Supplementary Information

Computational analysis of super-resolved in situ sequencing data reveals genes modified by immune-tumor contact events

Michal Danino-Levi[1,2,3], Tal Goldberg[1,2,3], Maya Keter[1], Nikol Akselrod[1], Noa Shprach-Buaron[1,2,3], Modi Safra[1,2,3], Gonen Singer[1,*], Shahar Alon[1,2,3,*]

[1]The Alexander Kofkin Faculty of Engineering

[2]Gonda Multidisciplinary Brain Research Center

[3]Institute of Nanotechnology and Advanced Materials, Bar-Ilan University, Ramat Gan, 5290002, Israel

*Corresponding authors: shahar.alon@biu.ac.il; gonen.singer@biu.ac.il

This file contains:

1) Materials and Methods
2) Supplementary Figures 1-26
3) Supplementary Tables 1-9
4) Supplementary Text: 'Guidelines on how to choose parameters for InSituSeg and fine tune them'
5) References

Materials and Methods

Description of the datasets

Biopsies were collected from patients at Dana Farber Cancer Institute and originally described in (Alon *et al.* 2021). Prior to any study procedures, the patients provide written informed consent for a research biopsy and subsequent analysis of tumor and normal samples, as approved by the Dana-Farber/Harvard Cancer Center Institutional Review Board (DF/HCC Protocol 05-246). The sample utilized in this study was an 18-gauge core needle biopsy (~6x0.8 mm) of a liver metastasis obtained from a 66-year-old woman with a known diagnosis of hormone receptor positive metastatic breast cancer. The region sequenced in situ with ExSeq was 1347 x 621 x 8 microns in size (before expansion). After 3.3x physical expansion of the tissue, it was imaged using 76 fields of view (FOV) of 40x objective, with a resolution of 0.17 microns in X&Y and 0.4 microns in Z (post expansion). In this biopsy, 297 genes were interrogated. These genes were chosen to represent various aspects of breast cancer biology, metastasis, and the tumor-immune-microenvironment, as well as cell types and programs discovered from the single cell and single nucleus RNAseq data (Alon *et al.* 2021). The list of genes included: non-tumor marker genes (including T cells and Macrophage marker genes)- *CD3G*, *CD68*, *FOXP3*, *CD4*, *CD8A*, *CD3D*, *CD3E*, *HLA-DRA*; tumor marker genes- *EGFR*, *GRB7*, *ERBB2*, *PGR*, *CD44*, *CD24*, *ALDH1A3*, *EPCAM*, *KRT19*, *KRT18*, *CDH1*; B cells marker genes- *IGHG1*, *IGHG4*, *IGKC*, *IGHM*; Fibroblast marker genes- *HSPG2*, *SULF1*.

Segmentation of cell bodies

We developed a new segmentation pipeline, termed InSituSeg, that takes advantage of the dense mapping of genes in situ for segmentation of cell bodies in 3D, using staining of cell nuclei and RNA locations (Fig. 2A). The steps of the segmentation pipeline are as follows (Fig. 2B):

A. **Illumination correction.** The input for the segmentation code is the 3D image of DAPI nuclear staining, in addition to the spatial locations of the mRNA molecules in that image which was utilized only at the end of the code. Each field of view (FOV) of the DAPI image was divided into 3x3 sub-fields to account for unequal illumination inside a given FOV. While this step accounts for different illumination, it can create gaps in the homogeneity of the segmentation at the sub-fields borders. Therefore, this step is

not mandatory in the pipeline and should be avoided when the illumination is even in a given FOV.

B. **Detection of nuclei pixels.** The first step is to find the locations of the nuclei pixels, using the observation that the strongest pixels in a DAPI-stained image are in the nucleus. This is done with simple pixel thresholding, collecting the strongest pixels, sorted by intensities. This parameter is initially set to 5%, and can be manually adjusted. Using visual inspection, we determined that setting this parameter to 2-5% for the nucleus is suitable for almost all FOVs (Table S2).

C. **Refine nuclei voxels.** Next, the code corrects and refines the nuclei voxels. This procedure starts with a pre-processing denoising step using a median filter, by replacing, for each pixel, its intensity value with the median value of the surrounding pixels, taking a voxel of 9 pixels by 9 pixels by 9 pixels. Then the pictures are converted into binary images, and the holes were filled via morphological operations performed on each z-plane separately, using the 'imclose' function (Matlab) followed by the hole filling function 'imfill' (Matlab), and then some isolated pixels were removed using the 'imopen' function (Matlab). Then, objects which are smaller than the size of the nucleus but are bigger than pixel-size (i.e. large scale noise), were removed by connected component analysis (He *et al.* 2014). The connected components were sorted by the number of voxels, and the bottom 50% were removed using the Matlab functions 'bwconncomp', 'regionprops' and 'bwareaopen'. Again, the threshold was adjusted manually and visual inspection determined that 40-60% is suitable for almost all FOVs (Table S2).

D. **Split large nuclei.** Next, the code detects large nuclei objects, which might in fact be two or more individual nuclei combined, and makes an attempt to split them. This is done by first sorting the nuclei objects by voxel size and detecting the top 20% of objects. This size threshold was manually adjusted to a value of 10-25% and was found fitting for most FOVs (Table S2). Focusing on the large objects, the code makes an attempt to split them using the following assumption – if the large object indeed consists of several nuclei, then it should be possible to detect borders, within the large object, that divide the large object into smaller ones. These border regions are expected to have low intensity ('drop off') compared to the rest of the object pixels. Thus, by iteratively increasing the threshold for nuclei pixel detection in step (B), and repeating the nuclei refinement procedure in step (C), the nuclei can be split; Specifically, instead of picking the top 5% of all pixels as belonging to nuclei in step (B), we increase the detection

threshold (thus selecting fewer pixels as belonging to the nuclei) to 4.5% of all pixels, and then 4% and so on, until a cutoff threshold value of 0.5% was reached, and for each threshold step (C) was repeated (Table S2). When a nucleus was split into two smaller nuclei in one of the iterative steps, the resulting nuclei were examined again to determine if their size is above the size threshold. If their size was below the size threshold, the iterative procedure was terminated for these nuclei. Otherwise, the iterative procedure continued for all large nuclei until a cutoff threshold value was reached.

E. **Detect cell body voxels using watershed segmentation.** Here, the cell nuclei, detected in the previous steps, serve as seeds for watershed segmentation (Kowal *et al.* 2020). The 'watershed' function (Matlab) receives two inputs: the first is the nuclei objects as described above, and the second is all the cell body pixels. The cell body pixels are calculated in a similar manner to that of the nuclei pixels (step (B) above), i.e. with a simple pixel thresholding that allows collecting strong pixels, sorted by intensities. We used an initial cutoff of the top 15% of the pixels, again sorted by intensities, and visual inspection determined that setting this parameter to 10-20% is suitable for detecting cell body pixels for almost all FOVs (Table S2). For adjacent cells, we validated that the cell boundaries, after the watershed segmentation, indeed correspond to a clear drop off in the DAPI signal between the cells (Fig. S2).

F. **Assign mRNA molecules into cell bodies.** In this step, the spatial locations of the mRNA molecules in that image are overlaid on the cell bodies detected in the previous step. For each cell, the mRNA molecules that are within the given cell body object are assigned to that cell. Next, for mRNA molecules that are outside all the cell body objects, an attempt is made to assign them to nearby cells. This is done in the following way: (i) for each mRNA molecule location, its distance to the surface of each neighboring cell body is calculated using the function 'dsearchn' (Matlab). To be considered further, the mRNA molecule has to be within 0.6 microns of the surface of the nearest cell body (this distance is termed 'R1'). (ii) To avoid wrong assignments, we mark the distance of the mRNA molecule to the surface of the second nearest cell body (this distance is termed 'R2'). Only if the difference between these distances (R2-R1) was larger than 1.2 microns, the mRNA molecule was assigned to the closest cell body. The cutoff numbers were calculated via visual inspection (Fig. S3). We directly tested the effect of the RNA reassignment step (step F), and by-and-large the identity of proximity-induced genes is the same with and without the RNA which are outside

cell bodies (Fig. S4). For example, for T cells which are proximal to tumor cells, 15 proximity-induced genes are detected without these RNA molecules, out of 17 genes overall. This is consistent with the fact that 95.3% (895,510 out of 939,764) of the sequenced RNA molecules in this sample are inside the cell bodies as detected via InSituSeg, and therefore the effect of the RNA reassignment step is minor. Finally, we note that in some cases, a z-position shift can occur between the mRNA molecule's location file and the raw DAPI images. This is a result of the ExSeq image processing that generated the original mRNA molecule's location file. In cases when the shift occurs, it is corrected by adding a fixed z-value to the mRNA molecule's location.

Sensitivity to human variability in choosing InSituSeg parameters

To examine how significant is the human variability factor in the fine tuning procedure of InSituSeg, we compared the segmentation results when ExSeq was run by two individuals using the same two randomly chosen fields of view (Fig. S6). To quantitatively measure the agreement between the segmentation of the two individuals, we measured the overlap between the segmented cell bodies as described in Methods section 'Comparison between InSituSeg segmentation and cytosolic with membrane stain segmentation' (Fig. S9), but note that here the calculation was performed in 3D, due to the thickness of the ExSeq samples. In addition, we also measured the differences in the parameter set that the two individuals used (Fig. S6).

Cell segmentation using mRNA locations alone (Fig. S7C)

To obtain a 'cell'-like segmentation using mRNA locations alone, the locations of the mRNA molecules were clustered in space utilizing a normal mixture modeling algorithm using the 'mclust' R package with the EEV (Ellipsoidal, Equal Volume and shape) model. The number of clusters for each FOV was chosen according to the Bayesian Information Criterion (BIC) score. Clusters with a number of mRNA molecules below the median, computed using all of the clusters, were considered background, i.e., non-cells, and therefore were removed.

Comparison between InSituSeg segmentation and cytosolic with membrane stain segmentation

To directly quantify the ability of InSituSeg to identify cell bodies, we compared the segmentation obtained with InSituSeg to segmentation based on cytosolic and membrane

staining. For this, we downloaded publicly available MERFISH data from VIZGEN website of 'mouse liver map data set', and specifically 'Mouse Liver 1 Slice 1': https://info.vizgen.com/mouse-liver-data?submissionGuid=d4ca3624-04dd-4865-b72f-f4b0b41cec71. This data contained the locations of 347 genes inside a mouse liver tissue section, in addition to both cytosolic and membrane stains, as well as DAPI stain. The data was downloaded after selecting channels 'Cellbound2', 'Cellbound3' and 'DAPI'. After downloading the MERFISH data, two FOVs were randomly selected with a size of 100x100 microns, to match the FOV size of the ExSeq datasets (before expansion). Next, the segmentation tool Cellpose (Stringer et al. 2021) was utilized to segment the cells in two ways: given the DAPI signal alone, and the DAPI signal with the cytosolic and membrane staining. Cellpose was used with default settings and the 'cyto2' option. Next, the same cells were segmented using InSituSeg given the DAPI signal alone as input. Then we performed two comparisons: 1) between the segmentation of the cells using Cellpose given the DAPI signal and the cytosolic and membrane staining, to the segmentation of the cells using InSituSeg given the DAPI signal alone (Fig. S9, top row). 2) between the segmentation of the cells using Cellpose given the DAPI signal and the cytosolic and membrane staining to the segmentation of the cells using Cellpose given the DAPI signal alone (Fig. S9, bottom row). The comparisons between the segmentation schemes were done in 2D (i.e., one z section) since the MERFISH data was 2D in nature; only 7 z sections were present in the MERFISH data, from which only one main z section contained more than a few localized transcripts. We compared the area of the 'matching' cells, i.e., cells that were identified using both segmentations (Fig. S9, same color palette in left and middle plots). Note that the definition of matching cells depends on an overlap area cutoff (Fig. S9, right plot). Accordingly, the average overlap area between cells in the two segmentations depends on the overlap area cutoff (Fig. S9, right plot, red line). A tradeoff exists between the number of matching cells and the overlap area cutoff (Fig. S9, right plot, blue line). We choose a working point (Fig. S9, right plot, green circle, and pink dashed line) to maximize the number of matching cells. These comparisons allow us to directly quantify the ability of InSituSeg to identify cell bodies.

Clustering segmented cells

In order to identify and cluster the segmented cells according to their expression pattern, we utilized the R toolkit Seurat (Hao *et al.* 2021), and followed the analysis in (Alon *et al.* 2021). Briefly, we used a supervised approach of using selected genes for dimension reduction (Becht

et al. 2018). These genes were described in the Methods section 'Description of the datasets'. Then, Seurat's principal component analysis (PCA)-based expression clustering was performed. Genes with an expression level that is higher in a given cluster compared to the other clusters were used to mark a given cluster. The Seurat 'FindAllMarkers' function was utilized to detect putative gene markers for each cluster; for putative gene markers with p-values less than 1e-10 (as determined by 'FindAllMarkers'), we assigned the cluster with the known cell type annotation of the marker gene, otherwise, we marked the cluster as "unknown". The cell's clustering was displayed using the Uniform manifold approximation and projection (UMAP) representation (Becht *et al.* 2018) (Fig. S10C).

Detecting differentially expressed genes

For any pair of cell clusters X and Y, cluster X was partitioned into two subsets: a subset of X cells that are proximal to Y cells, and a subset of X cells that are not proximal to Y cells. Physical proximity was measured using the smallest Euclidean distance between the mRNA molecules (sequencing reads) in two adjacent cells, using the python function 'cdist'. Computing distances between mRNA molecules is possible because of the improved cell body segmentation, and adds stringency compared to computing the smallest distances between the voxels of two adjacent cell bodies. We set a threshold of 3 microns (before expansion) for that distance. Here we make the simple assumption that if the cells are far apart then physical interaction did not occur between them. The cutoff distance between two adjacent cells, i.e., the minimal distance between the two cells' boundaries that below it the cells are considered adjacent, is an important parameter in the analysis of proximity-induced genes. Importantly, we confirmed that the results obtained are robust to the threshold value (Fig. S11-12), by examining the sensitivity to threshold values 4, 2, 1, and 0.5 microns (before expansion). The small distance cutoff of 1 and 0.5 microns increases the likelihood that the adjacent immune and tumor cells are physically touching. The super-resolution of ExSeq made it possible to measure these distances. To detect differentially expressed genes that are triggered by proximity between immune and tumor cells with the different distance cutoffs, we calculated gene expression change (fold change) and p-value per gene using DESeq2 with permutation analysis (as detailed below). We compared the proximity-induced genes detected via differential expression analysis when using either 3 microns or 4, 2, 1 and 0.5 microns as the distance cutoff between a non-tumor cell and a tumor cell (Fig. S11-12, Table S3). Furthermore, we compared the number of non-tumor cells detected as adjacent to tumor cells

when using either 3 microns or 1 and 0.5 microns (Fig. S12). Note that the 3 microns distance cutoff for cell-cell proximity is utilized throughout the paper unless otherwise mentioned, since it captures more adjacent cells and therefore improves statistical power.

We tested each one of the 7 non-tumor cell clusters that correspond to B cells (2 subtypes), T cells (3 subtypes), macrophage (1 subtype) and fibroblast (1 subtype), against each one of the 5 cell clusters that correspond to tumor cell types (Fig. S10C). In addition, merged non-tumor subtypes, i.e., the two B cell, and T cell subtypes combined, were tested against each one of the 5 tumor subtypes and against the tumor subtypes combined. Each of the 7 non-tumor cell clusters was also tested against the combined tumor subtypes. In total, this amounts to 54 comparisons performed to observe gene expression differences in non-tumor cell types when in proximity to tumor cell types. In a similar manner, 54 reciprocal comparisons were performed to observe differences in tumor cell types when in proximity to non-tumor cell types. So overall 108 comparisons were performed.

Gene expression change (fold change) and p-value per gene in each comparison were calculated using DESeq2 (Love, Huber, and Anders 2014), and we proceeded with genes that had Benjamini-Hochberg false discovery rate (FDR) of 0.1. To further assess the statistical significance of the results, the X cells were randomly partitioned into two subsets, of the same size as the two original subsets, and the fold changes and the p-values of all genes between the two random subsets were recalculated using DESeq2. The random partition procedure was repeated 1000 times. From these random realizations, bootstrap p-values were calculated for each gene in each comparison as (1+x)/1000 where x is the number of realizations in which the p-value was below that of the original detected gene. For each comparison, only genes that had both FDR less than 0.1 (using the DESeq2 p-values) and whose bootstrap p-value was less than 0.05 were considered differentially expressed genes. The genes were divided into upregulated genes with positive fold change, i.e. genes with expression levels increase as a result of proximity between two cell types, and downregulated genes with negative fold change, i.e. genes whose expression levels decrease as a result of this proximity (Fig. S14). To avoid errors that result from inaccurate boundaries detection of two adjacent cells, we filtered genes in two different ways (Table S6): 1) We filtered upregulated genes detected in X cells if they are known cell markers for the Y cells (the known marker genes are listed in the Methods section ‘Description of the datasets’). 2) We filtered genes detected in X cells (i.e., induced in the subset of X cells that are proximal to Y cells compared to the subset of X cells which are not

proximal) if they are highly differentially expressed in the Y cells (i.e., induced in the subset of Y cells that are proximal to X cells compared to the subset of Y cells which are not proximal). Highly differentially expressed was defined as having p-values lower than 1e-5. The genes filtered by each filtering method are given in Table S6. Overall, after the filtering based on canonical markers, 93% and 95% of the proximity-induced genes were retained, when non-tumor cells are proximal to tumor cells and vice versa, respectively (Table S6). With the filtering of the most differentially expressed genes, 94% and 98% of the proximity-induced genes were retained, when non-tumor cells are proximal to tumor cells and vice versa, respectively (Table S6). Importantly, when comparing the lists of proximity-induced genes after the two filtering methods, 94% and 97% of the proximity-induced genes are the same, when non-tumor cells are proximal to tumor cells and vice versa, respectively (Table S6). Thus, a high degree of overlap exists between the two different filtering methods: based on gene markers and based on highly differentially expressed genes in the other direction of proximity. This increases our confidence that after the filtering steps only a small number of proximity-induced genes are a result of mis-segmentation.

Estimation of the effect of the physical expansion on the number of resolved transcripts and proximity-induced genes

Detecting neighboring RNA molecules when exploring a large number of genes in a tissue is challenging. Physical expansion can address some of this challenge. We estimate that without expansion the signal from nearby transcripts is likely to overlap in space and therefore can't be resolved (Fig. S13). This estimation was done by considering three factors: 1) The physical size of the rolonies (i.e., the padlocks which bind single molecule RNA, after amplification) which is 300nm in diameter (Alon et al., Science, 2021). 2) The point spread function (PSF) of the imaging setup utilized for in situ sequencing of the samples (Fig. S13A,B); beads with 100nm diameter result with PSF that has full-width at half maximum (FWHM) of 270nm (Fig. S13A,B). 3) The physical expansion factor which is 3.3.

The calculation was done as follows: The diameter of the rolonies (yellow circles in Fig S13C,D) multiplied by a factor of 2.7 (FWHM of 270nm divided by 100nm beads) gives a final diameter of 810nm. Therefore, the rolonies should be at least 0.81 microns apart from each other to avoid overlap. In contrast, without expansion the rolonies should be at least 2.7 microns apart from each other to avoid overlap.

An example of this benefit of expansion is shown in Fig. S13C,D. The physical locations of the identified transcripts (yellow circles) in a randomly selected region of a sample analyzed with ExSeq (Figure 4B) are shown in Fig. S13C, and the estimated physical locations of the identified transcripts in the same region, but without the physical expansion, are shown in Fig. S13D. The physical size of the rolonies (yellow circles) remains the same, but the number of pixels in X and Y was reduced by the expansion factor of 3.3, resulting in likely overlapping rolonies (overlapping yellow circles). This analysis reveals that when the expansion factor is artificially removed, the majority of rolonies will likely not be detected due to overlap: only 77,906 out of 939,764 sequenced RNA remain. Next, we applied differentially expressed analysis as described in the Methods section 'Detecting differentially expressed genes', using the 77,906 non-overlapping transcripts, and revealed that this dramatic decrease in resolved trasncripts is also reflected in the number of proximity-induced genes detected (Fig. S13E). For example, only 2 genes are detected as proximity-induced genes for T cells close to tumor cells, instead of 17 in the original analysis (Fig. S13E).

Pre-processing for the machine learning pipeline

We applied machine learning tools to detect genes that their expressions separate, for cell type X, cells that are proximal to cell type Y versus non proximal cells. Specifically, we performed 108 comparisons between non-tumor cell types and tumor cell types as described in the Methods section 'Detecting differentially expressed genes' above. In contrast to the detection of differentially expressed genes described above, machine learning tools can detect genes that change their expression in concert due to the proximity between cells.

We applied a machine learning pipeline on each one of the 108 comparisons (Fig. S16). In every comparison, a boolean value was assigned to each cell of type X, representing whether or not it was in physical proximity with cells from type Y. Physical proximity was measured as in the Methods section 'Detecting differentially expressed genes'. We set the Boolean value to 'false' if the distance between the cells is larger than the physical distance threshold, or to 'true' otherwise. As in the Methods section 'Detecting differentially expressed genes', we confirmed that the results obtained are robust to the threshold value by examining the sensitivity to two other threshold values (Fig. S17). The Boolean value was the target class, i.e. the feature for which we wanted to gain a deeper understanding. Accordingly, we used supervised machine learning algorithms to uncover relationships between the gene expression

profile in each cell to the target class of the cell. The expression profile in each cell is the number of sequencing reads for each one of the 297 genes studied.

Machine learning pipeline

The dataset of each comparison was randomly split into training and testing sets with a ratio of 8:2, respectively. The split was stratified so the class distribution was retained. The testing set was not used during the training phase and was only used at the end to evaluate how well the model is generalized to unseen data.

In most of the comparisons, the majority of cells in the dataset were not in close proximity to cells from a different cell type. Therefore, the dataset was imbalanced, i.e., the 'false' class label had a high number of observations compared to the 'true' label. To adjust the class distribution, we utilized two over-sampling methods, Synthetic Minority Over-sampling Technique (SMOTE) (Chawla *et al.* 2002) and Random Over Sampler (ROS) (Batista *et al.* 2004) (see below). On the training set, we applied the stratified k-fold cross validation strategy (Tan and Gilbert 2003). In k-fold cross validation, the training set is equally divided into k different subsets, such that k rounds of validation are performed each with k-1 subsets used as a training set, while each instance in the data set appears only once as part of the validation data. The use of a stratified k-fold ensured that the relative distributions of the target class were considered when generating the subsets. The value of k was between 3 to 6, depending on the number of cells with the minority label in each fold in a given comparison (as mentioned above, the minority label was in most cases 'true'); the minimal number of k was set to 3, and in cases were the number of cells in each fold in the minority label was >20, we increased the k up to k=6.

Overall four machine learning classifiers were applied on the dataset: Decision Trees (Quinlan 1986), Random Forest (Ho 2002), XGBoost (Chen and Guestrin 2016) and CatBoost (Dorogush, Ershov, and Gulin 2018) (Fig. S16). Decision Tree is a classifier with a high level of interpretability, which is important since our aim is to find genes that change their expression due to the proximity between cells. Random Forest, XGBoost and CatBoost are all based on Decision Trees models, however with a low level of interpretability. To find the best hyperparameters ('best model') for each classification algorithm, we used hyperparameter tuning, i.e., we ran multiple combinations of the classifiers' hyperparameters. The performance of a classifier with a specific set of parameter's values was evaluated based on the results of

the k validation subsets using the k-fold cross validation; for each classifier this results with k values of area under curve (AUC). To choose the best set of parameters, we used Python's scikit-learn (Pedregosa *et al.* 2012) methods GridSearchCV and RandomizedSearchCV, and maximized the AUC values described above (Fig. S16). For each one of the machine learning classifiers, this procedure was performed twice: with the over-sampling methods SMOTE and ROS. The over-sampling method that produced the highest AUC values for each classifier was utilized.

To determine which one of the four classifiers used had the best performance (hence termed 'best' classifier), we compared the k values of AUC for each classifier. Student's t-test was used to determine if the difference between the classifiers was statistically significant. The best performing classifier was applied to the test dataset which was unused during the training phase.

Machine learning evaluation

To evaluate the performance of the best classifier, we first checked how sensitive the results are with respect to the initial (random) decision of which part of the dataset will serve as a train and which part will be the test. Then we compared the results obtained to the results of the same dataset, but with the class labels shuffled such that it should not contain biological meaning.

Specifically, we followed these steps:

A. We repeated the pipeline (Fig. S16, but only for the best classifier) 30 times for every comparison, but each time the data was split differently (and randomly) into training and testing sets with a ratio of 8:2. The AUC value of the test data was recorded each time (using the best model parameters as described above), resulting in 30 AUC values.
B. We generated non-biological realizations of the dataset. These realizations were constructed by randomly shuffling the labels of proximal and non-proximal cells (the 'true' and 'false' class labels), in each comparison, in a way that kept their relative proportions. This procedure was repeated 30 times, and each time we ran the complete pipeline (Fig. S16, but only for the best classifier). This resulted in 30 AUC values of non-biological datasets for each comparison.
C. Finally, for each comparison, we checked whether the 30 AUC values of the best model, obtained using the original unshuffled dataset, are higher compared to the ones obtained

using shuffled datasets. A p-value for each comparison was generated using the Student's t-test; note that the 30 AUC values, both from the biological and non-biological data, were normally distributed. We kept only the machine learning results for every comparison that had a Benjamini-Hochberg false discovery rate (FDR) of 1e-4. The stringent FDR value was chosen to further increase the confidence of the machine learning results, given the low number of cells used in the classification (Tables S4-5).

Finally, for comparisons that passed the aforementioned test, the best classifier was applied to the complete dataset, i.e. without splitting into train and test. The classifier model was rebuilt as discussed above. To avoid errors that result from inaccurate boundaries detection of two adjacent cells, we filtered upregulated genes detected in X cells if they are known cell markers for the Y cells (the known marker genes are listed in the Methods section 'Description of the datasets'). The features (genes) that had the highest influence on the classification are presented in Fig. S18. The feature importance values were calculated using SHapley Additive exPlanations (SHAP) technique (Lundberg and Lee 2017).

cNMF analysis

We implemented cNMF (Kotliar *et al.* 2019) analysis with the aim of detecting a battery of genes that change their expression together as a result of proximity between immune and tumor cells. Using this analysis we discovered gene signatures, namely gene expression programs ('GEP), which define cell types as well as cell states. Using the python package 'cnmf' (Kotliar *et al.* 2019), we created cNMF object and ran the command 'run_nmf'. The input was the gene expression (counts) matrix, including 297 genes and 2400 cells (we dropped cells that belong to 'unknown' cell type). After applying the cNMF algorithm, a gene expression matrix decomposition yielded a pair of low rank matrices. One matrix, named 'usage matrix' (N cells x K GEP components), accounts for the probability of each cell expressing each GEP, and the second matrix, 'components matrix' (K GEP components x G genes), presents the combinations of genes that are expressed in each GEP. The package was run using default parameters, with 500 iterations per K, and without filtering any iterations. To decide how many components will fit our data, we ran cNMF for a range of Ks from 8 to 29. We continued with K=18 since its stability-error ratio is the greatest compared to other values of K (Fig. S19A).

Detection of cell type GEPs

Based on the usage matrix obtained from the cNMF analysis, we extracted GEP usage values for the different cell types, after aggregating the cells into five main categories: T cell, B cell, Macrophage, Fibroblast and Tumor. To test whether a specific GEP is over-represented in a given cell type, we performed a permutation analysis. Specifically, we performed 100,000 realizations of the original dataset, in each realization we randomly shuffled cell type labels between the individual cells, while preserving the original number of cells in each cell type. Then, for each cell in each cell type we compared the GEP usage in the shuffled dataset to the usage in the original dataset. For each realization this comparison was performed separately for each GEP in each cell type using a right-tailed nonparametric Wilcoxon Rank-Sum test, with the null hypothesis that the average usage in all the cells in the original dataset is the same as the average usage in the permuted dataset. Cases in which the null hypothesis was rejected, using a confidence of 0.005 (Benjamini-Hochberg FDR), indicated that the usage in the original dataset was higher than the usage in the shuffled data. Tso estimate the significance of these cases, we calculated a p-value, defined as the number of times the null hypothesis was not rejected (plus one), divided by the total number of realizations. For example, a permutation analysis p-value of 1e-5 means that the null hypothesis was rejected in all the realizations, suggesting that a specific GEP is over-represented in the original dataset. Finally, the permutation analysis p-value was corrected for multiple testing using the Benjamini-Hochberg procedure (Fig. S19B).

Detection of proximity-related GEPs

Next, we examined whether GEPs can be overexpressed or under expressed in a cell type as a result of physical distance from other cell types ('proximity-related' GEPs). First, we divided each non-tumor cell type (T cell, B cell, macrophage, fibroblast) into two subgroups - cells proximal to tumor cells versus cells that are not close to tumor cells. Proximity was defined as 3 micrometer distance, before expansion, between the cell bodies boundaries. Similarly, we divided the tumor cells into two groups: cells that are proximal to non-tumor cells, and tumor cells distant from non-tumor cells. Then, for each GEP in each cell type, we performed a two-tailed nonparametric Wilcoxon Rank-Sum test, comparing the usage of that GEP in the proximal cells subgroup to the usage in the non-proximal subgroup. In cases when the null hypothesis (equal means) was rejected with Benjamini-Hochberg FDR < 0.005, we further tested the detected GEPs using permutation analysis. 100,000 realizations of the original dataset were created, in each realization, the identity of the individual cell as belonging to the

proximal subgroup versus the non-proximal subgroup was shuffled in each cell type separately, while preserving the total number of cells in each subgroup. Next, we applied a two-tailed nonparametric Wilcoxon Rank-Sum test to compare the GEP usage of the shuffled subgroups (proximal versus non-proximal) in each cell type. For each realization, we counted the number of times the Wilcoxon p-value of the shuffled data was lower than the Wilcoxon p-value obtained using the original data, for the same cell type and GEP. The total counts (plus 1) divided by the number of realizations is the permutation analysis p-value of a specific GEP in an individual cell type. The GEPs permutation analysis p-values are presented in Fig. S19C. Only p-values that passed a Benjamini-Hochberg multiple test correction (FDR<0.05) are presented, and the corresponding GEPs are treated as proximity-related GEPs.

Quantifying the statistical significance of overlapping genes

We assessed the statistical significance of the overlapping genes between any two detection methods (differential expression, machine learning and matrix factorization) with a bootstrapping approach. For any pair of cell clusters X and Y, the X cells were randomly partitioned into two subsets, of the same size as the two original subsets (originally these subsets were the X cells that are proximal to Y cells, and the X cells that are not proximal to Y cells). The random partition procedure was repeated 100 times. To assess the statistical significance of the overlap between the differential expression approach and the other approaches, the p-values of all genes, when comparing the two random subsets, were calculated using DESeq2. To assess the statistical significance of the overlap between the machine learning and matrix factorization approach, the genes that have the highest influence on the classification of the two random subsets were detected using the machine learning pipeline. Next, in each random realization, we sorted the genes according to their p-values (from lowest to highest) or according to their influence on the classification (from highest to lowest) and took the same number of genes as obtained with the real data. Finally, in each random realization, we examined the number of overlapping genes when comparing two detection methods. Bootstrap p-values were calculated for each comparison as (1+x)/100 where x is the number of realizations in which the number of overlapping genes between the two detection methods was above the original number of the overlapping genes.

Scale-down analysis

To explore the number of proximity-induced genes as a fraction of the data utilized, we performed a scale-down analysis (Fig. S20-21). Specifically, we have randomly sampled fields of view from the full sample size, and repeated the full analysis of the detection of proximity-induced genes using differential expression analysis (Methods section 'Detecting differentially expressed genes'). We randomly sampled 100 times each fraction of data going from 40% to 95% in increments of 5%. Then we calculated for each data fraction the average and the standard error of the number of overlapping upregulated proximity-induced genes (i.e., the overlap between the genes detected with a specific fraction of data and the full dataset). We also calculated the average and the standard error of the number of adjacent cells for each fraction of data. Next, we examined if the results fit a linear or a quadratic curve (Fig. S20-21). This analysis revealed a linear trend between the fraction of the data utilized and the number of proximity-induced genes revealed in T cells (Fig. S20, left panel). Importantly, a linear trend is also observed between the number of proximity-induced genes in T cells and the number of adjacent T cells and tumor cells (Fig. S20, right panel). The linear trend is also evident in B cells, Fibroblasts, and Macrophages (Fig. S21). Thus, the number of adjacent non-tumor and tumor cells present in the studied biopsy might suggest the number of proximity-induced genes that can be detected.

Moran's I calculation

We implemented Moran's I calculation in the context of spatially-resolved transcriptomics (Hao *et al.* 2021; Hu *et al.* 2021). Moran's I computes the correlation of an expression of a gene with itself through space, and produces a number between -1 and 1 (akin to a correlation coefficient), that measures the dependence of the expression of the gene on spatial location. The Moran's I values are calculated by dividing the space into bins and then computing the spatial autocorrelation between these bins.

Similarly to a recent implementation (Miller *et al.* 2021), we account for non-uniform cell distribution in the tissue. We compute a p-value for the spatial dependence of each gene by taking into account the locations of all the genes expressed in the tissue. To highlight the need for this, consider the following scenario: say that there is a high density of cells in a particular location in the tissue, and as a result many genes have high expression levels in that location. In this scenario the distribution of gene expression in space will not be uniform, and therefore the Moran's I might be high for nearly all genes tested. This will result in the trivial solution that nearly all the genes are spatially-dependent, which is technically correct but simply a result

of the tissue topography. However, if the distribution of locations of all the genes in the tissue is considered, then we can pinpoint specific genes that have higher spatial dependence relative to other expressed genes.

In our implementation the user does not need to decide in advance how to divide the sample into spatial bins. Instead of pre-defining the spatial bins, we iterate over many possible bin sizes and select the grid that produces the most robust results for the spatially-dependent genes (details below).

Similar to other implementations, we process the image in 2D, by a max projection of the gene expression from 3D to 2D. It is possible to implement Moran's I in 3D, but it is less useful for tumor tissues since typically the length in the Z axis is two (or more) orders of magnitude smaller compared to the X and Y axes; for the tissue processed here the numbers are 8 microns in Z axis compared to 1347 x 621 microns in the X and Y axes.

To compute a p-value for the spatial dependence of each gene (Fig. S22-23-24), our implementation of the Moran's I includes the following steps:

A. The complete image, i.e. all FOVs combined, is binned using a rectangular grid in the X and Y axes. We start with a coarse grid, by dividing the smallest dimension by 10. The size in the other dimension is determined such that the resulting X&Y bin is as close as possible to a square, taking into account the size of the tissue. For example, for the studied tissue the X dimension is initially divided by 20 whereas the Y dimension is by 10.
B. Moran's I is computed for all the genes, for the specific grid. For each gene, the number of mRNA molecules (i.e. sequencing reads) in each grid bin is calculated. This data is the input for the 'moransI' Python function which computes Moran's I.
C. To account for the overall distribution of genes in the tissue, for each gene with N mRNA molecules localized in space, we randomly pick N locations from the locations of all mRNA molecules of all the genes. The Moran's I is then computed using the randomly chosen locations as in step (B) above.
D. To calculate the p-value for the spatial dependence of each gene, we repeat step (C) 100 times for each gene and fit a normal distribution for the resulting Moran's I values. The original Moran's I value for each gene (step B), and the normal distribution, allow us to calculate the gene's spatial dependence p-value.

E. Accounting for multiple testing on all the genes, only genes with FDR<0.01 are recorded.
F. The grid is iteratively refined, by adding 5 more divisions to the smallest axis in each iteration; i.e. 10, 15, 20 divisions and so on, whereas the divisions in the other dimension are modified as described in step (A). For each resulting grid, we repeat steps (B)-(E) above. The last iteration is when the division creates bins which are roughly the size of an individual cell, which is ~10 microns in the studied tissue. This parameter, as well as the initial division and the step size for the divisions, can be modified by the user.
G. At this point for each grid tested we have a list of spatially-dependent genes. To choose the 'best' grid, we compare the lists of genes between grids. Specifically, starting from the second iteration, for each grid we ask what fraction of its list of genes is shared with the grid from a previous iteration (coarser grid) and with the grid from the next iteration (finer grid). Say 50 genes are detected as spatially-dependent in iteration i, and 20 of these genes are also detected in both iteration i-1 and iteration i+1, then for iteration i we record the fraction 0.4 (20/50).
H. The grid with the local highest maximal fraction is chosen and the genes that are detected as spatially-dependent in that grid are reported.

We examined if the genes detected as spatially variable using Moran's I are a consequence of the cell type spatial variability. Alternatively, these genes can be spatially variable in spite (or in excess) of the cell type spatial variability. To test these two possibilities, we did the following: for a given gene, in a given cell type, we computed the Moran's I of the RNA molecules from that given gene. Importantly, we used only locations which are inside the cells of the given cell type. Next, the significance of this Moran's I score was compared to the background of the given cell type distribution. This was done by realization analysis: the same number of RNA molecules (as that of the given gene) were randomly selected from all RNA locations inside cells of the given cell type, and the Moran's I was calculated for this realization. From 100 realizations the p-value of the original Moran's I score was calculated. This analysis clearly revealed that many genes are spatially variable in excess of cell type spatial variability. Examples of such genes are in Fig. S23 and a list of the most significant genes is in Table. S9.

We also implemented Moran's I on the level of cells from a given cell type. I.e. for each cell type we generate a p-value for the spatial dependence of the cells in the given type. The

implementation is identical to the implementation for genes discussed above, with the following modifications: For each cell type we recorded the locations of all the cells. The cell's location was defined as the mean position of mRNA molecules assigned to a given cell. The calculation was performed after maximal projection, as discussed above. The locations of the cells for each cell type were utilized for the calculation of Moran's I, with the grid search as discussed above. To calculate the p-value, for each cell type with N cells, we randomly picked the locations of N cells from all the cells, regardless of cell type. Although most of the cell types were found to be spatially-dependent in a statistically significant manner, a clear difference was evident between the non-tumor cell types and tumor cells. T cells, B cells and Fibroblasts have a clear spatial dependence and accordingly low p-values, whereas tumor cell types are distributed similarly to random cells and accordingly have higher p-values (Fig. S25-26). For example, ALDH1A3-positive tumor cells are very distinct in their molecular content (Fig. S10C), but the cells are distributed in the tissue in a manner that lacks clear spatial dependence (Fig. S26I). On the other extreme, all the B cell types are highly spatially-dependent (p-value<1e-15; Fig. S25A and Fig. S26A-B).

Supplementary Figures

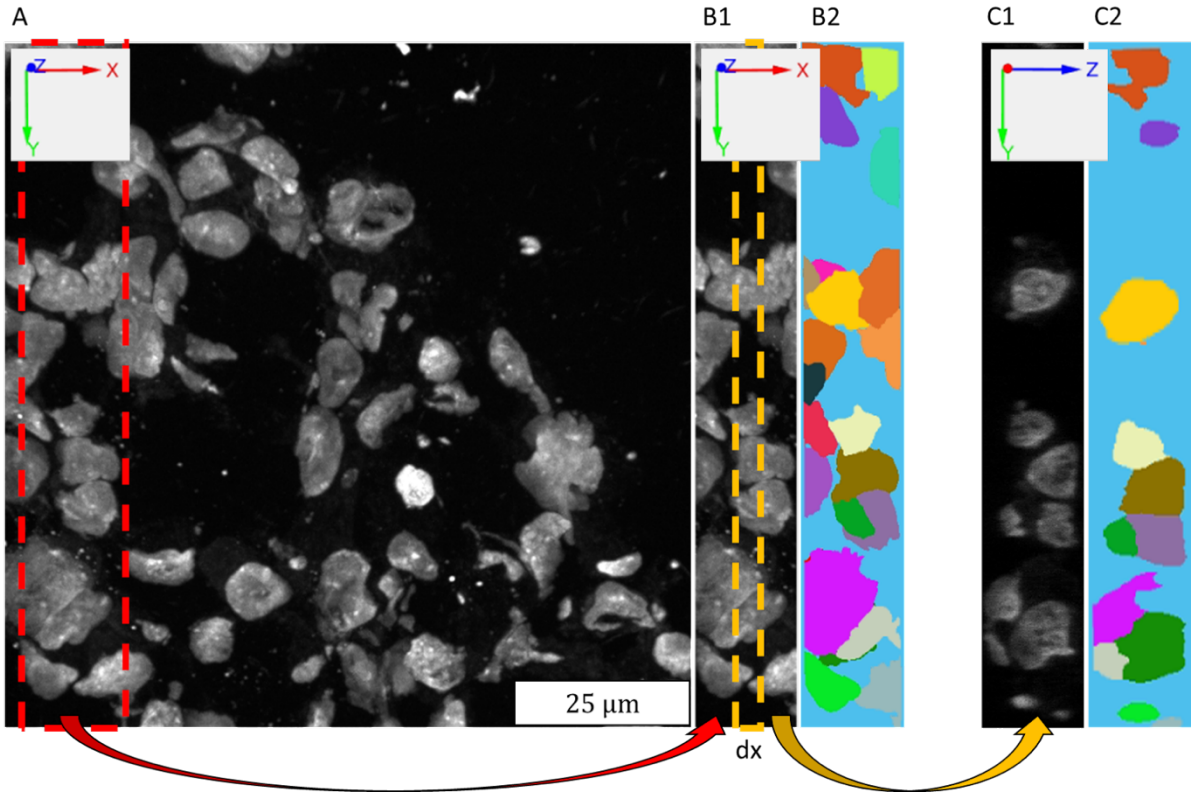


Figure S1. The importance of the z-axis for cell segmentation: cells can overlap in the x-y axes but the z-axis planes can allow 3D separation. InSituSeg considers the 3D volume of the cells in order to separate them and allow a 3D segmentation of the cell bodies, resulting in a better assignment of mRNA molecules to the cells. A) x-y plane of one FOV showing a max projection of the DAPI signal. B) x-y plane of the marked region (red square) in A (B1), and with cells segmented by InSituSeg in 3D, each color represents a different cell (B2). C) Side view, i.e., the z-y plane, of the marked region (yellow square) in B1 (C1), and with the cells segmented by InSituSeg in 3D, each color represents a different cell (C2). Note that when looking only at the x-y plane (B) there are several cells which seem as one. However, when looking at the z-axis as well, these cells are evidently different (C).

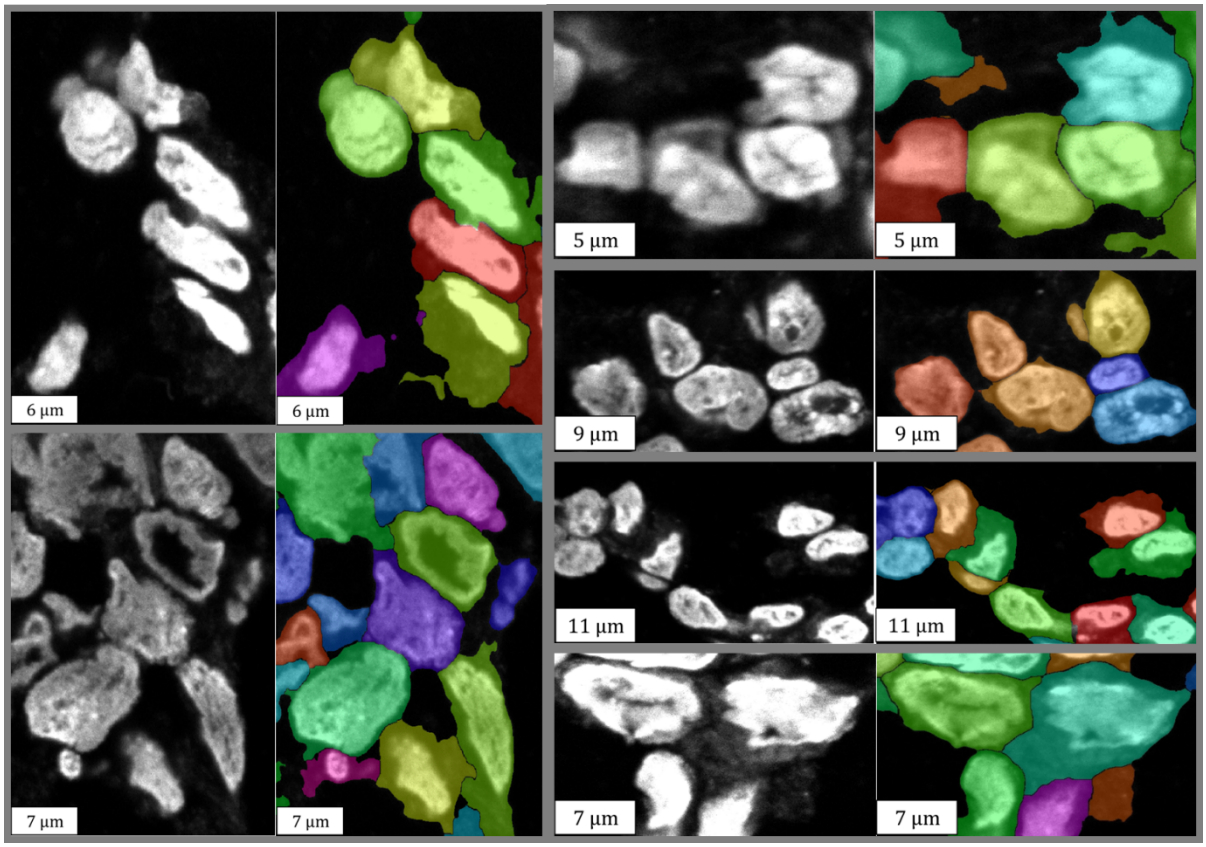


Figure S2. InSituSeg enables the detection of cell-cell contact boundaries due to a marked drop off in the DAPI intensity between cells. In each of the six zoomed-in examples, the left image is the DAPI signal of the cells, and the right image is the cell body masks generated by InSituSeg on top of the DAPI signal of the cells (each color represents a different cell).

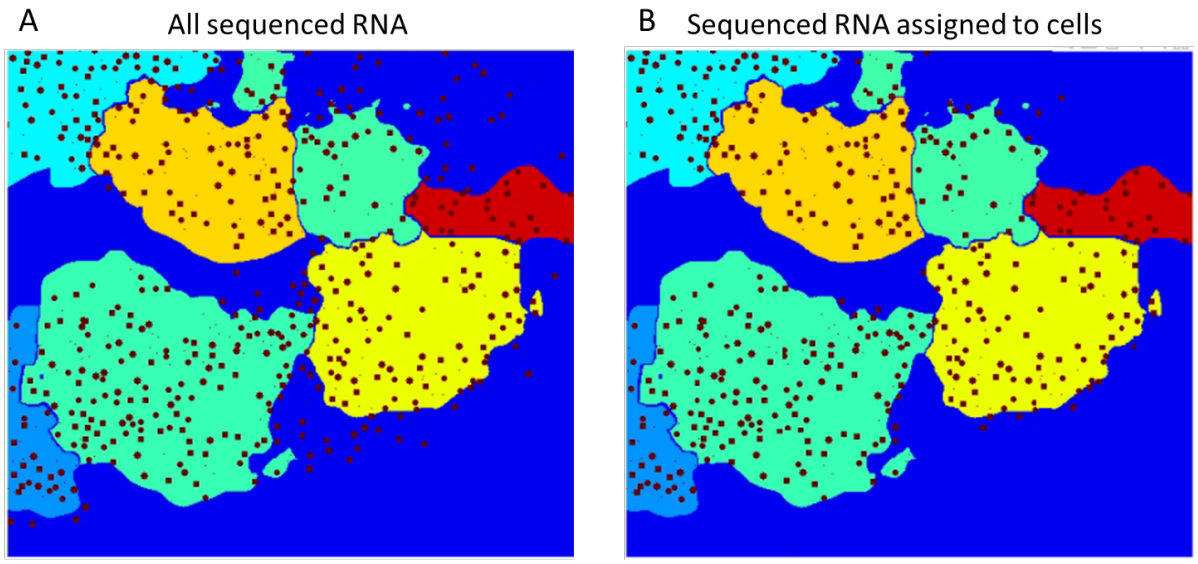


Figure S3. Refinement of the assignment of mRNA molecules into cell bodies with InSituSeg. For mRNA molecules that are outside the cell body objects, InSituSeg attempts to assign them to nearby cells. Out of all the sequenced RNA that are outside the cell bodies, only RNA molecules that are close (<1 micron) to one cell body, but are not close to other cell bodies, are assigned to the closest cell body. Dark blue represents regions outside segmented cells, other colors represent segmented cells, and brown spots represent sequenced RNA molecules. Note that from all the brown spots (RNA molecules) located in the dark blue region (i.e., outside segmented cells) before this refinement step (left panel), only some were assigned to segmented cells with high certainty (right panel). The region presented was randomly chosen.

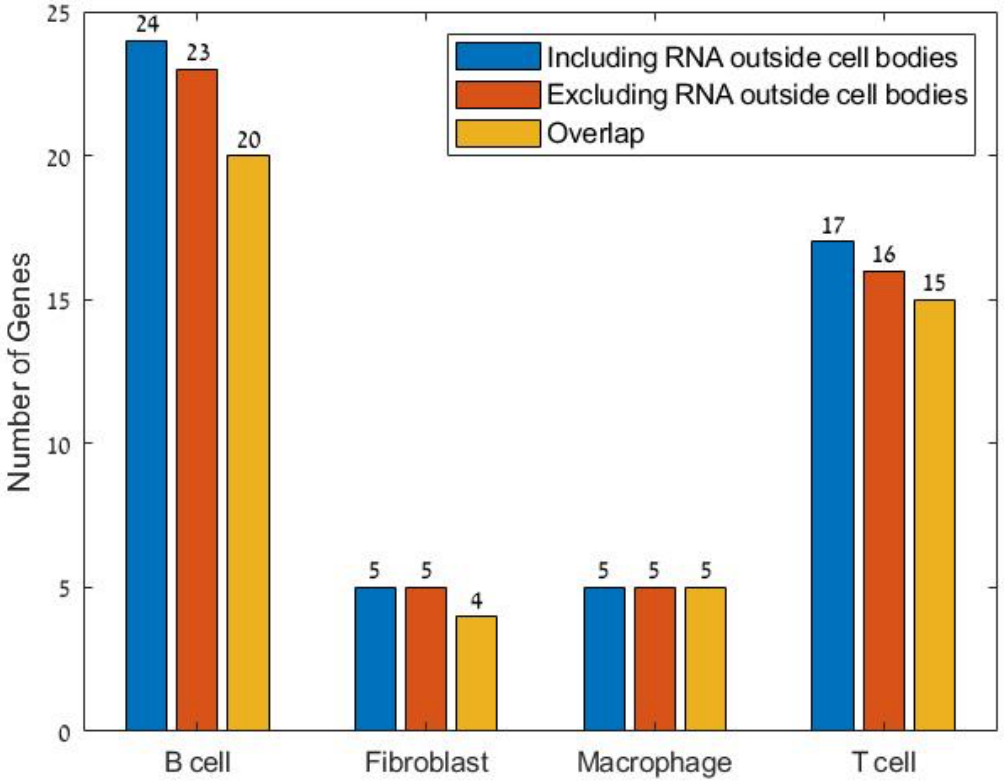


Figure S4. Comparison of the results of proximity-induced genes in cell types proximal to tumor cells, with and without the inclusion of mRNA molecules outside the cell bodies. Overall, 95.3% (895,510 out of 939,764) of the sequenced RNA molecules in this sample are inside the cell bodies as detected via InSituSeg. Differentially expressed analysis as described in the 'Detecting differentially expressed genes' section was performed to detect proximity-induced genes. This analysis reveals that the mRNA molecules outside the cell bodies have little influence on the number of proximity-induced genes detected. For example, 16 genes are detected as proximity-induced genes for T cells close to tumor cells, instead of 17 in the original analysis, and 15 of them overlap.

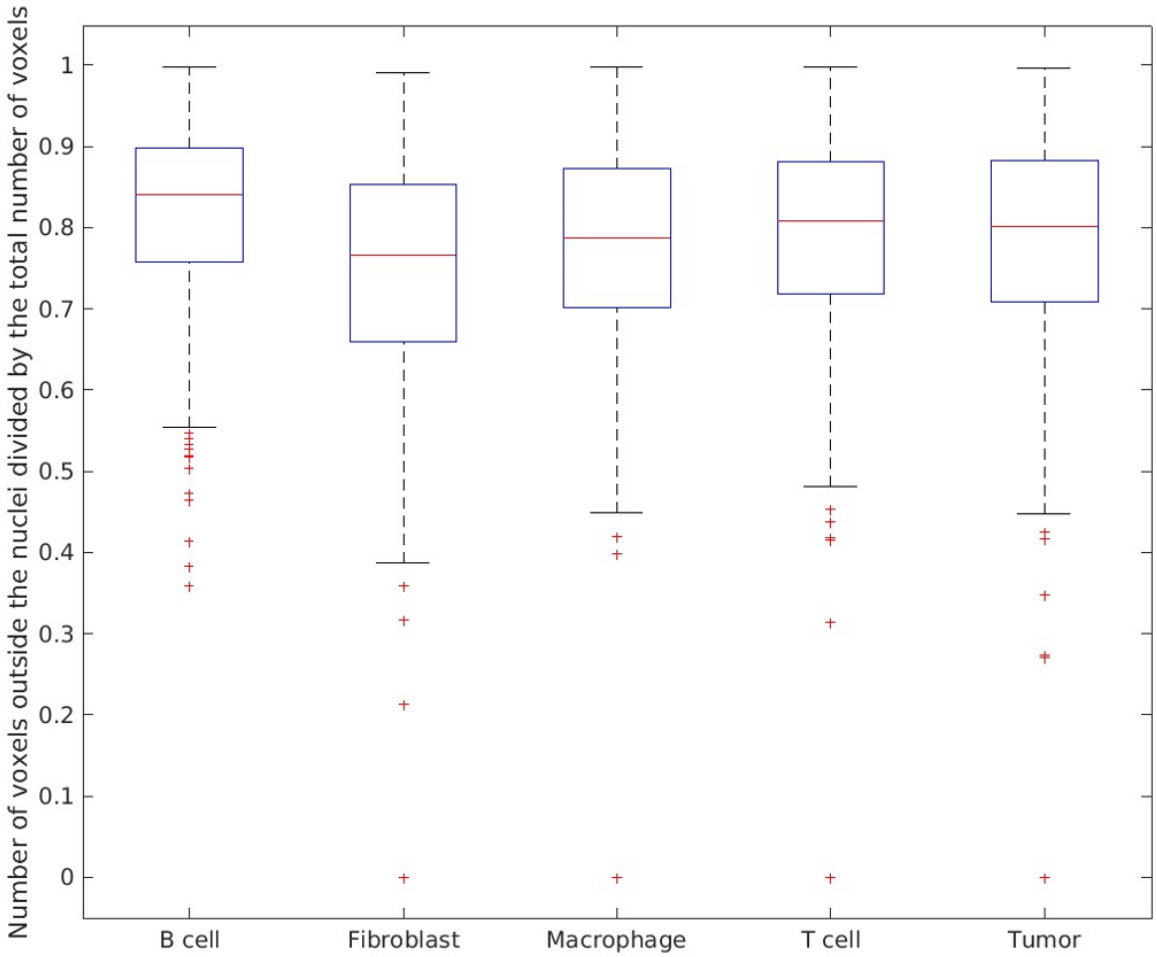


Figure S5. Dependency between the number of voxels outside the nucleus (as detected with InSituSeg), normalized by the total number of voxels in the cell, and cell type. We tested the effect of the cell type on the size of the cell body detected. Specifically, we used the DAPI cytoplasmic staining to segment individual cells via InSituSeg, and then asked what fraction of the voxels are outside the nucleus, as a function of the cell type. The number of voxels identified in the cytosol and the cell type were found to be dependent (One-way ANOVA, p-value=2e-19 with a null hypothesis of equal group means).

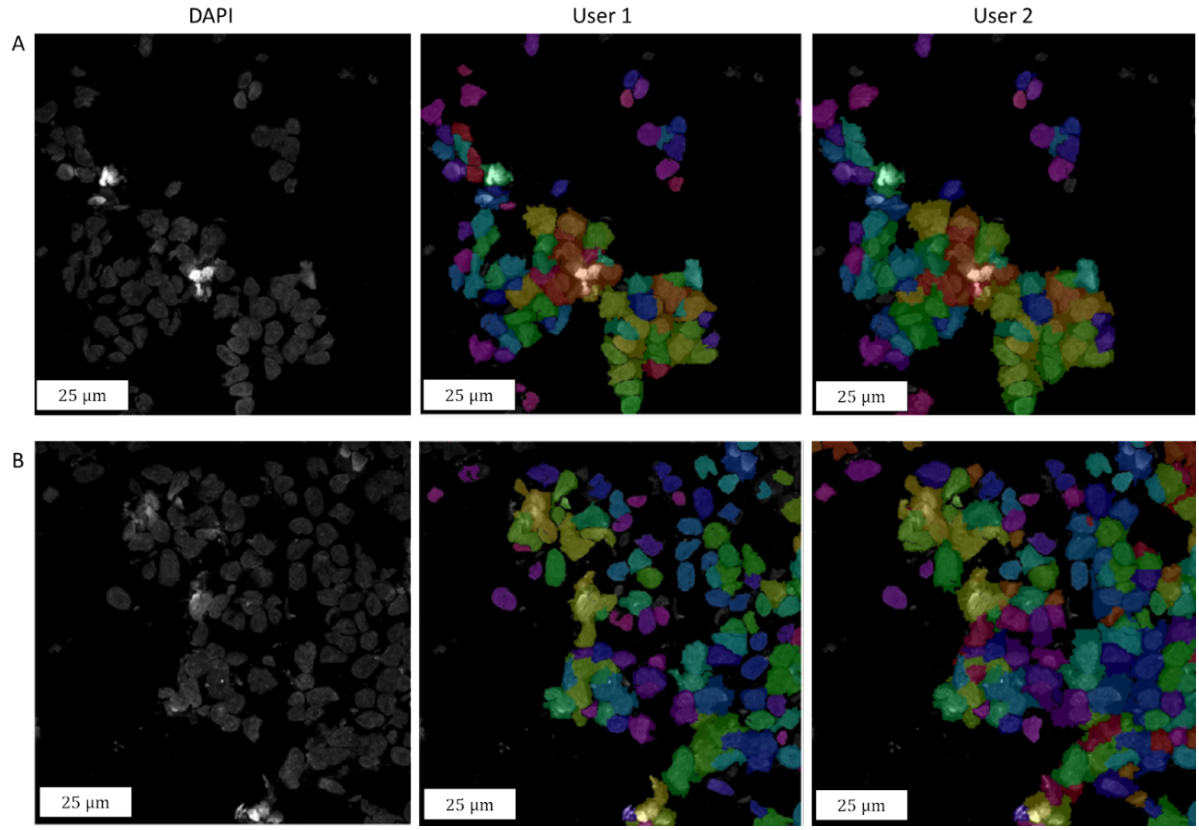


Figure S6. The sensitivity of the InSituSeg analysis to the choices made by the user running the segmentation. InSituSeg results between two individuals who performed the segmentation independently are compared (user 1 in the middle column and user 2 in the right column), using two random fields of view (A and B). We compared the volume of 'matching' cells, i.e., cells which were identified using both segmentations (the color palette in the middle and right columns is the same). High overlap is observed between the 3D segmented cell bodies of the two users (Methods): 94% in A and 92% in B.

	FOV	medfiltmask	min_nuc	min_som	closing-mask	opening-mask	area_remove_quant	area_big_quant	Iterations_step	down-sampling
User 1	A	3	0.978	0.89	1	1	0.3	0.6	0.001	4
User 2	A	3	0.975	0.8	1	1	0.4	0.54	0.001	4
User 1	B	3	0.96	0.83	1	1	0.55	0.6	0.001	4
User 2	B	3	0.955	0.75	1	1	0.4	0.77	0.001	4

The table above compares the segmentation parameters using InSituSeg of two individuals (User 1 and User 2). The segmentation was done twice using two randomly-selected fields of view (A and B), and the chosen parameters are clearly similar. The differences in the parameter set were less than 3% on average. The supplementary text in the Methods: 'Guidelines on how to choose parameters for InSituSeg and fine tune them' gives detailed guidelines about each parameter and how to fine tune their values.

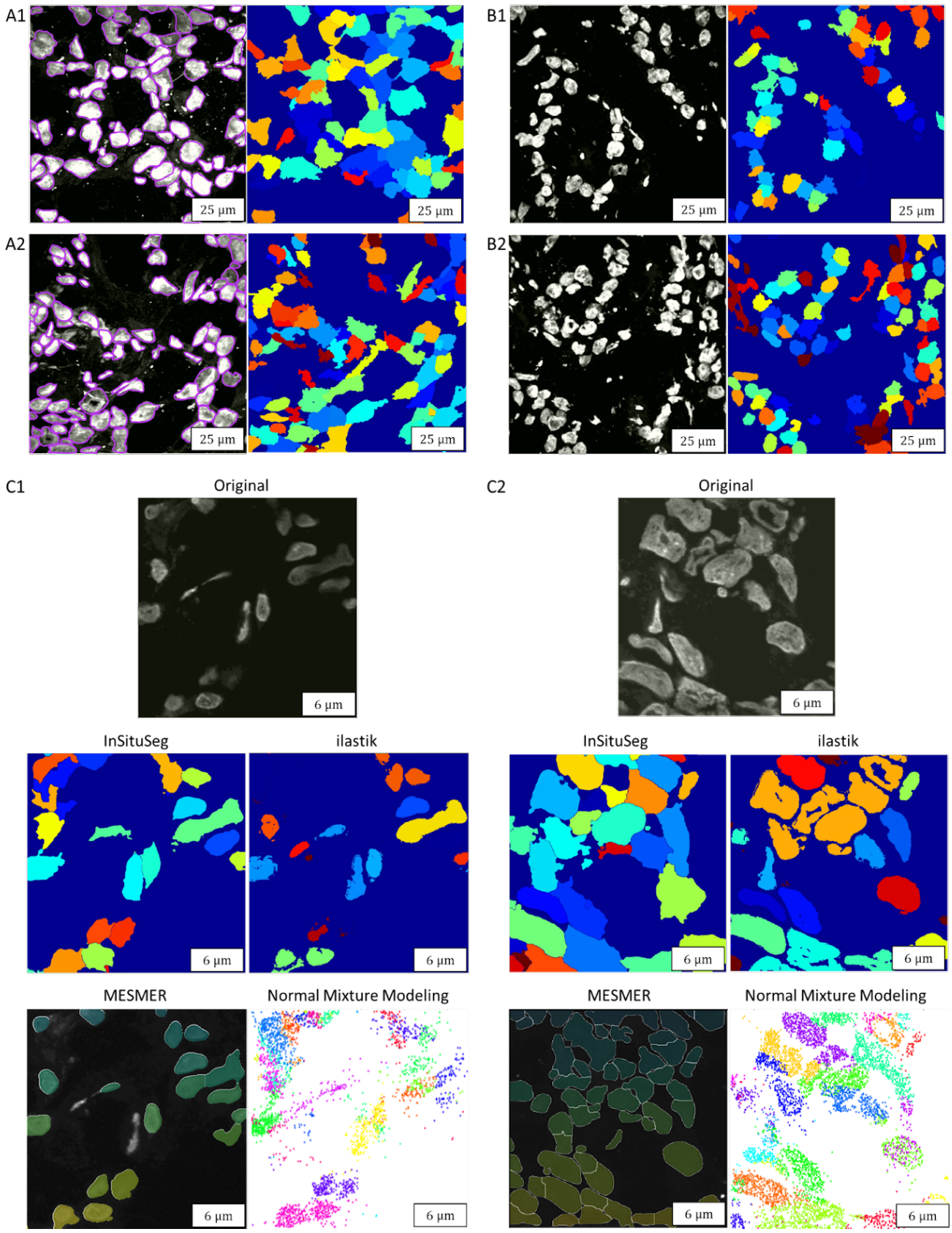


Figure S7. Comparing InSituSeg to manual segmentation and to other available segmentation methods. A) Two examples (A1 top row and A2 bottom row) of comparison between the results of manual segmentation (left) to InSituSeg (right). InSituSeg enables the detection of cell bodies, whereas manual segmentation detects mostly the cell nuclei. B) Two examples (B1 top row and B2 bottom row) of applying InSituSeg on a different core biopsy that was analyzed by expansion sequencing. C) Two

examples (C1 left and C2 right) of a comparison between the results of the segmentation tools ilastik (Berg *et al.* 2019) and Mesmer (Greenwald et al. 2022), to InSituSeg, given the same raw DAPI signal (top). The results of clustering using RNA locations alone are also presented ('Normal Mixture Modeling', see Methods). The regions presented were randomly chosen. Note that these images are 2D projections (via maximal projection) of 3D images and therefore some cells seem truncated.

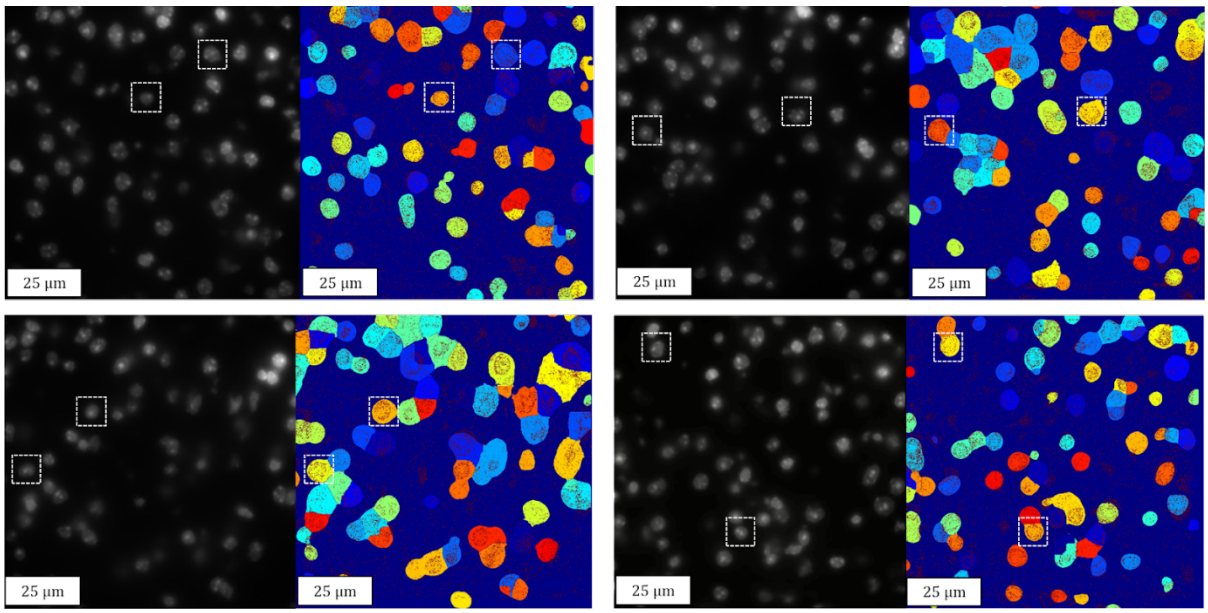


Figure S8. InSituSeg can be applied to MERFISH data for cell body segmentation in 3D. Four FOVs are shown, in each one the left image shows the DAPI signal of the cells in the MERFISH dataset, and the right image shows the segmented cell bodies generated by InSituSeg on the same FOV, with each color represents a different cell. In the right images, the locations of the mRNA molecules (red spots) are overlaid on the InSituSeg segmentation. The white squares highlight a few examples in which InSituSeg allows the detection of cell bodies and not only cell nuclei. Of note, in these cases the segmented cell bodies overlap with the locations of the RNA molecules. The MERFISH data is of coronal, 10µm thick, slices of the mouse primary motor cortex (Zhang *et al.* 2021). The DAPI data and the locations of the mRNA molecules ('spot locations') were downloaded from https://download.brainimagelibrary.org/cf/1c/cf1c1a431ef8d021/, and specifically we processed the sample 'mouse1_sample1_raw'.

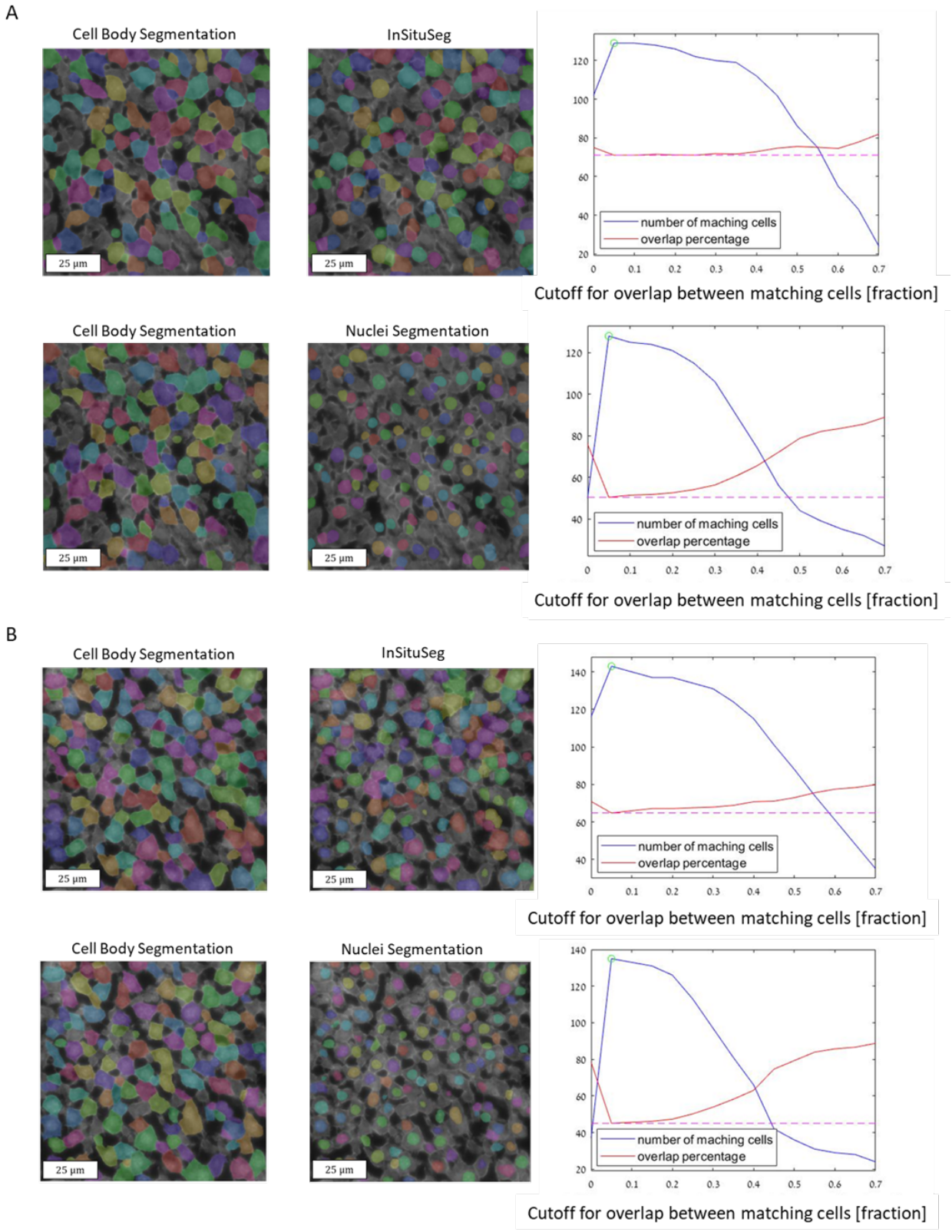
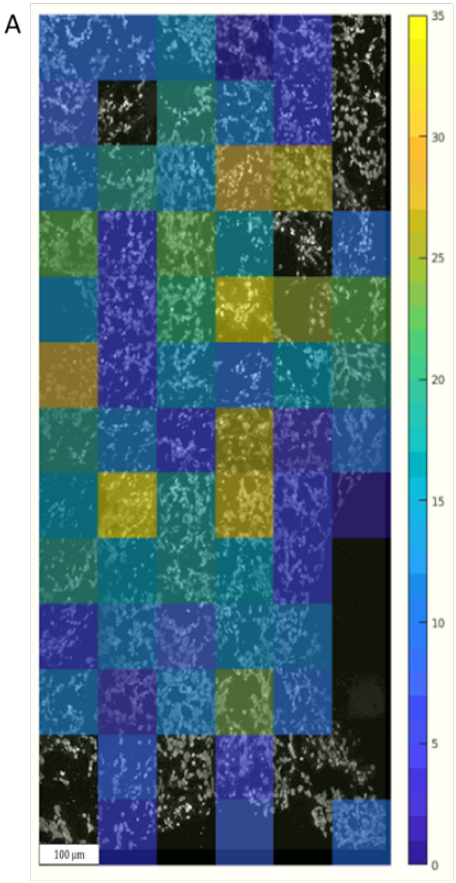
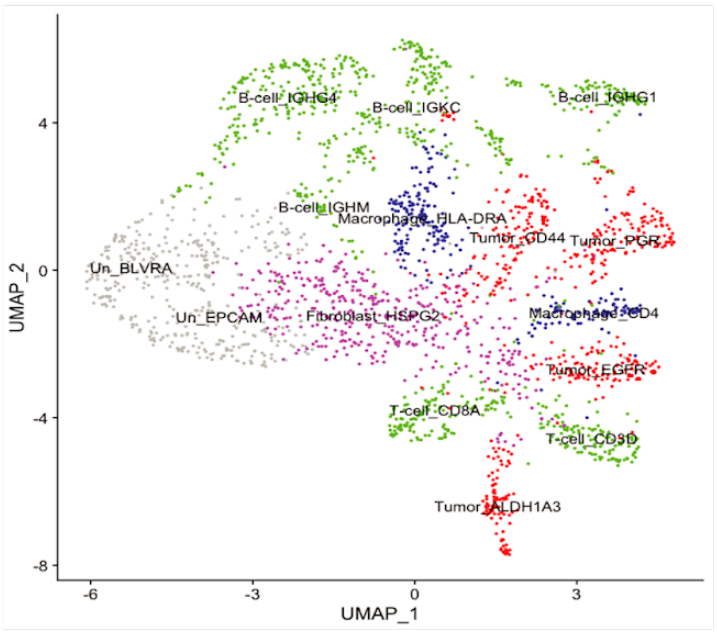
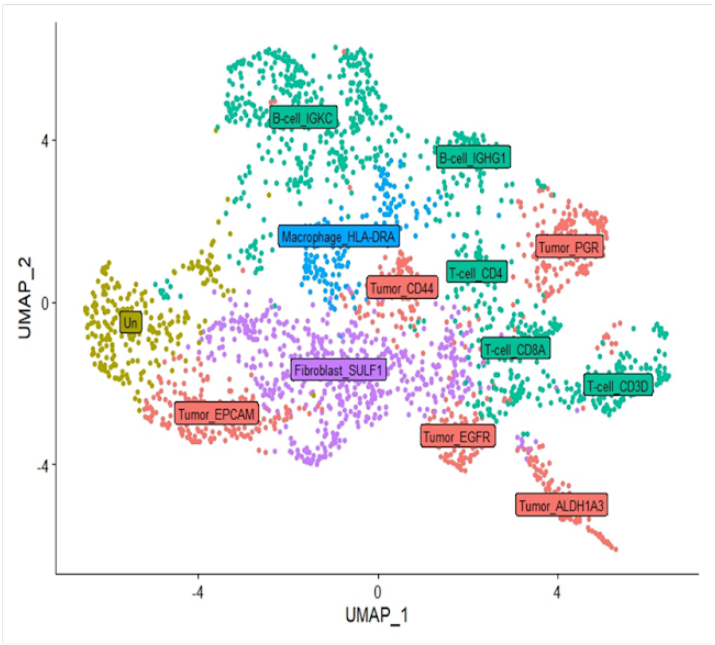


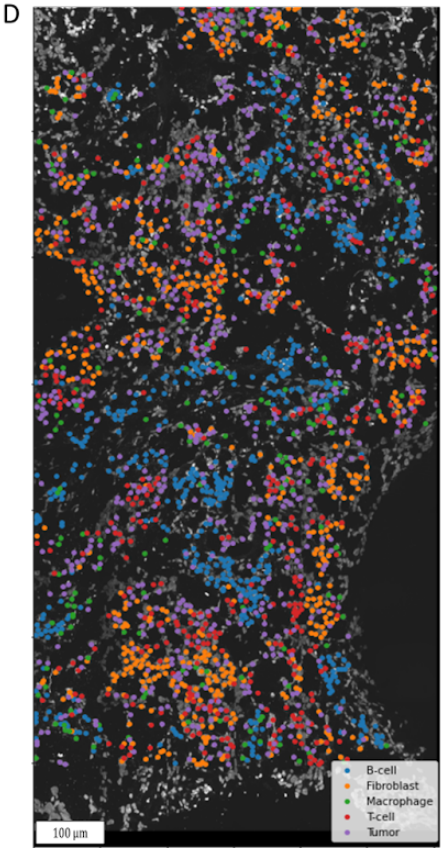
Figure S9. Comparison between segmentation based on cytosolic and membrane staining to segmentation obtained with InSituSeg. A) one field of view, 100x100 microns in size, randomly chosen from the MERFISH data (Methods). The MERFISH data contained the location of 347 genes inside a mouse liver tissue section, in addition to both cytosolic and membrane stains, as well as DAPI

stain. Top row: Cellpose was utilized to segment the cell bodies (left plot), using both cytosolic and membrane stains, and the results are compared to InSituSeg (middle plot), which uses only DAPI data. We compared the area of 'matching' cells, i.e., cells which were identified using both segmentations (same color palette in left and middle plots). Note that the definition of matching cells depends on an overlap area cutoff (right plot). Accordingly, the overlap percentage between the two segmentations depends on the overlap area cutoff (right plot, red line). There is a tradeoff between the number of matching cells and the overlap area cutoff (right plot, blue line). We choose a working point (right plot, green circle, and pink dashed line) to maximize the number of matching cells. Overall 129 matching cells were analyzed, and the average area overlap was 71%. Bottom row: same as the top row, but using segmentation of nuclei instead of InSituSeg segmentation. Overall 128 matching cells were analyzed, and the average area overlap was 50%. B) Same as (A), but using a second field of view, 100x100 microns in size, randomly chosen from the MERFISH data. Top row: Overall 143 matching cells were analyzed, and the average area overlap was 65%. Bottom row: Overall 135 matching cells were analyzed, and the average area overlap was 45%.


A
35
30
25
20
15
10
5
0
100 µm


B
B-cell_IGHG4
B-cell_IGKC
B-cell_IGHG1
B-cell_IGHM
Macrophage_HLA-DRA
Tumor_CD44
Tumor_PGR
Un_BLVRA
Un_EPCAM
Fibroblast_HSPG2
Macrophage_CD4
Tumor_EGFR
T-cell_CD8A
T-cell_CD3D
Tumor_ALDH1A3
UMAP_2
UMAP_1
4
0
-4
-8
-6
-3
0
3


C
B-cell_IGKC
B-cell_IGHG1
Macrophage_HLA-DRA
Tumor_PGR
T-cell_CD4
Tumor_CD44
Un
T-cell_CD8A
Fibroblast_SULF1
T-cell_CD3D
Tumor_EPCAM
Tumor_EGFR
Tumor_ALDH1A3
UMAP_2
UMAP_1
4
0
-4
-4
0
4


D
B-cell
Fibroblast
Macrophage
T-cell
Tumor
100 µm

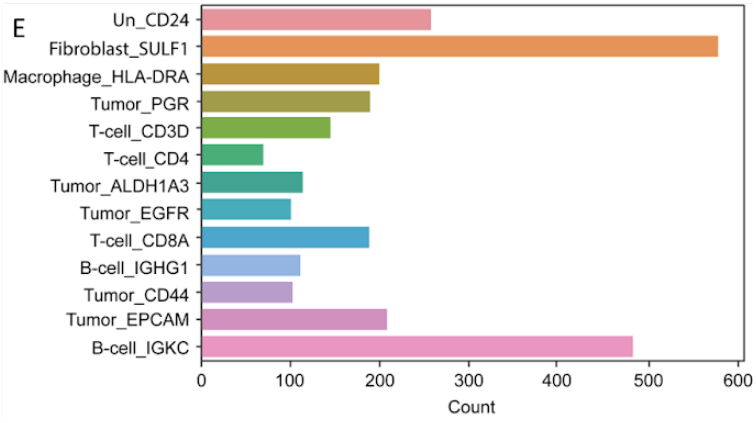

E
Un_CD24
Fibroblast_SULF1
Macrophage_HLA-DRA
Tumor_PGR
T-cell_CD3D
T-cell_CD4
Tumor_ALDH1A3
Tumor_EGFR
T-cell_CD8A
B-cell_IGHG1
Tumor_CD44
Tumor_EPCAM
B-cell_IGKC
0
100
200
300
400
500
600
Count

Figure S10. The biopsy cell statistics after InSituSeg. A) For each FOV, the percent improvement in the number of RNA assigned to the cells using InSituSeg compared to manual segmentation is presented. Overall, manual segmentation of the nuclei using the tool VAST (Berger, Seung, and Lichtman 2018) resulted in 2,395 cells, and 771,904 reads were assigned to them. Using InSituSeg, 2,748 cells were detected, and 939,764 reads were assigned to them. (B-C) Agreement between the Uniform Manifold Approximation and Projection (UMAP) representation of PCA-based expression clustering using manual segmentation (B), and InSituSeg segmented cells (C). The major cell types are represented by different colors: green for T cells and B cells; red for tumor cells; blue for macrophage; magenta for fibroblast; and gray/yellow for unannotated clusters. (D-E) After InsituSeg, the spatial locations of the major cell types in the analyzed tissue biopsy (D), and the number of cells in each clustered cell type (E).

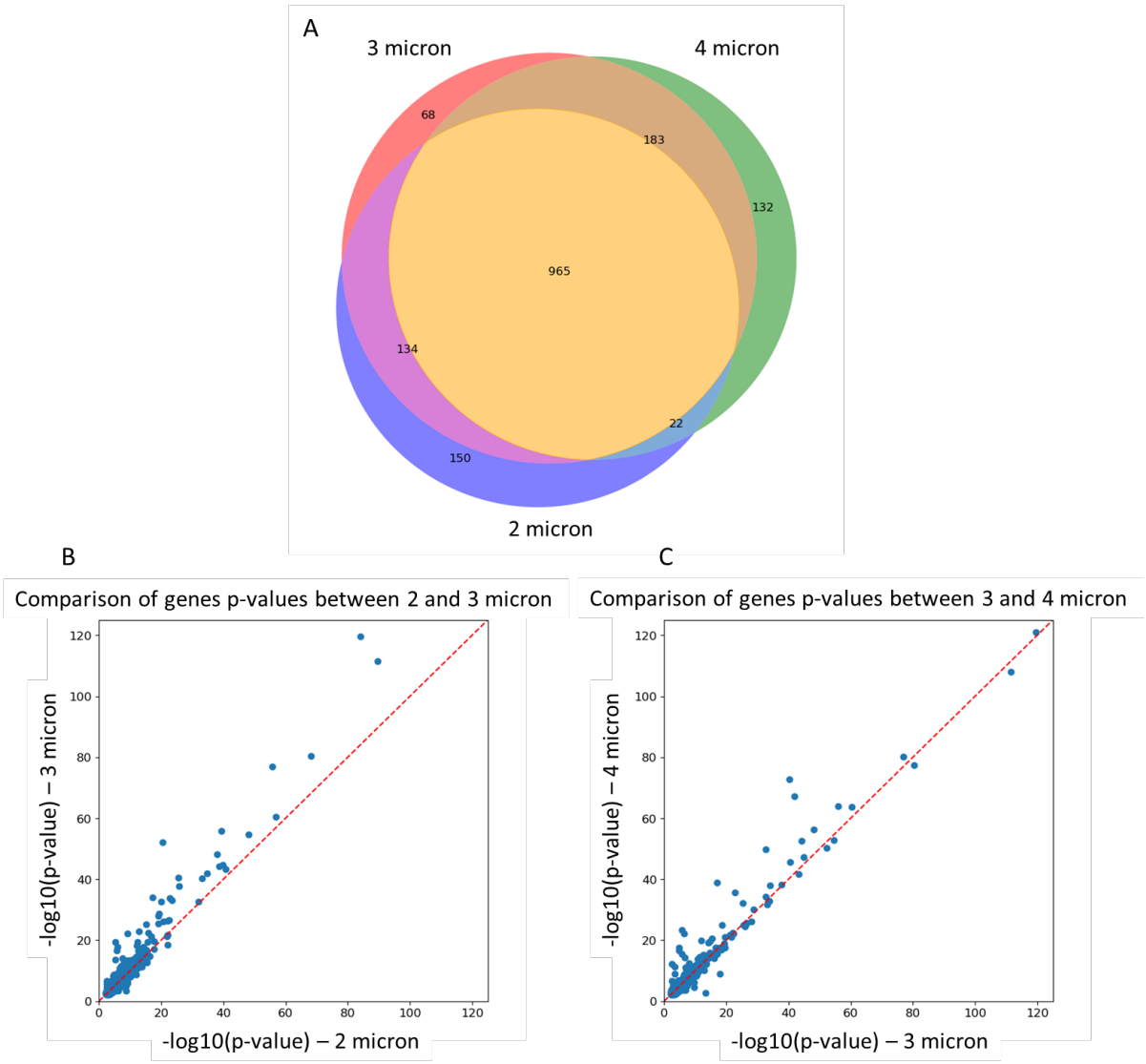


Figure S11. The differential expression analysis is robust to changes in the distance that defines proximity between cells. The differential expression analysis was performed with a distance of 3 microns (before expansion) as the cutoff for proximal cells. To check the robustness of the results, we recalculated all the differentially expressed genes using 2 and 4 microns. A) Venn diagram showing the overlap between the differentially expressed genes detected using 2, 3 and 4 microns as the distance cutoff. The number of genes detected is presented inside the Venn diagram: genes detected only using 3 microns (red); genes detected only using 4 microns (green); genes detected only using 2 microns (purple); genes detected using 3 and 4 microns (brown); genes detected using 2 and 4 microns (light blue); genes detected using 2 and 3 microns (pink); genes detected using 2, 3 and 4 microns (yellow). B) Comparison of the p-values of the genes detected using either 2 microns or 3 microns as a distance cutoff. Pearson's correlation is 0.97 and the correlation p-value is <1e-100. (C) Comparison of the p-value of the genes detected using either 3 microns or 4 microns as a distance cutoff. Pearson's correlation is 0.97 and the correlation p-value is <1e-100.

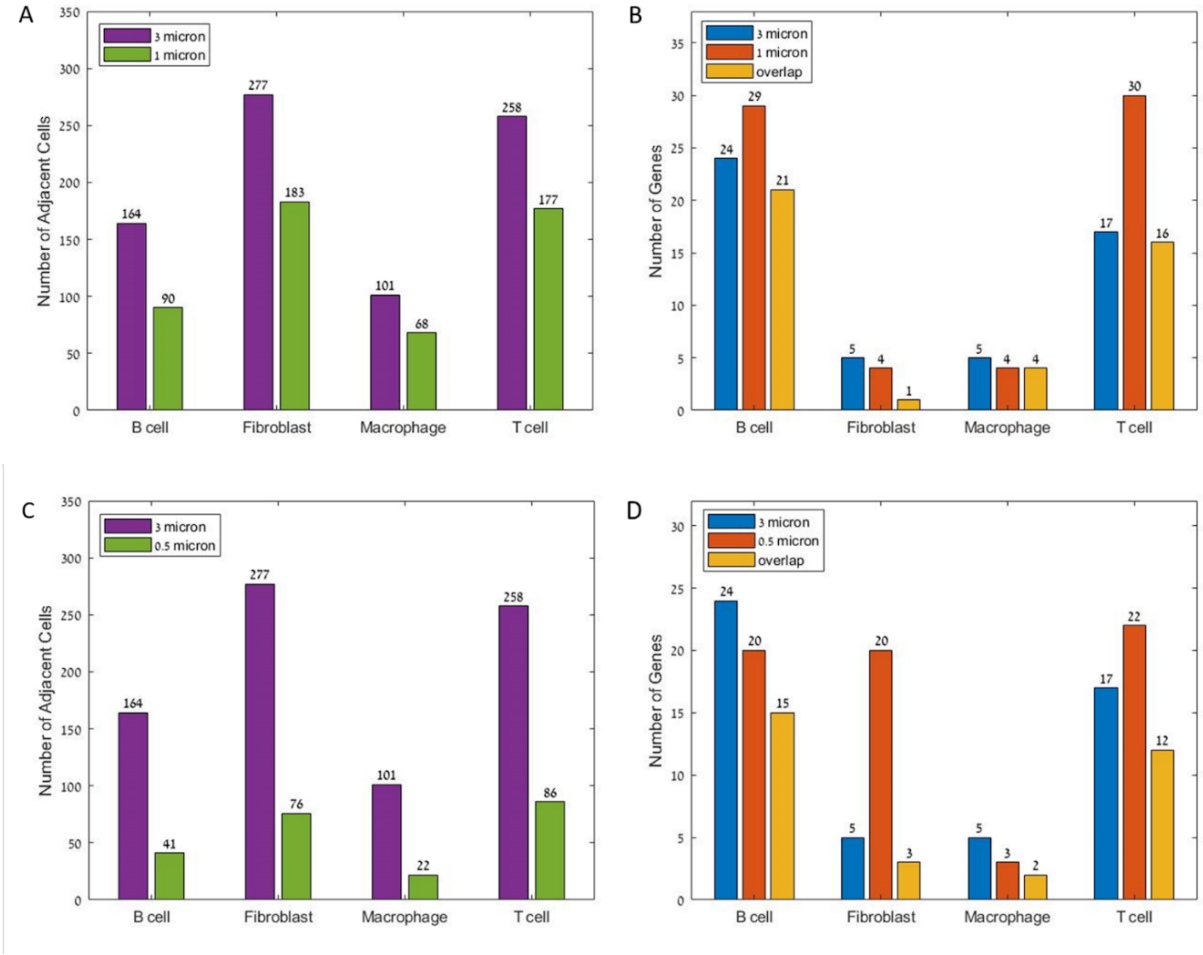


Figure S12. Most of the proximity-induced genes are detected with half a micron threshold of proximity. With 0.5 micron threshold, the adjacent immune and tumor cells are likely to be physically touching. A) The number of non-tumor cells detected as adjacent to tumor cells when using either 3 microns or 1 micron for the proximity cutoff between a non-tumor cell and a tumor cell. For example, out of the 258 T cells that were considered proximal to tumor cells with a distance cutoff of 3 microns, 177 were detected with a distance cutoff of 1 micron. B) Comparison of the proximity-induced genes detected via differential expression analysis when using either 3 microns or 1 micron for the proximity cutoff between a non-tumor cell and a tumor cell. For example, 17 genes were detected as proximity-induced in T cells close to tumor cells when using a 3 microns distance cutoff, and 16 out of them were detected using 1 micron distance cutoff. (C-D), same as (A-B) but using 0.5 microns proximity cutoff instead of 1 micron. For example, out of the 258 T cells that were considered proximal to tumor cells with a distance cutoff of 3 microns, 86 were detected with a distance cutoff of 0.5 microns. In addition, 17 genes were detected as proximity-induced in T cells close to tumor cells when using a 3 microns distance cutoff, and 12 out of them were detected using 0.5 microns distance cutoff.

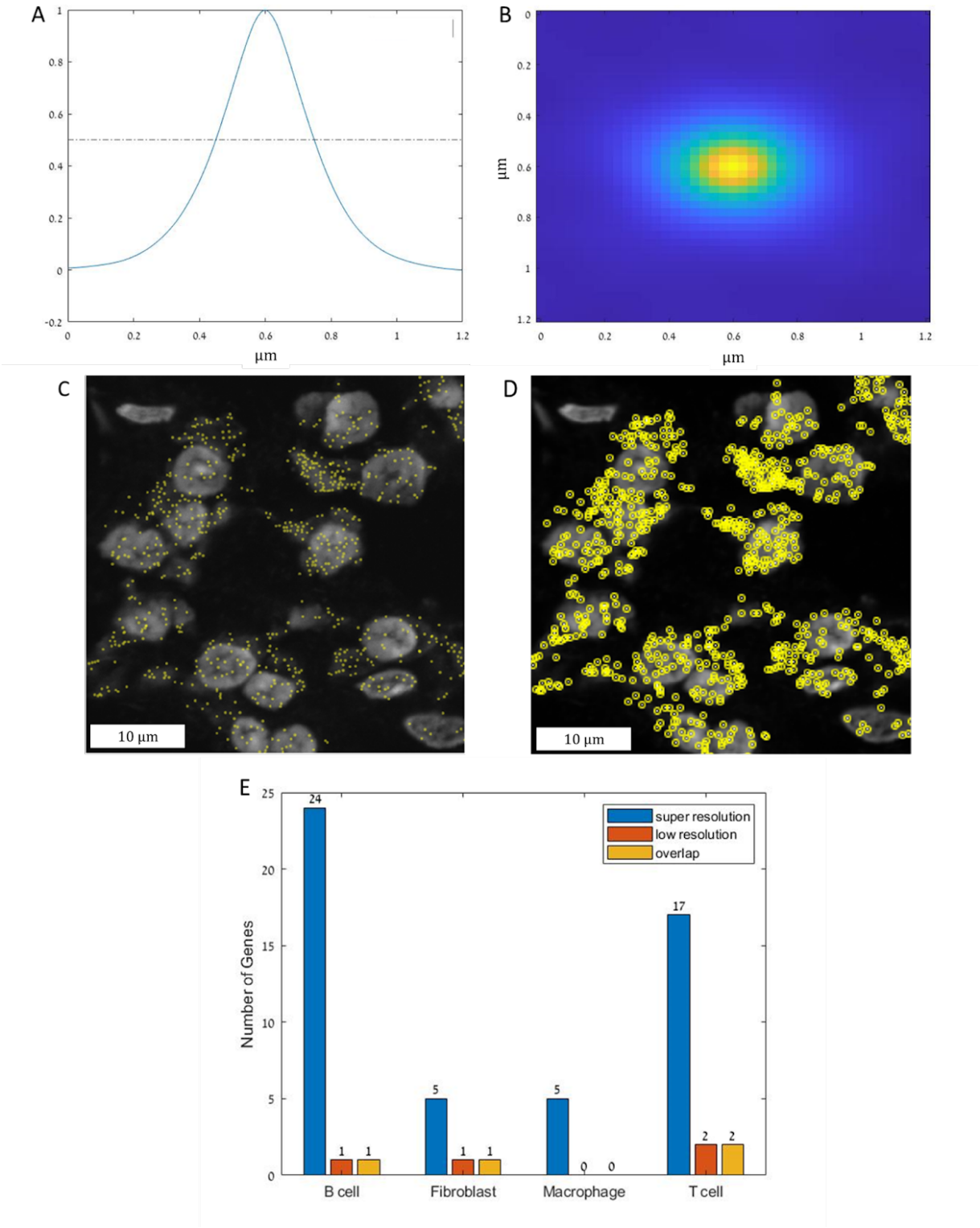
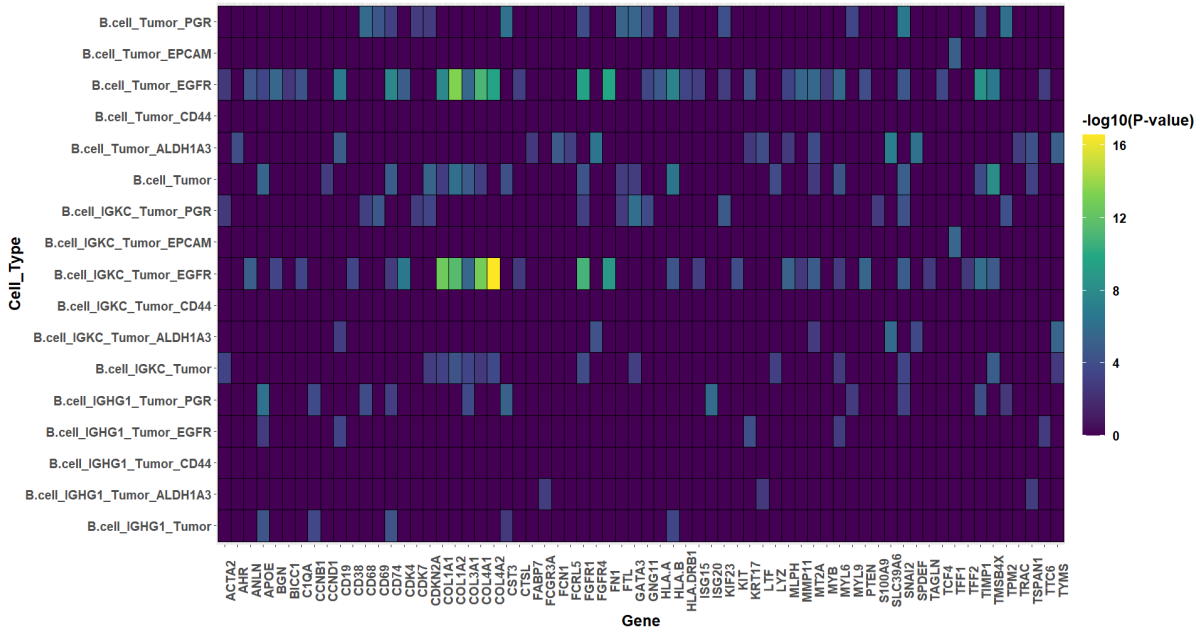
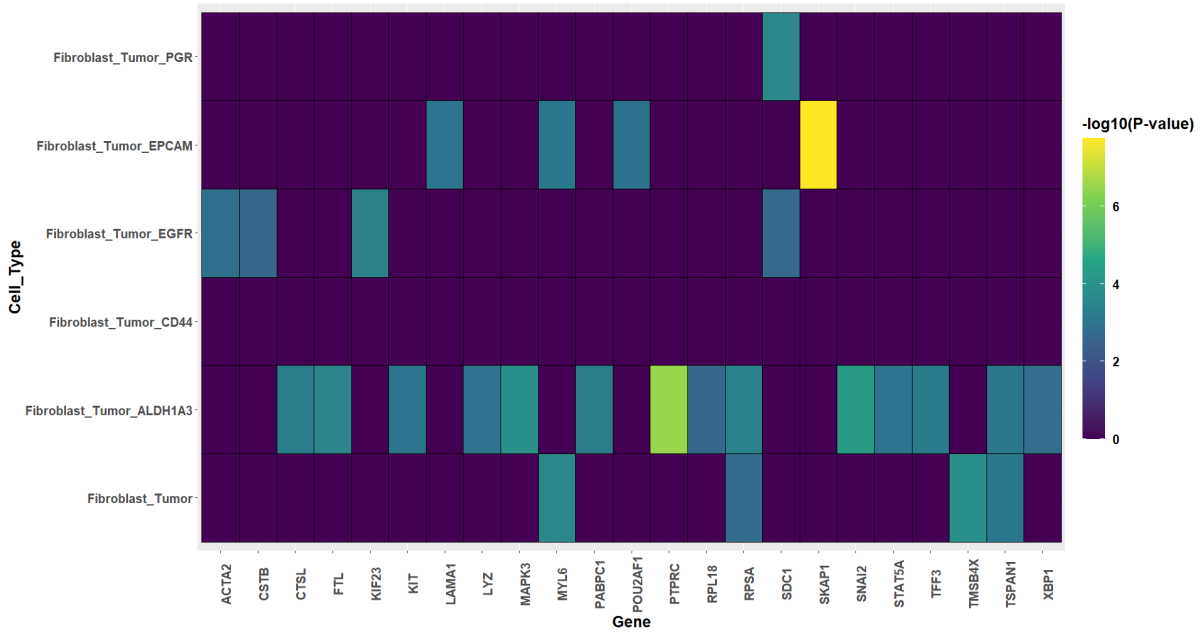


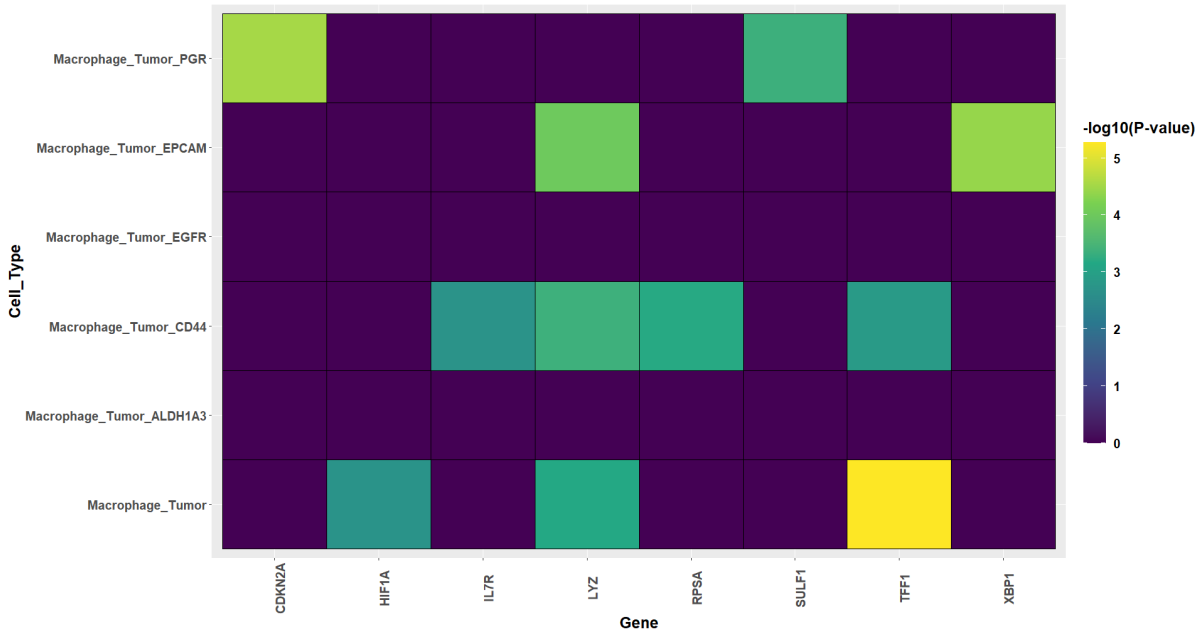
Figure S13. Expansion allows resolving more transcripts and significantly improves detection of proximity-induced genes. A-B) Point spread function (PSF) of the imaging setup utilized for in situ sequencing of the samples. Beads with 100nm diameter were used, and the full-width at half maximum (FWHM) is 270nm. C) The physical location of the identified transcripts in a randomly selected region of a sample analyzed with expansion sequencing (Figure 4B). The diameter of the transcripts (yellow

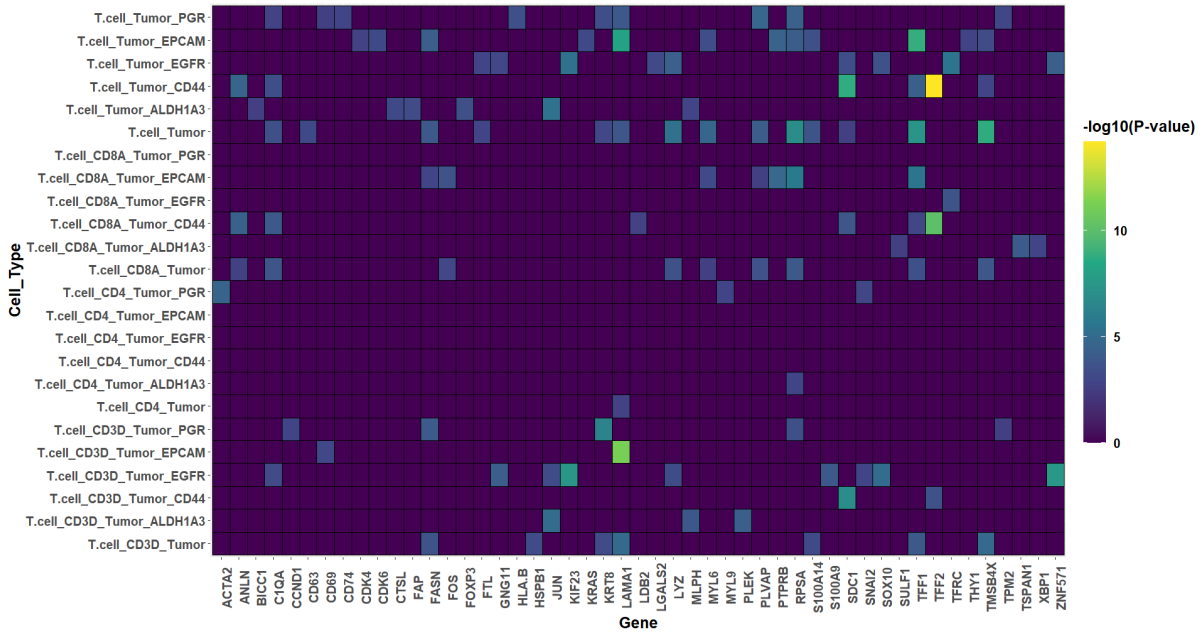
circles) was set to be 300nm, which is the average diameter of the rolonies (Alon et al. 2021), multiplied by a factor of 2.7 (FWHM of 270nm divided by 100nm beads). D) Estimated physical location of the identified transcripts in the same randomly selected region as (C), but without the physical expansion. The physical size of the rolonies (yellow circles) remains the same as (C), but the number of pixels in X and Y was reduced by the expansion factor of 3.3, resulting in likely overlapping rolonies (overlapping yellow circles). Overall, only 77,906 out of 939,764 sequenced RNA molecules in this sample are estimated to be non-overlapping when the expansion factor is artificially removed. (D) Applying differentially expressed analysis to the 77,906 non-overlapping transcripts reveals that the number of proximity-induced genes (super resolution) is expected to decrease dramatically without expansion (low resolution).

Upregulated genes when comparing non-tumor cell types to tumor cell types

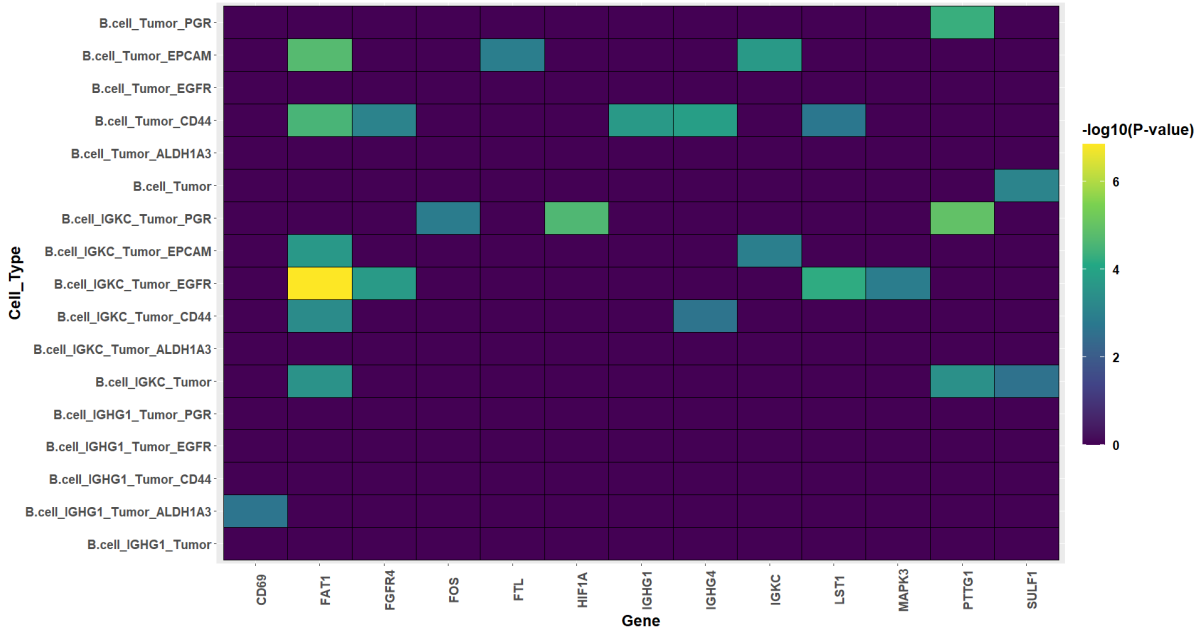


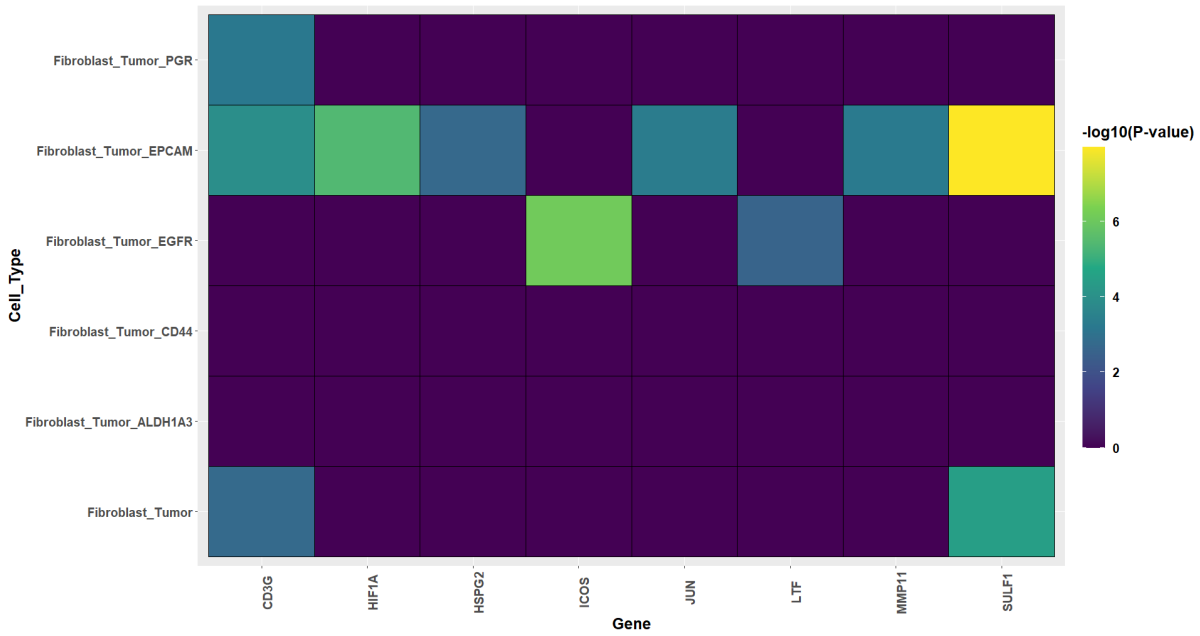


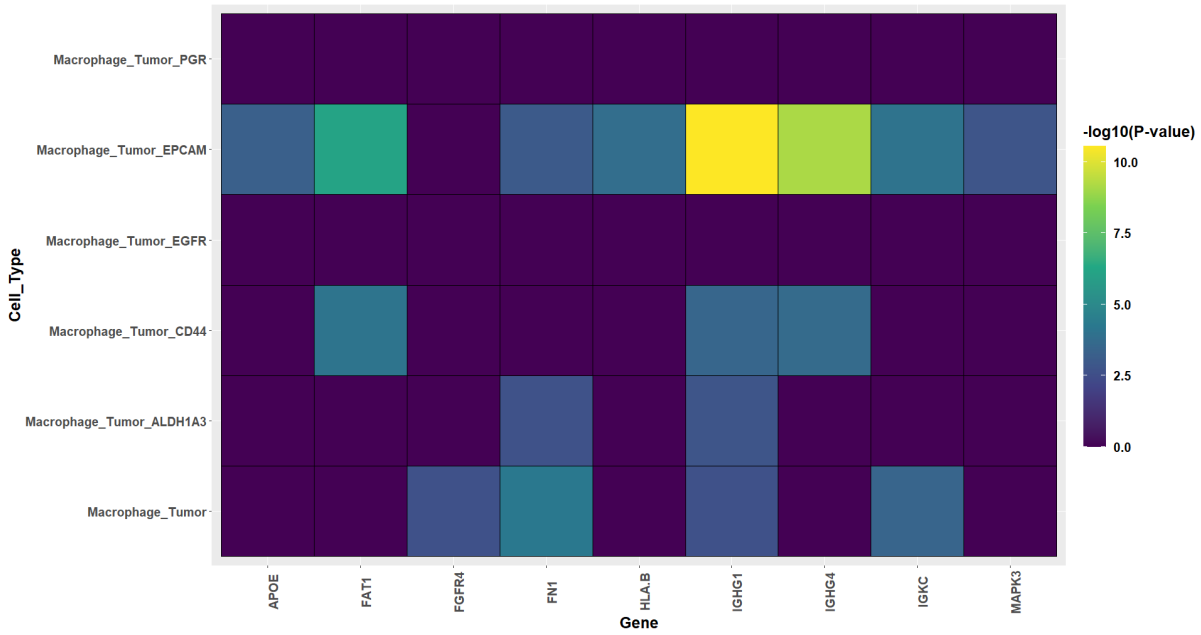

Macrophage_Tumor_PGR
Macrophage_Tumor_EPCAM
Macrophage_Tumor_EGFR
Macrophage_Tumor_CD44
Macrophage_Tumor_ALDH1A3
Macrophage_Tumor
Cell_Type
CDKN2A
HIF1A
IL7R
LYZ
RPSA
SULF1
TFF1
XBP1
Gene
-log10(P-value)
5
4
3
2
1
0

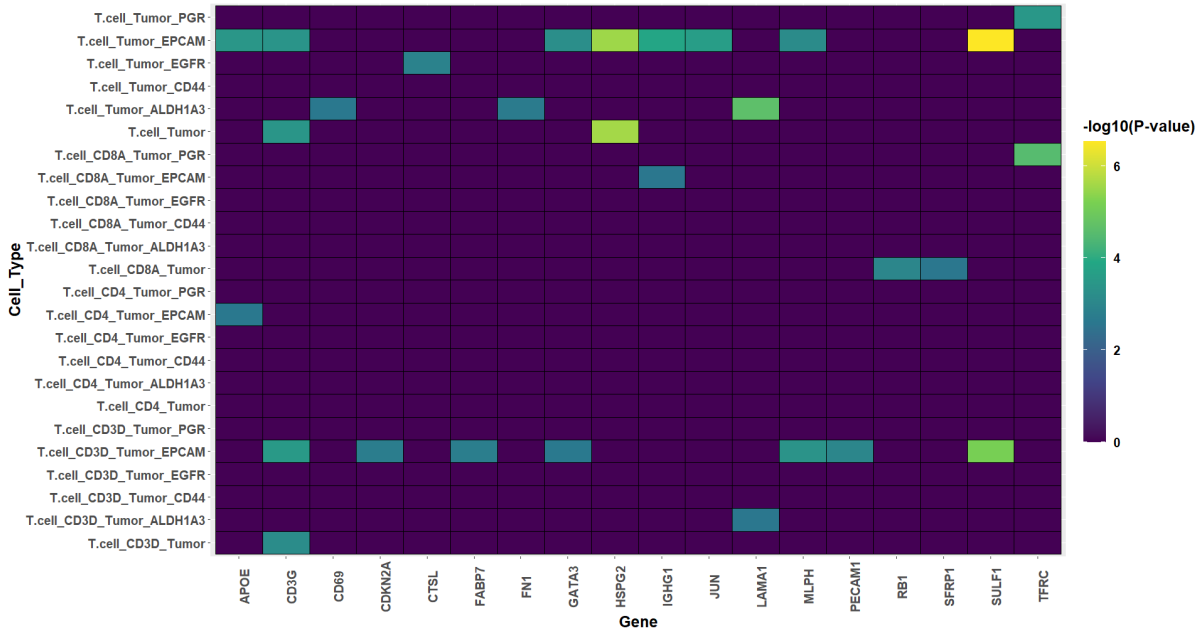

T.cell_Tumor_PGR
T.cell_Tumor_EPCAM
T.cell_Tumor_EGFR
T.cell_Tumor_CD44
T.cell_Tumor_ALDH1A3
T.cell_Tumor
T.cell_CD8A_Tumor_PGR
T.cell_CD8A_Tumor_EPCAM
T.cell_CD8A_Tumor_EGFR
T.cell_CD8A_Tumor_CD44
T.cell_CD8A_Tumor_ALDH1A3
T.cell_CD8A_Tumor
T.cell_CD4_Tumor_PGR
T.cell_CD4_Tumor_EPCAM
T.cell_CD4_Tumor_EGFR
T.cell_CD4_Tumor_CD44
T.cell_CD4_Tumor_ALDH1A3
T.cell_CD4_Tumor
T.cell_CD3D_Tumor_PGR
T.cell_CD3D_Tumor_EPCAM
T.cell_CD3D_Tumor_EGFR
T.cell_CD3D_Tumor_CD44
T.cell_CD3D_Tumor_ALDH1A3
T.cell_CD3D_Tumor
Cell_Type
ACTA2
ANLN
BICC1
C1QA
CCND1
CD63
CD69
CD74
CDK4
CDK6
CTSL
FAP
FASN
FOS
FOXP3
FTL
GNG11
HLA.B
HSPB1
JUN
KIF23
KRAS
KRT8
LAMA1
LDB2
LGALS2
LYZ
MLPH
MYL6
MYL9
PLEK
PLVAP
PTPRB
RPSA
S100A14
S100A9
SDC1
SNAI2
SOX10
SULF1
TFF1
TFF2
TFRC
THY1
TMSB4X
TPM2
TSPAN1
XBP1
ZNF571
Gene
-log10(P-value)
10
5
0

Downregulated genes when comparing non-tumor cell types to tumor cell types

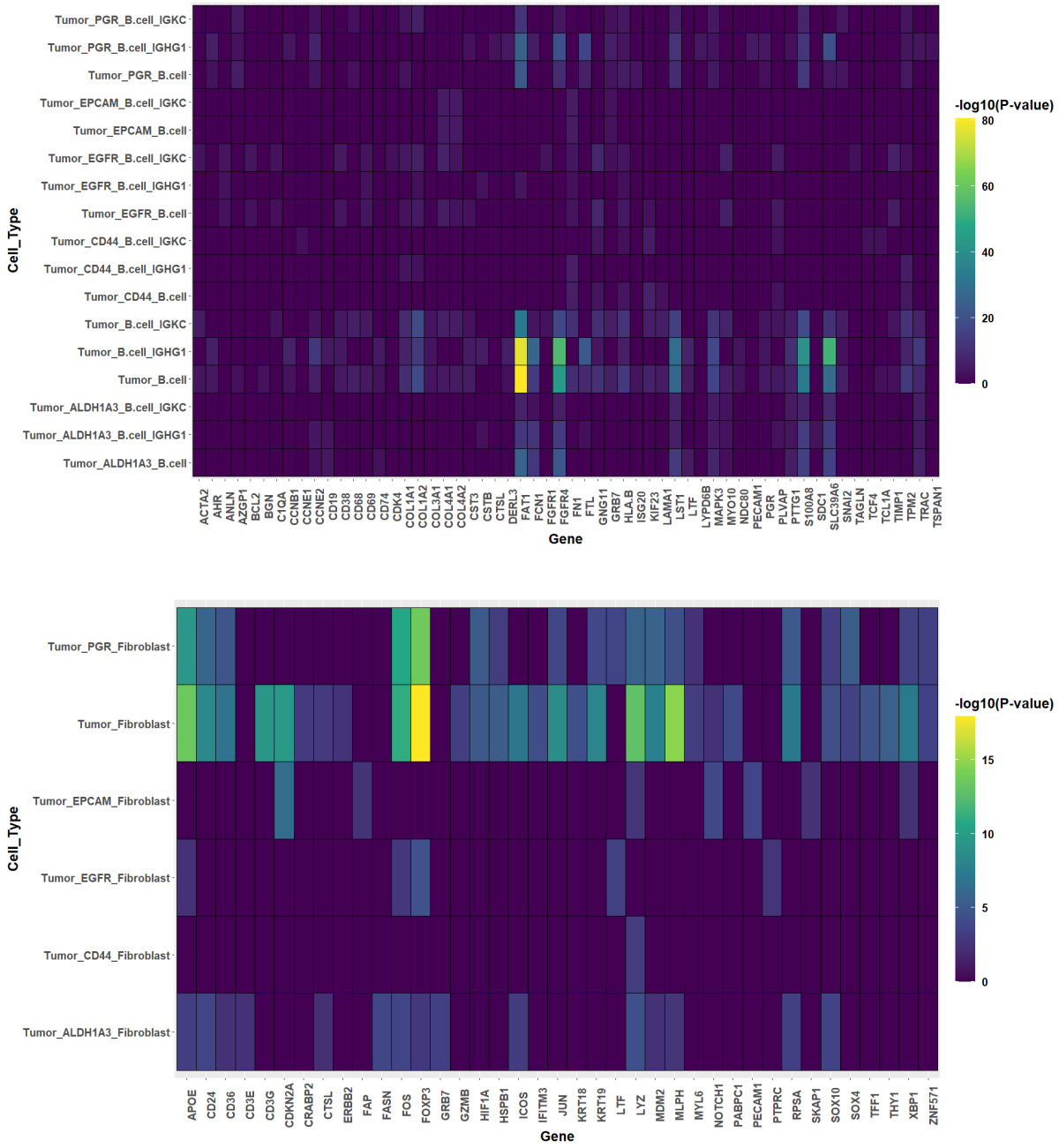


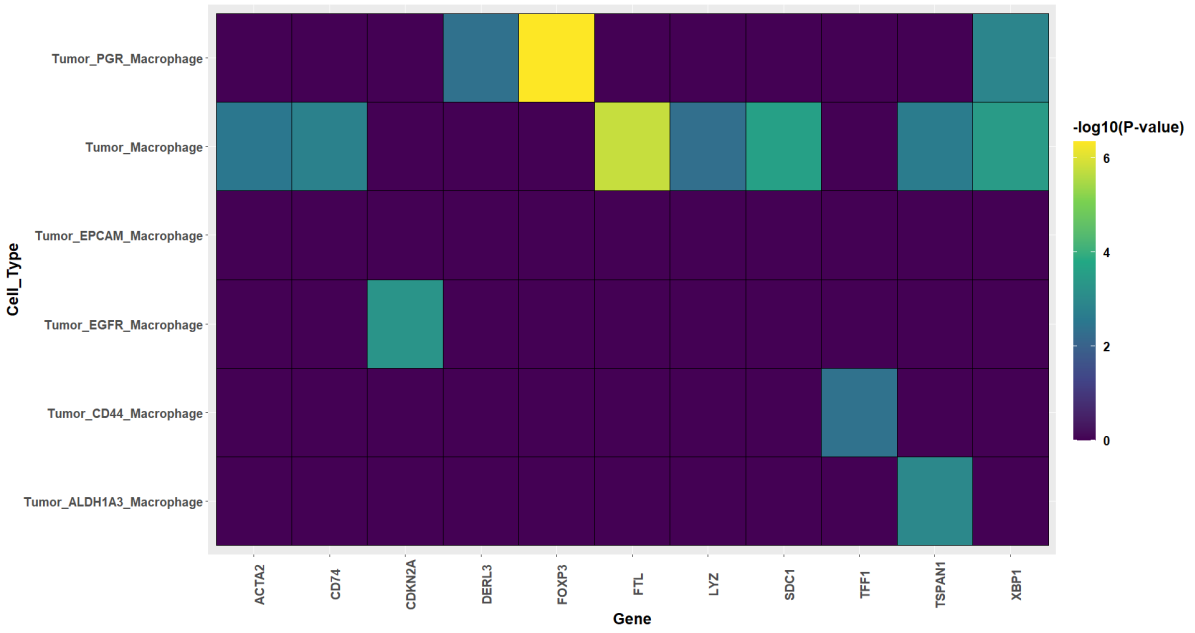


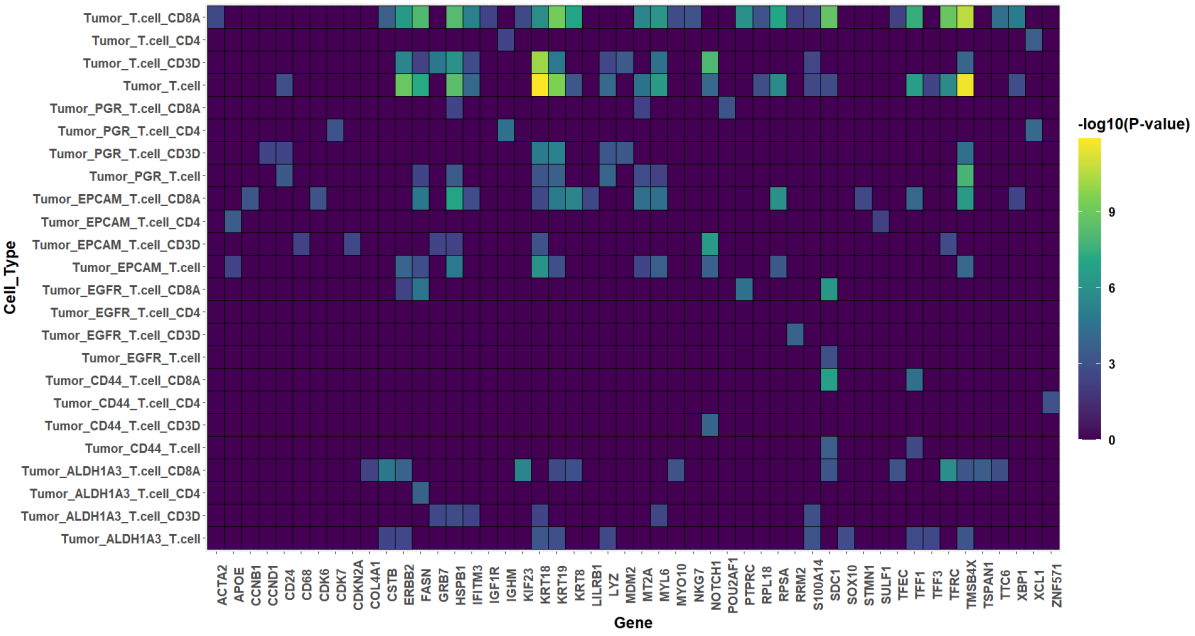

Macrophage_Tumor_PGR
Macrophage_Tumor_EPCAM
Macrophage_Tumor_EGFR
Macrophage_Tumor_CD44
Macrophage_Tumor_ALDH1A3
Macrophage_Tumor
Cell_Type
APOE
FAT1
FGFR4
FN1
HLA.B
IGHG1
IGHG4
IGKC
MAPK3
Gene
-log10(P-value)
10.0
7.5
5.0
2.5
0.0


T.cell_Tumor_PGR
T.cell_Tumor_EPCAM
T.cell_Tumor_EGFR
T.cell_Tumor_CD44
T.cell_Tumor_ALDH1A3
T.cell_Tumor
T.cell_CD8A_Tumor_PGR
T.cell_CD8A_Tumor_EPCAM
T.cell_CD8A_Tumor_EGFR
T.cell_CD8A_Tumor_CD44
T.cell_CD8A_Tumor_ALDH1A3
T.cell_CD8A_Tumor
T.cell_CD4_Tumor_PGR
T.cell_CD4_Tumor_EPCAM
T.cell_CD4_Tumor_EGFR
T.cell_CD4_Tumor_CD44
T.cell_CD4_Tumor_ALDH1A3
T.cell_CD4_Tumor
T.cell_CD3D_Tumor_PGR
T.cell_CD3D_Tumor_EPCAM
T.cell_CD3D_Tumor_EGFR
T.cell_CD3D_Tumor_CD44
T.cell_CD3D_Tumor_ALDH1A3
T.cell_CD3D_Tumor
Cell_Type
APOE
CD3G
CD69
CDKN2A
CTSL
FABP7
FN1
GATA3
HSPG2
IGHG1
JUN
LAMA1
MLPH
PECAM1
RB1
SFRP1
SULF1
TFRC
Gene
-log10(P-value)
6
4
2
0

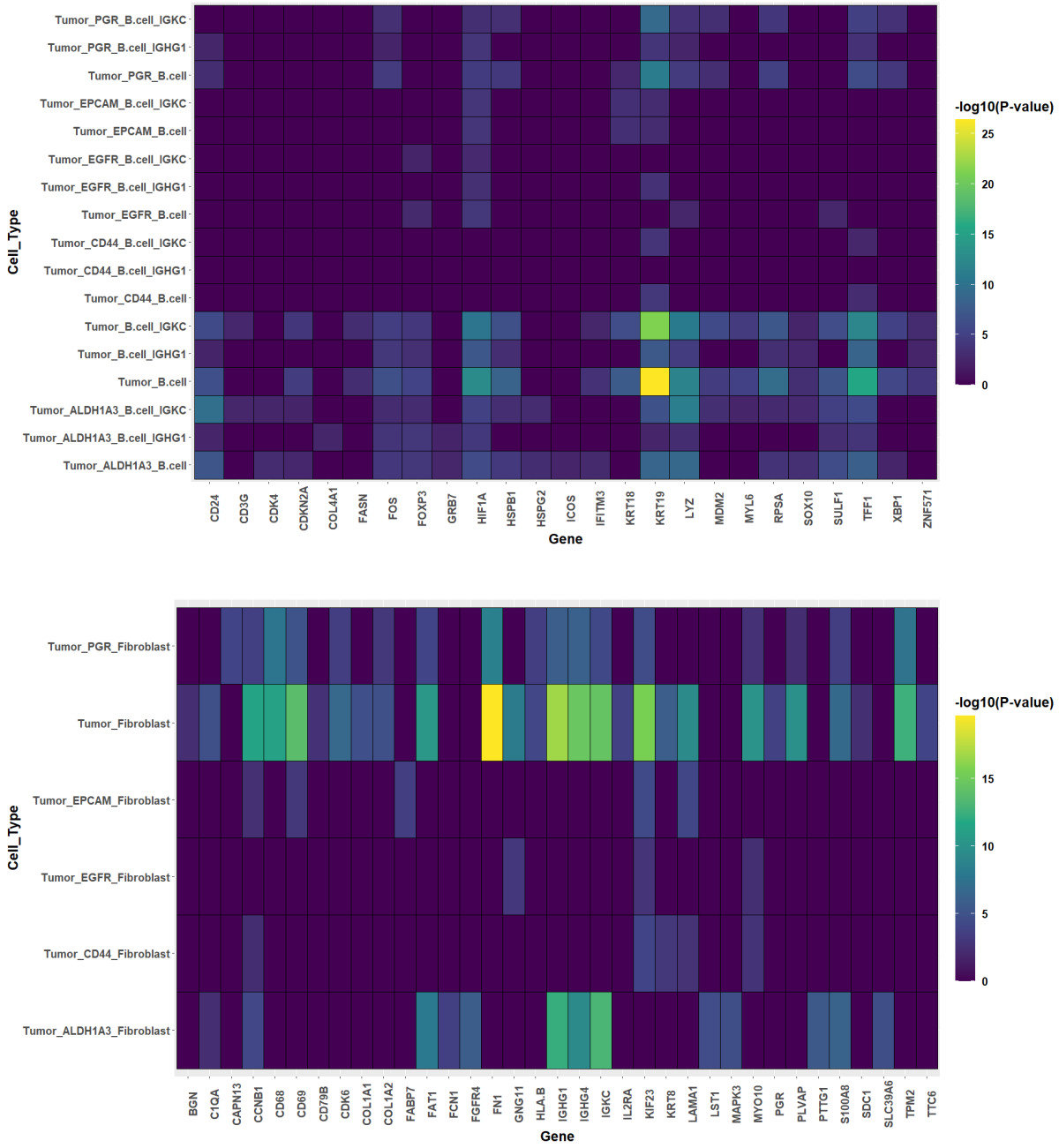
Upregulated genes when comparing tumor cell types to non-tumor cell types




Tumor_PGR_Macrophage
Tumor_Macrophage
Tumor_EPCAM_Macrophage
Tumor_EGFR_Macrophage
Tumor_CD44_Macrophage
Tumor_ALDH1A3_Macrophage
Cell_Type
ACTA2
CD74
CDKN2A
DERL3
FOXP3
FTL
LYZ
SDC1
TFF1
TSPAN1
XBP1
Gene
-log10(P-value)
6
4
2
0


Tumor_T.cell_CD8A
Tumor_T.cell_CD4
Tumor_T.cell_CD3D
Tumor_T.cell
Tumor_PGR_T.cell_CD8A
Tumor_PGR_T.cell_CD4
Tumor_PGR_T.cell_CD3D
Tumor_PGR_T.cell
Tumor_EPCAM_T.cell_CD8A
Tumor_EPCAM_T.cell_CD4
Tumor_EPCAM_T.cell_CD3D
Tumor_EPCAM_T.cell
Tumor_EGFR_T.cell_CD8A
Tumor_EGFR_T.cell_CD4
Tumor_EGFR_T.cell_CD3D
Tumor_EGFR_T.cell
Tumor_CD44_T.cell_CD8A
Tumor_CD44_T.cell_CD4
Tumor_CD44_T.cell_CD3D
Tumor_CD44_T.cell
Tumor_ALDH1A3_T.cell_CD8A
Tumor_ALDH1A3_T.cell_CD4
Tumor_ALDH1A3_T.cell_CD3D
Tumor_ALDH1A3_T.cell
Cell_Type
ACTA2
APOE
CCNB1
CCND1
CD24
CD68
CDK6
CDK7
CDKN2A
COL4A1
CSTB
ERBB2
FASN
GRB7
HSPB1
IFITM3
IGF1R
IGHM
KIF23
KRT18
KRT19
KRT8
LILRB1
LYZ
MDM2
MT2A
MYL6
MYO10
NKG7
NOTCH1
POU2AF1
PTPRC
RPL18
RPSA
RRM2
S100A14
SDC1
SOX10
STMN1
SULF1
TFEC
TFF1
TFF3
TFRC
TMSB4X
TSPAN1
TTC6
XBP1
XCL1
ZNF571
Gene
-log10(P-value)
9
6
3
0

Downregulated genes when comparing tumor cell types to non-tumor cell types



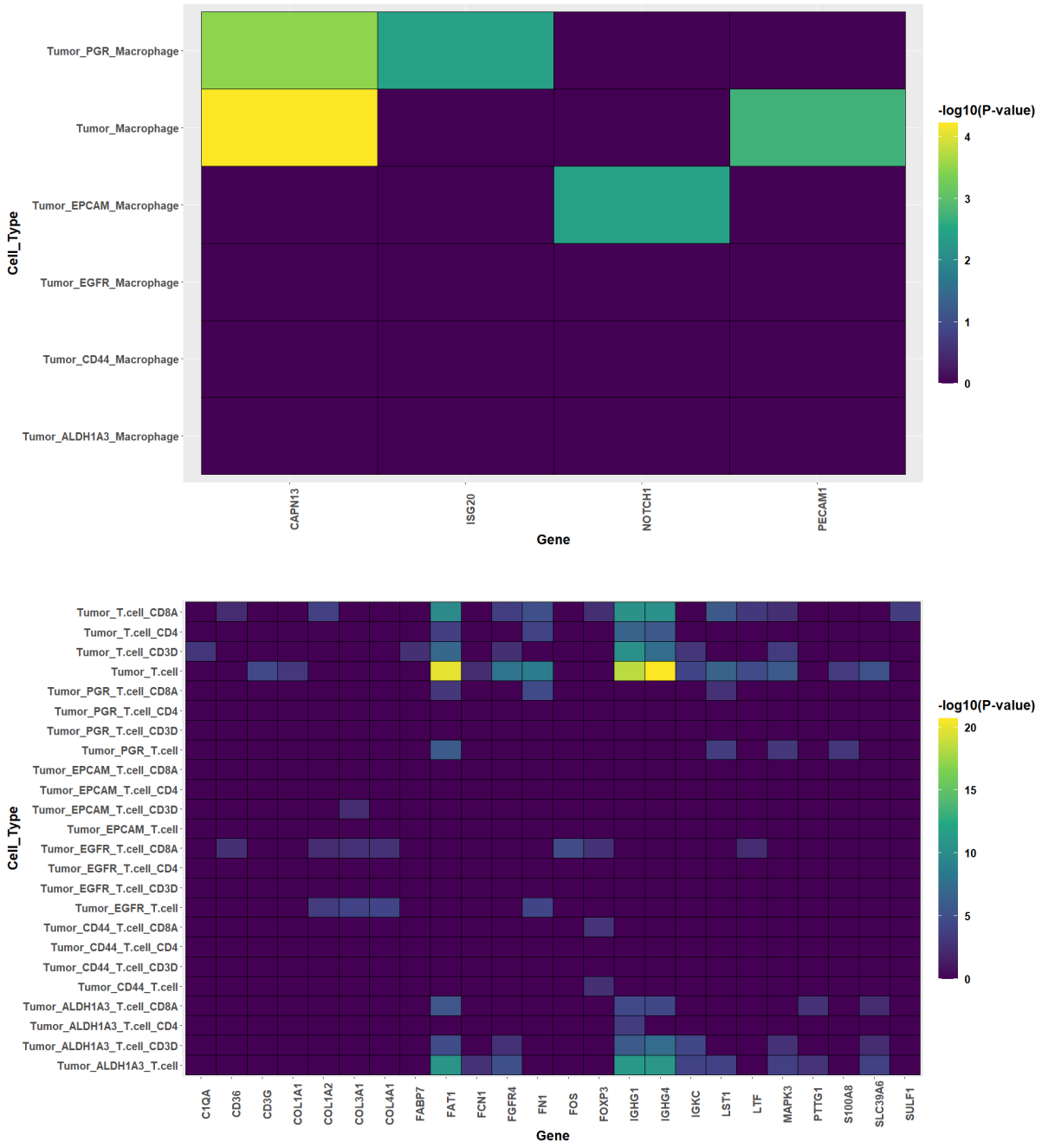


Figure S14. Heatmap showing the p-values of all the statistically significant differentially expressed genes (FDR<0.1) detected in 108 combinations of non-tumor and tumor cell types. Genes with FDR≥0.1 were given a value of 0 (black squares). Each row gives the name of a cell type X and a cell type Y, and we detected the genes upregulated or downregulated when comparing the cells in X that are proximal to Y, versus cells in X not proximal to Y. For example, in the row marked as Tumor_PGR_Tcell_CD8A, we detected genes that are upregulated or downregulated when PGR-positive tumor cells are in proximity to CD8A-positive T cells, versus PGR-positive tumor cells which are not proximal to CD8A-positive T cells. In that example, the genes detected as upregulated were *HSPB1*, *MT2A*, *POU2AF1*, and the genes detected as downregulated were *FAT1* and *FN1*.

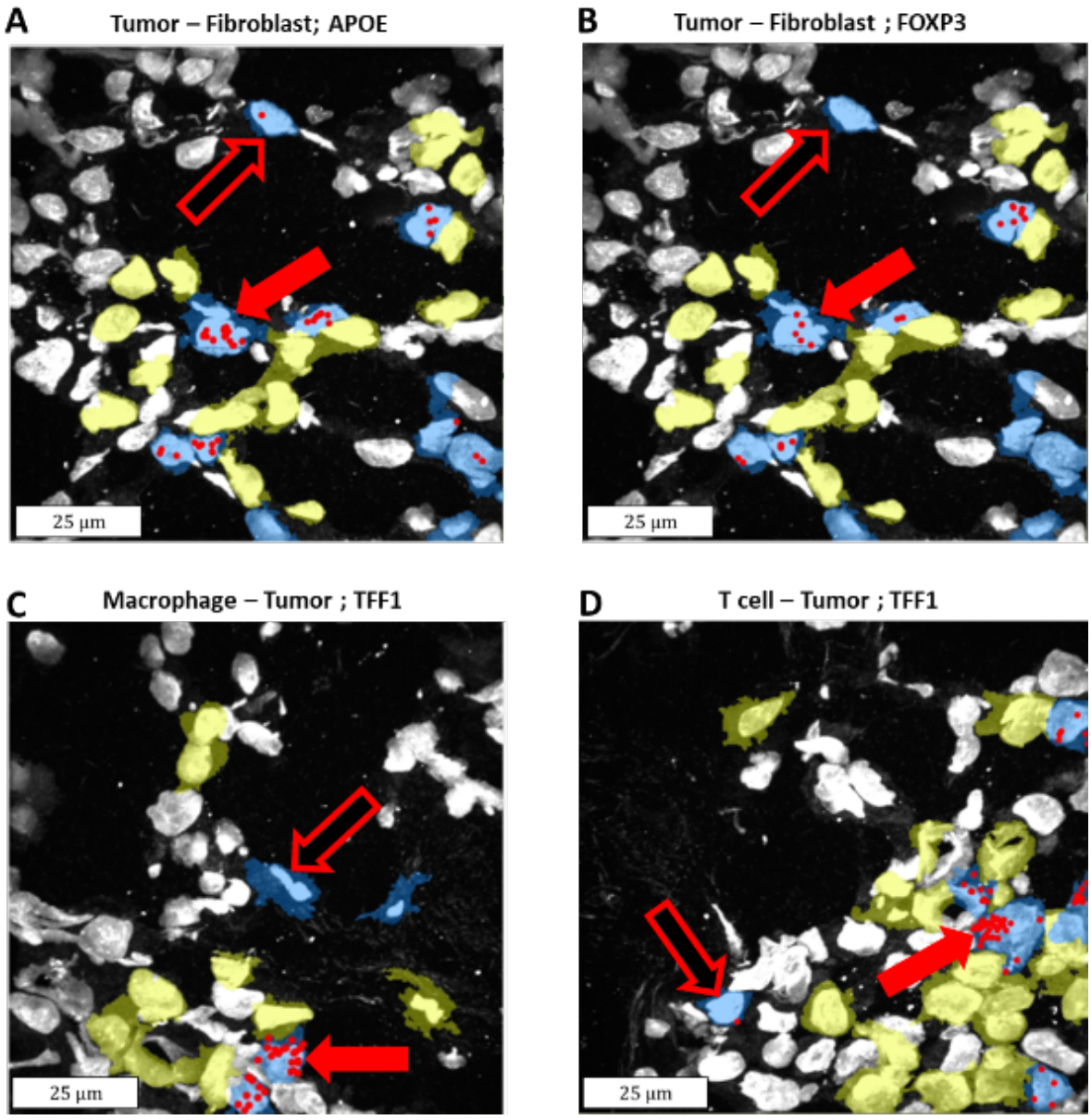


Figure S15. Additional examples of genes identified as induced by proximity between different cell types. The detected genes were upregulated in the subset of X cells which are proximal to Y cells, compared to X cells which are not proximal to the Y cells. Sequencing reads locations (red spots) of four induced genes are overlaid on the DAPI staining of the nuclei, as well as the segmentation of the X cell type (blue) and the Y cell type (yellow). In each example, only the X cell type and the Y cell type segmentations are presented. The cell bodies were detected using InSituSeg, and the cell types were identified using clustering of the gene expression profiles. Genes upregulated in X cells due to proximity to Y cells have more red spots when proximal to Y cells (exemplars in full red arrows) versus X cells which are not proximal (exemplars in hollow red arrows). A) The gene *APOE* was detected by differential expression (DE), when examining all tumor cells that are proximal to fibroblast cells. B) The gene *FOXP3* was detected by DE, when examining all tumor cells that are proximal to fibroblast

cells. C) the gene *TFF1* was detected by DE, by machine learning (ML) and by matrix factorization (MF), when examining all macrophage cells that are proximal to all tumor cells. D) the gene *TFF1* was detected by DE and by MF, when examining all T cells that are proximal to all tumor cells. Each panel shows a max projection of a subset region from the biopsy, acquired with a 40X objective, 100 x 100 microns in size (before expansion). DE was performed with DeSeq2 (Love, Huber, and Anders 2014), ML with CatBoost(Dorogush, Ershov, and Gulin 2018) (Dorogush, Ershov, and Gulin 2018), and MF with cNMF (Kotliar *et al.* 2019). Permutation analysis was performed on all methods to assess statistical significance.

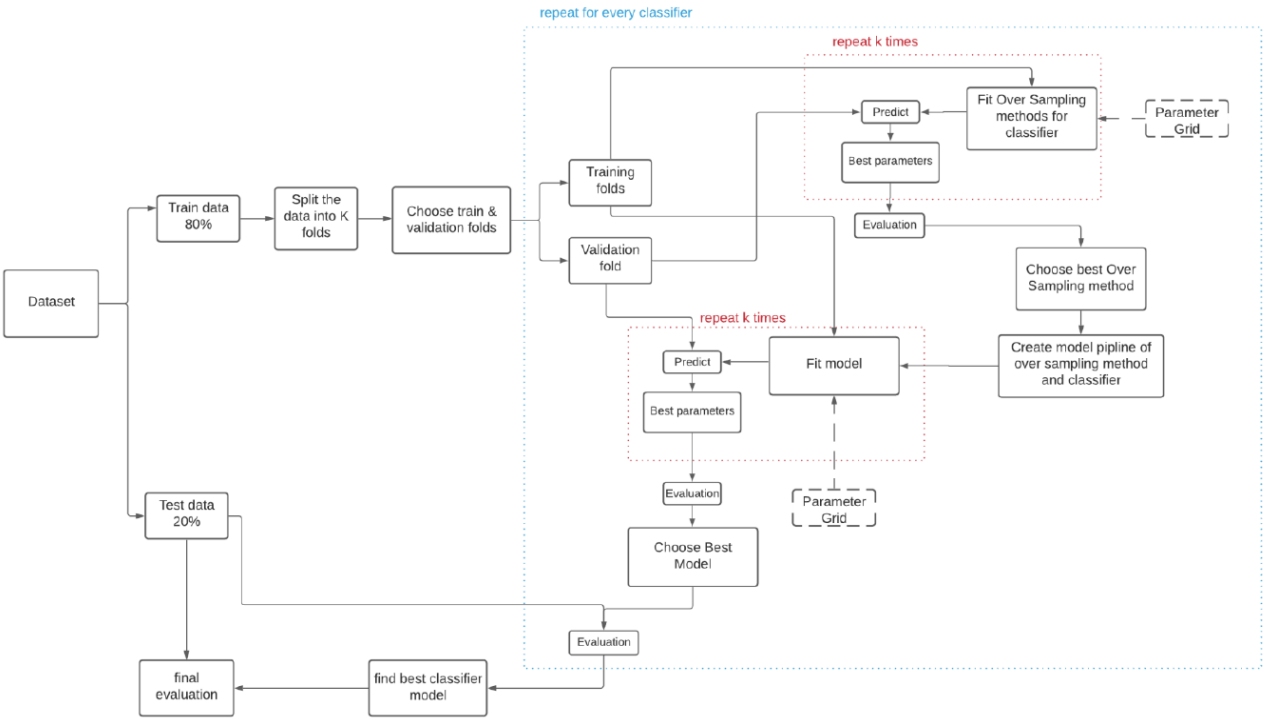


Figure S16. General scheme of the machine learning analysis process.

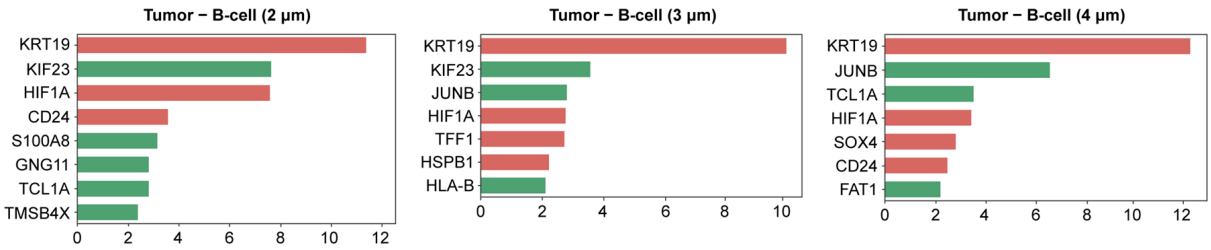
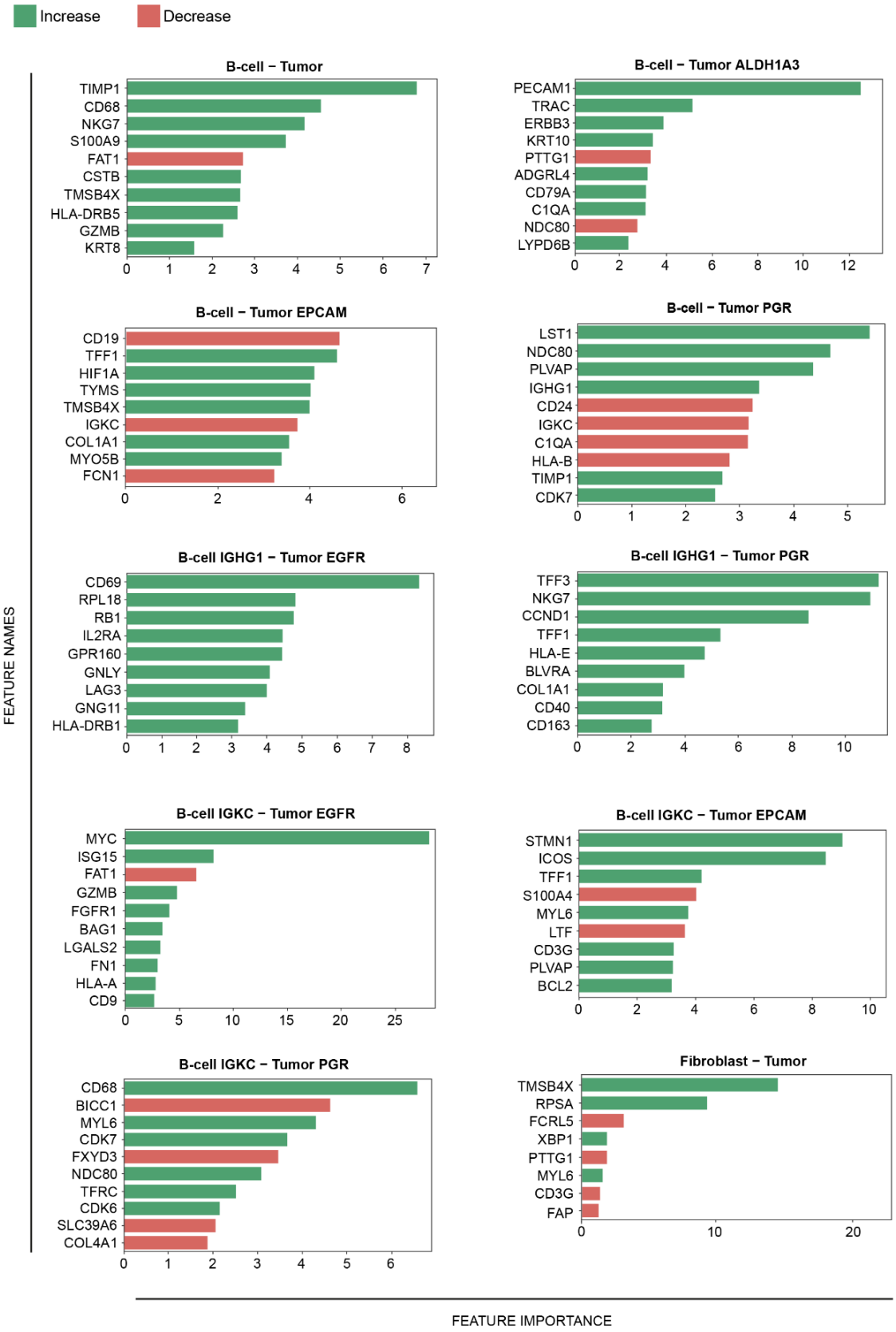
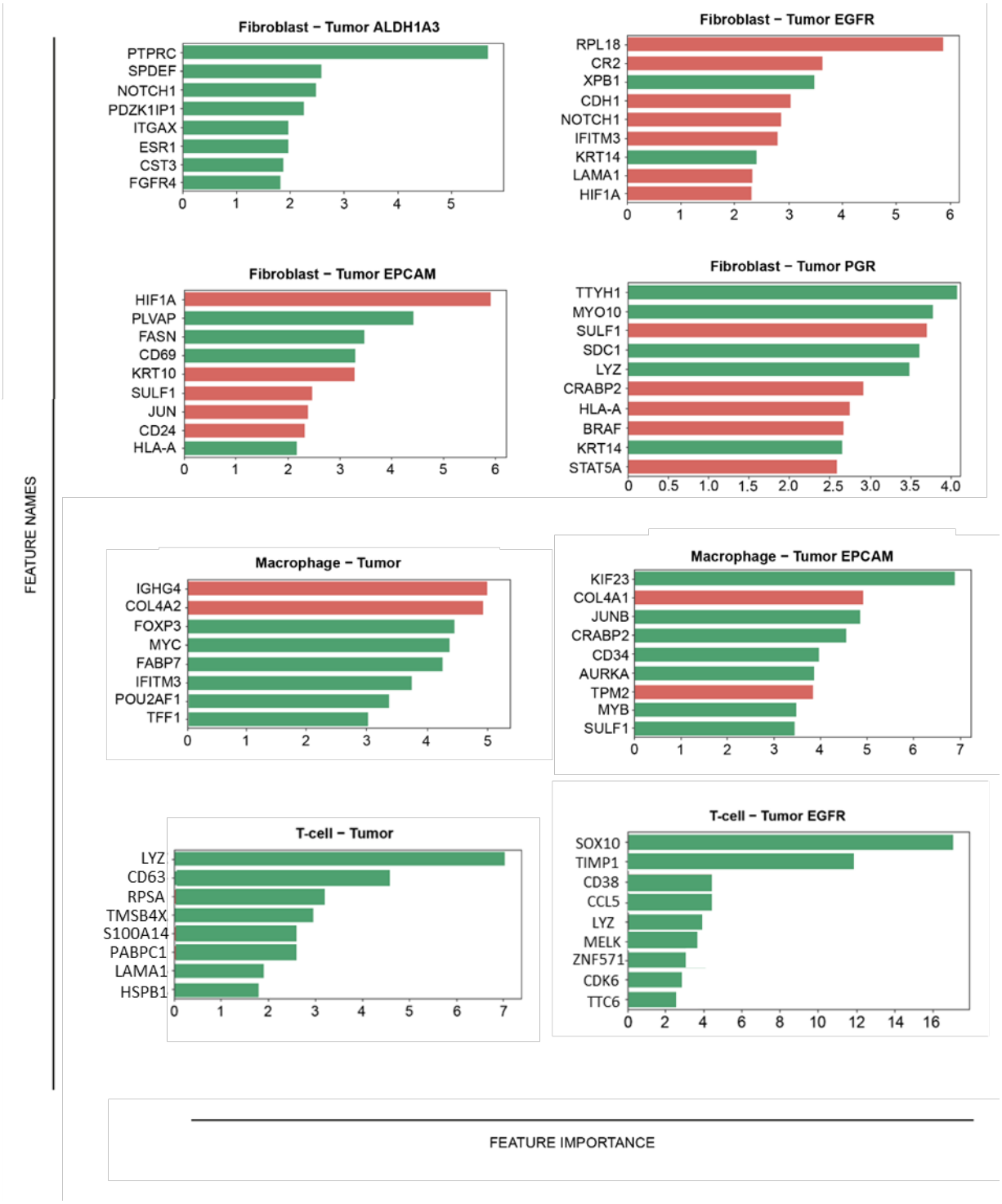
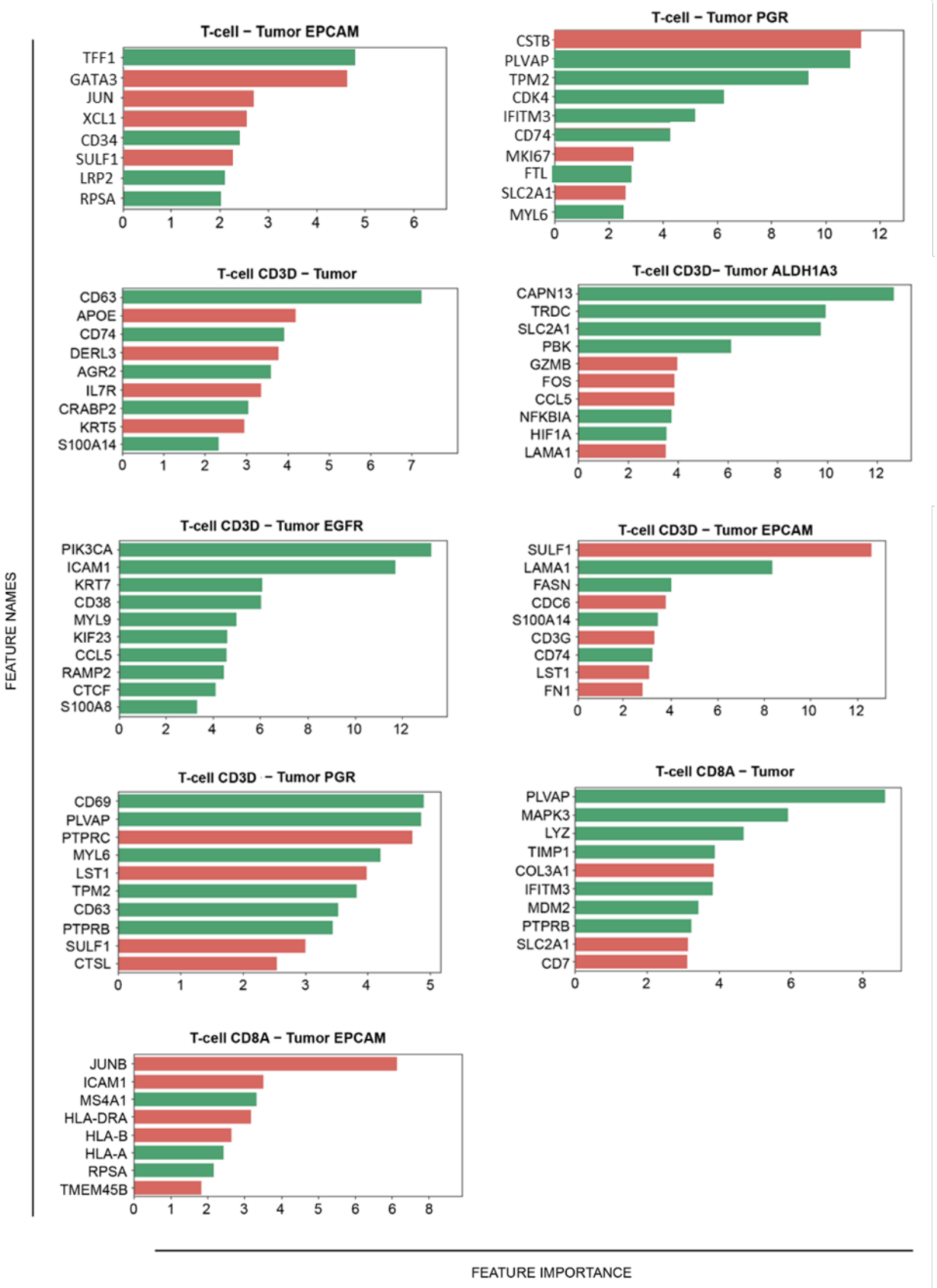


Figure S17. Comparing the features (genes) detected using machine learning classification as a function of the distance that defines proximity between cells. We detected the genes with the highest influence on the classification between: (a) tumor cells (from all types) that are in proximity to B cells (from all types), and (b) tumor cells (from all types) which are not proximal to B cells (from all types). This classification (middle panel) was performed with a distance of 3 microns (before expansion) as the cutoff for proximal cells. To check the robustness of the results, we recalculated the machine learning classification using 2 microns (left panel) and 4 microns (right panel). The majority of the genes detected using 3 microns as the proximity cutoff are robust to changes in this cutoff. Note that some of these genes are increased (green bars) when the cells are in proximity and some are decreased (red bars).

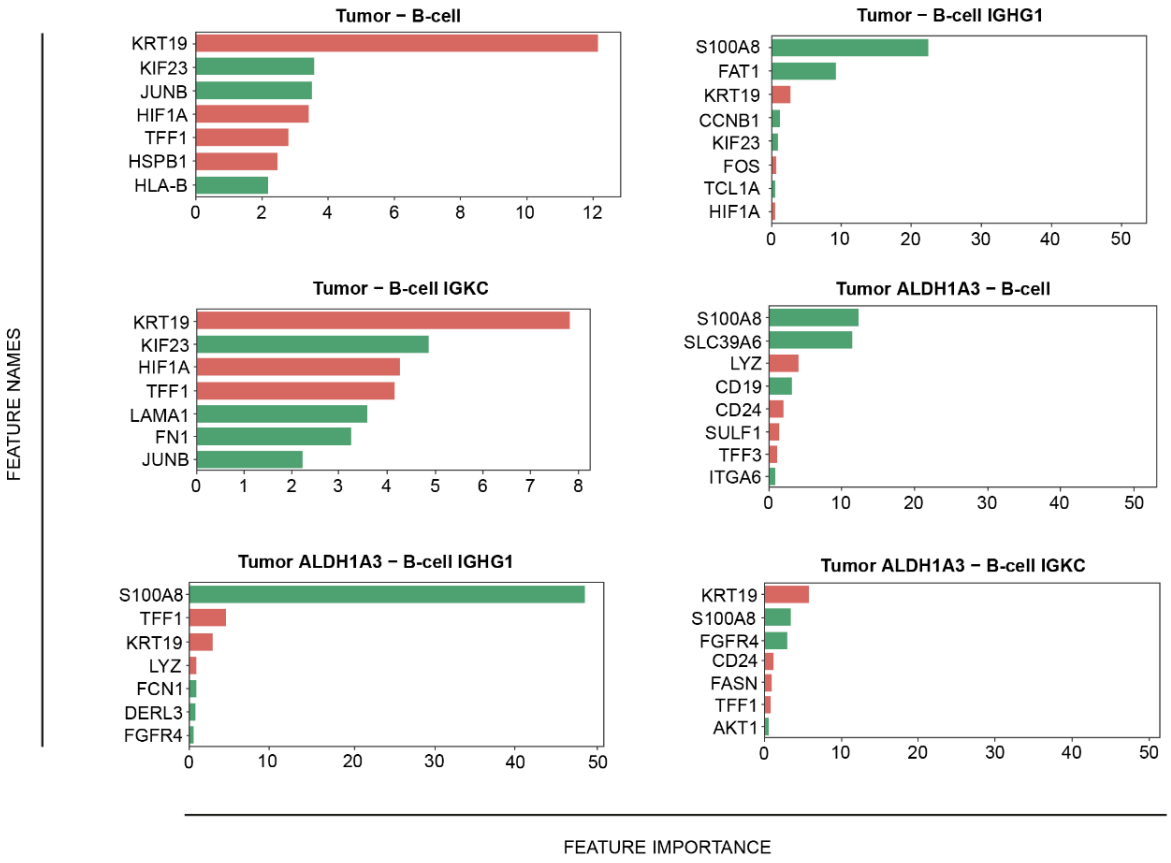
Genes detected using machine learning when comparing non-tumor cell types to tumor cell types

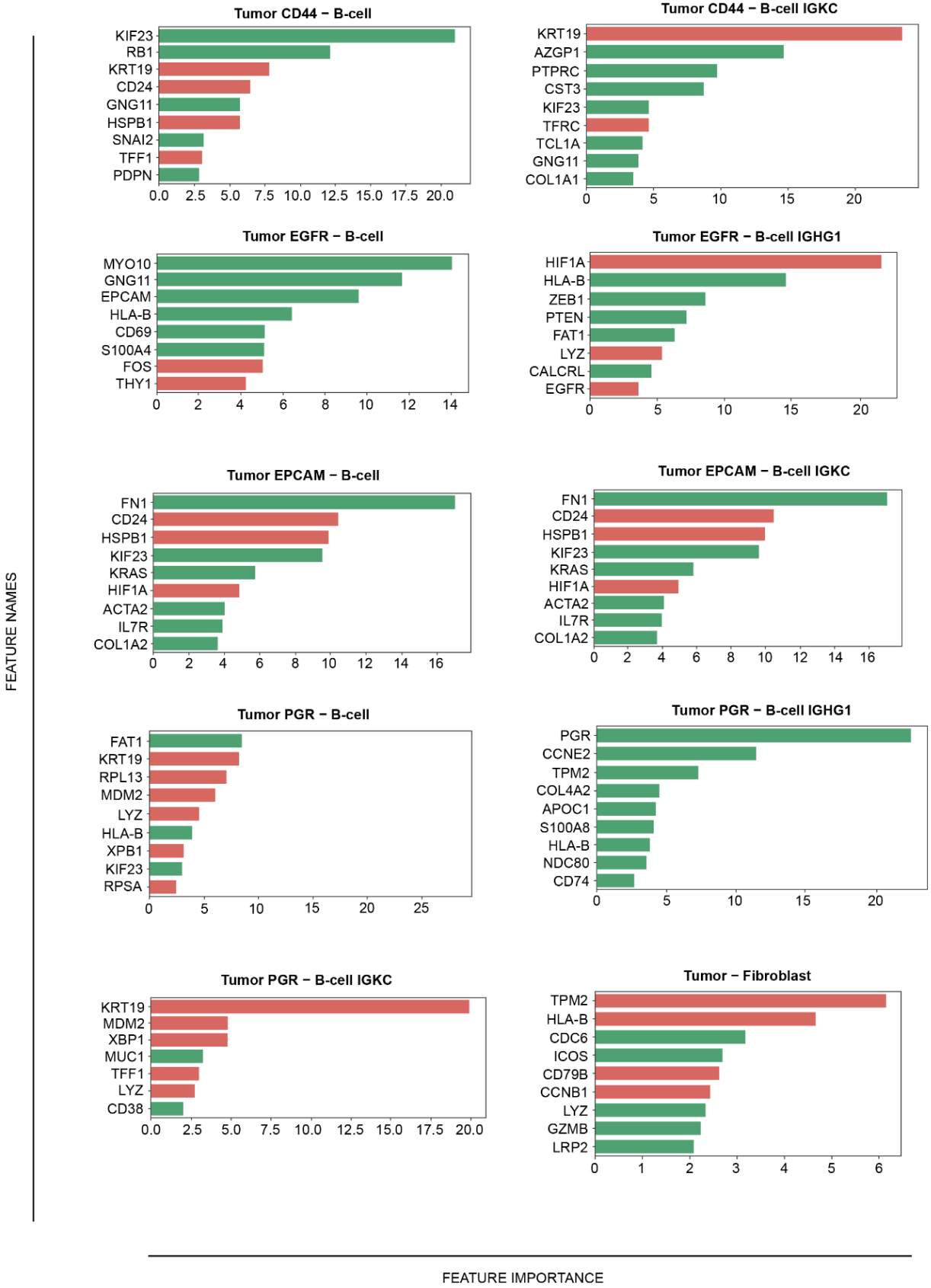


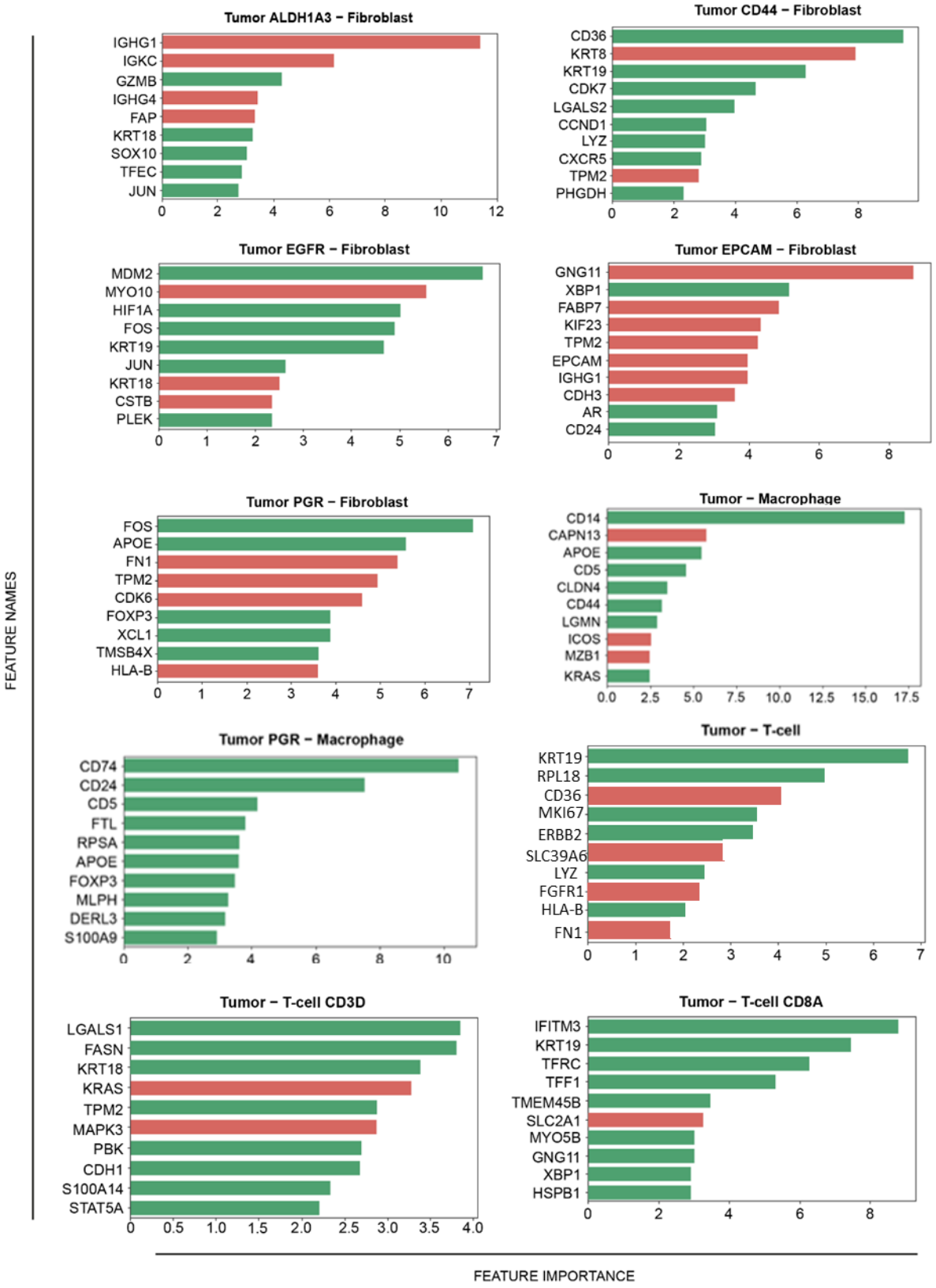

Fibroblast − Tumor ALDH1A3
PTPRC
SPDEF
NOTCH1
PDZK1IP1
ITGAX
ESR1
CST3
FGFR4
0 1 2 3 4 5
Fibroblast − Tumor EGFR
RPL18
CR2
XPB1
CDH1
NOTCH1
IFITM3
KRT14
LAMA1
HIF1A
0 1 2 3 4 5 6
Fibroblast − Tumor EPCAM
HIF1A
PLVAP
FASN
CD69
KRT10
SULF1
JUN
CD24
HLA-A
0 1 2 3 4 5 6
Fibroblast − Tumor PGR
TTYH1
MYO10
SULF1
SDC1
LYZ
CRABP2
HLA-A
BRAF
KRT14
STAT5A
0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Macrophage − Tumor
IGHG4
COL4A2
FOXP3
MYC
FABP7
IFITM3
POU2AF1
TFF1
0 1 2 3 4 5
Macrophage − Tumor EPCAM
KIF23
COL4A1
JUNB
CRABP2
CD34
AURKA
TPM2
MYB
SULF1
0 1 2 3 4 5 6 7
T-cell − Tumor
LYZ
CD63
RPSA
TMSB4X
S100A14
PABPC1
LAMA1
HSPB1
0 1 2 3 4 5 6 7
T-cell − Tumor EGFR
SOX10
TIMP1
CD38
CCL5
LYZ
MELK
ZNF571
CDK6
TTC6
0 2 4 6 8 10 12 14 16
FEATURE NAMES
FEATURE IMPORTANCE


T-cell – Tumor EPCAM
TFF1
GATA3
JUN
XCL1
CD34
SULF1
LRP2
RPSA
0 1 2 3 4 5 6
T-cell – Tumor PGR
CSTB
PLVAP
TPM2
CDK4
IFITM3
CD74
MKI67
FTL
SLC2A1
MYL6
0 2 4 6 8 10 12
T-cell CD3D – Tumor
CD63
APOE
CD74
DERL3
AGR2
IL7R
CRABP2
KRT5
S100A14
0 1 2 3 4 5 6 7
T-cell CD3D– Tumor ALDH1A3
CAPN13
TRDC
SLC2A1
PBK
GZMB
FOS
CCL5
NFKBIA
HIF1A
LAMA1
0 2 4 6 8 10 12
T-cell CD3D – Tumor EGFR
PIK3CA
ICAM1
KRT7
CD38
MYL9
KIF23
CCL5
RAMP2
CTCF
S100A8
0 2 4 6 8 10 12
T-cell CD3D – Tumor EPCAM
SULF1
LAMA1
FASN
CDC6
S100A14
CD3G
CD74
LST1
FN1
0 2 4 6 8 10 12
T-cell CD3D – Tumor PGR
CD69
PLVAP
PTPRC
MYL6
LST1
TPM2
CD63
PTPRB
SULF1
CTSL
0 1 2 3 4 5
T-cell CD8A – Tumor
PLVAP
MAPK3
LYZ
TIMP1
COL3A1
IFITM3
MDM2
PTPRB
SLC2A1
CD7
0 2 4 6 8
T-cell CD8A – Tumor EPCAM
JUNB
ICAM1
MS4A1
HLA-DRA
HLA-B
HLA-A
RPSA
TMEM45B
0 1 2 3 4 5 6 8
FEATURE NAMES
FEATURE IMPORTANCE

Genes detected using machine learning when comparing tumor cell types to non-tumor cell types




Tumor CD44 − B-cell
KIF23
RB1
KRT19
CD24
GNG11
HSPB1
SNAI2
TFF1
PDPN
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
Tumor CD44 − B-cell IGKC
KRT19
AZGP1
PTPRC
CST3
KIF23
TFRC
TCL1A
GNG11
COL1A1
0 5 10 15 20
Tumor EGFR − B-cell
MYO10
GNG11
EPCAM
HLA-B
CD69
S100A4
FOS
THY1
0 2 4 6 8 10 12 14
Tumor EGFR − B-cell IGHG1
HIF1A
HLA-B
ZEB1
PTEN
FAT1
LYZ
CALCRL
EGFR
0 5 10 15 20
Tumor EPCAM − B-cell
FN1
CD24
HSPB1
KIF23
KRAS
HIF1A
ACTA2
IL7R
COL1A2
0 2 4 6 8 10 12 14 16
Tumor EPCAM − B-cell IGKC
FN1
CD24
HSPB1
KIF23
KRAS
HIF1A
ACTA2
IL7R
COL1A2
0 2 4 6 8 10 12 14 16
Tumor PGR − B-cell
FAT1
KRT19
RPL13
MDM2
LYZ
HLA-B
XPB1
KIF23
RPSA
0 5 10 15 20 25
Tumor PGR − B-cell IGHG1
PGR
CCNE2
TPM2
COL4A2
APOC1
S100A8
HLA-B
NDC80
CD74
0 5 10 15 20
Tumor PGR − B-cell IGKC
KRT19
MDM2
XBP1
MUC1
TFF1
LYZ
CD38
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
Tumor − Fibroblast
TPM2
HLA-B
CDC6
ICOS
CD79B
CCNB1
LYZ
GZMB
LRP2
0 1 2 3 4 5 6
FEATURE NAMES
FEATURE IMPORTANCE


Tumor ALDH1A3 – Fibroblast
IGHG1
IGKC
GZMB
IGHG4
FAP
KRT18
SOX10
TFEC
JUN
0 2 4 6 8 10 12
Tumor CD44 – Fibroblast
CD36
KRT8
KRT19
CDK7
LGALS2
CCND1
LYZ
CXCR5
TPM2
PHGDH
0 2 4 6 8
Tumor EGFR – Fibroblast
MDM2
MYO10
HIF1A
FOS
KRT19
JUN
KRT18
CSTB
PLEK
0 1 2 3 4 5 6 7
Tumor EPCAM – Fibroblast
GNG11
XBP1
FABP7
KIF23
TPM2
EPCAM
IGHG1
CDH3
AR
CD24
0 2 4 6 8
Tumor PGR – Fibroblast
FOS
APOE
FN1
TPM2
CDK6
FOXP3
XCL1
TMSB4X
HLA-B
0 1 2 3 4 5 6 7
Tumor – Macrophage
CD14
CAPN13
APOE
CD5
CLDN4
CD44
LGMN
ICOS
MZB1
KRAS
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Tumor PGR – Macrophage
CD74
CD24
CD5
FTL
RPSA
APOE
FOXP3
MLPH
DERL3
S100A9
0 2 4 6 8 10
Tumor – T-cell
KRT19
RPL18
CD36
MKI67
ERBB2
SLC39A6
LYZ
FGFR1
HLA-B
FN1
0 1 2 3 4 5 6 7
Tumor – T-cell CD3D
LGALS1
FASN
KRT18
KRAS
TPM2
MAPK3
PBK
CDH1
S100A14
STAT5A
0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Tumor – T-cell CD8A
IFITM3
KRT19
TFRC
TFF1
TMEM45B
SLC2A1
MYO5B
GNG11
XBP1
HSPB1
0 2 4 6 8
FEATURE NAMES
FEATURE IMPORTANCE

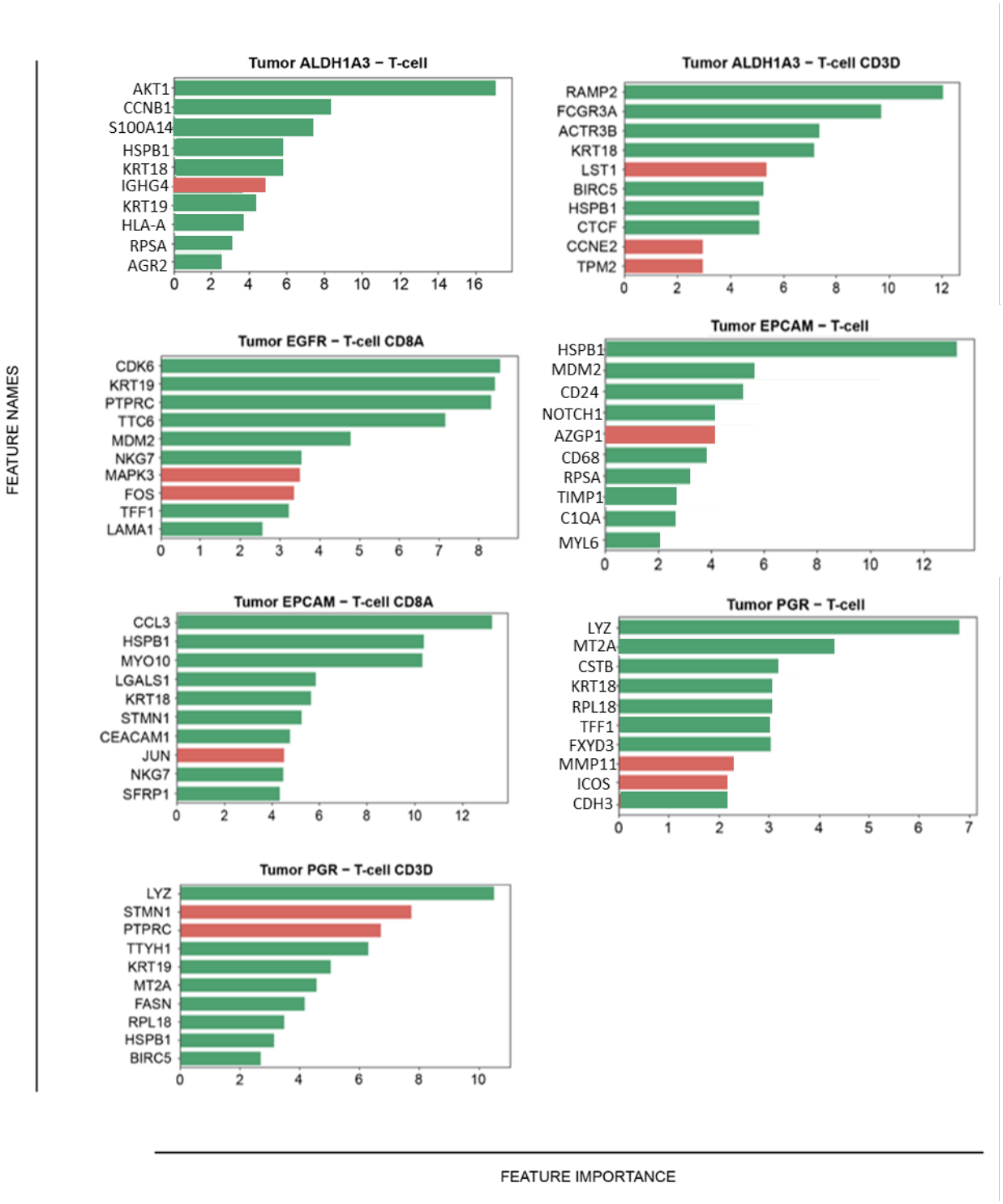


Figure S18. The 10 features (genes) that had the highest influence on the machine learning classification in all the combinations of non-tumor and tumor cell types. To avoid errors that result from inaccurate segmentation of two adjacent cells, we filtered differentially expressed genes detected in X cells if they are known cell markers for the Y cells. Only combinations with FDR<1e-4 are shown. For example, in the panel marked as 'Tumor EPCAM – T-cell CD8A', we detected the genes with the highest influence on the classification between: (a) EPCAM-positive tumor cells that are in proximity

to CD8A-positive T cells, and (b) EPCAM-positive tumor cells which are not proximal to CD8A-positive T cells. Note that some of these genes are increased (green bars) when the cells are in proximity, and some are decreased (red bars).

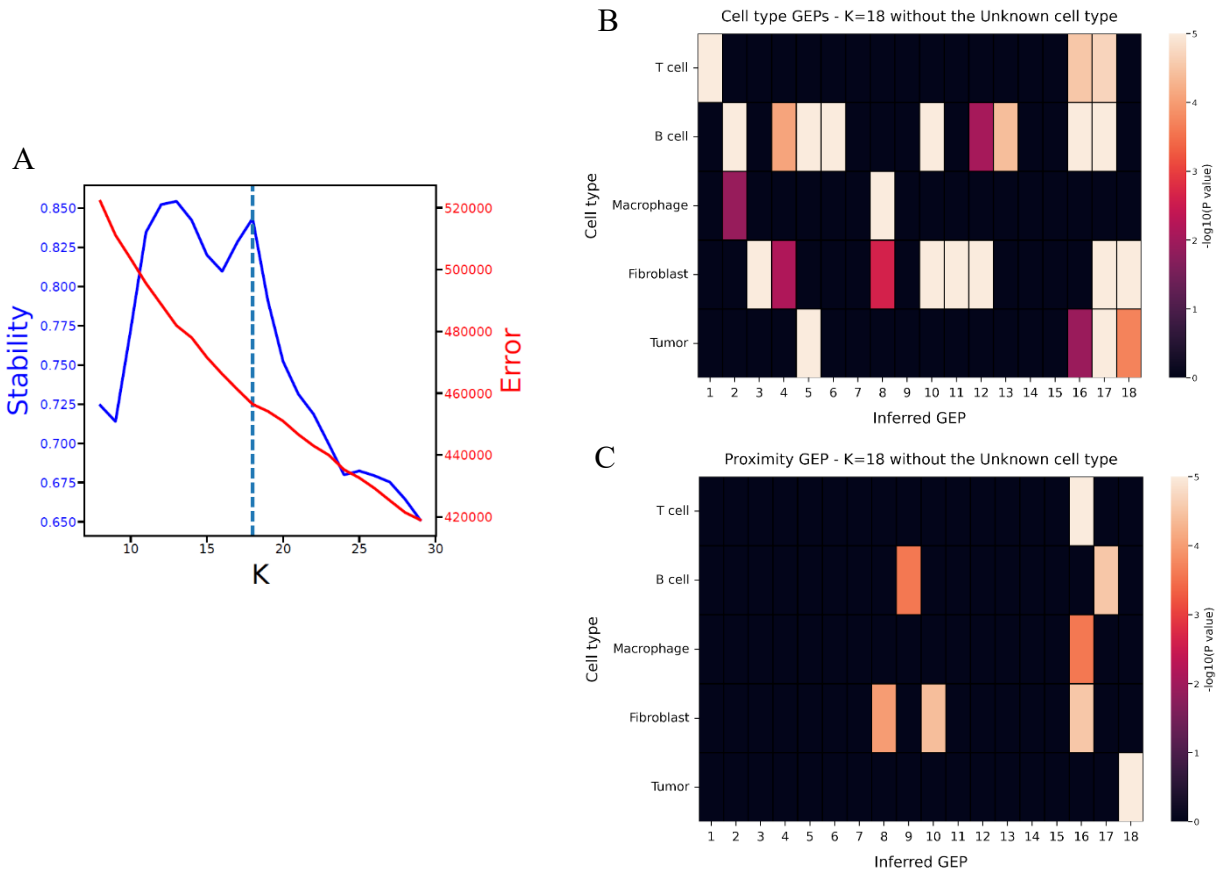


Figure S19. Cell type GEPs and proximity-related GEPs resulting from the cNMF analysis. A) The stability and error of the cNMF analysis as a function of K, the number of GEP components. B) Over-represented GEPs in specific cell types ('cell type' GEPs), and their corresponding p-values. C) Proximity-related GEPs, i.e., GEPs which are overexpressed or under expressed as a result of physical distance between cell types, and their corresponding p-values. In (B-C), the p-values were computed using permutation analysis, and only p-values with Benjamini-Hochberg FDR<0.05 are presented.

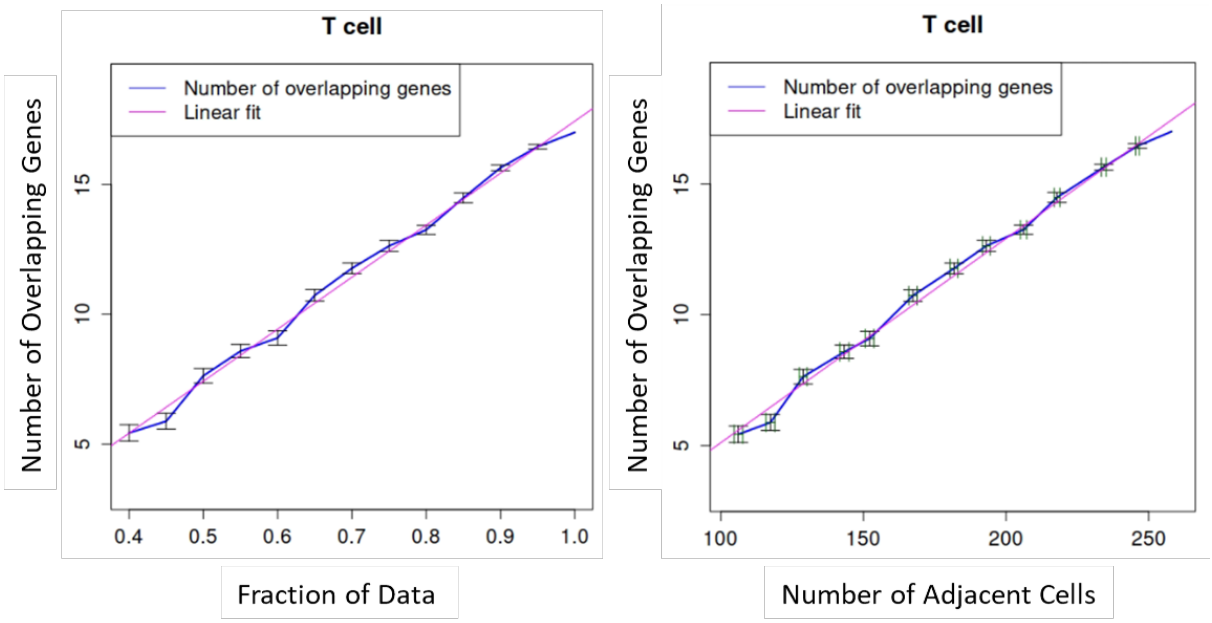
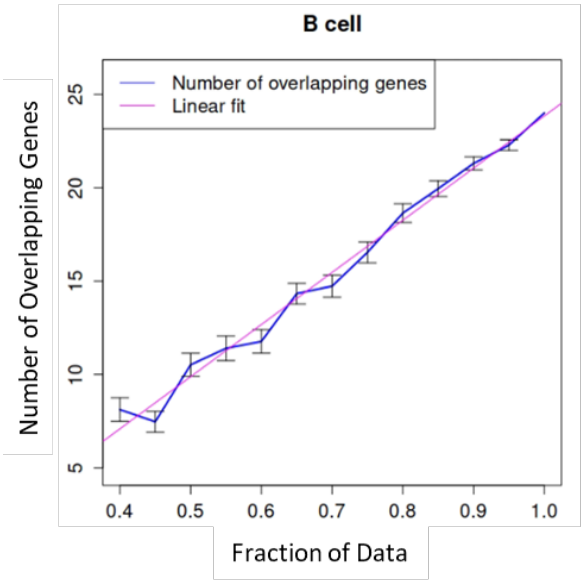
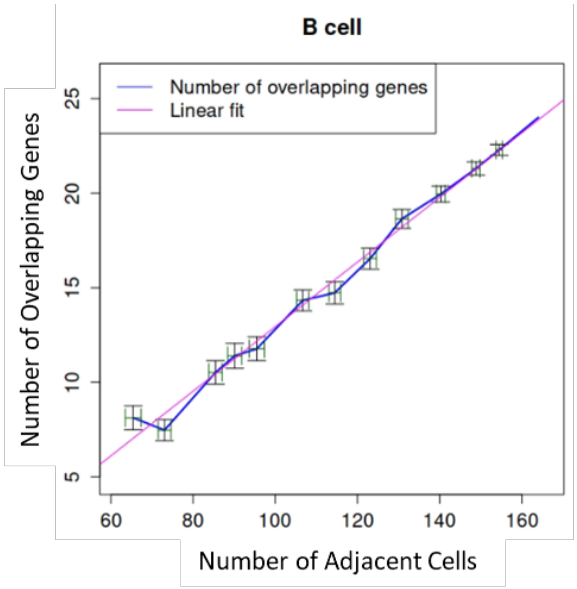
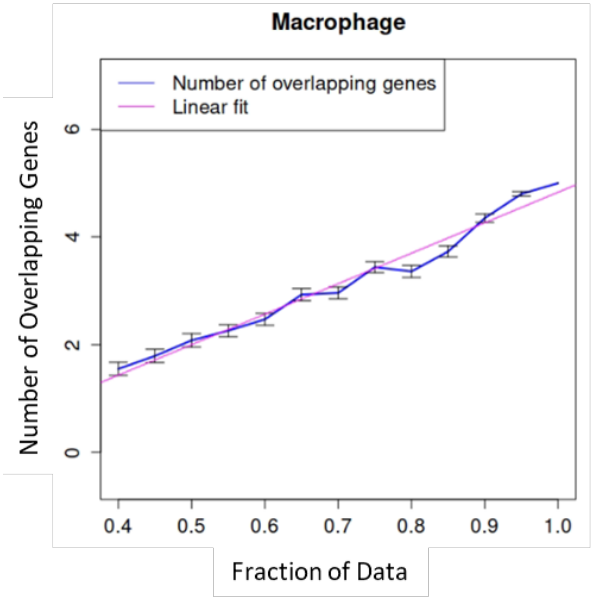
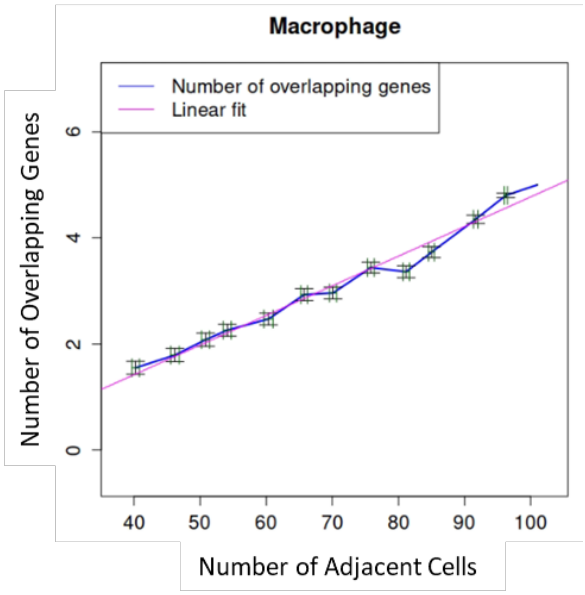


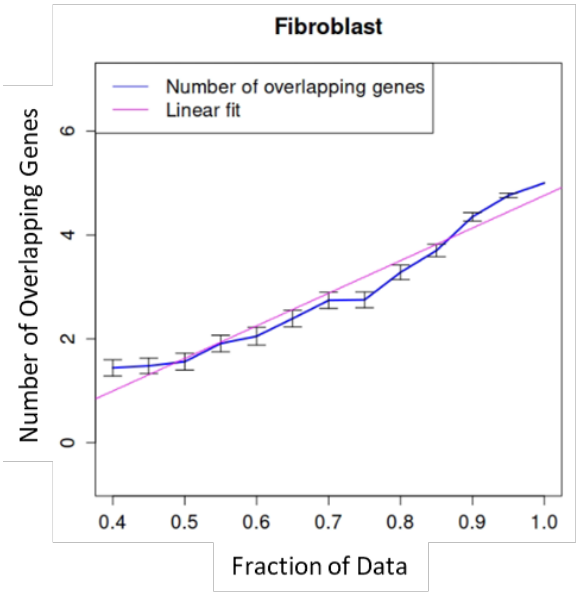
Figure S20. Scale-down analysis reveals the number of proximity-induced genes detected as a function of tissue size and number of adjacent cells. For each fraction of data, fields of view were randomly sampled 100 times from the full dataset (Methods section 'Scale-down analysis'). The average (blue line) and the standard error (vertical line) of the number of overlapping upregulated proximity-induced genes (i.e., overlap with the genes detected using the full dataset) detected is presented as a function of the fraction of data (left panel), and as a function of the number of adjacent cells (right panel). Adjacent cells are defined as the number of T cells which are in close proximity to tumor cells. Note that the number of adjacent cells can be different in each random realization, and therefore the average and the standard error of this value are shown (horizontal lines in the right panel). A linear trend between the number of proximity-induced genes and the fraction of the data utilized is evident (left panel, pink line; p-value 8e-14). A linear trend is also evident between the number of proximity-induced genes and the number of adjacent T cells and tumor cells (right panel, pink line; p-value 7e-14).


B cell
Number of overlapping genes
Linear fit
Number of Overlapping Genes
Fraction of Data


B cell
Number of overlapping genes
Linear fit
Number of Overlapping Genes
Number of Adjacent Cells


Macrophage
Number of overlapping genes
Linear fit
Number of Overlapping Genes
Fraction of Data


Macrophage
Number of overlapping genes
Linear fit
Number of Overlapping Genes
Number of Adjacent Cells


Fibroblast
Number of overlapping genes
Linear fit
Number of Overlapping Genes
Fraction of Data

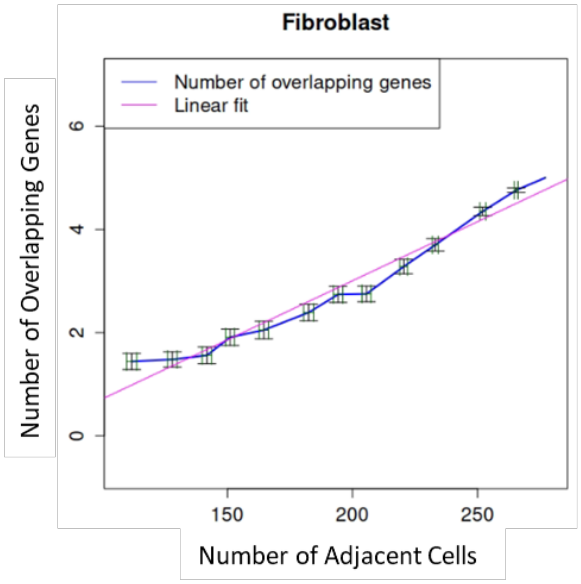

Fibroblast
Number of overlapping genes
Linear fit
Number of Overlapping Genes
Number of Adjacent Cells

Figure S21. Scale-down analysis reveals the number of proximity-induced genes detected as a function of tissue size and number of adjacent cells. For each fraction of data, fields of view were randomly sampled 100 times from the full dataset (Methods section 'Scale-down analysis'). The average (blue line) and the standard error (vertical line) of the number of overlapping upregulated proximity-induced genes (i.e., overlap with the genes detected using the full dataset) detected is presented as a function of the fraction of data (left panels), and as a function of the number of adjacent cells (right panels). Adjacent cells are defined as the number of non-tumor cells from a given cell type which are in close proximity to tumor cells. Note that the number of adjacent cells can be different in each random realization, and therefore the average and the standard error of this value are shown (horizontal lines in the right panels). A linear trend between the number of proximity-induced genes and the fraction of the data utilized is evident (left panels, pink line; p-values: B cells 7e-12, Macrophage 3e-10, Fibroblast 6e-09). A linear trend is also evident between the number of proximity-induced genes and the number of adjacent non-tumor and tumor cells (right panel, pink line; p-values: B cells 8e-13, Macrophage 1e-10, Fibroblast 2e-09).

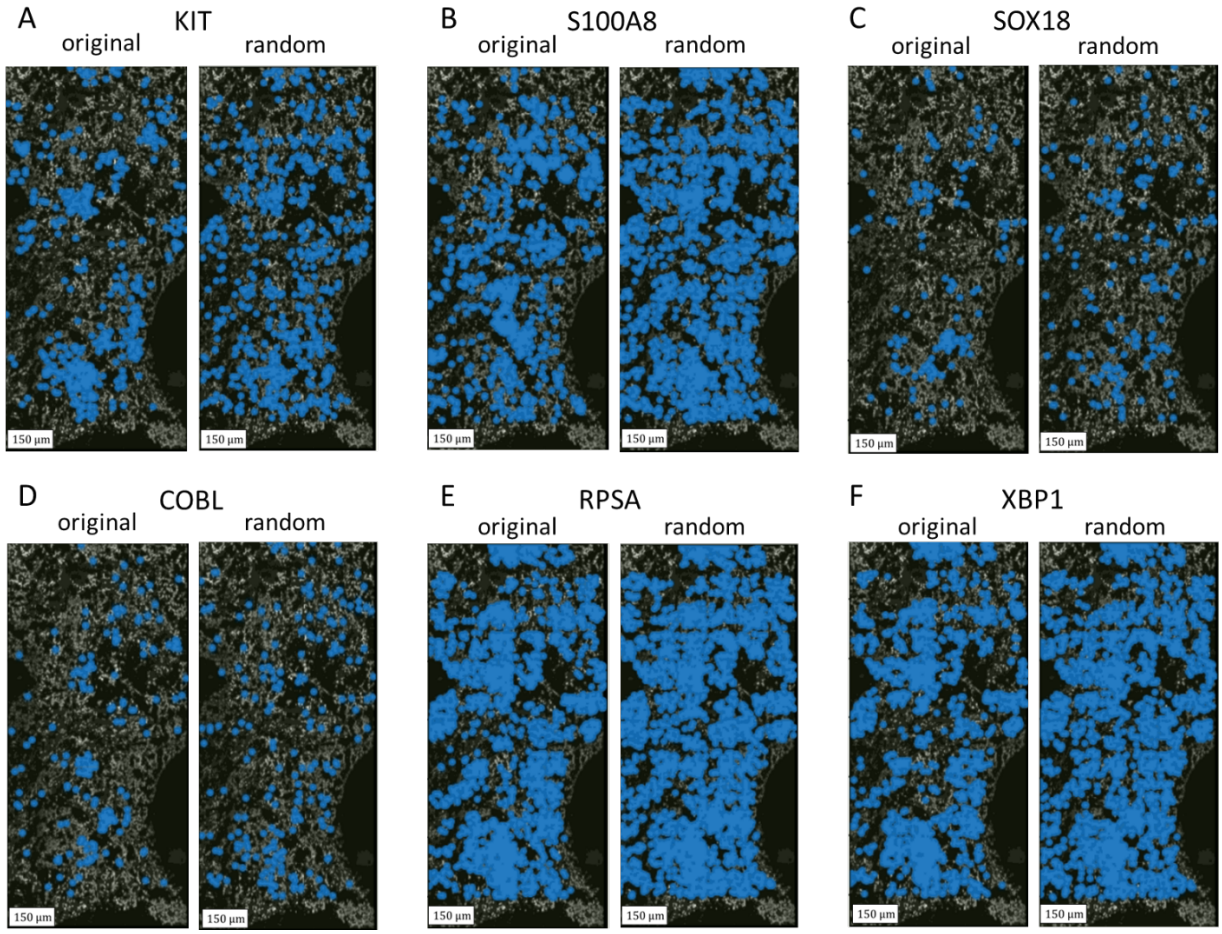


Figure S22. Detection of spatially-dependent genes using Moran's I calculation and permutation analysis. The top detected genes, sorted by p-value from the lowest to the highest, are presented. For each gene, two images are shown: the left shows the true locations of the gene reads, and the right is one non-biological realization. Below four values for each gene are shown: 1) Moran's I value of the true spatial distribution, marked as 'original' in the figure, (2) an average of 100 Moran's I values of non-biological spatial distribution, (3) Moran's I value of one specific non-biological spatial distribution, which is marked as 'random' in the figure, and (4) the resulting p-value. A) (1) 0.39, (2) 0.18, (3) 0.18, (4) <1e-15. B) (1) 0.46, (2) 0.35, (3) 0.36, (4) 3.33e-13. C) (1) 0.26, (2) 0.05, (3) 0.04, (4) <1e-15. D) (1) 0.2, (2) 0.07, (3) 0.04, (4) 4.12e-15. E) (1) 0.54, (2) 0.45, (3) 0.46, (4) 2.51e-14. F) (1) 0.6, (2) 0.41, (3) 0.42, (4) <1e-15.

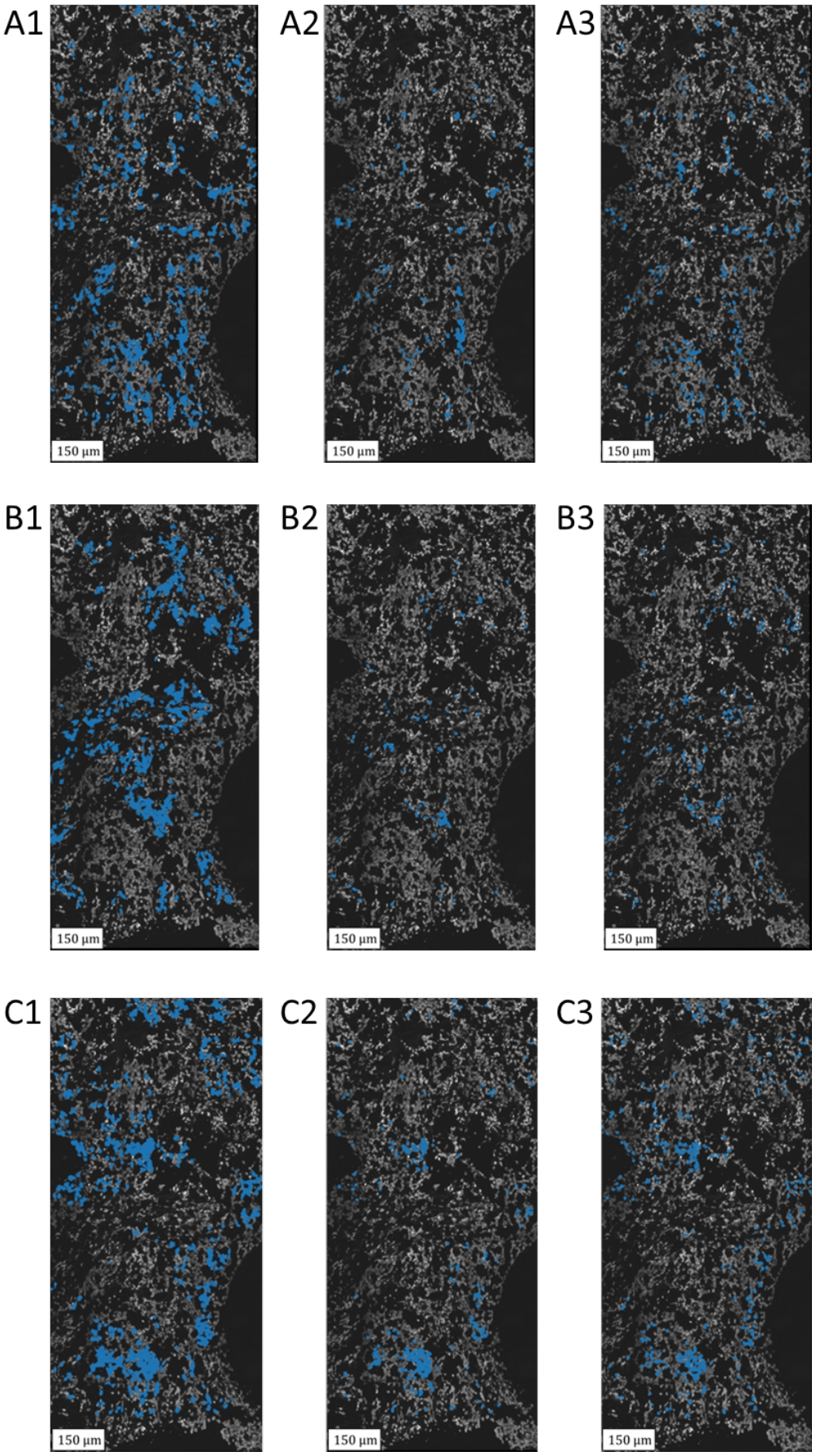

A1
A2
A3
B1
B2
B3
C1
C2
C3
150 µm

Figure S23. Example of three genes TTC6, ISG20 and AURKA which are spatially-dependent in spite of cell type spatial variability. (A1) all genes detected in T cells (blue), (A2) the locations of the gene TTC6 in T cells (blue), and (A3) the realization of gene locations inside T cells (blue), with the total sum of spots equal to (B). (B1-3), same as (A1-3), respectively, but for the gene ISG20 in B cells. (C1-3), same as (A1-3), respectively, but for the gene AURKA in Fibroblast. Moran's I of TTC6: 0.37, compared to 0.17±0.02 (all realizations), and 0.16 (realization shown in A3). Moran's I of ISG20: 0.38, compared to 0.17±0.03 (all realizations), and 0.15 (realization shown in B3). Moran's I of AURKA: 0.54, compared to 0.34±0.03 (all realizations), and 0.30 (realization shown in C3).

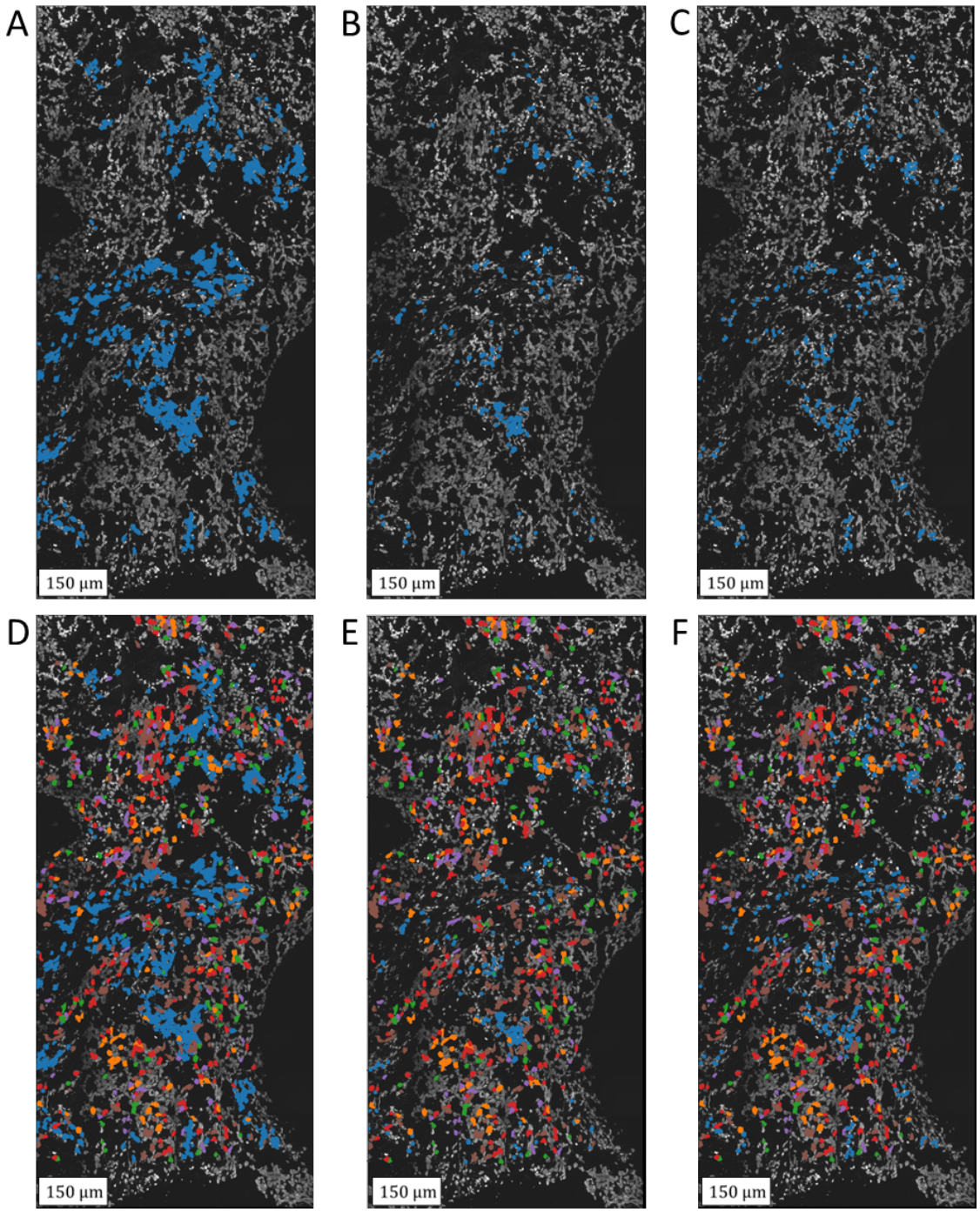


Figure S24A. The gene *AHR* is spatially-dependent, but without a visually clear correlation to the locations of tumor cells. (A) all genes detected in B cells (blue), (B) the locations of the gene *AHR* in B cells (blue), and (C) the realization of gene locations inside B cells (blue), with the total sum of spots equal to (B). (D-F), same as (A-C), respectively, but overlaid with the locations of tumor cell types: ALDH1A3-positive (orange), EGFR-positive (green), EPCAM-positive (red), CD44-positive (purple) and PGR-positive (brown). Moran's I of *AHR*: 0.44, compared to 0.27±0.03 (all realizations), and 0.28 (realization shown in (C)).

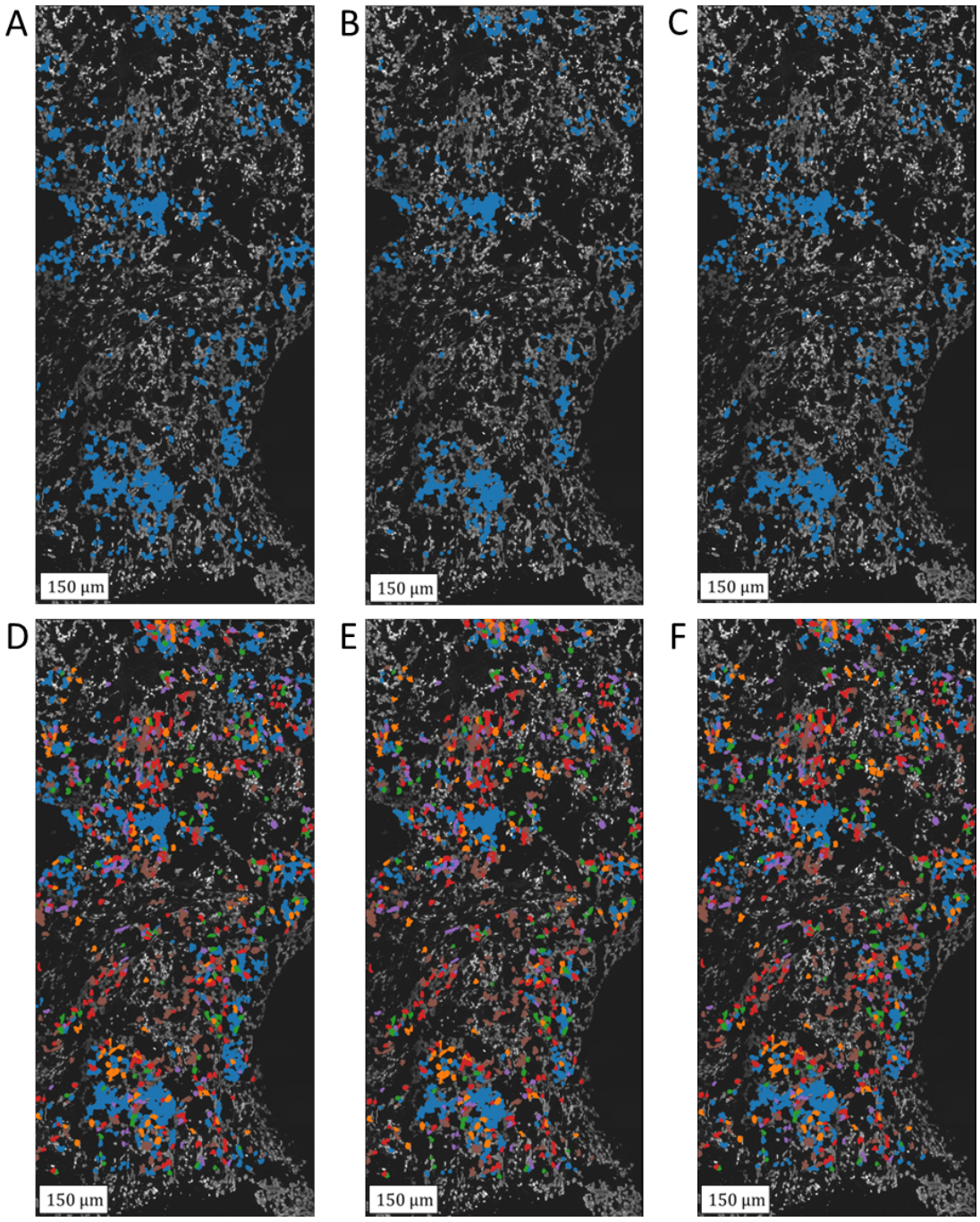


Figure S24B. The gene *XBP1* is spatially-dependent, but without a visually clear correlation to the locations of tumor cells. (A) all genes detected in fibroblasts (blue), (B) the locations of the gene *XBP1* in fibroblasts (blue), and (C) the realization of gene locations inside fibroblasts (blue), with the total sum of spots equal to (B). (D-F), same as (A-C), respectively, but overlaid with the locations of tumor cell types: ALDH1A3-positive (orange), EGFR-positive (green), EPCAM-positive (red), CD44-positive (purple) and PGR-positive (brown). Moran's I of *XBP1*: 0.63, compared to 0.51±0.02 (all realizations), and 0.51 (realization shown in (C)).

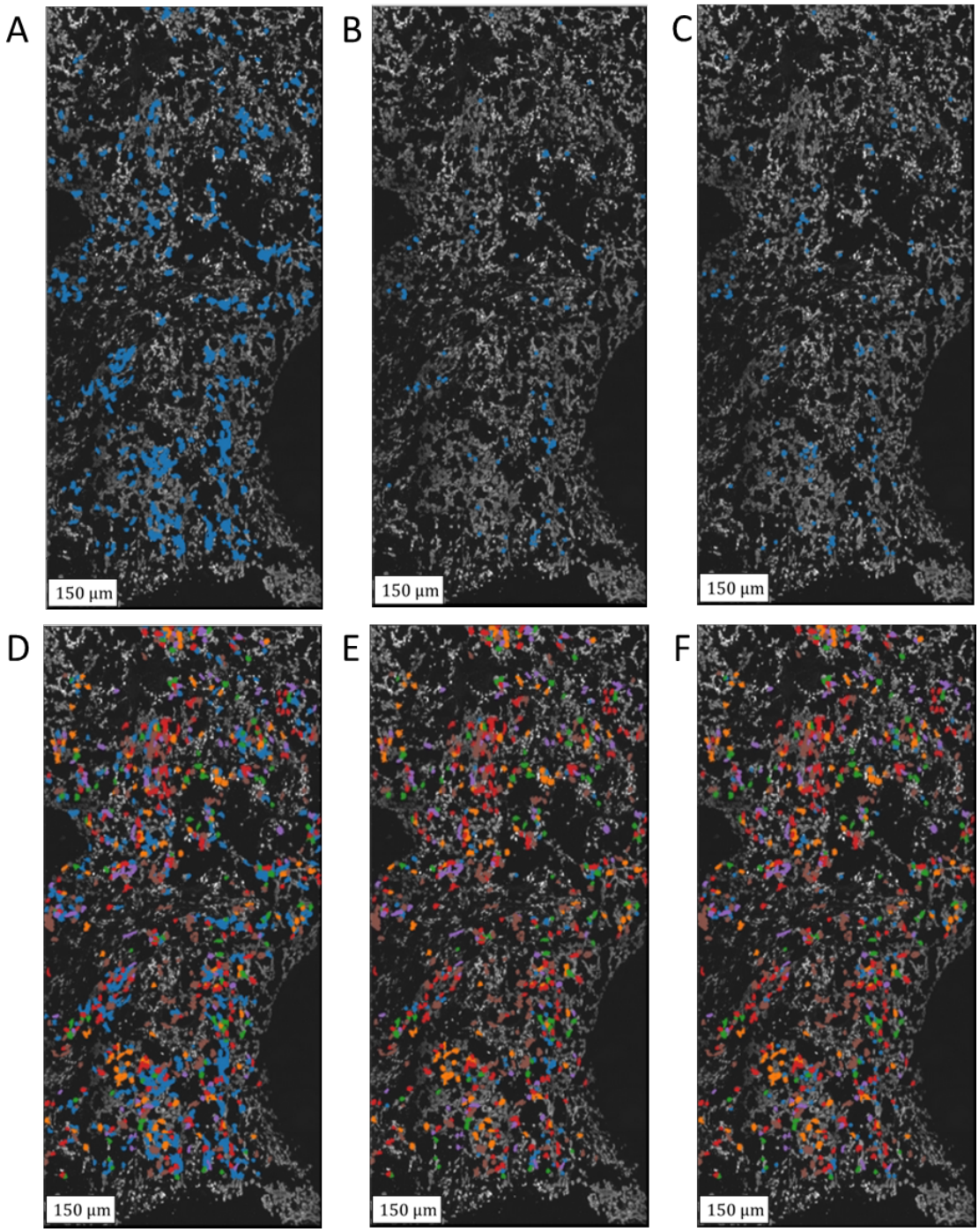


Figure S24C. The gene *ICAM1* is spatially-dependent, but without a visually clear correlation to the locations of tumor cells. (A) all genes detected in T cells (blue), (B) the locations of the gene *ICAM1* in T cells (blue), and (C) the realization of gene locations inside T cells (blue), with the total sum of spots equal to (B). (D-F), same as (A-C), respectively, but overlaid with the locations of tumor cell types: ALDH1A3-positive (orange), EGFR-positive (green), EPCAM-positive (red), CD44-positive (purple) and PGR-positive (brown). Moran's I of *ICAM1*: 0.24, compared to 0.11±0.03 (all realizations), and 0.14 (realization shown in (C)).

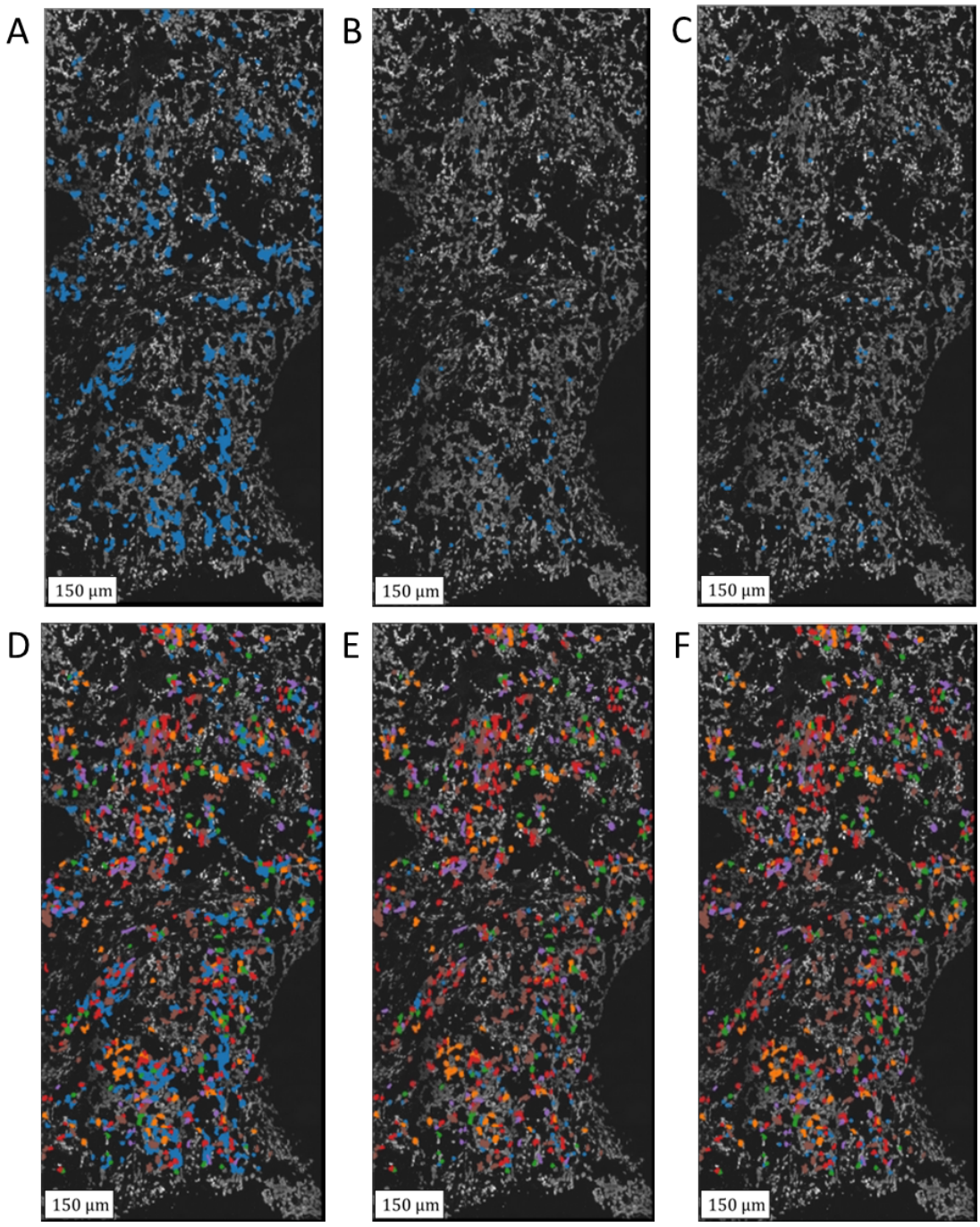


Figure S24D. The gene *LGMN* is spatially-dependent, but without a visually clear correlation to the locations of tumor cells. (A) all genes detected in T cells (blue), (B) the locations of the gene *LGMN* in T cells (blue), and (C) the realization of gene locations inside T cells (blue), with the total sum of spots equal to (B). (D-F), same as (A-C), respectively, but overlaid with the locations of tumor cell types: ALDH1A3-positive (orange), EGFR-positive (green), EPCAM-positive (red), CD44-positive (purple) and PGR-positive (brown). Moran's I of *LGMN*: 0.18, compared to 0.08±0.03 (all realizations), and 0.05 (realization shown in (C)).

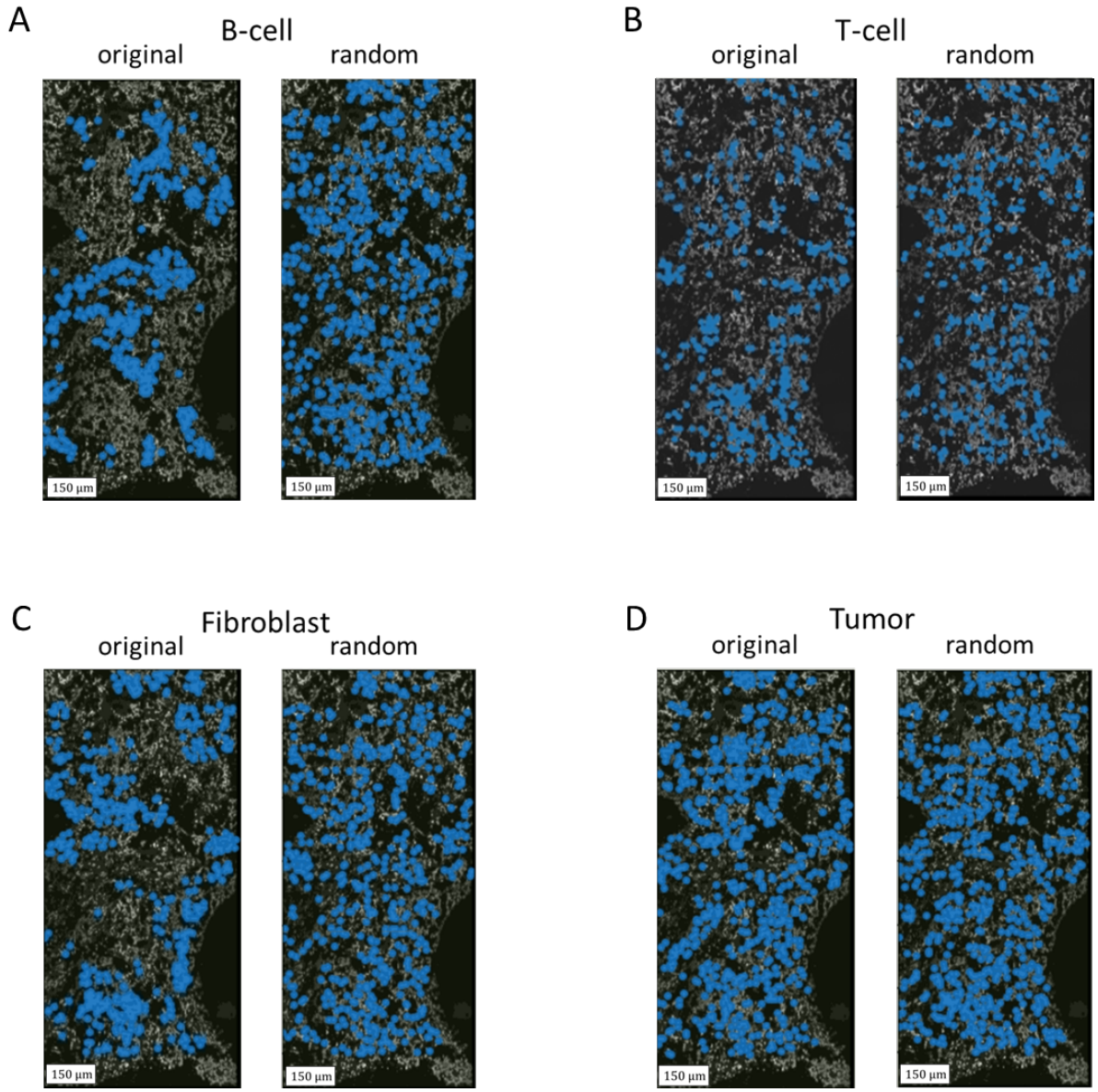


Figure S25. Detection of spatially-dependent cell types using Moran's I calculation and permutation analysis. For each cell type, two images are shown: the left shows the true locations of the cells, and the right is one non-biological realization. Below four values for each cell type are shown: 1) Moran's I value of the true spatial distribution, marked as 'original' in the figure, (2) an average of 100 Moran's I values of non-biological spatial distribution, (3) Moran's I value of one specific non-biological spatial distribution, which is marked as 'random' in the figure, and (4) the resulting p-value. A) (1) 0.38, (2) 0.08, (3) 0.07, (4) <1e-15. B) (1) 0.17, (2) 0.06, (3) 0.04, (4) <1e-15. C) (1) 0.29, (2) 0.08, (3) 0.1, (4) <1e-15. D) (1) 0.14, (2) 0.1, (3) 0.13, (4) 3.84e-3.

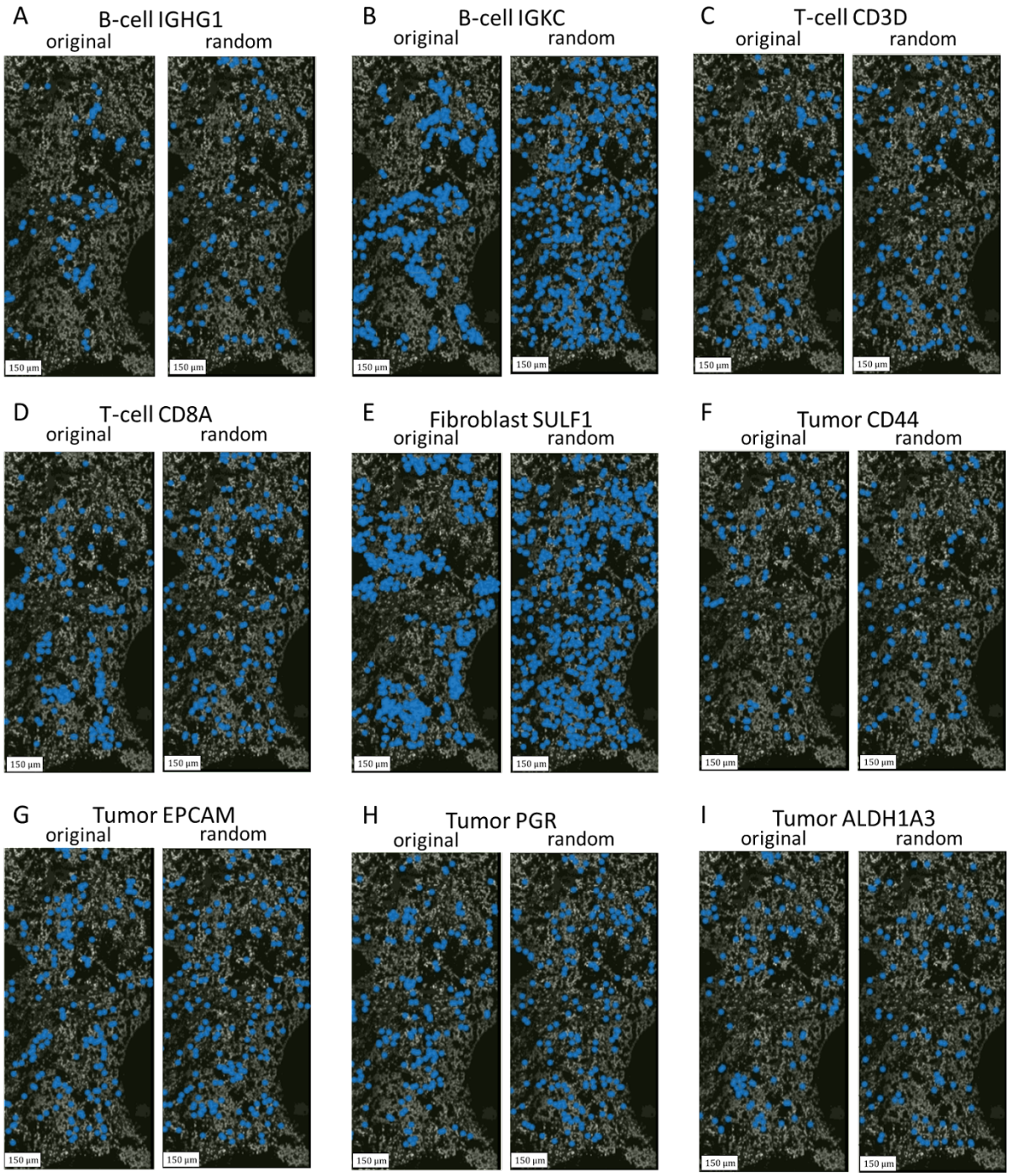


Figure S26. Detection of spatially-dependent cell subtypes using Moran's I calculation and permutation analysis. A-H) For each cell subtype, two images are shown: the left shows the true locations of the cells, and the right is one non-biological realization. Below four values for each cell subtype are shown: 1) Moran's I value of the true spatial distribution, marked as 'original' in the figure, (2) an average of 100 Moran's I values of non-biological spatial distribution, (3) Moran's I value of one specific non-biological spatial distribution, which is marked as 'random' in the figure, and (4) the resulting p-value. (I) similar representation for ALDH1A3-positive tumor cell type, but note that this cell type was not detected as spatially-dependent with statistical significance. Excluding (I), only cell

subtypes with FDR<0.01 are shown.

A) (1) 0.24, (2) 0.02, (3) 0.03, (4) <1e-15. B) (1) 0.29, (2) 0.07, (3) 0.07, (4) <1e-15.

C) (1) 0.1, (2) 0.02, (3) 0.02, (4) 1.41e-12. D) (1) 0.17, (2) 0.03, (3) 0.04, (4) <1e-15.

E) (1) 0.29, (2) 0.08, (3) 0.08, (4) <1e-15. F) (1) 0.05, (2) 0.01, (3) 0.03, (4) 1.65e-3.

G) (1) 0.07, (2) 0.03, (3) 0.03, (4) 3.33e-4. H) (1) 0.09, (2) 0.03, (3) 0.02, (4) 2.36e-7.

I) (1) 0.04, (2) 0.02, (3) 0.03, (4) 0.05.

Supplementary Tables

Table S1. The names of 297 genes in the expansion sequencing panel utilized in this study. The table is attached as a separate spreadsheet file.

FOV	min_nuc	min_som	max_som	area_remove_quant	area_big_quant
0	0.95	0.85	0.95	0.5	0.75
1	0.96	0.85	0.96	0.4	0.75
2	0.95	0.85	0.95	0.6	0.78
3	0.93	0.8	0.93	0.5	0.73
4	0.96	0.85	0.96	0.9	0.75
5	0.945	0.85	0.945	0.6	0.82
6	0.96	0.85	0.96	0.55	0.85
7	0.97	0.85	0.97	0.65	0.75
8	0.975	0.85	0.975	0.6	0.85
9	0.97	0.85	0.97	0.7	0.75
10	0.975	0.85	0.975	0.5	0.89
11	0.96	0.85	0.96	0.5	0.85
14	0.97	0.85	0.97	0.5	0.75
15	0.94	0.8	0.94	0.4	0.91
16	0.89	0.85	0.89	0.5	0.73
17	0.975	0.85	0.975	0.6	0.93
18	0.955	0.85	0.955	0.5	0.8
19	0.97	0.85	0.97	0.5	0.905
20	0.945	0.8	0.945	0.4	0.85
21	0.94	0.85	0.94	0.4	0.9
22	0.935	0.85	0.935	0.5	0.75
23	0.96	0.85	0.96	0.6	0.86
24	0.965	0.85	0.965	0.7	0.85
26	0.95	0.8	0.95	0.65	0.87
27	0.97	0.85	0.97	0.8	0.82
28	0.95	0.8	0.95	0.5	0.85
29	0.94	0.8	0.94	0.5	0.75
30	0.945	0.75	0.945	0.5	0.94
31	0.94	0.7	0.94	0.5	0.9
32	0.96	0.8	0.96	0.6	0.75
33	0.95	0.85	0.95	0.5	0.75
34	0.95	0.75	0.95	0.5	0.75
35	0.95	0.75	0.95	0.5	0.75
36	0.945	0.75	0.945	0.5	0.99
37	0.95	0.8	0.95	0.5	0.87
40	0.95	0.8	0.95	0.35	0.92
41	0.95	0.75	0.95	0.5	0.9
42	0.945	0.75	0.945	0.4	0.8
43	0.95	0.8	0.95	0.5	0.9
44	0.97	0.75	0.97	0.5	0.95
45	0.975	0.8	0.975	0.5	0.81
46	0.98	0.8	0.98	0.5	0.94
47	0.96	0.8	0.96	0.5	0.8
48	0.97	0.8	0.97	0.6	0.75
49	0.96	0.8	0.96	0.4	0.85
50	0.97	0.8	0.97	0.5	0.74
51	0.97	0.8	0.97	0.5	0.85
52	0.97	0.8	0.97	0.35	0.9
53	0.95	0.8	0.95	0.5	0.8
54	0.95	0.8	0.95	0.4	0.92
55	0.98	0.8	0.98	0.5	0.94
57	0.975	0.8	0.975	0.5	0.85
58	0.95	0.8	0.95	0.5	0.91
59	0.95	0.8	0.95	0.5	0.91
60	0.945	0.8	0.945	0.5	0.79
61	0.93	0.8	0.93	0.85	0.8
62	0.97	0.8	0.97	0.4	0.89
63	0.97	0.8	0.97	0.4	0.92
66	0.97	0.8	0.97	0.9	0.85
71	0.945	0.8	0.945	0.4	0.87
72	0.97	0.8	0.97	0.4	0.93
73	0.97	0.8	0.97	0.4	0.9
74	0.975	0.8	0.975	0.4	0.85
75	0.96	0.8	0.96	0.5	0.84
76	0.965	0.8	0.965	0.4	0.9

Table S2. The parameters used in the segmentation in each field of view.

FOV = the field of view analyzed.

min_nuc = the minimum percentile of pixel intensity for the representation of nuclei.

min_som = the minimum percentile of pixel intensity for representation of the cell bodies.

max_som = the maximum percentile of pixel intensity for representation of the cell bodies.

area_remove_quant = the maximum percentile of area for removing objects that are suspected of

being noise.

area_big_quant = the minimum percentile of area for splitting objects that are suspected to be combined.

In addition to the parameters presented in the table, several constant parameters were used (explained below): medfiltmask=9, max_nuc=1, closingmask=3, openingmask=3, max_val_quant=0.9999, patch_num_x=3, patch_num_y=3.

medfiltmask = the size of the median filter.

max_nuc = the maximum percentile of pixel intensity for the representation of nuclei.

closingmask = the size of the closing mask.

openingmask = the size of the opening mask.

max_val_quant = the minimum percentile of pixel intensity for representation of noise.

patch_num_x = the number of divisions of FOV to sub-fields in the x-axis.

patch_num_y = the number of divisions of FOV to sub-fields in the y-axis.

Table S3. Comparison of the genes detected by differential expression analysis between 0.5, 1 and 3 microns as the distance cutoff between proximal cells. The table is attached as a separate spreadsheet file.

cell type X [1]	cell type Y [1]	number of k folds	median number of cells in false category	median number of cells in true category	the ratio between minimal and maximal	mean of 30 AUC values	mean of 30 AUC values of non biological data	FDR [2]
Bcell	Tumor	6	72	28	0.389	0.615	0.521	1.02E-07
Bcell	Tumor_EPCAM	3	193	6	0.031	0.701	0.510	9.78E-05
Bcell	Tumor_ALDH1A3	3	182	17	0.093	0.639	0.486	5.19E-09
Bcell	Tumor_PGR	4	129	20	0.155	0.617	0.519	8.83E-06
Bcell_IGHG1	Tumor_PGR	3	32	5	0.156	0.764	0.519	9.89E-07
Bcell_IGHG1	Tumor_EGFR	3	35	3	0.086	0.697	0.457	9.69E-05
Bcell_IGKC	Tumor_EGFR	3	157	5	0.032	0.708	0.468	7.69E-09
Bcell_IGKC	Tumor_EPCAM	3	156	6	0.038	0.679	0.498	7.06E-05
Bcell_IGKC	Tumor_PGR	4	105	16	0.152	0.620	0.498	2.09E-06
Fibroblast	Tumor_EPCAM	4	120	25	0.208	0.681	0.528	6.52E-11
Fibroblast	Tumor_ALDH1A3	4	122	23	0.189	0.616	0.504	2.09E-06
Fibroblast	Tumor_PGR	3	169	24	0.142	0.612	0.496	9.89E-07
Fibroblast	Tumor_EGFR	3	170	23	0.135	0.604	0.506	2.24E-06
Fibroblast	Tumor	6	50	46	0.920	0.598	0.493	8.98E-10
Macrophage	Tumor_EPCAM	3	56	10	0.179	0.735	0.514	4.03E-08
Macrophage	Tumor	4	25	25	1.000	0.693	0.487	4.21E-07
Tcell	Tumor_EPCAM	5	55	25	0.455	0.700	0.509	2.30E-14
Tcell	Tumor	6	24	43	0.558	0.689	0.527	9.47E-13
Tcell	Tumor_PGR	4	79	22	0.266	0.628	0.486	4.64E-13
Tcell	Tumor_EGFR	3	113	20	0.177	0.581	0.497	3.74E-05
Tcell_CD3D	Tumor_EGFR	3	42	6	0.143	0.803	0.529	1.75E-07
Tcell_CD3D	Tumor_PGR	3	37	11	0.297	0.694	0.528	1.16E-05
Tcell_CD3D	Tumor_EPCAM	3	35	13	0.371	0.686	0.505	1.02E-07
Tcell_CD3D	Tumor_ALDH1A3	3	41	7	0.171	0.664	0.409	1.70E-09
Tcell_CD3D	Tumor	3	18	30	0.600	0.630	0.489	9.48E-07
Tcell_CD8A	Tumor_EPCAM	3	40	22	0.550	0.672	0.478	2.61E-10
Tcell_CD8A	Tumor	3	21	42	0.500	0.636	0.551	9.69E-05

Table S4. Summary statistics of the machine learning classifications (ML) for each one of the comparisons between non-tumor cells and tumor cells. Only comparisons with FDR<1e-4 in the machine learning classification are presented.

[1]The comparisons were performed between: (a) the cells from type X that are in proximity (true label) to cell type Y, and (b) the cells from type X that are not in proximity (false label) to cell type Y. [2]The false discovery rate (FDR) of the machine learning classification, which takes into account 54 multiple tests, as 54 comparisons between non-tumor cells and tumor cells were performed.

cell type X [1]	cell type Y [1]	number of k folds	median number of cells in false category	median number of cells in true category	the ratio between minimal and maximal	mean of 30 AUC values	mean of 30 AUC values of non biological data	FDR [2]
Tumor	Bcell	5	121	23	0.190	0.936	0.491	1.01E-31
Tumor	Bcell_IGHG1	3	226	13	0.058	0.953	0.478	1.23E-21
Tumor	Bcell_IGKC	5	124	20	0.161	0.901	0.511	1.13E-24
Tumor	Fibroblast	6	73	46	0.630	0.817	0.475	2.81E-32
Tumor	Macrophage	6	97	22	0.227	0.586	0.503	2.36E-07
Tumor	Tcell	6	71	48	0.676	0.655	0.497	3.35E-17
Tumor	Tcell_CD3D	6	98	21	0.214	0.634	0.507	1.44E-10
Tumor	Tcell_CD8A	6	92	27	0.293	0.652	0.502	4.36E-15
Tumor_ALDH1A3	Bcell	3	29	9	0.310	0.976	0.475	4.94E-21
Tumor_ALDH1A3	Bcell_IGHG1	3	33	5	0.152	0.942	0.473	8.59E-16
Tumor_ALDH1A3	Bcell_IGKC	3	31	7	0.226	0.926	0.482	1.89E-18
Tumor_ALDH1A3	Fibroblast	3	18	20	0.900	0.736	0.522	2.45E-09
Tumor_ALDH1A3	Tcell	3	25	13	0.520	0.653	0.495	6.59E-06
Tumor_ALDH1A3	Tcell_CD3D	3	32	6	0.188	0.729	0.501	9.00E-09
Tumor_CD44	Bcell	3	32	3	0.094	0.894	0.476	2.68E-12
Tumor_CD44	Bcell_IGKC	3	32	3	0.094	0.741	0.498	1.38E-05
Tumor_CD44	Fibroblast	3	21	14	0.667	0.648	0.453	4.99E-08
Tumor_EGFR	Bcell	3	28	6	0.214	0.806	0.486	1.42E-11
Tumor_EGFR	Bcell_IGHG1	3	31	2	0.065	0.777	0.421	1.34E-05
Tumor_EGFR	Fibroblast	3	19	14	0.737	0.623	0.482	9.88E-06
Tumor_EGFR	Tcell_CD8A	3	26	8	0.308	0.723	0.528	1.95E-05
Tumor_EPCAM	Bcell	3	65	4	0.062	0.759	0.545	2.37E-06
Tumor_EPCAM	Bcell_IGKC	3	65	4	0.062	0.775	0.423	5.38E-12
Tumor_EPCAM	Fibroblast	4	32	20	0.625	0.724	0.513	3.37E-12
Tumor_EPCAM	Tcell	4	27	25	0.926	0.574	0.483	6.25E-05
Tumor_EPCAM	Tcell_CD8A	3	50	19	0.380	0.633	0.473	2.45E-09
Tumor_PGR	Bcell	3	48	15	0.313	0.912	0.526	3.96E-17
Tumor_PGR	Bcell_IGHG1	3	58	5	0.086	0.902	0.473	4.36E-15
Tumor_PGR	Bcell_IGKC	3	50	13	0.260	0.908	0.498	1.90E-18
Tumor_PGR	Fibroblast	3	46	17	0.370	0.876	0.524	1.98E-18
Tumor_PGR	Macrophage	3	52	11	0.212	0.631	0.515	5.43E-05
Tumor_PGR	Tcell	3	39	24	0.615	0.654	0.486	1.21E-10
Tumor_PGR	Tcell_CD3D	3	52	11	0.212	0.629	0.490	1.16E-08

Table S5. Summary statistics of the machine learning classifications (ML) for each one of the comparisons between tumor cells and non-tumor cells. Only comparisons with FDR<1e-4 in the machine learning classification are presented.

[1]The comparisons were performed between: (a) the cells from type X that are in proximity (true label) to cell type Y, and (b) the cells from type X that are not in proximity (false label) to cell type Y. [2]The false discovery rate (FDR) of the machine learning classification, which takes into account 54 multiple tests, as 54 comparisons between tumor cells and non-tumor cells were performed.

Table S6. Summary of the results of the three detection methods: differential expression, machine learning, and matrix factorization. The table includes, for each comparison between two cell types, the significant genes in each method (names and number of genes) and the overlapping genes between the methods (names, number of genes, percent overlap, and p-value of the overlap). The table is attached as a separate spreadsheet file.

	K=18		K=18 - including cells in types marked as 'unknown'		K=16		K=20	
GEP	**cell type-related**	**Proximity-related**	**cell type-related**	**Proximity-related**	**cell type-related**	**Proximity-related**	**cell type-related**	**Proximity - related**
1	**T cell**		T cell		T cell		T cell	
2	**B cell, Mac**		B cell	Tumor	B cell, Mac	Tumor	B cell	
3	**Fib**		Fib				Fib	
4	**B cell**, **Fib**		B cell				B cell, Fib	
5	**B cell** [IGHG1][8], **Tumor**		B cell [IGHG1][8], Tumor		B cell [IGHG1][8], Tumor		B cell [IGHG1][8], Tumor	
6	**B cell** [IGHM, IGKC][8]		B cell [IGHM, IGKC][8]		B cell [IGHM, IGKC][8], Tumor		B cell [IGHM, IGKC][8], Tumor	
7								
8	**Mac** [HLA-DRA][8], **Fib**	**Fib**	Mac [HLA-DRA][8]	Fib	Mac [HLA-DRA][8], Fib		Mac [HLA-DRA][8], Fib	
9		**B cell**[1] P-value-DE[9] = 0.07 P-value-ML[10] = 0.37		B cell		B cell		
ss	**B cell, Fib** [SULF1][8]	**Fib**[2] P-value-DE[9] = 0.21 P-value-ML[10] = 0.71	B cell, Fib [SULF1][8]	Fib	B cell, Fib [SULF1][8]	Fib	B cell, Fib [SULF1][8]	Fib
11	**Fib** [SULF1][8]		Fib [SULF1][8]		Fib [SULF1][8]		Fib [SULF1][8]	
12	**B cell, Fib** [HSPG2][8]		Fib [HSPG2] [8]		B cell, Mac, Fib [HSPG2][8]		B cell, Fib [HSPG2][8]	

13	**B cell**		B cell		B cell, Tumor	Tumor	B cell, Tumor	
14								
15								
16	**T cell, B cell, Tumor** [KRT18, KRT19][8]	**T cell**[3] P-value-DE[9] < 0.01 P-value-ML[10] < 0.01 **Mac**[4] P-value-DE[9] = 0.17 P-value-ML[10] = 0.09 **Fib**[5] P-value-DE[9] = 0.04 P-value-ML[10] < 0.01	T cell, B cell, Fib, Tumor [KRT18, KRT19][8]	T cell, Mac, Fib	T cell, B cell, Fib	T cell, Fib	T cell, B cell, Tumor [KRT18, KRT19][8]	T cell, Mac, Fib
17	**T cell, B cell** [IGHG4, IGHG1, IGKC][8], **Fib, Tumor** [CD24][8]	**B cell**[6] P-value-DE[9] = 0.92 P-value-ML[10] = 0.74	T cell, B cell [IGHG1, IGHG4, IGKC][8], Fib, Tumor [CD24][8]	B cell	T cell, B cell [IGHG1, IGHG4, IGKC][8], Fib, Tumor [CD24][8]	B cell	T cell, B cell [IGHG1, IGHG4, IGKC][8], Fib, Tumor [CD24][8]	
18	**Fib, Tumor** [PGR][8]	**Tumor**[7] P-value-DE[9] = 0.7	Fib, Tumor [PGR][8]	Tumor	Fib, Tumor	T cell, Fib, Tumor	Fib, Tumor [EPCAM, PGR][8]	Tumor
19								
20								

Table S7. Sensitive tests for cell type GEPs and proximity-related GEPs. Comparison of significant GEP when the dataset was based on either K=18 (the baseline), K=16, or K=20, and also with K=18 but including cells in types marked as 'unknown'. The calculations were performed as described for the baseline condition. Fib=fibroblast; Mac=Macrophage. Only cell types with FDR<0.05 in bootstrapping analysis are presented (both for cell type GEP and proximity-related GEP). The genes highly-associated with each GEP (cell type and proximity-related) are presented in Table S8.

[1]a B cell proximity-related GEP. Out of the 15 genes highly-associated with this GEP, 6 genes overlapped with the 23 genes detected as differentially expressed when comparing B cells proximal

versus not proximal to tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping genes are: COL1A2, COL1A1, COL4A1, COL3A1, FGFR1, TIMP1. These overlapping genes were all present in the sensitive tests for K=16 and K=18 including 'unknown' cell type. Out of the 15 genes highly-associated with this GEP, a single gene (TIMP1) overlapped with the 10 genes detected by machine learning (Fig. S18).

[2] a Fibroblast proximity-related GEP. Out of the 15 genes highly-associated with this GEP, 2 genes overlapped with the 7 genes detected as differentially expressed when comparing Fibroblast cells proximal versus not proximal to tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping genes are: SULF1, CD3G. These overlapping genes were all present in the sensitive tests for K=16, K=20 and K=18 including 'unknown' cell type. Out of the 15 genes highly-associated with this GEP, a single gene (CD3G) overlapped with the 10 genes detected by machine learning (Fig. S18).

[3] a T cell proximity-related GEP. Out of the 15 genes highly-associated with this GEP, 8 genes overlapped with the 19 genes detected as differentially expressed when comparing T cells proximal versus not proximal to tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping genes are: KRT18, MYL6, TFF1, TMSB4X, RPSA, S100A14, FASN, CD63. These overlapping genes were all present in the sensitive tests for K=16, K=20 and K=18 including 'unknown' cell type. Out of the 15 genes highly-associated with this GEP, 5 genes (HSPB1, TMSB4X, RPSA, S100A14, CD63) overlapped with the 10 genes detected by machine learning (Fig. S18).

[4] a Macrophage proximity-related GEP. Out of the 15 genes highly-associated with this GEP, 2 genes overlapped with the 9 genes detected as differentially expressed when comparing Macrophage cells proximal versus not proximal to tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping genes are: TFF1, KRT19. These overlapping genes were all present in the sensitive tests for K=20 and K=18 including 'unknown' cell type. Out of the 15 genes highly-associated with this GEP, 3 genes (TFF1, KRT19, IFITM3) overlapped with the 10 genes detected by machine learning (Fig. S18).

[5] a Fibroblast proximity-related GEP. Out of the 15 genes highly-associated with this GEP, 4 genes overlapped with the 7 genes detected as differentially expressed when comparing Fibroblast cells proximal versus not proximal to tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping genes are: KRT18, MYL6, TMSB4X, RPSA. These overlapping genes were all present in the sensitive tests for K=16, K=20 and K=18 including 'unknown' cell type. Out of the 15 genes highly-associated with this GEP, 6 genes (KRT18, MYL6, KRT19, TMSB4X, RPSA, XBP1) overlapped with the 10 genes detected by machine learning (Fig. S18).

[6] a B cell proximity-related GEP. Out of the 15 genes highly-associated with this GEP, a single gene overlapped with the 23 genes detected as differentially expressed when comparing B cells proximal versus not proximal to tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping gene is: LYZ. This overlapping gene was present in the sensitive tests for K=16, K=20 and

K=18 including ‘unknown’ cell type. Out of the 15 genes highly-associated with this GEP, a single gene (FAT1) overlapped with the 10 genes detected by machine learning (Fig. S18).

[7]a tumor proximity-related GEP. Out of the 15 genes highly-associated with this GEP, 2 genes overlapped with the 22 genes detected as differentially expressed when comparing tumor cells proximal versus not proximal to not-tumor cells (Fig. S14, but note that this figure excludes tumor markers). The overlapping genes are: CD69 and KIF23 were present in the sensitive tests for K=16, K=20 and K=18 including ‘unknown’ cell type.

[8] cell type marker genes associated with the given GEP.

[9] P-value of the observed overlap between the genes detected by cNMF and by differential expression analysis (DE), bootstrapping.

[10] P-value of the observed overlap between the genes detected by cNMF and by machine learning analysis (ML), bootstrapping.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	SFRP1	RPL18	CSTB	IGF1R	CALCRL	MS4A1	TFRC	LGALS2	GNLY	DCN	BCL2	EGFR	MYLK	ERBB2	TMEM45B
2	CD24	FOS	ZNF571	SOX10	LYZ	CD36	MDM2	PBK	FGFR2	IL7R	GRB7	PDPN	PTPRC	MLPH	EFNA5
3	KIT	CD3E	CTSL	CSRP2	KRT14	PTPRC	CR2	MSR1	PTPRB	GZMB	BANK1	MKI67	KRT10	MYL6	CD44
4	IGFBP5	TSPAN1	SNAI2	PDCD1	GATA3	CD7	COL3A1	KRT5	SLC2A1	ZEB1	TMSB10	UBE2T	AHR	FGFR1	CR2
5	SLC39A6	FGFR4	NDC80	IGHG1	CD163	FCN1	FTL	CCNE2	TRAC	AHR	PIK3CA	ADGRL4	CLDN4	FCRL5	NR3C1
6	IGHM	IGKC	COBL	LYPD6B	MELK	FOXA1	PDCD1	CD40LG	NDRG2	AGR2	AZGP1	JUNB	ORC6	MMP9	FGFR1
7	ACTA2	KRAS	PTPRB	POU2AF1	PLEK	MT2A	PTPRC	TFRC	CDK7	FASN	CCL5	S100A9	CDH11	MYO10	PLVAP
8	CD74	HLA-DRB1	HLA-B	SPP1	GPNMB	HLA-DRA	APOE	HLA-DRB5	VIM	FTL	TP53	HLA-A	CD8A	MMP9	PRLR
9	COL1A2	FN1	COL1A1	CD38	COL4A2	COL4A1	ISG15	COL3A1	ANLN	FGFR1	PTEN	TIMP1	TAGLN	BGN	MMP11
10	SULF1	HIF1A	FOXP3	MLPH	APOE	JUN	CD3G	MMP11	MMP12	SOX4	FABP7	CD5	PI3	SLC2A1	LAMC1
11	ICOS	CDH11	CD2	GATA3	CD40LG	S100A9	LRP2	CDKN2A	KRT18	MYL6	SULF1	KRT19	TMSB4X	ERBB2	S100A14
12	PECAM1	HSPG2	NOTCH1	GRB7	STAT5A	CDKN2A	ICOS	FOXC1	APOC1	SKAP1	CTSL	CD3G	HLA-DRB5	LRP2	COL4A1
13	FAP	CCNB1	TYMS	THY1	LILRB1	COL3A1	CD96	LDB2	CAPN13	CDK4	TRDC	FOS	CCNE1	SKAP1	ELF5
14	TFF2	ICAM1	SDC1	TTC6	MUC1	ANLN	PTEN	PTPRC	MYO10	NKG7	CDK6	CTLA4	DERL3	NF1	LGALS2
15	CST3	SOX18	NUF2	XCL1	SNAI2	GNG11	ISG20	MYL9	GATA3	LYPD6B	KIF23	CLU	TCF4	IL32	AHR
16	KRT18	MYL6	TFF1	KRT19	HSPB1	TMSB4X	RPSA	XBP1	IFITM3	S100A14	FASN	CD63	TFF3	RPL13	CCND1
17	LST1	FAT1	IGHG4	ACTG2	MAPK3	PTTG1	IGHG1	S100A8	ITGA6	CD14	CD24	IGKC	LYZ	FCN1	EIF3E
18	TPM2	CD69	LAMA1	CD79B	MYO10	IL2RA	PLVAP	CAPN13	KRT8	KIF23	PGR	TMEM45B	GNG11	RB1	C1QA

Table S8. Genes associated with the detected GEPs. Each GEP (rows) is represented by a list of 15 genes in descending contribution order.

Table S9. Moran's I analysis of genes that are spatially variable in excess to cell type spatial variability. The table is attached as a separate spreadsheet file.

Supplementary Text

Guidelines on how to choose parameters for InSituSeg and fine tune them

When applying InSituSeg to a field of view (FOV) with DAPI staining, fine tuning various parameters can improve the results obtained, in terms of the cell bodies detected.

The min_nuc parameter

The first parameter to fine tune is min_nuc parameter which determines the minimum percentile of pixel intensity for nuclei detection. The first run of InSituSeg will be performed with the default parameter for this value. After the first run with the default parameter, this parameter can be fine tuned using the matlab table labeled 'll1', which will be generated after the InSituSeg run and can be viewed using the matlab function sliceViewer. This table is printed in the lines of code following the Processing_Image function and preceding the step of splitting connected cells.

With the min_nuc parameter there is a trade off - since this parameter is the minimum detected intensity, setting a low value can result in detection of practically all nuclei voxels. Alas, this can also result in over detection of nuclei voxels, which can falsy merge nearby cells into one object (with one cell ID). On the other hand, setting a high value for this parameter will likely result in a clear separation between the nuclei, but some nuclei will likely remain undetected.

Therefore, if multiple nuclei in the FOV are combined and assigned to a single cell ID instead of being separated into distinct cell IDs, it is advisable to opt for a higher min_nuc value. A higher min_nuc value will result in detecting a smaller fraction of the nuclei voxels, therefore potentially leading to the separation of combined nuclei. Conversely, if some nuclei are not detected in the 'll1' output image, a lower min_nuc value should be set to encompass nuclei with lower intensities.

Note that connected cells can be split later in the splitting connected cells step.

The area_big_quant parameter

Another parameter requiring adjustment is area_big_quant, representing the minimum percentile of nuclei voxels for splitting potentially combined objects (i.e., the number of voxels for each nucleus is sorted from the nucleus with the smallest number of voxels to the largest,

and the area_big_quant parameter is the minimum percentile for the nucleus to be considered potentially combined). The nuclei that have a number of voxels above this percentile will proceed to the stage of splitting connected cells. The determination of the area_big_quant value relies on the 'S1' vector variable generated after the Processing_Image function. This vector captures the number of voxels of each nucleus area, and this vector is utilized in establishing the value of the area_big_quant parameter.

To estimate the 'ideal' value area_big_quant parameter, a manual examination can be helpful: examine the matlab table labeled 'll1' using the matlab function sliceViewer, and manually detect the largest nucleus which should remain unsplit, because it appears to be one cell. Using sliceViewer, hover over the aforementioned cell and locate the cell ID. Then locate this cell ID in the 'S1' vector to obtain the percentile of the number of voxels of this nucleus from all the nuclei. Ideally, the area_big_quant value should be slightly greater than that of the aforementioned cell. Consequently, this ensures that the specified nucleus, along with smaller nuclei falling below the area_big_quant threshold, will not undergo splitting.

The iterations_step parameter

To make sure that neighboring cells are not merged together, the pipeline performs an iterative procedure to split potentially connected cells. This is done by gradually increasing min_nuc for nuclei exceeding the area_big_quant cutoff, which in turn leads to the detection of stronger voxels, and therefore can allow the splitting of merged nuclei. During these iterations, the area_big_quant is fixed, and the user can adjust the iterations_step parameter.

For nuclei exceeding the threshold size (i.e., higher than the area_big_quant cutoff), the min_nuc parameter value undergoes iterative increments (as mentioned above), resulting in fewer pixels being selected as part of the nuclei and leading to the generation of split objects. In instances where certain nuclei remain unsplit even after reaching the maximum min_nuc value, it is advisable to decrease the iterations_step parameter (i.e., increase the number of iterations) for splitting connected cells. With a low number of iterations, there is a risk for a nucleus to be undetected beyond a specific iteration, primarily because of a high min_nuc threshold value. Prior to this iteration, the nucleus may maintain an overly large size due to the low min_nuc threshold. Hence, an intermediate jump is necessary to achieve a point where the large nucleus undergoes division into several smaller nuclei. It's important to be aware that a high iteration count results in a prolonged runtime.

The area_remove_quant parameter

An additional parameter to consider is the area_remove_quant, representing the maximum percentile of area for eliminating objects suspected of being noise (i.e., objects with a very low number of voxels are likely to be noise, and not real nuclei). The area_remove_quant parameter can be fine tuned using the second segmented image, denoted as matlab table labeled 'll_nuc'. This table will be generated after the InSituSeg run, and can be viewed using the matlab function sliceViewer. This segmented image is generated after the pipeline step of splitting connected cells. We recommend using the default area_remove_quant value, unless multiple small objects, which appear to be unrelated to cells, are observed in the 'll_nuc' image. In cases where visual inspection of the 'll_nuc' image still reveals small 'noisy' objects, or in cases where cells are overly split to multiple cells, the area_remove_quant value should be higher.

The min_som parameter

min_som represents the minimum percentile of pixel intensity required for the representation of cell bodies. The definition of min_som is very similar to that of min_nuc, and indeed, if the min_som parameter is set to be equal to min_nuc (this is not recommended…), only nuclei will be detected as the cell bodies. In contrast, setting the parameter to be lower than min_nuc, will allow detection of the hues surrounding the nuclei, thus allowing detection of the cell bodies. The min_som parameter can be fine tuned using the third segmented image, denoted as matlab table labeled 'll_som'. This table will be generated after the InSituSeg run, and can be viewed using the matlab function sliceViewer. As a general guideline, it is recommended to choose a min_som value that is approximately ten-twenty units smaller than the min_nuc value. Ultimately, the min_som parameter plays a key role in determining the cell body size. Thus, to augment the cell body size, one should decrease the min_som value, whereas to reduce the cell body size, a higher min_som value is preferable.

The medfiltmask parameter

The InSituSeg procedure starts with a pre-processing denoising step utilizing a median filter. During this stage, each pixel's intensity value is substituted with the median value of the neighboring pixels. The size of the window encompassing these pixels, denoting the number of pixels considered in the 3D volume, is defined as the 'medfiltmask' parameter. It is essential to calibrate this 'medfiltmask' parameter to the downsampled image size (see below), ensuring that it achieves a balance between image smoothing and resolution preservation. The

medfiltmask parameter can be fine tuned using an additional image, denoted as a matlab table labeled ‘img3d_fil’. This table will be generated during the InSituSeg run, in the ‘Filtering Median Filter’ part, and can be viewed using the matlab function sliceViewer. For instance, if no downsampling is applied (i.e., downsampling parameter equals 1), the 'medfiltmask' parameter is advised to be set to [9,9,9] for 2048 by 2048 pixels image. Conversely, if the downsampling parameter is set to 4 (see below), or similarly if the original image size is smaller than 2048 by 2048 pixels, we recommend using a 'medfiltmask' parameter value of [3,3,3].

The downsampling parameter

The full-resolution DAPI staining image of the designated field of view (FOV), typically in the form of an h5 or tiff file, is an input of InSituSeg. This image is read in MATLAB as a table. However, executing the pipeline with the original full-resolution image may consume a significant amount of time, potentially even hours for 150 serial sections of 2048 by 2048 pixel image. We have observed that downsampling the image by a factor of 2-4 yields nearly identical segmentation results while significantly reducing the runtime to minutes instead of hours. Consequently, we recommend utilizing the downsampling parameter for downsampling. This parameter is employed in the built-in MATLAB function 'imresize3,' which resizes a 3D volumetric intensity image by a factor defined by the downsampling parameter. For instance, if a value of 4 is set for the downsampling parameter, the resulting volume will be four times smaller than the original image volume (table). The entire InSituSeg process is then applied to the downsampled image table.

The closingmask parameter

The InSituSeg procedure starts with a pre-processing denoising step utilizing the ‘imclose’ Matlab function, which is morphological close operation. During this stage, the morphological close operation dilates the image and then erodes the dilated image using the same structuring element for both operations. This stage is useful for filling small holes in the image while preserving the shape and size of large holes and objects in the image. The size of the structuring element is defined as the 'closingmask' parameter. It is essential to calibrate this 'closingmask' parameter to the downsampled image size, ensuring that it achieves a balance between filling small holes and preserving separate cells. For instance, if no downsampling is applied (i.e., downsampling parameter equals 1), the 'closingmask' parameter is advised to be set to 3 for 2048 by 2048 pixels image. Conversely, if the downsampling parameter is set to 4, or similarly

if the original image size is smaller than 2048 by 2048 pixels, we recommend using a 'closingmask' parameter value of 1.

The openingmask parameter

The InSituSeg procedure starts with a pre-processing denoising step utilizing the ‘imopen’ Matlab function, a morphological opening operation. During this stage, the morphological opening operation erodes the image and then dilates the eroded image using the same structuring element for both operations. This stage is useful for removing small objects from the image while preserving the shape and size of larger objects in the image. The size of the structuring element is defined as the 'openingmask' parameter. It is essential to calibrate this 'openingmask' parameter to the downsampled image size, ensuring that it achieves a balance between removing small objects and preserving the cells. For instance, if no downsampling is applied (i.e., downsampling parameter equals 1), the 'openingmask' parameter is advised to be set to 3 for 2048 by 2048 pixels image. Conversely, if the downsampling parameter is set to 4, or similarly if the original image size is smaller than 2048 by 2048 pixels, we recommend using a 'openingmask' parameter value of 1.

References

Alon, Shahar, Daniel R. Goodwin, Anubhav Sinha, Asmamaw T. Wassie, Fei Chen, Evan R. Daugharthy, Yosuke Bando, et al. 2021. "Expansion Sequencing: Spatially Precise in Situ Transcriptomics in Intact Biological Systems." *Science (New York, N.Y.)* 371 (6528): eaax2656.

Batista, Gustavo E. A. P. A., Gustavo E. A. P. Batista, Ronaldo C. Prati, and Maria Carolina Monard. 2004. "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data." *ACM SIGKDD Explorations Newsletter*. https://doi.org/10.1145/1007730.1007735.

Becht, Etienne, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. 2018. "Dimensionality Reduction for Visualizing Single-Cell Data Using UMAP." *Nature Biotechnology*, December. https://doi.org/10.1038/nbt.4314.

Berg, Stuart, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, et al. 2019. "Ilastik: Interactive Machine Learning for (Bio)Image Analysis." *Nature Methods* 16 (12): 1226–32.

Berger, Daniel R., H. Sebastian Seung, and Jeff W. Lichtman. 2018. "VAST (Volume Annotation and Segmentation Tool): Efficient Manual and Semi-Automatic Labeling of Large 3D Image Stacks." *Frontiers in Neural Circuits* 12 (October): 88.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. "SMOTE: Synthetic Minority Over-Sampling Technique." *Journal of Artificial Intelligence Research*. https://doi.org/10.1613/jair.953.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785.

Dorogush, Anna Veronika, Vasily Ershov, and Andrey Gulin. 2018. "CatBoost: Gradient Boosting with Categorical Features Support." https://doi.org/10.48550/ARXIV.1810.11363.

Greenwald, Noah F., Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, et al. 2022. "Whole-Cell Segmentation of Tissue Images with Human-Level Performance Using Large-Scale Data Annotation and Deep Learning." *Nature Biotechnology* 40 (4): 555–65.

Hao, Yuhan, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck 3rd, Shiwei Zheng, Andrew Butler, Maddie J. Lee, et al. 2021. "Integrated Analysis of Multimodal Single-Cell Data." *Cell* 184 (13): 3573-3587.e29.

He, Yong, Yunlong Meng, Hui Gong, Shangbin Chen, Bin Zhang, Wenxiang Ding, Qingming Luo, and Anan Li. 2014. "An Automated Three-Dimensional Detection and Segmentation Method for Touching Cells by Integrating Concave Points Clustering and Random Walker Algorithm." *PloS One* 9 (8): e104437.

Ho, Tin Kam. 2002. “Random Decision Forests.” In *Proceedings of 3rd International Conference on Document Analysis and Recognition*. IEEE Comput. Soc. Press. https://doi.org/10.1109/icdar.1995.598994.

Hu, Jian, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J. Irwin, Edward B. Lee, Russell T. Shinohara, and Mingyao Li. 2021. “SpaGCN: Integrating Gene Expression, Spatial Location and Histology to Identify Spatial Domains and Spatially Variable Genes by Graph Convolutional Network.” *Nature Methods* 18 (11): 1342–51.

Kotliar, Dylan, Adrian Veres, M. Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A. Melton, and Pardis C. Sabeti. 2019. “Identifying Gene Expression Programs of Cell-Type Identity and Cellular Activity with Single-Cell RNA-Seq.” *ELife* 8 (July). https://doi.org/10.7554/eLife.43803.

Kowal, Marek, Michał Żejmo, Marcin Skobel, Józef Korbicz, and Roman Monczak. 2020. “Cell Nuclei Segmentation in Cytological Images Using Convolutional Neural Network and Seeded Watershed Algorithm.” *Journal of Digital Imaging* 33 (1): 231–42.

Love, Michael I., Wolfgang Huber, and Simon Anders. 2014. “Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2.” *Genome Biology* 15 (12): 550.

Lundberg, Scott, and Su-In Lee. 2017. “A Unified Approach to Interpreting Model Predictions.” https://doi.org/10.48550/ARXIV.1705.07874.

Miller, Brendan F., Dhananjay Bambah-Mukku, Catherine Dulac, Xiaowei Zhuang, and Jean Fan. 2021. “Characterizing Spatial Gene Expression Heterogeneity in Spatially Resolved Single-Cell Transcriptomic Data with Nonuniform Cellular Densities.” *Genome Research* 31 (10): 1843–55.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2012. “Scikit-Learn: Machine Learning in Python.” https://doi.org/10.48550/ARXIV.1201.0490.

Quinlan, J. R. 1986. “Induction of Decision Trees.” *Machine Learning*. https://doi.org/10.1007/bf00116251.

Stringer, Carsen, Tim Wang, Michalis Michaelos, and Marius Pachitariu. 2021. “Cellpose: A Generalist Algorithm for Cellular Segmentation.” *Nature Methods* 18 (1): 100–106.

Tan, Aik Choon, and David Gilbert. 2003. “Ensemble Machine Learning on Gene Expression Data for Cancer Classification.” *Applied Bioinformatics* 2 (3 Suppl): S75-83.

Zhang, Meng, Stephen W. Eichhorn, Brian Zingg, Zizhen Yao, Kaelan Cotter, Hongkui Zeng, Hongwei Dong, and Xiaowei Zhuang. 2021. “Spatially Resolved Cell Atlas of the Mouse Primary Motor Cortex by MERFISH.” *Nature* 598 (7879): 137–43.