

Package: LongOmicsTraj (via r-universe)

October 3, 2026

Type Package

Title Longitudinal Omics Trajectory Clustering and Analysis Engine

Version 0.99.0

Description Provides methods for estimating, classifying and comparing longitudinal molecular trajectories across repeated study visits using empirical, mixed-effects, generalized additive model and polynomial regression estimators. Consecutive molecular changes are converted into interpretable Omics Trajectory Signatures (OTS) describing stable, sustained, transient and rebound temporal behaviours, enabling topology-aware clustering and consensus trajectory inference. LongOmicsTraj further provides a flexible query framework for directly retrieving molecular features satisfying simple or complex trajectory definitions, including multi-transition patterns, logical combinations of trajectory states, group-specific behaviours, and comparisons of trajectory topologies across treatment arms, experimental conditions or other longitudinal study arms.

License GPL-3

Encoding UTF-8

Depends R (>= 4.5.0), MultiAssayExperiment

Imports methods, data.table, S4Vectors, SummarizedExperiment, BiocParallel, ggplot2, dplyr, tibble, tidyr, rlang, scales, withr, stats4, lme4, emmeans, mgcv

Suggests testthat (>= 3.0.0), matrixStats, cluster, knitr, rmarkdown, flexmix, patchwork, BiocStyle, maSigPro

VignetteBuilder knitr

biocViews Software, TimeCourse, Transcriptomics, Clustering, SystemsBiology

BugReports <https://github.com/nzkermani/LongOmicsTraj/issues>

URL <https://github.com/nzkermani/LongOmicsTraj>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libicu-dev zlib1g-dev

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

Date/Publication 2026-08-04 00:18:46 UTC

RemoteUrl <https://github.com/BiocStaging/LongOmicsTraj>

RemoteRef HEAD

RemoteSha c92f35a6d48426c6164fd02a12b89bec01bddeec

Contents

compute_vmm_deltas	3
LongOmicsTraj-class	4
lot_cluster_purity	4
lot_cluster_topology_score	6
lot_compare_estimators	7
lot_compute_deltas	9
lot_compute_thresholds	10
lot_consensus	12
lot_create	13
lot_from_clusters	16
lot_from_empirical	17
lot_from_flexmix_clusters	25
lot_glucold	28
lot_glucold_clinical	28
lot_glucold_expression	29
lot_glucold_ots	29
lot_mae	30
lot_map_to_ots	32
lot_plot_cluster_selection	33
lot_plot_cluster_topology	35
lot_plot_cluster_trajectory_arms	39
lot_plot_ots_distribution	41
lot_plot_topology_deviation	42
lot_plot_trajectory	44
lot_query	45
lot_summary	46
plot_cluster_spaghetti	47
plot_cluster_spaghetti_single	48
print.lot_summary	48
show,LongOmicsTraj-method	49

Index

50

compute_vmm_deltas	<i>Compute adjacent visit-to-visit deltas from a Visit Mean Matrix</i>
--------------------	--

Description

Compute adjacent visit-to-visit deltas from a Visit Mean Matrix

Usage

```
compute_vmm_deltas(vmm, visits)
```

Arguments

vmm	A data.frame with columns object_id, object_type, group, visit, estimated_value, estimator.
visits	Character vector giving the ordered visits.

Value

A data.frame of adjacent deltas.

Examples

```
visits <- c(
  "Baseline",
  "Week 4",
  "Week 12"
)

vmm <- data.frame(
  object_id = rep(
    c("gene_1", "gene_2"),
    each = 3
  ),
  object_type = "gene",
  group = "treated",
  visit = rep(
    visits,
    times = 2
  ),
  estimated_value = c(
    1.0, 1.6, 1.2,
    2.0, 2.0, 2.5
  ),
  estimator = "empirical",
  stringsAsFactors = FALSE
)

deltas <- compute_vmm_deltas(
```

```
    vmm = vmm,
    visits = visits
  )

  deltas[
    ,
    c(
      "object_id",
      "transition",
      "delta_hat"
    )
  ]
}
```

LongOmicsTraj-class	<i>The LongOmicsTraj Class</i>
---------------------	--------------------------------

Description

S4 container for supervised and unsupervised longitudinal omics topology analysis.

Slots

- mae MultiAssayExperiment object storing raw/normalised multi-omics data.
- vmm Visit Mean Matrix as S4Vectors::DataFrame.
- deltas Adjacent visit-to-visit changes.
- thresholds Feature/object-specific topology thresholds.
- ots Overall Trajectory State labels.
- relationships Child-parent relationships, e.g. gene-to-cluster.
- results List of query, enrichment, diagnostic, or model outputs.
- cache List for reusable intermediate objects.
- settings List of analysis settings.
- history List recording processing steps.

lot_cluster_purity	<i>Calculate topology purity within clusters</i>
--------------------	--

Description

Calculates the dominant Omics Trajectory Signature, topology purity and entropy within each cluster and clinical group.

Usage

```
lot_cluster_purity(assignments, purity_threshold = 0.9)
```

Arguments

assignments A data frame containing feature-to-cluster assignments and feature-level OTS labels. Required columns are 'gene_id', 'cluster', 'group' and 'topology_label'.

purity_threshold Numeric value between 0 and 1 used to classify a cluster-group combination as sufficiently topology-homogeneous.

Details

Topology purity is defined as the proportion of features belonging to the most frequent OTS class within a cluster-group combination.

Value

An object of class 'lot_cluster_purity' containing:

cluster_group_purity Purity and entropy statistics for each cluster-group combination.

overall Overall minimum, mean and weighted purity, passing fraction and maximum normalised entropy.

Examples

```
assignments <- data.frame(
  gene_id = paste0("G", 1:10),
  cluster = c(rep(1, 5), rep(2, 5)),
  group = "Withdrawal",
  topology_label = c(
    rep("up_down", 4),
    "flat_flat",
    rep("down_up", 5)
  )
)

result <- lot_cluster_purity(
  assignments,
  purity_threshold = 0.90
)

result$cluster_group_purity
result$overall
```

lot_cluster_topology_score

Calculate a topology-aware cluster score

Description

Summarises topology purity, dominant-topology diversity and duplication for a fitted clustering solution.

Usage

```
lot_cluster_topology_score(
  purity,
  k = NULL,
  bic = NA_real_,
  aic = NA_real_,
  minimum_cluster_size = NA_integer_
)
```

Arguments

purity	A 'lot_cluster_purity' object returned by [lot_cluster_purity()], or a data frame containing cluster-level purity results.
k	Optional requested number of clusters.
bic	Optional BIC value for the fitted clustering solution.
aic	Optional AIC value for the fitted clustering solution.
minimum_cluster_size	Optional size of the smallest fitted cluster.

Details

This function complements likelihood-based criteria such as BIC by asking whether the fitted clusters represent distinct longitudinal trajectory behaviours. A solution with many clusters but few unique dominant topologies has low topology coverage and may reflect over-partitioning.

For each clinical group, topology coverage is defined as:

$$\text{topology coverage} = \frac{\text{number of unique dominant topologies}}{\text{number of observed clusters}}$$

A value of 1 indicates that every cluster has a different dominant topology. Lower values indicate that multiple clusters share the same dominant trajectory behaviour.

The duplicate topology count is:

$$\text{number of clusters} - \text{number of unique dominant topologies}$$

The function reports purity and coverage separately. It does not yet combine BIC and topology metrics into a single weighted score.

Value

An object of class 'lot_cluster_topology_score' containing:

group_summary Topology diversity and duplication statistics for each clinical group.

overall Overall topology-aware summary of the clustering solution.

Examples

```
assignments <- data.frame(
  gene_id = paste0("G", 1:12),
  cluster = c(
    rep(1, 4),
    rep(2, 4),
    rep(3, 4)
  ),
  group = "Withdrawal",
  topology_label = c(
    rep("flat_flat", 4),
    rep("flat_flat", 3),
    "flat_up",
    rep("down_up", 4)
  )
)

purity <- lot_cluster_purity(
  assignments,
  purity_threshold = 0.90
)

score <- lot_cluster_topology_score(
  purity,
  k = 3,
  bic = 125.4,
  minimum_cluster_size = 4
)

score$group_summary
score$overall
```

lot_compare_estimators

Compare trajectory estimators

Description

Combines Omics Trajectory Signature assignments generated by two or more trajectory estimators into a common comparison table.

Combines Omics Trajectory Signature assignments from two or more 'LongOmicsTraj' objects into a common table suitable for agreement and consensus analyses.

Usage

```
lot_compare_estimators(object, ..., .names = NULL)
```

```
## S4 method for signature 'LongOmicsTraj'
```

```
lot_compare_estimators(object, ..., .names = NULL)
```

Arguments

<code>object</code>	A ‘LongOmicsTraj’ object containing OTS assignments.
<code>...</code>	Additional ‘LongOmicsTraj’ objects containing OTS assignments.
<code>.names</code>	Optional character vector giving estimator labels. When omitted, labels of the form “estimator_1”, “estimator_2” and so on are used.

Details

Each supplied object must contain OTS assignments generated by ‘lot_map_to_ots()’. The original estimator column is replaced by the labels supplied through ‘.names’, allowing results from different modelling strategies to be compared directly.

Value

An ‘S4Vectors::DataFrame’ containing the combined OTS assignments.

An ‘S4Vectors::DataFrame’ containing the combined OTS assignments and an ‘estimator’ column identifying their source.

Examples

```
data("lot_glucold_ots")

## For illustration, compare two stored trajectory analyses.
comparison <- lot_compare_estimators(
  lot_glucold_ots,
  lot_glucold_ots,
  .names = c(
    "analysis_1",
    "analysis_2"
  )
)

comparison <- as.data.frame(comparison)

table(
  comparison$estimator
)

utils::head(
  comparison[
    ,
    c(
      "object_id",
```

```

        "group",
        "topology_label",
        "estimator"
    )
]
)
```

lot_compute_deltas *Compute between-visit molecular changes*

Description

Computes feature-level changes between consecutive visits from stored visit-level molecular trajectories.

Calculates feature-level changes between consecutive visits from the visit-level molecular trajectories stored in a 'LongOmicsTraj' object.

Usage

```
lot_compute_deltas(object, ...)

## S4 method for signature 'LongOmicsTraj'
lot_compute_deltas(object, assay, visits, overwrite = FALSE)
```

Arguments

object	A 'LongOmicsTraj' object.
...	Additional arguments passed to the corresponding method.
assay	Name of the assay to process.
visits	Character vector giving the visits in chronological order.
overwrite	Logical; overwrite existing delta results when 'TRUE'.

Value

A 'LongOmicsTraj' object containing updated molecular-change results.

A 'LongOmicsTraj' object with updated delta results.

Examples

```
data("lot_glucold_ots")

visits <- c(
  "Baseline",
  "6 Months",
  "30 Months"
)
```

```
fitted <- lot_compute_deltas(
  object = lot_glucoold_ots,
  assay = "transcriptomics",
  visits = visits,
  overwrite = TRUE
)

deltas <- as.data.frame(
  lot_deltas(fitted)
)

utils::head(
  deltas[
    ,
    c(
      "object_id",
      "group",
      "transition",
      "delta_hat"
    )
  ]
)
```

lot_compute_thresholds

Compute molecular-change thresholds

Description

Computes thresholds used to classify molecular changes between consecutive visits.

Computes feature-specific thresholds used to classify molecular changes between consecutive visits as down, flat or up.

Usage

```
lot_compute_thresholds(object, ...)

## S4 method for signature 'LongOmicsTraj'
lot_compute_thresholds(
  object,
  assay,
  method = c("fixed", "global_mad", "gene_mad", "hybrid", "quantile"),
  delta_min = NULL,
  k = 0.25,
  prob = 0.75,
  visits,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

object	A 'LongOmicsTraj' object.
...	Additional arguments passed to the corresponding method.
assay	Name of the assay for which thresholds should be computed.
method	Threshold-estimation method. One of "fixed", "global_mad", "gene_mad", "hybrid" or "quantile".
delta_min	Optional minimum threshold value. When 'NULL', the median absolute delta across all objects and transitions is used.
k	Numeric scaling factor applied to MAD-based thresholds.
prob	Probability used for quantile-based threshold estimation.
visits	Character vector giving the visits in chronological order. When omitted, 'object@settings\$visits' is used.
overwrite	Logical; overwrite existing thresholds for the selected assay when 'TRUE'.
verbose	Logical; print progress messages.

Details

Thresholds are calculated from the delta estimates already stored in the 'LongOmicsTraj' object. The function supports fixed, global MAD, feature-specific MAD, hybrid and quantile-based thresholding.

Delta values are grouped by molecular object and converted into a matrix of adjacent-visit changes. The available thresholding methods are:

'fixed' Uses the same 'delta_min' threshold for every object.

'global_mad' Uses a single threshold based on the global median MAD.

'gene_mad' Uses an object-specific MAD threshold.

'hybrid' Uses the maximum of 'delta_min' and the object-specific MAD threshold.

'quantile' Uses the maximum of 'delta_min' and an object-specific absolute-delta quantile.

The function supports both gene-level and cluster-level delta results. Object type is inferred from identifiers beginning with "cluster_".

Value

A 'LongOmicsTraj' object containing updated threshold results.

A 'LongOmicsTraj' object with updated threshold results.

Examples

```
data("lot_glucold_ots")

visits <- c(
  "Baseline",
  "6 Months",
  "30 Months"
```

```

)

fitted <- lot_compute_thresholds(
  object = lot_glucold_ots,
  assay = "transcriptomics",
  method = "hybrid",
  visits = visits,
  overwrite = TRUE
)

thresholds <- as.data.frame(
  lot_thresholds(fitted)
)

utils::head(
  thresholds[
    ,
    c(
      "object_id",
      "delta_g",
      "threshold_method"
    )
  ]
)

```

lot_consensus

*Calculate consensus trajectory assignments***Description**

Combines Omics Trajectory Signature assignments from multiple estimators into a consensus topology label.

Combines OTS assignments stored in a 'DFrame' and returns a consensus topology label for each molecular feature.

Usage

```
lot_consensus(x, min_agreement = 2L, mode = c("strict", "majority"))
```

```
## S4 method for signature 'DFrame'
```

```
lot_consensus(x, min_agreement = 2L, mode = c("strict", "majority"))
```

Arguments

x	A 'DFrame' containing OTS assignments from multiple estimators.
min_agreement	Minimum number of assignments that must agree before a consensus topology is assigned.
mode	Consensus rule. "majority" assigns the most frequent topology when its count is at least 'min_agreement'. "strict" assigns a topology only when all available assignments agree and the number of assignments is at least 'min_agreement'.

Details

Multiple estimator and group-specific assignments are first collapsed to one topology label per molecular feature, estimator and group. Consensus is then calculated across the resulting assignments.

Value

An object containing consensus topology assignments.

A ‘DFrame’ containing one row per molecular feature with the consensus topology, agreement count and number of available assignments.

Examples

```
assignments <- S4Vectors::DataFrame(
  object_id = rep(
    c("gene_1", "gene_2"),
    each = 3
  ),
  estimator = rep(
    c("empirical", "lmm", "gam"),
    times = 2
  ),
  group = "treated",
  topology_label = c(
    "up_down",
    "up_down",
    "flat_flat",
    "down_up",
    "down_up",
    "down_up"
  )
)

consensus <- lot_consensus(
  assignments,
  min_agreement = 2L,
  mode = "majority"
)

as.data.frame(consensus)
```

lot_create

*Create a LongOmicsTraj object***Description**

Creates a ‘LongOmicsTraj’ object from a numeric molecular-data matrix and sample-level clinical metadata.

Usage

```
lot_create(
  expression,
  metadata,
  assay = "transcriptomics",
  sample_col = "sample_id",
  subject_col = "subject_id",
  visit_col = "visit",
  group_col = "group",
  visit_levels = NULL,
  time_values = NULL,
  time_col = "time",
  feature_type = "feature"
)
```

Arguments

expression	A numeric matrix or matrix-like object. Rows represent molecular features and columns represent samples.
metadata	A data frame containing one row per sample.
assay	Character string giving the assay name. The default is "transcriptomics".
sample_col	Character string naming the sample identifier column in 'metadata'. The default is "sample_id".
subject_col	Character string naming the participant identifier column in 'metadata'. The default is "subject_id".
visit_col	Character string naming the visit column in 'metadata'. The default is "visit".
group_col	Character string naming the clinical or treatment-group column in 'metadata'. The default is "group".
visit_levels	Optional character vector giving the chronological order of visits. When supplied, the visit column is converted to an ordered factor.
time_values	Optional named numeric vector mapping visit labels to numerical time. For example, 'c("Baseline" = 0, "6 Months" = 6, "30 Months" = 30)'.
time_col	Character string used for the numerical time column created from 'time_values'. The default is "time".
feature_type	Character string describing the molecular features. The default is "feature".

Details

The input matrix should contain molecular features in rows and samples in columns. The metadata should contain one row per sample. Sample IDs are checked, aligned and stored in a 'Summarized-Experiment', which is then placed inside a 'MultiAssayExperiment'.

'lot_create()' performs the following steps:

1. Checks that the expression data are numeric.
2. Checks that feature and sample identifiers are available.

3. Checks the required metadata columns.
4. Aligns metadata rows to expression-matrix columns.
5. Optionally defines visit order and numerical time.
6. Creates a 'SummarizedExperiment'.
7. Creates a 'MultiAssayExperiment'.
8. Returns a valid 'LongOmicsTraj' object.

Value

A valid 'LongOmicsTraj' object.

Examples

```
data("lot_glucold_expression")
data("lot_glucold_clinical")

visits <- c(
  "Baseline",
  "6 Months",
  "30 Months"
)

time_values <- c(
  "Baseline" = 0,
  "6 Months" = 6,
  "30 Months" = 30
)

lot <- lot_create(
  expression = lot_glucold_expression,
  metadata = lot_glucold_clinical,
  assay = "transcriptomics",
  visit_levels = visits,
  time_values = time_values
)

methods::is(
  lot,
  "LongOmicsTraj"
)

names(
  MultiAssayExperiment::experiments(
    lot_mae(lot)
  )
)
```

lot_from_clusters	<i>Cluster molecular features by longitudinal trajectory</i>
-------------------	--

Description

Groups molecular features according to similarities in their longitudinal profiles.

Clusters molecular features using their combined group-by-visit longitudinal profiles. Cluster centres are stored as visit-level molecular models, while feature-to-cluster memberships and clustering diagnostics are stored in the results and relationships slots.

Usage

```
lot_from_clusters(object, ...)

## S4 method for signature 'LongOmicsTraj'
lot_from_clusters(
  object,
  assay,
  group_col = "group",
  visit_col = "visit",
  visits,
  k = 4L,
  estimator = c("kmeans", "pam"),
  assay_data = NULL,
  scale_features = TRUE,
  standardise_ids = FALSE,
  seed = 1L,
  require_gene_ots = FALSE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

object	A 'LongOmicsTraj' object.
...	Additional clustering arguments passed to the method.
assay	Name of the assay to analyse.
group_col	Metadata column defining clinical or treatment groups.
visit_col	Metadata column defining visits.
visits	Character vector giving the chronological visit order.
k	Number of clusters.
estimator	Clustering method: "kmeans" or "pam".
assay_data	Optional assay-matrix name. The first available matrix is used by default.

scale_features	Logical; standardise each feature trajectory before clustering. The default is 'TRUE'.
standardise_ids	Logical; convert feature identifiers to uppercase before clustering and matching OTS results. The default is 'FALSE'.
seed	Random seed used by k-means.
require_gene_ots	Logical; require existing feature-level OTS results.
overwrite	Logical; overwrite an existing visit-level model.
verbose	Logical; print progress messages.

Details

When feature-level OTS results already exist, they are preserved and summarised within each trajectory cluster.

Each molecular feature is represented by a vector containing its mean value for every observed group-by-visit combination. Clustering is therefore based on the feature's combined longitudinal behaviour across all included groups.

When 'scale_features = TRUE', clustering and cluster-level OTS mapping use standardised feature profiles. Both raw and scaled cluster centres are retained in 'object@results\$cluster_profiles'.

Value

A 'LongOmicsTraj' object containing cluster-level trajectories and feature-to-cluster relationships.

A 'LongOmicsTraj' object containing cluster-centre trajectories, feature-to-cluster relationships, raw and scaled cluster profiles, clustering diagnostics and, when available, feature-level OTS distributions within clusters.

lot_from_empirical	<i>Estimate trajectories using a maSigPro-style polynomial model</i>
--------------------	--

Description

Fits polynomial regression models to longitudinal molecular features and stores predicted group-by-visit trajectories in the visit mean matrix.

Estimate feature-level longitudinal molecular trajectories from repeated measurements stored in a 'LongOmicsTraj' object.

Usage

```
lot_from_empirical(object, ...)
```

```
lot_from_lmm(object, ...)
```

```
lot_from_gam(object, ...)
```

```
lot_from_masigpro(object, ...)  
  
## S4 method for signature 'LongOmicsTraj'  
lot_from_empirical(  
  object,  
  assay,  
  visits,  
  group_col = "group",  
  visit_col = "visit",  
  estimator = "empirical",  
  overwrite = FALSE,  
  verbose = TRUE  
)  
  
## S4 method for signature 'LongOmicsTraj'  
lot_from_gam(  
  object,  
  assay,  
  formula_fixed,  
  group_col = "group",  
  visit_col = "visit",  
  subject_col = "subject_id",  
  visits,  
  estimator = "gam",  
  overwrite = FALSE,  
  verbose = TRUE  
)  
  
## S4 method for signature 'LongOmicsTraj'  
lot_from_lmm(  
  object,  
  assay,  
  formula_fixed,  
  subject_col = "subject_id",  
  group_col = "group",  
  visit_col = "visit",  
  visits,  
  estimator = "lmm",  
  BPPARAM = NULL,  
  overwrite = FALSE,  
  verbose = TRUE  
)  
  
## S4 method for signature 'LongOmicsTraj'  
lot_from_masigpro(  
  object,  
  assay,
```

```

    group_col = "group",
    visit_col = "visit",
    time_col = "time",
    subject_col = "subject_id",
    visits,
    degree = NULL,
    estimator = "masigpro",
    overwrite = FALSE,
    verbose = TRUE
  )

```

Arguments

object	A ‘LongOmicsTraj’ object.
...	Additional method-specific arguments passed to the selected trajectory-estimation method.
assay	Name of the assay or experiment to analyse.
visits	Character vector giving the chronological visit order.
group_col	Metadata column containing clinical or treatment groups.
visit_col	Metadata column containing visit labels.
estimator	Character label recorded for the fitted estimator.
overwrite	Logical; overwrite existing visit-level trajectory results.
verbose	Logical; print progress messages.
formula_fixed	Fixed-effects model formula used by the linear mixed-effects estimator.
subject_col	Metadata column containing subject identifiers.
BPPARAM	A ‘BiocParallelParam’ object controlling parallel execution.
time_col	Metadata column containing numeric time values.
degree	Polynomial degree used by the regression-based trajectory estimator.

Details

The available methods include empirical visit-level summaries, linear mixed-effects models, generalised additive models and maSigPro-style polynomial regression.

Value

A ‘LongOmicsTraj’ object with updated visit mean trajectories.

A ‘LongOmicsTraj’ object with updated visit-level molecular model results stored in the ‘vmm’ component.

Examples

```

data("lot_glucold")

visits <- c(
  "Baseline",
  "6 Months",
  "30 Months"
)

## Work on a copy of the example object
lot_empirical <- lot_glucold

## Extract the transcriptomics experiment
experiments <- MultiAssayExperiment::experiments(
  lot_empirical@mae
)

se <- experiments[["transcriptomics"]]

metadata <- as.data.frame(
  SummarizedExperiment::colData(se)
)

## Define the chronological visit order
metadata$visit <- factor(
  metadata$visit,
  levels = visits,
  ordered = TRUE
)

## Ensure the grouping variable is treated as a factor
metadata$group <- factor(
  metadata$group
)

stopifnot(
  !anyNA(metadata$visit)
)

## Return the updated metadata to the experiment
SummarizedExperiment::colData(se) <- S4Vectors::DataFrame(
  metadata,
  row.names = rownames(metadata)
)

experiments[["transcriptomics"]] <- se

lot_empirical@mae <- MultiAssayExperiment::MultiAssayExperiment(
  experiments = experiments
)

## Calculate empirical group-by-visit means

```

```

lot_empirical <- lot_from_empirical(
  object = lot_empirical,
  assay = "transcriptomics",
  visits = visits,
  group_col = "group",
  visit_col = "visit",
  estimator = "empirical",
  overwrite = TRUE,
  verbose = FALSE
)

## Inspect the Visit Mean Matrix
utils::head(
  as.data.frame(lot_empirical@vmm)
)

stopifnot(
  nrow(lot_empirical@vmm) == 852L,
  all(
    unique(
      as.character(
        lot_empirical@vmm$estimator
      )
    ) == "empirical"
  )
)

if (requireNamespace("mgcv", quietly = TRUE)) {

  data("lot_glucold")

  visits <- c(
    "Baseline",
    "6 Months",
    "30 Months"
  )

  ## Work on a copy of the example object
  lot_gam <- lot_glucold

  ## Extract the transcriptomics experiment
  experiments <- MultiAssayExperiment::experiments(
    lot_gam@mae
  )

  se <- experiments[["transcriptomics"]]

  metadata <- as.data.frame(
    SummarizedExperiment::colData(se)
  )

  ## Define chronological visit order

```

```

metadata$visit <- factor(
  metadata$visit,
  levels = visits,
  ordered = TRUE
)

## Create numeric time values used by the smooth
metadata$time <- c(
  "Baseline" = 0,
  "6 Months" = 6,
  "30 Months" = 30
)[as.character(metadata$visit)]

## Ensure grouping variables are factors
metadata$group <- factor(
  metadata$group
)

metadata$subject_id <- factor(
  metadata$subject_id
)

stopifnot(
  !anyNA(metadata$visit),
  !anyNA(metadata$time)
)

## Return the updated metadata to the experiment
SummarizedExperiment::colData(se) <- S4Vectors::DataFrame(
  metadata,
  row.names = rownames(metadata)
)

experiments[["transcriptomics"]] <- se

lot_gam@mae <- MultiAssayExperiment::MultiAssayExperiment(
  experiments = experiments
)

## Estimate group-specific smooth trajectories
lot_gam <- lot_from_gam(
  object = lot_gam,
  assay = "transcriptomics",
  formula_fixed =
    y ~
    group +
    mgcv::s(
      time,
      by = group,
      k = 3
    ) +
    mgcv::s(
      subject_id,

```

```

        bs = "re"
      ),
      group_col = "group",
      visit_col = "visit",
      subject_col = "subject_id",
      visits = visits,
      overwrite = TRUE,
      verbose = FALSE
    )

    ## Inspect the Visit Mean Matrix
    utils::head(
      as.data.frame(lot_gam@vmm)
    )

    stopifnot(
      nrow(lot_gam@vmm) == 852L,
      all(
        unique(
          as.character(lot_gam@vmm$estimator)
        ) == "gam"
      )
    )
  }

  if (
    requireNamespace("lme4", quietly = TRUE) &&
    requireNamespace("emmeans", quietly = TRUE) &&
    requireNamespace("BiocParallel", quietly = TRUE)
  ) {
    data(lot_glucold)

    lot_lmm <- lot_from_lmm(
      object = lot_glucold,
      assay = "transcriptomics",
      formula_fixed = y ~ group * visit + (1 | subject_id),
      visits = c(
        "Baseline",
        "6 Months",
        "30 Months"
      ),
      BPPARAM = BiocParallel::SerialParam(),
      overwrite = TRUE
    )

    lot_lmm
  }

  data("lot_glucold")

  visits <- c(

```

```

    "Baseline",
    "6 Months",
    "30 Months"
  )

  ## Work on a copy of the example object
  lot_masigpro <- lot_glucold

  ## Extract the transcriptomics experiment
  experiments <- MultiAssayExperiment::experiments(
    lot_masigpro@mae
  )

  se <- experiments[["transcriptomics"]]

  metadata <- as.data.frame(
    SummarizedExperiment::colData(se)
  )

  ## Define chronological visit order
  metadata$visit <- factor(
    metadata$visit,
    levels = visits,
    ordered = TRUE
  )

  ## Create numeric visit times required by the polynomial model
  metadata$time <- c(
    "Baseline" = 0,
    "6 Months" = 6,
    "30 Months" = 30
  )[as.character(metadata$visit)]

  stopifnot(
    !anyNA(metadata$visit),
    !anyNA(metadata$time)
  )

  ## Return the updated metadata to the experiment
  SummarizedExperiment::colData(se) <- S4Vectors::DataFrame(
    metadata,
    row.names = rownames(metadata)
  )

  experiments[["transcriptomics"]] <- se

  lot_masigpro@mae <- MultiAssayExperiment::MultiAssayExperiment(
    experiments = experiments
  )

  ## Estimate group-specific polynomial trajectories
  lot_masigpro <- lot_from_masigpro(
    object = lot_masigpro,

```

```

    assay = "transcriptomics",
    group_col = "group",
    visit_col = "visit",
    time_col = "time",
    subject_col = "subject_id",
    visits = visits,
    degree = 2,
    overwrite = TRUE,
    verbose = FALSE
  )

  ## Inspect the Visit Mean Matrix
  utils::head(
    as.data.frame(lot_masigpro@vmm)
  )

  data("lot_glucold")

  visits <- c(
    "Baseline",
    "6 Months",
    "30 Months"
  )

  fitted <- lot_from_empirical(
    object = lot_glucold,
    assay = "transcriptomics",
    visits = visits,
    overwrite = TRUE
  )

  utils::head(
    as.data.frame(
      lot_vmm(fitted)
    )
  )

```

lot_from_flexmix_clusters

Cluster molecular features using FlexMix trajectory models

Description

Fits finite mixtures of longitudinal regression trajectories to molecular features within one clinical or treatment group. Candidate cluster resolutions are evaluated using likelihood-based model statistics and LongOmicsTraj topology purity.

Usage

```
lot_from_flexmix_clusters(
```

```

object,
assay,
group,
visits,
k = 2:5,
assay_data = NULL,
group_col = "group",
visit_col = "visit",
sample_col = "sample_id",
trajectory_source = c("estimated", "mean"),
trajectory_estimator = NULL,
model = c("quadratic", "linear"),
scale_features = TRUE,
purity_threshold = 0.9,
required_passing_fraction = 0.8,
min_cluster_size = 5L,
nrep = 10L,
seed = 42L,
store_fits = FALSE,
overwrite = FALSE,
verbose = TRUE
)

```

Arguments

object	A ‘LongOmicsTraj’ object containing molecular data and feature-level OTS results.
assay	Name of the experiment in the ‘MultiAssayExperiment’.
group	Clinical or treatment group to analyse.
visits	Character vector giving the chronological visit order.
k	Integer vector containing candidate numbers of clusters.
assay_data	Optional assay-matrix name. The first available matrix is used by default.
group_col	Metadata column containing clinical groups.
visit_col	Metadata column containing visits.
sample_col	Metadata column containing sample identifiers.
trajectory_source	Source of the gene trajectories: “estimated” uses ‘object@vmm’, while “mean” calculates visit-level assay means.
trajectory_estimator	Optional estimator used to select trajectories from ‘object@vmm’, such as “masigpro”, “gam”, “lmm” or “empirical”.
model	Trajectory regression model: “quadratic” or “linear”.
scale_features	Logical; standardise each molecular-feature trajectory before clustering.
purity_threshold	Minimum dominant-OTS proportion required for a cluster to pass the topology-purity criterion.

required_passing_fraction	Minimum fraction of clusters required to pass the topology-purity criterion.
min_cluster_size	Minimum allowed number of molecular features in a cluster.
nrep	Number of repeated FlexMix initialisations for each candidate value of 'k'.
seed	Random seed.
store_fits	Logical; store the fitted FlexMix models in 'object@results'.
overwrite	Logical; overwrite existing FlexMix clustering results.
verbose	Logical; print progress messages.

Details

Trajectories can be obtained from model-estimated values already stored in 'object@vmm', or calculated as mean molecular values at each visit from the original assay matrix.

The selected solution is the smallest candidate 'k' for which:

- the required fraction of clusters reaches 'purity_threshold'; and
- all clusters contain at least 'min_cluster_size' features.

If no candidate satisfies both criteria, the solution with the lowest BIC is selected.

When 'trajectory_source = "estimated"', LongOmicsTraj trajectories stored in 'object@vmm' are used. This is generally preferred because trajectories may already account for repeated measurements, smoothing or other model assumptions.

Value

A 'LongOmicsTraj' object containing model-comparison statistics, the selected cluster resolution, feature assignments, cluster trajectories, OTS distributions and topology-purity summaries.

Examples

```
data("lot_glucold_ots")

lot_flexmix <- lot_from_flexmix_clusters(
  object = lot_glucold_ots,
  assay = "transcriptomics",
  group = "ICS 6 months then withdrawal",
  visits = c(
    "Baseline",
    "6 Months",
    "30 Months"
  ),
  trajectory_source = "estimated",
  trajectory_estimator = "masigpro",
  k = 2:5,
  purity_threshold = 0.90,
  required_passing_fraction = 0.80,
  min_cluster_size = 5
)
```

lot_glucold	<i>Cleaned longitudinal transcriptomics dataset from the GLUCOLD study</i>
-------------	--

Description

A pre-processed ‘LongOmicsTraj’ object containing an integrated ‘MultiAssayExperiment’ of 71 candidate and hypothesis-driven genes across longitudinal milestones from the GLUCOLD study.

Usage

```
data(lot_glucold)
```

Format

A ‘LongOmicsTraj’ object containing a ‘MultiAssayExperiment’ with one experiment named “transcriptomics”.

The internal ‘colData’ contains:

sample_id Character. Unique sample identifier.

subject_id Character. Unique participant identifier.

visit Factor. Chronological study visit.

group Factor. Treatment-allocation group.

time Numeric or factor representation of study time.

treatment Factor. Original clinical treatment allocation.

Source

NCBI Gene Expression Omnibus accession GSE36221.

lot_glucold_clinical	<i>GLUCOLD clinical metadata</i>
----------------------	----------------------------------

Description

Clinical and longitudinal sample metadata used to construct the example ‘LongOmicsTraj’ objects.

Usage

```
data(lot_glucold_clinical)
```

Format

A data frame containing sample identifiers, subject identifiers, study visits, treatment groups, time and treatment allocation.

Source

NCBI Gene Expression Omnibus accession GSE36221.

lot_glucold_expression

GLUCOLD expression matrix

Description

Pre-processed longitudinal transcriptomic measurements for 71 selected candidate and hypothesis-driven genes from the GLUCOLD study.

Usage

```
data(lot_glucold_expression)
```

Format

A numeric matrix with molecular features in rows and samples in columns.

Source

NCBI Gene Expression Omnibus accession GSE36221.

lot_glucold_ots

GLUCOLD LongOmicsTraj object with OTS results

Description

A processed 'LongOmicsTraj' object containing model-estimated molecular trajectories, between-visit changes, thresholds and Omics Trajectory Signature assignments.

Usage

```
data(lot_glucold_ots)
```

Format

A 'LongOmicsTraj' object.

Source

Derived from NCBI Gene Expression Omnibus accession GSE36221.

`lot_mae`*Access components of a LongOmicsTraj object*

Description

Accessor functions for retrieving the principal data structures stored in a 'LongOmicsTraj' object. These functions provide a stable interface to the object and should be used instead of direct slot access.

Usage

```
lot_mae(object)

lot_vmm(object)

lot_deltas(object)

lot_thresholds(object)

lot_ots(object)

lot_relationships(object)

lot_results(object)

## S4 method for signature 'LongOmicsTraj'
lot_mae(object)

## S4 method for signature 'LongOmicsTraj'
lot_vmm(object)

## S4 method for signature 'LongOmicsTraj'
lot_deltas(object)

## S4 method for signature 'LongOmicsTraj'
lot_thresholds(object)

## S4 method for signature 'LongOmicsTraj'
lot_ots(object)

## S4 method for signature 'LongOmicsTraj'
lot_relationships(object)

## S4 method for signature 'LongOmicsTraj'
lot_results(object)
```

Arguments

object A 'LongOmicsTraj' object.

Details

- 'lot_mae()'** Returns the underlying [MultiAssayExperiment](#) containing the molecular assays and sample metadata.
- 'lot_vmm()'** Returns the visit-level molecular model matrix containing estimated molecular values for each feature, group and visit.
- 'lot_deltas()'** Returns estimated molecular changes between consecutive visits.
- 'lot_thresholds()'** Returns the feature-specific thresholds used to classify molecular changes as increasing, decreasing or stable.
- 'lot_ots()'** Returns the Omics Trajectory Signature assignments obtained by combining the classified transitions across visits.
- 'lot_relationships()'** Returns relationships among molecular objects, such as feature-to-cluster assignments.
- 'lot_results()'** Returns additional analysis results stored by trajectory estimators, clustering procedures and diagnostic functions.

Value

The requested component of the 'LongOmicsTraj' object. The exact class depends on the accessor:

- 'lot_mae()' returns a 'MultiAssayExperiment'.
- 'lot_vmm()', 'lot_deltas()', 'lot_thresholds()', 'lot_ots()' and 'lot_relationships()' return 'S4Vectors::DataFrame' objects.
- 'lot_results()' returns a named list.

Examples

```
data("lot_glucold_ots")

## Extract the underlying experiment container.
mae <- lot_mae(lot_glucold_ots)
names(MultiAssayExperiment::experiments(mae))

## Inspect the visit-level molecular estimates.
vmm <- lot_vmm(lot_glucold_ots)
dim(vmm)
utils::head(as.data.frame(vmm))

## Examine the estimated changes between adjacent visits.
deltas <- lot_deltas(lot_glucold_ots)
dim(deltas)
utils::head(
  as.data.frame(deltas)[
    ,
    c(
```

```

        "object_id",
        "group",
        "transition",
        "delta_hat"
    )
]
)

## Inspect the thresholds used to classify molecular changes.
thresholds <- lot_thresholds(lot_glucold_ots)
utils::head(
  as.data.frame(thresholds)[
    ,
    c(
      "object_id",
      "threshold_method",
      "delta_g"
    )
  ]
)

## Inspect the resulting Omics Trajectory Signatures.
ots <- lot_ots(lot_glucold_ots)
table(
  as.data.frame(ots)$topology_label
)

## Relationships and additional results may be empty when no
## clustering analysis has been performed.
relationships <- lot_relationships(lot_glucold_ots)
nrow(relationships)

results <- lot_results(lot_glucold_ots)
names(results)

```

lot_map_to_ots

Map molecular changes to Omics Trajectory Signatures

Description

Classifies molecular changes between consecutive visits and combines them into feature-level Omics Trajectory Signature labels.

Classifies feature-level molecular changes between consecutive visits and combines the transition states into Omics Trajectory Signature labels.

Usage

```
lot_map_to_ots(object, visits, overwrite = FALSE, ...)
```

```
## S4 method for signature 'LongOmicsTraj'
```

```
lot_map_to_ots(object, visits, overwrite = FALSE, ...)
```

Arguments

object	A 'LongOmicsTraj' object.
visits	Character vector giving the visits in chronological order.
overwrite	Logical; overwrite existing OTS results when 'TRUE'.
...	Additional arguments reserved for future extensions.

Details

The method uses delta estimates and thresholds already stored in the 'LongOmicsTraj' object.

Value

A 'LongOmicsTraj' object containing updated OTS assignments.

A 'LongOmicsTraj' object with updated OTS assignments.

Examples

```
data("lot_glucold_ots")

visits <- c(
  "Baseline",
  "6 Months",
  "30 Months"
)

fitted <- lot_map_to_ots(
  object = lot_glucold_ots,
  visits = visits,
  overwrite = TRUE
)

utils::head(
  as.data.frame(
    lot_ots(fitted)
  )
)
```

lot_plot_cluster_selection

Plot topology-aware cluster selection

Description

Displays likelihood-based and topology-aware statistics across candidate numbers of FlexMix clusters.

Usage

```
lot_plot_cluster_selection(
  object,
  metrics = c("BIC", "weighted_purity", "passing_fraction", "minimum_cluster_size"),
  purity_threshold = 0.9,
  required_passing_fraction = 0.8,
  min_cluster_size = 5L,
  show_selected = TRUE
)
```

Arguments

<code>object</code>	A 'LongOmicsTraj' object containing FlexMix model-comparison results.
<code>metrics</code>	Metrics to display. Supported values are "BIC", "weighted_purity", "passing_fraction", "minimum_cluster_size", "minimum_purity" and "mean_purity".
<code>purity_threshold</code>	Reference purity threshold.
<code>required_passing_fraction</code>	Required fraction of clusters passing the purity rule.
<code>min_cluster_size</code>	Minimum acceptable cluster size.
<code>show_selected</code>	Logical; mark the selected cluster resolution.

Value

A 'ggplot' object.

Examples

```
data("lot_glucold_ots")

example_object <- lot_glucold_ots

example_results <- lot_results(
  example_object
)

example_results$flexmix_model_comparison <- S4Vectors::DataFrame(
  k = 2:4,
  BIC = c(540, 505, 512),
  weighted_purity = c(0.76, 0.91, 0.88),
  passing_fraction = c(0.50, 1.00, 0.75),
  minimum_cluster_size = c(18, 12, 6),
  minimum_purity = c(0.68, 0.87, 0.72),
  mean_purity = c(0.78, 0.92, 0.86)
)

example_results$flexmix_selected_k <- 3L
```

```

methods::slot(
  example_object,
  "results"
) <- example_results

plot <- lot_plot_cluster_selection(
  object = example_object
)

plot

```

lot_plot_cluster_topology

Plot trajectory topology profiles for fitted clusters

Description

Displays the representative longitudinal trajectory and the within-cluster Omics Trajectory Signature composition for each selected FlexMix cluster.

Usage

```

lot_plot_cluster_topology(
  object,
  clusters = NULL,
  group = NULL,
  value_type = c("scaled", "raw"),
  show_features = TRUE,
  max_feature_lines = 50L,
  purity_threshold = 0.9,
  sort_by = c("cluster", "purity", "size", "dominant_topology"),
  seed = 1L
)

```

Arguments

object	A 'LongOmicsTraj' object containing FlexMix clustering results.
clusters	Optional vector of cluster identifiers. By default, all clusters in the selected FlexMix solution are shown.
group	Optional clinical group. When omitted, the group stored in the selected FlexMix results is used.
value_type	Trajectory values to display: "scaled" uses the clustering values, while "raw" uses the original estimated trajectory values.
show_features	Logical; display individual feature trajectories.
max_feature_lines	Maximum number of individual feature trajectories drawn per cluster.

purity_threshold	Reference purity threshold shown in the OTS panel.
sort_by	Ordering of clusters: "cluster", "purity", "size" or "dominant_topology".
seed	Random seed used when subsampling feature trajectories.

Details

The left-hand panel shows the representative cluster trajectory and, optionally, the individual molecular-feature trajectories. The right-hand panel shows the proportion of features assigned to each OTS class.

Value

A 'patchwork' plot object.

Examples

```
data("lot_glucold_ots")

example_object <- lot_glucold_ots

example_results <- lot_results(
  example_object
)

example_results$flexmix_gene_assignments <-
  S4Vectors::DataFrame(
    gene_id = c(
      "gene_1",
      "gene_2",
      "gene_3",
      "gene_4"
    ),
    cluster = c(
      "1",
      "1",
      "2",
      "2"
    ),
    group = "treated"
  )

example_results$flexmix_cluster_trajectories <-
  S4Vectors::DataFrame(
    cluster = rep(
      c("1", "2"),
      each = 3
    ),
    group = "treated",
    visit = rep(
      c(
        "Baseline",
```

```

        "Month 6",
        "Month 12"
    ),
    times = 2
),
mean_clustering_value = c(
    0, 1, 0,
    0, -1, 0.5
),
mean_trajectory_value = c(
    5, 6, 5,
    5, 4, 5.5
)
)

example_results$flexmix_ots_distribution <-
S4Vectors::DataFrame(
    cluster = c(
        "1",
        "1",
        "2",
        "2"
    ),
    group = "treated",
    topology_label = c(
        "up_down",
        "flat_flat",
        "down_up",
        "flat_flat"
    ),
    n_genes = c(
        3L,
        1L,
        3L,
        1L
    ),
    proportion = c(
        0.75,
        0.25,
        0.75,
        0.25
    )
)

example_results$flexmix_cluster_purity <-
S4Vectors::DataFrame(
    cluster = c(
        "1",
        "2"
    ),
    group = "treated",
    n_genes = c(
        2L,

```

```

      2L
    ),
    dominant_topology = c(
      "up_down",
      "down_up"
    ),
    purity = c(
      0.75,
      0.75
    )
  )
)

example_results$flexmix_trajectory_data <-
  S4Vectors::DataFrame(
    gene_id = rep(
      c(
        "gene_1",
        "gene_2",
        "gene_3",
        "gene_4"
      ),
      each = 3
    ),
    visit = rep(
      c(
        "Baseline",
        "Month 6",
        "Month 12"
      ),
      times = 4
    ),
    clustering_value = c(
      0, 1.1, 0.1,
      0, 0.9, -0.1,
      0, -1.1, 0.6,
      0, -0.9, 0.4
    ),
    mean_expression = c(
      5, 6.1, 5.1,
      5, 5.9, 4.9,
      5, 3.9, 5.6,
      5, 4.1, 5.4
    )
  )
)

methods::slot(
  example_object,
  "results"
) <- example_results

plot <- lot_plot_cluster_topology(
  object = example_object,
  group = "treated",

```

```

    show_features = FALSE
  )

plot

```

lot_plot_cluster_trajectory_arms

Plot trajectory arms within molecular-feature clusters

Description

Visualises the distribution of feature-level Omics Trajectory Signatures within each molecular-feature cluster. Each panel represents one cluster and clinical group. Arrows emerge from a shared cluster node and represent the distinct trajectory classes found inside that cluster.

Usage

```

lot_plot_cluster_trajectory_arms(
  object,
  clusters = NULL,
  groups = NULL,
  top_n = NULL,
  min_proportion = 0,
  show_counts = TRUE,
  show_percent = TRUE,
  scale_width = TRUE,
  ncol = NULL
)

```

Arguments

object	A ‘LongOmicsTraj’ object containing ‘cluster_ots_distribution’ results.
clusters	Optional character vector of cluster identifiers to show. By default, all clusters are included.
groups	Optional character vector of clinical or treatment groups to show. By default, all groups are included.
top_n	Optional maximum number of trajectory arms displayed within each cluster-group panel. By default, all trajectory classes are shown.
min_proportion	Minimum within-cluster trajectory proportion required for display. The default is ‘0’.
show_counts	Logical; include feature counts in trajectory labels.
show_percent	Logical; include percentages in trajectory labels.
scale_width	Logical; scale arrow width by trajectory proportion.
ncol	Number of facet columns.

Details

Arrow width is proportional to the fraction of clustered features assigned to each trajectory class.

Value

A ‘ggplot’ object.

Examples

```
data("lot_glucold_ots")

example_object <- lot_glucold_ots

example_results <- lot_results(
  example_object
)

example_results$cluster_ots_distribution <-
  S4Vectors::DataFrame(
    cluster_id = c(
      "cluster_1",
      "cluster_1",
      "cluster_2",
      "cluster_2"
    ),
    group = "treated",
    topology_label = c(
      "up_down",
      "flat_flat",
      "down_up",
      "flat_down"
    ),
    count = c(
      8L,
      2L,
      7L,
      3L
    ),
    proportion = c(
      0.8,
      0.2,
      0.7,
      0.3
    )
  )

methods::slot(
  example_object,
  "results"
) <- example_results

plot <- lot_plot_cluster_trajectory_arms(
```

```

    object = example_object,
    groups = "treated"
  )

plot

```

lot_plot_ots_distribution

Plot the distribution of Omics Trajectory Signatures

Description

Summarises the proportion or number of molecular features assigned to each Omics Trajectory Signature within each group.

Usage

```

lot_plot_ots_distribution(
  object,
  estimator = NULL,
  type = c("proportion", "count"),
  palette = "Set3"
)

```

Arguments

object	A ‘LongOmicsTraj’ object containing computed OTS results.
estimator	Optional estimator name used to filter the OTS results.
type	Either “proportion” or “count”.
palette	Name of an RColorBrewer qualitative palette.

Value

A ‘ggplot’ object.

Examples

```

data("lot_glucold_ots")

plot <- lot_plot_ots_distribution(
  object = lot_glucold_ots,
  estimator = "masigpro",
  type = "proportion"
)

plot

```

```
lot_plot_topology_deviation
```

Plot topology deviations within clusters

Description

Creates a PAMR-inspired nail plot showing how strongly each Omics Trajectory Signature is enriched or depleted within each cluster compared with its overall frequency.

Usage

```
lot_plot_topology_deviation(
  object,
  clusters = NULL,
  group = NULL,
  min_abs_deviation = 0,
  top_n = NULL,
  sort_topologies = TRUE
)
```

Arguments

<code>object</code>	A ‘LongOmicsTraj’ object containing selected FlexMix clustering results.
<code>clusters</code>	Optional vector of clusters to show.
<code>group</code>	Optional clinical group.
<code>min_abs_deviation</code>	Minimum absolute deviation required for display.
<code>top_n</code>	Optional maximum number of topologies displayed per cluster.
<code>sort_topologies</code>	Logical; order topologies by deviation within each cluster.

Details

For topology ‘t’ in cluster ‘c’, deviation is calculated as:

$$D_{ct} = p_{ct} - p_t$$

where ‘p_{ct}’ is the within-cluster topology proportion and ‘p_t’ is the overall topology proportion.

Value

A ‘ggplot’ object.

Examples

```
data("lot_glucold_ots")

example_object <- lot_glucold_ots

example_results <- lot_results(
  example_object
)

example_results$flexmix_ots_distribution <-
  S4Vectors::DataFrame(
    cluster = c(
      "1",
      "1",
      "2",
      "2"
    ),
    group = "treated",
    topology_label = c(
      "up_down",
      "flat_flat",
      "up_down",
      "flat_flat"
    ),
    n_genes = c(
      8L,
      2L,
      3L,
      7L
    ),
    proportion = c(
      0.8,
      0.2,
      0.3,
      0.7
    )
  )

methods::slot(
  example_object,
  "results"
) <- example_results

plot <- lot_plot_topology_deviation(
  object = example_object,
  group = "treated"
)

plot
```

lot_plot_trajectory *Plot longitudinal trajectories for selected molecular features*

Description

Plots observed longitudinal measurements for one or more molecular features across clinical groups. Individual-subject profiles, group means, uncertainty intervals and assigned Omics Trajectory Signatures can be displayed.

Usage

```
lot_plot_trajectory(
  object,
  features,
  assay = "transcriptomics",
  assay_data = NULL,
  groups = NULL,
  estimator = NULL,
  show_subjects = FALSE,
  summary = c("mean_se", "mean_ci", "mean"),
  facet_scales = c("free_y", "fixed"),
  ncol = NULL
)
```

Arguments

object	A ‘LongOmicsTraj’ object.
features	Character vector of feature identifiers, such as gene symbols.
assay	Name of the experiment in the ‘MultiAssayExperiment’.
assay_data	Optional name of the matrix within the ‘SummarizedExperiment’. By default, the first available matrix is used.
groups	Optional character vector of groups to display. By default, all available groups are included.
estimator	Optional estimator used to retrieve OTS labels.
show_subjects	Logical; if ‘TRUE’, individual-subject trajectories are displayed as faint lines.
summary	One of “mean_se”, “mean_ci” or “mean”.
facet_scales	Either “free_y” or “fixed”.
ncol	Number of facet columns.

Value

A ‘ggplot’ object.

Examples

```
data("lot_glucold_ots")

lot_plot_trajectory(
  lot_glucold_ots,
  features = c("ATP2A3", "CAT"),
  groups = c(
    "ICS 30 months",
    "ICS 6 months then withdrawal",
    "Placebo"
  ),
  estimator = "masigpro",
  show_subjects = TRUE
)
```

lot_query	<i>Query LongOmicsTraj results</i>
-----------	------------------------------------

Description

Retrieves molecular features or trajectory results matching user-defined criteria.

Creates an exclusion condition for use in 'lot_query()'. Features whose topology labels match any value in 'x' are excluded.

Filters Omics Trajectory Signature assignments stored in a 'LongOmicsTraj' object.

Usage

```
lot_query(object, ...)

NOT(x)

## S4 method for signature 'LongOmicsTraj'
lot_query(
  object,
  object_type = NULL,
  assay = NULL,
  group = NULL,
  topology = NULL,
  conditions = NULL,
  estimator = NULL
)
```

Arguments

object	A 'LongOmicsTraj' object.
...	Query arguments passed to the corresponding method.
x	Character vector of topology labels to exclude.

object_type	Optional character vector of molecular object types.
assay	Optional character vector of assays.
group	Optional character vector of clinical or treatment groups.
topology	Optional character vector of OTS topology labels.
conditions	Optional named list defining group-specific topology conditions. Each element may be a character vector, a 'NOT()' condition, or a list with 'include' and/or 'exclude' elements.
estimator	Optional character vector of trajectory estimators.

Details

Queries may be restricted by molecular object type, assay, group, estimator or topology label. The 'conditions' argument supports group-specific cross-arm queries.

Simple filters are applied directly to the OTS table. Cross-group queries supplied through 'conditions' require the same molecular feature to satisfy all named group-specific conditions.

Value

A filtered result object.

An object of class "'lot_not'" representing an exclusion condition.

A 'DataFrame' containing the filtered OTS assignments.

Examples

```
data("lot_glucold_ots")

flat <- lot_query(
  object = lot_glucold_ots,
  topology = "flat_flat"
)

nrow(flat)

utils::head(
  as.data.frame(flat)
)
```

lot_summary

Summarise a LongOmicsTraj object

Description

Summarises assays, study design, available estimators and Omics Trajectory Signature distributions stored in a 'LongOmicsTraj' object.

Usage

```
lot_summary(object, estimator = NULL, top_n = 5L)
```

Arguments

object	A 'LongOmicsTraj' object.
estimator	Optional estimator used to filter OTS results.
top_n	Number of most common OTS classes to retain per group.

Value

An object of class 'lot_summary'. The object is a list containing dataset dimensions, study design information, result availability and OTS summaries.

Examples

```
data("lot_glucold_ots")

result <- lot_summary(
  lot_glucold_ots,
  estimator = "masigpro"
)

result$assay_summary
result$top_ots
```

plot_cluster_spaghetti

Plot clustered molecular-feature trajectories

Description

Creates one spaghetti plot per molecular-feature cluster. Individual feature trajectories are coloured by their Omics Trajectory Signature, while the cluster-level representative trajectory is shown in black.

Usage

```
plot_cluster_spaghetti(object, assay_name, visits_vector, max_clusters = 3L)
```

Arguments

object	A 'LongOmicsTraj' object containing cluster relationships, gene-level OTS results, cluster trajectories and OTS metrics.
assay_name	Name of the assay to display.
visits_vector	Character vector giving visits in chronological order.
max_clusters	Maximum number of clusters to plot.

Value

Invisibly returns a named list of ‘ggplot’ objects.

plot_cluster_spaghetti_single	<i>Plot one clustered molecular-feature trajectory profile</i>
-------------------------------	--

Description

Creates a spaghetti plot for a single molecular-feature cluster. Individual feature trajectories are coloured by their Omics Trajectory Signature, while the representative cluster trajectory is displayed in black.

Usage

```
plot_cluster_spaghetti_single(object, cluster_id, assay_name, visits_vector)
```

Arguments

object	A ‘LongOmicsTraj’ object containing cluster relationships, gene-level OTS results, cluster trajectories and OTS metrics.
cluster_id	Identifier of the cluster to display.
assay_name	Name of the assay to display.
visits_vector	Character vector giving visits in chronological order.

Value

A ‘ggplot’ object.

print.lot_summary	<i>Print a LongOmicsTraj summary</i>
-------------------	--------------------------------------

Description

Print a LongOmicsTraj summary

Usage

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

Arguments

x	A ‘lot_summary’ object.
...	Additional arguments, currently ignored.

Value

The summary object, returned invisibly.

show, LongOmicsTraj-method

Display a LongOmicsTraj object

Description

Prints a concise summary of the assays, study design and analysis results stored in a ‘LongOmicsTraj’ object.

Usage

```
## S4 method for signature 'LongOmicsTraj'
show(object)
```

Arguments

object A ‘LongOmicsTraj’ object.

Value

The object is returned invisibly.

Examples

```
data("lot_glucold")

## Printing the object invokes the show() method.
lot_glucold

## The object is returned invisibly.
invisible(lot_glucold)
```

Index

- * **datasets**
 - lot_glucold, [28](#)
 - lot_glucold_clinical, [28](#)
 - lot_glucold_expression, [29](#)
 - lot_glucold_ots, [29](#)
- compute_vmm_deltas, [3](#)
- LongOmicsTraj-class, [4](#)
- lot_accessors (lot_mae), [30](#)
- lot_cluster_purity, [4](#)
- lot_cluster_topology_score, [6](#)
- lot_compare_estimators, [7](#)
- lot_compare_estimators, LongOmicsTraj-method (lot_compare_estimators), [7](#)
- lot_compute_deltas, [9](#)
- lot_compute_deltas, LongOmicsTraj-method (lot_compute_deltas), [9](#)
- lot_compute_thresholds, [10](#)
- lot_compute_thresholds, LongOmicsTraj-method (lot_compute_thresholds), [10](#)
- lot_consensus, [12](#)
- lot_consensus, DFrame-method (lot_consensus), [12](#)
- lot_create, [13](#)
- lot_deltas (lot_mae), [30](#)
- lot_deltas, LongOmicsTraj-method (lot_mae), [30](#)
- lot_from_clusters, [16](#)
- lot_from_clusters, LongOmicsTraj-method (lot_from_clusters), [16](#)
- lot_from_empirical, [17](#)
- lot_from_empirical, LongOmicsTraj-method (lot_from_empirical), [17](#)
- lot_from_flexmix_clusters, [25](#)
- lot_from_gam (lot_from_empirical), [17](#)
- lot_from_gam, LongOmicsTraj-method (lot_from_empirical), [17](#)
- lot_from_lmm (lot_from_empirical), [17](#)
- lot_from_lmm, LongOmicsTraj-method (lot_from_empirical), [17](#)
- lot_from_masigpro (lot_from_empirical), [17](#)
- lot_from_masigpro, LongOmicsTraj-method (lot_from_empirical), [17](#)
- lot_glucold, [28](#)
- lot_glucold_clinical, [28](#)
- lot_glucold_expression, [29](#)
- lot_glucold_ots, [29](#)
- lot_mae, [30](#)
- lot_mae, LongOmicsTraj-method (lot_mae), [30](#)
- lot_map_to_ots, [32](#)
- lot_map_to_ots, LongOmicsTraj-method (lot_map_to_ots), [32](#)
- lot_ots (lot_mae), [30](#)
- lot_ots, LongOmicsTraj-method (lot_mae), [30](#)
- lot_plot_cluster_selection, [33](#)
- lot_plot_cluster_topology, [35](#)
- lot_plot_cluster_trajectory_arms, [39](#)
- lot_plot_ots_distribution, [41](#)
- lot_plot_topology_deviation, [42](#)
- lot_plot_trajectory, [44](#)
- lot_query, [45](#)
- lot_query, LongOmicsTraj-method (lot_query), [45](#)
- lot_relationships (lot_mae), [30](#)
- lot_relationships, LongOmicsTraj-method (lot_mae), [30](#)
- lot_results (lot_mae), [30](#)
- lot_results, LongOmicsTraj-method (lot_mae), [30](#)
- lot_summary, [46](#)
- lot_thresholds (lot_mae), [30](#)
- lot_thresholds, LongOmicsTraj-method (lot_mae), [30](#)
- lot_trajectory_estimators

(lot_from_empirical), [17](#)
lot_vmm (lot_mae), [30](#)
lot_vmm, LongOmicsTraj-method (lot_mae),
[30](#)
MultiAssayExperiment, [31](#)
NOT (lot_query), [45](#)
plot_cluster_spaghetti, [47](#)
plot_cluster_spaghetti_single, [48](#)
print.lot_summary, [48](#)
show, LongOmicsTraj-method, [49](#)