 represents the occurrence of one or more mutation on a gene. Using this mutation-patient binary matrix to perform stratification can be ineffective because gene pattern overlap among patients can be small. Ideally, patients with distinct mutated gene should be stratified as same subtype if those genes are in fact in the same pathway or gene module. Mutation matrix smoothing is proposed in NBS paper(13) to solved this. It uses network propagation technique (*15*) to propagate mutation signal along the connected genes based on the known gene or protein network. The network based mutation smoothing increases the accuracy of patient stratification greatly according to NBS paper. It takes a very sparse binary gene-patient binary matrix and returns a relatively smooth matrix where mutated genes and their neighboring genes having value from 0 to 1 based on distance and network topology. Leveraging their toolbox, we generated smoothed matrix for all our simulated and real mutation matrixes. The propagation follows suggested parameters in their original paper.

Non-smooth graph-regularized NMF (ns-gNMF)

Based on Pascual-Montano's paper of original nsNMF (16), we proposed ns-gNMF, which is implemented with both python and matlab code (extended from 'Nimpy' module and 'netnmf' module). The proof of converge and multiplicative update rule are similar to original NMF paper(17).

Objective function is presented in Euclidean distance form:

$$O = \|\boldsymbol{V} - \boldsymbol{WSH}\|^2 + \lambda Tr(\boldsymbol{HLH}^T)$$

where the ***V*** is a *m*n* gene-patient matrix for an *m* patient cohort with *n* genes mutation/expression measured. ***V*** can be approximately factorized into two matrices ***W*** (*m*k*) and ***H*** (*k*n*), containing information of gene's clusters and patient's stratification respectively. *k* is the number of clusters, which can be defined by user. ***L*** is the graph Laplacian, which can be calculated from PPI network adjacency matrix ***A*** and corresponding degree matrix ***D***, ***L*** = ***D*** – ***A***. In this case, we also have a matrix controlling the sparseness of the factorization result, called smoothing matrix ***S***, which is in following form:

$$\boldsymbol{S} = (1 - \theta)\boldsymbol{I} + \frac{\theta}{q}\boldsymbol{11}^T$$

Where θ is a value in (0, 1) and often needed to be determined by data, ***I*** is an identity matrix and ***1*** is vector of ones.

Following the multiplicative updating rule to achieve the local minima:

$$w_{ij} \leftarrow w_{ij}\frac{(\boldsymbol{VH}^T)_{ij}}{(\boldsymbol{WHH}^T)_{ij}}$$

$$h_{ij} \leftarrow h_{ij}\frac{(\boldsymbol{W}^T\boldsymbol{V} + \lambda\boldsymbol{HA})_{ij}}{(\boldsymbol{W}^T\boldsymbol{WH} + \lambda\boldsymbol{HD})_{ij}}$$

Since currently there is no theoretical way to find value of λ, we empirically estimated it by trying a series of values like 0.1, 10,100,1000, etc.

Multiple graph-regularized NMF (m-gNMF)

Inspired by jNMFMA (18, 19) and GNMF (20), we formulized m-gNMF, which is implemented with both python and matlab code (extended from 'Nimfa' module and 'SNMNMF' module, respectively). The proof of converge and multiplicative update rule are very similar to original NMF paper(17).

Objective function:

$$O = \|\boldsymbol{V}_1 - \boldsymbol{W}\boldsymbol{H}_1\|^2 + \|\boldsymbol{V}_2 - \boldsymbol{W}\boldsymbol{H}_2\|^2 + \lambda_1 Tr(\boldsymbol{H}_1\boldsymbol{L}_1\boldsymbol{H}_1^T) + \lambda_2 Tr(\boldsymbol{H}_2\boldsymbol{L}_2\boldsymbol{H}_2^T)$$

where $\boldsymbol{V}_i$ (i=1,2) are the input patient-gene matrix, just as we described above. Following the same notation we defined above, $\boldsymbol{W}$, $\boldsymbol{H}_i$ (i=1,2) are the factorized results representing gene or patients subgroups. $\boldsymbol{L}$ is the graph Laplacian calculated from the known PPI network.

Following the multiplicative updating rule to achieve the local minima:

$$w_{ij} \leftarrow w_{ij} \frac{(\boldsymbol{V}_1\boldsymbol{H}_1^T + \boldsymbol{V}_2\boldsymbol{H}_2^T)_{ij}}{(\boldsymbol{W}\boldsymbol{H}_1\boldsymbol{H}_1^T + \boldsymbol{W}\boldsymbol{H}_2\boldsymbol{H}_2^T)_{ij}}$$

$$h_{ij}^1 \leftarrow h_{ij}^1 \frac{(\boldsymbol{W}^T\boldsymbol{V}_1 + \lambda_1\boldsymbol{H}_1\boldsymbol{A}_1)_{ij}}{(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}_1 + \boldsymbol{H}_1\boldsymbol{D}_1)_{ij}}$$

$$h_{ij}^2 \leftarrow h_{ij}^2 \frac{(\boldsymbol{W}^T\boldsymbol{V}_2 + \lambda_2\boldsymbol{H}_2\boldsymbol{A}_2)_{ij}}{(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}_2 + \boldsymbol{H}_2\boldsymbol{D}_2)_{ij}}$$

The estimation of λ follows the method described under ns-gNMF section.

Existing NMF algorithms used for comparison

In paper we compared our methods with existing NMF based metods. For standard NMF, we used 'Nmf' class from Nimfa package (python implementation), and nnls class from Li's NMF Toolbox (matlab implementation). For graph-regularized nmf, we used 'netnmf' class from NBS (matlab), 'GNMF' class from Cai's released code (matlab), and a 'Gnmf' class (python) I implemented based on Nimfa's framework (see manifest of submitted code).

Consensus Clustering

The stochastic nature of NMF-based methods require consensus clustering to promote result's robustness and evaluate the stability (21) . In this paper, as a default, all the NMF based clustering algorithm was performed with multiple runs to stratify input cohort into subgroups. More specifically, the clustering algorithm was performed 50 to 100 times and all samples and all genes are selected for each run with randomized initiation (no subsampling). As a way of evaluating stratification's stability, clear blocks observed in consensus clustering represent stable clusters. Both our implemented python and matlab codes support consensus clustering. The hierarchical clustering (HC) heatmap is created by R script.

Rand index (RI) and Adjusted Rand Index (ARI)

In this paper, clustering accuracy is reported as the Adjusted Rand Index (ARI) (22) between the known stratification of simulation and the stratification generated by our method. Adjusted

Rand Index (ARI) calculates the overlap between two sets with the consideration of effect of randomness. The original Rand Index (RI) can be value from 0 to 1, representing from completely no overlap to perfect overlap. RI is simply the number of agreements between two stratifications divided by n select 2, where n is the total number of elements. ARI, instead, takes into account of the situation that the index is less than the expected index. Therefore, the score can be lower than 0, meaning it is less accuracy than expected random assignments.

$$ARI = \frac{Index - Expected\ Index}{Max\ Index - Expected\ Index}\ .$$

We adopted NBS's matlab function to calculate ARI and RI.

Sparseness

Sparseness of a matrix is defined as the average sparseness of its column vectors. The value of sparseness is from 0 to 1. Sparseness is 1 for the case where the vector contains a single nonzero component and is 0 when all components of the vector are equal. We directly used sparseness() function in Nimfa module in python.

Cophenetic correlation coefficient

Cophenetic correlation coefficient is a stability measurement of the consensus matrix of multiple NMF runs. It is based on the average of connectivity matrices. As a model selection method, Brunet (21) first proposed to observe this coefficient as factorization rank increases and to choose the first rank where the magnitude of the coefficient begins to fall. We used 'scipy.cluster.hierarchy' class to calculate Cophenetic correlation coefficient.

Survival Analysis

The clustering result from our proposed method was combined with survival time information downloaded from TCGA. The R package 'survival' was used to generate Kaplan-Meier plot and conduct log-rank test.

Results

Evaluating the performance of a non-smoothed graph regularized NMF (ns-gNMF) on simulated data

First we want to examine whether adding a smoothing matrix will lead to a better performance than standard NMF and graph regularized NMF.

Before we compare different methods, we conducted multiple runs to find the proper parameter for gNMF and ns-gNMF.

In gNMF, λ is key parameter and controls the influence of graph regularization term. As described in the Methods section, empirical estimation of λ for all of our simulation data is in the magnitude of 10^2 to 10^3, which is the ratio of two terms in objective function. Following GNMF paper (*20*), we further tested the influence of λ with value 0, 10, 100, 1000, 10000

(Figure 1A) on a smoothed simulation data. We then used clustering accuracy as a benchmark to evaluate the suitable λ. Clustering accuracy is evaluated by the overlap between our stratification results and known patient subtype assignment. In order to eliminate the impact of overlaps caused by randomness, we use Adjusted Rand Index (ARI) as the measurement. As a result, we choose λ=1000, as it is better than 10000 and slightly better than 10, 100, and standard NMF with no regularization. This is interesting to observe, because under exact settings with NBS (the same PPI and simulation parameters), we found the greatest improvement shown in NBS paper is not due to the gNMF model they implemented, but the network-based smoothing step. In fact, network regularization only improves clustering accuracy by 1.3% comparing to standard NMF.

A key parameter for our ns-gNMF is θ, which controls the values in smoothing matrix and promotes the sparseness of result matrices. However, there is also no theoretical and efficient way of searching best θ value. In original nsNMF (16) paper, authors suggest trying out values from 0.1 to 0.9. We used a smoothed simulation data with driver frequency at 10% to ensure a good clustering accuracy (for justification, see Figure 2C), and tried six θ values. As shown in Figure 1B, accuracy stays roughly same from 0.1 to 0.3 and decrease dramatically when θ goes higher. At the same time, sparseness and converging speed increases when θ becomes larger. Since for our task of patient stratification, the accuracy matters most, we decided to choose 0.1 as θ value for rest of the experiments. λ in ns-gNMF was estimated by the same way described above and set as 1000.

Figure 1

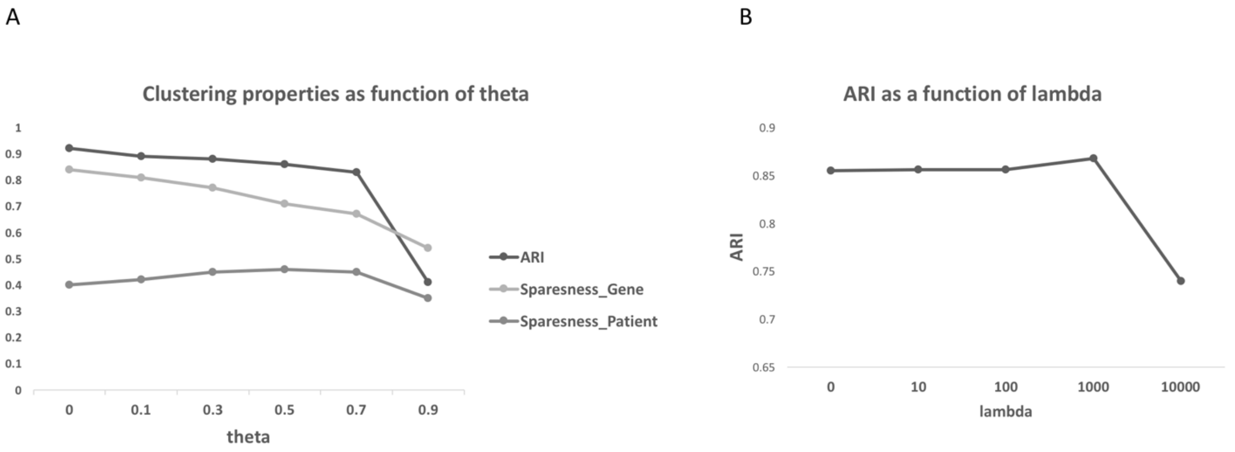

Figure 1. Determine model parameters for simulation test. A) Using an accuracy (Adjusted Rand Index, ARI, see Methods) as a criterion to choose θ. Matrix sparseness for both W (gene-subgroup) matrix and H (patient-subgroup) matrix are calculated. The simulation dataset has driver frequency 0.1. B) Using our accuracy (Adjusted Rand Index, ARI, see Methods) as a criterion to choose λ. The simulation dataset has driver frequency 0.075. A, B) Consensus clustering of 50 runs is performed to generate all above values.

Finally, after fixed model parameters, we began to compare our ns-gNMF with state-of-art NMF based methods using simulated data. To do that, we simulated five patient-gene mutation matrix with driver mutation frequency from 2.5% to 12.5% (Details see Methods), and run three versions of NMF-based patient stratification on simulated data (Figure 2). Panel A shows the convergence of both ns-gNMF and gNMF. It is clear that the residual of the cost function starts to become stable after first 50 iterations while the max iteration is set to 500. Fixing driver mutation frequency at 7.5%, we further tested consensus clustering stableness (Figure 2B). In panel B, standard NMF represents original NMF clustering on binary matrix without smoothing, NBS represents gNMF clustering on smoothed mutation matrix, ns-gNMF represents ns-gNMF clustering on smoothed mutation matrix. As seen from panel B, comparing to standard NMF with binary input, gNMF with smoothed matrix (NBS) and ns-gNMF give more stable stratification result among each runs. Standard NMF with binary mutation input can only give reasonable stable result when driver mutation frequency as high as 10%, which is usually not the case. Being consistent with previous NBS study, gNMF provides a stable result when driver mutation frequency higher than 7.5%. Our ns-gNMF shows nearly identical performance. Panel C illustrates the adjusted Rand index as a measure of clustering accuracy. As expected, to reach the similar high accuracy, gNMF and ns-gNMF lowering the requirements of driver mutation frequency from standard NMF's 12.5% to 7.5%, which echoes NBS results. Note that we added a discontinuous line representing original NMF with smoothed mutation data, which shows a little less accuracy than gNMF. The result replicates what we observed during selecting of λ. Our proposed ns-gNMF proposed shows no significant difference, but slightly less accurate than gNMF. In general, our proposed ns-gNMF performs similar to gNMF under θ value 0.1. They have simaler convergence speed, similar stability during multiple randomized initiation runs. There are several possible explanations about the mediocre performance of ns-gNMF in discussion section.

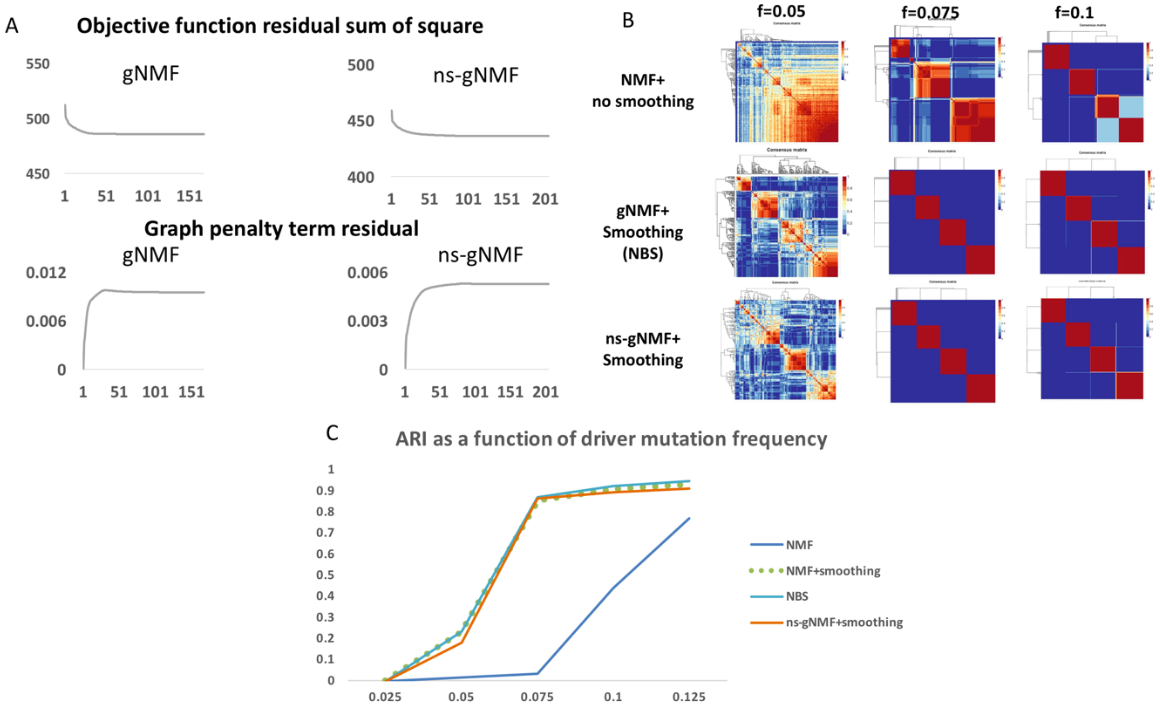

Figure 2. Simulation test for ns-gNMF. A) Residual sum square of objective function and the graph term during the update to convergence. Max iteration is set to 50. B) Consensus matrix of three different methods, and three different driver mutation frequency cohorts. C) Accuracy (ARI) as a function of driver mutation frequency. Comparing different methods under different driver mutation frequency cohorts.

Evaluating the performance of multiple input graph regularized NMF (m-gNMF) on simulated data

First of all, we want to point out this simulation is designed to test performance of multiple input gNMF, not necessary for mutation or expression. We would like to assume no matter what type of measurements (expression, methylation, protein), after proper transformation, it will be similar to a smoothed mutation matrix, where only a small set of genes in small number of pathways are drivers and of high values, rest of genes are passengers and of lower values (mean is lower).

In order to evaluate the benefits of stacking multiple molecular profiles together to perform patient stratification, we experimented following tests based on different sets of simulation data controlling cohort subtype structure and disease driver gene frequency (Figure 3). We simulated two major extreme conditions: Condition A, two inputs (namely input 1, input 2) have the same underlying patient stratification. Condition B, two inputs have completely different patient stratification, which is achieved by permuting input 1 or input 2 generated in condition A. Note that though input 1 and input 2 share same cohort stratification, they are driven by different gene clusters (orange and green), which resembles in real world driver mutated genes and highly expressed genes are not coming from same cluster in PPI network due to the complications caused by *trans* regulation and other mechanisms. Then for each for the condition we evaluated the performance by comparing clustering accuracy and stability.

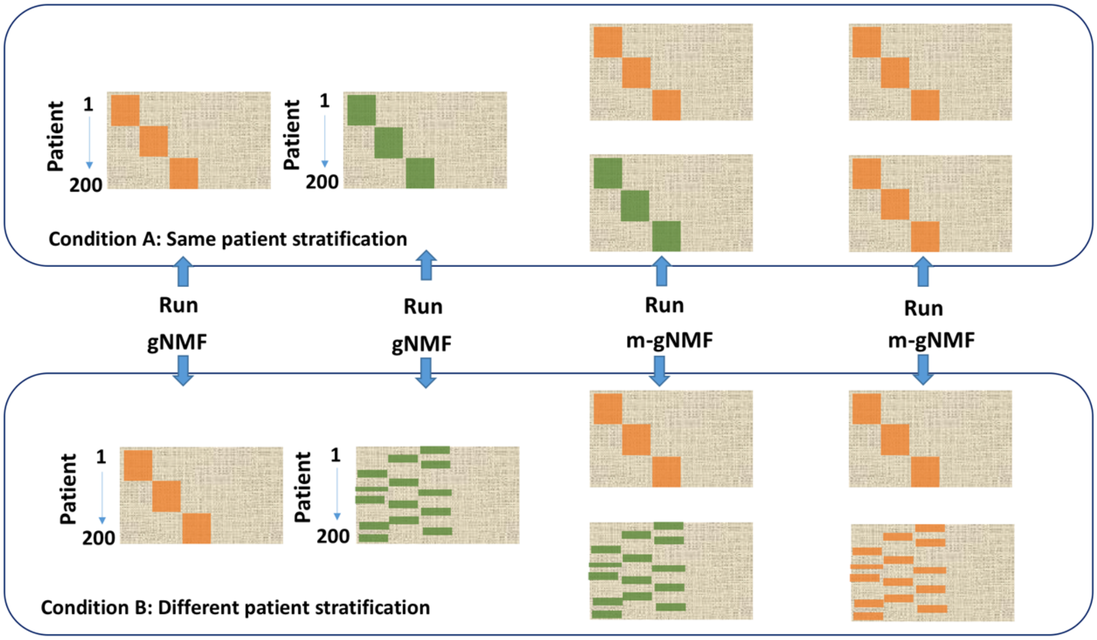
Patient
1
200
Patient
1
200
Condition A: Same patient stratification
Run
gNMF
Run
gNMF
Run
m-gNMF
Run
m-gNMF
Patient
1
200
Patient
1
200
Condition B: Different patient stratification

Figure 3. A demonstration of two simulated conditions to evaluate m-gNMF comparing to existing methods. In the top datasets, the two inputs are under same patient stratification, the bottom datasets two inputs are under different patient stratification. Patients are sorted in the same order from 1 to 200 in this illustration. Orange and green represents different set of driver genes.

In Figure 4, we first show the comparison result under the condition A, where the patients in the two inputs have same cluster membership yet different network modules (orange and green in Figure 3). The comparison is comprised of four methods, including two single input gNMF (equivalent to NBS), combined different inputs m-gNMF, and a control m-gNMF with two same inputs. The comparison is between single gNMF of each of the input matrix (smoothed, applies to all experiments below), m-gNMF with the two input matrixes, and a control that m-gNMF takes two identical matrixes. Parameter λ for gNMF was estimated again as 10^3 while for m-gNMF was 10^-2 for both inputs. Under different driver mutation frequency settings, m-gNMF consistently shows significant higher accuracy than either single gNMF or m-gNMF with two identical matrixes. This indicates the integration of two matrices can be greatly helpful under the current simulation conditions. The improvement of m-gNMF is rather significant in driver mutation rate of 5% (ARI increased from 0.23 to 0.70), which further lowering the requirement of driver gene frequency to achieve accurate stratification. On the other hand, for our control run, m-gNMF with two identical inputs returns almost same clustering accuracy of using single gNMF on that input, which confirms the increase of accuracy is due to the integration of two different matrixes supporting same patient structure. The possible reason to cause this boosting of accuracy is very similar to add a known graph regularization for data from manifold structure. In this case, each input can be regarded as a customized regularization for another. We will discuss this more in discussion section.

In figure 5, we simulated another extreme case where the two input matrixes do not support the same patient stratification. We used the same input 1 and input 2 generated above, but this

Figure 4

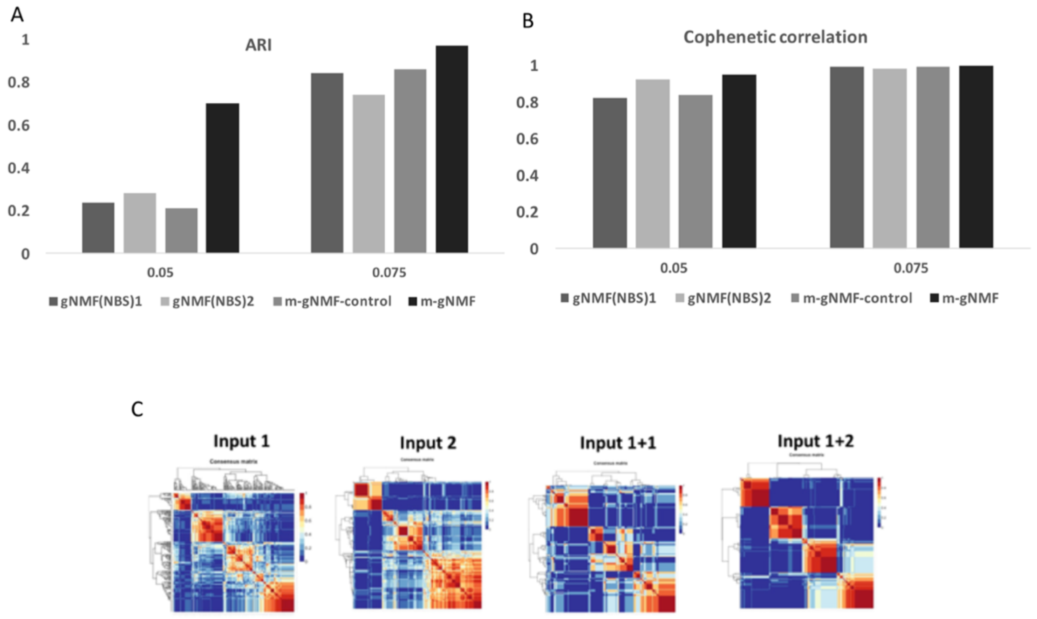

Figure 4. Simulation test under condition A (two inputs with same patient stratification). A) Clustering accuracy (Adjusted Rand Index, ARI) of four different combinations with different driver frequency. B) Clustering stability (Cophenetic correlation coefficient) of four different combinations with different driver frequency. C) Consensus matrix (200*200) of four different combinations with driver frequency 5%. A,B,C) All values are calculated based on consensus clustering of 50 times. Details for consensus clustering see Methods.

time we permutated each of them by patient and combined them in different ways (Figure 5). Specifically, the comparison is between single gNMF of each of the input matrix, m-gNMF with the two input matrixes, and a control that m-gNMF takes two originally identical matrixes yet one is permutated by patients. First by looking at consensus matrixes, it is easy to find though the permutation has no effect on clustering single input cohort as we expected, it does harm the m-gNMF (the bottom two). And the control test shows lowest stability. This can be explained by accuracy (Table 1). Notably, it seems there is always a matrix dominant the clustering result in table 1. Back in figure 4A we can see input 1 has better clustering accuracy than input 2. So after shuffling, the integrated result always follows input 1's patient stratification. This can be clearly seen in table 1. When input 2 is permutated, the m-gNMF result's accuracy is 0.79 (ARI), while input 1 is permutated, the combined result follows the permutated patient assignments, which leads to 0 ARI. Note that ARI is adjusted for the

Table 1

	Input 1	Input 2	Input 1 + Input 2 shuffle	Input 2 + Input 1 shuffle	Input 1+ Input 1 shuffle
ARI (relative to Input 1's underlying structure)	**0.84**	**0.74**	**0.79**	**0.00**	**0.46**

Table 1. Simulation test under condition B. Accuracy (ARI) of each gNMF or m-gNMF result comparing to input 1's patient stratification. Consensus clustering with 50 times run. Simulation input 1 and input 2 are under driver gene frequency 7.5%.

Figure 5

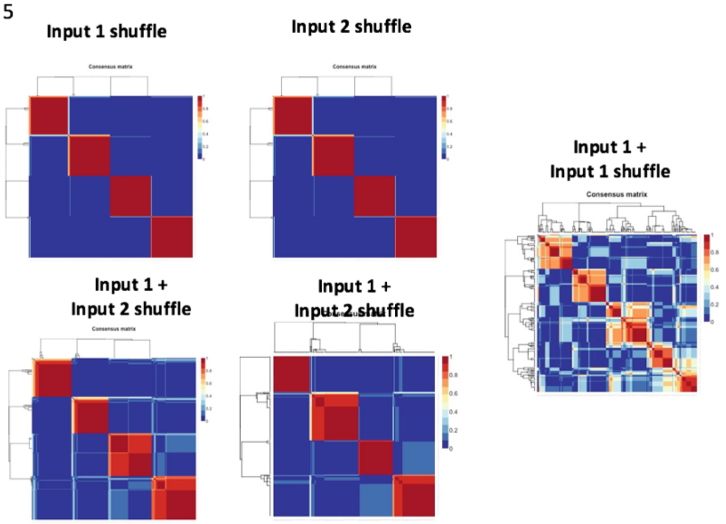

Figure 5. Simulation test under condition B. Consensus matrix of each gNMF or m-gNMF result with different combination of inputs. Consensus clustering with 50 times run. Simulation input 1 and input 2 are under driver gene frequency 7.5%.

expected random similarity, so 0 does not mean no overlap but stands for unadjusted accuracy around 50%--randomly assigned. The control case has no dominant input, thus gives worst accuracy (ARI 0.46).

Evaluating the performance of multiple input version of gNMF (m-gNMF) on real cancer data

Based on above simulation result, we decided for the real data, we will continue with our m-gNMF to compare with existing methods and skip the non-smooth technique. This is because m-gNMF shows positive results and improved accuracy comparing the existing gNMF methods, while ns-gNMF shows no added value comparing to existing gNMF.

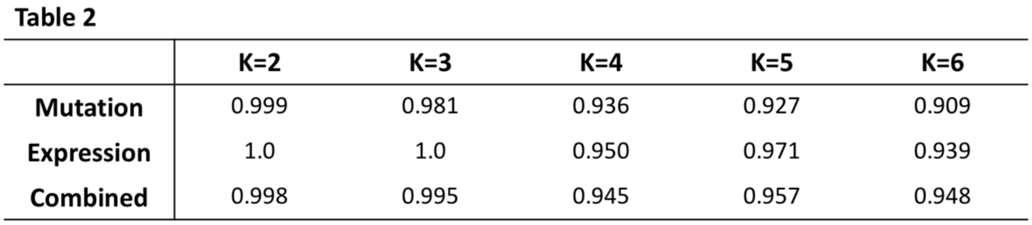

Table 2

	K=2	K=3	K=4	K=5	K=6
Mutation	0.999	0.981	0.936	0.927	0.909
Expression	1.0	1.0	0.950	0.971	0.939
Combined	0.998	0.995	0.945	0.957	0.948

Table 2. Clustering stability (Cophenetic correlation coefficient) of consensus clustering result based on TCGA OV mutation and expression data. Each consensus matrix is generated based on 50 runs.

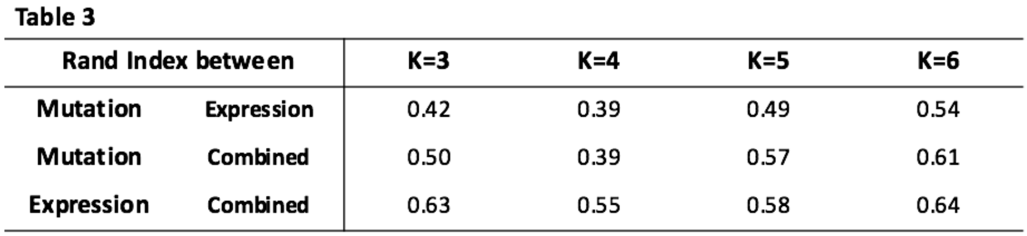

Table 3

Rand Index between		K=3	K=4	K=5	K=6
Mutation	**Expression**	0.42	0.39	0.49	0.54
Mutation	**Combined**	0.50	0.39	0.57	0.61
Expression	**Combined**	0.63	0.55	0.58	0.64

Table 3. Similarity (Rand Index, RI) between stratification results of different methods under different K. 'Mutation' represents smoothed somatic mutation clustered by gNMF, 'Expression' represents normalized gene expression data clustered by gNMF, 'Combined' represents using m-gNMF clustering mutation and expression together. Consensus clustering are done with 50 runs each.

Unlike the simulation data that we always know the best number of clustering (K), for real data set like this OV-TCGA dataset, we had to performed model section for K. Here we launched consensus clustering with different value of K, from 2 to 6. In Table 2, the clustering stability (Cophenetic correlation coefficient) was measured for both single input gNMF and two inputs m-gNMF with smoothed mutation and normalized expression data are shown with representing the stability of clustering. In this case, though there is a significant decrease of the coefficient at K=4, it is nor safe to select K=3 as the best number of cluster. The reason is in this case, from K=2 to K=6, Cophenetic values are all higher than 0.9, meaning all those consensus matrixes are of good stability. So Cophenetic coefficient alone is not sufficient to determine the value of K and it is helpful to look at the actual consensus matrix to see what are those clusters look like (Figure 6B). We observed 2~3 relatively stable clustering from K=4 to K=6, suggesting a relatively stable stratification. Then we considered mutual overlaps among different pairs of stratification (Table

3). We found K=5 and K=6 both maximized the overlaps between three outputs. Thus we decided to use K=5 to further evaluate our m-gNMF stratification.

There is no objective way to measure the accuracy of patient stratification for real data, but leveraging clinical information like survival time is common practice. The assumption is the patient's molecular features are closely related to the disease phenotype, so difference in survival time of cancer patient should be able to be separated based on molecular features. The same assumption applies to survival time. As we seen from figure 6A, under K=5, there are two major subgroups of patients are separated. Other subgroups are of relatively small number of patients. The log-rank test yields p-value 0.007 meaning the stratification generated by m-gNMF with mutation and expression is statistically significant.

Figure 6

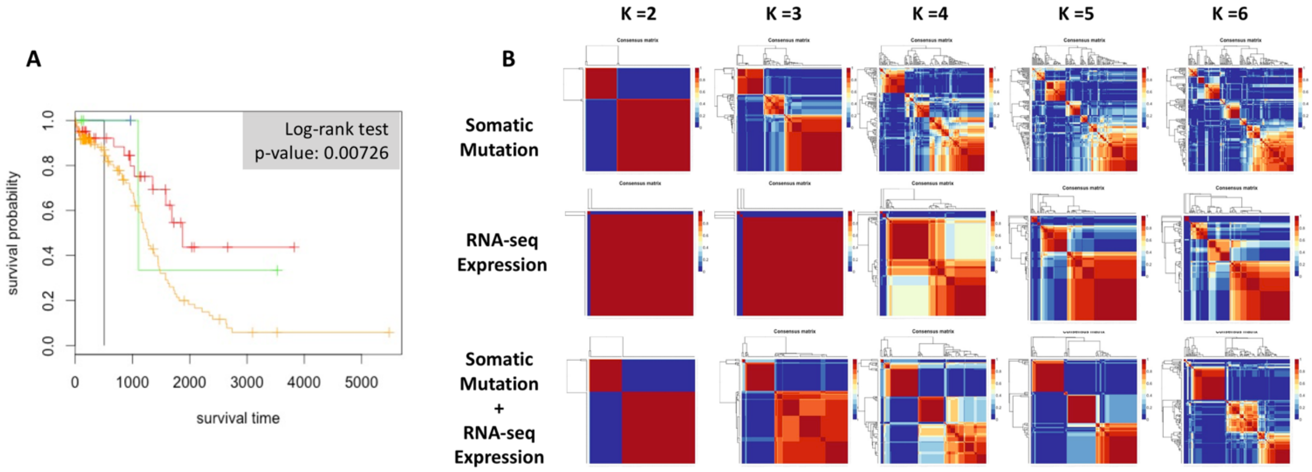

Figure 6. Mutation and expression based m-gNMF on ovarian cancer dataset. A) Survival analysis for stratification generated by m-gNMF with mutation end expression. K equals 5. B) Consensus matrices for gNMF and m-gNMF with different inputs with different K values.

Discussion

Strength and weakness of proposed methods

From the results above, we gained insights about the possible strength and weakness of our proposed integrated biology network-aware NMF-based patient stratification method.

First, it is obvious the integration of multiple inputs (m-gNMF) having a good chance to outperform the current single input NBS method in terms of clustering accuracy. Under the proper condition, it will increase the stratification accuracy and lowering the requirements of driver gene frequency. One of the main reason caused this improvement is that integrating multiple inputs is very similar to adding certain regularization on patient matrix (H). Especially for the case where the same stratification of cohort is shared by several inputs. In some sense, this is even better than adding PPI network regularization because this information is more customized and tailored for the specific cohort, not like the generic PPI network. However, the limitation is also apparent, it must under the condition that the two data types/inputs support similar patient stratification, which is normally not the case in real world problem. Another limitation lies in the graph. In fact, it is not an easy task to get a high quality, precise graph. It is, more often than not, we have the knowledge about the manifold structure the data coming from. This aspect will be expanded later.

Second, it is also clear that the non-smooth NMF technique is not suitable to be directly applied to NMF clustering with graph regularization. It leads to no significant changes on clustering stability and minor decrease of stratification accuracy. The reason behind the mediocre performance of ns-gNMF may due to three reasons: 1) from the residual plots we can see, nsNMF causes a sparser H matrix, which decreases the effect of graph regularization, leading to slightly worse accuracy. 2) Even in original nsNMF paper, the effect of smoothing matrix ***S*** on sparseness is not linear (they reached high sparseness at θ=0.5, θ=0.8). The sparseness peak is sensitive on data. It is possible in our case there is no peak and 0 gives highest sparseness. 3) There is no clear connection between sparseness and clustering accuracy.

Third, the test of m-gNMF on real TCGA cancer data shows promising results. It got relatively stable clusters among different K in range 4 to 6, and also got good overlap among different methods (gNMF with mutation, gNMF with expression). In addition, the stratification by m-gNMF shows significant difference on patient's survival time. The limitation of this method in real case is that the parameter lambda for graph regularization is hard to select. In the paper, we directly used parameters from simulation result, but it may not be suitable for other cases.

Future works

Beside above major conclusions from this study, there are several aspects demanding future developments.

First, the speed of m-gNMF. This can generally apply to most variations of NMF based algorithms. The stochastic nature of NMF and multiplicative updating based method takes a lot time to run and test. Lin (23) proposed an efficient way to promote the speed and convergence. It is possible and will be a good improvement to our current method.

Second, the correctness of knowledge of biological network. For graph regularized NMF, the performance relies on the quality of the graph. In our case, if the PPI network can faithfully reflect the biological network that the input data (eg. Mutation, expression, methylation) coming from, we can expect gNMF giving a reasonable accuracy. However, it is well known that PPI network are prone to false positive connection. It is also interesting to notice that some works of NMF clustering on multiple manifolds (24), by which we could considering integrating more possible interaction networks (eg. DNA structural level).

Third, the estimation of m-gNMF parameter λ. Unlike in simulation data, where λ is often determined by clustering accuracy, it is inconvenient to estimate it in real data. It's not trivial task, especially because this parameter is data sensitive can be as high as 10^3 or as low as 10^3 in previous studies (13, 19).

Last but not least, given our m-gNMF framework, we can easily extend to integrate more data types and related networks for comprehensive stratification.

Justification of minor deviation from proposal

A minor deviation: As mentioned at the end of introduction, I decided to separate my proposed method into m-gNMF and ns-gNMF, to evaluate them separately for simplicity and clarity.

Reference

1. J. C. Willis, G. M. Lord, Immune biomarkers: the promises and pitfalls of personalized medicine. *Nat Rev Immunol* **15**, 323-329 (2015).
2. A. M. Dulak *et al.*, Exome and whole-genome sequencing of esophageal adenocarcinoma identifies recurrent driver events and mutational complexity. *Nat Genet* **45**, 478-486 (2013).
3. D. Bertrand *et al.*, Patient-specific driver gene prediction and risk assessment through integrated network analysis of cancer omics profiles. *Nucleic Acids Res* **43**, e44 (2015).
4. N. Cancer Genome Atlas, Comprehensive molecular characterization of human colon and rectal cancer. *Nature* **487**, 330-337 (2012).
5. S. Zhang *et al.*, Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Res* **40**, 9379-9391 (2012).
6. D. Y. Cho, T. M. Przytycka, Dissecting cancer heterogeneity with a probabilistic genotype-phenotype model. *Nucleic Acids Res* **41**, 8011-8020 (2013).
7. C. Kandoth *et al.*, Mutational landscape and significance across 12 major cancer types. *Nature* **502**, 333-339 (2013).
8. S. Khakabimamaghani, M. Ester, Bayesian Biclustering for Patient Stratification. *Pac Symp Biocomput* **21**, 345-356 (2016).
9. R. Shen, A. B. Olshen, M. Ladanyi, Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* **25**, 2906-2912 (2009).
10. J. Sun, J. Bi, H. R. Kranzler, Multi-view singular value decomposition for disease subtyping and genetic associations. *BMC Genet* **15**, 73 (2014).
11. E. M. Borkowska *et al.*, Molecular subtyping of bladder cancer using Kohonen self-organizing maps. *Cancer Med* **3**, 1225-1234 (2014).
12. M. Liang, Z. Li, T. Chen, J. Zeng, Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach. *IEEE/ACM Trans Comput Biol Bioinform* **12**, 928-937 (2015).
13. M. Hofree, J. P. Shen, H. Carter, A. Gross, T. Ideker, Network-based stratification of tumor mutations. *Nat Methods* **10**, 1108-1115 (2013).
14. B. Vogelstein *et al.*, Cancer genome landscapes. *Science* **339**, 1546-1558 (2013).
15. M. D. Leiserson *et al.*, Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat Genet* **47**, 106-114 (2015).
16. A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, R. D. Pascual-Marqui, Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE Trans Pattern Anal Mach Intell* **28**, 403-415 (2006).
17. D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, 788-791 (1999).
18. H. Q. Wang, C. H. Zheng, X. M. Zhao, jNMFMA: a joint non-negative matrix factorization meta-analysis of transcriptomics data. *Bioinformatics* **31**, 572-580 (2015).
19. S. Zhang, Q. Li, J. Liu, X. J. Zhou, A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules. *Bioinformatics* **27**, i401-409 (2011).

20. D. Cai, X. He, J. Han, T. S. Huang, Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Trans Pattern Anal Mach Intell* **33**, 1548-1560 (2011).

21. J. P. Brunet, P. Tamayo, T. R. Golub, J. P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization. *Proc Natl Acad Sci U S A* **101**, 4164-4169 (2004).

22. W. M. Rand, Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* **66**, 846-850 (1971).

23. C.-J. Lin, On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks* **18**, 1589-1596 (2007).

24. B. Shen, L. Si. (2010).