

Package: rbcpd (via r-universe)

July 19, 2026

Type Package

Title An R package-wrapper for Fast Bayesian Coherent Point Drift and BCPD++ Registration

Version 0.1.0

Description Native R bindings to a bundled, pinned implementation of Bayesian Coherent Point Drift. The package implements the accelerated BCPD++ workflow in memory: representative-point downsampling, Nystrom low-rank deformation and probability approximations, k-d-tree probability search near convergence, and interpolation of the deformation back to every original moving point.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

SystemRequirements C++17, BLAS, LAPACK

NeedsCompilation yes

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Repository <https://bimsbbioinfo.r-universe.dev>

Date/Publication 2026-07-19 15:58:47 UTC

RemoteUrl <https://github.com/BIMSBbioinfo/rbcpd>

RemoteRef HEAD

RemoteSha 174b75a36e327567b6afd2f65bdd069c490a7ad0

Contents

apply_bcpd	2
bcpd_backend_info	2
bcpd_control	3
bcpd_register	4
print.bcpd_control	5
run_bcpd	6

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

apply_bcpd	<i>Apply a fitted BCPD transformation</i>
------------	---

Description

Apply a fitted BCPD transformation for new points from the moving coordinate system to fixed coordinate system

Usage

```
apply_bcpd(model, newdata)
```

Arguments

- | | |
|---------|--|
| model | a list of parameters defining fitted BPCD model |
| newdata | Numeric matrix in the original moving coordinate system. |

bcpd_backend_info	<i>Report Native BCPD Backend Capabilities</i>
-------------------	--

Description

Reports compile-time capabilities of the bundled native BCPD backend.

Usage

```
bcpd_backend_info()
```

Value

A list containing backend identity, pinned commit, OpenMP status, maximum thread count, and OpenCV status.

bcpd_control

*Configure Native BCPD and BCPD++ Registration***Description**

Creates and validates controls for the native BCPD backend and the downsample-register-interpolate BCPD++ workflow.

Usage

```
bcpd_control(
  transformation = c("similarity_nonrigid", "nonrigid", "affine_nonrigid",
    "affine", "similarity", "rigid"),
  omega = 0, lambda = 2, beta = 2, gamma = 1, tolerance = 1e-4,
  max_iterations = 150L, min_iterations = 1L,
  normalization = c("each", "fixed", "moving", "none"),
  downsample = 1000L, downsample_fixed = NULL,
  downsample_moving = NULL, nystrom_deformation = 100L,
  nystrom_probability = 300L, kd_tree = TRUE, kd_scale = 7,
  kd_radius = 0.15, kd_switch = 0.2, seed = 1L, verbose = FALSE
)
```

Arguments

transformation	Transformation model.
omega	Outlier probability in $[0, 1)$.
lambda	Positive deformation regularization.
beta	Positive deformation-kernel width.
gamma	Positive initial residual multiplier.
tolerance	Positive convergence threshold.
max_iterations	Maximum number of iterations.
min_iterations	Minimum iteration index before convergence testing.
normalization	Normalization strategy.
downsample	Representative-point count for both clouds. Zero uses all points.
downsample_fixed	Optional fixed-cloud override.
downsample_moving	Optional moving-cloud override.
nystrom_deformation	Nyström rank for the deformation kernel.
nystrom_probability	Nyström rank for probability calculations.
kd_tree	Use local k-d-tree probabilities near convergence.

kd_scale	Search radius in residual-standard-deviation units.
kd_radius	Maximum normalized search radius.
kd_switch	Residual standard deviation at which k-d-tree search starts.
seed	Non-negative random seed.
verbose	Print compact iteration progress.

Value

An object of class `bcpd_control`.

bcpd_register	<i>Register Point Clouds with BCPD or BCPD++</i>
---------------	--

Description

Deforms moving onto fixed. When representative-point downsampling is active, the deformation is interpolated back to all original moving points.

Usage

```
bcpd_register(fixed, moving, control = bcpd_control())

bcpd_pp(
  fixed, moving, downsample = 1000L, nystrom_deformation = 100L,
  nystrom_probability = 300L, tolerance = 1e-4,
  max_iterations = 150L, omega = 0, lambda = 2, beta = 2,
  seed = 1L, verbose = FALSE
)
```

Arguments

fixed	Numeric fixed or target point matrix. Rows are points.
moving	Numeric moving or source point matrix. Rows are points.
control	A <code>bcpd_control</code> object.
downsample	Representative points selected from each cloud.
nystrom_deformation	Deformation-kernel Nyström rank.
nystrom_probability	Probability Nyström rank.
tolerance	Convergence tolerance.
max_iterations	Maximum iterations.
omega	Outlier probability.
lambda	Deformation regularization.
beta	Deformation-kernel width.
seed	Random seed.
verbose	Print iteration progress.

Value

A bcpd_registration list. Its transformed component has the same dimensions as moving.

Examples

```
theta <- seq(0, 2 * pi, length.out = 101)[-101]
moving <- cbind(cos(theta), sin(theta))
fixed <- sweep(moving, 2, c(0.2, -0.1), "+")
## Not run:
fit <- bcpd_pp(fixed, moving, downsample = 60L)
plot(fixed, asp = 1)
points(fit$transformed, col = 2)

## End(Not run)
```

print.bcpd_control	<i>Print BCPD Objects</i>
--------------------	---------------------------

Description

Displays the most important settings or registration diagnostics.

Usage

```
## S3 method for class 'bcpd_control'
print(x, ...)
## S3 method for class 'bcpd_registration'
print(x, ...)
```

Arguments

x	A BCPD control or registration object.
...	Unused.

Value

The input object, invisibly.

run_bcpd

Register point clouds with native BCPD or BCPD++

Description

Rows are points and columns are coordinate dimensions. moving is deformed onto fixed. When the moving cloud is downsampled, the native backend interpolates the estimated deformation back to every original moving point.

Usage

```
run_bcpd(fixed, moving, control = bcpd_control())
```

Arguments

fixed	Numeric fixed/target point matrix.
moving	Numeric moving/source point matrix.
transformation	transformation
downsample	Representative points selected from each cloud.
nystrom_deformation	Deformation-kernel Nyström rank.
nystrom_probability	Probability Nyström rank.
tolerance	Convergence tolerance.
max_iterations	Maximum iterations.
omega	Outlier probability.
lambda	Deformation regularization.
beta	Deformation-kernel width.
seed	Random seed.
verbose	Print iteration progress.

Value

A list of results of the BCPD run. The most important component is transformed, which has the same dimensions as moving.

Index

`apply_bcpd`, [2](#)

`bcpd_backend_info`, [2](#)

`bcpd_control`, [3](#)

`bcpd_pp (bcpd_register)`, [4](#)

`bcpd_register`, [4](#)

`print.bcpd_control`, [5](#)

`print.bcpd_registration`
 (`print.bcpd_control`), [5](#)

`run_bcpd`, [6](#)