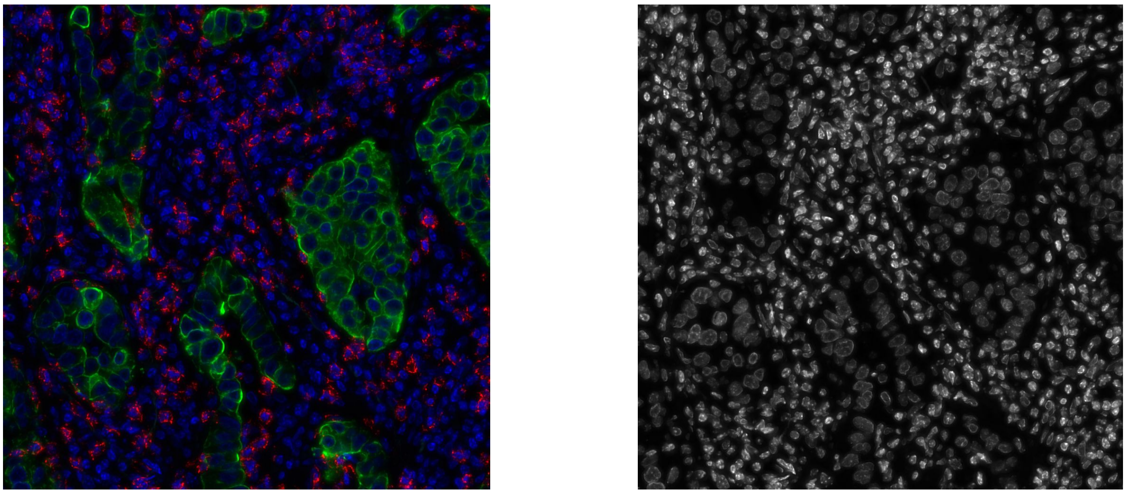
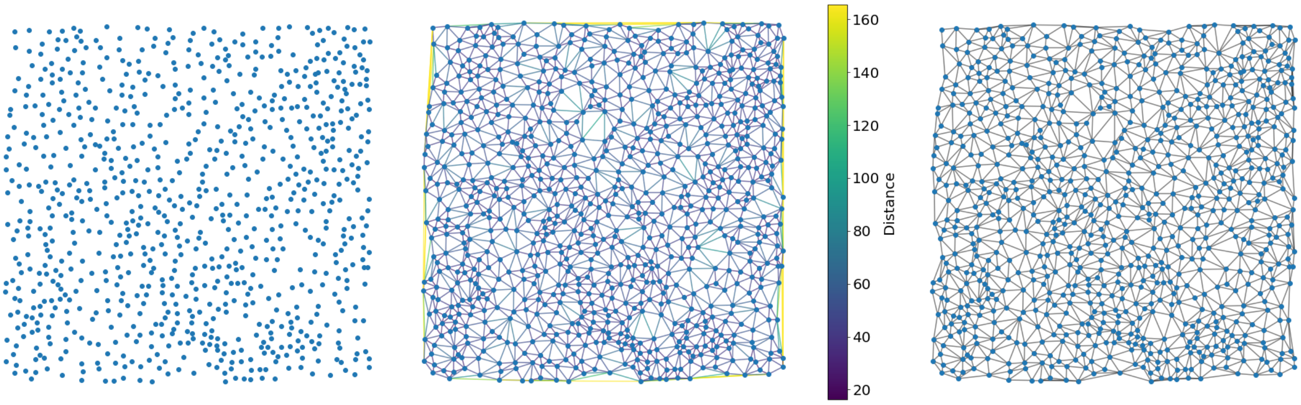
Tysserand - Fast reconstruction of spatial networks from bioimages
Supplementary Materials

Alexis Coullomb and Vera Pancaldi
Centre de Recherches en Cancérologie de Toulouse, Toulouse, France



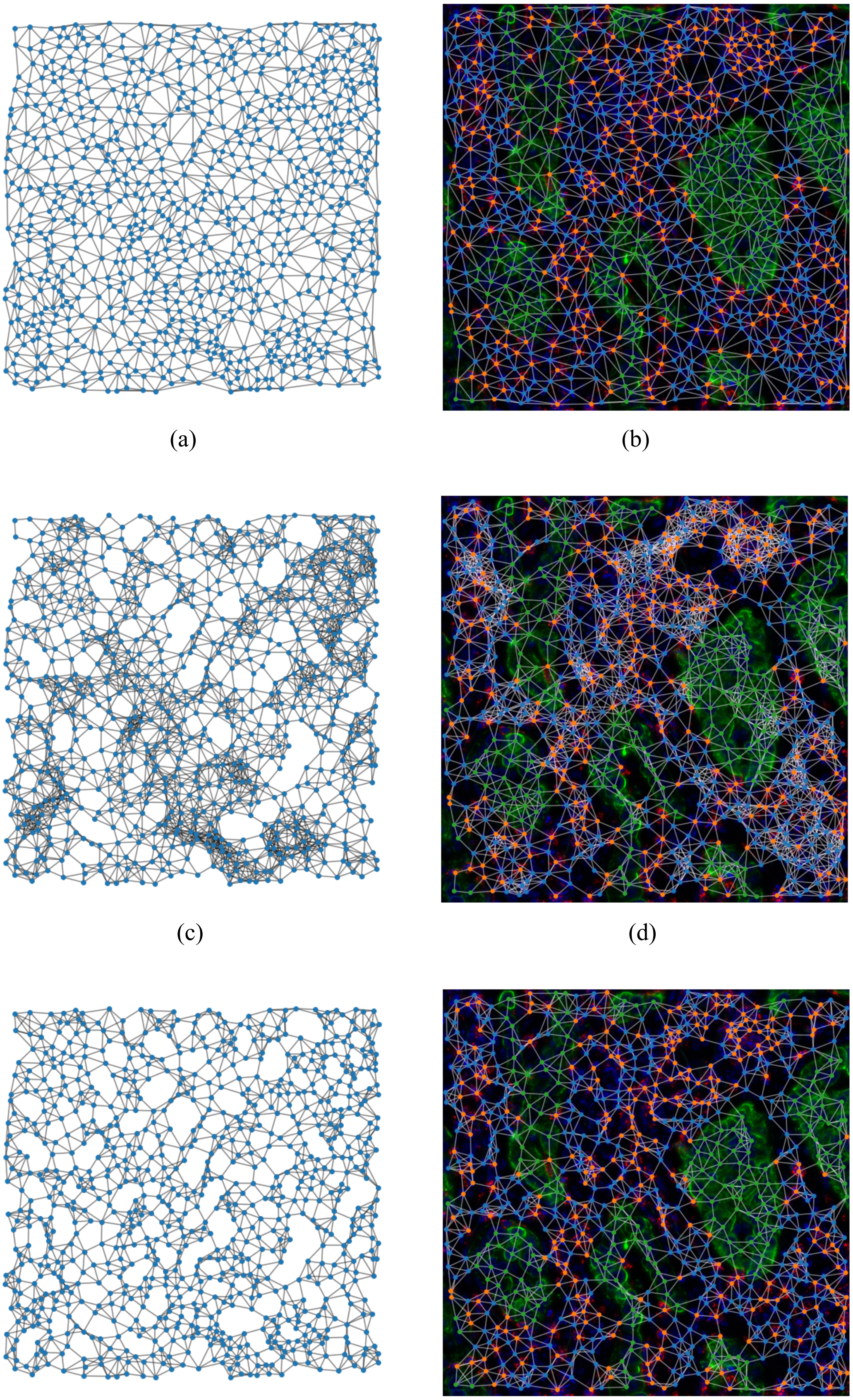
(a) (b)

Figure S1: Exemplary tissue image (a) extracted from a multiplex immunofluorescence (mIF) *Whole Slide Image* of a lung tumor biopsy. Pseudo colors are blue: DNA, green: PanCK (tumor), red: CD8 (cytotoxic T cells). Nodes positions are defined by manual marking of nuclei (b), nodes class (blue / green / orange color coded) are defined from markers in image (a).



(a) (b) (c)

Figure S2: (a) Scatter plot of 100 nodes, their coordinates are given by a 100x2 array. (b) Delaunay triangulation can produce long edges artifacts, most often at the border of the samples (yellow edges). Users can trim edges longer than a manually defined distance using distances plots like in (b), or define this threshold automatically at a distance corresponding to the 99th (or other) percentile of lengths. (c) The resulting cleaned network after edge trimming.

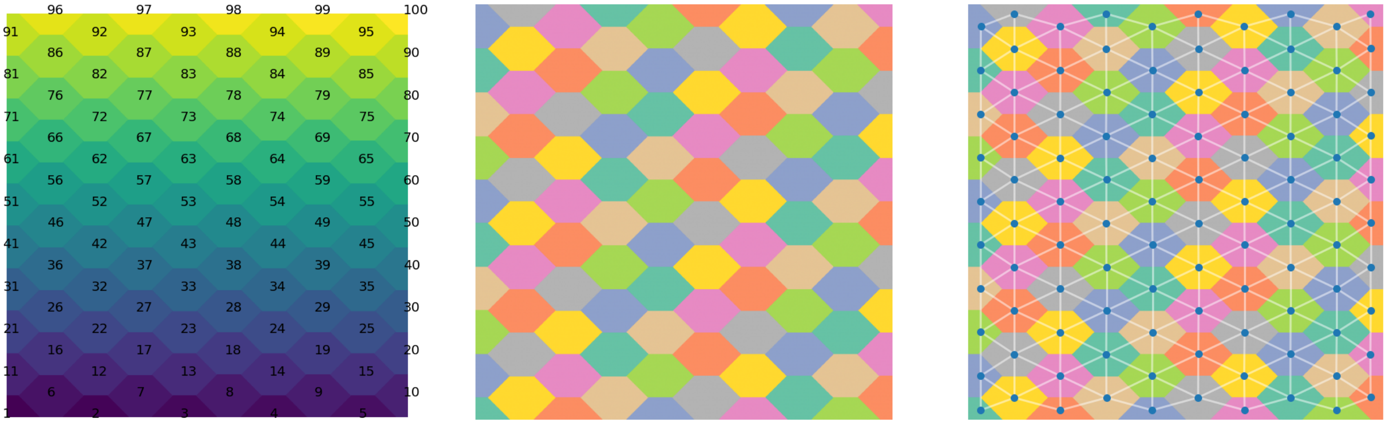


(a) (b)

(c) (d)

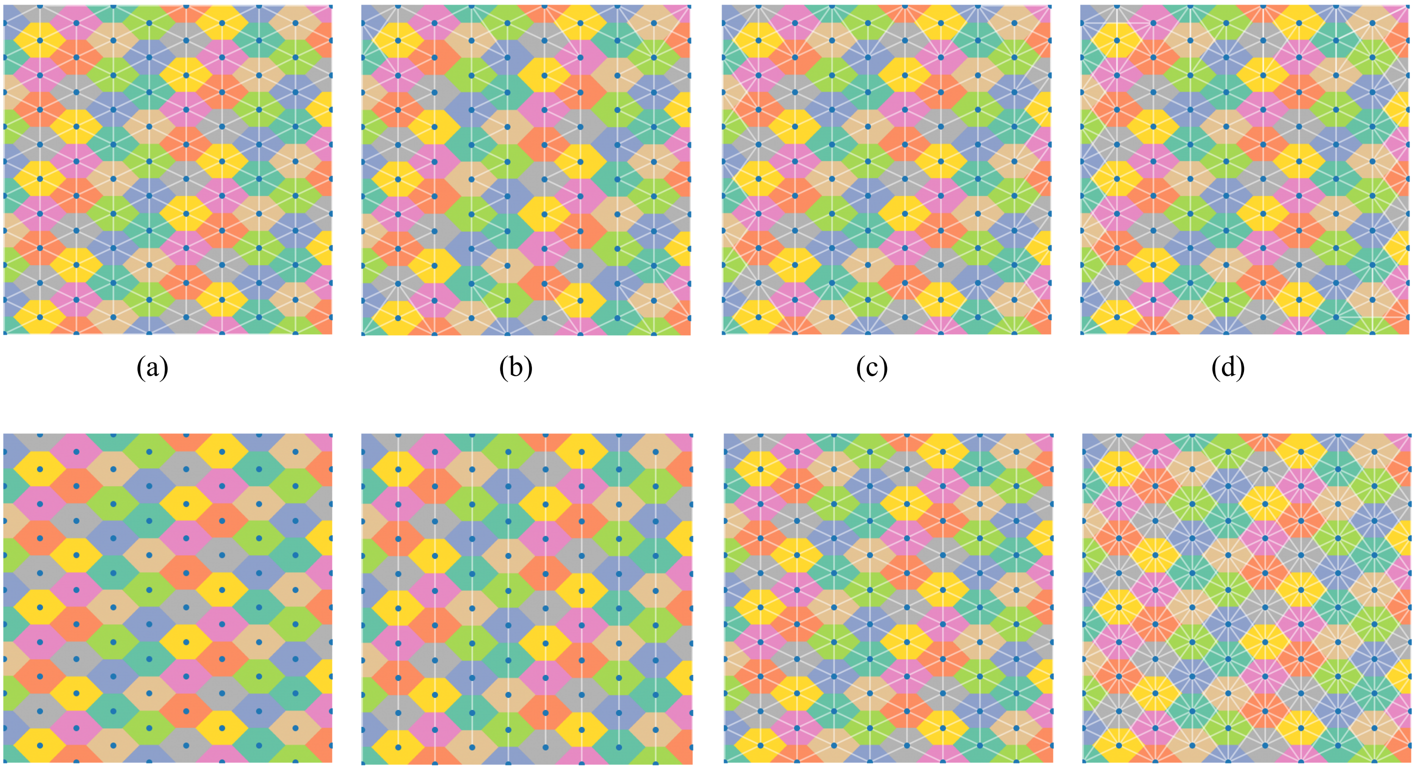
(e) (f)

Figure S3: Delaunay triangulation after edge trimming (a) and resulting network superimposed on the mIF image (b). Radial distance neighbors (r=60) network (c) and network superimposed on the mIF image (d). The variability in cell density induces very lowly and very highly connected areas. k-nearest neighbors (k=6) network (e) and network superimposed on the mIF image (f). Some areas are less connected than expected given the apparent cells contacts, and some edges go *through* neighbors.



(a) (b) (c)

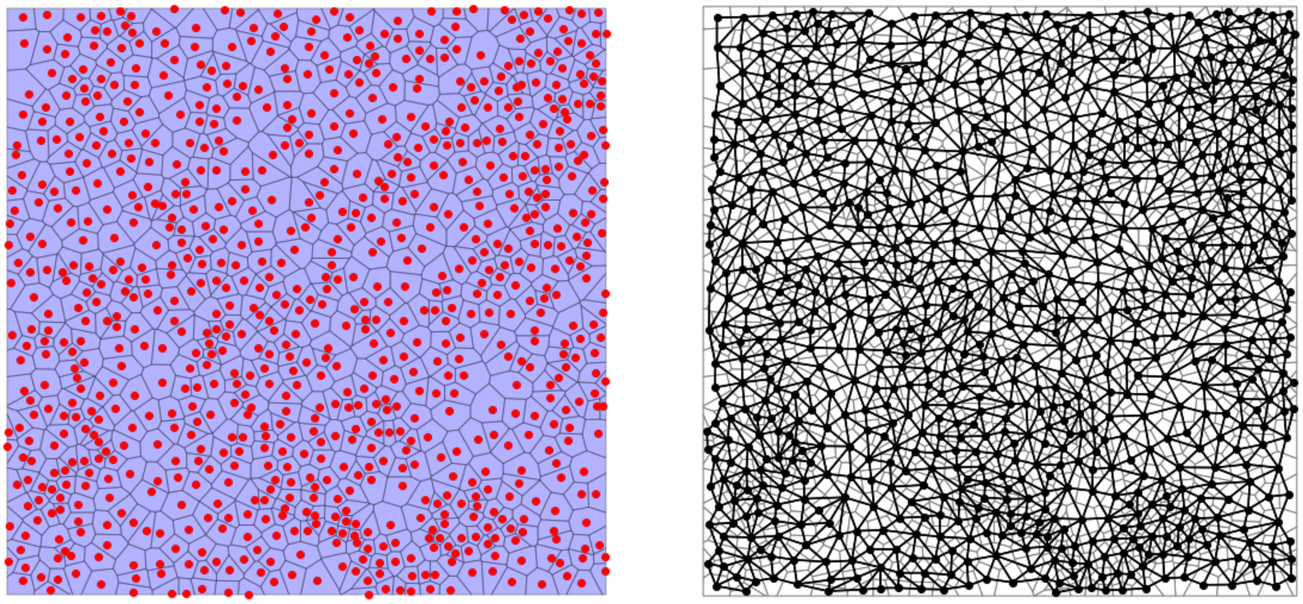
Figure S4: A segmentation image (a) is a 2D array containing areas of integers from 0 to K resulting from the detection of K areas (cells, nuclei, …). Here, areas are color-coded, and their integer value is displayed as the contrast between adjacent areas can be very low. We use a scikit-image function to display areas with a better contrast (b). The area contact method uses scikit-images functions to detect for each area the areas that are in direct contact or closer than a user-defined distance (c). On this generated set of tiles mimicking a tissue, the area contact method appears well suited to reconstruct tissue networks.



(a) (b) (c) (d)

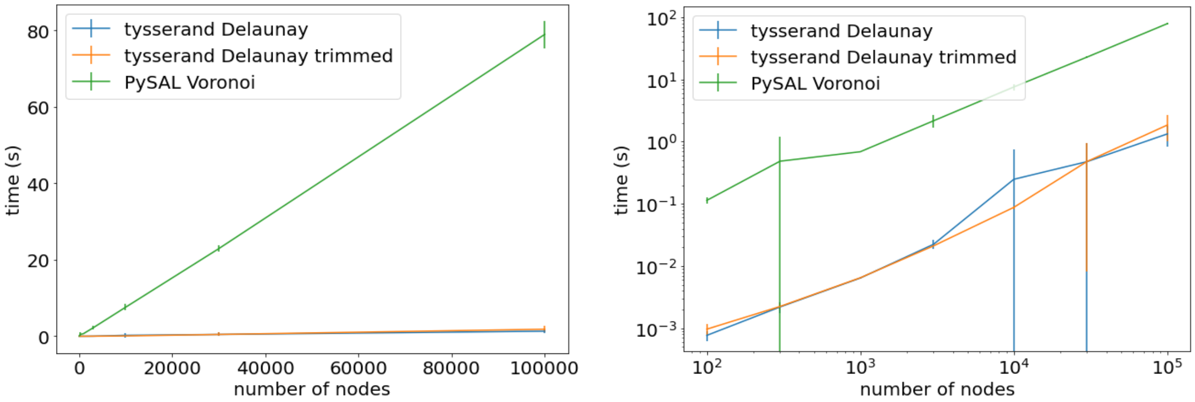
(e) (f) (g) (h)

Figure S5: Reconstruction methods for coordinates applied to generated tissue. Delaunay reconstruction (a) produces a network very similar to the area contact method, with only long edges artifact at the border of the network that can be easily corrected. K-nearest neighbor method with k=5 (b), k=6 (c) and k=7 (d). The theoretically ideal value of k is 6, but it produces artifacts from nodes on the border of the sample, these artifacts reach a substantial depth in the sample. To avoid them, with k=5 the core of the network is too poorly connected. With a higher value of k=7 artifacts from the sample border are more numerous, but supplementary edges in the core of the sample are overlapping, edge doublets are eliminated for all network construction methods. The networks obtained with the radial distance method are very sensitive to the radius parameter (r). For r=50 (e) there is no edge between nodes, for r=55 (f) the connectivity is too low, nodes are connected in one direction due to the sample asymmetry. For r= 75 (g) the resulting network is identical to the one obtained with the area contact method. On this very regular lattice this result is robust with higher values of r, until a threshold where many edges are defined between nodes, as with r=100 (h).



(a) (b)

Figure S6: (a) PySAL implementation of Voronoi tessellation results in 'shape objects' that are common in geosciences. (b) Contacts are computed between these objects and they define the Voronoi diagram (equivalent to Delaunay triangulation). Thanks to the use of contacts between shape objects, there are no long edges artifacts.



(a) (b)

Figure S7: Performance comparison between Delaunay/Voronoi implementations of tysserand (based on scipy) and PySAL for various sizes of sets of nodes. Tested sizes are 100, 300, 1000, 3000, 10000, 30000 and 100000 nodes. For each size, 10 random sets are generated and both implementations are tested on the same sets. Graphs display on uniform scale (a) or log-log scale (b) the mean of execution time (s), error bars are the standard deviation. Since tysserand's method can produce artifacts, the additional step of network correction is also considered ("tysserand Delaunay trimmed"). Tysserand implementation with network correction is always at least 42 times faster than PySAL implementation, which becomes an important point to consider when analysing multiple samples with several thousands of detected cells for instance.