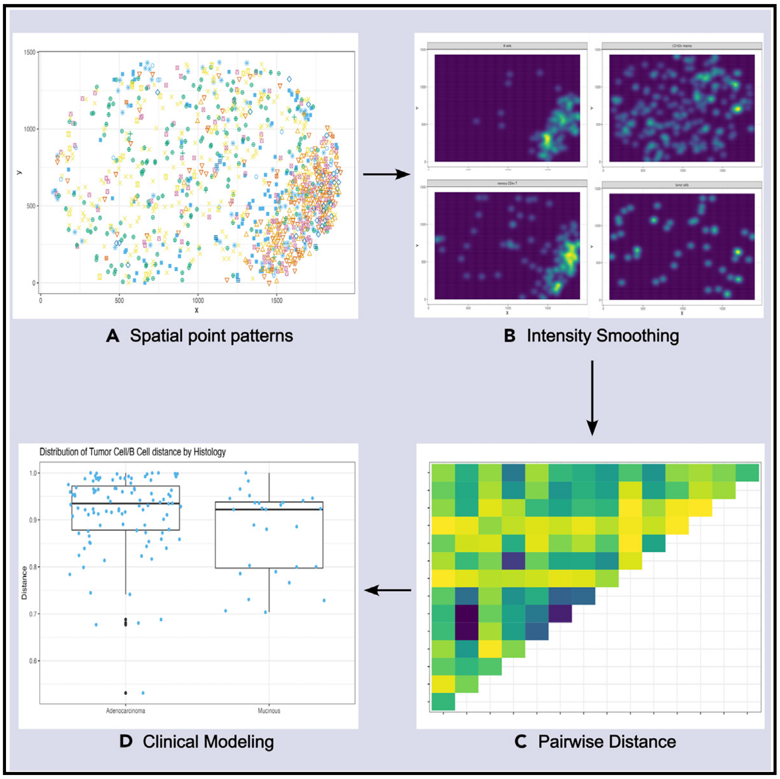
Descriptor

Patterns

DIMPLE: An R package to quantify, visualize, and model spatial cellular interactions from multiplex imaging with distance matrices

Graphical abstract



Highlights

- DIMPLE is an R package to analyze and visualize multiplex imaging data
- DIMPLE provides a scalable framework to quantify cellular interactions
- Statistical modeling links tumor microenvironment to patient-level outcomes

Authors

Maria Masotti, Nathaniel Osher, Joel Eliason, Arvind Rao, Veerabhadran Baladandayuthapani

Correspondence

mmasotti@umich.edu

In brief

The tumor microenvironment consists of multiple cell types that can interact in complex ways. Multiplex imaging technologies such as Vectra Polaris or PhenoCycler can be used to identify the precise spatial locations and phenotypes of cells in tissue samples of the tumor microenvironment. Masotti et al. present a software package, DIMPLE, that provides an end-to-end pipeline to quantify, visualize, and model spatial cellular interactions in multiplex imaging data.

Masotti et al., 2023, Patterns *4*, 100879
December 8, 2023 © 2023 The Author(s).
https://doi.org/10.1016/j.patter.2023.100879

Descriptor

DIMPLE: An R package to quantify, visualize, and model spatial cellular interactions from multiplex imaging with distance matrices

Maria Masotti,[1,3,*] Nathaniel Osher,[1] Joel Eliason,[2] Arvind Rao,[1,2] and Veerabhadran Baladandayuthapani[1,2]
[1]University of Michigan, Department of Biostatistics, Ann Arbor, MI 48109, USA
[2]University of Michigan, Department of Computational Medicine and Bioinformatics, Ann Arbor, MI 48109, USA
[3]Lead contact
*Correspondence: mmasotti@umich.edu

https://doi.org/10.1016/j.patter.2023.100879

THE BIGGER PICTURE The tumor microenvironment (TME), the ecosystem of immune cells, extracellular matrix, blood vessels, and other cells that surrounds a tumor, is emerging as the next frontier in cancer research. It is firmly established that the presence and prevalence of specific immune cells within and around the tumor can predict patient outcomes, including their response to treatment and the progression of cancer. Multiplex imaging (MI) technologies such as PhenoImager, PhenoCycler, MIBI, and others provide a detailed view of the TME. These rapidly evolving technologies enable the discrimination of numerous cell types while preserving their spatial context. This allows for quantification of spatial cellular interactions or the tendency of cell types to co-locate. Evidence is mounting that these cellular interactions, beyond mere presence and prevalence, are associated with patient outcomes.

1 2 3 4 5 **Development/Pre-production:** Data science output has been rolled out/validated across multiple domains/problems

SUMMARY

A major challenge in the spatial analysis of multiplex imaging (MI) data is choosing how to measure cellular spatial interactions and how to relate them to patient outcomes. Existing methods to quantify cell-cell interactions do not scale to the rapidly evolving technical landscape, where both the number of unique cell types and the number of images in a dataset may be large. We propose a scalable analytical framework and accompanying R package, `DIMPLE`, to quantify, visualize, and model cell-cell interactions in the TME. By applying `DIMPLE` to publicly available MI data, we uncover statistically significant associations between image-level measures of cell-cell interactions and patient-level covariates.

INTRODUCTION

The tumor microenvironment (TME) refers to the complex network of cells and other structures surrounding a tumor. This microenvironment is essential for the growth, survival, and metastasis of cancer cells.[1] Understanding the mechanisms underlying the TME is crucial for developing new and more effective cancer therapies.[2,3] Recent breakthroughs in multiplex imaging (MI) of tissues have allowed researchers to simultaneously visualize and quantify multiple biomarkers in a single tissue sample while preserving their spatial information.[4] Specific MI technologies include PhenoImager, formerly known as Vectra; PhenoCycler, formerly known as codetection by indexing (CODEX); multiplexed ion beam imaging by time-of-flight (MIBI); imaging mass cytometry (IMC); matrix-assisted laser desorption ionization mass spectrometry imaging (MALDI-MSI); and digital spatial analysis (GeoMx/DSP/CosMx). These technologies produce high-resolution maps of multiple functional and phenotypic markers on a single tissue section. Each single cell in the tissue can be phenotyped based on the marker intensities. One of the key analytical goals for high-resolution images from these technologies is to understand the interactions between different cells in the TME and how they contribute to tumor growth, metastasis, and drug resistance.

Spatial analysis of MI-derived data may offer insight into how cellular crosstalk and heterogeneity affect cancer prognoses and responses to treatment. Several recent studies discovered novel cellular interactions in the TME. An MI study on the lung adenocarcinoma TME in 153 patients with resected tumors found that expression of major histocompatibility complex (MHC) class II associates with tumor and immune interaction within the TME.[5] This suggests that cancer-cell-specific expression of MHC class II may represent a biomarker for the immune system's recognition and activation against the tumor. An MI study of ovarian cancer found that the proximity between tumor-associated macrophages and B cells or CD4 T cells significantly correlated with overall survival.[6] An MI study of tissue samples from the colorectal cancer (CRC) invasive front found that co-localization of PD-1+ CD-4+ T cells with granulocytes was positively correlated with survival in a high-risk patient subset.[7]

Despite structurally similar data, each of these studies attempts to quantify cellular interaction with profoundly different techniques. In the study of the lung cancer TME,[5] Euclidean distances between individual tumor cells and the nearest immune cell phenotypes were calculated. In the study of ovarian cancer,[6] an interaction variable based on the number of one cell type within a certain *r* of another was calculated. This measure was binned and used downstream in survival outcome modeling. In the study of CRC,[7] a complex clustering procedure was used to partition each image into a neighborhood. The neighborhoods were roughly defined by the prevalence of certain cells. Then, summary statistics were computed in each neighborhood and compared between neighborhoods and images. These studies demonstrate a lack of consensus on an approach for measuring cellular interaction in the TME.

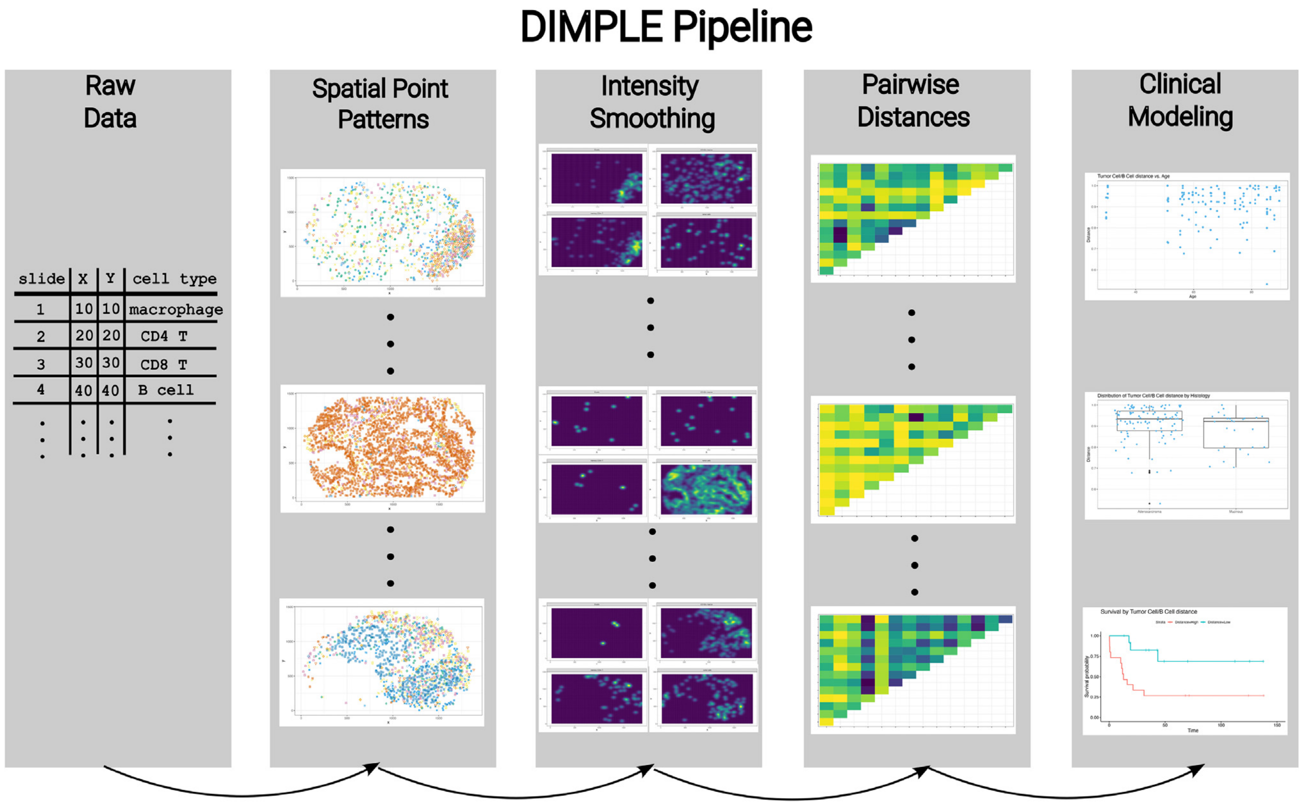
A handful of software tools exist to compute cell-level measures of interaction. One of the most widely used tools, the R package "spatstat,"[8] is a comprehensive set of functions for spatial point patterns. The R packages "spatialTIME"[9] and "spicyR"[10] also compute cell-level interaction measures and were designed specifically to store and analyze data from MI. These measures of interaction based at the cell level, such as Ripley's K,[11] Besag's L,[12] Marcon's M,[13] and nearest-neighbor distance,[14] may not readily scale with increasing numbers of cell types or images. Specifically, pairwise interactions grow quadratically with the number of cell types, so discovery-driven approaches can quickly become computationally infeasible. These cell-level measures are functions of a radius (*r*), which must be pre-specified in order to calculate a univariate measure of interaction for downstream analysis. Alternatively, more complex modeling procedures are required to study the heterogeneity of cell-cell interactions using these functions over a range of *r* as functional data inputs.[15] The values of these measures are not interpretable alone and require comparison with the function estimated under complete spatial randomness (CSR). The value of the function under CSR is highly sensitive to spatial inhomogeneity, uneven distribution of cells across the image, and "holes" or areas of missing data.[16] Unfortunately, these features are common in MI data, so calculation of the CSR must be performed by permutation for each individual image, which can, again, be quite computationally expensive.

To address these challenges, we present `DIMPLE` (distance matrices for multiplex imaging) along with an R package[17] and an accompanying R Shiny app, filling the gap for a scalable and statistically savvy data science toolkit for researchers conducting MI experiments (see Figure 1). Briefly, `DIMPLE` first computes a spatially smooth non-parametric kernel density estimate (KDE) of each cell type for each image, the parameters of which are user defined. Then, a distance or similarity metric is applied to each unique pairwise combination of the KDEs for each image. These univariate distance metrics are organized in matrix form and can be visualized using heatmaps or networks. Users have the option to attach patient-level metadata, allowing users to test associations between distance metrics and patient covariates or outcomes. `DIMPLE` is designed to accommodate data from a variety of sources, requiring only image IDs, spatial coordinates, and a cell type or marker status for each cell. The plotting functions and accompanying Shiny app generate publication-quality figures to visualize heterogeneity in the cell-cell interactions across patient-level covariates. Our framework is computationally efficient and requires no permutations to interpret measures of cellular interaction. The image-level and optional patient-level data along with computed intensities and distance matrices are organized with all relevant information in an R S3 object. `DIMPLE` is designed to appeal to novice R users. All core analyses including organizing and storing data, computing intensity smoothed point patterns, calculating distances, conducting statistical inference via regression modeling, and visualization can be done within the `DIMPLE` package. See Figure S4 for a flowchart of the core functions. For more savvy R users, outputs from the package can be easily used downstream in more complex statistical analysis such as survival analysis, mixed-effects modeling, clustering, and variable selection. The accompanying R Shiny app allows users to visually explore a `MltplxExperiment` object in a point-and-click fashion. The accompanying R package is free and open source and was designed to adhere to the FAIR principles; see section S4 of the supplemental information for details.

To illustrate the functionality of the `DIMPLE` software, we will utilize publicly available MI data of the lung adenocarcinoma TME in 153 patients with resected tumors from Johnson et al.[5] In the results, we describe each of the analysis steps (columns of Figure 1) using these data as examples. Results from our application of `DIMPLE` to these data reveal statistically significant associations between different types of immune cells and CK+ (pan-cytokeratin positive) tumor cells. These findings agree with findings from Johnson et al.,[5] who found that expression of MHC class II associates with tumor and immune interaction within the TME. We conclude the discussion with a tutorial for generating `DIMPLE` distance matrices at quantile-specific partitions of the MI data in generating and visualizing quantile-specific distance matrices, an overview of our R Shiny app in using the DIMPLE Shiny app, and a note on simulating MI data in simulation of MI data.

RESULTS

In this section, we describe how to use the `DIMPLE` package and provide complete coding examples. The data can be downloaded using the `VectraPolarisData` package.[18] See section S1 of the supplemental information for the R code used to download and process the data for use in `DIMPLE`. The overall analyses pipeline is organized into 4 main steps. Step 1 entails reading the data and storing them as a `MltplxExperiment`. Step 2 comprises the generation of


DIMPLE Pipeline
Raw Data
Spatial Point Patterns
Intensity Smoothing
Pairwise Distances
Clinical Modeling
Tumor Cell/B Cell distance vs. Age
Distribution of Tumor Cell/B Cell distance by Histology
Survival by Tumor Cell/B Cell distance

slide	X	Y	cell type
1	10	10	macrophage
2	20	20	CD4 T
3	30	30	CD8 T
4	40	40	B cell
⋮	⋮	⋮	⋮

Figure 1. Overview of DIMPLE pipeline, from raw data to clinical analysis
First, data are supplied to DIMPLE in a simple format. Second, a point pattern representation of the data is generated for each image. Third, cell-type intensity surfaces are estimated. Fourth, distances are computed from the intensities for each pair of cell types. Finally, pairwise distances may be easily used downstream in statistical analyses in combination with patient metadata.

intensity surface estimates for each cell type and image in the dataset. Next, step 3 entails generating distance matrix representations of cellular interaction for each image. Lastly, step 4 involves using the pairwise distances in downstream analysis. This section concludes with a tutorial of generating quantile-specific distance matrices to investigate patterns of cellular interaction at various partitions of the image.

Step 1: Input data and construct MltplxExperiment

To generate measures of cellular interaction using the DIMPLE package, the minimal amount of information needed is four vectors with length equal to the total number of cells in the MI dataset.

(1)x, the x coordinates of each cell
(2)y, the y coordinates of each cell
(3)marks, the cell type of each cell
(4)slide_id, the ID of the image that each cell is from

The new_MltplxExperiment function takes at least those four inputs: x, y, marks, and slide_id. It returns an S3 object of class MltplxExperiment. This object is a list of all of the slides in the dataset, each with its corresponding cell types and locations stored for easy access, along with additional metadata. It can be indexed the way a standard list is indexed, using single- and double-bracket expressions as demonstrated below.

We will store the example data in a MltplxExperiment called lung_experiment.

```
library(DIMPLE)
lung_EXPERIMENT = DIMPLE::NEW_MLTPLXEXPERIMENT(
X = CELL_X_COORDS,
Y = CELL_Y_COORDS,
MARKS = CELL_MARKS,
SLIDE_ID = SLIDE_IDS)
```

Each slide in the dataset is stored as a MltplxObject within the MltplxExperiment object. A MltplxObject represents a labeled collection of cell types and locations for each slide in the dataset. The S3 method plot is implemented for MltplxObject, which allows for quick inspection for a given slide. The following code demonstrates how to plot a MltplxObject, with the resulting plot output shown in Figure 2.

```
plot(lung_experiment[[1]])
plot(lung_experiment[[2]])
```

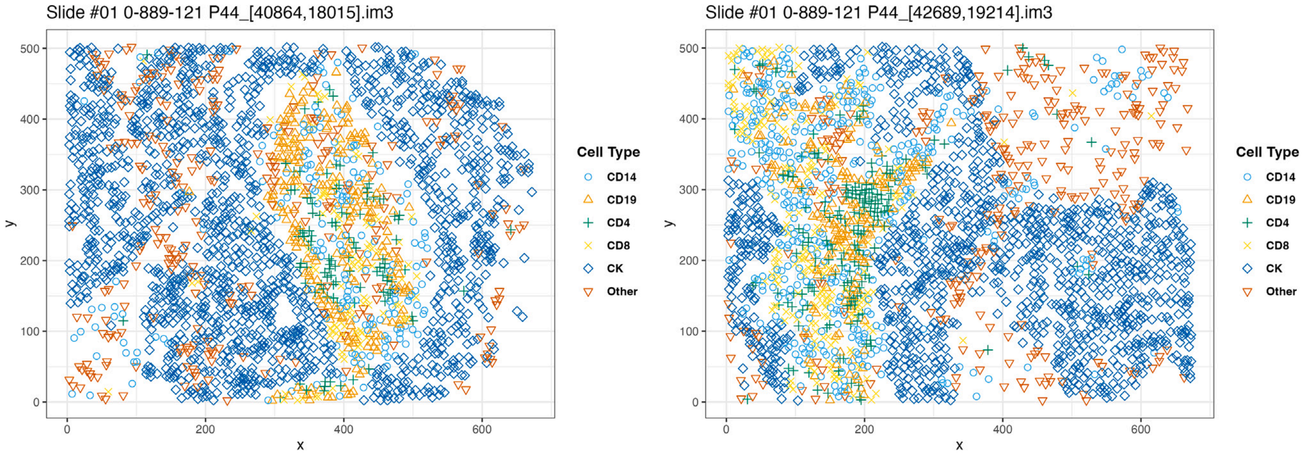


Figure 2. A point pattern plot of the first two images contained in `lung_experiment` generated with the `plot()` function

Combinations of color and shape represent the cell phenotype. These images come from the same patient, revealing heterogeneity within patient and over space—both hallmarks of MI data.

Step 2: Generating and visualizing cell-type intensities

The intensity of a point process can be thought of as an average, or first moment. The intensity function can be estimated non-parametrically by kernel estimation. The `DIMPLE` package uses an isotropic Gaussian kernel and Diggle's correction[19,20] to reduce the bias from edge effects. There are two arguments necessary to generate cell intensities: `ps` and `bw`. `ps` controls the size of the grid at which the intensity estimates will be generated. `ps` is short for "pixel size," since the resulting intensities can often resemble highly pixelated images. `bw` controls the smoothing of the resulting pixels, i.e., the degree to which the values of certain pixels should resemble their neighbors. Larger values of `bw` will result in "smoother" intensity surfaces. The smoothing bandwidth can be tuned by the user and should be chosen with some prior biological knowledge about the "radius of influence" that a given cell type has within a tissue. The calculated distance between cell types is sensitive to the choice of bandwidth, which can be seen in the simulation study (section S3 of the supplemental information). We note that in application to MI, the *r* at which users often look for cell-cell interaction is between 30 and 60 μm, as most cells in the TME have an *r* of between 5 and 20 μm. Thus, a smoothing bandwidth between 20 and 60 would indeed be appropriate for analysis of MI data.

`DIMPLE` makes it trivial to generate intensity grids for points of all types for entire `MltplxExperiment` objects as well as individual `MltplxObject` objects. To expedite computations, users may optionally implement intensity smoothing and calculation of distances in parallel in the manner shown below. This code also demonstrates how to plot the resulting intensity surfaces; see Figure 3 for the resulting plot output.

```
# uncomment below to use multiple cores to calcu-
late intensity
# plan(multisession(workers=2L))
lung_experiment = update_intensity(lung_experiment,
ps = 10,
bw = 30)
plot(lung_experiment[[1]]$mltplx_intensity)
plot(lung_experiment[[2]]$mltplx_intensity)
```

Step 3: Generating and visualizing pairwise distance matrices

Having generated appropriate intensity functions, a reasonable question to ask is how similar or different are the distributions of cells of different types within a given slide? For example, the intensity surfaces in the previous example seem to suggest that, for this specific patient, while there is considerable overlap between the positions of CD4+ and CD19+ cells, there is very little overlap between either of these types and CK+ cells.

To formalize this intuition in a way that can allow comparison across different slides, one can employ measures of distance between the intensities of different cell types within a given slide. Although any user-defined distance metric can be used, we prefer the Jensen-Shannon distance (JSD), which is a method of measuring similarity between two probability distributions.[21] It is a symmetrized and smoothed version of the Kullback-Leibler divergence (KLD). The JSD takes the square root of the Jensen-Shannon divergence so that it fulfills the axioms of a metric. The JSD is our preferred metric for this task because it is bounded by 0 (perfect overlap) and 1 (complete separation) and symmetric and does not suffer from inflation due to the presence of zeros (holes in the image). The JSD is defined by

$$JSD(P||Q) = \sqrt{\frac{1}{2}KLD(P||\mathcal{M}) + \frac{1}{2}KLD(Q||\mathcal{M})}$$

$$KLD(P||Q) = \sum_{x \in \mathcal{X}} P(x)\log_2\left(\frac{P(x)}{Q(x)}\right),$$

where P and Q are probability distributions defined on the same sample space $\mathcal{X}$ and $\mathcal{M} = \frac{1}{2}(P+Q)$.

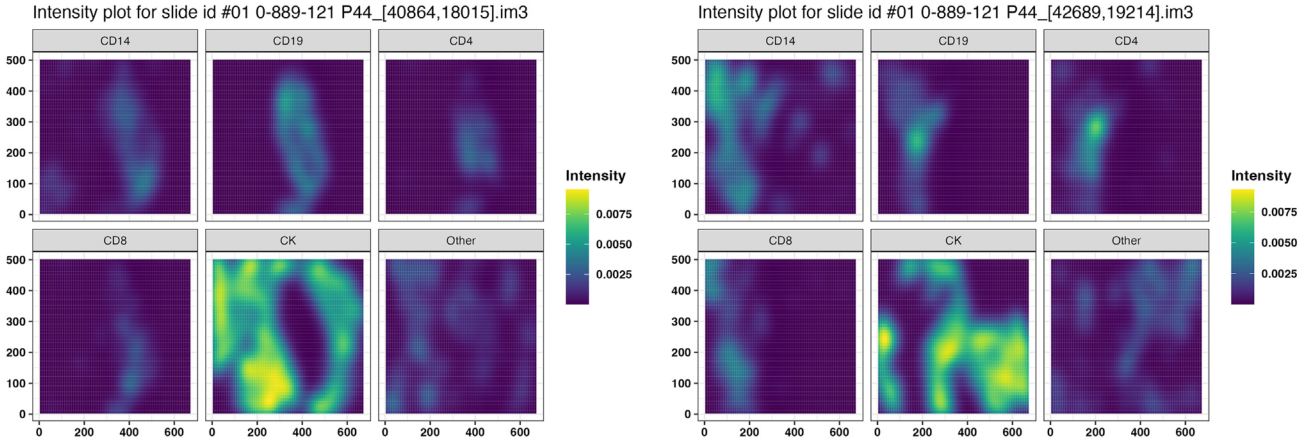


Figure 3. Estimated intensity surfaces for each cell phenotype from the two point patterns generated using the code above
Color represents intensity, with brighter color indicating greater intensity values.

In our case, P and Q are 2-dimensional spatial densities, or intensities, of two cell types on the same image. As the intensity functions are unknown, we will use their Gaussian KDEs, $\widehat{\lambda}^{(t)}$ and $\widehat{\lambda}^{(t')}$, where t and t' are two different cell types. To normalize the intensities, we will divide each by their sum. So, our equation for the JSD between the intensity of cell type t and cell type t' for one image is

$$\widehat{JSD}^{(t,t')} = \sqrt{\frac{1}{2}\sum_{u}\widehat{\lambda}^{(t)}(u)\log_2\left(\frac{2\widehat{\lambda}^{(t)}(u)}{\widehat{\lambda}^{(t)}(u)+\widehat{\lambda}^{(t')}(u)}\right)+\frac{1}{2}\sum_{u}\widehat{\lambda}^{(t')}(u)\log_2\left(\frac{2\widehat{\lambda}^{(t')}(u)}{\widehat{\lambda}^{(t)}(u)+\widehat{\lambda}^{(t')}(u)}\right)}.$$

where the us are the points at which the KDEs are estimated and are determined by the `ps` argument.

In addition to its favorable theoretical properties, the JSD is robust to varying radii of cell-cell interaction and the presence of holes in the image. We show through a simulation study that the JSD can capture cell-cell distance when two cell types are interacting at a close r versus a large r. Further, when a hole of data is deleted from the image, the JSD is minimally affected. See section S3 of the supplemental information for details on the simulation study.

Having computed intensities for each cell type in a `MltplxExperiment` object, pairwise distances can be added easily by specifying a distance metric. For a single image, we can construct an upper triangular or symmetric distance matrix by calculating $\widehat{JSD}^{(t,t')}$ for all $\binom{T}{2}$ sets of pairs where there are T unique cell types. Here, we use `jsd` to add distance matrices to the `MltplxExperiment`:

```
# uncomment below to use multiple cores to calcu-
late distance
# plan(multisession(workers=2L))
lung_experiment = update_dist(lung_experiment,
dist_metric = jsd)
plot_dist_matrix(lung_experiment[[1]])
plot_dist_matrix(lung_experiment[[2]])
```

The `dist_metric` can be user defined. Any function that takes in two vectors of the same length as an argument and produces a single scalar can be used in place of `jsd`. For example, a simple Euclidean distance measure could be used.

```
euclidean <- function(a, b) {
sqrt(sum((a - b)^2))
}
# The following code, if run, would compute the
distance using
# the previously defined function:
# lung_experiment = update_dist(lung_experiment,
# dist_metric = euclidean)
```

The resulting distance matrices in Figure 4 formalize our previous intuition. Since higher values of JSD indicate more different distributions (and vice versa for smaller values), it does indeed seem that there is quite a bit of overlap between the CD19+ and CD4+ cells, while the distributions of these cells are quite far from those of the CK+ cells in the two images within this patient. Note that we only plot the upper diagonal of the matrix. This is because a symmetric distance metric is employed. We also omit the diagonal because the distance from one cell type to itself will be zero regardless of the distance metric used. Boxplots of image-level

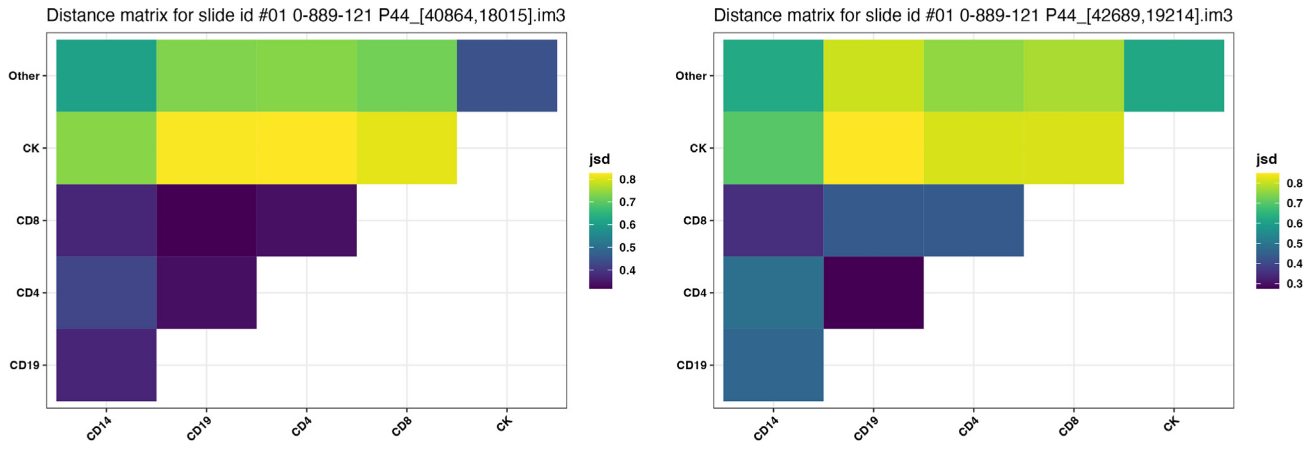


Figure 4. Estimated pairwise distances for the two point patterns generated using the code above
Color represents JSD, with brighter color indicating greater distance.

tumor-to-CD8 distances reveal significant within- and between-patient heterogeneity, as in Figure 5.

We conclude this section by noting that the changes made to the `lung_experiment` object can be discerned at a glance by simply invoking the `print` function on the object, implicitly or explicitly.

```
# equivalent to ''print(lung_experiment)''
lung_experiment
##MltplxExperiment with 761 slides
##Intensities generated with pixel size 10 and
bandwidth 30
##Distance matrices generated with jsd
##No attached metadata
```

Step 4: Adding patient metadata and visualizing and modeling associations

Patient-level data can also be stored in the `MltplxExperiment` for the purpose of visualizing and modeling cellular interactions with patient outcomes. Patient metadata can be attached to the `lung_experiment` object using the `update_metadata` function. The patient metadata must contain 2 columns, and each row corresponds to a `slide_id` in the `Mltplx Experiment`.

(1)`slide_id` contains the same set of IDs as the `MltplxExperiment`.
(2)`patient_id` links the `slide_id` to the patient identifier.

The metadata can be added to `lung_experiment` via

```
lung_experiment = update_metadata(lung_experiment, full_lung_metadata)
lung_experiment
##MltplxExperiment with 761 slides
##Intensities generated with pixel size 10 and
bandwidth 30
##Distance matrices generated with jsd
##Metadata has 14 columns
```

The generic function `as_tibble` facilitates downstream modeling and analysis by generating a tibble containing all of the columns that were already present in the metadata and a column for each pair of cell types. Users may easily include the pairwise distances as covariates in survival or regression modeling of patient outcomes, model them as outcomes in a mixed-effects model, or uncover natural groupings of the images through unsupervised clustering.

```
slide_level_tibble = as_tibble(lung_experiment)
```

Basic linear regression can be performed using the `lm_dist` function, which fits separate models for each pairwise distance to test for association with a variable from the patient metadata. In this example, "mhcII_status" is a column of the patient metadata and indicates whether a patient's tumor cells expressed a high or low amount of MHC class II. The user can specify variables to adjust for and aggregating functions to use. The default settings adjust for cell-type counts and aggregates images within patient via the median. The tested coefficient for each model can be displayed graphically using `plot_dist_regression_heatmap`, as shown in the following code sample. The resulting plot output is shown in Figure 6.

```
lung_lm = lm_dist(lung_experiment, group_factor = "mhcII_status")
plot_dist_regression_heatmap(lung_lm)
```

In this example data, low expression of MHC class II is associated with decreased infiltration of immune cells with CK+ tumor cells. This finding is in agreement with Johnson et al.[5]

As an alternative to aggregating distances within patient, distances may be modeled as outcomes in mixed-effect models with patient-specific random effects to account for within-patient correlations.

Generating and visualizing quantile-specific distance matrices

Patient outcomes or covariates may not be associated with pairwise cellular interactions on the scale of the entire image. Rather,

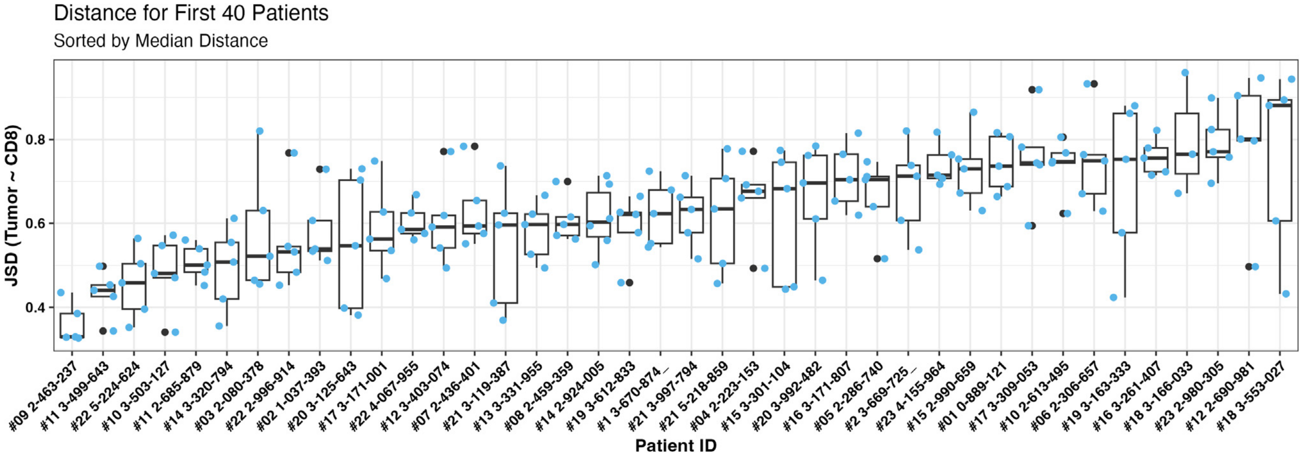


Figure 5. Distribution of biopsy distances between tumor cells and CD8 T cells within and between of first 40 patients in the dataset, sorted by median distance

cellular interaction at different regions of the image may be associated with outcomes. The distance matrices may be computed at various partitions of the image. We have implemented functionality to compute distance matrices in partitions of the image defined by user-defined quantiles of one cell-type intensity. The `update_qdist` function takes the following arguments.

(1)`dist_metric` determines the distance metric.
(2)`mask_type` determines by which cell-type intensity to partition.
(3)`q_probs` dataframe representing a range of quantiles by which to partition the intensity.

The following code partitions each image in `lung_experiment` into distinct areas based on quartiles of the intensity of CK+ cells.

```
# uncomment below to use multiple cores to calcu-
late quantile
# distance matrices
# plan(multisession(workers=2L))
lung_experiment = update_qdist(lung_experiment,
dist_metric = jsd,
mask_type = "CK",
q_probs = tibble(
from = c(0, 25, 50, 75),
to = c(25, 50, 75, 100)
))
```

The quantile-specific distance matrices can be visualized using the `plot_qdist` function, as shown below. The resulting plot output is shown in Figure 7.

```
plot_qdist_matrix(lung_experiment[[1]])
```

Across the different quartiles, the patterns of spatial interactions between different cell types can vary considerably. In this particular image, as the intensity of “CK”-phenotyped cells increases, the distance between the distributions of many of the various immune-phenotyped cells also increases. Interestingly, though, the distance between CD14-phenotyped cells and CK-phenotyped cells decreases. Varying levels of immune cell co-localization across regions of the tumor may indicate increasing concentrations of cytokines or other signaling molecules in the core tumor regions.[22] Quantile-specific distance matrices generated by `DIMPLE` allow one to probe and explore the natural heterogeneity that arises from such multifactorial and complex interactions in the TME.

Using the DIMPLE Shiny app

The R Shiny app is an additional tool to help researchers generate publication-ready plots using the outputted distance matrices and intensities from the R package (Figure 8). Plots included in this descriptor may be generated with this app. It produces color-blind-friendly visualizations of individual intensities and distance matrices, aggregated data, and model outputs.

The Shiny app accepts a `MltplxExperiment` saved as an `.RDS` data file. The computation of intensities and distance matrices is meant to be completed before loading the `MltplxExperiment` in the R Shiny app. This minimizes the computations being done within the Shiny app, allowing the user to iterate between visualizations relatively quickly. One can also explore the lung cancer data referenced in this tutorial with the click of a button.

The Shiny app is organized into the following 3 pages.

(1)Visualize multiplex object: visually explore individual `MltplxObjects` by selecting a slide ID.
(2)Analyze multiplex experiment: basic inference and visualization of model outputs in combination with patient-level metadata.
(3)Quantile-specific distances: explore, visualize, and make inferences on quantile-specific distance matrices.

Each plot may be downloaded as a PDF file. The app is freely available at https://bayesrx.shinyapps.io/dimple_shiny/.

Simulation of MI data

Finally, we have developed several functions to simulate MI data from intensity surfaces. These functions generate simulated MI data that may be used to test out parameters of the intensity

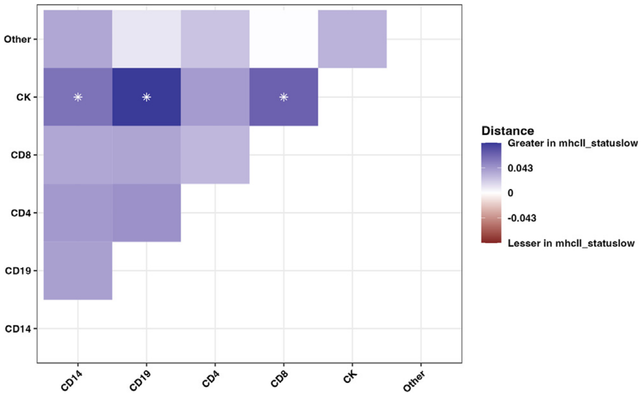


Figure 6. Estimated effect of MHC class II status low on pairwise cellular distances

Each square displays the regression coefficient in a linear regression model where the dependent variable is the median pairwise distance for each patient and the independent variable is a binary indicator of MHC class II status low (MHC class II status high is reference) for each patient. Each regression model is adjusted for overall prevalence of the cell types. Blue indicates a positive effect on distance, and red indicates a negative effect on distance. Stars indicate false discovery rate (FDR)-adjusted p values less than 0.05.

estimation or various distance metrics. We provide a tutorial of those functions in section S2 of the supplemental information.

DISCUSSION

We introduce DIMPLE, a statistical software package designed to probe the relationships between cell types obtained from MI experiments. DIMPLE converts spatial and phenotypic information captured in MI assays to continuously varying non-parametric KDEs of the point process intensity function with a user-defined smoothing bandwidth. Then, a user-defined distance metric is applied to the pairwise combinations of the KDEs for each cell type. The resulting distance matrix can be further explored and, when combined with patient-level metadata, can be used to identify potential biomarkers within the TME. All of this functionality is built around a single simple, flexible, and easily extendable data structure: the MltplxExperiment object.

The method proposed here takes cell types as input and treats them as *a priori* "known." We note that any conclusions drawn from our pipeline rely on the accuracy of the mechanism used for typing the cells. In principle, raw pixel-level protein markers may be used directly in place of cell-type intensity maps. However, in practice, it depends on the resolution of the pixel-level data. For example, the Vectra Polaris platform provides outputs of aggregated measures of pixel intensities at the nucleus, cytoplasm, and membrane of each detected cell. Furthermore, an analysis directly on the pixel-level intensity maps would answer a different scientific question than those we have focused on in our current work. It would investigate the relationships between different types of protein markers expressed in the biopsy. Often, this is not the primary analysis of interest. We are specifically interested in the relationships of distinct cell types (which may be defined by expression of combinations of protein markers).

While the MltplxExperiment object has some similarities with the SpatialExperiment class,[23] we would like to clarify how and why they are different. MltplxExperiment is designed specifically for MI technologies in which discrete cell types have already been annotated. On the other hand, SpatialExperiment is designed for a much broader class of spatial technologies, in which a potentially large number of markers or genes are imaged or sequenced (typical of spatial transcriptomics experiments), and this information is stored in a large "counts" matrix, analogous to that in single-cell RNA sequencing experiments. Since MltplxExperiment deals typically with a much smaller data size (at minimum, just the spatial coordinates and discrete type of each cell), rather than the full roster of markers/genes observed at each spot, we deemed that the overhead of all of the extra parts in SpatialExperiment was extraneous for our purposes. Furthermore, the SpatialExperiment object is primarily a data structure object, while the MltplxExperiment data structure integrates smoothly with the various analysis options available in DIMPLE. A MltplxExperiment object holds not only the raw data (spatial coordinates, cell types, and image metadata) but also the results of analyses such as smoothed intensity plots and distance matrices.

We acknowledge a couple important limitations of our proposed method. First, the smoothing bandwidth bw must be chosen *a priori* and can have a non-trivial influence on downstream results. We urge users to choose a biologically plausible "sphere of influence" for the smoothing bandwidth. Second, this method cannot quantify the tendency of cells to cluster spatially with their own type. Due to how we have defined intracellular distance, the distance resulting from a comparison of one cell type to itself would always be zero.

The expression level of functional markers may be an important confounder in the association between pairwise cellular distance and patient outcomes. The quantification of this type of interaction is currently very limited. If the functional marker can be represented by a binary variable for each cell (expressed/not expressed), it can be easily incorporated into the cell type and be readily used in the pipeline. For example, users could divide tumor cells into PD-L1+ tumor cells and PD-L1− tumor cells. In the future, we hope to develop additional methods to incorporate the functional markers and add support for other spatially varying covariates. These improvements will enhance the capabilities of DIMPLE and enable researchers to gain further insight into the relationships between different cell types in MI experiments.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Maria Masotti (mmasotti@umich.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The DIMPLE R package is open source and freely available on Github (https://github.com/nateosher/dimple). The Shiny app is available at https://bayesrx.shinyapps.io/dimple_shiny/. All original code has been deposited at Zenodo under DOI: https://doi.org/10.5281/zenodo.8327331. Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

The data used as illustration of our package were sourced from the VectraPolarisData package. This package can be installed from Bioconductor (https://bioconductor.org/packages/release/data/experiment/html/vectrapolarisdata.html).

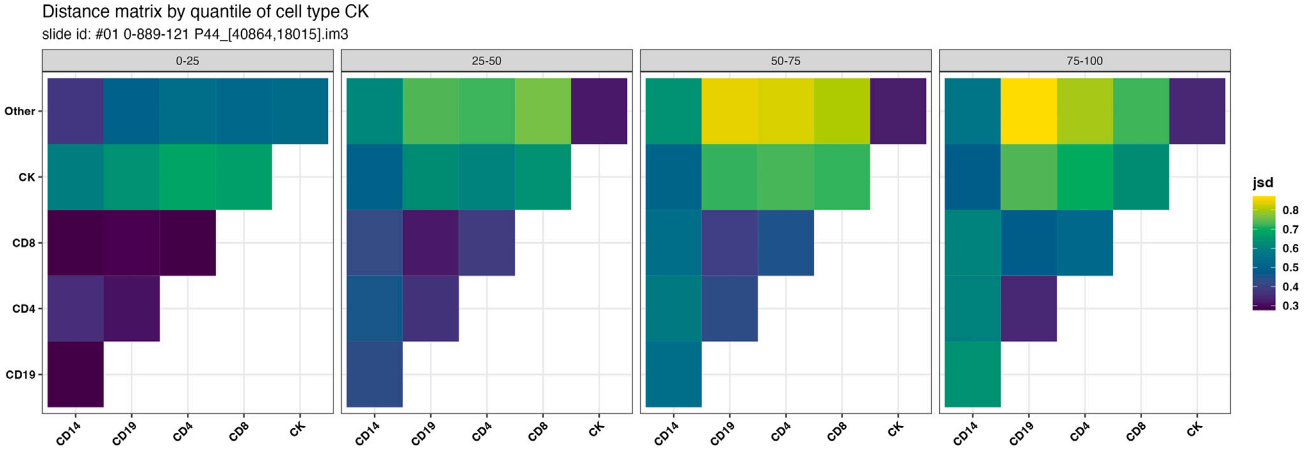


Figure 7. Quantile-specific distance matrices generated by the code above for the first point pattern faceted by the intensity of CK+ cells
The color represents JSD, with brighter color indicating greater distance.

SUPPLEMENTAL INFORMATION

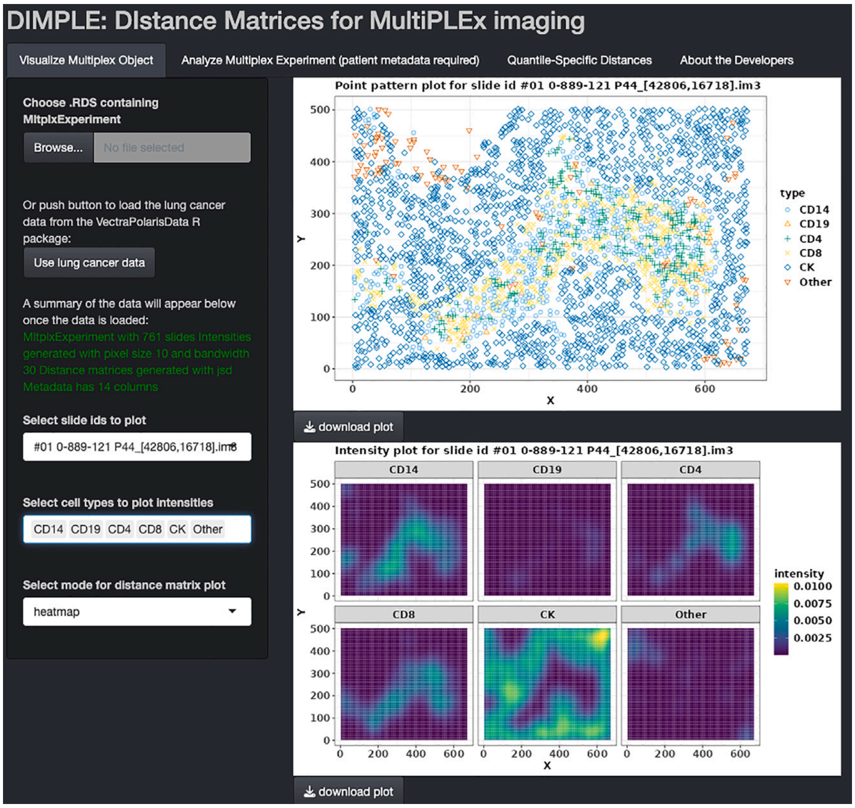
Supplemental information can be found online at https://doi.org/10.1016/j.patter.2023.100879.

ACKNOWLEDGMENTS

We thank Dr. Julia Wrobel for making available the data that is used in this manuscript. This work was supported by NCI grant R37 CA214955-01A1 (M.M., J.E., V.B., and A.R.), NIH grant P30 CA046592 (A.R., M.M., and V.B.), NIH grant R01 CA244845-01A1 (V.B.), and NIH grant T32 CA140044 (J.E.).

AUTHOR CONTRIBUTIONS

M.M., N.O., and J.E. developed the presented software and drafted the manuscript. V.B. provided input on writing, reviewing, and editing the manuscript. All authors contributed to conception of the project and approved the final version of the manuscript.



DIMPLE: DIstance Matrices for MultiPLEx imaging

Visualize Multiplex Object
Analyze Multiplex Experiment (patient metadata required)
Quantile-Specific Distances
About the Developers

Choose .RDS containing MltplxExperiment

Browse...
No file selected

Or push button to load the lung cancer data from the VectraPolarisData R package:

A summary of the data will appear below once the data is loaded:
MltplxExperiment with 761 slides Intensities generated with pixel size 10 and bandwidth 30 Distance matrices generated with jsd Metadata has 14 columns

Select slide ids to plot: #01 0-889-121 P44_[42806,16718].im3

Select cell types to plot intensities: CD14 CD19 CD4 CD8 CK Other

Select mode for distance matrix plot: heatmap

Figure 8. The R Shiny app allows users to explore and visualize MI data
Users can simply upload a `MltplxExperiment` object or explore the lung data referenced in this descriptor with the click of a button.

DECLARATION OF INTERESTS

A.R. serves as consulting member for Voxel Analytics, LLC; Telperian, Inc.; TCS, Ltd.; and Tempus Labs.

INCLUSION AND DIVERSITY

We support inclusive, diverse, and equitable conduct of research.

Received: July 7, 2023
Revised: September 11, 2023
Accepted: October 24, 2023
Published: November 20, 2023

REFERENCES

1. Anderson, N.M., and Simon, M.C. (2020). The tumor microenvironment. Curr. Biol. *30*, R921–R925. https://doi.org/10.1016/j.cub.2020.06.081.
2. Tan, W.C.C., Nerurkar, S.N., Cai, H.Y., Ng, H.H.M., Wu, D., Wee, Y.T.F., Lim, J.C.T., Yeong, J., and Lim, T.K.H. (2020). Overview of multiplex immunohistochemistry/immunofluorescence techniques in the era of cancer immunotherapy. Cancer Commun. *40*, 135–153. https://doi.org/10.1002/cac2.12023.
3. Lewis, S.M., Asselin-Labat, M.-L., Nguyen, Q., Berthelet, J., Tan, X., Wimmer, V.C., Merino, D., Rogers, K.L., and Naik, S.H. (2021). Spatial omics and multiplexed imaging to explore cancer biology. Nat. Methods *18*, 997–1012. https://doi.org/10.1038/s41592-021-01203-6.
4. van Dam, S., Baars, M.J.D., and Vercoulen, Y. (2022). Multiplex Tissue Imaging: Spatial Revelations in the Tumor Microenvironment. Cancers *14*, 3170. https://doi.org/10.3390/cancers14133170.
5. Johnson, A.M., Boland, J.M., Wrobel, J., Klezcko, E.K., Weiser-Evans, M., Hopp, K., Heasley, L., Clambey, E.T., Jordan, K., Nemenoff, R.A., and Schenk, E.L. (2021). Cancer Cell-Specific Major Histocompatibility Complex II Expression as a Determinant of the Immune Infiltrate Organization and Function in the NSCLC Tumor Microenvironment. J. Thorac. Oncol. *16*, 1694–1704. https://doi.org/10.1016/j.jtho.2021.05.004.
6. Steinhart, B., Jordan, K.R., Bapat, J., Post, M.D., Brubaker, L.W., Bitler, B.G., and Wrobel, J. (2021). The Spatial Context of Tumor-Infiltrating Immune Cells Associates with Improved Ovarian Cancer Survival. Mol. Cancer Res. *19*, 1973–1979. https://doi.org/10.1158/1541-7786.MCR-21-0411.
7. Schürch, C.M., Bhate, S.S., Barlow, G.L., Phillips, D.J., Noti, L., Zlobec, I., Chu, P., Black, S., Demeter, J., McIlwain, D.R., et al. (2020). Coordinated Cellular Neighborhoods Orchestrate Antitumoral Immunity at the Colorectal Cancer Invasive Front. Cell *182*, 1341–1359.e19. https://doi.org/10.1016/j.cell.2020.07.005.
8. Baddeley, A., Turner, R., and Rubak, E. (2023). Spatstat: Spatial Point Pattern Analysis, Model-Fitting, Simulation (Tests).
9. Creed, J.H., Wilson, C.M., Soupir, A.C., Colin-Leitzinger, C.M., Kimmel, G.J., Ospina, O.E., Chakiryan, N.H., Markowitz, J., Peres, L.C., Coghill, A., and Fridley, B.L. (2021). spatialTIME and iTIME: R package and Shiny application for visualization and analysis of immunofluorescence data. Bioinformatics *37*, 4584–4586. https://doi.org/10.1093/bioinformatics/btab757.
10. Canete, N.P., Iyengar, S.S., Ormerod, J.T., Baharlou, H., Harman, A.N., and Patrick, E. (2022). spicyR: spatial analysis of *in situ* cytometry data in R. Bioinformatics *38*, 3099–3105. https://doi.org/10.1093/bioinformatics/btac268.
11. Ripley, B.D. (1976). The Second-Order Analysis of Stationary Point Processes. J. Appl. Probab. *13*, 255–266. https://doi.org/10.2307/3212829.
12. Besag, J. (1977). Comments on Ripley's paper. J. Roy. Stat. Soc. *39*, 193–195.
13. Marcon, E., Puech, F., and Traissac, S. (2012). Characterizing the Relative Spatial Structure of Point Patterns. Int. J. Ecol. *2012*. e619281–11. https://doi.org/10.1155/2012/619281.
14. Baddeley, A., Rubak, E., and Turner, R. (2016). Spatial Point Patterns: Methodology and Applications with R (Champan & Hall/CRC Interdisciplinary Statistics Series, CRC Press, Taylor & Francis Group).
15. Vu, T., Wrobel, J., Bitler, B.G., Schenk, E.L., Jordan, K.R., and Ghosh, D. (2022). SPF: A spatial and functional data analytic approach to cell imaging data. PLoS Comput. Biol. *18*, e1009486. https://doi.org/10.1371/journal.pcbi.1009486.
16. Wilson, C.M., Ospina, O.E., Townsend, M.K., Nguyen, J., Moran Segura, C., Schildkraut, J.M., Tworoger, S.S., Peres, L.C., and Fridley, B.L. (2021). Challenges and Opportunities in the Statistical Analysis of Multiplex Immunofluorescence Data. Cancers *13*, 3031. https://doi.org/10.3390/cancers13123031.
17. Osher, N., Eliason, J., and Masotti, M. (2023). nateosher/DIMPLE:. https://doi.org/10.5281/zenodo.8327332 Version 1.0.0. .
18. Wrobel, J., and Ghosh, T. (2022). Vectrapolarisdata: Vectra Polaris and Vectra 3 Multiplex Single-Cell Imaging Data, R Package Version 1.0.0.
19. Jones, M.C. (1993). Simple boundary correction for kernel density estimation. Stat. Comput. *3*, 135–146. https://doi.org/10.1007/BF00147776.
20. Diggle, P. (2010). Nonparametric Methods. In Handbook of Spatial Statistics, A.E. Gelfand, P. Guttorp, and M. Fuentes, eds. (CRC Press).
21. Seal, S., Vu, T., Ghosh, T., Wrobel, J., and Ghosh, D. (2022). DenVar: density-based variation analysis of multiplex imaging data. Bioinform. Adv. *2*, vbac039. https://doi.org/10.1093/bioadv/vbac039.
22. Li, L., Yu, R., Cai, T., Chen, Z., Lan, M., Zou, T., Wang, B., Wang, Q., Zhao, Y., and Cai, Y. (2020). Effects of immune cells and cytokines on inflammation and immunosuppression in the tumor microenvironment. Int. Immunopharm. *88*, 106939. https://doi.org/10.1016/j.intimp.2020.106939.
23. Righelli, D., Weber, L.M., Crowell, H.L., Pardo, B., Collado-Torres, L., Ghazanfar, S., Lun, A.T.L., Hicks, S.C., and Risso, D. (2022). SpatialExperiment: infrastructure for spatially-resolved transcriptomics data in R using Bioconductor. Bioinformatics *38*, 3128–3131. https://doi.org/10.1093/bioinformatics/btac299.