Package ‘bgns’

September 27, 2026

Type Package

Title Biweight Graph and Network Statistics

Version 0.4.3

Description Provides memory-efficient biweight midcorrelation and exact bicor-based k-nearest-neighbor graph construction for dense and sparse numeric matrices. Dense, sparse, and mixed-input paths avoid materializing full dense similarity matrices for tidy and k-nearest-neighbor workflows where possible. The implementation supports pairwise finite-overlap handling and robust correlation-based graph construction for biological expression matrices and other high-dimensional numeric data.

License GPL-3

Encoding UTF-8

Depends R (>= 4.3.0)

Imports Matrix, methods, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

SystemRequirements C++17, optional OpenMP

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

URL https://github.com/metaddict/bgns

BugReports https://github.com/metaddict/bgns/issues

Author Aditya Kshirsagar [aut, cre]

Maintainer Aditya Kshirsagar <adityaksh4@gmail.com>

Repository CRAN

Date/Publication 2026-09-27 16:40:32 UTC

Contents

`bgns-package` *Biweight Graph and Network Statistics*

Description

bgns provides biweight midcorrelation kernels and exact bicor-based k-nearest-neighbor graph construction for dense matrices and sparse `dgCMatrix` inputs.

Details

bgns builds graphs from robust correlation structure, supporting workflows where similarity is better represented by coordinated variation. Tidy and KNN workflows are panelized so that a full dense similarity matrix does not need to be materialized.

Rows are observations and columns are items. In a KNN result, `col1` is the target column, `col2` is the selected neighbor, `val` is the bicor score, and `rank` is the within-target neighbor rank.

Runtime can be tuned with environment variables. `BGNS_MEM_MB` controls the scratch-memory budget in MB for panelization (default 256); it is not a limit on total process memory, which also includes inputs, outputs and other allocations. `BGNS_NUM_THREADS` controls OpenMP threads when available and defaults to at most two threads. `BGNS_SIMD` can be set to 0, false, or off to disable optional SIMD kernels when they are compiled in. `BGNS_BITSET_BETA` controls sparse overlap bitset construction (default 0.75). `BGNS_OMP_BUILD_THRESH` is the column-count threshold for parallel panel building (default 8). Set `min_overlap` through the function argument; it temporarily controls the native `BGNS_MIN_OVERLAP` setting. OpenMP settings do not control threads used independently by the BLAS library.

Author(s)

Aditya Kshirsagar

See Also

`bicor`, `bicor_knn`

`bicor` *Robust Biweight Midcorrelation*

Description

Compute biweight midcorrelations between columns of `x` and, optionally, between columns of `x` and `y`. Tidy output streams correlation edges without materializing a full dense correlation matrix.

Usage

```
bicor(x, y = NULL, pairwise.complete.obs = TRUE, spearman = FALSE,
      tidy = FALSE, threshold = -Inf, use_intersection_denominator = FALSE,
      min_overlap = 3)
```

Arguments

`x`	Numeric matrix with rows as observations and columns as items. Sparse objects inheriting from `sparseMatrix` are coerced to `dgCMatrix`; sparse inputs are supported when `tidy = TRUE`.
`y`	Optional numeric matrix or sparse matrix coercible to `dgCMatrix` with the same number of rows as `x`.
`pairwise.complete.obs`	Logical. If `TRUE`, compute each correlation on the finite pairwise overlap. If `FALSE`, pairs involving incomplete columns are omitted in tidy mode or returned as `NA` in dense matrix mode.
`spearman`	Logical. Accepted for API compatibility and ignored; this function computes biweight midcorrelation.
`tidy`	Logical. If `TRUE`, return a data frame with columns `col1`, `col2`, and `cor`.
`threshold`	Numeric. In tidy mode, keep pairs with $\|cor\| \geq threshold$. The default keeps all finite pairs.
`use_intersection_denominator`	Logical. For sparse or incomplete columns, use the overlap-specific denominator rather than full-column sums of squares.
`min_overlap`	Integer. Minimum number of shared finite observations required between two columns. The requirement also applies when both columns are fully finite. Defaults to 3.

Details

Column medians and scaled MADs (constant 1.4826) are estimated from finite observations. Biweight deviations use a tuning multiplier of 9. These estimates and weights are fixed before pairwise comparisons. The numerator uses shared finite observations; by default, the denominator uses full-column sums of squared weighted deviations. Setting `use_intersection_denominator = TRUE` restricts the sums to shared observations without refitting the weights. Implicit sparse entries are observed zeros. Row matching is positional.

Named edge outputs require unique, non-missing column names within each input. Unnamed inputs use column indices.

Input orientation is rows as observations and columns as variables, genes, cells, metacells, or other items to be compared.

When `tidy = FALSE`, both inputs must be dense base matrices. Sparse inputs are coerced to `dgCMatrix` and require `tidy = TRUE`. The tidy output stores pairwise correlations as an edge table and is intended for downstream graph construction.

For self-correlation matrices, a diagonal entry is 1 only when that column has a defined robust variance and satisfies `min_overlap`. Constant, all-missing, and otherwise ineligible columns have `NA` on the diagonal.

Value

A dense correlation matrix when `tidy = FALSE` and inputs are dense, or a data frame with columns `col1`, `col2`, and `cor` when `tidy = TRUE`.

Author(s)

Aditya Kshirsagar

See Also

`bicor_knn`

Examples

```
set.seed(1)
x <- matrix(rnorm(60), nrow = 12, ncol = 5)
x[1, 2] <- NA_real_

cm <- bicor(x)
cm

td <- bicor(x, tidy = TRUE, threshold = 0.1)
head(td)

x_sparse <- x
x_sparse[sample.int(length(x_sparse), 12)] <- 0
xs <- Matrix::Matrix(x_sparse, sparse = TRUE)
head(bicor(xs, tidy = TRUE, threshold = 0))
```

---

bicor_knn *k-Nearest Neighbors by Biweight Midcorrelation*

Description

Compute exact top-k neighbors per target column using biweight midcorrelation. The default streaming algorithm avoids storing a full dense similarity matrix.

Usage

```
bicor_knn(x, y = NULL, knn, pairwise.complete.obs = TRUE,
          threshold = -Inf, use_intersection_denominator = FALSE,
          direct_sparse = TRUE, bipartite_levels = c("strict", "separate"),
          min_overlap = 3)
```

Arguments

x	Numeric matrix or sparse matrix coercible to dgCMatrix with rows as observations and columns as source items.
y	Optional numeric matrix or sparse matrix coercible to dgCMatrix with the same number of rows as x. When supplied, neighbors are selected from columns of x for each target column of y.
knn	Integer. Maximum number of neighbors per target column. Requests exceeding the available candidates return all eligible neighbors.
pairwise.complete.obs	Logical. If TRUE, compute each edge on the finite overlap for that column pair. If FALSE, only pairs where both preprocessed columns are complete are eligible; pairs involving incomplete columns are omitted from the KNN output.
threshold	Numeric. Signed cutoff applied before top-k selection. Candidate bicor values must be strictly greater than threshold; for example, threshold = 0 keeps only positive bicor edges.
use_intersection_denominator	Logical. Use overlap-specific denominators for incomplete or sparse paths.
direct_sparse	Logical. If TRUE, retain top-k candidates while evaluating similarities in panels, which may be dense. If FALSE, a full similarity matrix may be used when inputs are fully finite, both use the same storage type, use_intersection_denominator = FALSE, and the estimated working allocation fits the scratch-memory budget. Otherwise use panels.
bipartite_levels	One of "strict" or "separate". Use "separate" when x and y have distinct column-name levels.
min_overlap	Integer. Minimum number of shared finite observations required between two columns. The requirement also applies when both columns are fully finite. Defaults to 3.

Details

Column medians and scaled MADs (constant 1.4826) are estimated from finite observations. Biweight deviations use a tuning multiplier of 9. These estimates and weights are fixed before pairwise comparisons. The numerator uses shared finite observations; by default, the denominator uses full-column sums of squared weighted deviations. Setting use_intersection_denominator = TRUE restricts the sums to shared observations without refitting the weights. Implicit sparse entries are observed zeros. Row matching is positional.

Named edge outputs require unique, non-missing column names within each input. Unnamed inputs use column indices.

Rows are observations and columns are items. With y = NULL, the result is a directed KNN graph among columns of x, with self-neighbors excluded. With y supplied, col1 is a target column of y and col2 is a selected neighbor column of x.

Neighbors are ordered by decreasing signed score, with smaller source-column indices resolving exact ties.

The function returns exact top-k neighbors under the implemented bicor score. It is memory-efficient because it keeps only candidate top-k buffers, but it is not an approximate subquadratic nearest-neighbor index. Sparse inputs other than `dgCMatrix` are coerced to `dgCMatrix`. Dense and sparse inputs may be mixed in bipartite calls, for example a dense matrix `x` and sparse `y`, or sparse `x` and dense `y`.

Value

A data frame with columns `col1`, `col2`, `val`, and `rank`. `val` is the bicor score and `rank` is the neighbor rank within each `col1` target.

Author(s)

Aditya Kshirsagar

See Also

`bicor`

Examples

```
set.seed(2)
x <- matrix(rnorm(80), nrow = 16, ncol = 5)
x[1, 2] <- NA_real_

kn <- bicor_knn(x, knn = 2, threshold = -Inf)
kn

kn_complete <- bicor_knn(x, knn = 2, pairwise.complete.obs = FALSE)
kn_complete

x_sparse <- x
x_sparse[sample.int(length(x_sparse), 16)] <- 0
xs <- Matrix::Matrix(x_sparse, sparse = TRUE)
bicor_knn(xs, knn = 2, threshold = 0)

y <- matrix(rnorm(48), nrow = 16, ncol = 3)
bicor_knn(xs, y, knn = 2, threshold = -Inf)
```

Index