Package: cellchatr (via r-universe)

June 4, 2026

Title Fast CellChat Database Querying Backed by Rust

Version 0.99.0

Description Accelerates three bottleneck steps in the CellChat workflow using Rust via extendr. Provides parallel Wilcoxon rank-sum testing with tie correction, HashSet-based gene filtering, and ligand-receptor pair matching as drop-in replacements for CellChat functions. Achieves up to 183x speedup on Wilcoxon testing using Rayon parallelism.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/rextendr/version 0.4.2

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0, xz

URL https://github.com/mayankgandhi13/cellchatr

BugReports https://github.com/mayankgandhi13/cellchatr/issues

biocViews SingleCell, GeneExpression, Transcriptomics, StatisticalMethod, Software

Depends R (>= 4.4.0)

Imports methods

Suggests knitr, rmarkdown, testthat (>= 3.0.0), CellChat, microbenchmark

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs xz-utils libclang-dev

Repository https://mayankgandhi13.r-universe.dev

Date/Publication 2026-05-05 03:18:28 UTC

RemoteUrl https://github.com/mayankgandhi13/cellchatr

RemoteRef main

RemoteSha 842b32ee5a3766255991d2ec4443ea2577e242dd

RemoteSubdir .

Contents

`hello_world` *Return string* `"Hello world!"` *to R.*

Description

Return string `"Hello world!"` to R.

Usage

```
hello_world()
```

Value

A character string "Hello world!"

Examples

```
hello_world()
```

`rust_match_lr_pairs` *Match overexpressed genes against CellChatDB ligand-receptor pairs.*

Description

Match overexpressed genes against CellChatDB ligand-receptor pairs.

Usage

```
rust_match_lr_pairs(overexpressed, lr_ligands, lr_receptors, lr_names)
```

Arguments

overexpressed	Character vector of overexpressed gene names.
lr_ligands	Character vector of ligands from CellChatDB.
lr_receptors	Character vector of receptors from CellChatDB.
lr_names	Character vector of interaction names from CellChatDB.

Value

Character vector of matched interaction names.

Complexity

- Time: O(m + r) where m = overexpressed genes, r = LR pairs
- Space: O(m) for HashSet
- R equivalent (%%in%% twice): O(m x r)

Examples

```
rust_match_lr_pairs(
  c("TGFB1", "VEGFA"),
  c("TGFB1", "MYC", "VEGFA"),
  c("TGFBR1", "MYCBP", "FLT1"),
  c("TGFB1_TGFBR1", "MYC_MYCBP", "VEGFA_FLT1")
)
```

---

rust_subset_genes *Filter expression matrix genes to only those present in CellChatDB.*

Description

Filter expression matrix genes to only those present in CellChatDB.

Usage

```
rust_subset_genes(expr_genes, db_genes)
```

Arguments

expr_genes	Character vector of gene names from expression matrix.
db_genes	Character vector of gene names from CellChatDB.

Value

Character vector of genes present in both inputs.

Complexity

- Time: O(n + m) where n = db_genes, m = expr_genes
- Space: O(n) for HashSet
- R equivalent (%%in%%): O(n x m)

Examples

```
rust_subset_genes(
  c("TGFB1", "VEGFA", "GAPDH"),
  c("TGFB1", "VEGFA", "MYC")
)
```

---

rust_wilcoxon_filter *Identify over-expressed genes per cell type using Wilcoxon rank-sum test.*

Description

Identify over-expressed genes per cell type using Wilcoxon rank-sum test.

Usage

```
rust_wilcoxon_filter(counts, labels, gene_names, pval_threshold)
```

Arguments

counts	NumericMatrix of gene expression (genes x cells).
labels	Character vector of cell type labels, one per cell.
gene_names	Character vector of gene names.
pval_threshold	P-value cutoff for significance.

Value

Character vector of over-expressed gene names.

Complexity

- Time: O((g/p) x k x c log c) where g = genes, p = CPU cores, k = cell types, c = cells
- Space: O(c) per gene
- R equivalent: O(g x k x c log c) — serial, no parallelism

Examples

```
counts <- matrix(c(rep(10, 50), rep(0, 50), rep(1, 100)),
  nrow = 2, byrow = TRUE)
labels <- c(rep("A", 50), rep("B", 50))
rust_wilcoxon_filter(counts, labels, c("TGFB1", "GAPDH"), 0.05)
```

Index