Cluster LOCO: Feature Importance for Interpreting Clusters

Anonymous authors
Paper under double-blind review

Abstract

Clustering is widely used for exploratory analysis and scientific discovery, driving insights from market segmentation to biological data analysis, but its outputs can be difficult to interpret, audit, and reproduce as modern datasets become increasingly large and complex. Reliable use of clustering requires understanding which features drive the discovered structure, yet feature-level explanations for clustering remain scarce compared with methods in supervised learning. Furthermore, existing clustering feature importance scores are often tied to specific algorithms and data assumptions. To address these challenges, we propose Cluster LOCO (Leave-One-Covariate-Out), a family of model-agnostic feature importance scores for clustering. Cluster LOCO is built on feature occlusion and clustering generalizability, defined as whether cluster labels learned on one subset of the data can be accurately predicted on held-out samples. For any chosen clustering algorithm, Cluster LOCO quantifies a feature's importance by measuring how much its removal degrades generalizability. We first introduce Cluster LOCO-Split, which relies on data splitting, and then extend it to Cluster LOCO-MP, a minipatch ensemble-based version designed for large-scale data. Across synthetic simulations and an application to cell-type discovery in single-cell transcriptomics, we show that Cluster LOCO more reliably recovers informative features than existing clustering feature importance methods.

1 Introduction

A fundamental task in unsupervised learning, clustering is used across disciplines ranging from the social sciences, astrophysics to biology (Handcock et al., 2007; Materne, 1978; Xu & Wunsch, 2010) to draw insights from data by forming groups or partitions. Yet clustering is not defined by one canonical objective. As emphasized by Luxburg et al. (2012), clustering has several use cases: it may be used for preprocessing, to organize, compress, or denoise data; for exploration, to reveal unknown structure and generate hypotheses; or for confirmation, to validate hypothesized groupings or support scientific discoveries. Because clustering intervenes upstream in the data science life-cycle (Yu & Kumbier, 2020), its outputs influence downstream analysis, modeling, and scientific interpretation. It is therefore consequential that clustering-driven conclusions be reliable, trustworthy, and reproducible.

At the same time, *reliability* is difficult to assess for clustering solutions because of the underlying assumptions and choices made by clustering models: for example, K-means requires specifying the number of clusters, and the algorithm will return exactly that many groups whether or not such structure is meaningful in the data (Allen et al., 2023). This challenge is amplified in modern datasets, where complex nonlinear patterns, interactions, and high dimensionality can limit the effectiveness of classical methods (Kriegel et al., 2009). In response, practitioners increasingly rely on deep clustering models (Min et al., 2018; Li et al., 2023) or heavily feature-engineered workflows (Ding & He, 2004; Jolliffe & Cadima, 2016; Jin & Wang, 2016; Stuart et al., 2019; Wolf et al., 2018). While these approaches can improve the detection of complex structure (e.g. nonlinearity) or domain-dependent specificity (e.g. zero-inflated data), they can also make the resulting clusters harder to understand and audit. For instance, in genomics, clustering has enabled meaningful discoveries of cell types and markers (Villani et al., 2017) while, on the other hand, computational studies centered on clustering have raised persistent concerns about reproducibility (Gibson, 2022). For clustering to support rigorous discovery, we therefore need tools that clarify why a clustering solution arises. In particular,

feature-level explanations can help identify which features the clustering solution relies on and promote *trust* (Gan et al., 2025).

We address this need by bringing the perspective of interpretable machine learning (IML) to clustering, establishing a useful notion of feature importance for clustering solutions. In many clustering applications, features are themselves meaningful and interpretable: they are the genes in transcriptomic data, words in text data, behavioral or measured attributes in the social sciences and often the object of downstream analysis. For example, in single-cell genomics, practitioners commonly interpret clusters by identifying "marker genes" i.e. genes that differ across the discovered clustering groups (Villani et al., 2017), through differential expression analysis, where genes are tested across clusters (Kiselev et al., 2019). While useful for annotation, this workflow can raise post-selection inference, or "double dipping" concerns because the same data are used both to define the clusters and to assess the features that distinguish them, a practice known to inflate the false discovery rate of significant genes (Zhang et al., 2019; DenAdel et al., 2024; Song et al., 2023). Moreover, features that differ across clusters are not necessarily the features that produced the clustering solution. We therefore focus on feature importance for clustering, which asks directly how much each feature contributes to the clustering structure itself.

This IML perspective is well established in supervised learning, where feature importance methods are widely used to explain model predictions (Molnar, 2018). In clustering, however, feature-level interpretability remains comparatively underdeveloped. Existing work has largely followed two directions: intrinsically interpretable clustering algorithms, built on interpretable supervised methods such as decision trees with a modified objective for clustering (Hu et al., 2024), and post-hoc explanations tailored to specific algorithms, mainly K-means (Nápoles et al., 2024; Kauffmann et al., 2024). While useful, these approaches can be difficult to scale to large datasets, might make particular data assumptions or constrain the practitioner to specific clustering models. On the other hand, feature selection in clustering enables handling large scale data and has been well studied with models leveraging sparsity via regularization (Witten & Tibshirani, 2010; Wang et al., 2018), filters or wrapper methods (Xing & Karp, 2001; Dash et al., 2002; Roth & Lange, 2003; Alelyani et al., 2014). While related, feature selection and feature importance answer different questions. Feature selection asks which features should be used to construct a clustering solution, often by optimizing a sparsity or clustering quality criterion. In contrast, feature importance asks, after a clustering solution has been obtained, which features contributed most to it. We instead seek a post-hoc interpretation of a chosen clustering model, separating the explanation of a clustering solution from the decisions used to produce it.

To formulate such a post-hoc interpretation, we turn to feature importance methods from supervised interpretable machine learning. In supervised learning, feature-level interpretability includes a wide range of model-specific and model-agnostic techniques. For model-agnostic techniques, we find three main types of feature importance metrics: feature permutation introduced via model-class reliance (Fisher et al., 2019; Breiman, 2001) inspects the model's "reliance" to each feature via its error; Shapley values (Shapley, 1953; Lundberg & Lee, 2017) and extensions (Sundararajan & Najmi, 2020; Mase et al., 2021; Verdinelli & Wasserman, 2023), a popular metric based in game-theoretical axioms which distributes feature value across features fairly; and feature ablation or occlusion which explains the change in prediction when a feature is removed (Lei et al., 2017; Rinaldo et al., 2019; Verdinelli & Wasserman, 2023).

In this paper, we propose a model-agnostic feature importance framework for clustering based on feature occlusion. Our approach is motivated by the Leave-One-Covariate-Out framework Lei et al. (2017): remove a feature and measure how much the clustering solution changes. If removing a feature substantially alters the clustering, then the feature is important for the discovered structure; if the clustering remains largely unchanged, then the feature is less important. This definition is straightforward to interpret, post hoc as it is applied after a clustering method has already been chosen, and is therefore flexible to complex clustering workflows. Our contributions are as follows: first, we introduce Cluster LOCO, a family of model-agnostic feature importance metrics for clustering based on feature occlusion via Cluster-LOCO-Split. Second, we develop Cluster LOCO-MP, a scalable minipatch-based extension designed for high-dimensional data. Third, we validate our approach in simulated settings, including low-dimensional examples with complex nonlinear structure and high-dimensional regimes, and compare it against existing feature importance methods. Lastly, we demonstrate our framework in a single-cell transcriptomics application where important features for clustering are biologically meaningful.

2 Feature importance scores for clustering via LOCO

2.1 Background for Cluster LOCO

Before introducing our Cluster LOCO family of metrics, we review two existing notions that inspired our metric: generalizability and LOCO.

Generalizability in clustering, also sometimes coined as **predictability**, was first proposed in the context of model selection by Lange et al. (2004) and Tibshirani & Walther (2005) as an alternative to stability-based validation techniques relying on sampling or bootstrapping non disjoint subsets of data (Ben-Hur et al., 2001; Levine & Domany, 2001). Instead, the stability of cluster solution is captured by a transfer predictor to measure how generalizable the cluster solution on one disjoint subset is to another. We apply this idea to the ML paradigm of data splitting, where a portion of the data is carved out as a training set, another held-out for calibration i.e. $\boldsymbol{X} = \boldsymbol{X}_{tr} \sqcup \boldsymbol{X}_{cal}$. Then the measure of generalizability of the clustering solution is defined as the error (or dissimilarity) between the clustering solution on the calibration set ($\boldsymbol{z}_{cal}$ cluster labels for $\boldsymbol{X}_{cal}$) with its transferred labels via a transfer classifier $\hat{f}_{tr}$ trained on the training data ($\boldsymbol{X}_{tr}$ with cluster labels $\boldsymbol{z}_{tr}$).

$$\text{generalizability error} = \text{Error}\big(\boldsymbol{z}_{cal}, \hat{f}_{tr}(\boldsymbol{X}_{cal}; (\boldsymbol{X}_{tr}, \boldsymbol{z}_{tr})\big) \tag{1}$$

This quantifies the *generalizability* of clustering solutions: a small value for the error (or dissimilarity) between clustering solutions and predictions on the calibration set means the clustering solution learned from the train set is very generalizable to the held-out calibration set. Example of error measures include the mean squared error (used observation-wise in Tibshirani & Walther (2005)'s *prediction strength* index), or the Hamming distance (which corresponds to the misclassification risk of the calibration set) in Lange et al. (2004). However, for clustering which involves multi-class labels, a more popular similarity metric for clustering label comparison is the adjusted-Rand index (ARI). In fact, the aforementioned error measures require a label alignment step, a linear assignment problem that can be solved using the Hungarian matching algorithm (Kuhn, 1955) while ARI is permutation invariant. Alternatively, we also use the multi-class hinge loss as point-wise error measure for Cluster LOCO when using a soft classifier. These scores empirically yield similar normalized feature importance (see Appendix Figure 5 for a comparison of error measures on a simple example).

Leave-One-Covariate-Out (LOCO) on the other hand is an extensively studied quantity borrowed from supervised learning (Lei et al., 2017; Verdinelli & Wasserman, 2023; Gan et al., 2023; Little et al., 2025) which quantifies the change in prediction error when removing a feature. In its LOCO-Split form, the LOCO error for feature j given test point (X, y) is obtained by the difference in prediction error on the test set in the absence of feature j (i.e. $\hat{f}_{tr}^{-j}$ fit on without-j training data $(\boldsymbol{X}_{tr,-j}, \mathbf{y}_{tr})$) and the prediction error on the test set given the full feature set for training (i.e. $\hat{f}_{tr}$ trained on $(\boldsymbol{X}_{tr}, \mathbf{y}_{tr})$) as shown in equation 2.

$$\text{LOCO}_j(X, y) = \text{Error}\big(y, \hat{f}_{tr}^{-j}(X_{-j}; (\boldsymbol{X}_{tr,-j}, \mathbf{y}_{tr})\big) - \text{Error}\big(y, \hat{f}_{tr}(X; (\boldsymbol{X}_{tr}, \mathbf{y}_{tr})\big) \tag{2}$$

Essentially, LOCO quantifies how much the model performance drops when retraining the model after excluding feature j to determine whether it was an important feature: if the performance degrades, then feature j is important whereas if the performance remains unchanged, feature j isn't important.

2.2 Cluster LOCO-Split: a generalizability feature importance score

The LOCO objective lends itself naturally to a clustering extension: one may remove a feature, re-cluster the data, measure the resulting change in cluster solution using internal validity indices or stability scores for example. However, popular validity indices such as the silhouette score (Rousseeuw, 1987; Karanikola et al., 2021) often rely on geometric assumptions (i.e. favoring compact or well-separated clusters). On the other hand, most stability criteria (Ben-Hur et al., 2001; Luxburg, 2009) measure the robustness of the clustering solution under resampling and the LOCO interpretation of the effect of feature occlusion is less

meaningful as scores reflect mostly the model's sensitivity. Therefore, we choose to use generalizability as a meaningful quantity for feature importance via LOCO: providing a model-agnostic and assumption-free feature importance. We give an illustrative example of both silhouette-based LOCO and stability-based LOCO scores' limitations in Section 3 compared to our generalizability-based Cluster LOCO-Split score.

In order to capture the change in generalizability of clustering solutions due to feature contribution, our score (in eq. 3) quantifies the change in generalizability error when removing a feature. Essentially, when an important feature is removed for the cluster algorithm, the clustering solution is expected to be less generalizable and the generalizability error without the feature increases.

Cluster LOCO-Split. For $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ split into $\boldsymbol{X}_{tr}$ and $\boldsymbol{X}_{cal}$, C_θ a clustering algorithm. Cluster the training set and get the cluster labels $\boldsymbol{z}_{tr} = C_\theta(\boldsymbol{X}_{tr})$, cluster the calibration set and get cluster labels $\boldsymbol{z}_{cal} = C_\theta(\boldsymbol{X}_{cal})$:

$$\hat{\Delta}_j(\boldsymbol{X}) = \underbrace{\text{Error}\big(\boldsymbol{z}_{cal}, \hat{f}_{tr}^{-j}(\boldsymbol{X}_{cal,-j}; \boldsymbol{X}_{tr,-j}, z_{tr})\big)}_{\text{generalizability error without } j} - \underbrace{\text{Error}\big(\boldsymbol{z}_{cal}, \hat{f}_{tr}(\boldsymbol{X}_{cal}; (\boldsymbol{X}_{tr}, \boldsymbol{z}_{tr})\big)}_{\text{generalizability error}} \tag{3}$$

We summarize our Cluster LOCO-Split algorithm in Algorithm 1, where the metric of dissimilarity used from the stability literature is usually the negative ARI. We start by splitting the data into training and calibration i.e. $\boldsymbol{X}_{tr} = \{X_i\}_{i\in\mathcal{I}_{tr}}$ and $\boldsymbol{X}_{cal} = \{X_i\}_{i\in\mathcal{I}_{cal}}$ respectively, and for cluster algorithm C_θ (where θ denotes all hyperparameters for the clustering algorithm), we separately obtain the cluster labels on each split $\boldsymbol{z}_{tr} = C_\theta(\boldsymbol{X}_{tr})$, $\boldsymbol{z}_{cal} = C_\theta(\boldsymbol{X}_{cal})$. We set $\{\boldsymbol{X}_{tr}, \boldsymbol{z}_{tr}\}$ to be the training set for the generalizability predictor while $\{\boldsymbol{X}_{cal}, \boldsymbol{z}_{cal}\}$ is used as unseen held-out set. Cluster LOCO-Split requires two generalizability classifiers, a base generalizability classifier $\hat{f}_{tr}$ trained on the full training set $(\boldsymbol{X}_{tr}, \boldsymbol{z}_{tr})$ and its *without feature-j* counterpart $\hat{f}_{tr}^{-j}$ trained on the without feature-j training set $(\boldsymbol{X}_{tr,-j}, \boldsymbol{z}_{tr})$. We derive the Cluster LOCO-Split score as the difference between the generalizability error leaving out feature j evaluated on the calibration set $(\boldsymbol{X}_{cal,-j}, \boldsymbol{z}_{cal})$ and the generalizability error with full features evaluated on the full feature calibration set $(\boldsymbol{X}_{cal}, \boldsymbol{z}_{cal})$.

Algorithm 1 Cluster LOCO-Split

Input: Unlabeled $\boldsymbol{X} \in \mathbb{R}^{N\times M}$

1: Split data into training and calibration sets: $\boldsymbol{X}_{tr} = \{X_i\}_{i\in\mathcal{I}_{tr}}$ and $\boldsymbol{X}_{cal} = \{X_i\}_{i\in\mathcal{I}_{cal}}$.
2: Cluster each split:
   (a) Cluster the training data: $\{X_i\}_{i\in\mathcal{I}_{tr}}$ to obtain cluster labels $\{\boldsymbol{z}_i\}_{i\in\mathcal{I}_{tr}}$.
   (b) Cluster the calibration data: $\{X_i\}_{i\in\mathcal{I}_{cal}}$ to obtain cluster labels $\{\boldsymbol{z}_i\}_{i\in\mathcal{I}_{cal}}$.
3: Fit generalizability classifiers
   (a) Fit $\hat{f}$ to the cluster-labeled training data $\{(X_i, \boldsymbol{z}_i)\}_{i\in\mathcal{I}_{tr}}$.
   (b) Fit $\hat{f}^{-j}$ to cluster-labeled without j training data $\{(X_{i,-j}, \boldsymbol{z}_i)\}_{i\in\mathcal{I}_{tr}}$.
4: Compute & return Cluster LOCO-Split

$$\hat{\Delta}_j^{\text{split}} := \text{Error}\big(\boldsymbol{z}_{cal}, \hat{f}_{tr}^{-j}(\boldsymbol{X}_{cal,-j}|(\boldsymbol{X}_{tr,-j}, \boldsymbol{z}_{tr}))\big) - \text{Error}\big(\boldsymbol{z}_{cal}, \hat{f}_{tr}(\boldsymbol{X}_{cal}|(\boldsymbol{X}_{tr}, \boldsymbol{z}_{tr}))\big)$$

Output: $\{\hat{\Delta}_j^{\text{split}}\}_{j=1}^M$

2.3 Scaling Cluster LOCO: a fast procedure for high-dimensional data

Cluster LOCO-Split inherits several limitations from data splitting and feature occlusion. First, splitting the data reduces the sample size available both for fitting the clustering model and for estimating the generalizability error. This can reduce accuracy and make the score sensitive to the specific train-calibration split. The issue is especially pronounced when clusters are unbalanced: if either split contains few observations from a cluster, the resulting generalizability estimate can become unstable. Second, because Cluster LOCO-Split removes one feature at a time, it can underestimate the importance of correlated features: in the presence of two correlated features, removing one may have little effect because the other remains in

the active feature set; as a result, both features may receive artificially low importance scores. Multi-fold splitting can mitigate some of the instability caused by data splitting, but it increases the computational cost substantially and may be impractical for large datasets as one would have to refit $F \times (M+1)$ models, where F the number of folds. In this section, we introduce Cluster LOCO-MP, a minipatch-based extension designed to address these limitations. By leveraging ensembles of small random subsets of observations and features, Cluster LOCO-MP provides a fast and more stable approximation of Cluster LOCO scores, improving scalability in large datasets.

2.3.1 Cluster LOCO-MP: Cluster LOCO with minipatch ensembles

To scale up our Cluster LOCO feature importance score for large-scale data, we introduce minipatches. Minipatches are tiny subsamples of the data that enable fast and extremely parallel model fitting via ensembling. Instead of traditional subsampling observations into batches (Breiman, 2001; Louppe, 2015), minipatches require simultaneous subsampling of $n << N$ observations and $m << M$ features, and yield a natural structure of the data in in- and out-of-patch features and in- and out-of-patch observations. Minipatches have been shown to have computational advantages in the supervised learning setting (Yao & Allen, 2021; Toghani & Allen, 2021; Gan et al., 2023), benefit from implicit ridge-like regularization useful in correlated settings (Yao et al., 2021) and have been used successfully for clustering in consensus clustering (Gan & Allen, 2022). By drawing on those properties of minipatches, our Cluster LOCO-MP score is fast with large datasets, and no longer sensitive to correlated features.

Algorithm 2 presents the full Cluster LOCO-MP procedure. The algorithm follows the same principle as Cluster LOCO-Split, but replaces a single train-calibration split with an ensemble of minipatches in a leave-one-out framework exploiting the natural in and out-of-patch structure. Given a base cluster algorithm and classification algorithm, we require to fix ahead a large number of minipatches B. Then our algorithm can be summarized in four steps: first, performing minipatch clustering and generalizability training; second and third, obtaining the ensemble LOO and LOCO-LOO predictions; and finally computing the scores. Since we cluster each minipatch and then train a generalizability classifier to predict the resulting cluster labels, they are arbitrary across minipatches and require to be aligned to a common labeling. We outline in our algorithm an alignment step using a reference clustering but when computing such a reference clustering is costly, one can alternatively use the pairwise overlap across minipatches to get approximate alignment. The aligned predictors are then aggregated out-of-training: the LOO ensemble predictor $\hat{H}$ averages predictions over minipatches that exclude the target observation, whereas the LOCO-LOO predictor $\hat{H}^{-j}$ further restricts this average to minipatches that also exclude feature j. The Cluster LOCO-MP score is then defined as the change in prediction error between the ensemble $\hat{H}^{-j}$ and $\hat{H}$.

The computational advantages of our method come primarily from the minipatch construction. Each minipatch contains only a small subset of observations and features, reducing the cost of fitting computationally expensive clustering procedures on large data. Moreover, because minipatches are generated and fitted independently, the training step is naturally parallelizable. For example, spectral clustering has time complexity $\mathcal{O}(N^3)$, which becomes $\mathcal{O}(Bn^3)$ under minipatch clustering, where $n \ll N$ is the number of observations in each minipatch and B is the number of minipatches. The ensemble scores can also be computed in parallel across features, making the implementation modular and adaptable to the available computational resources. Beyond computational efficiency, feature subsampling in the minipatch ensemble helps evaluate the contributions of correlated features. Because different minipatches contain different subsets of features, some include one feature of a correlated group but not another, while others exclude the group altogether. The resulting ensemble prediction error therefore reflects how much predictive information is lost when a feature is unavailable across the ensemble, even when it belongs to a set of correlated features.

2.3.2 Cluster LOCO-RAMPART

In high-dimensional applications, practitioners are often primarily interested in the top-k ranked features driving a clustering solution rather than the full set of important features. This is particularly relevant in applications such as genomics, where many features may be irrelevant or redundant (e.g. house-keeping genes) and interpretation typically focuses on a small set of candidate markers. In such cases, computing Cluster LOCO-MP scores for every feature can be both computationally expensive and statistically inefficient.

Algorithm 2 Cluster LOCO-MP

Input: Unlabeled $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N \in \mathbb{R}^M$ to cluster.

1: Perform minipatch consensus clustering and minipatch training:
   (a) Randomly sample B minipatches of size $n \times m$ indexed $(I_b, F_b)_{b=1}^B$.
   (b) Obtain a reference clustering $\boldsymbol{z}_0 = C_\theta(\boldsymbol{X})$ for alignment.
   (c) Cluster each minipatch b to get the minipatch cluster label $\tilde{\boldsymbol{z}}_{I_b} = C_\theta(\boldsymbol{X}_{I_b,F_b})$. Align $\tilde{\boldsymbol{z}}_{I_b}$ with $\boldsymbol{z}_{0,I_b}$ via Hungarian matching to get $\boldsymbol{z}_{I_b}$.
   (d) Calculate consensus clustering $\boldsymbol{z}^*$ where for $i = 1, \ldots, N$,

$$\boldsymbol{z}_i^* = \arg\max_{k \in [K]} \sum_{b \in [B]} \mathbf{1}(i \in I_b)\mathbf{1}(\boldsymbol{z}_i = k)$$

   (e) Train the generalizability classifiers on each minipatch b: $\hat{f}_b$ on $(\boldsymbol{X}_{I_b,F_b}, \boldsymbol{z}_{I_b})$.

2: Construct Cluster LOO ensemble prediction $\hat{H}$ for observation i:

$$\hat{H}(X_{i,:}) \quad := \quad \sum_{\{b \in [B]: i \notin I_b\}} \hat{f}_b(X_{i,F_b} | (\boldsymbol{X}_{I_b,F_b}, \boldsymbol{z}_{I_b}))$$

3: Construct Cluster LOCO-LOO ensemble prediction $\hat{H}^{-j}$ for features $j = 1, \ldots, M$ and observation i:

$$\hat{H}^{-j}(X_{i,-j}) \quad := \quad \sum_{\{b \in [B]: i \notin I_b, j \notin F_b\}} \hat{f}_b(X_{i,F_b} | (\boldsymbol{X}_{I_b,F_b}, \boldsymbol{z}_{I_b}))$$

4: Compute Cluster LOCO-MP scores for feature $j = 1, \ldots, M$:

$$\hat{\Delta}_j := \text{Error}(\boldsymbol{z}^*, \hat{H}^{-j}(\boldsymbol{X}_{:,-j})) - \text{Error}(\boldsymbol{z}^*, \hat{H}(\boldsymbol{X}))$$

Output: $\{\hat{\Delta}_j\}_{j=1}^M$

We therefore combine Cluster LOCO with the RAMPART framework of Chen et al. (2025), a data-adaptive procedure for top-k feature ranking. In RAMPART Cluster LOCO, Cluster LOCO-MP is used as the round-level feature scoring oracle: at round t, Cluster LOCO-MP produces importance scores $\{\hat{\Delta}_j^{(t)} : j \in S_t\}$ on the active feature set S_t. Features are then ranked from these scores, the bottom half is discarded, and the procedure repeats until only the top-k candidates remain. This yields an adaptive feature importance procedure: preliminary Cluster LOCO-MP estimates are used to screen and prioritize candidate features, and additional computation is concentrated on the most promising features retained through multiple rounds of fixed-batch sequential halving. Chen et al. (2025) showed that while theoretically the order of the number of minipatches required with this procedure is roughly the same as for the minipatch estimates, empirically, in their work and also in our sparse high-dimensional simulations in section 3, this adaptive procedure improves the identification of important features while reducing computational cost substantially. The underlying interpretation remains unchanged: features are important when their removal increases the generalizability error of the clustering solution. The full Cluster LOCO-RAMPART algorithm is provided in the Appendix (Algorithm 3).

2.4 Extensions and practicalities

2.4.1 Cluster-specific LOCO

In many applications, cluster-level feature interpretability is more informative than global feature interpretability: for instance, in genomics, this corresponds to understanding which genes are important for a particular cluster that might identify with a specific cell type. We defined earlier the Cluster LOCO importance metric in the most general sense, compatible with dissimilarity metrics as well as point-wise error scores. In the case of point-wise error scores like multi-class hinge loss, we can extend the family of Cluster LOCO global importance scores to cluster-level importance scores. Cluster LOCO-Split admits a natural cluster-specific interpretation: instead of aggregating over all data points, one can aggregate over each cluster and get a cluster-specific important score (see eq. 4). For feature j and cluster k, let $N_k^{cal} = \sum_{i \in \mathcal{I}_{cal}} \mathbf{1}(\boldsymbol{z}_i = k)$:

$$\hat{\Delta}_{j,k}^{\text{split}} = \sum_{i \in \mathcal{I}_{cal}} \frac{\mathbf{1}(\boldsymbol{z}_i = k)}{N_k^{cal}} \left(\text{Error}\big(\boldsymbol{z}_i, \hat{f}_{tr}^{-j}(X_{i,-j}; (\boldsymbol{X}_{tr,-j}, \boldsymbol{z}_{tr}))\big) - \text{Error}\big(\boldsymbol{z}_i, \hat{f}_{tr}(X_{i,:}; (\boldsymbol{X}_{tr}, \boldsymbol{z}_{tr}))\big) \right) \tag{4}$$

Similarly, we can derive the Cluster-specific LOCO-MP estimates for any point-wise error metric, where for feature j and cluster k,

$$\hat{\Delta}_{j,k}^{\text{MP}} = \sum_{i=1}^{N} \frac{\mathbf{1}(\boldsymbol{z}_i^* = k)}{\sum_{i=1}^{N} \mathbf{1}(\boldsymbol{z}_i^* = k)} \left(\text{Error}\big(\boldsymbol{z}_i^*, \hat{H}^{-j}(X_{i,-j})\big) - \text{Error}\big(\boldsymbol{z}_i^*, \hat{H}(X_{i,:})\big) \right) \tag{5}$$

This property makes our score highly interpretable at both a *local* and *global* scale which is not the case for most scores in the literature: only the SHAP-derived quantities and LRP admit sample-level to global-level interpretation. For clustering however it is especially important to be able to assess the model at the cluster level, especially when those clusters are used for scientific analysis in downstream tasks. We present such a case in section 4.

2.4.2 Choice of hyperparameters

Cluster LOCO is a family of post-hoc interpretability metrics and is not intended to replace careful selection and validation of the clustering model itself. The choice of clustering algorithm, distance or similarity measure, number of clusters, and other model-specific hyperparameters should be made before using Cluster LOCO metrics, drawing on appropriate domain knowledge and clustering validation procedures (see Yin & Hamerly (2009); Luxburg (2009); Handl et al. (2005); Ullmann et al. (2022); Gan et al. (2025). Once a clustering solution has been selected, Cluster LOCO metrics can then be used to interpret which features contribute most to that solution. For Cluster LOCO-Split and Cluster LOCO-MP, the main tuning choices are the train/calibration split ratios and minipatch size ratios respectively. For the former, both splits must contain enough observations to obtain meaningful cluster labels and stable generalizability estimates. We

recommend shuffling the data before splitting and using a balanced split, such as a 50% train and 50% calibration split, as a default. This choice helps avoid pathological splits in which rare or unbalanced clusters are poorly represented in either the training or calibration set. For Cluster LOCO-MP, the main hyperparameters are the minipatch size ratios. The minipatch size controls the tradeoff between computational efficiency and the accuracy of the score estimates: smaller minipatches are faster but may yield noisier clustering solutions, while larger minipatches better approximate the full-data clustering problem at greater computational cost. The minipatch size ratios can be tuned in general as discussed in Gan et al. (2023) and tend to be robust when selecting the appropriate ratios ($r_N = \frac{n}{N}, r_M = \frac{m}{M}$) in the range of 20%-50% for both observations and features (Gan & Allen, 2022).

3 Simulations

In this section, we evaluate our proposed Cluster LOCO-Split and Cluster LOCO-MP/LOCO-RAMPART methods against existing feature importance approaches in simulation settings. We compare our metrics against five baseline methods that provide feature importance for clustering: a prototype-based feature importance (PBFI) and Fuzzy C-Means Shapley (FCM SHAP) from Nápoles et al. (2024), permutation feature importance (PFI) (Pfaffel, 2020), an extension of layer-wise relevance propagation (LRP) for KMeans clustering proposed by Kauffmann et al. (2024); and IMPACC, a minipatch-based adaptive consensus clustering algorithm proposed by Gan & Allen (2022). Since no public code base was available for PBFI or FCM SHAP, we re-implemented both methods in Python. We also extended the KMeans-based PFI implementation, originally proposed in R, to arbitrary `scikit-learn`-style clustering algorithms, and similarly adapted IMPACC, originally in R as well, to be compatible with these algorithms.

3.1 Illustrative simulation for Cluster LOCO-Split

To illustrate the limitations of existing methods and how Cluster LOCO-Split is able to obtain better feature importance scores in even simple examples, we construct a small but challenging synthetic example for clustering. This synthetic clustering example consists of 3 signal features created from interlaced half circles with varying levels of overlap and noise shown in Figure 1a; details on the data-generating process are provided in the Appendix. We augment those 3 signal features with 2 pure noise features sampled from independent uniform distributions. The difficulty of this example stems from the nonlinear separability of the clusters, which challenges methods that favor convex cluster geometries.

Spectral Clustering recovers the true groups as indicated by an adjusted Rand index of 1, whereas methods that impose convex or linearly separable clusters, such as KMeans with an adjusted Rand index of 0.40, produce incorrect assignments as shown in Figure 1b. Consequently, feature importance methods using KMeans, including FCM-SHAP (where we used exact Shapley), LRP, PBFI, and the original permutation-based score PFI, explain a misspecified clustering model rather than the clustering structure of interest. As shown in Figure 1b, these methods assign high importance to noise features in this setting. We also investigated several LOCO-style scores based on different clustering validation criteria: we show two such variants constructed from popular scores, silhouette LOCO is based on the silhouette score change when removing one feature, stability LOCO computes the change in negative ARI under the model-explorer stability objective, using the framework of Ben-Hur et al. (2001). We find that our generalizability-based score, Cluster LOCO-Split, provides the most informative notion of feature importance in this simple yet challenging setting where it uniquely separates signal from noise and correctly orders the signal features by importance.

3.2 Cluster LOCO for large data via minipatch ensembles

We next applied Cluster LOCO-MP for large data: generating three base clustering simulations with $K = 7$ clusters, $N = 3500$ observations (500 observations per cluster), with fixed signal features $p^* = 10$ and increasing noise features $p_{\text{noise}} \in \{10, 40, 190, 490, 990\}$ covering different clustering difficulties. We propose two mixtures models: a Gaussian mixture with *onion* covariance structure (as proposed by Qiu & Joe (2006)) to introduce correlation in the features and a Gamma mixture model that generates data with heavier tails.

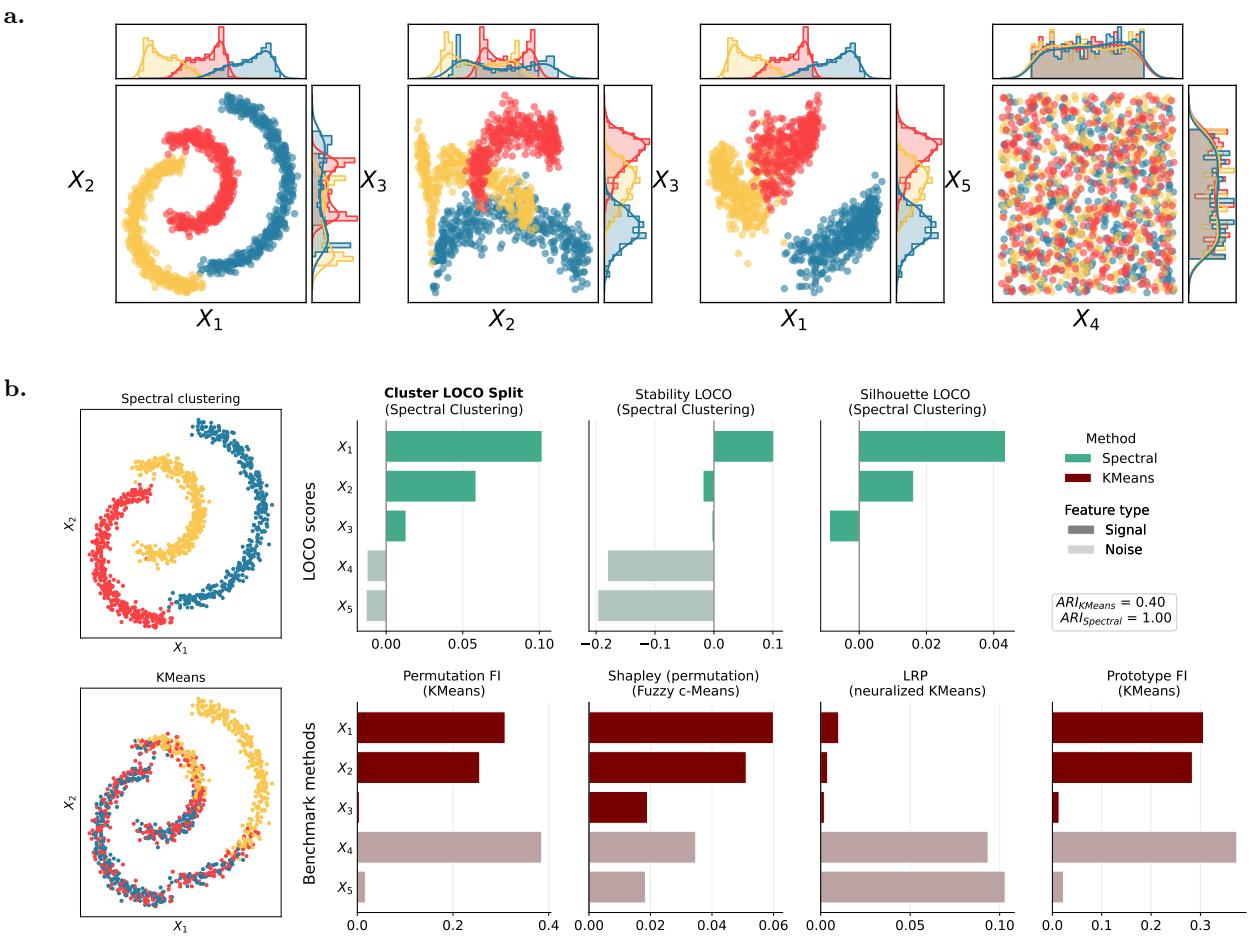

Figure 1: Cluster LOCO is a reliable feature importance score for complex clustering problems. **a. Simulated dataset** presents nonlinearly separable clusters, first three features are signal features, ordered by their importance of contribution. Removing X_1 leads to a harder clustering problem in the remaining signal space with overlapping nonlinearly separable clusters, removing X_2 leaves a moderately hard problem while removing X_3 leads to the easier problem hence the ordered importance of each feature. The last two features are pure noise features sampled from a uniform distribution. **b. Feature importance scores** of possible LOCO-style metrics and existing IML methods for clustering in the literature. For model-agnostic methods we use Spectral Clustering that yields the highest adjusted Rand index (ARI) on this problem while we report the method used for model-specific algorithms. Our Cluster LOCO importance score is the only metric able to recover the correct signal features with the correct ordering.

For a more complex scenario of clustering, we generate clusters inspired by the two-dimensional toy examples of moons and concentric circles (Ester et al., 1996): we sample our data in a two-dimensional embedding with controlled geometry and project up using an orthogonal projection to get any higher-dimensional extension. Each of those three base simulations has two difficulty levels created by varying cluster separation or signal-to-noise ratio into easier or harder clustering problems as shown in the PC-space scatter plot of the generated data in Figure 2 row 1 and 3. We report the top-10 hits (or precision at 10 features) of the true 10 signal features for each method i.e. $\frac{1}{10}|\{\text{top 10 features by method}\} \cap \{\text{true 10-feature set}\}|$. Our Cluster LOCO-MP and Cluster LOCO RAMPART methods were run with fixed $B = 5000$ and $B_{\text{rampart}} = 1000$ and minipatch ratio sizes $\alpha_N = 0.2$, $\alpha_M = 0.2$, while IMPACC was run with the default hyperparameters. Because FCM SHAP is extremely computationally expensive to run for exact Shapley value, we used in this example the SHAP permutation approximation of Shapley values with 20 iterations. For each simulation

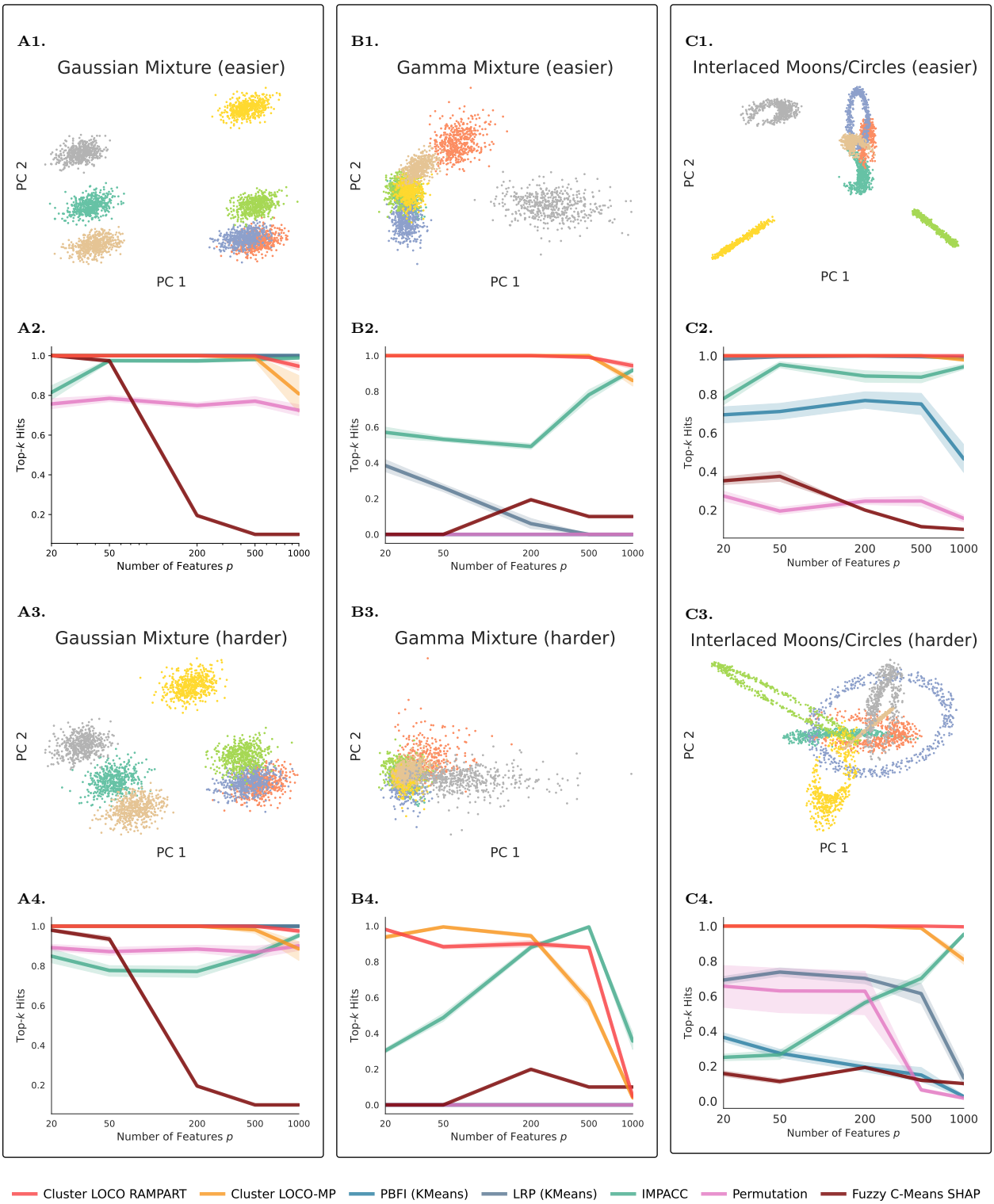

Figure 2: **Simulation results** are obtained across three settings: **A. Gaussian mixture**, **B. Gamma mixture**, and **C. interlaced moons/circles**. Model-agnostic algorithms used are KMeans for Gaussian mixtures, Gamma mixture EM for Gamma mixture, Spectral Clustering for interlaced moons/circles. For each setting, **rows 1 and 3** show the data colored by true labels in PC space, while **rows 2 and 4** report top-10 hits for feature importance with signal features $p^* = 10$ and noise features $p_{\text{noise}} \in \{10, 40, 190, 490, 990\}$. Model-specific feature importance scores' models are reported in the legend. Top-k hits were averaged over 100 replicates. Cluster LOCO-MP and RAMPART outperform existing methods on most tasks.

setting, model-agnostic methods were evaluated with the best performing algorithm among typical clustering algorithm of the `scikit-learn` environment: for Gaussian mixtures, we used KMeans, for the interlaced moons/circles, we used Spectral Clustering. For Gamma mixture, we implemented an EM Gamma mixture model that was best suited for clustering this type of data.

In Figure 2 panels A2 and A4, the performance of the different feature importance scores in the Gaussian mixture model case is reported via the top-10 hits: PBFI and LRP which are KMeans-based recover the correct features as expected, our Cluster LOCO-RAMPART and Cluster LOCO-MP scores also recover the correct top-10 features for $p \leq 500$, with a slight drop in performance at $p = 1000$ which could be solved by budgeting ahead more minipatches. In the easy case, IMPACC performs well, but improves especially when the number of noise features grows as it is designed for sparse features. FCM SHAP and PFI perform the worst overall in the easy and hard case of the Gaussian mixture. For FCM SHAP, the steep drop off of performance is due to the gap between exact Shapley and the approximation with fixed number of iterations while the number of features grow. In the two other simulation examples in Figure 2 columns B and C, our Cluster LOCO family scores outperforms all comparative scores the most definitely: in the Gamma mixture case, IMPACC achieves similar or better performance only when the features are extremely sparse (5% or less of the features are signal features), while in the easy case, both Cluster LOCO RAMPART and Cluster LOCO-MP have robust recovery of the top-10 features. We see similar results with the interlaced moons/circles (Figure 2 panels C2 and C4), Cluster LOCO-MP and Cluster LOCO RAMPART being extremely consistent and effective at recovering the true signal features. We note that in the easy case, LRP is also able to recover effectively the true signal when there is stronger separation in the data and nonlinearity does not affect KMeans as much when it was initialized well. However this performance drops when the clusters have more overlap in their supports (see Figure 2 panels C3 and C4). Overall, Cluster LOCO methods for large data perform as well as or outperform existing methods for those diverse clustering simulations, even typically "hard" clustering tasks, and we provide further simulation results in the Additional figures.

4 An application in single-cell transcriptomics to immune cells

We apply Cluster LOCO-MP and Cluster LOCO-RAMPART for scientific discovery in a real-world clustering application. In single-cell transcriptomics, clustering is routinely used to discover putative cell types or states, after which differentially expressed genes are identified post-hoc and interpreted as marker genes for the discovered groups. This workflow is embedded in popular pipelines such as Scanpy and Seurat (Wolf et al., 2018; Hao et al., 2021), and sometimes considered the gold standard for discovering cell-type identities (Luecken & Theis, 2019). However, this two-step procedure raises important statistical concerns: the same data are first used to define clusters and then reused to test for genes that distinguish those clusters, a form of *double dipping* or *data snooping* known to inflate false discovery rates (Lähnemann et al., 2020). Moreover, this sort of cell-type clustering for gene marker annotation is not a unified framework: the clustering algorithm choice can substantially change the downstream discoveries, and those new marker genes are implicitly dependent on the clustering solution, which can contribute to a lack of reproducibility of results (Gibson, 2022). Cluster LOCO addresses this gap by directly quantifying the contribution of each feature to clustering generalizability, providing a clear unified framework for feature importance in clustering rather than relying on separate downstream testing steps.

4.1 Peripheral Blood Mononuclear Cells dataset for immune cell types discovery

We study the Peripheral Blood Mononuclear Cells (PBMC) dataset published by Zheng et al. (2017) that was annotated using purified transcriptomics populations. Our goal is to cluster and understand the cluster-wise marker genes discovered using traditional pipelines of gene annotations (known to suffer from double dipping) and our proposal of feature importance. We use the processed data available via Scanpy (Wolf et al., 2018) that consists of 765 genes and 700 single cells, obtained after normalizing and scaling the data as reported on the 10X Genomics repository. The reported labels were obtained by classifying the single-cell data with 11 purified sub-populations of PBMC reference profiles and are referred to as *bulk labels* or *purified labels*. It is important here that we chose data that was not generated via clustering but via a correlation

approach for external validation of our results. To further illustrate our results, we compiled known marker annotations of human PBMCs from three sources: the Azimuth atlas (Stuart et al., 2019), as well as Ding et al. (2020) and Oelen et al. (2022) supplementary data on known markers of human PBMC cell types. When pooling known markers, we note that the granularity of cell typing was not always consistent, and this leads to some markers ambiguously marking multiple cell types.

4.2 Globally important genes for clustering

In this data application, global feature importance scores identify the genes most influential for the overall clustering solution. A reliable and interpretable model should reflect the biological reality, therefore we expect highest ranked features to align with some of our known marker genes defining cellular identity in human PBMC. We report in Figure 3 the results for six clustering feature importance metrics: Cluster LOCO-MP, Cluster LOCO-RAMPART with top-100 genes, IMPACC (consensus clustering with minipatches importance), LRP (layer-wise relevant propagation score), PBFI (prototype-based feature importance) and PFI (permutation feature importance). Following the best practices for model selection in clustering (Allen et al., 2023; Wycik et al., 2026), we chose hierarchical clustering with Ward linkage with $K = 10$ clusters to correspond with the 10 known purified labels as base clustering algorithm for our model-agnostic feature importance scores. We show in Figure 3a the clustering solutions in PC-space aligned with the reported labels for both hierarchical clustering and KMeans clustering that PBFI and PFI use. Cluster LOCO-MP was run with $B = 5000$ minipatches and minipatch ratio sizes $\alpha_N = 0.42$ for observations, $\alpha_M = 0.26$ for features. We aligned the obtained labels for each methods with the reported *purified* labels using the Hungarian algorithm.

To evaluate the different clustering feature importance methods, we first analyze the top 10 most globally important features in Figure 3b, where the highest-ranked features correspond to the most important genes driving the clustering structure. We first note that KMeans-based importance methods PBFI and PFI fail to find any known marker genes among the 10 most important genes. Among the remaining four methods, the CD14+ monocyte marker *FTL* is consistently important for Cluster LOCO-MP, Cluster LOCO-RAMPART, LRP and IMPACC, matching the original empirical findings in Zheng et al. (2017). Overall, Cluster LOCO-MP recovers a larger number of reference cell type-specific marker genes than the competing methods. Cluster LOCO-MP also demonstrates the strongest alignment with the biological ground truth, assigning high importance scores primarily to known marker genes from dendritic cells and monocytes, which correspond to the two most well-separated clusters in the PC space. On the other hand, LRP that uses a neuralized KMeans and therefore is more flexible than traditional KMeans recovers similar reference markers as Cluster LOCO-MP. In particular, Cluster LOCO-MP and LRP share important genes like *AIF1* and *LST1* that do not belong in our reference marker set but whose expression patterns have been documented in the monocyte and myeloid cell types in the PBMC literature (Thul & Lindskog, 2018; Leon-Oliva et al., 2023; Ferreira et al., 2024). However, LRP also presents *PSAP* a gene with lower immune-lineage specificity (Thul & Lindskog, 2018), and therefore with a less interpretable profile for cell typing. We note that IMPACC's top-ranked features include a mixture of CD34+ hematopoietic stem cell markers and monocyte markers. However, half of its top 10 most important genes lack documented cell-type specificity (e.g., *IGLL1*). These results underscore that in order to interpret important genes, the clustering solution needs to match a relevant biological truth. In particular, selecting hierarchical clustering as opposed to KMeans yields clustering solution that match a closer biological truth and therefore the solutions are necessarily more interpretable.

4.3 Cluster-wise important genes and marker selection

Global feature importance scores identify the genes most influential for the overall clustering solution, but cluster-level scores are needed to assess which genes characterize individual cell types. We compare here Cluster LOCO-MP at cluster-level with LRP also aggregated at cluster-level – other importance scores investigated earlier are solely global feature importance scores. In parallel, we implement the *clustering + differential gene expression* workflow commonly used in single-cell analysis and described in Scanpy tutorials (Wolf et al., 2018): data is clustered using Louvain clustering, identifying 11 clusters, and subsequent differential gene expression is obtained using t-test across Louvain clusters with Benjamini-Hochberg correction. We define differentially expressed genes (DEGs) using a 5% adjusted p-value cutoff and rank

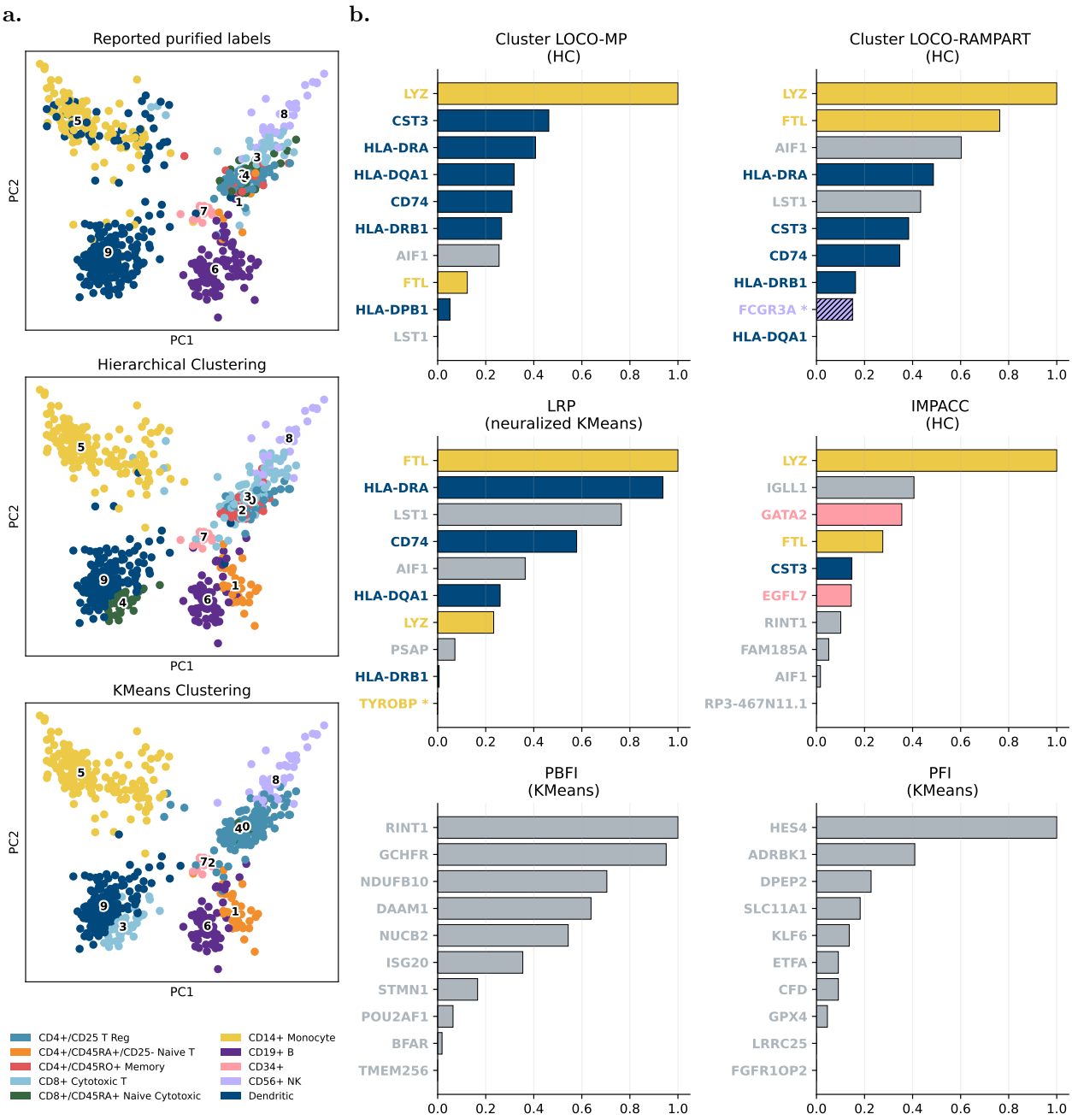

Figure 3: **Top 10 most globally important genes** obtained via different feature importance methods. **a. Normalized gene expression**, reported and clustered via base clustering models - Hierarchical Clustering and standard KMeans clustering, in PC space. **b. Feature importance scores**: we color the genes by known marker genes extracted from the Azimuth Human PBMC database Hao et al. (2021) and the manual annotations from Ding et al. (2020) and Stuart et al. (2019). Genes that appear not to be in any known PBMC cell type marker set are colored in grey. We note that Cluster LOCO-MP finds the most marker genes, KMeans-based methods fail to identify known markers in the important genes.

significant genes by the magnitude of their z-scores. We show in Figure 4a the data in PC-space colored with the respective methods, notably hierarchical clustering for Cluster LOCO-MP, neuralized KMeans for LRP and Louvain clustering for differential gene expression where labels have been aligned with the reported *purified* labels. Cluster LOCO-MP was also run with $B = 5000$ minipatches and minipatch ratio sizes $\alpha_N = 0.42$ for observations, $\alpha_M = 0.26$ for features for the cluster-level analysis.

The cluster-level feature importance are here analyzed for three well separated cell types (CD14+ monocytes, dendritic cells, CD56+ natural killer cells) and one cell type that is harder to cluster (CD34+ hematopoietic stem cells) shown in the PC space in Figure 4a. For each aligned cluster, we compare the top-ranked genes from Cluster LOCO-MP, LRP, and the standard Scanpy workflow (Figure 4b–d), with full top-10 rankings for all cell-type-aligned clusters reported in the Additional figures (Figures 10, 11, 12). Across all clusters, Cluster LOCO-MP recovers a larger number of cluster-specific marker genes than the competing methods. In particular, for the well separated clusters, Cluster LOCO-MP and LRP produce broadly similar rankings, and both are largely consistent with the DEG-based rankings. However, LRP and DEGs rank several genes with weaker cell-type specificity or genes absent from our marker reference set (*CFD, VIM, LAT2, SPI1, S100A10, PTPRCAP*). We also note that Cluster LOCO-MP and LRP scores for monocytes and dendritic reflect the mixing of true dendritic cells with monocytes as shown in the reported labels of Figure 3a, with the presence of shared markers. On the other hand, this cell type is largely over-clustered via Louvain (see Figure 12) where the DEGs of cluster 4, 9 and the unmatched cluster 10 are very similar. In this particular case study, DEGs show possible false positive signal, with larger amount of markers significant for clusters that do not match the cell type they identify (see table 2). For the CD34+ hematopoietic stem cell cluster, which is less cleanly separated than the monocyte, dendritic-cell, and NK clusters, Cluster LOCO-MP is largely more interpretable than LRP and differential gene expression analysis. In this case, Cluster LOCO-MP ranks three known CD34+ HSC markers among its top 10 genes, namely *PRSS57*, *EGFL7*, and *CYTL1*, whereas LRP and differential gene expression rank no known CD34+ HSC markers, in fact *LDHB* appears as a DEG although reported to rather be a marker for Naive T cells or Memory T cells. This pattern is consistent with the geometry of the learned clustering shown in the PC space: well-separated clusters yield more specific and interpretable markers, while less cleanly separated populations, particularly T-cell subtypes, lead to more ambiguous marker rankings across all methods. We note however that this validation against known biological markers is necessarily incomplete: genes absent from a reference marker set may reflect noise or poor specificity, but they may also represent missing markers, or previously under-characterized context-specific genes. Nevertheless, Cluster LOCO-MP most consistently prioritizes cluster-specific markers, suggesting that its local feature importance scores better reflect the features supporting the learned clustering structure.

5 Discussion

Clustering is often subject to debate regarding its rigor: *is clustering an art or science* (Luxburg et al., 2012)? In this work, we contribute to making clustering more scientifically rigorous by addressing a gap in the interpretability of clustering solutions. We devised Cluster LOCO, a novel family of feature importance metrics for clustering that are model-agnostic, flexible and scalable. Through synthetic and real-world evaluations, we have shown our method's flexibility and Cluster LOCO provides feature-level explanation improving the reliability and trustworthiness of clustering solutions, especially for downstream analysis. In particular, in our single-cell transcriptomics application, we addressed a current challenge in single-cell analysis by providing a new unified framework for gene marker annotation that informs practitioners with interpretable, cluster-specific genes corresponding to cell-types when the clustering algorithm identifies a cluster aligned to the biological cell states.

Furthermore, Cluster LOCO's model-agnostic design extends beyond typical clustering algorithms to encompass entire clustering pipelines: our framework can be applied on workflows that incorporate preliminary dimensionality reduction, or truly blackbox models, provided they can be applied to obtain new clustering solutions. Cluster LOCO was therefore implemented as an open-source Python library compatible with `scikit-learn`-style estimators and requires standard `fit` and `predict` functionality from the clustering model. As such, Cluster LOCO is universally applicable to both simple algorithms and sophisticated unsu-

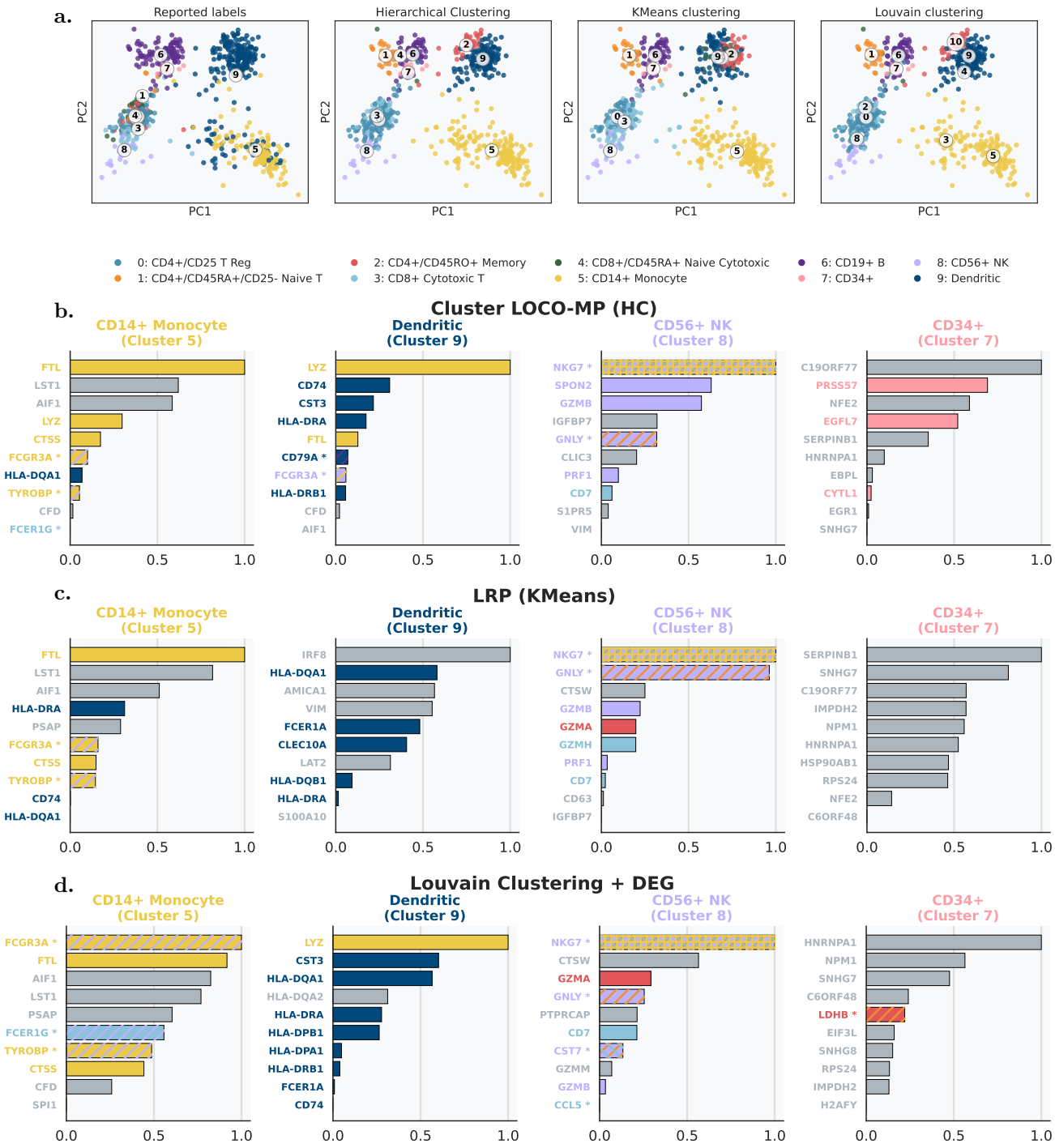

Figure 4: **Top 10 most important genes for clusters** corresponding to well separated cell types: CD14+ Monocytes (Cluster 5), Dendritic cells (Cluster 9), Natural Killer cells (Cluster 8) and CD34+ hematopoietic stem cells (Cluster 7). **a. Normalized gene expression** in PC space colored by cluster labels aligned with the reported labels. KMeans labels are from the neuralized KMeans model. **b. Cluster LOCO-MP** feature importance scores: Hierarchical Clustering was used as base clustering algorithm. **c. Layer-wise Relevant Propagation (LRP)** feature importance scores for neuralized KMeans. **d. Louvain clustering and differential gene expression**: genes ranked in size effect among those tested significant at level 0.05 in differential expression testing. Our method Cluster LOCO-MP captures more consistent cluster-specific markers.

pervised architectures, while retaining a straightforward feature interpretation. However, several limitations remain: first, although the minipatch formulation improves scalability, Cluster LOCO still requires repeated clustering because of feature occlusion, making it more computationally demanding than simpler post-hoc scores. For instance, in settings involving neural network-based clustering models, architecture-specific interpretability methods may be more computationally efficient, although typically less general than our proposed approach. Second, Cluster LOCO focuses on feature-level interpretability: it explains which variables are important for a clustering solution globally or cluster-wise, but it does not directly provide sample-level local explanations for why a particular observation was assigned to a particular cluster.

Future work could extend Cluster LOCO along both methodological and computational directions. Computationally, future work could further improve the efficiency of the minipatch procedure through adaptive sampling designs (e.g. multi-armed bandits, or active learning methods). Methodologically, the notion of generalizability could be adapted to unsupervised tasks beyond clustering, including dimensionality reduction, topic modeling, or representation learning. In each case, LOCO-style feature importance would ask which features are necessary for the learned structure to remain stable and generalizable under refitting, providing a path toward feature-level interpretability for a broader class of unsupervised workflows. Finally, an important direction is to further investigate the use of Cluster LOCO in single-cell transcriptomics. Our immune-cell analysis provides one case study in which cell-type labels were available from external validation experiments, but future work could evaluate the framework across additional datasets, cell-type annotation settings, pre-processing pipelines, and biological contexts. In particular, it would be useful to study how Cluster LOCO compares with marker discovery pipelines based on reference-atlas cell-type classification. More broadly, we believe Cluster LOCO can support a more transparent and statistically grounded use of clustering in scientific discovery.

References

Salem Alelyani, Jiliang Tang, and Huan Liu. Feature Selection for Clustering: A Review. In *Data Clustering*. Chapman and Hall/CRC, 2014. ISBN 978-1-315-37351-5. Num Pages: 32.

Genevera I. Allen, Luqin Gan, and Lili Zheng. Interpretable Machine Learning for Discovery: Statistical Challenges \& Opportunities, August 2023. URL http://arxiv.org/abs/2308.01475. arXiv:2308.01475 [stat.ML].

Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Biocomputing 2002*, Kauai, Hawaii, USA, December 2001. WORLD SCIENTIFIC. doi: 10.1142/9789812799623_0002. URL http://www.worldscientific.com/doi/abs/10.1142/9789812799623_0002.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Yuxi Chen, Tiffany Tang, and Genevera Allen. Top-k Feature Importance Ranking, September 2025. URL http://arxiv.org/abs/2509.15420. arXiv:2509.15420 [cs].

M. Dash, K. Choi, P. Scheuermann, and Huan Liu. Feature selection for clustering - a filter solution. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 115–122, December 2002. doi: 10.1109/ICDM.2002.1183893. URL https://ieeexplore.ieee.org/abstract/document/1183893.

Alan DenAdel, Michelle L. Ramseier, Andrew W. Navia, Alex K. Shalek, Srivatsan Raghavan, Peter S. Winter, Ava P. Amini, and Lorin Crawford. A knockoff calibration method to avoid over-clustering in single-cell RNA-sequencing, March 2024. URL https://www.biorxiv.org/content/10.1101/2024.03.08.584180v1. Pages: 2024.03.08.584180 Section: New Results.

Chris Ding and Xiaofeng He. K -means clustering via principal component analysis. In *Twenty-first international conference on Machine learning - ICML '04*, pp. 29, Banff, Alberta, Canada, 2004. ACM Press. doi: 10.1145/1015330.1015408. URL http://portal.acm.org/citation.cfm?doid=1015330.1015408.

Jiarui Ding, Xian Adiconis, Sean K. Simmons, Monika S. Kowalczyk, Cynthia C. Hession, Nemanja D. Marjanovic, Travis K. Hughes, Marc H. Wadsworth, Tyler Burks, Lan T. Nguyen, John Y. H. Kwon, Boaz Barak, William Ge, Amanda J. Kedaigle, Shaina Carroll, Shuqiang Li, Nir Hacohen, Orit Rozenblatt-Rosen, Alex K. Shalek, Alexandra-Chloé Villani, Aviv Regev, and Joshua Z. Levin. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology*, 38(6):737–746, June 2020. ISSN 1546-1696. doi: 10.1038/s41587-020-0465-8. URL https://www.nature.com/articles/s41587-020-0465-8.

Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *AAAI*, 1996.

Pedro F. Ferreira, Jack Kuipers, and Niko Beerenwinkel. Identifying hierarchical cell states and gene signatures with deep exponential families for single-cell transcriptomics, January 2024. URL https://www.biorxiv.org/content/10.1101/2022.10.15.512383v2. Pages: 2022.10.15.512383 Section: New Results.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of machine learning research : JMLR*, 20:177, 2019. ISSN 1532-4435. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC8323609/.

Luqin Gan and Genevera I. Allen. Fast and interpretable consensus clustering via minipatch learning. *PLOS Computational Biology*, 18(10):e1010577, October 2022. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1010577. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010577.

Luqin Gan, Lili Zheng, and Genevera I. Allen. Model-Agnostic Confidence Intervals for Feature Importance: A Fast and Powerful Approach Using Minipatch Ensembles, January 2023. URL http://arxiv.org/abs/2206.02088. arXiv:2206.02088 [stat].

Luqin Gan, Tarek M. Zikry, and Genevera I. Allen. Are machine learning interpretations reliable? A stability study on global interpretations, May 2025. URL http://arxiv.org/abs/2505.15728. arXiv:2505.15728 [stat].

Greg Gibson. Perspectives on rigor and reproducibility in single cell genomics. *PLoS Genetics*, 18(5): e1010210, May 2022. ISSN 1553-7390. doi: 10.1371/journal.pgen.1010210. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC9122178/.

Mark S. Handcock, Adrian E. Raftery, and Jeremy M. Tantrum. Model-Based Clustering for Social Networks. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 170(2):301–354, March 2007. ISSN 0964-1998, 1467-985X. doi: 10.1111/j.1467-985X.2007.00471.x. URL https://academic.oup.com/jrsssa/article/170/2/301/7085294.

Julia Handl, Joshua Knowles, and Douglas B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, August 2005. ISSN 1367-4803. doi: 10.1093/bioinformatics/bti517. URL https://doi.org/10.1093/bioinformatics/bti517.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, June 2021. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2021.04.048. URL https://www.cell.com/cell/abstract/S0092-8674(21)00583-3.

Lianyu Hu, Mudi Jiang, Junjie Dong, Xinying Liu, and Zengyou He. Interpretable Clustering: A Survey, September 2024. URL http://arxiv.org/abs/2409.00743. arXiv:2409.00743 [cs].

Jiashun Jin and Wanjie Wang. Influential features PCA for high dimensional clustering. *The Annals of Statistics*, 44(6):2323–2359, December 2016. ISSN 0090-5364, 2168-8966. doi: 10.1214/15-AOS1423. URL https://projecteuclid.org/journals/annals-of-statistics/volume-44/issue-6/Influential-features-PCA-for-high-dimensional-clustering/10.1214/15-AOS1423.full.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065): 20150202, April 2016. ISSN 1364-503X. doi: 10.1098/rsta.2015.0202. URL https://doi.org/10.1098/rsta.2015.0202.

A. Karanikola, C. M. Liapis, and S. Kotsiantis. Investigating cluster validation metrics for optimal number of clusters determination. *Intelligent Decision Technologies*, 15(4):809–824, 2021.

Jacob Kauffmann, Malte Esders, Lukas Ruff, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. From Clustering to Cluster Explanations via Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):1926–1940, February 2024. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2022.3185901. URL http://arxiv.org/abs/1906.07633. arXiv:1906.07633 [cs].

Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 20(5):273–282, May 2019. ISSN 1471-0056, 1471-0064. doi: 10.1038/s41576-018-0088-9. URL https://www.nature.com/articles/s41576-018-0088-9.

Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1): 1:1–1:58, March 2009. ISSN 1556-4681. doi: 10.1145/1497577.1497578. URL https://dl.acm.org/doi/10.1145/1497577.1497578.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. ISSN 1931-9193. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109.

Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, June 2004. ISSN 0899-7667. doi: 10.1162/089976604773717621.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference For Regression, March 2017. URL http://arxiv.org/abs/1604.04173. arXiv:1604.04173 [stat].

Diego De Leon-Oliva, Cielo Garcia-Montero, Oscar Fraile-Martinez, Diego Liviu Boaru, Luis García-Puente, Antonio Rios-Parra, Maria J. Garrido-Gil, Carlos Casanova-Martín, Natalio García-Honduvilla, Julia Bujan, Luis G. Guijarro, Melchor Alvarez-Mon, and Miguel A. Ortega. AIF1: Function and Connection with Inflammatory Diseases. *Biology*, 12(5):694, May 2023. doi: 10.3390/biology12050694. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10215110/.

E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11):2573–2593, 2001.

Shenghao Li, Hui Guo, Simai Zhang, Yizhou Li, and Menglong Li. Attention-based deep clustering method for scRNA-seq cell type identification. *PLOS Computational Biology*, 19(11):e1011641, November 2023. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011641. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011641.

Camille Little, Lili Zheng, and Genevera Allen. iLOCO: Distribution-Free Inference for Feature Interactions, May 2025. URL http://arxiv.org/abs/2502.06661. arXiv:2502.06661 [stat].

Gilles Louppe. Understanding Random Forests: From Theory to Practice, June 2015. URL http://arxiv.org/abs/1407.7502. arXiv:1407.7502 [stat].

Malte D. Luecken and Fabian J. Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):MSB188746, June 2019. ISSN 1744-4292. doi: 10.15252/msb.20188746. URL https://doi.org/10.15252/msb.20188746.

Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions, November 2017. URL http://arxiv.org/abs/1705.07874. arXiv:1705.07874 [cs].

Ulrike von Luxburg. Clustering Stability: An Overview. *Foundations and Trends® in Machine Learning*, 2 (3):235–274, 2009. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000008. URL http://arxiv.org/abs/1007.1075. arXiv:1007.1075 [stat].

Ulrike von Luxburg, Robert C. Williamson, and Isabelle Guyon. Clustering: Science or Art? In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 65–79. JMLR Workshop and Conference Proceedings, June 2012. URL https://proceedings.mlr.press/v27/luxburg12a.html.

David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J. McCarthy, Stephanie C. Hicks, Mark D. Robinson, Catalina A. Vallejos, Kieran R. Campbell, Niko Beerenwinkel, Ahmed Mahfouz, Luca Pinello, Pavel Skums, Alexandros Stamatakis, Camille Stephan-Otto Attolini, Samuel Aparicio, Jasmijn Baaijens, Marleen Balvert, Buys de Barbanson, Antonio Cappuccio, Giacomo Corleone, Bas E. Dutilh, Maria Florescu, Victor Guryev, Rens Holmer, Katharina Jahn, Thamar Jessurun Lobo, Emma M. Keizer, Indu Khatri, Szymon M. Kielbasa, Jan O. Korbel, Alexey M. Kozlov, Tzu-Hao Kuo, Boudewijn P.F. Lelieveldt, Ion I. Mandoiu, John C. Marioni, Tobias Marschall, Felix Mölder, Amir Niknejad, Alicja Rączkowska, Marcel Reinders, Jeroen de Ridder, Antoine-Emmanuel Saliba, Antonios Somarakis, Oliver Stegle, Fabian J. Theis, Huan Yang, Alex Zelikovsky, Alice C. McHardy, Benjamin J. Raphael, Sohrab P. Shah, and Alexander Schönhuth. Eleven grand challenges in single-cell data science. *Genome*

Biology, 21(1):31, February 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-1926-6. URL https://doi.org/10.1186/s13059-020-1926-6.

Masayoshi Mase, Art B. Owen, and Benjamin B. Seiler. Cohort Shapley value for algorithmic fairness. Technical Report arXiv:2105.07168, arXiv, May 2021. URL http://arxiv.org/abs/2105.07168. arXiv:2105.07168 [cs, econ, stat] type: article.

J. Materne. The structure of nearby clusters of galaxies. Hierarchical clustering and an application to the Leo region. *Astronomy and Astrophysics*, 63:401–409, February 1978. ISSN 0004-6361. URL https://ui.adsabs.harvard.edu/abs/1978A&A....63..401M. ADS Bibcode: 1978A&A....63..401M.

Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture. *IEEE Access*, 6:39501–39514, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2855437. URL https://ieeexplore.ieee.org/abstract/document/8412085.

Christoph Molnar. *Interpretable Machine Learning (Third Edition)*. Leanpub, February 2018. URL https://leanpub.next/interpretable-machine-learning.

Gonzalo Nápoles, Niels Griffioen, Samaneh Khoshrou, and Çiçek Güven. Feature Importance for Clustering. In Verónica Vasconcelos, Inês Domingues, and Simão Paredes (eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 31–45, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-49018-7. doi: 10.1007/978-3-031-49018-7_3.

Roy Oelen, Dylan H. de Vries, Harm Brugge, M. Grace Gordon, Martijn Vochteloo, Chun J. Ye, Harm-Jan Westra, Lude Franke, and Monique G. P. van der Wijst. Single-cell RNA-sequencing of peripheral blood mononuclear cells reveals widespread, context-specific gene expression regulation upon pathogenic exposure. *Nature Communications*, 13(1):3267, June 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-30893-5. URL https://www.nature.com/articles/s41467-022-30893-5.

Oliver Pfaffel. FeatureImpCluster: Feature Importance for Partitional Clustering, May 2020. URL https://CRAN.R-project.org/package=FeatureImpCluster. Institution: Comprehensive R Archive Network Pages: 0.1.5.

Weiliang Qiu and Harry Joe. Generation of Random Clusters with Specified Degree of Separation. *Journal of Classification*, 23(2):315–334, September 2006. ISSN 1432-1343. doi: 10.1007/s00357-006-0018-y. URL https://doi.org/10.1007/s00357-006-0018-y.

Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Comment: Statistical Inference from a Predictive Perspective. *Statistical Science*, 34(4):599–603, 2019. doi: 10.1214/19-STS748. URL https://www.researchgate.net/publication/338469585_Comment_Statistical_Inference_from_a_Predictive_Perspective.

Volker Roth and Tilman Lange. Feature Selection in Clustering Problems. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/hash/bb03e43ffe34eeb242a2ee4a4f125e56-Abstract.html.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Lloyd S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, volume 2, pp. 307–317. Princeton University Press, 1953.

Dongyuan Song, Kexin Li, Xinzhou Ge, and Jingyi Jessica Li. ClusterDE: a post-clustering differential expression (DE) method robust to false-positive inflation caused by double dipping. *Research Square*, pp. rs.3.rs, August 2023. doi: 10.21203/rs.3.rs-3211191/v1. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10418557/.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 177(7):1888–1902.e21, June 2019. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2019.05.031. URL https://www.cell.com/cell/abstract/S0092-8674(19)30559-8.

Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pp. 9269–9278. JMLR.org, July 2020.

Peter J. Thul and Cecilia Lindskog. The human protein atlas: A spatial map of the human proteome. *Protein Science : A Publication of the Protein Society*, 27(1):233–244, January 2018. ISSN 0961-8368. doi: 10.1002/pro.3307. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC5734309/.

Robert Tibshirani and Guenther Walther. Cluster Validation by Prediction Strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, September 2005. ISSN 1061-8600, 1537-2715. doi: 10.1198/106186005X59243. URL https://www.tandfonline.com/doi/full/10.1198/106186005X59243.

Mohammad Taha Toghani and Genevera I. Allen. MP-Boost: Minipatch Boosting via Adaptive Feature and Observation Sampling. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 75–78, January 2021. doi: 10.1109/BigComp51126.2021.00023. URL http://arxiv.org/abs/2011.07218. arXiv:2011.07218 [stat].

Theresa Ullmann, Christian Hennig, and Anne-Laure Boulesteix. Validation of cluster analysis results on validation data: A systematic framework. *WIREs Data Mining and Knowledge Discovery*, 12(3):e1444, 2022. ISSN 1942-4795. doi: 10.1002/widm.1444. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1444. _eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1444.

Isabella Verdinelli and Larry Wasserman. Feature Importance: A Closer Look at Shapley Values and LOCO, March 2023. URL http://arxiv.org/abs/2303.05981. arXiv:2303.05981 [stat].

Alexandra-Chloé Villani, Rahul Satija, Gary Reynolds, Siranush Sarkizova, Karthik Shekhar, James Fletcher, Morgane Griesbeck, Andrew Butler, Shiwei Zheng, Suzan Lazo, Laura Jardine, David Dixon, Emily Stephenson, Emil Nilsson, Ida Grundberg, David McDonald, Andrew Filby, Weibo Li, Philip L. De Jager, Orit Rozenblatt-Rosen, Andrew A. Lane, Muzlifah Haniffa, Aviv Regev, and Nir Hacohen. Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science (New York, N.Y.)*, 356(6335):eaah4573, April 2017. ISSN 1095-9203. doi: 10.1126/science.aah4573.

Binhuan Wang, Yilong Zhang, Will Wei Sun, and Yixin Fang. Sparse Convex Clustering. *Journal of Computational and Graphical Statistics*, 27(2):393–403, April 2018. ISSN 1061-8600. doi: 10.1080/10618600.2017.1377081. URL https://researchwith.njit.edu/en/publications/sparse-convex-clustering/.

Daniela M. Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, June 2010. ISSN 0162-1459. doi: 10.1198/jasa.2010.tm09415. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2930825/.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, February 2018. ISSN 1474-760X. doi: 10.1186/s13059-017-1382-0. URL https://doi.org/10.1186/s13059-017-1382-0.

Kai R. Wycik, Tiffany M. Tang, Tarek M. Zikry, and Genevera I. Allen. Cluster Analysis with Resampling for Validation and Exploration (CARVE), May 2026. URL http://arxiv.org/abs/2606.00327. arXiv:2606.00327 [stat.ME].

Eric P. Xing and Richard M. Karp. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17(suppl_1):S306–S315, June 2001. ISSN 1367-4811, 1367-4803. doi: 10.1093/bioinformatics/17.suppl_1.S306. URL https://academic.oup.com/bioinformatics/article/17/suppl_1/S306/262530.

Rui Xu and Donald C. Wunsch. Clustering Algorithms in Biomedical Research: A Review. *IEEE Reviews in Biomedical Engineering*, 3:120–154, 2010. ISSN 1937-3333, 1941-1189. doi: 10.1109/RBME.2010.2083647. URL http://ieeexplore.ieee.org/document/5594620/.

Tianyi Yao and Genevera I. Allen. Feature Selection for Huge Data via Minipatch Learning. Technical Report arXiv:2010.08529, arXiv, February 2021. URL http://arxiv.org/abs/2010.08529. arXiv:2010.08529 [cs, stat] type: article.

Tianyi Yao, Daniel LeJeune, Hamid Javadi, Richard G. Baraniuk, and Genevera I. Allen. Minipatch Learning as Implicit Ridge-Like Regularization. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 65–68, January 2021. doi: 10.1109/BigComp51126.2021.00021. URL https://ieeexplore.ieee.org/abstract/document/9373110. ISSN: 2375-9356.

Bing Yin and Greg Hamerly. Hierarchical Stability-Based Model Selection for Clustering Algorithms. In *2009 International Conference on Machine Learning and Applications*, pp. 217–222, December 2009. doi: 10.1109/ICMLA.2009.64. URL https://ieeexplore.ieee.org/document/5381839.

Bin Yu and Karl Kumbier. Veridical data science. *Proceedings of the National Academy of Sciences*, 117(8): 3920–3929, February 2020. doi: 10.1073/pnas.1901326117. URL https://www.pnas.org/doi/10.1073/pnas.1901326117.

Jesse M. Zhang, Govinda M. Kamath, and David N. Tse. Valid Post-clustering Differential Analysis for Single-Cell RNA-Seq. *Cell Systems*, 9(4):383–392.e6, October 2019. ISSN 2405-4712. doi: 10.1016/j.cels.2019.07.012. URL https://www.sciencedirect.com/science/article/pii/S2405471219302698.

Grace X. Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1): 14049, January 2017. ISSN 2041-1723. doi: 10.1038/ncomms14049. URL https://www.nature.com/articles/ncomms14049.

A Appendix

A.1 Simulation details

A.1.1 Interlaced moons toy example

The example is constructed from 2 interlaced moons (circular arcs) in both features 1, 2 and 3. We parametrize for the 3 clusters respectively:

- Cluster 1: $X_1 = 0.35 + 1.35\sin(T_1) + \epsilon_x$ and $X_2 = 0.5 + 1.35\cos(T_0) + \epsilon_y$ where $T_1 \sim U(0.05\pi, 1.05\pi)$ where $\epsilon_x, \epsilon_y \sim \mathcal{N}(0, 0.08^2)$. Then let $R = \sqrt{(X_2 - 0.1)^2 + (X_1 + 0.05)^2}$, $\tilde{X}_3 = -0.8\sin(1.5R) + 0.2\log(|1 + X_2|) + 0.9 + \epsilon_z$, $\epsilon_z \sim N(0, 0.08^2)$ as well.
- Cluster 2: $X_1 = -0.05 + \sin(T_2) + \epsilon_x$ and $X_2 = -0.05 + \cos(T_1) + \epsilon_y$ where $T_2 \sim U(0.95\pi, 1.95\pi)$ where $\epsilon_x, \epsilon_y \sim \mathcal{N}(0, 0.08^2)$. Then let $R = \sqrt{(X_2 - 0.1)^2 + (X_1 + 0.05)^2}$, $\tilde{X}_3 = 0.8\sin(1.5R) + 0.2\log(|1 + X_2|) - 0.7 + \epsilon_z$, $\epsilon_z \sim N(0, 0.08^2)$ as well.
- Cluster 3: $X_1 = -0.05 + 0.7\sin(T_3) + \epsilon_x$ and $X_2 = 0.55 + 0.7\cos(T_3) + \epsilon_y$ with $T_3 \sim U(-0.15\pi, 1.15\pi)$. Then let $R = \sqrt{(X_2 - 0.1)^2 + (X_1 + 0.05)^2}$, $\tilde{X}_3 = 0.8\sin(1.5R) + 0.5\log(|1 + X_2|) + 0.66 + \epsilon_z$, $\epsilon_z \sim N(0, 0.08^2)$ as well.

By construction, $\tilde{X}_3$ is a nonlinear transformation of X_1 and X_2 and is particularly correlated with X_2. We decorrelate $\tilde{X}_3$ from X_2 to get the final X_3 feature: $X_3 = \tilde{X}_3 - \frac{Cov(X_2, \tilde{X}_3}{Var(X_2)} X_2$. The data is then augmented with 2 features $X_4, X_5 \sim \mathcal{U}([-1, 2])$ independently.

A.1.2 Comparative simulations

For each simulation setting, we generated labeled data $(X_i, Y_i)_{i=1}^N$ where $Y_i \in \{1, \ldots, K\}$ denotes the cluster label and each cluster k contains n_k observations. The data generating process first produces a low-dimensional signal representation in latent dimension $\tilde{X} \in \mathbb{R}^{d_0}$ and is embedded into higher-dimensional signal feature space $\mathbb{R}^d$. We then append pure noise features, so $X_i \in \mathbb{R}^M$ where $M = d + d_{\text{noise}}$.

Gaussian Mixture. The standard Gaussian mixture model outputs generates the signal latent $\tilde{X}_i \in \mathbb{R}^{d_0}$, $Y_i \in \{1, \ldots, K\}$. Let $\alpha \in \mathbb{R}^{>0}$ be a parameter controlling cluster separation, $\mu_k \in \mathbb{R}^{d_0}, \Sigma_k \in \mathbb{R}^{d_0 \times d_0}$, then observations for cluster k are sampled from $\tilde{X}_i | Y_i = k \sim N(\alpha\mu_k, \Sigma_k)$ where Σ_k is obtained via an *onion covariance structure* Qiu & Joe (2006) that builds a correlation matrix layer-by-layer, hence its name, and μ_k's coordinates are sampled uniformly at random between $[-1, 1]$.

Gamma Mixture. The Gamma mixture introduces non-Gaussian marginal distributions while allowing dependence among features through a Gaussian copula. For each cluster k, let R_k denote a cluster-specific correlation matrix and let F_{kj} be the CDF of a gamma distribution with shape parameter a_j and cluster-specific scale parameter s_{kj}, $\Gamma(a_j, s_{kj})$. We define the joint distribution of $\tilde{X}_i \mid Y_i = k$ by the Gaussian copula

$$C_{R_k}(u_1, \ldots, u_{d_0}) = \Phi_{R_k}\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_{d_0})\right),$$

where Φ_{R_k} is the CDF of a $\mathcal{N}(0, R_k)$ with $R_k \in \mathbb{R}^{d_0, d_0}$, and Φ is the CDF of $\mathcal{N}(0, I)$. The samples are generated by drawing $z_i \sim \mathcal{N}(0, R_k)$, setting $U_{ij} = \Phi(z_{ij})$, and then applying the inverse gamma CDF $\tilde{X}_{ij} = F_{kj}^{-1}(U_{ij})$. This construction gives each coordinate a gamma marginal distribution while using R_k to control the dependence structure within cluster k.

Interlaced Moons/Circles. We generate half moons and circles in a two dimensional lower dimension first using mixtures of circles and half-moons. For each cluster k, a radius r_k and center $c_k \in \mathbb{R}^2$ are sampled, and the cluster shape is chosen to be either a full circle or a half-circle from a given probability vector. Points are sampled along the corresponding circular arc with additive Gaussian noise: $X_i = c_k + r_k(\cos\theta_i, \sin\theta_i) + \varepsilon_i$, where θ_i is sampled either on $[0, 2\pi]$ for circles or on an interval of length π for half-moons, with a random

orientation. Cluster centers are placed so that the pairwise overlap between the underlying circles is bounded by a fixed maximum overlap percentage. This produces nonlinear, partially interlaced cluster structures.

Embedding data in higher dimensions and noise features For high-dimensional simulations, we embed the low-dimensional signal into a larger ambient space using random feature maps: we generate random orthonormal projection maps, bringing the $\tilde{\boldsymbol{X}} \in \mathbb{R}^{N \times d_0}$ latent observations into $\hat{P}\tilde{\boldsymbol{X}} = \boldsymbol{X} \in \mathbb{R}^{N \times d}$. We also construct cluster-specific embeddings, allowing different clusters to be transformed through different random maps. Finally, we add pure noise signal $\mathbf{S} \in \mathbb{R}^{N \times d_{\text{noise}}}$ sampled from distributions such as Gaussian, Gamma with $a = 1, s = 1$, Student-t, uniform, triangular, or Laplace distributions. The final data generated by our simulator becomes $\boldsymbol{X} = [\hat{\boldsymbol{X}}, \mathbf{S}] \in \mathbb{R}^{N \times M}$.

A.2 Cluster LOCO-RAMPART algorithm

Algorithm 3 Cluster LOCO-RAMPART

Input: Unlabeled data $X \in \mathbb{R}^{N \times M}$, top-$k$ target k, minipatch sizes n, m, minipatches per round B.

1: Set $t = 0$. Initialize active feature set $S_0 = [M]$.
2: **while** $|S_t| > k$ **do**
3: Obtain round-t Cluster LOCO-MP estimates:

$$\{\hat{\Delta}_j^{(t)}\}_{j \in S_t} \leftarrow \text{Cluster LOCO-MP}(\boldsymbol{X}_{\cdot, S_t}, n, m, B)$$

4: Determine Cluster LOCO-MP estimates' ranks (in ascending order): $(\hat{r}_1^t, \ldots, r_{|S_t|}^t)$ at feature $(\hat{\tau}_1^t, \ldots, \hat{\tau}_{|S_t|}^t)$ respectively
5: Retain the top half of the candidate set $S_{t+1} \leftarrow \{\hat{\tau}_1^t, \cdots, \hat{\tau}_{|S_t|/2}^t\}$.
6: Set $t \leftarrow t + 1$.
7: **end while**

Output: $\{\hat{\Delta}_j^{(T)}\}_{j \in S_T}$, $|S_T| \approx k$.

A.3 Additional figures

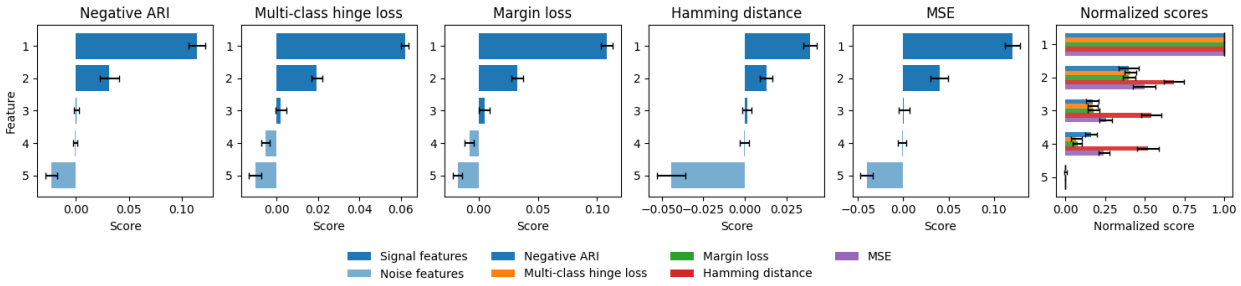

Figure 5: Comparison of Cluster LOCO-Split scores with different measures of error/dissimilarity averaged over 20 runs with standard errors for the toy example described in Section 3.1. Mean squared error and Hamming distance show the more difference in normalized scores compared to other feature importance scores but we see that this is an effect of the scaling, feature 5 has a "worse" effect for these two errors relative to the rest of the scores, normalizing inflates the contribution of feature 2, 3 and 4 since all scores are normalized to be positive with highest importance at 1.

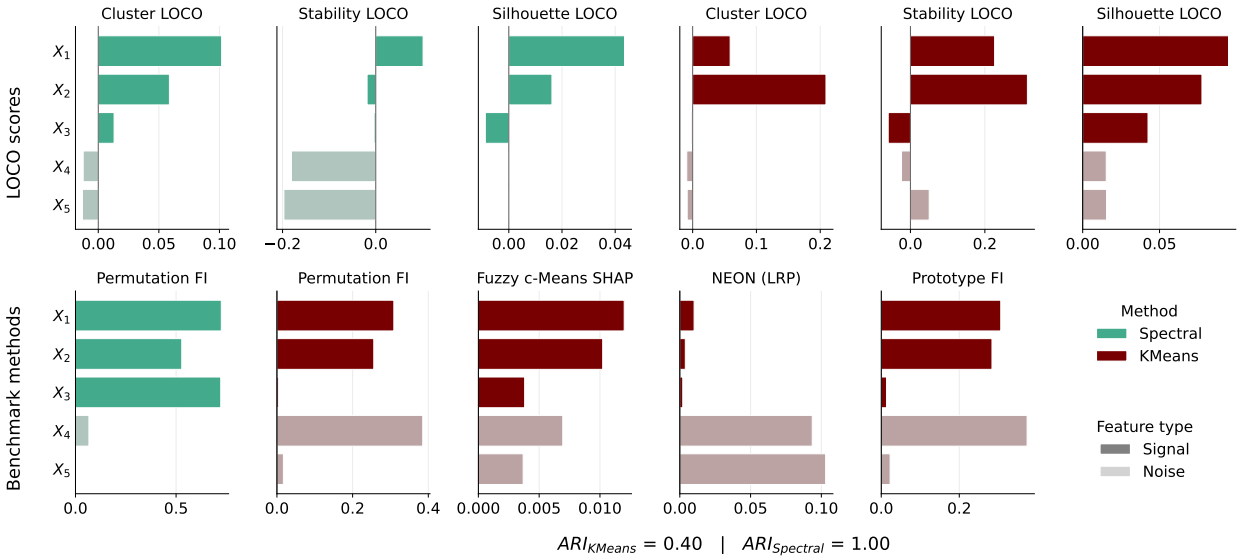

Figure 6: Extended comparison of Cluster LOCO-Split with existing feature importance scores: we compared other LOCO-style scores with stability metric and silhouette score, for both KMeans and Spectral Clustering.

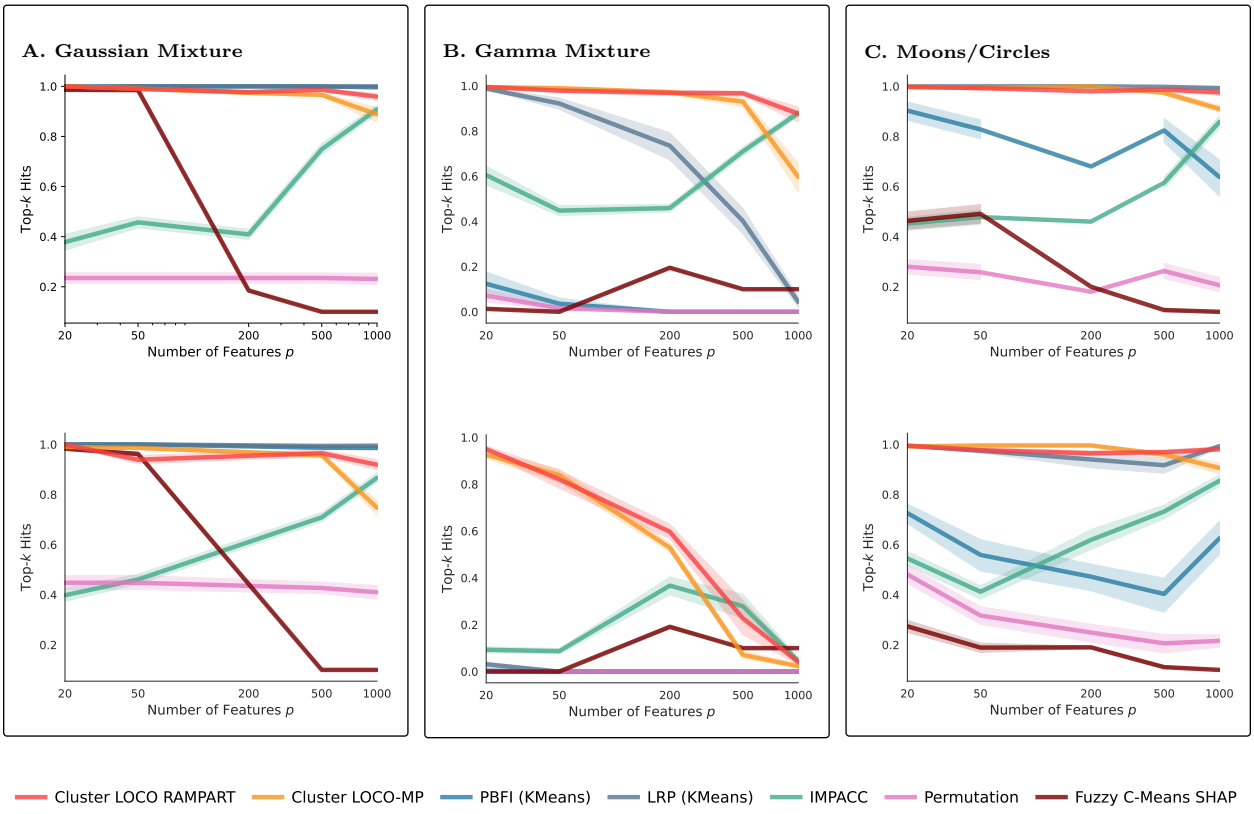

Figure 7: **Top-k hits** in three simulation settings for signal features $p^* = 10$ and noise features features $p_{\text{noise}} \in \{10, 40, 190, 490, 990\}$ with $K = 3$ clusters with $N = 500$ observations per cluster. Cluster LOCO-MP was run with $\alpha_M = \alpha_N = 0.2$ and $B = 5000$, Cluster LOCO-RAMPART with $B_{\text{rampart}} = 1000$. Model agnostic methods are obtained with KMeans for Gaussian mixtures, Gamma mixture EM for Gamma mixture, Spectral Clustering for Moons and Circles. Model-specific methods are reported with their base model in the legend. Top-k hits were reported averaged over 100 replicates.

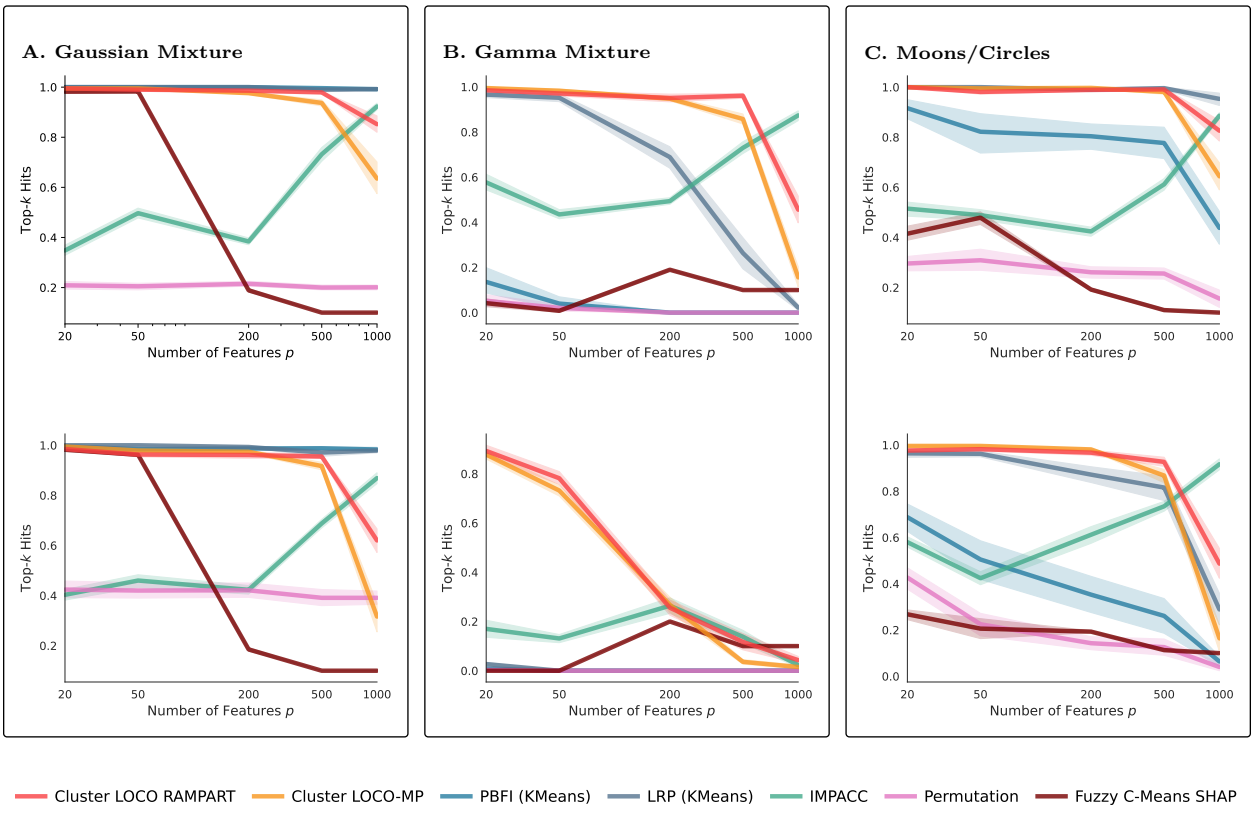

Figure 8: **Top-k hits** in three simulation settings for signal features $p^* = 10$ and noise features features $p_{\text{noise}} \in \{10, 40, 190, 490, 990\}$ with $K = 3$ clusters with $N = 300$ observations per cluster. Cluster LOCO-MP was run with $B = 5000$, Cluster LOCO-RAMPART with $B_{\text{rampart}} = 1000$ and adaptive $\alpha_N \in (0.1, 0.5)$, $\alpha_p \in (0.1, 0.5)$. Model agnostic methods are obtained with KMeans for Gaussian mixtures, Gamma mixture EM for Gamma mixture, Spectral Clustering for Moons and Circles. Model-specific methods are reported with their base model in the legend. Top-k hits were reported averaged over 100 replicates.

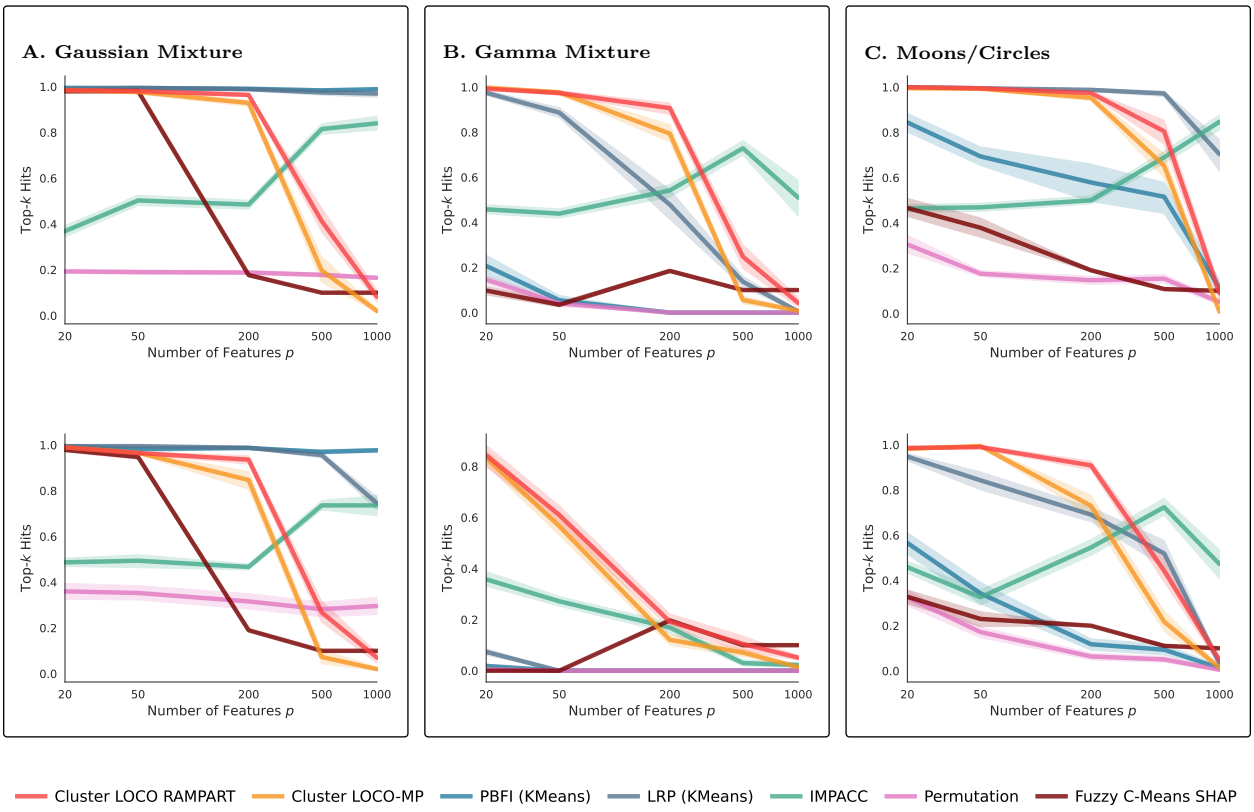

Figure 9: **Top-k hits** in three simulation settings for signal features $p^* = 10$ and noise features features $p_{\text{noise}} \in \{10, 40, 190, 490, 990\}$ with $K = 3$ clusters with $N = 100$ observations per cluster. Cluster LOCO-MP was run with $\alpha_M = \alpha_N = 0.2$ and $B = 5000$, Cluster LOCO-RAMPART with $B_{\text{rampart}} = 1000$. Model agnostic methods are obtained with KMeans for Gaussian mixtures, Gamma mixture EM for Gamma mixture, Spectral Clustering for Moons and Circles. Model-specific methods are reported with their base model in the legend. Top-k hits were reported averaged over 100 replicates.

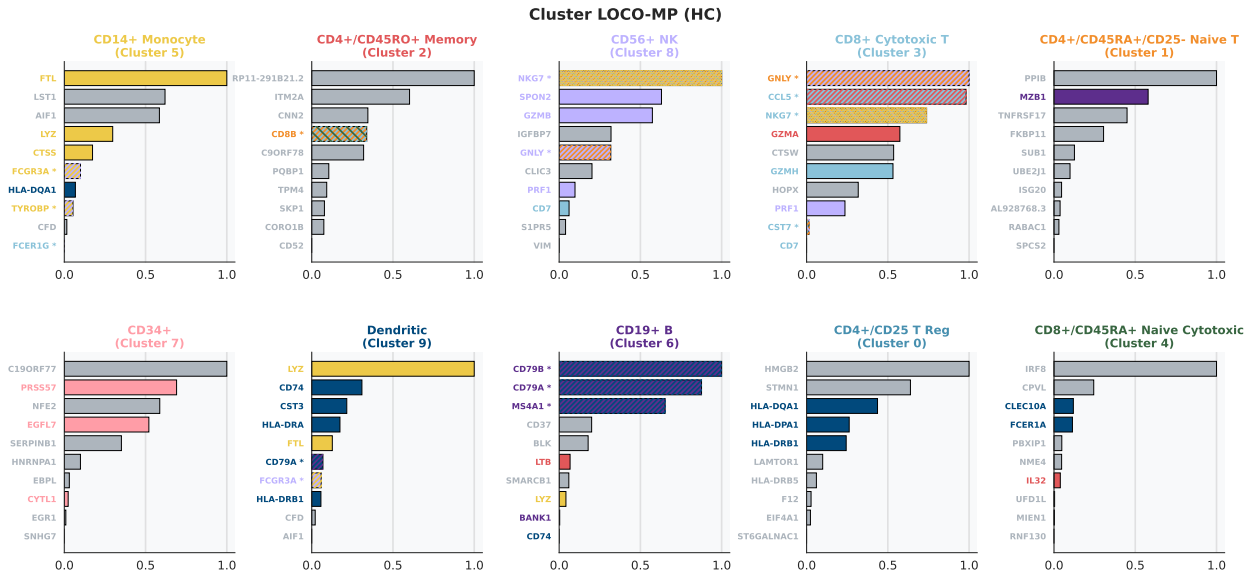

Figure 10: **Top 10 most important genes for clusters via Cluster LOCO-MP** feature importance scores: Hierarchical Clustering was used as base clustering algorithm. Labels were aligned with the reported *purified* labels. Known markers from our reference set are colored with the cell-type they identify.

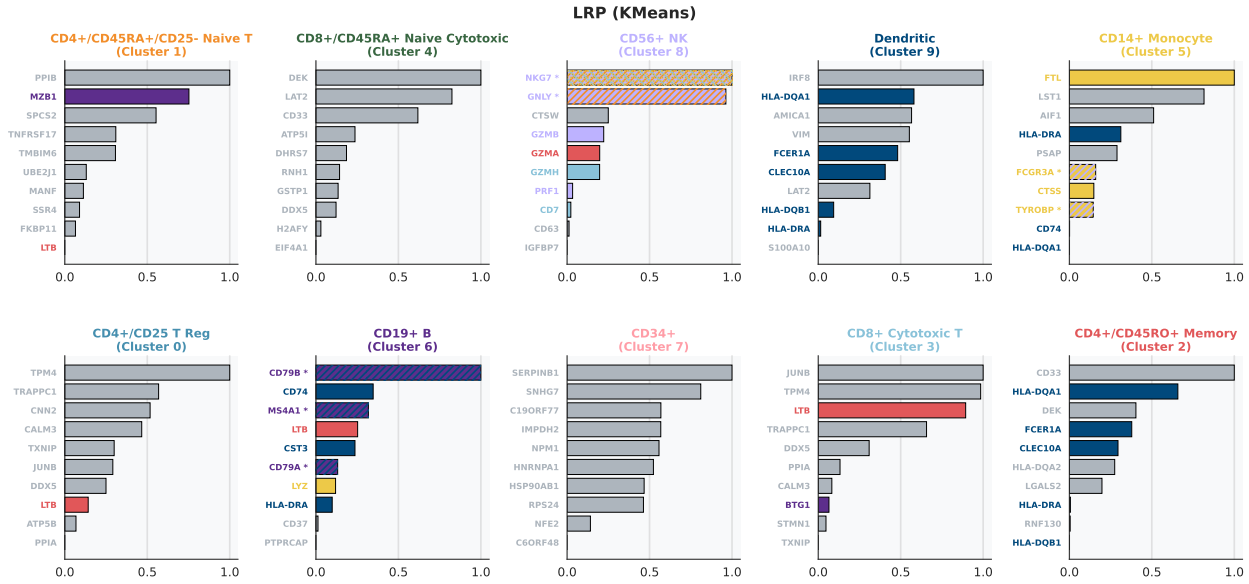

Figure 11: **Top 10 most important genes for clusters via LRP** feature importance scores, LRP uses a neuralized KMeans clustering algorithm. Labels were aligned with the reported *purified* labels. Known markers from our reference set are colored with the cell-type they identify.

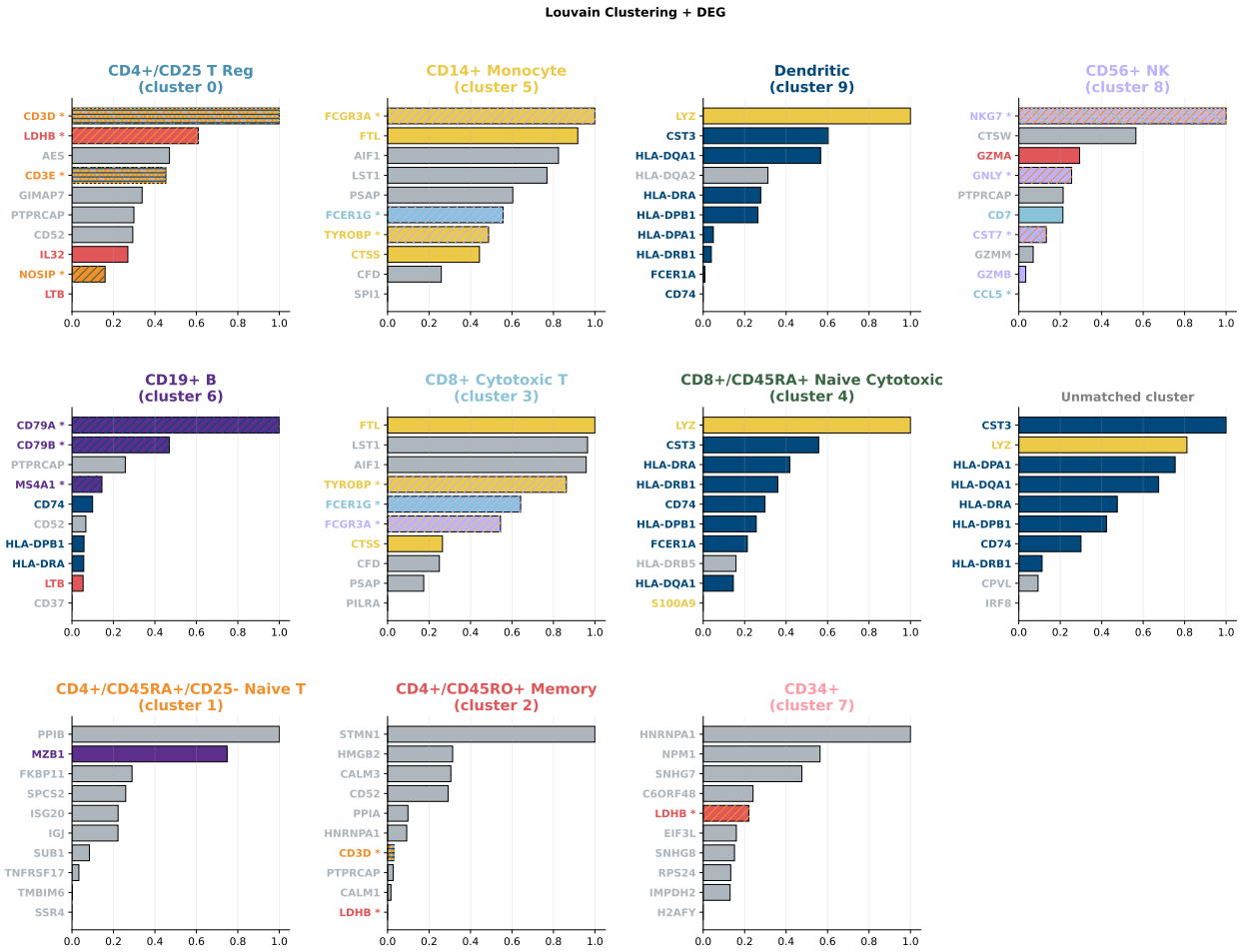

Figure 12: **Top 10 most important DEGs** obtained with Louvain clustering and t-test with BH correction. DEGs were filtered at significance level of 0.05 and ranked by z-score magnitude. Labels were aligned with the reported *purified* labels. Known markers from our reference set are colored with the cell-type they identify. We see the effect of overclustering in cluster 9, cluster 4 and the unmatched cluster.

A.4 Appendix tables

	Cluster 0 CD4+ T reg	Cluster 1 CD4+ Naive T	Cluster 2 CD4+ Mem T	Cluster 3 CD8+ Cytotoxic T	Cluster 4 CD8+ Naive T	Cluster 5 CD14+ Monocyte	Cluster 6 CD19+ B	Cluster 7 CD34+	Cluster 8 CD56+ NK	Cluster 9 Dendritic
Cluster	0/10	0/10	0/10	5/10	0/10	5/10	4/10	3/10	5/10	5/10
LOCO-MP	0/10	0/10	0/10	0/10	0/10	4/10	3/10	0/10	4/10	5/10
LRP	0/10	0/10	1/10	1/10	0/10	4/10	3/10	0/10	4/10	8/10
DEGs										

Table 1: Number of markers in the top-10 genes matching known markers of identified cell-type. We report in blue the highest proportion of consistent marker identified across methods for each cell types.

	Cluster 0 CD4+ T reg	Cluster 1 CD4+ Naive T	Cluster 2 CD4+ Mem T	Cluster 3 CD8+ Cytotoxic T	Cluster 4 CD8+ Naive T	Cluster 5 CD14+ Monocyte	Cluster 6 CD19+ B	Cluster 7 CD34+	Cluster 8 CD56+ NK	Cluster 9 Dendritic
Cluster	3/10	**1/10**	**1/10**	3/10	3/10	2/10	**3/10**	2/10	**1/10**	3/10
LOCO-MP	**1/10**	2/10	5/10	**2/10**	**0/10**	3/10	5/10	**0/10**	3/10	**0/10**
LRP	6/10	**1/10**	2/10	4/10	9/10	**1/10**	4/10	1/10	3/10	1/10
DEGs										

Table 2: Number of markers in the top-10 genes corresponding to known markers for **other** cell-types (possible false positive signal). We report in red the highest proportion of inconsistent markers across methods identified for each cell type.