VERSION 1

MAR 09, 2023

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.kxygx91zzg8j/v1

Protocol Citation: Mateusz Jundzill, Riccardo Spott, Mara Lohde, Martin Hölzer, Adrian Viehweger, Christian Brandt 2023. SOP - Routine SARS-CoV-2 sequencing data administration . **protocols.io** https://dx.doi.org/10.17504/protocols.io.kxygx91zzg8j/v1Version created by Mateusz Jundzill

Protocol status: Working
We use this protocol and it's working

Created: Feb 21, 2023

Last Modified: Mar 09, 2023

PROTOCOL integer ID:
77372

SOP - Routine SARS-CoV-2 sequencing data administration V.1

Mateusz Jundzill[1,2], Riccardo Spott[1], Mara Lohde[1], Martin Hölzer[3], Adrian Viehweger[4], Christian Brandt[1,5,2]

[1]Institute for Infectious Diseases and Infection Control, Jena University Hospital, Jena, Germany;
[2]Leibniz Center for Photonics in Infection Research (LPI), Jena 07747, Germany;
[3]Methodology and Research Infrastructure, Bioinformatics and Systems Biology (MF1), Robert Koch Institute, Berlin 13353, Germany;
[4]Medical Microbiology and Virology, University Hospital Leipzig, Leipzig 04103, Germany;
[5]InfectoGnostics Research Campus, Jena 07747, Germany

Mateusz Jundzill

ABSTRACT

This is a step-by-step standard operating protocol (SOP) for data administration using the MongoDB database for routine SARS-CoV-2 sequencing.
The collapsed note blocks contain command-line solutions that are usually faster but require more advanced IT knowledge.
Additionally, we have enclosed a Github repository with a code and script approach to automate multiple data administration procedures.
Please note that since these codes are specifically made to be used with the SARS-CoV-2 internal database, they will require adjustments to work with other databases.
In this protocol, we have focused on using the MongoDB Compass GUI, but there are other solutions available. We recommend choosing the most suited solution for your needs.

Keywords: MongoDB, routine sequencing, data administration, database

Creating New Entries

1 **Sample arrives from external or internal partner.**

2 **Sample entry creation.** Fill the template CSV file with the given information and import it through MongoDB Compass.

Example of a template CSV:

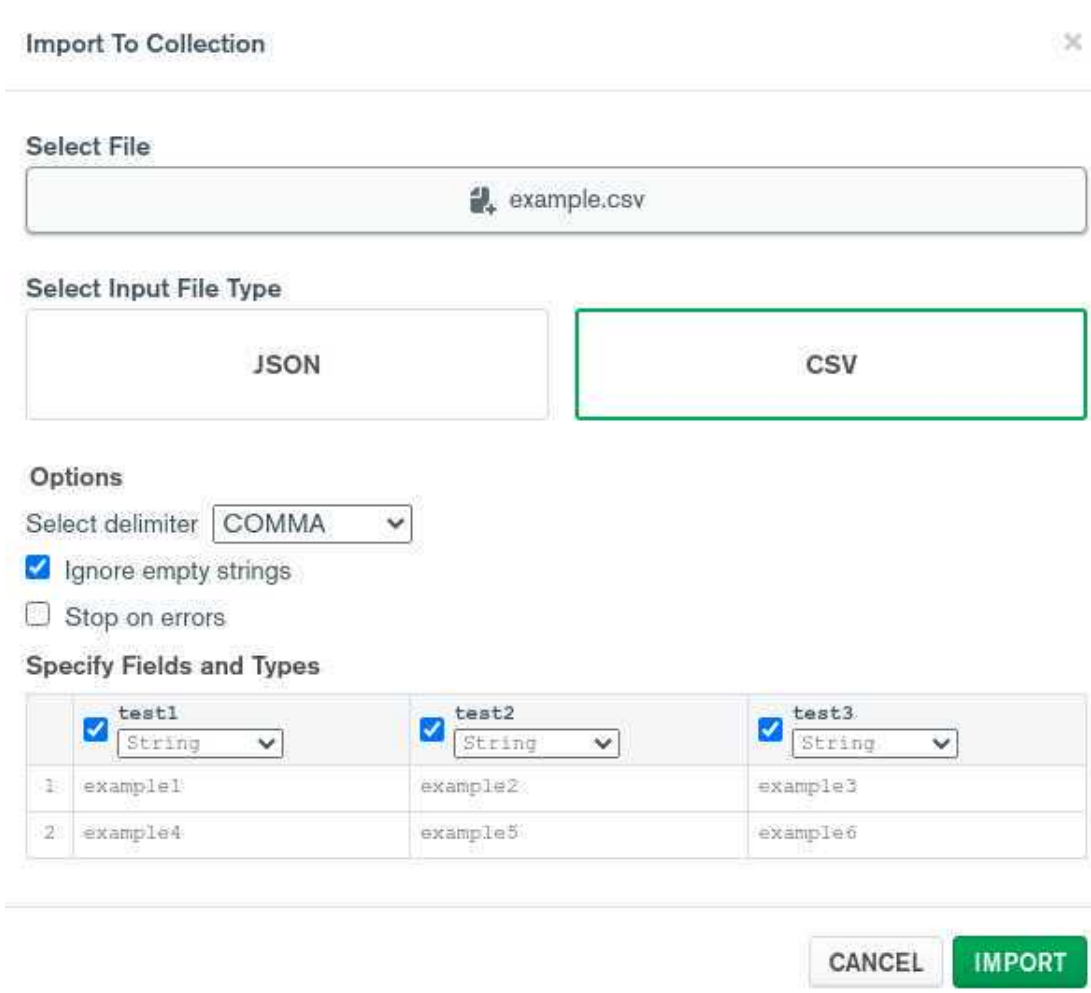
	A	B	C	D	E
	Sample_ID	Isolation_Date	Longitude	Latitude	...
	A12345	230101	11.586111	50.927223	...
	B12345	230101	11.586111	50.927223	...
	...	...	...	...	...

Longitude and latitude data can later be used for visualizing MongoDB Chart maps.

Import icon in MongoDB Compass.

Select *Import File* and choose a file for data upload. Pick *CSV* in the *Select Input File Type* option.

If the file is in proper format, a table with the headline *Specify Fields and Types* should appear, providing an overview of the imported data and allowing you to specify key types. Confirm by selecting *Import.*


Import To Collection
Select File
example.csv
Select Input File Type
JSON
CSV
Options
Select delimiter
COMMA
Ignore empty strings
Stop on errors
Specify Fields and Types
test1
String
test2
String
test3
String
1
example1
example2
example3
2
example4
example5
example6
CANCEL
IMPORT

Note

Command line import approach:
Enter the Docker container in the folder where CSV file is located.

```
docker run --rm -it -v $PWD:/input  mongo:4.0.22-xenial
/bin/bash
```

Execute command:

```
mongoimport \
--host "<hostname><:port>" \
--username <username> --password <password> \
--authenticationDatabase admin --ssl --db <database> \
--collection <collection name> --type csv --headerline \
--mode insert --file /input/<csv-file>
```

If Docker is not installed follow the official instalation manual.
If Mongo Tools are installed, the *mongoimport* command can be executed without entering the Docker container.

Sequencing preparation

3 **Select samples for laboratory work.** Access database using MongoDB Compass and search for samples that have an empty *Status* field.

Search query example:

```
{Status:""}
```

Search field in MongoDB Compass.

Fill *the Status* field with a short personal identifier and a sample barcode number.

Example:

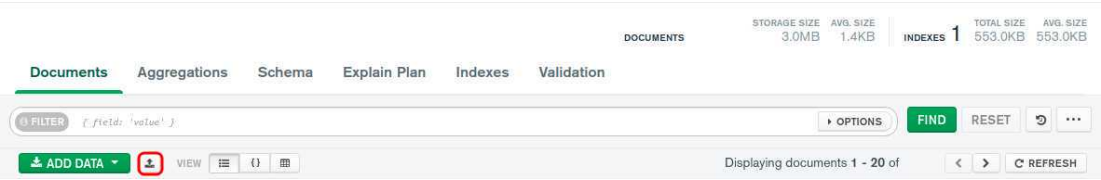
```
RS1_BC01
```

4 **(Optional) Export the sample list as a CSV file.** Search for the desired samples using MongoDB Compass and then export the results to a CSV file.

The exported CSV can also be used to update entries at a later step (see step 6).

Example of a queary that searches for a status that contais 'BC' (barcode).

```
{Status:{$regex: 'BC'}}
```



Export icon in MongoDB Compass.

Command line export approach:

Note

Command line export approach:

Enter the Docker container:

```
docker run --rm -it -v $PWD:/input  mongo:4.0.22-xenial
/bin/bash
```

Execute the following command:

```
query='{"Status":{"$regex":"BC"}}'

mongoexport --host "<hostname><:port>"\  --username <username>
--password <password>  \  --authenticationDatabase admin --ssl
--db <database> --collection <collection>  \  --
fields='_id,Sample_ID,Status,Ct_Value,Isolation_Date,PLZ,Seq_Re
ason,Sample_Type,Submitting_Lab,Sequencing_Lab,Storage,Comments
' \  --query "$query"  \  --type csv --out
/input/IMS_Sample_Overview.csv
```

5 **Print out the sample for the laboratory work.**

Sample analysis

6 **(Optional) Analyze the results your experiment** and update the list prepared in go to step #4 with any data that you want to store in the database.

Database entry update

7 **Update MongoDB database with new results.** There are three available solutions:
1. Manually update entries in MongoDB Compass.
Switch to the optimal data view in the application (table-like or JSON-like), depending on how many entries you need to view. Then, turn on the edit mode and in case you would like to add a

new key-value pair select the small plus sign. This will create a new key-value pair that you can modify. To modify an existing value, double-click on value you wish to modify. Confirm your changes by choosing *Update*.

2. Pure CSV solution. Export the list of analyzed samples by querying the common *Status* field fragment e.g. BC (Barcode):

```
{Status:{$regex: 'BC'}}
```

Update the exported CSV with new information and change the status of failed sequencing runs to *fail*. Then, delete the preexisting database entry and import the updated CSV file using MongoDB Compass.

Note

Command line update approach:

Enter the Docker container:

```
docker run --rm -it -v $PWD:/input  mongo:4.0.22-xenial
/bin/bash
```

Execute the following command:

```
for FILE in /input/*.json; do \
    echo "${FILE}" && \
    mongoimport \
        --host "<hostname><:port>"  --username <username> --
password <password> \
        --authenticationDatabase admin --ssl --db <database> \
        --collection <collection> --type json --mode merge --
file "${FILE}"; \
done 2>&1
```

8 **Check data integrity.** This could be performed in multiple ways:

1. Go to the MongoDB Compass Schema tab, select *Analyze Schema* and check your data for any discrepancies (such as numerical values out of bonds, repeated values, unusual data formats,

etc.)
2. Prepare MongoDB Charts data overview that works in a similar but customized way as MongoDB Compass Schema.
3. Use sanity scripts similar to the 'field_mod_exec.sh' script provided in the Github repository The script can also create a new field based on the given information. Please note that the provided scripts require modification and adjustments to work with databases other than the internal Routine SARS-CoV-2 sequencing database.

Prepare a report file

9 **Prepare report file for external partners.** Prepare results for the report by searching the database for the desired data, such as by Analysis Date or any other required field Then export the list with selected fields to a CSV file via MongoDB Compass.

Example to filter data by date of analysis:

```
{"Analysing_Date": <date of analysis>}
```

Note

Command line approach:

Enter the Docker container:

```
docker run --rm -it -v $PWD:/input  mongo:4.0.22-xenial
/bin/bash
```

Execute the following command:

```
query='{"Analysing_Date": <date of analysis>}'

mongoexport --host "<hostname><:port>"\
--username <username> --password <password> \
--authenticationDatabase admin --ssl --db <database> --
collection <collection> \
--fields='<fields to include in report>' --query "$query"  \
--type csv --out /input/Sample_Overview.csv
```