

DiscontClust: open program for automatic grouping of discontinuity orientations data

Ludger O. Suarez-Burgoa

Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia. losuarezb@unal.edu.co

Received: November 6th, 2018. Received in revised form: May 1st, 2019. Accepted: June 18th, 2019.

Abstract

This article describes the algorithm of the spectral method that was chosen to create an applied computational program, which performs an automatic clustering of oriented data of discontinuities-measurements in a rock mass. The clustering differentiation of discontinuity families normally is done by a heuristic selection of sets based on a density diagram of discontinuities. But, an automatic clustering based on a statistical algorithm may be faster and passive to be replicated. In addition, an automatic clustering that has been presented as an open program is useful for the implementation in other calculi in the process of data analysis of discontinuity orientations. The article explains in detail the algorithms and its implementation; it has validations and examples that guarantees that the program can be used.

Keywords: discontinuities; rock mass; automatic clustering; open code; Octave.

DiscontClust: programa abierto para el agrupamiento automático de datos de orientaciones de discontinuidades

Resumen

El presente artículo describe el algoritmo del método espectral que fue elegido para crear un programa computacional aplicado, este que hace el agrupamiento automático de datos de orientaciones de discontinuidades medidas en un macizo rocosos. La agrupación y diferenciación de familias de discontinuidades normalmente se hace a través de la selección heurística de grupos a partir de un diagrama de densidades de discontinuidades. Sin embargo, una agrupación automática y basada en un algoritmo estadístico puede ser más rápido y pasible a la réplica. Así mismo, un agrupamiento automático presentado en un programa abierto es útil para la implementación en otros cálculos dentro del análisis de datos de orientaciones de discontinuidades. El artículo explica en detalle del algoritmo y su implementación, tiene validaciones y ejemplos que garantizan que el programa puede ser usado.

Palabras clave: discontinuidades; macizos rocosos; agrupamiento automático; código abierto; Octave.

1. Introducción

No todas las discontinuidades en un macizo rocoso tienen un mismo origen en su formación (*e.g.* en distinto tiempo pueden existir diferentes estados de esfuerzos en el macizo rocoso); pero si fuera el caso, una sola causa que genera discontinuidades en el macizo rocoso puede crear más de un conjunto disjunto (*e.g.* diaclasas, fracturas de corte, de tracción y de compresión). Por tanto, un macizo rocoso puede tener más de una familia de discontinuidades.

En un macizo de un solo tipo de material rocoso, las

propiedades geométricas de cada una de las familias de discontinuidades (como la ondulación, la rugosidad, el espesor) por lo general son distintas. Una de las propiedades más influyentes para diferenciar cada una de las familias de discontinuidades es sin duda la *orientación espacial* de esta; por tanto, en la práctica se usa esta variable como la principal para diferenciar entre familias.

De este modo, se puede usar métodos de diferenciación de familias (métodos de agrupamiento de datos [en inglés *Clustering methods*]) para encontrar de forma automática las orientaciones más esperadas de un conjunto de datos que son

de un mismo macizo rocoso. Sin embargo, no es tan fácil porque la orientación espacial de un plano se la determina no por un escalar sino por un vector de 3×1 . Aun así, existen métodos para hacer este cálculo de diferenciación de familias por agrupamiento estadístico y además de forma automática. En este artículo se explica un algoritmo que posibilita el agrupamiento automático de orientaciones que se ha implementado en un software de aplicación denominado DiscontClust.

2. Método de agrupamiento de orientación de planos

El método que se usó en el desarrollo del software a aplicación para el agrupamiento automático de datos de orientaciones de planos es el *método espectral* presentado para el caso de discontinuidades en macizos rocosos por Jimenez-Rodriguez and Sitar (2006) [1].

Se toma en cuenta un conjunto de datos con N medidas de la orientación de las discontinuidades en la roca con el fin de agrupar en K familias (el número de familias la define uno mismo, debido a que el método no le dice a uno cuántas familias son las óptimas).

El algoritmo es el siguiente.

1. Calcule la *matriz de afinidad* $A \in \mathbb{R}^{N \times N}$ sabiendo que cada elemento de esa matriz está dada por

$$A_{ij} = \exp\left(\frac{-d^2(x_i, x_j)}{2\sigma^2}\right), \quad \text{si } i \neq j \quad (1)$$

y

$$A_{ij} = 0, \text{ si } i = j; \quad (2)$$

donde el operador $d^2(x_i, x_j)$ se define como la medida de similitud (la distancia) basada en la función seno de los vectores unitarios x_i y x_j ; es decir

$$d^2(x_i, x_j) = 1 - (x_i \cdot x_j)^2. \quad (3)$$

El escalar σ^2 controla cómo la medida de similitud entre dos medidas decae con la distancia entre estas dos orientaciones observadas; mientras mayor es el valor de σ menor es el decaimiento de afinidad cuando la distancia incrementa. El valor de σ^2 se recomienda se elija dentro del intervalo $[0.1^2, 0.15^2]$ para tener un valor adecuado de la agrupación. Jimenez-Rodriguez and Sitar (2006) [1] recomiendan un valor de $\sigma^2 = 0.12^2$.

2. Defina D , la matriz de afinidad, como la matriz diagonal de elementos (i, i) que se calcula como la suma de la i ésima fila de A . Aquí, cada elemento D_{ii} es la suma de las afinidades de la observación i respecto las demás observaciones en el conjunto total de datos.
3. Calcule la matriz L (llamada la matriz de afinidad normalizada) dada por

$$L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}. \quad (4)$$

4. Calcule los primeros K mayores autovalores de L y sus correspondientes autovectores v_1, v_2, \dots, v_K .
5. Cree la matriz V tras agrupar los K autovectores

$$V = [v_1, v_2, \dots, v_K]. \quad (5)$$

6. Forme la matriz U a partir de V tras normalizar esta última en cada fila, de tal forma que cada fila tenga una longitud unitaria; esto es

$$U_{ij} = \frac{v_{ij}}{(\sum_j v_{ij}^2)^{\frac{1}{2}}}. \quad (6)$$

7. Tomando en cuenta que cada fila de U representa un punto en el espacio \mathbb{R}^K , realice la agrupación de cada fila de U en K subconjuntos, esto usando el algoritmo de *K-means*.
8. Asigne el punto original x_i a un grupo j si y solo si la fila i de la matriz U se ha asignado al grupo j en el anterior paso.

El algoritmo de arriba realiza una transformación de las N observaciones de las orientaciones de las discontinuidades desde el espacio original cartesiano en \mathbb{R}^3 al nuevo espacio en \mathbb{R}^K . Las coordenadas de los puntos en el espacio transformado se dan por la normalización de las filas de aquella matriz que se obtiene de los autovectores provenientes de la descomposición espectral de la matriz de afinidad de las observaciones iniciales.

3. Programa computacional aplicado DiscontClust

El programa computacional aplicado DiscontClust fue codificado en el lenguaje intérprete **GNU Octave** para su versión igual o mayor a la 3.8.0; *i.e.* Octave ($\geq 3.8.0$). El lenguaje GNU Octave es un lenguaje que ha nacido con el fin de ofrecer al público la versión libre y no comercial del lenguaje **MATLAB**. En un principio (en la década de los 90 del siglo pasado y hasta mediados del primer decenio de este siglo), los dos lenguajes eran muy similares; sin embargo, hoy en día MATLAB ha reestructurado sus funciones de tal modo de posibilitar y ofrecer a sus usuarios el concepto de programación orientada a objetos y además MATLAB ha ido desarrollado una serie de paquetes completos para cada especialización de las ciencias y las disciplinas de la ingeniería. Hoy en día, MATLAB y Octave se los puede considerar como lenguajes distintos, con sus propios alcances y reglas, aunque con una sintaxis similar.

El núcleo principal del programa DiscontClust (*i.e.* mainToolBox) está compuesto por diez funciones procedimentales, de las cuales seis se usan para presentar los datos (*data presentation functions*), una para preparar los datos antes del análisis (*preanalysis function*), dos funciones que realizan tareas independientes (*auxiliar functions*) al ser llamado por una sola función principal (*main function*) que realiza el cálculo de agrupación; tal como se muestra en el listado de las funciones del programa en la Figura 1.

```

. DiscontClusy program
. Auxiliar functions
├── centerRows.m
├── sinebasesimdistance.m
. Data presentation functions
├── plot2dudata.m
├── plot3dudata.m
├── plotclusterssphericalproy.m
├── plotndudatasva.m
├── plotudata.m
├── plotunclusterssphericalproy.m
. Preanalysis function
├── sortem.m
. Main function
├── spectralclustering.m

```

Figura 1. Núcleo principal del programa DiscontClust.
Fuente: El Autor.

Estas diez funciones se alojan en un directorio para poder ser usado con la combinación de otros programas que requieran hacer esta operación. Las funciones se pueden hacer correr desde la consola de GNU Octave o desde otros programas.

Para tener mayores capacidades de preanálisis con datos de orientaciones espaciales de discontinuidades y para también tener mayores capacidades gráficas en el postanálisis usted debe usar la librería BuzyPlus que puede descargarlo de este enlace de GitHub (librería también desarrollado por el presente autor).

Esta librería es aparte a este desarrollo, pero unos cuantos archivos principales (no todos) se coloca dentro del directorio del presente desarrollo DiscontClust con el fin de tener las mínimas capacidades gráficas en los ejemplos y validaciones de esta herramienta.

De este modo, el programa DiscontClust tiene en su propio directorio las siguientes nueve funciones provenientes del programa BuzyPlus (Figura 2).

Para facilidad del usuario de DiscontClust todos estos archivos se han puesto en un *paquete* de instalación (llamado en Octave *package* y conocido con la sigla *pkg*).

```

. BuzyPlus program
├── createplungevec.m
├── createtrendvec.m
├── dipdirdip2pole.m
├── equalareapolar2planepolar.m
├── plotgreatcircnorth.m
├── plotplaneorientationdata.m
├── reduceminimal360angle.m
├── reduceminimal90angle.m
├── trendplunge2unitvect.m

```

Figura 2. Programa BuzyPlus.
Fuente: El Autor.

```

1; % This in order to be recognized as a script.

## Installing the package.
pkg install discontclust-0.1.0.tar.gz

## Loading the packages.
pkg load io
pkg load image
pkg load statistics
pkg load discontclust

```

Figura 3. Implementación de DiscontClust
Fuente: El Autor.

Este paquete se instala en la consola de Octave mediante el administrador de paquetes del mismo nombre; que para el caso presente se instalaría del siguiente modo

```
pkg install discontclust-0.1.0.tar.gz
```

El paquete tiene dependencias de tres librerías básicas de Octave, estas son *io* ($\geq 2.4.5$), *image* ($\geq 2.6.1$), y *statistics* ($\geq 1.3.0$).

Una vez instalado en su máquina, para su uso, el usuario debe cargarlo mediante la siguiente oración

```
pkg load discontclust
```

De este modo, si las tres librerías básicas están instaladas, el encabezado normal cuando se quiera hacer correr un archivo de lotes con la implementación de DiscontClust sería como se muestra en la Figura 3.

4. Funcionamiento del programa

El programa de aplicación DiscontClust necesita un archivo de texto como entrada donde deben estar las medidas de orientaciones de planos de discontinuidades que se van a someter al algoritmo de búsqueda automática.

En este archivo de texto plano (*plain text*), cada fila corresponde a la información de la orientación de una discontinuidad; de tal modo que si se tiene *m* datos se tiene *m* filas en el archivo. Cada fila está compuesta por dos números separados por un espacio; el primer número indica siempre la dirección de buzamiento de la discontinuidad (dado en grados sexagesimales) y el segundo número indica el buzamiento de la misma, también en grados sexagesimales. El siguiente listado muestra los primeros dos y finales dos datos de las orientaciones de la primera validación que se explica en la siguiente sección.

```

48 7
46 0
: :
277 58
278 55

```

Esto indica que se tiene discontinuidades con

orientaciones $\{(048,07), (046,00) \dots, (277,58), (278,55)\}$ dados en un par ordenado de dirección de buzamiento y buzamiento.

También el programa necesita de forma opcional que se introduzca el tipo de proyección esférica en el cual se quiere representar los datos agrupados e introducir el valor de la variable sigma. En caso que no se coloquen estos, los valores por defecto son a una proyección esférica equiárea y un valor de $\sigma = 0.12$. En el paquete mínimo incluido en DiscontClust solo acepta la proyección más usada, la proyección esférica equiárea; si quiere usar la equiángulo debe instalar el paquete BuzyPlus.

Las variables de salida son tres: knTParrayCell, lCell y knNormEigVecCell, pero la más importante es la primera variable knTParrayCell; las dos restantes son opcionales y sirven para hacer las gráficas de los puntos en el espacio \mathbb{R}^K .

La estructura de la variable de salida knTParrayCell es una lista (en Octave se llama una célula, i.e. Cell) de dimensión $1 \times K$ donde en cada elemento se almacena un arreglo de dimensión $m_k \times 2$, siendo m_k el número de datos perteneciente a cada familia. La primera columna del arreglo indica la dirección de buzamiento de la discontinuidad y la segunda columna indica el buzamiento. El siguiente listado muestra los primeros dos y últimos dos datos de las dos familias del anterior ejemplo.

```
{
  [1,1] =
    48   7
    46   0
    :   :
   211   1
    32   0
  [1,2] =
   105  55
    79  58
    :   :
   277  58
   278  55
}
```

El listado indica que en la primera familia se tiene agrupadas las orientaciones que tienen una orientación de $\{(048,07), (046,00) \dots, (211,01), (032,00)\}$ y en la segunda familia las orientaciones de $\{(105,55), (079,58) \dots, (277,58), (278,55)\}$. De esto, el dato (277,58) indica una discontinuidad con una orientación dada por una dirección de buzamiento de 277° y un buzamiento de 58° por ejemplo.

Esta variable se puede usar dentro del mismo Octave para posteriores desarrollos o se puede exportar a un archivo de texto; libertad de manipulación que tiene el usuario al ser un desarrollador en el lenguaje Octave.

La Fig. 4 muestra las variables de entrada y salida que entran en el algoritmo spectralclustering.

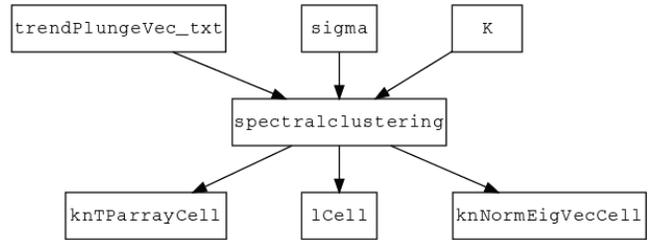


Figura 4. Variables de entrada y salida del algoritmo spectralclustering. Fuente: El Autor.

5. Validaciones

En esta sección se muestra las validaciones de programa DiscontClust tomando los ejemplos que están en el artículo de publicado de Jimenez-Rodriguez and Sitar (2006) [1] donde presentan este método.

5.1. Validación 1: macizo rocoso Sitio c1904

El material de esta validación la puede descargar de [este enlace](#).

Se digitalizaron los 327327 datos de la Fig. 3c del artículo de Jimenez-Rodriguez and Sitar (2006) [1]. La digitalización se hizo con el programa Geonetdigitizer (Suarez-Burgoa, 2015 [2]) y los datos se almacenaron en un archivo de texto llamado validation1data.txt (que puede [descargarlo aquí](#)).

Los datos de esa figura tomada como ejemplo para la validación de este programa corresponden a unos datos reproducidos por Herda et al., (1991) [3] catalogado como Sitio c1904 y corresponde a medidas de fracturas en un macizo rocoso.

Inicialmente, los datos digitalizados no están diferenciados, solo después de la aplicación del programa se obtiene el agrupamiento por familias de forma automática. Para ello, se hace correr el un archivo de lotes configurado para realizar la tarea llamando a las diversas funciones que componen el programa DiscontClust. Esta codificación es guardada en el archivo de lotes validation1dataSCR.m.

El siguiente listado muestra el contenido del archivo de lotes validation1dataSCR.

```
1; % This in order to be recognized as a script.
## This is the 'spectralClusteringSCR' script. It performs a discontinuity
## clustering in two groups of data loaded in the file 'validation1data.txt'
## file.
## Installing the 'DiscontClust' package.
pkg install discontclust-0.1.0.tar.gz
## Loading the packages.
pkg load io
pkg load image
pkg load statistics
pkg load discontclust
## General plot parameters.
projectionType = 'equalarea';
symbolSize = 5;
## Import data from 'txt' file.
trendPlungeArray = load('-ascii', 'validation1data.txt');
## Performing the spectral clustering.
K = 2; % Dimension.
sigma = 0.12; % Scaling parameter.
[knTParrayCell, lCell, knNormEigVecCell] = spectralclustering(trendPlungeArray,
K, sigma);
## Plotting the clustered data in the spherical projection.
plotclusterssphericalproy(knTParrayCell);
```

Figura 5. Archivo de lotes validation1dataSCR. Fuente: El Autor.

En las siguientes líneas se explica qué es lo que ordena a Octave el contenido del archivo de lotes mostrado arriba.

1. La primera línea indica al intérprete de Octave que es un archivo de lotes y no una función.
2. La segunda línea efectiva (efectiva, que no tiene comentario) instala el paquete *discontclust-0.1.0.tar.gz*.
3. Las siguientes tres líneas efectivas del código (*efectiva* que no tiene comentario) cargan librerías básicas de Octave. Respectivamente importa la librería *io* que es la librería para manipular los comandos de entrada y salida de datos; luego importa la librería *image* para graficar; y finalmente la librería *statistics* que tiene la función de K-means.
4. Inmediatamente después se carga la librería *discontclust*.
5. Después del comentario *General plot parameters* está claro que las dos siguientes líneas indican la proyección esférica con la que se graficarán los puntos (que representan las orientaciones) y el tamaño del símbolo de los puntos.
6. Del mismo modo, después del comentario *Import data from 'txt' file*, las siguientes líneas definen el nombre del archivo texto donde están almacenados los datos que lo importa del archivo para almacenarlo en un arreglo en la variable *trendPlungeArray*. Los datos deben estar en el directorio de trabajo de Octave.
7. Con el comentario *Performing the spectral clustering* se inicia el bloque para hacer el proceso de agrupamiento automático. Las dos primeras líneas definen el número de familias que se quiere encontrar con el algoritmo (que en este caso dice que serán dos) y asigna el valor de σ igual a 0.12. La tercera línea (la más importante) llama para hacer correr la función *spectralclustering* dando como variables de entrada: los datos de las orientaciones de las discontinuidades, el valor del número de familias y el valor de sigma previamente asignados. El resultado se almacena en una variable llamada *knTParrayCell*.

8. La última línea efectiva del archivo de lotes hace correr la función *plotclusterssphericalproy* con la celda que almacena todas las orientaciones agrupadas en dos grupos o familias, resultado del proceso de agrupamiento. Esta función (como su nombre lo indica) grafica los resultados en una ventana nueva mediante el programa *gnuplot*.

En la variable de salida *knTParrayCell* se almacena las mismas medidas pero agrupadas en *K* familias.

La Fig. 6 muestra cómo el programa reconoce dos familias de discontinuidades de forma automática, dando un resultado igual al presentado en el artículo original.

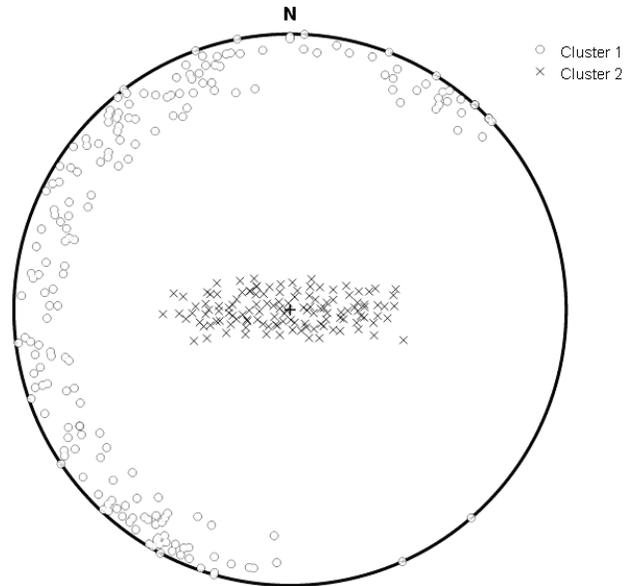


Figura 6. Diagrama esférico equiárea de la orientación de las discontinuidades, donde se diferencian dos familias a partir de la agrupación automática usando el programa *DiscontClust*. Fuente: El Autor.

OctaveOnline

Files

```

octave/discontclust-0.1.0/ dipdirip2pole.m
octave/discontclust-0.1.0/ doc-cache
octave/discontclust-0.1.0/doc/ discontclust.txt
octave/discontclust-0.1.0/doc/ macros.texi
octave/discontclust-0.1.0/ equalareapolar2planepolar.m
octave/discontclust-0.1.0/packinfo/ CITATION
octave/discontclust-0.1.0/packinfo/ COPYING
octave/discontclust-0.1.0/packinfo/ DESCRIPTION
octave/discontclust-0.1.0/packinfo/ INDEX
octave/discontclust-0.1.0/ plot2dudata.m
octave/discontclust-0.1.0/ plot3dudata.m
octave/discontclust-0.1.0/ plotclusterssphericalproy.m
octave/discontclust-0.1.0/ plotgreatcircnorth.m
octave/discontclust-0.1.0/ plotndudatasva.m
octave/discontclust-0.1.0/ plotplaneorientationdata.m
octave/discontclust-0.1.0/ plotudata.m
octave/discontclust-0.1.0/ plotuncclusterssphericalproy.m
octave/discontclust-0.1.0/ reduceminimal360angle.m
octave/discontclust-0.1.0/ reduceminimal90angle.m
octave/discontclust-0.1.0/ sinebasesimdistance.m
octave/discontclust-0.1.0/ sortem.m
octave/discontclust-0.1.0/ spectralclustering.m
octave/discontclust-0.1.0/ trendplunge2unitvect.m
validation1data.txt
validation1dataSCR.m
    
```

Drop Files Here to Upload

Tips and Tricks

The files you make on Octave Online will be saved for the next time you visit.

In the editor, you can use most of these keyboard shortcuts. You can also use these special keyboard shortcuts:

- Ctrl + Space** → Show the auto-completion menu
- Cmd/Ctrl + S** → Save the file
- Cmd/Ctrl/Win + R** → Run the file
- Cmd/Ctrl/Win + E** → Set focus to the prompt

Vars

```

# K
# ans
{1x2} knNormEigVecCell
{1x2} knTParrayCell
{1x2} lCell
(abc) projectionType
# sigma
# symbolSize
[326x2] trendPlungeArray
    
```

cwd: /home/oa

Plot: Line 2

ludgerito@gmail.com Sign Out

Sharing is currently disabled. If you enable it, you get a link that you can use for real-time collaboration with colleagues.

- Enable Sharing
- Change Password
- Feedback and Support
- Privacy Policy and EULA
- GitHub Project (AGPL)
- Change Theme
- Change/Reset Layout
- Follow on Twitter

Render New Plots Inline

Icons made by Boadan Rosu, Daniel Bruce, Sarfraz Shoukat, Icomoon, Freepik, Icon Works, Platinicon, and Simpleicon are from www.flaticon.com and licensed by CC BY 3.0

run validation1dataSCR

Figura 7. El entorno gráfico del programa Octave-Online luego de haber hecho correr el archivo de lotes de esta validación. Fuente: El Autor.

Esta validación (y cualquiera de las siguientes, más los ejemplos) se puede hacer correr en la nube ingresando a un desarrollo en línea de Octave en la página web <https://octave-online.net/>. Para ello debe crear una cuenta personal, dentro de ella subir los archivos *discontclust-0.1.0.tar.gz*, *validation1data.txt* y *validation1dataSCR.m* para finalmente hacer correr el archivo de lotes escribiendo en la terminal simplemente `run validation1dataSCR`. La Fig. 7 muestra el entorno gráfico de esta aplicación en la nube, donde se hizo correr el archivo de lotes de esta validación.

Se observa en la Fig. 7 que la ventana del entorno gráfico se divide en ocho puertos de vista. En la parte superior (en todo el ancho) está una franja con el nombre de la aplicación y en la parte derecha la opción del MENU que activa la vista larga vertical de la derecha, donde se observa el poseedor de la cuenta y una serie de opciones donde se puede compartir el desarrollo, cambiar la clave, ayudas, etc. Luego en la vista larga vertical de la parte izquierda se tiene información de los archivos que están en el directorio en la nube en la cuenta del usuario, se puede apreciar claramente el paquete con todos sus archivos y los dos archivos de extensión *.txt* y *.m*. En la parte central la vista se divide en cinco, en la parte izquierda se tiene algunas ayudas rápidas, mientras que en el resto se tiene las cuatro vistas, en la parte inferior derecha la línea de comando donde se ve `run validation1dataSCR`, por encima de este se muestra (sea el caso) el historial de la línea de comando. En la parte izquierda se tiene la lista variables que están en memoria, como ser *K*, *ans*, *knNormEigVecCell*, *lCell*, *projectionType*, *sigma*, *symbolSize* y *trendPlunge Array*.

Finalmente, en la parte superior se tiene la ventana donde aparece los resultados gráficos, que en este caso es el diagrama en proyección esférica con los resultados de las familias agrupadas, este gráfico se puede exportar en una imagen de extensión *.png* o un dibujo vectorial de extensión *.svg*.

¡Con Octave-OnLine no necesita instalar Octave en su máquina!

5.2. Validación 2: macizo rocoso mina San Manuel

El material de esta validación la puede descargar de [este enlace](#).

Para esta segunda validación, se digitalizó 283 puntos de la Fig. 4d del mismo artículo de Jimenez-Rodriguez and Sitar (2006) [1]. También, la digitalización se hizo con el programa Geonetdigitizer (Suarez-Burgoa, 2015 [2]) y los datos se almacenaron en un archivo de texto llamado *validation2data.txt*.

Los datos así digitalizados pertenecen a la toma de datos manual de un afloramiento de la mina de cobre San Manuel en Arizona EE.UU, reportado en Klose et al. (2005). [4]

Esta vez, se configuró el archivo de lotes de tal forma que con esos datos se encuentren tres familias.

La Fig. 8 muestra cómo el programa reconoce esas tres familias de discontinuidades de forma automática, dando

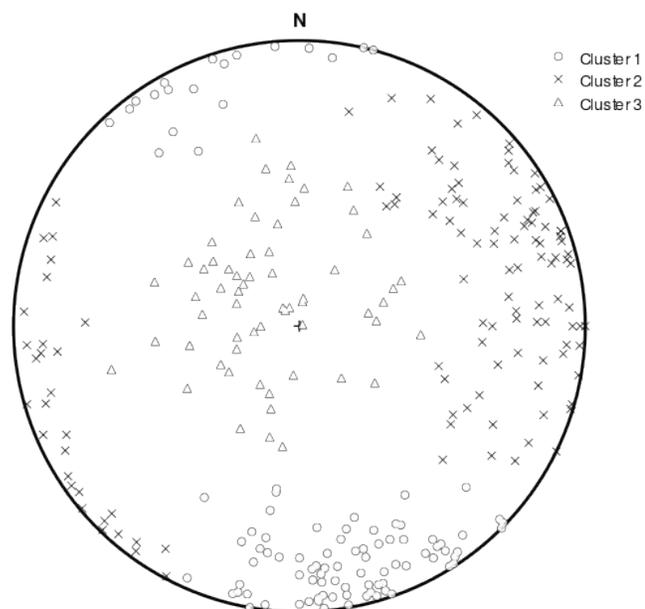


Figura 8. Diagrama esférico equiárea de la orientación de las discontinuidades, donde se diferencian tres familias a partir de la agrupación automática usando el programa *DiscontClust*.

Fuente: El Autor.

también un resultado igual al presentado en el artículo original.

5.3. Validación 2: macizo rocoso de serpentinita

El material de esta validación la puede descargar de [este enlace](#).

En esta tercera y última validación se digitalizó 174 datos que corresponden a medidas manuales (por el método de a línea de rastreo) de orientaciones de discontinuidades en un afloramiento de serpentinita al sur de España Jimenez-Rodriguez and Sitar (2006). [1]

En este caso, se hizo mayores procedimientos de análisis debido a que el artículo con el que se está validando el funcionamiento del programa *DiscontClust* obtiene para los mismos datos: dos y tres posibles familias de agrupación; asimismo, los autores grafican los datos en el espacio transformado \mathbb{R}^K para observar que las agrupaciones son correctas. Los mismos gráficos son producidos con el programa que se ha elaborado, con *DiscontClust*.

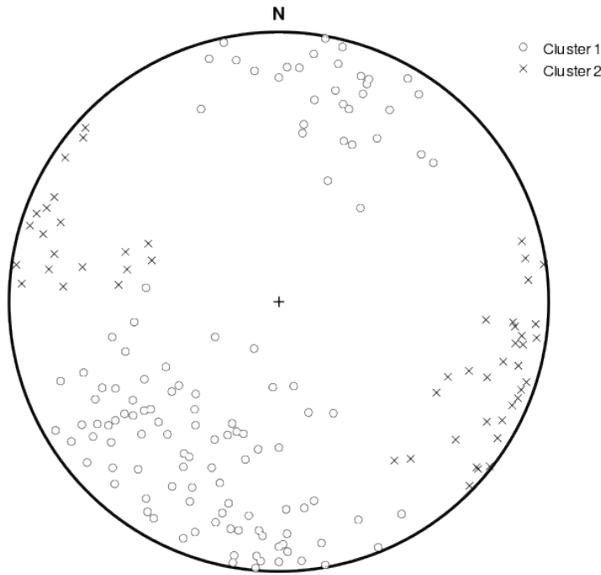
En el análisis, se obliga al programa que realice con los mismos datos tres análisis, un primer análisis en el que encuentre dos familias de discontinuidades, un segundo análisis que encuentre tres familias, y un tercer análisis que encuentre más de tres familias. Para los dos primeros casos el programa *DiscontClust* tiene sus representaciones gráficas en \mathbb{R}^K , pero para una dimensión de $K > 3$ no se tiene una representación gráfica por no poderse representar el espacio \mathbb{R}^K en un diagrama cartesiano ortogonal.

Del mismo modo al caso de los dos anteriores ejemplos,

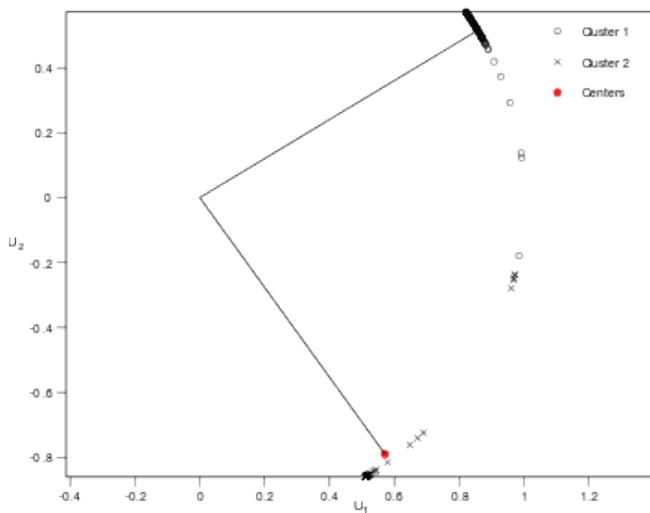
se observa que el programa *DiscontClust* ha respondido de forma exacta los resultados de la investigación con la que se está validando. La Fig. 9 muestra las representaciones gráficas en el diagrama estereográfico del proceso de agrupación a dos familias más su respectiva gráfica en el espacio \mathbb{R}^K , para $K = 2$.

Del mismo modo que el caso anterior, la Fig. 10 muestra las representaciones gráficas en el diagrama estereográfico del proceso de agrupación ahora de tres familias más su respectiva gráfica en el espacio \mathbb{R}^K , para $K = 3$.

La Fig. 11 muestra el diagrama esférico equiárea del proceso de agrupación a cuatro familias de los mismo datos.



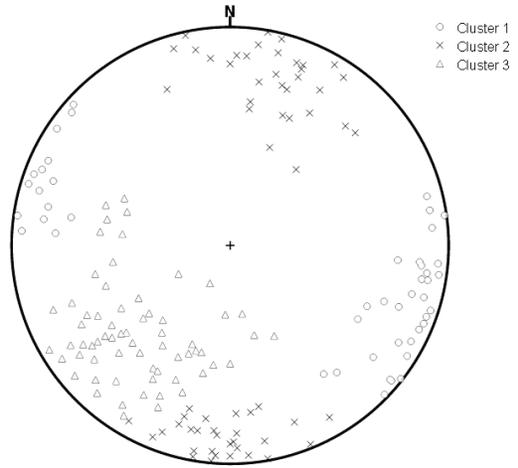
a) Diagrama de proyección esférica equiárea con las dos familias diferenciadas.



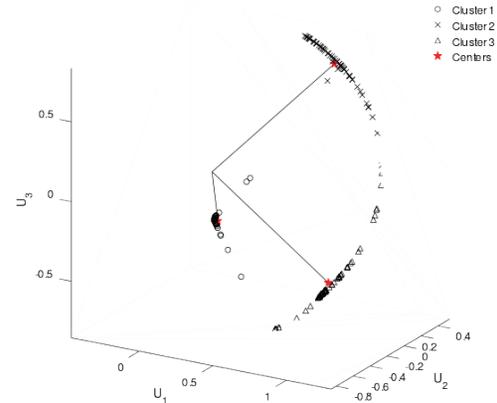
b) Puntos de los datos en el espacio transformado \mathbb{R}^2 concentrados en dos ejes.

Figura 9. Resultados del proceso de agrupación automática de datos de discontinuidades para dos familias.

Fuente: El Autor.



a) Diagrama de proyección esférica equiárea con las tres familias diferenciadas.



b) Puntos de los datos en el espacio transformado \mathbb{R}^3 concentrados en dos ejes.

Figura 10. Resultados del proceso de agrupación automática de datos de discontinuidades para tres familias.

Fuente: El Autor.

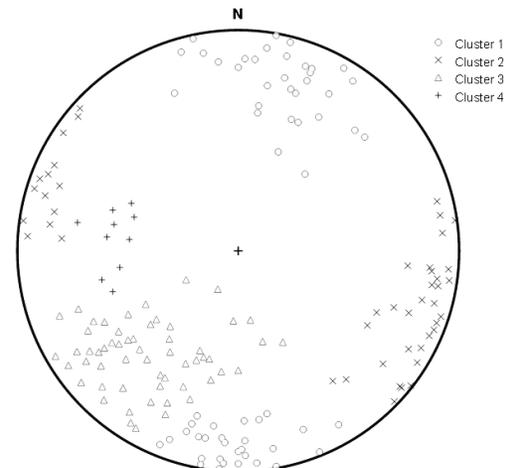


Figura 11. Diagrama esférico equiárea de la orientación de las discontinuidades, donde se diferencian cuatro familias a partir de la agrupación automática usando el programa *DiscontClust*.

Fuente: El Autor.

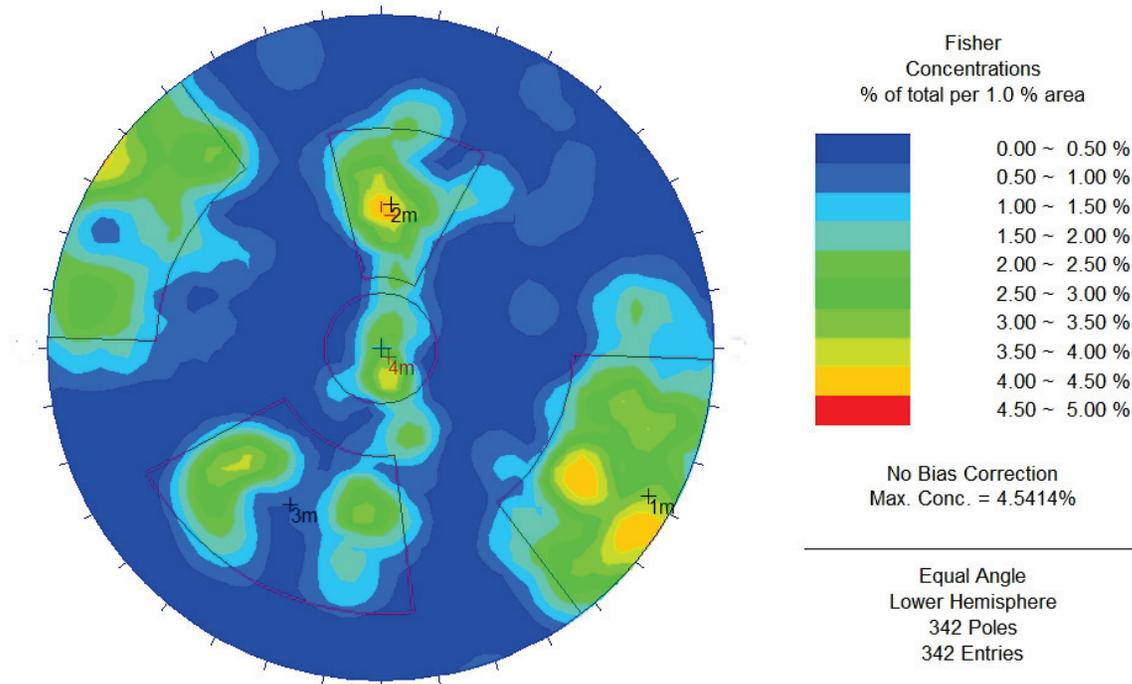


Figura 12. Diagrama esférico equiángulo de la orientación de las discontinuidades de la casa de máquinas de Porce 3, donde se diferencian cuatro familias encontradas de forma heurística manual a partir de un diagrama de concentraciones.
Fuente: Suarez-Burgoa, 2008 [5].

6. Ejemplos

A continuación se presenta dos ejemplos más con datos tomados en macizos rocosos en Colombia.

6.1. Ejemplo 1: macizo rocoso de la casa de máquinas de Porce 3

El material de esta validación la puede descargar de [este enlace](#).

Para este primer ejemplo, se tuvo acceso a la lectura de 342 datos de medidas de orientaciones de discontinuidades en el techo de la casa de máquinas del proyecto hidroeléctrico Porce 3 entre las progresivas $KM0 + 000$ a $KM0 + 120$. Con esos datos, Suarez-Burgoa (2008) [5] elaboró un diagrama de concentraciones de polos (mostrado en proyección equiangular) y a criterio heurístico se encontraron cuatro familias de discontinuidades agrupadas en tres ventanas rectangulares y una ventana circular, tal como se muestra en la Fig. 12. Esta es la práctica habitual en la actualidad por la carencia de una herramienta separada de distribución libre que posibilite la selección automática como la que se expone en este texto.

Con el programa *DiscontClust* se encontraron también cuatro familias, pero dos de ellas se alejan del criterio de selección actual. La Fig. 13 muestra los resultados del proceso de agrupación de familias (encontrada con este programa) puesto encima de la imagen del diagrama de

concentraciones de polos (obtenida con el programa *BuzyPlus*), aquel mencionado en el estudio (esta vez es una proyección equiárea porque el programa *DiscontClust* en su versión reducida limita solo a esta proyección).

Se observa que dos de las familias encontradas con el criterio heurístico manual son parecidas a las familias obtenida por el proceso automático, pero dos de ellas se sobreponen tal como se muestra en la Fig. 14.

Las familias *F2* y *F4* fueron encontradas por el proceso automático y las *G3* y *G4* por el método heurístico manual. Aquí hay una contradicción cuando por ambos criterios se obtienen conjuntos que comparten elementos.

La condición de la elección de familias es que estos sean conjuntos disjuntos (es decir, que no compartan elementos con otros grupos); esto indica a que uno de los métodos no está obteniendo los conjuntos de forma correcta. A la luz de un proceso automático bajo un fundamento matemático-estadístico, es más factible dar crédito al método automático que el método manual. Otro aspecto que uno puede observar, es que la agrupación de polos no necesariamente forma un conjunto escogido por una ventana rectangular o circular.

6.2. Ejemplo 2: macizo rocoso Anticlinal Monteralo

El material de esta validación la puede descargar de [este enlace](#).

Los datos que se analizan en este ejemplo corresponden a los planos de afloramientos todos ellos que pertenecen al

Anticlinal de Monterralo, ubicado en el Piedemonte Llanero en la Cordillera Oriental de Colombia. Los datos fueron digitalizados de la tesis de Sanchez-Villar (2011) [6] mediante el programa *GeonetDigitizer* [Suarez-Burgoa, 2015] [2] y luego fueron analizados con el programa *DiscontClust*.

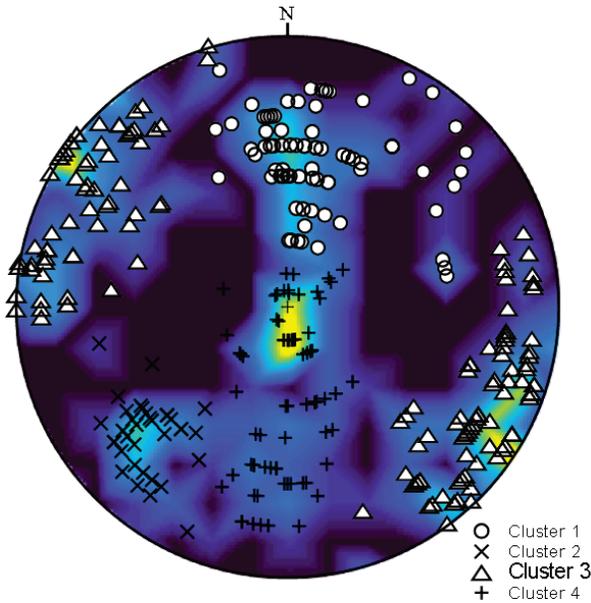


Figura 13. Diagrama esférico equiárea de la orientación de las discontinuidades de la casa de máquinas de Porce 3, donde se diferencian cuatro familias a partir de la agrupación automática usando el programa *DiscontClust*, en el fondo un diagrama de concentraciones logrado con el programa *BuzyPlus*.
Fuente: El Autor.

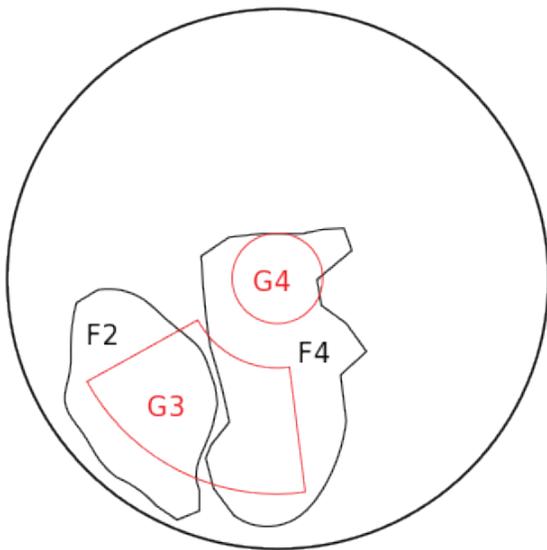


Figura 14. Esquema en proyección esférica equiárea comparando las diferencias de dos familias encontradas a partir de la agrupación automática usando el programa *DiscontClust* y dos familias encontradas de forma heurística manual a partir de un diagrama de concentraciones.
Fuente: El Autor.

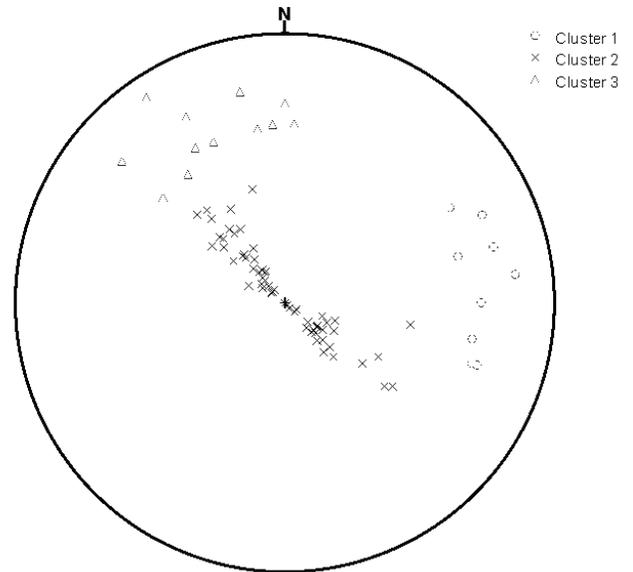


Figura 15. Esquema en proyección esférica equiárea de tres familias encontradas a partir de la agrupación automática usando el programa *DiscontClust* para el Anticlinal Monterralo.
Fuente: El Autor.

La Fig. 15 muestra que el programa encuentra tres familias de discontinuidades y que son coherentes con el concepto de un anticlinal: la primera familia corresponde al flanco sudeste, la tercera familia corresponde al flanco noroeste y la segunda a la parte central del anticlinal. Esto indica que la selección de familias de forma automática es coherente con el concepto de anticlinal para estos datos; y esa agrupación ayudaría a determinar si algún grupo de discontinuidades forman parte de una estructura más compleja, como fue con el presente ejemplo de un anticlinal.

7. Conclusiones

El algoritmo presentado por Jimenez-Rodriguez and Sitar (2006) [1] para la lograr el agrupamiento automático de datos de orientaciones de discontinuidades es uno entre muchos que logra esta tarea (como se detalla en la sección introductoria de Jimenez-Rodriguez and Sitar, 2006); sin embargo, la implementación de este algoritmo (que se ha hecho en este trabajo en el programa *Octave*) hace que el código sea robusto y corto, aumentando la eficiencia en la operación que desempeña.

Un proceso de agrupamiento de orientaciones de discontinuidades es más creíble y más susceptible a replicar si aquella elección se basa en un algoritmo más que aquel agrupamiento logrado por criterios heurísticos que dependen de la experiencia de la persona que analiza y de los resultados de un diagrama de densidades de polos.

El agrupamiento automático acelera el proceso de interpretación de discontinuidades en macizos rocosos si este se implementa en un código de libre uso que puede incluso hacerse correr en la nube.

El algoritmo presentado aquí convertido en un código en

el programa Octave y distribuido de forma libre se puede estudiar, entender, modificar y mejorar por medio del usuario, y además se puede implementar con otros desarrollos de cálculo donde esta operación sea necesaria. Además que se puede traducir a otros lenguajes computacionales a través de la interpretación del código abierto.

8. Descarga del programa

El código del programa DiscontClust está alojado en el sitio anfitrión de proyectos informáticos GitHub, él se descarga del siguiente enlace:

<https://github.com/losuarezburgoa/discontClust>

A través de este sitio se hace las respectivas descargas, aportes y peticiones de participación en el proyecto. Copyright© 2019 en adelante, Universidad Nacional de Colombia y Ludger O. Suárez Burgoa. Este código abierto es software libre: usted puede re-distribuirlo y/o modificarlo bajo los términos de la Licencia BSD-2 o superior vigente (<http://opensource.org/licenses/bsd-license.php>).

El autor se descarga de toda responsabilidad del uso del presente código computacional. Este código se distribuye con la esperanza de que sea útil, pero sin ninguna garantía; sin la garantía implícita en su comercialización o idoneidad para un propósito particular.

Agradecimientos

El autor agradece a la Universidad Nacional de Colombia (sede Medellín) por el apoyo financiero a través del proyecto Hermes bajo el código 36095 de la Convocatoria nacional de proyectos para el fortalecimiento de la investigación, creación e innovación de la Universidad Nacional de Colombia 2016-2018.

Referencias

- [1] Jimenez-Rodriguez, R. and Sitar, N., A spectral method for clustering of rock discontinuities sets. *International Journal of Rock Mechanics and Mining Sciences*, 43, pp. 1052-1061, 2006.
- [2] Suarez-Burgoa, L.O., Geonetdigitizer: código abierto para digitalizar información plasmada en mallas Wulff y Lambert, *Boletín de Ciencias de la Tierra*, 37, pp. 25-34, 2015.
- [3] Herda, H.H.W., Einstein, H.H. and Dershowitz, W.S., Problems with representation of rock-fractures clusters. *Journal of Geotechnical Engineering*, 117(11), pp. 1754-1771, 1991.
- [4] Klose, C., Seo, S. and Obermayer, K., A new clustering approach for partitioning directional data. *International Journal of Rock Mechanics and Mining Sciences and Geomechanical Abstracts*, 42(2), pp. 315-321, 2005.
- [5] Suarez-Burgoa, L.O., Rock mass mechanical behavior assessment at the Porco 3 underground hydropower central, Colombia, South America. MSc. Thesis, Facultad de Minas, Universidad Nacional de Colombia, Medellin, Colombia, 2008.
- [6] Sanchez-Villar, N.F., Desarrollo de patrones de fracturamiento y mecanismos de deformación del anticlinal de Monterralo, Pie de Monte Llanero, Cordillera Oriental de Colombia. Msc. Thesis, Departamento de Geociencias, Universidad Nacional de Colombia, Bogota DC, 2011.

L.O. Suarez-Burgoa, graduado como Ing. Civil de la Universidad Mayor de San Andrés (UMSA), Bolivia, MSc. en Ingeniería Geotécnica de la Universidad Nacional de Colombia, Dr. en Geotecnia de la Universidad de Brasilia (UnB), Brasil. Profesor asociado del Departamento de Ingeniería Civil de la Facultad de Minas de la Universidad Nacional de Colombia, sede Medellín. Desarrolló actividades como consultor e investigador por más de quince años en el campo de la geotecnia civil y minera en Bolivia, Chile, Brasil y Colombia, con énfasis en la mecánica de rocas. Actualmente es desarrollador de códigos computacionales en lenguajes intérpretes, tales como MatLab 11 (Octave 4.X), R y Python3, en temas relacionados a la geología aplicada y geotecnia; habiendo producido herramientas como: PyBimStab, PackCircles4Bims, JelinekStat, Buzy+, pyCSS, VolteoLab, GeonetDigitizer y SVG-stereographicPlot. Es director del semillero de investigación en geología matemática (www.geomecnica.org), parte del grupo de investigación de geotecnia del Departamento de Ingeniería Civil de la Facultad de Minas, Universidad Nacional de Colombia, sede Medellín. ORCID: 0000-0002-9760-0277.



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Medio Ambiente

Oferta de Posgrados

Especialización en Aprovechamiento de
Recursos Hidráulicos
Especialización en Gestión Ambiental
Maestría en Ingeniería Recursos Hidráulicos
Maestría en Medio Ambiente y Desarrollo
Doctorado en Ingeniería - Recursos Hidráulicos
Doctorado Interinstitucional en Ciencias del Mar

Mayor información:

E-mail: acma_med@unal.edu.co
Teléfono: (57-4) 425 5105