

# Package ‘GEOfastq’

December 24, 2024

**Type** Package

**Title** Downloads ENA Fastqs With GEO Accessions

**Version** 1.15.0

**Description** GEOfastq is used to download fastq files from the European Nucleotide Archive (ENA) starting with an accession from the Gene Expression Omnibus (GEO). To do this, sample metadata is retrieved from GEO and the Sequence Read Archive (SRA). SRA run accessions are then used to construct FTP and aspera download links for fastq files generated by the ENA.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**BugReports** <https://github.com/alexvpickering/GEOfastq/issues>

**Imports** xml2, rvest, stringr, RCurl, doParallel, foreach, plyr

**Suggests** BiocCheck, roxygen2, knitr, rmarkdown, testthat

**biocViews** RNASeq, DataImport

**VignetteBuilder** knitr

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

**git\_branch** devel

**git\_last\_commit** 7ac929a

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-12-24

**Author** Alex Pickering [cre, aut] (ORCID:  
<<https://orcid.org/0000-0002-0002-6759>>)

**Maintainer** Alex Pickering <[alexvpickering@gmail.com](mailto:alexvpickering@gmail.com)>

## Contents

ascpR . . . . .	2
crawl_gse . . . . .	2
crawl_gsms . . . . .	3
extract_gsms . . . . .	4
get_dldir . . . . .	4
get_ebi_fastqs . . . . .	5
get_fastqs . . . . .	5
<b>Index</b>	<b>7</b>

---

ascpR	<i>Utility function to run aspera</i>
-------	---------------------------------------

---

### Description

Utility function to run aspera

### Usage

```
ascpR(ascp_args, file, destDir = getwd())
```

### Arguments

ascp_args	Character vector of arguments to ascp.
file	Url to aspera file to download.
destDir	Path to directory to download files into.

### Value

return code from call to ascp

---

crawl_gse	<i>Get GSE text from GEO</i>
-----------	------------------------------

---

### Description

Get GSE text from GEO

### Usage

```
crawl_gse(gse_name)
```

### Arguments

gse_name	GEO study name to get metadata for
----------	------------------------------------

**Value**

Character vector of lines on GSE record.

**Examples**

```
gse_text <- crawl_gse('GSE111459')
```

---

crawl_gsms	<i>Crawls SRX pages for each GSM to get metadata.</i>
------------	---

---

**Description**

Goes to each GSM page to get SRX then to each SRX page to get some more metadata.

**Usage**

```
crawl_gsms(gsm_names, max.workers = 50)
```

**Arguments**

gsm_names	Character vector of GSMs.
max.workers	Maximum number of parallel workers to split task between

**Value**

data.frame

**Examples**

```
srp_meta <- crawl_gsms("GSM3031462")

# returns NULL because records on dbGAP for privacy reasons
srp_meta <- crawl_gsms("GSM2439650")

# example with empty values
srp_meta <- crawl_gsms('GSM4043025')
```

---

extract_gsms	<i>Extract GSMS needed to download RNA-seq data for a series</i>
--------------	--

---

**Description**

Extract GSMS needed to download RNA-seq data for a series

**Usage**

```
extract_gsms(gse_text)
```

**Arguments**

gse\_text            GSE text returned from [crawl\\_gse](#)

**Value**

Character vector of sample GSMS for the series gse\_name

**Examples**

```
gse_text <- crawl_gse('GSE111459')
gsm_names <- extract_gsms(gse_text)
```

---

get_dldir	<i>Gets part of path to download bulk RNAseq sample from EBI or NCBI</i>
-----------	--

---

**Description**

Gets part of path to download bulk RNAseq sample from EBI or NCBI

**Usage**

```
get_dldir(srr, type = c("ebi", "ncbi"))
```

**Arguments**

srr                SRR/ERR run name  
type               Either 'ebi' or 'ncbi'

**Value**

String path used by [get\\_fastqs](#).

**Examples**

```
get_dldir('SRR014242')
```

---

get_ebi_fastqs	<i>Download fastqs from EBI</i>
----------------	---------------------------------

---

**Description**

Much faster to use aspera than ftp

**Usage**

```
get_ebi_fastqs(
  srp_meta,
  srr_name,
  data_dir,
  method = c("ftp", "aspera"),
  max_rate = "300m"
)
```

**Arguments**

srp_meta	data.frame with SRP meta info. Returned from <a href="#">crawl_gsms</a> .
srr_name	Run accession as string.
data_dir	Path to folder that fastq files will be downloaded to. Will be created if doesn't exist.
method	One of 'aspera' or 'ftp'. 'aspera' is generally faster but requires the ascp command line utility to be on your path and in the authors experience frequently stalls.
max_rate	Used when method = 'aspera' only. Sets the target transfer rate. The default is '300m'.

**Value**

Integer return code from ascp or download.file.

---

get_fastqs	<i>Download and RNA-seq fastq data from EBI</i>
------------	---

---

**Description**

First tries to get RNA-Seq fastq files from EBI.

**Usage**

```
get_fastqs(srp_meta, data_dir, method = c("ftp", "aspera"), max_rate = "1g")
```

**Arguments**

<code>srp_meta</code>	data.frame with SRP meta info. Returned from <a href="#">crawl_gsms</a> .
<code>data_dir</code>	Path to folder that fastq files will be downloaded to. Will be created if doesn't exist.
<code>method</code>	One of 'aspera' or 'ftp'. 'aspera' is generally faster but requires the ascp command line utility to be on your path and in the authors experience frequently stalls.
<code>max_rate</code>	Used when method = 'aspera' only. Sets the target transfer rate. The default is '300m'.

**Value**

Named vector of integer return codes from ascp or `download.file`. Names are SRR runs.

**Examples**

```
gsm_name <- 'GSM3926903'  
srp_meta <- crawl_gsms(gsm_name)  
data_dir <- tempdir()  
res <- get_fastqs(srp_meta, data_dir)
```

# Index

## \* **internal**

ascpR, 2

get\_ebi\_fastqs, 5

ascpR, 2

crawl\_gse, 2, 4

crawl\_gsms, 3, 5, 6

extract\_gsms, 4

get\_dldir, 4

get\_ebi\_fastqs, 5

get\_fastqs, 4, 5