

# Package ‘demuxSNP’

January 19, 2026

**Title** scRNASeq demultiplexing using cell hashing and SNPs

**Version** 1.8.0

## Description

This package assists in demultiplexing scRNAseq data using both cell hashing and SNPs data. The SNP profile of each group is learned using high confidence assignments from the cell hashing data.

Cells which cannot be assigned with high confidence from the cell hashing data are assigned to their most similar group based on their SNPs.

We also provide some helper function to optimise SNP selection, create training data and merge SNP data into the SingleCellExperiment framework.

**URL** <https://github.com/michaelplynych/demuxSNP>

**BugReports** <https://github.com/michaelplynych/demuxSNP/issues>

**License** GPL-3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Depends** R (>= 4.3.0), SingleCellExperiment, VariantAnnotation, ensemblDb

**Imports** MatrixGenerics, BiocGenerics, class, Seqinfo, IRanges, Matrix, SummarizedExperiment, demuxmix, methods, KernelKnn, dplyr

**Suggests** knitr, rmarkdown, ComplexHeatmap, viridisLite, ggpubr, dittoSeq, EnsDb.Hsapiens.v86, BiocStyle, RefManageR, testthat (>= 3.0.0), Seurat

**biocViews** Classification, SingleCell

**VignetteBuilder** knitr

**LazyData** false

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/demuxSNP>

**git\_branch** RELEASE\_3\_22

**git\_last\_commit** 0c933a4

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.22

**Date/Publication** 2026-01-19

**Author** Michael Lynch [aut, cre] (ORCID: <https://orcid.org/0000-0001-9535-6461>),  
Aedin Culhane [aut] (ORCID: <https://orcid.org/0000-0002-1395-9734>)

**Maintainer** Michael Lynch <michael.lynch@ul.ie>

## Contents

add_snps . . . . .	2
commonvariants_1kgenomes_subset . . . . .	3
common_genes . . . . .	3
high_conf_calls . . . . .	4
multiplexed_scrnaseq_sce . . . . .	4
reassign . . . . .	5
reassign_balanced . . . . .	6
reassign_centroid . . . . .	7
reassign_jaccard . . . . .	8
subset_vcf . . . . .	9
vartrix_consensus_snps . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

add_snps	<i>Add SNPs to SingleCellExperiment object</i>
----------	--

---

## Description

Add SNPs to SingleCellExperiment object

## Usage

```
add_snps(sce, mat, thresh = 0.8)
```

## Arguments

sce	object of class SingleCellExperiment
mat	object of class matrix, output from VarTrix in 'consensus' mode (default)
thresh	threshold presence of SNP, defaults to 0.8

## Value

Updated SingleCellExperiment object with snps in altExp slot

## Examples

```
data(multiplexed_scrnaseq_sce, vartrix_consensus_snps)
multiplexed_scrnaseq_sce <- add_snps(sce = multiplexed_scrnaseq_sce,
mat = vartrix_consensus_snps,
thresh = 0.8)
```

---

```
commonvariants_1kgenomes_subset
```

*Sample vcf file*

---

**Description**

VCF file containing SNPs from a subset of the 1k Genomes common variants HG38 genome build.

**Usage**

```
data(commonvariants_1kgenomes_subset)
```

**Format**

An object of class CollapsedVCF with 2609 rows and 0 columns.

**Value**

```
commonvariants_1kgenomes_subset:
```

An object of class CollapsedVcf

**Source**

[https://cellsnp-lite.readthedocs.io/en/latest/snp\\_list.html](https://cellsnp-lite.readthedocs.io/en/latest/snp_list.html)

---

```
common_genes
```

*Return a character vector of top n most frequent genes from a Single-CellExperiment object.*

---

**Description**

Returns a character vector of the top n most frequently expressed genes from the counts of the SingleCellExperiment object. Expression is based on having a count > 0 in a given cell.

**Usage**

```
common_genes(sce, n = 100)
```

**Arguments**

```
sce
```

a SingleCellExperiment object

```
n
```

number of genes to be returned. Defaults to n=100.

**Value**

character vector of n most frequently expressed genes.

**Examples**

```
data(multiplexed_scrnaseq_sce)
multiplexed_scrnaseq_sce <- common_genes(multiplexed_scrnaseq_sce)
```

---

high_conf_calls	<i>Run demuxmix to determine high-confidence calls</i>
-----------------	--

---

**Description**

Run demuxmix to determine high-confidence calls

**Usage**

```
high_conf_calls(sce, assay = "HTO", pacpt = 0.95)
```

**Arguments**

sce	Object of class SingleCellExperiment with HTO (or similar) altExp assay
assay	Name of altExp for cell hashing counts to be retrieved from
pacpt	acceptance probability for demuxmix model

**Value**

Updated SingleCellExperiment object with logical vector indicating training data, data to be classified (all cells) and assigned labels for all cells.

**Examples**

```
data(multiplexed_scrnaseq_sce)
multiplexed_scrnaseq_sce <- high_conf_calls(multiplexed_scrnaseq_sce)
```

---

multiplexed_scrnaseq_sce	<i>SingleCellExperiment object containing multiplexed RNA and HTO data from six biological smamples</i>
--------------------------	---

---

**Description**

Example SingleCellExperiment object containing demultiplexed scRNAseq data from six donors, used throughout and built upon in demuxSNP workflow.

**Usage**

```
data(multiplexed_scrnaseq_sce)
```

**Format**

An object of class SingleCellExperiment with 259 rows and 2000 columns.

**Value**

```
multiplexed_scrnaseq_sce:
An object of class SingleCellExperiment
```

---

reassign	<i>Reassign cells using knn</i>
----------	---------------------------------

---

## Description

k-nearest neighbour classification of cells. Training data is intended to be labels of cells confidently called using cell hashing based methods and their corresponding SNPs. Prediction data can be remaining cells but can also include the training data. Doublets are simulated by randomly combining 'd' SNP profiles from each grouping combination.

## Usage

```
reassign(
  sce,
  k = 10,
  d = 10,
  train_cells = sce$train,
  predict_cells = sce$predict
)
```

## Arguments

sce	object of class SingleCellExperiment
k	number of neighbours used in knn, defaults to 10
d	number of doublets per group combination to simulate, defaults to 10
train_cells	logical vector specifying which cells to use to train classifier
predict_cells	logical vector specifying which cells to classify

## Value

A SingleCellExperiment with updated group assignments called 'knn'

## Examples

```
data(multiplexed_scrnaseq_sce, vartrix_consensus_snps)
multiplexed_scrnaseq_sce <- high_conf_calls(multiplexed_scrnaseq_sce)
multiplexed_scrnaseq_sce <- add_snps(sce = multiplexed_scrnaseq_sce,
  mat = vartrix_consensus_snps,
  thresh = 0.8)
multiplexed_scrnaseq_sce <- reassign(sce = multiplexed_scrnaseq_sce, k = 10)
```

---

reassign_balanced	<i>Reassign cells using balanced knn with jaccard distance</i>
-------------------	--

---

## Description

k-nearest neighbour classification of cells. Training data is intended to be labels of cells confidently called using cell hashing based methods and their corresponding SNPs. Prediction data can be remaining cells but can also include the training data. Doublets are simulated by randomly combining 'd' SNP profiles from each grouping combination.

## Usage

```
reassign_balanced(
  sce,
  k = 20,
  d_prop = 0.5,
  train_cells = sce$train,
  predict_cells = sce$predict,
  nmin = 50,
  n = NULL
)
```

## Arguments

sce	object of class SingleCellExperiment
k	number of neighbours used in knn, defaults to 10
d_prop	determines number of doublets simulated d, as a proportions of n (specified or calculated)
train_cells	logical vector specifying which cells to use to train classifier
predict_cells	logical vector specifying which cells to classify
nmin	min n per class (where available)
n	number of cells per group (otherwise will be calculated from data)

## Value

A SingleCellExperiment with updated group assignments called 'knn\_balanced'

## Examples

```
data(multiplexed_scrnaseq_sce, vartrix_consensus_snps)
multiplexed_scrnaseq_sce <- high_conf_calls(multiplexed_scrnaseq_sce)
multiplexed_scrnaseq_sce <- add_snps(sce = multiplexed_scrnaseq_sce,
  mat = vartrix_consensus_snps,
  thresh = 0.8)
multiplexed_scrnaseq_sce <- reassign_balanced(sce = multiplexed_scrnaseq_sce, k = 10, d=0.5)
```

---

reassign_centroid	<i>Reassign cells based on SNPs</i>
-------------------	-------------------------------------

---

## Description

Reassign cells based on SNPs

## Usage

```
reassign_centroid(  
  sce,  
  train_cells = sce$train,  
  predict_cells = sce$predict,  
  labels = sce$labels,  
  min_cells = 30,  
  key = "Hashtag"  
)
```

## Arguments

sce	SingleCellExperiment object
train_cells	logical, cells to be used for training
predict_cells	logical, cells to be used for prediction
labels	provisional cell labels
min_cells	minimum coverage (number of cells with read at SNP location) for SNP to be used for classification.
key	unique key in naming of singlet groups used with grep to remove doublet/negative/uncertain labels

## Value

character vector containing reassignments

## Examples

```
data(multiplexed_scrnaseq_sce, vartrix_consensus_snps)  
multiplexed_scrnaseq_sce <- high_conf_calls(multiplexed_scrnaseq_sce)  
multiplexed_scrnaseq_sce <- add_snps(sce = multiplexed_scrnaseq_sce,  
  mat = vartrix_consensus_snps,  
  thresh = 0.8)  
multiplexed_scrnaseq_sce<-reassign_centroid(multiplexed_scrnaseq_sce)
```

---

reassign_jaccard	<i>Reassign cells using knn with jaccard distance</i>
------------------	---

---

## Description

k-nearest neighbour classification of cells. Training data is intended to be labels of cells confidently called using cell hashing based methods and their corresponding SNPs. Prediction data can be remaining cells but can also include the training data. Doublets are simulated by randomly combining 'd' SNP profiles from each grouping combination.

## Usage

```
reassign_jaccard(
  sce,
  k = 10,
  d = 10,
  train_cells = sce$train,
  predict_cells = sce$predict
)
```

## Arguments

sce	object of class SingleCellExperiment
k	number of neighbours used in knn, defaults to 10
d	number of doublets per group combination to simulate, defaults to 10
train_cells	logical vector specifying which cells to use to train classifier
predict_cells	logical vector specifying which cells to classify

## Value

A SingleCellExperiment with updated group assignments called 'knn\_jaccard'

## Examples

```
data(multiplexed_scrnaseq_sce, vartrix_consensus_snps)
multiplexed_scrnaseq_sce <- high_conf_calls(multiplexed_scrnaseq_sce)
multiplexed_scrnaseq_sce <- add_snps(sce = multiplexed_scrnaseq_sce,
  mat = vartrix_consensus_snps,
  thresh = 0.8)
multiplexed_scrnaseq_sce <- reassign(sce = multiplexed_scrnaseq_sce, k = 10)
```



---

subset_vcf	<i>Subset common variants vcf file to only SNPs seen in 'top_genes'</i>
------------	---

---

**Description**

Subset common variants vcf file to only SNPs seen in 'top\_genes'

**Usage**

```
subset_vcf(vcf, top_genes, ensdb)
```

**Arguments**

vcf	object of class CollapsedVCF
top_genes	output from 'common_genes' function, alternatively character vector containing custom gene names.
ensdb	object of class EnsDb corresponding to organism, genome of data

**Value**

object of class CollapsedVCF containing subset of SNPs from supplied vcf seen in commonly expressed genes

**Examples**

```
data(multiplexed_scrnaseq_sce, commonvariants_1kgenomes_subset)
top_genes <- common_genes(multiplexed_scrnaseq_sce)
ensdb <- EnsDb.Hsapiens.v86::EnsDb.Hsapiens.v86
small_vcf <- subset_vcf(commonvariants_1kgenomes_subset, top_genes, ensdb)
```

---

vartrix_consensus_snps	<i>Sample VarTrix output</i>
------------------------	------------------------------

---

**Description**

A sample output from VarTrix corresponding to the sce SingleCellExperiment object for a subset of SNPs located in well observed genes.

**Usage**

```
data(vartrix_consensus_snps)
```

**Format**

An object of class matrix (inherits from array) with 2542 rows and 2000 columns.

**Value**

vartrix\_consensus\_snps:  
An object of class matrix

# Index

## \* datasets

commonvariants\_1kgenomes\_subset, [3](#)

multiplexed\_scrnaseq\_sce, [4](#)

vartrix\_consensus\_snps, [9](#)

add\_snps, [2](#)

common\_genes, [3](#)

commonvariants\_1kgenomes\_subset, [3](#)

high\_conf\_calls, [4](#)

multiplexed\_scrnaseq\_sce, [4](#)

reassign, [5](#)

reassign\_balanced, [6](#)

reassign\_centroid, [7](#)

reassign\_jaccard, [8](#)

subset\_vcf, [9](#)

vartrix\_consensus\_snps, [9](#)