Package ‘metadynminer3d’

October 13, 2022

Type Package

Title Tools to Read, Analyze and Visualize Metadynamics 3D HILLS Files from 'Plumed'

Version 0.0.2

Date 2022-04-14

Depends R (>= 3.3.0), metadynminer, rgl

LinkingTo Rcpp

Description Metadynamics is a state of the art biomolecular simulation technique. 'Plumed' Tribello, G.A. et al. (2014) <doi:10.1016/j.cpc.2013.09.018> program makes it possible to perform metadynamics using various simulation codes. The results of metadynamics done in 'Plumed' can be analyzed by 'metadynminer'. The package 'metadynminer' reads 1D and 2D metadynamics hills files from 'Plumed' package. As an addendum, 'metadynaminer3d' is used to visualize 3D hills. It uses a fast algorithm by Hosek, P. and Spiwok, V. (2016) <doi:10.1016/j.cpc.2015.08.037> to calculate a free energy surface from hills. Minima can be located and plotted on the free energy surface. Free energy surfaces and minima can be plotted to produce publication quality images.

LazyData true

License GPL-3

RoxygenNote 6.1.0

Imports Rcpp, misc3d

Suggests testthat

URL https://metadynamics.cz/metadynminer3d/

NeedsCompilation yes

Author Vojtech Spiwok [aut, cre] (<https://orcid.org/0000-0001-8108-2033>)

Maintainer Vojtech Spiwok <spiwokv@vscht.cz>

Repository CRAN

Date/Publication 2022-04-14 14:32:34 UTC

R topics documented:

`acealanme3d` *Hills from 30 ns metadynamics of AceAlaNme in water with three collective variable*

Description

Hills from 30 ns metadynamics of AceAlaNme (Amber99SB-ILDN) in water (TIP3P) with a Ramachandran dihedral phi and psi and peptide bond torsion omega as the collective variable.

Usage

```
acealanme3d
```

Format

hillsfile3d object

Source

http://www.metadynamics.cz/metadynminer/data/HILLS3d

feprof.minima3d *Calculate free energy profile for minima3d object*

Description

‘feprof.minima3d‘ calculates free energy profiles for free energy minima. It finds the global minimum at the ‘imax‘ and calculates the evolution of free energies of a local vs. the global free energy minimum. The free energy of the global minimum is constant (zero).

Usage

```
## S3 method for class 'minima3d'
feprof(minims, imax = NULL)
```

Arguments

minims	minima3d object.
imax	index of a hill from which summation stops (default the rest of hills).

Examples

```
library(metadynminer3d)
tfes<-fes(acealanme3d, imax=5000)
minima<-fesminima(tfes)
prof<-feprof(minima)
prof
```

fes.hillsfile3d *Calculate 3D free energy surface by Bias Sum algorithm*

Description

‘fes.hillsfile3d‘ sums up hills using fast Bias Sum algorithm.

Usage

```
## S3 method for class 'hillsfile3d'
fes(hills, imin = 1, imax = NULL, xlim = NULL,
  ylim = NULL, zlim = NULL, npoints = NULL)
```

Arguments

`hills`	hillsfile3d object.
`imin`	index of a hill from which summation starts (default 1).
`imax`	index of a hill from which summation stops (default the rest of hills).
`xlim`	numeric vector of length 1, giving the CV2 coordinates range.
`ylim`	numeric vector of length 2, giving the CV2 coordinates range.
`zlim`	numeric vector of length 2, giving the CV3 coordinates range.
`npoints`	resolution of the free energy surface in number of points.

Value

fes object.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
```

`fes2.hillsfile3d` *Calculate 3D free energy surface by conventional algorithm*

Description

‘fes2.hills3d‘ sums up hills using slow conventional algorithm. It can be used as a reference or when hill widths are variable.

Usage

```
## S3 method for class 'hillsfile3d'
fes2(hills, imin = 1, imax = NULL, xlim = NULL,
  ylim = NULL, zlim = NULL, npoints = NULL)
```

Arguments

`hills`	hillsfile3d object.
`imin`	index of a hill from which summation starts (default 1).
`imax`	index of a hill from which summation stops (default the rest of hills).
`xlim`	numeric vector of length 2, giving the CV1 coordinates range.
`ylim`	numeric vector of length 2, giving the CV2 coordinates range.
`zlim`	numeric vector of length 2, giving the CV3 coordinates range.
`npoints`	resolution of the free energy surface in number of points.

Value

fes object.

Examples

```
tfes<-fes2(acealanme3d, imax=100)
```

`fesminima.fes3d` *Find free energy minima in the fes3d object*

Description

'fesminima.fes3d' finds free energy minima on 3D free energy surface. The surface is divided by a 3D grid and minima are found for each bin. Next the program determines whether the minimum of a bin is a local minimum of the whole free energy surface. Free energy minima are labeled constitutively by capital letters.

Usage

```
## S3 method for class 'fes3d'
fesminima(inputfes, nbins = 8)
```

Arguments

`inputfes`	fes3d object.
`nbins`	number of bins for each CV (default 8).

Value

minima object.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
minima<-fesminima(tfes)
minima
```

`fespoint.hillsfile3d` *Calculate free energy at given point in the CV space*

Description

'fespoint' calculates free energy at given point in the CV space 'coord'. Hills are summed from 'imin' to 'imax'. Printed output can be suppressed by setting 'verb' to TRUE.

Usage

```
## S3 method for class 'hillsfile3d'
fespoint(hills, coord = NULL, imin = 1,
  imax = NULL, verb = T)
```

Arguments

`hills` hillsfile3d object.

`coord` coordinates of the point in the CV space.

`imin` index of a hill from which calculation of difference starts (default 1).

`imax` index of a hill from which summation stops (default the rest of hills).

`verb` if TRUE, the output is verbose (default TRUE).

Examples

```
fespoint(acealanme3d, c(0,0,0), imax=5000)
```

`head.hillsfile3d` *Print first n lines of hillsfile3d*

Description

‘head.hillsfile3d‘ prints first n lines of a hillsfile3d object.

Usage

```
## S3 method for class 'hillsfile3d'
head(x, n = 10, ...)
```

Arguments

`x` hillsfile3d object.

`n` number of lines (default 10).

`...` further arguments passed to or from other methods.

Examples

```
head(acealanme3d)
```

max.fes3d *Calculate maximum of 3D free energy surface*

Description

‘max.fes3d‘ calculates maximum of free energy in a fes3d object.

Usage

```
## S3 method for class 'fes3d'
max(inputfes, na.rm = NULL, ...)
```

Arguments

`inputfes`	fes3d object.
`na.rm`	a logical indicating whether missing values should be removed.
`...`	further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
max(tfes)
```

min.fes3d *Calculate minimum of 3D free energy surface*

Description

‘min.fes3d‘ calculates minimum of free energy in a fes3d object.

Usage

```
## S3 method for class 'fes3d'
min(inputfes, na.rm = NULL, ...)
```

Arguments

`inputfes`	fes3d object.
`na.rm`	a logical indicating whether missing values should be removed.
`...`	further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
min(tfes)
```

oneminimum.fes3d *Creates one ad hoc 3D free energy minimum for a fes object*

Description

‘oneminimum.fes3d‘ creates an ad hoc 3D free energy minimum on free energy surface. This can be used to calculate 3D free energy surface evolution at arbitrary point of free energy surface.

Usage

```
## S3 method for class 'fes3d'
oneminimum(inputfes, cv1, cv2, cv3)
```

Arguments

inputfes	fes3d object.
cv1	the value of collective variable 1.
cv2	the value of collective variable 2.
cv3	the value of collective variable 3.

Value

minima object.

Examples

```
tfes<-fes(acealanme3d, imax=1000)
minima<-fesminima(tfes)
minima<-minima+oneminimum(tfes, cv1=0, cv2=0, cv3=0)
minima
```

plot.fes3d *Plot 3D free energy surface object*

Description

‘plot.fes3d‘ plots 3D free energy surface using .

Usage

```
## S3 method for class 'fes3d'
plot(x, xlab = NULL, ylab = NULL, zlab = NULL,
  xlim = NULL, ylim = NULL, zlim = NULL, level = NULL,
  col = NULL, alpha = NULL, main = NULL, sub = NULL, fill = TRUE, ...)
```

Arguments

`x` fes3d object.

`xlab` a title for the x axis: see ’title’.

`ylab` a title for the y axis: see ’title’.

`zlab` a title for the z axis: see ’title’.

`xlim` numeric vector of length 2, giving the x coordinates range.

`ylim` numeric vector of length 2, giving the y coordinates range.

`zlim` numeric vector of length 2, giving the z coordinates range.

`main` an overall title for the plot: see ’title’.

`sub` a sub title for the plot: see ’title’.

`level` number or numeric vector of levels at which to draw 3D isosurface.

`col` color of the free energy surface. It can be a single color or a vector with multiple colors for multiple 3D isosurfaces.

`alpha` number or numeric vector of alpha levels (transparency) of 3D isosurfaces.

`fill` a logical value indicating whether 3D isosurface is ploted as solid surface (True) or wireframe (False).

`...` further arguments passed to or from other methods.

Examples

```
tfes3d<-fes(acealanme3d, imax=5000)
plot(tfes3d)
```

`plot.hillsfile3d` *Plot hillsfile3d object*

Description

‘plot.hillsfile3d‘ plots hillsfile object. It plots CV1 vs CV2 vs CV3.

Usage

```
## S3 method for class 'hillsfile3d'
plot(x, xlab = "CV1", ylab = "CV2", zlab = "CV3",
  main = NULL, sub = NULL, col = "orange", ...)
```

Arguments

`x` hillsfile object.

`xlab` a title for the x axis: see ’title’.

`ylab` a title for the y axis: see ’title’.

`zlab` a title for the z axis: see ’title’.

`main` an overall title for the plot: see ’title’.

`sub` a sub title for the plot: see ’title’.

`col` color code or name, see ’par’.

`...` further arguments passed to or from other methods.

Examples

```
plot(acealanme3d)
```

`plot.minima3d` *Plot minima3d object*

Description

‘plot.minima3d‘ plots 3D free energy surface with minima. The free energy surface is plotted the same way as by plot.fes3d with additional minima labels.

Usage

```
## S3 method for class 'minima3d'
plot(x, xlab = "CV1", ylab = "CV2", zlab = "CV3", level = NULL,
  col = NULL, alpha = NULL, main = NULL, sub = NULL, fill = TRUE, ...)
```

Arguments

`x` minima3d object.

`main` an overall title for the plot: see ’title’.

`sub` a sub title for the plot: see ’title’.

`xlab` a title for the x axis: see ’title’.

`ylab` a title for the y axis: see ’title’.

`zlab` a title for the z axis: see ’title’.

`level` number or numeric vector of levels at which to draw 3D isosurface.

`col` color of the free energy surface. It can be a single color or a vector with multiple colors for multiple 3D isosurfaces.

`alpha` number or numeric vector of alpha levels (transparency) of 3D isosurfaces.

`fill` a logical value indicating whether 3D isosurface is ploted as solid surface (True) or wireframe (False).

`...` further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
minima<-fesminima(tfes)
plot(minima)
```

---

plotheights.hillsfile3d

Plot evolution of heights of hills in hillsfile3d object

Description

‘plotheights.hillsfile3d‘ plots evolution of heights of hills. In well tempered metadynamics hill heights decrees with flooding of the free energy surface. Evolution of heights may be useful to evaluate convergence of the simulation.

Usage

```
## S3 method for class 'hillsfile3d'
plotheights(hills, ignoretime = FALSE, xlab = NULL,
  ylab = NULL, xlim = NULL, ylim = NULL, main = NULL, sub = NULL,
  col = "black", asp = NULL, lwd = 1, axes = TRUE)
```

Arguments

`hills`	hillsfile object.
`ignoretime`	time in the first column of the HILLS file will be ignored.
`xlab`	a title for the x axis: see ’title’.
`ylab`	a title for the y axis: see ’title’.
`xlim`	numeric vector of length 2, giving the x coordinates range.
`ylim`	numeric vector of length 2, giving the y coordinates range.
`main`	an overall title for the plot: see ’title’.
`sub`	a sub title for the plot: see ’title’.
`col`	color code or name, see ’par’.
`asp`	the y/x aspect ratio, see ’plot.window’.
`lwd`	line width for drawing symbols see ’par’.
`axes`	a logical value indicating whether both axes should be drawn on the plot.

Examples

```
plotheights(acealanme3d)
```

print.fes3d *Print minimum and maximum of 3D free energy surface*

Description

‘print.fes3d‘ prints dimensionality, minimum and maximum of free energy in a fes object

Usage

```
## S3 method for class 'fes3d'
print(x, ...)
```

Arguments

x	fes3d object
...	further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
tfes
```

print.hillsfile3d *Print hillsfile3d*

Description

‘print.hillsfile3d‘ prints dimensionality and size of a hillsfile object.

Usage

```
## S3 method for class 'hillsfile3d'
print(x, ...)
```

Arguments

x	hillsfile3d object.
...	further arguments passed to or from other methods.

Examples

```
acealanme3d
```

print.minima3d *Print minima3d object*

Description

'print.minima3d' prints 3D free energy minima (identifier, values of bins and collective variables and free energy).

Usage

```
## S3 method for class 'minima3d'
print(x, ...)
```

Arguments

`x`	minima object.
`...`	further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
minima<-fesminima(tfes)
minima
```

read.hills3d *Read 3D HILLS from Plumed*

Description

'read.hills3d' reads a HILLS file generated by Plumed and returns a hillsfile3d object. User can specify whether some collective variables are periodic.

Usage

```
read.hills3d(file = "HILLS", per = c(FALSE, FALSE, FALSE),
  pcv1 = c(-pi, pi), pcv2 = c(-pi, pi), pcv3 = c(-pi, pi),
  ignoretime = FALSE)
```

Arguments

`file`	HILLS file from Plumed.
`per`	logical vector specifying periodicity of collective variables.
`pcv1`	periodicity of CV1.
`pcv2`	periodicity of CV2.
`pcv3`	periodicity of CV3.
`ignoretime`	time in the first column of the HILLS file will be ignored.

Value

hillsfile object.

Examples

```
l1<-"1 -1.587 -2.969  3.013 0.3 0.3 0.3 1.111 10"
l2<-"2 -1.067  2.745  2.944 0.3 0.3 0.3 1.109 10"
l3<-"3 -1.376  2.697  3.049 0.3 0.3 0.3 1.080 10"
l4<-"4 -1.663  2.922 -3.065 0.3 0.3 0.3 1.072 10"
fourhills<-c(l1,l2,l3,l4)
tf <- tempfile()
writeLines(fourhills, tf)
read.hills3d(tf, per=c(TRUE,TRUE))
```

read.plumed3d *Read 3D free energy surface from PLUMED sum_hills*

Description

‘read.plumed3d‘ reads 3D free energy surface from PLUMED sum_hills. The grid in the inputfile must contain the same number of points for CV1, CV2 and CV3. It does not use the header of the file. Periodicity must be specified.

Usage

```
read.plumed3d(file = "fes.dat", per = c(FALSE, FALSE, FALSE))
```

Arguments

`file`	input file from PLUMED sum_hills.
`per`	logical vector specifying periodicity of collective variables.

Value

fes3d object.

Examples

```
l1<-" -3.14 -3.14 -3.14 -61.13 -47.43  19.00   2.04"
l2<-" -1.05 -3.14 -3.14 -70.72  25.95  25.78   2.43"
l3<-"  1.05 -3.14 -3.14 -65.58   8.34   2.82  -3.09"
l4<-" -3.14 -1.05 -3.14 -51.31 -43.88 -19.91   1.51"
l5<-" -1.05 -1.05 -3.14 -66.43   7.67 -22.45  -0.39"
l6<-"  1.05 -1.05 -3.14 -61.08  -7.50  -7.36  -0.83"
l7<-" -3.14  1.05 -3.14 -53.07 -55.12   0.19  -0.28"
l8<-" -1.05  1.05 -3.14 -62.81  36.19   1.65   0.45"
l9<-"  1.05  1.05 -3.14 -65.28  22.84  11.47   0.59"
l10<-" -3.14 -3.14 -1.05 -13.03 -32.17   8.24 -35.25"
```

```
l11<-" -1.05 -3.14 -1.05 -21.88  17.89  21.91 -51.20"
l12<-"  1.05 -3.14 -1.05 -14.49   3.60   6.04 -44.05"
l13<-" -3.14 -1.05 -1.05  -2.26  -7.00  -7.01 -10.65"
l14<-" -1.05 -1.05 -1.05  -8.21   3.69 -22.89 -28.48"
l15<-"  1.05 -1.05 -1.05  -1.10   0.52   3.59  -1.99"
l16<-" -3.14  1.05 -1.05  -3.75 -11.70  -5.65 -15.36"
l17<-" -1.05  1.05 -1.05  -1.15   5.75   1.05  -2.42"
l18<-"  1.05  1.05 -1.05 -10.67   8.23 -10.42 -36.77"
l19<-" -3.14 -3.14  1.05  -4.64 -13.79  10.51  14.96"
l20<-" -1.05 -3.14  1.05  -7.80  12.24  20.59  23.03"
l21<-"  1.05 -3.14  1.05  -5.32   3.46   3.17  21.99"
l22<-" -3.14 -1.05  1.05  -2.06  -6.59   0.17  10.04"
l23<-" -1.05 -1.05  1.05  -9.69   8.43  -0.97  36.97"
l24<-"  1.05 -1.05  1.05  -0.19  -0.44  -0.26   0.91"
l25<-" -3.14  1.05  1.05  -7.98 -23.02   3.97  26.98"
l26<-" -1.05  1.05  1.05  -4.64  13.66  -9.74  10.15"
l27<-"  1.05  1.05  1.05 -13.42  15.78  16.36  41.60"
twentysevenpoints<-c(l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,
                     l11,l12,l13,l14,l15,l16,l17,l18,l19,l20,
                     l21,l22,l23,l24,l25,l26,l27)
tf <- tempfile()
writeLines(twentysevenpoints, tf)
read.plumed3d(tf, per=c(TRUE,TRUE,TRUE))
```

summary.fes3d *Print summary of 3D free energy surface*

Description

'summary.fes3d' prints minimum and maximum of free energy in a fes object.

Usage

```
## S3 method for class 'fes3d'
summary(object, ...)
```

Arguments

`object`	fes3d object.
`...`	further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
summary(tfes)
```

`summary.hillsfile3d` *Print summary for hillsfile3d*

Description

'summary.hillsfile3d' prints dimensionality, size and collective variable ranges of a hillsfile3d object.

Usage

```
## S3 method for class 'hillsfile3d'
summary(object, ...)
```

Arguments

`object`	hillsfile3d object.
`...`	further arguments passed to or from other methods.

Examples

```
summary(acealanme3d)
```

`summary.minima3d` *Print minima3d object summary*

Description

'summary.minima3d' prints summary for 3D free energy minima (identifier, values of bins and collective variables, free energy and equilibrium populations).

Usage

```
## S3 method for class 'minima3d'
summary(object, temp = 300, eunit = "kJ/mol", ...)
```

Arguments

`object`	minima3d object
`temp`	temperature in Kelvins
`eunit`	energy units (kJ/mol or kcal/mol, kJ/mol is default)
`...`	further arguments passed to or from other methods.

Examples

```
tfes<-fes(acealanme3d, imax=5000)
minima<-fesminima(tfes)
summary(minima)
```

`tail.hillsfile3d` *Print last n lines of hillsfile3d*

Description

‘tail.hillsfile3d‘ prints last n lines of a hillsfile3d object.

Usage

```
## S3 method for class 'hillsfile3d'
tail(x, n = 10, ...)
```

Arguments

`x`	hillsfile3d object.
`n`	number of lines (default 10).
`...`	further arguments passed to or from other methods.

Examples

```
tail(acealanme3d)
```

Index