Package: hipergator (via r-universe)

May 17, 2026

Title HiPerGator SLURM Interface for R

Version 0.1.0

Description Simple interface for submitting and managing SLURM jobs on the University of Florida HiPerGator cluster. Provides authentication, data transfer, job submission, monitoring, and result retrieval with configuration management for streamlined workflows.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports cli, fs, glue, processx, rlang

Suggests testthat (>= 3.0.0)

Config/pak/sysreqs cmake make libuv1-dev

Repository https://flmnh-ai.r-universe.dev

Date/Publication 2025-10-14 01:20:49 UTC

RemoteUrl https://github.com/flmnh-ai/hipergator

RemoteRef HEAD

RemoteSha a2778f4170efb222b6d0a0a5ac065088ae613eb7

Contents

`ensure_hpg_job_env` *Check job status*

Description

Check job status

Usage

```
ensure_hpg_job_env(job)
```

Arguments

`job`	An `hpg_job` object
`refresh`	Force refresh from cluster (default: `FALSE`)

Value

Status string such as "PENDING", "RUNNING", or "COMPLETED"

hpg_authenticate *Authenticate with HiPerGator using SSH multiplexing*

Description

Establishes an SSH master connection to HiPerGator with Duo MFA support. Uses SSH multiplexing so subsequent connections are fast and don't require re-authentication.

Usage

```
hpg_authenticate(host = NULL, user = NULL, timeout = 60, quiet = FALSE)
```

Arguments

host	SSH hostname (uses configuration if NULL)
user	SSH username (uses configuration if NULL)
timeout	Connection timeout in seconds
quiet	Suppress status messages (default: FALSE)

Value

SSH target string for subsequent connections

Examples

```
## Not run:
# Use configuration defaults
target <- hpg_authenticate()

# Override defaults
target <- hpg_authenticate(host = "hpg", user = "myuser")

## End(Not run)
```

hpg_cancel *Cancel a running job*

Description

Cancel a running job

Usage

```
hpg_cancel(job)
```

Arguments

job	An hpg_job object

hpg_check_connection *Check if SSH master connection is active*

Description

Check if SSH master connection is active

Usage

```
hpg_check_connection(host = NULL, user = NULL)
```

Arguments

host	SSH hostname (uses configuration if NULL)
user	SSH username (uses configuration if NULL)

Value

Logical indicating if connection is active

hpg_cluster_config *Create cluster configuration*

Description

Create cluster configuration

Usage

```
hpg_cluster_config(host, user = NULL, port = 22)
```

Arguments

host	SSH hostname
user	SSH username
port	SSH port (default: 22 for password auth, use 2222 for SSH keys)

Value

A cluster configuration object

Examples

```
## Not run:
cluster <- hpg_cluster_config(host = "hpg.rc.ufl.edu", user = "myuser")
cluster_with_keys <- hpg_cluster_config(host = "hpg.rc.ufl.edu", user = "myuser", port = 2222)

## End(Not run)
```

hpg_config *Get flattened HiPerGator configuration*

Description

Returns a simple list containing commonly used fields (host, user, base_dir, and timeout values) derived from the structured configuration established via `hpg_configure()`.

Usage

```
hpg_config()
```

Value

List with host, user, port, base_dir, temp_dir, timeout_hours, poll_interval, and max_jobs.

hpg_configuration *Get current HiPerGator configuration*

Description

Get current HiPerGator configuration

Usage

```
hpg_configuration(show = FALSE)
```

Arguments

show	Whether to display configuration details

Value

Configuration list

hpg_configure *Configure HiPerGator settings*

Description

This is the primary entry point for defining how hipergator connects to the cluster, where it stores files, and how long it waits between status checks. You can pass structured configuration objects created with `hpg_cluster_config()`, `hpg_storage_config()`, and `hpg_defaults()`, or use the convenience arguments to update individual fields.

Usage

```
hpg_configure(
  host = NULL,
  user = NULL,
  port = NULL,
  base_dir = NULL,
  temp_dir = NULL,
  timeout_hours = NULL,
  poll_interval = NULL,
  max_jobs = NULL,
  cluster = NULL,
  storage = NULL,
  defaults = NULL
)
```

Arguments

`host`	Optional SSH hostname (character).
`user`	Optional SSH username (character).
`port`	Optional SSH port (numeric).
`base_dir`	Optional base directory for remote jobs (character).
`temp_dir`	Optional temporary directory under the base directory.
`timeout_hours`	Optional default timeout (hours).
`poll_interval`	Optional default polling interval (seconds).
`max_jobs`	Optional maximum concurrent jobs tracked.
`cluster`	Optional object produced by `hpg_cluster_config()`.
`storage`	Optional object produced by `hpg_storage_config()`.
`defaults`	Optional object produced by `hpg_defaults()`.

Value

Invisibly returns the complete structured configuration.

Examples

```
## Not run:
hpg_configure(host = "hpg", user = "myuser", base_dir = "/blue/group/project")

hpg_configure(
  cluster = hpg_cluster_config(host = "hpg", user = "myuser"),
  storage = hpg_storage_config(base_dir = "/blue/group/project")
)

## End(Not run)
```

---

hpg_defaults *Create default settings configuration*

Description

Create default settings configuration

Usage

```
hpg_defaults(timeout_hours = 8, poll_interval = 30, max_jobs = 100)
```

Arguments

timeout_hours	Default job timeout in hours
poll_interval	Default polling interval in seconds
max_jobs	Maximum concurrent jobs

Value

A defaults configuration object

Examples

```
## Not run:
defaults <- hpg_defaults(timeout_hours = 12, poll_interval = 60)

## End(Not run)
```

hpg_disconnect *Disconnect SSH master connection*

Description

Disconnect SSH master connection

Usage

```
hpg_disconnect(host = NULL, user = NULL)
```

Arguments

host	SSH hostname (uses configuration if NULL)
user	SSH username (uses configuration if NULL)

hpg_download *Download files from HiPerGator*

Description

Download files from HiPerGator

Usage

```
hpg_download(
  target,
  remote_path,
  local_path,
  verify_files = NULL,
  quiet = FALSE
)
```

Arguments

target	SSH target (from hpg_authenticate)
remote_path	Remote file or directory path
local_path	Local destination path
verify_files	Optional character vector of files that must exist
quiet	Suppress status messages (default: FALSE)

Examples

```
## Not run:
target <- hpg_authenticate()
hpg_download(target, "/blue/mygroup/project/output/", "results/")

## End(Not run)
```

`hpg_exists` *Check if remote path exists*

Description

Check if remote path exists

Usage

```
hpg_exists(target, remote_path)
```

Arguments

`target`	SSH target (from hpg_authenticate)
`remote_path`	Remote path to check

Value

Logical indicating if path exists

`hpg_get_error` *Get job error information*

Description

Get job error information

Usage

```
hpg_get_error(job)
```

Arguments

`job`	An `hpg_job` object

Value

List containing error output and exit code

hpg_gpu *Create GPU resource specification*

Description

Create GPU resource specification

Usage

```
hpg_gpu(count = 1, type = NULL, memory = NULL)
```

Arguments

count	Number of GPUs
type	GPU type (e.g., "a100", "b200", "rtx6000")
memory	GPU memory requirement (optional)

Value

A GPU resource object

Examples

```
## Not run:
gpu <- hpg_gpu(count = 2, type = "a100")
gpu <- hpg_gpu(count = 1, type = "rtx6000", memory = "24gb")

## End(Not run)
```

hpg_job_info *Get detailed job information*

Description

Get detailed job information

Usage

```
hpg_job_info(job)
```

Arguments

job	An hpg_job object

Value

Named list with job metadata

hpg_mkdir *Create directories on HiPerGator*

Description

Create directories on HiPerGator

Usage

```
hpg_mkdir(target, remote_paths, quiet = FALSE)
```

Arguments

target	SSH target (from hpg_authenticate)
remote_paths	Character vector of remote directory paths to create
quiet	Suppress status messages (default: FALSE)

Examples

```
## Not run:
target <- hpg_authenticate()
hpg_mkdir(target, c("/blue/mygroup/project/data", "/blue/mygroup/project/output"))

## End(Not run)
```

hpg_reset_config *Reset HiPerGator configuration to defaults*

Description

Reset HiPerGator configuration to defaults

Usage

```
hpg_reset_config()
```

hpg_reset_configuration *Reset configuration to defaults*

Description

Reset configuration to defaults

Usage

```
hpg_reset_configuration()
```

`hpg_resources` *Create HiPerGator resource specification*

Description

Create HiPerGator resource specification

Usage

```
hpg_resources(
  cores = 4,
  memory = "24gb",
  time = "02:00:00",
  partition = "hpg-turin",
  gpu = NULL,
  modules = NULL,
  conda_env = NULL,
  exclusive = FALSE
)
```

Arguments

`cores`	Number of CPU cores
`memory`	Memory requirement (e.g., "24gb", "500mb")
`time`	Time limit in HH:MM:SS format
`partition`	SLURM partition
`gpu`	GPU specification from hpg_gpu() or NULL
`modules`	Character vector of modules to load
`conda_env`	Conda environment to activate
`exclusive`	Whether to request exclusive node access

Value

A resource specification object

Examples

```
## Not run:
# Basic CPU job
resources <- hpg_resources(cores = 8, memory = "32gb", time = "04:00:00")

# GPU job
resources <- hpg_resources(
  cores = 16,
  memory = "64gb",
  time = "12:00:00",
```

```
  gpu = hpg_gpu(count = 2, type = "l4"),
  partition = "hpg-turin"
)

# With environment setup
resources <- hpg_resources(
  cores = 4,
  memory = "16gb",
  time = "02:00:00",
  modules = c("conda", "cuda"),
  conda_env = "/blue/group/envs/pytorch"
)

## End(Not run)
```

`hpg_show_base_dir` *Show current base directory and explain path resolution*

Description

Show current base directory and explain path resolution

Usage

```
hpg_show_base_dir()
```

`hpg_show_config` *Show current configuration*

Description

Show current configuration

Usage

```
hpg_show_config()
```

hpg_storage_config *Create storage configuration*

Description

Create storage configuration

Usage

```
hpg_storage_config(base_dir, temp_dir = NULL)
```

Arguments

base_dir	Base directory for job files
temp_dir	Temporary directory (optional)

Value

A storage configuration object

Examples

```
## Not run:
storage <- hpg_storage_config(base_dir = "/blue/mygroup/projects")

## End(Not run)
```

hpg_submit *Submit a job to HiPerGator*

Description

This is the primary entry point for launching work on the cluster. It uses the structured configuration helpers for safe path resolution and produces the unified job object described in new_hpg_job().

Usage

```
hpg_submit(
  resources,
  command,
  job_name = NULL,
  working_dir = ".",
  ssh_target = NULL,
  quiet = FALSE,
  ...
)
```

Arguments

resources	Resource specification created with hpg_resources()
command	Command to execute remotely
job_name	Optional SLURM job name (defaults to a timestamped value)
working_dir	Remote working directory relative to configured base dir
ssh_target	Optional SSH target override (defaults to configuration)
quiet	Suppress status messages (default: FALSE)
...	Named environment variables to export before running the command

Value

An object of class hpg_job

Examples

```
## Not run:
resources <- hpg_resources(cores = 8, memory = "32gb", time = "04:00:00")
job <- hpg_submit(resources, "python train.py", job_name = "training_run")
hpg_wait(job)

## End(Not run)
```

---

hpg_suggest_partitions

Get partition suggestions based on resources

Description

Get partition suggestions based on resources

Usage

```
hpg_suggest_partitions(resources)
```

Arguments

resources	An hpg_resources object

Value

Character vector of suggested partitions

hpg_upload *Upload files to HiPerGator*

Description

Upload files to HiPerGator

Usage

```
hpg_upload(
  target,
  local_path,
  remote_path,
  exclude_patterns = c("__pycache__", "*.pyc", ".git", ".DS_Store"),
  delete_extra = FALSE,
  quiet = FALSE
)
```

Arguments

target	SSH target (from hpg_authenticate)
local_path	Local file or directory path
remote_path	Remote destination path
exclude_patterns	Patterns to exclude from transfer
delete_extra	Remove files in destination that don't exist in source
quiet	Suppress status messages (default: FALSE)

Examples

```
## Not run:
target <- hpg_authenticate()
hpg_upload(target, "my_data/", "/blue/mygroup/project/data/")

## End(Not run)
```

hpg_wait *Wait for job completion*

Description

Wait for job completion

Usage

```
hpg_wait(
  job,
  timeout_hours = NULL,
  poll_interval = NULL,
  show_progress = TRUE,
  log_path = NULL,
  progress_callback = NULL
)
```

Arguments

job	An hpg_job object
timeout_hours	Maximum hours to wait (defaults from configuration)
poll_interval	Poll interval in seconds (defaults from configuration)
show_progress	Display progress messages
log_path	Optional remote path to monitor for progress updates
progress_callback	Optional function called with log contents during each poll

Value

The job object (invisibly)

---

print.hpg_job *Print method for* hpg_job

Description

Print method for hpg_job

Usage

```
## S3 method for class 'hpg_job'
print(x, ...)
```

Index