

# Package ‘Herper’

May 29, 2024

**Type** Package

**Title** The Herper package is a simple toolset to install and manage conda packages and environments from R

**Version** 1.14.0

**Description** Many tools for data analysis are not available in R, but are present in public repositories like conda. The Herper package provides a comprehensive set of functions to interact with the conda package management system. With Herper users can install, manage and run conda packages from the comfort of their R session. Herper also provides an ad-hoc approach to handling external system requirements for R packages. For people developing packages with python conda dependencies we recommend using basilisk (<https://bioconductor.org/packages/release/bioc/html/basilisk.html>) to internally support these system requirements pre-hoc.

**URL** <https://github.com/RockefellerUniversity/Herper>

**Depends** R (>= 4.0), reticulate

**Imports** utils, rjson, withr, stats

**Suggests** BiocStyle, testthat, knitr, rmarkdown

**License** GPL-3

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.2.0

**VignetteBuilder** knitr

**biocViews** Infrastructure, Software

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

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

**git\_last\_commit** 388d929

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

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-29

**Author** Matt Paul [aut] (<<https://orcid.org/0000-0002-3020-7729>>),  
 Thomas Carroll [aut, cre] (<<https://orcid.org/0000-0002-0073-1714>>),  
 Doug Barrows [aut],  
 Kathryn Rozen-Gagnon [ctb]

**Maintainer** Thomas Carroll <tc.infomatics@gmail.com>

## Contents

conda_search . . . . .	2
export_CondaEnv . . . . .	3
install_CondaSysReqs . . . . .	4
install_CondaTools . . . . .	5
list_CondaEnv . . . . .	6
list_CondaPkgs . . . . .	7
with_CondaEnv . . . . .	7

<b>Index</b>	<b>10</b>
--------------	-----------

---

conda_search	<i>Search package availability</i>
--------------	------------------------------------

---

### Description

Search package availability

### Usage

```
conda_search(package, channel = NULL, print_out = TRUE, pathToMiniConda = NULL)
```

### Arguments

package	Package to search for. If an exact match is found, the function will return true (assuming 'package_version' is left NULL or is a valid entry). If there is not an exact match and other packages contain this text, the function will return FALSE but the alternative options will be printed if print_out = TRUE.
channel	Channels for to search in (bioconda and conda-forge are defaults).
print_out	Either True or FALSE indicating whether to print out information about available builds and channels for the search entry.
pathToMiniConda	Path to miniconda installation. If this is set to NULL (default), then the output of 'reticulate::miniconda_path()' is used.

### Value

TRUE/FALSE

**Author(s)**

Doug Barrows and Matt Paul

**Examples**

```
condaPaths <- install_CondaTools("salmon", "herper_env")
conda_search("salmon")
```

---

export\_CondaEnv      *Export Conda environment.*

---

**Description**

Export Conda environment

Import Conda environment

**Usage**

```
export_CondaEnv(
  env_name,
  yml_export = NULL,
  pathToMiniConda = NULL,
  depends = TRUE
)
```

```
import_CondaEnv(
  yml_import,
  name = NULL,
  pathToMiniConda = NULL,
  install = TRUE,
  channels = NULL
)
```

**Arguments**

env_name	Name of environment you want to save
yml_export	Destination for exported environment yml file
pathToMiniConda	NULL Path to miniconda installation
depends	if FALSE will only include packages explicitly installed and not dependencies
yml_import	conda environment yml file
name	Name of the environment to create.
install	TRUE/FALSE whether to install miniconda at path if it doesn't exist.
channels	Channels for miniconda (bioconda and conda-forge are defaults).

**Value**

Nothing returned. Output written to file.

Nothing returned. Output written to file.

**Author(s)**

Matt Paul

---

install\_CondaSysReqs    *Install Conda requirements listed in the System Requirement field of description*

---

**Description**

Install Conda requirements

**Usage**

```
install_CondaSysReqs(
  pkg,
  channels = NULL,
  env = NULL,
  pathToMiniConda = NULL,
  updateEnv = FALSE,
  SysReqsAsJSON = FALSE,
  SysReqsSep = ", ",
  verbose = FALSE
)
```

**Arguments**

pkg	Package to install Conda System Requirements from.
channels	Channels for miniconda (bioconda and conda-forge are defaults).
env	Name of Conda environment to install tools into.
pathToMiniConda	NULL Path to miniconda installation
updateEnv	Update existing package's conda environment if already installed.
SysReqsAsJSON	Parse the SystemRequirements in JSON format (see Details). Default is TRUE.
SysReqsSep	Separator used in SystemRequirement field.
verbose	Print system messages from conda on progress (Default is FALSE). There is a third option "silent" which suppresses Herper and Conda messaging.

**Value**

Nothing returned. Output written to file.

**Author(s)**

Thomas Carroll

**Examples**

```
testPkg <- system.file("extdata/HerperTestPkg", package = "Herper")
install.packages(testPkg, type = "source", repos = NULL)
condaPaths <- install_CondaSysReqs("HerperTestPkg", SysReqsAsJSON = FALSE)
system2(file.path(condaPaths$pathToEnvBin, "samtools"), args = "--help")
```

---

install\_CondaTools      *Install Conda requirements.*

---

**Description**

Install Conda requirements

**Usage**

```
install_CondaTools(
  tools,
  env,
  channels = NULL,
  pathToMiniConda = NULL,
  updateEnv = FALSE,
  search = FALSE,
  verbose = FALSE
)
```

**Arguments**

tools	Vector of software to install using conda.
env	Name of Conda environment to install tools into.
channels	Channels for miniconda (bioconda and conda-forge are defaults).
pathToMiniConda	NULL Path to miniconda installation
updateEnv	Update existing package's conda environment if already installed.
search	Whether to search for the package name and version before installing. It is highly recommended this be set to TRUE as information about available versions or similar packages will be included in the output if the exact match is not found.
verbose	Print system messages from conda on progress (Default is FALSE). There is a third option "silent" which suppresses Herper and Conda messaging.

**Value**

Nothing returned. Output written to file.

**Author(s)**

Thomas Carroll

**Examples**

```
condaPaths <- install_CondaTools("salmon", "herper_env")
system2(file.path(condaPaths$pathToEnvBin, "salmon"), args = "--help")
```

---

list_CondaEnv	<i>List Conda environments.</i>
---------------	---------------------------------

---

**Description**

List Conda environments

**Usage**

```
list_CondaEnv(pathToMiniConda = NULL, allCondas = FALSE, env = NULL)
```

**Arguments**

pathToMiniConda	NULL Path to miniconda installation
allCondas	Logical. Whether to return conda environments, for all discoverable conda installs, or just the conda specified in pathToMiniConda.
env	Environment name. If this is supplied to list_CondaEnv, it will query whether that environment is present in the given conda.

**Value**

Conda environment names and the file paths to their conda installation are printed to the screen. If environment name is supplied a TRUE/FALSE will be returned depending on whether that environment is present or not.

**Author(s)**

Matt Paul

**Examples**

```
condaPaths <- install_CondaTools("salmon", "herper_env")
list_CondaEnv()
list_CondaEnv( env = "herper_env")
```

---

list\_CondaPkgs            *List Conda packages.*

---

**Description**

List Conda packages

**Usage**

```
list_CondaPkgs(env, pathToMiniConda = NULL, pkg = NULL)
```

**Arguments**

env                    environment to look in  
pathToMiniConda        NULL Path to miniconda installation  
pkg                    Package name. If this is supplied to list\_CondaPkg, it will query whether that package is present in the given environment.

**Value**

Conda package information is printed to the screen. If package name is supplied a TRUE/FALSE will be returned depending on whether that package is present or not.

**Author(s)**

Matt Paul

**Examples**

```
condaPaths <- install_CondaTools("salmon", "herper_env")  
list_CondaPkgs("herper_env")
```

---

with\_CondaEnv            *Use Conda environments.*

---

**Description**

Use Conda environments  
Use Conda environments

**Usage**

```

with_CondaEnv(
  new,
  code,
  pathToMiniConda = NULL,
  path_action = "prefix",
  pythonpath_action = "replace",
  perl5lib_action = "replace",
  path_additional = NULL,
  pythonpath_additional = NULL,
  perl5lib_additional = NULL
)

local_CondaEnv(
  new = list(),
  pathToMiniConda = NULL,
  path_action = "prefix",
  pythonpath_action = "replace",
  perl5lib_action = "replace",
  path_additional = NULL,
  pythonpath_additional = NULL,
  perl5lib_additional = NULL,
  .local_envir = parent.frame()
)

```

**Arguments**

<code>new</code>	The name of conda environment to include in the temporary R environment.
<code>code</code>	Code to execute in the temporary R environment
<code>pathToMiniConda</code>	Path to miniconda.
<code>path_action</code>	Should new values "replace", "prefix" or "suffix" existing PATH variable.
<code>pythonpath_action</code>	Should new values "replace", "prefix" or "suffix" existing PYTHONPATH variable.
<code>perl5lib_action</code>	Should new values "replace", "prefix" or "suffix" existing PERL5LIB variable.
<code>path_additional</code>	Additional paths to suffix to existing PATH variable.
<code>pythonpath_additional</code>	Additional paths to suffix to existing PYTHONPATH variable.
<code>perl5lib_additional</code>	Additional paths to suffix to existing PERL5LIB variable.
<code>.local_envir</code>	The environment to use for scoping.

**Value**

Nothing returned.



*with\_CondaEnv*

9

**Author(s)**

Thomas Carroll

**Examples**

```
local_CondaEnv(new = "herper_env")
```

# Index

`conda_search`, [2](#)

`export_CondaEnv`, [3](#)

`import_CondaEnv (export_CondaEnv)`, [3](#)

`install_CondaSysReqs`, [4](#)

`install_CondaTools`, [5](#)

`list_CondaEnv`, [6](#)

`list_CondaPkgs`, [7](#)

`local_CondaEnv (with_CondaEnv)`, [7](#)

`with_CondaEnv`, [7](#)