

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE GEOCIÊNCIAS

DESENVOLVIMENTO DE PROGRAMA DE ANÁLISE DE DESCONTINUIDADES
PARA GEOLOGIA ESTRUTURAL APLICADA

Luiz Antonio Urtiga e Silva

Orientador: Prof.Dr. Ginaldo Ademar da Cruz Campanha
Co-orientador: Pedro Refinetti Rodrigues Martins

MONOGRAFIA DE TRABALHO DE FORMATURA
(TF-2012 / 29)

SÃO PAULO
2012

Luiz Antonio Urtiga e Silva

Ginaldo Ademir da Cruz Campanha
Orientador

“721”

Abu Musa Jabir Ibn Hayyan (a.k.a. Geber)

Sumário

	Pág.
Agradecimentos	4
Resumo	5
Abstract	6
1-Introdução	7
2-Metas e Objetivos	8
3-Fundamentação Teórica	9
3.1- Representação vetorial de atitudes estruturais	9
3.2- Frequência de descontinuidades	10
3.3- Identificação de famílias	12
3.4- Cálculo da orientação média e da dispersão angular	14
3.5- Cálculo das interseções	15
3.6- Distribuição espacial de frequências	16
4-Materiais e Métodos	17
5-Resultados Obtidos	18
6-Interpretação e discussão dos Resultados	19
6.1- Identificação de famílias	19
6.2- Correção da frequência de descontinuidades	19
6.3- Cálculo das Interseções	20
6.4- Distribuição espacial das frequências	20
6.5- Estatísticas dos dados	20
7-Conclusões	21
8-Referências Bibliográficas	22

Agradecimentos

Agradeço ao Pedro Refinetti Rodrigues Martins por me dar a idéia para o projeto e o auxílio durante a execução, ao Prof. Dr. Ginaldo Ademar da Cruz Campanha por aceitar o projeto de trabalho de formatura e sua colaboração como orientador, aos meus colegas por sempre estarem disponíveis para uma conversa e a minha família por me apoiar nesses 21 anos.

Resumo

Desenvolveu-se rotinas computacionais para análise de descontinuidades, a partir de formulações teóricas existentes na bibliografia especializada, de modo a preencher uma lacuna existente nos pacotes computacionais comerciais disponíveis atualmente. Elaborou-se rotinas na linguagem de programação Python visando a identificação automatizada de famílias, a caracterização de parâmetros estatísticos de orientação das famílias obtidas, o cálculo das intersecções entre famílias e a caracterização da frequência de descontinuidades em três dimensões. Utilizou-se o processo de Poisson para identificar descontinuidades não aleatórias e agrupá-las de acordo com sua proximidade, a estatística de dados direcionais de Fisher para caracterizar a orientação média e dispersão das famílias, o produto vetorial para obter as intersecções entre diferentes famílias de descontinuidades, e a correção direcional da frequência de descontinuidades aplicada a uma malha de contagem para representação no espaço 3D. Utilizou-se conjuntos de dados pré-estabelecidos, com diferentes tipos de distribuição e dispersão, para testar as rotinas desenvolvidas. Estes testes mostraram o desempenho correto das rotinas, destacando-se a forte influência da correção direcional de frequências na distribuição dos dados, e do ângulo de abertura do cone de busca no número de famílias identificadas.

Palavras-chave: Descontinuidades, fraturas, identificação de famílias, estatística, frequência, programação.

Abstract

The project developed computational algorithms for the analysis of discontinuities, from existing theoretical formulations in the literature, in order to fill a gap in the currently available commercial computer packages. Routines in the Python programming language have been elaborated for the automated identification of families, elaboration of statistical parameters characterizing the orientation of the families, calculation of intersections between families and characterization of frequency of discontinuities in three dimensions. The program utilized the Poisson process to identify non-random discontinuities and group them according to their proximity, the Fisher statistics to characterize average orientation and dispersion of families, the cross product to get the intersections between different families of discontinuities, and the correction of directional frequency discontinuities applied to a mesh count for representation in 3D space. We used data sets predetermined, with different distribution and dispersion, to test routines developed. These tests showed the correct performance of routines, highlighting the strong influence of the correction of directional frequency distribution of the data, and the opening angle of the search cone on the number of families identified.

Keywords: Discontinuity, fractures, cluster identification, statistics, frequency, programming.

1- Introdução

Descontinuidade é um termo utilizado em geotecnia para referir-se, de forma genérica, a estruturas planares. O termo não faz distinção à natureza ou processo genético da feição, apesar destes frequentemente controlarem as características das descontinuidades. Fisicamente, descontinuidades representam quebras no meio rochoso contínuo, cuja resistência à tração é nula ou negligenciável. Em função disso, descontinuidades têm enorme influência nas características mecânicas e hidráulicas dos maciços rochosos. Dessa forma, a análise de descontinuidades e respectivas características é uma etapa fundamental para projetos de engenharia de rochas.

Descontinuidades não ocorrem com orientação absolutamente aleatória: por motivos mecânicos, elas ocorrem com algum grau de agrupamento, ao redor de orientações preferenciais associadas aos mecanismos de formação. Um conjunto de descontinuidades paralelas ou subparalelas é denominado de família (Hudson & Harrison, 1997).

Em geotecnia, a maioria dos métodos de análise de estabilidade de taludes e escavações subterrâneas baseia-se no conceito de famílias, frequentemente representando cada família pela sua orientação média.

Dessa forma, a correta identificação de famílias de descontinuidades tem muita relevância para projetos de geotecnia, mas seu método comum de identificação baseia-se em critérios subjetivos, que muitas vezes acabam por subestimar ou sobrestimar determinadas estruturas. Por este motivo, a utilização de métodos estatísticos para a determinação de famílias e seus parâmetros estatísticos é de grande ajuda na melhor compreensão dos dados, especialmente em situações com uma quantidade muito grande de dados com um grande espalhamento. Para tanto o uso de recursos computacionais torna o trabalho rápido e prático, criando assim uma ferramenta extra para a análise de maciços rochosos a custos mais baixos.

2- Metas e objetivos

Este projeto de pesquisa visou desenvolver rotinas computacionais para análise de descontinuidades, a partir de formulações teóricas existentes na bibliografia especializada, de modo a preencher uma lacuna existente nos pacotes computacionais comerciais disponíveis atualmente. Resumidamente, visou-se elaborar rotinas para:

- a) Identificação automatizada de famílias;
- b) Caracterização de parâmetros estatísticos de orientação para as famílias obtidas;
- c) Cálculo das intersecções entre famílias;
- d) Caracterização da frequência de descontinuidades em três dimensões

3- Fundamentação Teórica

3.1- Representação vetorial de atitudes estruturais

O sistema usado normalmente para a representação de planos e linhas em geologia estrutural, baseado no azimute do rumo e no mergulho, apesar de ser mais prático para a obtenção e visualização dos dados, não é o mais adequado em termos de geometria analítica e de programação (Millman, 2008). Para tanto a transformação das atitudes em vetores unitários é a melhor solução.

A orientação de um vetor unitário pode ser definida pelos seus cossenos diretores l , m , n , que correspondem aos cossenos dos ângulos que o vetor faz respectivamente com os eixos de referência x, y, z , conforme ilustrado na Figura 1.

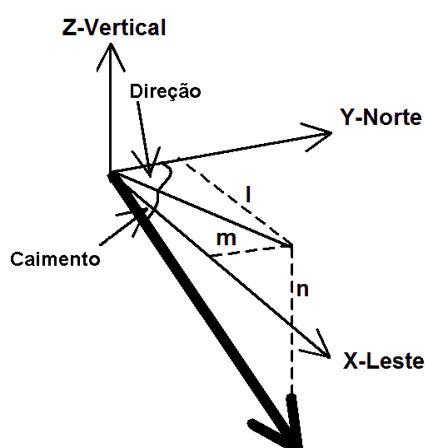


Figura 1 Descrição do sistema adotado de representação vetorial de dados estruturais. (modificado de University of Alberta 2011)

A figura 1 mostra as relações entre a atitude de uma linha, representada pela sua direção ou *trend* (T) e caimento ou *plunge* (P), e os cossenos diretores l , m , n , do vetor unitário equivalente (Campanha & Quintanilha, 1996). A relação entre esses valores é dada por:

$$\begin{aligned} l &= \sin T * \cos P \\ m &= \cos T * \cos P \\ n &= -\sin P \end{aligned} \quad (1)$$

Uma descontinuidade, no entanto, não é uma linha, e sim um plano. Neste caso, para utilizar as equações acima com relação ao pólo de um plano, é necessário determinar a atitude da reta normal a esse plano. Considerando um plano com rumo do mergulho *dir* e mergulho *dip*, sua normal tem *trend* T e *plunge* P segundo:

$$\begin{aligned} dir > 180^\circ : T &= dir - 180^\circ \\ dir < 180^\circ : T &= dir + 180^\circ \\ P &= 90^\circ - dip \end{aligned} \quad (2)$$

3.2- Frequência de descontinuidades

Define-se a *frequência*, simbolizada aqui por λ , como o número de descontinuidades por unidade de comprimento. No campo pode ser obtida através do levantamento de uma linha de varredura (linha de *scan*) ou de uma sondagem, dividindo-se o número de estruturas medidas (N) pelo comprimento da linha (L).

$$\lambda = N/L \quad (3)$$

Numericamente a frequência é o inverso do espaçamento médio (E) entre as descontinuidades.

$$\lambda = 1/E \quad (4)$$

Amostragens lineares de descontinuidades ("linhas de scan", ou linhas de varredura) sempre vão fornecer frequências maiores para descontinuidades ortogonais à linha de amostragem, e para tanto é necessário a aplicação de uma correção geométrica utilizando o ângulo entre a normal à descontinuidade e a linha de amostragem, originalmente conhecida como correção de Terzaghi (Terzaghi, 1965). A figura 2 abaixo ilustra esse conceito.

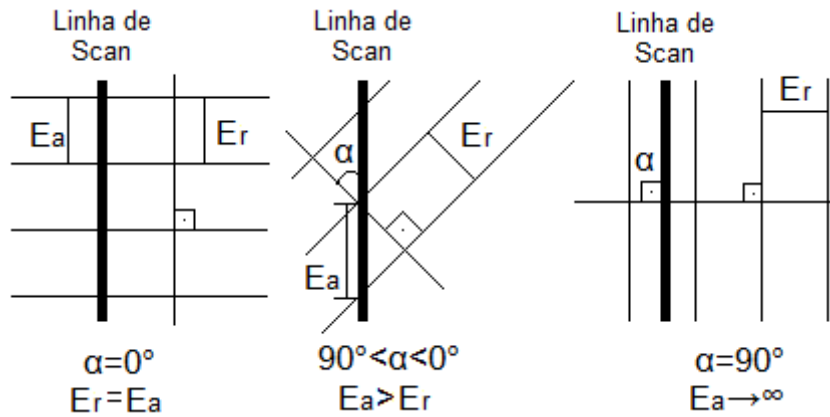


Figura 2 Relações entre direção de uma linha de scan e o espaçamento sua frequência aparente; E_r – espaçamento real, E_a - espaçamento aparente; lembrando que a frequência é o inverso do espaçamento.

Considerando os cossenos diretores l_1, m_1, n_1 do polo de um plano e a linha de scan de cossenos diretores l_2, m_2, n_2 , pode-se obter o ângulo α entre eles através do produto escalar:

$$\cos \alpha = l_1 l_2 + m_1 m_2 + n_1 n_2 \quad (5)$$

Desse modo, a frequência λ_s ao longo de uma linha de scan mostra a seguinte relação com a frequência λ na direção perpendicular à descontinuidade:

$$\lambda_s = \lambda \cos \alpha \quad (6)$$

As frequências obtidas no campo podem ser corrigidas pela multiplicação por um peso w , igual ao inverso do cosseno de α . Consegue-se assim uma função que varia de 1, para um plano perpendicular à linha de *scan* e portanto com amostragem perfeita, à infinito, para um plano paralelo a linha de *scan* que portanto nunca será amostrado.

$$w = 1 / \cos \alpha \quad (7)$$

As contagens de densidades de polos por área do estereograma também podem ser corrigidas da mesma forma.

Um problema nesta abordagem é que pode gerar valores de correção absurdos, tendendo a infinito, para ângulos α muito altos. Dessa maneira é necessário estabelecer um limite para essa correção. Seguindo a recomendação de Priest (1993), o presente trabalho adotou o valor de α máximo = 84,3°. Esse é o valor que permite que o w tenha um valor máximo de 10.

3.3- Identificação de famílias

Em sua publicação clássica de análise de descontinuidades, Priest (1993) indica que, ao se analisar um conjunto de dados de orientação de descontinuidades utilizando-se princípios de estatística baseados na distribuição de Poisson, é possível identificar padrões de agrupamento (*clustering*) nos dados e, conseqüentemente, efetuar a separação estatística de famílias. Na realidade, Priest (1993) reproduz o método desenvolvido por Shanley & Mahtab (1976) e modificado por Mahtab & Yegulalp (1982). Este método consiste em uma análise do universo de dados (orientação das normais dos planos) que, baseada na distribuição de Poisson, identifica os pontos que estatisticamente não têm distribuição aleatória. Depois, compara a proximidade entre os pontos selecionados para agrupá-los em famílias.

Deste modo o algoritmo utilizado baseia-se no processo homogêneo de Poisson. Este é caracterizado por um parâmetro λ , chamado de intensidade, que é o número esperado de eventos ocorridos por unidade de tempo, em uma equação arbitrária qualquer $P(t)$ onde t é uma medida de tempo que varia no campo $(t, t+v)$ e k é diferença entre os resultados delas, de maneira que essa relação é dada por:

$$[P(t + v) - P(t) = k] = \frac{e^{-\lambda v} (\lambda v)^k}{k!} \quad (8)$$

Modificando as variáveis para a utilização em descontinuidades (Priest, 1993), o intervalo de tempo é substituído por um intervalo genérico que começa em zero, logo $P(t)=0$ e $P(t+v)$ é $P(v)$ definindo uma probabilidade P para k eventos em um intervalo com frequência λ .

$$P(k, v) = \frac{e^{-\lambda v} (\lambda v)^k}{k!} \quad (9)$$

No contexto das descontinuidades, o intervalo v pode ser substituído por um cone de busca de um ângulo pré-determinado ψ . Desse ângulo é possível retirar uma área c , que é a subtração da área do cone no estereograma, dado por:

$$c = 1 - \cos \psi \quad (10)$$

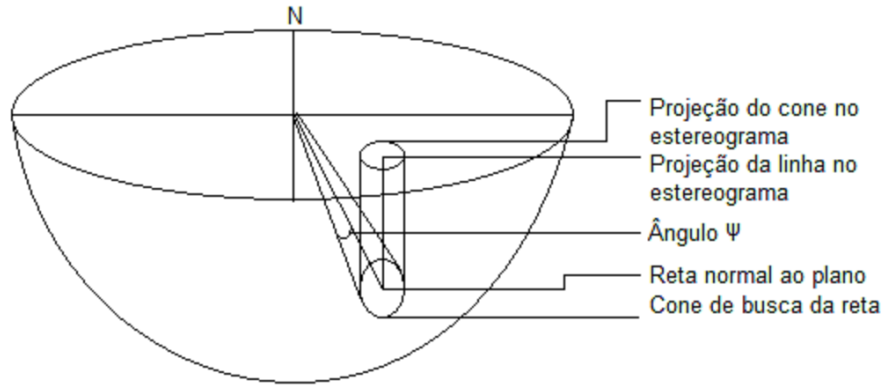


Figura 3 Representação esquemática de um cone de busca na projeção estereográfica

Substituindo a equação (10) na equação (9) para um intervalo de n descontinuidades a frequência λ será igual ao valor de n .

$$P(k, c) = \frac{e^{-nc} (nc)^k}{k!} \quad (11)$$

Com base nesta relação, é possível conhecer a probabilidade P na qual k descontinuidades dentro de um cone de busca de ângulo ψ indica perda de aleatoriedade. Esta probabilidade é dada pela equação a seguir:

$$P(t, c) = 1 - \frac{\sum_{j=0}^t e^{-nc} (nc)^j}{j!} \quad (12)$$

De maneira que a medida que t aumenta a probabilidade $P(t, c)$ diminui até o ponto em que a distribuição deixa de ser aleatória, numericamente descrito quando $P(t, c) < c$.

3.4- Cálculo da orientação média e da dispersão angular

Depois da identificação das principais famílias de descontinuidades é possível determinar alguns parâmetros estatísticos para caracterizar melhor sua orientação e dispersão angulares. No caso utilizou-se a distribuição de Fisher (Cheeney, 1983), que se aplica a distribuições de dados de orientação que tendem a formar concentrações circulares em torno de um vetor médio. A orientação média de uma família, dada pelos seus cossenos diretores (l_m, m_m, n_m) pode ser calculada com a utilização dos cossenos diretores (l_i, m_i, n_i) de w dados de uma família segundo:

$$\begin{aligned} l_m &= \frac{\sum_{i=0}^w l_i}{w} \\ m_m &= \frac{\sum_{i=0}^w m_i}{w} \\ n_m &= \frac{\sum_{i=0}^w n_i}{w} \end{aligned} \quad (13)$$

A dispersão angular pode ser obtida através do parâmetro de Fisher K :

$$\begin{aligned} r &= \sqrt{l_m^2 + m_m^2 + n_m^2} \\ K &= \frac{w-1}{w-r} \end{aligned} \quad (14)$$

Com base no valor K de Fisher é possível conhecer o ângulo θ do cone que representa o valor de limite de confiança para essa distribuição, seguindo a fórmula:

$$\cos \theta \approx 1 + \frac{\ln(1 - P(> \theta))}{k} \quad (15)$$

3.5- Cálculo das interseções

A determinação da atitude de linhas de interseção entre famílias de descontinuidades é relevante em geotecnia na medida em que é um parâmetro usado por exemplo na análise de rupturas de talude em cunha, ou seja, determinadas por dois planos de descontinuidade.

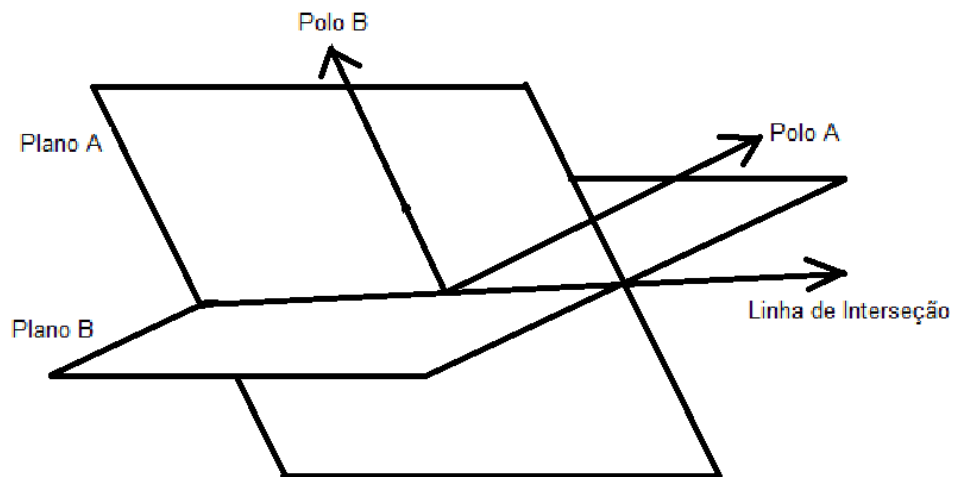


Figura 4 Planos, Polos e a Linha de Interseção resultante

A atitude da interseção entre dois planos é igual ao produto vetorial dos polos dos dois planos, ou seja, é um vetor perpendicular aos dois polos.

O produto vetorial pode ser obtido pela resolução do determinante da matriz abaixo composta pelos componentes i,j,k do vetor resultante do produto vetorial e pelos cossenos diretores dos planos N1 e N2:

$$N1 \times N2 = DET \begin{bmatrix} i & j & k \\ l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \end{bmatrix} \quad (16)$$

Comparando os resultados entre as descontinuidades de duas famílias temos todas as possíveis interseções entre esses planos.

3.6 Distribuição espacial das frequências

Conhecendo a orientação média e a frequência (medida em campo) das famílias é possível criar um modelo simples de distribuições das frequências em 3D.

Se tivermos duas famílias de descontinuidades, suas frequências numa certa direção S podem ser somadas:

$$\lambda_s = \lambda_1 \cos \alpha_1 + \lambda_2 \cos \alpha_2 \quad (17)$$

Para n famílias de descontinuidades (Hudson e Priest, 1983):

$$\lambda_s = \sum_{i=1}^n \lambda_i \cos \alpha_i \quad (18)$$

Lançando-se essas somatórias de frequências das várias famílias em todas as direções no espaço, por exemplo, através de um estereograma, obten-se um modelo de distribuição das frequências em 3D.

4- Materias e Métodos

Para a execução do projeto foram necessários, primeiramente, dados de descontinuidades para testar a validade e as capacidades das rotinas a serem elaboradas nas diferentes situações em que poderão ser utilizadas pelo usuário final. Os dados pré-definidos foram desenvolvidos usando critérios próprios com objetivo de localizar erros matemáticos. Os dados reais foram retirados de trabalhos realizados durante o curso de graduação.

O código foi feito usando a linguagem de programação Python, (Python Software Foundation, 2010), por ser gratuita, multiplataforma e de fácil compreensão. . Utilizou-se a versão 2.7 da linguagem, por esta ser a mais bem distribuída na atualidade. Utilizou-se também as bibliotecas NumPy 1.6.1 (biblioteca científica) e Matplotlib 1.1.0 (biblioteca de plotagem 2D), além do ambiente de desenvolvimento integrado IDLE, para simplificar o processo de busca por erros.

5- Resultados Obtidos

Esse trabalho gerou como resultado quatro diferentes rotinas computacionais que foram adicionadas ao programa OpenStereo (Grohmann *et al*, 2011), para fácil utilização.

Todas as rotinas executam inicialmente a transformação dos dados estruturais (rumo, mergulho, em graus) em cossenos diretores, e ao final dos cálculos retornam o resultado no formato *Rumo, Mergulho em graus*. Esse método foi usado porque o OpenStereo faz as plotagens da interface gráfica com os dados no formato usado em geologia estrutural.

A primeira rotina é responsável pela identificação das famílias (anexo 16). Ela começa adquirindo parâmetros de origem dos dados e o ângulo do cone de busca escolhido. Usando o método de Poisson ela descobre a quantidade de pontos mínima para que a distribuição deixe de ser aleatória e faz a contagem para cada ponto. Por fim ela agrupa pontos não aleatórios pela proximidade em famílias. Caso haja necessidade a função também executa a correção linear pelo método de Terzaghi antes da definição das famílias.

A terceira rotina calcula as somatórias de frequência das várias famílias (anexo 17). Ela utiliza os dados de orientação e frequência de um grupo de famílias e calcula as frequências aparentes de todos os membros para cada pólo em uma rede uniforme, conseguindo assim uma distribuição 3D de frequências.

A segunda rotina trata das interseções (anexo 18). Ela utiliza duas diferentes famílias ou grupos de pontos escolhido pelo usuário e calcula a interseção entre cada ponto de uma família e todos os pontos da outra. A rotina termina retornando os dados de interseção para a saída gráfica.

A quarta rotina trabalha com as estatísticas (anexo 19). Ela adquire os parâmetros da família escolhida e utiliza esse dados para o cálculo de seu rumbo de mergulho e mergulho médios e utilizando isso para conseguir no fim o seu índice de Fisher K. Por fim é usado o parâmetro de Fisher K para retirar ângulos de confiança para a distribuição. No programa foram usados ângulos de 70% e 90% de confiança.

Os resultados obtidos a partir da aplicação das rotinas estão ilustrados nos Anexos 1 a 15 e são discutidos na seção 6, a seguir.

6- Interpretação e Discussão dos Resultados

6.1- Identificação de famílias

Os resultados dos testes de identificação de famílias apresentam resultados coerentes com o esperado para o método convencional, feito à mão. Os anexos 1 a 4 mostram uma comparação do resultado obtido com isolinhas de densidade para vários casos diferentes. O anexo 1 trata de uma família de grande dispersão, o 2 uma distribuição aparentemente uniforme de pontos com uma família bem caracterizada, o 3 uma guirlanda e o 4 famílias subverticais.

Um fator muito importante do método estudado foi comparar os resultados por ordem da abertura do cone de busca escolhido (ângulo ψ). Os anexos de 5 a 7 tratam um conjunto de dados utilizando cones de busca de ângulos apicais diferentes. É possível ver que com um ângulo de 5° o resultado apresenta mais famílias com uma dispersão menor. Com 10° de ângulo a dispersão aumenta e famílias pequenas próximas se unem. Por fim, a 15° de ângulo a dispersão aumenta mais, adicionando às famílias uma quantidade maior de membros.

6.2- Correção da frequência de descontinuidades

Para exemplificar a correção de amostragem, o anexo 8 mostra a comparação entre as isolinhas de densidade para os mesmos dados do item 6.1 sem nenhuma correção e com uma correção considerando uma linha de *scan* de direção 90/30. Os resultados mostram que se em uma situação sem correção temos apenas uma grande concentração, com a correção são geradas outras concentrações importantes no 3° e 4° quadrantes.

Para exemplificar a influência da correção de Terzaghi nos resultados da rotina de identificação de famílias foi utilizado o conjunto de dados do item 6.1 com um cone de busca de 10° (anexo 6) e correções para linhas de *scan* teóricas de direções 90/30 e 180/30.. Os anexos 9 e 10 apresentam os resultados. No anexo 9 é possível ver que a correção aumentou o tamanho das famílias localizadas no 3° e 4° quadrante do estereograma e uma família nova foi criada com direção média 150/60. No anexo 10 a situação inverte-se, com os pontos localizados no 1° e 2° quadrante crescendo e a família localizada na direção média 360/45 desaparecendo.

6.3- Cálculo das Interseções

As medidas de interseção entre as famílias são mostradas nos anexos 11 a 13. O anexo 11 mostra a interseção entre famílias direcionalmente opostas com mergulho próximo, o anexo 12 a interseção entre famílias consecutivas em um mesmo quadrante e o anexo 13 o resultado para interseções entre famílias subverticais.

6.4- Distribuição espacial das frequências

Para exemplificar as rotinas de frequências foram utilizadas duas famílias, de medidas de rumo de mergulho e mergulho 312/53 e 146/45 em duas situações distintas: uma em que ambas têm a mesma frequência, e outra em que a segunda tem o dobro da frequência da primeira. Os resultados são mostrados nos anexos 14 e 15.

6.5- Estatísticas dos dados

Para a demonstração dos cálculos estatísticos foram utilizados os valores de direção de mergulho e mergulho para todas as famílias dos anexos de 1 a 4. O resultado são os parâmetros de direção do mergulho médio, mergulho médio e Fisher K para cada família.

7-Conclusões

O objetivo principal desse trabalho era a criação de uma ferramenta de fácil utilização para a análise de descontinuidades. As rotinas desenvolvidas objetivaram, portanto, equilibrar uma resposta automática com a personalização dos resultados. As funções escolhidas foram suficientes, mas um programa nunca está completo, a quantidade de outras funções que poderiam ser adicionadas as já existentes e de pequenos ajustes que poderiam ser feitos é ilimitada.

Os testes realizados mostraram o desempenho correto das rotinas, destacando-se a forte influência da correção direcional de frequências na distribuição dos dados, e do ângulo de abertura do cone de busca no número de famílias identificadas.

Os resultados obtidos no trabalho mostram-se compatíveis em comparação com o esperado em resultados adquiridos pelos métodos convencionais de observação, mas diferem em ter replicabilidade e, tendo uma origem estatística, conseguem identificar padrões que o olho humano não é capaz de perceber. Ao mesmo tempo o conhecimento geológico continua indispensável para um melhor resultado, sendo responsável pela escolha dos parâmetros que serão utilizados.

8-Referências Bibliográficas

- Campanha, G. A. C., & Quintanilha, J. A.. 1996. Análise Estatística de Dados Estruturais; Estatística de Dados Direcionais Tridimensionais, **In:** Carneiro, Celso Dal Re (editor) *Projeção estereográfica para análise de estruturas*, UNICAMP / CPRM / IPT, São Paulo, pp. 51-58
- Cheeney, R. F. 1983. *Statistical methods in geology*. George Allen & Unwin, London. 169 pp.
- Deere, D.U., Hendron, A.J., Jr., Patton, F.D., and Cording, E.J., (1967), Design of Surface and Near-Surface Construction **In:** Fairhurst C. (editor) *Failure and Breakage of Rock*, Soc. of Min. Eng., AIME, Nova Iorque, pp. 237-302.
- Grohmann, C.H., Campanha, G.A.C. and Soares Junior, A.V., 2011. OpenStereo: um programa Livre e multiplataforma para análise de dados estruturais. **In:** *XIII Simpósio Nacional de Estudos Tectônicos*.
- Hudson, J.A.; Priest, S.D. 1983. Discontinuity frequency in rock masses. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 20(2), 73 - 89.
- Hudson, J.A.; Harrison, J.P. 1997. *Engineering Rock Mechanics: An Introduction to the Principles*. Elsevier Science, London, 456 pp.
- Mahtab, M.A. & Yegulalp T.M. 1982. *A Rejection Criterion for Definition of Clusters in Orientation Data*. Proceedings 23rd US Symposium on Rock Mechanics. Berkeley, California. 760p.
- Millman, J., Vaught, T., 2008. *The State of SciPy* . *In: 7th Annual Python in Science Conference*
- Priest S.D. 1993. *Discontinuity Analysis for Rock Engineering*. Chapman & Hall, London, 475 pp.
- Priest, S.D. & Hudson, J.A. 1976. *Estimation of discontinuity spacing and trace length using scan line surveys*. *International Journal Rock Mechanics, Mining Sciences and Geomechanics*, 18:183-197.

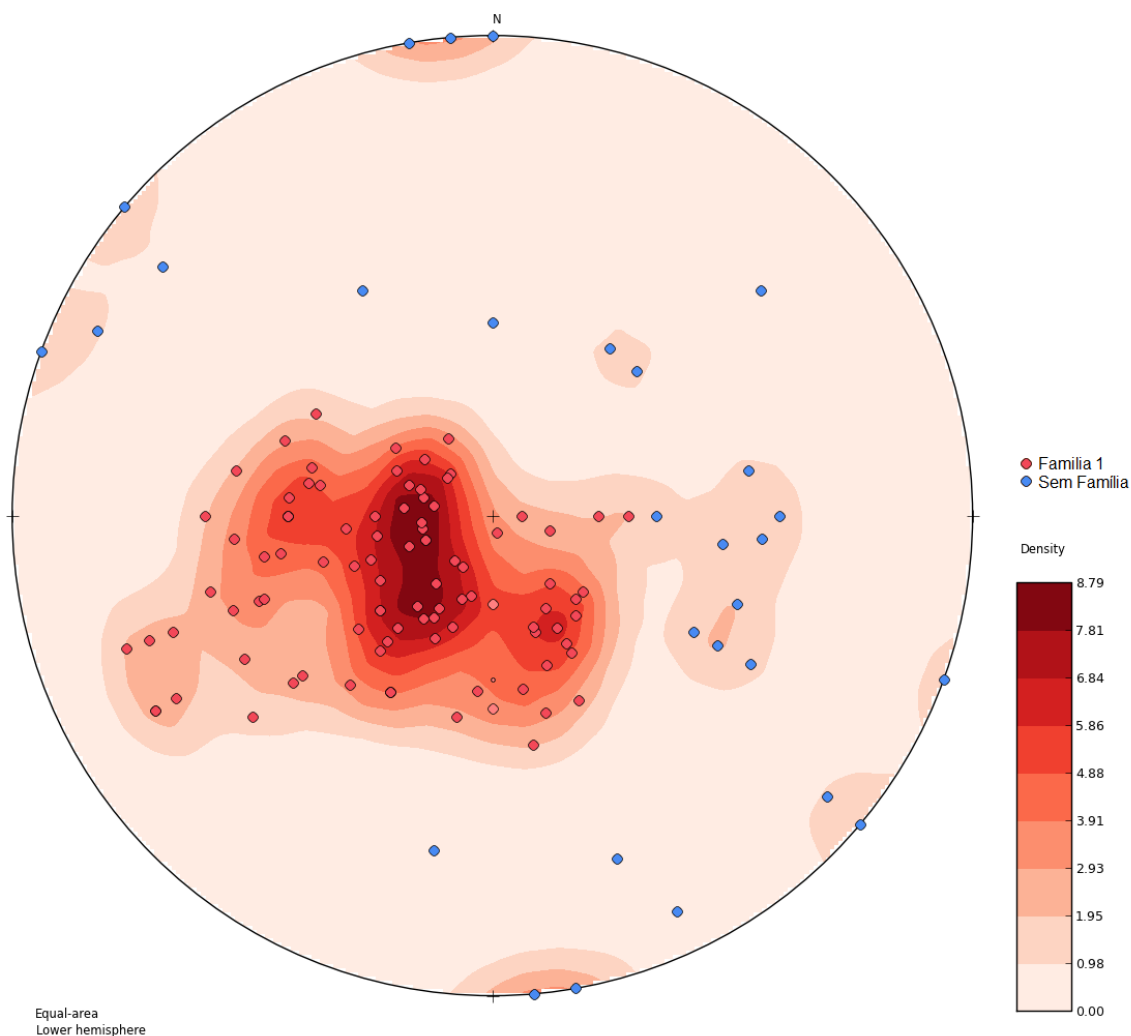
Python Software Foundation 2010. Python, version 2.7: programming language software. Reston, Virginia, Zope Corporation.

Shanley, R.J. & Mahtab, M.A. 1976. *Delineation and analysis of centers in orientation data*. Math Geology 8:9-23

Terzaghi, R.D. 1965. *Sources of error in joint surveys*. *Géotechnique*, 15, 287–304.

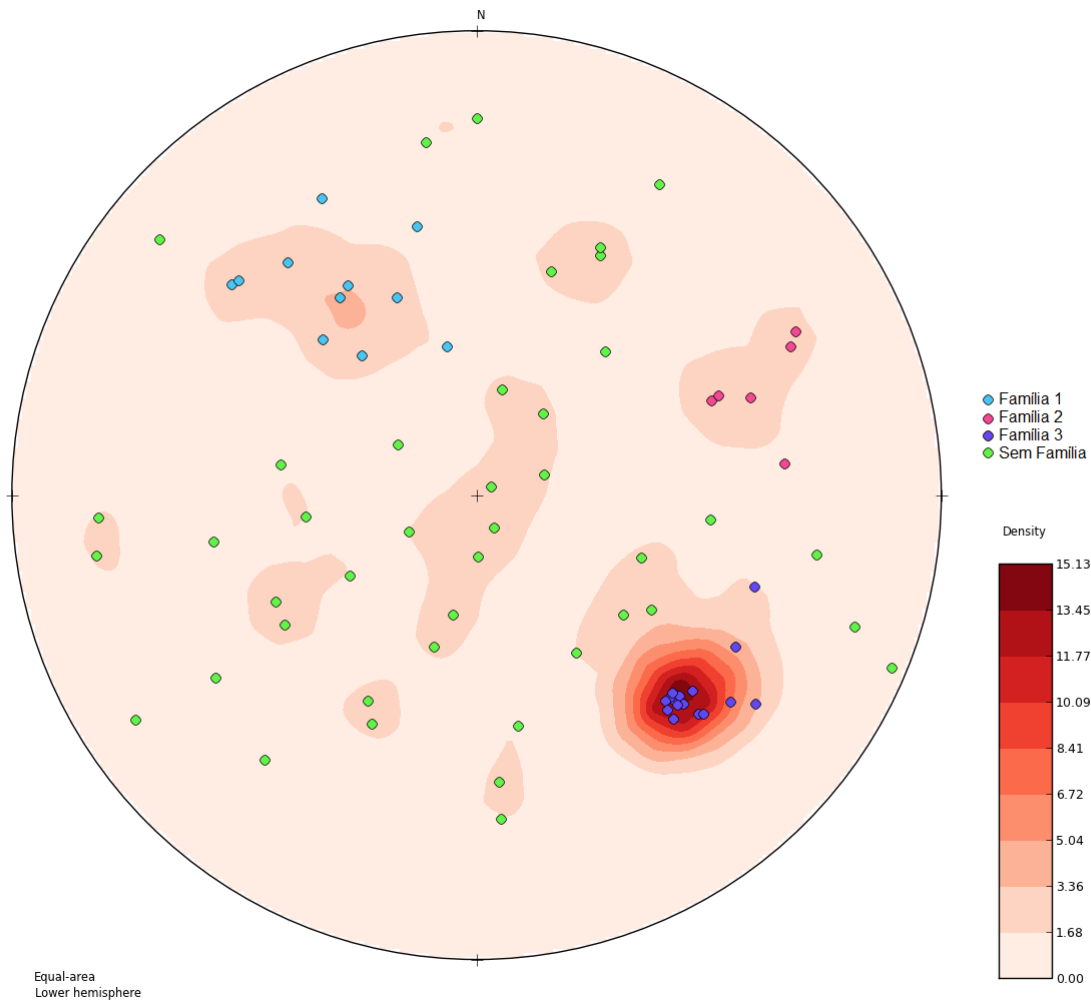
University of Alberta. 2011. *Orientation data analysis*. Disponível em <http://courses.eas.ualberta.ca/eas421/lecturepages/orientation.html>. Acessado em 14 jul 2012.

Anexo 1 – Identificação automatizada de famílias. Resultado para uma família com dispersão moderada.



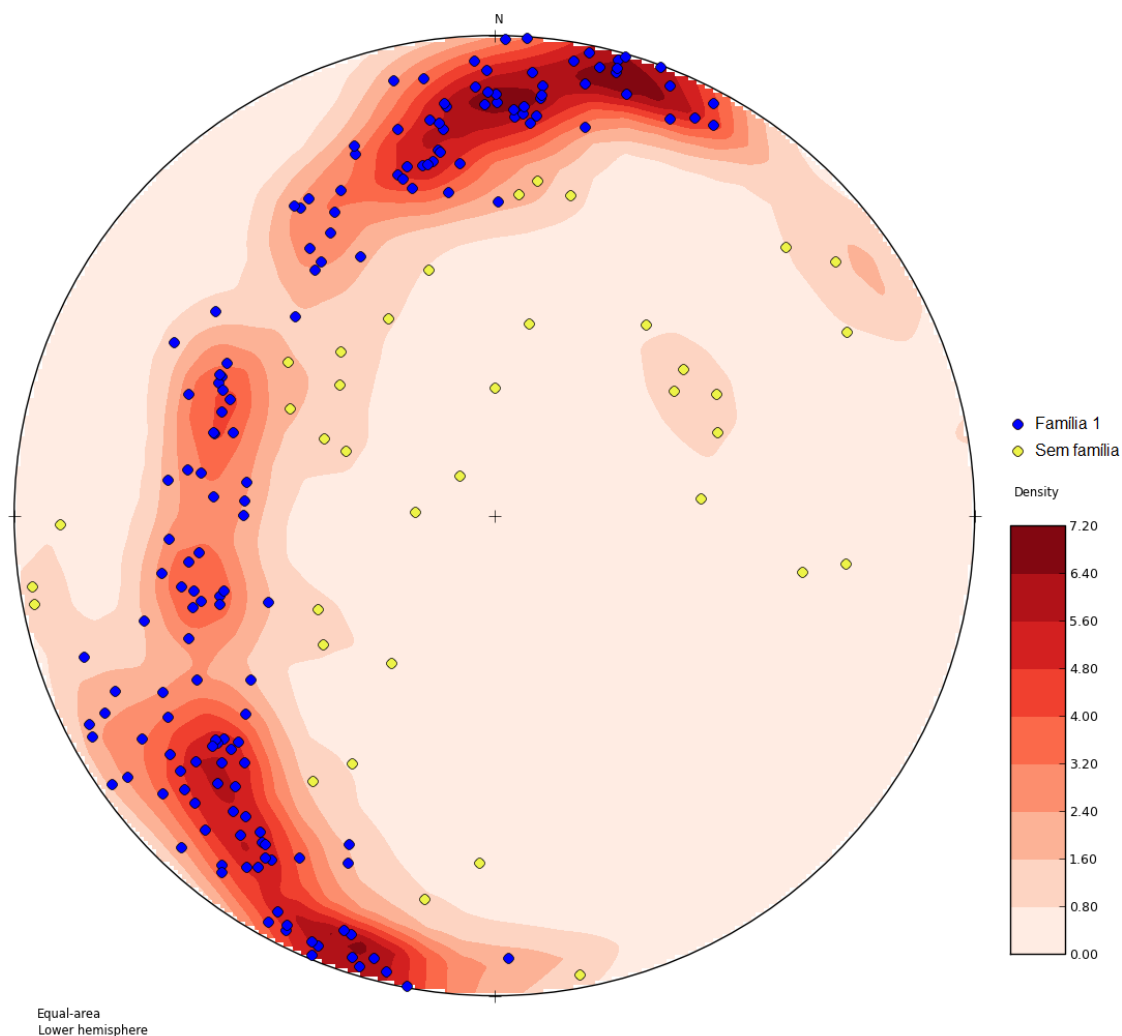
Estatísticas	Família 1
N	97
Rumo médio	120
Mergulho médio	28
Fisher K	117.344
Cone de Confiança - 70%	27°
Cone de Confiança - 90%	37°

Anexo 2 – Identificação automatizada de famílias. Resultado para uma distribuição aparentemente uniforme com uma única concentração visível



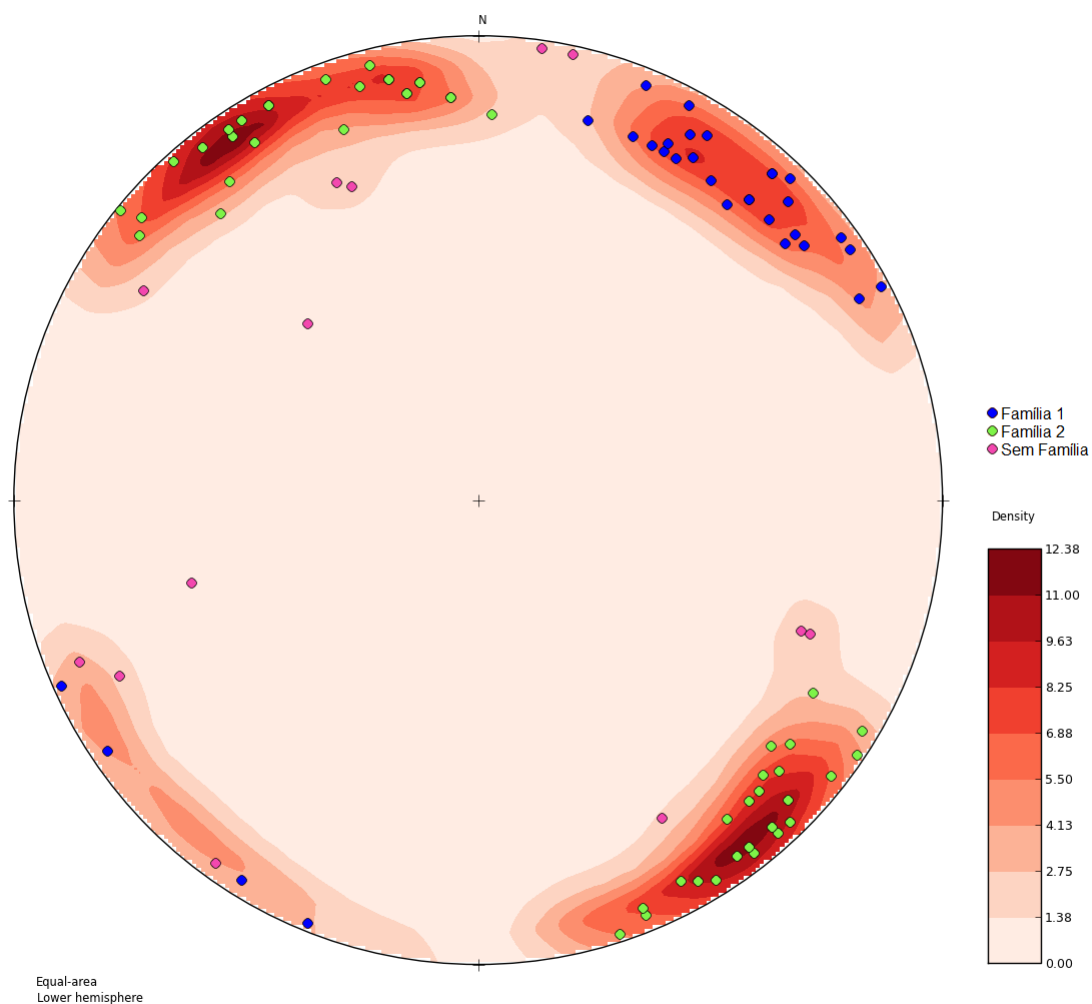
Estatísticas	Família 1	Família 2	Família 3
N	11	6	17
Rumo médio	132	250	326
Mergulho médio	36	55	45
Fisher K	133	69	127
Cone de Confiança - 70%	7°	10°	15°
Cone de Confiança - 90%	10°	15°	21°

Anexo 3 – Identificação automatizada de famílias. Resultado para dados em formato de guirlanda



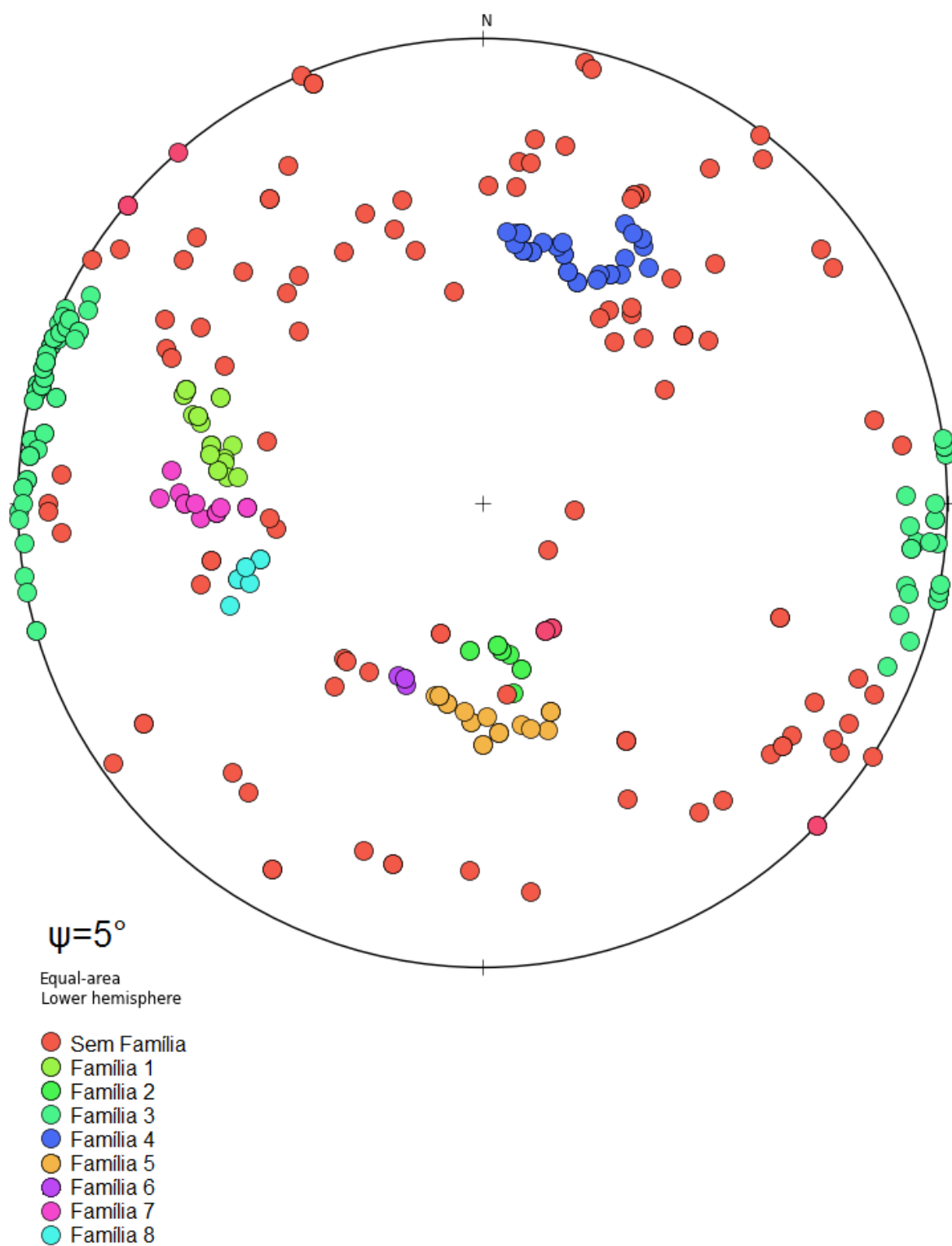
Estatísticas	Família 1
N	163
Rumo médio	101
Mergulho médio	72
Fisher K	3,01531
Cone de Confiança - 70%	64°
Cone de Confiança - 90%	94°

Anexo 4 – Identificação automatizada de famílias. Resultado para um conjunto de dados com duas famílias subverticais

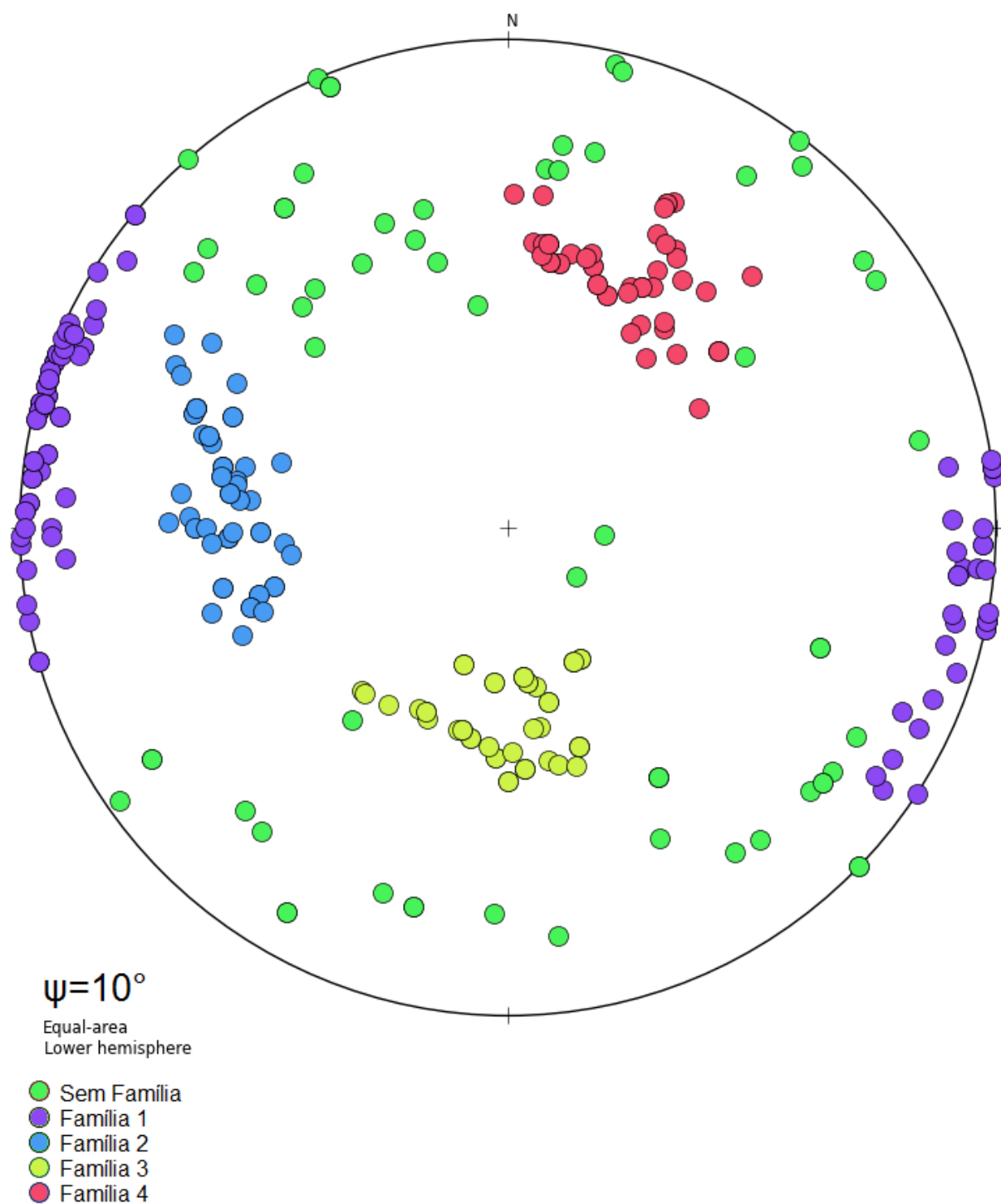


Estatísticas	Família 1	Família 2
N	50	31
Rumo médio	233	296
Mergulho médio	81	80
Fisher K	28,2351	18,0541
Cone de Confiança - 70%	88°	47°
Cone de Confiança - 90%	146°	67°

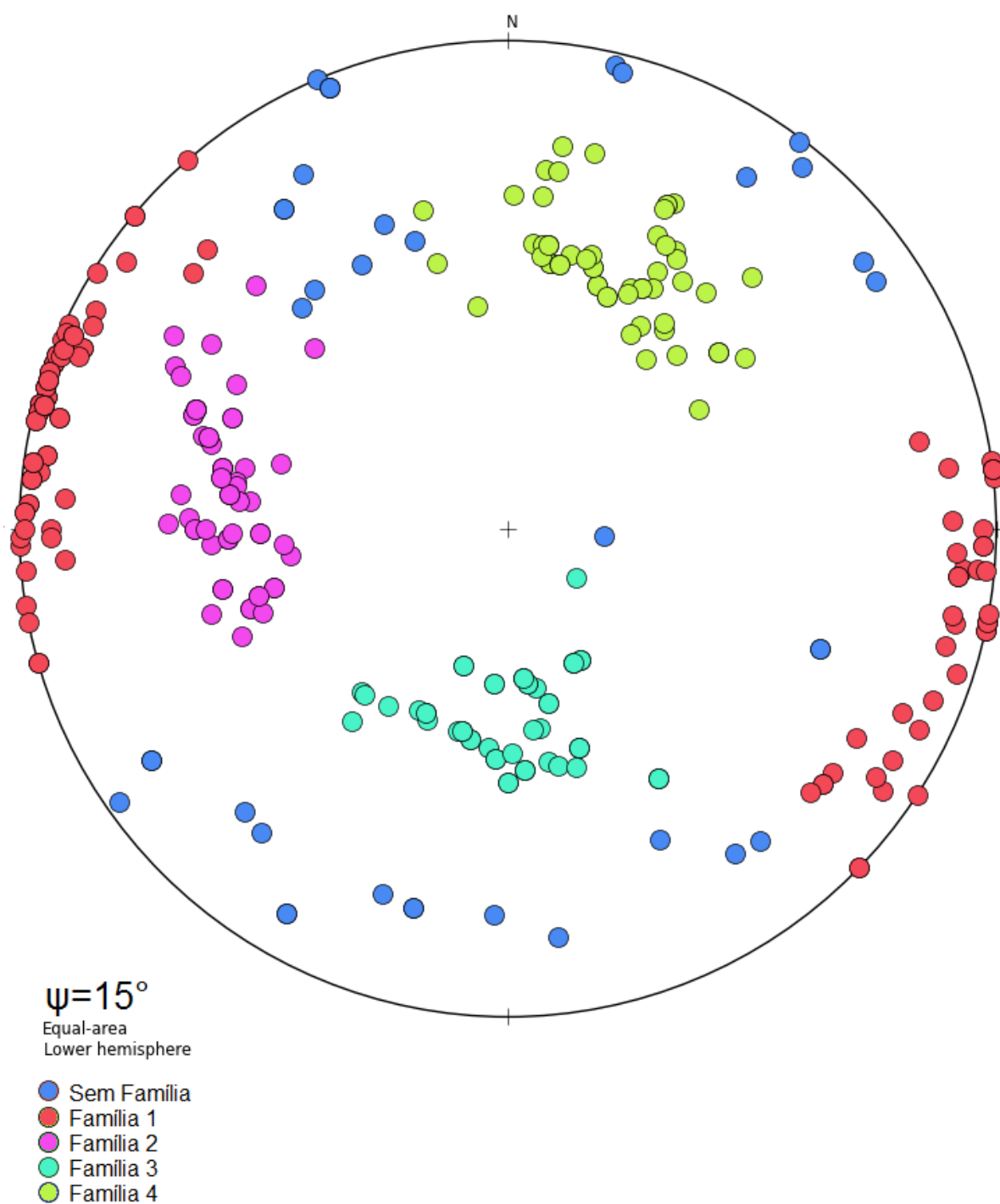
Anexo 5 – Resultado da rotina de identificação de famílias para um conjunto de dados com várias famílias, utilizando um cone de busca de 5°



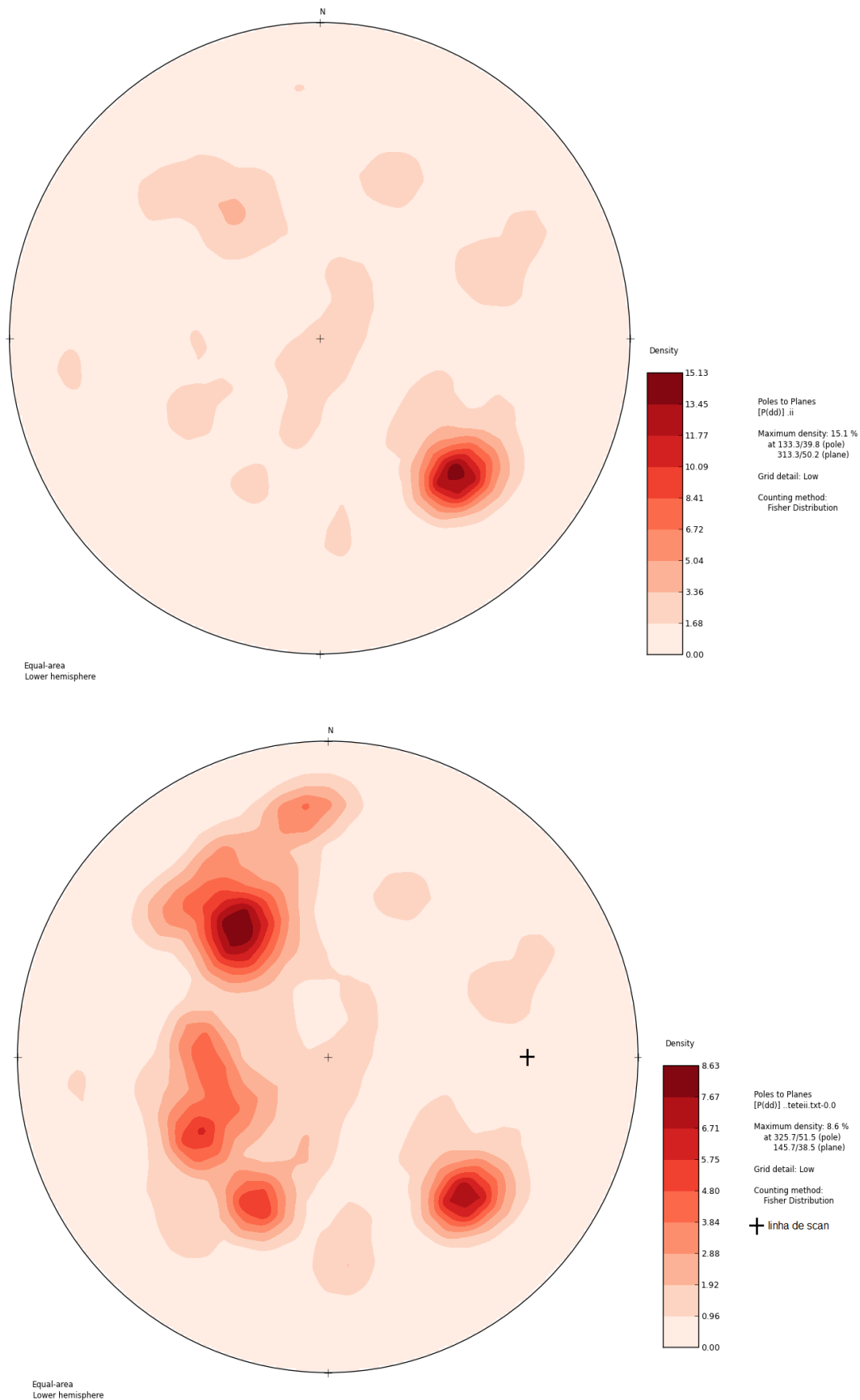
Anexo 6 – Resultado da rotina de identificação de famílias para um conjunto de dados com várias famílias, utilizando um cone de busca de 10°.



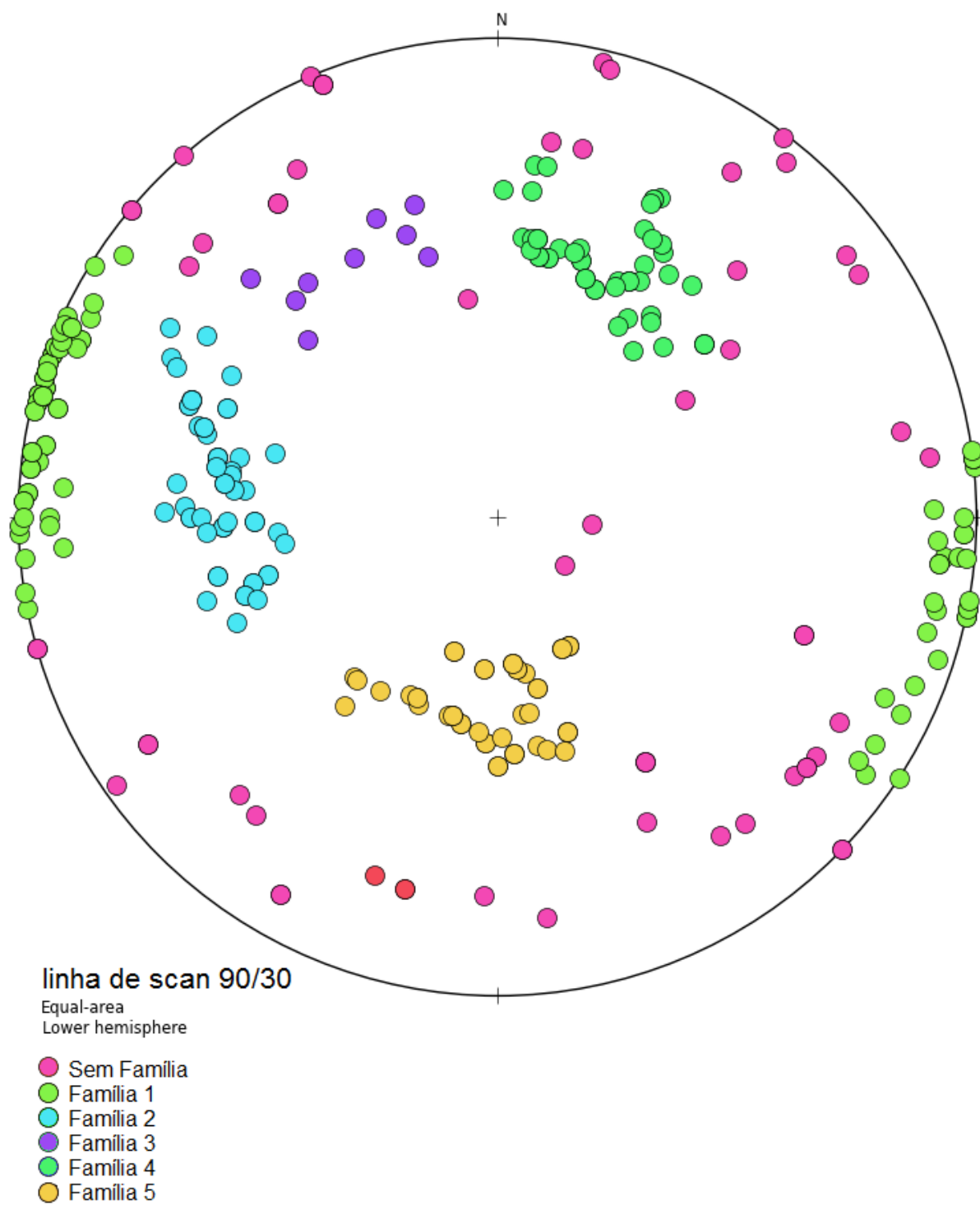
Anexo 7 – Resultado da rotina de identificação de famílias para um conjunto de dados com várias famílias, utilizando um cone de busca de 15°.



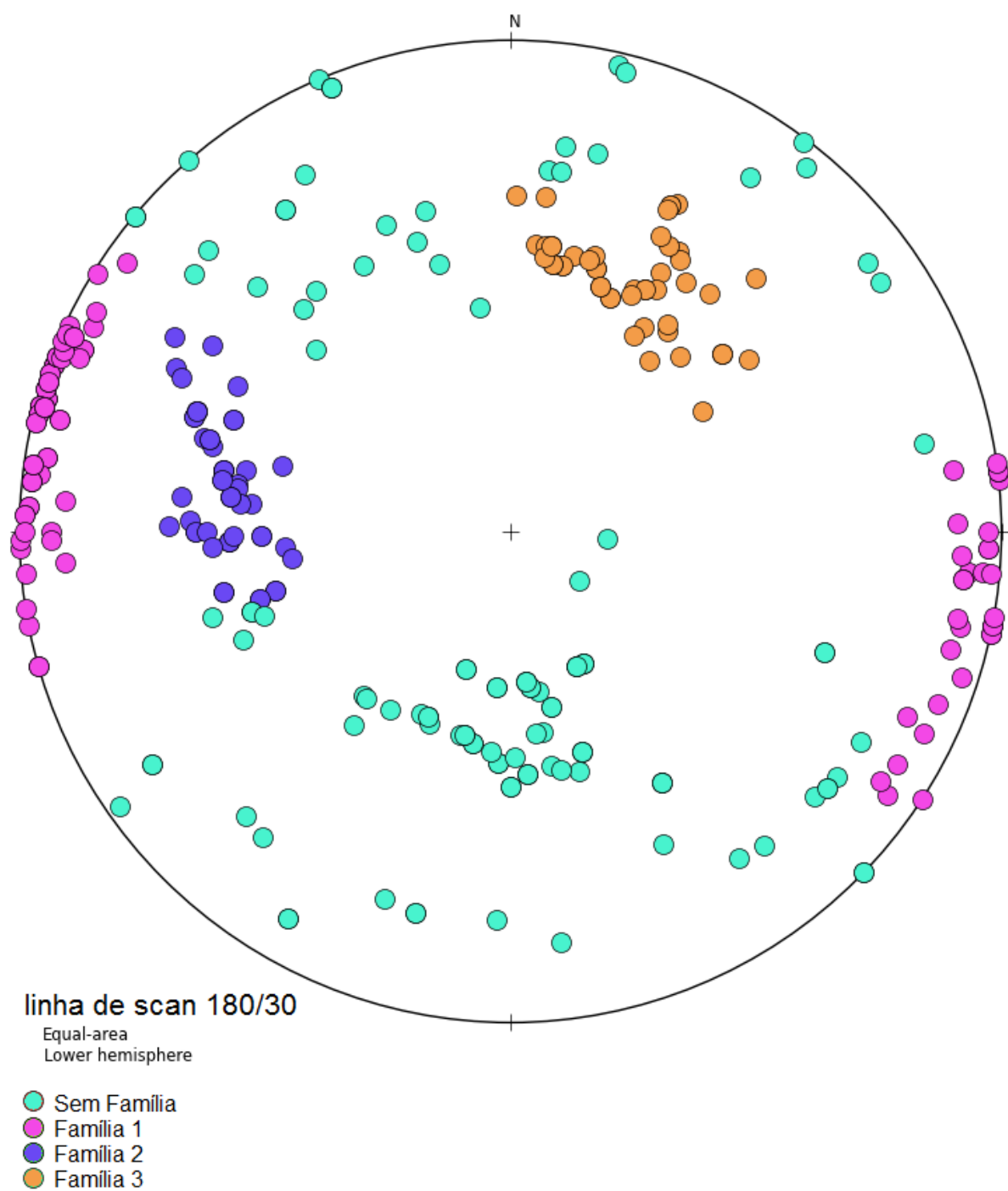
Anexo 8 - Efeito da correção de Terzaghi na geometria das isolinhas de densidade.
Figura superior: sem correção. Figura inferior: com correção para uma linha de amostragem de orientação 90/30



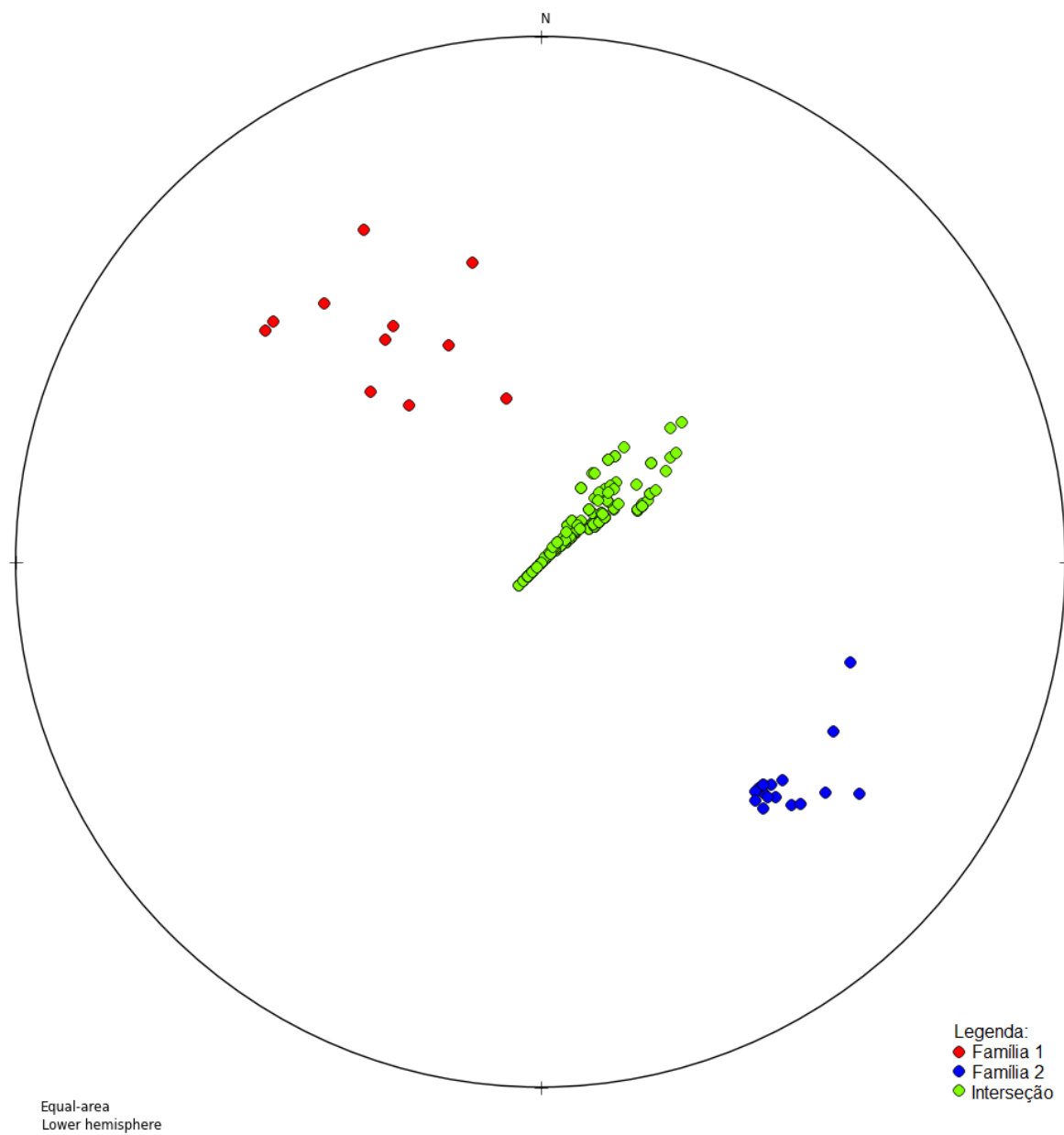
Anexo 9 – Identificação automatizada de famílias utilizando a correção de Terzaghi.
para uma linha de scan 90/30



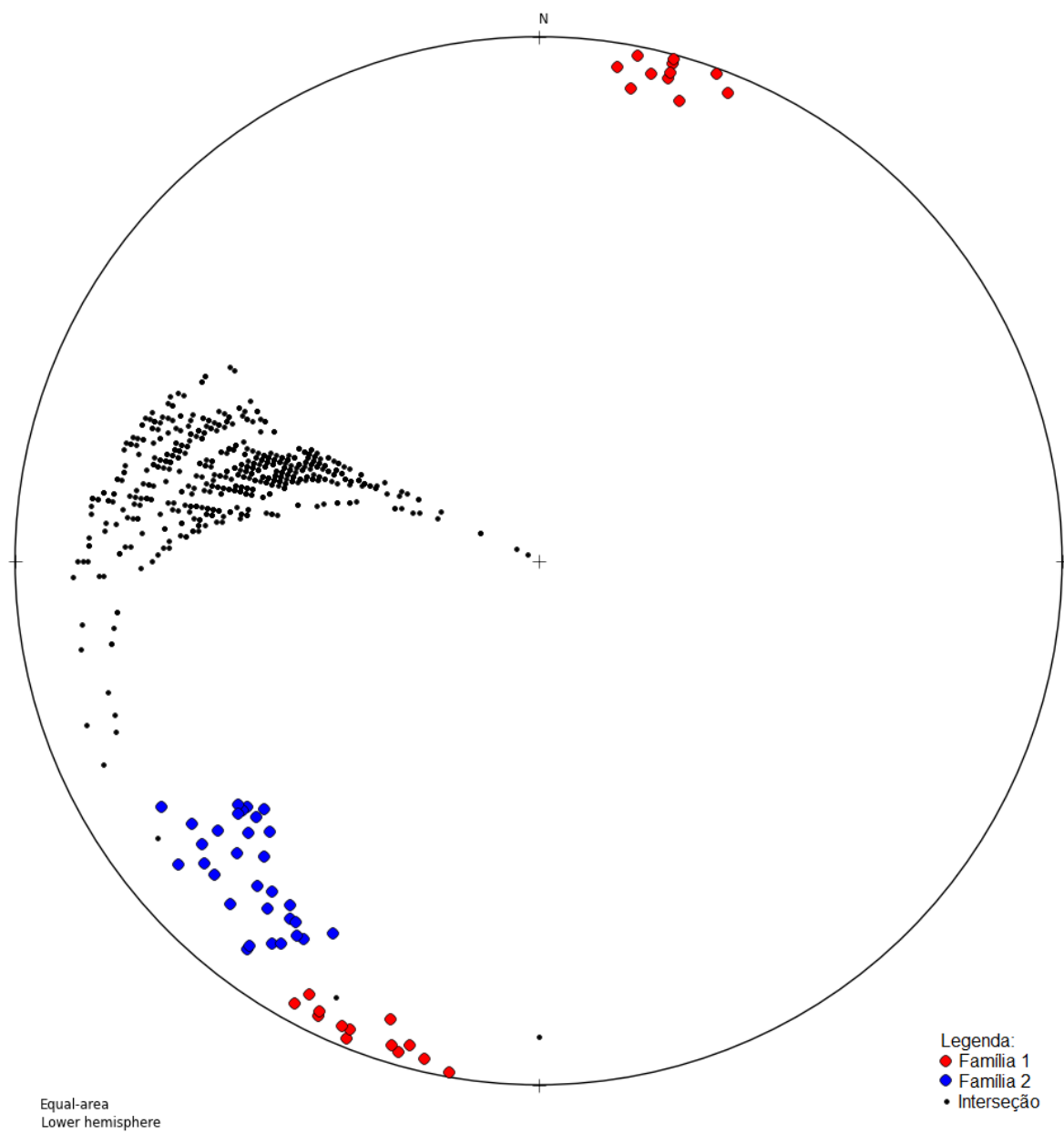
Anexo 10 - Identificação automatizada de famílias utilizando a correção de Terzaghi.
para uma linha de scan 180/30



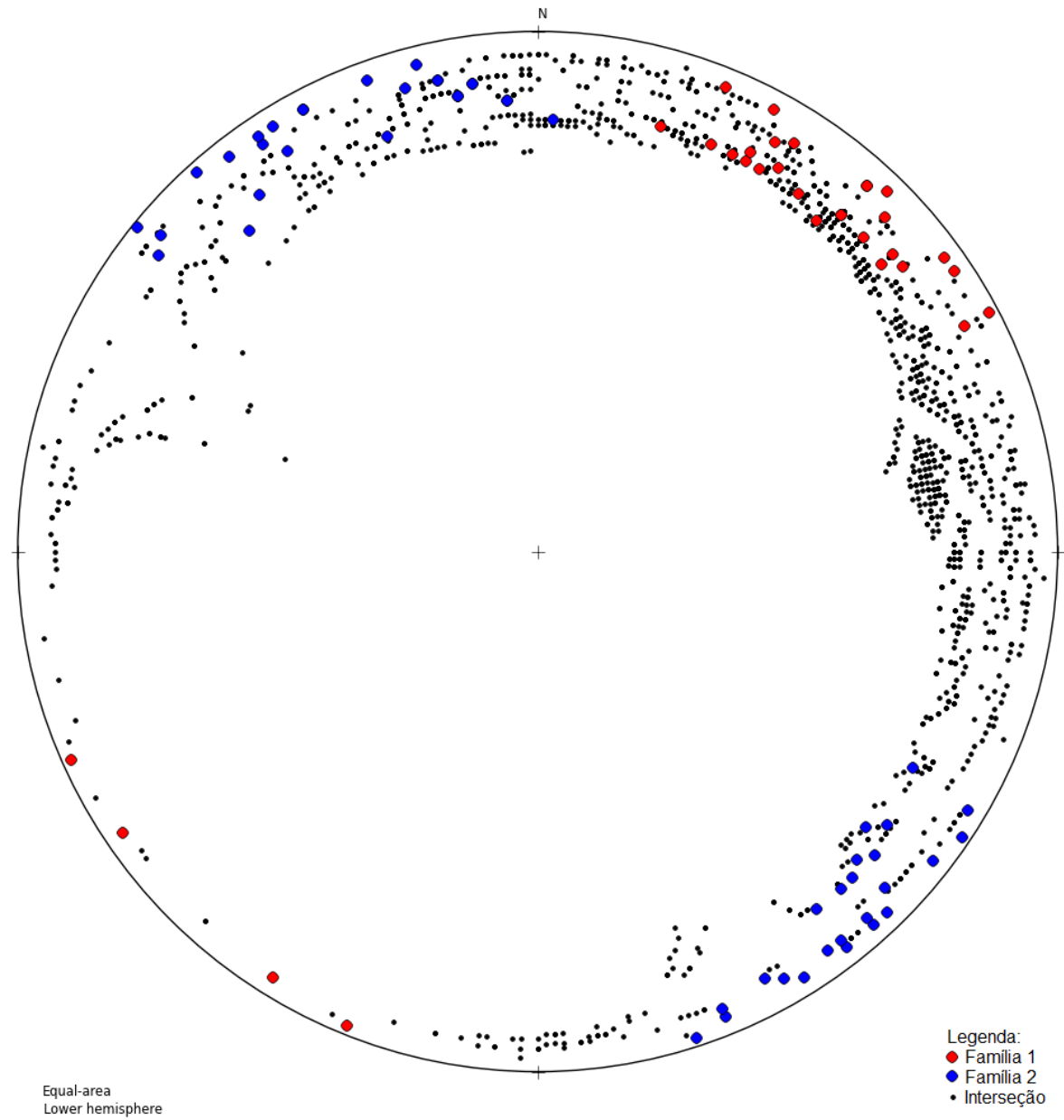
Anexo 11 – Interseções entre famílias direcionalmente opostas com mergulho próximo



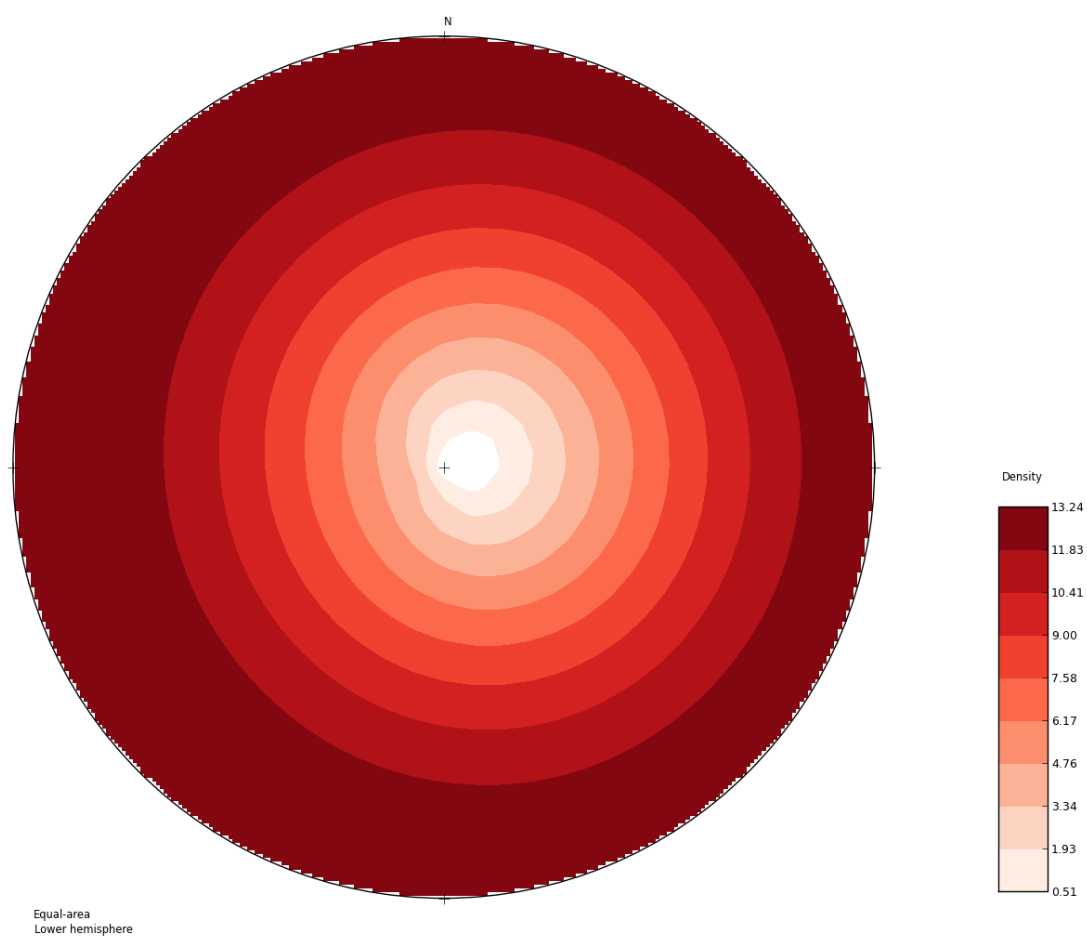
Anexo 12 – Interseções entre famílias consecutivas



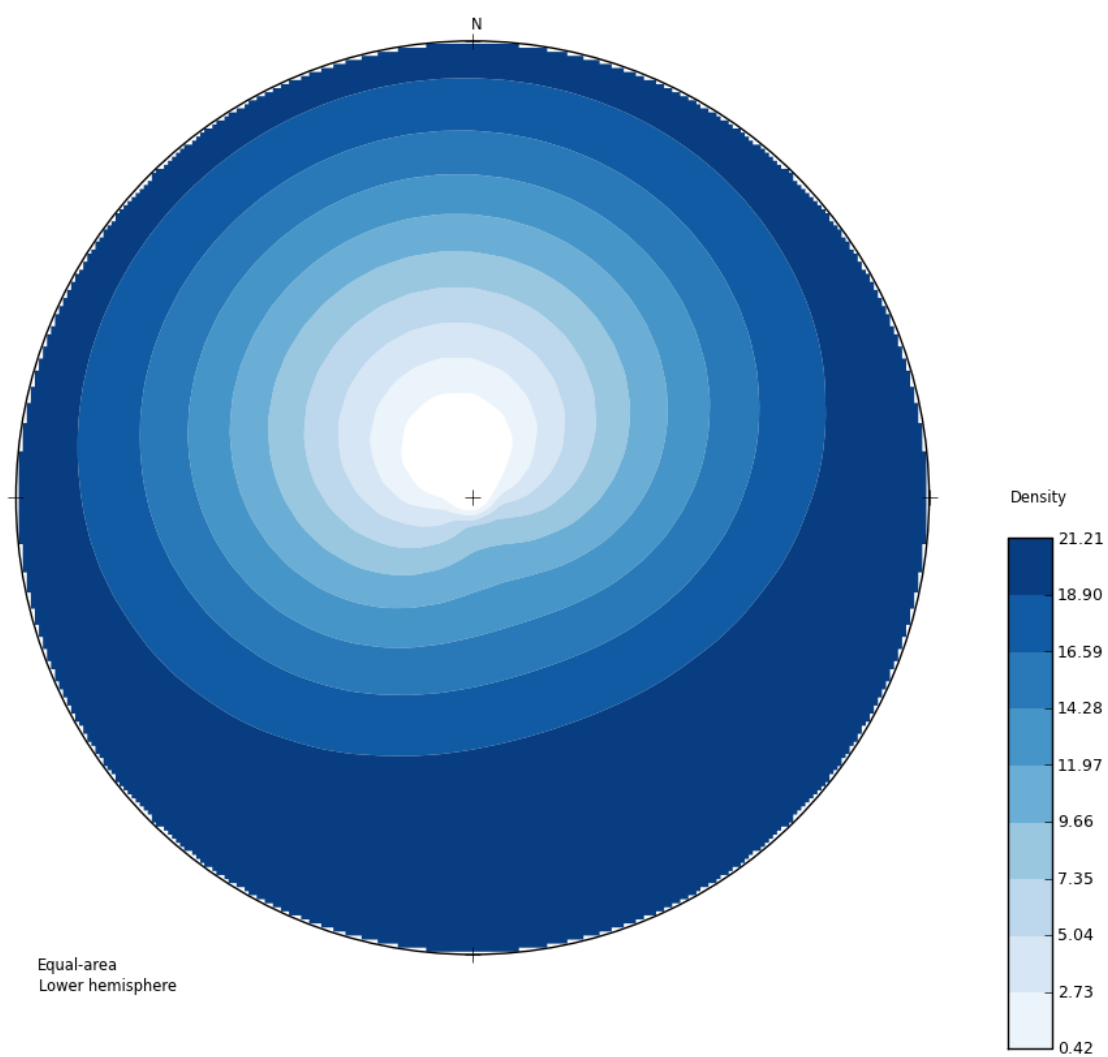
Anexo 13 – Interseções entre famílias subverticais



Anexo 14



Anexo 15



Anexo 16 – Rotina para identificação de famílias

```
def familias(ad,ang,tz):
#ad=dados de dipdir/dip ang=angulo de cone de busca
    import numpy as np
    import math
    import csv
    for i in ad:
        if i[0]+180>360:
            i[0]=i[0]-180
        else:
            i[0]=i[0]+180
        i[1]=90-i[1]
    dircos = np.array([[np.sin(azim) * np.cos(theta), np.cos(azim) * np.cos(theta), np.sin(theta)] for
    azim, theta in np.radians(ad)])
# cálculo do tcritico
    n=len(dircos)
    c=1-math.cos(math.radians(ang))
    nc=n*c
    j=0
    t=150
    pp=0
    tcrit=0
    p=0
    while j<t:
        p=p+(((math.exp(-1*nc))*(math.pow(nc,j)/math.factorial(j))))
        j=j+1
        pp=1-p
        if pp>=c:
            tcrit=tcrit+1
    print "Valor do t crítico: ", tcrit
# cálculo do ângulo entre pontos
    n=len(dircos)
    wmag=np.ones((n,1))
    If tz!=0:
        wmag= terzaghi(dircos,tz)
    cos=[]
    n=len(dircos)
    fam= np.zeros((n,5))
    n=0
    den=0
    for i in dircos:
        for j in dircos:
            c=math.fabs(i[0]*j[0]+i[1]*j[1]+i[2]*j[2])
            if c>0.99999:
                c=1
            elif c<-0.99999:
                c=-1
            else:
                c=c
            cos=math.degrees(math.acos(c))
            if cos<=ang:
                den=den+wmag
        if den>=tcrit:
            fam[n]=[i[0],i[1],i[2],100,0]
            n=n+1
        else:
            fam[n]=[i[0],i[1],i[2],0,0]
            n=n+1
```



```

        den=0
    n=1
    c=0
    cos=0
    x=0
# definir familias
    for i in fam:
        if i[3]!=0:
            if i[3]==100:
                i[3]=n
                for j in fam:
                    if j[3]!=0:
                        c=math.fabs(i[0]*j[0]+i[1]*j[1]+i[2]*j[2])
                        if c>0.99999:
                            c=1
                        elif c<-0.99999:
                            c=-1
                        else:
                            c=c
                        cos=math.degrees(math.acos(c))
                        if cos<=ang:
                            j[3]=n
            n=n+1
        else:
            x=i[3]
            for j in fam:
                if j[3]!=0:
                    c=math.fabs(i[0]*j[0]+i[1]*j[1]+i[2]*j[2])
                    if c>0.99999:
                        c=1
                    elif c<-0.99999:
                        c=-1
                    else:
                        c=c
                    cos=math.degrees(math.acos(c))
                    if cos<=ang:
                        j[3]=x

# adicionar não densos as familias
    for i in fam:
        i[4]=i[3]
    for i in fam:
        if i[3]!=0:
            n=i[3]
            for j in fam:
                c=math.fabs(i[0]*j[0]+i[1]*j[1]+i[2]*j[2])
                if c>0.99999:
                    c=1
                elif c<-0.99999:
                    c=-1
                else:
                    c=c
                cos=math.degrees(math.acos(c))
                if cos<=ang:
                    if j[3]==0:
                        j[4]=i[3]

# imprimir resultado
    ad=np.loadtxt(nome)
    n=len(ad)
    final=np.zeros((n,3))
    n=0
    for i in fam:
        e,f=ad[n]

```

```

        final[n]=[e,f,i[4]]
        n=n+1
    writ = csv.writer(open("c:/..muitospontos.txt", "w"), dialect='excel-tab')
    writ.writerows(final)
    writ = None

def terzaghi(dircos,tz):
    fx=tz[0]
    fy=tz[1]
    vx=np.sin(fx*np.pi/180)*np.cos(fy*np.pi/180)
    vy=np.cos(fx*np.pi/180)*np.cos(fy*np.pi/180)
    vz=np.sin(fy*np.pi/180)
    maximo=84.3
    #_calculo do angulo_#
    terzang=[]
    for i in dircos:
        cos=math.fabs(vx*i[0]+vy*i[1]+vz*i[2])
        if cos>0.99999:
            cos=1
        elif cos<-0.99999:
            cos=-1
        else:
            cos=cos
        angulo=math.degrees(math.acos(cos))
        if angulo>maximo:
            angulo=maximo
        terzang=terzang+[cos]
    #___Determinação do fator de correção W___#
    w=[]
    for i in terzang:
        tok=1/(math.cos(math.radians(i)))
        w=w+[tok]
    #___Determinação do fator de correção normalizado Wni___#
    wni=[]
    wnis=math.fsum(w)
    l=len(w)
    for i in w:
        tok=(i*l)/wnis
        wni=wni+[tok]
    #___Determinação dos cossenos diretores corrigidos___#
    n=len(wni)
    wdircos=np.zeros((n,3))
    n=0
    x=[]
    while n<wnis-1:
        x=dircos[n]*wni[n]
        wdircos[n]=[x[0],x[1],x[2]]
        n=n+1
    #___Determinação da magnitude dos vetores corrigidos___#
    wmag=[]
    x=[]
    for i in wdircos:
        smag=math.sqrt(math.pow(i[0],2)+math.pow(i[1],2)+math.pow(i[2],2))
        wmag=wmag+[smag]
    return wmag

```

Anexo 17 – Rotina de identificação de frequências

```
def frequencia(freq):
#     freq=[dipdir,dip,freq] para cada família
        import math
        import numpy as np
        import csv
        contagem=([[0,90],[10,90],[20,90],[30,90],[40,90],[50,90],[60,90],[70,90],[80,90],[90,90]
,[100,90],[110,90],[120,90],[130,90],[140,90],[150,90],[160,90],[170,90],[180,90],[190,90],[200,
90],[210,90],[220,90],[230,90],[240,90],[250,90],[260,90],[270,90],[280,90],[290,90],[300,90],[3
10,90],[320,90],[330,90],[340,90],[350,90],[155,84],[165,82],[171,79],[180,78],[188,80],[196,82]
,[204,83],[147,80],[155,75],[163,70],[171,67],[180,67],[190,67],[198,70],[206,75],[214,80],[136,
78],[142,70],[150,63],[160,58],[170,55],[180,52],[191,54],[202,59],[212,62],[219,70],[225,80],[1
24,80],[130,70],[135,60],[144,53],[154,48],[167,43],[180,41],[194,43],[208,48],[217,53],[225,60]
,[232,70],[237,80],[116,83],[118,73],[121,63],[126,54],[135,45],[148,39],[162,34],[180,32],[189,
34],[214,38],[225,45],[233,54],[238,62],[243,75],[245,83],[106,80],[109,68],[112,57],[119,47],[1
25,37],[136,30],[154,23],[180,20],[208,23],[180,20],[208,23],[225,30],[236,38],[244,48],[248,58]
,[252,68],[254,80],[99,78],[100,65],[102,54],[106,44],[110,33],[118,23],[135,15],[180,10],[225,1
5],[243,23],[252,33],[257,42],[258,54],[260,65],[262,78],[90,77],[90,65],[90,52],[90,42],[90,32],[
90,21],[90,10],[90,0],[270,10],[270,22],[270,32],[270,42],[270,53],[270,65],[270,78],[82,78],[81,
65],[79,54],[77,43],[71,31],[65,23],[45,14],[0,10],[315,15],[298,22],[288,32],[284,42],[281,54],[2
79,65],[278,78],[74,80],[72,69],[69,58],[64,47],[57,38],[45,30],[25,24],[0,20],[334,23],[316,29],[3
04,36],[298,46],[292,56],[288,68],[286,80],[65,84],[64,72],[58,62],[52,52],[44,45],[32,38],[18,33]
,[0,31],[340,33],[324,38],[317,45],[308,52],[302,62],[298,72],[296,83],[57,82],[51,70],[45,61],[36
,54],[28,47],[14,43],[0,41],[347,43],[334,47],[324,54],[316,61],[308,70],[304,80],[44,78],[38,70],[
34,80],[30,62],[20,58],[11,55],[0,54],[349,55],[339,58],[330,64],[322,70],[316,80],[26,86],[18,70]
,[9,67],[0,66],[350,67],[342,70],[334,76],[327,80],[24,85],[14,84],[8,81],[0,80],[352,81],[344,84],[
355,85]])
# cálculo dos cossenos diretores
        x=directioncosines(contagem)
        y=directioncosines(freq)
        n=len(x)
        xi= np.zeros((n,4))
        n=0
        for i in contagem:
            a,b,c=x[n]
            xi[n]=[a,b,c,0]
            n=n+1
        n=len(y)
        yi= np.zeros((n,4))
        n=0
        for i in freq:
            a,b,c=y[n]
            yi[n]=[a,b,c,i[2]]
            n=n+1
# cálculo das frequências aparentes
        c=0
        h=0
        cos=0
        for i in xi:
            for j in yi:
                c=math.fabs(i[0]*j[0]+i[1]*j[1]+i[2]*j[2])
                if c>0.99999:
                    c=1
                elif c<-0.99999:
                    c=-1
                else:
                    c=c
```

```

        cos=math.degrees(math.acos(c))
        h=h+math.fabs((j[3])/(math.sin(cos)))
    i[3]=h
    h=0
    for i in contagem:
        if i[0]+180>360:
            i[0]=i[0]-180
        else:
            i[0]=i[0]+180
        i[1]=90-i[1]
    n=len(xi)
    costagem=np.zeros((n,3))
    n=0
# resultado final
    for i in xi:
        a,b=contagem[n]
        costagem[n]=[a,b,i[3]]
        n=n+1
    return costagem

```

Anexo 18 – Rotina para cálculo de interseção

```
def intersec(dadosa,dadosb):
# dadosa=primeira família , dadosb= segunda família
    import numpy as np
    import math
# transformação em cossenos diretores
    for i in dadosa:
        if i[0]+180>360:
            i[0]=i[0]-180
        else:
            i[0]=i[0]+180
        i[1]=90-i[1]
        dircosa = np.array([[np.sin(azim) * np.cos(theta), np.cos(azim) * np.cos(theta),
np.sin(theta)] for azim, theta in np.radians(dadosa)])
    for i in dadosb:
        if i[0]+180>360:
            i[0]=i[0]-180
        else:
            i[0]=i[0]+180
        i[1]=90-i[1]
        dircosb = np.array([[np.sin(azim) * np.cos(theta), np.cos(azim) * np.cos(theta),
np.sin(theta)] for azim, theta in np.radians(dadosb)])
# calculo da linha de interseção
    intersec=[]
    for i in dircosa:
        for j in dircosb:
            intersec=intersec+[[i[1]*j[2]-i[2]*j[1],i[2]*j[0]-i[0]*j[2],i[0]*j[1]-i[1]*j[0]]]
# transformação em graus
    saida=[]
    a=0
    b=0
    for i in intersec:
        a, b = CalcSphere(i[0], i[1], i[2])
        saida=saida + [[a,b]]
    return saida
```

Anexo 19 – Rotina para cálculo de estatísticas

```
def estat(ad):
#ad=dados de dipdir/dip
    import numpy as np
    import math
    import csv
# transformação em cossenos diretores
    for i in ad:
        if i[0]+180>360:
            i[0]=i[0]-180
        else:
            i[0]=i[0]+180
        i[1]=90-i[1]
        dircos = np.array([[np.sin(azim) * np.cos(theta), np.cos(azim) * np.cos(theta),
np.sin(theta)] for azim, theta in np.radians(ad)])
# calculo da media
        x=0
        y=0
        z=0
        for i in dircos:
            x=x+i[0]
            y=y+i[1]
            z=z+i[2]
        n=len(dircos)
        xi=x/n
        yi=y/n
        zi=z/n
# resultante r
        r=math.sqrt(xi**2+yi**2+zi**2)
        xm=xi/r
        ym=yi/r
        zm=zi/r
# calculo do fisher K
        k=1/(1-r)
# dispersão
        p1=0.7
        p2=0.9
        cos1=1+(math.log(1-p1)/k)
        cos2=1+(math.log(1-p2)/k)
        disp1=math.degrees(math.acos(cos1))
        disp2= math.degrees(math.acos(cos2))
        return xm, ym, zm, k, disp1, disp2
```