

Package: OrnTx (via r-universe)

October 15, 2026

Title Expression Atlas for Ornamental Plant Transcriptomics

Version 0.99.2

Description The first R/Bioconductor package for ornamental plant transcriptomics. Provides a complete RNA-seq pipeline (download, QC, alignment, import), a curated *Rosa chinensis* expression atlas, a 64-gene floriculture trait database anchored to *Arabidopsis* orthologs, multi-tool import support (STAR, featureCounts, HTSeq, Salmon), auto-detection of three *Rosa chinensis* gene ID formats, and cross-species validation on *Tagetes erecta*.

License GPL-3

URL <https://github.com/lashikakarnal/OrnTx>

BugReports <https://github.com/lashikakarnal/OrnTx/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.4.0)

Imports SummarizedExperiment, S4Vectors, ggplot2, tidyr, stats, utils

Suggests DESeq2, edgeR, dplyr, methods, testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

biocViews RNASeq, GeneExpression, DifferentialExpression, Transcriptomics, Visualization, DataImport, Annotation, Alignment

LazyData false

Config/pak/sysreqs libicu-dev zlib1g-dev

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

Date/Publication 2026-10-15 07:00:08 UTC

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

RemoteRef HEAD

RemoteSha 34ba0effa903e6d2d931389fe5156caa4aeab942

Contents

alignReads	2
annotateDEResults	3
annotateGenes	4
annotateWithNCBI	5
annotateWithTraits	5
buildAtlas	6
buildGenomeIndex	7
buildRosaAtlas	8
checkTools	9
downloadSRA	9
findOrthologs	10
floriculture_traits	11
getFlowerGenes	12
getRosaAnnotation	12
importCountMatrix	13
importCustomAtlas	14
importRosaCounts	16
marigold_carotenoid_genes	17
normalizeCounts	17
ornTheme	18
plotAtlas	18
plotPCA	19
plotVolcano	20
rosa_annotation	21
rosa_example	22
rose_ncbi_annotation	22
runDE	23
runFullPipeline	24
runQC	25
standardizeStage	26
Index	27

alignReads

Align RNA-seq reads using STAR

Description

Aligns FASTQ files to a reference genome using STAR and generates ReadsPerGene.out.tab count files compatible with importCustomAtlas(). Requires STAR to be installed.

Usage

```
alignReads(
  fastq_dir,
  index_dir,
  output_dir = "aligned",
  paired = TRUE,
  threads = 16L,
  ram_gb = 60,
  strandedness = "reverse"
)
```

Arguments

fastq_dir	Directory containing cleaned FASTQ files.
index_dir	Path to STAR genome index directory.
output_dir	Directory for alignment output.
paired	Logical. TRUE for paired-end (default).
threads	Threads per sample (default 16).
ram_gb	RAM for BAM sorting in GB (default 60).
strandedness	Library strandedness (default "reverse").

Value

Invisibly returns paths to count files.

Examples

```
checkTools()
## Not run:
alignReads(
  fastq_dir = "clean_reads/",
  index_dir = "star_index/",
  output_dir = "aligned/"
)

## End(Not run)
```

annotatedDEResults	<i>Annotate DE results</i>
--------------------	----------------------------

Description

Adds locus names to DE results from runDE().

Usage

```
annotateDEResults(de_results, species = "Rosa chinensis")
```

Arguments

`de_results` data.frame from `runDE()` with a `gene_id` column.
`species` Character. Default "Rosa chinensis".

Value

data.frame with `locus_name` and `gene_biotype` columns added.

Examples

```
data("rosa_example", package = "OrnTx")
de <- runDE(rosa_example,
            design = ~ tissue_simple,
            contrast = c("tissue_simple", "petal", "abscission zone"))
head(annotateDEResults(de))
```

annotateGenes	<i>Annotate gene IDs with Rosa chinensis locus information</i>
---------------	--

Description

Maps generic gene IDs (`gene0`, `gene1`, etc.) from the RchiOBHm-V2 reference genome to official *Rosa chinensis* locus names and gene biotype information.

Usage

```
annotateGenes(gene_ids, species = "Rosa chinensis")
```

Arguments

`gene_ids` Character vector of gene IDs to annotate (e.g., `c("gene24707", "gene47555")`).
`species` Character. Currently supports "Rosa chinensis".

Value

A data.frame with columns:

- `gene_id` — input gene ID
- `locus_name` — official RchiOBHm-V2 locus name
- `gene_biotype` — gene type (protein_coding, lncRNA, etc.)

Examples

```
ann <- annotateGenes(c("gene10715", "gene21520", "gene37167"))
print(ann)
```

annotateWithNCBI	<i>Annotate gene IDs using NCBI functional annotation</i>
------------------	---

Description

Adds functional product descriptions to DE results when gene IDs are in NCBI LOC format (e.g., LOC112190511). This complements `annotateGenes()` which works with `RchiOBHm` locus names.

Usage

```
annotateWithNCBI(de_results)
```

Arguments

`de_results` A `data.frame` with a `gene_id` column containing NCBI LOC gene identifiers.

Value

The input `data.frame` with product column added.

Examples

```
data(rose_ncbi_annotation)
head(rose_ncbi_annotation)

data(rosa_ncbi_annotation)
de <- data.frame(gene_id = rose_ncbi_annotation$ncbi_gene_id[1:5],
                 log2FoldChange = rnorm(5))
annotateWithNCBI(de)
```

annotateWithTraits	<i>Annotate DE results with trait information</i>
--------------------	---

Description

Identifies known floriculture trait genes in differential expression results.

Usage

```
annotateWithTraits(
  de_results,
  trait = "all",
  species = "all",
  gene_col = NULL,
  mapping = NULL
)
```

Arguments

de_results	A data.frame from runDE().
trait	Trait categories to check, or 'all'.
species	Species for evidence filtering, or 'all'.
gene_col	Optional column name containing gene symbols.
mapping	Optional data.frame with gene_id and arabidopsis_id columns for ortholog-based matching.

Value

A data.frame with the original columns plus additional trait annotation columns merged from the floriculture trait database.

Examples

```
color_genes <- getFlowerGenes(trait = 'flower_color')
head(color_genes[, c('gene_symbol', 'pathway')])

de <- data.frame(
  gene_id      = c('CHS', 'CHI', 'F3H', 'gene001'),
  log2FoldChange = c(2.1, 1.8, 1.5, -0.3),
  significant   = c(TRUE, TRUE, TRUE, FALSE)
)
annotateWithTraits(de, trait = 'flower_color', gene_col = 'gene_id')
```

buildAtlas	<i>Build an expression atlas from count data</i>
------------	--

Description

Build an expression atlas from count data

Usage

```
buildAtlas(se, group_var = "stage", assay_name = "counts")
```

Arguments

se	A SummarizedExperiment object from importRosaCounts()
group_var	Column name in colData for grouping samples
assay_name	Which assay to use (default "counts")

Value

A SummarizedExperiment with atlas slot

Examples

```
library(OrnTx)
data(rosa_example)
se_norm <- normalizeCounts(rosa_example, method = "CPM")
```

buildGenomeIndex	<i>Build STAR genome index</i>
------------------	--------------------------------

Description

Builds a STAR genome index from FASTA and GTF. Required once per genome. Requires STAR to be installed.

Usage

```
buildGenomeIndex(
  genome_fasta,
  gtf_file,
  index_dir = "star_index",
  threads = 16L,
  ram_gb = 60,
  sa_index_nbases = 13L
)
```

Arguments

genome_fasta	Path to reference genome FASTA file.
gtf_file	Path to genome annotation GTF file.
index_dir	Output directory for genome index.
threads	Threads for index building (default 16).
ram_gb	Available RAM in GB (default 60).
sa_index_nbases	SA index parameter (default 13).

Value

Invisibly returns the index directory path.

Examples

```
checkTools()
## Not run:
buildGenomeIndex(
  genome_fasta = "rosa_genome.fna",
  gtf_file      = "rosa_annotation.gtf",
  index_dir     = "star_index/"
```

```
)  
  
## End(Not run)
```

buildRosaAtlas	<i>Build Rosa chinensis Expression Atlas</i>
----------------	--

Description

Build Rosa chinensis Expression Atlas

Usage

```
buildRosaAtlas(  
  counts_dir,  
  metadata_file,  
  strandedness = "reverse",  
  min_samples = 10  
)
```

Arguments

counts_dir	Path to directory containing ReadsPerGene.out.tab files
metadata_file	Path to SraRunTable.txt
strandedness	One of "reverse", "forward", "unstranded"
min_samples	Minimum samples a gene must be expressed in

Value

A SummarizedExperiment object

Examples

```
ann <- getRosaAnnotation()  
nrow(ann)  
## Not run:  
se <- buildRosaAtlas(  
  counts_dir = "path/to/counts",  
  metadata_file = "path/to/SraRunTable.txt"  
)  
  
## End(Not run)
```

checkTools	<i>Check which RNA-seq tools are available</i>
------------	--

Description

Checks fastp, STAR, fasterq-dump availability.

Usage

```
checkTools()
```

Value

Named logical vector of tool availability.

Examples

```
checkTools()
```

downloadSRA	<i>Download RNA-seq data from NCBI SRA</i>
-------------	--

Description

Downloads SRA files and converts to FASTQ using fasterq-dump or direct AWS S3 download.

Usage

```
downloadSRA(  
  accessions,  
  output_dir = "raw_reads",  
  threads = 4L,  
  compress = TRUE,  
  method = c("aws", "fasterq-dump")  
)
```

Arguments

accessions	Character vector of SRA run accessions.
output_dir	Directory to save FASTQ files.
threads	Threads for fasterq-dump (default 4).
compress	Compress output with gzip (default TRUE).
method	"fasterq-dump" or "aws" (direct S3, faster).

Value

Invisibly returns paths to downloaded FASTQ files.

Examples

```
checkTools()
## Not run:
downloadSRA(
  accessions = c("SRR10160942", "SRR10160944"),
  output_dir = "raw_reads/",
  threads    = 4
)

## End(Not run)
```

findOrthologs	<i>Find orthologs in floriculture trait database</i>
---------------	--

Description

Maps gene symbols or Arabidopsis IDs to curated trait gene information.

Usage

```
findOrthologs(gene_ids, id_type = c("symbol", "arabidopsis"), species = "any")
```

Arguments

gene_ids	Character vector of gene IDs to query.
id_type	One of "symbol" or "arabidopsis".
species	Source species (informational).

Value

A data.frame of matched trait genes.

Examples

```
findOrthologs(c("CHS", "DFR", "ANS"), id_type = "symbol")
```

floriculture_traits	<i>Floriculture Trait Gene Database</i>
---------------------	---

Description

A curated database of 64 genes associated with economically important floriculture traits, anchored to *Arabidopsis thaliana* gene IDs to enable cross-species queries regardless of annotation quality. Covers flower color, floral scent, senescence/vase life, floral development, and transcriptional regulation.

Usage

```
data(floriculture_traits)
```

Format

A data frame with 64 rows and 9 variables:

gene_symbol Gene symbol (e.g., "CHS", "DFR", "FT")

arabidopsis_id *Arabidopsis thaliana* AGI locus ID

trait Specific trait subcategory

trait_category Broad trait category

pathway Biochemical pathway name

rose_evidence Evidence level in *Rosa chinensis*

marigold_evidence Evidence level in *Tagetes erecta*

key_pmid Key PubMed reference ID

Value

A data.frame with 64 rows and columns including gene symbol, trait category, *Arabidopsis* ortholog, and *Rosa chinensis* gene ID.

Source

Literature-curated from published floriculture transcriptomics studies. See key_pmid column for references.

getFlowerGenes	<i>Get curated floriculture trait gene sets</i>
----------------	---

Description

Returns curated gene sets anchored to Arabidopsis orthologs for cross-species trait analysis.

Usage

```
getFlowerGenes(trait = "all", species = "all", evidence = "all")
```

Arguments

trait	Character. One of "flower_color", "scent", "senescence", "development", "regulatory", or "all".
species	Character. "Rosa chinensis", "Tagetes erecta", or "all".
evidence	Character. "Strong", "Moderate", "Weak", or "all".

Value

A data.frame of curated trait genes.

Examples

```
color_genes <- getFlowerGenes(trait = "flower_color")
head(color_genes)
```

getRosaAnnotation	<i>Get Rosa chinensis annotation database</i>
-------------------	---

Description

Returns the complete built-in annotation table for Rosa chinensis RchiOBHm-V2 genome containing 51,302 genes.

Usage

```
getRosaAnnotation()
```

Value

A data.frame with columns gene_id, locus_name, gene_biotype.

Examples

```
ann <- getRosaAnnotation()
head(ann)
nrow(ann)
```

importCountMatrix	<i>Import count data from any RNA-seq tool</i>
-------------------	--

Description

Accepts count matrices from STAR, featureCounts, HTSeq, Salmon, or any pre-loaded matrix. Unlike importCustomAtlas() which reads STAR output files, this function works with already-loaded matrices or files from any quantification tool.

Usage

```
importCountMatrix(
  counts,
  metadata,
  species = "Unknown species",
  file_format = "auto",
  sample_col = NULL,
  min_count = 5L,
  min_samples = 3L
)
```

Arguments

counts	File path or pre-loaded matrix/data.frame with genes as rows and samples as columns.
metadata	Data.frame with sample metadata. Row names or first column must match column names of counts.
species	Scientific name of the plant species.
file_format	One of "auto", "STAR", "featureCounts", "HTSeq", "Salmon", "matrix". Default "auto" detects format.
sample_col	Column in metadata containing sample IDs.
min_count	Minimum count for filtering (default 5).
min_samples	Minimum samples expressing gene (default 3).

Value

A SummarizedExperiment object.

Examples

```
counts <- matrix(rpois(500, 10), nrow=50, ncol=10)
rownames(counts) <- paste0("gene", 1:50)
colnames(counts) <- paste0("sample", 1:10)
meta <- data.frame(Sample = colnames(counts),
                  tissue = rep(c("petal", "leaf"), 5))
se <- importCountMatrix(counts, meta,
  species = "Rosa chinensis", sample_col = "Sample")
```

importCustomAtlas	<i>Import RNA-seq data from any plant species into OrnTx</i>
-------------------	--

Description

A universal import function that accepts STAR ReadsPerGene.out.tab files from any plant species. Unlike buildRosaAtlas() which is Rosa chinensis specific, this function works for any plant species including ornamental flowers, fruit crops, vegetables, and model plants.

Usage

```
importCustomAtlas(
  counts_dir,
  metadata_file,
  species = "Unknown species",
  sample_col = NULL,
  strandedness = c("reverse", "forward", "unstranded"),
  min_count = 5L,
  min_samples = 3L,
  annotation_file = NULL,
  sep = ",",
)
```

Arguments

counts_dir	Character. Path to directory containing ReadsPerGene.out.tab files from STAR alignment.
metadata_file	Character. Path to a CSV or TSV file with sample metadata. Must have a column matching sample IDs. See Details.
species	Character. Scientific name of the plant species, e.g., "Chrysanthemum morifolium", "Tagetes erecta".
sample_col	Character. Column name in metadata containing sample IDs that match count file names (default "Run").
strandedness	Character. Library strandedness: "reverse" (most dUTP libraries, default), "forward", or "unstranded".
min_count	Integer. Minimum count threshold for filtering (default 5).
min_samples	Integer. Minimum number of samples a gene must be expressed in (default 3).
annotation_file	Character. Optional path to GTF/GFF file for gene name annotation. If provided, gene IDs are mapped to gene names automatically.
sep	Character. Metadata file separator: ",", " for CSV (default) or "\t" for TSV.

Details

The metadata file should contain one row per sample with at minimum a column of sample IDs matching the count file names. For example, if count files are named `Sample1_ReadsPerGene.out.tab` and `Sample2_ReadsPerGene.out.tab`, the metadata must have a column containing `Sample1` and `Sample2`.

Recommended metadata columns for ornamental plant studies:

- `tissue` - tissue type (petal, leaf, root, etc.)
- `dev_stage` - developmental stage
- `cultivar` - cultivar or accession name
- `treatment` - experimental treatment
- `replicate` - biological replicate number

Value

A `SummarizedExperiment` object with:

- `assays`: raw counts matrix
- `colData`: sample metadata
- `rowData`: gene annotation (if GTF provided)
- `metadata`: species name and import parameters

Examples

```
# Check available tools
ann <- getRosaAnnotation()
nrow(ann)
## Not run:
# Example 1: Chrysanthemum morifolium
se <- importCustomAtlas(
  counts_dir = "path/to/chrysanthemum/counts/",
  metadata_file = "chrysanthemum_metadata.csv",
  species = "Chrysanthemum morifolium",
  strandedness = "reverse"
)

# Example 2: Tagetes erecta (marigold)
se <- importCustomAtlas(
  counts_dir = "path/to/marigold/counts/",
  metadata_file = "marigold_metadata.csv",
  species = "Tagetes erecta",
  annotation_file = "marigold_genome.gtf"
)

# Example 3: Any plant species
se <- importCustomAtlas(
  counts_dir = "path/to/counts/",
  metadata_file = "metadata.csv",
  species = "Tulipa gesneriana"
```

```

)

# After import - standard analysis works identically
se_norm <- normalizeCounts(se, method = "CPM")
plotPCA(se_norm, color_by = "tissue")
runDE(se, design = ~ tissue,
      contrast = c("tissue", "petal", "leaf"))

## End(Not run)

```

importRosaCounts	<i>Import STAR count files into OrnTx</i>
------------------	---

Description

Import STAR count files into OrnTx

Usage

```
importRosaCounts(file_paths, col_data, strandedness = "reverse")
```

Arguments

file_paths	Named character vector of paths to ReadsPerGene.out.tab files.
col_data	A data.frame with sample metadata.
strandedness	One of "reverse", "forward", "unstranded".

Value

A SummarizedExperiment object.

Examples

```

ann <- getRosaAnnotation()
head(ann)
## Not run:
files <- c(Sample1 = "path/to/sample1/ReadsPerGene.out.tab")
importRosaCounts(files, col_data = metadata)

## End(Not run)

```

marigold_carotenoid_genes
<i>Marigold Carotenoid Gene Mapping</i>

Description

A curated mapping of Tagetes erecta gene IDs to carotenoid biosynthesis gene symbols and Arabidopsis orthologs. Based on differential expression analysis between orange (V-01) and yellow (V-01M) flower color genotypes.

Usage

data(marigold_carotenoid_genes)

Format

A data frame with 4 rows and 3 variables:
gene_id Tagetes erecta gene ID (QVD17 format)
gene_symbol Carotenoid gene symbol (CCS, BCH, LCYe, PSY)
arabidopsis_id Arabidopsis thaliana ortholog AGI ID

Value

A data.frame containing carotenoid biosynthesis gene identifiers and annotations for *Tagetes erecta*.

Source

Differential expression analysis of NCBI BioProject PRJNA562616 using OrnTx runDE().

normalizeCounts	<i>Normalize count data</i>
-----------------	-----------------------------

Description

Normalize count data

Usage

normalizeCounts(se, method = c("CPM", "TMM", "DESeq2_vst"), design = ~1)

Arguments

se A SummarizedExperiment object.
method One of "CPM", "TMM", or "DESeq2_vst".
design Optional formula for DESeq2 methods.

Value

SummarizedExperiment with normalized assay added.

Examples

```
data(rosa_example)
se_norm <- normalizeCounts(rosa_example, method = "CPM")
```

ornTheme	<i>OrnTx ggplot2 theme</i>
----------	----------------------------

Description

OrnTx ggplot2 theme

Usage

```
ornTheme(base_size = 12)
```

Arguments

base_size Base font size

Value

A ggplot2 theme

Examples

```
p <- ggplot2::ggplot(data.frame(x=1:5, y=1:5),
  ggplot2::aes(x, y)) + ornTheme()
print(p)
```

plotAtlas	<i>Plot expression atlas heatmap</i>
-----------	--------------------------------------

Description

Plot expression atlas heatmap

Usage

```
plotAtlas(atlas, top_n = 50, title = "Expression Atlas")
```

Arguments

atlas	A SummarizedExperiment from buildAtlas()
top_n	Number of top variable genes to show
title	Plot title

Value

A ggplot2 heatmap object

Examples

```
data(rosa_example)
se_norm <- normalizeCounts(rosa_example, method = 'CPM')
atlas <- buildAtlas(se_norm, group_var = 'tissue_simple',
                    assay_name = 'CPM')
plotAtlas(atlas, top_n = 30, title = 'Rosa chinensis Atlas')

data(rosa_example)
se_norm <- normalizeCounts(rosa_example, method = "CPM")
atlas <- buildAtlas(se_norm, group_var = "tissue_simple",
                    assay_name = "CPM")
plotAtlas(atlas, top_n = 30, title = "Rosa chinensis Atlas")
```

plotPCA	<i>PCA plot of expression data</i>
---------	------------------------------------

Description

PCA plot of expression data

Usage

```
plotPCA(
  se,
  assay_name = "counts",
  color_by = "stage",
  shape_by = NULL,
  label_points = TRUE
)
```

Arguments

se	A SummarizedExperiment object
assay_name	Which assay to use
color_by	Column in colData for point color
shape_by	Column in colData for point shape
label_points	Show sample labels

Value

A ggplot2 PCA plot

Examples

```
data(rosa_example)
se_norm <- normalizeCounts(rosa_example, method = "CPM")
```

plotVolcano	<i>Volcano plot of DE results</i>
-------------	-----------------------------------

Description

Volcano plot of DE results

Usage

```
plotVolcano(
  de_results,
  alpha = 0.05,
  lfc_threshold = 1,
  top_n = 20,
  title = "Volcano Plot"
)
```

Arguments

de_results	Data.frame from runDE()
alpha	Significance threshold
lfc_threshold	LCF threshold
top_n	Number of genes to label
title	Plot title

Value

A ggplot2 volcano plot

Examples

```
data(rosa_example)
results <- runDE(rosa_example, design = ~ tissue_simple,
  contrast = c('tissue_simple', 'petal', 'abscission zone'))
plotVolcano(results, title = 'Petal vs AZ')

data(rosa_example)
results <- runDE(rosa_example,
  design = ~ tissue_simple,
```

```
contrast = c("tissue_simple", "petal", "abscission zone"))
plotVolcano(results, title = "Petal vs Abscission Zone")
```

rosa_annotation	<i>Rosa chinensis</i> Gene Annotation Database
-----------------	--

Description

A curated annotation table mapping generic gene IDs from the RchiOBHm-V2 reference genome to official *Rosa chinensis* locus names and gene biotype classifications.

Usage

```
data(rosa_annotation)
```

Format

A data frame with 51,302 rows and 3 variables:

- gene_id** Generic gene identifier (e.g., "gene0", "gene1")
- locus_name** Official RchiOBHm-V2 locus name
- gene_biotype** Gene biotype (protein_coding, lncRNA, etc.)

Value

A data.frame with gene annotation data for *Rosa chinensis*, including gene IDs, symbols, descriptions, and chromosomal coordinates.

Source

NCBI RefSeq GCF_002994745.2 (RchiOBHm-V2)

References

Hibrand Saint-Oyant L, et al. (2018) Nature Plants 4: 473-484.

rosa_example	<i>Rosa chinensis Example Expression Dataset</i>
--------------	--

Description

A small subset of the OrnTx Rosa chinensis expression atlas for use in package examples and testing. Contains 500 highly variable genes across 6 samples: 3 petal and 3 abscission zone.

Usage

```
data(rosa_example)
```

Format

A SummarizedExperiment with 500 genes x 6 samples.

assays counts: raw read counts matrix

colData sample metadata including tissue_simple

Value

A [SummarizedExperiment](#) object with 6 RNA-seq samples (3 petal, 3 abscission zone) from *Rosa chinensis*.

Source

NCBI SRA BioProjects PRJNA562083 and PRJNA594099

rose_ncbi_annotation	<i>Rosa chinensis NCBI Functional Annotation</i>
----------------------	--

Description

Functional annotation for 30,924 Rosa chinensis genes from the NCBI RefSeq annotation of the RchiOBHm-V2 reference genome (GCF_002994745.2). Contains gene product descriptions extracted from the genomic GTF file. Gene IDs use NCBI LOC format (e.g., LOC112190511).

Usage

```
data(rose_ncbi_annotation)
```

Format

A data frame with 30,924 rows and 2 variables:

ncbi_gene_id NCBI gene identifier (LOC format)

product Gene product description from NCBI annotation

Value

A data.frame with NCBI-sourced gene annotation for *Rosa chinensis*, including RefSeq IDs and functional descriptions.

Source

NCBI RefSeq GCF_002994745.2 (RchiOBHm-V2) GTF annotation. https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_002994745.2/

runDE

Differential Expression Analysis

Description

Differential Expression Analysis

Usage

```
runDE(  
  se,  
  design = ~tissue_clean,  
  contrast = NULL,  
  alpha = 0.05,  
  lfc_threshold = 1  
)
```

Arguments

se	A SummarizedExperiment object
design	Formula for experimental design
contrast	Character vector of length 3
alpha	Significance threshold
lfc_threshold	Minimum log2 fold change

Value

A data.frame of DE results

Examples

```
data(rosa_example)  
results <- runDE(rosa_example,  
  design = ~ tissue_simple,  
  contrast = c("tissue_simple", "petal", "abscission zone"))
```

runFullPipeline	<i>Run complete RNA-seq pipeline from FASTQ to SummarizedExperiment</i>
-----------------	---

Description

Convenience wrapper running the complete OrnTx pipeline: QC (fastp) -> Alignment (STAR) -> Import -> Normalize. For SRA download first, use downloadSRA() separately. Requires fastp and STAR installed (Linux/macOS/HPC only).

Usage

```
runFullPipeline(
  fastq_dir,
  genome_fasta,
  gtf_file,
  metadata_file,
  species = "Unknown species",
  work_dir = "ornAtlas_pipeline",
  threads = 16L,
  strandedness = "reverse",
  paired = TRUE,
  skip_qc = FALSE,
  skip_index = FALSE
)
```

Arguments

fastq_dir	Directory containing raw FASTQ files.
genome_fasta	Path to reference genome FASTA.
gtf_file	Path to genome annotation GTF.
metadata_file	Path to sample metadata CSV/TSV.
species	Scientific name of the plant species.
work_dir	Working directory for intermediate files.
threads	Threads for alignment (default 16).
strandedness	Library strandedness (default "reverse").
paired	Logical. Paired-end data (default TRUE).
skip_qc	Logical. Skip fastp QC step (default FALSE).
skip_index	Logical. Skip index building if built.

Value

A CPM-normalized SummarizedExperiment.

Examples

```
checkTools()
## Not run:
se <- runFullPipeline(
  fastq_dir      = "raw_reads/",
  genome_fasta   = "rosa_genome.fna",
  gtf_file       = "rosa_annotation.gtf",
  metadata_file  = "metadata.csv",
  species        = "Rosa chinensis",
  threads        = 16
)
plotPCA(se, color_by = "tissue")

## End(Not run)
```

runQC

Quality control of FASTQ files using fastp

Description

Runs fastp QC on FASTQ files. Supports paired and single-end data. Requires fastp to be installed.

Usage

```
runQC(
  input_dir,
  output_dir = "clean_reads",
  paired = TRUE,
  threads = 4L,
  min_length = 36L,
  quality = 20L
)
```

Arguments

input_dir	Directory containing raw FASTQ files.
output_dir	Directory for clean FASTQ output.
paired	Logical. TRUE for paired-end (default).
threads	Threads per sample (default 4).
min_length	Minimum read length after trimming (default 36).
quality	Minimum base quality score (default 20).

Value

Invisibly returns paths to cleaned FASTQ files.

Examples

```
checkTools()
## Not run:
runQC(input_dir="raw_reads/", output_dir="clean_reads/")

## End(Not run)
```

standardizeStage	<i>Standardize developmental stage vocabulary</i>
------------------	---

Description

Harmonizes inconsistent developmental stage names across studies into a controlled vocabulary. Solves the problem of "bud", "Bud", "floral bud", "Stage 1" all meaning the same thing in different publications.

Usage

```
standardizeStage(stage_vector)
```

Arguments

stage_vector Character vector of stage names to standardize.

Value

Character vector of standardized stage names.

Examples

```
raw_stages <- c("bud", "Bud", "floral bud", "Stage 1",
               "open flower", "OF", "full bloom", "senescent")
standardizeStage(raw_stages)
```

Index

* datasets

- floriculture_traits, [11](#)
- marigold_carotenoid_genes, [17](#)
- rosa_annotation, [21](#)
- rosa_example, [22](#)
- rose_ncbi_annotation, [22](#)

- alignReads, [2](#)
- annotateDEResults, [3](#)
- annotateGenes, [4](#)
- annotateWithNCBI, [5](#)
- annotateWithTraits, [5](#)

- buildAtlas, [6](#)
- buildGenomeIndex, [7](#)
- buildRosaAtlas, [8](#)

- checkTools, [9](#)

- downloadSRA, [9](#)

- findOrthologs, [10](#)
- floriculture_traits, [11](#)

- getFlowerGenes, [12](#)
- getRosaAnnotation, [12](#)

- importCountMatrix, [13](#)
- importCustomAtlas, [14](#)
- importRosaCounts, [16](#)

- marigold_carotenoid_genes, [17](#)

- normalizeCounts, [17](#)

- ornTheme, [18](#)

- plotAtlas, [18](#)
- plotPCA, [19](#)
- plotVolcano, [20](#)

- rosa_annotation, [21](#)

- rosa_example, [22](#)
- rose_ncbi_annotation, [22](#)
- runDE, [23](#)
- runFullPipeline, [24](#)
- runQC, [25](#)

- standardizeStage, [26](#)
- SummarizedExperiment, [22](#)