**Figure e tabella da rivedere**

**Simulation**

To assess schist reliability, we applied our methodology on 4 scRNA-Seq dataset simulated with a widely used tool: SymSim [1]. This simulator makes use of many parameters to models the processes from which the data are generated. Given the complexity due to the number of parameters to tune, we used the experience gained by [2]. Therefore, we adapted to our case the GT\_simulation.R script available in their GitHub Repository (<https://github.com/BIMIB-DISCo/review-scRNA-seq-DENOISING>).

The core of the script is the SimulateTrueCounts() function from SymSim. The parameters evaluated for that function by [2] are: number of cells and genes to produce (ncells\_total and ngenes), the number of cells in the smallest population (min\_popsize), number of evfs (nevf), number of differential evfs between populations (n\_de\_evf), type of evf (evf\_type), the kinetic parameters affected by evfs (vary), the probability that the effect size is not 0 (gene\_effect\_prob), the standard deviation of the normal distribution where the non-zero gene effect sizes are sampled from (gene\_effects\_sd), parameter of the standard deviation of evf values within the same population (Sigma), random seed (randseed) and the cell developmental tree (phyla). The latest is used with the options Phyla3() or Phyla5(), which results are a population with 3 or 5 subclusters respectively. Compared to [2], we also evaluated the i\_minpop parameter, to decide which of the subclusters should be the smallest. The main difference in our simulation is the definition of Phyla8(), a function which result is a population where 8 subclusters are present.

In this way we could assess the applicability of schist and compare its results to Leiden algorithm in different settings: we tested dataset with a reduce number of clusters easily separable and equally populated, and later on more complex datasets where clusters are interrelated and a rare cell population is present in various phylogenetic position with respect to the others.

In Table 1 the parameters characterization is reported for each of the 4 dataset we produced.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | D1 (pop3) | D2 (pop5) | D3 (pop8 b) | D4 (pop d) |
| ncells\_total | 3.000 | 3.000 | 10.000 | 10.000 |
| ngenes | 10.000 | 10.000 | 10.000 | 10.000 |
| min\_popsize | 600 | 600 | 10 | 50 |
| nevf | 60 | 60 | 60 | 60 |
| n\_de\_evf | 9 | 9 | 18 | 12 |
| evf\_type | Discrete | Discrete | Discrete | Discrete |
| vary | S | S | S | S |
| gene\_effect\_prob | 0.1 | 0.1 | 0.1 | 0.1 |
| gene\_effects\_sd | 0.5 | 0.5 | 0.5 | 0.5 |
| Sigma | 0.5 | 0.5 | 0.5 | 0.5 |
| randseed | 10 | 10 | 10 | 10 |
| phyla | Phyla3() | Phyla5() | Phyla8() | Phyla8() |
| i\_minpop | 1 | 1 | 1 | 6 |

By running the modified script, we obtained our ground truth for each simulation (known cell-cluster association). To make the analysis fair, datasets were processed using the same approach already applied on the real datasets.

We observed on UMAP maps the results for each NSBM’s level, and for Leiden results (by restricting the resolution parameter to be one of [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]). We took note of changes in modularity, adjust rand index (ARI) and adjust mutual information (AMI) for every NSBM’s level / Leiden’s resolution.

***D1 and D2***

Dataset D1 and D2 were built with the predefined Phyla3() and Phyla5() functions respectively. They are characterized by a total of 3.000 cells for 10.000 genes, numbers that were slightly reduced after the filtering. Looking at the ground truth UMAP (Figure 1), D1 seems to have 3 big clusters (that should be well characterizable), and sparse cells in the space between the clusters, that could be source of misclassification. D2 instead, appears like an almost uniform cluster, where the subpopulations are equally represented and highly interconnected (especially population 4 and 5).

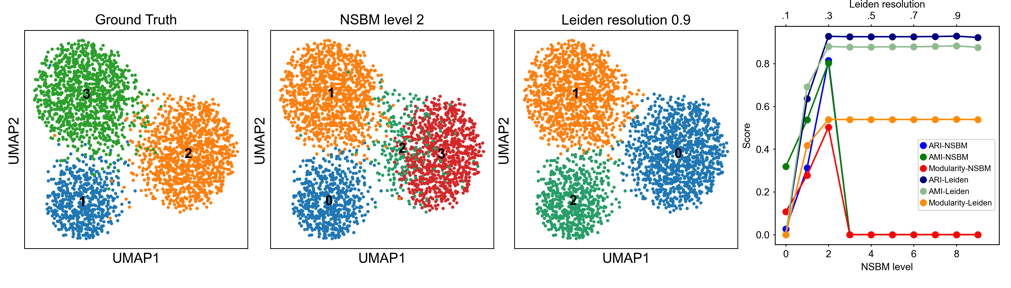
The nested model applied on D1 and D2 returned a not so good identification of the real populations. We report here only the results on D1 as they are comparable on what was observed in D2.

Figure Dataset D1. It is composed of 3.000 cells and 10.000 genes.

The Leiden algorithm performs better than NSBM in this setting if we consider the modularity, ARI and AMI scores.

In D1 there is no level in which the true clusters are correctly named. The highest scores (ARI, AMI, modularity) were obtained at level 2, where 4 clusters are computed. At level 1 the number of clusters has grown exponentially, at level 3 only one cluster is returned.

The Leiden algorithm is instead able to identify 3 clusters from resolution 0.3 to 1, even if it is not completely correct in the assignment of the sparse cells to a cluster. The highest scores were obtained for resolution 0.9.

We observed for both NSBM and Leiden algorithms a correspondence between the highest modularity and the highest AMI/ARI. This is especially true for NSBM, where there is a great difference in the scores looking at one level or another. This observation may be used as an indicator of which level would be better to analyse in the first instance (even if each level is a possible and correct choice). For the Leiden algorithm this is not an indicator because we observed that the scores remained almost constant after reaching the plateau. This means that the resolution choice remains arbitrary in a wide range (for D1 the range is [0.3-1]).

***D3 and D4***

D3 and D4 are datasets computed with Phyla8() function: 10.000 cells for 10.000 genes are divided in 8 clusters.

The main differences between D3 and D4 are due to the n\_de\_evf parameter that was modified from 18 in D3 to 12 in D4. Another difference is in the characterization of the rare population (rare population that was not in D1 and D2).

The D3 ground truth UMAP plot show 3 clusters (6,7,8) that are very close to each other, making a cloud. Other 3 clusters (2,3,5) are near the cloud and sparse cells are in the space between them and the cloud. Population 4 is clearly separable. Cluster 1 represents the rare population composed of 10 cells, in the plot positioned close to population 2.

D4 differs from D3 for the characterization of the rare population: in D4 it is cluster 6 (50 cells only) and it is almost overlapping with cluster 5. Clusters appears more distant compared to D3.

NSBM and Leiden algorithms applied on D3 and D4 resulted in comparable modularity and AMI/ARI scores. Like observed for D1 and D2, also for D3 and D4 the best level to analyse the dataset seems to be the one with the maximum scores.

In D3 the Leiden algorithm fails to identify an adequate number of clusters: it returns up to 6 different clusters and it is unable to recover the rare population.

The main accomplishment for NSBM is instead to have the rare population identified in the level with maximum scores. It is also remarkable that NSBM gives a better classification of the population inside the cloud compared to Leiden.

D4 clusters calling is almost identical between NSBM and Leiden: both algorithms fail to recognize the rare population and to distinguish the 7 and 8 clusters.

Probably in the case of the D4 rare population, the design was too complex for the algorithms to be efficient.

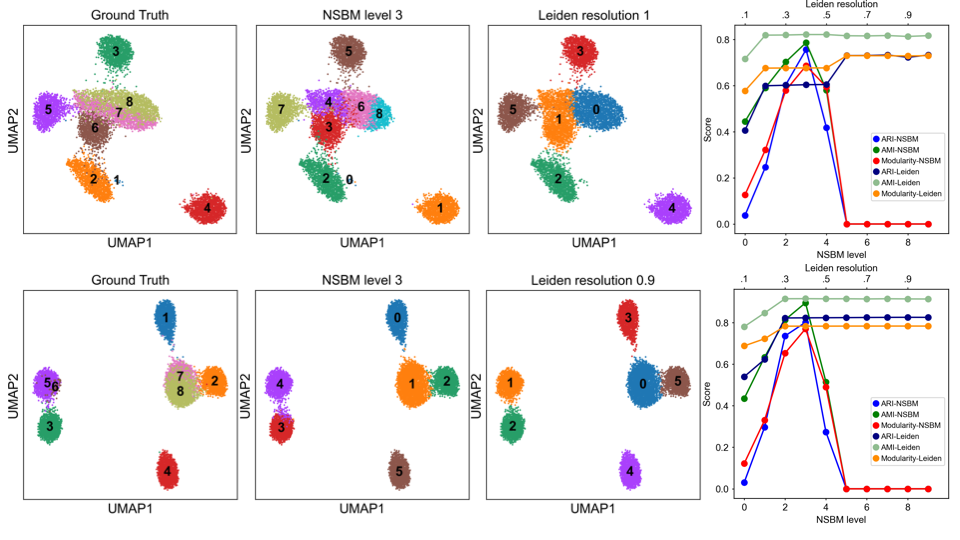


Figure 2 D3 and D4

***Fast option***

D1-D4 were analysed also with the fast\_model=True option, resulting in a decreased ability to correctly identify cells true identity.

***Comments-Conclusion***

NSBM and Leiden algorithms were applied on simulated datasets characterized by 3, 5 or 8 subpopulations. In the latest case, a rare population was also defined.

Modularity, AMI and ARI were evaluated varying level (for NSBM) or resolution (for Leiden). Not much difference was observed in the scores for the two algorithms. For NSBM, the scores trend can be used to choose which level to analyse: the level with the maximum values has clusters that highly resemble the ground truth. For Leiden, this conclusion can not be draw as scores reached a plateau: a wide range of resolution values results similar behaviour.

In simpler contexts (D1 and D2) Leiden seems to perform better compare to NSBM, which instead seems to prevail (or at least has equal results) if the population structure is more complex (D3 and D4).

Given the importance of rare population characterization, and the high probability to have samples where the expected number of clusters is more than 3, NSBM can be a valid alternative to the application of Leiden algorithm.

[1] X. Zhang, C. Xu, and N. Yosef, “Simulating multiple faceted variability in single cell RNA sequencing,” *Nat. Commun.*, vol. 10, no. 1, 2019.

[2] L. Patruno, D. Maspero, F. Craighero, F. Angaroni, M. Antoniotti, and A. Graudenzi, “A review of computational strategies for denoising and imputation of single-cell transcriptomic data,” *Brief. Bioinform.*, pp. 1–20, 2020.