Package 'tlsR'

April 9, 2026

Type Package

Title Detection and Spatial Analysis of Tertiary Lymphoid Structures

Version 0.2.0

Date 2026-04-02

Description Fast, reproducible detection and quantitative analysis of tertiary lymphoid structures (TLS) in multiplexed tissue imaging. Implements Independent Component Analysis Trace (ICAT) index, local Ripley's K scanning, automated K Nearest Neighbor (KNN)-based TLS detection, and T-cell clusters identification as described in Amiryousefi et al. (2025) <doi:10.1101/2025.09.21.677465>.

License MIT + file LICENSE

URL https://github.com/labsyspharm/tlsR

Depends R (>= 4.0.0)

Imports dbscan (>= 1.1-10), fastICA (>= 1.2-3), FNN (>= 1.1.3), spatstat.explore (>= 3.0-0), spatstat.geom (>= 3.0-0), ggplot2 (>= 3.4.0), rlang (>= 1.0.0), methods

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Ali Amiryousefi [aut, cre] (ORCID: <https://orcid.org/0000-0002-6317-3860>), Jeremiah Wala [aut] (ORCID: <https://orcid.org/0000-0001-6591-1620>), Peter Sorger [ctb] (ORCID: <https://orcid.org/0000-0002-3364-1838>)

Maintainer Ali Amiryousefi <ali_amiryousefi@hms.harvard.edu>

Repository CRAN

Date/Publication 2026-04-09 17:00:02 UTC

Contents

`tlsR-package` *tlsR: Detection and Spatial Analysis of Tertiary Lymphoid Structures*

Description

Fast, reproducible detection and quantitative analysis of tertiary lymphoid structures (TLS) in multiplexed tissue imaging data.

Typical workflow

1. Load or prepare a named list of data frames (`ldata`), one per tissue sample. Each data frame must contain columns `x`, `y` (spatial coordinates in microns), and `phenotype` (character: `"B cell"` / `"T cell"` / other).
2. Run `detect_TLS` to label B+T co-localised regions.
3. (Optional) Run `scan_clustering` to identify windows of significant immune clustering via local Ripley's L.
4. Run `calc_icat` to score the internal linearity/organisation of each detected TLS.
5. Run `detect_tic` to identify T-cell clusters outside TLS.
6. Use `summarize_TLS` to obtain a tidy summary table.
7. Use `plot_TLS` to produce publication-ready spatial plots.

Author(s)

Maintainer: Ali Amiryousefi <ali_amiryousefi@hms.harvard.edu> (ORCID)

Authors:

- Jeremiah Wala <jeremiah_wala@dfci.harvard.edu> (ORCID)

Other contributors:

- Peter Sorger (ORCID) [contributor]

References

Amiryousefi et al. (2025) doi:10.1101/2025.09.21.677465

See Also

Useful links:

- https://github.com/labsyspharm/tlsR

`calc_icat` *Calculate ICAT (Immune Cell Arrangement Trace) Index*

Description

Quantifies the linear / organised spatial arrangement of cells within a detected TLS by applying FastICA to the (x, y) coordinates of TLS cells and returning the trace of the estimated mixing matrix. Higher values indicate more structured (linear) cell organisation.

If the requested TLS contains fewer than 3 cells, or FastICA does not converge, the function returns `NA_real_` with an informative message rather than throwing an error.

Usage

```
calc_icat(patientID, tlsID, ldata = NULL)
```

Arguments

`patientID`	Character. Sample name in `ldata`.
`tlsID`	Numeric or integer. TLS identifier (value of `tls_id_knn` for the TLS of interest).
`ldata`	Named list of data frames, or `NULL` to use the global `ldata` object (deprecated; pass explicitly).

Value

A single numeric value (the ICAT index), or `NA_real_` if computation is not possible (fewer than 3 cells, or FastICA did not converge).

Examples

```
data(toy_ldata)
ldata <- detect_TLS("ToySample", k = 30, ldata = toy_ldata)
if (max(ldata[["ToySample"]]$tls_id_knn, na.rm = TRUE) > 0) {
  icat <- calc_icat("ToySample", tlsID = 1, ldata = ldata)
  icat
}
```

detect_tic *Detect Tumor-Infiltrating T-cell Clusters (TIC)*

Description

Applies HDBSCAN to T cells that lie outside of previously detected TLS regions to identify spatially compact T-cell clusters (TIC). Phenotype labels "T cell" and "T cells" are both accepted.

Usage

```
detect_tic(sample, min_pts = 10L, min_cluster_size = 10L, ldata = NULL)
```

Arguments

sample	Character. Sample name in ldata.
min_pts	Integer. HDBSCAN minPts parameter: minimum cluster size (default 10). Smaller values detect more, smaller clusters.
min_cluster_size	Integer. Minimum number of T cells for a HDBSCAN cluster to be retained; smaller clusters are merged back into noise (label 0). Default 10.
ldata	Named list of data frames, or NULL to use the global ldata object (deprecated; pass explicitly).

Value

The input ldata list with the sample data frame augmented by one new column:

tcell_cluster_hdbscan Integer. 0 = noise / not a T-cell cluster; positive integer = TIC cluster ID. Non-T-cell rows receive NA.

Examples

```
data(toy_ldata)
ldata <- detect_TLS("ToySample", k = 30, ldata = toy_ldata)
ldata <- detect_tic("ToySample", ldata = ldata)
table(ldata[["ToySample"]]$tcell_cluster_hdbscan, useNA = "ifany")
```

detect_TLS *Detect Tertiary Lymphoid Structures using a KNN-density approach*

Description

Identifies TLS candidates by finding regions of high local B-cell density that also contain a sufficient number of nearby T cells (B+T co-localisation). Phenotype labels "B cell" and "B cells" (and their T-cell equivalents) are both accepted.

Usage

```
detect_TLS(
  LSP,
  k = 30L,
  bcell_density_threshold = 10,
  min_B_cells = 50L,
  min_T_cells_nearby = 20L,
  max_distance_T = 50,
  ldata = NULL
)
```

Arguments

LSP	Character. Sample name in ldata.
k	Integer. Number of nearest neighbours used for density estimation (default 30, calibrated for 0.325 um/px imaging).
bcell_density_threshold	Numeric. Minimum average 1/k-distance (in microns) for a B cell to be considered locally dense (default 15).
min_B_cells	Integer. Minimum B cells per candidate TLS cluster (default 50).
min_T_cells_nearby	Integer. Minimum T cells within max_distance_T microns of the candidate cluster centre (default 30).
max_distance_T	Numeric. Search radius (microns) for T-cell proximity check (default 50).
ldata	Named list of data frames, or NULL to use the global ldata object (deprecated; pass explicitly).

Value

The input ldata list, with the data frame for LSP augmented by three new columns:

tls_id_knn Integer. 0 = non-TLS cell; positive integer = TLS cluster ID.

tls_center_x Numeric. X coordinate of the TLS centre for TLS cells; NA otherwise.

tls_center_y Numeric. Y coordinate of the TLS centre for TLS cells; NA otherwise.

Examples

```
data(toy_ldata)
ldata <- detect_TLS("ToySample", k = 30, ldata = toy_ldata)
table(ldata[["ToySample"]]$tls_id_knn)
plot(ldata[["ToySample"]]$x, ldata[["ToySample"]]$y,
     col = ifelse(ldata[["ToySample"]]$tls_id_knn > 0, "red", "gray"),
     pch = 19, cex = 0.5, main = "Detected TLS in toy data")
```

---

plot_TLS *Plot Spatial Map of TLS and T-cell Clusters*

Description

Produces a ggplot2 scatter plot of cell positions, coloured by TLS membership, T-cell cluster membership, and background phenotype.

Usage

```
plot_TLS(
  sample,
  ldata = NULL,
  show_tic = TRUE,
  point_size = 0.4,
  alpha = 0.6,
  tls_palette = c("#0072B2", "#009E73", "#CC79A7", "#D55E00", "#56B4E9", "#F0E442"),
  tic_colour = "#E69F00",
  bg_colour = "grey80"
)
```

Arguments

sample	Character. Sample name in ldata.
ldata	Named list of data frames, or NULL to use the global ldata object (deprecated; pass explicitly).
show_tic	Logical. Colour T-cell clusters in a distinct colour? Default TRUE.
point_size	Numeric. Point size (default 0.4).
alpha	Numeric. Point transparency (default 0.6).
tls_palette	Character vector of colours for TLS IDs. Recycled if there are more TLS than colours.
tic_colour	Character. Colour for T-cell cluster cells (default "#E69F00").
bg_colour	Character. Colour for background cells (default "grey80").

Value

A ggplot object (invisibly). Build and print the returned object to display the plot.

Examples

```
data(toy_ldata)
ldata <- detect_TLS("ToySample", k = 30, ldata = toy_ldata)

  p <- plot_TLS("ToySample", ldata = ldata)
  print(p)
```

`scan_clustering` *Scan Tissue for Local Immune Cell Clustering (Ripley's L)*

Description

Applies a sliding-window centred L-function analysis across the tissue to identify spatially localised windows with statistically significant immune cell clustering. Each window is tested against a Monte Carlo envelope of complete spatial randomness (CSR).

Usage

```
scan_clustering(
  ws = 500,
  sample,
  phenotype = c("T cells", "B cells", "Both"),
  plot = FALSE,
  creep = 1L,
  nsim = 39L,
  min_cells = 10L,
  ldata = NULL
)
```

Arguments

`ws`	Numeric. Window side length in microns (default `500`).
`sample`	Character. Sample name in `ldata`.
`phenotype`	One of `"T cells"`, `"B cells"`, or `"Both"`.
`plot`	Logical. Show a diagnostic plot for each significant window? (default `FALSE`).
`creep`	Integer. Grid density factor; higher values give a finer scanning grid (default `1`).
`nsim`	Integer. Number of Monte Carlo simulations for envelope estimation (default `39`, giving a pointwise significance level of 0.05 under CSR).
`min_cells`	Integer. Minimum cell count required in a window before it is analysed (default `10`).
`ldata`	Named list of data frames, or `NULL` to use the global `ldata` object (deprecated; pass explicitly).

Value

A named list of results for significant windows. Each element is itself a list with:

`Lest` The `spatstat` `Lest` object.

`envelope` The Monte Carlo envelope object.

`window_center` Numeric vector `c(cx, cy)` of window centre coordinates.

`n_cells` Integer. Cell count in this window.

Returns an empty list (invisibly) when no significant windows are found.

Examples

```
data(toy_ldata)

  models <- scan_clustering(ws = 500, sample = "ToySample",
                            phenotype = "B cells", plot = FALSE,
                            nsim = 19, ldata = toy_ldata)
  length(models)
```

`summarize_TLS` *Summarize Detected TLS Across Samples*

Description

Produces a tidy `data.frame` with one row per sample summarising the number of detected TLS, their sizes, and (optionally) ICAT scores.

Usage

```
summarize_TLS(ldata, calc_icat_scores = FALSE)
```

Arguments

`ldata`	Named list of data frames as returned by `detect_TLS` (and optionally `detect_tic`).
`calc_icat_scores`	Logical. Should ICAT scores be computed for each TLS and appended as a list-column? Default `FALSE`.

Value

A `data.frame` with columns:

`sample` Sample name.

`n_TLS` Number of TLS detected.

`total_cells` Total cells in the sample.

`TLS_cells` Number of cells assigned to any TLS.

`TLS_fraction` Fraction of all cells that are TLS cells.

`mean_TLS_size` Mean cells per TLS (`NA` if n_TLS = 0).

`n_TIC` Number of T-cell clusters detected by `detect_tic` (`NA` if not yet run).

`icat_scores` List-column of ICAT scores per TLS (only when `calc_icat_scores = TRUE`).

Examples

```
data(toy_ldata)
ldata <- detect_TLS("ToySample", k = 30, ldata = toy_ldata)
summarize_TLS(ldata)
```

---

toy_ldata *Toy Multiplexed Imaging Data*

Description

A small synthetic dataset mimicking multiplexed tissue imaging data, used in package examples and tests. The list contains one sample named `"ToySample"`.

Usage

```
toy_ldata
```

Format

A named list with one element:

`ToySample` A `data.frame` with the following columns:

- `x` Numeric. X coordinate in microns.
- `y` Numeric. Y coordinate in microns.
- `phenotype` Character. Cell phenotype label. Values are `"B cell"`, `"T cell"`, and `"Other"`.

Source

Synthetically generated for package examples.

References

Amiryousefi et al. (2025) doi:10.1101/2025.09.21.677465

Examples

```
data(toy_ldata)
str(toy_ldata)
table(toy_ldata[["ToySample"]]$phenotype)
```

Index