Finite mixture model: decompose a cytogram into "*n*" sub adjacent asymmetric Gaussian bells. A Mathematica 9 script.

Andrea Butturini (abutturini@ub.edu)

Dep. of **Biologia Evolutiva, Ecologia i Ciències Ambientals**, Universitat de Barcelona. Avd. Diagonal 643 08028 Barcelona, Catalonia.

This script implements the Laplacian filter and the finite mixture approach to decompose a surface into an arbitrary number "n" of bi-dimensional asymmetric Gaussian probability distributions. Finite mixture are widely used in data mining or pattern recognition. Here, this tool in implemented to decompose an arbitrary surface f(x,y) with some peaks and shoulders. This approach can be useful to decompose any bidimensional data array.
This document describes how to use this approach to deconvolve a cytogram.

The script works with Mathematica (version 9 or latest) however the Laplacian filter and the finite mixture should be implement with any other mathematical software.

The scripts includes different steps:
1) To import the original cytogram and convert it into a matrix *f(x,y)* of dimension "n x n"
2) To detect all potential pics in *f(x,y)* with the laplacian filter, ∇^2, and create the list L_n that include the coordinate of each potential peak
3) To extract all *i* distinct proper subsets of L_n: $P(L_n)=\{\{\mu_1\}_1,\{\mu_2\}_2,\{\mu_3\}_3,\{\mu_1,\mu_2\}_4 \ldots\ldots \{\mu_1,\mu_2,\mu_{3,\ldots},\mu_n\}_i\}$ (where $i=2^n-1$)
3) To run the finite mixture model for each subset
4) To select the bet model with the optimal number of peak according the Bayesian Information Criterion (BIC) descriptor.

Deconvolution approach and model description

1) **Laplacian operator of $f_{(xy)}$ ($\nabla^2 f$)**

Local minima in the $\nabla_2 f$ surface provide the information about the number and position of potential peaks that might decompose the surface *f(x,y).*
A simple visual inspection of the f(x,y) provides a rapid identification of local maxima of the *f(x,y).*
To detect more subtle peaks, the $\nabla_2 f$ surface emerged as an interesting tool. The visual analysis of the local minima of the $\nabla_2 f$ surface allows to identify the position of these more subtle peaks. This step can be automated. In Mathematica the automatic search of the local minima can be performed with the built-in function "NMinimize" (Search method is the Nelder-Mead).
The following lines detailed the code.

2) **The asymmetric bi-dimensional Gaussian probability distribution**

According to the Finite Distribution mixture Modelling (FDM), the complex surface $f_{(x,y)}$ of a bivariate cytogram (i.e., FSC vs green fluorescence) is described as the sum of n subjacent peaks (eq. 1):

$$f_{(x,y)} = \sum_{i=1}^{n} c_{(x,y)_i} \qquad (1)$$

Each peak represents a subgroup that fits a predefined probabilistic density functions ($c_{(x,y)}$). Here, an asymmetric parameter (r) was incorporated into the Gaussian PDF probability model (36) in order to cope with asymmetries and long tails (37) (eq. 2):

$$c_{(x,y)_i} = a_i e^{-\left(\begin{cases} \frac{(\mu-\mu_i)^2}{2\sigma_i^2} & \text{if } \mu>\mu_i \\ \frac{(\mu-\mu_i)^2}{2r_i^2\sigma_i^2} & \text{otherwise} \end{cases}\right)} \qquad (2)$$

If the skewness r_i (r_{ix}, r_{iy}) is equal to the unity, equation 2 is equivalent to a Gaussian distribution defined by its mean μ_i (μ_{ix}, μ_{iy}), deviation σ_i (σ_{ix}, σ_{iy}), and height a_i (a_{ix}, a_{iy}).

Next, the code that uses the Mathematica language.

THIS FILE IMPLEMENT THE FINITE MIXTURE APPROACH TO DECONVOLVE A CYTOGRAM INTO "n" ASYMMETRIC BIDIMENSIONAL GAUSSIAN PEAKS

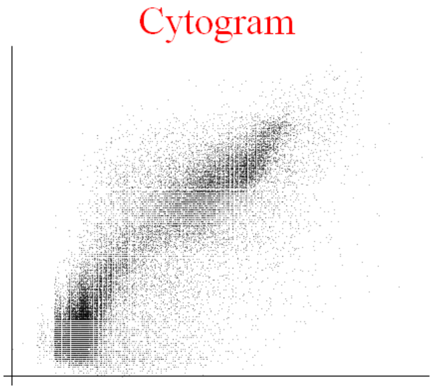
```
SAMPLE="T6";
 MESANY="april12"; (*The identification name of the sample of this example*)
```

Import the cytogram data from the file generated by the cytometer and create a raw-data table with the coordinate of each “no zero” value and save in a *txt file. The user fixes the type of the original data set and directory.

```
(*import the cytogram data from the file generated by the cytometer. In this example the data set has been generated by Apogee software*)

dati[SAMPLE_,MESANY_]:=Module[{ss},

ss=Import["F:\\CITOGRAMAS\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_LS1vsGreen.csv", "Data"];
   ss[[2;;-1]]
   ];
NdatimNOZero[{SAMPLE_,MESANY_}]:=Module[{datiNum,fff,jjj,ZZZ},
  datiNum=dati[SAMPLE,MESANY];
  fff=Drop[Flatten[Position[datiNum[[All,3]],Except[0]]],1];
  jjj=Map[datiNum[[#]]&, fff];
  ZZZ=Flatten[Table[Table[jjj[[n, {1,2}]],{jjj[[n,3]]}], {n,1,Length[jjj]}],1];

Export["F:\\CITOGRAMAS\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_LS1vsGreenNdatimNOZero.txt",ZZZ,"Table"]
  ] (*Generate a rawdata table with coordiante of each "no zero" value and save in a *txt file*)
```

Import the cytogram data from the file generated by the cytometer and create a rawdata table with the coordinate of each “no zero” value and save the rawdata matrix in a *txt file

```
CytogramPlot=ListPlot[Import["F:\\CITOGRAMAS\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_LS1vsGreenNdatimNOZero.txt", "Table"],PlotRange→All, PlotLabel→Style["Cytogram",20, Red],PlotStyle->{Black, Opacity[0.35],PointSize[0.0025]},Ticks→None, ImageSize→215, AspectRatio→0.8] (*A disporsion plot of the cytogram*)
```



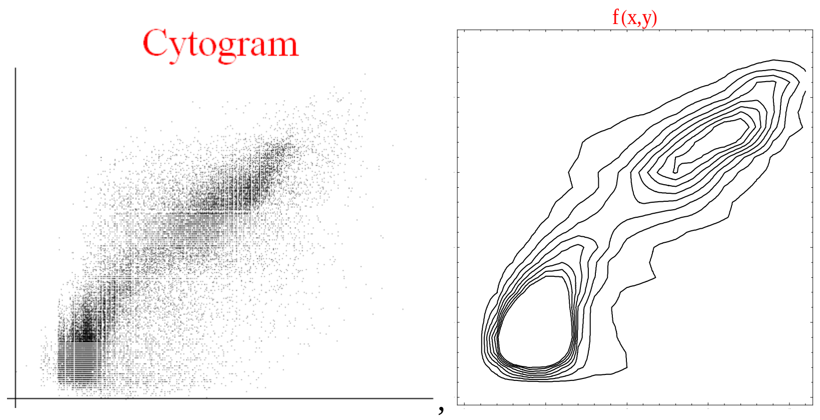
Steps to create a f(x, y) matrix from the cytogram data. In this example the dimension of the f(x,y) is 40 x 40

```
MBG=1024;(*Maximun binning grid of flow cytometer software (user specific parameter. It depends on used cytometer)*)
TABLEDATa[SAMPLE_,MESANY_]:=
 Module[{dat,EventValues,MatrDAT},

  dat=Module[{ss},

ss=Import["F:\\CITOGRAMAS\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_LS1vsGreen.csv", "Data"];(*import the cytogram data*)
    ss[[2;;-1]]
    ];

  EventValues[vv_]:=Module[{n},n=vv MBG;
    dat[[n+1;;n+MBG]][[All,3]]];
  MatrDAT=Table[EventValues[n], {n,0,MBG-1}];

Export["F:\\CITOGRAMAS\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_LS1vsGreenTABLE.txt",MatrDAT, "Table"] (*Convert the cytogram into a matrix 1024 x 1024 and save the matrix data into a *.txt file. User can change the names*)
  ]
TABLEDATa[SAMPLE,MESANY](*To create a the matrix 1024 x 1024 a save it in a *txt file*)

ImportDATTable[SAMPLE_,MESANY_]:=Take[Import["F:\\Citogramas\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_LS1vsGreenTABLE.txt",  "Table"], {50,850-1 },{50,850-1 }];
(*Import the  1024 x 1024 matrix and I star to reduce it. At this stage the matrix is 800 x 800The ranges “{50,850-1 },{50,850-1 }" are for to center the cytogram and to eliminate the perimeter of cytogram without information*)

int=Round[400 0.05]; (*VIGILAR AL 400!!!!*)
MATRIFIN00[SAMPLE_,MESANY_]:=Module[{PartTable},
```

```
 PartTable=Partition[ImportDATTable[SAMPLE,
MESANY],{int,int},int];ReplaceAll[Table[Plus@@Plus@@PartTable[[x,y]],
{x,1,PartTable//Length},{y,1,PartTable//Length}],x_?(#<0&)→0]] (*the final f(x,y) matrix of dimension 40
x 40 *)


ArrayExport[MOSTRA_,MESANY_]:=Export["F:\\Citogramas\\AllSamples_"<>MESANY<>"_TABLE_\\
"<>MOSTRA<>"_"<>MESANY<>"_LS1vsGreenARRAYDAT.txt",MATRIFIN00[MOSTRA,MESANY],
"Table" ]
(*Save the f(x,y) matrix in a *txt file*)

MATRIFIN[{SAMPLE_, MESANY_}]:=Module[{jj, cc},
 jj=Import["F:\\Citogramas\\AllSamples_"<>  MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>
MESANY<>"_LS1vsGreenARRAYDAT.txt","Table"];

 n=3;
 cc=ArrayPad[MovingAverage[jj,n],{{(n-1)/2},{0}}, "Extrapolation", InterpolationOrder→1];
 ReplaceAll[cc ,x_?(#<0&)→ 0]
 ]


F:\CITOGRAMAS\AllSamples_april12_TABLE_\T6_april12_LS1vsGreenTABLE.txt
```

A contour Plot the the f(x,y) matrix and compare with the original cytogram

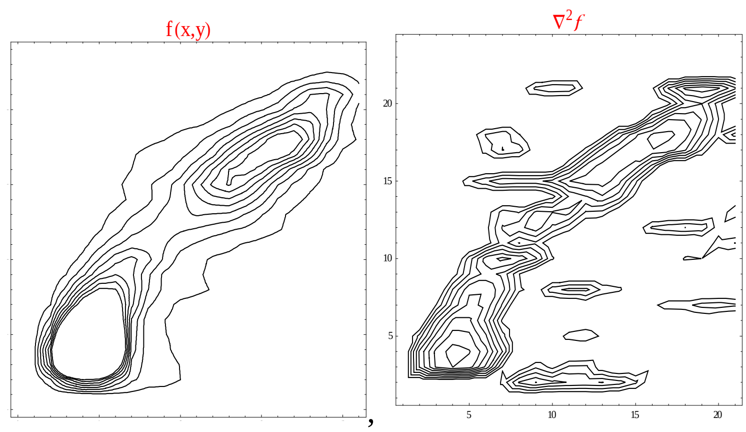
```
LCPfxy=ListContourPlot[MATRIFIN[{SAMPLE,MESANY}][[3;;23,12;;35]]//Transpose,InterpolationOrd
er→1,Contours→8,ContourStyle→Black,ContourShading→None, PlotRange→Automatic , Frame→True,
FrameStyle→Directive[Black,0],PlotLabel→Style["f(x,y)",20, Red],Ticks→None];(*a Contour Plot of
f(x,y)*)


Grid[{{CytogramPlot,LCPfxy}}]
```



Steps to create a the laplacian $\nabla_2 f$ of f(x, y) matrix. This is a new matrix saved in a *txt file.

Local minima in the $\nabla^2 f$ surface provide the information about the number and position of potential peaks that might decompose the surface f(x,y).
A simple visual inspection of the f(x,y) provides a rapid identification of local maxima of the f(x,y). To detect more subtle peaks, the $\nabla^2 f$ surface emerged as an useful tool. The visual analysis of the local minima of the $\nabla^2 f$ surface allows to identify the position of these more subtle peaks. However, this step can be automated. In Mathematica this automatic search of the local minima with the built-in function "NMinimize" (Search method is the Nelder-Mead).

```
LAplacianMATRIFIN[{SAMPLE_, MESANY_
}]:=Module[{EEM,ExpandMAEEM,expandedndegree,ExpandEEM,degreeMovingverage,LIEEM},
  EEM=ReplaceAll[LaplacianGaussianFilter[ MATRIFIN[{SAMPLE,
MESANY}],1(*pixelRadius*)],x_?(#>0&)→0];
  expandedndegree=1/5;
  LIEEM=ListInterpolation[EEM, InterpolationOrder→1];
  ExpandEEM=Table[LIEEM[y,x],
{y,1,Dimensions[EEM][[1]],expandedndegree},{x,1,Dimensions[EEM][[2]],expandedndegree}];
  (*ExpandEEM//Dimensions*)
  degreeMovingverage=Round[If[OddQ[Round[1/expandedndegree]]==False,1/expandedndegree-
1,1/expandedndegree]];

  ExpandMAEEM=MovingAverage[ExpandEEM,degreeMovingverage];

  Take[ArrayPad[ExpandMAEEM,{{(degreeMovingverage-1)/2},{0}}, "Extrapolation",
InterpolationOrder→1],{1,-1,1/expandedndegree},{1,-1,1/expandedndegree}]
  ]

ExportLaplacian[{SAMPLE_, MESANY_}]:=Export[
"F:\\Citogramas\\AllSamples_"<>MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_"<>"Seco
ndDerivARRAYDAT_ROUNDED_.txt",LAplacianMATRIFIN[{SAMPLE, MESANY}],"Table" ]

 LAplacianPlot=ListContourPlot[Log[{1-LAplacianMATRIFIN[{SAMPLE, MESANY,
TRESHOLD}][[3;;23,12;;35]]//Transpose}], PlotRange→{10,1.0}, Contours→12, ContourShading→None,
ContourStyle→Black, Ticks→None, PlotLabel→Style["∇2𝑓",20, Red]];
Grid[{{LCPfxy,LAplacianPlot}}]
```

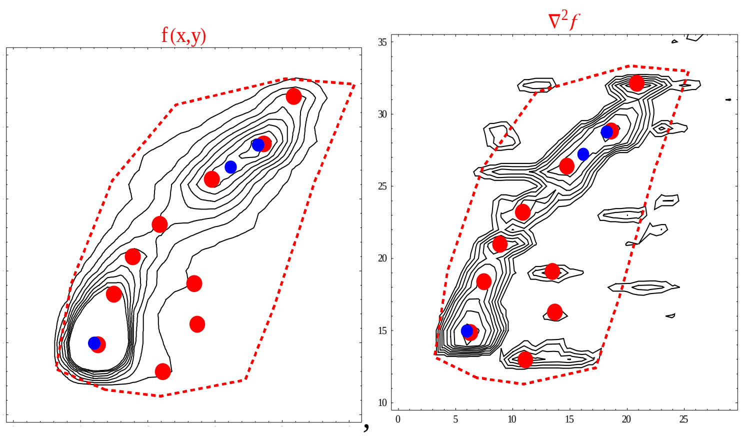


In this example the coordinates of the potential peaks (L_n) are:

```
MAXPEAKS0={{14.98,6.012},{28.76,18.23},{27.22,16.18}}; (*these are the local maxima in the f(x,y)
matrix *)
LOCALMINIMA0={{14.87,6.295},{18.39,7.469},{20.98,8.879},{26.38,14.75},{28.84,18.63},{32.13,20.8
6},{23.21,10.88},{12.99,11.11},{19.1,13.46},{16.28,13.69}};(*these are the local minima in the ∇^2𝑓
matrix *)

Boundary={{3.157,13.17},{4.295,19.12},{7.357,26.27},{12.08,31.54},{20.22,33.35},{25.38,32.97},{22.67
,26.72},{22.5,26.42},{19.35,17.38},{17.25,12.42},{10.95,11.29},{6.832,11.74}};
```

(*Here the boundary of the ∇^2𝑓 surface are defined. Eventual peaks that apepread outside these limit are considered. The user can change or remove the boundary*)

```
Pl1=Show[ListContourPlot[MATRIFIN[{SAMPLE,MESANY}]//Transpose,InterpolationOrder→1,Contours→8,ContourStyle→Black,ContourShading→None, PlotRange→{0,300}, Frame→True,
FrameStyle→Directive[Black,0],PlotLabel→Style["f(x,y)",20, Red],Ticks→None],
  ListPlot[Map[Reverse,LOCALMINIMA0],PlotStyle→{Red, PointSize[0.045]}],
  ListPlot[Map[Reverse,MAXPEAKS0],PlotStyle→{Blue, PointSize[0.035]}],
  Graphics[{

    FaceForm[None],EdgeForm[Directive[Dashed,Thick,Red]],Polygon[ Boundary]}],
  PlotRange→{10,35}
  ];
Pl2=Show[ListContourPlot[Log[{1-LAplacianMATRIFIN[{SAMPLE, MESANY,
TRESHOLD}]//Transpose}], PlotRange→{10,1.0}, Contours→12, ContourShading→None,
ContourStyle→Black, Ticks→None, PlotLabel→Style["∇2𝑓",20, Red]],
  ListPlot[Map[Reverse,LOCALMINIMA0],PlotStyle→{Red, PointSize[0.045]}],
  ListPlot[Map[Reverse,MAXPEAKS0],PlotStyle→{Blue, PointSize[0.035]}],
  Graphics[{

    FaceForm[None],EdgeForm[Directive[Dashed,Thick,Red]],Polygon[ Boundary]}],
  PlotRange→{10,35}
  ];
Grid[{{Pl1,Pl2}}]
```



In this example some local maxima and local minima nearly overlaps. Therefore in the definitive list of the potential peaks (L_n) the duplicate coordinates are removed.In the example the definitive list of L_n included 11 potential peaks

```
LISTAUNION=Module[{BSFt,RADIInfuelncia},
 RADIInfuelncia=1.5;
 NCD=
  Module[{yy,yy2,re,filt,filt2,res3,res2,res,ser0,ser1,ser2,INIT,FINAL,FOLLW},
  res= Join[MAXPEAKS0,LOCALMINIMA0];
  Off[Part::"partw"];
  BSFt[{y_,x_}]:=ListInterpolation[matr, InterpolationOrder→1, Method→"Spline"][x,y];
  ser0=Table[Nearest[res,res[[n]],{Infinity,RADIInfuelncia}],{n,1,Length[res]}];
     yy=Table[Max[Map[ BSFt,ser0[[n]]]]  ,{n,1,Length[ser0]}];
  yy2=DeleteDuplicates[yy];
  re=Map[BSFt, res];
```

```
    filt=Flatten[Table[Position[re, yy2[[n]]],{n,1,Length[re]}] ];
        res2=res[[filt]];
    Label[INIT];
    ser0=Table[Nearest[res2,res2[[n]],{Infinity,RADIInfuelncia}],{n,1,Length[res2]}];
    yy=Table[Max[Map[ BSFt,ser0[[n]]]]  ,{n,1,Length[ser0]}];
    yy2=DeleteDuplicates[yy];
    re=Map[BSFt, res2];
    filt2=Flatten[Table[Position[re, yy2[[n]]],{n,1,Length[re]}] ];
    res3=res2[[filt2]];
    If[Length[filt]>Length[filt2],  Goto[FOLLW], Goto[FINAL]];
    Label[FOLLW];
    res2=res3;
    filt=filt2;
    Goto[INIT];
    Label[FINAL];
    (* Complement[ N[res2[[filt2]]]//Round  ,EXCLUSAREA]*)
     N[res2[[filt2]]]
    ];
  NCD[[Flatten[Position[Map[BSFt,NCD],x_?(#> 0.&)]]]]
  ] (*the definitive list of potenial peak in the f(x,y)*)

Export[ "F:\\CITOGRAMAS\\AllSamples_"<>  MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>
MESANY<>"_CandidatePEaksROUNDED.txt",LISTAUNION,"Table"]
```

The following code lines are to automatize the search of local maxima in f(x,y) and local minima in $\nabla_2 f$.

These function can substitute the visual search of the potential peaks (L_n) described previously.

```
(*
matr=MATRIFIN[{SAMPLE, MESANY, }];
secDeriv=LAplacianMATRIFIN[{SAMPLE, MESANY }];
{LINEES,COLUMNES}=Dimensions[matr]
 MULTIPL=10;
{ ExpandRat,ShrinkRat,ContractRat,ReflectRat}={2,.5,.5,1};

MATRINTERP=ListInterpolation[ matr, InterpolationOrder→1];
secDerivINTERP=ListInterpolation[secDeriv, InterpolationOrder→1];


MAXPEAKS0bis=Module[{yyy0,iii,PPP,PPP2,eeee, p1,POINTSEARCH},
yyy0=Table[ NMaximize[{MATRINTERP[x,y],1≤y≤COLUMNES-1&&1≤x≤LINEES-
1},{{y,1,COLUMNES},{x,1,LINEES}},Method→{"NelderMead","ExpandRatio"→ ExpandRat,
"ShrinkRatio"→ShrinkRat,"ContractRatio"→ContractRat,"ReflectRatio"→ReflectRat,
"RandomSeed"→i}, WorkingPrecision→10],{i, 2 MULTIPL Max[COLUMNES,LINEES]}][[All,2]];


iii=Transpose[{yyy0[[All,1,2]],yyy0[[All,2,2]]}];

PPP=Union[Round[iii]];
PPP2=Table[Nearest[iii, PPP[[n]]], {n,Length[PPP]}];
eeee=Table[{Mean[PPP2[[n,All,1]]], Mean[PPP2[[n,All,2]]]}, {n, 1, Length[PPP2]}];

POINTSEARCH[PoligonVertex_, PUNTO_]:=Module[{r,area2,angtest,testpoint,p,pt},
(* RotationMatrix[π/2] for y sign change and subsequent x,y Swap *)
                    r:={{0,-1},{1,0}};
                    (* calculate polygon signed area *)
```

```
                    area2[pts_]:=1/2 Total[pts (pts//RotateRight[r.#&/@#]&),2];
                    (* test for positive angle sign between two 2D vectors *)
                    angtest[p1:{x1_,y1_},p2:{x2_,y2_}]:=p1.r.p2>0;
                    (* test if point pt inside convex polygon poly *)
                    testpoint[poly_,pt_]:=
                              (* translate poly such that pt becomes origin *)
                              poly//#-pt&/@#&//
                              (* pt is inside if adjacent vertex vector pair angles are all positive or all negative *)
                              {#,RotateLeft@#}&//Transpose//angtest@@#&/@#&//Equal@@#&;

With[{z=PoligonVertex,pz= PUNTO},
                    If[testpoint[z,pz],1,0]]
];
 p1=eeee[[Flatten[Position[Map[POINTSEARCH[POLIGON,#]&,eeee],1]]]]
];



LOCALMINIMA0bis=Module[{yyy0,iii,PPP,PPP2,eeee, p1,POINTSEARCH},
yyy0=Table[ NMinimize[{secDerivINTERP[x,y],1≤y≤COLUMNES-1&&1≤x≤LINEES-
1},{{y,1,COLUMNES},{x,1,LINEES}},Method→{"NelderMead","ExpandRatio"→ ExpandRat,
"ShrinkRatio"→ShrinkRat,"ContractRatio"→ContractRat,"ReflectRatio"→ReflectRat,
"RandomSeed"→i}, WorkingPrecision→10],{i, 2 MULTIPL Max[COLUMNES,LINEES]}][[All,2]];


iii=Transpose[{yyy0[[All,1,2]],yyy0[[All,2,2]]}];

PPP=Union[Round[iii]];
PPP2=Table[Nearest[iii, PPP[[n]]], {n,Length[PPP]}];
eeee=Table[{Mean[PPP2[[n,All,1]]], Mean[PPP2[[n,All,2]]]}, {n, 1, Length[PPP2]}];

POINTSEARCH[PoligonVertex_, PUNTO_]:=Module[{r,area2,angtest,testpoint,p,pt},
                    r:={{0,-1},{1,0}};
                    area2[pts_]:=1/2 Total[pts (pts//RotateRight[r.#&/@#]&),2];
                    angtest[p1:{x1_,y1_},p2:{x2_,y2_}]:=p1.r.p2>0;
                    testpoint[poly_,pt_]:=
                              poly//#-pt&/@#&//
                              {#,RotateLeft@#}&//Transpose//angtest@@#&/@#&//Equal@@#&;
With[{z=PoligonVertex,pz= PUNTO},
                    If[testpoint[z,pz],1,0]]
];
 p1=eeee[[Flatten[Position[Map[POINTSEARCH[POLIGON,#]&,eeee],1]]]]
];
*)
```

A list of subsets (P(L_n)) of L_n is generated in This example the length of the P(L_n) is 2047 combinations (2n-1, n=11)

```
ImportPotentialPeaks[{SAMPLE_, MESANY_}]:=Import[ "F:\\CITOGRAMAS\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_CandidatePEaksROUNDED.txt","Table"]

AllSubSets=Reverse[Drop[Subsets[ImportPotentialPeaks[{SAMPLE,MESANY}]],1]] (*the list
Subscript[L, n] is created*);
```

The finite mixing model decomposition code.

```
DECOMPOMPOSITION[{SAMPLE_, MESANY_,SUBSET_ (*desde 0 fins al num max de coordinates del
ImportPotentialPeaks*),ITERATIONMAX_}]:=Module[{EEM,LINEES,COLUMNES,coord,EEMXY,COORDCompleto,NEWCOORINATEASDEFINITIVAS,NWCinputStepA,AsymBidnormalDistrib,ABD,ADPARAMETERS,COST,
datReal,StepB,RES,yyyy,LIEEM,IP,POTCOORD,COORDSubSets,INPINDISPENSABLES(*,ImportPotentialPeaks,AllSubSets*)},

  Needs["MultivariateStatistics`"];
  EEM=MATRIXUMBRAL[{SAMPLE, MESANY, TRESHOLD}];
  {LINEES,COLUMNES}=Dimensions[EEM];
  LIEEM=ListInterpolation[EEM,  Method→"Spline"];

  coord=Table[{y,x},{x,1,Dimensions[EEM][[1]],1},{y,1,Dimensions[EEM][[2]],1}];
   EEMXY=Table[Flatten[{coord[[n,m]],EEM[[n,m]]}],
{m,1,Dimensions[coord][[2]]},{n,1,Dimensions[coord][[1]]}];

  COORDCompleto=Flatten[ EEMXY,1][[All,{1,2}]];

  (*ImportPotentialPeaks =Import[ "D:\\Citogramas\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_CandidatePEaksROUNDED.txt","Table"];

  AllSubSets=Reverse[Drop[Subsets[ImportPotentialPeaks ],1]];*)

  NEWCOORINATEASDEFINITIVAS=AllSubSets[[SUBSET]];
  NWCinputStepA=AllSubSets[[SUBSET]];



  AsymBidnormalDistrib[{aax_,aay_,Kost_, sigmax_,sigmay_ ,alfax_, alfay_}]:=Kost
```

$$e^{-\left(\mathrm{If}\left[x\leq aax,\frac{(x-aax)^2}{2\,sigmax^2},\frac{(x-aax)^2}{2\,alfax^2\,sigmax^2}\right]+\mathrm{If}\left[y\leq aay,\frac{(y-aay)^2}{2\,sigmay^2},\frac{(y-aay)^2}{2\,alfay^2\,sigmay^2}\right]\right)};$$

```
  ABD[n_]:=Module[{iu},
    iu=ToString[n];
    AsymBidnormalDistrib[{
      Symbol["aax"<>iu],Symbol["aay"<>iu],Symbol["Kost"<>iu],
Symbol["sigmax"<>iu],Symbol["sigmay"<>iu] ,Symbol["alfax"], Symbol["alfay"]}
     ]
    ];

  ADPARAMETERS[VALINIT_ ]:=Module[{ParDistrNorm },

    ParDistrNorm=Module[{pars,ParsL, rrr},
      rrr=Length[Transpose[VALINIT][[1]]];
      pars[x_]:={Symbol["aax"<>ToString[x] ],Symbol["aay"<>ToString[x] ],
Symbol["sigmax"<>ToString[x] ], Symbol["sigmay"<>ToString[x] ],Symbol["Kost"<>ToString[x]],
        Symbol["alfax"],
        Symbol["alfay"]
        };
      ParsL = Sort[Flatten[Map[pars,  Range[rrr]]]];
      Join[Transpose[{Take[ParsL , 2 rrr],Flatten[Transpose[VALINIT]]}],Take[ParsL , -5rrr]]
      ]];
```

```
  COST[nPICOS_ ]:=Module[{pars,ParsL,rrr},
    rrr=nPICOS;
    pars[x_]:=Symbol["Kost"<>ToString[x]];
    ParsL = Sort[Flatten[Map[pars,  Range[rrr]]]]
    ] (*CONDICION QUE LOS CONST sean >0*);
   datReal=Flatten[EEMXY,1];

  StepB=Module[{MODELO,COSTNEGATIVESPOSITION ,begin,end,follow,PARAMINITIALS},
    Label[begin];

    PARAMINITIALS=Union[Table[Symbol["aax"<>ToString[n]]-
>NEWCOORINATEASDEFINITIVAS[[n,1]], {n,Length[NEWCOORINATEASDEFINITIVAS] }],
      Table[Symbol["aay"<>ToString[n]]->NEWCOORINATEASDEFINITIVAS[[n,2]],
{n,Length[NEWCOORINATEASDEFINITIVAS] }]] ;


    MODELO=Plus@@Map[ABD,Range[1,Length[NEWCOORINATEASDEFINITIVAS ]]]/.
PARAMINITIALS;

    yyyy=
NonlinearModelFit[datReal,MODELO,DeleteDuplicates[Drop[ADPARAMETERS[NEWCOORINATEAS
DEFINITIVAS], Length[NEWCOORINATEASDEFINITIVAS]*2
]],{x,y},MaxIterations→ITERATIONMAX,PrecisionGoal→∞,ConfidenceLevel→.999,
(*NormFunction→(Norm[#, Infinity]&),*)AccuracyGoal→1000];

    COSTNEGATIVESPOSITION=Module[
      {yty,val,elem, rr},
      yty=Position[Partition[Drop[yyyy["BestFitParameters"],2],
Length[NEWCOORINATEASDEFINITIVAS]][[1, All,2]], x_?(#<0&)] ;


      (*Individuio si hi han pics negatius!!!*)
      val= COST[Length[NEWCOORINATEASDEFINITIVAS]][[Flatten[yty]]];

      rr=Table[
        elem= Drop[Characters[ToString[val[[v]]]] ,4] ;
        If[Length[elem]<2, ToExpression[elem[[1]]],10ToExpression[elem[[-2]]] +ToExpression[elem[[-1]]] ],
{v,1,Length[val]}];
      Partition[rr,1]
      ]
    If [Length[COSTNEGATIVESPOSITION]≥  1, Goto[follow], Goto[end]];
    Label[follow];
        NEWCOORINATEASDEFINITIVAS=Delete[

NEWCOORINATEASDEFINITIVAS,COSTNEGATIVESPOSITION]  ;

        Goto[begin];
    Label[end];
         yyyy
    ];

  RES=Flatten[Join[{SAMPLE,
      SUBSET,
      Length[AllSubSets[[SUBSET]]],
      Length[NWCinputStepA],
```

```
    Length[NEWCOORINATEASDEFINITIVAS],NEWCOORINATEASDEFINITIVAS,
    Map[StepB, {"AdjustedRSquared","BIC"}]}]];
  Export["F:\\CITOGRAMAS\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"SummStat_ROUNDED_"<>ToString[Length[NWCinputStepA]]<>"_"<>ToString[ SUBSET]<>"_.txt", RES, "Table"];
  RES

  ]

BESTDECOMPOMPOSITIONVisualization[{SAMPLE_, MESANY_,SUBSET_ (*desde 0 fins al num max de coordinates del ImportPotentialPeaks*),ITERATIONMAX_}]:=Module[{EEM,LINEES,COLUMNES,coord,EEMXY,LIEEM,COORDCompleto,NEWCOORINATEASDEFINITIVAS,NWCinputStepA,AsymBidnormalDistrib,ABD,ADPARAMETERS,COST, datReal,StepB, RES,yyyy(*,ImportPotentialPeaks,AllSubSets*)},

  Needs["MultivariateStatistics`"];
  EEM=MATRIXUMBRAL[{SAMPLE, MESANY, TRESHOLD}];
  {LINEES,COLUMNES}=Dimensions[EEM];
  LIEEM=ListInterpolation[EEM,  Method→"Spline"];

  coord=Table[{y,x},{x,1,Dimensions[EEM][[1]],1},{y,1,Dimensions[EEM][[2]],1}];
   EEMXY=Table[Flatten[{coord[[n,m]],EEM[[n,m]]}],
{m,1,Dimensions[coord][[2]]},{n,1,Dimensions[coord][[1]]}];

  COORDCompleto=Flatten[ EEMXY,1][[All,{1,2}]];

  (*ImportPotentialPeaks =Import[ "D:\\Citogramas\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_CandidatePEaksROUNDED.txt","Table"];

  AllSubSets=Reverse[Drop[Subsets[ImportPotentialPeaks ],1]];*)

  NEWCOORINATEASDEFINITIVAS=AllSubSets[[SUBSET]];
  NWCinputStepA=AllSubSets[[SUBSET]];



  AsymBidnormalDistrib[{aax_,aay_,Kost_, sigmax_,sigmay_ ,alfax_, alfay_}]:=Kost
```

$$e^{-\left(\mathrm{If}\left[x\leq aax,\frac{(x-aax)^2}{2\,sigmax^2},\frac{(x-aax)^2}{2\,alfax^2\,sigmax^2}\right]+\mathrm{If}\left[y\leq aay,\frac{(y-aay)^2}{2\,sigmay^2},\frac{(y-aay)^2}{2\,alfay^2\,sigmay^2}\right]\right)};$$

```
  (*The asymmetric bidimensional Gaussian distribution model*)

 ABD[n_]:=Module[{iu},
   iu=ToString[n];
   AsymBidnormalDistrib[{
    Symbol["aax"<>iu],Symbol["aay"<>iu],Symbol["Kost"<>iu],
Symbol["sigmax"<>iu],Symbol["sigmay"<>iu] ,Symbol["alfax"], Symbol["alfay"]}
    ]
   ];


  ADPARAMETERS[VALINIT_ ]:=Module[{ParDistrNorm },

   ParDistrNorm=Module[{pars,ParsL, rrr},
    rrr=Length[Transpose[VALINIT][[1]]];
```

```
     pars[x_]:={Symbol["aax"<>ToString[x] ],Symbol["aay"<>ToString[x] ],
Symbol["sigmax"<>ToString[x] ], Symbol["sigmay"<>ToString[x] ],Symbol["Kost"<>ToString[x]],
       Symbol["alfax"],
       Symbol["alfay"]
       };
     ParsL = Sort[Flatten[Map[pars,  Range[rrr]]]];
     Join[Transpose[{Take[ParsL , 2 rrr],Flatten[Transpose[VALINIT]]}],Take[ParsL , -5rrr]]
     ]];

  COST[nPICOS_ ]:=Module[{pars,ParsL,rrr},
    rrr=nPICOS;
    pars[x_]:=Symbol["Kost"<>ToString[x]];
    ParsL = Sort[Flatten[Map[pars,  Range[rrr]]]]
    ] (*CONDICION QUE LOS CONST sean >0*);
   datReal=Flatten[EEMXY,1];

  StepB=Module[{MODELO,COSTNEGATIVESPOSITION ,begin,end,follow,PARAMINITIALS},
    Label[begin];

    PARAMINITIALS=Union[Table[Symbol["aax"<>ToString[n]]-
>NEWCOORINATEASDEFINITIVAS[[n,1]], {n,Length[NEWCOORINATEASDEFINITIVAS] }],
     Table[Symbol["aay"<>ToString[n]]->NEWCOORINATEASDEFINITIVAS[[n,2]],
{n,Length[NEWCOORINATEASDEFINITIVAS] }]] ;


    MODELO=Plus@@Map[ABD,Range[1,Length[NEWCOORINATEASDEFINITIVAS ]]]/.
PARAMINITIALS;


    yyyy=
NonlinearModelFit[datReal,MODELO,DeleteDuplicates[Drop[ADPARAMETERS[NEWCOORINATEAS
DEFINITIVAS], Length[NEWCOORINATEASDEFINITIVAS]*2
]],{x,y},MaxIterations→ITERATIONMAX,PrecisionGoal→∞,ConfidenceLevel→.999,
(*NormFunction→(Norm[#, Infinity]&),*)AccuracyGoal→1000];

    COSTNEGATIVESPOSITION=Module[
     {yty,val,elem, rr},
     yty=Position[Partition[Drop[yyyy["BestFitParameters"],2],
Length[NEWCOORINATEASDEFINITIVAS]][[1, All,2]], x_?(#<0&)] ;


     (*Individuio si hi han pics negatius!!!*)
     val= COST[Length[NEWCOORINATEASDEFINITIVAS]][[Flatten[yty]]];

     rr=Table[
       elem= Drop[Characters[ToString[val[[v]]]] ,4] ;
       If[Length[elem]<2, ToExpression[elem[[1]]],10ToExpression[elem[[-2]]] +ToExpression[elem[[-1]]] ],
{v,1,Length[val]}];
     Partition[rr,1]
     ]

    If [Length[COSTNEGATIVESPOSITION]≥  1, Goto[follow], Goto[end]];
    Label[follow];
```

```
        NEWCOORINATEASDEFINITIVAS=Delete[
NEWCOORINATEASDEFINITIVAS,COSTNEGATIVESPOSITION] ;
        Goto[begin];
    Label[end];
         yyyy
    ];

  RES=Flatten[Join[{SAMPLE,
      SUBSET,
      Length[AllSubSets[[SUBSET]]],
      Length[NWCinputStepA],
      Length[NEWCOORINATEASDEFINITIVAS],NEWCOORINATEASDEFINITIVAS,
      Map[StepB, {"AdjustedRSquared","BIC"}]}]];

  Export["F:\\CITOGRAMAS\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_SummStat_ROUNDED_"<>ToString[
SUBSET]<>"_.txt", RES, "Table"];

  ME={StepB, NEWCOORINATEASDEFINITIVAS,datReal};



  MEsel=ME[[1]];
  NEWCOORINATEASDEFINITIVAS=ME[[2]];
  datReal=ME[[3]];
  (*Print[MEsel["BestFitParameters"]];*)

  ObsVsSim=Transpose[MEsel[{"Response","PredictedResponse"}]];

Export["F:\\CITOGRAMAS\\AllSamples_"<>SAMPLE<>"_"<>MESANY<>"_PLOT_SUMMARY_ObsV
sSimul_"<>ToString[Length[SUBSET]]<>"_.txt",ObsVsSim,"Table"];


LPPredictedVsObserved=ListPlot[ObsVsSim,FrameLabel→{"observed","predicted"},Frame→True,Axes→F
alse, PlotRange→All,
    PlotLabel→Style[Map[StepB, {"AdjustedRSquared","BIC"}], 15,Red,Bold]];
  MaxSimulat=Max[MEsel[ "PredictedResponse"]];
  AreaTotSimulat=Plus@@MEsel[ "PredictedResponse"];

  DATSIMULAT=Transpose[Append[Transpose[datReal[[All,{1,2}]]],MEsel[ "PredictedResponse"]]];
  CONTOURPLOTSIMULAT=Show[ListContourPlot[DATSIMULAT, PlotLabel→"Simulated"], ListPlot[
NEWCOORINATEASDEFINITIVAS, PlotStyle→{Red, PointSize[0.05]}]];


  PEAKSCANDIDATSORIG=NWCinputStepA;

  CONTOURPLOTOBSERV=Show[ListContourPlot[datReal
,PlotLabel→Style["Sample="<>SAMPLE<>"_"<>MESANY, 15,Red,Bold]], ListPlot[
NEWCOORINATEASDEFINITIVAS, PlotStyle→{Red, PointSize[0.04]}] ,
    ListPlot[  PEAKSCANDIDATSORIG , PlotStyle→{Black, PointSize[0.025]}]
    ];
```

```
PLOTS=GraphicsGrid[{{CONTOURPLOTOBSERV,CONTOURPLOTSIMULAT,LPPredictedVsObserve
d}}];
  Print[PLOTS];

  ME

  ]
VISUALSIMULATBestRandomSearch[ {SAMPLE_, MESANY_,SUBSET_,ITERATIONMAX_ }
]:=Module[{ME,
MEsel,NEWCOORINATEASDEFINITIVAS,ObsVsSim,LPPredictedVsObserved,MaxSimulat,AreaTotSim
ulat,CONTOURPLOTSIMULAT,PEAKSCANDIDATSORIG,CONTOURPLOTOBSERV,PLOTS,PARA
MINITIALSfinals,AsymBidnormalDistrib(*,MATRIXCURVESNDIVIDUALS*),SIMMATRIX,param,lab
el,TABLESUMMARYindividualPeaks,datReal},


  ME=BESTDECOMPOMPOSITIONVisualization[{SAMPLE, MESANY,SUBSET,ITERATIONMAX}];

  MEsel=ME[[1]];
  NEWCOORINATEASDEFINITIVAS=ME[[2]];
  datReal=ME[[3]];
  (*Print[MEsel["BestFitParameters"]];*)

  ObsVsSim=Transpose[MEsel[{"Response","PredictedResponse"}]];
  Export[ "F:\\CITOGRAMAS\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_SUMMARY_BestRandom_Search_ObsVs
Simul_ROUNDED_.txt",ObsVsSim,"Table"];


LPPredictedVsObserved=ListPlot[ObsVsSim,FrameLabel→{"observed","predicted"},Frame→True,Axes→F
alse,
    PlotLabel→Style["   # Peaks = "<>ToString[Length[ NEWCOORINATEASDEFINITIVAS]],
15,Red,Bold]];
  MaxSimulat=Max[MEsel[ "PredictedResponse"]];
  AreaTotSimulat=Plus@@MEsel[ "PredictedResponse"];


CONTOURPLOTSIMULAT=Show[ListContourPlot[Transpose[Append[Transpose[datReal[[All,{1,2}]]],
MEsel[ "PredictedResponse"]]], PlotLabel→"Simulated"], ListPlot[ NEWCOORINATEASDEFINITIVAS,
PlotStyle→{Red, PointSize[0.05]}]];

  PEAKSCANDIDATSORIG=Import[SAMPLE<>"_"<>
MESANY<>"_CandidatePEaksROUNDED.txt","Table"];

  CONTOURPLOTOBSERV=Show[ListContourPlot[datReal ,PlotLabel→Style["Sample= "<>SAMPLE,
15,Red,Bold]], ListPlot[ NEWCOORINATEASDEFINITIVAS, PlotStyle→{Red, PointSize[0.04]}] ,
    ListPlot[Map[Reverse,PEAKSCANDIDATSORIG], PlotStyle→{Black, PointSize[0.025]}]
    ];


PLOTS=GraphicsGrid[{{CONTOURPLOTOBSERV,CONTOURPLOTSIMULAT,LPPredictedVsObserve
d}}];
  (*Print[PLOTS];*)
```

```
 Export["F:\\CITOGRAMAS\\AllSamples_"<>  MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>
MESANY<>"_PLOT_SUMMARY_BestRandom_Search_EXTENDED_ROUNDED_.jpg",PLOTS,"JPEG
"];

 PARAMINITIALSfinals=Union[Table[Symbol["aax"<>ToString[n]]-
>NEWCOORINATEASDEFINITIVAS[[n,2]], {n,Length[NEWCOORINATEASDEFINITIVAS] }],
   Table[Symbol["aay"<>ToString[n]]->NEWCOORINATEASDEFINITIVAS[[n,1]],
{n,Length[NEWCOORINATEASDEFINITIVAS] }]] ;
 AsymBidnormalDistrib[{aax_,aay_,Kost_, sigmax_,sigmay_ ,alfax_, alfay_}]:=Kost
```

$$e^{-\left(\text{If}\left[x\leq aax,\frac{(x-aax)^2}{2\,sigmay^2},\frac{(x-aax)^2}{2\,alfay^2\,sigmay^2}\right]+\text{If}\left[y\leq aay,\frac{(y-aay)^2}{2\,sigmax^2},\frac{(y-aay)^2}{2\,alfax^2\,sigmax^2}\right]\right)};$$

```
 (*The asymmetric bidimensional Gaussian model*)

 ABD[n_]:=Module[{iu},
   iu=ToString[n];
   AsymBidnormalDistrib[{
     Symbol["aax"<>iu],Symbol["aay"<>iu],Symbol["Kost"<>iu],
Symbol["sigmax"<>iu],Symbol["sigmay"<>iu] ,Symbol["alfax"], Symbol["alfay"]}
    ]
   ];
 ZZ=Join[MEsel["BestFitParameters"], PARAMINITIALSfinals];
 (*Print[ZZ];*)


 MATRIXCURVESNDIVIDUALS=Table[
   ss[{y_,x_}]:=Evaluate[{y,x,Evaluate[ABD[n]/.ZZ]}];

    Map[ss,datReal[[All,1;;2]]] ,
   {n,1,Length[NEWCOORINATEASDEFINITIVAS]}];



 Print[Table[
   Show[CONTOURPLOTSIMULAT,
    ListContourPlot[ MATRIXCURVESNDIVIDUALS[[n]], PlotRange→All(*{0,
MaxSimulat/3}*),ContourShading→None,ContourStyle→{{Black,Thickness[0.007]}}]],
   {n,1,Length[NEWCOORINATEASDEFINITIVAS]}]];

 SIMMATRIX=Transpose[{MATRIXCURVESNDIVIDUALS[[1,All,2]],
MATRIXCURVESNDIVIDUALS[[1,All,1]],Plus@@MATRIXCURVESNDIVIDUALS[[1;;-1,All,3]]}];

 Export[ "F:\\CITOGRAMAS\\AllSamples_"<>  MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>
MESANY<>"_Simulat_PercentIntegrat_BestRandom_Search_ROUNDED_.txt",SIMMATRIX,"Table"];

 (*Print[ListContourPlot[SIMMATRIX, PlotLabel→"Simlated"]];*)

 param[PARAM_, n_]:=Module[
   {rr,SBBestPAram},rr=Position[ MEsel["BestFitParameters"],If[n==0,
Symbol[PARAM],Symbol[PARAM<>ToString[n]]]]]+{{0,1}};
   SBBestPAram[{z_,w_}]:=MEsel["BestFitParameters"][[z,w]];
   SBBestPAram[rr[[1]]]
   ];


 label={
```

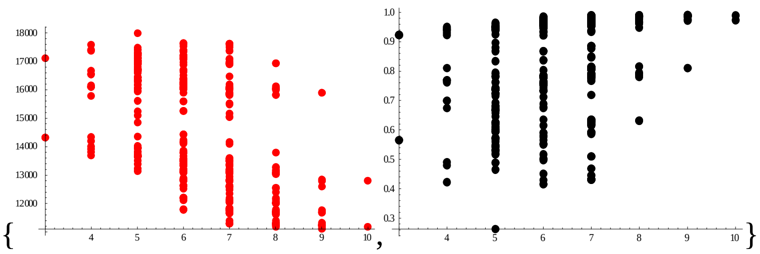
```
    ToString[Style["#Peak", Red,Bold,12]],
    ToString[Style["aax", Red,Bold,12]],
    ToString[ Style["aay", Red,Bold,12]],
    ToString[Style[ "PeakAlt", Red,Bold,12]],
    ToString[ Style["PeakArea(%)", Red,Bold,12]],
    ToString[ Style["PeakArea(Absolute)", Red,Bold,12]],
    ToString[ Style["sigmax", Red,Bold,12]],
    ToString[ Style["sigmay", Red,Bold,12]],
    ToString[ Style["alfax", Red,Bold,12]],
    ToString[ Style["alfay", Red,Bold,12]],
    ToString[ Style["Kost", Red,Bold,12]]};

  TABLESUMMARYindividualPeaks=Prepend[Table[Flatten[{n,
       NEWCOORINATEASDEFINITIVAS[[n]],
       Max[MATRIXCURVESNDIVIDUALS[[n,All,3]]],
       Plus@@MATRIXCURVESNDIVIDUALS[[n,All,3]]/AreaTotSimulat 100,
       Plus@@MATRIXCURVESNDIVIDUALS[[n,All,3]],
       param["sigmax",n],
       param["sigmay",n],
       param["alfax",0],
       param["alfay",0],

       param["Kost",n]
       }
      ], {n,1,Length[NEWCOORINATEASDEFINITIVAS]}],label] //TableForm;

  TABLESUMMARYindividualPeaks//Print;

  Print[Style["created the file : ", Red, 16, Bold],
   Export["F:\\CITOGRAMAS\\AllSamples_"<>
MESANY<>"_TABLE_\\"<>SAMPLE<>"_"<>MESANY<>"_ASYMETRIC_DISTRIBUTIONS_TableSummaryIndividualPeaks_BestRandom_Search_ROUNDED_.txt",TABLESUMMARYindividualPeaks,
"Table"]
   ];

  Print[Style["The file is in the following directory: ", Red, 16, Bold], Style[ToString[Directory[]], 16,Bold,
Black]];

  ]
```

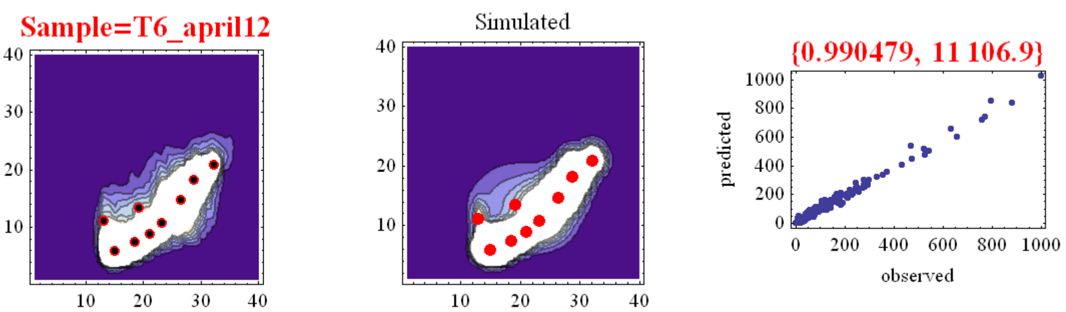
The decomposition is executed for each subset of P(Ln). In this example the decomposition is run 2047 times. To reduce the computation time the subsets i that can not generate a saddisfactory outputs can be removed.

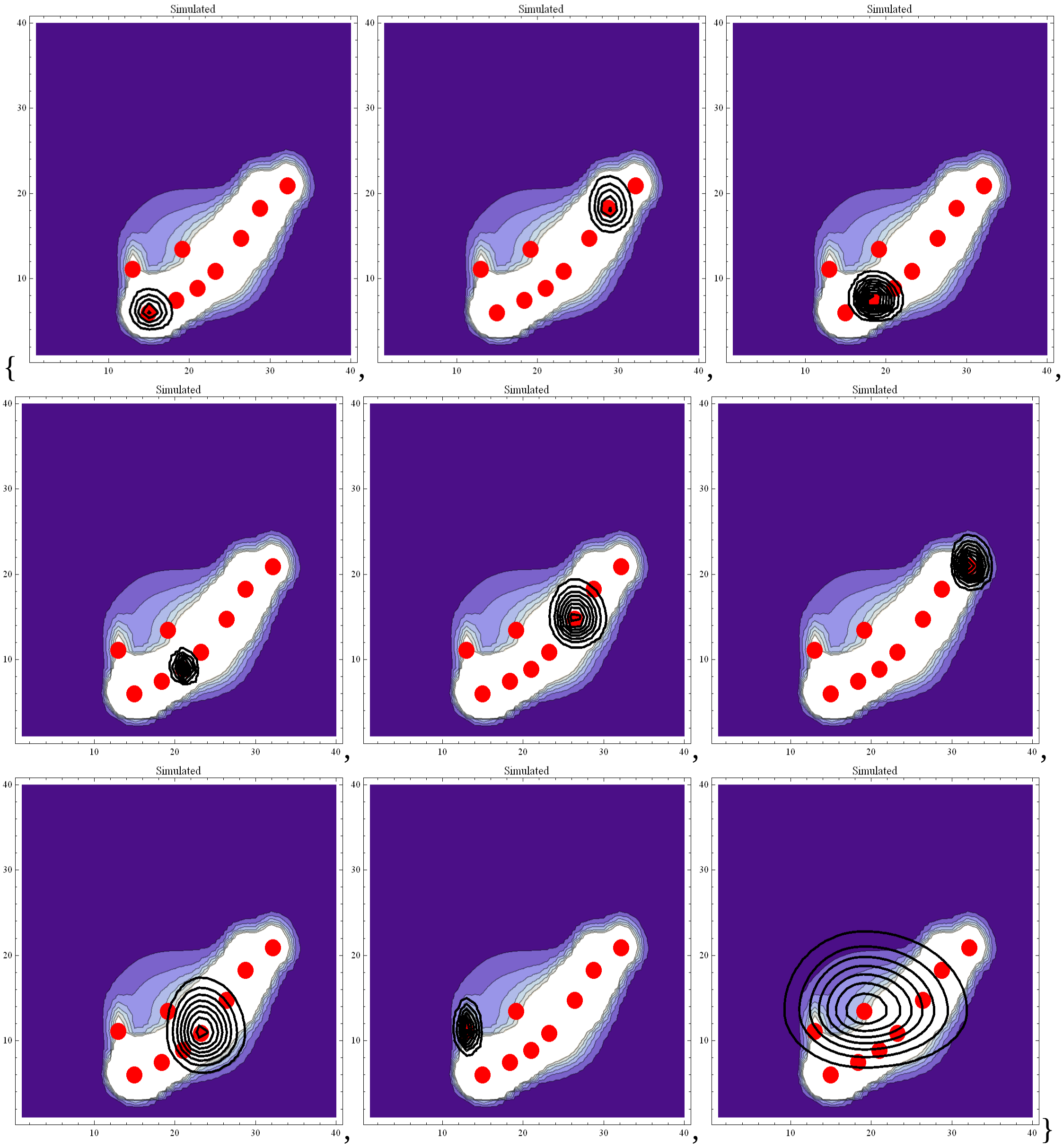
```
MODELEXECUT[SAMPLE_,Ninit_]:=Module[{},
   AllSubSets=Reverse[Drop[Subsets[ImportPotentialPeaks[{SAMPLE,MESANY}]],1]];
   Table[DECOMPOMPOSITION[{SAMPLE, MESANY,n (*desde 0 fins al num max de coordinates del
ImportPotentialPeaks*),50}], {n,Ninit,Length[AllSubSets]}]
   ];

MODELEXECUT["T6",1]
```

The following code lines are : 1) to summarize and visualize the results; 2) visualize the selected best model according the BIC criteria; 3) Visualize each peak of the selected peak.

```
SummStatIMPORT[{SAMPLE_, MESANY_}]:=Module[{aa,hh},
  SetDirectory["F:\\CITOGRAMAS\\AllSamples_"<>MESANY<>"_TABLE_"];
  aa=FileNames[SAMPLE<>"_"<>MESANY<>"SummStat_ROUNDED_*"];
  hh[n_]:=Import[aa[[n]],"List"];
  Table[hh[n], {n,1,Length[aa]}]]


SummStatIMPORTNumCOORDINATBIC[{SAMPLE_, MESANY_} ]:=SummStatIMPORT[{SAMPLE, MESANY}][[All,{5,-1}]]
SummStatIMPORTNumCOORDINATRsqrt[{SAMPLE_, MESANY_} ]:=SummStatIMPORT[{SAMPLE, MESANY}][[All,{5,-2}]]

BESTRESULTS[{SAMPLE_, MESANY_}]:=Module[{uu,DAT},
   DAT=SummStatIMPORT[{SAMPLE, MESANY}];
   Sort[DAT,#1[[-1]]<#2[[-1]]&]
   ];



BR=BESTRESULTS[{SAMPLE, MESANY}];

Partition[Delete[BR[[1]],{{1},{2}, {3}, {4},{5}, {-1},{-2}}],2]


{ListPlot[SummStatIMPORTNumCOORDINATBIC[{SAMPLE, MESANY }],PlotStyle→{Red, PointSize[0.023]}, PlotRange→All],ListPlot[SummStatIMPORTNumCOORDINATRsqrt[{SAMPLE, MESANY }], PlotStyle→{Black, PointSize[0.023]}, PlotRange→All]}

BR[[1;;20]]//TableForm

VISUALSIMULATBestRandomSearch[ {SAMPLE,  MESANY,BR[[1,2]],100}  ]
```







{
{#Peak, aax, aay, PeakAlt, PeakArea(%), PeakArea(Absolute), sigmax, sigmay, alfax, alfay, Kost},
{1, 14.98, 6.012, 1024.05, 38.4639, 11734.9, 1.2275, 1.07329, 1.29269, 1.41483, 1024.2},
{2, 28.76, 18.23, 255.596, 13.4695, 4109.37, 1.22002, 1.48025, 1.29269, 1.41483, 261.713},
{3, 18.39, 7.469, 227.114, 10.6556, 3250.91, 1.34463, 1.10769, 1.29269, 1.41483, 250.865},
{4, 20.98, 8.879, 74.3193, 1.23991, 378.282, -0.703424, 0.82997, 1.29269, 1.41483, 74.733},
{5, 26.38, 14.75, 209.81, 13.9869, 4267.22, 1.43485, 1.5639, 1.29269, 1.41483, 218.689},
{6, 32.13, 20.86, 93.3765, 3.38273, 1032.03, 1.02389, -1.22686, 1.29269, 1.41483, 94.4389},
{7, 23.21, 10.88, 83.9179, 10.792, 3292.51, -2.02603, -2.21315, 1.29269, 1.41483, 84.4319},
{8, 12.99, 11.11, 36.2212, 0.987571, 301.296, 0.691782, 1.37808, 1.29269, 1.41483, 36.3391},
{9, 19.1, 13.46, 15.4485, 7.02195, 2142.31, 4.86925, 3.25309, 1.29269, 1.41483, 15.5584}
}