

CatStat

Seismic Catalog and Bulletin Quality Control Analysis MATLAB Package

Table of Contents

- I. [License](#)
- II. [Installation](#)
 - A. [GitHub](#)
 - B. [Download this code](#)
 - C. [Creating a local copy of a repository](#)
 - D. [Seeing the changes you made to a file](#)
 - E. [Adding a file to the local copy of a repository](#)
 - F. [Committing \(locally!\) the changes you've made to any files](#)
 - G. [Pushing those changes to GitHub](#)
 - H. [Submit a pull request via the github.com website](#)
 - 1. [Adding changes from the upstream master](#)
 - I. [How to re-sync your local copy with the "authoritative" repository](#)
 - J. [How to fork a new project...](#)
 - K. [MATLAB](#)
- III. [Catalogue/Bulletin Format Supported](#)
 - A. [Catalog Format 1: ComCat CSV](#)
 - B. [Catalog Format 2: LibComCat CSV](#)
 - C. [Catalog Format 3: ISC-Gem Catalog CSV](#)
- IV. [Single-Catalog QC Report \(mkQCReport.m\)](#)
 - A. [Input File \(initMkQCReport.dat\)](#)
 - B. [How to Run mkQCReport.m](#)
 - C. [Detailed Explanation of Algorithms/Methods Used](#)
 - D. [mkQCReport.m](#)
 - E. [QCReport.m](#)
 - 1. [loadcat.m](#)
 - 2. [trimcat.m](#)
 - 3. [basiccatsum.m](#)
 - 4. [catalogsize.m](#)
 - 5. [plotcatmap.m](#)
 - 6. [catdensplot.m](#)
 - 7. [plotcatdeps.m](#)
 - 8. [eventfreq.m](#)
- V. [Multiple-Catalog Comparison QC Report \(mkQCmulti.m\)](#)
 - A. [Input File \(initMkQCmulti.dat\)](#)
 - B. [Detailed Explanation of Algorithms/Methods Used](#)
- VI. [References](#)
- VII. [Caveats and Known Bugs](#)
- VIII. [Appendix](#)
 - A. [Region Information](#)

License

Unless otherwise noted, this software is in the public domain because it contains materials that originally came from the United States Geological Survey, an agency of the United States Department of Interior. For more information, see the official USGS copyright policy at http://www.usgs.gov/visual-id/credit_usgs.html#copyright

This information is preliminary or provisional and is subject to revision. It is being provided to meet the need for timely best science. The information has not received final approval by the USGS and is provided on the condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from the authorized or unauthorized use of the information.

Installation

GitHub

Download this code

```
git clone https://github.com/usgs/CatStat
```

Creating a local copy of a repository

```
git clone https://github.com/usgs/CatStat
```

Seeing the changes you made to a file

```
git diff FILE
```

Adding a file to the local copy of a repository

```
git status
```

```
git add FILE
```

Note: Changing the hardcoded path in the main script will indicate a difference between the local and master branch.

Committing (locally!) the changes you've made to any files

```
git commit -am "LOG MESSAGE"
```

```
git status
```

Pushing those changes to GitHub

```
git push origin master
```

Submit a pull request via the github.com website

Adding changes from the upstream master

Fetch the branches and their respective commits from upstream repository

```
git fetch upstream
```

Merge changes into LOCAL master branch, syncing with the upstream without losing local changes

```
git merge upstream/master
```

Review status of repository

git status

Sync local repository to the merged changes

git push origin master

How to re-sync your local copy with the "authoritative" repository

List the current configured remote repository for your fork

git remote -v

Specify new remote upstream repository to be synced with fork

git remote add upstream https://github.com/usgs/CatStat %

Verify new upstream repository

git remote -v

Then follow directions above for adding changes from the upstream master

For reference:

<https://help.github.com/articles/configuring-a-remote-for-a-fork/>

<https://help.github.com/articles/syncing-a-fork/>

How to fork a new project...

Set up Git & authentication to GitHub

Navigate to your fork of the new project repository on GitHub

git clone URL

Then follow directions above for syncing with original, upstream repository

For reference:

<https://help.github.com/articles/fork-a-repo/>

<https://help.github.com/articles/set-up-git/>

MATLAB

In order to run QCreport and QCmulti from any directory, you must add the CatStat directory to your MATLAB path. Adding the following line to your MATLAB *startup.m* file is the simplest way to accomplish this.

`addpath(genpath(PATH TO CATSTAT))`

Once this is added to your startup.m file, and assuming MATLAB is currently running, simply enter the command '*startup*' and MATLAB will run and read the startup.m script.

By successfully downloading the GitHub repository and adding the above line to the *startup.m* script, CatStat will be successfully installed. **No additional software or MATLAB packages are needed to run QCreport or QCmulti.**

Catalogue/Bulletin Format Supported

As of 1 August 2016, there are only two file formats supported by CatStat, which are explained below.

Catalog Format 1: ComCat CSV

If the data were obtained using the USGS NEIC API tools or with the ComCat search GUI, the output CSV file is explained [here](#) and contain the following fields:

time, latitude, longitude, depth, mag, magType, nst, gap, dmin, rms, net, id, updated, place, type, horizontalError, depthError, magError, magNst, status, locationSource, magSource

The time formats that are currently accepted under this format are:

yyyy-mm-ddTHH:MM:SS.FFFZ
yyyy-mm-ddTHH:MM:SS.FFF
yyyy-mm-dd HH:MM:SS.FFFZ
yyyy-mm-dd HH:MM:SS.FFF
yyyy-mm-ddTHH:MM:SSZ
yyyy-mm-ddTHH:MM:SS
yyyy-mm-dd HH:MM:SSZ
yyyy-mm-dd HH:MM:SS
yyyy-mm-dd HH:MM

Note: Work is currently underway to make the date-time field accept a variety of formats.

Catalog Format 2: LibComCat CSV

If the data were obtained using the *getcsv.py* tool in the [LibComCat Python package](#), the output CSV file will currently contain the following fields:

EventID, Time, Latitude, Longitude, Depth, Mag, EventType

The time formats that are currently accepted under this format are:

yyyy-mm-ddTHH:MM:SS.FFFZ
yyyy-mm-ddTHH:MM:SS.FFF
yyyy-mm-dd HH:MM:SS.FFFZ
yyyy-mm-dd HH:MM:SS.FFF
yyyy-mm-ddTHH:MM:SSZ
yyyy-mm-ddTHH:MM:SS
yyyy-mm-dd HH:MM:SSZ

yyyy-mm-dd HH:MM:SS
yyyy-mm-dd HH:MM

Note: Work is currently underway to make the date-time field accept a variety of formats.

Catalog Format 3: ISC-Gem Catalog CSV

If the data were obtained from [ISC](#), or in a similar format, the resulting CSV will contain 59 rows of header information and the following fields:

Date,lat,lon,smajax,sminax,strike,q,depth,unc,q,mw,unc,q,s,mo,fac,mo_auth,mpp,mpr,mrr,mrt,mtp,mtt,eventid

However, only the EventID, date, lat, lon, dep, and Mw columns are used in mkQCReport.m

The time format that are current accepted under this format are:

yyyy-mm-dd HH:MM:SS.FF (native ISC-GEM time format)

Note: Work is currently underway to make the date-time field accept a variety of formats.

Single-Catalog QC Report (mkQCReport.m)

The single-catalog QC Report generated by *mkQCReport.m* provides the user with a baseline set of statistics and graphs characterizing and depicting the contents of the catalogue/bulletin. It's purpose is to allow the user to efficiently assess the quality of the data and determine any large error, *i.e.* missing data, systematic catalogue errors, and duplicate postings or entries for the same events. The results are not intended to be a complete investigation of any given catalogue, but as a general, and preliminary look into the completeness and overall quality of data contained in the data set.

Input File (initMkQCReport.dat)

The MATLAB algorithm used to read the contents of the initMkQCReport.dat file relies on line numbers to parse the data. Therefore, order is very important.

The following is an example input file (please save input files as initMkQCReport.dat)

```
% Catalog Data File Name and Path (for completeness, always include full path)
path/to/catalog/dataset/datasetname.csv
% Human readable catalog description
String describing the catalog
% Catalog Format: 1 = ComCat CSV, 2 = libcomcat CSV, 3 = ISC-GEM CSV
Integer specifying catalog format number
```

```
% UTC offset [hours]
Integer either + or - depending on time zone
% Time Zone Name
String - Name of time zone for use in plot labels
% Authoritative Region, if none put all
String - See Appendix for abbreviations
%Directory to Put Report File Into
Example_Report_Directory
%File Format for Report (HTML default / PDF) See MATLAB Publish documentation
See publish documentation
% Show code in report (will be removed in future releases)
false
```

How to Run mkQCreport.m

As long as the appropriate steps were taken during [installation](#), mkQCreport can be run from any directory. For example, say I want to run a QC report on the Southern California Catalog and that my catalog is located in /Users/usgs/CI/ as CI.csv. I create a file called “initMkQCreport.dat,” and fill out the fields contained in the above example as:

```
% Catalog Data File Name and Path (for completeness, always include full path)
/Users/usgs/CI/CI.csv
% Human readable catalog description
Southern California Seismic Catalog
% Catalog Format: 1 = ComCat CSV, 2 = libcomcat CSV, 3 = ISC-GEM CSV
1
% UTC offset [hours]
-8
% Time Zone Name
Pacific Time Zone
% Authoritative Region, if none put all
ci
%Directory to Put Report File Into
CI_Catalog_Report
%File Format for Report (HTML default / PDF) See MATLAB Publish documentation
html
% Show code in report (will be removed in future releases)
false
```

Now, within MATLAB, change directories to the folder contained the input file created above (*cd* /Users/usgs/CI) and type *mkQCreport*. This should begin producing a report in the specified format (html/xml/pdf etc) under the directory specified on line 12 on the input file. If for some reason you do not see “Using local initMkQCreport.dat file” in the MATLAB command window, then either

initMkQCreport.dat isn't in your current working directory, or there was a mistake in your path. Because of the way mkQCreport is currently set-up, the program will run, but will use the default *initMkQCreport.dat* file.

Detailed Explanation of Algorithms/Methods Used

mkQCreport.m

This is the parent script responsible for reading the input file and generating the report, through the use of the *publish* command in MATLAB.

QCreport.m

QCreport.m is the script that ties the algorithms and functions used in generating the statistics and relevant plots for the report. Work is currently underway for each functions to be self-sustaining (i.e. can comment out all but a few functions, and the active functions will still produce output without relying on other “less-important” functions).

loadcat.m

This function loads catalog/bulletin data from a variety of [formats](#) (ComCat, LibComCat, and ISC-GEM). This loading script requires *mkQCreport.m* to run prior in order to read the input file and gather the needed information i.e., catalog location, format, etc.. If it is desired to run *loadcat.m* without relying on *mkQCreport.m*, the input must be a MATLAB structure with the file location and path saved under *.file*, and the format saved under *.format*.

The output from this function will be a MATLAB data structure with the following fields:

- .name -- name of the catalog (Human readable catalog description from input file)
- .file -- name and path of the catalog file (from input file)
- .data -- Hypocenter information for the catalog (origin time, latitude, longitude, depth, magnitude)
- .id -- Event IDs (rows correspond with rows in .data)
- .evtype -- Event type associated with events in .data

trimcat.m

If a region other than ‘all’ is selected, this functions will return spatially filtered data using the MATLAB function [inpolygon](#). For region information and abbreviations, please refer to the [appendix](#). Work is currently underway to allow for temporal filtering (an option that is currently working in *mkQCMulti.m*). If a region is not selected (i.e. all), this function is by-passed.

basiccatsum.m

This function provides basic catalog information and statistics, and requires the input to be the data structure created by *loadcat.m*. If the desire is to run this function individually the data structure passed must contain:

.name - Catalog Name
.file - Catalog File and Path
.data - Hypocenter Information (Origin Time, latitude, longitude, depth, magnitude)
.evtype - Event Type for corresponding event in .data

There is variable output for this function, but it will print out a variety of statistics including catalog time-span, minimum and maximum latitude, longitude, depth, and magnitude. Number of 0 and NaN magnitudes and depths, as well as a count of each event type contained in the catalog.

catalogsize.m

This function determines if plots should be made by year, month, or day based on the time span of the catalog. Technically, the only information needed for this function is a data structure with origin time (in MATLAB datenum format) in the first column. The output will be either 1: if more than 5 month-year combinations and more than 3 years in total, use yearly plotting; 2: If more than 5 month-year combinations, but less than 3 years in total, then use monthly plotting; 3: If less than 5 month-year combination, then use daily plotting. This function will eventually be assimilated into *basiccatsum.m*.

plotcatmap.m

As indicated in the function name, this function produces an event map of the catalog, it also returns data relevant to earthquakes (i.e. *equakevent*, *equakevent_ids*), which are in the same format as *.data* created by *loadcat.m*. The required input is the data structure created in *loadcat.m*, or modified by *trimcat.m*. This function also requires the user to pass a variable that defines the regional polygon. Remember, if no region is desired the region must be set to 'all', or the program will crash.

The map produced by this function will display country and state boundaries (solid black lines), regional polygon boundary (thick dashed black lines), if selected, earthquakes (red dots), and other seismic events (blue dots). The map will automatically adjust in order to best display the given data.

catdensplot.m

This function creates two seismicity density plots for the catalog. The first is a density plot with linear scaling, while the second has logarithmic scaling. No output other than the maps is produced, and the function requires data in the format of the data structure produced in *loadcat.m*.

plotcatdeps.m

This function produces plot(s) of earthquake depth distribution in the catalog. Only one plot is produced if no depth exceed 50 km. However, if there are depths deeper than 50 km, three plots will be produced. The first will be a histogram of those events more shallow than 50 km. The second and third plots will be a histogram of those events with depths greater than 50 km and a map showing the location of the deeper events. This function requires a matrix of hypocenter information and name of the region in consideration. Again, if no region is selected, 'all' must be the region name.

eventfreq.m

This function creates a histogram of number of earthquakes per day. It requires two inputs: a matrix, as described before, of hypocenter information for earthquakes in the catalog, and the sizenum determined in [catalogsize.m](#).

hreventfreq.m

As with eventfreq.m, this function requires two inputs: a matrix of hypocenter data and the sizenum for the catalog. A plot showing the number of events per hour of the day in the respective time zone of the catalog (see [Time Zone](#)). Typically, one should see a general decrease in the number of earthquakes detected during the day, as anthropomorphic noise increases (reference needed??).

inteventspace.m

The temporal spacing between subsequent events is an important indicator of the completeness of a catalog (REFERENCE??). This function, unlike the previous two, requires the catalog data structure created in [loadcat.m](#), and the sizenum determined by [catalogsize.m](#). There is no variable output, but this function will print a few statistics to the screen and produce a stem plot of interevent times in days. In the future, we hope to use this function as a stepping stone in determining catalog completeness time.

catmagdistrib.m

This function determines magnitude statistics and distribution of earthquake events throughout the catalog. It will print out minimum and maximum magnitudes, as well as provide a count of NaN and 0 magnitude events. The plot that is produced is an undithered scatter plot of magnitudes through time.

catmaghist.m

A magnitude histogram is produced by this function. Input is the egevents matrix created in [plotcatmap.m](#).

magspecs.m

If the catalog has more than 5 month-year combinations and more than 3 years in total, this function will produce a plot of magnitude “bands” counts per year. For example, it’ll count all the earthquakes that are greater than 0 but less than 1 by year. It’ll do the same until it reaches the “band” that contains the maximum magnitude in the catalog. Work is underway to better control the display of these graphs within the QC report.

plotyrmedmag.m

This function calculates and plots the yearly median magnitude for earthquakes only, and requires a matrix of earthquake hypocenters and sizenum created in [plotcatmap.m](#) and [catalogsize.m](#), respectively.

Multiple-Catalog Comparison QC Report (mkQCmulti.m)

The multiple-catalog QC Report generated by *mkQCmulti.m* provides the user with a baseline set of statistics and graphs characterizing and depicting the contents of each catalogue/bulletin, as well as events deemed to be matching, missing, or duplicated between the catalogs/bulletins. Its purpose is to allow the user to efficiently assess the quality of the data and determine any large errors, *i.e.* missing data, systematic catalogue errors, and duplicate postings or entries for the same events. The results are not intended to be a complete investigation of any given catalogue, but as a general, and preliminary look into the completeness and overall quality of data contained in each data set.

Input File (initMkQCmulti.dat)

The MATLAB algorithm used to read the contents of the initMkQCmulti.dat file relies on line numbers to parse the data. Therefore, order is very important.

The following is an example input file (please save input files as initMkQCmulti.dat)

```
% First catalog data file name
/Users/usgs/CI/CI.csv
% Human readable catalog description
String describing the first catalog
% Catalog Format 1 = ComCat CSV; 2 = libcomcat CSV format; 3 = ISC-GEM CSV
Integer specifying catalog format number
% Second catalog Data file name
/path/to/first/catalog/catalog2.csv
% Human readable catalog description
String describing the second catalog
% Catalog Format 1 = ComCat CSV; 2 = libcomcat CSV format
Integer specifying catalog format number
% Time Window
Time window in seconds for event association (matching time tolerance)
% Distance Range
Distance window in kilometers for event association (matching distance tolerance)
% Region
String - See Appendix for abbreviations
% Set Mag Limit
Lower Magnitude limit to be considered
% Set Mag Difference Max
Magnitude tolerance for event association (set to 9999, if not needed)
% Set Depth difference tolerance
Depth tolerance in kilometers for event association (set to 9999, if not needed)
```

```

% Diretory to put report in
Example_Report_Directory
% File format for report
See publish documentation
% show code in report (will be removed in future versions)
true/false
% Display event list (will display missing/matching/etc. Event lists)
Yes / no
% Compared Authoritative IDs (Select to compare EventIDs or to check for authoritative region
solutions)
yes/no

```

How to Run mkQCmulti.m

As long as the appropriate steps were taken during [installation](#), *mkQCmulti* can be run from any directory. For example, say I want to compare the Southern California Catalog currently in ComCat with a CSV file provided by CI that contains events they tried to upload. The files are located in */Users/usgs/CI/CI.csv* and */Users/usgs/CI/CI_Prenpload.csv*. I create a file called “initMkQCmulti.dat,” and fill out the fields contained in the above example as:

```

% First catalog data file name
/Users/usgs/CI/CI.csv
% Human readable catalog descriptcity ion
CI from ComCat
% Catalog Format 1 = ComCat CSV; 2 = libcomcat CSV format; 3 = ISC-GEM CSV
1
% Second catalog Data file name
/Users/usgs/CI/CI_Prenpload.csv
% Human readable catalog description
CI Prenpload CSV
% Catalog Format 1 = ComCat CSV; 2 = libcomcat CSV format
2
% Time Window
16
% Distance Range
100
% Region
ci
% Set Mag Limit
-10
% Set Mag Difference Max
1
% Set Depth difference tolerance

```

25

% Directory to put report in

CI_CIPre_Compare

% File format for report

html

% show code in report (will be removed in future versions)

false

% Display event list (will display missing/matching/etc. Event lists)

yes

% Compared Authoritative IDs (Select to compare EventIDs or to check for authoritative region solutions)

yes

Now, within MATLAB, change directories to the folder contained the input file created above (*cd /Users/usgs/CI*) and type *mkQCmulti*. This should begin producing a report in the specified format (html/xml/pdf etc) under the directory specified on line 12 on the input file. If for some reason you do not see “Using local initMkQCmulti.dat file” in the MATLAB command window, then either initMkQCmulti.dat isn’t in your current working directory, or there was a mistake in your path. Because of the way mkQCmulti is currently set-up, the program will not run if it cannot find the input file.

Detailed Explanation of Algorithms/Methods Used

References

Caveats and Known Bugs

Appendix

I. Region Information