Instructions for Python, Matlab, and R codes used in the manuscript

‘Python algorithm’ Folder:

- **SubRegion_generate.py**: segment gigapixel whole slide images into processable small tiles. In this study each tile (subregion) is 200*200 pixels. The output is for cell extraction.

- **Color_method.py**: assign coordinate to every pixel of downsampled whole slide images. The output is used to generate equidistant sections.

- (examplar script) **Case1_cKDTree_CD3CD4.py**: calculate DoC score for CD3 and CD4 for Case 1. The script uses KDTree algorithm (implemented in C language) to accelerate the process to search nearest neighbors.

- **Case1 subfolder**: CD3+, CD4+, CD8+, CD20+, and FoxP3+ cell coordinates for Case 1. Coordinate files are used as input for cKDTree algorithm.

‘Matlab algorithms’ folder:

- **‘Registration’ subfolder**:
  - **Case_1_reginfo:**
    - Transformation matrices (included in ‘_SecInfo’ folders)
    - Annotations for CD4 (as the reference to compare the local and global registration algorithm)
    - CD4+ cells coordinate
  - **Contour:**
    - Registered contour under local and global registration.
  - **Case1_Contour_globalReg.m:** register the original annotations for target whole slides images using global registration transformation matrices.
  - **Case1_Contour_localReg.m:** register the original annotations for target whole slides images using local registration transformation matrices.
  - **Case1_Coords_Registration.m:** register the cell coordinates part by part using the

local transformation matrices.

  - **Reg_Integration.m:** Integrate registered (local registration) subregions into a complete image.
  - **Registration_validation.R:** Evaluate the performance of local registration by calculating the DICE score for both local and global method and then compare using Wilcoxon test.

- **‘Segmentation subfolder**:
  - **‘Sources’ subfolder:**
    - Case1 subfolder:
      - CD3 subfolder:
        - ‘Coords’ subfolder: local cell coordinate for each subregion.
        - ‘Processed subregions’ subfolder: subregion images labeled with algorithmic detected cells.
        - ‘subregions subfolder’: raw subregion images.
      - CD4, 8, 20, FoxP3 subfolders: same as above.
    - Case2 – 5 subfolders: same as above.
  - **Main.m:** main script for cell extraction and coordinate record. Capable of batch processing.
  - **Coordinate_record.m:** Matlab function script, record cell coordinate while processing each subregion.
  - **rgb2cmyk.m:** Matlab function script, converts RGB channel to CMYK channel.
  - **Tumor_Reconstruct.m:** take all subregion coordinate files as input, convert local coordinate into global coordinate and then combine.
  - **regionSelect.m:** rules out preliminary cells which failed to pass the morphometric filters from the cell candidate pool.
- **RandomSelect.R**: select processed subregion images using systemic sampling methods to

evaluate the performance of segmentation algorithm.

- **'Spatial statistics' folder:**

- **MiFunction.R:** function script.

- **fractalAnalysis.R:** perform fractal analysis to determine window size for spatial model fitting (Fig.S4).

- **IF_CT_generate.R:** obtain invasive front and tumor regions.

- **N_generate_and_area.R:** obtain normal regions and calculate areas for IF, CT, and N.

- **Annotation_vis.R:** visualize region annotations (Fig.5A).

- **Section_generate.R:** generate equidistant sections (rings) towards the tumor boundary (Fig.4C).

- **Intra-hetero.R:** visualize cell density variations across each specimen (Figs.5D, S8) and generate density vs distance profile from N to CT (immune infiltration profile).

- **Density_Profile_WSI.R:** calculate standard error for each density along the immune infiltration profile (Figs.5C, S7).

- **Density_Profile_IF.R:** generate density-distance profile along the invasive front and associated standard errors (Figs.4D, 5E, S9).

- **Shannon_Entropy.R:** use a modified version of Shannon's Entropy to gauge the spatial heterogeneity across the specimen (Fig.6).

- **subregion_fit.R:** perform spatial model fitting for a typical clustered point pattern and a typical non-clustered point pattern (Fig.7).

- **Cluster_stats.R:** perform clustering and morphometric analysis (Figs.8, S12).

- **Chisq-test-WSI.R**: perform chi-square test for densities (over equidistant sections generated by script 'Section_generate.R', Fig.S5).

- **Chisq-test-IF.R:** perform chi-square test for densities (over regions generated by Voronoi tessellation, Fig.S6).

- **Correlation_analysis.R:** perform correlation analysis (Figs.9, 10).