

# Package ‘SmoothR’

June 10, 2024

**Title** Manage R Sessions while using SLURM repetitively

**Version** 0.0.0.9000

**Description** Manage R Sessions while using SLURM repetitively

**License** GPL-3

**Imports** dplyr,  
lubridate,  
sessioninfo

**Suggests** testthat

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

## R topics documented:

AddFat . . . . .	2
checkDir . . . . .	2
checkpoint . . . . .	2
CopyRunSmoothRScript . . . . .	3
InitNow . . . . .	3
InitSmoothR . . . . .	4
print_args . . . . .	4
QuitNow . . . . .	4
ReadFromConfig . . . . .	5
RunNow . . . . .	5
SafeExecute . . . . .	6
SaveNow . . . . .	7
setAccount . . . . .	7
SetConfig . . . . .	8
setEmail . . . . .	8
UpdateConfig . . . . .	9
UpdateJobTime . . . . .	9
UpdateNumThreads . . . . .	10
UpdateOutputDir . . . . .	10
UpdatePartition . . . . .	10
UpdateSuffix . . . . .	11
<b>Index</b>	<b>12</b>

---

AddFat	<i>Add or Update FAT Configuration</i>
--------	--

---

### Description

Adds or updates the FAT setting in the configuration file. It validates and accepts only 'T', 'F', 't', 'f' as values. If no value is provided, the current configuration is printed.

### Usage

```
AddFat(fat_value = NULL)
```

### Arguments

fat_value	Optional character indicating the FAT setting ('T', 'F', 't', 'f'); default NULL.
-----------	---

---

checkDir	<i>Change Working Directory or Verify Current Directory</i>
----------	---

---

### Description

Verifies if the specified 'output\_dir' exists and changes the working directory to it. Throws an error if the directory does not exist or does not match the current directory.

### Usage

```
checkDir(output_dir)
```

### Arguments

output_dir	The target directory to validate or change to.
------------	--

---

checkpoint	<i>Print Custom Checkpoint Message in SLURM Output</i>
------------	--

---

### Description

Outputs a checkpoint message with a user-defined comment. Useful for tracking progress in non-interactive SLURM tasks.

### Usage

```
checkpoint(comment)
```

### Arguments

comment	A character string for the checkpoint comment.
---------	--

**Value**

Prints to the console; returns invisible NULL.

**Examples**

```
checkpoint("Data loaded successfully")
checkpoint("Model training started")
```

---

CopyRunSmoothRScript    *Copy runSmoothR.sh Script to a Specified Directory*

---

**Description**

This function copies the 'runSmoothR.sh' script from the 'shell\_helpers' folder of the package to a specified folder.

**Usage**

```
CopyRunSmoothRScript(dest_folder)
```

**Arguments**

dest\_folder      The destination folder where the script will be copied.

---

InitNow                      *Initialize Session After Slurm Execution*

---

**Description**

Loads a specified R session file. Aborts if run interactively or if the session file does not exist. Continues without loading if the file is missing.

**Usage**

```
InitNow(session_file_name = NULL)
```

**Arguments**

session\_file\_name  
The name of the session file to load, defaults to the first command line argument with ".RData" appended.

**Examples**

```
InitNow()
```

---

InitSmoothR

*Activate runSmoothR.sh by Copying and Adding to \$PATH*


---

### Description

This function copies 'runSmoothR.sh' to a specified directory and adds the directory to the \$PATH variable by updating the appropriate shell configuration file, only if the path is not already present.

### Usage

```
InitSmoothR(force = F)
```

---

print\_args

*Print Command Line Arguments*


---

### Description

This function prints all command line arguments provided to the R script when run from the command line. It will not execute in interactive sessions to ensure it is used specifically for command-line operations.

### Usage

```
print_args()
```

### Value

Prints each command-line argument passed to the script; returns invisible NULL in interactive mode.

---

QuitNow

*Quit R Session when not Interactive*


---

### Description

Terminates the current R session non-interactively with a success status. Useful in scripts and batch processing. Does not save the session before exiting. Pair with SaveNow() to save data first.

### Usage

```
QuitNow()
```

### Examples

```
QuitNow()
```

---

ReadFromConfig	<i>Read Configuration Values from a File</i>
----------------	--

---

**Description**

Retrieves a specific configuration value by key from a file formatted as 'export KEY='VALUE''. Throws an error if the key is not found or if the file does not exist.

**Usage**

```
ReadFromConfig(key_to_check)
```

**Arguments**

key_to_check	The configuration key to retrieve.
config_file	Path to the configuration file, defaults to '~/.temp_shell_exports'.

**Value**

The value associated with the provided key.

---

RunNow	<i>RunNow Function</i>
--------	------------------------

---

**Description**

This function is designed to submit R scripts to a Slurm job scheduler. It initializes the environment by reading configurations from a shell script located in the user's home directory. The function retrieves the 'OUTPUT\_DIR' environment variable, and submits the job using a designated Slurm script. The 'script\_name' variable is defined here and logged for subsequent analysis. If 'script\_name' is not redefined during execution, its initial value is used.

**Usage**

```
RunNow(script_name = NULL, verbose = TRUE)
```

**Arguments**

config_file	A string specifying the path to the configuration file for setting required environment variables. Defaults to "~/.temp_shell_exports".
output_file	A string specifying the path to the file where stdout and stderr from the Slurm job will be directed. If NULL, outputs will not be redirected. Defaults to NULL.

**Details**

Instead of using this function, one might find it convenient to use a multiplexed terminal for live reporting. This can be achieved by running: `./runSmoothR.sh script_name.R` [optional args, which will start from position 6 in args]

Note: The optional arguments are available starting from the sixth position in 'args'.

**Value**

The value of the ‘OUTPUT\_DIR’ environment variable, which indicates the output directory.

**Examples**

```
RunNow() # Uses default configuration file and no output redirection
RunNow(output_file = "path/to/output.log")
```

---

SafeExecute

*Safely Evaluate Expressions with Optional Logging*

---

**Description**

Evaluates an expression while handling errors and warnings to prevent unexpected script termination. If enabled, captures and logs messages, warnings, and errors to a file, facilitating smooth operation in production or critical scripts requiring uninterrupted execution.

**Usage**

```
SafeExecute(
  expr,
  logging = FALSE,
  log_file = "R_console_log_file.log",
  envir = parent.frame()
)
```

**Arguments**

expr	Expression to evaluate, which can range from a single command to a block of code.
logging	Boolean; enables logging of all runtime messages, warnings, and errors when set to TRUE.
log_file	String; specifies the file path for logging, effective only if ‘logging’ is TRUE.
envir	Evaluation environment, defaulting to the parent frame.

**Value**

Returns the result of the evaluated expression, or NULL in the event of an error or warning.

**Examples**

```
SafeExecute({
  x <- rnorm(100)
  if (mean(x) > 0) "Positive" else "Negative"
}, logging = TRUE, log_file = "run_log.txt")
```

---

SaveNow	<i>Save the Current R Session for Slurm Execution and Log Session Information</i>
---------	---

---

### Description

This function saves the current R session into a .RData file, enabling the session to be resumed in future Slurm job executions. It also logs session details into a timestamped text file, aiding in reproducibility and debugging. The session file name and timestamp can be specified; if not, defaults are used.

### Usage

```
SaveNow(
  session_file_name = NULL,
  timestamp = format(Sys.time(), "%Y%m%d-%H%M%S")
)
```

### Arguments

session_file_name	string (optional); the filename for saving the session. If not provided, the first command line argument will be used with "Data" appended.
timestamp	string (optional); the timestamp to append to the session info file. Defaults to the current datetime if not provided.

### Examples

```
SaveNow() # Save using default settings
```

---

setAccount	<i>Set USER_ACCOUNT Environment Variable</i>
------------	--

---

### Description

Sets the USER\_ACCOUNT environment variable to the specified account name and displays it.

### Usage

```
setAccount(accountName)
```

### Arguments

accountName	Account name to set as USER_ACCOUNT.
-------------	--------------------------------------

### Examples

```
setAccount("snic123456")
```

---

**SetConfig***Set Environment Configuration Interactively*

---

**Description**

Interactively prompts for various configuration settings and writes them to a shell script in the user's home directory. The script, formatted for environment variable export, is intended for shell sourcing. Includes conversion of job time from minutes to D-H:M:S format.

**Usage**

```
SetConfig(force = F)
```

**Arguments**

None                      Parameters are input interactively.

**Value**

No return value; outputs file path and contents to console.

**Examples**

```
set_config_interactively() # Run this in an interactive R session
```

---

**setEmail***Set USER\_E\_MAIL Environment Variable*

---

**Description**

Sets the USER\_E\_MAIL environment variable to the specified email and displays it.

**Usage**

```
setEmail(email)
```

**Arguments**

email                      Email address to set as USER\_E\_MAIL.

**Examples**

```
setEmail("user@example.com")
```



---

UpdateConfig	<i>Update Configuration Setting</i>
--------------	-------------------------------------

---

**Description**

This function updates a specific configuration key in the ‘.temp\_shell\_exports‘ file located in the user’s home directory. If the key does not exist, it adds the key with the specified value.

**Usage**

UpdateConfig(key, value)

**Arguments**

- |       |                                  |
|-------|----------------------------------|
| key   | The configuration key to update. |
| value | The new value for the key.       |

**Value**

Invisible NULL; updates are made in-place.

**Examples**

UpdateConfig("NUM\_THREADS", "8")

---

UpdateJobTime	<i>Update Job Time</i>
---------------	------------------------

---

**Description**

Update Job Time

**Usage**

UpdateJobTime(JOB\_TIME\_MINUTES)

**Arguments**

- |                  |                                 |
|------------------|---------------------------------|
| JOB_TIME_MINUTES | New job time to set in minutes. |
|------------------|---------------------------------|

---

UpdateNumThreads	<i>Update the Number of Threads</i>
------------------	-------------------------------------

---

**Description**

Update the Number of Threads

**Usage**

UpdateNumThreads(NUM\_THREADS)

**Arguments**

NUM\_THREADS      New number of threads to set.

---

UpdateOutputDir	<i>Update Output Directory</i>
-----------------	--------------------------------

---

**Description**

Update Output Directory

**Usage**

UpdateOutputDir(OUTPUT\_DIR)

**Arguments**

OUTPUT\_DIR      New output directory to set.

---

UpdatePartition	<i>Update Job Partition</i>
-----------------	-----------------------------

---

**Description**

Update Job Partition

**Usage**

UpdatePartition(PARTITION)

**Arguments**

PARTITION      New partition to set.

---

UpdateSuffix	<i>Update File Suffix</i>
--------------	---------------------------

---

**Description**

Update File Suffix

**Usage**

UpdateSuffix(SUFFIX)

**Arguments**

SUFFIX	New suffix for the files to set.
--------	----------------------------------

# Index

AddFat, [2](#)

checkDir, [2](#)

checkpoint, [2](#)

CopyRunSmoothRScript, [3](#)

InitNow, [3](#)

InitSmoothR, [4](#)

print\_args, [4](#)

QuitNow, [4](#)

ReadFromConfig, [5](#)

RunNow, [5](#)

SafeExecute, [6](#)

SaveNow, [7](#)

setAccount, [7](#)

SetConfig, [8](#)

setEmail, [8](#)

UpdateConfig, [9](#)

UpdateJobTime, [9](#)

UpdateNumThreads, [10](#)

UpdateOutputDir, [10](#)

UpdatePartition, [10](#)

UpdateSuffix, [11](#)