FMC—Earthquake focal mechanisms data management, cluster and classification

José A. Álvarez-Gómez

Department of Geodynamics, Stratigraphy and Palaeontology; Faculty of Geology; Complutense University of Madrid, C/ José Antonio Novais, 12. 28040 Madrid, Spain

Seismicity is frequently used to deduce the tectonics of a region. The study of earthquakes as a tectonic component, seismotectonics, has grown as one of the key research areas on active tectonics, especially from the analysis of earthquake focal mechanisms. FMC computes the different earthquake parameters that can be obtained from focal mechanism data, classifies the rupture type of each focal mechanism, performs a clustering analysis of the data if required by the user, outputs the parameters in different formats and generates a classification diagram from the input data.

1. Motivation and significance

The first earthquake focal mechanism determination methods, based on P wave first motion polarities, were developed in the first half of the 20th century, more specifically in Japan where a dense seismic network was available [1–5]. Since 1960, computers have allowed the numerical determination of fault-plane solutions with different, more robust, methods [e.g. 6–8]. In parallel with the development of modern seismology, the theory of plate tectonics changed the way geologists understand the Earth. Consequently, the study of earthquakes related to plate tectonics was developed at the same time, the basic concepts of seismotectonics and the lithospheric deformation were established [e.g. 9–12].

Since the 70s, earthquake focal mechanisms started to be computed in a systematic way and global catalogues of focal mechanisms. Because of the continuous increase in data available, we need new tools to analyse it systematically. In order to represent focal mechanism populations, Frohlich and Apperson [13] proposed a diagram to visualize focal mechanism data as a function of the rupture type. This representation is popular and widely used in seismotectonics to represent the focal mechanisms of the study area [e.g. 14–19]. However, since it is significantly distorted towards the lower corners [20], Kagan [21] used the Kaverina equal-area projection [22] to avoid them. The difference between both diagrams is similar to that found between gnomonic and Lambert azimuthal equal-area projections (see the FMC manual for more details).

The aim of the FMC program is to provide a simple but powerful tool to manage focal mechanism data, classify the events according to the earthquake double-couple (DC) rupture type, and optionally perform a clustering analysis and plot a classification diagram based on DC characteristics. The combination of these functionalities allows a deeper analysis of earthquake ruptures...
Table 1

Focal mechanisms inversion and analysis software.

<table>
<thead>
<tr>
<th>Software</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single focal mechanism inversion and plotting programs</td>
<td></td>
</tr>
<tr>
<td>FOCMEC</td>
<td>Snoke, 1984 [23]</td>
</tr>
<tr>
<td>EARTHFLOWORM</td>
<td>Johnson et al. 1995 [24]</td>
</tr>
<tr>
<td>Coral tools</td>
<td>Crecer 1997 [25]</td>
</tr>
<tr>
<td>Rake</td>
<td>Louvari &amp; Kiratzi, 1997 [26]</td>
</tr>
<tr>
<td>Geotouch</td>
<td>Lees, 2000 [27]</td>
</tr>
<tr>
<td>FIFS/PACK</td>
<td>Gasperini &amp; Vannucci [28]</td>
</tr>
<tr>
<td>Dishansh 2005</td>
<td>Srivastava et al. 2006 [29]</td>
</tr>
<tr>
<td>MIRONE</td>
<td>Luis, 2007 [30]</td>
</tr>
<tr>
<td>3DFM</td>
<td>Labay &amp; Haeussler, 2007 [31]</td>
</tr>
<tr>
<td>Earthquake Focal Mechanism</td>
<td>Scherbaum et al. 2009 [32]</td>
</tr>
<tr>
<td>SeisComp</td>
<td>Hanka et al. 2010 [33]</td>
</tr>
<tr>
<td>ModPAD</td>
<td>Krieger &amp; Heimann, 2012 [34]</td>
</tr>
<tr>
<td>Computer programs in seismology</td>
<td>Herrmann, 2013 [35]</td>
</tr>
<tr>
<td>focalmech</td>
<td>Conder, 2017 [36]</td>
</tr>
<tr>
<td>Grond (Pyrocko)</td>
<td>Heimann et al. 2018 [37]</td>
</tr>
<tr>
<td>bb</td>
<td>CERL, 2019 [38]</td>
</tr>
<tr>
<td>Focal Mechanisms</td>
<td>Hoffrich, 2019 [39]</td>
</tr>
<tr>
<td>psmeca, pscoupe (GMT)</td>
<td>Patau, 2019 [40]</td>
</tr>
<tr>
<td>PyTDMT (ObsPy)</td>
<td>Bernardi, 2019 [41]</td>
</tr>
</tbody>
</table>

Software population analysis and plotting programs.

| Geotouch                  | Lees, 2000 [27]     |
| EQuakes                   | Listur, 2010 [42]   |
| TFM tools                 | Khalil & Al-Arifi, 2018 [43] |

The program input and outputs can be performed by means of ASCII files or using standard input (or redirection "<"), standard output (screen or redirection ">") and pipes ("|"). By default, FMC will read the input and write the output as a Harvard CMT (psmeca formatted) ASCII file. The input format can be changed by the program option modifier "-i", while the output format is selected with the "-o" modifier.

Data should be entered into the program using one of the three focal mechanism formats of the GMT (Generic Mapping Tools) package [59]. The formats are the Harvard CMT convention, the two nodal planes old Harvard CMT format for psmeca, and the single nodal plane Aki and Richards [70] convention. The former is a complete format that can be downloaded directly from the Global CMT site (http://www.globalcmt.org/), while the latter is the simplest way to incorporate earthquake rupture data.

Optionally, FMC will produce a Kaverina-type DC classification diagram (with the program option "-p"). FMC uses matplotlib libraries and can generate figures in different formats (emf, eps, jpeg, jpg, pdf, png, ps, raw, rgb, svg, svgz, tif, tiff). The format is determined automatically from the plot file name extension.

The diagram is based on the Kaverina [22] projection technique, used also by Kagan [21], but it incorporates a DC classification similar to the geological conceptual classification of faults. The earthquakes are classified into seven types according to the values of the P, T and B Centroid Moment Tensor axes following a simple algorithm and are opportunely represented on the Kaverina diagram (Fig. 1). This classification is very similar to the one used by Johnston et al. [71–73]. Currently, FMC produces only DC classification diagrams. For source-type classification diagrams, the reader can refer to recent works on the subject [48,74,75].

Common practice when working with seismic moment tensors requires decomposition of the tensor, which is iso-deviatoric following the procedure implemented by Gasperini and Vannucci [28]. The compensated linear vector dipole ratio, fclvd, which measures how different a source is from a “pure” double couple, is computed as defined by Frohlich and Apperson [13]. In order to obtain the nodal planes of the double couple, the orientation of the main axes (P, B, T) is computed from the deviatoric moment tensor (the P- and T-axis being the same for the DC and the CLVD components). Nodal plane orientations and slip vectors are obtained geometrically from the P- and T-axis. Inverse computation can also be performed, obtaining the P-, T- and B-axis orientations from the nodal planes (in fact only one nodal plane is necessary as both are mutually orthogonal and kinematically constrained). In this case, the moment tensor obtained is a pure DC with fclvd = 0.

FMC implements the hierarchical agglomerative clustering algorithms from SciPy to group data. The advantages of these algorithms are their versatility, as the user can choose between a number of metrics and grouping methods, their capacity to automatically select a minimum number of clusters without a priori estimation, and their potential to work with different parameters with different scales and strong different populations in clusters.

2. Software description

The program has been designed with the modularity and versatility of the classical UNIX-like tools. It is called from the command line and can be easily integrated into shell scripts ("NIX systems") or batch files (DOS/Windows systems).

FMC was originally programmed in Python 2.7.3 using several common Python libraries: sys, argparse, os, NumPy (version 1.14 or higher) and matplotlib. Since version 1.3, FMC also works on Python 3. The core functions for focal mechanism data manipulation adapt some FORTRAN subroutines from Gasperini & Vannucci [28].

2.1. Software functionalities

The program input and outputs can be performed by means of ASCII files or using standard input (or redirection "<"), standard output (screen or redirection ">") and pipes ("|"). By default, FMC will read the input and write the output as a Harvard CMT (psmeca formatted) ASCII file. The input format can be changed by the program option modifier "-i", while the output format is selected with the "-o" modifier.

Data should be entered into the program using one of the three focal mechanism formats of the GMT (Generic Mapping Tools) package [59]. The formats are the Harvard CMT convention, the two nodal planes old Harvard CMT format for psmeca, and the single nodal plane Aki and Richards [70] convention. The former is a complete format that can be downloaded directly from the Global CMT site (http://www.globalcmt.org/), while the latter is the simplest way to incorporate earthquake rupture data.

Optionally, FMC will produce a Kaverina-type DC classification diagram (with the program option "-p"). FMC uses matplotlib libraries and can generate figures in different formats (emf, eps, jpeg, jpg, pdf, png, ps, raw, rgb, svg, svgz, tif, tiff). The format is determined automatically from the plot file name extension.

The diagram is based on the Kaverina [22] projection technique, used also by Kagan [21], but it incorporates a DC classification similar to the geological conceptual classification of faults. The earthquakes are classified into seven types according to the values of the P, T and B Centroid Moment Tensor axes following a simple algorithm and are opportunely represented on the Kaverina diagram (Fig. 1). This classification is very similar to the one used by Johnston et al. [71–73]. Currently, FMC produces only DC classification diagrams. For source-type classification diagrams, the reader can refer to recent works on the subject [48,74,75].

Common practice when working with seismic moment tensors requires decomposition of the tensor, which is iso-deviatoric following the procedure implemented by Gasperini and Vannucci [28]. The compensated linear vector dipole ratio, fclvd, which measures how different a source is from a “pure” double couple, is computed as defined by Frohlich and Apperson [13]. In order to obtain the nodal planes of the double couple, the orientation of the main axes (P, B, T) is computed from the deviatoric moment tensor (the P- and T-axis being the same for the DC and the CLVD components). Nodal plane orientations and slip vectors are obtained geometrically from the P- and T-axis. Inverse computation can also be performed, obtaining the P-, T- and B-axis orientations from the nodal planes (in fact only one nodal plane is necessary as both are mutually orthogonal and kinematically constrained). In this case, the moment tensor obtained is a pure DC with fclvd = 0.

FMC implements the hierarchical agglomerative clustering algorithms from SciPy to group data. The advantages of these algorithms are their versatility, as the user can choose between a number of metrics and grouping methods, their capacity to automatically select a minimum number of clusters without a priori estimation, and their potential to work with different parameters with different scales and strong different populations in clusters.

2.2. Available command line switches

The program uses different options or flags controlling the following aspects: (i) input formats, (ii) output formats, (iii) plotting options and (iv) clustering options. Additionally there is the common "-v" verbose option for information.
2.2.1. Input

FMC input can be given as an ASCII file or as standard input, from a pipe ("|") or a redirection ("<"). The following codes are then equivalent:

FMC.py input-file.dat
cat input-file.dat | FMC.py
FMC.py < input-file.dat

If no input is given, then FMC will show the on-screen help. The input format is specified with the optional flag "-i", and the possible values are:

**CMT** Harvard Centroid Moment Tensor (by default):

- longitude, latitude, depth, mrr, mtt, mff, mrt, mrf, mtf, Exponent (dyn.cm), X plot, Y plot (for GMT), ID

**AR** Aki and Richards one plane convention:

- longitude, latitude, depth, strike, dip, rake, magnitude (Mw), X plot, Y plot (for GMT), ID

**P** Focal mechanism both nodal planes:

- longitude, latitude, depth, strike A, dip A, rake A, strike B, dip B, rake B, Scalar seismic moment mantissa, Exponent (dyn.cm), X plot, Y plot (for GMT), ID

2.2.2. Output

FMC output format can be selected among the following options with the flag "-o":

**CMT** Harvard Centroid Moment Tensor (psmeca compatible):

- longitude, latitude, depth, mrr, mtt, mff, mrt, mrf, mtf, Exponent (dyn.cm), X plot, Y plot (for GMT), ID, TYPE

**P** Focal mechanism both nodal planes (psmeca compatible):

- longitude, latitude, depth, strike A, dip A, rake A, strike B, dip B, rake B, Scalar seismic moment mantissa, Exponent (dyn.cm), X plot, Y plot (for GMT), ID, TYPE

**AR** Focal mechanism one plane (psmeca compatible):

- longitude, latitude, depth, strike, dip, rake, magnitude (Mw), X plot, Y plot (for GMT), ID, TYPE

**K** Kaverina diagram position for plotting outside FMC:

- [X Kaverina diagram, Y Kaverina diagram, Mw, Depth, ID, TYPE]

**ALL** All parameters obtained:

- longitude, latitude, depth, mrr, mtt, mff, mrt, mrf, Exponent (dyn.cm), Scalar seismic moment (dyn.cm), Mw, strike A, dip A, rake A, strike B, dip B, rake B, Slip trend A, Slip plunge A, Slip trend B, Slip plunge B, P trend, P plunge, B trend, B plunge, T trend, T plunge, fclvd, isotropic component, X Kaverina diagram, Y Kaverina diagram, ID, TYPE

**CUSTOM** In case you need any focal mechanism parameters in any order you can use the CUSTOM option and give the requested parameters using the flag "-of". The output parameters need to be listed separated by commas. The accepted parameter names are listed below and can be seen on the terminal using FMC.py -helpFields

### Table 2
Parameter names used in FMC.

<table>
<thead>
<tr>
<th>Code</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>lon</td>
<td>longitude</td>
</tr>
<tr>
<td>lat</td>
<td>latitude</td>
</tr>
<tr>
<td>dep</td>
<td>depth</td>
</tr>
<tr>
<td>mrr</td>
<td>mrr centroid moment tensor component</td>
</tr>
<tr>
<td>mtt</td>
<td>mtt centroid moment tensor component</td>
</tr>
<tr>
<td>mff</td>
<td>mff centroid moment tensor component</td>
</tr>
<tr>
<td>mrt</td>
<td>mrt centroid moment tensor component</td>
</tr>
<tr>
<td>mrf</td>
<td>mrf centroid moment tensor component</td>
</tr>
<tr>
<td>mtf</td>
<td>mtf centroid moment tensor component</td>
</tr>
<tr>
<td>mant</td>
<td>mantissa of the seismic moment tensor</td>
</tr>
<tr>
<td>expo</td>
<td>exponent of the seismic moment tensor</td>
</tr>
<tr>
<td>Mo</td>
<td>Scalar seismic moment</td>
</tr>
<tr>
<td>Mw</td>
<td>Moment (or Kanamori) magnitude</td>
</tr>
<tr>
<td>posX</td>
<td>X plotting position for GMT psmeca</td>
</tr>
<tr>
<td>posY</td>
<td>Y plotting position for GMT psmeca</td>
</tr>
<tr>
<td>ID</td>
<td>ID of the event</td>
</tr>
<tr>
<td>clas</td>
<td>Focal mechanism rupture type</td>
</tr>
<tr>
<td>strA</td>
<td>Strike of nodal plane A</td>
</tr>
<tr>
<td>dipA</td>
<td>Dip of nodal plane A</td>
</tr>
<tr>
<td>rakeA</td>
<td>Rake of nodal plane A</td>
</tr>
<tr>
<td>strB</td>
<td>Strike of nodal plane B</td>
</tr>
<tr>
<td>dipB</td>
<td>Dip of nodal plane B</td>
</tr>
<tr>
<td>rakeB</td>
<td>Rake of nodal plane B</td>
</tr>
<tr>
<td>slipA</td>
<td>Slip sense of plane A</td>
</tr>
<tr>
<td>plungA</td>
<td>Plunge of slip vector of plane A</td>
</tr>
<tr>
<td>slipB</td>
<td>Slip sense of plane B</td>
</tr>
<tr>
<td>plungB</td>
<td>Plunge of slip vector of plane B</td>
</tr>
<tr>
<td>trendp</td>
<td>Trend of P axis</td>
</tr>
<tr>
<td>plungp</td>
<td>Plunge of P axis</td>
</tr>
<tr>
<td>trendb</td>
<td>Trend of B axis</td>
</tr>
<tr>
<td>plunbg</td>
<td>Plunge of B axis</td>
</tr>
<tr>
<td>trendt</td>
<td>Trend of T axis</td>
</tr>
<tr>
<td>plungt</td>
<td>Plunge of T axis</td>
</tr>
<tr>
<td>fclvd</td>
<td>Compensated linear vector dipole ratio</td>
</tr>
<tr>
<td>iso</td>
<td>Moment tensor isotropic component</td>
</tr>
<tr>
<td>x_kav</td>
<td>x position on the Kaverina diagram</td>
</tr>
<tr>
<td>y_kav</td>
<td>y position on the Kaverina diagram</td>
</tr>
</tbody>
</table>

2.2.3. Plot

FMC uses several flags to customize the plot.

- **-p** This flag activates the plotting. It must be followed by the name of the figure file that will be produced. The name chosen for the file (without the extension) is used as a title for the plot.

- **-pc** With this flag the user specifies the parameter that is used to fill the symbols. A colour palette is produced with the selected parameter value range.

- **-pa** This flag is used to annotate the symbols with a certain parameter.

- **-pg** If present, the program will plot grid-lines with the specified angular spacing on the diagram plot (10° by default).

With -pc and -pa, the parameters must be given with their corresponding internal name as listed in Table 2.

2.2.4. Clustering

With the increment of focal mechanisms available for studying the seismotectonics of an area or a seismic series, the need for tools to perform statistical analyses has grown. The clustering analysis is now a common statistical tool that allows a large amount of different clustering strategies to be adapted to any problem.

I have chosen to include hierarchical clustering (connectivity-based clustering) as a clustering algorithm in FMC. Although other algorithms can work well with focal mechanisms, hierarchical
clustering presents a suite of linkage methods and distance metrics that makes this algorithm very versatile and users can choose which method and metric is better suited to their specific tasks. This clustering algorithm groups every event into a cluster and consequently every focal mechanism is used and incorporated in the analysis. Other alternative clustering algorithms will be included in future releases of FMC after they have been tested. The parameters for the clustering are passed by several optional flags. If any of the following flags is given in the command line FMC will perform the clustering analysis using some default options if needed. When a clustering analysis is done, by default FMC will shade the symbols in the diagram using the cluster number unless a different parameter is stated with `-pc` flag.

- **-cm** Method to be used in the clustering analysis.\(^1\)

The options are:

- **single** single/minimum/nearest
  \[d(u, v) = \min(\text{dist}(u[i], v[j]))\]
- **complete** complete/max/farthest point
  \[d(u, v) = \max(\text{dist}(u[i], v[j]))\]
- **average** average/UPGMA
  \[d(u, v) = \frac{1}{|J|} \sum_{j \in J} d(u[i], v[j])\]
- **weighted** weighted/WPGMA
  \[d(u, v) = \frac{\text{dist}(s, v) + \text{dist}(t, v))}{2}\]
- **centroid** centroid/UPGMC [default]
  \[\text{dist}(s, t) = \|c_s - c_t\|_2\]
  where \(c_s\) and \(c_t\) are the centroids of clusters \(s\) and \(t\), respectively. When two clusters \(s\) and \(t\) are combined into a new cluster \(u\), the new centroid is computed over all the original objects in clusters \(s\) and \(t\). The distance then becomes the Euclidean distance between the centroid of \(u\) and the centroid of a remaining cluster \(v\) in the forest.

- **median** median/WPGMC, assigns \(d(s, t)\) like the centroid method. When two clusters \(s\) and \(t\) are combined into a new cluster \(u\), the average of centroids \(s\) and \(t\) give the new centroid.

- **ward** Ward variance minimization algorithm.
  \[d(u, v) = \sqrt{\frac{|s| + |t|}{T} d(v, s)^2 + \frac{|v| + |t|}{T} d(v, t)^2 - \frac{|v|}{T} d(s, t)^2}\]
  where \(u\) is the newly joined cluster consisting of clusters \(s\) and \(t\), \(v\) is an unused cluster in the forest, \(T = |s| + |t| + |v|\), and \(|s|\) is the cardinality of its argument.

Methods “centroid”, “median” and “ward” are correctly defined only if Euclidean pairwise metric is used. When analysing rupture characteristics (for example the position on the Kaverina diagram) the choices of “centroid” or “median” linkage methods are reasonable. When analysing the spatial position of the events, for example in a seismic series, maybe the “single” or “complete” linkage methods are better suited.

- **-ce** Metric used to measure distances between events parameters.\(^2\) These metrics work with non-Boolean vectors.

By default FMC uses Euclidean distance, which is a reasonable choice when working with numerical values on the same units. For example, when using only the epicentral position of the events, or the position of the events in the Kaverina diagram (rupture type) or any other clustering approximation using a limited number of parameters in a common physical magnitude and unit. When performing more complex clustering analysis, with parameters such as slip direction, epicentral position and magnitude, the Euclidean metric will produce anomalous results, as the parameters are in different units and magnitudes. To avoid this problem, the different parameters can be normalized so they can be compared. The Mahalanobis metric, for example, normalizes each parameter with its covariance matrix, so they can be used together.

When analysing seismotectonic data in areas, a useful approximation is to split the clustering into several steps, for example, performing first a rupture-type clustering and then spatial clustering or vice versa, rather than mixing all the parameters in one analysis.

- **braycurtis** The Bray–Curtis distance between two points \(u\) and \(v\) is
  \[d(u, v) = \frac{\sum |u_i - v_i|}{\sum |u_i + v_i|}\]

- **canberra** The Canberra distance between two points \(u\) and \(v\) is
  \[d(u, v) = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|}\]

- **chebyshev** The Chebyshev distance between two \(n\)-vectors \(u\) and \(v\) is the maximum norm-1 distance between their respective elements. More precisely, the distance is given by
  \[d(u, v) = \max_i |u_i - v_i|\]

- **cityblock** City block or Manhattan distance between the points.

- **correlation** Correlation distance between vectors \(u\) and \(v\). This is
  \[1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{\|u - \bar{u}\|_2 \|v - \bar{v}\|_2}\]

- **cosine** Cosine distance between vectors \(u\) and \(v\),
  \[1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2}\]

- **euclidean** Distance between \(m\) points using Euclidean distance (2-norm). [Default]

- **hamming** Normalized Hamming distance, or the proportion of those vector elements between two \(n\)-vectors \(u\) and \(v\) which disagree.

- **jaccard** Jaccard distance between the points. Given two vectors, \(u\) and \(v\), the Jaccard distance is the proportion of those elements \(u[i]\) and \(v[i]\) that disagree.

---

\(^1\) The details on the clustering algorithms shown in this section are taken from the Scipy documentation.

\(^2\) The details on the clustering algorithms shown in this section are taken from the Scipy documentation.
3. Examples of use

- Obtaining nodal planes from moment tensor

  Command:
  
  ```
  echo -2.54 37.09 12 3.4669 -2.0652 5.5321 6.2368 -1.8004 -5.1775 22 X Y ID | FMC.py -i AR -o CMT
  ```

  Result:
  
  ```
  #Longitude Latitude Slip_trend_A Slip_plunge_A mrf mtf mtt mff mrt Exponent_(dyn-cm) X_position(GMT) Y_position(GMT) ID Rupture_type
  -2.54 37.09 12.0 -13.9911 100.925 -47.5101 2.54 37.09 206.709 -13.9911 100.925 -47.5101
  ```

- Obtaining moment tensor from one nodal plane

  Command:
  
  ```
  echo -2.54 37.09 12 190.925 42.4899 206.709 4.6 X Y ID | FMC.py -i AR -o CMT
  ```

  Result:
  
  ```
  #Longitude Latitude Depth_(km) mrr mtt mff mrt mrf mtf Exponent_(dyn-cm) X_position(GMT) Y_position(GMT) ID Rupture_type
  -2.54 37.09 12.0 -3.56563 -2.21928 5.78491 6.70126 -1.61249 -5.19047 22.0 X Y ID N-SS
  ```

- Obtaining all the parameters from CMT input file and storing in an ASCII file

  Command:
  
  ```
  FMC.py -o CUSTOM -of lon,lat,slipA,plungA,slipB,plungB japan_CMT.dat
  ```

- Using CUSTOM output to obtain event location and slip vector of both nodal planes

  Command:
  
  ```
  FMC.py -p 'Japan 2011 clusters.png' japan_CMT.dat -cn 0
  ```

- Obtaining moment tensor from one nodal plane

  Command:
  
  ```
  echo -2.54 37.09 12 190.925 42.4899 206.709 4.6 X Y ID | FMC.py -i AR -o CMT
  ```

  Result:
  
  ```
  #Longitude Latitude Depth_(km) mrr mtt mff mrt mrf mtf Exponent_(dyn-cm) X_position(GMT) Y_position(GMT) ID Rupture_type
  -2.54 37.09 12.0 -3.56563 -2.21928 5.78491 6.70126 -1.61249 -5.19047 22.0 X Y ID N-SS
  ```

4. Impact and conclusions

The main research question addressed by FMC is the improvement of the seismotectonic analysis of regions and seismic series. FMC is a powerful tool that allows the user to obtain a deeper insight into the processes responsible for the seismicity, be it natural or human-induced.

With the use of FMC, the user can use a straightforward method to obtain parameters related to the earthquake focal mechanism, especially the double-couple, and can also produce diagrams allowing easy visualization of the DC earthquake rupture mechanisms. The hierarchical clustering analysis is an approximation that is not frequently used in seismotectonics due to its complexity. Its implementation in FMC provides the seismology and seismotectonics communities with a user-friendly tool.

In general, FMC facilitates management of focal mechanism parameters and the implementation of new research approximations to seismicity analysis. To date (see the list of published works in Section 1), the use of the previous version of FMC has improved the quality and clarity of seismotectonic data representation as well as the variety of analyses that can be performed with focal mechanisms, and has provided a basic tool to improve our understanding of the details of tectonic and seismic processes. With the recently implemented clustering algorithm in FMC, seismotectonic analysis will gain a completely new perspective.

FMC is being used by the seismotectonic community, formed by geophysicists, seismologists, earthquake geologists and structural geologists. It has been used in academic research, as well as...
in seismic and tsunami risk consultancy. It improves the quality of the analysis performed and reduces the time spent on data processing and management.

FMC is growing as a versatile tool that can be implemented easily on automated scripts of data analysis and representation in conjunction with other software suites such as GMT [69] that lack similar tools.

Acknowledgements

I would like to thank Jorge L. Giner Robles, Alberto Jiménez Díaz, Julissa Sanjur, Junqing Liu, Lester Anderson and Andrei Bala, who have tested different versions of the software and suggested improvements for the code. Two anonymous reviewers, and especially R. Myhill, made suggestions to improve the program and the manuscript. This work has been partially supported by the project QuakeStep (CGL2017-83931-C3-1-P) funded by the
Fig. 3. Plot result from the command "FMC.py -p 'Japan 2011 clusters.png' japan_CMT_2011.dat -cn 0".

Fig. 4. Plotting with GMT psmeca of the clusters obtained with the command "FMC.py -p 'Japan 2011 spatial clusters.png' japan_CMT_2011.dat -cn 4 -ci lon,lat".

Ministry of Science, Innovation and Universities of the Spanish Government, Spain.

References


