For this study the methods were evaluated using six different datasets and were in different run modes. By running the clustering methods using the unfiltered and filtered datasets, we can (to some extent) investigate the effect of the filtering and normalisation steps on the clustering. RaceID, Linnorm, TSCAN and Seurat each have their cell or genewise filters implemented; these were included when the unfiltered datasets were used.

By running the methods under the default mode, we can investigate the performances with a minimum user effort. Also, some methods are able to autodetect the number of clusters, and this setting was included when running in the default mode. The methods were then run with a refined parameter setting and the annotated number of clusters. Note that the number of parameters varied considerably between the methods; in this study, we chose only the parameter that is regarded as the most important.

The clusterings of the methods were assessed by the use of the ARI and F1 metrics. The ARI scores for the different run modes are shown in Figure \ref{fig:ariall} and \ref{fig:ariradar}. According to the dataset used, Figure \ref{fig:aridiff1} shows the differences in ARI scores between the method with the highest ARI score and the other methods. The differences between the ARI scores of the different run modes are shown in Figure \ref{}. The comparison of the different run modes is strictly speaking only possible for deterministic methods such as CIDR, and is not possible for stochastic methods (e.g. tSNEkmeans ). The F1 scores for the filtered dataset are shown in Figure \ref{fig:f1pointplot}. The results for the run mode with the default setting, the unfiltered data and the optimal number of clusters are shown in the Appendix (see Figure \ref{fig:f1poindef},\ref{fig:f1pointunf} and \ref{fig:f1pointopt}.)

The datasets and simulations used varied in the number of cells, the library sizes, the number of subpopulations, the zero-fractions per gene and the type of the expression values (see Table \ref{tbl:data} and Figure \ref{}). In order to assess the accuracies of the methods some ground truth of the type of subpopulations is needed. Here, we used the annotation given by the authors of the datasets. This may not be correct; the cell types could be wrongly annotated, or the annotated clusters might consist of more refined unknown subpopulations. However, here we used this annotation because it was seen as the best possible information available. The datasets were chosen such that there was a range in clustering difficulty. As an objective measure of the clustering difficulty, the average silhouette coefficient is used.

\paragraph{}

The average silhouette width of the Kumar dataset is with 0.53 the highest for all datasets. It consists of three distinct cell populations and is the simplest of all the datasets. A high proportion of the variances are explained by the cell type or the batch effect, as these two are not separable (see Figure \ref{}e ). The number of expressed genes in cell population \dots is substantially higher than in the other two cell types. This dataset can be seen as a benchmark for the dataset as no method should have problems in clustering this dataset. Using the filtered datasets the methods SC3, pcaReduce, SIMLR, CIDR and ZINBWaVE achieved a correct partition of the cells. The other methods also achieved high accuracies with ARI scores between 0.97 and 0.99. The F1 scores give a more in-depth view of the actual partitioning, and for the filtered data we have similar high F1 scores for each of the subpopulations, showing that no method failed to cluster one of the subpopulations. In contrast to the uniform results with the filtered datasets, the results were more variable when running in default mode and with the unfiltered datasets. Running with the default setting, and automatically detecting the clusters, methods RaceID, SC3 and TSCAN failed to detect the correct number of clusters. RaceID and TSCAN partitioned the cells into four clusters and SC3 into five clusters. Additionally, TSCAN failed to correctly partition the cells into three subgroups. Note that for SC3 and TSCAN the extra cluster consisted only of a few cells and had only a marginal effect on the final clustering. ZINBWaVE, tSNEkmeans, pcaReduce and Linnorm failed in clustering the three populations correctly.

With the unfiltered datasets, the methods show similar results where most of the methods again achieved ARI scores close to one. There are two exceptions that had a drop in the performance: Linnorm and ZINBWaVE. These two methods failed by clustering one particular subpopulation.

\paragraph{}

The simDataKumar simulation has an average silhouette width of 0.15 and is one of the simpler datasets for clustering. Three out of four subpopulations in the dataset are distinct, with a high proportion of DE genes. Only 5 \% of the subpopulation Group 1 are DE genes and based on the tSNE representation of the dataset the two subpopulations Group1 and Group2 are not distinguishable (see Figure \ref{}). For the filtered data and run with the annotated number of clusters, the methods SC3, Seurat, pcaREduce SIMLR and CIDR had high ARI scores between 0.95 and 1.00. Also, Linnorm and TSCAN showed a somewhat lower performance with an ARI score of 0.87 and 0.90, respectively. However, tSNEkmeans, TSCAN and RaceID failed to correctly cluster the dataset with ARI scores between 0.31 and 0.65. The low ARI scores are due to failing to partition the subpopulation Group 1.

When running under the default mode, pcaReduce also dropped in performance. TSCAN failed in clustering the results due to the high threshold for the zero expression.

CIDR, RaceID, SC3 and Seurat, the methods with an autodetect function, were all able to correctly identify the number of clusters. Running the data in default mode had no impact on the clusterings for SC3, CIDR, RaceID, tSNEkmeans and SIMLR. However, pcaREduce, Seurat and ZINBWaVE had a different partition. Using unfiltered data affected the methods Linnorm, RaceID and TSCAN. Note that for Linnorm and RaceID this could be due to the stochasticity of the method. For the other methods, it had no impact, and the ARI scores were stable.

The Zheng dataset is a mixture of four populations of PBMC cells. The two subpopulations CD19+B and CD14+monocytes are distinct cell populations, whereas the naive cytotoxic and regulatory T cells are overlapping populations. The tSNE representations show that CD19+B and CD14+monocytes form two separate clusters, with a third cluster that consists of the two nested populations naive cytotoxic and regulatory T cells. The dataset has a medium difficulty with an average silhouette width of 0.1. The dataset has a low sequencing depth and a high dropout rate.

The performances given by the methods were highly variable on the different run modes. With the filtered data SC3, pcaReduce and tSNEkmeans had ARI scores near one. Also, Seurat, Linnorm and SIMLR had high accuracies with ARI scores between 0.88 and 0.92. CIDR dropped in performance when compared to its performance for other datasets. The Zheng datasets consist of UMI counts, for which CIDR is not designed.

When running in the default mode CIDR, SC3 and Seurat detected an extra cluster, which explains the lower performance for Seurat and SC3 in the default run mode.

With the exception of method SIMLR (for large-scale data), using filtered data had no impact on the performance of the methods.

The more difficult simulation simDataKumar2 has an average silhouette width of 0.03. For this dataset, all the proportions of DE genes are relatively low with five to eight percent. The subpopulation Group 3 is distinguishable from the other three in the tSNE representations. Group 1, Group 2 and Group 4 form a single non-separable cluster in the tSNE representation. SC3, SIMLR, Seurat, CIDR and pcaReduce were mostly able to correctly cluster the cells when using the filtered datasets and run with the annotated number of clusters. The ARI scores were between 0.9 and 1.00. The other methods showed profoundly lower performances. When running under the default mode, SC3 and RaceID detected only three of the clusters when using the autodetect function of the methods, explaining the drop in the ARI scores for these methods. The methods Linnorm, tSNEkmeans, TSCAN and to a lesser extent SIMLR (large scale) showed a decrease in the ARI scores. For the other methods using the unfiltered did not affect the clusterings.

The Trapnell dataset is not a mixture of distinct cell populations, and the development of the populations followed a time-dependent trajectory. In tSNE space, the non-differentiated cells form a distinct cell cluster. However, the more differentiated cells later on the time axis are, at least in tSNE space, non-separable. It is also notable that the batch explains a higher proportion of the variance than the cell type (see Figure \ref{} ). The average silhouette width is 0.04 and it is one of the more difficult datasets.

The methods showed the lowest ARI scores for this dataset, and the maximum ARI score achieved SC3 with 0.55, showing the difficulty to cluster this dataset. The other methods all had ARI scores below 0.5. TSCAN is specially developed for this scenario \citep{ji2015tscan}. However, the method did not perform any better than the other methods. To improve the clustering results for TSCAN, it is possible to provide a starting point for the trajectory. However, this was not done in this study. By running the dataset in the default mode, the ARI scores varied considerably compared to the run mode with the filtered and annotated number of clusters. Indeed, depending on the method, the scores were better or worse or stable. SC3 detected additional three clusters. RaceID detected only one single cluster. CIDR and Seurat found three subpopulations but failed to correctly label the cells. Except for TSCAN, the filtering led to an increase in the ARI scores. Without filtering most of the methods failed to cluster the dataset as they had scores around zero.

Koh was the dataset that had the highest number of subpopulations; ten clusters were annotated by the authors of the original study. During the cell filtering of the Koh dataset one subpopulation was wrongly detected as outlier cells and was removed during the filtering steps (D3GARPpCrdcM). The average silhouette width is -0.04 and the dataset is one with the highest difficulty. In the tSNE representation, some cell types are easily distinguishable, whereas other are nested. On average, 10 \% of the variance is explained by the cell type. Similar to the Zheng dataset, the Koh data has a low sequencing depth and a high dropout rate. For the filtered data only SC3 was able to correctly identify all the nine subpopulations with the highest ARI score of 0.96. CIDR, SIMLR, pcaReduce and Seurat also showed good performances with ARI scores between 0.88 and 0.92. In the middle field are the methods Linnorm tSNEkmeans, Linnorm and ZINBWaVE with ARI scores between 0.62 and 0.8. RaceID failed for this data with an ARI score of 0.26. Compared to other run modes, the clustering was the overall worst for each method when running with the unfiltered dataset, indicating the importance for pre-filtering this dataset. Except for the unstable methods, Linnorm and tSNEkmeans, running the methods in default mode only had a small impact on the ARI scores. However, the number of detected clusters varied greatly between the methods with an auto detect function. CIDR, TSCAN and RaceID missed one or more clusters, and SC3 detected an additional three clusters.

Overall, Kumar provided no difficulties for most of the methods. Most methods also achieved high scores for the SimDataKumar2 and the Zheng datasets. The Trapnell dataset was a challenge for the methods, and only low accuracies were achieved. When looking at the F1 scores and the size of the clusters, wrongly assigned clusters tend to be smaller in size. The investigated methods either used PCA or tSNE for dimension reduction. For the clustering, either graph-based, K-means, hierarchical clustering or combinations of these are used. No connection between the performance and the type of the dimension reduction or clustering approach can be seen.

\subsection{Evaluation of the performances of the methods}

Based on the average scores, SC3 was the best performing method when running with the filtered and unfiltered datasets. Exceptions are seen in the default and unfiltered setting when using the Trapnell and the Zheng datasets. A pre-filtering step and a fine-tuning of the parameters for this method is recommended, as this improved the accuracy of the method. It is noteworthy that the method was able to correctly classify more than 95\% percent of the cells for the Kumar, Koh, simDataKumar, simDataKumar2 and Zheng datasets. Even if it had low scores for the Trapnell data, it achieved the highest accuracies of all methods in this dataset. SC3 can detect the numbers of clusters automatically and did so correctly for the simulated datasets. For the other datasets, the number of clusters was higher than in the annotation. It is unknown whether this is due to the existence of more refined subpopulations.

Similar to SC3, Seurat achieved high accuracies. The method had the highest average ARI score when running under the default settings. This is particularly interesting if the number of clusters is unknown, as Seurat determines the number of clusters automatically (albeit, it is possible to adjust the granularity of the clusterings). However, Seurat’s performance levels dropped when used with the Koh and Zheng datasets. The filtering of the Koh dataset led to an improvement in the accuracies but otherwise, when using the filtered datasets, only small changes were detected. According to \citet{butler2017integrated}, the resolution parameter for the function is crucial to the ability to determine the number of clusters, and it is recommended that the method is tested using different values of this parameter. In this study, we were only able to run the methods on a small range of values in the parameter, as else it was not possible to run the method. However, Seurat was able to detect the correct number of subpopulations in most of the datasets and run modes. The exceptions are in the Zheng data where an additional cluster was detected and the Koh datasets were the method missed one subpopulation. The default for the number of neighbours for the kNN is set to 30, and no or only small differences in the clustering are achieved when this parameter was set to 10 percent of the dataset (see Figure \ref{fig:diff}).

Although CIDR showed an overall high level of accuracy, it had one of the lowest performances for the Zheng dataset of all the tested methods. A possible reason for its low performance is that the expression values are UMI counts and the data has a low sequencing depth, leading to a wrong model fit in the imputation procedure. Filtering of the datasets improved this method’s performance for two out of the six different datasets, else it had no impact on the performance of the method. It was attempted to further improve this method’s performance by selecting an appropriate number of PCs; however, no improvements in accuracy were achieved. Except for the Koh and Zheng dataset, the method was able to identify the correct number of subpopulations.

In comparison to the other well-performing methods, SIMLR had similar or higher scores for Koh, Kumar, and the simulated datasets. It dropped in performance for the Trapnell, Zheng and Trapnell data. Filtering the datasets led to an improvement for the Trapnell and Koh datasets. Mean scaling of the dataset, one of the parameter settings, had no impact on the performances. However, it is necessary to do so as else the method is not able to perform the eigenvalue decomposition of the learned similarity matrix.

The method tSNEkmeans achieved similar accuracies compared to the high-performing methods when used with the simple Kumar, simDataKumar, and Zheng datasets, but it showed consistently low accuracies for the Koh dataset. The k-means algorithm assumes spherical clusters which, in the tSNE representation of the dataset Koh, are not given. The influence of a reduced number of PCs as input for the tSNE dimension reduction was dependent on the datasets. However, though this improved the performance of the Kumar data, it had a negative impact on the Koh dataset. Nevertheless, no changes in ARI scores were detected for the other datasets. It is unclear whether gene filtering had a positive or negative influence on the clustering, as it varied highly between the datasets. The method is unstable, and the changes in the performances could also be due to the stochasticity of the method.

Linnorms performance is in the medium range. It was able to cluster the Kumar and the Zheng datasets correctly but was in medium range for the other dataset when compared to well-performing methods as SC3 and Seurat. Changes in the filtered parameters seemed crucial and led to an improvement in the clustering results for the Zheng, Kumar and simDataKumar datasets. However, the score for the Koh dataset was worse. The minimum zero fraction was set to 0.75, which is probably too high if the high dropout rate in the Koh data is considered. However, note that the method was highly unstable, and the results for the different run modes may not be comparable. Linnorm had relatively high accuracies for the unfiltered Trapnell and Zheng datasets. This could be due to the filtering functions that are implemented in the method.

The methods ZINBWaVE and TSCAN only had high accuracies for the Kumar datasets. Filtering of the data improved the accuracies for ZINBWaVE, whereas for TSCAN it only improved the performance for the Koh dataset. Otherwise, the ARI scores didn't change or were worse for the data. The method TSCAN is designed to find trajectories. This type of data is given in the Trapnell data. Here the method achieved ARI scores between 0.6 and 0.7, which is not better than other methods. In this study, a start and end point of the trajectory could be given which, according to the authors, could improve the results. In this study, however, this was not done. It was not possible to run TSCAN in the default mode for the Zheng and the simulated datasets.

RaceID had the lowest performance and returned only a high ARI score over all runmodes when used with the simple Kumar data. It performed well for the Zheng dataset with the default setting but failed for the other two runmodes. If this is due to a poor choice of the parameter settings, or it was due to the stochasticity of the method remains unclear. The method is based on absolute transcript counts which can explain the bad performance in the non-UMI based datasets.

Overall, SC3, Seurat and SIMLR are the methods with high accuracies. Also, CIDR performed comparable to the before mentioned methods, except when running with UMI counts. Due to the stochasticity of pcaReduce and tSNEkmeans, the results are unstable, and the performance was dependent on the actual run. Linnorm, TSCAN, ZINBWaVE and, in particular, RaceID showed overall low ARI scores.

Using filtered data generally improved the results for the Koh and the Trapnell dataset, with only two exceptions, namely TSCAN and Linnorm.

For the other datasets, the genewise filtering had different effects on the clustering results. Mostly the ARI scores were stable. The methods Linnorm, RaceID tSNEkmeans, TSCAN and ZINBWaVE showed strong changes in the ARI scores in at least one of these datasets. If this changes are due to the stochasticity of the methods is unclear.

All methods showed differing ARI scores for at least one dataset when comparing the default mode and when run with the annotated number of clusters. It is not clear whether the methods perform better under the default settings or with a changed parameter setting and naturally depend on the method. As stated above, especially for tSNEKmeans, pcaREduce, RaceID and Linnorm, any changes in the clustering could be due to the stochasticity of the methods. It was also highly dependent on the dataset, and for most methods, it only had a slight impact, negative or positive, on the ARI score. However, we note that for SC3 the ARI scores were higher when the number of clusters is given.

The methods were run under a range of the number of clusters, and for each clustering result, the ARI score was computed. Seurat did not allow for direct control of the number of clusters. Instead, the resolution parameter was used. However, it was only possible to run the method on a small range of the resolution parameter. The results are shown in Figure \ref{fig:arirangeall}. The methods behaved differently depending on the difficulty of the datasets and the number of subpopulations. For example, most of the methods had clear maximums when used with the simple Kumar dataset, whereas, when used with the more difficult Koh dataset, the methods showed a monotonic increase in the ARI and reached a plateau within five to ten cluster. For the Trapnell data, SC3, Linnorm, TSCAN and tSNEkmeans had clear maximum values with two clusters and then had a decrease in the scores for a higher number of clusters. The methods CIDR, pcaReduce, RaceID and SIMLR had no clear maximums and all had a plateau between two to five clusters. For the Zheng dataset, tSNEkmeans, SC3, Linnorm, ZINBWaVE and SIMLR showed a similar behaviour with clear maximum with four clusters. The other methods had no such clear maximum values, or the maximum score is not at the annotated four clusters. Generally, the methods tSNEkmeans and SC3 showed a similar behaviour.

Subsampling without replacement was used to assess the stability of the methods. Based on the wide range of algorithms for the methods, the methods showed results with varying levels of stability (see Figure \ref{fig:stab}). The deterministic method CIDR is stable. Seurat and TSCAN were mostly stable and showed some outlying runs with a slight decrease in the ARI scores. This is in contrast to Linnorm and pcaReduce, which had some strong outlying runs. pcaReduce and RaceID are both very unstable, and the assignment of the cells to the respective cluster varied greatly. Here, the simple Kumar dataset is used, and it can be expected that the methods will behave even more unstable with a more difficult dataset.

The runtimes for each method can be seen in Figure \ref{fig:runtimelog}. They were highly different, ranging across two magnitudes for the studied methods. The fastest methods were Seurat, CIDR, and SIMLR (large-scale), whereas the methods pcaReduce, SIMLR, RaceID, and SC3 showed the highest runtimes. It is notable that SC3 and SIMLR both have a non-linear increase in runtime, making their use infeasible with a larger dataset consisting of thousands of cells. However, for bigger data sets ( $>$ 5000 cells) \citet{} recommend clustering the data with the use of support vector machines which improves the runtime. For small-scale datasets, Seurat is one of the fastest methods. However, when used with the larger Zheng dataset (with 2,000 cells), its runtime lies in the middle ground.