# *systemPipeRdata*: NGS workflow templates and sample data

***Author: Daniela Cassol (danielac@ucr.edu) and Thomas Girke (thomas.girke@ucr.edu)***

**Last update: 19 November, 2017**

**Package**

systemPipeRdata 1.5.1

## Contents

1 Introduction . . . . . 2

2 Getting Started . . . . . 2

2.1 Installation . . . . . 2

2.2 Loading package and documentation . . . . . 2

2.3 Generate workflow template . . . . . 2

2.4 Run workflows . . . . . 3

2.5 Return paths to sample data . . . . . 3

3 Version information . . . . . 4

4 Funding . . . . . 5

References . . . . . 5

Note: the most recent version of this vignette can be found here and a short overview slide show here.

# 1 Introduction

`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` (Girke 2014). 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. The directory structure of the workflow templates and the sample data used by `systemPipeRdata` are described here.

Back to Table of Contents

# 2 Getting Started

## 2.1 Installation

The R software for using `systemPipeRdata` can be downloaded from CRAN. The `systemPipeRdata` package can be installed from within R as follows: zR‘{r install, eval=FALSE} source(“http://bioconductor.org/biocLite.R”) # Sources the biocLite.R installation script biocLite(“tgirke/systemPipeRdata”, build_vignettes=TRUE, dependencies=TRUE) # Installs from github biocLite(“systemPipeRdata”) # Installs from Bioconductor once available there “‘

Back to Table of Contents

## 2.2 Loading package and documentation

```
library("systemPipeRdata") # Loads the package
```

```
library(help="systemPipeRdata") # Lists package info
vignette("systemPipeRdata") # Opens vignette
```

Back to Table of Contents

## 2.3 Generate workflow template

Load one of the available NGS workflows into your current working directory. The following does this for the `varseq` template. The name of the resulting workflow directory can be specified under the `mydirname` argument. 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.

```
genWorkenvir(workflow="varseq", mydirname=NULL)
setwd("varseq")
```

On Linux and OS X systems the same can be achieved from the command-line of a terminal with the following commands. `{.sh generate_workenvir_from_shell, eval=FALSE} $ Rscript -e "systemPipeRdata::genWorkenvir(workflow='varseq', mydirname=NULL)"`

The workflow templates generated by *genWorkenvir* contain the following preconfigured directory structure:

```
workflow_name/            # *.Rnw/*.Rmd scripts and targets file
                param/    # parameter files for command-line software
                data/     # inputs e.g. FASTQ, reference, annotations
                results/  # analysis result files
```

Back to Table of Contents

## 2.4 Run workflows

Next, run from within R the chosen sample workflow by executing the code provided in the corresponding **.Rnw* template file. If preferred the corresponding **.Rmd* or **.R* versions can be used instead. Alternatively, one can run an entire workflow from start to finish with a single command by executing from the command-line *'make -B'* within the workflow directory (here *'varseq'*). Much more detailed information on running and customizing *systemPipeR* workflows is available in its overview vignette here. This vignette can also be opened from R with the following command.

```
library("systemPipeR") # Loads systemPipeR which needs to be installed via biocLite() from Bioconductor
```

```
vignette("systemPipeR", package = "systemPipeR")
```

Back to Table of Contents

## 2.5 Return paths to sample data

The location of the sample data provided by *systemPipeRdata* can be returned as a *list*.

```
pathList()
## $targets
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/param/targets.txt"
##
## $targetsPE
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/param/targetsPE.txt"
##
## $annotationdir
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/annotation/"
##
## $fastqdir
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/fastq/"
##
## $bamdir
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/bam/"
##
```

```
## $paramdir
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/param/"
##
## $workflows
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/workflows/"
##
## $chipseq
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/workflows/chipseq/"
##
## $rnaseq
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/workflows/rnaseq/"
##
## $riboseq
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/workflows/riboseq/"
##
## $varseq
## [1] "/home/dcassol/R/x86_64-pc-linux-gnu-library/3.4/systemPipeRdata/extdata/workflows/varseq/"
```

Back to Table of Contents

# 3 Version information

```
sessionInfo()
## R version 3.4.2 (2017-09-28)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    parallel  methods   stats     graphics  grDevices utils     datasets  base
##
## other attached packages:
##  [1] systemPipeRdata_1.5.1      systemPipeR_1.12.0         ShortRead_1.36.0
##  [4] GenomicAlignments_1.14.0   SummarizedExperiment_1.8.0 DelayedArray_0.4.1
##  [7] matrixStats_0.52.2         Biobase_2.38.0             BiocParallel_1.12.0
## [10] Rsamtools_1.30.0           Biostrings_2.46.0          XVector_0.18.0
## [13] GenomicRanges_1.30.0       GenomeInfoDb_1.14.0        IRanges_2.12.0
## [16] S4Vectors_0.16.0           BiocGenerics_0.24.0        BiocStyle_2.6.0
##
```

```
## loaded via a namespace (and not attached):
##  [1] edgeR_3.20.1            RMySQL_0.10.13          bit64_0.9-7             splines_3.4.2
##  [5] assertthat_0.2.0        latticeExtra_0.6-28     RBGL_1.54.0             blob_1.1.0
##  [9] GenomeInfoDbData_0.99.1 yaml_2.1.14             Category_2.44.0         progress_1.1.2
## [13] RSQLite_2.0             backports_1.1.1         lattice_0.20-35         limma_3.34.1
## [17] digest_0.6.12           checkmate_1.8.5         RColorBrewer_1.1-2      colorspace_1.3-2
## [21] htmltools_0.3.6         Matrix_1.2-11           plyr_1.8.4              GSEABase_1.40.0
## [25] pkgconfig_2.0.1         XML_3.98-1.9            pheatmap_1.0.8          biomaRt_2.34.0
## [29] genefilter_1.60.0       bookdown_0.5            zlibbioc_1.24.0         GO.db_3.5.0
## [33] xtable_1.8-2            scales_0.5.0            brew_1.0-6              tibble_1.3.4
## [37] annotate_1.56.1         ggplot2_2.2.1           GenomicFeatures_1.30.0  lazyeval_0.2.1
## [41] survival_2.41-3         magrittr_1.5            memoise_1.1.0           evaluate_0.10.1
## [45] fail_1.3                hwriter_1.3.2           GOstats_2.44.0          graph_1.56.0
## [49] tools_3.4.2             prettyunits_1.0.2       BBmisc_1.11             stringr_1.2.0
## [53] sendmailR_1.2-1         munsell_0.4.3           locfit_1.5-9.1          AnnotationDbi_1.40.0
## [57] compiler_3.4.2          rlang_0.1.4             grid_3.4.2              RCurl_1.95-4.8
## [61] rjson_0.2.15            AnnotationForge_1.20.0  base64enc_0.1-3         bitops_1.0-6
## [65] rmarkdown_1.7           codetools_0.2-15        gtable_0.2.0            DBI_0.7
## [69] R6_2.2.2                knitr_1.17              rtracklayer_1.38.0      bit_1.1-12
## [73] rprojroot_1.2           Rgraphviz_2.22.0        stringi_1.1.5           BatchJobs_1.6
## [77] Rcpp_0.12.13
```

Back to Table of Contents

# 4 Funding

This project was supported by funds from the National Institutes of Health (NIH) and the National Science Foundation (NSF).

Back to Table of Contents

# References

Girke, Thomas. 2014. "systemPipeR: NGS Workflow and Report Generation Environment." UC Riverside. https://github.com/tgirke/systemPipeR.