Package: lrmap (via r-universe)

November 29, 2026

Type Package

Title Multi-Scale Spatial Co-Location of Ligands and Receptors Between Cell Types

Version 0.1.0

Description For every sender and receiver cell type and every ligand-receptor pair, measures how much ligand from the sender type surrounds receiver cells that express the receptor, at three spatial scales: touching cells (contact), short-range neighbours (paracrine) and the regional neighbourhood (niche). Significance comes from toroidal shifts of the sender layer against the receiver layer, which keep each layer's own spatial pattern. Transcripts leaked between touching cells can be estimated and removed (lrmap_leak()).

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.2)

Imports Matrix, RANN, stats, methods, utils

Suggests testthat (>= 3.0.0), edgeicg, lrmaprs

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://github.com/drieslab/lrmap

BugReports https://github.com/drieslab/lrmap/issues

Repository https://drieslab.r-universe.dev

Date/Publication 2026-11-25 22:25:26 UTC

RemoteUrl https://github.com/drieslab/lrmap

RemoteRef HEAD

RemoteSha 3621da9a3c4adb0edee5393f008d03a359ae71cf

Contents

lrmap 2
lrmap_compare 4
lrmap_leak 5
lrmap_local 7
lrmap_results 8
lrmap_simulate 9

Index **11**

`lrmap` *Multi-scale co-location of ligands and receptors between cell types*

Description

For each sender type A, receiver type B (A != B unless `same_type`) and ligand-receptor pair (L, R), the score at scale s is the mean over receiver cells i of type B of R_i * X_i, where X_i is the ligand carried by type-A cells around i: sum over neighbours j of type A of w_s(i, j) L_j, divided by the summed weight of all neighbours. Expression is on the composition scale (counts / library * 100). Scales: contact (k nearest cells, weight exp(-d / median d)), paracrine (Gaussian, `sigma_p`) and niche (Gaussian, `sigma_n`); paracrine and niche are evaluated on grids. The score is high when receptor-expressing B cells sit among ligand-expressing A cells.

Usage

```
lrmap(
  counts,
  coords,
  types,
  lr,
  lib = NULL,
  scales = c("contact", "paracrine", "niche"),
  k = 8,
  sigma_p = 25,
  sigma_n = 150,
  bin_p = 8,
  bin_n = 25,
  B = 200L,
  min_shift = 450,
  min_det = 0.05,
  min_cells = 30L,
  same_type = FALSE,
  leak = NULL,
  leak_pool = c("median", "q1", "none"),
  spill_edges = NULL,
  chunk = NULL,
  grid_mem = 4e+07,
```

```
  z_cand = 1.5,
  backend = c("auto", "rust", "r"),
  threads = 1L,
  seed = 1L,
  verbose = TRUE
)
```

Arguments

counts	raw counts: a genes x cells matrix (dense or sparse) with gene rownames, or a reader function(genes) returning a genes x cells matrix for the requested genes that exist (rows named; cells in the order of coords), e.g. reading from a disk-backed store. Only the genes of lr are read, once.
coords	cells x 2 coordinates (microns).
types	cell type per cell.
lr	ligand-receptor table: columns ligand and receptor, optionally ligand_subunits and receptor_subunits (subunit genes joined by "_"). A complex ligand is the minimum over its subunits, a complex receptor the mean.
lib	library size per cell (all genes). Default: column sums of counts when it is a matrix holding all genes; required with a reader.
scales	any of "contact", "paracrine", "niche", and "expression": the expression-only reference used by single-cell analyses (mean ligand in the sender type x mean receptor in the receiver type against random relabelling of cells, no spatial information), on the same tests and values; compare with lrmap_compare().
k	contact neighbours.
sigma_p, sigma_n	paracrine and niche kernel widths (microns).
bin_p, bin_n	grid bins for paracrine and niche (microns); keep at most a third of the kernel width.
B	number of toroidal shifts.
min_shift	minimum shift length (microns).
min_det	a pair is tested for (A, B) when the ligand is detected in at least min_det of A cells and the receptor (every subunit) in at least min_det of B cells.
min_cells	minimum cells of the sender and of the receiver type.
same_type	also test A = B.
leak	optional leak (spillover) correction: a data.frame with receiver, sender (type names), lambda and optionally mu (e.g. from lrmap_leak()), an edgeICG fit (its leak coefficients are read; edgeICG is not needed), or TRUE to estimate them with lrmap_leak() on all genes of counts (matrix input only). Transcripts leaked from touching cells of other types are removed from every gene before ligand and receptor values are formed: y - sum_j wn_ij lambda y_j - cmu_i a_t, with a_t the receiver type's first-stage baseline.
leak_pool	how leak coefficients of pairs with little contact are pooled when leak is an edgeICG fit or a plain table (see lrmap_leak(), argument pool); tables from lrmap_leak() are used as given.

spill_edges	optional touching edges for the leak correction (recv, send, w; cell indices in the order of coords), e.g. from cell outlines; default: the k nearest neighbours, weight exp(-d / median d).
chunk	ligands blurred at a time; default from the grid size (about grid_mem numbers per chunk).
grid_mem	numbers held per blurred chunk (sets the default chunk).
z_cand	the ligand-side part of the sampling variance on grid scales is computed for tests with z above this before it (adding variance only lowers z).
backend	"auto" uses the compiled kernels of the lrmaprs package when it is installed.
threads	threads for the compiled kernels (1: the calling thread only; 0: all cores).
seed	seed for the shifts.
verbose	progress messages.

Details

The null distribution comes from toroidal shifts of the whole sender layer (cells with their types and expression) against the receiver layer: each layer keeps its own spatial pattern, only their alignment is broken. z = (score - mean of shifted scores) / their standard deviation; p is one-sided (co-location above chance); q is Benjamini-Hochberg within each scale.

The score measures co-location: where and at which range communication through a pair is possible. Within one co-located pair of cell types, ligand-receptor pairs differ mainly through their expression and its spatial pattern; a call does not show that the pair is used. The type-level co-location (coloc) is reported separately so pair-level results can be read against it. Giotto's original cell-cell communication test (spatCellCellcom) is the contact-only predecessor of this analysis.

Three tests are reported for every spatial row. q (with z, fold): co-location of ligand-expressing senders and receptor-expressing receivers against shifted sender layers; this includes the co-location of the two cell types, so any pair expressed in two neighbouring types is called. q_type (with z_type, fold_type; the default of lrmap_results()): co-location beyond that of the two cell types, i.e. whether ligand-high senders sit nearer receptor-high receivers than other cells of the two types do, calibrated by the same shifts. q_shuffle: the same question against shuffling ligand and receptor values within their cell types (layout fixed, exact moments); it ignores smooth spatial trends of expression and is anti-conservative when they are present.

Value

an LRmap object: table (one row per tested sender, receiver, pair and scale), coloc (type-level co-location per scale), n_pairs, n_covered, settings, timing.

---

lrmap_compare *Expression-only calls versus spatial co-location*

Description

For a fit with the "expression" scale and at least one spatial scale: one row per tested sender, receiver and ligand-receptor pair, with whether it is called on expression alone and at which spatial scales it is co-located, and whether the two cell types co-locate at all (coloc).

Usage

```
lrmap_compare(
  x,
  test = c("type", "shuffle", "colocation"),
  q_thr = 0.05,
  min_fold = 1.25
)
```

Arguments

x	an LRmap object fitted with scales including "expression".
test, q_thr, min_fold	call rule for the spatial scales (as in lrmap_results()).

Value

data.frame: sender, receiver, ligand, receptor, expr_call, expr_z, spatial_scales (called scales joined by "+", "" if none), spatial_call, types_colocated, and class: "both", "expression only", "spatial only" or "neither".

---

lrmap_leak *Estimate leak (spillover) coefficients between touching cells*

Description

Fits the leak model used by lrmap (leak =) and by edgeICG: for each receiver type t and touching sender type k, lambda is the fraction of the sender's expression that appears in the receiver and mu a share of the receiver's own baseline, both estimated across genes from a first-stage regression of expression on the cell's neighbourhood composition (contact, paracrine and niche shares). Use many genes (all genes of the panel); the coefficients are per type pair, not per gene.

Usage

```
lrmap_leak(
  counts,
  coords,
  types,
  lib = NULL,
  genes = NULL,
  spill_edges = NULL,
  k = LEAK_DEF$k,
  sigma_p = LEAK_DEF$sigma_p,
  sigma_n = LEAK_DEF$sigma_n,
  bin = LEAK_DEF$bin,
  min_cells = LEAK_DEF$min_cells,
  min_det = LEAK_DEF$min_det,
  min_support = LEAK_DEF$min_support,
```

```
  pool = c("median", "q1", "none"),
  block = 300L,
  verbose = TRUE
)
```

Arguments

counts	raw counts: genes x cells matrix with gene rownames, or a reader function(genes) (as in lrmap()); with a reader, genes and lib are required.
coords	cells x 2 coordinates (microns).
types	cell type per cell.
lib	library size per cell (default: column sums of counts).
genes	genes to fit on (default: all rows of counts). Genes detected in fewer than min_det of the cells of every receiver type are dropped.
spill_edges	optional touching edges (recv, send, w; cell indices in the order of coords), e.g. from cell outlines; default: k nearest neighbours with weight exp(-d / median d).
k, sigma_p, sigma_n, bin	neighbourhood settings of the first-stage regression.
min_cells	receiver types with fewer cells are not fitted.
min_det	gene filter (see genes).
min_support	contact support (the receivers' contact shares of the sender type, summed: about the number of fully touching receivers) at which a pair's own estimate gets half the weight in pooling.
pool	how pairs with little contact are handled. "median" (default): every pair's estimate is shrunk toward the median of the well-supported pairs (support >= min_support) of its receiver type (all well-supported pairs if it has none), with weight support / (support + min_support); pairs without a usable estimate (little contact, outside the physical range of lambda between 0 and 1 and mu between -1 and 1, or a receiver type too small to fit) get that median. "q1": the same with the first quartile for lambda (less correction). "none": no pooling; pairs with support below min_support or without a usable estimate are not corrected.
block	genes read and fitted at a time.
verbose	progress messages.

Value

data.frame with receiver, sender, lambda, mu (used for the correction), support, status ("ok", "low support", "out of range" or "no estimate", prefixed "pooled: " when the pooled value was used), and the pair's own estimates lambda_raw, mu_raw; attributes n_genes, pool.

`lrmap_local` *Per-cell ligand exposure and local score for one ligand-receptor pair*

Description

The per-cell quantities behind one row of an `lrmap()` table, for maps: the ligand of each sender cell, the receptor of each receiver cell, the ligand each receiver is exposed to from sender cells at the chosen scale (X_i, with the kernels of `lrmap()`), and the local score R_i * X_i. Averaged over the receivers, the local score is the table's `score` (before leak removal).

Usage

```
lrmap_local(
  counts,
  coords,
  types,
  sender,
  receiver,
  ligand,
  receptor,
  scale = c("contact", "paracrine", "niche"),
  lib = NULL,
  k = 8,
  sigma_p = 25,
  sigma_n = 150,
  bin_p = 8,
  bin_n = 25
)
```

Arguments

`counts`	raw counts: a genes x cells matrix with gene rownames, or a reader `function(genes)` (see `lrmap()`); only the ligand and receptor genes are read.
`coords`	cells x 2 coordinates (microns).
`types`	cell type per cell.
`sender`, `receiver`	sender and receiver cell types.
`ligand`, `receptor`	genes of the ligand and the receptor; several genes for a complex (ligand: minimum over subunits, receptor: mean).
`scale`	"contact", "paracrine" or "niche".
`lib`	library size per cell (all genes); default: column sums of `counts` when it is a matrix.
`k`, `sigma_p`, `sigma_n`, `bin_p`, `bin_n`	kernel settings, as in `lrmap()`.

Value

data.frame, one row per cell (in the order of `coords`): `type`, `ligand` (sender cells, else NA), `receptor` (receiver cells, else NA), `exposure` (receiver cells), `local` (= receptor x exposure).

`lrmap_results` *Significant ligand-receptor co-location*

Description

Significant ligand-receptor co-location

Usage

```
lrmap_results(
  x,
  scale = c("contact", "paracrine", "niche", "expression"),
  test = c("type", "shuffle", "colocation"),
  q_thr = 0.05,
  min_fold = 1.25
)
```

Arguments

`x`	an `LRmap` object.
`scale`	scales to keep.
`test`	which null: "type" (pair co-location beyond the co-location of the two cell types, calibrated by the shifts; `q_type`, `fold_type`), "shuffle" (ligand and receptor values shuffled within their cell types, layout fixed; `q_shuffle`, `fold_type`) or "colocation" (shifted sender layer: co-location of ligand-expressing senders and receptor-expressing receivers, including the co-location of the two types; `q`, `fold`). The expression scale always uses `q` and `fold`.
`q_thr`	q-value threshold.
`min_fold`	minimum fold (see `test`).

Value

rows of `x$table` passing the thresholds, by q.

`lrmap_simulate` *Simulate a section with ligand-receptor co-location scenarios*

Description

Cells on a jittered hexagonal lattice. Type A forms round clusters; the layout places type B:

- `touch`: a ring of B directly around each A cluster (A and B touch);
- `gap`: a ring of C (`gap` microns) around each A cluster, then a 50 micron ring of B;
- `far`: A clusters in the left fifth, B clusters in the right fifth;
- `indep`: A, B and C clusters at independent random positions. Other cells are D. Gene `L` is expressed in A and gene `R` in B. With `leak_R`, `R` is expressed in A instead (not at all in B), and `leak` > 0 moves transcripts from touching cells of other types into each cell, so B cells next to A pick up `R`. Genes `G1..` have random type profiles and smooth spatial modulation; `lr` pairs them at random as decoys.

Usage

```
lrmap_simulate(
  layout = c("touch", "gap", "far", "indep"),
  size = 1500,
  spacing = 15,
  n_clusters = 20,
  r_a = 40,
  mu_l = 4,
  mu_r = 4,
  leak = 0,
  leak_R = FALSE,
  n_genes = 30,
  n_decoy = 40,
  gap = 35,
  lib_mean = 300,
  induce = 0,
  seed = 1
)
```

Arguments

`layout`	one of "touch", "gap", "far", "indep".
`size`	side of the square section (microns).
`spacing`	lattice spacing (microns).
`n_clusters`	A clusters.
`r_a`	A cluster radius (microns).
`mu_l`, `mu_r`	mean counts of L in A and R in B (at the mean library).

leak	fraction of each touching neighbour's expression leaked into a cell (other types only).
leak_R	express R in A instead of B (leak scenario).
n_genes	decoy genes.
n_decoy	decoy pairs.
gap	width of the C ring in the gap layout (microns).
lib_mean	mean library size.
induce	interface induction: L in A cells and R in B cells scale with (1 + induce x the cell's contact share of the other type), renormalised to the same type mean (0: expression is type-level only).
seed	random seed.

Value

list(counts (genes x cells), coords, types, lib, lr, leak (table of the true coefficients)).

Index

lrmap, 2
lrmap(), *6, 7*
lrmap_compare, 4
lrmap_compare(), *3*
lrmap_leak, 5
lrmap_leak(), *3*
lrmap_local, 7
lrmap_results, 8
lrmap_results(), *4, 5*
lrmap_simulate, 9