

# Package ‘neurohcp’

May 9, 2026

**Type** Package

**Title** Human 'Connectome' Project Interface

**Version** 0.11.0

**Description** Downloads and reads data from Human 'Connectome' Project  
<<https://db.humanconnectome.org>> using Amazon Web Services ('AWS')  
'S3' buckets.

**License** GPL-2

**LazyData** true

**LazyLoad** true

**Depends** R (>= 2.10)

**Imports** digest, httr, utils, xml2 (>= 1.0.1), base64enc, aws.s3,  
methods

**Suggests** dplyr, knitr, rmarkdown, covr

**RoxygenNote** 7.3.2

**URL** <https://db.humanconnectome.org>

**BugReports** <https://github.com/muschelli2/neurohcp/issues>

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**Author** John Muschelli [aut, cre],  
Adi Gherman [ctb]

**Maintainer** John Muschelli <[muschelli2@gmail.com](mailto:muschelli2@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-04-01 18:20:05 UTC

## Contents

bucketlist . . . . .	2
download_hcp_dir . . . . .	3
download_hcp_file . . . . .	4
get_hcp_file . . . . .	5
hcp_1200_scanning_info . . . . .	5
hcp_900_scanning_info . . . . .	6
hcp_aws_url . . . . .	6
hcp_ids . . . . .	7
hcp_list_files . . . . .	8
hcp_scanning_info . . . . .	9
make_aws_call . . . . .	10
parse_list_files . . . . .	11
set_aws_api_key . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

bucketlist	<i>Retrieve S3 Bucket List</i>
------------	--------------------------------

---

### Description

Lists the Buckets available with the Key/Secret

### Usage

```
bucketlist(region = "us-east-1", access_key = NULL, secret_key = NULL, ...)
```

### Arguments

region	Region of S3 Bucket
access_key	Amazon S3 Access Key
secret_key	Amazon S3 Secret Key
...	arguments to pass to <a href="#">bucket_list_df</a>

### Value

List of Buckets

### Examples

```
if (have_aws_key()){
  bucketlist()
}
```

---

download_hcp_dir	<i>Download an entire directory from HCP</i>
------------------	----------------------------------------------

---

### Description

Downloads a directory/folder from HCP database

### Usage

```
download_hcp_dir(  
  prefix,  
  delimiter = "",  
  outdir = tempfile(),  
  verbose = TRUE,  
  ...  
)  
  
download_fcp_dir(...)  
  
download_openneuro_dir(...)
```

### Arguments

prefix	Folder to download
delimiter	Delimiter for files
outdir	Output directory
verbose	Should diagnostic values be printed?
...	additional arguments to pass to <a href="#">hcp_list_files</a>

### Value

List of return from calling [hcp\\_list\\_files](#), the output directory, and all destination files (not subsetted by those that did in fact download)

### Examples

```
if (have_aws_key()) {  
  prefix = "HCP/100307/release-notes"  
  res = download_hcp_dir(prefix = prefix, verbose = FALSE)  
}
```

---

download\_hcp\_file      *Download HCP file*

---

## Description

Downloads a file from HCP S3 Bucket

## Usage

```
download_hcp_file(  
  path_to_file,  
  destfile = NULL,  
  verbose = TRUE,  
  error = TRUE,  
  ...  
)  
  
download_fcp_file(...)  
  
download_openneuro_file(...)
```

## Arguments

path_to_file	Path to file on HCP S3 Bucket
destfile	Destination filename
verbose	should progress be added to downloading?
error	Should the function error if the return was bad?
...	arguments to pass to <a href="#">hcp_aws_url</a>

## Value

Output filename that was downloaded

## Examples

```
if (have_aws_key()){  
  path_to_file <- "HCP_900/100206/MNINonLinear/100206.164k_fs_LR.wb.spec"  
  download_hcp_file(path_to_file = path_to_file)  
}
```

---

get_hcp_file	<i>Get HCP file</i>
--------------	---------------------

---

**Description**

Wraps a [make\\_aws\\_call](#) to a [GET](#) statement to get the file

**Usage**

```
get_hcp_file(path_to_file = "/", ..., verbose = TRUE, verb = "GET")
```

```
head_hcp_file(...)
```

```
get_fcp_file(...)
```

```
head_fcp_file(...)
```

```
get_openneuro_file(...)
```

```
head_openneuro_file(...)
```

**Arguments**

path_to_file	Path to file on HCP S3 Bucket
...	arguments to pass to <a href="#">make_aws_call</a>
verbose	Should the URL be printed?
verb	passed to <a href="#">VERB</a>

**Value**

Result of GET

---

hcp_1200_scanning_info	<i>Scanning Information for HCP 1200 Data</i>
------------------------	-----------------------------------------------

---

**Description**

A data.frame with all the available scanning information for the HCP 1200 data.

**Usage**

```
hcp_1200_scanning_info
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 69615 rows and 18 columns.

---

hcp\_900\_scanning\_info *Scanning Information for HCP 900 Data*

---

### Description

A data.frame with all the available scanning information for the HCP 900 data.

### Usage

```
hcp_900_scanning_info
```

### Format

An object of class tbl\_df (inherits from tbl, data.frame) with 34406 rows and 18 columns.

---

hcp\_aws\_url *Construct AWS S3 String*

---

### Description

Constructs url string for AWS S3 Bucket to pass in for HCP downloading

### Usage

```
hcp_aws_url(
  path_to_file = "",
  bucket = "hcp-openaccess",
  region = "us-east-1",
  access_key = NULL,
  secret_key = NULL,
  lifetime_minutes = 20,
  query = NULL,
  verb = "GET",
  sign = TRUE
)
```

### Arguments

path_to_file	Path to file on HCP S3 Bucket
bucket	Bucket to download from
region	Region of S3 Bucket
access_key	Amazon S3 Access Key
secret_key	Amazon S3 Secret Key

lifetime_minutes	Time that connection can be opened
query	additional query to add to url
verb	httr VERB to be used
sign	Should the url be signed?

**Value**

Character of the url to be passed to httr 'VERB's

**Examples**

```
if (have_aws_key()){
  path_to_file <- "HCP_900/100206/MNINonLinear/100206.164k_fs_LR.wb.spec"
  hcp_aws_url(path_to_file)
}
```

---

hcp\_ids

*Get list of HCP Ids*

---

**Description**

Returns list of IDs of folders in the HCP database

**Usage**

```
hcp_ids(group = c("HCP", "HCP_900", "HCP_1200", "HCP_Retest"))
```

**Arguments**

group            Group of IDs to return

**Value**

Character vector

**Examples**

```
if (have_aws_key()) {
  head(hcp_ids("HCP"))
  head(hcp_ids("HCP_900"))
}
```

---

hcp_list_files	<i>List files from HCP bucket</i>
----------------	-----------------------------------

---

### Description

Lists a set of files from an HCP bucket for a specific sub-folder

### Usage

```
hcp_list_files(
  prefix = "",
  delimiter = NULL,
  query = NULL,
  marker = NULL,
  ...,
  sign = TRUE
)
```

```
hcp_list_files(prefix = "", delimiter = NULL, query = NULL, marker = NULL, ...)
```

```
openneuro_list_files(
  prefix = "",
  delimiter = NULL,
  query = NULL,
  marker = NULL,
  ...
)
```

```
hcp_list_dirs(prefix = "HCP/", ...)
```

```
hcp_list_dirs(prefix = "data/Projects/", ...)
```

```
openneuro_list_dirs(prefix = NULL, ...)
```

### Arguments

prefix	directory folder to list files. If "", then it will be the root path
delimiter	Delimiter to list files. For example ".mat", for ".mat"
query	Additional query arguments
marker	the marker to start the listing, needed for pagination of results
...	additional arguments passed to <a href="#">get_hcp_file</a>
sign	Should the url be signed with an API key?

### Value

List with the result of the GET command, the parsed result, and the content from the result.

**Examples**

```

if (have_aws_key()){
x = hcp_list_files(prefix = "HCP/100307/unprocessed/3T/Diffusion",
  delimiter="bval")
stopifnot(x$Parsed_result$ListBucketResult$Name[[1]] == "hcp-openaccess")
t1_niis = hcp_list_files(prefix = "HCP/100307/T1w",
  delimiter = ".nii.gz")
all_dirs = hcp_list_dirs("HCP/")
}

if (have_aws_key()){
res = hcp_list_dirs("HCP/")
projects = unlist(parse_list_files(res)$prefixes)
projects = unname(projects)
projects = unname(projects)
head(projects)
head(basename(projects))
stopifnot("100307" %in% basename(projects))
}
if (have_aws_key()){
res = fcp_list_dirs()
projects = unlist(parse_list_files(res)$prefixes)
projects = unname(projects)
head(projects)
head(basename(projects))
stopifnot("ABIDE" %in% basename(projects))
}
res = openneuro_list_dirs()
projects = unlist(parse_list_files(res)$prefixes)
projects = unname(projects)
head(projects)
if (length(projects) > 0) {
head(basename(projects))
stopifnot("ds000002" %in% basename(projects))
}
}

```

---

hcp\_scanning\_info

*Scanning Information for HCP Data*


---

**Description**

A data.frame with all the available scanning information for the HCP data.

**Usage**

```
hcp_scanning_info
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 34406 rows and 18 columns.

---

make_aws_call	<i>Construct AWS S3 Call</i>
---------------	------------------------------

---

**Description**

Constructs GET information string for AWS S3 Bucket

**Usage**

```
make_aws_call(
  path_to_file = "/",
  bucket = "hcp-openaccess",
  region = "us-east-1",
  access_key = NULL,
  secret_key = NULL,
  lifetime_minutes = 5,
  query = NULL,
  verb = "GET",
  sign = TRUE
)
```

**Arguments**

path_to_file	Path to file on HCP S3 Bucket
bucket	Bucket to download from
region	Region of S3 Bucket
access_key	Amazon S3 Access Key
secret_key	Amazon S3 Secret Key
lifetime_minutes	Time that connection can be opened
query	additional query to add to verb command
verb	httr VERB to be used
sign	Should the url be signed?

**Value**

Character of the url to be passed to httr 'VERB's

**Examples**

```
if (have_aws_key()){
  path_to_file <- paste0(
    "HCP_900/100206/MNINonLinear/",
    "100206.164k_fs_LR.wb.spec")
  result = make_aws_call(path_to_file)
```

```
# don't want this stuff printed because it has keys
result$headers$secret_key = NULL
result$headers$access_key = NULL
result$query$AWSAccessKeyId = NULL
result$query$Signature = NULL
result
}
```

---

parse_list_files	<i>Parse listed files from HCP bucket</i>
------------------	-------------------------------------------

---

### Description

This parses the result from [hcp\\_list\\_files](#) and organizes the files into `data.frames`

### Usage

```
parse_list_files(ret)
```

### Arguments

`ret` object with element `parsed_result`, usually from [hcp\\_list\\_files](#)

### Value

List of 2 `data.frames`, the contents and the `commonprefixes` elements from the list

### Examples

```
if (have_aws_key()){
  ret = hcp_list_files(prefix = "HCP/100307/unprocessed/3T/Diffusion")
  parsed = parse_list_files(ret)
  stopifnot(!is.null(parsed$contents))
}
```

---

set_aws_api_key	<i>Set Amazon AWS Key</i>
-----------------	---------------------------

---

### Description

Sets and returns the AWS keys. This will error if not all are specified.

**Usage**

```
set_aws_api_key(  
    access_key = NULL,  
    secret_key = NULL,  
    default_region = "us-east-1",  
    error = TRUE  
)  
  
have_aws_key()
```

**Arguments**

access_key	Amazon access key. If NULL then looks at the AWS_ACCESS_KEY_ID system variable.
secret_key	Amazon secret key. If NULL then looks at the AWS_SECRET_ACCESS_KEY system variable.
default_region	Amazon default region. If NULL then looks at the AWS_DEFAULT_REGION system variable.
error	Should this function error if things are not specified?

**Value**

List of access\_key, secret\_key, and default\_region

# Index

## \* datasets

- hcp\_1200\_scanning\_info, 5
- hcp\_900\_scanning\_info, 6
- hcp\_scanning\_info, 9

bucket\_list\_df, 2  
bucketlist, 2

download\_fcp\_dir (download\_hcp\_dir), 3  
download\_fcp\_file (download\_hcp\_file), 4  
download\_hcp\_dir, 3  
download\_hcp\_file, 4  
download\_openneuro\_dir  
    (download\_hcp\_dir), 3  
download\_openneuro\_file  
    (download\_hcp\_file), 4

fcp\_list\_dirs (hcp\_list\_files), 8  
fcp\_list\_files (hcp\_list\_files), 8

GET, 5  
get\_fcp\_file (get\_hcp\_file), 5  
get\_hcp\_file, 5, 8  
get\_openneuro\_file (get\_hcp\_file), 5

have\_aws\_key (set\_aws\_api\_key), 11  
hcp\_1200\_scanning\_info, 5  
hcp\_900\_scanning\_info, 6  
hcp\_aws\_url, 4, 6  
hcp\_ids, 7  
hcp\_list\_dirs (hcp\_list\_files), 8  
hcp\_list\_files, 3, 8, 11  
hcp\_scanning\_info, 9  
head\_fcp\_file (get\_hcp\_file), 5  
head\_hcp\_file (get\_hcp\_file), 5  
head\_openneuro\_file (get\_hcp\_file), 5

make\_aws\_call, 5, 10

openneuro\_list\_dirs (hcp\_list\_files), 8  
openneuro\_list\_files (hcp\_list\_files), 8

parse\_list\_files, 11

set\_aws\_api\_key, 11

VERB, 5