

# Package ‘ClonoCluster’

February 7, 2022

**Title** A method for using clonal origin to inform transcriptome clustering

**Version** 0.0.1

**Description** Package to facilitate the integration of clonal clusters with traditional transcriptome based clustering to generate intermediate levels of clustering for analysis.

**License** See LICENSE

**Imports** magrittr,  
data.table,  
Seurat,  
ggplot2,  
cowplot,  
scales,  
purrr,  
stringr,  
testthat,  
stats,  
uwot,  
reticulate,  
irlba,  
ROCR,  
ggalluvial,  
utils,  
testthat,  
parallel

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Config/testthat/edition** 3

## R topics documented:

barcode_warp . . . . .	2
build_barcode_matrix . . . . .	2
build_barcode_matrix_fast . . . . .	3
c25 . . . . .	3
cast_confusion . . . . .	4
clonocluster . . . . .	4
clonocluster_model . . . . .	5
cw_colors . . . . .	5

dt2m . . . . .	6
engage_warp . . . . .	6
FindAllMarkers_Seurat . . . . .	7
Find_Markers_ROC . . . . .	7
irlba_wrap . . . . .	8
Plot_alluvia . . . . .	8
Plot_alluvia_counts . . . . .	9
Plot_alluvia_track . . . . .	10
ROCR_wrap . . . . .	11
tdt . . . . .	12
ttheme . . . . .	12
umap_matrix . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

barcode_warp	<i>Warp principal components towards linear projection of barcodes.</i>
--------------	---

---

## Description

Warp principal components towards linear projection of barcodes.

## Usage

```
barcode_warp(irl, bt, s)
```

## Arguments

irl	Matrix. Principal components matrix, output from <code>clonocluster::irlba_wrap</code> .
bt	Data table. Barcode table of two columns cell IDs ("rn") and barcodes ("Barcode").
s	Numeric. Warp factor, from 0 to 10.

## Value

A matrix of the same dimensions as `irl` suitable for further reduction by UMAP.

---

build_barcode_matrix	<i>Build a sparse matrix representing the barcode graph.</i>
----------------------	--

---

## Description

Build a sparse matrix representing the barcode graph.

## Usage

```
build_barcode_matrix(bt, value = NULL)
```

**Arguments**

bt	Data.table. Columns are "rn" (cell ids) and "Barcode".
value	Numeric. A value to be used for edge weights within barcodes. Default NULL will use the reciprocal of barcode size.

**Value**

A sparse matrix of class dgCMatrix suitable for clonocluster\_model or Seurat::FindClusters.

---

`build_barcode_matrix_fast`

*Build a sparse matrix representing the barcode graph.*

---

**Description**

Build a sparse matrix representing the barcode graph.

**Usage**

```
build_barcode_matrix_fast(bt, value = NULL)
```

**Arguments**

bt	Data.table. Columns are "rn" (cell ids) and "Barcode".
value	Numeric. A value to be used for edge weights within barcodes. Default NULL will use the reciprocal of barcode size.

**Value**

A sparse matrix of class dgCMatrix suitable for clonocluster\_model or Seurat::FindClusters.

---

`c25`

*internal function, vector of colors for discrete plots.*

---

**Description**

internal function, vector of colors for discrete plots.

**Usage**

```
c25
```

**Format**

An object of class character of length 25.

---

cast_confusion	<i>Generate confusion matrix for two clusterings, then return summary statistics.</i>
----------------	---

---

### Description

Generate confusion matrix for two clusterings, then return summary statistics.

### Usage

```
cast_confusion(clusters, barcodes)
```

### Arguments

clusters	Data.table. Columns are cell ids and cluster assignment.
barcodes	Data.table. Columns are cell ids and barcode assignment.

### Value

A data.table with summary statistics from the contingency tables, for each barcode, the cluster(s) with the highest number of true positives is returned.

---

clonocluster	<i>Return hybrid clusters for a range of alpha values.</i>
--------------	--

---

### Description

Return hybrid clusters for a range of alpha values.

### Usage

```
clonocluster(
  irl,
  bt,
  alpha = c(0, 0.5, 1),
  beta = 0.1,
  res = 1,
  method = "fast",
  ...
)
```

### Arguments

irl	Matrix. Principal components matrix, output from <code>clonocluster::irlba_wrap</code> .
bt	Data table. Barcode table of two columns cell IDs ("rn") and barcodes ("Barcode").
alpha	Numeric. Alpha parameter or vector of values to iterate over. Ranges from 0 to 1. A value of zero reflects only transcriptome edges, a value of one reflects only barcodes.

beta	Numeric. Exponent on alpha to adjust curve sloping. Default is 0.1.
res	Numeric. Resolution parameter passed to community detection algorithm.
method	c("fast", "index"), how the barcode matrix is made, default is fast, if its going too slow try "index".
...	Additional arguments passed to Seurat::RunModularityClustering.

**Value**

A long format data table with four columns, cell ID ("rn"), alpha value, beta value, and cluster assignment ("Group").

---

clonocluster_model	<i>Return the clonocluster network graph for clustering.</i>
--------------------	--

---

**Description**

Return the clonocluster network graph for clustering.

**Usage**

```
clonocluster_model(alpha, beta = 0.1, m, nm)
```

**Arguments**

alpha	Numeric. Between zero and one, with zero representing complete transcriptome and 1 representing complete barcode graphs.
beta	Numeric. Exponent on alpha to adjust curve sloping. Default is 0.1.
m	Object of class dgCMatrix, the graph of transcriptome edges derived from SNN jaccard index. Usually from Seurat::FindNeighbors.
nm	Output of build_barcode_matrix or build_barcode_matrix_fast.

**Value**

A sparse matrix of class dgCMatrix suitable for Seurat::RunModularityClustering.

---

cw_colors	<i>Initialize vector of plot colors published by Claus Wilke.</i>
-----------	---

---

**Description**

Initialize vector of plot colors published by Claus Wilke.

**Usage**

```
cw_colors
```

**Format**

An object of class character of length 14.

---

dt2m	<i>Convert a data.table to matrix.</i>
------	--

---

**Description**

Convert a data.table to matrix.

**Usage**

```
dt2m(dt)
```

**Arguments**

dt	A data.table to be converted to a matrix.
----	---

**Value**

A matrix from the data.table, with the first table column as rownames of the matrix.

---

engage_warp	<i>Warp principal components and return modified umap.</i>
-------------	--

---

**Description**

Warp principal components and return modified umap.

**Usage**

```
engage_warp(pca, bt, s, ...)
```

**Arguments**

pca	Matrix. Principal components matrix, output from <code>clonocluster::irlba_wrap</code> .
bt	Data table. Barcode table of two columns cell IDs ("rn") and barcodes ("Barcode").
s	Numeric. Warp factor, from 0 to 10.
...	Other arguments passed to <code>umap_matrix</code> .

**Value**

UMAP output from the warped PCA.

---

FindAllMarkers\_Seurat *A wrapper to call Seurat to FindAllMarkers on given groups.*

---

### Description

A wrapper to call Seurat to FindAllMarkers on given groups.

### Usage

```
FindAllMarkers_Seurat(so, clust, method = "roc", ...)
```

### Arguments

so	A Seurat object or path to one.
clust	A table of cellIDs and group assignments.
method	Passed to the test.use argument of Seurat::FindAllMarkers, default is "roc".
...	All other arguments passed to Seurat::FindAllMarkers.

### Value

A data.table of the Seurat::FindAllMarkers output.

---

Find\_Markers\_ROC *Fast AUC calculation wrapper for clusters.*

---

### Description

Fast AUC calculation wrapper for clusters.

### Usage

```
Find_Markers_ROC(dl, cm, n_threads = 1)
```

### Arguments

dl	A data.table of cluster assignments, output from clonocluster, with minimum columns c("rn", "Group", "alpha").
cm	Matrix. A count matrix.
n_threads	Integer. Passed to mc.cores for parallel::mclapply

### Value

A table of marker AUCs and thresholds for all clusters.

---

irlba_wrap	<i>A wrapper to run approximate PCA on a count matrix.</i>
------------	--

---

### Description

A wrapper to run approximate PCA on a count matrix.

### Usage

```
irlba_wrap(gt, npc = 100, seed_use = 42)
```

### Arguments

gt	Matrix. Rows are cell IDs, columns are gene counts.
npc	Integer. Number of PCs to compute with <code>irlba::irlba</code> , default is 100.
seed_use	Integer. RNG seed for PCA, default is 42.

### Value

A matrix with cell as rows, PCs as features/columns.

---

Plot_alluvia	<i>Plot a Sankey Diagram from a table of unique IDs, and two or more columns indicating membership groups to progress through, in order.</i>
--------------	--

---

### Description

Plot a Sankey Diagram from a table of unique IDs, and two or more columns indicating membership groups to progress through, in order.

### Usage

```
Plot_alluvia(
  dl,
  bl,
  title = "",
  xlab = "",
  ylab = "",
  flow_alpha = 0.2,
  node_alpha = 1,
  stack_colors = TRUE,
  border_size = 2,
  label_nodes = TRUE,
  label_size = 2,
  ltype = "label",
  reverse = TRUE,
  cols = ClonoCluster::c25
)
```



**Arguments**

dl	A data table, with three columns, c("rn", "Group", "alpha").
bl	A data table, with two columns, c("rn", "Barcode")
title	Character. Passed to ggtitle.
xlab	Character. Passed to xlab.
ylab	Character. Passed to ylab.
flow_alpha	Numeric. Alpha value for the alluvial ribbons.
node_alpha	Numeric. Alpha value for the strata, passed to geom_rect.
stack_colors	Boolean. Should nodes be represented as stacked colors representing the contributing alluvia? Default TRUE.
border_size	Numeric. Rectangle thickness around the sankey nodes. Default 2.
label_nodes	Boolean. Should clusters be labeled with geom_label? Default TRUE.
label_size	Numeric. size of geom_text labels font.
ltype	One of c("label", "text"). Passed as the geom to annotation. Default is "label"
reverse	Boolean. Default TRUE, return plot colored by clusters and barcodes.
cols	Character. Vector of colors values, default is c25.

**Value**

Returns a ggplot object (or two if reverse == TRUE) of the Sankey Plot.

---

Plot_alluvia_counts	<i>Plot a Sankey Diagram from a table of unique IDs, and two or more columns indicating membership groups to progress through, in order.</i>
---------------------	--

---

**Description**

Plot a Sankey Diagram from a table of unique IDs, and two or more columns indicating membership groups to progress through, in order.

**Usage**

```
Plot_alluvia_counts(
  dl,
  counts,
  title = "",
  xlab = "",
  ylab = "",
  flow_alpha = 0.2,
  node_alpha = 1,
  border_size = 2,
  label_nodes = TRUE,
  label_size = 2,
  ltype = "text",
  col_start = "gray100",
  col_end = "darkblue"
)
```

**Arguments**

dl	A data table, with at least three columns, c("rn", "Group", "alpha"), from clonocluster.
counts	A matrix with one count column and rownames equivalent to "rn".
title	Character. Passed to ggtitle.
xlab	Character. Passed to xlab.
ylab	Character. Passed to ylab.
flow_alpha	Numeric. Alpha value for the alluvial ribbons.
node_alpha	Numeric. Alpha value for the strata, passed to geom_rect.
border_size	Numeric. Rectangle thickness around the sankey nodes. Default 2.
label_nodes	Boolean. Should clusters be labeled with geom_label? Default TRUE.
label_size	Numeric. size of geom_text labels font.
ltype	One of c("label", "text"). Passed as the geom to annotation. Default is "label"
col_start	Character. Starting value for color gradient, default is "gray100".
col_end	Character. Ending value for color gradient, default is "darkblue".

**Value**

Returns a ggplot object of the Sankey Plot, alluvia are directly proportional to counts and node colors are the average of all contributing cells.

---

Plot_alluvia_track	<i>Plot a Sankey Diagram and track a list of cellIDs.</i>
--------------------	---

---

**Description**

Plot a Sankey Diagram and track a list of cellIDs.

**Usage**

```
Plot_alluvia_track(
  dl,
  ids,
  title = "",
  xlab = "",
  ylab = "",
  flow_alpha = 0.2,
  node_alpha = 1,
  border_size = 2,
  label_nodes = TRUE,
  label_size = 2,
  ltype = "text",
  cols = ClonoCluster::cw_colors,
  alluvia_cols = cols,
  col2 = "gray100",
  orientation = "bottom"
)
```

**Arguments**

dl	A data table, with three columns, c("rn", "Group", "alpha"), from clonocluster.
ids	List. A list of character vectors containing rownames in dl (cellIDs) for each group to be tracked.
title	Character. Passed to ggtitle.
xlab	Character. Passed to xlab.
ylab	Character. Passed to ylab.
flow_alpha	Numeric. Alpha value for the alluvial ribbons.
node_alpha	Numeric. Alpha value for the strata, passed to geom_rect.
border_size	Numeric. Rectangle thickness around the sankey nodes. Default 2.
label_nodes	Boolean. Should clusters be labeled with geom_label? Default TRUE.
label_size	Numeric. size of geom_text labels font.
ltype	One of c("label", "text"). Passed as the geom to annotation. Default is "label"
cols	Character. Vector of colors values of length equal to the number of groups to be tracked, node colors.
alluvia_cols	Character. Vector of colors values of length equal to the number of groups to be tracked, alluvium colors, default is same as cols.
col2	Character. Color of untracked cells, default "gray100".
orientation	One of c("top", "bottom"), where colored rectangles will be stacked on the nodes. Default is "bottom".

**Value**

Returns a ggplot object of the Sankey Plot.

---

ROCR_wrap	<i>Fast AUC calculation.</i>
-----------	------------------------------

---

**Description**

Fast AUC calculation.

**Usage**

```
ROCR_wrap(x, y, return_curve = FALSE)
```

**Arguments**

x	Numeric. First vector of values.
y	Numeric. Second vector of values.
return_curve	Logical. Default FALSE. Return the ROC curve values.

**Value**

AUC value (always > 0.5) or ROC curve values.

---

tdt	<i>Flip a count matrix data.table.</i>
-----	--

---

**Description**

Flip a count matrix data.table.

**Usage**

```
tdt(dt)
```

**Arguments**

dt	A data.table to be flipped.
----	-----------------------------

**Value**

A data table that has been transposed

---

ttheme	<i>internal function, plot theming.</i>
--------	---

---

**Description**

internal function, plot theming.

**Usage**

```
ttheme
```

**Format**

An object of class theme (inherits from gg) of length 3.

---

umap_matrix	<i>Run uwot::umap on a matrix using Seurat v2.0 defaults.</i>
-------------	---

---

**Description**

Run uwot::umap on a matrix using Seurat v2.0 defaults.

**Usage**

```
umap_matrix(
  ce,
  metric = "cosine",
  n_neighbors = 30L,
  n_components = 2L,
  min_dist = 0.3,
  learning_rate = 1,
  spread = 1,
  set_op_mix_ratio = 1,
  local_connectivity = 1L,
  repulsion_strength = 1,
  negative_sample_rate = 5L,
  init = "spectral",
  seed_use = 42,
  ...
)
```

**Arguments**

<code>ce</code>	Matrix. Input matrix to reduce.
<code>metric</code>	Character. Default "cosine", other options <code>c("euclidean", "manhattan", "hamming", "correlation", "categorical")</code> .
<code>n_neighbors</code>	Integer. Nearest neighbors to search, default 30.
<code>n_components</code>	Integer. Number of UMAP dimensions to return.
<code>min_dist</code>	Numeric. Minimum distance between points, default 0.3.
<code>learning_rate</code>	Numeric. Initial learning rate for optimization. Default 1.
<code>spread</code>	Numeric. Effective scale of embedded points. Default 1.
<code>set_op_mix_ratio</code>	Numeric. Default is 1, pure fuzzy union.
<code>local_connectivity</code>	Integer. Number of nearest neighbors to consider at local level. Default 1.
<code>repulsion_strength</code>	Numeric. Weighting applied to negative sampling. Default 1.
<code>negative_sample_rate</code>	Integer. Ratio of negative to positive samples to use for optimization of embedding. Default 5.
<code>init</code>	Character. Default "spectral". Type of initialization for coordinates, other options <code>c("normlaplacian", "random", "lvrandom", "laplacian", "pca", "spca", "agspectral")</code> or matrix of coordinates.
<code>seed_use</code>	Numeric. RNG seed passed to <code>set_seed</code> , default is 42.

**Value**

A data table of the input rownames, UMAP components, and seed used.

# Index

## \* datasets

- c25, [3](#)
- cw\_colors, [5](#)
- ttheme, [12](#)

- barcode\_warp, [2](#)
- build\_barcode\_matrix, [2](#)
- build\_barcode\_matrix\_fast, [3](#)

- c25, [3](#)
- cast\_confusion, [4](#)
- clonocluster, [4](#)
- clonocluster\_model, [5](#)
- cw\_colors, [5](#)

- dt2m, [6](#)

- engage\_warp, [6](#)

- Find\_Markers\_ROC, [7](#)
- FindAllMarkers\_Seurat, [7](#)

- irlba\_wrap, [8](#)

- Plot\_alluvia, [8](#)
- Plot\_alluvia\_counts, [9](#)
- Plot\_alluvia\_track, [10](#)

- ROCR\_wrap, [11](#)

- tdt, [12](#)
- ttheme, [12](#)

- umap\_matrix, [12](#)