

# Package ‘mmnet’

November 8, 2016

**Type** Package

**Title** A metagenomic pipeline for systems biology

**Version** 1.12.0

**Date** 2014-7-15

**Author** Yang Cao, Fei Li

**Maintainer** Yang Cao <yiluheihe@gmail.com>, Fei Li  
<pittacus@gmail.com>

**Description** This package gives the implementations microbiome metabolic network constructing and analyzing. It introduces a unique metagenomic systems biology approach, mapping metagenomic data to the KEGG global metabolic pathway and constructing a systems-level network. The system-level network and the next topological analysis will be of great help to analysis the various functional properties, including regulation and metabolic functionality of the metagenome.

**Depends** R (>= 2.14), igraph, biom

**Imports** Biobase, RJSONIO, stringr, reshape2, ggplot2, KEGGREST, plyr, XML, RCurl, flexmix, Matrix, methods, tools

**Suggests** RCytoscape, graph, knitr

**VignetteBuilder** knitr

**License** GPL (>= 2)

**LazyLoad** yes

**biocViews** GraphsAndNetwork, Sequencing, Pathways, Microbiome, SystemsBiology

**NeedsCompilation** no

## R topics documented:

mmnet-package . . . . .	2
anno . . . . .	2
checkMgrastMetagenome . . . . .	3
constructMetabolicNetwork . . . . .	4
constructSSN . . . . .	4
delMgrastInbox . . . . .	5
differentialAnalyzeNet . . . . .	6
estimateAbundance . . . . .	7

generateMgrastWebkey . . . . .	8
generateMgrastWebsession . . . . .	9
getKOPathwayInfo . . . . .	9
getMgrastAnnotation . . . . .	10
listMgrastInbox . . . . .	11
listMgrastProject . . . . .	12
loadMetabolicData . . . . .	12
loginMgrast . . . . .	13
preprocessKOMetabolites . . . . .	14
RefDbcache . . . . .	14
saveMetabolicData . . . . .	15
showMetagenomicNet . . . . .	15
submitMgrastJob . . . . .	16
topologicalAnalyzeNet . . . . .	17
updateKEGGPathway . . . . .	18
uploadMgrast . . . . .	19

<b>Index</b>	<b>21</b>
--------------	-----------

---

mmnet-package	<i>A metagenomic pipeline for systems biology</i>
---------------	---

---

## Description

The **mmnet** add-on is an implementation of mapping metagenomic data to the KEGG global metabolic pathway and constructing a systems-level network. It is of great help to analysis the various functional properties, including regulation and metabolic functionality of the metagenome.

## Details

This package contains functions for mapping metagenomic data to the KEGG global metabolic pathway and constructing systems-level network. First, the information of enzymes, corresponding substrates and products is parsed from the KEGG PATYWAY database with function [updateKEGGPathway](#). Second, sequence functional ontology annotation with KEGG is preprocessing on MGRAST, KO abundance can be obtained by parse the MGRAST ontology annotation files with function [estimateAbundance](#). Finally, global metabolic pathway and systems-level network are made to analyze the metabolic functionality of metagenome with function [constructMetabolicNetwork](#), visualization with function [showMetagenomicNet](#)

---

anno	<i>annotation profile of two sample data sets</i>
------	---

---

## Description

The data give the functional annotation profiles of two sample data sets.

## Usage

```
data(anno)
```

**Format**

The format is: List of 4 each element represents functional annotation profile of one sample

**Details**

Metagenome ID

obese mgm4440613.3

lean mgm4440823.3

**Examples**

```
data(anno)
```

---

checkMgrastMetagenome *Check whether the MGRAST annotation is completed*

---

**Description**

Check the whether MGRAST annotation is completed of the metagenomic sequence.

**Usage**

```
checkMgrastMetagenome(login.info, metagenome.id, public = TRUE)
```

**Arguments**

`login.info` login.info a list of login info generated after user login into MGRAST which consists of webkey of the file, the invalid time of your webkey, cookie, websession and curlhandle. It is unique for each users. Only needed while checking the private project.

`metagenome.id` a character, metagenome.id the metagenome id MGRAST assigned while user submit metagenome sequence to MGRAST.

`public` logical, whether the metagenome of your MGRAST project is public

**Value**

logical, whether the metagenome annotation is completed, it will be NULL while metagenome annotation is not completed.

**Examples**

```
checkMgrastMetagenome(metagenome.id = "4440616.3")
```

---

`constructMetabolicNetwork`*Metabolic network construction of microbiome*

---

**Description**

Metabolic network construction according to the metabolites among enzymes

**Usage**

```
constructMetabolicNetwork(path = Sys.getenv("HOME"))
```

**Arguments**

`path` character, the path that reference data saved, default is the user's HOME directory. If does not exists, prestored reference metabolic data (RefDbCache) in *mmnet* will be loaded.

**Value**

referece metabolic network of igraph

**See Also**

[loadMetabolicData](#)

**Examples**

```
## may take a long time
# data(RefDbcache)
# constructMetabolicNetwork(RefDbcache)
```

---

`constructSSN`*construct the state specific network*

---

**Description**

Given a set of KOs of a sample with a specific state and the reference metabolic network, this function creates and returns sub-network with only the supplied nodes (KOs) and any edges between them.

**Usage**

```
constructSSN(abundance)
```

**Arguments**

abundance abundance of KOs of a given specific state, may a numeric vector which can be obtained with [estimateAbundance](#)  
 a given Biological Observation Matrix (BIOM) format, the ncol of BIOM file represents number of samples. The BIOM file format (canonically pronounced biome) is designed to be a general-use format for representing biological sample by observation contingency tables. More details on BIOM file can be found in <http://biom-format.org/> R package *biom*. There is also a example BIOM file in directory "inst/extdata" of *mmnet*.  
 It could be obtained with [estimateAbundance](#). Moreover, while samples has been annotated with other tools (e.g. blast to KEGG locally), users could create their own BIOM file represents enzymatic genes abundance.

**Details**

Reference metabolic network has been saved in the RefDbcache. There are varies sets of KOs and its abundance between metagenomic samples. With this function, users can create a specified sub-network of the reference network for each sample.

**Value**

new graph object will be return, and the abundance is added as a vertex meta-attributes. The number of graph is equal to the samples.

**See Also**

[RefDbcache](#), [estimateAbundance](#)

**Examples**

```
# load reference metabolic network
data(RefDbcache)

data(anno)
abund <- estimateAbundance(anno[[1]])
subnet <- constructSSN(abund)
```

---

delMgrastInbox

*Delete the files in you MG-RAST inbox*


---

**Description**

Once job is submiited to MG-RAST, user can delete the redundant sequence.

**Usage**

```
delMgrastInbox(login.info, sequence)
```

**Arguments**

login.info web information while login to MGRAST , more details see [loginMgrast](#)  
 sequence character, represents the file your will delete

**Details**

Users can delete files from the inbox on MG-RAST, Filenames in your Inbox are undergoing analysis and cannot be moved or submitted to a different process until analysis is complete.

---

differentialAnalyzeNet

*the KO difference abundance of specific state*

---

**Description**

The function computes the difference abundance with three methods and show the differential abundance comparison metabolic networks.

**Usage**

```
differentialAnalyzeNet (ssns, sample.state, method = c("OR", "rank", "JSD"), cutoff, p.value = TR
```

**Arguments**

ssns	the state specific network for differential comparison
sample.state	a character vector, represents the state of sample. Apparently, its length is equal to number of samples
method	method used to measure the difference abundance, more description in details
cutoff	numeric vector with length 2, specifies the difference threshold
p.value	logical, whether p value is caculated, Wilcoxon rank-sum test to limit enzymes that enriched or depleted
Visualization	logical, whether plot the differential metabolic network

**Details**

This function only applicable to compare the abundance profile between samples in different states (e.g., lean vs. obese). States is added to the abundance profile as the name attribute. Abundance were normalized within each sample to represent the relative abundance of each enzyme (KO) in each sample before comparison, thereby accounting for differences in sampling depth.

The selection of method to measure the enrichment is a non-trivial component of comparative metagenomic analysis, since statistical results can be heavily influenced by missing data, sample size, data magnitute, normalization technique, or assumptions of the distribution of values.

Odds ratio (OR) is a widely recognized measurement as well as has a number of benefits. The differential abundance score of each enzyme, defined as  $\text{abs}[\log_2(\text{OR})]$ , provides a measure of the extent to which an enzyme's abundance differs in samples from a given state, relative to samples in another state.

RANK: first ranking the enzymes within each sample from most abundant to least abundant, then measure the difference between the mean reank of each enzyme between samples in different states.

JSD is the Jensen-Shannon divergence measure to quantify the similarity bewteen the distributions of enzyme in samples in different state. Note it only suitable for a large enough sample size.

cutoff: more description see [showMetagenomicNet](#)

**Value**

a igraph with the diference abundance as a node attribute

---

estimateAbundance	<i>Getting KO (KEGG ontology) and abundance information</i>
-------------------	---

---

### Description

This function parses MGRAST functional annotation files to retrieve the KO of the KEGG, and estimate the abundance of each KO.

### Usage

```
estimateAbundance(KOAnno)
```

### Arguments

KOAnno	for multiple samples: a list, each element represents MGRAST functional annotation file suffixed with "expand.ontology" of one sample, includes md5 value of the database sequence hit followed by sequence or cluster ID, similarity information, and annotation information. Users can download it in the MGRAST website or with function <a href="#">getMgrastAnnotation</a> . for a single sample: a data frame, represents the MGRAST functional annotation file suffixed with "expand.ontology" of the sample
--------	--

### Details

With the annotation file, some values are encoded as index numbers in the annotation files of the argument. Column fields are as follows:

1. hit id (md5sum) index number
2. query id (either fasta ID or cluster ID)
3. percentage identity
4. alignment length
5. e-value
6. function annotation index number
7. functional category (ontology) index number
8. DB source index number

This function is just to get the KEGG ontology from the ontology annotation files. Users can substract reference networks by mapping this annotation to the selected reference database (KEGG PATHWAY).

### Value

Return a a given Biological Observation Matrix (BIOM) format, the ncol of the BIOM file is equal to the number of sample, and the colnames of the contingency table represents healthy state of samples. The BIOM file format (canonically pronounced biome) is designed to be a general-use format for representing biological sample by observation contingency tables. More details on BIOM file can be found in <http://biom-format.org/> R package *biom*. There is also a example BIOM file in directory "inst/extdata" of *mmnet*. , the name of the vector is the KO number and elements represents abundance.

**See Also**

[constructMetabolicNetwork](#)

**Examples**

```
#load the test ontology data
data(anno)
KOAbund <- estimateAbundance(anno[[1]])
```

---

generateMgrastWebkey *Generate your webkey on MGRAS*

---

**Description**

This function help users to generate a webkey which is essential for data uplodng to MGRAS

**Usage**

```
generateMgrastWebkey(login.info)
```

**Arguments**

login.info	list, reponse information while user login to MGRAS, contains webkey, invalid time of webkey, cookie, websseion and curlhandle, see <a href="#">loginMgrast</a> for more details
------------	--

**Value**

webkey	character, it will be valid for a limited time only (the next value)
invalid.until	character, with corresponding to the webkey, just the limit time of your webkey

**See Also**

[loginMgrast](#)

**Examples**

```
# not run
# generateMgrastWebkey(login.info$curlhandle)
```



---

```
generateMgrastWebsession
```

*Generate your websession*

---

### Description

Return your websession while login to MG-RAST

### Usage

```
generateMgrastWebsession(cookie)
```

### Arguments

cookie	web cookie, a small piece of data sent from a website and stored in a user's web browser while the user login to MG-RAST
--------	--

### Details

A session token is a unique identifier that is generated and sent from a server to a client to identify the current interaction session. The client usually stores and sends the token as an HTTP cookie. This function helps users to get websession from the cookie.

### Value

character

### See Also

[loginMgrast](#)

### Examples

```
##  
# generateWebsession(cookie)
```

---

```
getKOPathwayInfo
```

*Get the KEGG metabolic pathway information*

---

### Description

This function phases the KEGG metabolic pathway information, and obtains the KO or enzyme reaction, metabolites, for each pathway

### Usage

```
getKOPathwayInfo(koPathway)
```

### Arguments

koPathway	the metabolic KO pathway name, can be obtained with the <i>KEGGREST</i> package
-----------	---

**Value**

list of the KEGG metabolic pathway information

**See Also**

[updateKEGGPathway](#)

---

getMgrastAnnotation    *Get the annotation profile from MGRAST*

---

**Description**

The function downloads the annotation profile in MGRAST

**Usage**

```
getMgrastAnnotation(MetagenomeID, evalue = 5, identity = 60, length = 15, resource = c(source = "KO"))
```

**Arguments**

MetagenomeID	character, the metagenome ID in mgrast
evalue	expectation negative exponent value threshold for saving hits, default is 5
identity	minimum percent identity of matches, default is 60
length	value for minimum alignment length cutoff, default is 15
resource	a two length vector, the source you want to download, e.g., c("KO","ontology") represents the ontology annotation with the KO database , more details in MGRAST
login.info	a list of login info generated after user login into MGRAST which consists of webkey of the file, the invlid time of your webkey, cookie, websession and curl-handle. It is unique for each users

**Value**

annotation profile

**References**

<http://api.metagenomics.anl.gov/1/api.html#annotation>

**Examples**

```
##
# getMgrastAnnotation("mgm4447943.3")
```

---

listMgrastInbox	<i>List the contents of the user inbox</i>
-----------------	--

---

## Description

List the file and file information user uploaded to MGRAST.

## Usage

```
listMgrastInbox(login.info)
```

## Arguments

login.info list, reponse information while user login to MGRAST, contains webkey, invalid time of webkey, cookie, websseion and curlhandle, see [loginMgrast](#) for more details

## Details

There is a delay between upload completion and before job submission because sequence statistics is begin to calculate once file uploaded . This may be on the order of seconds to hours depending on file size. This files cannot be submitted to create job and annotation until analysis is complete. File name, file (calculation completed) statistics information and under analysis files which appear in the locks will be list with this function.

## Value

a list, contains file details user uploaded

## See Also

[submitMgrastJob](#)

## Examples

```
## login into MGRAST for webkey
# login.info <- loginMgrast("yiluheihei","heihei")

## not run need user login to MGRAT
#listMgrastInbox(login.info)
```

---

listMgrastProject	<i>List all the private projects you submitted to MGRAST</i>
-------------------	--

---

### Description

List all your private projects information on MGRAST, the information included job id, MGRAST project id and MGRAST id ("Accession numbers") which completed calculation.

### Usage

```
listMgrastProject(login.info)
```

### Arguments

login.info	a list of login info generated after user login into MGRAST which consists of webkey of the file, the invalid time of your webkey, cookie, websession and curl-handle. It is unique for each users
------------	--

### Details

Time MGRAST annotated a metagenome depend on the size of your dataset and and the current server load. In practice the time taken will range between a few hours and a week.

It will list all the MGRAST metagenome id for each user's project which was completely annotated, it also means the in progress metagenome of a project will not show until annotation complete. Only this metagenome list by this function can user to view the annotation relusts.

### Value

Return all the private projects, a list contains the job name users created while job submission, and MGRAST id ("Accession numbers of metagenome"), MGRAST id will be NULL while sample annotation is not completed, and project id MGRAST assigned corresponding to the job id. metagenome.id), "?verbosity=metadata")

### Examples

```
# listMgrastProject(login.info)
```

---

loadMetabolicData	<i>Metabolic data loading</i>
-------------------	-------------------------------

---

### Description

This function load the updated reference metabolic data into workspace.

### Usage

```
loadMetabolicData(path = Sys.getenv("HOME"))
```

**Arguments**

path character, the path that reference data saved, default is the user's HOME directory. If does not exists, prestored reference metabolic data (RefDbCache) in *mmnet* will be loaded.

**See Also**

[link{updateKEGGPathway}](#), [constructMetabolicNetwork](#)

**Examples**

```
##
# loadMetabolicData(path = Sys.getenv("HOME"))
```

---

loginMgrast

*Login to the MGRAST*

---

**Description**

Login to MGRAST, obtain your Websession and webkey for data uploading, as an authenticate of subsequent operation about MGRAST (e.g. data upload, job submission).

**Usage**

```
loginMgrast(user, userpwd)
```

**Arguments**

user character, username you have registered on MGRAST  
 userpwd character, password correspondance to your username

**Details**

Sequence functional annotation is the most necessary and important step for metagenomic analysis. MGRAST (metagenomics RAST) server is a free and public utomated analysis platform for metagenomes providing quantitative insights into microbial populations based on sequence data, also very fast, stable and extensible. Thus, MGRAST was took for sequence functional annotation in our package *mmnet*.

Webkey, session and curlhandle would be generated when users login to MGRAST with this function, which is the key for data uploading and sequence annotation.

**Value**

webkey character, e.g. "8Dvg2d5DCp7KsWKBPzY2GS4i", the webkey of your account which is essential for data uploding to MGRAST. However, it will be valid for a limited time only (the next value). Users will generate a new webkey with fucntion [generateMgrastWebkey](#) for convenient.

invalid.until character, with corresponding to the webkey, just the limit time of your webkey

cookie data saved on user's computer while login to MGRAST

session character, a userid that MGRAST server assigned

curlhandle class of CURLHandle, the curl handle while users to MGRAST which is saved for user's sebsequent analysis with MGRAST.

**See Also**

[uploadMgrast](#)

**Examples**

```
# need a username and password on MGRAST
# login.info <- loginMgrast("mmnet","mmnet")
```

---

preprocessKOMetabolites

*Preprocessing of KO's (enzyme's) metabolites*

---

**Description**

This function preprocess the KO information to delete the redundant metabolites

**Usage**

```
preprocessKOMetabolites(ko.info)
```

**Arguments**

ko.info            KO information, can be obtained by [getKOPathwayInfo](#)

**Details**

When we get the KO information first, there are reversible reaction, duplicate KOs and metabolites. This function merge all the KO information together and delete the redundant information.

**Value**

list, is composed of KO, substrates and products

**See Also**

[getKOPathwayInfo](#), [updateKEGGPathway](#)

---

RefDbcache

*Reference data for global metabolic construction*

---

**Description**

The metabolic pathway data contains KOs, substrates and products, as well as a constructed reference global network

**Usage**

```
data(RefDbcache)
```

**Format**

The format is: List of 7 KO, substrate, product, user, date, version, reference network :

**Examples**

```
data(RefDbcache)
```

---

saveMetabolicData	<i>Save a new version of the reference metabolic data</i>
-------------------	---

---

**Description**

This function saves the updated reference KEGG metabolic data and global network in your computer.

**Usage**

```
saveMetabolicData(RefDbcache, path = Sys.getenv("HOME"))
```

**Arguments**

RefDbcache	a list, the updated KEGG Pathway information, more details in <code>link{preprocessKOMetabolites}</code>
path	character, the path that reference data saved, default is the user's HOME directory.

**See Also**

`link{preprocessKOMetabolites}`

**Examples**

```
##
# saveMetabolicData(RefDbcache, path = Sys.getenv("HOME"))
```

---

showMetagenomicNet	<i>Visualiation of metabolic network</i>
--------------------	--

---

**Description**

This function able to plot metabolic network to any R device.

**Usage**

```
showMetagenomicNet(net, mode = c("ref", "ssn", "compared"), method = c("OR", "rank", "JSD"), cutoff
```

**Arguments**

net	the graph to visualization, reference metabolic network or sub-network of different samples
mode	a string, represents the property of your network, for more description in details
method	method used to measure the difference abundance, more description in <a href="#">differentialAnalyzeNet</a>
cutoff	numeric vector with length 2, specifies the difference threshold, more description in details
...	additional plotting parameters. See <a href="#">igraph.plotting</a> for the complete list.

**Details**

There are three different metabolic networks: ref - reference global network, ssn - state specific, compared - compared network between different state network.

More details on method to measure the difference abundance see [differentialAnalyzeNet](#)

cutoff: two-fold threshold commonly used in the odds-ratio test, cutoff = c(0.5, 2); enzymes with the 10% highest calculated differences and the 10% lowest differences, cutoff = c(0.1, 0.9)

**Value**

Returns NULL, invisibly.

**Examples**

```
# reference network
data(RefDbcache)
showMetagenomicNet(RefDbcache$network, mode="ref")
```

---

submitMgrastJob	<i>Create a job on MGRAST</i>
-----------------	-------------------------------

---

**Description**

Submit your file to create a job after data uploaded.

**Usage**

```
submitMgrastJob(login.info, seqfile, new_project)
```

**Arguments**

login.info	a list of login info generated after user login into MGRAST which consists of webkey of the file, the invalid time of your webkey, cookie, websession and curl-handle. It is unique for each users
seqfile	character, the sequence name you want to analysis
new_project	character, the project name



**Details**

Users have to specify a project to upload a job to MG-RAST, just new project supported in this current version of our package. When the submission process has been successfully completed, MG-RAST ID's ("Accession numbers") will be automatically assigned and the data will be removed from your inbox.

**Value**

return the MGRAST ID of your sequence and the user name

**See Also**

[listMgrastInbox](#), [uploadMgrast](#)

**Examples**

```
## login into MGRAST for webkey
# login.info <- loginMgrast("yiluheihei", "heihei")

##
#submitMgrastJob(login.info, seq, new_project)
```

---

topologicalAnalyzeNet *Analyze the specific state network*

---

**Description**

Calculate Topological features of each node and linked with abundance attributes.

**Usage**

```
topologicalAnalyzeNet(g, Scatterplot = TRUE, .properties = c("betweennessCentrality", "degree", "C"))
```

**Arguments**

<code>g</code>	a igraph network of a specific state, it can be obtained with function <a href="#">constructSSN</a>
<code>Scatterplot</code>	logical, if show the scatterplot between topological features and abundance
<code>.properties</code>	instead of specifying the topological features in free form via the ... argument, one can specify them as a vector
<code>mode</code>	Character string, "out" for out-degree, "in" for in-degree and "all" for the sum of the two.
<code>...</code>	topological features that user want to calculate.

**Details**

Five most presented vertex topological features is provided by this function.

**betweenness centrality:** defined by the number of geodesics (shortest paths) going through a vertex, more details in [betweenness](#) of package igraph

**degree:** the most basic structural property, represents the number of its adjacent edges.

**clusteringCoefficient:** measures the probability that the adjacent vertices of a vertex are connected

**pageRank:** Calculates the Google PageRank for the vertices in network

**Value**

a graph that add all the calculated topological features as the meta-attributes

**See Also**

[constructSSN](#)

**Examples**

```
##
data(anno)
data(RefDbcache)
KOAbund <- estimateAbundance(anno[[1]])

subnet <- constructSSN(KOAbund)
topologicalAnalyzeNet(subnet)
```

---

updateKEGGPathway	<i>Updating and saving the reference (KEGG metabolic pathway) data</i>
-------------------	--

---

**Description**

Parsing the KEGG Markup Language (KGML) file of KEGG metabolic pathway, obtaining the KO and corresponding metabolites and constructing the reference network.

**Usage**

```
updateKEGGPathway(path = Sys.getenv("HOME"))
```

**Arguments**

path	A character, the path that reference data saved, default is the user's HOME directory.
------	--

**Details**

The KEGG metabolic pathway is the best organized part of the KEGG/PATHWAY database, each metabolic pathway can be viewed as a network of KOs. The KEGG Markup Language (KGML) is an exchange format of the KEGG pathway maps, which is converted from internally used KGML+ (KGML+SVG) format. It will retrieve all KOs, products and substrates from the KGML files of KEGG metabolic pathway. There is an initial reference data saved in the subdirectory data of this package. However, KEGG metabolic pathway updated frequently, this function will update the data which contains KO, substrates, products and reference metabolic network. . For more details, see [constructMetabolicNetwork](#).

This function updates the KEGG reference data, and saving in the dir.*mmnet* under users Specified, default is the user's home directory. Date, user name and the R VERSION also saved in a log file.

**Note**

For this function, download KGML file and reference metabolic constructing may take a long time

**References**

<http://www.kegg.jp/kegg/xml/docs/>

**See Also**

[constructMetabolicNetwork,RefDbcache](#)

**Examples**

```
#updateMetabolicNetwork()
```

---

uploadMgrast

*Upload your sequence to MGRAST*

---

**Description**

Sequence can be uploaded to MGRAST for functional annotation.

**Usage**

```
uploadMgrast(login.info, file)
```

**Arguments**

login.info	a list of login info generated after user login into MGRAST which consists of webkey of the file, the invlid time of your webkey, cookie, websession and curl-handle. It is unique for each users.
file	the sequence file user want to upload, more description in details.

**Details**

MG-RAST supports each user a temporary storage location. This inbox provides temporary storage for data to be submitted for analysis. Users can also delete the files in your inbox.

Ensure the file not exsited in the inbox before upload. Uploaded files may be removed from your inbox after 72 hours. Please perform annotation of your files within that time frame. There is a delay between upload completion and appearing in this table due to sequence statistics calculations. This may be on the order of seconds to hours depending on file size. Uses can check your file state with the function [listMgrastInbox](#).

Your uploaded sequence data can be in FASTA, FASTQ or SFF format. These are recognized by the file name extension with valid extensions for the appropriate formats .fasta, .fna, .fastq, .fq, and .sff and FASTA and FASTQ files need to be in plain text ASCII. Compressing large files will reduce the upload time and the chances of a failed upload, you can use Zip (.zip) and gzip (.gz) as well as tarred gzipped files (.tgz) but not rar. We suggest you upload raw data (in FASTQ or SFF format) and let MG-RAST perform the quality control step, see here for details.

**See Also**

[listMgrastInbox,submitMgrastJob](#)

**Examples**

```
## login into MGRASST for webkey
# login.info <- loginMgrast("yiluheihei","heihei")

## upload
# uploadMgrast(login.info, file="")
```

# Index

anno, [2](#)

betweenness, [17](#)

checkMgrastMetagenome, [3](#)

constructMetabolicNetwork, [2](#), [4](#), [8](#), [13](#), [18](#),  
[19](#)

constructSSN, [4](#), [17](#), [18](#)

delMgrastInbox, [5](#)

differentialAnalyzeNet, [6](#), [16](#)

differentialAnalyzeNet  
(differentialAnalyzeNet), [6](#)

estimateAbundance, [2](#), [5](#), [7](#)

generateMgrastWebkey, [8](#), [13](#)

generateMgrastWebsession, [9](#)

getKOPathwayInfo, [9](#), [14](#)

getMgrastAnnotation, [7](#), [10](#)

igraph.plotting, [16](#)

listMgrastInbox, [11](#), [17](#), [19](#)

listMgrastProject, [12](#)

loadMetabolicData, [4](#), [12](#)

loginMgrast, [5](#), [8](#), [9](#), [11](#), [13](#)

mmnet-package, [2](#)

preprocessKOMetabolites, [14](#)

RefDbcache, [5](#), [14](#), [19](#)

saveMetabolicData, [15](#)

showMetagenomicNet, [2](#), [6](#), [15](#)

submitMgrastJob, [11](#), [16](#), [19](#)

topologicalAnalyzeNet, [17](#)

updateKEGGPathway, [2](#), [10](#), [14](#), [18](#)

uploadMgrast, [14](#), [17](#), [19](#)