

# Package ‘vccp’

July 22, 2025

**Title** Vine Copula Change Point Detection in Multivariate Time Series

**Version** 0.1.1

**Author** Xin Xiong [aut, cre],  
Ivor Cribben [aut]

**Maintainer** Xin Xiong <xinxiong@hsph.harvard.edu>

**Description** Implements the Vine Copula Change Point (VCCP) methodology for the estimation of the number and location of multiple change points in the vine copula structure of multivariate time series. The method uses vine copulas, various state-of-the-art segmentation methods to identify multiple change points, and a likelihood ratio test or the stationary bootstrap for inference. The vine copulas allow for various forms of dependence between time series including tail, symmetric and asymmetric dependence. The functions have been extensively tested on simulated multivariate time series data and fMRI data. For details on the VCCP methodology, please see Xiong & Cribben (2021).

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Imports** VineCopula, stats, graphics, mosum, mvtnorm

**Suggests** knitr, rmarkdown

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-05-29 10:40:02 UTC

## Contents

getTestPlot . . . . .	2
mvn.sim.2.cps . . . . .	2
vccp . . . . .	3
vccp.fun . . . . .	4

Index	7
-------	---

---

getTestPlot	<i>Plot output from the VCCP model</i>
-------------	--

---

**Description**

This function plots the change points in the network structure between multivariate time series detected by the VCCP model.

**Usage**

```
getTestPlot(vccp_result)
```

**Arguments**

`vccp_result` A list generated from `vccp.fun`.

**Value**

No return value, called for a plotting purpose.

**See Also**

[vccp.fun](#)

**Examples**

```
## Simulate MVN data with 2 change points
data = cbind(1:180, mvn.sim.2.cps(180,8,seed=101))
## Change point detection using VCCP (it may take several minutes to complete...)
result = vccp.fun(data, "NBS", test = "V")
## Plot the result
getTestPlot(result)

result.2 = vccp.fun(data, "NBS", test = "B")
## Plot the result
getTestPlot(result.2)
```

---

mvn.sim.2.cps	<i>Simulate multivariate normal data with 2 change points</i>
---------------	---

---

**Description**

This function simulates a multivariate normal data set with 2 change points in the network structure.

**Usage**

```
mvn.sim.2.cps(nobs, n_ser, seed = 101)
```

**Arguments**

nobs	A positive integer, which defines the length of the time series. It must be a multiple of 3 since change points occur at time points $T/3$ and $2T/3$ .
n_ser	A positive integer number indicating the dimensionality of the time series. n_ser must be equal to or larger than 8 as mvn.sim.2.cps generates 3 different network structures among 8 connected nodes. The remaining variables are generated as independent data.
seed	A positive integer with default value equal to 101. It is used to ensure reproducibility.

**Value**

A  $nobs * n\_ser$  matrix with 2 change points at time points  $nobs/3+1$  and  $nobs*2/3+1$ .

**Examples**

```
## Simulate MVN data with 2 change points
data <- mvn.sim.2.cps(180, 8, seed = 101)
```

---

vccp	<i>vccp: Detect multiple change points in the vine copula structure of multivariate time series by Vine Copula Change Point Model</i>
------	---

---

**Description**

The vccp package implements the Vine Copula Change Point (VCCP) methodology for the estimation of the number and location of multiple change points in the vine copula structure of multivariate time series. The method uses vine copulas, various state-of-the-art segmentation methods to identify multiple change points, and a likelihood ratio test or the stationary bootstrap for inference. The vine copulas allow for various forms of dependence between time series including tail, symmetric and asymmetric dependence. The functions have been extensively tested on simulated multivariate time series data and fMRI data. For details on the VCCP methodology, please see Xiong & Cribben (2021).

**vccp functions**

[mvn.sim.2.cps](#), [getTestPlot](#) and [vccp.fun](#)

**Author(s)**

Xin Xiong, Ivor Cribben (<cribben@ualberta.ca>)

**References**

"Beyond linear dynamic functional connectivity: a vine copula change point model", Xiong and Cribben (2021), bioRxiv 2021.04.25.441254.

## Examples

```
# See examples in the function vccp.fun.
```

---

vccp.fun	<i>Multiple change point detection in the vine copula structure of multivariate time series</i>
----------	---

---

## Description

This function detects multiple change points in the vine copula structure of a multivariate time series using vine copulas, various state-of-the-art segmentation methods to identify multiple change points, and a likelihood ratio test or the stationary bootstrap for inference.

## Usage

```
vccp.fun(
  X,
  method = "NBS",
  delta = 30,
  G = 0.1,
  M = NA,
  test = "V",
  CDR = "D",
  trunc_tree = NA,
  family_set = 1,
  pre_white = 0,
  ar_num = 1,
  p = 0.3,
  N = 100,
  sig_alpha = 0.05
)
```

## Arguments

X	A numerical matrix representing the multivariate time series, with the columns representing its components. If multiple subjects are included (panel data), vertically stack the subject data and identify timestamps of each subject in the first column.
method	A character string, which defines the segmentation method. If method = "NBS", which is the default method, then the adapted binary segmentation is used. Similarly, if method = "OBS", "MOSUM" or "WBS", then binary segmentation, MOSUM and wild binary segmentation are used, respectively.
delta	A positive integer number with default value equal to 30. It is used to define the minimum distance acceptable between change points. In general, $\text{delta} \geq 5 \times \text{ncol}(X)$ is recommended to ensure sufficient data when estimating the vine copula model.

G	A positive real number between 0 and 1 with default value equal to 0.1. It is used to define the moving sum bandwidth relative to T in MOSUM when method = "MOSUM" is chosen. Alternatively, a positive integer less than half of the time series length can be set to define the absolute bandwidth.
M	A positive integer with default value equal to $\text{floor}(9 \cdot \log(T))$ (T is the length of the time series). It represents the number of sub-samples in WBS when method = "WBS" is chosen.
test	A character string, which defines the inference method used. If test = "V", which is the default method, the Vuong test is performed. If test = "B", the stationary bootstrap is performed.
CDR	A character string, which defines the vine structure. If CDR = "D", which is the default method, a D-vine is used. Similarly, if CDR = "C" or CDR = "R", a C-vine or an R-vine is used, respectively.
trunc_tree	A positive integer, which defines the level of truncation for the vine copula. If trunc_tree = "NA", which is the default value, the Vine contains $\text{dim}(X)[2]-2$ levels of trees.
family_set	A positive integer, which defines the bivariate copula family. If familyset = 1, which is the default value, only the Gauss copula is selected and VCCP detects change points in the linear correlation graph. Coding of pair-copula families is the same as in <a href="#">BiCop</a> .
pre_white	A positive integer, which defines whether the data is pre-whitened. If pre-white = 0, which is the default value, no pre-whitening is performed. If pre_white = 1, an autoregressive time series model (method: yule-walker) is used to preprocess the raw data.
ar_num	A positive integer, which defines the maximum order of model to fit to preprocess the data (see pre_white). If ar_num = 1, which is the default value, then an AR(1) model is fit to the data.
p	A positive real number between 0 and 1 which is defined as the block size in the stationary bootstrap method ( $\text{rgeom}(T, p)$ ) if test = "B" is chosen. If p = 0.3, which is the default value, each resampled block has 1/0.3 time points on average.
N	A positive integer, which defines the number of the stationary bootstrap resamples used. The default value is N = 100.
sig_alpha	A positive real number between 0 and 1, which defines the significance level of the inference test. The default values is 0.05.

## Details

The time series  $X_t$  is of dimensionality p and we are looking for changes in the vine copula structure between the different time series components  $X_{\{t\}^{\{1\}}}$ ,  $X_{\{t\}^{\{2\}}}$ ,  $\dots$ ,  $X_{\{t\}^{\{p\}}}$ . VCCP uses vine copulas, various state-of-the-art segmentation methods to identify multiple change points, and a likelihood ratio test or the stationary bootstrap for inference.

## Value

A list with the following components:

loc_of_cpts	The locations of the detected change points.
no_of_cpts	The number of detected change points.
test_df	A dataframe containing the test result.
compute_time	Time (in minutes) to run vccp.fun.
T	The length of the time series data.
sig_alpha	The significance level for the inference test.

### Author(s)

Xin Xiong, Ivor Cribben (<cribben@ualberta.ca>)

### References

"Beyond linear dynamic functional connectivity: a vine copula change point model", Xiong and Cribben (2021), bioRxiv 2021.04.25.441254.

### See Also

[getTestPlot](#)

### Examples

```
## Simulate MVN data with 2 change points
data <- cbind(1:180, mvn.sim.2.cps(180, 8, seed = 101))
T <- 180
## Change point detection using VCCP (it may take several minutes to complete...)
result.NV <- vccp.fun(data, method = "NBS", delta = 30, test = "V")
## Plot the results
getTestPlot(result.NV)
#title("VCCP: NBS + Vuong")

## Change point detection using NBS and stationary bootstrap for inference
result.NB <- vccp.fun(data, method = "NBS", delta = 30, test = "B")
## Plot the results
getTestPlot(result.NB)
title("VCCP: NBS + Stationary Bootstrap")
```

# Index

BiCop, [5](#)

getTestPlot, [2](#), [3](#), [6](#)

mvn.sim.2.cps, [2](#), [3](#)

vccp, [3](#)

vccp.fun, [2](#), [3](#), [4](#)