Package 'penm'

September 29, 2022

Type Package

Title Build and Perturb Elastic Network Models

Version 0.2.0.9000

Author Julian Echave

Maintainer Julian Echave <jechave@unsam.edu.ar>

Description Functions to calculate ENM models, mutate ENMs by perturbing them, perform single-site mutional scans to calculate average mutation-response matrices, and perform double-site mutational scans to calculate compensation matrices.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Suggests testthat

Depends R (>= 2.10),

Imports matrixStats,
pracma,
bio3d,
dplyr,
jefuns,
magrittr,
Matrix,
purrr,
stats,
tibble,
tidyr

R topics documented:

penm-package 2
admrs 2
amrs 3
delta_energy 4
delta_motion_by_mode 5
delta_motion_by_site 5
delta_structure_by_mode 6

delta_structure_by_site 7
mrs_motion_by_mode 8
mrs_motion_by_site 8
mrs_structure_by_mode 9
mrs_structure_by_site 10
sdmrs 10
set_enm 12
smrs 12

Index **14**

`penm-package` *penm: Build and Perturb Elastic Network Models*

Description

Functions to calculate ENM models, mutate ENMs by perturbing them, perform single-site mutional scans to calculate average mutation-response matrices, and perform double-site mutational scans to calculate compensation matrices.

Details

The `penm` package includes functions to calculate various Elastic Network Models for proteins, perform normal mode analysis, and using lfenm, obtain mutant proteins and the corresponding mutant ENMs. In addition, it has functions to scan the various average-responses w.r.t. single-site mutations and double-site mutations.

`admrs` *Calculate a double-mutational-scan matrix analytically*

Description

Returns a compensation matrix: element (i,j) measures the degree of compensation of structural deformations produced by pairs of mutations at sites i and j. It uses analytical methods (closed formulas). Two measures are implemented: "mean_max" (default), the structural compensation maximized over mutations at j and averaged over mutations at i; "max_max" is the structural compensation maximized over mutations at i and j.

Usage

```
admrs(wt, mut_dl_sigma, mut_sd_min, option = "mean_max", response = "dr2")
```

Arguments

`wt`	is the (wild-type) protein to mutate (an object obtained using `set_enm`)
`mut_dl_sigma`	is the standard deviation of a normal distribution from which edge-length perturbations are picked (LFENM model).
`mut_sd_min`	is integer sequence-distance cutoff, only edges with `sdij` >= `mut_sd_min` are mutated
`option`	is either "mean_max" (default) or "max_max", depending on which compensation measure is desired.
`response`	is the response desired, which maybe either "dr2", "de2", or "df2"

Details

For details see doi:10.7717/peerj.11330

Value

A compensation matrix, rows are initially mutated site, j is compensation site

See Also

Other mutscan functions: `amrs()`, `sdmrs()`, `smrs()`

Examples

```
## Not run:
pdb <- bio3d::read.pdb("2acy")
wt <- set_enm(pdb, node = "ca", model = "ming_wall", d_max = 10.5, frustrated = FALSE)
dmat <- admrs(wt, mut_dl_sigma = 0.3, mut_sd_min = 1, option = "max_max", response = "dr2")

## End(Not run)
```

`amrs` *Calculate a mutation-response matrix analitically*

Description

Returns a mutation-response matrix It uses an analytical method (closed formulas). For details see doi:10.7717/peerj.11330

Usage

```
amrs(wt, mut_dl_sigma, mut_sd_min, option = "site", response = "dr2")
```

Arguments

`wt`	is the (wild-type) protein to mutate (an object obtained using `set_enm`)
`mut_dl_sigma`	is the standard deviation of a normal distribution from which edge-length perturbations are picked (LFENM model).
`mut_sd_min`	is integer sequence-distance cutoff, only edges with `sdij` >= `mut_sd_min` are mutated
`option`	is either "site" (default) or "mode"
`response`	is either "dr2" (default), "de2", or "df2"

Details

A site-by-site response matrix has elements Mij that measure the response (e.g. deformation) of site i averaged over mutations at site j. A mode-by-site response matrix has elements Mnj that measure the response (e.g. deformation) along mode n averaged over mutations at site j.

It may calculate either site-by-site or mode-by site response matrices Three type of response may be calculated, "dr2" (dr2ij and dr2nj), "de2" (de2ij and de2nj), and "df2" (df2ij and df2nj).

Value

A response matrix, columns are mutated sites, rows are responses, of a given site or normal mode.

See Also

Other mutscan functions: `admrs()`, `sdmrs()`, `smrs()`

Examples

```
## Not run:
pdb <- bio3d::read.pdb("2acy")
wt <- set_enm(pdb, node = "ca", model = "ming_wall", d_max = 10.5, frustrated = FALSE)
mrs_matrix <- amrs(wt,  mut_dl_sigma = 0.3, mut_sd_min = 1, option = "site", resonse = "dr2")

## End(Not run)
```

`delta_energy` *Calculate energy differences between a mutant and wild type*

Description

Calculate energy differences between a mutant and wild type

Usage

```
calculate_dvm(wt, mut)

calculate_dg_entropy(wt, mut, beta)

calculate_dvs(wt, mut, ideal = wt)
```

Arguments

`wt`	A protein object with `xyz` defined
`mut`	A second protein object with `xyz` defined

Details

'calculate_dvm' calculates the minimum-energy difference between `mut` and `wt`

'calculate_dg_enetropy' calculates the entropic free energ difference between `mut` and `wt`

'calculate_dvs' calculates the ideal-conformation stress-energy difference between `mut` and `wt`

Value

A (scalar) energy difference between mutant and wild type.

delta_motion_by_mode *Compare two protein ensembles mode by mode (mode-dependent profiles)*

Description

Given two proteins, compare their conformational ensembles (fluctuation patterns), the proteins' `cmat`, and normal modes (Principal components) are assumed known.

Usage

```
delta_motion_dmsfn(wt, mut)

delta_motion_dhn(wt, mut)

delta_motion_rwsipn(wt, mut)

delta_motion_nhn(wt, mut)
```

Arguments

`wt`	A protein object with `enm` defined
`mut`	A second protein object with `enm` defined

Details

(This version works only for wt and mut with no indels)

'delta_motion_dmsfn' returns mode-dependent profile of changes of mean-square fluctuations $\delta\sigma_n^2$

'delta_motion_dhn' returns mode-dependent profile of entropy differences δH_n

'delta_motion_rwsipn' returns mode-dependent profile of rwsip similarity

'delta_motion_nhn' returns mode-dependent profile of mode conservation measure nH_n

Value

A vector (`x_n`) of size `nmodes`, where `x_n` is the property compared, for mode `n`.

delta_motion_by_site *Compare the motions of two proteins site by site*

Description

Given two proteins, compare their conformational ensembles (fluctuation patterns), the proteins' `cmat`, and normal modes (Principal components) are assumed known.

Usage

```
delta_motion_dmsfi(wt, mut)

delta_motion_dbhati(wt, mut)

delta_motion_rwsipi(wt, mut)

delta_motion_dhi(wt, mut)
```

Arguments

wt	A protein object with enm defined
mut	A second protein object with enm defined

Details

(This version works only for wt and mut with no indels)

'delta_motion_dmsfi' returns site-dependent profile of changes of mean-square fluctuations $\delta\sigma_i^2$

'delta_motion_dbhati' returns site-dependent profile of dbhat distances

'delta_motion_rwsipi' returns site-dependent profile of rwsip similarity

'delta_motion_dhi' returns site-dependent profile of dh similarity

Value

A vector (x_i) of size nsites, where x_i is the property compared, for site i.

---

delta_structure_by_mode

Compare two protein structures in nm representation

Description

(This version works only for wt and mut with no indels)

Usage

```
delta_structure_dr2n(wt, mut)

delta_structure_de2n(wt, mut)

delta_structure_df2n(wt, mut)
```

Arguments

wt	A protein object with xyz and enm defined
mut	A second protein object with xyz defined

Details

'delta_structure_dr2n' calculates de square of the mode-contributions to $\delta\mathbf{r} = \mathbf{Cf}$

'delta_structure_de2n' calculates de square of the mode-contributions to $\delta\mathbf{e} = \mathbf{C}^{1/2}\mathbf{f}$

'delta_structure_df2n' calculates de square of the mode-contributions to the force vecgtor $\mathbf{f}$

Value

A vector with contributions of each normal mode to the given property

---

`delta_structure_by_site` *Calculate site-dependent profiles of structure differences between two proteins*

Description

This version works only for wt and mut with no indels

Usage

```
delta_structure_dr2i(wt, mut)

delta_structure_de2i(wt, mut, kmat_sqrt)

delta_structure_df2i(wt, mut)

delta_structure_dvmi(wt, mut)

delta_structure_dvsi(wt, mut)

delta_structure_dvsi_same_topology(wt, mut)
```

Arguments

`wt`	A protein object with `xyz` defined
`mut`	A second protein object with `xyz` defined

Details

'delta_structure_dr2i' returns the square of structural difference vector $\mathbf{Cf}$

'delta_structure_de2i' returns the square of deformation energy vector $\mathbf{C}^{1/2}\mathbf{f}$

'delta_structure_df2i' returns the square of force vector $\mathbf{f}$

'delta_structure_dvmi' returns the difference of site-dependent minimum-energy contributions

'delta_structure_dvsi' returns the difference of site-dependent stress-energy contributions

'delta_structure_dvsi_same_topology' returns the difference of site-dependent stress-energy contributions, assumes no change in topology

Value

A vector (`x_i`) of size `nsites`, where `x_i` is the property compared, for site i.

mrs_motion_by_mode *Calculate mode-by-site motion-response matrices*

Description

Calculate mode-by-site motion-response matrices

Usage

```
mrs_motion_dmsfnj(mutants)

mrs_motion_dhnj(mutants)

mrs_motion_nhnj(mutants)

mrs_motion_rwsipnj(mutants)
```

Arguments

mutants	A tibble of single-point mutants generated using ‘generate_mutants‘

Details

‘mrs_motion_dmsfnj()‘ calculates the change of msf along mode n averaged over mutations at site j.

‘mrs_motion_dhnj()‘ calculates change of entropy contribution of mode n averaged over mutations at site j.

‘mrs_motion_nhnj()‘ calculates conservation score nh for mode n averaged over mutations at site j.

‘mrs_motion_rwsipnj()‘ calculates rwsip for mode n averaged over mutations at j

Value

a response matrix of the form R_{nj} (response mode is n, mutated site is j)

mrs_motion_by_site *Calculate site-dependent motion-response matrices*

Description

Calculate site-dependent motion-response matrices

Usage

```
mrs_motion_dmsfij(mutants)

mrs_motion_dhij(mutants)

mrs_motion_rwsipij(mutants)

mrs_motion_dbhatij(mutants)
```

Arguments

mutants	A tibble of single-point mutants generated using ‘generate_mutants‘

Details

‘mrs_motion_dmsfij()‘ calculates the change of msf of site i due to mutations at j, averaged over mutations at j

‘mrs_motion_dhij()‘ calculates the change in entropy of site i due to mutations at j averaged over mutations at j

‘mrs_motion_rwsipij()‘ calculates rwsip between mutant and wt site i distributions due to mutations at j, averaged over mutations at j

‘mrs_motion_dbhatij()‘ calculates dbhat distance between wt and mut site i distributions, averaged over mutations at j.

Value

a response matrix of the form R_{ij} (response site is i, mutated site is j)

---

mrs_structure_by_mode *Calculate mode-by-site structure-response matrices*

Description

Calculate mode-by-site structure-response matrices

Usage

```
mrs_structure_df2nj(mutants)

mrs_structure_de2nj(mutants)

mrs_structure_dr2nj(mutants)
```

Arguments

mutants	A tibble of single-point mutants generated using ‘generate_mutants‘

Details

‘mrs_structure_df2nj()‘ calculates the force matrix f2(n, j) averaged over mutations at j

‘mrs_structure_de2nj()‘ calculates the energy matrix de2(n, j) averaged over mutations at j

‘mrs_structure_dr2nj()‘ calculates the structural-difference matrix dr2(n, j) averaged over mutations at j

Value

a response matrix of the form R_{nj} (response mode is n, mutated site is j)

mrs_structure_by_site *Calculate site-by-site structure-response matrices*

Description

Calculate site-by-site structure-response matrices

Usage

```
mrs_structure_df2ij(mutants)

mrs_structure_de2ij(mutants)

mrs_structure_dr2ij(mutants)

mrs_structure_dvsij(mutants)
```

Arguments

mutants	A tibble of single-point mutants generated using ‘generate_mutants‘

Details

‘mrs_structure_df2ij()‘ calculates the force matrix `df2(i, j)` averaged over mutations at `j`

‘mrs_structure_de2ij()‘ calculates the energy-difference matrix `de2(i, j)` averaged over mutations at `j`

‘mrs_structure_dr2ij()‘ calculates the structural difference matrix `dr2(i, j)` averaged over mutations at `j`

‘mrs_structure_dvsij()‘ calculates the stress-energy matrix `dvs(i, j)` averaged over mutations at `j`

Value

a response matrix of the form R_{ij} (response site is `i`, mutated site is `j`)

sdmrs *Calculate a double-mutational-scan matrix numerically (simulation-based)*

Description

Returns a compensation matrix: element (i,j) measures the degree of compensation of structural deformations produced by pairs of mutations at sites i and j. It uses a simulation method (calculates responses for various instances of forces, then calculates means or maxima) Two measures are implemented: "mean_max" (default), the structural compensation maximized over mutations at j and averaged over mutations at i; "max_max" is the structural compensation maximized over mutations at i and j.

Usage

```
sdmrs(
  wt,
  nmut,
  mut_dl_sigma,
  mut_sd_min,
  option = "mean_max",
  response = "dr2",
  seed = 1024
)
```

Arguments

wt	is the (wild-type) protein to mutate (an object obtained using set_enm)
nmut	is the number of mutations per site to simulate
mut_dl_sigma	is the standard deviation of a normal distribution from which edge-length perturbations are picked (LFENM model).
mut_sd_min	is integer sequence-distance cutoff, only edges with sdij >= mut_sd_min are mutated
option	is either "mean_max" (default) or "max_max", depending on which compensation measure is desired.
response	is the response desired, which maybe either "dr2", "de2", or "df2"
seed	seed for random generation of mutations

Details

For details see doi:10.7717/peerj.11330

Value

A compensation matrix, rows are initially mutated site, j is compensation site

See Also

Other mutscan functions: admrs(), amrs(), smrs()

Examples

```
## Not run:
pdb <- bio3d::read.pdb("2acy")
wt <- set_enm(pdb, node = "ca", model = "ming_wall", d_max = 10.5, frustrated = FALSE)
dmat <- sdmrs(wt, nmut = 10, mut_dl_sigma = 0.3, mut_sd_min = 1, option = "max_max", response = "dr2")

## End(Not run)
```

`set_enm` *Set up 'prot' object*

Description

'set_enm' set's up a 'prot' object containing information on ENM structure, parameters, and normal modes

Usage

```
set_enm(pdb, node, model, d_max, frustrated)
```

Arguments

`pdb`	pdb object obtained using bio3d::read.pdb
`node`	parameter specifying how network nodes should be built: "sc" (side chains) or "ca" (alpha carbons)
`model`	parameter specifying model type: "anm", "ming_wall", "hnm", "hnm0", "pfanm", "reach"
`d_max`	distance cutoff used to define enm contacts
`frustrated`	logical value indicating whether to include frustrations in calculation of kmat

Value

an object of class 'prot', which is a list 'lst(param, node, graph, eij, kmat, nma)'

Examples

```
## Not run:
pdb <- bio3d::read.pdb("2acy")
set_enm(pdb, node = "ca", model = "ming_wall", d_max = 10.5, frustrated = FALSE)
set_enm(pdb, node = "sc", model = "anm", d_max = 12.5, frustrated = TRUE)

## End(Not run)
```

`smrs` *Calculate a mutation-response matrix numerically (simulation-based)*

Description

Returns a mutation-response matrix It uses a simulation method (averages over perturbations). For details see doi:10.7717/peerj.11330

Usage

```
smrs(
  wt,
  nmut,
  mut_dl_sigma,
  mut_sd_min,
  option = "site",
  response = "dr2",
  seed = 1024
)
```

Arguments

wt	is the (wild-type) protein to mutate (an object obtained using set_enm)
nmut	is the number of mutations per site to simulate
mut_dl_sigma	is the standard deviation of a normal distribution from which edge-length perturbations are picked (LFENM model).
mut_sd_min	is integer sequence-distance cutoff, only edges with sdij >= mut_sd_min are mutated
option	is either "site" (default) or "mode"
response	is either "dr2" (default), "de2", or "df2"
seed	is a seed for random-number generation of mutations

Details

A site-by-site response matrix has elements Mij that measure the response (e.g. deformation) of site i averaged over mutations at site j. A mode-by-site response matrix has elements Mnj that measure the response (e.g. deformation) along mode n averaged over mutations at site j.

It may calculate either site-by-site or mode-by site response matrices Three type of response may be calculated, "dr2" (dr2ij and dr2nj), "de2" (de2ij and de2nj), and "df2" (df2ij and df2nj).

Value

A response matrix, columns are mutated sites, rows are responses, of a given site or normal mode.

See Also

Other mutscan functions: admrs(), amrs(), sdmrs()

Examples

```
## Not run:
pdb <- bio3d::read.pdb("2acy")
wt <- set_enm(pdb, node = "ca", model = "ming_wall", d_max = 10.5, frustrated = FALSE)
mrs_matrix <- smrs(wt, nmut = 10, mut_model = "lfenm", mut_dl_sigma = 0.3, mut_sd_min = 1, seed = 1024)

## End(Not run)
```

Index

∗ **mutscan functions**
admrs, 2
amrs, 3
sdmrs, 10
smrs, 12

admrs, 2, *4*, *11*, *13*
amrs, *3*, 3, *11*, *13*

calculate_dg_entropy *(*delta_energy*)*, 4
calculate_dvm *(*delta_energy*)*, 4
calculate_dvs *(*delta_energy*)*, 4

delta_energy, 4
delta_motion_by_mode, 5
delta_motion_by_site, 5
delta_motion_dbhati *(*delta_motion_by_site*)*, 5
delta_motion_dhi *(*delta_motion_by_site*)*, 5
delta_motion_dhn *(*delta_motion_by_mode*)*, 5
delta_motion_dmsfi *(*delta_motion_by_site*)*, 5
delta_motion_dmsfn *(*delta_motion_by_mode*)*, 5
delta_motion_nhn *(*delta_motion_by_mode*)*, 5
delta_motion_rwsipi *(*delta_motion_by_site*)*, 5
delta_motion_rwsipn *(*delta_motion_by_mode*)*, 5
delta_structure_by_mode, 6
delta_structure_by_site, 7
delta_structure_de2i *(*delta_structure_by_site*)*, 7
delta_structure_de2n *(*delta_structure_by_mode*)*, 6
delta_structure_df2i *(*delta_structure_by_site*)*, 7
delta_structure_df2n *(*delta_structure_by_mode*)*, 6
delta_structure_dr2i *(*delta_structure_by_site*)*, 7
delta_structure_dr2n *(*delta_structure_by_mode*)*, 6
delta_structure_dvmi *(*delta_structure_by_site*)*, 7
delta_structure_dvsi *(*delta_structure_by_site*)*, 7
delta_structure_dvsi_same_topology *(*delta_structure_by_site*)*, 7

mrs_motion_by_mode, 8
mrs_motion_by_site, 8
mrs_motion_dbhatij *(*mrs_motion_by_site*)*, 8
mrs_motion_dhij *(*mrs_motion_by_site*)*, 8
mrs_motion_dhnj *(*mrs_motion_by_mode*)*, 8
mrs_motion_dmsfij *(*mrs_motion_by_site*)*, 8
mrs_motion_dmsfnj *(*mrs_motion_by_mode*)*, 8
mrs_motion_nhnj *(*mrs_motion_by_mode*)*, 8
mrs_motion_rwsipij *(*mrs_motion_by_site*)*, 8
mrs_motion_rwsipnj *(*mrs_motion_by_mode*)*, 8
mrs_structure_by_mode, 9
mrs_structure_by_site, 10
mrs_structure_de2ij *(*mrs_structure_by_site*)*, 10
mrs_structure_de2nj *(*mrs_structure_by_mode*)*, 9
mrs_structure_df2ij *(*mrs_structure_by_site*)*, 10
mrs_structure_df2nj *(*mrs_structure_by_mode*)*, 9
mrs_structure_dr2ij *(*mrs_structure_by_site*)*, 10
mrs_structure_dr2nj *(*mrs_structure_by_mode*)*, 9
mrs_structure_dvsij *(*mrs_structure_by_site*)*, 10

penm *(*penm-package*)*, 2
penm-package, 2

sdmrs, *3*, *4*, 10, *13*

set_enm, 12
smrs, *3*, *4*, *11*, 12