```
(* SVD Analysis of Elutriation *)
```

```
(* Normalize Selected Data *)
```

```
(* Read Selected Data from Select_Data.txt *)

<< LinearAlgebra`MatrixManipulation`;
<< NumericalMath`TrigFit`;
<< Graphics`Graphics`;
<< Graphics`Arrow`;
Off[General::"spell"];

(* Define HardDrive *)

name = "Marzipan";

(* Elutriation *)

stream = StringJoin[name, ":Desktop Folder:PNAS Data:Select_Elutriation.txt"];
matrix = ReadList[stream, Word, RecordLists -> True, NullWords -> True];
{genes, arrays} = Dimensions[matrix] - {2, 1}
Clear[stream];
```

```
{5981, 14}
```

```
genenames = TakeRows[
  TakeColumns[matrix, {1, 1}],
  {3, genes + 2}];
arraynames = TakeColumns[
  TakeRows[matrix, {1, 2}],
  {2, arrays + 1}];
matrix = TakeColumns[
  TakeRows[matrix, {3, genes + 2}],
    {2, arrays + 1}];
matrix = ToExpression[matrix];

sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[arraynames[[2, a]]]
    ]
   ], {a, 1, arrays}]];
size = Sort[sizes, OrderedQ[{#2, #1}]&][[1]];
Do[
 Do[arraynames[[2, a]] = StringJoin[ToString[arraynames[[2, a]]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, arrays}];
```

```
(* Examine Raw Data *)
```

```
(* Calculate Singular Value Decomposition *)
```

```
correlation = Dot[Transpose[matrix], matrix] / (arrays - 1);
{eigenexpressions, eigengenes} = Eigensystem[correlation];
eigenexpressions = Sqrt[(arrays - 1) * eigenexpressions];
Clear[correlation];
Do[eigengenes[[a]] = -eigengenes[[a]],
 {a, 1, arrays}];
eigenarrays = Dot[eigengenes, Transpose[matrix]];
Do[
 eigenarrays[[a]] = eigenarrays[[a]] / eigenexpressions[[a]],
 {a, 1, arrays}];
eigenarrays = Transpose[eigenarrays];
arraycorrelations = Dot[DiagonalMatrix[eigenexpressions], eigengenes];
genecorrelations = Dot[eigenarrays, DiagonalMatrix[eigenexpressions]];
genecorrelations = Transpose[genecorrelations];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, arrays}] / Log[arrays]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.14
```

```
(* Create Fractions Bar Charts Displays *)
```

```
fractions[[2]]
```

```
0.0311582
```

```
limit = 0.04;
```

```
Clear[gridx, framex, framey, sizes];
gridx = Table[a, {a, 0, limit, N[limit / 4]}];
framex = gridx;
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[framex[[a]]]
    ]
   ], {a, 1, 5}]];
Do[
 Do[framex[[a]] = StringJoin[ToString[framex[[a]]], " "],
    {b, 1, 4 - sizes[[a]]}],
   {a, 1, 5}];
framex = Table[{gridx[[a]], framex[[a]]}, {a, 1, 5}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 5}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 2}];
table = Table[fractions[[arrays - a]], {a, 0, arrays - 2}];
g = BarChart[
  table,
  BarOrientation -> Horizontal,
  PlotRange -> {{0, limit * 1.0001}, {0.5, arrays - 1 + 0.5}},
  AspectRatio -> 1,
  Axes -> False,
  Frame -> True,
  FrameTicks -> {None, framey, framex, None},
  FrameLabel -> {None, None, None, None},
  GridLines -> {gridx, None},
  DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.5}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> 1.25,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
gridx = Table[a, {a, 0, 1, 0.2}];
framex = gridx;
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[framex[[a]]]
     ]
    ], {a, 1, 6}]];
Do[
 Do[framex[[a]] = StringJoin[ToString[framex[[a]]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, 6}];
framex = Table[{gridx[[a]],
   framex[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 1}];
labelx = ColumnForm[
  {"(b) Eigenexpression Fraction", StringJoin["d = ", ToString[entropy]], " "}, Center];
g = BarChart[
  Table[fractions[[arrays - a]], {a, 0, arrays - 1}],
  BarOrientation -> Horizontal,
  PlotRange -> {{0, 1.0001}, {0.5, arrays + 0.5}},
  AspectRatio -> 1,
  Axes -> False,
  Frame -> True,
  FrameTicks -> {None, framey, framex, None},
  FrameLabel -> {None, None, labelx, None},
  GridLines -> {gridx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.8}, {0, 0}, {0, 1}];
g2 = Show[{g,
   Graphics[{RGBColor[1, 1, 0.8], Rectangle[{0.1, 0.6}, {0.98, 12.6}]}],
   Graphics[{Rectangle[{0.1, 0.6}, {0.98, 12.6}, g1]}]},
 AspectRatio -> 1.35,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
(* Create Eigengenes 2D Red & Green Raster Display *)

contrast = 3;
displaying = Table[
  If[contrast * eigengenes[[i, j]] > 0,
   If[contrast * eigengenes[[i, j]] < 1, {contrast * eigengenes[[i, j]], 0}, {1, 0}],
   If[contrast * eigengenes[[i, j]] > -1, {0, -contrast * eigengenes[[i, j]]}, {0, 1}]],
  {i, 1, arrays}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[2, a]]}, {a, 1, arrays}];
framey = Table[{a + 1 - 0.5, arrays - a}, {a, 0, arrays - 1}];
labely = "Eigengenes";
labelx = ColumnForm[{"(a) Arrays", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, arrays, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 2, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.8}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> 1.05,
 PlotRange -> All,
 DisplayFunction -> Identity];

(* Create Selected Eigengenes Graph Display *)

eigengenes1 = Chop[TrigFit[eigengenes[[1]], 2, {x, arrays - 1}], 0.05]
eigengenes2 = Chop[TrigFit[eigengenes[[2]], 2, {x, arrays - 1}], 0.175]
eigengenes3 = Chop[TrigFit[eigengenes[[3]], 4, {x, arrays - 1}], 0.15]
eigengenes4 = Chop[TrigFit[eigengenes[[4]], 4, {x, arrays - 1}], 0.15]
```

0.266135

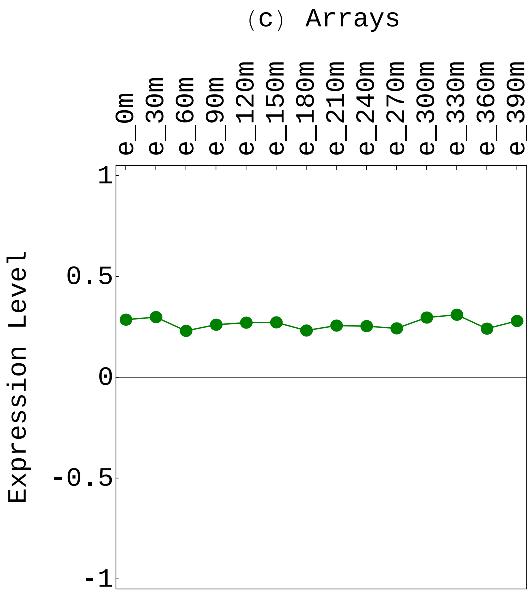
$$0.241146\ \mathrm{Cos}\left[\frac{2\pi x}{13}\right]$$

$$0.322034\ \mathrm{Sin}\left[\frac{2\pi x}{13}\right]$$

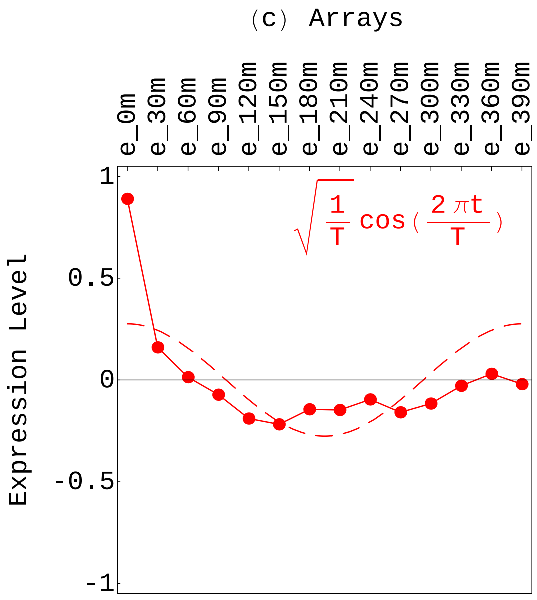
$$0.190452\ \mathrm{Cos}\left[\frac{2\pi x}{13}\right]$$

```
eigengenes2 = 0.276 * Cos[2 * Pi * x / 13];
eigengenes3 = 0.39 * Sin[2 * Pi * x / 13];
eigengenes4 = 0.276 * Cos[2 * Pi * x / 13];
```

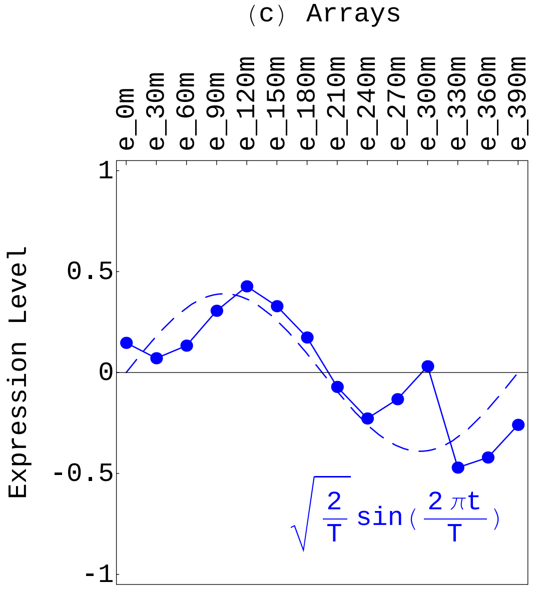
```
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {-1, -0.5, 0, 0.5, 1};
coordinates = Table[{a - 1, eigengenes[[1, a]]}, {a, 1, arrays}];
points = Table[Point[coordinates[[a]]], {a, 1, arrays}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[0, 0.5, 0], PointSize[0.022], points}],
   Graphics[{RGBColor[0, 0.5, 0], line}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
 FrameTicks -> {None, framey, framex, None},
 PlotRange -> {-1.05, 1.05},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 0.24}, {0, 0}, {0, 1}];
p1 = Show[g,
  AspectRatio -> 1.05,
  PlotRange -> All];
```



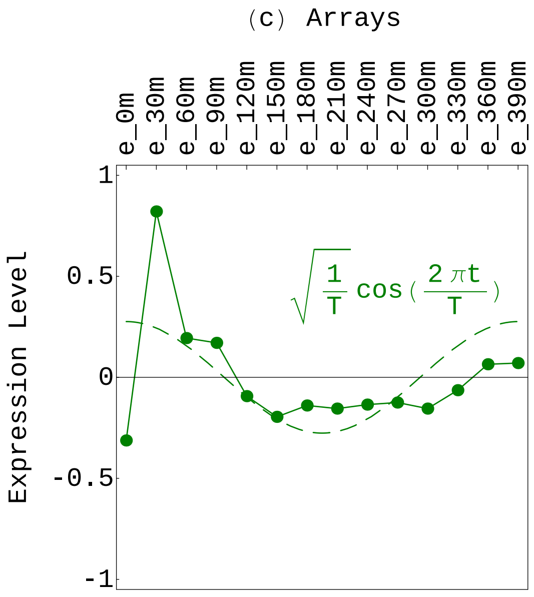
```
graph = Plot[
  eigengenes2,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[1, 0, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {-1, -0.5, 0, 0.5, 1};
coordinates = Table[{a - 1, eigengenes[[2, a]]},  {a, 1, arrays}];
points = Table[Point[coordinates[[a]]],  {a, 1, arrays}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[1, 0, 0], PointSize[0.022], points}],
   Graphics[{RGBColor[1, 0, 0], line}],
   graph,
   Graphics[{RGBColor[1, 0, 0], Text["√(1/T) cos(2 π t/T)", {9, 0.8}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
 FrameTicks -> {None, framey, framex, None},
 PlotRange -> {-1.05, 1.05},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 0.24}, {0, 0}, {0, 1}];
p2 = Show[g,
  AspectRatio -> 1.05,
  PlotRange -> All];
```



```
graph = Plot[
  eigengenes3,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[0, 0, 1], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {-1, -0.5, 0, 0.5, 1};
coordinates = Table[{a - 1, eigengenes[[3, a]]}, {a, 1, arrays}];
points = Table[Point[coordinates[[a]]], {a, 1, arrays}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[0, 0, 1], PointSize[0.022], points}],
   Graphics[{RGBColor[0, 0, 1], line}],
   graph,
   Graphics[{RGBColor[0, 0, 1], Text["√(2/T) sin(2πt/T)", {9, -0.7}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
 FrameTicks -> {None, framey, framex, None},
 PlotRange -> {-1.05, 1.05},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 0.24}, {0, 0}, {0, 1}];
p3 = Show[g,
  AspectRatio -> 1.05,
  PlotRange -> All];
```

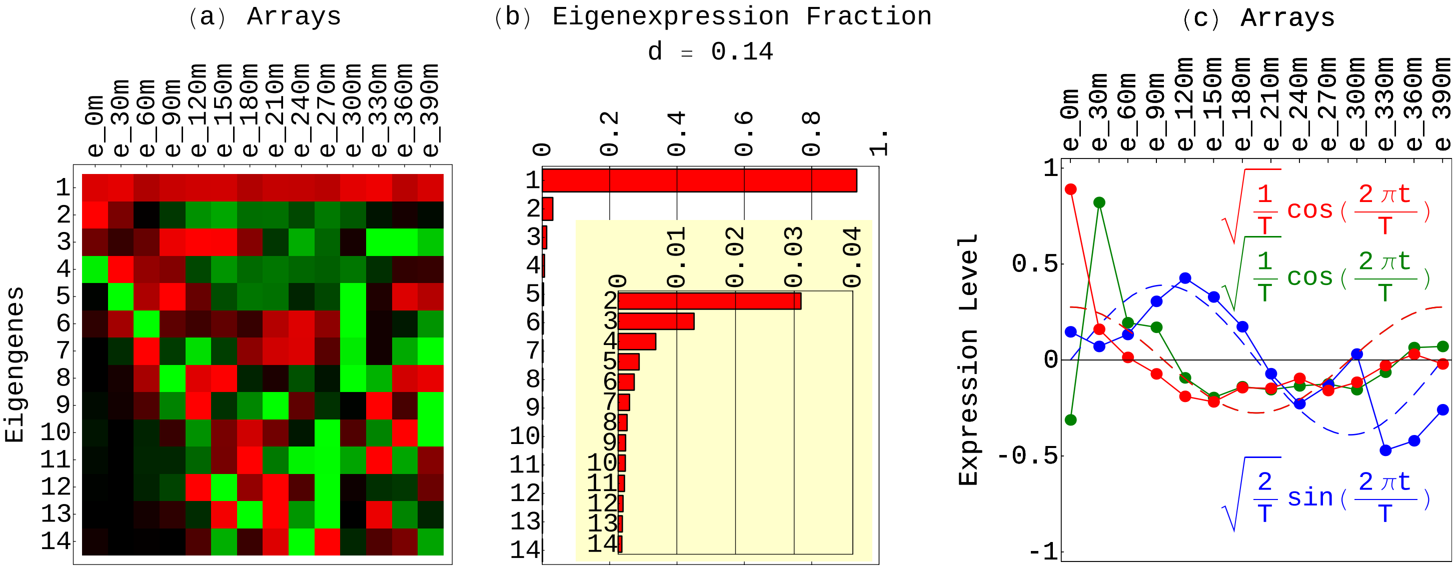


```
graph = Plot[
  eigengenes4,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[0, 0.5, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {-1, -0.5, 0, 0.5, 1};
coordinates = Table[{a - 1, eigengenes[[4, a]]}, {a, 1, arrays}];
points = Table[Point[coordinates[[a]]], {a, 1, arrays}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[0, 0.5, 0], PointSize[0.022], points}],
   Graphics[{RGBColor[0, 0.5, 0], line}],
   graph,
   Graphics[{RGBColor[0, 0.5, 0], Text["√(1/T) cos(2 π t/T)", {9, 0.45}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
 FrameTicks -> {None, framey, framex, None},
 PlotRange -> {-1.05, 1.05},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 0.24}, {0, 0}, {0, 1}];
p4 = Show[g,
  AspectRatio -> 1.05,
  PlotRange -> All];
```



```
(* Display Selected Eigengenes *)

g3 = Show[{p4, p3, p2},
  DisplayFunction -> Identity];
```

```
(* Display Eigengenes, Fractions and Selected Eigengenes *)

Show[GraphicsArray[{g1, g2, g3}],
 GraphicsSpacing -> -0.125];
```



```
(* and Remove the Additive Constant *)
```

```
(* Reconstruct Data Without Additive Constant *)

eigenexpressions[[1]] = 0;
matrix = Dot[eigenarrays, DiagonalMatrix[eigenexpressions], eigengenes];
```

```
(* Remove the Multiplicative Constant *)
```

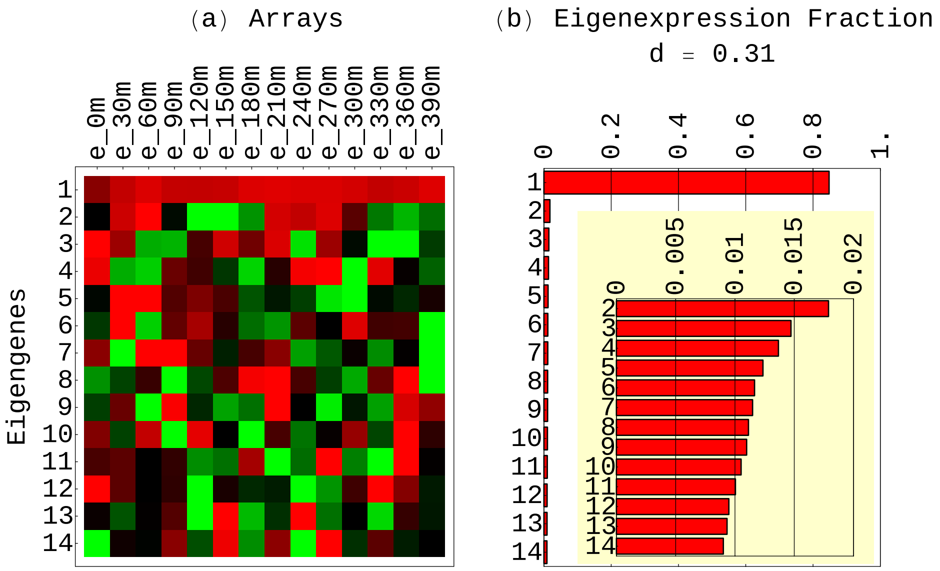
```
(* Calculate Singular Value Decomposition *)

normalization = Log[matrix^2];
correlation = Dot[Transpose[normalization], normalization] / (arrays - 1);
{eigenexpressions, eigengenes} = Eigensystem[correlation];
eigenexpressions = Sqrt[(arrays - 1) * eigenexpressions];
Clear[correlation];
eigengenes[[2]] = -eigengenes[[2]];
eigenarrays = Dot[eigengenes, Transpose[normalization]];
Do[
 eigenarrays[[a]] = eigenarrays[[a]] / eigenexpressions[[a]],
 {a, 1, arrays}];
eigenarrays = Transpose[eigenarrays];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, arrays}] / Log[arrays]];
entropy = N[Round[100 * entropy] / 100]

0.31
```

```
(* Create Fractions Bar Charts Displays *)

fractions[[2]]

0.0178904

limit = 0.02;
```

```
Clear[gridx, framex, framey, sizes];
gridx = Table[a, {a, 0, limit, N[limit / 4]}];
framex = gridx;
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[framex[[a]]]
    ]
   ], {a, 1, 5}]];
Do[
 Do[framex[[a]] = StringJoin[ToString[framex[[a]]], " "],
    {b, 1, 5 - sizes[[a]]}],
   {a, 1, 5}];
framex = Table[{gridx[[a]], framex[[a]]}, {a, 1, 5}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 5}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 2}];
table = Table[fractions[[arrays - a]], {a, 0, arrays - 2}];
g = BarChart[
  table,
  BarOrientation -> Horizontal,
  PlotRange -> {{0, limit * 1.0001}, {0.5, arrays - 1 + 0.5}},
  AspectRatio -> 1,
  Axes -> False,
  Frame -> True,
  FrameTicks -> {None, framey, framex, None},
  FrameLabel -> {None, None, None, None},
  GridLines -> {gridx, None},
  DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 2}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> 1.25,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
gridx = Table[a, {a, 0, 1, 0.2}];
framex = gridx;
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[framex[[a]]]
    ]
   ], {a, 1, 6}]];
Do[
 Do[framex[[a]] = StringJoin[ToString[framex[[a]]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, 6}];
framex = Table[{gridx[[a]],
   framex[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 1}];
labelx = ColumnForm[
  {"(b) Eigenexpression Fraction", StringJoin["d = ", ToString[entropy]], " "}, Center];
g = BarChart[
  Table[fractions[[arrays - a]], {a, 0, arrays - 1}],
  BarOrientation -> Horizontal,
  PlotRange -> {{0, 1.0001}, {0.5, arrays + 0.5}},
  AspectRatio -> 1,
  Axes -> False,
  Frame -> True,
  FrameTicks -> {None, framey, framex, None},
  FrameLabel -> {None, None, labelx, None},
  GridLines -> {gridx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.8}, {0, 0}, {0, 1}];
g2 = Show[{g,
   Graphics[{RGBColor[1, 1, 0.8], Rectangle[{0.1, 0.6}, {0.98, 13}]}],
   Graphics[{Rectangle[{0.1, 0.6}, {0.98, 13}, g1]}]},
 AspectRatio -> 1.35,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
(* Create Eigengenes 2 D Red & Green Raster Display *)

contrast = 3;
displaying = Table[
  If[contrast * eigengenes[[i, j]] > 0,
   If[contrast * eigengenes[[i, j]] < 1, {contrast * eigengenes[[i, j]], 0}, {1, 0}],
   If[contrast * eigengenes[[i, j]] > -1, {0, -contrast * eigengenes[[i, j]]}, {0, 1}]],
  {i, 1, arrays}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[2, a]]}, {a, 1, arrays}];
framey = Table[{a + 1 - 0.5, arrays - a}, {a, 0, arrays - 1}];
labely = "Eigengenes";
labelx = ColumnForm[{"(a) Arrays", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, arrays, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 2, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.8}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> 1.05,
 PlotRange -> All,
 DisplayFunction -> Identity];

(* Display Both Eigengenes & Fractions *)

Show[GraphicsArray[{g1, g2}],
 GraphicsSpacing -> -0.125];
```



```
(* Reconstruct the Data Without Multiplicative Constant *)
```

```
eigenexpressions[[1]] = 0;
normalization = Dot[eigenarrays, DiagonalMatrix[eigenexpressions], eigengenes];
normalization = Sqrt[Exp[normalization]];
matrix = Sign[matrix];
matrix = N[matrix * normalization];
Clear[normalization];
```

```
(* Examine Normalized Data *)
```

```
(* Calculate Singular Value Decomposition *)
```

```
correlation = Dot[Transpose[matrix], matrix] / (arrays - 1);
{eigenexpressions, eigengenes} = Eigensystem[correlation];
eigenexpressions = Sqrt[(arrays - 1) * eigenexpressions];
Clear[correlation];
eigengenes[[1]] = -eigengenes[[1]];
eigenarrays = Dot[eigengenes, Transpose[matrix]];
Do[
 eigenarrays[[a]] = eigenarrays[[a]] / eigenexpressions[[a]],
 {a, 1, arrays}];
eigenarrays = Transpose[eigenarrays];
arraycorrelations = Dot[DiagonalMatrix[eigenexpressions], eigengenes];
genecorrelations = Dot[eigenarrays, DiagonalMatrix[eigenexpressions]];
genecorrelations = Transpose[genecorrelations];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, arrays}] / Log[arrays]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.88
```

```
(* Create Fractions Bar Charts Displays *)
```

```
fractions[[1]]
```

```
0.22391
```

```
limit = 0.25;
```

```
Clear[gridx, framex, framey, sizes];
gridx = Table[a, {a, 0, limit, N[limit / 5]}];
framex = gridx;
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[framex[[a]]]
    ]
   ], {a, 1, 6}]];
Do[
 Do[framex[[a]] = StringJoin[ToString[framex[[a]]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, 6}];
framex = Table[{gridx[[a]], framex[[a]]}, {a, 1, 6}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 1}];
labelx = ColumnForm[
  {"(b) Eigenexpression Fraction", StringJoin["d = ", ToString[entropy]], " "}, Center];
g = BarChart[
  Table[fractions[[arrays - a]], {a, 0, arrays - 1}],
  BarOrientation -> Horizontal,
  PlotRange -> {{0, limit * 1.0001}, {0.5, arrays + 0.5}},
  AspectRatio -> 1,
  Axes -> False,
  Frame -> True,
  FrameTicks -> {None, framey, framex, None},
  FrameLabel -> {None, None, labelx, None},
  GridLines -> {gridx, None},
  DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.8}, {0, 0}, {0, 1}];
g2 = Show[g,
 AspectRatio -> 1.35,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

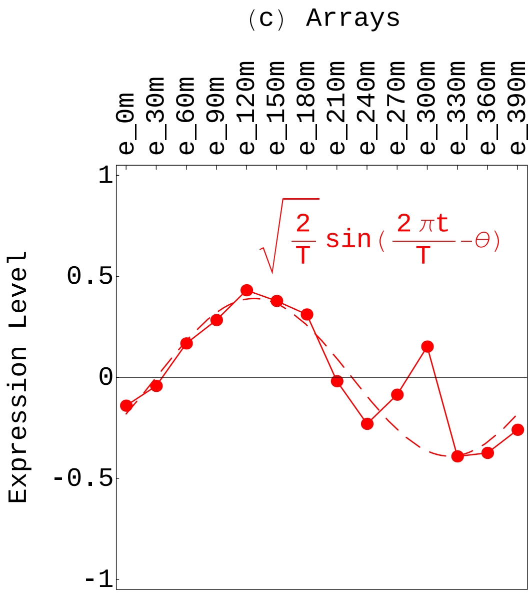
```
(* Create Eigengenes 2 D Red & Green Raster Display *)

contrast = 3;
displaying = Table[
  If[contrast * eigengenes[[i, j]] > 0,
   If[contrast * eigengenes[[i, j]] < 1, {contrast * eigengenes[[i, j]], 0}, {1, 0}],
   If[contrast * eigengenes[[i, j]] > -1, {0, -contrast * eigengenes[[i, j]]}, {0, 1}]],
  {i, 1, arrays}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[2, a]]}, {a, 1, arrays}];
framey = Table[{a + 1 - 0.5, arrays - a}, {a, 0, arrays - 1}];
labely = "Eigengenes";
labelx = ColumnForm[{"(a) Arrays", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, arrays, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 2, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 1.8}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> 1.05,
 PlotRange -> All,
 DisplayFunction -> Identity];

(* Create Selected Eigengenes Graph Display *)

eigengenes1 = Chop[TrigFit[eigengenes[[1]], 2, {x, arrays - 1}], 0.12]
eigengenes2 = Chop[TrigFit[eigengenes[[2]], 2, {x, arrays - 1}], 0.1]
```

$$-0.133012\,\mathrm{Cos}\left[\frac{2\,\pi\,x}{13}\right]+0.292161\,\mathrm{Sin}\left[\frac{2\,\pi\,x}{13}\right]$$

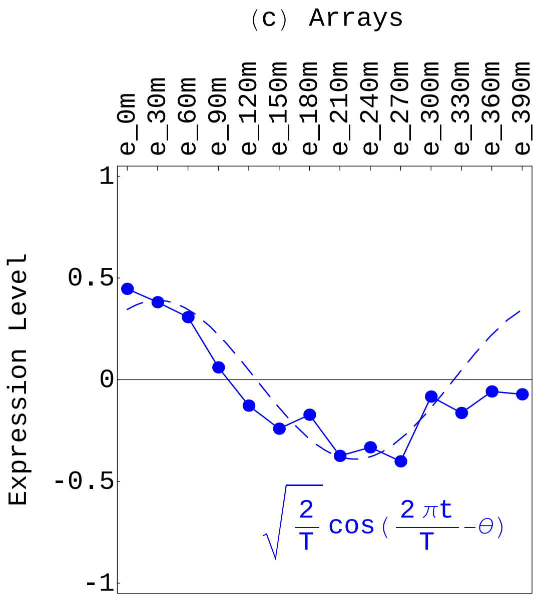
$$0.304829\,\mathrm{Cos}\left[\frac{2\,\pi\,x}{13}\right]+0.121684\,\mathrm{Sin}\left[\frac{2\,\pi\,x}{13}\right]$$

```
eigengenes1 = 0.39 * Sin[2 * Pi * (x - 1) / 13];
eigengenes2 = 0.39 * Cos[2 * Pi * (x - 1) / 13];
```

```
graph = Plot[
  eigengenes1,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[1, 0, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {-1, -0.5, 0, 0.5, 1};
coordinates = Table[{a - 1, eigengenes[[1, a]]},  {a, 1, arrays}];
points = Table[Point[coordinates[[a]]],  {a, 1, arrays}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[1, 0, 0], PointSize[0.022], points}],
   Graphics[{RGBColor[1, 0, 0], line}],
   graph,
   Graphics[{RGBColor[1, 0, 0], Text["√(2/T) sin(2 πt/T -θ)", {8.5, 0.7}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
 FrameTicks -> {None, framey, framex, None},
 PlotRange -> {-1.05, 1.05},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 0.24}, {0, 0}, {0, 1}];
p1 = Show[g,
  AspectRatio -> 1.05,
  PlotRange -> All];
```

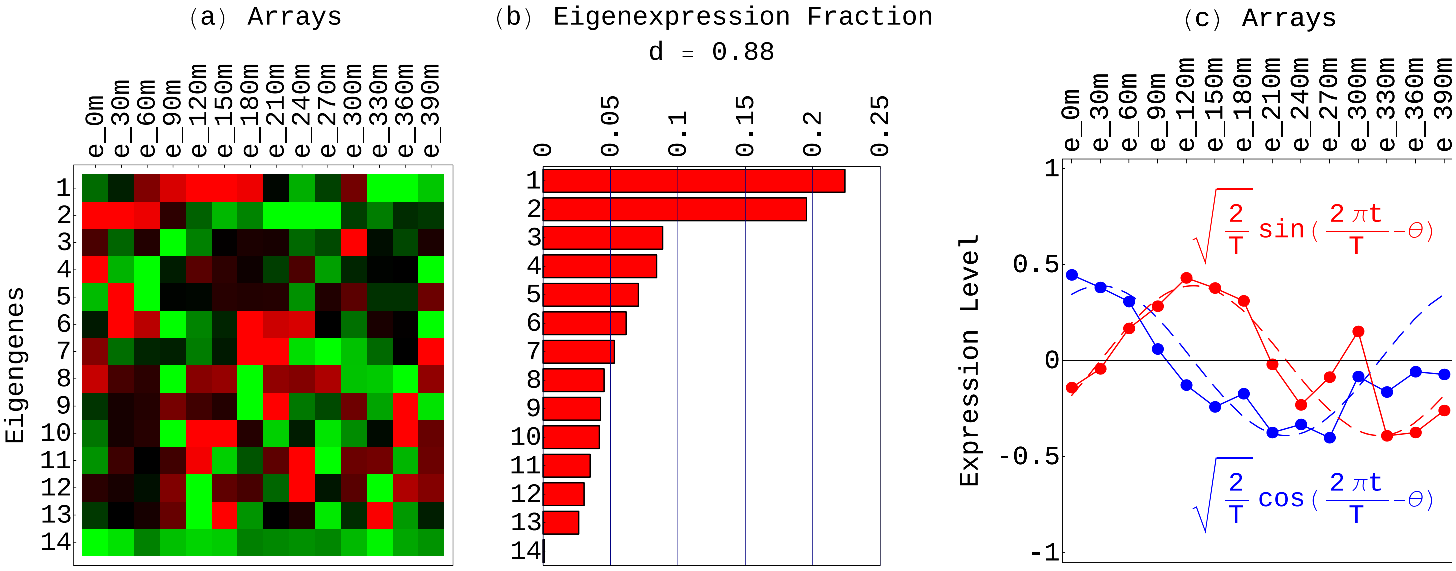


```
graph = Plot[
  eigengenes2,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[0, 0, 1], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {-1, -0.5, 0, 0.5, 1};
coordinates = Table[{a - 1, eigengenes[[2, a]]},  {a, 1, arrays}];
points = Table[Point[coordinates[[a]]],  {a, 1, arrays}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[0, 0, 1], PointSize[0.022], points}],
   Graphics[{RGBColor[0, 0, 1], line}],
   graph,
   Graphics[{RGBColor[0, 0, 1], Text["√(2/T) cos(2πt/T-θ)", {8.5, -0.7}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
 FrameTicks -> {None, framey, framex, None},
 PlotRange -> {-1.05, 1.05},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 0.24}, {0, 0}, {0, 1}];
p2 = Show[g,
  AspectRatio -> 1.05,
  PlotRange -> All];
```



```
(* Display Selected Eigengenes *)

g3 = Show[{p2, p1},
 DisplayFunction -> Identity];
```

```
(* Display Eigengenes, Fractions and Selected Eigengenes *)

Show[GraphicsArray[{g1, g2, g3}],
 GraphicsSpacing -> -0.125];
```



```
(* Create Parameter Graphs of Arrays According to Projections on Eigenarrays *)

labelx = ColumnForm[{"Array Correlation with |α2〉"}, Center];
labely = ColumnForm[{"Array Correlation with |α1〉"}, Center];
matrix = Transpose[matrix];
coordinates = Table[
  {arraycorrelations[[2, a]] / Sqrt[Dot[matrix[[a]], matrix[[a]]]],
   arraycorrelations[[1, a]] / Sqrt[Dot[matrix[[a]], matrix[[a]]]]},
  {a, 1, arrays}];
matrix = Transpose[matrix];
points1 = {Point[coordinates[[1]]], Point[coordinates[[2]]], Point[coordinates[[14]]]};
points2 = Table[Point[coordinates[[a]]], {a, 3, 6}];
points3 = Table[Point[coordinates[[a]]], {a, 7, 8}];
points4 = Table[Point[coordinates[[a]]], {a, 9, 10}];
points5 = Table[Point[coordinates[[a]]], {a, 11, 13}];
textcoordinates = coordinates;
Do[textcoordinates[[a, 1]] =
  If[textcoordinates[[a, 1]] > 0, textcoordinates[[a, 1]] + 0.085, textcoordinates[[a, 1]] - 0.085],
 {a, 1, 9}];
Do[textcoordinates[[a, 1]] =
  If[textcoordinates[[a, 1]] > 0, textcoordinates[[a, 1]] + 0.11, textcoordinates[[a, 1]] - 0.11],
 {a, 10, arrays}];
textcoordinates[[4]] = textcoordinates[[4]] + {0, 0.02};
textcoordinates[[8]] = textcoordinates[[8]] + {0, -0.02};
textcoordinates[[13]] = textcoordinates[[13]] + {0.01, 0.04};
textcoordinates[[14]] = textcoordinates[[14]] + {0.085, 0.085};
texts = Table[Text[a, textcoordinates[[a]]], {a, 1, arrays}];
zerophase = N[ArcTan[arraycorrelations[[1, 1]] / arraycorrelations[[2, 1]]]];
```

```
p = Show[{
  Graphics[{RGBColor[1, 1, 0], PointSize[0.035], points1}],
  Graphics[{RGBColor[0, 0.5, 0], PointSize[0.035], points2}],
  Graphics[{RGBColor[0, 0, 1], PointSize[0.035], points3}],
  Graphics[{RGBColor[1, 0, 0], PointSize[0.035], points4}],
  Graphics[{RGBColor[1, 0.5, 0], PointSize[0.035], points5}],
  Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.6, -0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.8, -0.85}]}],
  Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-0.875, -0.75}]}],
  Graphics[{RGBColor[0, 0, 1], Text["S", {-0.95, 0.5}]}],
  Graphics[{RGBColor[0, 0.5, 0], Text["G1", {0.55, 0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Text["(a)", {-0.9, 0.95}]}],
  Graphics[texts],
  Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 1]}],
  Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 0.5]}],
  Graphics[{RGBColor[0, 0, 0], Circle[{0, 0}, 0.45, {zerophase, 0}]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[
     {0.45 * Cos[-0.05], 0.45 * Sin[-0.05]}, {0.45 * Cos[0], 0.45 * Sin[0]},
     HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}],
  Graphics[{RGBColor[0, 0, 0], Circle[{0, 0}, 0.9, {zerophase, -2 * zerophase}]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[
      {0.9 * Cos[-2 * zerophase - 0.05], 0.9 * Sin[-2 * zerophase - 0.05]},
      {0.9 * Cos[-2 * zerophase], 0.9 * Sin[-2 * zerophase]},
      HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[{0, 0}, coordinates[[1]],
      HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}],
  Graphics[{RGBColor[0, 0, 0], Text["r", {0.65, -0.35}]}],
  Graphics[{RGBColor[0, 0, 0], Text["θ", {0.35, -0.05}]}]},
 AspectRatio -> 1,
 PlotRange -> {{-1.05, 1.05}, {-1.05, 1.05}},
 Frame -> True,
 FrameTicks -> False,
 FrameLabel -> {labelx, labely, None, None},
 GridLines -> {{{0, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
 DisplayFunction -> Identity];
p = FullGraphics[p];
p[[1, 2]] = p[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {-1.18, 0}, {0, 0}, {0, 1}];
p1 = Show[p,
 AspectRatio -> 1.,
 PlotRange -> All,
 DisplayFunction -> Identity];

(* Create Parameter Graphs of Genes According to Projections on Eigengenes *)

labelx = ColumnForm[{"Gene Correlation with |γ2〉"}, Center];
labely = ColumnForm[{"Gene Correlation with |γ1〉"}, Center];
coordinates = Table[
  {genecorrelations[[2, a]] / Sqrt[Dot[matrix[[a]], matrix[[a]]]],
   genecorrelations[[1, a]] / Sqrt[Dot[matrix[[a]], matrix[[a]]]]},
  {a, 1, genes}];
points = Table[Point[coordinates[[a]]],  {a, 1, genes}];
stream = StringJoin[name, ":Desktop Folder:PNAS Data:Classify_Elutriation.txt"];
list = ReadList[stream, Word, RecordLists -> True, NullWords -> True];
list = Drop[list, 2];
```

```
position = Position[list, "M/G1"];
points1 = Table[Point[coordinates[[position[[a, 1]]]]], {a, 1, Dimensions[position][[1]]}];
radii1 = Table[Sqrt[coordinates[[position[[a, 1]], 1]]^2 + coordinates[[position[[a, 1]], 2]]^2],
   {a, 1, Dimensions[position][[1]]}];
Dimensions[points1][[1]]
position = Position[list, "G1"];
points2 = Table[Point[coordinates[[position[[a, 1]]]]], {a, 1, Dimensions[position][[1]]}];
radii2 = Table[Sqrt[coordinates[[position[[a, 1]], 1]]^2 + coordinates[[position[[a, 1]], 2]]^2],
   {a, 1, Dimensions[position][[1]]}];
Dimensions[points2][[1]]
position = Position[list, "S"];
points3 = Table[Point[coordinates[[position[[a, 1]]]]], {a, 1, Dimensions[position][[1]]}];
radii3 = Table[Sqrt[coordinates[[position[[a, 1]], 1]]^2 + coordinates[[position[[a, 1]], 2]]^2],
   {a, 1, Dimensions[position][[1]]}];
Dimensions[points3][[1]]
position = Position[list, "S/G2"];
points4 = Table[Point[coordinates[[position[[a, 1]]]]], {a, 1, Dimensions[position][[1]]}];
radii4 = Table[Sqrt[coordinates[[position[[a, 1]], 1]]^2 + coordinates[[position[[a, 1]], 2]]^2],
   {a, 1, Dimensions[position][[1]]}];
Dimensions[points4][[1]]
position = Position[list, "G2/M"];
points5 = Table[Point[coordinates[[position[[a, 1]]]]], {a, 1, Dimensions[position][[1]]}];
radii5 = Table[Sqrt[coordinates[[position[[a, 1]], 1]]^2 + coordinates[[position[[a, 1]], 2]]^2],
   {a, 1, Dimensions[position][[1]]}];
Dimensions[points5][[1]]
```

```
112

295

69

116

192
```

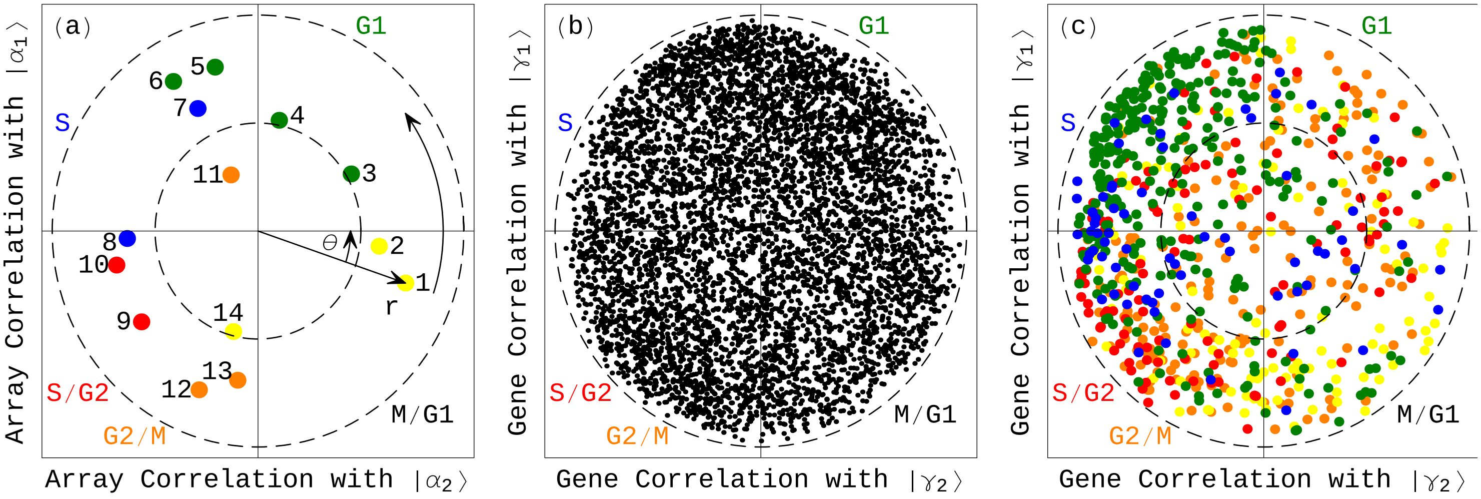
```
radii = AppendRows[{radii1}, {radii2}, {radii3}, {radii4}, {radii5}][[1]];
radii = Sort[radii, OrderedQ[{{#1}, {#2}}]&];
N[Round[radii[[143]] * 100] / 100]
N[Round[radii[[144]] * 100] / 100]
```

```
0.49

0.5
```

```
(* 784 cell cycle genes, 112 in M/G1, 295 in G1, 69 in S, 116 in S/G2, 192 in G2/M. *)

(* 641 with more than 25% of normalized expression in the cell cycle subspace. *)
```

```
p = Show[{
  Graphics[{RGBColor[0, 0, 0], PointSize[0.01], points}],
  Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.6, -0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.8, -0.85}]}],
  Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-0.875, -0.75}]}],
  Graphics[{RGBColor[0, 0, 1], Text["S", {-0.95, 0.5}]}],
  Graphics[{RGBColor[0, 0.5, 0], Text["G1", {0.55, 0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Text["(b)", {-0.9, 0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 1]}]},
 AspectRatio -> 1,
 PlotRange -> {{-1.05, 1.05}, {-1.05, 1.05}},
 Frame -> True,
 FrameTicks -> False,
 FrameLabel -> {labelx, labely, None, None},
 GridLines -> {{{0, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
 DisplayFunction -> Identity];
```

```
p = FullGraphics[p];
p[[1, 2]] = p[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {-1.18, 0}, {0, 0}, {0, 1}];
p2 = Show[p,
  AspectRatio -> 1.,
  PlotRange -> All,
  DisplayFunction -> Identity];

p = Show[{
   Graphics[{RGBColor[1, 0.5, 0], PointSize[0.02], points5}],
   Graphics[{RGBColor[1, 1, 0], PointSize[0.02], points1}],
   Graphics[{RGBColor[1, 0, 0], PointSize[0.02], points4}],
   Graphics[{RGBColor[0, 0.5, 0], PointSize[0.02], points2}],
   Graphics[{RGBColor[0, 0, 1], PointSize[0.02], points3}],
   Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.6, -0.95}]}],
   Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.8, -0.85}]}],
   Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-0.875, -0.75}]}],
   Graphics[{RGBColor[0, 0, 1], Text["S", {-0.95, 0.5}]}],
   Graphics[{RGBColor[0, 0.5, 0], Text["G1", {0.55, 0.95}]}],
   Graphics[{RGBColor[0, 0, 0], Text["(c)", {-0.9, 0.95}]}],
   Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 1]}],
   Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 0.5]}]},
  AspectRatio -> 1,
  PlotRange -> {{-1.05, 1.05}, {-1.05, 1.05}},
  Frame -> True,
  FrameTicks -> False,
  FrameLabel -> {labelx, labely, None, None},
  GridLines -> {{{0, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
  DisplayFunction -> Identity];
p = FullGraphics[p];
p[[1, 2]] = p[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {-1.18, 0}, {0, 0}, {0, 1}];
p3 = Show[p,
  AspectRatio -> 1.,
  PlotRange -> All,
  DisplayFunction -> Identity];

(* Display Both Arrays & Genes Parameter Graphs *)

Show[GraphicsArray[{p1, p2, p3}],
  GraphicsSpacing -> 0];
```



```
(* Sort Normalized Data *)
```

```
(* Define the Initial Phase According to the Initial Array *)
```

```
zerophase = N[ArcTan[arraycorrelations[[1, 1]] / arraycorrelations[[2, 1]]] / Pi]
```

```
-0.103286
```

```
(* Sort Data by Phase *)
```

```
radii = Table[Sqrt[coordinates[[a, 1]]^2 + coordinates[[a, 2]]^2], {a, 1, genes}];
coordinates = Table[
  {genecorrelations[[2, a]] / Sqrt[genecorrelations[[1, a]]^2 + genecorrelations[[2, a]]^2],
   genecorrelations[[1, a]] / Sqrt[genecorrelations[[1, a]]^2 + genecorrelations[[2, a]]^2],
   N[ArcTan[genecorrelations[[1, a]] / genecorrelations[[2, a]]] / Pi],
   radii[[a]]},
  {a, 1, genes}];
sortmatrix = AppendRows[coordinates, genenames, matrix];
sortmatrix = Sort[sortmatrix, OrderedQ[{{#1}, {#2}}]&];
negative1 = 3106;
positive1 = 3107;
sortmatrix[[negative1, 1]]
sortmatrix[[positive1, 1]]
```

```
-0.00016311
```

```
0.000620348
```

```
sortmatrix = Transpose[Drop[Transpose[sortmatrix], {1}]];
sortmatrix = AppendColumns[
  Sort[
   TakeRows[sortmatrix, {1, negative1}],
   OrderedQ[{{#2}, {#1}}]&],
  Sort[
   TakeRows[sortmatrix, {positive1, genes}],
   OrderedQ[{{#1}, {#2}}]&]
  ];
sortmatrix = Transpose[Drop[Transpose[sortmatrix], {1}]];
```

```
(* Correct for the Phase & Calibrate According to the Initial Phase *)

Do[sortmatrix[[a, 1]] = sortmatrix[[a, 1]] - zerophase,
 {a, 1, negative1}];
Do[sortmatrix[[a, 1]] = sortmatrix[[a, 1]] + 1 - zerophase,
 {a, positive1, genes}];
negative2 = 1322;
positive2 = 1323;
sortmatrix[[negative2, 1]]
sortmatrix[[positive2, 1]]
```

```
-0.000992767

0.00072612
```

```
sortmatrix = AppendColumns[
  TakeRows[sortmatrix, {positive2, genes}],
  TakeRows[sortmatrix, {1, negative2}]
  ];
positive3 = 4659;
negative3 = 4660;
sortmatrix[[positive3, 1]]
sortmatrix[[negative3, 1]]
```

```
1.60309

-0.395832
```

```
Do[sortmatrix[[a, 1]] = sortmatrix[[a, 1]] + 2,
 {a, negative3, genes}];

(* Reconstruct Data With Sorted Genes *)

matrix = TakeColumns[sortmatrix, {4, arrays + 3}];
eigenarrays = Dot[eigengenes, Transpose[matrix]];
Do[
 eigenarrays[[a]] = eigenarrays[[a]] / eigenexpressions[[a]],
 {a, 1, arrays}];
eigenarrays = Transpose[eigenarrays];
arraycorrelations = Dot[DiagonalMatrix[eigenexpressions], eigengenes];
genecorrelations = Dot[eigenarrays, DiagonalMatrix[eigenexpressions]];
genecorrelations = Transpose[genecorrelations];
```

```
(* Classify Gene Phases into Cell Cycle Phases *)

phases = TakeColumns[sortmatrix, {1, 1}];
ph1 = -zerophase;
ph2 = 0.5 - zerophase;
ph3 = 1 - zerophase;
ph4 = 1.25 - zerophase;
ph5 = 1.5 - zerophase;

endph5 = 260;
beginph1 = 261;
phases[[endph5]] - ph1
phases[[beginph1]] - ph1

{-0.000367577}

{0.000229068}

endph1 = 1784;
beginph2 = 1785;
phases[[endph1]] - ph2
phases[[beginph2]] - ph2

{-0.0000519195}

{0.000528725}

endph2 = 3128;
beginph3 = 3129;
phases[[endph2]] - ph3
phases[[beginph3]] - ph3

{-0.00136339}

{0.000154703}

endph3 = 3816;
beginph4 = 3817;
phases[[endph3]] - ph4
phases[[beginph4]] - ph4

{-0.000410391}

{0.000773805}

endph4 = 4659;
beginph5 = 4660;
phases[[endph4]] - ph5
phases[[beginph5]] - ph5

{-0.000197463}

{0.00088155}

(* 5891 yeast genes, 1582 in M/G1, 1524 in G1, 1344 in S, 688 in S/G2, 843 in G2/M. *)
(* Create Classified Sorted Data 2 D Red & Green Raster Display *)

contrast = 2;
displaying = Table[
  If[contrast * matrix[[i, j]] > 0,
   If[contrast * matrix[[i, j]] < 1, {contrast * matrix[[i, j]], 0}, {1, 0}],
   If[contrast * matrix[[i, j]] > -1, {0, -contrast * matrix[[i, j]]}, {0, 1}]],
  {i, 1, genes}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[2, a]]}, {a, 1, arrays}];
framey = {
 {genes - endph5 / 2, "M/G1"},
 {genes - (endph5 + endph1) / 2, "G1"},
 {genes - (endph1 + endph2) / 2, "S"},
 {genes - (endph2 + endph3) / 2, "S/G2"},
 {genes - (endph3 + endph4) / 2, "G2/M"},
 {(genes - endph4) / 2, "M/G1"}};
gridy = {
  {genes - endph1 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph2 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph3 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph4 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph5 + 0.5, {RGBColor[0, 0, 0]}}};
labelx = "(a)  Arrays";
labely = ColumnForm[{" ", "Genes", " ", " ", " ", " ", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, genes, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1,
 Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, gridy},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {1., 0.}] ->
   Text[a, {b - 1, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 1000}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 450}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

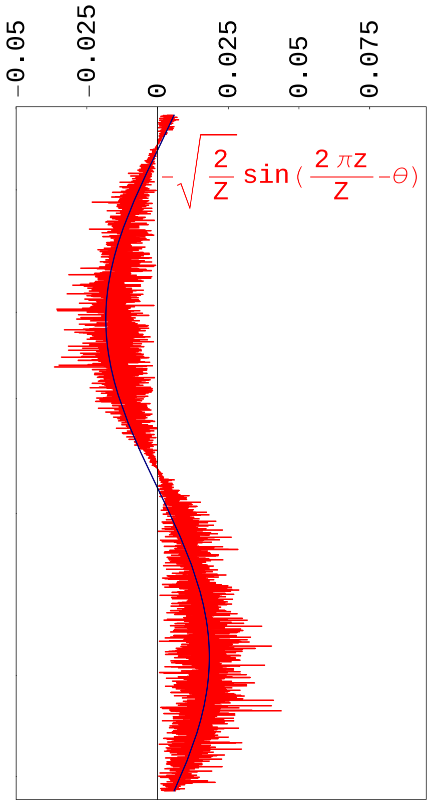
```
(* Create Classified Sorted Eigenarrays 2 D Red & Green Raster Display *)

contrast = 50;
displaying = Table[
  If[contrast * eigenarrays[[i, j]] > 0,
   If[contrast * eigenarrays[[i, j]] < 1, {contrast * eigenarrays[[i, j]], 0}, {1, 0}],
   If[contrast * eigenarrays[[i, j]] > -1, {0, -contrast * eigenarrays[[i, j]]}, {0, 1}]],
  {i, 1, genes}, {j, 1, arrays}];
labelx = "(b) Eigenarrays";
labely = ColumnForm[{" ", " ", " ", " ", " ", " ", " ", " "}, Center];
framex = Table[{a - 0.5, ToString[a]}, {a, 1, arrays}];
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     framex[[a, 2]]]], {a, 1, arrays}]];
Do[
 Do[framex[[a, 2]] = StringJoin[framex[[a, 2]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, arrays}];
framey = {
 {genes - endph5 / 2, "    ", 0},
 {genes - (endph5 + endph1) / 2, "  ", 0},
 {genes - (endph1 + endph2) / 2, " ", 0},
 {genes - (endph2 + endph3) / 2, "    ", 0},
 {genes - (endph3 + endph4) / 2, "    ", 0},
 {(genes - endph4) / 2, "    ", 0}};
gridy = {
  {genes - endph1 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph2 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph3 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph4 + 0.5, {RGBColor[0, 0, 0]}},
  {genes - endph5 + 0.5, {RGBColor[0, 0, 0]}}};
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, genes, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1,
 Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 GridLines -> {None, gridy},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {1., 0.}] ->
   Text[a, {b - 1, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 1000}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 450}, {0, 0}, {0, 1}];
g2 = Show[g,
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
(* Create Classified Selected Eigenarrays Graph Display *)

eigenarrays = Transpose[eigenarrays];

eigenarrays1 = Chop[TrigFit[eigenarrays[[1]], 1, {x, genes - 1}], 0.0025]
eigenarrays2 = Chop[TrigFit[eigenarrays[[2]], 1, {x, genes - 1}], 0.0025]
```

$$0.00389157\,\mathrm{Cos}\left[\frac{\pi x}{2990}\right] - 0.0166061\,\mathrm{Sin}\left[\frac{\pi x}{2990}\right]$$

$$-0.0165388\,\mathrm{Cos}\left[\frac{\pi x}{2990}\right] - 0.00401255\,\mathrm{Sin}\left[\frac{\pi x}{2990}\right]$$

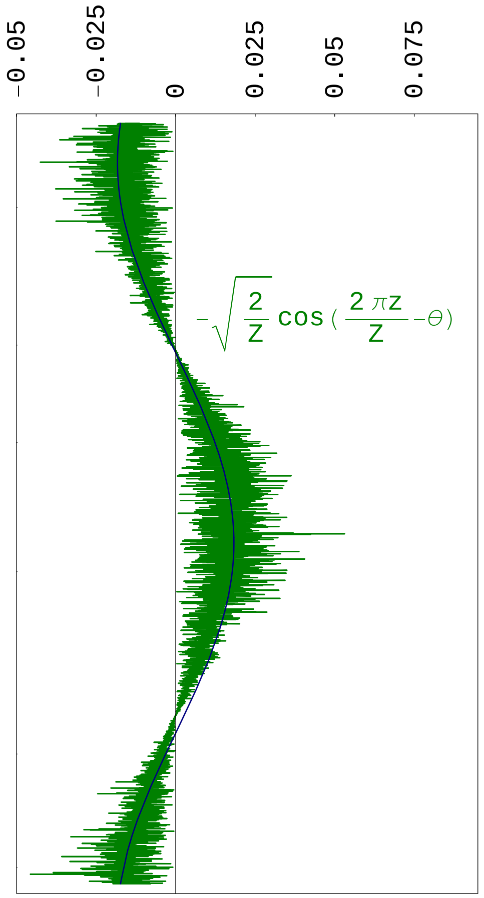
```
zerophase = N[ArcTan[arraycorrelations[[1, 1]] / arraycorrelations[[2, 1]]] / Pi];
eigenarrays1 = -0.0183 * Sin[Pi * (x + Round[zerophase * 2990]) / 2990];
eigenarrays2 = -0.0183 * Cos[Pi * (x + Round[zerophase * 2990]) / 2990];

graph = ParametricPlot[{eigenarrays1, -x},
  {x, genes - 1, 0},
  PlotStyle -> {RGBColor[0, 0, 0.5]},
  DisplayFunction -> Identity];
labelx = "(c) Expression Level";
labely = ColumnForm[{" ", " ", " ", " ", " ", " ", " ", " "}, Center];
framex = {
  {-0.05, "-0.05 "}, {-0.025, "-0.025"}, {0, "0      "},
  {0.025, "0.025 "}, {0.05, "0.05  "}, {0.075, "0.075 "}};
framey = {
 {-genes + endph5 / 2, "    ", 0},
 {-genes + (endph5 + endph1) / 2, "  ", 0},
 {-genes + (endph1 + endph2) / 2, " ", 0},
 {-genes + (endph2 + endph3) / 2, "    ", 0},
 {-genes + (endph3 + endph4) / 2, "    ", 0},
 {-(genes - endph4) / 2, "    ", 0}};
coordinates = Table[{eigenarrays[[1, a]], -a + 1},  {a, 1, genes}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[1, 0, 0], line}],
   graph,
   Graphics[{RGBColor[1, 0, 0], Text["-√(2/Z) sin(2πz/Z-θ)", {0.0475, -500}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 FrameTicks -> {None, framey, framex, None},
 GridLines -> {{{0, RGBColor[0, 0, 0]}}, None},
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> {{-0.05, 0.095}, {75, -genes - 75 + 1}},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {1., 0.}] ->
   Text[a, {b - 0.0095, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 1000}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 450}, {0, 0}, {0, 1}];
p1 = Show[g,
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> All];
```



(c) Expression Level

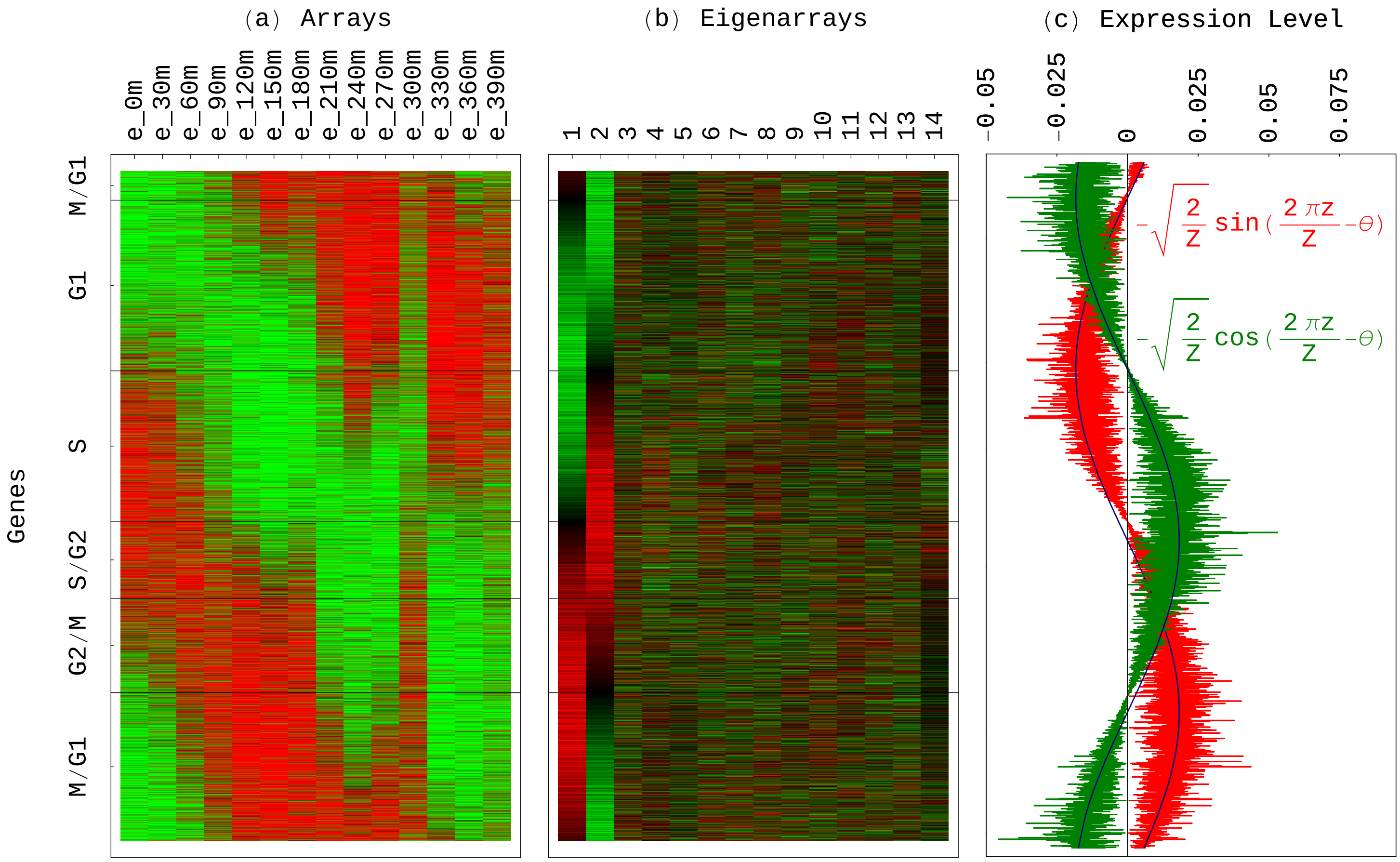
```
graph = ParametricPlot[{eigenarrays2, -x},
  {x, 0, genes - 1},
  PlotStyle -> {RGBColor[0, 0, 0.5]},
  DisplayFunction -> Identity];
labelx = "(c) Expression Level";
labely = ColumnForm[{" ", " ", " ", " ", " "}, Center];
framex = {
  {-0.05, "-0.05 "}, {-0.025, "-0.025"}, {0, "0     "},
  {0.025, "0.025 "}, {0.05, "0.05  "}, {0.075, "0.075 "}};
framey = {
 {-genes + endph5 / 2, "    ", 0},
 {-genes + (endph5 + endph1) / 2, "  ", 0},
 {-genes + (endph1 + endph2) / 2, " ", 0},
 {-genes + (endph2 + endph3) / 2, "    ", 0},
 {-genes + (endph3 + endph4) / 2, "    ", 0},
 {-(genes - endph4) / 2, "    ", 0}};
coordinates = Table[{eigenarrays[[2, a]], -a + 1},  {a, 1, genes}];
line = Line[coordinates];
g = Show[
  {Graphics[{RGBColor[0, 0.5, 0], line}],
   graph,
   Graphics[{RGBColor[0, 0.5, 0], Text["-√(2/Z) cos(2 πz/Z-θ)", {0.0475, -1500}]}]},
 Frame -> True,
 FrameLabel -> {None, labely, labelx, None},
 FrameTicks -> {None, framey, framex, None},
 GridLines -> {{{0, RGBColor[0, 0, 0]}}, None},
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> {{-0.05, 0.095}, {75, -genes + 1 - 75}},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {1., 0.}] ->
   Text[a, {b - 0.0095, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 1000}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 450}, {0, 0}, {0, 1}];
p2 = Show[g,
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> All];
```

(c) Expression Level



```
(* Display Classified Selected Sorted Eigenarrays *)

g3 = Show[{p1, p2},
 DisplayFunction -> Identity];
```

```
(* Display Classified Sorted Data, Eigenarrays and Selected Eigenarrays *)

Show[GraphicsArray[{g1, g2, g3}],
 GraphicsSpacing -> -0.225];
```



```
(* Save Sorted Normalized Data in Sort_Data.txt *)

sortmatrix = AppendRows[
  Table[{a}, {a, 1, genes}],
  N[Round[TakeColumns[sortmatrix, {1, 2}] * 1000] / 1000],
  TakeColumns[sortmatrix, {3, 3}],
  N[Round[TakeColumns[sortmatrix, {4, arrays + 3}] * 100000] / 100000]];
sortmatrix = AppendColumns[
  AppendRows[{{" ", " ", " ", " "}, {" ", "Phase", "Radius", "UID"}}, arraynames],
  sortmatrix];
stream = OpenWrite[
  StringJoin[name, ":Desktop Folder:PNAS Data:Sort_Elutriation.nb"],
  PageWidth -> Infinity];
Write[
 stream,
 OutputForm[
  TableForm[sortmatrix, TableSpacing -> {0, 1}]
 ]];
Close[stream];
Clear[sortmatrix];
```

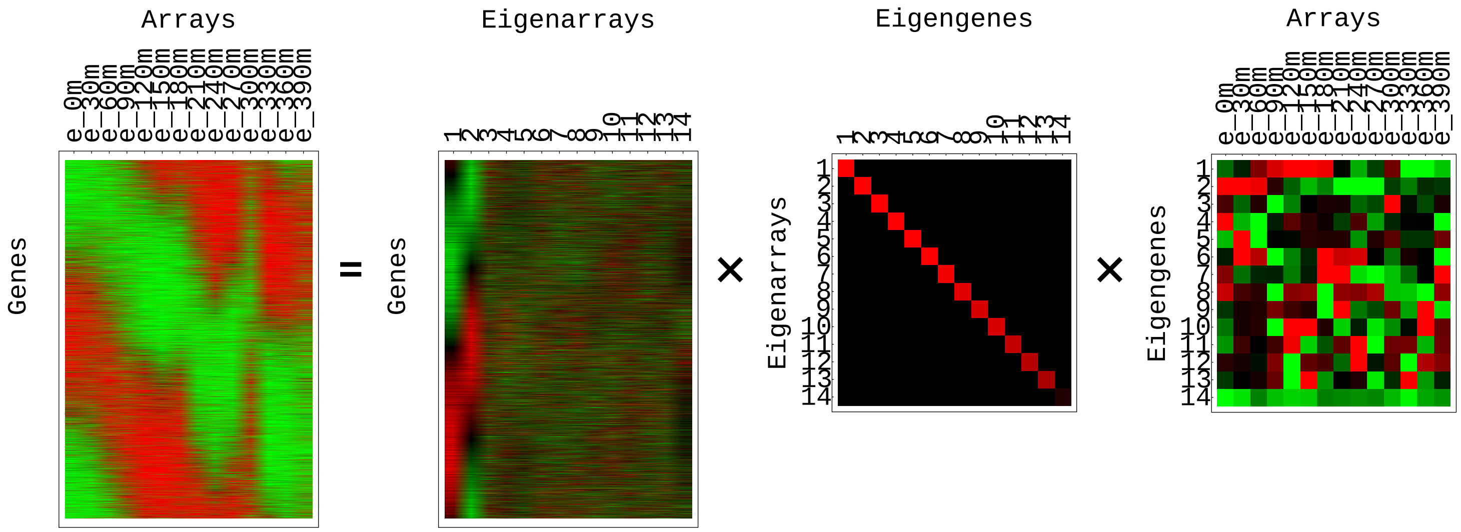
```
(* Display Singular Value Decomposition of Sorted Normalized Data  *)

eigenarrays = Transpose[eigenarrays];

(* Create Sorted Data 2 D Red & Green Raster Display *)

contrast = 2;
displaying = Table[
  If[contrast * matrix[[i, j]] > 0,
   If[contrast * matrix[[i, j]] < 1, {contrast * matrix[[i, j]], 0}, {1, 0}],
   If[contrast * matrix[[i, j]] > -1, {0, -contrast * matrix[[i, j]]}, {0, 1}]],
  {i, 1, genes}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[2, a]]}, {a, 1, arrays}];
labelx = "Arrays";
labely = ColumnForm[{" ", "        Genes", " ", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, genes, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, None, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 1850}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 850}, {0, 0}, {0, 1}];
g1 = Show[g,
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
(* Create Sorted Eigenarrays 2 D Red & Green Raster Display *)

contrast = 50;
displaying = Table[
  If[contrast * eigenarrays[[i, j]] > 0,
   If[contrast * eigenarrays[[i, j]] < 1, {contrast * eigenarrays[[i, j]], 0}, {1, 0}],
   If[contrast * eigenarrays[[i, j]] > -1, {0, -contrast * eigenarrays[[i, j]]}, {0, 1}]],
  {i, 1, genes}, {j, 1, arrays}];
labelx = "Eigenarrays";
labely = ColumnForm[{" ", "        Genes", " ", " ", " "}, Center];
framex = Table[{a - 0.5, ToString[a]}, {a, 1, arrays}];
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     framex[[a, 2]]
    ]
   ], {a, 1, arrays}]];
Do[
 Do[framex[[a, 2]] = StringJoin[framex[[a, 2]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, arrays}];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, genes, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, None, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 1850}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 850}, {0, 0}, {0, 1}];
g2 = Show[g,
 AspectRatio -> GoldenRatio * 1.15,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

```
(* Create Eigenexpressions 2 D Red & Green Raster Display *)

contrast = 0.0075;
eigenexpression = DiagonalMatrix[eigenexpressions];
displaying = Table[
  If[contrast * eigenexpression[[i, j]] > 0,
   If[contrast * eigenexpression[[i, j]] < 1, {contrast * eigenexpression[[i, j]], 0}, {1, 0}],
   If[contrast * eigenexpression[[i, j]] > -1, {0, -contrast * eigenexpression[[i, j]]}, {0, 1}]],
  {i, 1, arrays}, {j, 1, arrays}];
framex = Table[a, {a, 1, arrays}];
sizes = Flatten[
  Table[
   Dimensions[
    Characters[
     ToString[framex[[a]]]
    ]
   ], {a, 1, arrays}]];
Do[
 Do[framex[[a]] = StringJoin[ToString[framex[[a]]], " "],
    {b, 1, size - sizes[[a]]}],
   {a, 1, arrays}];
framex = Table[{a - 0.5, framex[[a]]}, {a, 1, arrays}];
framey = Table[{a + 1 - 0.5, arrays - a}, {a, 0, arrays - 1}];
labely = ColumnForm[{" ", "Eigenarrays", " "}, Center];
labelx = ColumnForm[{"Eigengenes", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, arrays, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 3}, {0, 0}, {0, 1}];
g3 = Show[g,
 AspectRatio -> 1.05,
 PlotRange -> All,
 DisplayFunction -> Identity];
```

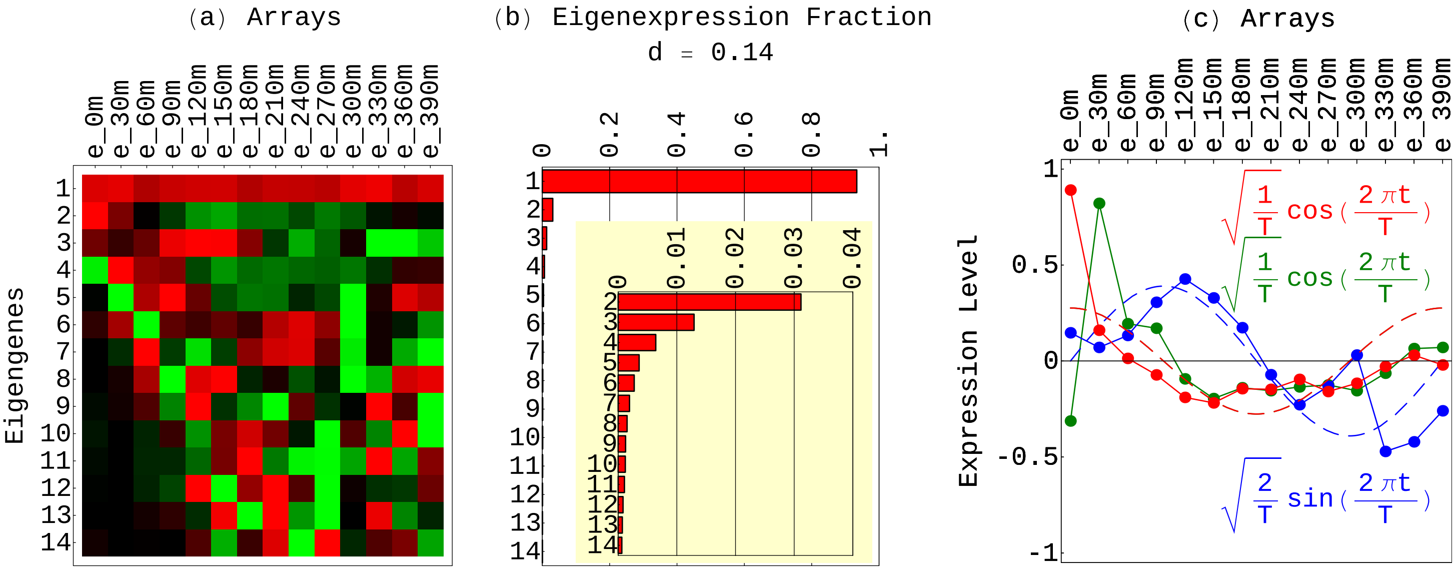
```
(* Create Eigengenes 2 D Red & Green Raster Display *)

contrast = 3;
displaying = Table[
  If[contrast * eigengenes[[i, j]] > 0,
   If[contrast * eigengenes[[i, j]] < 1, {contrast * eigengenes[[i, j]], 0}, {1, 0}],
   If[contrast * eigengenes[[i, j]] > -1, {0, -contrast * eigengenes[[i, j]]}, {0, 1}]],
  {i, 1, arrays}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[2, a]]}, {a, 1, arrays}];
framey = Table[{a + 1 - 0.5, arrays - a}, {a, 0, arrays - 1}];
labely = ColumnForm[{" ", "Eigengenes", " "}, Center];
labelx = ColumnForm[{"Arrays", " ", " "}, Center];
g = Show[
 Graphics[
  RasterArray[
   Table[
    RGBColor[
     displaying[[i, j, 1]], displaying[[i, j, 2]], 0
     ],
    {i, arrays, 1, -1}, {j, 1, arrays}
   ]]],
 AspectRatio -> 1, Frame -> True,
 FrameTicks -> {None, framey, framex, None},
 FrameLabel -> {None, labely, labelx, None},
 DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
   Text[labely, {b - 3, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., -1.}] ->
   Text[labelx, {b, c + 2.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
   Text[a, {b, c + 3.}, {0, 0}, {0, 1}];
g4 = Show[g,
 AspectRatio -> 1.05,
 PlotRange -> All,
 DisplayFunction -> Identity];

(* Display Singular Value Decomposition of Sorted Normalized Data *)

equal = Show[Graphics[
   Text[StyleForm["=", FontSize -> 20, FontWeight -> Bold], {0, 0}]
  ], DisplayFunction -> Identity];
times = Show[Graphics[
   Text[StyleForm["×", FontSize -> 20, FontWeight -> Bold], {0, 0}]
  ], DisplayFunction -> Identity];
```

```
Show[{
  Graphics[{Rectangle[{0, 0}, {175, 325}, g1]}],
  Graphics[{Rectangle[{170, 0}, {205, 325}, equal]}],
  Graphics[{Rectangle[{185, 0}, {360, 325}, g2]}],
  Graphics[{Rectangle[{355, 0}, {390, 325}, times]}],
  Graphics[{Rectangle[{370, 42.5}, {545, 325}, g3]}],
  Graphics[{Rectangle[{540, 0}, {575, 325}, times]}],
  Graphics[{Rectangle[{555, 42.5}, {730, 325}, g4]}]}],
 PlotRange -> All];
```

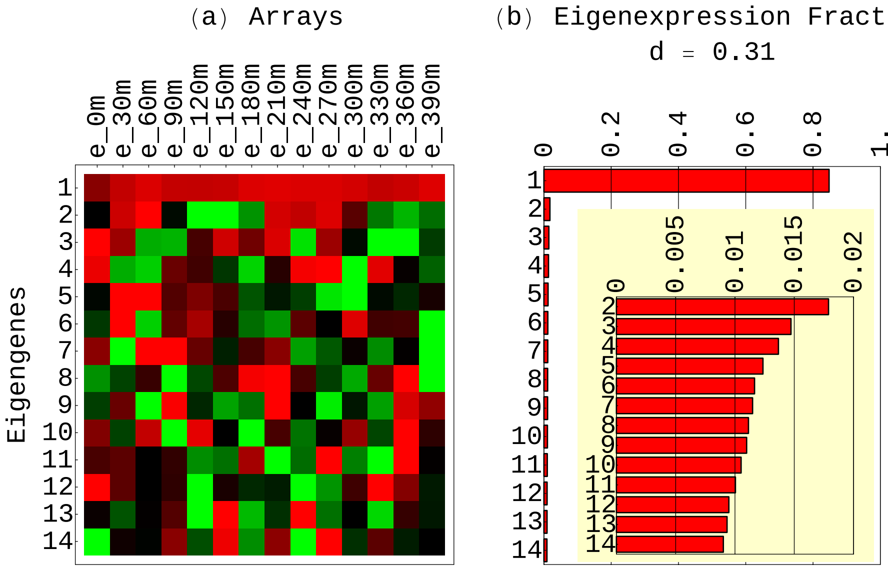


(* Summarize Elutriation Analysis *)

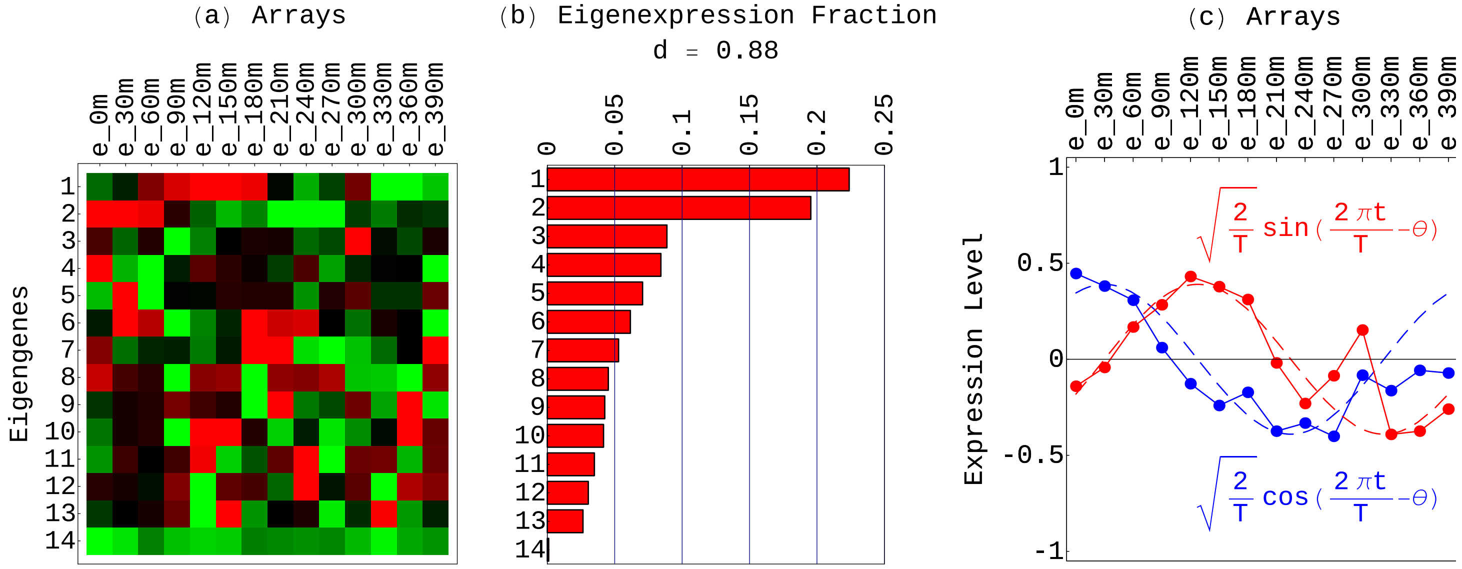
(* Centering by Removing the Additive Constant, i.e., the Steady State *)



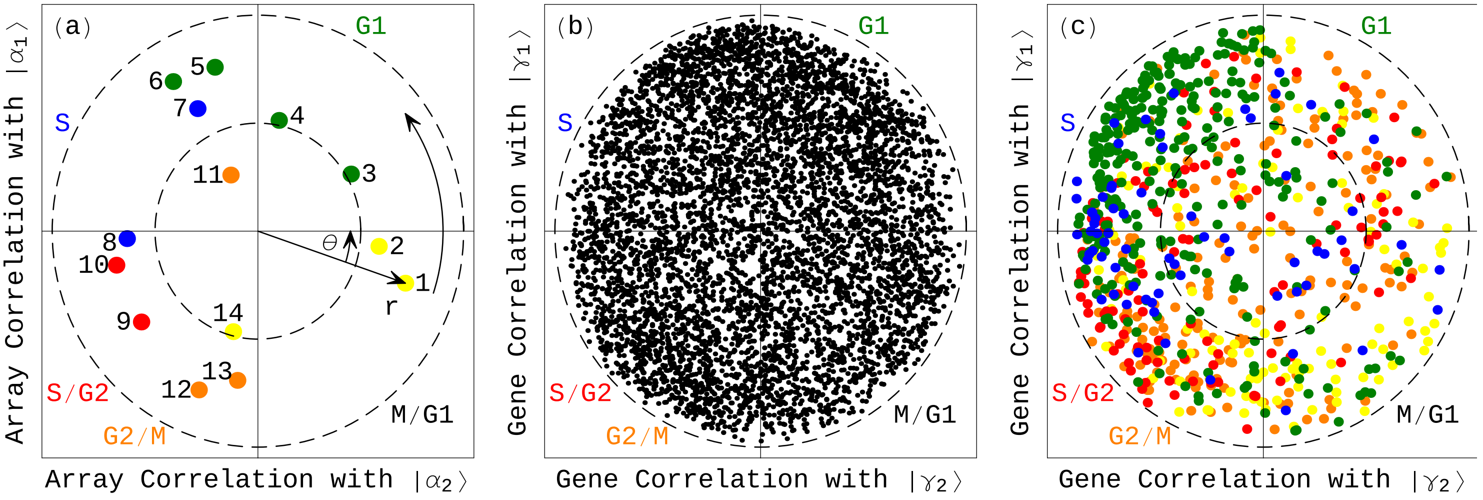
(* Normalizing the Variances by Removing the Multiplicative Constant *)



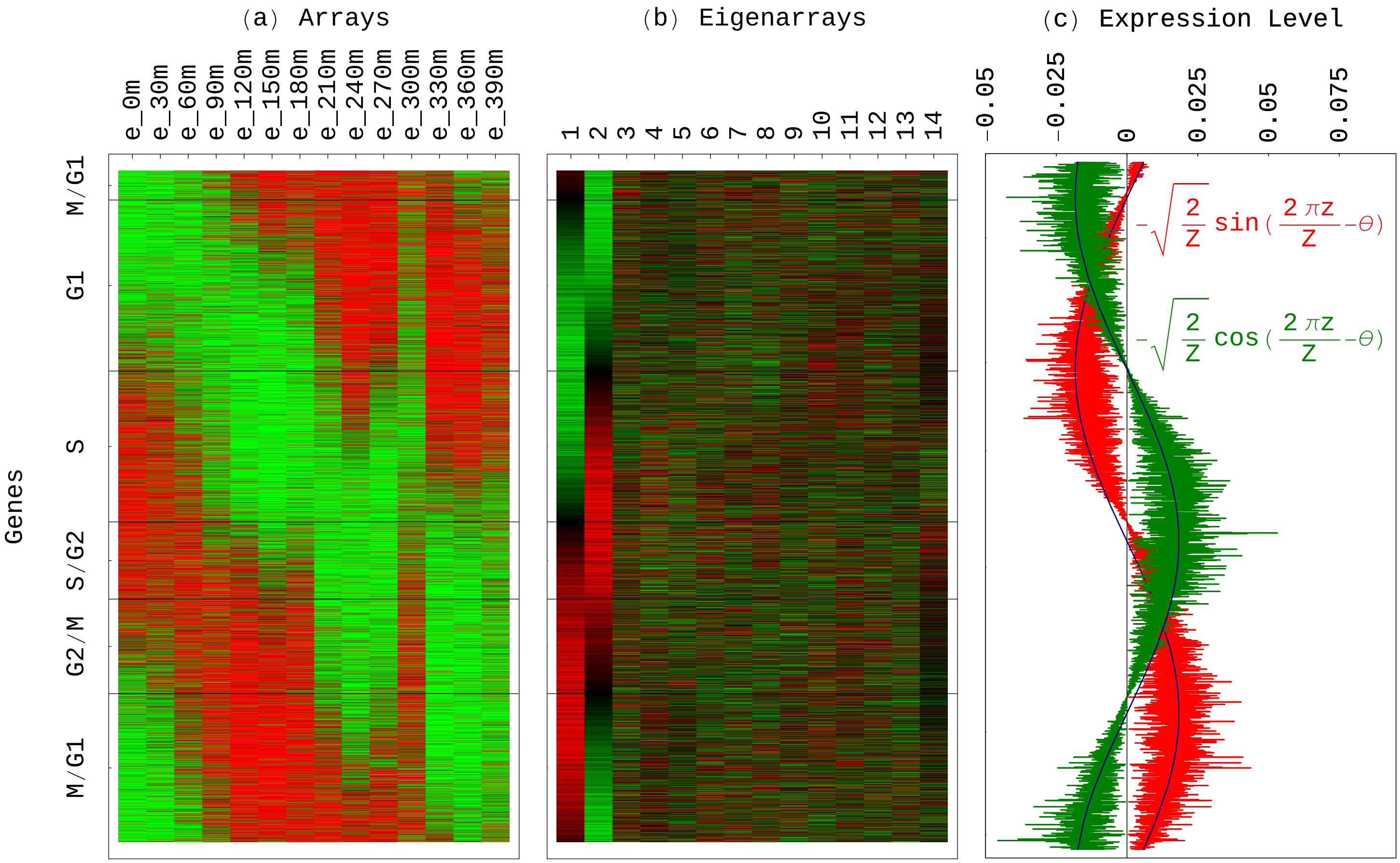
(* Normalized Data *)



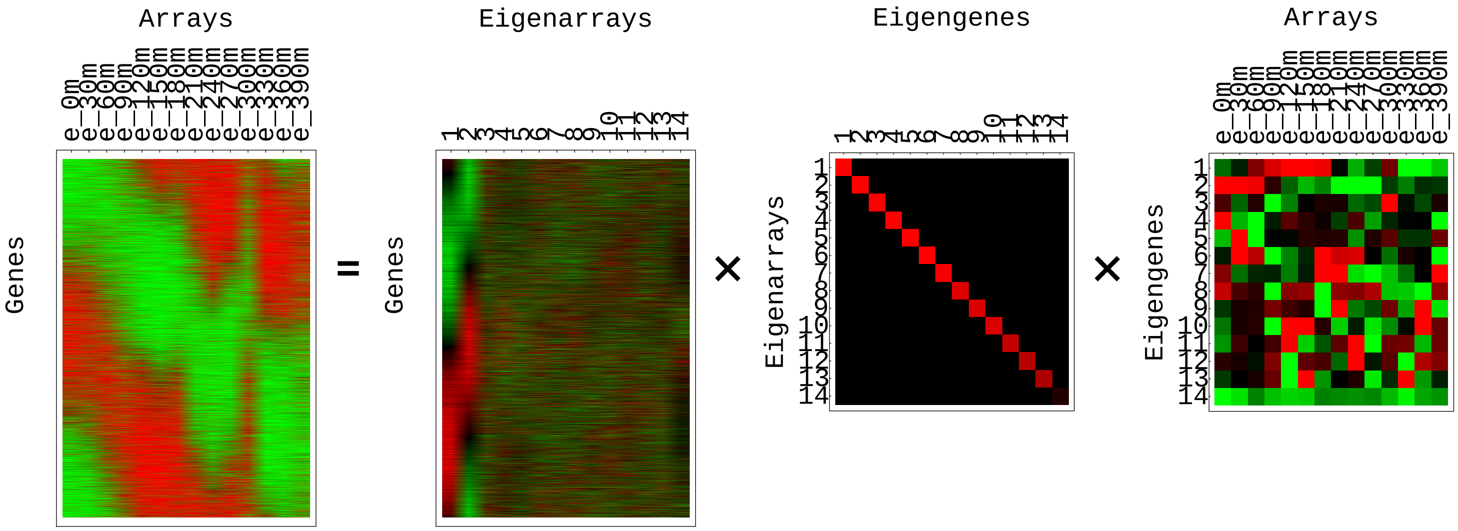
(* Sorting the Normalized Data *)



```
(* Classified Sorted Normalized Data *)
```



```
(* Summarized SVD *)
```


Arrays
e_0m
e_30m
e_60m
e_90m
e_120m
e_150m
e_180m
e_210m
e_240m
e_270m
e_300m
e_330m
e_360m
e_390m
Genes
=
Eigenarrays
Genes
×
Eigengenes
Eigenarrays
×
Arrays
e_0m
e_30m
e_60m
e_90m
e_120m
e_150m
e_180m
e_210m
e_240m
e_270m
e_300m
e_330m
e_360m
e_390m
Eigengenes