From puncta to retinas: Verbose workflow of data cleaning and transformations

1) Human annotation using the lasso tool in Xenium explorer was performed for each slice by visualizing RBPMS and GRM6 puncta, with the workflow consisting of drawing the lasso region through the GRM6 rich region or null space flanking an RBPMS rich region. RBPMS is also strongly expressed in pigment epithelium cells, so in ambiguous regions or regions suspected of being pigment epithelium OPN1m/s were also visualized to clarify decisions.
    a) Lasso files stored in **/media/sam/Data2/xenium_rbpms_coordinates**
2) Lasso regions were consolidated by slide and polygons were cleaned up in **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/Baysor_hand_drawn_ROIs.ipynb** these polygons were then used to subset the relevant t**ranscripts.parquet** file and to store the subset parquet in a new folder in the user specified output directory
    a) Output directory with each slide and slice subdirectories **/media/sam/Data2/baysor_rbpms_consolidated**
3) Baysor prediction of segmentation was performed in the same **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/Baysor_hand_drawn_ROIs.ipynb** as well as in the limited version of this script **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/Baysor_hand_drawn_ROIs_RunOnly.ipynb**
    a) The main segmentation files generated are:
        i) **os.path.join(output_path, 'segmentation_polygons_2d.json')**
        ii) **os.path.join(output_path, 'segmentation_polygons_3d.json')**
        iii) **os.path.join(output_path, 'segmentation.csv')**
    b) The baysor command executed in bash via python's **subprocess** command is: **baysor run "{parq_path}" -c "/home/sam/baysor_analysis/xenium.toml" -o "{output_path}" --prior-segmentation-confidence=.2 :cell_id**
4) The **segmentation.csv** is converted into an expression matrix in R via **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/Baysor_segmentation_cleaner_counter.R**
    a) The counts are first filtered in the following order by:
        i) Removing all transcripts baysor predicted are noise
        ii) Removing all transcripts with a qv < 20
        iii) Filtering non-target probe transcripts
        iv) Removing all baysor predicted cells with fewer than 100 transcripts
    b) In addition to counting each transcript associated with a cell, this function also computes simple geometric descriptions of the transcript point cloud.
        i) Volume is computed as the volume of the convex hull by geometry::convhulln as per this stack overflow example
        ii) x_range = max(x) - min(x),
        iii) y_range = max(y) - min(y),
        iv) z_range = max(z) - min(z),
        v) rect_vol is computed as the product of the x, y and z ranges
        vi) Elongation is computed as the ratio of the first two principal components
        vii) Flatness is computed as the ratio of the second two principal components

viii) Compactness is the ratio of the largest to smallest PCA axis
ix) Sphericity is the ratio of the minimum and maximum PCA

c) The output file is stored as **'expression_matrix.csv'** in the same folder as the respective baysor outputs

5) The **'expression_matrix.csv'** is then passed through CutleNet inference using **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/Baysor_hand_drawn_ROIs_CuttleNet_InferenceOnly.ipynb** and the **'expression_matrix.csv'** has the column Prediction added to it which contains the argmax prediction of the network. All other classification predictions for the respective cells are stored in **'{path[:-21]}ClassProbabilities.csv'** in the same folder
6) DAPI statistics are extracted from the respective slide's maximum projection tif file by **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/baysor_dapi_extractor.ipynb** specifically, the respective **morphology_focus.ome.tif** is subset by each slice's **segmentation_polygons_2d.json** and the mean, std, min, and max DAPI values within a given cell's polygon are stored in **os.path.join(output_path,"dapi_statistics.csv")** while a visualization of the polygon and the DAPI image are stored in **os.path.join(output_path, "dapi_baysor_segs.tif")**
7) All **dapi_statistics.csv** and **expression_matrix.csv** files are merged together into a single file in R by **/home/sam/scRNAseq/Xenium/XenExplorer_Registration/baysor_retina_stats.R** the output final csv file is stored as **/media/sam/Data2/baysor_rbpms_consolidated/all_retinas_prediction_expmat.csv**
8) This same R script next filters the data to contain only RGCs, and further subsets these to contain only cells with a normalized mean DAPI value above 0.05, where the normalization is defined as X-min(DAPI_min)/ (max(DAPI_max) -min(DAPI_min)) grouped by the respective slide to account for illumination differences across the slide, as well as by cells whose volume is greater than 500 um^3 which was the lower limit observed in Bae 2018. Finally, these filtered dataframes have the nearest neighbor distance and ID computed and added as respective columns
    a) The filtered files are respectively **/media/sam/Data2/baysor_rbpms_consolidated/rgc_retinas_prediction_expmat.csv** and **/media/sam/Data2/baysor_rbpms_consolidated/filtered_rgc_prediction_expmat.csv**
9) This same R script also performs simple descriptive statistics including the NNRI and voroni tessellation index calculations, as well as the respective panels for main manuscript Figures 1 and 2
10) The X, and Y centroid positions in **Filtered_rgc_prediction_expmat.csv** are then transformed in **Igor** by manual alignment of the stained tissue transparencies to produce the final dataset