Imputation Methods for single cell RNA-Seq data

Marmar Moussa
Ion Măndoiu
Computer Science & Engineering Department
University of Connecticut

Imputation Methods for scRNA-seq data

There are two types of people in this world:

1. Those who can infer information from missing data

Drop-outs

- occurring because of inefficient mRNA capture,
- or naturally due to low number of RNA transcripts and the stochastic nature of gene expression,

→the result is capturing only a fraction of the transcriptome of each cell and hence data that has a high degree of sparsity.

Existing single cell RNA-Seq imputation methods

- The DrImpute R package implements imputation for scRNA-Seq based on clustering the data.
- First DrImpute computes the distance between cells using Spearman and Pearson correlations, then it performs cell clustering based on each distance matrix, followed by imputing zero values multiple times based on the resulting clusters, and finally averaging the imputation results to produce a nal value for the drop-outs.

Existing single cell RNA-Seq imputation methods

- The scImpute R package makes the assumption that most genes have a bimodal expression pattern that can be described by a mixture model with two components.
- The first component is a Gamma distribution used to account for the drop-outs, while the second component is a Normal distribution to represent the actual gene expression levels.
- The parameters in the mixture model are estimated using Expectation-Maximization (EM)

Existing single cell RNA-Seq imputation methods

- Weighted K-nearest neighbors (KNNimpute), a method originally developed for microarray data, selects genes with expression profiles similar to the gene of interest to impute missing values.
- For instance, consider a gene A that has a missing value in cell 1, KNN will find K other genes which have a value present in cell 1, with expression most similar to A in cells 2 to N, where N is the total number of cells.
- A weighted average of values in cell 1 for the K genes closest in Euclidean distance is then used as an estimate for the missing value for gene A.

Proposed method: locality sensitive imputation (LSImpute)

- Given a set S of n cells , start by selecting m cells with highest similarity level, until at least m_min are selected or the highest pair similarity drops below a given threshold.
- Cluster the m cells using a suitable clustering algorithm into ``tight'' clusters.
- For each cluster, replace zeros for each gene j with values imputed based on the expression levels of gene j in all cells within the cluster.
- The selected cells now have imputed values and the clusters they form are collapsed into their respective centroids.
- The centroids are pooled together with unselected cells to form a new S.

Proposed method: locality sensitive imputation (LSImpute)

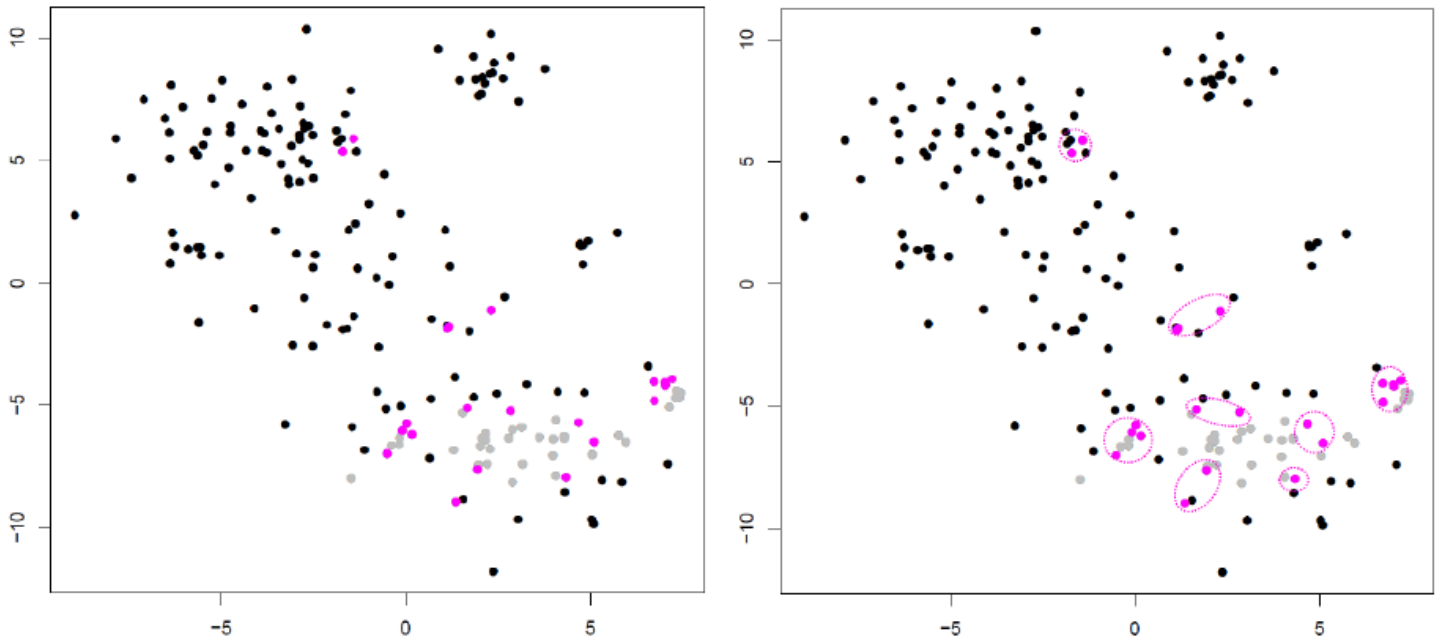


Fig. 1. Illustration of Steps 1 (left) and 2 (right) of the LSImpute algorithm. Gray dots represent already processed cells and collapsed centroids from previous iterations. Pink dots represent cells in pairs with highest similarity level which are selected for clustering.

Data

- ultra-deep scRNA-Seq data generated for 209 somatosensory neurons isolated from the mouse dorsal root ganglion (DRG)
- An average of 31.5M 2*100 read pairs
- detection of an average of 10,950+/-1,218 genes per cell.
- simulate varying levels of drop-out effects : 50K, 100K, 200K, 300K, 400K, 500K, 1M, 5M , 10M, respectively 20M read pairs per cell.
- As ground truth we used TPM values determined by running IsoEM2 on the full set of reads.

Data

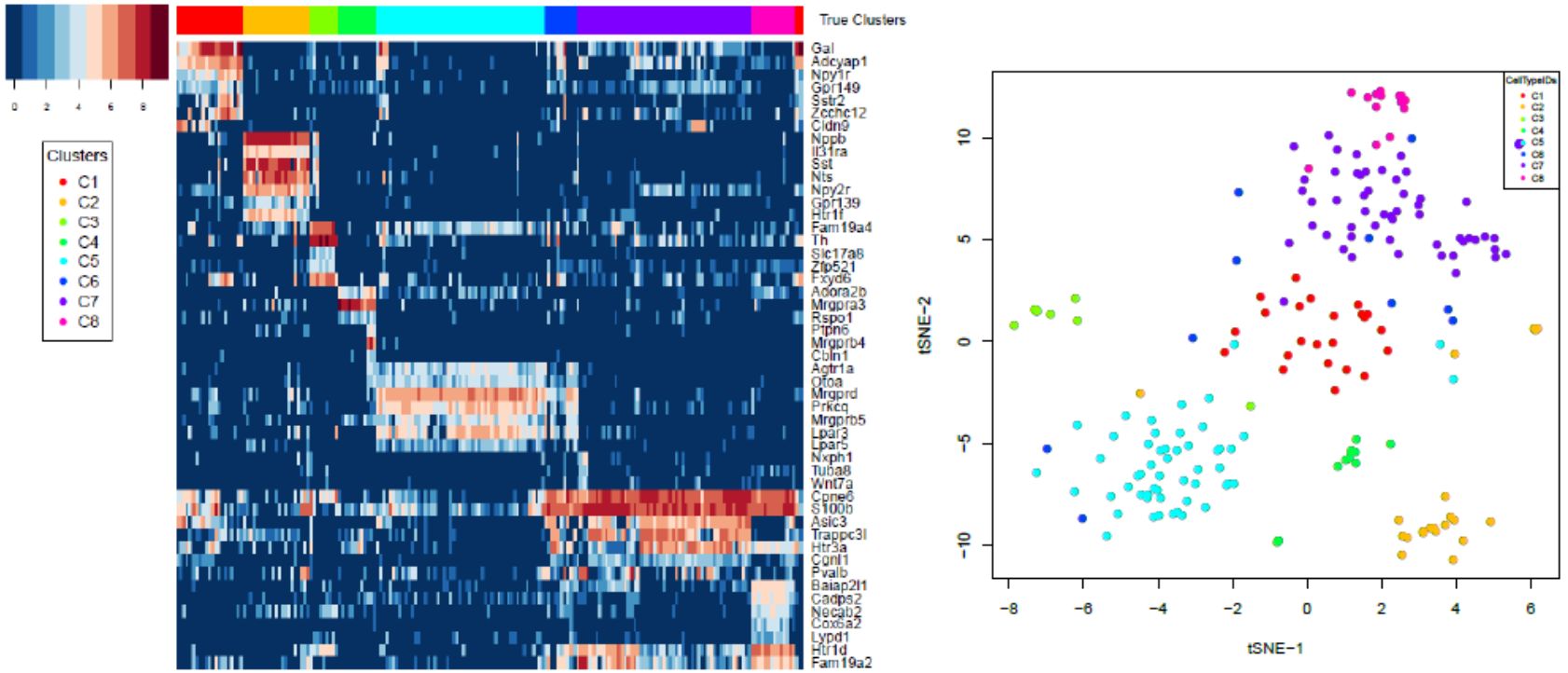
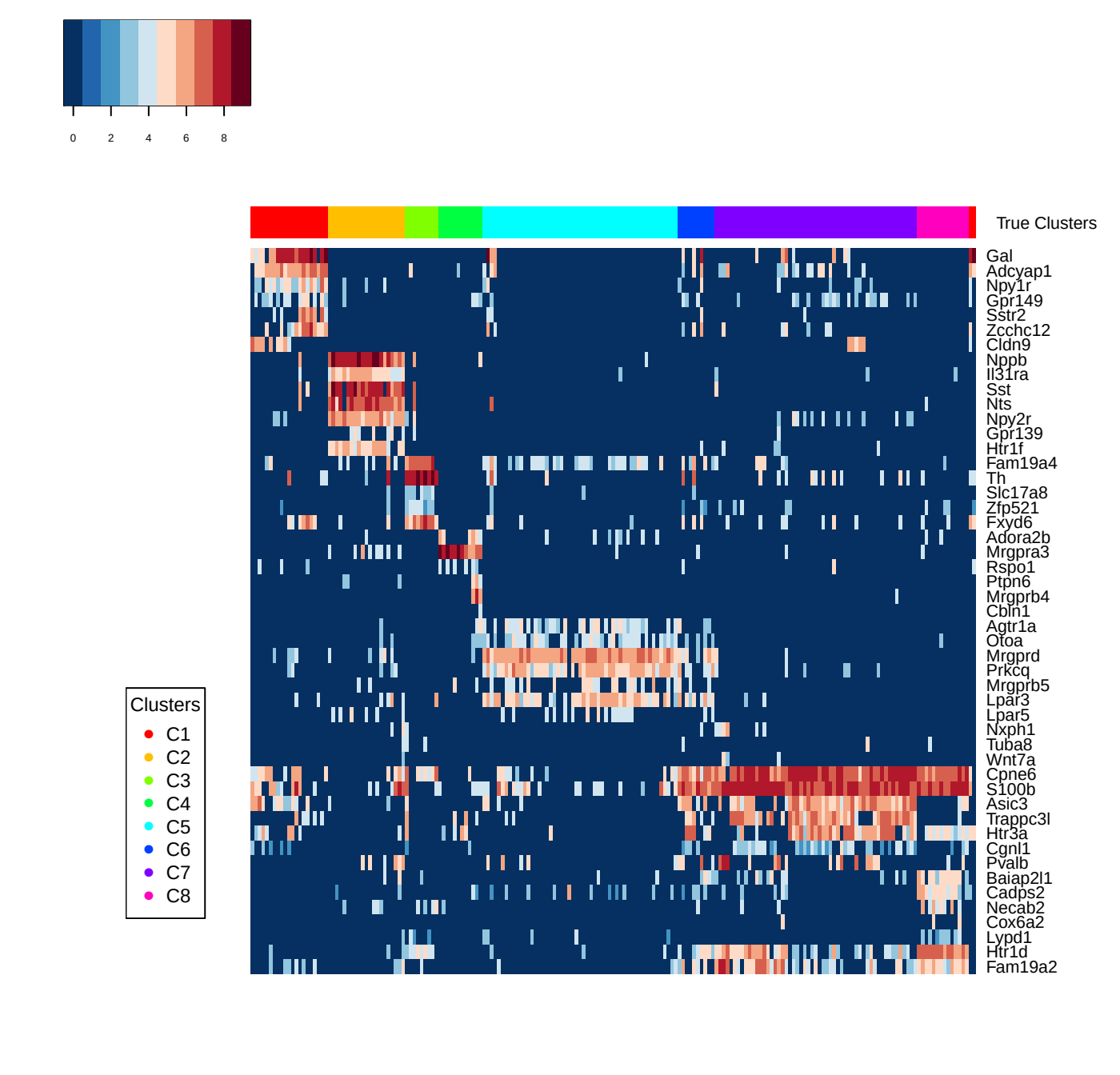
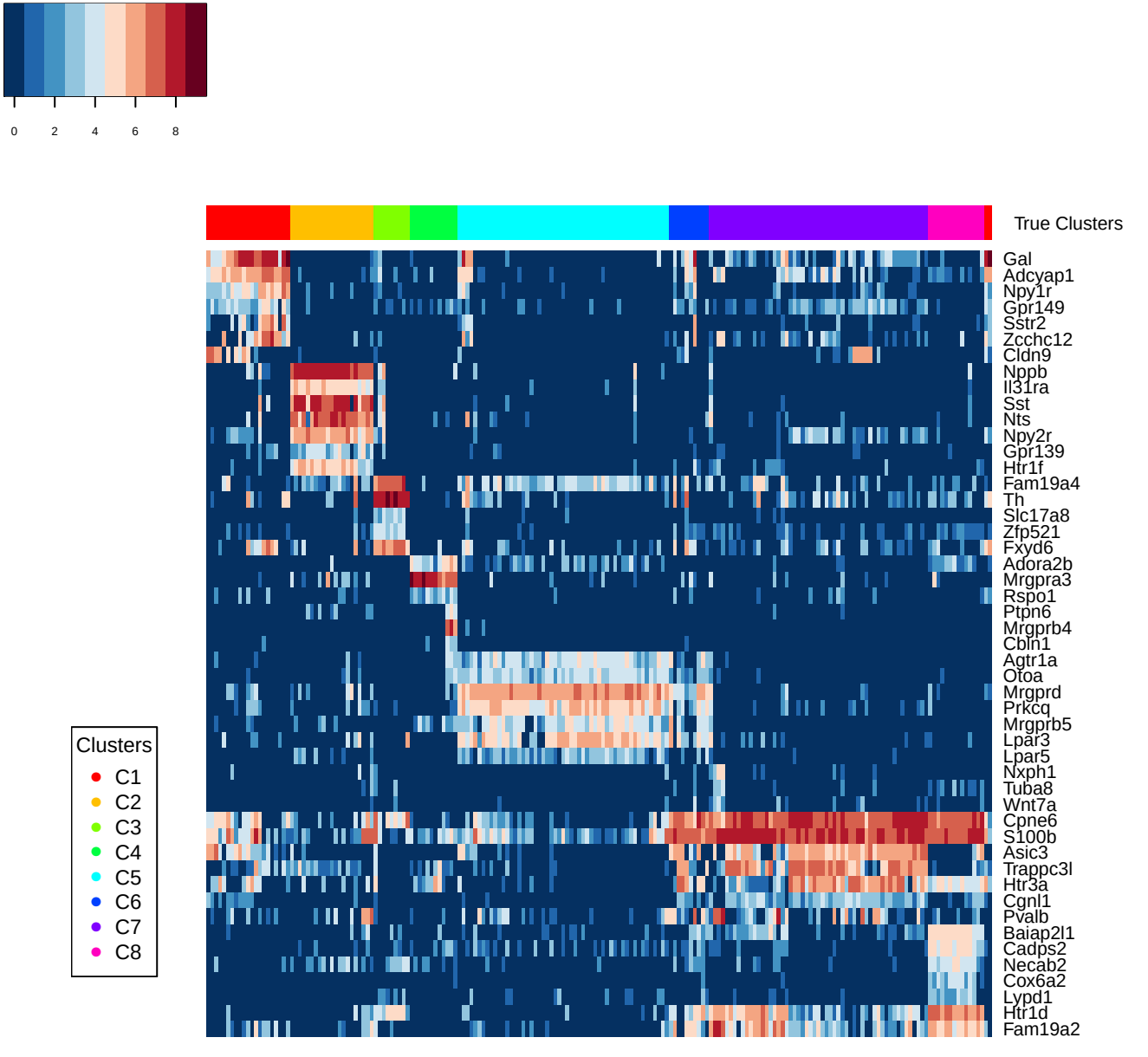
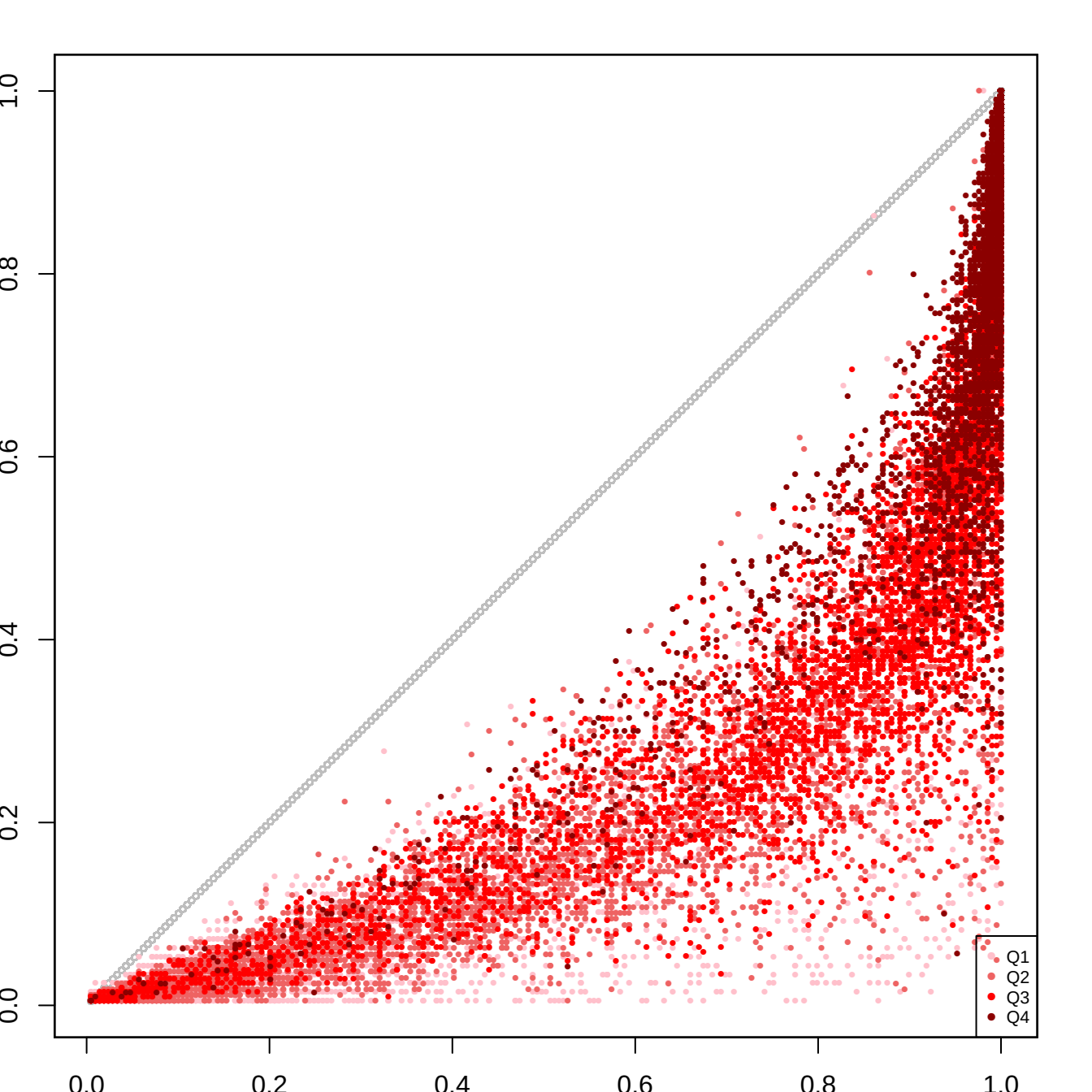
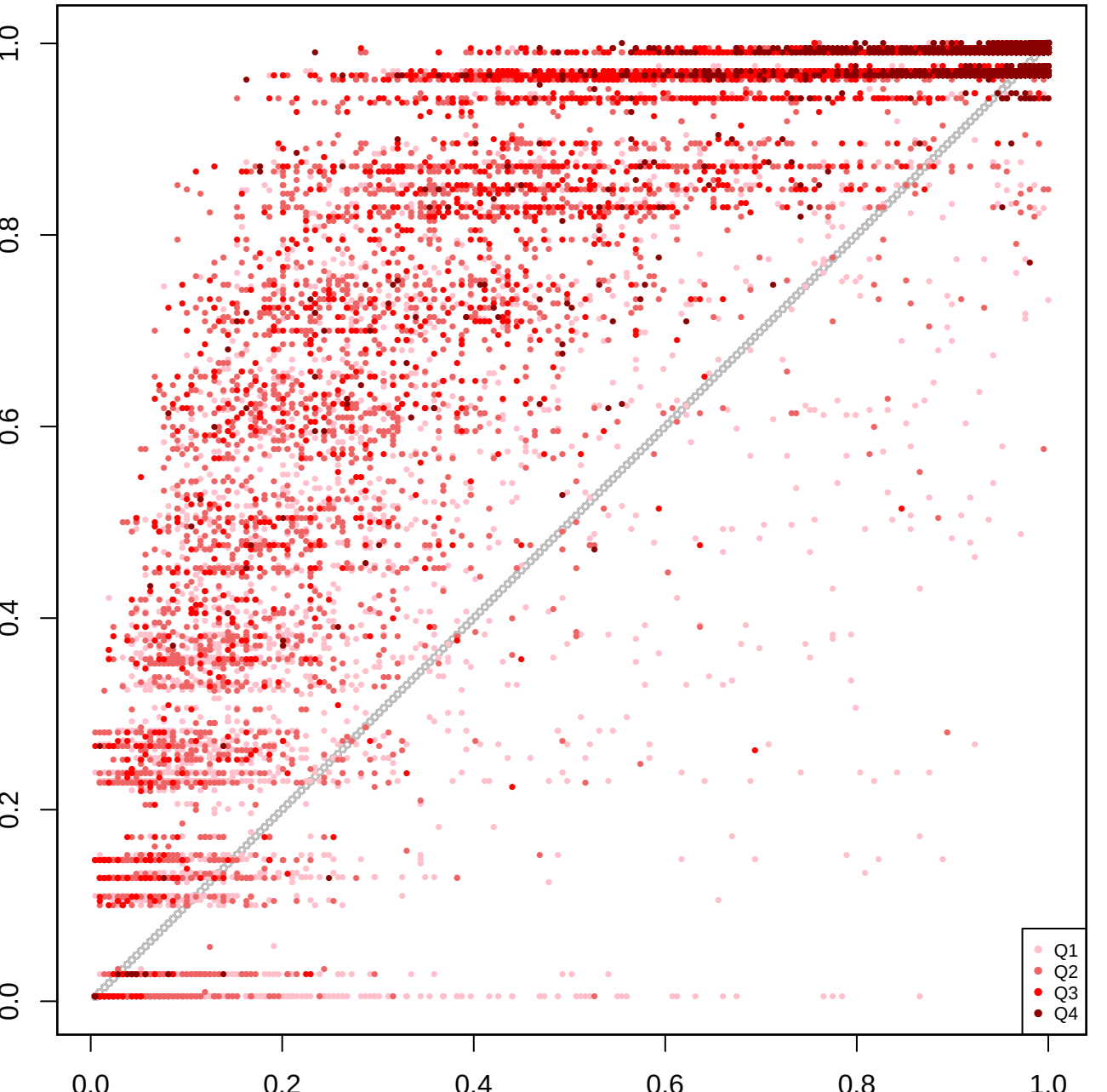


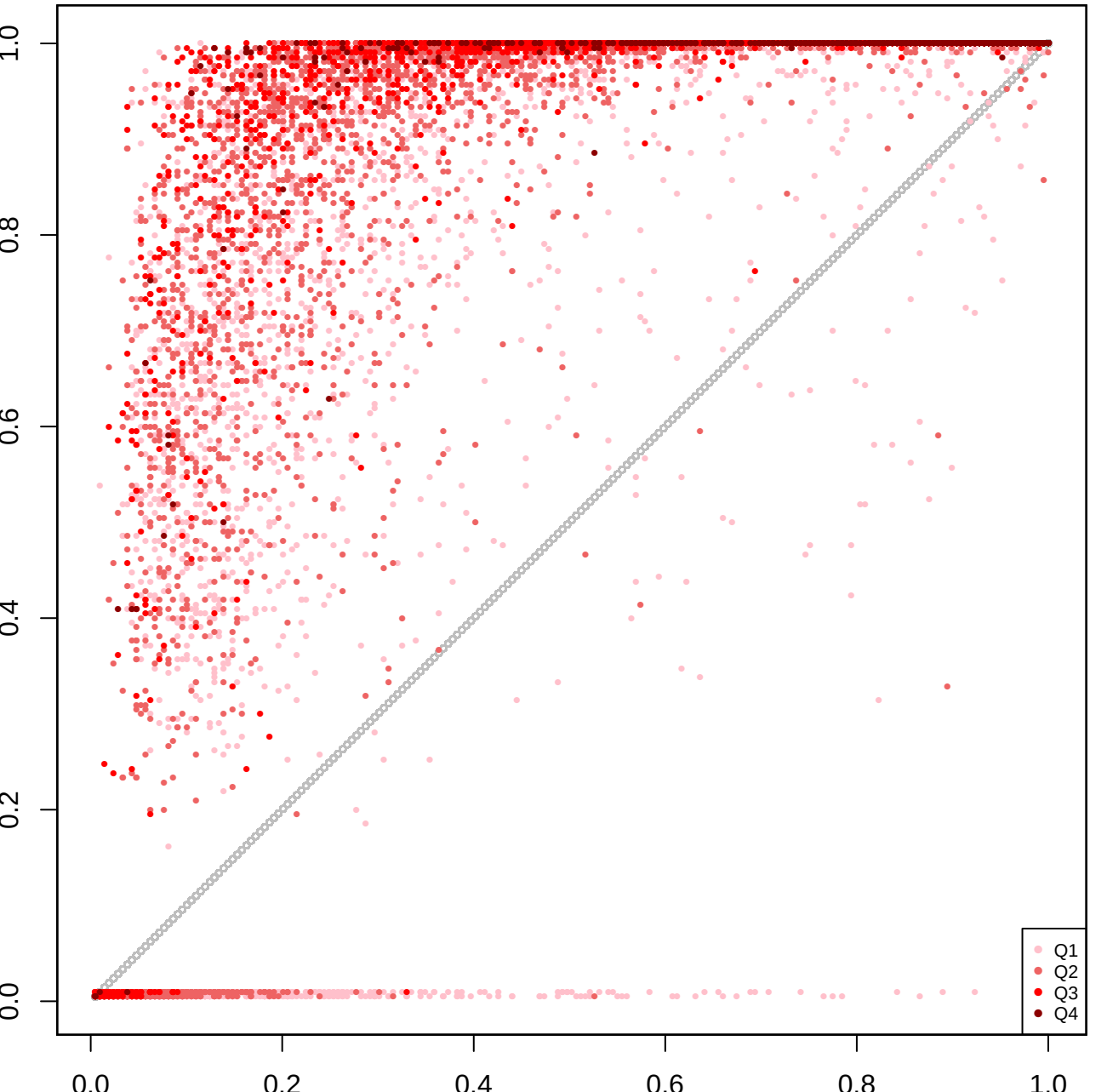
Fig. 2. Heatmap of log-transformed TPM values of marker genes identified for DRG neurons in [7] (left) and t-SNE plot showing the 8 clusters from [7] (right).

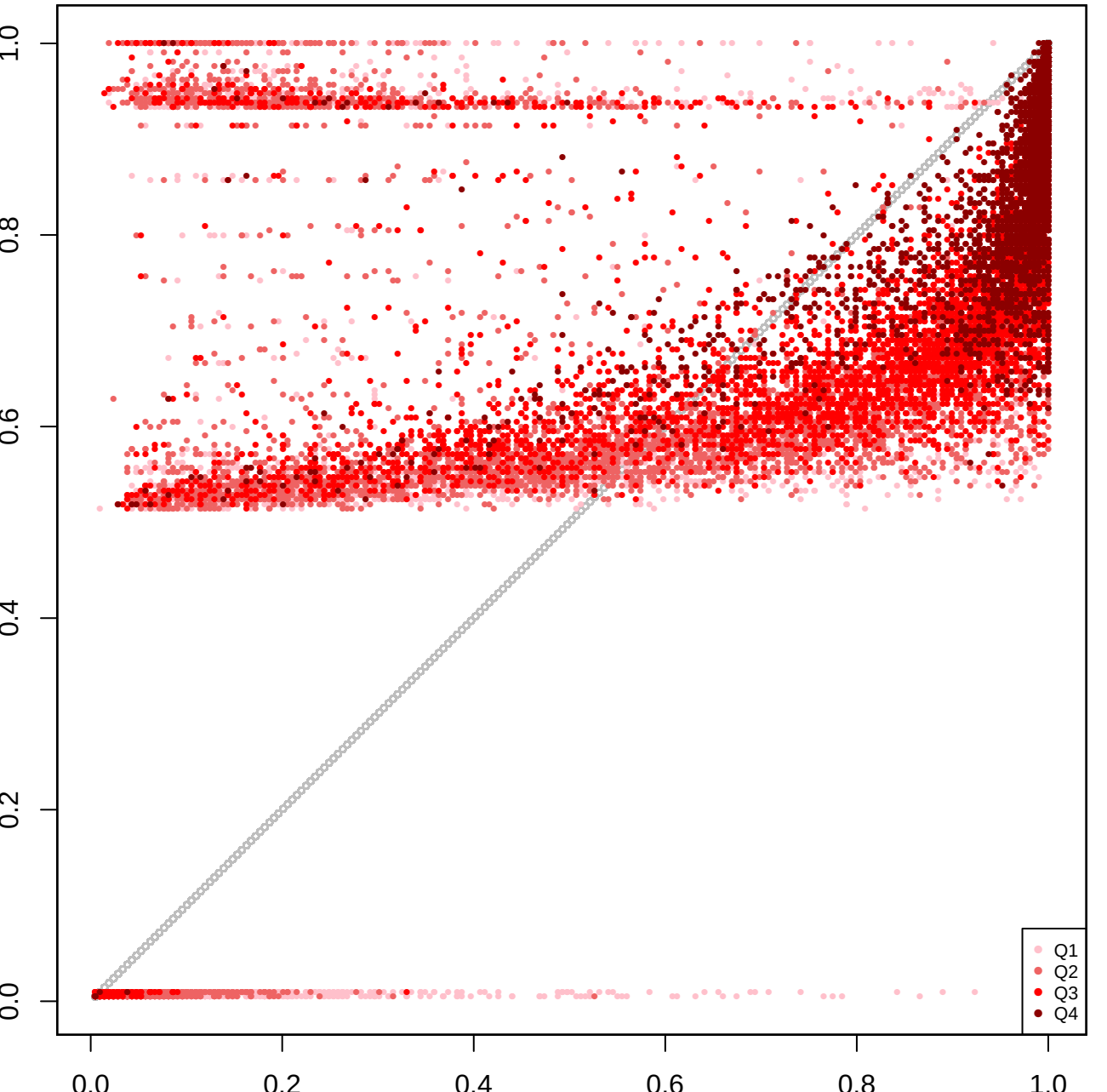

0
2
4
6
8
True Clusters
Gal
Adcyap1
Npy1r
Gpr149
Sstr2
Zcchc12
Cldn9
Nppb
Il31ra
Sst
Nts
Npy2r
Gpr139
Htr1f
Fam19a4
Th
Slc17a8
Zfp521
Fxyd6
Adora2b
Mrgpra3
Rspo1
Ptpn6
Mrgprb4
Cbln1
Agtr1a
Otoa
Mrgprd
Prkcq
Mrgprb5
Lpar3
Lpar5
Nxph1
Tuba8
Wnt7a
Cpne6
S100b
Asic3
Trappc3l
Htr3a
Cgnl1
Pvalb
Baiap2l1
Cadps2
Necab2
Cox6a2
Lypd1
Htr1d
Fam19a2
Clusters
C1
C2
C3
C4
C5
C6
C7
C8

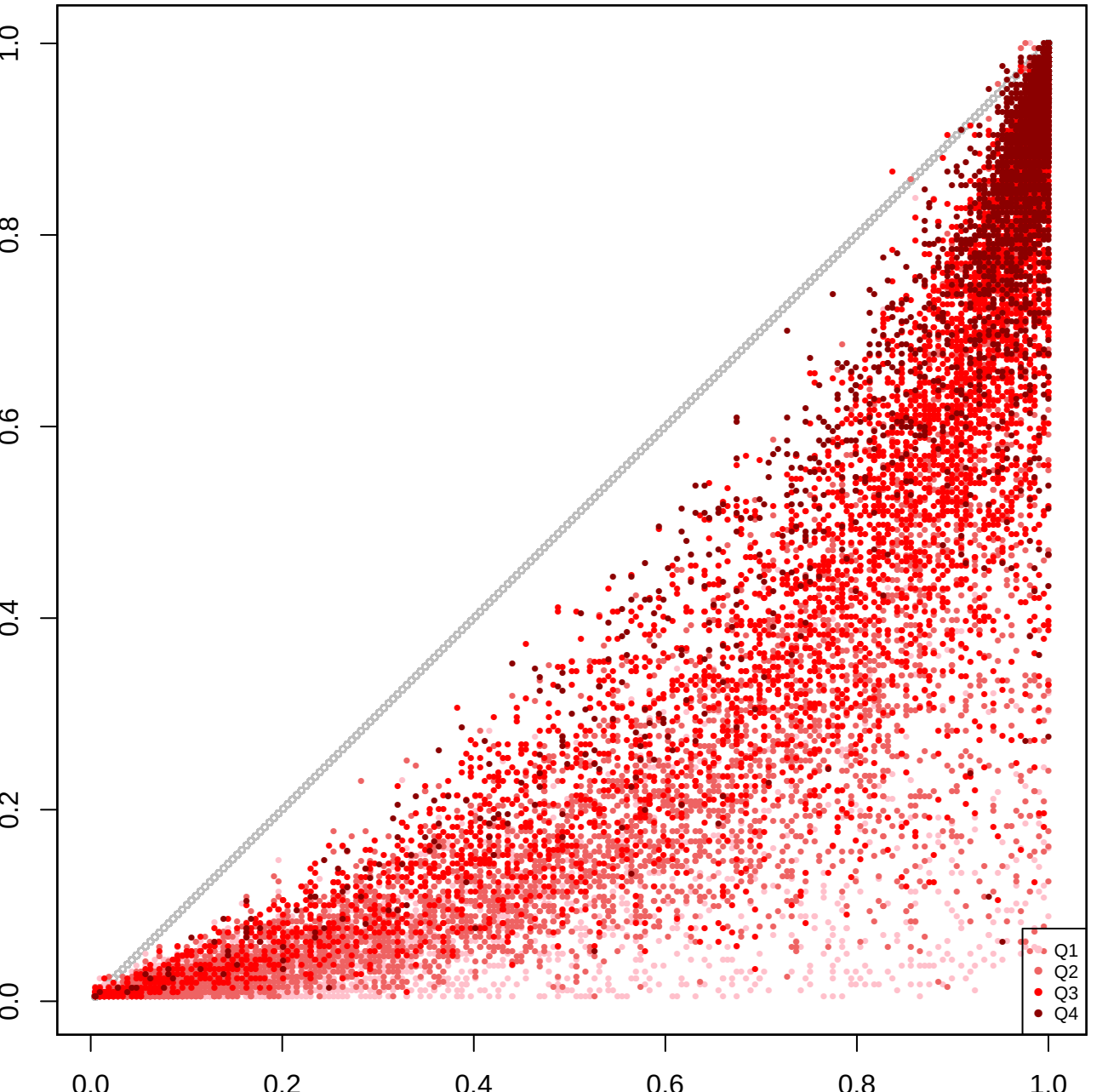

0
2
4
6
8
True Clusters
Gal
Adcyap1
Npy1r
Gpr149
Sstr2
Zcchc12
Cldn9
Nppb
Il31ra
Sst
Nts
Npy2r
Gpr139
Htr1f
Fam19a4
Th
Slc17a8
Zfp521
Fxyd6
Adora2b
Mrgpra3
Rspo1
Ptpn6
Mrgprb4
Cbln1
Agtr1a
Otoa
Mrgprd
Prkcq
Mrgprb5
Lpar3
Lpar5
Nxph1
Tuba8
Wnt7a
Cpne6
S100b
Asic3
Trappc3l
Htr3a
Cgnl1
Pvalb
Baiap2l1
Cadps2
Necab2
Cox6a2
Lypd1
Htr1d
Fam19a2
Clusters
C1
C2
C3
C4
C5
C6
C7
C8

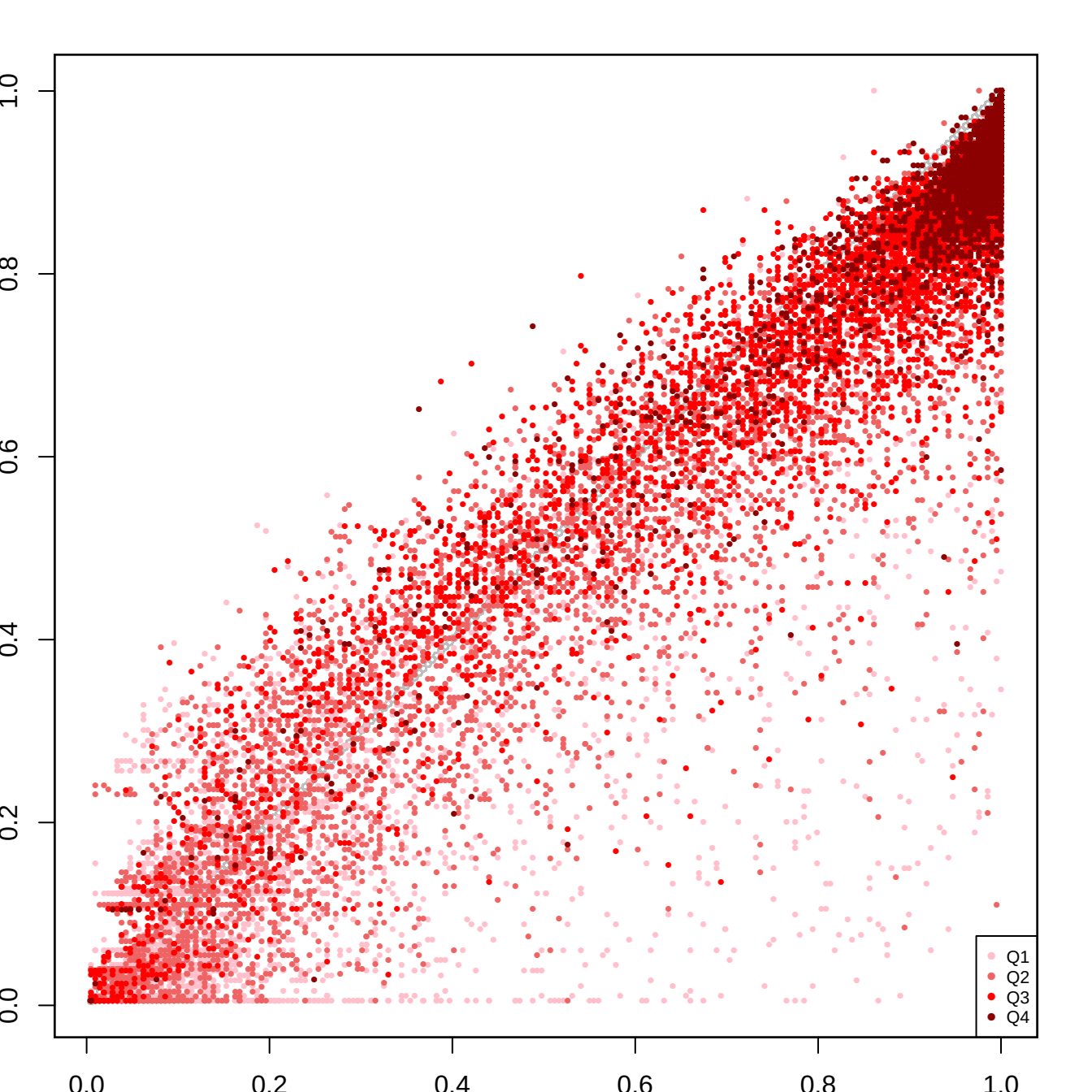

0.0
0.2
0.4
0.6
0.8
1.0
Q1
Q2
Q3
Q4


0.0
0.2
0.4
0.6
0.8
1.0
Q1
Q2
Q3
Q4


0.0
0.2
0.4
0.6
0.8
1.0
Q1
Q2
Q3
Q4


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Q1
Q2
Q3
Q4


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Q1
Q2
Q3
Q4


0.0
0.2
0.4
0.6
0.8
1.0
Q1
Q2
Q3
Q4

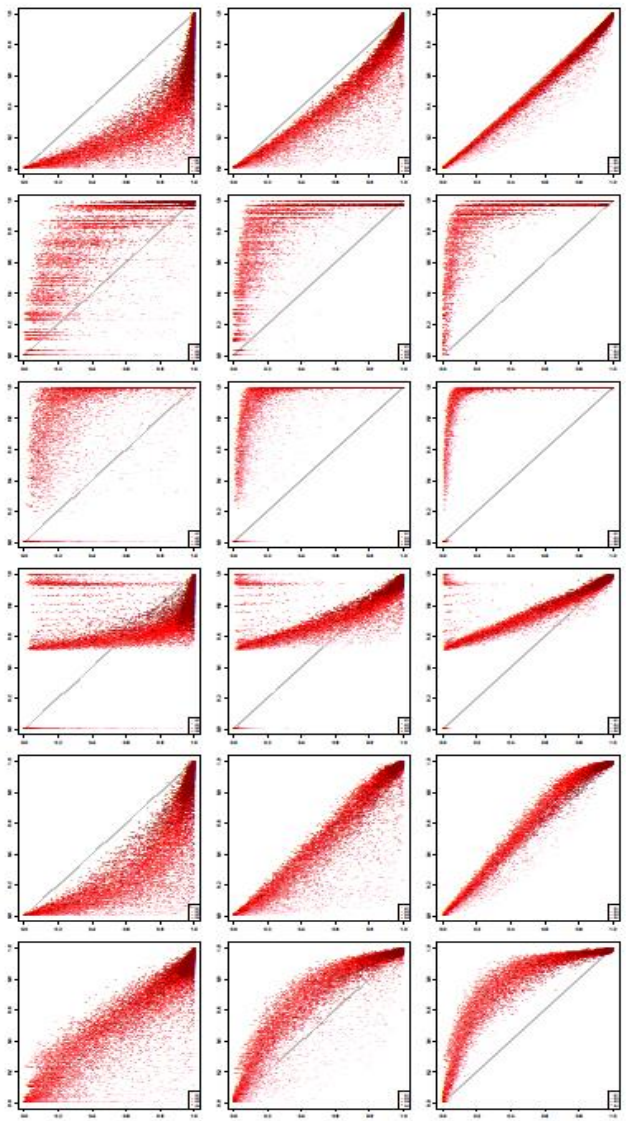


Fig. 3. True vs. imputed detection fractions (columns, left to right: 100K, 1M, 10M read pairs per cell; rows, top to bottom: Raw Data, DrImpute, scImpute, KNNImpute, LSImputeMed, and LSImputeMean).

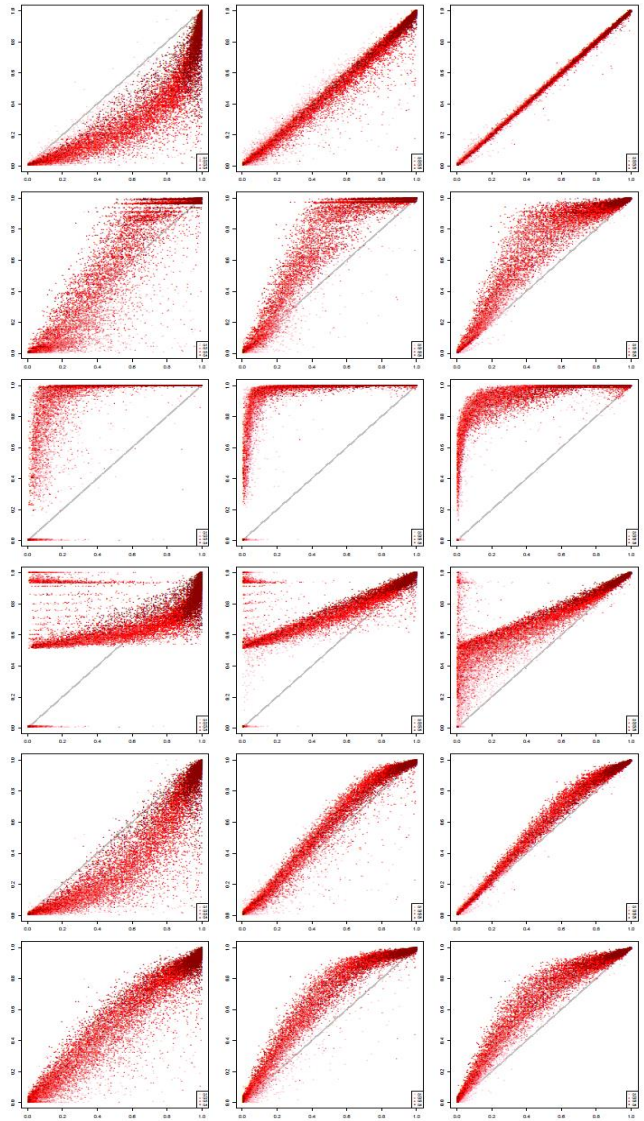
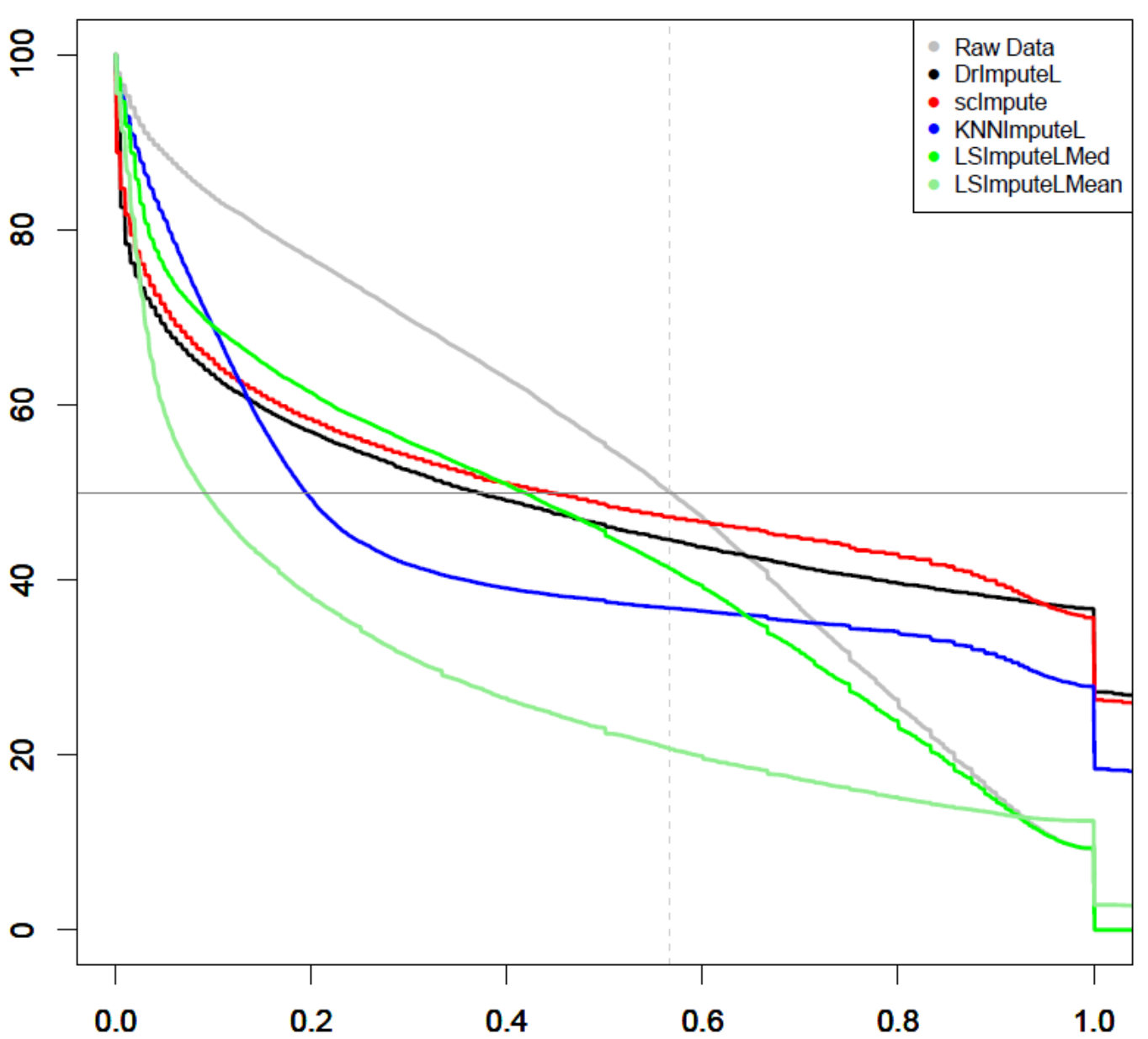
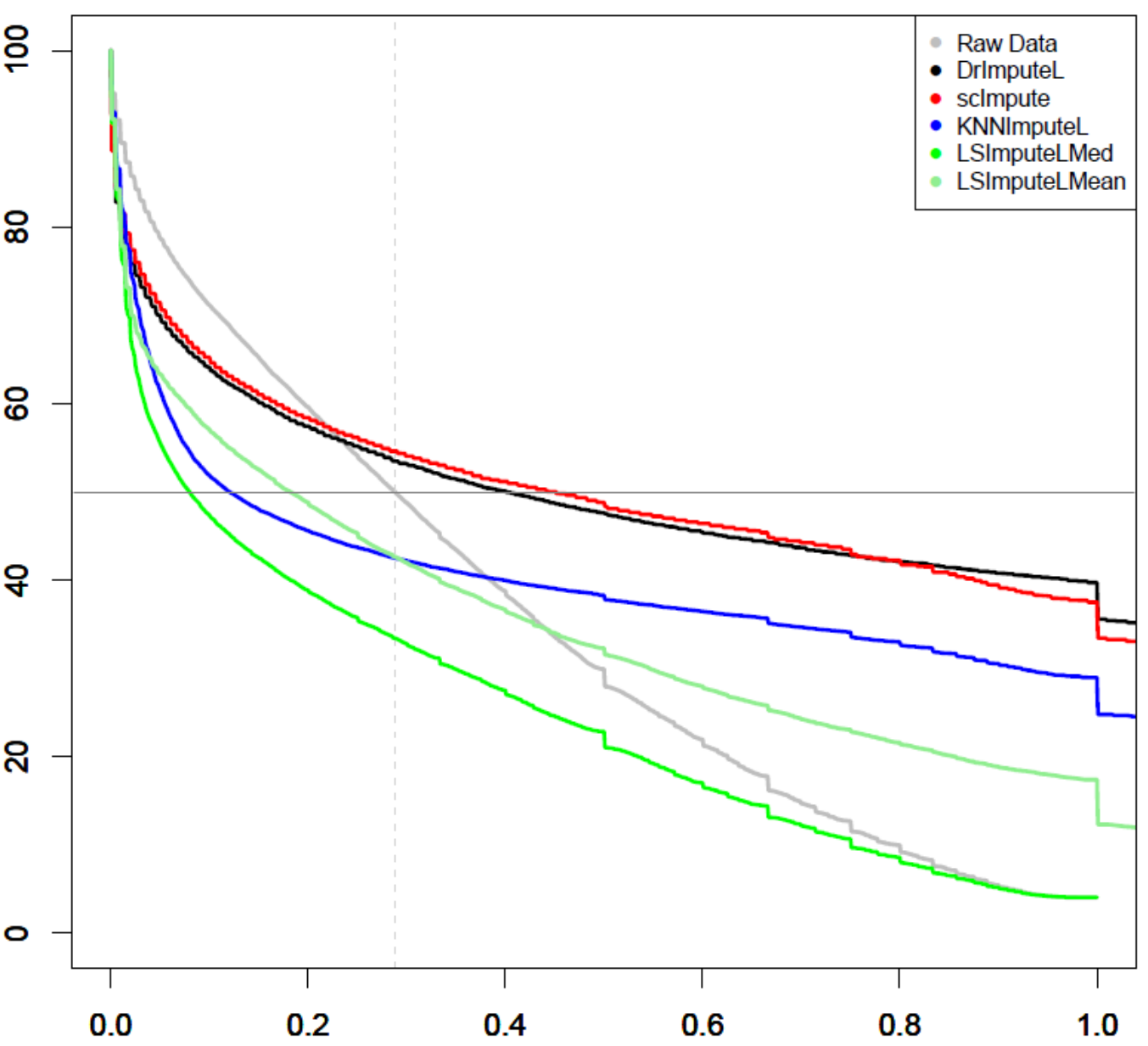


Fig. 1. Rounded True vs. imputed detection fractions (columns, left to right: 100K, 1M, 10M read pairs per cell; rows, top to bottom: Raw Data, DrImpute, scImpute, KNNImpute, LSImputeMed, and LSImputeMean).


Raw Data
DrImputeL
scImpute
KNNImputeL
LSImputeLMed
LSImputeLMean
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0


Raw Data
DrImputeL
scImpute
KNNImputeL
LSImputeLMed
LSImputeLMean
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0

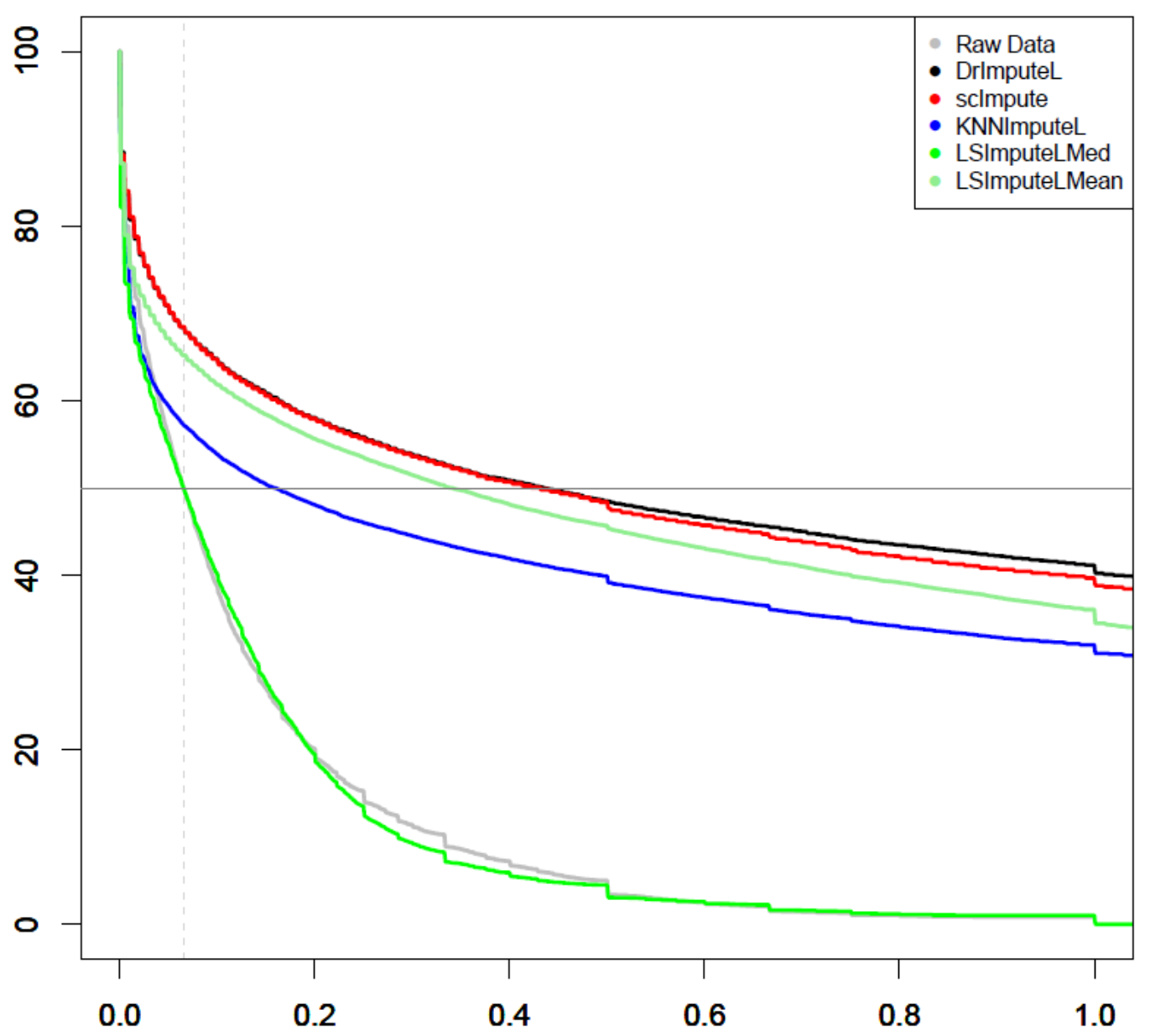
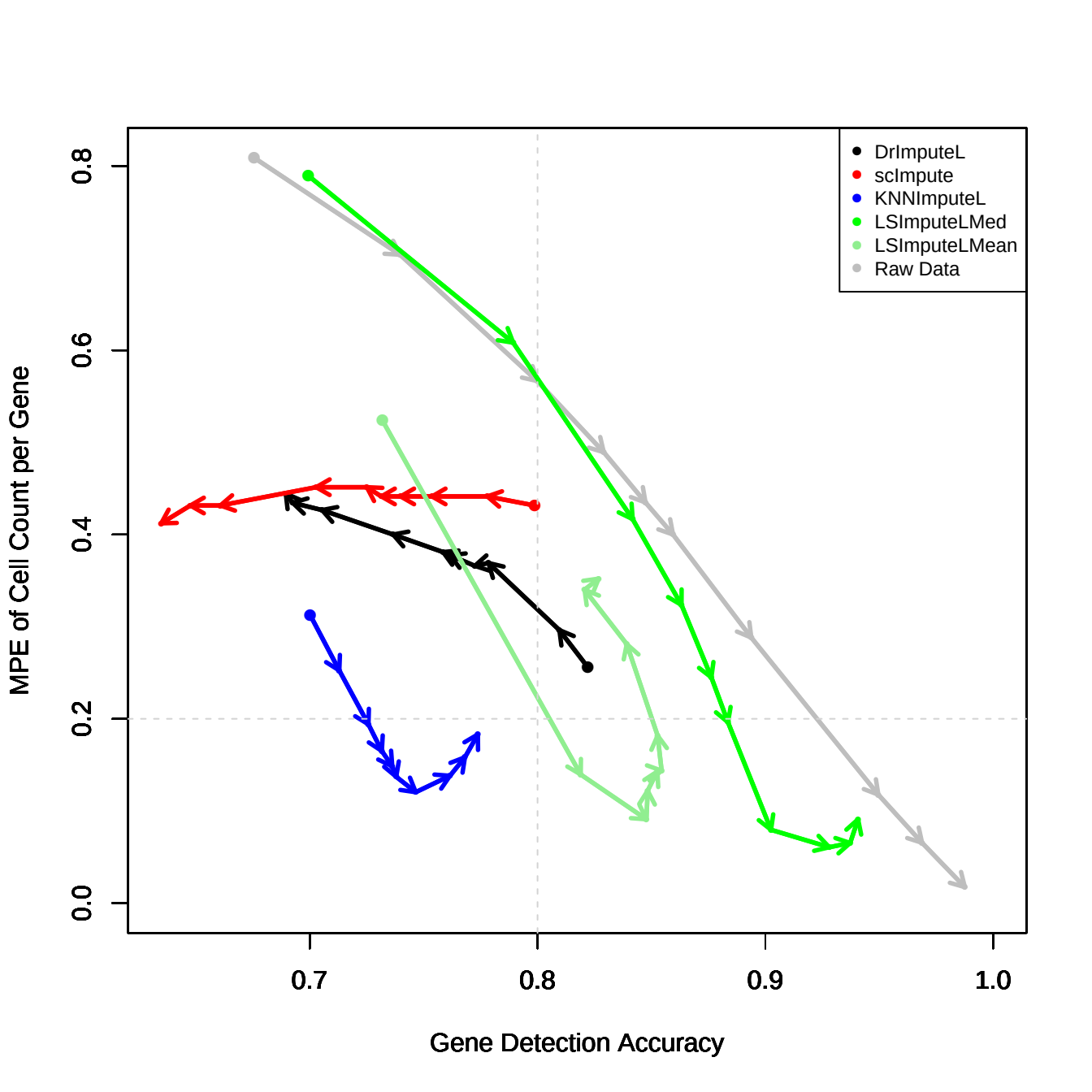
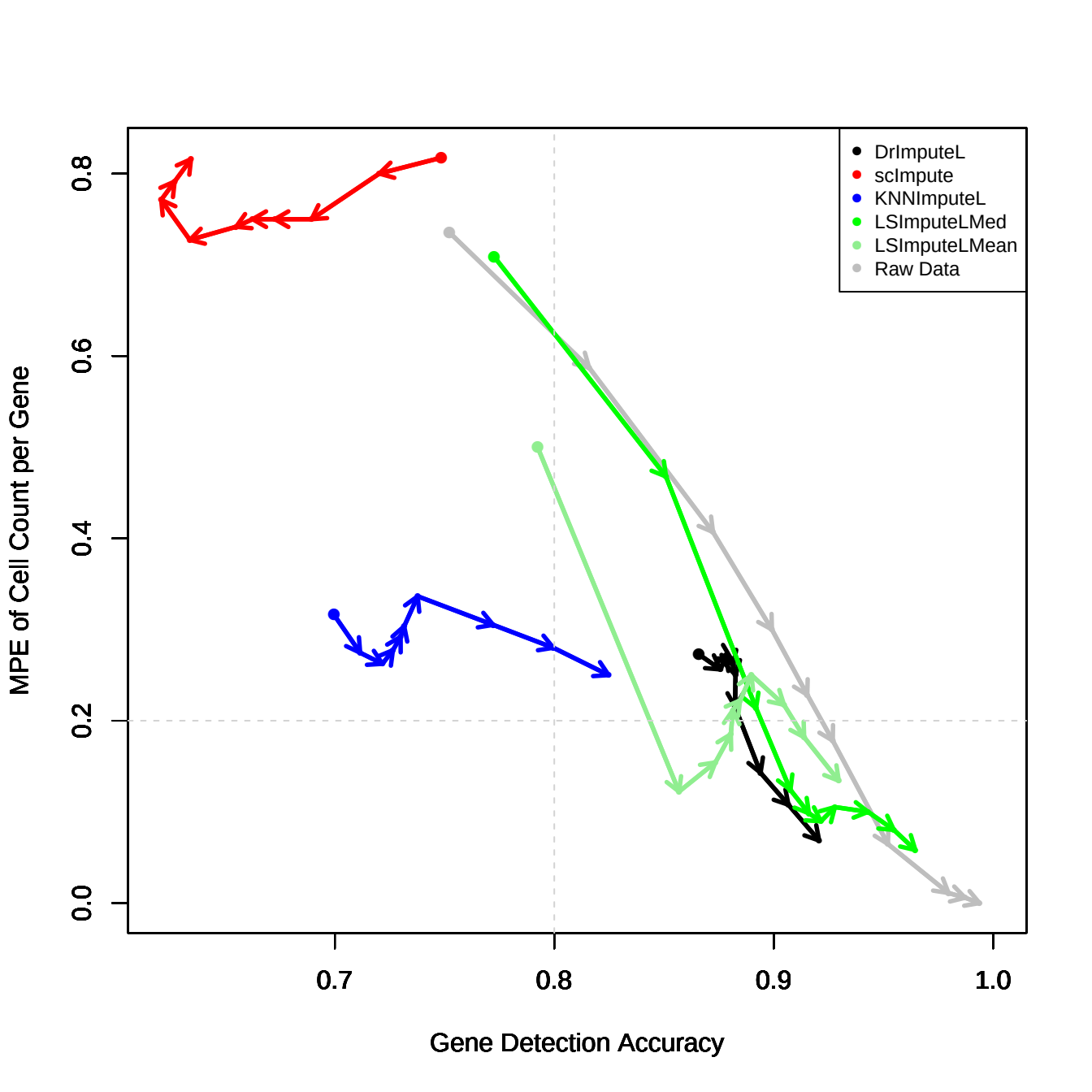

Raw Data
DrImputeL
scImpute
KNNImputeL
LSImputeLMed
LSImputeLMean
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0

Table 1. Gene detection accuracy

Data	Not Rounded						Rounded					
	Raw	Dr.	KNN.	sc.	LSMd	LSMn	Raw	Dr.	KNN.	sc.	LSMd	LSMn
50K	0.676	**0.822**	0.700	0.799	0.699	0.732	0.752	**0.866**	0.748	0.700	0.773	0.792
100K	0.740	0.810	0.713	0.778	0.790	**0.819**	0.816	**0.876**	0.720	0.712	0.851	0.857
200K	0.800	0.778	0.726	0.754	0.842	**0.848**	0.872	0.878	0.689	0.722	**0.892**	0.873
300K	0.830	0.772	0.732	0.740	**0.863**	0.848	0.899	0.880	0.673	0.726	**0.908**	0.880
400K	0.847	0.762	0.736	0.731	**0.876**	0.853	0.915	0.882	0.663	0.730	**0.916**	0.881
500K	0.859	0.759	0.738	0.725	**0.884**	0.854	**0.927**	0.883	0.655	0.732	0.922	0.885
1M	0.894	0.737	0.747	0.703	**0.902**	0.853	**0.952**	0.882	0.634	0.738	0.928	0.890
5M	**0.950**	0.705	0.762	0.661	0.928	0.840	**0.980**	0.894	0.621	0.772	0.943	0.905
10M	**0.969**	0.692	0.768	0.648	0.937	0.821	**0.987**	0.907	0.627	0.800	0.955	0.914
20M	**0.988**	0.690	0.774	0.635	0.941	0.827	**0.994**	0.921	0.634	0.825	0.965	0.930


DrImputeL
scImpute
KNNImputeL
LSImputeLMed
LSImputeLMean
Raw Data
MPE of Cell Count per Gene
0.0
0.2
0.4
0.6
0.8
0.7
0.8
0.9
1.0
Gene Detection Accuracy


DrImputeL
scImpute
KNNImputeL
LSImputeLMed
LSImputeLMean
Raw Data
MPE of Cell Count per Gene
0.0
0.2
0.4
0.6
0.8
0.7
0.8
0.9
1.0
Gene Detection Accuracy

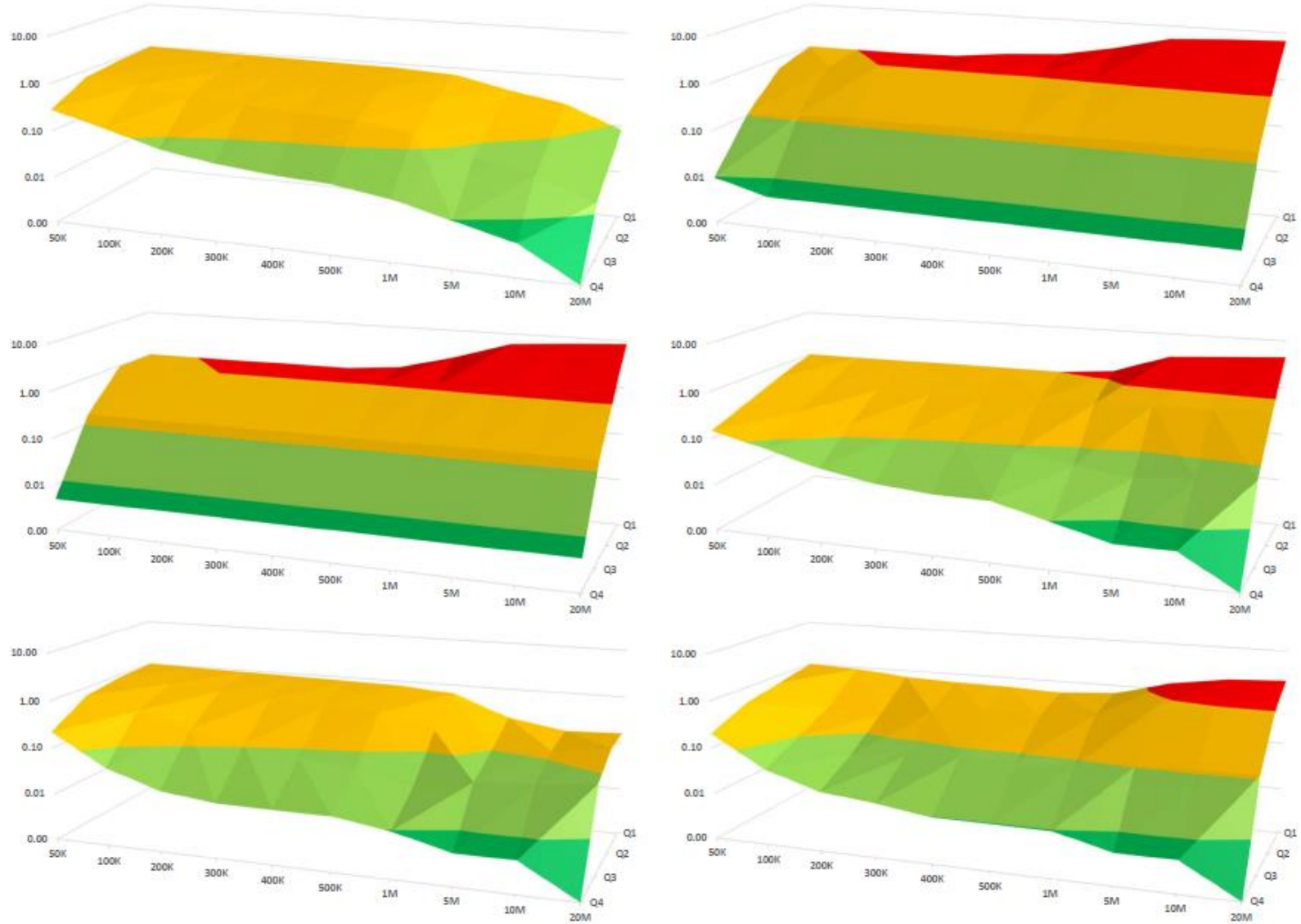


Fig. 5. Surface plots indicating Median Percent Error value in log scale (y-axis) for each depth (x-axis) in each quantile (z-axis) for each method starting clockwise from top left: Raw data, DrImpute, scImpute, KNNImpute, LSImputeMed and LSImputeMean

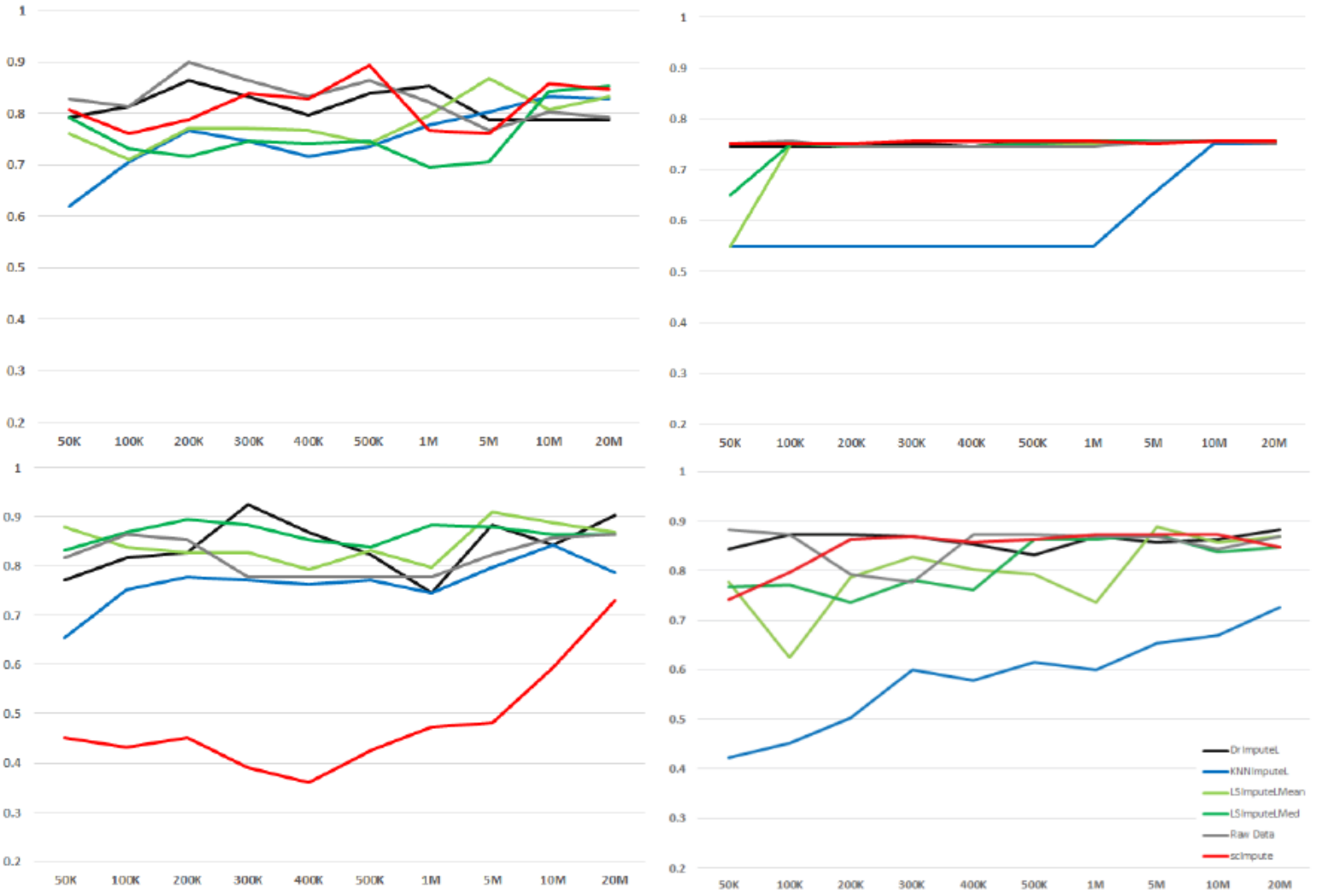


Fig. 7. Clustering Micro Accuracy for clustering algorithms clockwise from top left: PCA based hierarchical clustering using Spearman correlation, Seurat, TF-IDF_Top_C clustering and PCA based spherical k-means clustering