intCC: An efficient weighted integrative consensus clustering of multimodal data

Can Huang and Pei Fen Kuan*

Department of Applied Mathematics and Statistics,
Stony Brook University,
Stony Brook, NY 11794, USA
**E-mail: peifen.kuan@stonybrook.edu*
www.ams.sunysb.edu

High throughput profiling of multiomics data provides a valuable resource to better understand the complex human disease such as cancer and to potentially uncover new subtypes. Integrative clustering has emerged as a powerful unsupervised learning framework for subtype discovery. In this paper, we propose an efficient weighted integrative clustering called intCC by combining ensemble method, consensus clustering and kernel learning integrative clustering. We illustrate that intCC can accurately uncover the latent cluster structures via extensive simulation studies and a case study on the TCGA pan cancer datasets. An R package intCC implementing our proposed method is available at `https://github.com/candsj/intCC`.

Keywords: Integrative clustering; Consensus clustering; Multiomics data; Ensemble learning.

1. Introduction

Recent advancements in high throughput technologies have enabled rapid profiling of different omics data, including genomics, epigenomics, transcriptomics, proteomics and metabolomics which allow for in-depth study of the complex regulatory patterns from a systems biology perspective. For example, the Cancer Genome Atlas (TCGA) has generated over 2.5 petabytes of multiomics data. Such datasets offer the opportunity to explore the heterogeneity underpinning diseases such as cancer via unsupervised learning based on clustering framework, which could help define cancer subtypes, bringing us a step closer towards personalized medicine.

In multimodal data structure, e.g., the different omics data, a key challenge in data analysis is in identifying the most appropriate approach for data integration. For unsupervised clustering over multimodal data, these include the choice of a single step *versus* two-step approach. A single step approach is also known as joint modeling which combines all datasets together. Two-step approach works by clustering each dataset separately, followed by integration of these clusters.

A number of integrative clustering methods and tools have been proposed to date. This includes Bayesian Consensus Clustering (BCC[1]), iCluster,[2] iClusterPlus,[3] Cluster Of Clusters Analysis (COCA[4]), Clusternomics[5] and kernel learning integrative clustering (KLIC[6]).

BCC, Clusternomics and iClusterPlus are based on Bayesian modeling framework and rely on Markov Chain Monte Carlo (MCMC) algorithm for fitting the model. These methods also assume that the probability model for each dataset is specified. However, softwares for BCC and Clusternomics currently only implement the algorithms for Gaussian distributed dataset, thus limiting the applicability of these methods to non-Gaussian datasets such as SNPs, mutation or copy number datasets.

On the other hand, iCluster works by assuming a Gaussian latent variable model for inferring the cluster structures, whereas iClusterPlus increases the versatility of iCluster by incorporating statistical models for continuous, binary, multinomial count datasets via a Bayesian latent variable model and employs MCMC algorithm for sampling from its posterior distribution for statistical inference. However, software implementation of iClusterPlus currently is limited to integrative clustering of at most four datasets. Since the model involves tuning a number of parameters, the bottleneck is the computational time when the number of datasets or features increases.

Another popular integrative clustering approach is COCA[4] which was first introduced to define cancer subtypes by clustering six different datasets, namely DNA copy number, DNA methylation, mRNA expression, microRNA expression, protein expression, and somatic point mutation. COCA works by first clustering each dataset using consensus clustering,[7] followed by clustering the binary matrix generated by aggregating the clusters obtained from each dataset. While this approach is robust and easily scalable to a large number of datasets, a limitation of COCA is that all datasets contribute equally to the final clustering which affects the accuracy of the clusters obtained, especially in scenario in which certain dataset is less reliable.

Taking inspiration from COCA and multiple kernel learning,[8,9] KLIC[6] was developed to address the pitfall of COCA. Similar to COCA, KLIC works by first applying consensus clustering to each dataset. The authors proved that these consensus matrices are positive semi-definite kernels, which can then be used as input in multiple kernel k-means clustering and allows for weights to be estimated for each kernel via a two-step optimization strategy and convex quadratic programming. This approach allows for more informative dataset to contribute more to the overall clustering. Currently, KLIC runs one clustering algorithm on each dataset to generate the consensus matrix.

In this paper, we seek to extend the KLIC framework to a more robust integrative clustering by proposing a two layer weighted integrative clustering which allows for more than one clustering algorithm to be run on each dataset, i.e, ensemble clustering and aggregated together via an efficient weight estimation.

2. Methods

Our proposed method can be viewed as a combination of (a) ensemble clustering, i.e, aggregating multiple clustering algorithms, (b) consensus clustering, i.e., resampling, and (c) kernel learning integrative clustering. While some papers use ensemble and consensus clustering interchangeably, in this paper, we refer to ensemble clustering as a collection of multiple clustering algorithms, e.g., k-means, hierarchical clustering or partitioning around medoid (PAM),

whereas consensus clustering as a framework which draws a random sample from either the sample or feature space. We now briefly describe the consensus clustering and kernel learning integrative clustering framework.

Consensus clustering was originally proposed by Monti et al (2003).[7] The main idea behind consensus clustering is to apply a resampling scheme on the sample or feature dimension under the assumption that different subsamples drawn from the dataset should not differ much in the clustering results. The resampling scheme allows one to assess the stability of the cluster assignments and the robustness of the dataset to perturbations, thus could aid in deriving a more stable and reliable result that reveals the real structure underlying the dataset.

A key element derived from the consensus clustering is the consensus matrix which measures the agreement among samples. For a dataset with N samples, the consensus matrix $\mathcal{M}$ is a $N \times N$ matrix whose element $M(i, j)$ denotes the proportion of sample i and sample j in the same cluster during the resampling iterations. Values which are close to 1 (and vice versa 0) indicate that the two samples are always assigned to the same cluster (and vice versa different clusters). $1 - \mathcal{M}$ is a distance measure which can be used to derive a final clustering result.

Cabassi and Kirk (2020)[6] proved that the consensus matrix is positive semi-definite and thus can be used as input in kernel learning integrative clustering via the application of multiple kernel k-means algorithm. The kernel k-means algorithm utilizes the kernel trick by projecting the data into a non-linear feature space via a kernel. This overcomes the drawback of regular k-means clustering which cannot identify clusters that are not linearly separable in the original input space. The integration of the multimodal data within the kernel learning integrative clustering involves a convex sum of the kernels, i.e., consensus matrix from each dataset, and the estimation of the weights in the convex sum. In the KLIC integrative clustering algorithm of Cabassi and Kirk (2020),[6] the authors adopted the optimization strategy proposed by Gonen and Margolin (2014)[10] which involves a convex quadratic programming.

In this paper, we reason that the weights in the kernel learning integrative clustering can be estimated by utilizing the fuzziness in the consensus matrix. Furthermore, we extend the framework of KLIC by allowing multiple base clustering algorithms, e.g., k-means, hierarchical clustering, PAM, to be applied within each dataset and aggregated, i.e., ensemble clustering[11] which has been shown to enhance the robustness of clustering results compared to individual clustering algorithm. To this end, we propose an efficient weight estimation method and a two layer weighted integrative consensus clustering.

2.1. *Weight estimation*

The consensus matrix can be used to assess cluster stability and composition. As a motivating example, we generate two datasets, each with 10 features and 100 samples. For both datasets, we assume that there are 3 clusters with cluster sizes 20, 30 and 50. All the features are generated from the Gaussian distribution. For dataset 1, 9 out of the 10 features are informative, where the means of cluster 1, 2 and 3 are 1, -1 and 0, respectively with unit variance. For dataset 2, 3 out of the 10 features are informative, where the means of cluster 1, 2 and 3 are 0.2, -0.2 and 0, respectively with unit variance. Non-informative features are generated from

standard Gaussian distribution. We designate datasets 1 and 2 as having high and and low signal-to-noise-ratio (SNR), respectively and run consensus clustering on both datasets using 100 iterations of k-means and resampling 80% of samples and features in each iteration. Figure 1 shows the heatmaps of the consensus matrices. The diagonal blocks plot the in-cluster values, whereas the off diagonals blocks plot the out-of-cluster values. For the low SNR dataset, the off diagonal blocks are much noisier compared to the high SNR dataset. We argue that this can be used to derive the weights in the multiple kernel integrative clustering. Specifically, we define the weights based on the ratio of in-cluster proportion to out-of-cluster proportion using the cluster estimated by the algorithm itself. Clustering result closer to the real structure tends to have higher in-cluster proportion and lower out-of-cluster proportion. In other words, datasets with a higher ratio of in-cluster proportion to out-of-cluster proportion will be assigned larger weights.

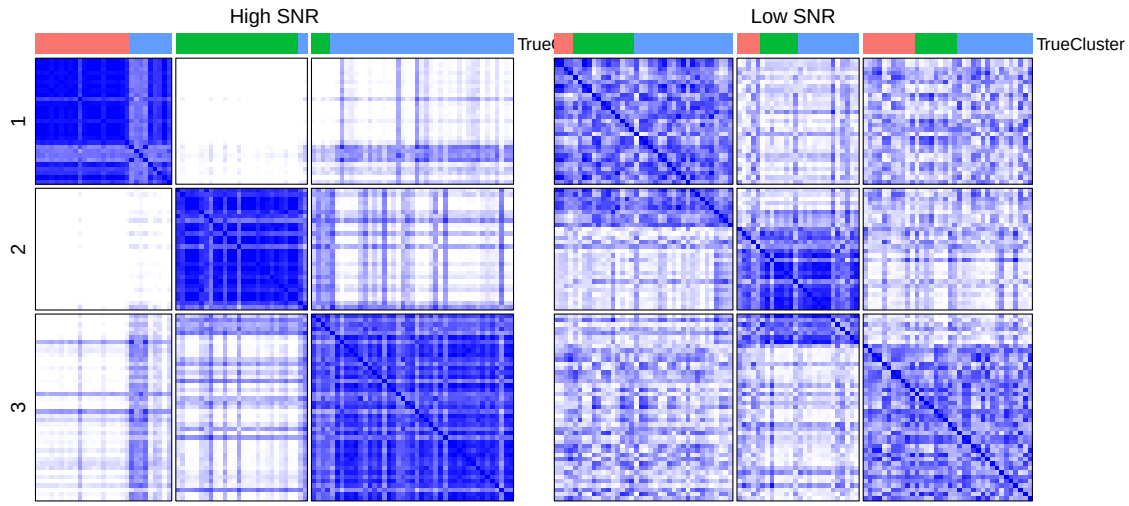

Fig. 1. Heatmaps of consensus matrices for high and low signal-to-noise ratio (SNR) datasets. True cluster membership is given in the annotation above each heatmap. Predicted cluster membership corresponds to the three gap-separated blocks in each heatmap.

Without loss of generality, we consider $\mathcal{P}$ consensus matrices $\mathcal{M}_1, ..., \mathcal{M}_P$ for number of clusters K. Here, the consensus matrices could arise by applying different clustering algorithms to the same dataset or could denote consensus matrices derived from different datasets. We further define:

$W_{in}^p(k)$: in-cluster proportion for cluster k of consensus matrix $\mathcal{M}_p$.
$W_{out}^p(k)$: out-of-cluster proportion for cluster k of consensus matrix $\mathcal{M}_p$.
W_{in}^p: average in-cluster proportion across all clusters of consensus matrix $\mathcal{M}_p$.
W_{out}^p: average out-of-cluster proportion across all clusters of consensus matrix $\mathcal{M}_p$.
R_p: ratio of in-cluster proportion to out-of-cluster proportion for consensus matrix $\mathcal{M}_p$.
W_p: weight for consensus matrix $\mathcal{M}_p$.

We propose calculating the weights as follows:

$$W_{in}^p(k) = \frac{\sum_{i\in k, j\in k} M_p(i,j)}{\sum I\{i \in k, j \in k\}}, \quad W_{in}^p = \frac{\sum_{k=1}^K W_{in}^p(k)}{K}$$
$$W_{out}^p(k) = \frac{\sum_{i\in k, j\notin k} M_p(i,j)}{\sum I\{i \in k, j \notin k\}}, \quad W_{out}^p = \frac{\sum_{k=1}^K W_{out}^p(k)}{K}$$
$$R_p = \frac{W_{in}^p}{W_{out}^p}$$
$$W_p = \frac{R^p}{\sum_{i=1}^P R^i}$$

In practice, true cluster membership is unknown, thus the weights will be computed based on predicted cluster membership. Using this formula, $W_1 = 0.726$ and $W_2 = 0.274$ for the consensus matrices derived based on predicted cluster membership of the two datasets above.

2.2. *Two Layer Weighted Integrative Consensus Clustering*

We now describe our proposed two layer weighted integrative consensus clustering. We assume that there are D datasets, X_1,...,X_D, and number of clusters K.

Layer 1: For each dataset X_d where $d = 1, 2, ..., D$:

(1) Perform ensemble clustering using P different clustering methods, where $p = 1, 2, ..., P$. This will generate consensus matrices $\mathcal{M}_p^d$, where $p = 1, 2, ..., P$.
(2) Compute the weights $w_1^d, w_2^d, ..., w_P^d$ for each consensus matrix $\mathcal{M}_1^d, \mathcal{M}_2^d, ..., \mathcal{M}_P^d$.
(3) Define the weighted consensus matrix $\mathcal{M}_{weight}^d$ as $\mathcal{M}_{weight}^d = \sum_{p=1}^P w_p^d \times \mathcal{M}_p^d$.
(4) Apply a clustering algorithm, e.g., PAM or hierarchical clustering, to each weighted consensus matrix $\mathcal{M}_{weight}^d$.

Layer 2:

(1) For the weighted consensus matrix $\mathcal{M}_{weight}^1, \mathcal{M}_{weight}^2, ..., \mathcal{M}_{weight}^D$, compute the weights $W_1, W_2, ..., W_D$.
(2) Define the weighted of weighted consensus matrix $\mathcal{M}_{weight}$ as $\mathcal{M}_{weight} = \sum_{d=1}^D W_d \times \mathcal{M}_{weight}^d$.
(3) Apply a clustering algorithm, e.g., PAM or hierarchical clustering, to $\mathcal{M}_{weight}$ to derive a final clustering result.

We provide a flowchart in Figure 2 summarizing our proposed two layer weighted integrative consensus clustering. Our method is implemented as a GitHub R package intCC available at `https://github.com/candsj/intCC`.

3. Simulation studies

We conduct simulation studies to compare the performance of our proposed two layer weighted integrative consensus clustering intCC against other integrative clustering methods which are implemented for both Gaussian and non-Gaussian distributed datasets, namely KLIC[6] and iClusterPlus.[3]

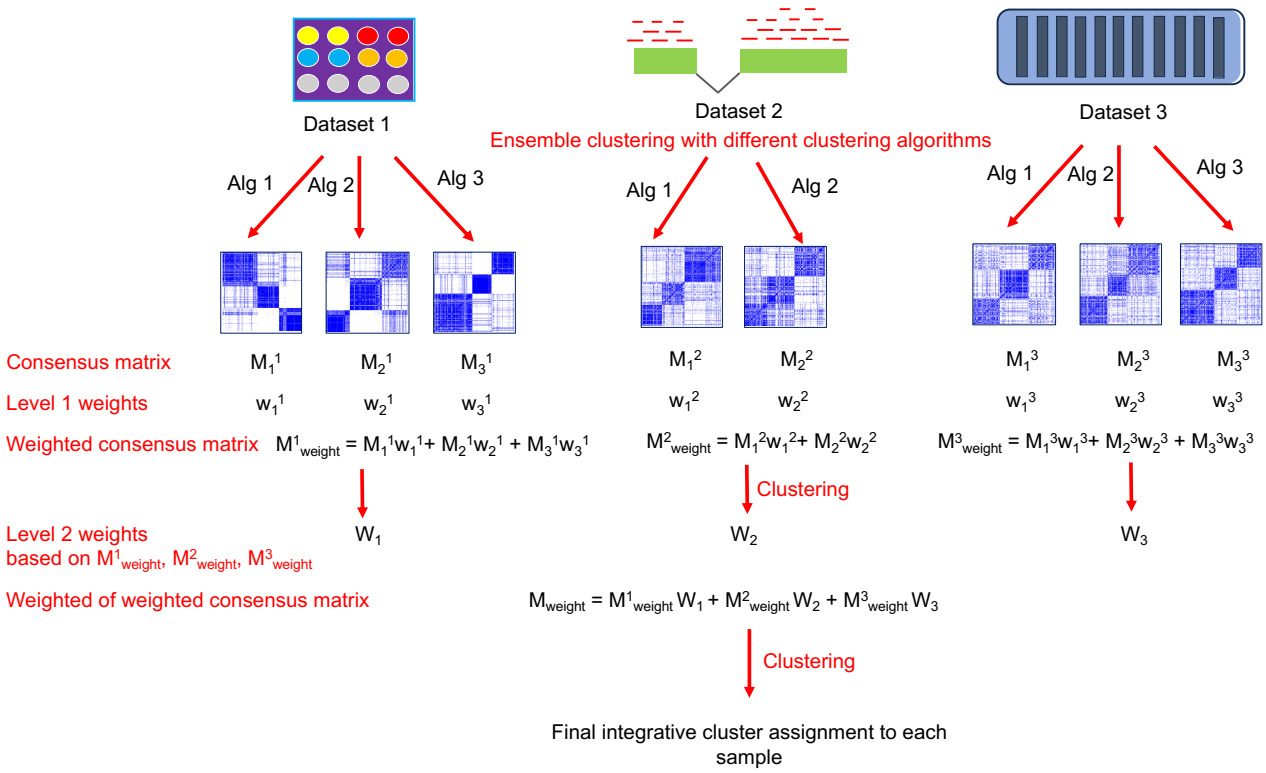

Fig. 2. Flowchat describing our proposed algorithm.

3.1. *Datasets*

Unlike Cabassi and Kirk (2020)[6] which only considered data simulated from Gaussian distributions, we follow the strategy of Mo et al. (2013)[3] where we generate datasets from different distributions, including Gaussian (e.g., M-values from DNA methylation, microarray data such as gene expression), binomial (e.g., somatic mutations), Poisson (e.g., count data from sequencing technologies such as RNA-Seq data or copy number data represented as number of copies gained or lost) and multinomial (e.g., copy number data states represented as gain, normal or loss, or SNP data) distributions. This is to ensure that our proposed method is applicable to integration of continuous, binary, count and categorical types of datasets. For Settings 1-6, we set the sample size and the true number of clusters to be 60 and 3, respectively in which each cluster consists of 20 samples. We vary the number of informative and non-informative, i.e., noise features. The parameters used in our simulations for Settings 1-6 are provided in Supplementary Table 1. Settings 7-9 follow from the simulation setup of of Cabassi and Kirk (2020).[6] We consider several simulation settings, namely:

(1) Setting 1: 4 datasets following Gaussian, binomial, Poisson and multinomial distribution, respectively. Each dataset has 30 features, in which 15 features are informative and the rest are noise features.
(2) Setting 2: 4 datasets includes normal, binomial, Poisson and multinomial distribution, respectively. Each dataset has 30 features, in which 15 features are informative and the

rest are noise features. Informative features have slighly lower signal compared to the Setting 1.

(3) Setting 3: 3 datasets following Gaussian, binomial and Poisson distribution, respectively. Each dataset has 30 features, in which 15 features are informative and the rest are noise features.

(4) Setting 4: 5 datasets following Gaussian, binomial, Poisson, multinomial and Gaussian distribution, respectively. Each dataset has 30 features. For the first 4 datasets, 15 features are informative and the rest are noise features. The 5th dataset follows a Gaussian distribution in which all features are noise features.

(5) Setting 5: 4 datasets following Gaussian, binomial, Poisson and multinomial distribution, respectively. Each dataset has 500 features, in which 100 features are informative and the rest are noise features.

(6) Setting 6: 4 datasets following Gaussian, binomial, Poisson and multinomial distribution, respectively. Each dataset has 500 features, in which 250 features are informative and the rest are noise features.

(7) Setting 7: 4 datasets following Gaussian distribution with similar parameter setting. Each dataset consists of 300 samples with 6 clusters of size 50 samples each. There are 2 features with no noise feature. For cluster k, $\mu = k \times (\text{separation level} - 1)/2, \sigma = 1$, $k = 1, 2, 3, 4, 5, 6$. Separation level $= 4$ is used in this setting.

(8) Setting 8: 4 datasets following Gaussian distribution with different parameter setting. Each dataset consists of 300 samples with 6 clusters of size 50 samples each. There are 2 features with no noise feature. For cluster k, $\mu = k \times (\text{separation level} - 1)/2, \sigma = 1$, $k = 1, 2, 3, 4, 5, 6$. Varying separation levels $= 1, 2, 3, 4$ are used in this setting. Only 3 datasets are used as input. We consider 4 dataset combinations, namely 123, 124, 134, 234. Here 123 implies that the clustering algorithms are applied to only datasets 1, 2 and 3.

(9) Setting 9 (nested cluster structure): 2 datasets following Gaussian distribution, in which each dataset consists of 300 samples. There are 2 features with no noise feature. Dataset 1 has 6 clusters of size 50 samples each. Dataset 2 has 3 clusters of size 100 samples each. For cluster k, $\mu = k \times (\text{separation level} - 1)/2, \sigma = 1$, $k = 1, 2, 3, 4, 5, 6$ for dataset 1 and $k = 1, 2, 3$ for dataset 2. Separation level $= 4$ is used in this setting.

Each setting is repeated 100 times. Additional simulation settings including multivariate Gaussian distribution are provided in Supplementary Material.

3.2. *Clustering algorithms*

We apply several clustering strategies based on our proposed method intCC, KLIC[6] and iClusterPlus.[3] To evaluate the advantage of ensemble clustering, i.e., applying multiple clustering algorithms to each dataset, we also include our proposed method which only runs a single clustering algorithm to each dataset. We denote this as one layer weighted integrative consensus clustering. We also compare application of PAM and hierarchical clustering to the weighted consensus matrix in deriving a final clustering result. These methods are denoted as:

(1) iClusterPlus: applying iClusterPlus with the data type specified.

(2) KLIC-k-means: KLIC by applying k-means to each dataset for generating the consensus matrix.
(3) KLIC-Hclust: KLIC by applying hierarchical clustering to each dataset for generating the consensus matrix.
(4) 1 layer intCC-k-means (PAM): One layer weighted integrative consensus clustering by applying k-means to each dataset for generating the consensus matrix, followed by PAM to derive a final clustering result.
(5) 1 layer intCC-Hclust (PAM): One layer weighted integrative consensus clustering by applying hierarchical clustering to each dataset for generating the consensus matrix, followed by PAM to derive a final clustering result.
(6) 1 layer intCC-k-means (Hclust): One layer weighted integrative consensus clustering by applying k-means to each dataset for generating the consensus matrix, followed by hierarchical clustering to derive a final clustering result.
(7) 1 layer intCC-Hclust (Hclust): One layer weighted integrative consensus clustering by applying hierarchical clustering to each dataset for generating the consensus matrix, followed by hierarchical clustering to derive a final clustering result.

To obtain an unbiased comparison to our two layer approach, we also apply KLIC with multiple clustering algorithms. In other words, suppose there are 4 datasets and two clustering algorithms are applied to each dataset, there will be a total of 8 consensus matrices, i.e., akin to applying KLIC to 8 datasets. KLIC is applied using these 8 consensus matrices as input in the multiple kernel integrative clustering. Additionally, to illustrate the advantage of two layer approach, we also include another one layer approach in which we apply a single layer weight estimation to the 8 consensus matrices. These methods are denoted as:

(8) 2 layer intCC-2 methods (PAM): Two layer weighted integrative consensus clustering by applying both k-means and hierarchical clustering to each dataset for generating the consensus matrices, followed by PAM to derive a final clustering result.
(9) 2 layer intCC-2 methods (Hclust): Two layer weighted integrative consensus clustering by applying both k-means and hierarchical clustering to each dataset for generating the consensus matrices, followed by hierarchical clustering to derive a final clustering result.
(10) KLIC-2-methods: KLIC by applying both k-means and hierarchical clustering to each dataset for generating the consensus matrices.
(11) 1 layer intCC-2 methods (PAM): One layer weighted integrative consensus clustering by applying both k-means and hierarchical clustering to each dataset for generating the consensus matrices, followed by PAM to derive a final clustering result.
(12) 1 layer intCC-2 methods (Hclust): One layer weighted integrative consensus clustering by applying both k-means and hierarchical clustering to each dataset for generating the consensus matrices, followed by hierarchical clustering to derive a final clustering result.

For Settings 1-8, we apply each method by setting the number of clusters to be the true number of clusters. In practice, one can tune the optimal number of clusters using criteria such as the silhouette method,[12] gap statistics,[13] Dunn index[14] or the delta K method.[7] For Setting 9, we consider (a) global clustering, where we set the number of clusters to be the

same throughout for both individual dataset and final integrative clustering, i.e., either 3 or 6 throughout (we denote these strategies as "Global K=3" and "Global K=6"), and (b) separate clustering, where we use the true number of clusters for individual dataset, i.e., 6 for dataset 1 and 3 for dataset 2, and consider both $K = 3$ and $K = 6$ in the final integrative clustering (we denote these strategies as "Separate K=3" and "Separate K=6"). Additionally, due to the poor performance of iClusterPlus and the long computational time, we omit iClusterPlus for Settings 4-6. We compare the performance of the clustering methods via the average adjusted rand index (ARI). We also report the weight estimation time of intCC and KLIC.

3.3. *Results*

We summarize the ARI for each simulation setting in Figure 3. Overall, results show that our proposed methods, namely 2 layer intCC-2 methods (PAM) and 1 layer intCC-k-means (PAM) perform well across all simulation settings. To explain this observation, without loss of generality, we summarize the ARI within each simulated dataset of Setting 4 in Figures 4A and 4B. The ARI by applying k-means as the base algorithm in the consensus clustering within each dataset is significantly better than hierarchical clustering in the simulated datasets considered in this paper. Thus, it is not surprising that methods which use k-means as the base clustering algorithm in the consensus clustering yield better performance. However, in practice the best base clustering algorithm is sometimes unknown. Thus, the 2 layer intCC which aggregates multiple base clustering algorithms can automatically assign higher weights to the better algorithm as shown in our simulation studies, as evident from the estimated weights in Figures 4C and 4D. It is also worth noting that our method assigns significantly smaller weights to the 5th dataset in which all the features are noise features. Additionally, using PAM to derive a final clustering result in general yields better performance compared to hierarchical clustering. We also note that the performance of iClusterPlus is significantly poorer compared to other methods, consistent with the findings of Cabassi and Kirk (2020).[6] Moreover, extending KLIC to run multiple base clustering algorithms, i.e., KLIC-2-methods has lower ARI compared to our proposed method, implying that the current KLIC framework does not yield a straightforward extension to incorporate ensemble clustering.

Without loss of generality, we also report the weight calculation time for KLIC and our proposed method intCC for Setting 1 (60 samples) and Setting 7 (300 samples) in Table 1, which shows that our proposed weight calculation is computationally efficient and yields good operating characteristics.

4. Case study

We illustrate our proposed method intCC on the TCGA pan cancer datasets.[15] There are 5 datasets across 12 cancer types which represent different tissues of origin, including DNA copy number, DNA methylation, mRNA expression, microRNA expression and protein expression data. To minimize bias in the comparison, we use the same preprocessing pipeline as previously described.[6,15]

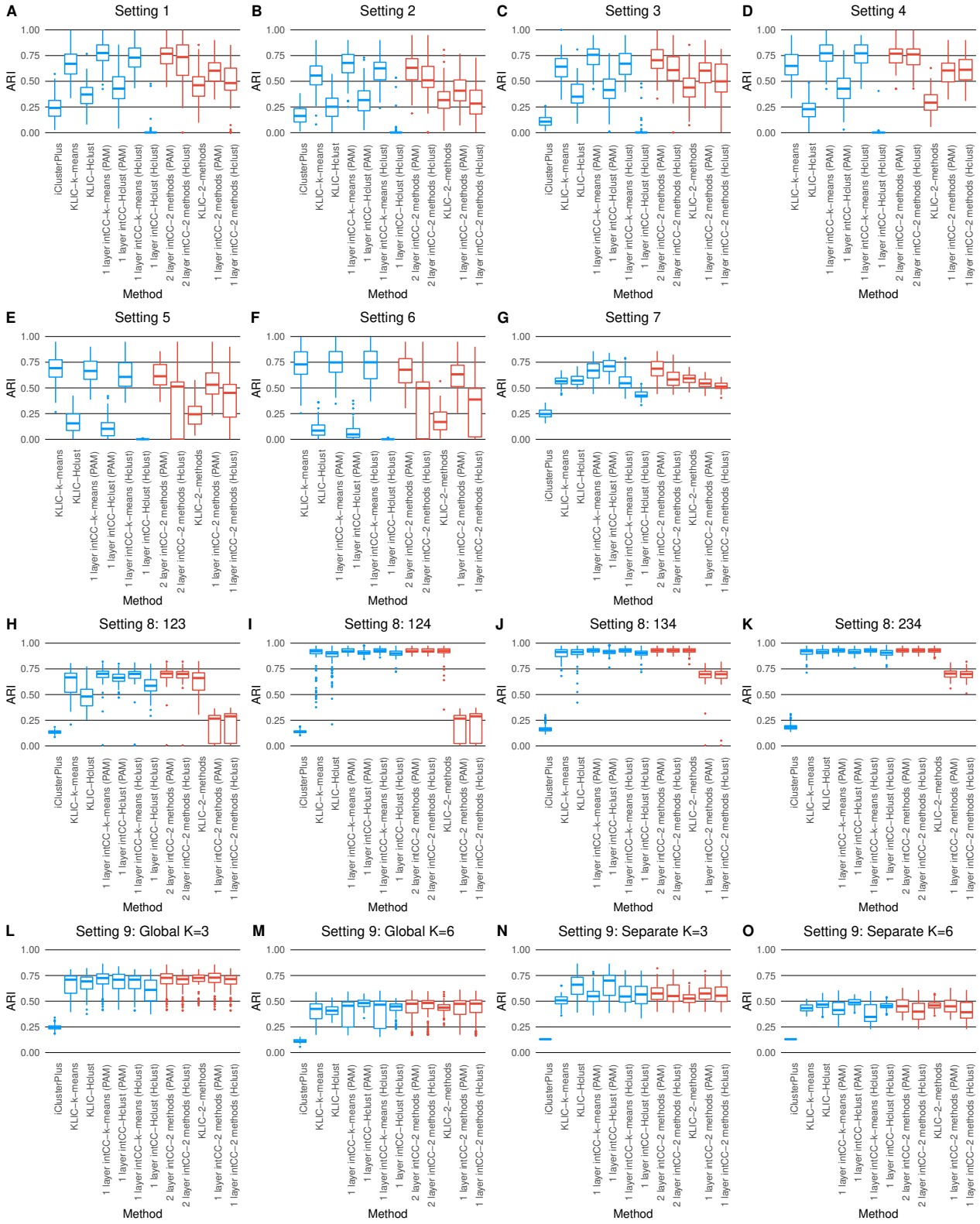

Fig. 3. Distribution of ARI across all methods and simulation settings. Blue (red) boxplots are methods which apply one (two) clustering algorithm(s) per dataset. A-G. Settings 1-7. H-K. Setting 8 with different dataset combinations as input. L-O. Setting 9 with different strategies for setting number of clusters for individual dataset and final integrative clustering.

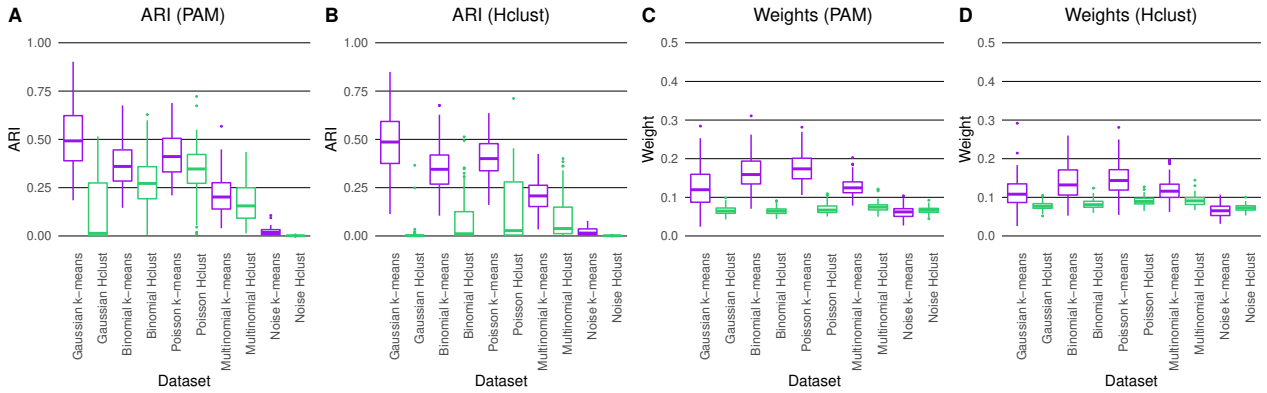

Fig. 4. A-B. Distribution of ARI within each simulated dataset of Setting 4. C-D. Distribution of estimated weights from intCC within each simulated dataset of Setting 4. Purple (green) boxplots are results by applying k-means (hierarchical clustering) algorithm in the consensus clustering. A, C. Using PAM to derive a final clustering result. B, D. Using hierarchical clustering to derive a final clustering result.

Table 1. Weight calculation time comparison.

Method	Setting 1 (seconds)	Setting 7 (seconds)
KLIC-k-means	0.541	7.209
KLIC-Hclust	0.791	8.241
1 layer intCC-k-means (PAM)	0.000879	0.00330
1 layer intCC-Hclust (PAM)	0.000882	0.00335
1 layer intCC-k-means (Hclust)	0.000876	0.00333
1 layer intCC-Hclust (Hclust)	0.000909	0.00332
2 layer intCC-2 methods (PAM)	0.00273	0.0103
2 layer intCC-2 methods (Hclust)	0.00265	0.0102
KLIC-2-methods	2.428	27.592
1 layer intCC-2 methods (PAM)	0.00155	0.00673
1 layer intCC-2 methods (Hclust)	0.00152	0.00686

Cabassi and Kirk (2020)[6] followed the same procedures described in Hoadley et al. (2014)[15] in setting the number of clusters for each dataset, except for microRNA expression in which the authors identified 8 as the number of clusters. We also set the number of clusters for each dataset following Cabassi and Kirk (2020).[6] Subsequently, we apply our proposed method intCC to obtain an integrative clustering across these datasets using the PAM algorithm to derive a final clustering result. Our method also selects 10 as the optimal number of clusters based on the average silhouette criterion, similar to KLIC.[6] Figure 5A compares the cluster membership of our method intCC against the results of KLIC, with ARI 0.693, whereas Figures 5B and 5C compare the cluster membership of intCC and KLIC against the 12 cancer type annotation, respectively. The ARI between intCC and cancer type annotation associated with tissues of origin is 0.754, whereas the ARI between KLIC and cancer type annotation is

0.585, indicating that the cluster membership of intCC yields a higher consistency with tissues of origin in the TCGA pan cancer datasets. Further investigation into the clusters obtained by intCC versus KLIC among subset of breast invasive carcinoma (BRCA) indicates that the results from intCC yield a higher consistency with the TCGA-BRCA molecular subtypes compared to the results from KLIC (Supplementary Material).

The estimated weights of each dataset for intCC and KLIC are (DNA copy number, DNA methylation, mRNA expression, miRNA expression, protein expression) =(0.073, 0.401,0.045, 0.272, 0.209) and (0.309, 0.192, 0.168, 0.183, 0.148), respectively. intCC assigns a higher weight to DNA methylation data, whereas KLIC assigns a higher weight to the copy number data, which could explain the differences observed in cluster memberships obtained by these two methods. Finally, the weight calculation time for intCC is 0.43 second, whereas the weight calculation time for KLIC via quadratic programming is > 10 hours on an Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz.

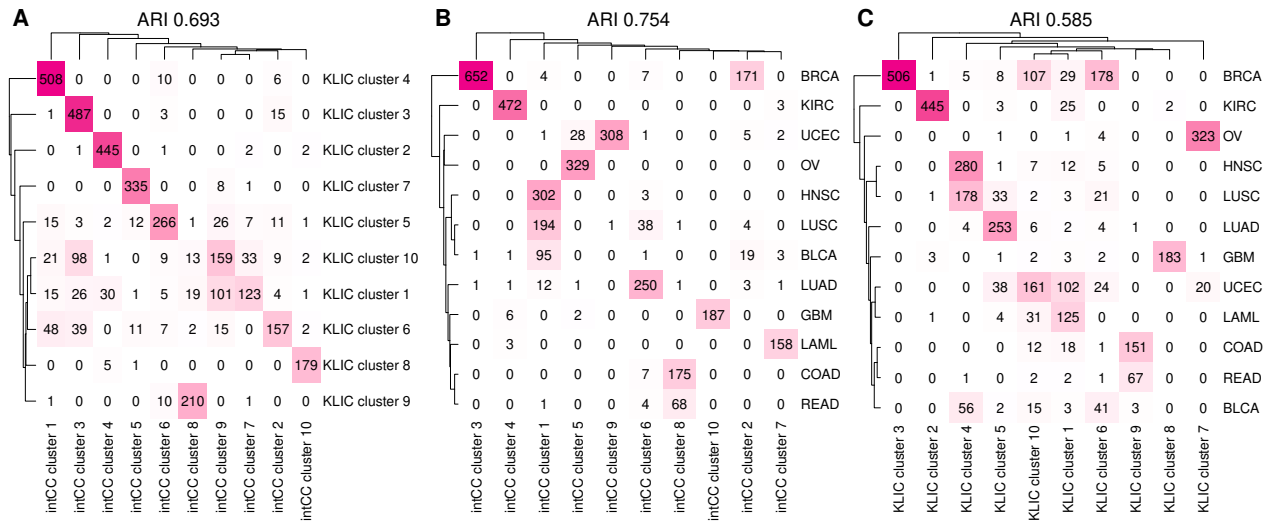

Fig. 5. Heatmaps of coincidence matrices comparing A. intCC clusters to KLIC clusters, B. intCC clusters to cancer type annotation, C. KLIC clusters to cancer type annotation. The ARI is reported in the header of each plot.

5. Discussion

The rapid development of high throughput technologies has provided an avenue to scientists to decipher the complex human diseases from a systems biology perspective via multiomics profiling. Integrative clustering has become a powerful approach to dissect the heterogeneity underpinning these diseases, e.g., to define new cancer subtypes which may help inform treatment efforts. In this paper, we extend the framework of KLIC[6] which recasts the integrative clustering model into multiple kernel learning framework by utilizing the consensus matrices estimated from consensus clustering as input. Specifically, our model further incorporates the ensemble learning via an aggregation of multiple base clustering algorithms to enhance the

robustness of multiple kernel integrative clustering model. This is to safeguard against applying a single base clustering algorithm that performs poorly on the dataset. Additionally, we also propose an efficient weight estimation to combine the consensus matrices. Our simulation studies show that the proposed two layer weighted integrative clustering yields better performance overall.

Conceptually, the weight estimation is analogous to the heuristics of multiple kernel support vector machine (MKL-SVM) based on kernel-target alignment.[16–18] Specifically, MKL-SVM is developed for supervised learning and the kernel-target alignment depends on the true binary class labels. For a fixed cluster membership, this is equivalent to multi-class classification. One can extend the kernel-target alignment for multi-class classification by dividing the problem into several binary classification subproblems (e.g., one-versus-all or all-pairs). However, how to optimally combine the results across these binary subproblems is not trivial and may require longer computational time compared to our proposed method.

Besides identifying appropriate and robust clustering algorithms, another important research question in unsupervised learning is in tuning the optimal number of clusters. Several metrics have been proposed for this task, including the silhouette method,[12] gap statistics,[13] Dunn index[14] and the delta K method.[7] An immediate extension to our intCC framework is to aggregate the different metrics/criteria for selecting the optimal number of clusters.

Supplementary Material and Code

Supplementary Material is available online at
`http://www.ams.sunysb.edu/~pfkuan/PDF/SM_PSB2024.pdf`.
The R code implementing intCC is available online at `https://github.com/candsj/intCC`.

Acknowledgments

This work is supported in part by CDC/NIOSH award U01OH012257. The findings and conclusions presented in this article are those of the authors and do not represent the official position of NIOSH, the CDC or the U.S. Public Health Service.
Conflict of Interest: None declared.

References

1. E. F. Lock and D. B. Dunson, Bayesian consensus clustering, *Bioinformatics* **29**, 2610 (2013).
2. R. Shen, A. B. Olshen and 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 (2009).
3. Q. Mo, S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, C. Sander, R. S. Powers, M. Ladanyi and R. Shen, Pattern discovery and cancer gene identification in integrated cancer genomic data, *Proceedings of the National Academy of Sciences* **110**, 4245 (2013).
4. B. . W. H. . H. M. S. C. L. . . P. P. J. . K. R. 13, G. data analysis: Baylor College of Medicine Creighton Chad J. 22 23 Donehower Lawrence A. 22 23 24 25, I. for Systems Biology Reynolds Sheila 31 Kreisberg Richard B. 31 Bernard Brady 31 Bressler Ryan 31 Erkkila Timo 32 Lin Jake 31 Thorsson Vesteinn 31 Zhang Wei 33 Shmulevich Ilya 31 *et al.*, Comprehensive molecular portraits of human breast tumours, *Nature* **490**, 61 (2012).

5. E. Gabasova, J. Reid and L. Wernisch, Clusternomics: Integrative context-dependent clustering for heterogeneous datasets, *PLoS Computational Biology* **13**, p. e1005781 (2017).
6. A. Cabassi and P. D. Kirk, Multiple kernel learning for integrative consensus clustering of omic datasets, *Bioinformatics* **36**, 4789 (2020).
7. S. Monti, P. Tamayo, J. Mesirov and T. Golub, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data, *Machine Learning* **52**, 91 (2003).
8. F. R. Bach, G. R. Lanckriet and M. I. Jordan, Multiple kernel learning, conic duality, and the smo algorithm, in *Proceedings of the Twenty-First International Conference on Machine Learning*, (Banff, Canada, 2004).
9. G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui and M. I. Jordan, Learning the kernel matrix with semidefinite programming, *Journal of Machine Learning Research* **5**, 27 (2004).
10. M. Gönen and A. A. Margolin, Localized data fusion for kernel k-means clustering with application to cancer biology, *Advances in Neural Information Processing Systems* **27** (2014).
11. O. Sagi and L. Rokach, Ensemble learning: A survey, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **8**, p. e1249 (2018).
12. P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* **20**, 53 (1987).
13. R. Tibshirani, G. Walther and T. Hastie, Estimating the number of clusters in a data set via the gap statistic, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**, 411 (2001).
14. M. Halkidi, Y. Batistakis and M. Vazirgiannis, On clustering validation techniques, *Journal of Intelligent Information Systems* **17**, 107 (2001).
15. K. A. Hoadley, C. Yau, D. M. Wolf, A. D. Cherniack, D. Tamborero, S. Ng, M. D. Leiserson, B. Niu, M. D. McLellan, V. Uzunangelov *et al.*, Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin, *Cell* **158**, 929 (2014).
16. N. Cristianini, J. Shawe-Taylor, A. Elisseeff and J. Kandola, On kernel-target alignment, *Advances in neural information processing systems* **14** (2001).
17. C. Cortes, M. Mohri and A. Rostamizadeh, Two-stage learning kernel algorithms, *Proceedings of the 27 th International Conference on Machine Learning* , 239 (2010).
18. S. Qiu and T. Lane, A framework for multiple kernel support vector regression and its applications to sirna efficacy prediction, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **6**, 190 (2008).

PACIFIC SYMPOSIUM ON BIOCOMPUTING 2024

PACIFIC SYMPOSIUM ON BIOCOMPUTING 2024

Kohala Coast, Hawaii, USA,
3 – 7 January 2024

Edited by

Russ B. Altman
Stanford University, USA

Lawrence Hunter
University of Colorado Health Sciences Center, USA

Marylyn D. Ritchie
University of Pennsylvania, USA

Tiffany Murray
Stanford University, USA

Teri E. Klein
Stanford University, USA

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI • TOKYO

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

Online ISSN: 2335-6936
Print ISSN: 2335-6928

Library of Congress Control Number: 2023949005

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

BIOCOMPUTING 2024
Proceedings of the Pacific Symposium

ISBN 978-981-12-8642-1 (ebook)
ISBN 978-981-12-8641-4 (print)