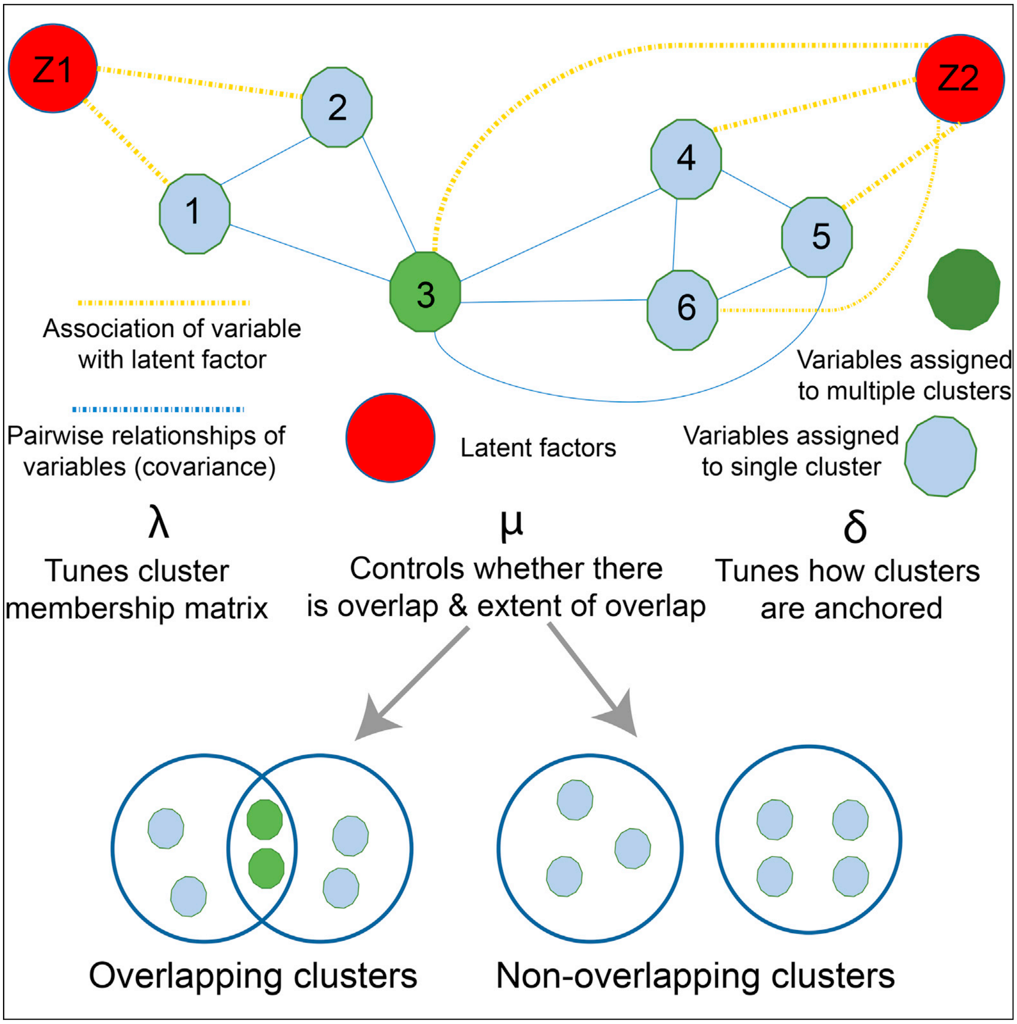
iScience

CellPress

Article

Latent Model-Based Clustering for Biological Discovery


Z1
Z2
1
2
3
4
5
6
Association of variable with latent factor
Pairwise relationships of variables (covariance)
Latent factors
Variables assigned to multiple clusters
Variables assigned to single cluster
λ
Tunes cluster membership matrix
μ
Controls whether there is overlap & extent of overlap
δ
Tunes how clusters are anchored
Overlapping clusters
Non-overlapping clusters

Xin Bing,
Florentina Bunea,
Martin Royer,
Jishnu Das

fb238@cornell.edu (F.B.)
jd327@cornell.edu (J.D.)

HIGHLIGHTS

LOVE is a robust, scalable, and versatile latent model-based clustering method

Has theoretical guarantees, and can generate overlapping and non-overlapping clusters

Generates meaningful clusters from datasets spanning a range of biological domains

Using established benchmarks, outperforms 13 state-of-the-art methods across datasets

Bing et al., iScience *14*, 125–135
April 26, 2019 © 2019 The Authors.
https://doi.org/10.1016/j.isci.2019.03.018

iScience

CellPress

Article

Latent Model-Based Clustering for Biological Discovery

Xin Bing,[1] Florentina Bunea,[1,*] Martin Royer,[1,2] and Jishnu Das[3,4,5,*]

SUMMARY

LOVE, a robust, scalable latent model-based clustering method for biological discovery, can be used across a range of datasets to generate both overlapping and non-overlapping clusters. In our formulation, a cluster comprises variables associated with the same latent factor and is determined from an allocation matrix that indexes our latent model. We prove that the allocation matrix and corresponding clusters are uniquely defined. We apply LOVE to biological datasets (gene expression, serological responses measured from HIV controllers and chronic progressors, vaccine-induced humoral immune responses) resulting in meaningful biological output. For all three datasets, the clusters generated by LOVE remain stable across tuning parameters. Finally, we compared LOVE's performance to that of 13 state-of-the-art methods using previously established benchmarks and found that LOVE outperformed these methods across datasets. Our results demonstrate that LOVE can be broadly used across large-scale biological datasets to generate accurate and meaningful overlapping and non-overlapping clusters.

INTRODUCTION

One of the most critical aspects of handling large biological datasets is identifying and accurately quantifying similarities and differences in the data. Clustering is one of the most popular ways to do this, and many clustering algorithms with specific biological applications have been developed over the last two decades. However, despite the availability of numerous clustering algorithms, three key issues still remain unaddressed. Most clustering methods use heuristics to assign clusters and do not come with rigorous statistical performance guarantees. Second, existing clustering methods work well only for specific datasets. A comprehensive benchmarking of 13 well-known methods across 24 datasets revealed that there was no universal best performer; rather, methods typically worked best for the datasets that they were specifically designed for (Wiwie et al., 2015). Even in our own experience, the choice of an appropriate clustering method is highly specific to the dataset being analyzed (Das et al., 2012, 2013, 2015; Vo et al., 2016). Furthermore, clustering methods typically work for generating either overlapping or non-overlapping clusters, but not both.

These led us to envision a clustering approach that comes with rigorous statistical guarantees regarding both cluster identification and assignment of variables to clusters. The method would be generically applicable across a wide range of datasets, as there would be no assumptions regarding how the data were generated. Furthermore, the method is designed to estimate the type of clusters that best fit a dataset, which in some cases may be overlapping and in other situations non-overlapping. Here, we report LOVE, a robust and scalable (Table 2) latent factor model-based clustering method with all the above properties. We apply LOVE to three datasets with very different properties and correlation structures (Table 1) and show that LOVE generates stable, biologically meaningful, and accurate clusters in all three cases, outperforming state-of-the-art methods. Furthermore, we demonstrate that LOVE consistently outperforms 13 previously benchmarked state-of-the-art clustering methods (Wiwie et al., 2015) across datasets (Table 3).

RESULTS

Formulation of LOVE

We consider the following latent factor model,

$$X = AZ + E$$

where $X \in \mathbb{R}^p$ represents p variables (to be clustered); $Z \in \mathbb{R}^K$ denotes K latent, unobservable factors corresponding to the K clusters; $A \in \mathbb{R}^{p \times K}$ represents the membership matrix assigning p variables to K clusters; and E denotes an error term corresponding to random noise.

[1] Department of Statistical Science, Cornell University, Ithaca, NY 14853, USA

[2] Department of Mathematics, Universite Paris-Sud, 91405 Orsay, France

[3] Ragon Institute of MGH, Harvard, MIT, Cambridge, MA 02139, USA

[4] Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[5] Lead Contact

*Correspondence: fb238@cornell.edu (F.B.), jd327@cornell.edu (J.D.)

https://doi.org/10.1016/j.isci.2019.03.018

Dataset	Number of Variables	Number of Measurements	Quantiles of Correlation of Measurements (0%, 25%, 50%, 75%, 100%)				
Gene expression	16,134	114	−0.51	−0.06	−0.01	0.04	1.00
Progressors and controllers	19	72	−0.68	−0.10	0.08	0.30	1.00
Vaccine-induced humoral immune responses	60	191	−0.99	0.03	0.22	0.44	1.00

Table 1. Applicability of LOVE across Datasets with Different Correlation Structures

We define *K* clusters based on the *K* latent factors. Assignment of variables to clusters is based on the membership matrix A, which assigns some variables to only one cluster, to anchor that cluster, and other variables to multiple clusters.

There are three main steps in LOVE (Figure 1). The first step involves determination of the covariance network connecting the variables of interest, and the structure of the underlying latent variables (Figure 1A). The second involves inferring the strongest connections between variables. Based on the strength of these connections, variables are designated as "mixed" and "unmixed." Mixed variables are defined as those that are associated with multiple latent factors, whereas unmixed variables are those that are associated with a single latent factor. The unmixed variables are then used to identify the unique clusters (Figure 1B). The final step comprises assignment of the mixed variables to multiple clusters based on the membership matrix *A* (Figure 1C). There are three primary tuning parameters in LOVE—determination of cluster anchors using delta, membership matrix determination using lambda, and thresholding during assignment of variables to clusters using mu. The mathematical details of each step, as well as the associated parameter tuning, are provided in the Transparent Methods. We also provide code to implement each step of LOVE as well as associated detailed documentation (Data S1). LOVE is highly scalable as a method. The determination of pure and mixed variables is an O(*n*^2) algorithm, where *n* is the number of variables to be clustered. The estimation of the membership matrix A involves solving *K* linear programs (where *K* is the number of latent factors). To practically test the scalability of LOVE across datasets with different numbers of variables, we tested LOVE on a wide range of datasets from a few hundred (10^2) to a million variables (10^6). Runtimes on a single core ranged from under a second to ∼140 h (Table 2). Thus even for the largest dataset of a million variables, this extrapolates to a runtime of a few hours on a typical server or cluster node. Thus LOVE is highly scalable and can efficiently cluster even ultralarge datasets in hours.

Overlapping Clustering Using LOVE

To test overlapping clustering using LOVE, we used a previously described compendium of human gene expression data (Das et al., 2012). The dataset corresponds to expression measurements for 16,134 genes across 114 different points in the cell cycle. Using LOVE, we obtained >1,000 overlapping clusters, which corresponded well with prior biological expectation. For example, we obtained three overlapping clusters, where cluster 1 contained the genes RWDD3, BRD30S, and IRF2; cluster 2 contained the genes BRF3OS, IRF2, CTGF, INHBA, and INHBB; and cluster 3 contained the genes INHBA, INHBB, and COLA1A2 (Figure 2A). Based on KEGG pathway annotations, RWDD3 is associated with nuclear factor (NF)-κB signaling and CTGF is associated with leishmaniasis, whereas the genes shared between clusters 1 and 2—BRD3OS and IRF2—are known to be associated with both NF-κB signaling and leishmaniasis (Figure 2A). CTGF is also associated with the KEGG inflammatory bowel disease (IBD) pathway, and COL1A2, the only gene in cluster 3, is in the KEGG pathogenic *E. coli* infection pathway. Completely consistent with this, the genes shared between LOVE clusters 2 and 3—INHBA and INHBB—are associated with both the IBD and the pathogenic *E. coli* pathways. These results show that the clusters detected by LOVE correspond to specific functions, illustrating that the latent factors in our modeling formulation are not only mathematical entities but also have underlying biological significance. Genes that are only in one cluster, but not others, define the biological relevance of the clusters; this perfectly matches our mathematical latent factor formulation. Furthermore, LOVE is very specific at detecting meaningful overlaps based on shared biological functions—different sets of genes were detected as overlapping between clusters 1 and 2 and 2 and 3, and both sets were consistent with prior biological expectations based on genes that are present in only one of the clusters. The clusters presented in Figure 2A, and the corresponding overlaps, serve as examples

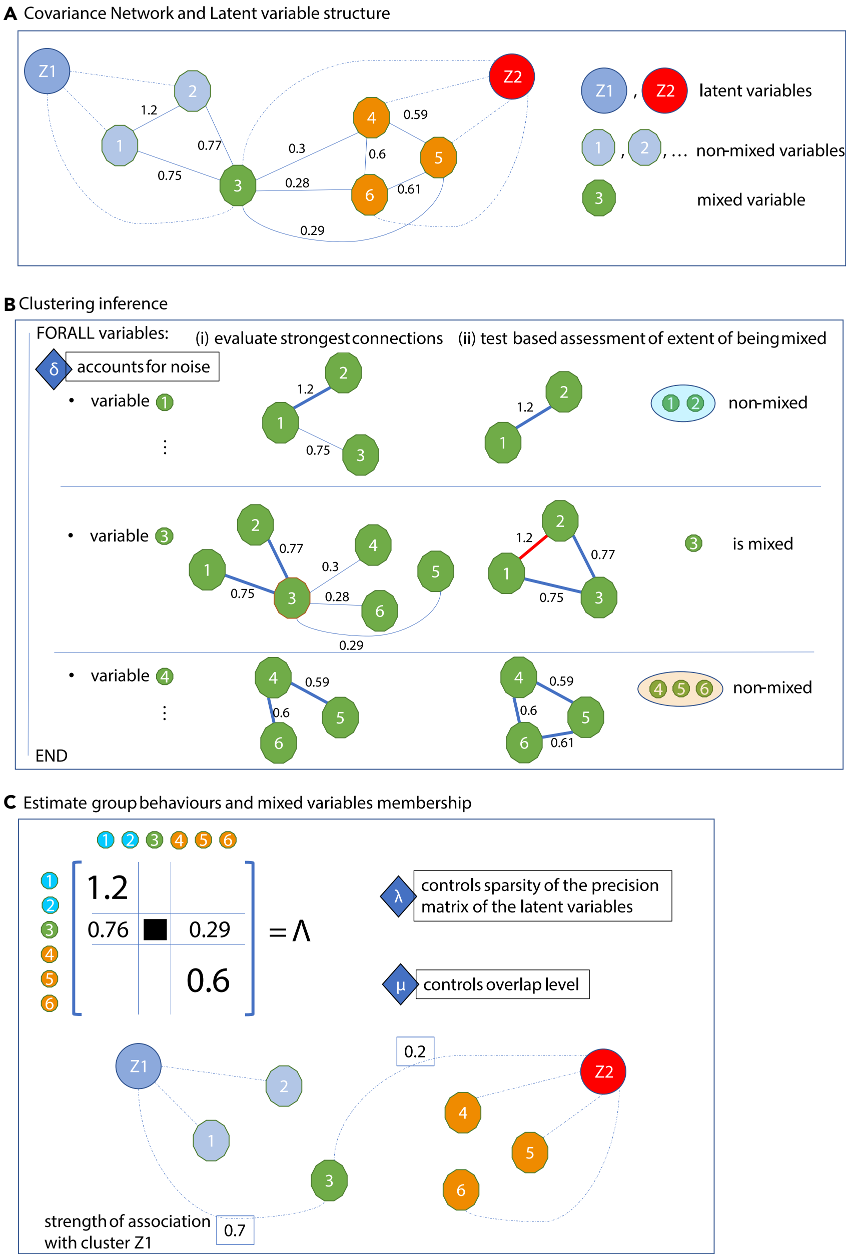

A Covariance Network and Latent variable structure
Z1
Z2
2
1
3
4
5
6
1.2
0.77
0.75
0.3
0.28
0.59
0.6
0.61
0.29
Z1 , Z2 latent variables
1 , 2 , … non-mixed variables
3 mixed variable
B Clustering inference
FORALL variables:
δ accounts for noise
• variable 1
⋮
• variable 3
• variable 4
⋮
END
(i) evaluate strongest connections
(ii) test based assessment of extent of being mixed
1.2
0.75
1 2 non-mixed
0.77
0.3
0.28
0.29
3 is mixed
0.59
0.6
0.61
4 5 6 non-mixed
C Estimate group behaviours and mixed variables membership
1 2 3 4 5 6
1.2
0.76
0.29
0.6
= Λ
λ controls sparsity of the precision matrix of the latent variables
μ controls overlap level
0.2
Z1
Z2
strength of association with cluster Z1
0.7

Figure 1. Overview of LOVE

(A–C) Schematic illustrating the steps of the LOVE algorithm. (A) Estimation of the covariance network and latent variable structure, (B) cluster inference based on strength of connections, and (C) final assignment of clusters based on estimation of the membership matrix.

of how LOVE detects biologically meaningful relationships in an unsupervised framework. Next, we systematically examine the quality of all the clusters and the corresponding overlaps.

As clusters correspond to biological functions, one expects pleiotropic genes to be overlapping across clusters as these carry out several functions. We defined pleiotropic genes based on the network degree of the proteins encoded by these genes, i.e., protein-protein interaction network hubs were defined as pleiotropic. This is a standard way to characterize multiplicity of function as proteins perform their functions by interacting with other proteins (Rolland et al., 2014; Vo et al., 2016; Yu et al., 2008) and previous studies have shown that network hubs are the most functionally important genes (Albert et al., 2000; Jeong et al.,

Dataset	Number of Variables	Number of Measurements	Runtime
Gene expression	16,134	114	4 min
Progressors and controllers	19	72	0.021 s
Vaccine-induced humoral immune responses	60	191	0.023 s
TCGA[a,b]	293	293	0.064 s
Cassini[b]	250	2	0.027 s
Synthetic_1	100,000	100	1.4 h
Synthetic_2	1,000,000	100	142.4 h

Table 2. Runtimes of LOVE for Different Datasets

We record the running time of LOVE on different datasets with one specified value for each tuning parameter on a single core of a machine (2.2 GHz Intel Core i7) with 16GB RAM.

[a] For the TCGA dataset, we have a pairwise similarity matrix of dimension $n \times n$ where, n = number of variables ($n = 293$).

[b] The TCGA and Cassini datasets are obtained from Wiwie et al. (2015).

2000; Yu et al., 2008). We used a consensus high-quality protein interaction network to define hubs (Das and Yu, 2012) and found that hubs belonged to significantly more clusters than non-hubs (Figure 2B, $p = 1.2 \times 10^{-10}$ using a Mann-Whitney U test). Thus the assignment of overlapping clusters was consistent with biological expectation—pleotropic genes were more likely to be assigned to multiple clusters than non-pleotropic genes. We then compared our results with two existing clustering methods—fuzzy Cmeans clustering (Bezdek et al., 1984) and ClusterOne (Nepusz et al., 2012). Fuzzy Cmeans clustering is a well-established and widely used distance-metric-based algorithm. ClusterOne is graph based and has recently been demonstrated to be superior to several similar approaches (Nepusz et al., 2012). Thus fuzzy Cmeans clustering and ClusterOne are two state-of-the-art methods, use orthogonal concepts, and serve as excellent benchmarks to compare against. We found that hubs were assigned to significantly more clusters by LOVE than they were by Cmeans clustering or ClusterOne (Figure 2C, $p < 10^{-10}$ using a Mann-Whitney U test), suggesting that the overlaps detected by LOVE are more consistent with prior biological expectation than the overlaps detected by other methods.

We also checked whether overlapping genes are also assigned to appropriate clusters, i.e., the clusters from LOVE coincide with prior biological expectation as defined by protein network modules. We found that the network distances (i.e., minimum path length between the two nodes) between overlapping genes and non-overlapping genes from the same cluster were low. This distribution of network distances had a median of 3.3, and 75% of network distances were under 3.6. Thus most overlapping and non-overlapping genes from the same cluster are within four hops (<25% of the network diameter) away in the protein network.

Although the above analyses show that genes with multiple functions are correctly assigned by LOVE to multiple appropriate clusters, a good clustering method should also not assign genes with similar expression levels to multiple clusters. To test this, we looked at how housekeeping genes, as defined by stable expression across 16 human tissue types (Eisenberg and Levanon, 2013), were distributed across the clusters generated by LOVE. As housekeeping genes are uniformly expressed, i.e., have low variability in their expression levels, ideally these should only be assigned to one or a few clusters and not the other clusters. To systematically test this, we calculated the under-representation of housekeeping genes in the clusters generated by LOVE and quantified this using an under-representation index (see Transparent Methods). We found that housekeeping genes were under-represented across most LOVE clusters (Figure 2D) and the corresponding index was significantly higher ($p < 10^{-10}$ using a Mann-Whitney U test) for LOVE compared with fuzzy Cmeans clustering and ClusterOne (Figure 2D). These results illustrate that LOVE not only accurately identifies overlaps but also effectively discriminates between basally expressed genes and genes with specific expression profiles.

We then explored how biologically relevant the clusters discovered by LOVE are. To define relevance, we examined whether a cluster was over-represented for at least one known Gene Ontology biological

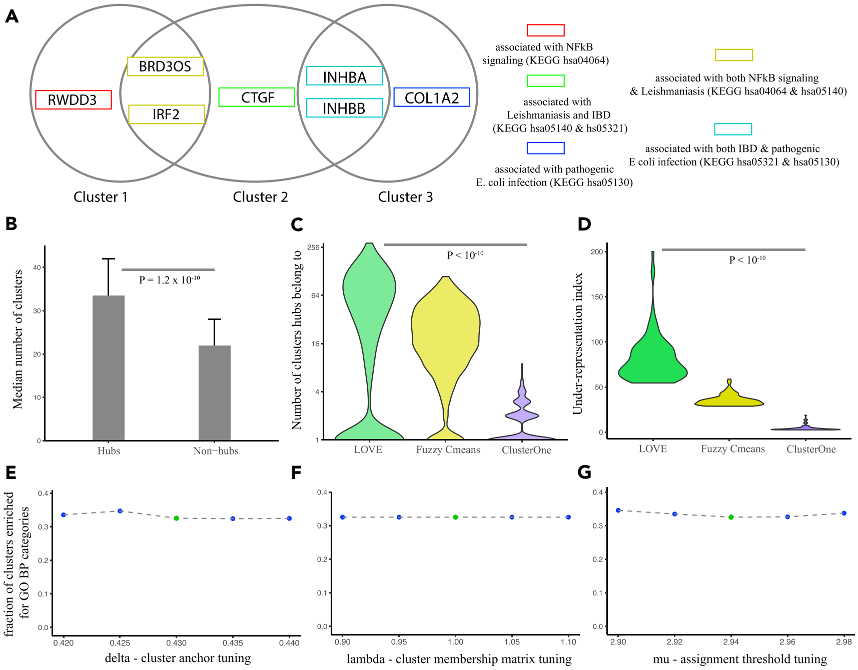


Figure 2. Overlapping Clustering Using LOVE

(A) An example of how overlapping clusters detected by LOVE are consistent with known biological pathways, both in terms of the genes assigned to single clusters and the genes shared between clusters.

(B) Median number of LOVE clusters that protein network hubs and non-hubs belong to. Error bars correspond to a decile around the median. P value calculated using a Mann-Whitney *U* test.

(C) Distributions of the number of clusters protein network hubs belong to, for each of the three methods—LOVE, fuzzy Cmeans clustering, and ClusterOne. P values calculated using a Mann-Whitney *U* test ($P < 10^{-10}$ for LOVE vs Fuzzy Cmeans and LOVE vs Cluster One).

(D) Distribution of under-representation indices corresponding to how under-represented housekeeping genes are across clusters generated by the three methods—LOVE, fuzzy Cmeans clustering, and ClusterOne. P values calculated using a Mann-Whitney U test ($P < 10^{-10}$ for LOVE vs Fuzzy Cmeans and LOVE vs ClusterOne).

(E) Fraction of clusters enriched for GO BP categories across a range of delta values, the cluster anchor tuning parameter.

(F) Fraction of clusters enriched for GO BP categories across a range of lambda values, the cluster membership matrix tuning parameter.

(G) Fraction of clusters enriched for GO BP categories across a range of mu values, the assignment threshold tuning parameter.

process (GO BP) category (Ashburner et al., 2000). Significant over-representation was defined using a false discovery rate cutoff of 0.05 (p value calculated from a hypergeometric test followed by Benjamini-Hochberg multiple testing correction) and computed using WebGestalt (Wang et al., 2013). The number of biologically relevant clusters identified using this approach represents a lower bound on the actual number of biologically relevant clusters as current GO annotations are not complete. Thus any cluster enriched for at least one GO BP category is definitely biologically relevant, whereas clusters not enriched for a GO BP category may still be meaningful. Despite this stringent evaluation criterion, we found that at optimal parameter settings, >30% of clusters detected by LOVE are enriched for a GO BP category (Figures 2E–2G), suggesting that the latent variables and corresponding clusters detected by LOVE are highly relevant biologically. Furthermore, several clusters are enriched for multiple GO categories, suggesting that we accurately recapitulate an even higher fraction of functional similarity relationships. Overall, we found 1,723 over-represented GO categories across 1,222 clusters.

Finally, we tested how the relevance of the clusters discovered by LOVE changed when key tuning parameters of the method are varied. We first performed a grid search around the optimal delta, the parameter that determines cluster anchors, i.e., which ''unmixed'' variables will serve to define clusters. We found that across a range of parameter values around the optimal delta, the fraction of clusters known to be biologically relevant remained stable (Figure 2E). Next, we performed a similar analysis with lambda—the parameter used to tune the membership matrix based on the conditional independence structure of the variables. Again, the fraction of clusters known to be biologically relevant remained stable around the optimal lambda (Figure 2F). We also observed similar results with a grid search around the optimal mu—the thresholding parameter that determines tuning of the latent variables (Figure 2G). Thus LOVE discovers biologically meaningful clusters across a range of parameter choices.

Non-overlapping Clustering Using LOVE

Most methods are good at either generating overlapping or non-overlapping clusters (Wiwie et al., 2015). However, due to the inherent formulation of LOVE, it can be used for either purpose. To test the effectiveness of LOVE in generating non-overlapping clusters, we chose a recently published dataset of humoral immune measurements from 19 human subjects from two distinct clinical phenotypes—long-term HIV controllers and chronic progressors (Sadanand et al., 2018). For each subject, 18 different measurements of antibody-effector functions and titers were available at four different time points, corresponding to a total of 72 measurements (Sadanand et al., 2018). This dataset is different with regard to several key aspects from the earlier gene expression dataset. First, the desired clusters here are non-overlapping as HIV controllers and chronic progressors are clinically distinct groups and are known to be very different in terms of their humoral responses (Alter et al., 2018; Sadanand et al., 2018). Second, the sources of biological and technical variance in the two datasets are different. In terms of biological variability, the modulation of transcript expression levels across time points in the cell cycle is structurally very different from variation across human subjects with different clinical phenotypes. The extent of technical noise is also different as microarray measurements are relatively noisy, whereas this dataset comprises serological measurements collected using modern methods. Finally, the number of entities being clustered (number of input variables for LOVE) is also very different. The gene expression dataset had >16,000 genes profiled over 114 different points in the cell cycle. The dataset of controllers and progressors have 19 human subjects. The differences across these two datasets reveal the inherent variation across different biological datasets. Testing LOVE on two extremes provides an opportunity to benchmark how the clustering method performs at different ends of the spectrum.

The 19 human subjects were split into two clusters—one of that comprised eight progressors and two controllers and the other comprised seven controllers and two progressors (Figure 3A). As each cluster primarily comprises individuals from one clinical phenotype, the latent factors in this case can be interpreted as the average humoral signature corresponding to each phenotype. Thus LOVE comes up with biologically meaningful latent factors for this dataset too, illustrating that the model formulation is both intuitive and interpretable.

Next, we evaluated the performance of LOVE in terms of accuracy, true positive rate, and true negative rate. Although clustering by definition is unsupervised, we were able to measure these metrics for this dataset as we already know the clinical outcomes for each human subject. We assumed that the ideal result would be two clusters—one comprising only controllers, and the other comprising only chronic progressors. Based on this definition of ground truth, we obtained an accuracy, a true positive rate, and a true negative rate each of ~80% for LOVE (Figure 3B). We contend that an accuracy of ~80% is really high, especially for an unsupervised method, as previously even a supervised approach had a median classification accuracy of ~75% (Sadanand et al., 2018). We also found that LOVE outperformed both Cmeans clustering and ClusterOne in terms of all three metrics—accuracy, true positive, and true negative rate (Figure 3B).

Next, we sought to explore how LOVE performs across a range of assignments. Although the most optimal assignments correspond to all subjects being assigned to a cluster with high accuracy (as shown in Figure 3B), we wanted to check how LOVE does across a range of assignment thresholds (including those where not all subjects are assigned to clusters). A receiver operating characteristic curve drawn across assignment thresholds revealed robust performance across thresholds (Figure 3C, area under the curve = 0.82).

Finally, we wanted to evaluate how stable LOVE is across the three key parameters—delta (cluster anchor tuning), lambda (membership matrix tuning), and mu (latent variable tuning). We performed a grid search around the optimal parameters and found that all the three indicators of performance—accuracy, false positive rate (1, true positive rate), and false negative rate (1, true negative rate), are stable across a range of tuning parameters (Figures 3D–3F). These results demonstrate that LOVE is able to accurately cluster even if somewhat less than optimal parameter choices are made and are analogous to those observed for overlapping clustering.

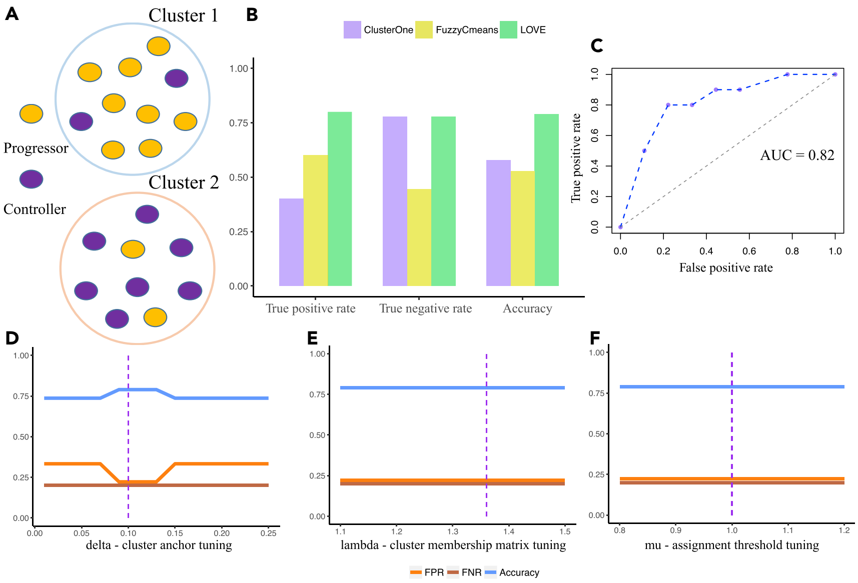


Figure 3. Non-overlapping Clustering Using LOVE

(A) Distribution of HIV controllers and chronic progressors in the two LOVE clusters.

(B) True positive rate, true negative rate, and accuracy for each of the three methods—LOVE, fuzzy Cmeans clustering, and ClusterOne.

(C) Receiver operating characteristic curve illustrating the performance of LOVE across a range of assignment thresholds for the membership matrix.

(D) Variation of true positive rate, true negative rate, and accuracy for LOVE across a range of delta values, the cluster anchor tuning parameter.

(E) Variation of true positive rate, true negative rate, and accuracy for LOVE across a range of lambda values, the cluster membership matrix tuning parameter.

(F) Variation of true positive rate, true negative rate, and accuracy for LOVE across a range of mu values, the assignment threshold tuning parameter.

LOVE on High-Dimensional Data

Finally, we sought to evaluate LOVE on a high-dimensional dataset of vaccine-induced humoral immune responses. Recently, we found that the route of immunization, even for the same immunogen, can modulate mechanisms of protection in the context of vaccination against simian immunodeficiency virus (SIV) (Ackerman et al., 2018). Our study had three vaccination arms—IM239 (administration of the SIVmac239 immunogen intramuscularly), IM mosaic (administration of a mosaic envelope immunogen intramuscularly), and AE239 (administration of the SIVmac239 immunogen via inhaled aerosol). We found that each vaccine arm induced a distinct profile of humoral immune responses (Ackerman et al., 2018). We sought to evaluate whether these differences that we had captured using a supervised approach (Ackerman et al., 2018) could also be discovered using LOVE, an unsupervised clustering method. Furthermore, here the data are high dimensional, i.e., the number of measured humoral immune responses >> the number of primates. This is typical in a study involving human subjects or non-human primates, as the cost per subject or primate is high and there are ethical guidelines outlining the maximum number of primates that can be used in such studies. Thus sample sizes for these studies are usually much smaller than the number of measured analytes. Thus having a clustering method that works on high-dimensional data is of paramount importance.

LOVE split the six primates into three clusters, each of which was clearly enriched for one of the vaccination arms (Figure 4A). Here too, the latent variable formulation has an intuitive biological explanation; they correspond to the induced humoral signature corresponding to each vaccination strategy. Overall, LOVE correctly assigned 80% of primates to the corresponding vaccination arm (Figure 4B). This is significantly better than the performance of the two other clustering methods—fuzzy Cmeans and ClusterOne (Figure 4B). Fuzzy Cmeans was reasonably accurate at discriminating between the IM239 and IM mosaic arms, i.e., the vaccination arms with different immunogens, arguably the easier split. However, it failed to discriminate the AE239 arm, i.e., the arm that had the same immunogen as IM239, but differed only

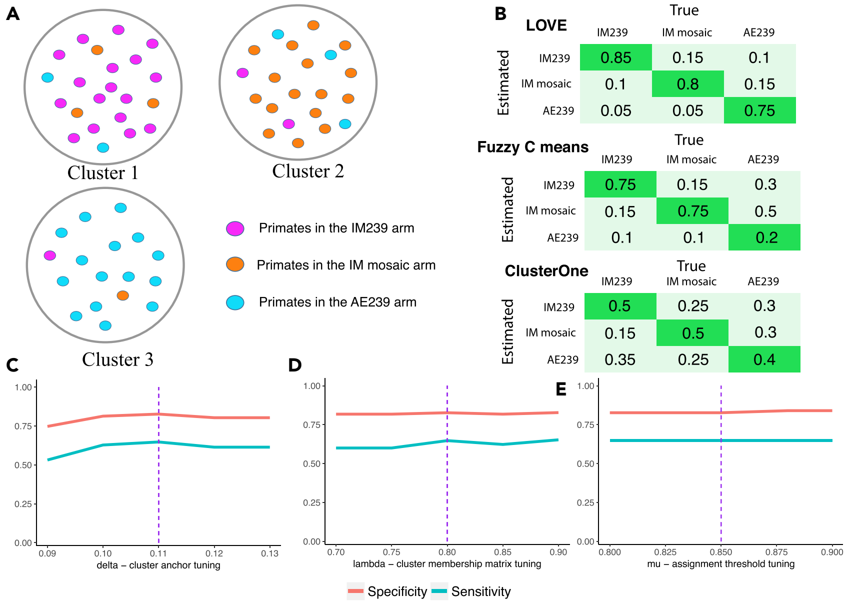


Figure 4. Using LOVE on High-Dimensional Data

(A) Distribution of primates from the three vaccination arms in the three clusters generated by LOVE.

(B) Confusion matrix for each of the three methods, LOVE, fuzzy Cmeans clustering, and ClusterOne, showing the fractions of primates correctly or incorrectly assigned to the different vaccination arms.

(C) Variation of specificity and sensitivity for LOVE across a range of delta values, the cluster anchor tuning parameter.

(D) Variation of specificity and sensitivity for LOVE across a range of lambda values, the cluster membership matrix tuning parameter.

(E) Variation of specificity and sensitivity for LOVE across a range of mu values, the assignment threshold tuning parameter.

based on the route of administration (Figure 4B). ClusterOne had a high error rate overall (Figure 4B). Furthermore, although LOVE is completely unsupervised, the accuracy obtained is comparable to what we had previously obtained using a supervised approach (Ackerman et al., 2018). Together, these results show that LOVE is very accurate even when clustering a high-dimensional dataset and significantly outperforms existing methods.

We also evaluated how stable LOVE is across the three key tuning parameters—delta (cluster anchor tuning), lambda (membership matrix tuning), and mu (latent variable tuning). A grid search around the optimal parameters revealed that both the sensitivity and specificity of LOVE are stable across a range of parameter values (Figures 4C–4E). As for the previous two datasets, these results confirm that LOVE is able to accurately cluster even if somewhat less than optimal parameter choices are made.

Benchmarking LOVE against a Wide Range of Clustering Methods

Our previous results demonstrate that LOVE works well across datasets with different properties both in terms of size and correlation structure. We also showed that LOVE outperforms two state-of-the art methods that use different approaches—fuzzy Cmeans clustering and ClusterOne. To comprehensively compare LOVE's performance in a wide range of existing methods, we used previously established benchmarks for 13 different clustering methods across datasets (Wiwie et al., 2015). Also, both the methods to benchmark against and the datasets were chosen by an independent study (Wiwie et al., 2015). Based on F1 scores (harmonic mean of precision and recall), LOVE outperforms the 13 existing methods on two very different datasets (Table 3). The first dataset quantified similarities between 293 The Cancer Genome Atlas (TCGA) clinical samples across three different cancer types—breast cancer, lung cancer, and glioblastoma. Specifically, the data were a pairwise similarity matrix of dimension 293 × 293 ($n \times n$, where $n = 293$, the number of variables/samples). For this dataset, LOVE outperformed 11 of the 13 other methods, in terms of the F1 score, and had a rank of 3 across the 14 methods (Table 3). The other dataset on which we benchmarked LOVE was synthetic and low dimensional (we have already evaluated LOVE on a high-dimensional dataset earlier), purposely chosen to be very

Dataset	TCGA	Synthetic_cassini
Affinity propagation	0.389	0.524
ClusterDP	0.921	1
ClusterOne	0.678	0.524
Density-based spatial clustering of applications with noise	0.944	1
Fanny	0.914	0.957
Hierarchical clustering	0.998	1
Cmeans clustering	–	0.78
Partitioning around medoids	0.9	0.95
Markov clustering	0.678	0.524
Molecular complex detection (MCODE)	0.894	0.992
Self-organizing maps	–	0.778
Spectral clustering	0.5	1
Transitivity clustering	0.986	0.885
LOVE	0.976	0.984

Table 3. F1 Scores for LOVE and 13 Other Methods for Different Datasets

different from the previous dataset. Here, the dataset consisted of 250 variables, and each variable had two corresponding features (i.e., each variable could be represented by a dot on a two-dimensional dot-plot). Again, LOVE outperformed nine of the 13 other methods, in terms of the F1 score, and had a rank of 5 across the 14 methods (Table 3). Overall, LOVE consistently had one of the highest ranks across the two datasets (Table 3). The only other method that had comparable performance across these two datasets was hierarchical clustering. However, hierarchical clustering does not support fuzzy clustering, whereas LOVE can generate both overlapping and non-overlapping clusters. This is an inherent strength of LOVE. Moreover, hierarchical clustering is not applicable to some of the other datasets, such as the high-dimensional dataset of vaccine-induced humoral immune responses, used in this study. Hierarchical clustering typically uses L_2 or allied L_p norms and clusters based on a distance metric. For high-dimensional datasets with highly correlated features, these L_p norms pose as inherent bias as the distance metric will be skewed toward the trends of the "larger" correlation blocks. Thus among all the methods evaluated in his study, LOVE is the only method that works consistently well across datasets of varying sizes (both number of variables and number of features) and correlation structures. This strength is rooted in LOVE's theoretical principles, as our method makes no distributional assumptions regarding the data-generating mechanisms, beyond the latent factor model formulation.

DISCUSSION

Here, we present a versatile, robust, and scalable clustering method—LOVE. Our method comes with numerous statistical guarantees regarding identifiability of clusters that existing methods do not provide. Furthermore, whereas our method uses covariance as a measure of similarity, our approach will work for any similarity measure, linear or non-linear, as long as the measure satisfies certain very generic criteria regarding its decomposition (please see Transparent Methods for additional details). The only assumptions that LOVE make are regarding the decomposition of the matrix of similarity measures, the covariance matrix in our case. It makes no further assumptions regarding the data-generating mechanism; for instance, we do not need to know or assume the distribution of the data. Most existing methods work well for specific datasets only, as there are underlying assumptions regarding how the data were generated (Wiwie et al., 2015), but LOVE is broadly applicable as it makes no such assumptions. Furthermore, whereas most existing methods generate either overlapping or non-overlapping clusters, LOVE can generate both kinds of clusters. Finally, the clusters generated by LOVE are highly stable across parameter choices.

We successfully applied LOVE to three very different systems-scale datasets. Although the nature of the data varies both within each kind of dataset (e.g., the structure and quality of gene expression measurements vary depending on the platform or technology used to generate it), and across datasets (different datasets had different correlation structures as summarized in Table 1), the theoretical guarantees provided by LOVE remain unchanged. Further benchmarking against 13 state-of-the-art methods demonstrated that LOVE is the only approach that has consistently high performance across datasets with varying properties (Table 3). This is primarily due to the latent model formulation of LOVE, which does not make any assumptions regarding data-generating mechanisms. Furthermore, the latent factors are not only mathematical constructs but also biologically meaningful and context dependent. Given these unique and novel features, we anticipate that LOVE will be widely adopted in systems biology analyses and open new avenues of biological discovery.

Limitations of Study

LOVE's inherent formulation fits most typical contexts—either each variable belongs to a single cluster (non-overlapping clustering) or some variables belong to a single cluster, whereas the others belong to multiple clusters (overlapping clustering). However, in a scenario where all variables belong to multiple clusters, LOVE would not perform optimally as the method assumes that there are at least some variables that belong to only a single cluster and uses these variables to determine the latent factors. Furthermore, LOVE, like any typical clustering method, is unsupervised. However, we envision being able to extend our latent model framework to classify variables in a supervised fashion (i.e., taking into account an outcome variable or outcome labels).

METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2019.03.018.

ACKNOWLEDGMENTS

This work was supported by NSF grant NSF-DMS 1712709 to F.B.

AUTHOR CONTRIBUTIONS

J.D. and F.B. conceived of, designed, and supervised the study. J.D., X.B., and M.R. analyzed data. J.D. and X.B. wrote the manuscript with inputs from all authors.

DECLARATION OF INTERESTS

The authors declare that they have no competing interests.

Received: August 27, 2018
Revised: November 26, 2018
Accepted: March 18, 2019
Published: April 26, 2019

REFERENCES

Ackerman, M.E., Das, J., Pittala, S., Broge, T., Linde, C., Suscovich, T.J., Brown, E.P., Bradley, T., Natarajan, H., Lin, S., et al. (2018). Route of immunization defines multiple mechanisms of vaccine-mediated protection against SIV. Nat. Med. *24*, 1590–1598.

Albert, R., Jeong, H., and Barabasi, A.L. (2000). Error and attack tolerance of complex networks. Nature *406*, 378–382.

Alter, G., Dowell, K.G., Brown, E.P., Suscovich, T.J., Mikhailova, A., Mahan, A.E., Walker, B.D., Nimmerjahn, F., Bailey-Kellogg, C., and Ackerman, M.E. (2018). High-resolution definition of humoral immune response correlates of effective immunity against HIV. Mol. Syst. Biol. *14*, e7881.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. *25*, 25–29.

Bezdek, J.C., Ehrlich, R., and Full, W. (1984). FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. *10*, 191–203.

Das, J., Gayvert, K.M., Bunea, F., Wegkamp, M.H., and Yu, H. (2015). ENCAPP: elastic-net-based prognosis prediction and biomarker discovery for human cancers. BMC Genomics *16*, 263.

Das, J., Mohammed, J., and Yu, H. (2012). Genome-scale analysis of interaction dynamics reveals organization of biological networks. Bioinformatics *28*, 1873–1878.

Das, J., Vo, T.V., Wei, X., Mellor, J.C., Tong, V., Degatano, A.G., Wang, X., Wang, L., Cordero, N.A., Kruer-Zerhusen, N., et al. (2013). Cross-species protein interactome mapping reveals species-specific wiring of stress response pathways. Sci. Signal. *6*, ra38.

Das, J., and Yu, H. (2012). HINT: high-quality protein interactomes and their applications in understanding human disease. BMC Syst. Biol. *6*, 92.

Eisenberg, E., and Levanon, E.Y. (2013). Human housekeeping genes, revisited. Trends Genet. *29*, 569–574.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., and Barabasi, A.L. (2000). The large-scale organization of metabolic networks. Nature *407*, 651–654.

Nepusz, T., Yu, H., and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods *9*, 471–472.

Rolland, T., Tasan, M., Charloteaux, B., Pevzner, S.J., Zhong, Q., Sahni, N., Yi, S., Lemmens, I., Fontanillo, C., Mosca, R., et al. (2014). A proteome-scale map of the human interactome network. Cell *159*, 1212–1226.

Sadanand, S., Das, J., Chung, A.W., Schoen, M.K., Lane, S., Suscovich, T.J., Streeck, H., Smith, D.M., Little, S.J., Lauffenburger, D.A., et al. (2018). Temporal variation in HIV-specific IgG subclass antibodies during acute infection differentiates spontaneous controllers from chronic progressors. AIDS *32*, 443–450.

Vo, T.V., Das, J., Meyer, M.J., Cordero, N.A., Akturk, N., Wei, X., Fair, B.J., Degatano, A.G., Fragoza, R., Liu, L.G., et al. (2016). A proteome-wide fission yeast interactome reveals network evolution principles from yeasts to human. Cell *164*, 310–323.

Wang, J., Duncan, D., Shi, Z., and Zhang, B. (2013). WEB-based GEne SeT AnaLysis Toolkit (WebGestalt): update 2013. Nucleic Acids Res. *41*, W77–W83.

Wiwie, C., Baumbach, J., and Rottger, R. (2015). Comparing the performance of biomedical clustering methods. Nat. Methods *12*, 1033–1038.

Yu, H., Braun, P., Yildirim, M.A., Lemmens, I., Venkatesan, K., Sahalie, J., Hirozane-Kishikawa, T., Gebreab, F., Li, N., Simonis, N., et al. (2008). High-quality binary protein interaction map of the yeast interactome network. Science *322*, 104–110.

ISCI, Volume *14*

Supplemental Information

Latent Model-Based Clustering for Biological Discovery

Xin Bing, Florentina Bunea, Martin Royer, and Jishnu Das

Supplement to “Latent model-based clustering for biological discovery”

1 Model introduction

In this work, we consider the problem of clustering variables into groups that are allowed to overlap. To formalize this problem, we assume that: (i) each group is represented by one unobservable latent factor, (ii) each group has observable variables that anchor it, in that they are only associated with one latent variable/group, and (iii) possibly many variables have multiple group association.

To be specific, we consider the following latent factor model, which has been first introduced and analyzed theoretically in Bing et al. (2017):

$$X = AZ + E \tag{1}$$

where $X \in \mathbb{R}^p$ is a vector of p variables to be clustered, $Z \in \mathbb{R}^K$ denotes the vector of K latent, unobservable, factors, $A \in \mathbb{R}^{p\times K}$ is the membership matrix assigning p variables to K groups , and E denotes the random error. In model (1), X, Z and E are random with $\mathbb{E}[E] = 0$ and E and Z are independent. We assume that the covariance matrix of the errors, $Cov(E) := \Gamma = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ is diagonal, and that the covariance matrix of the latent factors, $Cov(Z) = C$ is strictly positive definite. Without loss of generality, we can assume that X and Z have mean zero since we can always subtract their means. The data consists in $X_1, \ldots, X_n$ assumed to be independent and identically distributed as X given by (1):

$$X_i = AZ_i + E_i, \tag{2}$$

for $1 \le i \le n$, and with $Cov(Z_i) = C$, for all i. We emphasize that only X is observable, and that both the group number K and the membership matrix A are unknown, and are the parameters of interest, to be estimated from the data. Our final clusters of X are defined via A. To be more specific, for each group $k \in \{1, \ldots, K\}$, we cluster all X_j, $j \in \{1, \ldots, p\}$, into group k if $A_{jk} \neq 0$, that is,

$$G_k = \{j \in \{1, \ldots, p\} : A_{jk} \neq 0\}, \qquad \text{for each } k \in \{1, \ldots, K\}. \tag{3}$$

Since each row of A is allowed to have more than one non-zero entries, we expect that groups are overlapped.

Model (1) is not identifiable without further conditions, in that one can always write $AZ = AQQ^T Z$, for any orthogonal matrix Q. Therefore, without imposing further structural assumptions, the cluster allocation matrix A cannot be uniquely defined. We thus assume the following model specifications:

(i) $\sum_{k=1}^{K} |A_{jk}| \leq 1$ and $A_{j\cdot}$ is sparse, for each $j = \{1, \ldots, p\}$;

(ii) For each $k = \{1, \ldots, K\}$, there exists at least two indices $j \neq \ell \in \{1, \ldots, p\}$ such that $|A_{jk}| = |A_{\ell k}| = 1$;

(iii) $\Delta(C) := \min_{k \neq k'}(C_{kk} \wedge C_{k'k'} - |C_{kk'}|) > 0$, with $a \wedge b := \min\{a, b\}$.

Our specification (i) rules out the scaling ambiguity between A and Z. It also allows the existence of some variable which is not associated with any group, that is $A_{jk} = 0$ for all $k \in \{1, \ldots, K\}$. This is practically meaningful. For example, while analyzing a gene expression dataset, it allows genes with poor/noisy expression measurements to not be associated with any latent factor, where each latent factor could correspond to a unique biological function (as demonstrated in the main manuscript). Specification (ii) requires that, for each group, we have at least two variables that are solely associated with this group. In many areas of factor analysis, this assumption is arguably the most well-received, such as psychology (McDonald, 1999) and non-negative matrix factorization (Donoho and Stodden, 2004) in computer science, to name just a few. It has the following practical implication: if X_i satisfies (ii), then it is only related to one Z_k, for some k. Then, the cluster corresponding to the unobservable Z_k inherits the properties of X_i, which clarifies the cluster interpretation, and renders the multi-clustering association meaningful. We name variables as in (ii) the *non-mixed* variables and define its set as

$$\mathcal{I} = \{I_1, \ldots, I_K\}, \qquad I_k = \{j \in \{1, \ldots, p\} : |A_{jk}| = 1, A_{jk'} = 0, \forall k' \neq k\}.$$

Correspondingly, we name the variables in the complement set *mixed* variables. Specification (iii) implies $|Z_k| \neq |Z_{k'}|$ almost surely for any two latent variables, and can be viewed as the minimal assumption to make two latent variables, and therefore two clusters, distinguishable.

Under model (1) and assuming that (i) - (iii) hold, Theorems 1 and 2 in Bing et al. (2017) show that K and $\mathcal{I}$ can be uniquely determined from $\Sigma := Cov(X)$ up to a group permutation. Moreover, the allocation matrix A can also be determined, uniquely, from Σ up to a $K \times K$ signed label permutation.

In practice, instead of having access to the theoretical covariance matrix Σ, we only have access to the data that consists in n i.i.d. copies of the vector X, organized in the $n \times p$ data matrix $\mathbf{X} := (X_1, \ldots, X_n)^T$, where by slight abuse of notation we denote by $X_i \in \mathbb{R}^p$ the i-th measurement on the vector X. By using $\widehat{\Sigma} = \frac{1}{n} X^T X$ as an estimator of Σ, and assuming that X has sub-Gaussian tails, Algorithms 1 and 2 in Bing et al. (2017) yield statistically accurate estimates $\widehat{\mathcal{I}}$ and $\widehat{A}$ of $\mathcal{I}$ and A, respectively, as shown in Theorems 3 and 4 of Bing et al. (2017), under suitable conditions. After estimating A, we can estimate the clusters G_k by

$$\widehat{G}_k = \{j \in [p] : \widehat{A}_{jk} \neq 0\}, \qquad \text{for each } k \in \{1, \ldots, \widehat{K}\}. \tag{4}$$

Under suitable conditions, Part 3 of Remark 4 in Bing et al. (2017) guarantees that $\widehat{G}_k = G_k$ for all $1 \leq k \leq K$, with high probability, up to label permutation. This shows that the clusters defined by our latent variable model can be estimated consistently, via a scalable algorithm.

Remark: The original paper (Bing et al., 2017) focuses on clustering varaibles based on their covariance structure which is a linear measure of the dependency. In particular, model (1) is an example when the linear similarity measure $\Sigma := Cov(X)$ has the following decomposition

$$\Sigma = ACA^T + \Gamma. \tag{5}$$

Since the definition of clusters in (3) is via A and A is identified from the linear similarity measure Σ, we can naturally extend this to any non-linear similarity measure of variables. To be precisely, let $\mathcal{K} \in \mathbb{R}^{p \times p}$ be any similarity measure of those p variables. If $\mathcal{K}$ satisfies the following decomposition

$$\mathcal{K} = ACA^T + \Gamma$$

with A and C following our model specifications (i) - (iii) and Γ being diagonal, then as long as one can estimate $\mathcal{K}$ well from the observed data, LOVE algorithm is still applicable to estimate the matrix A, from which the clusters can be constructed.

2 Identifiability and estimation of $I, \mathcal{I}, A$ and G

The details given in this section have been developed in Bing et al. (2017). For the convenience of the reader, and to keep this work self-contained, we repeat them here.

We first present two theorems which guarantee that I, $\mathcal{I}$, A and G are identifiable. The criterion of Theorem 1 is constructive for the later estimation.

2.1 Identifiability

Let

$$M_i := \max_{j \in [p] \setminus \{i\}} |\Sigma_{ij}| \tag{6}$$

be the largest absolute value of the entries of row i of Σ excluding $|\Sigma_{ii}|$. Let S_i be the set of indices for which M_i is attained:

$$S_i := \big\{ j \in [p] \setminus \{i\} : |\Sigma_{ij}| = M_i \big\}. \tag{7}$$

Theorem 1. *Assume that model (1) and (i) - (iii) hold. Then:*

(**a**) $i \in I \quad \Longleftrightarrow \quad M_i = M_j$ *for all* $j \in S_i$.

(**b**) *The pure variable set I can be determined uniquely from $\Sigma := Cov(X)$. Moreover, its partition $\mathcal{I} := \{I_a\}_{1 \le a \le K}$ is unique and can be determined from Σ up to label permutations.*

The identifiability of the allocation matrix A and that of the collection of clusters $\mathcal{G} = \{G_1, \ldots, G_k\}$ in (3) use the results from Theorem 1 in crucial ways. We state the result in Theorem 2 below.

Theorem 2. *Assume that Model (1) with (i) - (iii) holds. Then, there exists a unique matrix A, up to a signed permutation, such that $X = AZ + E$. This implies that the associated overlapping clusters G_a, for $1 \le a \le K$, are identifiable, up to label switching.*

2.2 Estimation

We develop estimators from the observed data, which is assumed to be a sample of n i.i.d. copies $X^{(1)}, \ldots, X^{(n)}$ of $X \in \mathbb{R}^p$, where p is allowed to be larger than n. Our estimation procedure consists of the following four steps:

(1) Estimate the pure variable set I, the number of clusters K and the partition $\mathcal{I}$;

(2) Estimate A_I, the submatrix of A with rows $A_{i\cdot}$ that correspond to $i \in I$;

(3) Estimate A_J, the submatrix of A with rows $A_{j\cdot}$ that correspond to $j \in J$;

(4) Estimate the overlapping clusters $\mathcal{G} = \{G_1, \ldots, G_K\}$.

2.2.1 Estimation of I and $\mathcal{I}$

Given the different nature of their entries, we estimate the submatrices A_I and A_J separately. For the former, we first estimate I and its partition $\mathcal{I} = \{I_1, \ldots, I_K\}$, which can be both uniquely constructed from Σ, as shown by Theorem 1. We use the constructive proof of Theorem 1 for this step, replacing the unknown Σ by the sample covariance matrix

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} X^{(i)}(X^{(i)})^T.$$

Specifically, we iterate through the index set $\{1, 2, \ldots, p\}$, and use the sample version of part (**a**) of Theorem 1 to decide whether an index i is pure. If it is not deemed to be pure, we add it to the set that estimates J. Otherwise, we retain the estimated index set $\widehat{S}_i$ of S_i defined in (7), which corresponds to an estimator of M_i given by (6). We then use the constructive proof of part (**b**) of Theorem 1 to declare $\widehat{S}_i \cup \{i\} := \widehat{I}^{(i)}$ as an estimator of one of the partition sets of $\mathcal{I}$. The resulting procedure has complexity $O(p^2)$, and we give all the specifics in Algorithm 1 of Section 2.2.5. The algorithm requires the specification of a tuning parameter δ, which will be discussed in Section 3.

2.2.2 Estimation of the allocation submatrix A_I

Given the estimators $\widehat{I}$, $\widehat{K}$ and $\widehat{\mathcal{I}} = \{\widehat{I}_1, \ldots, \widehat{I}_{\widehat{K}}\}$ from Algorithm 1, we estimate the matrix A_I by a $|\widehat{I}| \times \widehat{K}$ matrix with rows $i \in \widehat{I}$ consisting of $\widehat{K} - 1$ zeros and one entry equal to either $+1$ or -1 as follows. For each $a \in [\widehat{K}]$,

(1) Pick an element $i \in \widehat{I}_a$ at random, and set $\widehat{A}_{ia} = 1$. Note that $\widehat{A}_{ia}$ can only be $+1$ or -1 by the definition of a pure variable.

(2) For the remaining $j \in \widehat{I}_a \setminus \{i\}$, we set $\widehat{A}_{ja} = \text{sign}(\widehat{\Sigma}_{ij})$.

This procedure induces a partition of $\widehat{I}_a = \widehat{I}_a^1 \cup \widehat{I}_a^2$, where $\widehat{I}_a^1$ and $\widehat{I}_a^2$ are defined below:

$$\begin{cases} \widehat{A}_{ka} = \widehat{A}_{la}, & \text{for } k, l \in \widehat{I}_a^1 \text{ or } k, l \in \widehat{I}_a^2 \\ \widehat{A}_{ka} \neq \widehat{A}_{la}, & \text{for } k \in \widehat{I}_a^1 \text{ and } l \in \widehat{I}_a^2 \end{cases}. \tag{8}$$

2.2.3 Estimation of the allocation submatrix A_J

We continue by estimating the matrix A_J, row by row. To motivate our procedure, we begin by highlighting the structure of each row $A_{j\cdot}$ of A_J, for $j \in J$. We recall that $A_{j\cdot}$ is sparse, with $\|A_{j\cdot}\|_1 \leq 1$, for each $j \in J$, as specified by assumption (i). In addition, model (1) subsumes a further constraint on each row $A_{j\cdot}$ of A, as explained below. To facilitate notation, we rearrange Σ, A and $\Gamma := \text{Cov}(E)$ as follows:

$$\Sigma = \begin{bmatrix} \Sigma_{II} & \Sigma_{IJ} \\ \Sigma_{JI} & \Sigma_{JJ} \end{bmatrix}, \quad A = \begin{bmatrix} A_I \\ A_J \end{bmatrix} \quad \text{and} \quad \Gamma = \begin{bmatrix} \Gamma_{II} & 0 \\ 0 & \Gamma_{JJ} \end{bmatrix}.$$

Model (1) implies the following decomposition of the covariance matrix Σ of X:

$$\Sigma = \begin{bmatrix} \Sigma_{II} & \Sigma_{IJ} \\ \Sigma_{JI} & \Sigma_{JJ} \end{bmatrix} = \begin{bmatrix} A_I C A_I^T & A_I C A_J^T \\ A_J C A_I^T & A_J C A_J^T \end{bmatrix} + \begin{bmatrix} \Gamma_{II} & 0 \\ 0 & \Gamma_{JJ} \end{bmatrix}.$$

In particular, $\Sigma_{IJ} = A_I C A_J^T$. Thus, for each $i \in I_a$ with some $a \in [K]$ and $j \in J$, we have

$$A_{ia}\Sigma_{ij} = A_{ia}^2 \sum_{b=1}^{K} A_{jb} C_{ab} = \sum_{b=1}^{K} A_{jb} C_{ab} = C_{a\cdot}^T A_{j\cdot}. \tag{9}$$

Averaging display (9) over all $i \in I_a$ yields

$$\frac{1}{|I_a|} \sum_{i \in I_a} A_{ia} \Sigma_{ij} = C_{a\cdot}^T A_{j\cdot}, \quad \text{for each } a \in [K]. \tag{10}$$

For each $j \in J$, we let

$$\beta^j := A_{j\cdot}$$

and

$$\theta^j = \left(\frac{1}{|I_1|} \sum_{i \in I_1} A_{i1} \Sigma_{ij}, \ldots, \frac{1}{|I_K|} \sum_{i \in I_K} A_{iK} \Sigma_{ij} \right)^T. \tag{11}$$

Since $A_{ia} \in \{-1, 1\}$, for each $i \in I_a$ and $a \in [K]$, the entries of θ^j are respective averages of the sign corrected entries of Σ corresponding to the partition of the pure variable set. Summarizing, modeling assumption (i) and equation (10) above show that the estimation of A_J reduces to estimating, for each $j \in J$, a K-dimensional vector β^j that is sparse, with norm $\|\beta^j\|_1 \leq 1$, and that satisfies the equation

$$\theta^j = C\beta^j.$$

Both C and θ^j, for each $j \in J$, can be estimated directly from the data as follows. For each $j \in \widehat{J}$, we estimate the a-th entry of θ^j by

$$\widehat{\theta}^j_a = \frac{1}{|\widehat{I}_a|} \sum_{i \in \widehat{I}_a} \widehat{A}_{ia} \widehat{\Sigma}_{ij}, \; a \in [\widehat{K}], \tag{12}$$

and compute

$$\widehat{C}_{aa} = \frac{1}{|\widehat{I}_a|(|\widehat{I}_a| - 1)} \sum_{i,j \in \widehat{I}_a, i \neq j} |\widehat{\Sigma}_{ij}|, \qquad \widehat{C}_{ab} = \frac{1}{|\widehat{I}_a||\widehat{I}_b|} \sum_{i \in \widehat{I}_a, j \in \widehat{I}_b} \widehat{A}_{ia}\widehat{A}_{ib}\widehat{\Sigma}_{ij}, \tag{13}$$

for each $a \in [\widehat{K}]$ and $b \in [\widehat{K}] \setminus \{a\}$ to form the estimator $\widehat{C}$ of C. The estimates (12) and (13) rely crucially on having first estimated the pure variables and their partition, according to the steps described in Sections 2.2.1 and 2.2.2 above.

We have developed a computationally efficient method to estimate β^j. We exploit the fact that the square matrix C is invertible and take the equation $\beta^j = C^{-1}\theta^j$ as our starting point. The idea is to first construct a pre-estimator $\bar{\beta}^j = \widehat{\Omega}\widehat{\theta}^j$, based on an appropriate estimator $\widehat{\Omega}$ of the precision matrix $\Omega := C^{-1}$, followed by a sparse projection of $\bar{\beta}^j$. Alternatively, and recommended to speed up the computation, we could use a simple hard threshold operation in the second step. To estimate Ω, we propose the linear program

$$(\widehat{\Omega}, \hat{t}) \quad = \arg \min_{t \in \mathbb{R}^+, \; \Omega \in \mathbb{R}^{\widehat{K} \times \widehat{K}}} t \tag{14}$$

subject to

$$\Omega = \Omega^T, \quad \|\Omega\widehat{C} - I\|_\infty \leq \lambda t, \quad \|\Omega\|_{\infty,1} \leq t, \tag{15}$$

with tuning parameter λ. After we compute $\bar{\beta}^j = \widehat{\Omega}\widehat{\theta}^j$, for each $j \in \widehat{J}$, we solve the following optimization problem

$$\widehat{\beta}^j = \arg \min_{\beta \in \mathbb{R}^{\widehat{K}}} \|\beta\|_1 \tag{16}$$

subject to

$$\|\beta - \bar{\beta}^j\|_\infty \leq \mu, \tag{17}$$

for some tuning parameter μ that is proportional to $\|C^{-1}\|_{\infty,1}$, to obtain our final estimate $\widehat{\beta}^j$ as the optimal solution of this linear program. This solution is also sparse and properly scaled, in accordance to our model specification (i). Then, $\widehat{A}_{\widehat{J}}$ is the matrix with rows $\widehat{\beta}^j$, for $j \in \widehat{J}$. Our final estimator $\widehat{A}$ of A is obtained by concatenating $\widehat{A}_{\widehat{I}}$ and $\widehat{A}_{\widehat{J}}$.

2.2.4 Estimation of the overlapping groups

Recalling the definition of groups in (3), the overlapping groups are estimated by

$$\widehat{\mathcal{G}} = \{\widehat{G}_1, \ldots, \widehat{G}_{\widehat{K}}\}, \quad \widehat{G}_a = \{i \in [p] : \widehat{A}_{ia} \neq 0\}, \text{ for each } a \in [\widehat{K}]. \tag{18}$$

Variables X_i that are associated (via $\widehat{A}$) with the same latent factor Z_a are therefore placed in the same group $\widehat{G}_a$. To accommodate potential pure noise variables, we further define

$$G_0 := \{j \in \{1, \ldots, p\} : \ A_{ja} = 0, \text{ for all } a \in \{1, \ldots, K\}\} \tag{19}$$

as the pure noise cluster. We can estimate G_0 in (19) by

$$\widehat{G}_0 = \{i \in [p] : \widehat{A}_{ia} = 0, \text{ for all } a \in [\widehat{K}]\}. \tag{20}$$

However, our main focus is on $\mathcal{G}$ because it completely determines G_0.

2.2.5 LOVE: A Latent variable model approach for OVErlapping clustering.

We give below the specifics of Algorithm 1, motivated in Section 2.2.1, and summarize our final algorithm, LOVE in Algorithm 2.

3 Identifying tuning parameters

In order to apply Algorithms 1 and 2 in Bing et al. (2017), there are three tuning parameters (δ, λ, μ) to be chosen. We clarify them in the sequel.

1. The tuning parameter δ is used for finding $\widehat{\mathcal{I}}$ and defined as

$$\delta := \max_{1 \leq i < j \leq p} |\widehat{\Sigma}_{ij} - \Sigma_{ij}|.$$

   When X has sub-Gaussian tail, we know $\delta = c\sqrt{\log(p \vee n)/n}$ for some constant c depending on the variance of X with $a \vee b := \max\{a, b\}$. Thus, we can choose a fine grid for the leading constant of δ with rate equal to $\sqrt{\log(p \vee n)/n}$, and use cross-validation to find the best leading constant (cf. page 25 in Bing et al. (2017)).

2. To see the role of λ, note that we need to estimate C^{-1} in the Algorithm 2 by using $\widehat{C}$ which is not guaranteed to be positive definite. Thus, a linear program is proposed in (14) and (15) in Bing et al. (2017) to deal with this issue, and the resulting estimator of C^{-1} is denoted by $\widehat{\Omega}$. The rate of λ is the same as that of δ.

3. Finally, in order to obtain sparsity of $\widehat{A}$, a soft-thresholding procedure is proposed for the estimation of each row of A as (16) and (17) in Bing et al. (2017). The tuning parameter μ controls the sparsity of the resultant $\widehat{A}$. The rate of μ is $\|C^{-1}\|_{\infty,1}\sqrt{\log(p \vee n)/n}$ where $\|C^{-1}\|_{\infty,1} := \max_k \sum_{k'=1}^{K} |C_{kk'}|$. In practice, we can replace C^{-1} by its estimate obtained in the previous step.

Practically, as long as we use cross-validation to select δ^{cv}, Bing et al. (2017) recommends using $\lambda = \delta^{cv}$ and $\mu = \|\widehat{\Omega}\|_{\infty,1}\delta^{cv}$ where $\widehat{\Omega}$ is the estimate of C^{-1}. It is worth mentioning that, the algorithm is very robust to μ and λ as long as δ is selected.

4 Gene expression dataset

The data has expression values corresponding to $p = 16134$ genes as columns and $n = 114$ time-points as rows.

- For *LOVE*, we select δ from the grid $c\sqrt{\log p/n}$ with $c \in \{0.3, 0.31, \ldots, 0.49, 0.50\}$ by cross-validation . The selected δ^{cv} is $0.43\sqrt{\log p/n}$. We then use $\lambda = \delta^{cv}$ as recommended in Bing et al. (2017). For selecting μ, recall that it controls the sparsity of $\widehat{A}$ which determines the size of each group via (4). We tune μ to have a large number of clusters with size between $[50, 1000]$ and we end up using $\mu = 1.91\|\widehat{\Omega}\|_{\infty,1}\delta^{cv}$.

- For *FuzzyCmeans*, we use the function `cmeans` in the `R`-package `e1071` (Bezdek, 1981). Recall that $\mathbf{X} \in \mathbb{R}^{114\times16134}$ where the rows correspond to independent observations. Therefore, *FuzzyCmeans* is designed to cluster rows instead of columns. This suggests that it is meaningless to naively apply *FuzzyCmeans* to $\mathbf{X}^T$ to cluster 16134 genes. Moreover, the number of observations is too small to make the result meaningful. To remedy this, we try to apply *FuzzyCmeans* to $\mathbf{X}^T\mathbf{X}$ instead; however it is still outperformed by *LOVE*.

The number of clusters needs to be pre-determined. To make the results comparable with *LOVE*, we specify it equal to 1222 which is the number of clusters obtained from *LOVE*. Since *FuzzyCmeans* doesn't yield sparsity of its membership matrix (the quantity analogous to our $\widehat{A}$), we manually threshold it by using different thresholding levels. To tune the thresholding level, we choose it via a grid such that it yields the largest proportion of clusters with size within $[50, 1000]$, in accordance of the biological expectation. The thresholded membership matrix is used to define clusters as (4).

- For *ClusterOne*, we first compute the weighted gene co-expression network using pairwise correlations as edge weights. Significant edges are defined when the (unsigned) correlation is superior to threshold 0.5, with the added condition that the Benjamini-Hochberg multiple testing correction produces a p-value associated with the t-test inferior to 0.05. On that weighted correlation graph we apply the ClusterOne method aiming to detect highly cohesive, potentially overlapping complexes (Nepusz et al., 2012). We generate overlapping clusters using all default parameters, specifically a default density threshold of 0.3.

To calculate the under-representation index of housekeeping genes in each cluster, we first measure the significance of the under-representation using a hypergeomtric test. We then report the distributon of under-representation indices as the $-\log$ of the 100 most significant p-values for each method.

5 Dataset of progressors and controllers

We have 19 subjects (10 progressors and 9 controllers). For each subject, we have 18 measurements (antibody-dependent effector functions and antibody titers) across 4 time points. We use all time points for each subject and end up with a 72×19 data matrix $\mathbf{X}$. There are 1.5% missing values and we impute them using means.

LOVE could directly be applied to this dataset if the rows were independent and identically distributed. However, measurements of effector functions and titers at different time points are independent, but not necessarily identically distributed. We explain below why *LOVE* can still be used in in this situation. To start with, we note that for this data set $n = 72$ and $p = 19$, with the interpretation of n and p given by model (1), but where the data consists now in independent observations $X_i \in \mathbb{R}^{19}$, $1 \le i \le n$, and each observation is allowed to have a different distribution, in that for each $i \in \{1, \dots, n\}$ we have

$$X_i = AZ_i + E_i, \tag{21}$$

with $Cov(Z_i) = G^i$. This emphasizes the fact that the vector containing the i-th clinical phenotype (controller vs progressor) for all 19 subjects is linked, via an allocation matrix A, to an unobservable latent factor vector $Z_i \in \mathbb{R}^K$, but that unlike (2), where $Cov(Z_i) = C$, for all i, here we allow the covariance structure of the latent factors to change with i. The random error term $E_i \in \mathbb{R}^p$ is assumed to have mean zero and diagonal covariance matrix. Then, (21) implies

$$M^i := \mathbb{E}[X_i X_i^T] = A\mathbb{E}[Z_i Z_i^T]A^T + \mathbb{E}[E_i E_i^T] := AG^iA^T + \Gamma^i,$$

with Γ^i being diagonal, and recalling that G^i denotes the covariance matrix of the latent factors, for $i = 1, \dots, n$. This further yields

$$\overline{M} := \frac{1}{n}\sum_{i=1}^{n} M^i = A\left(\frac{1}{n}\sum_{i=1}^{n} G^i\right)A^T + \frac{1}{n}\sum_{i=1}^{n}\Gamma^i := A\overline{G}A^T + \overline{\Gamma}.$$

A close look at the development of the *LOVE* algorithm in Bing et al. (2017) reveals the fact that it can be employed to estimate an allocation matrix A from a decomposition as above whenever: A satisfies the requirements (i) and (ii) introduced in the first section, $\overline{G}$ satisfies the mild requirement (iii) from the first section, and $\overline{M}$ can be estimated well from the data. Assuming that the theoretical model requirements are satisfied, an estimator of $\overline{M} \in \mathbb{R}^{p\times p}$ is given by $\widehat{M} = n^{-1}\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{p\times p}$, which can be used as the input of the *LOVE* algorithm. One of the tuning parameters, δ, is defined now as $\delta := \max_{1\le i<j\le p}|\widehat{M}_{ij} - \overline{M}_{ij}|$, and will continue to be proportional to $n^{-1/2}$, up to logarithmic terms, under appropriate conditions on the error distributions. The other two tuning parameters will change accordingly, but will have the same interpretation as in the previous analysis. Specifically:

- For *LOVE*, we choose δ from $c\sqrt{\log n/n}$ with $c \in \{0.01, 0.02, \dots, 0.19, 0.2\}$ and we ended up with $\delta^{cv} = 0.1\sqrt{\log n/n}$. We choose $\lambda = 1.36\delta^{cv}$ via the criterion on page 25 in Bing et al. (2017). Finally, $\mu = \|\widehat{\Omega}\|_{\infty,1}\delta^{cv}$ is used as recommended; for this analysis $\widehat{\Omega}$ is an estimator of the inverse matrix $\overline{G}^{-1}$. We demonstrate in Figure 3 that our results are robust to the choice of λ. We use the largest membership weight to assign each subject to one cluster.The initial groups are merged to generate 2 clusters (a-priori expectation is 2 clusters as there are two different clinical phenotypes).

- For *FuzzyCmeans*, using the arguments outlined in the previous section, we apply *FuzzyCmeans* to $X^T X$ by using the true group number $K = 2$. Since we are interested in the performance of non-overlapping, we manually assign each subject to one cluster by using the largest membership weight. The fuzziness parameter is the default value as the resultant clusters change little when we vary the fuzziness paramter.
- For *ClusterOne*, we use the same procedure as above, we apply the method with default parameters on the weighted gene co-expression network using pairwise correlations as edge weights. The significant edges are defined as the previous section. However, here we use an FDR cutoff of 0.2 so that the correlation graph is not too sparse.

6 Dataset of vaccine-induced humoral immune responses

We have 60 primates in 3 vaccination arms - IM239, IM mosaic and AE239. For each primate, we measured 191 different humoral immune responses post vaccination. The data matrix $\mathbf{X} \in \mathbb{R}^{191\times 60}$ contains 60 columns representing 60 primates and 191 rows of measurements. We impute the missing values using k-nearest-neighbour imputation where k = 5. Since each humoral immune response is on a different scale, we standardize each measurement to have unit variance. Note that the 191 measurements here are independent but not necessarily identically distributed. From the explanations of the previous dataset (Dataset of progressors and controllers), $LOVE$ cound be directly applied to $n^{-1}\mathbf{X}^T\mathbf{X}$ in this dataset.

- For $LOVE$, we choose $\delta = c\sqrt{\log n/n}$ from the grid $c \in \{0.05, 0.06, \ldots, 0.39, 0.4\}$ which gives $\delta^{cv} = 0.11\sqrt{\log n/n}$. The chosen λ^{cv} is $0.8\delta^{cv}$ via the criterion on page 25 in Bing et al. (2017). The thresholding parameter μ is set to be $0.85\|\widehat{\Omega}\|_{\infty,1}\delta^{cv}$ in order to guarantee that no primate is excluded. To obtain non-overlapping groups, we assign each primate to the group having the largest membership weight. The initial groups are merged to generate 3 clusters (a-priori expectation is 3 clusters as there are 3 different vaccination arms).

- For *FuzzyCmeans*, we use the true group number $K = 3$ as input and assign each primate based on the largest weight of the membership matrix to obtain non-overlapping groups. Since the fuzziness parameter controls the level of overlap in *FuzzyCmeans*, we manually tune it from the grid of $\{1.1, 1.2, \ldots, 2.9, 3\}$ based on the best performance in terms of the confusion matrix. We end up with a fuzziness parameter of 1.5.

- For *ClusterOne*, since the pairwise correlation is rather weak in this dataset, *ClusterOne* tends not to assign many subjects into any group. To obtain the upper bound of performance for *ClusterOne*, we used effect size and significance thresholds that allowed most primates to be assigned to at least one group. We used an FDR cutoff of 0.2 as in the earlier section. Even after tuning parameters to ensure that most primates are assigned to at least one cluster, some remained unassigned. Each of these primates was randomly assigned to one of the clusters.

References

Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algoritms.* Plenum Press.

Bing, X., Bunea, F., Yang, N. and Wegkamp, M. (2017). Adaptive estimation in structured factor models with applications to overlapping clustering. *Under review in the Annals of Statistics (arXiv:1704.06977v3)* .

Donoho, D. and Stodden, V. (2004). When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems 16* (S. Thrun, L. K. Saul and P. B. Schölkopf, eds.). MIT Press, 1141–1148.

McDonald, R. (1999). *Test Theory: A Unified Treatment.* Taylor & Francis.

Nepusz, T., Yu, H. and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods* **9** 471 – 472.

Algorithm 1 Estimate the partition of the pure variables $\mathcal{I}$ by $\widehat{\mathcal{I}}$

```
procedure PureVar($\widehat{\Sigma}$, $\delta$)
    $\widehat{\mathcal{I}} \leftarrow \emptyset$.
    for all $i \in [p]$ do
        $\widehat{I}^{(i)} \leftarrow \left\{l \in [p] \setminus \{i\} : \max_{j \in [p] \setminus \{i\}} |\widehat{\Sigma}_{ij}| \le |\widehat{\Sigma}_{il}| + 2\delta\right\}$
        $Pure(i) \leftarrow True$.
        for all $j \in \widehat{I}^{(i)}$ do
            if $\left| |\widehat{\Sigma}_{ij}| - \max_{k \in [p] \setminus \{j\}} |\widehat{\Sigma}_{jk}| \right| > 2\delta$ then
                $Pure(i) \leftarrow False$,
                break
            end if
        end for
        if $Pure(i)$ then
            $\widehat{I}^{(i)} \leftarrow \widehat{I}^{(i)} \cup \{i\}$
            $\widehat{\mathcal{I}} \leftarrow$ Merge($\widehat{I}^{(i)}$, $\widehat{\mathcal{I}}$)
        end if
    end for
    return $\widehat{\mathcal{I}}$ and $\widehat{K}$ as the number of sets in $\widehat{\mathcal{I}}$.
end procedure

function Merge($\widehat{I}^{(i)}$, $\widehat{\mathcal{I}}$)
    for all $G \in \widehat{\mathcal{I}}$ do                          ▷ $\widehat{\mathcal{I}}$ is a collection of sets
        if $G \cap \widehat{I}^{(i)} \neq \emptyset$ then
            $G \leftarrow G \cap \widehat{I}^{(i)}$                   ▷ Replace $G \in \widehat{\mathcal{I}}$ by $G \cap \widehat{I}^{(i)}$
            return $\widehat{\mathcal{I}}$
        end if
    end for
    $\widehat{I}^{(i)} \in \widehat{\mathcal{I}}$                     ▷ add $\widehat{I}^{(i)}$ in $\widehat{\mathcal{I}}$
    return $\widehat{\mathcal{I}}$
end function
```

Algorithm 2 The LOVE procedure for overlapping clustering.

```
Require: $\widehat{\Sigma}$ from I.I.D. data $(X^{(1)}, ..., X^{(n)})$, the tuning parameters $\delta$, $\lambda$ and $\mu$.
Apply Algorithm 1 to obtain the number of clusters $\widehat{K}$, the estimated set of pure variables $\widehat{I}$ and its partition of $\widehat{\mathcal{I}}$.
Estimate $A_I$ by $\widehat{A}_{\widehat{I}}$ from (8).
Estimate $C^{-1}$ by $\widehat{\Omega}$ from (14) and $\bar{\beta}^j$ for each $j \in \widehat{J}$.
Estimate $A_J$ by $\widehat{A}_{\widehat{J}}$ from (16). Combine $\widehat{A}_{\widehat{I}}$ with $\widehat{A}_{\widehat{J}}$ to obtain $\widehat{A}$.
Estimate overlapping groups $\widehat{\mathcal{G}} = \{\widehat{G}_1, ..., \widehat{G}_{\hat{K}}\}$ from (18) by using $\widehat{A}$.
Output $\widehat{A}$ and $\widehat{\mathcal{G}}$.
```