

# Package: slurmtools (via r-universe)

September 25, 2024

**Title** slurm tooling  
**Version** 0.0.0.9000  
**Description** What the package does (one paragraph).  
**License** GPL (>= 3)  
**Encoding** UTF-8  
**Roxygen** list(markdown = TRUE)  
**RoxygenNote** 7.3.2  
**Imports** brio, dplyr, fs, glue, jsonlite, processx, purrr, rlang,  
tibble, utils, whisker, withr, here, cli, stringi  
**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), bbr  
**Config/testthat/edition** 3  
**VignetteBuilder** knitr  
**Repository** <https://a2-ai.r-universe.dev>  
**RemoteUrl** <https://github.com/a2-ai/slurmtools>  
**RemoteRef** HEAD  
**RemoteSha** 638c111abbfc5b24b35faabf2b7216dbecd51fd

## Contents

check_slurm_partitions . . . . .	2
get_slurm_jobs . . . . .	2
get_slurm_partitions . . . . .	3
lookup_partitions_by_cpu . . . . .	3
partition_advice . . . . .	4
process_slurm_partitions . . . . .	4
run_sinfo . . . . .	5
submit_nonmem_model . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

check\_slurm\_partitions

*throws error if the number of requested CPUs exceeds the number of CPUs available in the requested partition*

---

### Description

throws error if the number of requested CPUs exceeds the number of CPUs available in the requested partition

### Usage

```
check_slurm_partitions(ncpu, partition, cache = TRUE)
```

### Arguments

ncpu	number of CPUs requested by user
partition	name of partition requested by user
cache	optional argument to forgo caching

### Examples

```
## Not run:
check_slurm_partitions(17, "cpu2mem4gb")
check_slurm_partitions(3, "cpu2mem4gb")
check_slurm_partitions(5, "cpu4mem32gb")
check_slurm_partitions(5, "cpu4mem32gb")
check_slurm_partitions(100, "cpu32mem128gb")
check_slurm_partitions(2, "cpu2mem4gb")

## End(Not run)
```

---

get\_slurm\_jobs            *get slurm jobs*

---

### Description

get slurm jobs

### Usage

```
get_slurm_jobs(user = NULL)
```

### Arguments

user	user filter
------	-------------

---

`get_slurm_partitions`    *get list of partition names for the given cluster*

---

**Description**

get list of partition names for the given cluster

**Usage**

```
get_slurm_partitions(cache = TRUE)
```

**Arguments**

cache                    optional argument to forgo caching

---

`lookup_partitions_by_cpu`  
*get table of each partition's number of CPUs and memory*

---

**Description**

- gets the raw string output from [run\\_sinfo\(\)](#)
- converts to a data frame
- reorders and removes \* from default partition with [process\\_slurm\\_partitions\(\)](#)

**Usage**

```
lookup_partitions_by_cpu(cache = TRUE)
```

**Arguments**

cache                    optional argument to forgo caching

**Value**

the processed table of each partition's number of CPUs and memory

---

partition_advice	<i>get partition suggestions</i>
------------------	----------------------------------

---

### Description

In a call to `submit_slurm_model()`, if the number of requested CPUs exceeds the number of CPUs available in the requested partition, `check_slurm_partitions()` errors. This function follows up with a message providing one or two suggestions for the partition with the smallest sufficient number of CPUs and least amount of memory. If there are no partitions with enough CPUs to accommodate the number requested, this function's return message clarifies this.

### Usage

```
partition_advice(ncpu, partition, avail_cpus_table, cache)
```

### Arguments

ncpu	number of CPUs requested by user
partition	name of partition requested by user
avail_cpus_table	table of partitions with respective number of CPUs and memory
cache	optional argument to forgo caching

### Value

string with suggestion upon `check_slurm_partitions()` error

---

process_slurm_partitions	<i>manipulate partition table for usability</i>
--------------------------	-------------------------------------------------

---

### Description

manipulate partition table for usability

### Usage

```
process_slurm_partitions(table)
```

### Arguments

table	the table created by <code>lookup_partitions_by_cpu</code>
-------	------------------------------------------------------------

### Value

the table such that the default partition will be first and will have the asterisk removed

---

run_sinfo	<i>get list of each partition's number of CPUs and memory</i>
-----------	---------------------------------------------------------------

---

**Description**

get list of each partition's number of CPUs and memory

**Usage**

```
run_sinfo()
```

**Value**

the raw partition-cpu-memory string output from sinfo

---

submit_nonmem_model	<i>submit a nonmem model to slurm in parallel</i>
---------------------	---------------------------------------------------

---

**Description**

submit a nonmem model to slurm in parallel

**Usage**

```
submit_nonmem_model(
  .mod,
  partition = get_slurm_partitions(),
  ncpu = 1,
  overwrite = FALSE,
  dry_run = FALSE,
  ...,
  slurm_job_template_path = getOption("slurmtools.slurm_job_template_path"),
  submission_root = getOption("slurmtools.submission_root"),
  bbi_config_path = getOption("slurmtools.bbi_config_path"),
  slurm_template_opts = list()
)
```

**Arguments**

.mod	a path to a model or a bbi nonmem model object
partition	name of the partition to submit the model
ncpu	number of cpus to run the model against
overwrite	whether to overwrite existing model results
dry_run	return the command that would have been invoked, without invoking

```
... arguments to pass to processx::run
slurm_job_template_path
    path to slurm job template
submission_root
    directory to track job submission scripts and output
bbi_config_path
    path to bbi.yaml file for bbi configuration
slurm_template_opts
    choose slurm template
```

# Index

check\_slurm\_partitions, 2  
check\_slurm\_partitions(), 4

get\_slurm\_jobs, 2  
get\_slurm\_partitions, 3

lookup\_partitions\_by\_cpu, 3

partition\_advice, 4  
process\_slurm\_partitions, 4  
process\_slurm\_partitions(), 3

run\_sinfo, 5  
run\_sinfo(), 3

submit\_nonmem\_model, 5