

# Package: gStream (via r-universe)

October 16, 2024

**Version** 0.2.0

**Date** 2019-04-28

**Title** Graph-Based Sequential Change-Point Detection for Streaming Data

**Author** Hao Chen and Lynna Chu

**Maintainer** Hao Chen <hxchen@ucdavis.edu>

**Depends** R (>= 3.0.1)

**Description** Uses an approach based on k-nearest neighbor information to sequentially detect change-points. Offers analytic approximations for false discovery control given user-specified average run length. Can be applied to any type of data (high-dimensional, non-Euclidean, etc.) as long as a reasonable similarity measure is available. See references (1) Chen, H. (2019) Sequential change-point detection based on nearest neighbors. The Annals of Statistics, 47(3):1381-1407. (2) Chu, L. and Chen, H. (2018) Sequential change-point detection for high-dimensional and non-Euclidean data <[arXiv:1810.05973](https://arxiv.org/abs/1810.05973)>.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-05-01 22:00:03 UTC

## Contents

distM1 . . . . .	2
gStream . . . . .	2
gstream . . . . .	3

<b>Index</b>	<b>6</b>
--------------	----------

---

distM1	<i>An distance matrix constructed from L2 distance</i>
--------	--

---

### Description

This is the variable name for a distance matrix in the "Example" data. It is constructed from a sequence of 40 observations of dimension 10. The first 20 observations are considered historical observations. There is a change in mean at  $t = 10$ .

---

gStream	<i>Graph-Based Sequential Change-Point Detection</i>
---------	--

---

### Description

This package can be used to estimate change-points in a sequence of sequentially generated observations, where the observation can be a vector or a data object, e.g., a network. A distance matrix is required.

The function `gstream` will report the graph-based test statistics and the thresholds used in the stopping rules for a given average run length.

### Author(s)

Hao Chen and Lynna Chu

Maintainer: Hao Chen (hxchen@ucdavis.edu) and Lynna Chu (lbchu@ucdavis.edu)

### References

Chen, H. (2019) Sequential Change-point Detection Based on Nearest Neighbors. *The Annals of Statistics*, 47(3):1381-1407.

Chu, L. and Chen, H. (2018) Sequential Change-point Detection for High-dimensional and non-Euclidean Data. arXiv:1810.05973.

### See Also

[gstream](#)

### Examples

```
# This example contains two distance matrices constructed using L2 distance (distM1 and distM2).
# In this example, the data is treated as if it were being observed sequentially
# in order to illustrate how the package works.
```

```
# Example:
# distM1 is a distance matrix constructed from a dataset with n=40 observation.
# The first 20 observations are treated as historical observations.
```

```

# It has been determined that there are no change-points among the
# first 20 observations (see package gSeg for offline change-point detection).
# There is change in mean when tau = 20 (This means a change happens 20 observations
# after we start the tests. We start the test at N0+1 = 21.)
# The following code shows the data generating scheme to create distM1:
# (uncomment to run)
# N0 = 20 # the first 20 observations are historical observations
# N1 = N0 + 10
# N2 = N1 + 10
# d = 10
# vmu = 10
# set.seed(15)
# y1 = matrix(0,N1,d)
# y2 = matrix(0,N2-N1,d)
# for (i in 1:N1) y1[i,] = rnorm(d)
# for (i in 1:(N2-N1)) y2[i,] = rnorm(d, vmu)
# y = rbind(y1,y2)
# distM1 = as.matrix(dist(y))
# diag(distM1) = max(distM1)+100

# Uncomment the following to run
# N0 = 20
# L = 20 # the k-nn graph is constructed on only the L most recent observations.
# k = 1

# r1= gstream(distM1, L, N0, k, statistics="all", n0=0.3*L, n1=L-0.3*L,
# ARL=200,alpha=0.05, skew.corr=TRUE,asyp=FALSE)

# output results based on all four statistics; the scan statistics can be found in r1$scanZ
# r1$tauhat # reports the locations where a change-point is detected
# r1$b # reports the analytical approximations of the thresholds used in the stopping rules

# Set ARL = 10,000
# r1= gstream(distM1, L, N0, k, statistics="all", n0=0.3*L, n1=L-0.3*L,
# ARL=10000,alpha=0.05, skew.corr=TRUE,asyp=FALSE) # uncomment to run this function

```

---

gstream

*Sequential Change-Point Detection based on k-Nearest Neighbors*


---

## Description

This function finds change-points in the sequence when the underlying distribution changes. It reports four graph-based test statistics and the analytical approximations for thresholds used in their corresponding stopping rules.

## Usage

```

gstream(distM, L, N0, k, statistics = c("all", "o", "w", "g", "m"),
n0 = 0.3*L, n1 = 0.7*L, ARL = 10000, alpha = 0.05, skew.corr = TRUE, asyp = FALSE)

```

**Arguments**

distM	A distance matrix constructed based on some distance measure.
L	The number of observations the k-NN graph will be constructed from.
N0	The number of historical observations.
k	A fixed integer used to construct k-NN graph.
statistics	The scan statistic to be computed. A character indicating the type of scan statistic desired. The default is "all". "all": specifies to compute <b>all</b> of the scan statistics: original, weighted, generalized, and max-type; "o", "ori" or "original": specifies the <b>original</b> edge-count scan statistic; "w" or "weighted": specifies the <b>weighted</b> edge-count scan statistic; "g" or "generalized": specifies the <b>generalized</b> edge-count scan statistic; and "m" or "max": specifies the <b>max</b> -type edge-count scan statistic.
n0	The starting index to be considered as a candidate for the change-point. We recommend you set this to be $0.2 * L$
n1	The ending index to be considered as a candidate for the change-point. For example, $n1 = L - n0$ .
ARL	The average run length: the expectation of the stopping rule when there is no change-point.
alpha	The probability of an early stop.
skew.corr	Default is TRUE. If skew.corr is TRUE, the average run length approximation would incorporate skewness correction.
asymp	Default is FALSE. If asymp is TRUE, the average run length approximation will be based on the asymptotic analytical formulas.

**Value**

Returns a list with items scanZ, b and tauhat for each type of statistic specified. See below for more details.

scanZ	A vector of the test statistic (maximum of the scan statistics) for each time $n = N0+1, \dots, N$ . ori: A vector of the original scan statistics (standardized counts) if statistic specified is "all" or "o". weighted: A vector of the weighted scan statistics (standardized counts) if statistic specified is "all" or "w". generalized: A vector of the generalized scan statistics (standardized counts) if statistic specified is "all" or "g". max.type: A vector of the max-type scan statistics (standardized counts) if statistic specified is "all" or "m".
b	Thresholds used in the stopping rules for each test statistic. These thresholds are based on analytical approximations of the average run length.
tauhat	Estimate of the locations of change-points based on the thresholds.

**See Also**[gStream](#)**Examples**

```
# This example contains two distance matrices (distM1 and distM2).
# Information on how distM1 and distM2 are generated can be found in gStream.

# data(Example)

# Example:
# distM1 is a distance matrix constructed from a dataset with n=40 observation.
# The first 20 observations are treated as historical observations.
# It has been determined that there are no change-points among the
# first 20 observations (see package gSeg for offline change-point detection).
# There is change in mean when tau = 20 (This means a change happens 20 observations
# after we start the tests. We start the test at N0+1 = 21.)

# Uncomment the following to run
# N0 = 20
# L = 20 # the k-nn graph is constructed on only the L most recent observations.
# k = 1

# r1= gstream(distM1, L, N0, k, statistics="all", n0=0.3*L, n1=0.7*L,
# ARL=200,alpha=0.05, skew.corr=TRUE, asymp=FALSE)
# output results based on all four statistics; the scan statistics can be found in r1$scanZ
# r1$tauhat # reports the locations where a change-point is detected
# r1$b # reports the analytical approximations of the thresholds used in the stopping rules

# Set ARL = 10,000
# r1= gstream(distM1, L, N0, k, statistics="all", n0=0.3*L, n1=L-0.3*L,
# ARL=10000,alpha=0.05, skew.corr=TRUE, asymp=FALSE) # uncomment to run this function
```

# Index

`distM1`, [2](#)

`gStream`, [2](#), [5](#)

`gstream`, [2](#), [3](#)