# Package ‘CCMnet’

February 13, 2026

**Type** Package

**Title** Congruence Class Models for Networks

**Version** 0.1.4

**Description** Provides an implementation of Congruence Class Models for generating networks. It facilitates sampling networks based on specific topological properties and attribute mixing patterns using a Markov Chain Monte Carlo framework. The implementation builds upon code from the 'ergm' package; see Handcock et al. (2008) <doi:10.18637/jss.v024.i01>.

**License** GPL-3

**Encoding** UTF-8

**Imports** dplyr, ergm, ggplot2, gtools, igraph, intergraph, kableExtra, mvtnorm, network, RBesT, rlang, stats, tibble, tidyr, utils

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**NeedsCompilation** yes

**Author** Ravi Goyal [aut, cre],
Statnet Development Team [ctb, cph]

**Maintainer** Ravi Goyal <`ravi.j.goyal@gmail.com`>

**Repository** CRAN

**Date/Publication** 2026-02-13 16:40:14 UTC

## Contents

---

## `CCM_theoretical_check` *Compare MCMC Estimates with Theoretical Distributions*

---

### Description

This function compares the empirical MCMC distribution from `sample_ccm` with the theoretical distribution from the CCM model. It is used as a diagnostic tool to evaluate model adequacy.

### Usage

```
CCM_theoretical_check(fit, n_sim = nrow(fit$mcmc_stats))
```

### Arguments

| | |
|---|---|
| `fit` | An object returned by `ccm_sample`. |
| `n_sim` | The number of samples drawn from the theoretical distribution |

### Value

A diagnostic plot comparing empirical and theoretical distributions.

### Examples

```
ccm_sample <- sample_ccm(
  network_stats = list("edges"),
  prob_distr = list("poisson"),
  prob_distr_params = list(list(350)),
  population = 50
)
ccm_sample<- CCM_theoretical_check(ccm_sample, n_sim = 1000)
plot(ccm_sample, stats = "edges", type = "hist", include_theoretical = TRUE)
```

---

## `CCM_traceplot` *Plot MCMC Trace for sample_ccm*

---

### Description

`CCM_traceplot` produces a trace plot of the MCMC samples from `sample_ccm`. This is used to diagnose convergence and mixing.

### Usage

```
CCM_traceplot(object, stats = NULL, ...)
```

### Arguments

| | |
|---|---|
| `object` | A `ccm_sample` object. |
| `stats` | string. Which statistic to plot. |
| `...` | Additional arguments passed to `plot()`. |

### Value

A trace plot for the selected MCMC chain.

### Examples

```
ccm_sample <- sample_ccm(
  network_stats = list("edges"),
  prob_distr = list("poisson"),
  prob_distr_params = list(list(350)),
  population = 50
)
CCM_traceplot(ccm_sample, stats = "edges")
```

---

## `sample_ccm` *Fit a Congruence Class Model (CCM)*

### Description

`sample_ccm` fits a Congruence Class Model using an MCMC framework to sample networks that match specific topological property distributions. It facilitates sampling based on specified network statistics (e.g., edges, degree distribution, mixing patterns) and their associated probability distributions.

### Usage

```
sample_ccm(
  network_stats,
  prob_distr,
  prob_distr_params,
  population,
  sample_size = 1000L,
  burnin = 200000L,
  interval = 1000L,
  cov_pattern = NULL,
  initial_g = NULL,
  use_initial_g = FALSE,
  partial_network = as.integer(0),
  obs_nodes = NULL,
  Obs_stats = NULL,
  remove_var_last_entry = FALSE,
  stats_only = TRUE
)
```

## Arguments

| | |
|---|---|
| `network_stats` | Character vector of statistic names to be constrained (e.g., "edges", "degree", "density"). |
| `prob_distr` | Character vector of probability distribution names corresponding to each statistic. |
| `prob_distr_params` | List of parameter sets for each specified distribution. |
| `population` | Integer. The number of nodes in the network. |
| `sample_size` | Integer. Number of MCMC samples to return. Default is 1000. |
| `burnin` | Integer. Number of MCMC iterations to discard before sampling begins. Default is 200,000. |
| `interval` | Integer. Thinning interval (number of iterations between samples). Default is 1000. |
| `cov_pattern` | Integer vector. Optional group labels or covariate patterns for nodes. |
| `initial_g` | An `igraph` object. The starting graph for the MCMC chain. |
| `use_initial_g` | Logical. If TRUE, the MCMC chain starts from `initial_g`. |
| `partial_network` | Integer. Reserved for future use in partial network observation. |
| `obs_nodes` | Integer vector. Reserved for future use in specifying observed nodes. |
| `Obs_stats` | Character vector of additional network statistics to monitor during sampling. |
| `remove_var_last_entry` | Logical. If TRUE, removes the variance constraint from the last entry of the distribution. |
| `stats_only` | Logical. If TRUE, only sufficient statistics are returned; otherwise, network objects are included. |

## Value

An object of class `ccm_sample` containing:

- `mcmc_stats`: A data frame of sampled network statistics.
- `population`: The number of nodes in the network.
- `prob_distr`: The distributions used for constraints.
- `prob_distr_params`: Parameters used for the constraints.
- `network_stats`: The names of the statistics constrained.
- `cov_pattern`: The covariate pattern used.
- `theoretical`: Theoretical distribution values, if calculated.
- `g`: A list of sampled graphs.

### Examples

```
# Basic sampling of a random graph with an edge constraint
ccm_sample <- sample_ccm(
  network_stats = list("edges"),
  prob_distr = list("poisson"),
  prob_distr_params = list(list(350)),
  population = 50
)
summary(ccm_sample)
plot(ccm_sample, stats = "edges", type = "hist")
```

# Index