

# Package ‘GEOfastq’

April 12, 2022

**Type** Package

**Title** Downloads ENA Fastqs With GEO Accessions

**Version** 1.2.0

**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** RELEASE\_3\_14

**git\_last\_commit** 866d6af

**git\_last\_commit\_date** 2021-10-26

**Date/Publication** 2022-04-12

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

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

## R topics documented:

<code>crawl_gse</code> . . . . .	2
<code>crawl_gsms</code> . . . . .	2

extract_gsms . . . . .	3
get_dldir . . . . .	4
get_fastqs . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

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

*Extract GSMs needed to download RNA-seq data for a series*

---

**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_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**

srp_meta	data.frame with SRP meta info. Returned from <a href="#">crawl_gsms</a> .
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**

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

`crawl_gse`, 2, 3  
`crawl_gsms`, 2, 4  
`extract_gsms`, 3  
`get_dldir`, 4  
`get_fastqs`, 4, 4