Integrative Imaging and Molecular Diagnostics Lab
Spatial analysis

Leon H. Kloker
Institute for Computational and Mathematical Engineering
Stanford University
leonkl@stanford.edu

1 Brief overview of project

The goal of this project is to implement a range of spatial biomarkers, based on the single cell map of H&E images (cores), that can be correlated with clinical endpoints in order predict treatment response. The single cell map in this setting consists of a csv file containing the x, y coordinates, areas as well as cell types of all cells in the core. The dataset used here consists of cores of 86 patients and their censored overall survival in months post-surgery.

2 Biomarkers

The amount of cell types appearing in the single cell map is from here on denoted as T. Moreover, N denotes the amount of actual biomarker values arising from one of the following biomarker class definitions. The name of the biomarkers is equal to the name of the function calculating the biomarker expressions in the `core.py` file for consistency between the code and this report.

2.1 First-order

1. `fraction_cell_type`:
   $N = T$
   The fraction of each cell type is calculated for each core as
   #(cells of TYPE in core) / #(cells in core).
2. `density_cell_type`:
   $N = T + 1$
   The areal density of each cell type and the overall cell density is calculated for each core as
   #(cells of TYPE in core) / #(area of core)
3. `area_cell_type`:
   $N = T + 1$
   The average area of each cell type and the overall average cell area is calculated.

2.2 Higher-order

1. `neighbouring_cell_type_distance_cutoff`:
   $N = T(T + 1)$
   The average amount of cells of type 2 within a given radius around cells of type 1 is

calculated. This average is calculated for each possible combination of cell types. Moreover, the average amount of cells regardless of their type within a given radius around cells of type1 is also calculated.

2. `neighbouring_cell_type_amount_cutoff`
   $N = T^2$
   The average fraction of cells of type 2 among the k-nearest cells around cells of type 1 is calculated. Thus, here, instead of cells within a given radius, the k-nearest cells are considered. The fraction is calculated for each possible combination of cell types.
3. `smallest_distance_cell_type`
   $N = T^2$
   The average smallest distance between cells of type 1 to cells of type 2 is calculated for each possible combination of cell types.
4. `g_function`
   $N = T(T+1)$
   The G-function for each possible combination of cell types is calculated. The G-function evaluated at a given radius is equal to the amount of shortest distances from cells of type 1 to cells of type 2 that are smaller than said radius. Then, the integral of the difference between the empirical G-function and the theoretical G-function based on complete spatial randomness is calculated and used as a biomarker.
5. `k_function`
   $N = T(T+1)$
   The K-function for each possible combination of cell types is calculated. The K-function evaluated at a given radius is equal to the fraction of distances from cells of type 1 to cells of type 2 that are smaller than said radius. Then, the integral of the difference between the empirical K-function and the theoretical K-function based on complete spatial randomness is calculated and used as a biomarker.

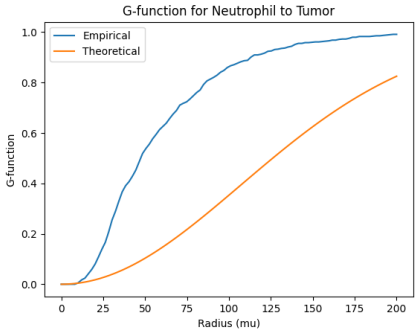

Figure 1: Empirical and theoretical G-function plotted over the radius for the core A-1.

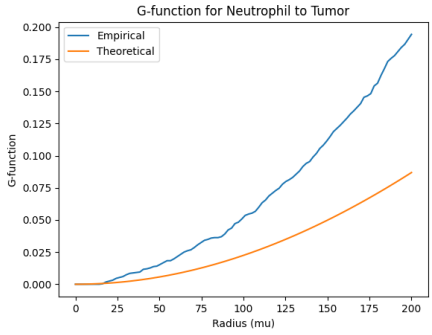

Figure 2: Empirical and theoretical G-function plotted over the radius for the core A-3.

2.3 Cellular neighbourhoods

Cellular neighbourhoods are constructed by calculating the distribution of cell types among the k-nearest cells to each cell in the core. The distribution vector associated with every cell is then used to cluster the cells of all cores in the entire dataset into M different neighbourhoods (e.g. Lymphocyte enriched neighbourhoods). Based on the neighbourhood each cell is assigned to, new biomarkers can be implemented. For inference on unseen data, the centroids of the clusters can be fixed and the cells in the previously unseen core can thus be clustered according to proximity of their distribution vectors to the existing centroids.

1. `fraction_cellular_neighbourhoods`
   $N = M(T+1)$
   The fraction of cells regardless of their type that belong to a certain cellular neighbourhood are calculated. Moreover, this fraction is also calculated for each cell individually.

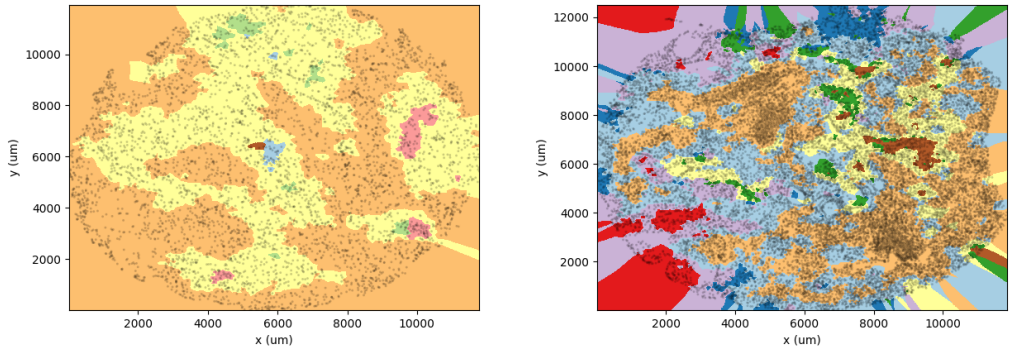

Figure 3: The cells are clustered into 8 different cellular neighbourhoods based on the cell type distribution among their 50 nearest neighbours. The plots show the Voronoi diagram of the core A-3 and A-7 on the left and right, respectively, according to the cellular neighbourhood each cell belongs to. The cell locations are plotted as transparent black points.

2. `entropy_cellular_neighbourhoods`
   $N = T + 1$
   The entropy of the distribution of cellular neighbourhoods among all cells as well as only the cells of a certain type is calculated.

3. `g_function_cellular_neighbourhoods`
   $N = M(M + 1)$ The G-function for each possible combination of neighbourhood types is calculated (i.e. similar to the calculation of the G-function previously, now however, the cells are grouped by neighbourhood instead of cell type). Then, the integral of the difference between the empirical G-function and the theoretical G-function based on complete spatial randomness is calculated and used as a biomarker.

4. `k_function_cellular_neighbourhoods`
   $N = M(M + 1)$ The K-function for each possible combination of neighbourhood types (analogous to the cellular neighbourhood G-function) is calculated. Then, the integral of the difference between the empirical K-function and the theoretical K-function based on complete spatial randomness is calculated and used as a biomarker.

2.4 Brief summary of all biomarkers

The biomarkers under consideration range from simpel first-order to more sophisticated spatial features that encapsulate the distribution of cell types with respect to each other. For the four distinct cell types appearing in this dataset, we end up with 14 first-order features consisting of cell type fractions, areal densities and average areas.

Second-order features are made up of the fractions of cell type 1 that are in vicinity of cell type 2 or the average smallest distance from cells of type 1 to type 2. Additionally, the G-function and Ripley's K-function can be calculated for each possible combination of cell types and compared to the theoretical estimate based on complete spatial randomness in order to quantify the clustering of cell types. This leads to 92 additional second-order biomarkers.

Finally, so as to provide an even more fine-grained description of the tumor microenvironment, cellular neighbourhoods can be defined. Here, cells are clustered into 8 distinct cellular neighbourhoods based on the distribution of cell types among the 50 closest cells surrounding each cell. Now, features such as the fraction of a certain cell type within a neighbourhood or the information entropy of the neighbourhood distribution are considered. Additionally, the neighbourhood assignment can be interpreted as a re-labeling of the cells based on which the G- and K-functions are recalculated. The clustering of the cells into 8 different neighbourhoods yields 189 auxiliary biomarkers purely based on cellular neighbourhoods leaving us with 295 biomarkers overall.

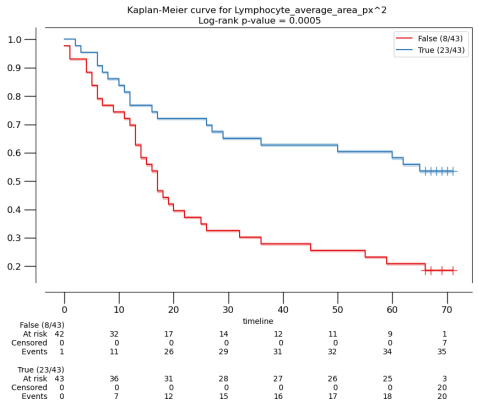

Figure 4: Kaplan-Meier curve when the patient population is split into two groups, one with an average Lymphocyte area bigger (True) than the entire dataset median and the other one with a smaller Lymphocyte area (False).

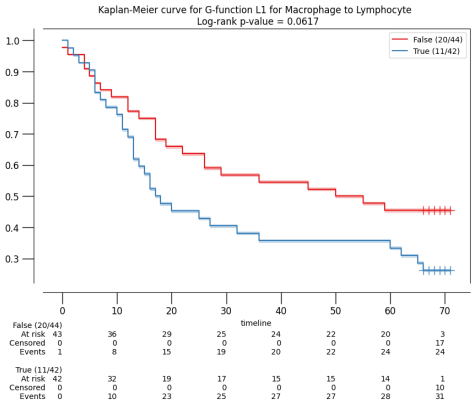

Figure 5: Kaplan-Meier curve when the patient population is split into two groups according to the median integral difference of the empirical G-function to its theoretical curve based on the smallest distances from Macrophages to Lymphocytes.

3 Survival analysis tools

3.1 Kaplan-Meier curve

The Kaplan-Meier curve is a statistical tool commonly used in medical and health research to estimate the survival rates of patients over time. It serves as a visualization of the survival or failure function in time-to-event analysis, where the event in question might be death, remission, or recurrence of disease.

The log-rank test is used alongside the Kaplan-Meier curve to compare the survival distributions of two or more groups. The log-rank p-value, in particular, helps determine whether there's a statistically significant difference between these groups. If the p-value is less than the commonly used threshold of 0.05, it suggests that there's a significant difference in survival among the groups being compared.

The Kaplan-Meier curve can be plotted for each biomarker and varies depending on where the threshold for splitting the patient population into two groups is set. Figure 3.1 and 3.1 depict the Kaplan-Meier curves for the average Lymphocyte area and the G-1 function between Macrophages and Lymphocytes when the patients are split by the median biomarker expression.

3.2 Cox regression

Cox regression, also known as the Cox proportional hazards model, is another tool used in survival analysis that allows for the investigation of how multiple variables can simultaneously influence the time to a specified event. Unlike Kaplan-Meier curves, which require splitting the patient population based on an arbitrarily chosen threshold of a biomarker expression, Cox regression models are able to incorporate the entire spectrum of patient data without the need for such a binary split.

In order to investigate how strongly each biomarker impacts survival outcome, a univariate cox model is built for each biomarker yielding the hazard ratio and cox p-value, both quantifying the significance of the influence on survival. Employing this tool with the dataset containing 86 patients and their overall survival post-surgery yields 19 biomarkers with a cox p-value of less than 0.1 and 7 biomarkers with $p < 0.05$ as listed in table 3.2.

Biomarker	Cox Regression p-value
Lymphocyte_average_area_mu^2	0.0328
Neutrophil_average_area_mu^2	0.0470
Neutrophil_density_mu^2	0.0783
Fraction of Lymphocyte among 50 closest cells next to Macrophage	0.0701
Average amount of Tumor cells within 50mu of Macrophage	0.0909
G-function L1 for Lymphocyte to Lymphocyte	0.0861
G-function L1 for Macrophage to Lymphocyte	0.0336
G-function L1 for Macrophage to Tumor	0.0701
K-function L1 for Lymphocyte to Neutrophil	0.0884
K-function L1 for Neutrophil to Neutrophil	0.0738
K-function L1 for Tumor to Neutrophil	0.0834
G-function L1 for neighbourhood 1 / 8 to neighbourhood 2 / 8	0.0186
G-function L1 for neighbourhood 2 / 8 to neighbourhood 4 / 8	0.0811
G-function L1 for neighbourhood 7 / 8 to neighbourhood 1 / 8	0.0634
G-function L1 for neighbourhood 7 / 8 to neighbourhood 7 / 8	0.0467
G-function L1 for neighbourhood 8 / 8 to neighbourhood 7 / 8	0.0801
Fraction of Lymphocyte that belong to neighbourhood 3 / 8	0.0274
Fraction of Macrophage that belong to neighbourhood 3 / 8	0.0322
Fraction of Macrophage that belong to neighbourhood 4 / 8	0.0713

4 Implementation details

The `main.ipynb` file contains an exemplary use case of the core and dataset classes in order to load a new dataset from a directory and calculate a range of biomarker expressions and their p-values.

4.1 Core class

The `core.py` file defines the core class, which is used to load cell coordinates of all cells in a core, the cell types, cell areas and patient survival from a csv file. Moreover, all the biomarker functions are defined in this class and function definitions that are not biomarkers always end on an underscore to distinguish between the two types of functions.

4.2 Dataset class

The `dataset.py` file defines the dataset class, which is used to load multiple cores of an entire dataset at once. In addition to the cores, the patient survival can be loaded into the dataset as well. From then on, the `calculate_biomarker` function can be used to calculate biomarker expressions of every core in the dataset for either specific classes of biomarkers or all implemented biomarkers at once. The function `log_rank_test` can be used to calculate the log-rank p-value when the population is split by median, mean and optimal threshold to minimize the p-value. The function `univariate_cox_model` can be used to calculate the hazard ratio and cox p-value using a cox model that only considers one single biomarker as variable at once. The `kaplan_meier` function can be used to create a Kaplan-Meier plot of a specific biomarker. The `save` and `load` functions can be used to save and load a dataset to and from a file with all calculated p-values and biomarker expressions.