Package 'SexSeek'

November 3, 2026

Title Estimate Genetic Sex from Expression Counts

Version 0.1.0

Description Infers the genetic sex of samples, donors or cells from bulk or single-cell RNA sequencing counts. Accepts dense or sparse matrices, delimited text files, 10x Genomics outputs, and 'Seurat', 'SingleCellExperiment' and 'SummarizedExperiment' objects.

License MIT + file LICENSE

URL https://sexseek.saketlab.org, https://github.com/saketlab/SexSeek

BugReports https://github.com/saketlab/SexSeek/issues

Encoding UTF-8

Language en-GB

Depends R (>= 4.1)

Imports ggplot2, Matrix, methods, stats, utils

Suggests genevintage (>= 0.1.3), hdf5r, knitr, rmarkdown, SeuratObject, SingleCellExperiment, SummarizedExperiment, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Saket Choudhary [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-5202-7633>)

Maintainer Saket Choudhary <saketc@iitb.ac.in>

Repository CRAN

Date/Publication 2026-11-03 15:50:18 UTC

Contents

CompareSex . . . 2
DetectSpecies . . . 3
EstimateSex.list . . . 3
ExampleData . . . 6
GametologPairs . . . 7
MarkerCellFraction . . . 7
PanelFor . . . 8
PlotCellFraction . . . 9
PlotSexEvidence . . . 9
Read10xDir . . . 10
Read10xH5 . . . 11
ResolveSpecies . . . 12
SexChromosomeGenes . . . 12
SexPanels . . . 13
SexSpecies . . . 14
SummariseComparison . . . 14

Index **15**

`CompareSex` *Compare calls with recorded sex*

Description

Joins `EstimateSex()` calls to the sex recorded for each unit. Only `female`/`male` (or `F`/`M`, any case) are scored; other labels are kept as recorded.

Usage

```
CompareSex(calls, recorded)
```

Arguments

`calls`	Output of `EstimateSex()`.
`recorded`	Recorded sex as a character vector named by unit, matching `calls$unit`.

Value

`calls` with three columns added: `recorded` (as given), `recorded_sex` (`"female"`, `"male"` or `NA`) and `agrees`: `TRUE` when the verdict is the recorded sex, `FALSE` when it is the other sex, and `NA` when the call abstains (`uncertain`, `possible_mixed`, `unknown`) or no usable sex is recorded.

Examples

```
ex <- ExampleData("mouse_bulk")
calls <- EstimateSex(ex$counts, species = ex$species, group = colnames(ex$counts))
cmp <- CompareSex(calls, ex$recorded)
table(recorded = cmp$recorded_sex, call = cmp$verdict)
```

DetectSpecies *Detect species from gene identifiers or symbols*

Description

Tries stable-ID prefixes first; they are unambiguous and cost one regex per species. Falls back to scoring the input symbols against each species' panel symbols, which is weaker: panels overlap heavily between close relatives, so the returned confidence is what tells you whether to trust it.

Usage

```
DetectSpecies(features)
```

Arguments

features	A character vector of gene identifiers or symbols, typically rownames(counts).

Value

A list with species, confidence (high, medium, low) and method (id_prefix, symbol, or none). species is NULL when nothing matched.

Examples

```
DetectSpecies(c("ENSMUSG00000086503", "ENSMUSG00000069045"))
DetectSpecies(c("ENSG00000229807.13", "ENSG00000012817"))
```

EstimateSex.list *Estimate genetic sex from expression counts*

Description

Scores two complementary lines of evidence, expression of sex-specific chromosome markers and expression of the X-inactivation marker, and reports the state they jointly imply. It does not force a male/female answer.

Usage

```
## S3 method for class 'list'
EstimateSex(x, ..., group = NULL, per_cell = FALSE)

EstimateSex(x, ...)

## Default S3 method:
EstimateSex(
  x,
  species = NULL,
  group = NULL,
  per_cell = FALSE,
  annotation = NULL,
  input_scale = c("auto", "counts", "normalised"),
  ...
)

## S3 method for class 'character'
EstimateSex(x, ...)

## S3 method for class 'Seurat'
EstimateSex(
  x,
  species = NULL,
  group = NULL,
  per_cell = FALSE,
  assay = NULL,
  annotation = NULL,
  input_scale = c("auto", "counts", "normalised"),
  ...
)

## S3 method for class 'SingleCellExperiment'
EstimateSex(
  x,
  species = NULL,
  group = NULL,
  per_cell = FALSE,
  assay = "counts",
  annotation = NULL,
  input_scale = c("auto", "counts", "normalised"),
  ...
)

## S3 method for class 'SummarizedExperiment'
EstimateSex(
  x,
  species = NULL,
```

```
  group = NULL,
  per_cell = FALSE,
  assay = "counts",
  annotation = NULL,
  input_scale = c("auto", "counts", "normalised"),
  ...
)
```

Arguments

x	A counts matrix, sparse matrix, data frame, file path, or a supported container (Seurat, SingleCellExperiment).
...	Passed to methods.
group	Optional grouping of columns into units, as a vector the length of `ncol(x)` or the name of a column in the object's metadata. Defaults to treating the whole matrix as one unit.
per_cell	Report one row per column instead of aggregating.
species	Species name, scientific or common. Detected from the gene identifiers when `NULL`.
annotation	Optional sex-chromosome table from `SexChromosomeGenes()`, or `"genevintage"` to fetch the newest annotation for the resolved species. `NULL` uses shipped markers without downloading. A supplied table must belong to the input species; use the full chromosome inventory, including zero-expression genes. Curated markers absent from it are excluded.
input_scale	`"auto"` checks every stored value for fractional counts; `"counts"` requires integer-like values; `"normalised"` explicitly declares normalised or estimated expression and caps confidence even if rounded. Integer-like values alone cannot establish raw-count provenance.
assay	Assay to read counts from. Defaults to the object's active or first assay.

Details

A named list of matrices is scored one element per unit; elements may have different gene lists and are never zero-padded. `group` is not accepted for a list. With `per_cell = TRUE` each unit is `<element>:<column>`.

Absence of Y signal is **not** evidence of female: loss of Y in aged male blood, shallow sequencing and incomplete annotation all look identical to a female sample. Most species in the shipped panel have no annotated X-inactivation marker at all, so for those a female verdict is never available and the honest answer is `uncertain`.

For single-cell input the default is to aggregate, and the biological unit is the **donor**, not the library. Pass `group` for anything that may hold more than one donor. A male cell can carry zero Y UMIs through dropout alone, and ambient RNA in a mixed-sex run carries Y transcripts into female droplets, which is a direct false-male mechanism. `per_cell = TRUE` is therefore **unvalidated**: it has not been tested against cell-level donor labels or doublet calls, and its output should be read as evidence, not as a call.

Value

A data frame, one row per unit, with verdict, heuristic confidence, marker CPM, gametolog and inactivation fractions, depth and diagnostic flags. Confidence categories are not calibrated probabilities. `inact_measured` is `FALSE` when the species panel has no inactivation marker or the input lacks its row; `inact_score` is then 0 without being measured. `per_cell = TRUE` adds `per_cell_unvalidated` and caps confidence at medium.

Examples

```
path <- system.file("extdata", "example_counts.tsv.gz", package = "SexSeek")

# A file path works directly; species is detected from the gene IDs.
# Without `group` the whole matrix is one unit: two donors mixed.
EstimateSex(path)[, c("unit", "verdict", "flags")]

# One unit per sample.
counts <- as.matrix(read.delim(path, row.names = 1))
res <- EstimateSex(counts, group = colnames(counts))
res[, c("unit", "verdict", "confidence", "gametolog_frac", "inact_frac")]
```

ExampleData *Example datasets with recorded sex*

Description

Excerpts of public GEO datasets whose sample records state each donor's sex. Each keeps the rows `EstimateSex()` scores for the species plus an `OTHER` row with the rest of each library, so calls match the full matrix. Excerpts are too small for species detection; pass `species`.

Usage

```
ExampleData(dataset = NULL)
```

Arguments

`dataset`	A name from `ExampleData()$dataset`. `NULL` lists them.

Value

With `dataset = NULL`, a data frame of the available datasets. Otherwise a list with `dataset`, `study`, `species`, `assay`, `tissue`, `counts` (a matrix; a named list of sparse matrices, one per donor, for single-cell; transcript rows for `"mouse_tx"`), `recorded` (the recorded sex named by sample, as written in the record), `samples` (sample accession, title and recorded sex) and `tx2gene` (the transcript-to-gene table for `"mouse_tx"`, otherwise `NULL`).

Examples

```
ExampleData()
ex <- ExampleData("rat_bulk")
dim(ex$counts)
ex$recorded
```

`GametologPairs` *X/Y gametolog pairs*

Description

Each row pairs a Y gene with its X-linked homolog. Scoring uses complete pairs from the selected marker tier. A common library-size multiplier cancels in Y/(Y+X), but gene-specific capture, length, mapping and tissue effects remain. The resulting fraction requires empirical calibration.

Usage

```
GametologPairs()
```

Value

A data frame with one row per pair: `scientific_name`, the Y and X gene IDs and symbols, and the panel tier of the Y gene.

Examples

```
gp <- GametologPairs()
gp[gp$scientific_name == "Homo sapiens", c("y_gene_name", "x_gene_name")]
```

`MarkerCellFraction` *Fraction of cells carrying any sex-specific marker count*

Description

For each unit, the share of columns (cells) with at least one count across the species' scoring-tier sex-specific markers (Y in XY species, W in ZW species).

Usage

```
MarkerCellFraction(x, species, group = NULL)

## S3 method for class 'list'
MarkerCellFraction(x, species, group = NULL)

## Default S3 method:
MarkerCellFraction(x, species, group = NULL)
```

Arguments

x	Counts with cells in columns (matrix, sparse matrix, data frame or file path), or a named list of them, one per unit.
species	Species name, scientific or common.
group	For a matrix, the unit of each column. NULL treats the whole matrix as one unit.

Value

A data frame with unit, n_cells and fraction.

Examples

```
ex <- ExampleData("mouse_sc")
MarkerCellFraction(ex$counts, ex$species)
```

PanelFor *The panel rows for one species*

Description

Defaults to the core tier, which is what scoring uses: a curated set of uniquely mappable, somatic-tissue markers. The full inventory carries every gene on the sex chromosomes and is far too permissive to call from: ampliconic and testis-restricted Y families in particular fire on tissue, not on genotype.

Usage

```
PanelFor(species, tier = "core")
```

Arguments

species	A scientific name, as it appears in SexSpecies().
tier	Which tiers to keep. NULL returns every row.

Value

A data frame of panel rows.

Examples

```
PanelFor("Homo sapiens")[, c("gene_name", "role", "tier")]
PanelFor("Gallus gallus")[, c("gene_name", "role", "tier")]
```

PlotCellFraction *Plot the fraction of cells carrying a marker count*

Description

Bars of `MarkerCellFraction()` per unit. When `recorded` is given, each bar is topped by the recorded sex (circle female, triangle male). The result is a ggplot, so scales and theme can be replaced.

Usage

```
PlotCellFraction(x, species, recorded = NULL, group = NULL, main = NULL)
```

Arguments

`x`	Counts with cells in columns (matrix, sparse matrix, data frame or file path), or a named list of them, one per unit.
`species`	Species name, scientific or common.
`recorded`	Optional recorded sex named by unit; see `CompareSex()`.
`group`	For a matrix, the unit of each column. `NULL` treats the whole matrix as one unit.
`main`	Plot title.

Value

A ggplot whose `data` is the `MarkerCellFraction()` table, with `recorded_sex` when `recorded` was given.

Examples

```
ex <- ExampleData("mouse_sc")
PlotCellFraction(ex$counts, ex$species, ex$recorded, main = ex$study)
```

PlotSexEvidence *Plot the evidence behind each call*

Description

One point per unit: sex-specific marker expression against inactivation marker expression, both as log10(CPM + 1). Colour is the call; the shape is the recorded sex when it is given (circle female, triangle male, cross when not recorded). When no unit has a measured inactivation marker (none in the species panel, or its row absent from the input), the plot has one axis, marker expression per unit.

Usage

```
PlotSexEvidence(calls, recorded = NULL, main = NULL)
```

Arguments

calls	Output of EstimateSex() or CompareSex().
recorded	Optional recorded sex named by unit; see CompareSex().
main	Plot title.

Details

The result is a ggplot, so scales and theme can be replaced, e.g. + ggplot2::scale_shape_manual(values = c(female = 1, male = 2)).

Value

A ggplot. Its data is calls, with the recorded columns when recorded was given.

Examples

```
ex <- ExampleData("mouse_bulk")
calls <- EstimateSex(ex$counts, species = ex$species, group = colnames(ex$counts))
PlotSexEvidence(calls, ex$recorded, main = ex$study)
```

Read10xDir *Read a 10x MatrixMarket triplet from a directory*

Description

Expects matrix.mtx, a barcode file and a feature file, each optionally gzipped, as CellRanger writes them. Matrix::readMM does the parsing, so this costs no dependency beyond what the package already needs.

Usage

```
Read10xDir(dir, feature_column = 1L)
```

Arguments

dir	Directory holding the triplet.
feature_column	Which column of the feature file to use as rownames. CellRanger v3 writes id, symbol, type; column 1 is the Ensembl identifier, which is what species detection and panel matching want.

Value

A sparse matrix with gene rownames and barcode colnames.

Examples

```
dir <- file.path(tempdir(), "tenx")
dir.create(dir, showWarnings = FALSE)
m <- Matrix::sparseMatrix(i = c(1, 2), j = c(1, 2), x = c(5, 3), dims = c(2, 2))
Matrix::writeMM(m, file.path(dir, "matrix.mtx"))
writeLines(
  c("ENSG00000229807\tXIST", "ENSG00000012817\tKDM5D"),
  file.path(dir, "features.tsv")
)
writeLines(c("AAAC-1", "AAAG-1"), file.path(dir, "barcodes.tsv"))
Read10xDir(dir)
unlink(dir, recursive = TRUE)
```

Read10xH5 *Read a 10x HDF5 counts file*

Description

Handles both the CellRanger v2 layout (a single named group holding the genome) and the v3 layout (a `matrix` group with a `features` subgroup). Needs the `hdf5r` package, which is a suggested dependency.

Usage

```
Read10xH5(path, feature_column = c("id", "name"))
```

Arguments

`path`	Path to a `.h5` file.
`feature_column`	For v3 files, `"id"` for Ensembl identifiers or `"name"` for symbols. Identifiers are preferred; symbols cannot distinguish species.

Value

A sparse matrix with gene rownames and barcode colnames.

Examples

```
path <- tempfile(fileext = ".h5")
h5 <- hdf5r::H5File$new(path, mode = "w")
g <- h5$create_group("matrix")
g[["data"]] <- c(5L, 3L)
g[["indices"]] <- c(0L, 1L)
g[["indptr"]] <- c(0L, 1L, 2L)
g[["shape"]] <- c(2L, 2L)
g[["barcodes"]] <- c("AAAC-1", "AAAG-1")
f <- g$create_group("features")
f[["id"]] <- c("ENSG00000229807", "ENSG00000012817")
f[["name"]] <- c("XIST", "KDM5D")
```

```
h5$close_all()
Read10xH5(path)
unlink(path)
```

`ResolveSpecies` *Resolve a user-supplied species name to a scientific name*

Description

Accepts a scientific name, a common name, or any alias in the shipped alias table, in any case.

Usage

```
ResolveSpecies(name)
```

Arguments

`name`	A species name.

Value

The scientific name, or `NULL` if unrecognised.

Examples

```
ResolveSpecies("mouse")
ResolveSpecies("CHICKEN")
ResolveSpecies("Danio rerio")
```

`SexChromosomeGenes` *Retrieve sex-chromosome annotations with genevintage*

Description

Returns a chromosome inventory, not a diagnostic marker panel. Pass the result to `EstimateSex(annotation = ...)` to reconcile curated markers against this annotation. Fetch once and reuse across datasets.

Usage

```
SexChromosomeGenes(
  species,
  release = NULL,
  assembly = NULL,
  source = NULL,
  mapping = NULL
)
```

Arguments

species	Scientific or common species name.
release, assembly, source	Annotation selection passed to genevintage::sex_genes(). Pin these for reproducible analyses.
mapping	Optional local annotation with id, name, chr, biotype columns. Supplying this avoids downloads.

Value

A data frame with id, name, chr, and biotype columns.

Examples

```
# A local mapping avoids any download.
mapping <- data.frame(
  id = c("ENSG00000229807", "ENSG00000012817", "ENSG00000075624"),
  name = c("XIST", "KDM5D", "ACTB"),
  chr = c("X", "Y", "7"),
  biotype = c("lncRNA", "protein_coding", "protein_coding")
)
SexChromosomeGenes("human", mapping = mapping)
```

SexPanels *Sex-linked gene panels for every supported species*

Description

One row per sex-linked gene. role is one of Y, X, W, Z, inactivation (XIST, RSX) or male_specific (roX1, roX2).

Usage

```
SexPanels()
```

Value

A data frame with one row per gene and columns including scientific_name, gene_id, gene_name, chr, role and tier.

Examples

```
panels <- SexPanels()
table(panels$role)
```

SexSpecies *Supported species and their sex-determination systems*

Description

Supported species and their sex-determination systems

Usage

```
SexSpecies()
```

Value

A data frame with one row per species, including its sex-determination `system`, `heterogametic` sex and panel `status`.

Examples

```
head(SexSpecies()[, c("scientific_name", "common_name", "system", "status")])
```

SummariseComparison *Count agreements with recorded sex*

Description

Count agreements with recorded sex

Usage

```
SummariseComparison(x)
```

Arguments

x	Output of `CompareSex()`, or a named list of them.

Value

One row per comparison: `samples`, and how many calls `agree`, `disagree` or `abstain` (including samples with no usable recorded sex).

Examples

```
ex <- ExampleData("rat_bulk")
calls <- EstimateSex(ex$counts, species = ex$species, group = colnames(ex$counts))
SummariseComparison(CompareSex(calls, ex$recorded))
```

Index

CompareSex, 2
CompareSex(), *9*, *10*, *14*

DetectSpecies, 3

EstimateSex *(*EstimateSex.list*)*, 3
EstimateSex(), *2*, *6*, *10*
EstimateSex.list, 3
ExampleData, 6

GametologPairs, 7
genevintage::sex_genes(), *13*

MarkerCellFraction, 7
MarkerCellFraction(), *9*

PanelFor, 8
PlotCellFraction, 9
PlotSexEvidence, 9

Read10xDir, 10
Read10xH5, 11
ResolveSpecies, 12

SexChromosomeGenes, 12
SexChromosomeGenes(), *5*
SexPanels, 13
SexSpecies, 14
SexSpecies(), *8*
SummariseComparison, 14