Package 'systemPipeRdata'

February 26, 2026

Title systemPipeRdata: Workflow templates and sample data

Version 2.14.5

Date 2026-02-16

Author Thomas Girke

Maintainer Thomas Girke <thomas.girke@ucr.edu>

Encoding UTF-8

biocViews Genetics, Infrastructure, DataImport, Sequencing, RNASeq, ChIPSeq, MethylSeq, SNP, GeneExpression, Coverage, GeneSetEnrichment, Alignment, QualityControl, ImmunoOncology, RiboSeq, WorkflowStep

Description systemPipeRdata complements the systemPipeR workflow management system (WMS) by offering a collection of pre-designed data analysis workflow templates. These templates are easily accessible and can be readily loaded onto a user's system with a single command. Once loaded, the WMS can immediately utilize these templates for efficient end-to-end analysis, serving a wide range of data analysis needs.

Depends R (>= 3.6.0)

Imports methods, Biostrings, BiocGenerics, jsonlite, remotes

Suggests GenomicFeatures, DT, txdbmaker, GenomicRanges, IRanges, Rsamtools, ShortRead, rtracklayer, RUnit, BiocStyle, knitr, rmarkdown, systemPipeR, kableExtra, magrittr, dplyr, gert

VignetteBuilder knitr

License Artistic-2.0

NeedsCompilation no

URL https://github.com/tgirke/systemPipeRdata

RoxygenNote 7.3.3

git_url https://git.bioconductor.org/packages/systemPipeRdata

git_branch RELEASE_3_22

git_last_commit 66d3705

git_last_commit_date 2026-02-16

Repository Bioconductor 3.22

Date/Publication 2026-02-26

Contents

`availableWF` *List Available Workflows Templates at* `systemPipeRdata` *package*

Description

This function checks the workflow templates availability from `systemPipeRdata` package and also from systemPipeR Organization on GitHub.

Usage

```
availableWF(github = FALSE)
```

Arguments

`github`	logical. If `TRUE`, it will return current experimental workflow templates available on `systemPipeR` Organization.

Details

Internally, this function uses the GitHub API, and there is an access limit per hour. For more details, please check: `system("curl -i https://api.github.com/users/<username>")`.

Value

Workflow templates are printed on the console, and also return an invisible list with the names of the workflows templates availabe at `systemPipeRdata` package. If `github = TRUE`, it will return an addicional `data.frame` with current workflow templates available on `systemPipeR` Organization in the list.

Note

We are assuming that workflow templates repositories under systemPipeR Organization content the keyword "Workflow Template" on the Description section and "Topics" section, we expected "systempiper" and "release" or "development" words.

Author(s)

Daniela Cassol

See Also

`genWorkenvir`.

Examples

```
availableWF()
## Not run:
## List Workflow Templates from \code{systemPipeR} Organization
availableWF(github = TRUE)

## End(Not run)
```

---

genWorkenvir *Generate workflow templates*

Description

Generates workflow templates for systemPipeR package. The template environemts contain a predifined directory structure along with run parameter files and sample data. The strucutre of the workflow templates and the sample data are described in all details in the Overview Vignette of the systemPipeR package.

Usage

```
genWorkenvir(workflow, mydirname=NULL, bam=FALSE)
```

Arguments

workflow	character string of workflow templates to be generated. Supported values can be checked with the \link{availableWF}() function.
mydirname	Specifies the name of the workflow directory. The default NULL uses the name of the chosen workflow. An error is issued if a directory of the same name and path exists already.
bam	If bam=TRUE pregenerated short read alignment (BAM) files will be included in the results directory of the workflow environment. Note, these BAM files have been generated with the HISAT2 aligner using the FASTQ files provided in the data directory. The default bam=FALSE omits this step meaning no BAM files will be copied into the results directory.

Details

Check the output of \link{availableWF}() to the current workflow templates available on systemPipeR Organization.

For an interactive() session, the readline() function provides the option choose between proceeding or not, through options: yes or no. For non-interactive use, if there is no package install, the option yes will be selected.

Value

Workflow directory containing sample data and parameter files along with the following subdirectories:

param/	stores parameter files
data/	stores input data

results/	stores output results

For more details, please consult the Overview Vignette (HTML) of the systemPipeR package (http://bioconductor.org/packages/systemPipeR).

Author(s)

Thomas Girke and Daniela Cassol

Examples

```
## Return location of sample data
samplepaths <- pathList()
## Not run:
## Generate varseq workflow environment
genWorkenvir(workflow="varseq", mydirname=NULL)
setwd("varseq")

## End(Not run)
```

---

genWorkenvir_gh *Clone a workflow repository using gert and enforce version requirements*

Description

Convenience wrapper around `gert::git_clone()` that clones a workflow repository and enforces minimum package version requirements specified in the workflow's `manifest.yml` file.

Usage

```
genWorkenvir_gh(
  url,
  mydirname,
  force_min_version = FALSE,
  cleanup_on_fail = TRUE,
  ...
)
```

Arguments

url	GitHub HTTPS URL of the workflow repository to clone.
mydirname	Target directory to clone into.
force_min_version	Logical; if `FALSE` (default), enforce `bioc_min` requirements and stop on version mismatch. If `TRUE`, continue with a warning (at your own risk). Missing required packages always cause an error.
cleanup_on_fail	Logical; if `TRUE` (default) and a strict version check fails, remove the cloned directory before stopping.
...	Additional arguments passed to `gert::git_clone()`.

Details

If `manifest.yml` contains a block like:

```
systemPipeR:
  bioc_min: "2.17.1"

systemPipeRdata:
  bioc_min: "2.15.4"
```

then the installed versions of the corresponding packages are checked.

By default, version mismatches cause an error and the cloned directory is removed to avoid leaving a partially initialized workflow.

Value

Invisibly returns the normalized path of the cloned repository.

Examples

```
## Not run:
## Strict (default)
genWorkenvir_gh(
  "https://github.com/systemPipeR/sprwf-new.git",
  "sprwf-new"
)

## Allow version mismatch (not recommended)
genWorkenvir_gh(
  "https://github.com/systemPipeR/sprwf-new.git",
  "sprwf-new",
  force_min_version = TRUE
)

## End(Not run)
```

---

getData_gh *Install a shared workflow data directory from a GitHub tag archive*

Description

Downloads a GitHub tag archive (`.zip`) for a data repository and installs the contents of the `data/` directory into the workflow repository (default destination: `"data"`).

Usage

```
getData_gh(
  data_repo = "ORG/cwl-data",
  dest = "data",
  tag = NULL,
  default_tag = "v1.0.0",
  source_subdir = "data",
  force = TRUE,
```

```
  mode = c("backup", "side_by_side", "ask", "abort", "overwrite"),
  quiet = FALSE
)
```

Arguments

data_repo	Data repository identifier. Either "ORG/repo" or a GitHub URL like "https://github.com/ORG/rep If left at the default placeholder ("ORG/cwl-data") and manifest.yml exists, the repo is read from the manifest.
dest	Destination directory in the workflow repository (default "data").
tag	Git tag to install. If NULL, the tag is resolved in priority: 1. environment variable DATA_TAG 2. manifest.yml data: tag: (and data: repo: if data_repo is left at default) 3. default_tag
default_tag	Fallback tag used if no other tag source is available.
source_subdir	Subdirectory within the data repo archive to copy (default "data").
force	Logical; only relevant when mode="overwrite". If FALSE, refuse destructive overwrite.
mode	Behavior when dest exists and is non-empty: "backup" (default), "side_by_side", "ask", "abort", or "overwrite".
quiet	Logical; suppress progress messages.

Details

The data repository URL and tag are typically specified in the workflow repository's manifest.yml under the data: block (keys repo and tag).

Versioned snapshots via Git tags: This function installs files from a GitHub *tag archive* URL of the form https://github.com/<ORG>/<REPO>/archive/refs/tags/<TAG>.zip. To publish an updated parameter (or data) pack, developers must create and push a new Git tag in the corresponding repository (and update the workflow manifest.yml tag: field accordingly). Creating a GitHub Release for the tag is optional but recommended for discoverability.

Users can override the tag without editing the manifest via the environment variable PARAM_TAG (or DATA_TAG for sample data).

Value

Invisibly returns a list with elements dest, tag, repo, backup, and mode.

Examples

```
## Not run:
## Run from inside a workflow repo that has manifest.yml:
getData_gh()

## Developer workflow (run in param/data repo):
## git tag -a v1.2.0 -m "Update CWL params"
## git push --tags
## Then update manifest.yml in each workflow repo to tag: "v1.2.0"

## End(Not run)
```

getParam_gh *Install a shared CWL parameter directory from a GitHub tag archive*

Description

Downloads a GitHub tag archive (`.zip`) for a parameter repository and installs the contents of a subdirectory (default: `"param"`) into the workflow repository (default destination: `"param"`).

Usage

```
getParam_gh(
  param_repo = "ORG/cwl-params",
  dest = "param",
  tag = NULL,
  default_tag = "v1.0.0",
  source_subdir = "param",
  force = TRUE,
  mode = c("backup", "side_by_side", "ask", "abort", "overwrite"),
  quiet = FALSE
)
```

Arguments

`param_repo`	Parameter repository identifier. Either `"ORG/repo"` or a GitHub URL like `"https://github.com/OR` If left at the default placeholder (`"ORG/cwl-params"`) and `manifest.yml` exists, the repo is read from the manifest.
`dest`	Destination directory in the workflow repository (default `"param"`).
`tag`	Git tag to install. If `NULL`, the tag is resolved in priority: 1. environment variable `PARAM_TAG` 2. `manifest.yml` `param: tag:` (and `param: repo:` if `param_repo` is left at default) 3. `default_tag`
`default_tag`	Fallback tag used if no other tag source is available.
`source_subdir`	Subdirectory within the parameter repo archive to copy (default `"param"`).
`force`	Logical; only relevant when `mode="overwrite"`. If `FALSE`, refuse destructive overwrite.
`mode`	Behavior when `dest` exists and is non-empty: `"backup"` Rename existing `dest` to `dest.BAK-<timestamp>` and install fresh (default). `"side_by_side"` Install into `dest.NEW-<tag>-<timestamp>` without modifying existing `dest`. `"ask"` Prompt interactively; refuses in non-interactive sessions. `"abort"` Stop with an error if `dest` exists and is non-empty. `"overwrite"` Delete contents of `dest` and install fresh (requires `force=TRUE`).
`quiet`	Logical; suppress progress messages.

Details

The parameter repository URL and tag are typically specified in the workflow repository's `manifest.yml` under the `param:` block (keys `repo` and `tag`).

Versioned snapshots via Git tags: This function installs files from a GitHub *tag archive* URL of the form `https://github.com/<ORG>/<REPO>/archive/refs/tags/<TAG>.zip`. To publish an updated parameter (or data) pack, developers must create and push a new Git tag in the corresponding repository (and update the workflow `manifest.yml` `tag:` field accordingly). Creating a GitHub Release for the tag is optional but recommended for discoverability.

Users can override the tag without editing the manifest via the environment variable `PARAM_TAG` (or `DATA_TAG` for sample data).

Value

Invisibly returns a list with elements:

dest Normalized destination path where parameters were installed.

tag Resolved tag used for installation.

repo Resolved `ORG/repo` used for archive download.

backup Backup directory path (if a backup occurred), otherwise `NA`.

mode Effective mode used.

Examples

```
## Not run:
## Run from inside a workflow repo, usually generated by genWorkenvir_gh,
## that has manifest.yml:
getParam_gh()

## Explicit repo and tag:
getParam_gh(param_repo = "systemPipeR/sprwfcmp-param", tag = "v1.1.1")

## Safer install without touching existing param/:
getParam_gh(mode = "side_by_side")

## Developer workflow (run in param/data repo):
## git tag -a v1.2.0 -m "Update CWL params"
## git push --tags
## Then update manifest.yml in each workflow repo to tag: "v1.2.0"

## End(Not run)
```

---

getSubsetReads *Subsetting fastq data*

Description

Returns subsets of fastq files data based on specific mapping regions or list of genes or GRanges object.

Usage

```
getSubsetReads(args,
  geneList = NULL,
  gr = NULL,
  MappingRegion = 1:1e+05,
  sample_range = 90000:1e+05,
  truncate_refs = TRUE,
  id_read_number = TRUE,
  annotation = "data/tair10.gff",
  reference = "data/tair10.fasta",
  annot_outname = "tair10_sub.gff",
  ref_outname = "tair10_sub.fasta",
  outdir = "data/subset/",
  silent = FALSE
)
```

Arguments

args	object of class SYSargs2.
geneList	selected genes list to retrieve the reads from the fastq file.
gr	an object containing genomic ranges to retrieve the reads from the fastq file.
MappingRegion	integers ranges of start and end of chromosome position to retrieve the reads from the fastq file.
sample_range	random range to subsetted the fastq file.
truncate_refs	logical. If TRUE it will generate reference genome and annotation subset file.
id_read_number	if fastq file contains sequence name with read number (`$ri --defline-seq '@$sn[_$rn]/$ri'`).
annotation	path to annotation file.
reference	path to reference genome.
annot_outname	character name of the annotation output file.
ref_outname	character name of the reference genome output file.
outdir	path to output directory.
silent	if set to TRUE, all messages returned by the function will be suppressed.

Value

Workflow directory containing sample data and parameter files along with the following subdirectories:

param/	stores parameter files
data/	stores input data
results/	stores output results

For more details, please consult the Overview Vignette (HTML) of the systemPipeR package (http://bioconductor.org/packages/systemPipeR).

Author(s)

Thomas Girke, Shiyuan Guo and Daniela Cassol

Examples

```
## Not run:
getSubsetReads(args, MappingRegion = 1:900, sample_range = 800:900, outdir = "data/subset/", silent = FALSE)
getSubsetReads(args, MappingRegion = 1:900, sample_range = NULL, outdir = "data/subset/", silent = FALSE)

## End(Not run)
```

---

pathList *Return location of sample data*

Description

Function to return paths to sample data provided by sytemPipeRdata package.

Usage

```
pathList()
```

Value

list

Author(s)

Thomas Girke

Examples

```
samplepaths <- pathList()
```

Index