User Manual for MeganServer1.0

Hans-Joachim Ruscheweyh
hans-joachim.ruscheweyh@id.ethz.ch

June 21, 2016

Contents

1   Introduction

MeganServer is a companion tool to MEGAN6 [2]. It allows researchers to outsource the storage of often very large metagenomic datasets to a dedicated server. The datasets remain available via a secured network connection and can be accessed in MEGAN6 as if they would be local files. For the purpose of communication with MEGAN6 MeganServer provides a RESTful API based on JSON.

For testing we operate an instance of MeganServer at http://megan-db.org/Public/ (username/password=guest) at the Center for Bioinformatics in Tübingen, Germany. This instance provides access to MEGAN files containing the complete taxonomic and functional analysis of 130 published metagenome samples. Under "Asari" we list 12 human gut samples from a recent study on antibiotic selection pressure [6], labeled "Alice" and "Bob". This dataset consists of 612 million aligned reads (out of 830 million reads) and nearly 10 billion alignments, requiring 229 GB to host. Under "Daisy", we list four human gut samples published in [5]. These datasets contain a total of 6.6 million aligned reads, requiring 6 GB to host. Under "Permafrost"', we list 12 permafrost soil samples published in [4]. These datasets contains 52 million aligned reads (out of 250 million reads) and 750 million alignments, requiring 38 GB to host. Under "LouisEtAl2016", we list 92 human gut microbiota samples published in [3]. These datasets contain 1.4 billion reads and 22 billion alignments, and require 600 GB to host.

2   Help with MeganServer

Please do not hesitate to contact hans-joachim.ruscheweyh@id.ethz.ch or post a question on the MEGAN wiki page (http://megan.informatik.uni-tuebingen.de/) if there is any problem with installation or usage of MeganServer.

3   Installation

MeganServer requires to be launched inside of a Java servlet engine. For that purpose we distribute three variants of MeganServer. The **Standalone** variant includes the Java servlet engine and can be launched in terminal or via a graphical user interface. The **Web Archive** variant contains the raw war file and requires to be copied to a servlet engine. Finally, the **Source** distribution allows one to additionally build the application from the source provided at github:

- **Standalone**: The standalone version of MeganServer targets users who would like to test the application locally. Besides the MeganServer program this bundle also ships with an Java servlet engine (Jetty) and scripts to start and stop MeganServer out of the box.

- **Web Archive**: With this version we target users who would like to integrate

MeganServer in an existing web server/servlet engine environment such as Tomcat or Jetty.

- **Source**: The source code of MeganServer is publicly released under the GPL3 license on GitHub https://github.com/danielhuson/megan6server

3.1 Standalone

Prerequisites: MeganServer requires an installed JAVA runtime, version 8.

Download MeganServer_standalone.zip from http://ab.inf.uni-tuebingen.de/software/MeganServer. The archive includes:

- **bin** Folder with scripts to start and stop MeganServer
  - **start.sh**: Script to start MeganServer on Linux and OS X
  - **start.bat**: Script to start MeganServer on Windows
  - **stop.sh**: Script to stop MeganServer on Linux and OS X
  - **stop.bat**: Script to stop MeganServer on Windows
- **lib** Folder with program libraries
  - **MeganServer.war**: The MeganServer application
  - **ServerStarter.jar**: The graphical user interface to start MeganServer
  - **jetty-runner-9.2.7.v20150116.jar**: The web server which hosts the MeganServer application
  - **commons-codec-1.6.jar**: An additional library
  - **start.jar**: An additional library
  - **TableLayout.jar**: An additional library
- **properties** Folder with configuration files. See Section 4
  - **credentials.txt**
  - **log4j.properties**
  - **MeganServer.properties**
  - **info.txt**
- **log** Folder where the application's log files are stored
- **manual.pdf**: This manual

The application can be started with the start.(sh|bat) command (for full list of command line options see below). The server will then be accessible via http://localhost:8080/MeganServer/. To stop the server execute stop.(sh|bat). MeganServer needs to know where on the file system the RMA files can be found. Please read Section 4.1 for configuration options.

3.1.1 start.(sh|bat)

--name: Define name of MeganServer instance. Default=Meganserver

--port: Port on which MeganServer will be launched. Default=8080

--stop-port: Port to stop MeganServer. Default=8181

--stop-key: Stop key. Default=7ce31e4eb617c85efbff9450bf9c2fd2

--max-memory: Max memory for MeganServer. Default=4G

--help: Shows this help

Example: `'./start.sh --name MyMeganServer --port 1234'` will launch MeganServer on http://www.localhost:1234/MyMeganServer

3.1.2 stop.(sh|bat)

--stop-port: Port to stop MeganServer. Default=8181

--stop-key: Stop key. Default=7ce31e4eb617c85efbff9450bf9c2fd2

--help: Shows this help

Example: `'./stop.sh --stop-port 1235 --stop-key RandomStopKey'` will stop MeganServer which has been started with stop port 1235 and stop key RandomStopKey

3.2 Web Archive

Prerequisites: MeganServer requires an installed JAVA runtime (at least version 8). Furthermore a servlet container such as Tomcat or Jetty must be installed.

Download MeganServer_webarchive.zip from http://ab.inf.uni-tuebingen.de/data/software/MeganServer/download/MeganServer_webarchive.zip. The archive includes:

- MeganServer.war: The web archive containing the application and configuration files

- **manual.pdf**: This manual

The application has to be launched inside a servlet engine such as Tomcat (see http://tomcat.apache.org/) or Jetty (see: http://www.eclipse.org/jetty/). Configuration as described in Section 4 takes place after the first launch. Optionally unzip the war file and edit the config files in the *WEB-INF/classes/* folder before the first launch.

4 Configuration

4.1 Metagenomic Data Storage

MEGAN6 operates on so called Read Match Archives (short RMA), a binary file format that incorporates reads, matches, metadata and classification data and is currently available in version 2, 3 and 6. This file structure is also being used by MeganServer in order to publish data via the web service. Furthermore, MeganServer can also operate on so-called *meganized* DAA files. DAA files are emitted by the aligner DIAMOND [1]. MEGAN6 is distributed with the tool to then meganize these datasets http://ab.inf.uni-tuebingen.de/data/software/megan6/download/welcome.html.

All data files (RMA v2, v3, v6 and meganized DAA files) must be located in a base folder that is accessible by MeganServer. MeganServer will then recursively screen this folder searching for suitable data files. The base folder can be changed by modifying the entry `rma.rootFolder` in the *meganserver.properties* file. MeganServer will then find RMA files in this folder and all subfolders and expose those via the web service. This property can be set before starting the server or while running. For the latter option one must then reload the properties file by calling the method described in 6.4.1.

4.2 Log Files

Applicable only when using the webarchive version

By default MeganServer creates a log file in the folder where the web application is launched. The location of this file can be changed by altering the property `log4j.appender.file.File` in the *log4.properties* file. This operation requires a restart of the web application (or the servlet engine)

4.3 User credentials

The *credentials.txt* file contains information about all users that are allowed to access a MeganServer instance. Its entries can be changed either by editing the file (md5 hashed) or via web service using the commands as described in Section 6.4.3.

The default user with administrative permissions has the username/password *admin*. It is advisable to change the password of this user before the first start of the server.

Adding a new user *guest* with the password *guest* and user permissions one needs to add this line to the credentils file:

`guest 084e0343a0486ff05530df6c705c8bb4 true false`

4.4   Information Text

The *info.txt* gives administrators the possibility to share information, such as what data is stored on the server and/or who to send an email for account requests. The *info.txt* can be found together with the other configuration files and can contain either plaintext or html.

5   MEGAN

The metagenomics analysis tool MEGAN6 comes with a graphical user interface to connect and exchange data with MeganServer. With this connection one can work on remote datasets just as if they would be present on a local hard drive. The connection can be established via the graphical user interface of MEGAN as shown in Figure 1
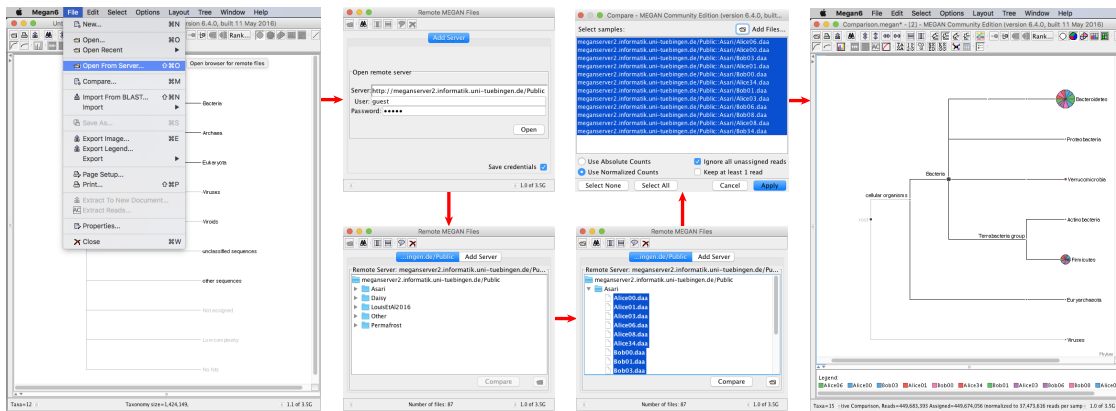

Figure 1: MEGAN6 has inbuilt capabilities to connect and communicate with MeganServer. MeganServer is implemented in that way that users can take advantage of the full set of functions that MEGAN6 offers without the need to have large data files physically present. MEGAN6 provides a graphical user interface to connect to one or multiple MeganServer instances at a time. Once connected users can browse through remotely hosted datasets that are organized in a tree-like structure. One or multiple datasets can then be opened in MEGAN6 and their content can be explored in the same way as if data files would be physically present.

Please read the MeganServer section in the MEGAN manual for further information http://ab.inf.uni-tuebingen.de/data/software/megan6/download/welcome.html.

6   MeganServer RESTful API

MeganServer uses a JSON based RESTful API to broadcast its services as well as transfer data. MEGAN6 is fully capable to communicate with MeganServer via this API. However, the API can be used even without MEGAN6 using, for example, a simple web browser. This section describes all API commands in detail.

6.1 Parameter

The API uses queries to retrieve data. The selection of data is performed by methods and parameters. Since some parameters are recurring in multiple methods we will introduce these parameters in this section and in the next section the methods.

6.1.1 Identifier

MeganServer accesses datasets and data inside datasets with unique identifiers. MeganServer knows three types of identifier.

Dataset/File Identifier The dataset/file identifier uniquely identifies a single dataset on a MeganServer instance. The parameter on the API is `fileId` and is either a number such as **1242324557** or the path to the dataset on the server such as **Permafrost/-core2_activelayer_day2.rma6**

Read Identifier Each read inside a dataset has a numerical identifier. This identifier is unique inside a dataset but the identifier is not required to be unique among datasets. Two different datasets can have same read identifier. On the API we refer to the read identifier as `readUid`.

Match Identifier Each match inside a read has a numerical identifier. This identifier is unique inside a dataset but the identifier is not required to be unique among datasets. Two different datasets can have the same read identifier. On the API we refer to the match identifier as `matchUid`.

6.1.2 General Parameters

DataSelection When requesting reads or matches one can select which data fields should be returned. Currently following values are supported:

```
useRead, useReadName, useReadHeader, useReadSequence, useMateUId,
useReadLength, useReadComplexity, useReadNumberOfMatches, useMatchText,
useMatchIgnore, useMatchBitScore, useMatchLength, useMatchTaxonId,
useMatchSeedId, useMatchKeggId, useMatchCogId, useMatchExpected, useMatchRefSeq
```

Therefore the parameter `dataSelection="useReadName,useReadLength"` would tell MeganServer to return only the read name and the read length for a given query. If this parameter is skipped MeganServer will use the default value which returns all possible data fields.

FindSelection and regEx MeganServer supports searching for data matching regular expressions inside datasets. One has to specify where to search (FindSelection) and what to search for (regEx).

The FindSelection currently supports four data fields that can be searched:

`useMatchText, useReadHeader, useReadName, useReadSequence`

The regular expression can be any string or normal regular expression (See MEGAN manual for further information).

A possible combination of these both parameters which returns data on all sequences that look like "ACTG" would then be `FindSelection="userReadSequence"&regEx="ACTG"`

minScore and maxExpected These both parameters are meant to give users the possibility to filter matches by quality. The minScore parameter defines the minimal bitscore an alignment must have in order to appear in the results. The maxExpected value allows users to filter by E-Value. A possible parameter combination would be `minScore="100"&maxExpected="0.0001"`.

The default values are for minScore 0 and for maxExpected 100000.

classification and classId(s) These both parameters are meant to give users the possibility to filter matches by classification and classification ids. MEGAN currently supports four classifications, namely NCBI taxonomy, SEED, COG and KEGG. Each of these classifications has identifier for e.g genes or taxa. Therefore the parameter `classification="Taxonomy"&classId="562"` will tell the executing method to only return reads that stem from E.coli.

6.2 General

6.2.1 help

- **Name**: help

- **Outline**: Get a list of all accessible commands of MeganServer

- **Parameters**

  – No Parameters

- **Examples**:

  – http://megan-db.org/Public/

  – http://megan-db.org/Public/help

6.2.2 info

- **Name**: info
- **Outline**: Get information about the data stored on this MeganServer instance
- **Parameters**
  - No Parameters
- **Examples**:
  - http://megan-db.org/Public/info

6.2.3 about

- **Name**: about
- **Outline**: Get a version number, contact information and build date of MeganServer
- **Parameters**
  - No Parameters
- **Examples**:
  - http://megan-db.org/Public/About

6.3 Reading RMA files

6.3.1 listDatasets

- **Name**: listDatasets
- **Outline**: Retrieve a list of datasets that is known to the MeganServer instance.
- **Parameters**
  - **includeMetadata**
    * **Name**: includeMetadata
    * **Outline**: include metadata to the result
    * **Required** : false
    * **Default Value** : false
- **Examples**:
  - http://megan-db.org/Public/listDatasets
  - http://megan-db.org/Public/listDatasets?includeMetadata=True

6.3.2 getFileUid

- **Name**: getFileUid

- **Outline**: Get the unique identifier for a dataset

- **Parameters**

  - **fileId**

    * **Name**: fileId (see 6.1.1)
    * **Outline**: An identifier of a dataset. Either a path, a name or a number
    * **Required** : true
    * **Default Value** : –

- **Examples**:

  - http://megan-db.org/Public/getFileUid?fileId=Permafrost/core2_
    activelayer_day2.rma6
  - http://megan-db.org/Public/getFileUid?fileId=1242324557

6.3.3 getAuxiliary

- **Name**: getAuxiliary

- **Outline**: Get metadata and taxonomic/functional overview on a single dataset

- **Parameters**

  - **fileId**

    * **Name**: fileId (see 6.1.1)
    * **Outline**: An identifier of a dataset. Either a path, a name or a number
    * **Required** : true
    * **Default Value** : –

- **Examples**:

  - http://megan-db.org/Public/getAuxiliary?fileId=Permafrost/core2_
    activelayer_day2.rma6

6.3.4 getRead

- **Name**: getRead

- **Outline**: Get data on a single sequence together with all its information

- **Parameters**

- **fileId**
  * **Name**: fileId (see 6.1.1)
  * **Outline**: An identifier of a dataset. Either a path, a name or a number
  * **Required** : true
  * **Default Value** : –

- **readUid**
  * **Name**: readUid (see 6.1.1)
  * **Outline**: The numerical identifier of the read
  * **Required** : true
  * **Default Value** : 0

- **minScore**
  * **Name**: minScore (see 6.1.2)
  * **Outline**: The minimal bitscore a match has to have in order to be present in the result
  * **Required** : false
  * **Default Value** : 0

- **maxExpected**
  * **Name**: maxExpected (see 6.1.2)
  * **Outline**: The maximal E-Value a match is allowed to have in order to be present in the result
  * **Required** : false
  * **Default Value** : 100000

- **dataSelection**
  * **Name**: dataSelection (see 6.1.2)
  * **Outline**: Definition of fields that should be present in the result
  * **Required** : false
  * **Default Value** : useRead,useReadName,useReadHeader,useReadSequence, useMateUId,useReadLength,useReadComplexity,useReadNumberOfMatches, useMatchText,useMatchIgnore,useMatchBitScore,useMatchLength, useMatchTaxonId,useMatchSeedId,useMatchKeggId,useMatchCogId,useMatchExpected, useMatchRefSeq

- **Examples**:

  - http://megan-db.org/Public/getRead?fileId=1958397319&readUid=7679
  - http://megan-db.org/Public/getRead?fileId=1958397319&readUid=7679& minScore=10
  - http://megan-db.org/Public/getRead?fileId=1958397319&readUid=7679& minScore=10&maxExpected=0.01&dataSelection="useReadName,useReadLength"

6.3.5 getAllReadsIterator

- **Name**: getAllReadIterator

- **Outline**: Get data of all reads of a single dataset. Returns one page of results and a link to the next page of results.

- **Parameters**

    - **fileId**
        * **Name**: fileId (see 6.1.1)
        * **Outline**: An identifier of a dataset. Either a path, a name or a number
        * **Required** : true
        * **Default Value** : –

    - **minScore**
        * **Name**: minScore (see 6.1.2)
        * **Outline**: The minimal bitscore a match has to have in order to be present in the result
        * **Required** : false
        * **Default Value** : 0

    - **maxExpected**
        * **Name**: maxExpected (see 6.1.2)
        * **Outline**: The maximal E-Value a match is allowed to have in order to be present in the result
        * **Required** : false
        * **Default Value** : 100000

    - **dataSelection**
        * **Name**: dataSelection (see 6.1.2)
        * **Outline**: Definition of fields that should be present in the result
        * **Required** : false
        * **Default Value** : useRead,useReadName,useReadHeader,useReadSequence, useMateUId,useReadLength,useReadComplexity,useReadNumberOfMatches, useMatchText,useMatchIgnore,useMatchBitScore,useMatchLength, useMatchTaxonId,useMatchSeedId,useMatchKeggId,useMatchCogId,useMatchExpected, useMatchRefSeq

- **Examples**:

    - http://megan-db.org/Public/getAllReadsIterator?fileId=1958397319

6.3.6 getReadsIterator

- **Name**: getReadIterator

- **Outline**: Get data of all reads of a single dataset that are assigned to a node in the taxonomy or functional classification. Returns one page of results and a link to the next page of results.

- **Parameters**

    – **fileId**
      * **Name**: fileId (see 6.1.1)
      * **Outline**: An identifier of a dataset. Either a path, a name or a number
      * **Required** : true
      * **Default Value** : –

    – **classId**
      * **Name**: classId (see 6.1.2)
      * **Outline**: The id of a classification such as NCBI taxonomy, KEGG or SEED
      * **Required** : true
      * **Default Value** : –

    – **classification**
      * **Name**: classification (see 6.1.2)
      * **Outline**: One of the classifications MEGAN supports
      * **Required** : true
      * **Default Value** : –

    – **minScore**
      * **Name**: minScore (see 6.1.2)
      * **Outline**: The minimal bitscore a match has to have in order to be present in the result
      * **Required** : false
      * **Default Value** : 0

    – **maxExpected**
      * **Name**: maxExpected (see 6.1.2)
      * **Outline**: The maximal E-Value a match is allowed to have in order to be present in the result
      * **Required** : false
      * **Default Value** : 100000

    – **dataSelection**
      * **Name**: dataSelection (see 6.1.2)
      * **Outline**: Definition of fields that should be present in the result

∗ **Required** : false

∗ **Default Value** : useRead,useReadName,useReadHeader,useReadSequence, useMateUId,useReadLength,useReadComplexity,useReadNumberOfMatches, useMatchText,useMatchIgnore,useMatchBitScore,useMatchLength, useMatch-TaxonId,useMatchSeedId,useMatchKeggId,useMatchCogId,useMatchExpected, useMatchRefSeq

- **Examples**:

  – http://megan-db.org/Public/getReadsIterator?fileId=1958397319&classification=Taxonomy&classId=1

6.3.7 getReadsForMultipleClassIds

- **Name**: getReadsForMultipleClassIds

- **Outline**: Get data of all reads of a single dataset that are assigned to a number of nodes in the taxonomy or functional classification. Returns one page of results and a link to the next page of results.

- **Parameters**

  – **fileId**

    ∗ **Name**: fileId (see 6.1.1)
    ∗ **Outline**: An identifier of a dataset. Either a path, a name or a number
    ∗ **Required** : true
    ∗ **Default Value** : –

  – **classId**

    ∗ **Name**: classId (see 6.1.2)
    ∗ **Outline**: The ids of a classification such as NCBI taxonomy, KEGG or SEED
    ∗ **Required** : true
    ∗ **Default Value** : –

  – **classification**

    ∗ **Name**: classification (see 6.1.2)
    ∗ **Outline**: One of the classifications MEGAN supports
    ∗ **Required** : true
    ∗ **Default Value** : –

  – **minScore**

    ∗ **Name**: minScore (see 6.1.2)
    ∗ **Outline**: The minimal bitscore a match has to have in order to be present in the result
    ∗ **Required** : false

* **Default Value** : 0

  – **maxExpected**

    ∗ **Name**: maxExpected (see 6.1.2)
    ∗ **Outline**: The maximal E-Value a match is allowed to have in order to be present in the result
    ∗ **Required** : false
    ∗ **Default Value** : 100000

  – **dataSelection**

    ∗ **Name**: dataSelection (see 6.1.2)
    ∗ **Outline**: Definition of fields that should be present in the result
    ∗ **Required** : false
    ∗ **Default Value** :   useRead,useReadName,useReadHeader,useReadSequence, useMateUId,useReadLength,useReadComplexity,useReadNumberOfMatches, useMatchText,useMatchIgnore,useMatchBitScore,useMatchLength, useMatch-TaxonId,useMatchSeedId,useMatchKeggId,useMatchCogId,useMatchExpected, useMatchRefSeq

* **Examples**:

  – http://megan-db.org/Public/getReadsForMultipleClassIds?fileId= 1958397319&classification=Taxonomy&classIds=1,2

6.3.8 getFindAllReadsIterator

* **Name**: getFindAllReadIterator

* **Outline**: Get data of all reads of a single dataset and searches for text patterns. Those fulfilling the pattern are returned. Returns one page of results and a link to the next page of results.

* **Parameters**

  – **fileId**

    ∗ **Name**: fileId (see 6.1.1)
    ∗ **Outline**: An identifier of a dataset. Either a path, a name or a number
    ∗ **Required** : true
    ∗ **Default Value** : –

  – **regEx**

    ∗ **Name**: regEx (see 6.1.2)
    ∗ **Outline**: Regular expression that is being searched for
    ∗ **Required** : true
    ∗ **Default Value** : –

- **findSelection**
  - ∗ **Name**: findSelection (see 6.1.2)
  - ∗ **Outline**: Where to search
  - ∗ **Required** : false
  - ∗ **Default Value** : useMatchText,useReadHeader,useReadName,useReadSequence

- **Examples**:

  - http://megan-db.org/Public/getFindAllReadsIterator?fileId=1958397319&regEx=Bacteria&findSelection=useMatchText

6.3.9 loadPagedReads

- **Name**: loadPagedReads

- **Outline**: Load the next page of an readblockiterator

- **Parameters**

  - **pageId**
    - ∗ **Name**: pageId
    - ∗ **Outline**: Next page Id. Retrieved from the previous page
    - ∗ **Required** : true
    - ∗ **Default Value** : –

- **Examples**: Pages and pageIds are dynamic. This is why the link in the example will not work. Try to insert your own pageId.

  - http://megan-db.org/Public/loadPagedReads?pageId=823880593

6.3.10 getNumberOfReads

- **Name**: getNumberOfReads

- **Outline**: Get number of reads for one dataset

- **Parameters**

  - **fileId**
    - ∗ **Name**: fileId (see 6.1.1)
    - ∗ **Outline**: An identifier of a dataset. Either a path, a name or a number
    - ∗ **Required** : true
    - ∗ **Default Value** : –

- **Examples**:

  - http://megan-db.org/Public/getNumberOfReads?fileId=1958397319

6.3.11 getNumberOfMatches

- **Name**: getNumberOfMatches

- **Outline**: Get number of alignments for one dataset

- **Parameters**

    - **fileId**

        * **Name**: fileId (see 6.1.1)
        * **Outline**: An identifier of a dataset. Either a path, a name or a number
        * **Required** : true
        * **Default Value** : –

- **Examples**:

    - http://megan-db.org/Public/getNumberOfMatches?fileId=1958397319

6.3.12 getAllClassificationNames

- **Name**: getAllClassifcationNames

- **Outline**: Get the names of all classifications present in this dataset

- **Parameters**

    - **fileId**

        * **Name**: fileId (see 6.1.1)
        * **Outline**: An identifier of a dataset. Either a path, a name or a number
        * **Required** : true
        * **Default Value** : –

- **Examples**:

    - http://megan-db.org/Public/getAllClassificationNames?fileId=1958397319

6.3.13 getClassSize

- **Name**: getClassSize

- **Outline**: Get the number of reads assigned to one node in a classification

- **Parameters**

    - **fileId**

        * **Name**: fileId (see 6.1.1)

- * **Outline**: An identifier of a dataset. Either a path, a name or a number
- * **Required** : true
- * **Default Value** : –

- **classId**
  - * **Name**: classId (see 6.1.2)
  - * **Outline**: The id of a classification such as NCBI taxonomy, KEGG or SEED
  - * **Required** : true
  - * **Default Value** : –

- **classification**
  - * **Name**: classification (see 6.1.2)
  - * **Outline**: One of the classifications MEGAN supports
  - * **Required** : true
  - * **Default Value** : –

- **Examples**:

  - http://megan-db.org/Public/getClassSize?fileId=1958397319&classification=Taxonomy&classId=1

6.3.14   getClassificationBlock

- **Name**: getClassificationBlock

- **Outline**: Get the number of reads assigned to all nodes in a classification

- **Parameters**

  - **fileId**
    - * **Name**: fileId (see 6.1.1)
    - * **Outline**: An identifier of a dataset. Either a path, a name or a number
    - * **Required** : true
    - * **Default Value** : –

  - **classification**
    - * **Name**: classification (see 6.1.2)
    - * **Outline**: One of the classifications MEGAN supports
    - * **Required** : true
    - * **Default Value** : –

- **Examples**:

  - http://megan-db.org/Public/getClassificationBlock?fileId=1958397319&classification=Taxonomy

6.3.15 isReadOnly

- **Name**: isReadOnly

- **Outline**: Determines if user has write permissions. Currently all datasets are readOnly, no matter if a user is admin or not.

- **Parameters**

    – **fileId**

      * **Name**: fileId (see 6.1.1)
      * **Outline**: An identifier of a dataset. Either a path, a name or a number
      * **Required** : true
      * **Default Value** : –

- **Examples**:

    – http://megan-db.org/Public/isReadOnly?fileId=1958397319

6.3.16 getClassificationSize

- **Name**: getClassificationSize

- **Outline**: Get the number of nodes with reads in a classification

- **Parameters**

    – **fileId**

      * **Name**: fileId (see 6.1.1)
      * **Outline**: An identifier of a dataset. Either a path, a name or a number
      * **Required** : true
      * **Default Value** : –

    – **classification**

      * **Name**: classification (see 6.1.2)
      * **Outline**: One of the classifications MEGAN supports
      * **Required** : true
      * **Default Value** : –

- **Examples**:

    – http://megan-db.org/Public/getClassificationSize?fileId=1958397319&
    classification=Taxonomy

6.4 Admin

Accessible with admin permission only!

6.4.1 admin/updateDatasets

- **Name**: admin/updateDatasets

- **Outline**: Reads the root location from the *MeganServer.properties* file and updates the list of datasets

- **Parameters**

    – None

- **Examples**:

    – http://megan-db.org/Public/admin/updateDatasets

6.4.2 admin/addUser

- **Name**: addUser

- **Outline**: Get the number of nodes with reads in a classification

- **Parameters**

    – **userName**
        * **Name**: userName
        * **Outline**: Name of the new user
        * **Required** : true
        * **Default Value** : –

    – **password**
        * **Name**: password
        * **Outline**: password for the user
        * **Required** : true
        * **Default Value** : –

    – **isAdmin**
        * **Name**: isAdmin
        * **Outline**: Determines if the new user should get admin rights
        * **Required** : true
        * **Default Value** : –

- **Examples**:

– http://megan-db.org/Public/admin/addUser?userName=Megan&password=
Server&isAdmin=false

6.4.3 admin/listUsers

- **Name**: listUsers

- **Outline**: Get the list of active users

- **Parameters**

  – No parameters

- **Examples**:

  – http://megan-db.org/Public/admin/listUsers

6.4.4 admin/removeUser

- **Name**: admin/removeUser

- **Outline**: Delete user and revoke credentials

- **Parameters**

  – **userName**
    * **Name**: userName
    * **Outline**: Name of the new user
    * **Required** : true
    * **Default Value** : –

- **Examples**:

  – http://megan-db.org/Public/admin/removeUser?userName=Megan

6.4.5 admin/getLog

- **Name**: admin/getLog

- **Outline**: Returns log entries

- **Parameters**

  – No parameters

- **Examples**:

  – http://megan-db.org/Public/admin/getLog

7   Acknowledgements

References

[1] B. Buchfink, C. Xie, and D.H. Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12:59–60, 2015.

[2] D.H. Huson, , Sina Beier, Benjamin Buchfink, Isabell Flade, Anna Górska, Mohamed El-Hadidi, Suparna Mitra, Hans-Joachim Ruscheweyh, and Rewati Tappu. MEGAN6 - microbiome analysis involving hundreds of samples and billions of reads. Submitted, 2015.

[3] Sandrine Louis, Rewati-Mukund Tappu, Antje Damms-Machado, Daniel H. Huson, and Stephan C. Bischoff. Characterization of the gut microbial community of obese patients following a weight-loss intervention using whole metagenome shotgun sequencing. *PLoS ONE*, 11(2):1–18, 02 2016.

[4] Rachel Mackelprang, Mark Waldrop, Kristen DeAngelis, Maude David, Krystle Chavarria, Steven Blazewicz, Edward Rubin, and Janet Jansson. Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. *Nature*, 480(7377):368–371, 2011.

[5] Anita Y Voigt, Paul I Costea, Jens Roat Kultima, Simone S Li, Georg Zeller, Shinichi Sunagawa, and Peer Bork. Temporal and technical variability of human gut metagenomes. *Genome Biol*, 16:73, 2015.

[6] M. Willmann, M. El-Hadidi, D.H. Huson, M. Schütz, C. Weidenmaier, I.B. Autenrieth, and S. Peter. Antibiotic selection pressure determination through sequence-based metagenomics. *Antimicrobial Agents and Chemotherapy*, 59(12):7335–45, Dec 2015.