

# Package ‘systemPipeRdata’

September 17, 2024

**Title** systemPipeRdata: Workflow templates and sample data

**Version** 2.8.0

**Date** 2023-10-10

**Author** Thomas Girke

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

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

**Description** systemPipeRdata is a helper package to generate with a single command NGS workflow templates that are intended to be used by its parent package systemPipeR. The latter is an environment for building end-to-end analysis pipelines with automated report generation for next generation sequence (NGS) applications such as RNA-Seq, RIBO-Seq, ChIP-Seq, VAR-Seq and many others. Detailed examples for using systemPipeRdata are given in systemPipeR's overview vignette.

**Imports** methods, Biostrings, BiocGenerics, jsonlite, remotes

**Suggests** GenomicFeatures, GenomicRanges, IRanges, Rsamtools, ShortRead, rtracklayer, RUnit, BiocStyle, knitr, rmarkdown, systemPipeR

**VignetteBuilder** knitr

**License** Artistic-2.0

**NeedsCompilation** no

**URL** <https://github.com/tgirke/systemPipeRdata>, <https://systempipe.org/>

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

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** b03c247

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-09-17

## Contents

|                          |          |
|--------------------------|----------|
| availableWF . . . . .    | 2        |
| genWorkenvir . . . . .   | 3        |
| getSubsetReads . . . . . | 5        |
| pathList . . . . .       | 6        |
| <b>Index</b>             | <b>7</b> |

---

|             |  |
|-------------|--|
| availableWF | <i>List Available Workflows Templates at systemPipeRdata package</i> |
|-------------|--|

---

### 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 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

Return a list with the names of the workflows templates available at systemPipeRdata package. If `github = TRUE`, it will return an additional data.frame with current workflow templates available on systemPipeR Organization.

### 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 environments contain a predefined directory structure along with run parameter files and sample data. The structure 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, ref="master", subdir=NULL, url=NULL, urlname=NULL)
```

## Arguments

|           |  |
|-----------|--|
| workflow  | character string of workflow templates to be generated. Supported values can be checked with the <code>\link{availableWF}()</code> function. Workflow name containing a '/' are treated as GitHub repositories and installed using <code>BiocManager::\link{install_github}</code> . For Github workflow templates, the format <code>systemPipeR/repo</code> is required. See Details.   |
| 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 <code>bam=TRUE</code> 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 <code>bam=FALSE</code> omits this step meaning no BAM files will be copied into the results directory. |
| ref       | Desired GitHub reference for the branch name. Default to master branch.  |
| subdir    | subdirectory within GitHub repo that contains the R package, if it is required.  |
| url       | character string of a specifies the URL of a different version of the R Markdown workflow template or other file to download. The default NULL copies the current version available in the 'systemPipeRdata' or the workflow template on GitHub being selected.  |
| urlname   | character string with the name where the downloaded file is saved. This is argument is required when the url is provided.  |

## Details

When installing GitHub Workflow Packages under **systemPipeR Organization**, it is requested to the remotes package function `BiocManager::\link{install_github}()` to build vignettes and also install all the dependencies, via `dependencies=TRUE, build_vignettes=TRUE`.

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

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, url=NULL, urlname=NULL)
setwd("varseq")

## List Workflow Templates from \code{systemPipeRdata} package and \code{systemPipeR} Organization
availableWF(github = TRUE)
## Generate 'systemPipeR/systemPipeChIPseq' workflow environment
genWorkenvir(workflow="systemPipeR/systemPipeChIPseq", mydirname=NULL, ref="master", subdir=NULL)
setwd("systemPipeChIPseq")

## Download a specific R Markdown file
genWorkenvir(workflow="systemPipeR/systemPipeRNAseq", mydirname="rnaseq", url = "https://raw.githubusercontent.com")

## End(Not run)
```

---

|                |                              |
|----------------|------------------------------|
| getSubsetReads | <i>Subsetting fastq data</i> |
|----------------|------------------------------|

---

### 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 ( <code>\$ri --define-seq '@\$sn[_\$rn]/\$ri'</code> ). |
| 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 | <i>Return location of sample data</i> |
|----------|---------------------------------------|

---

**Description**

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

**Usage**

```
pathList()
```

**Value**

```
list
```

**Author(s)**

Thomas Girke

**Examples**

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

# Index

## \* **utilities**

- availableWF, [2](#)
- genWorkenvir, [3](#)
- getSubsetReads, [5](#)
- pathList, [6](#)

availableWF, [2](#)

genWorkenvir, [2](#), [3](#)  
getSubsetReads, [5](#)

pathList, [6](#)