

```
(* © Orly Alter 2001 *)
```

```
(* All Rights Reserved *)
```

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

```
(* Estimate Missing Data *)
```

```
(* Read Raw Data from Raw_Data.txt *)
```

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

```
(* Define HardDrive *)
```

```
name = "Marzipan";
```

```
(* Elutriation *)
```

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

```
{6113, 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}];
```

```

(* Count Null Data *)

counter = Table[0, {genes}];
Do[
  counter[[a]] = counter[[a]] + Count[matrix[[a]], Null],
  {a, 1, genes}];

(* Locate Gene Position of Null Data *)

Clear[positions];
positions = Table[0, {a, 1, arrays + 1}];
Do[
  positions[[a]] = Flatten[
    Position[Flatten[counter], a - 1]],
  {a, 1, arrays + 1}];
numbers = Flatten[
  Table[
    Dimensions[positions[[a]],
    {a, 1, arrays + 1}]];

(* Create Display of Gene Position of Null Data *)

framex = Table[a - 1, {a, 2, arrays + 1}];
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, framex[[a]]}, {a, 1, arrays}];
framey = {20, 40, 60, 80, 100};
g = BarChart[Table[numbers[[a]], {a, 2, arrays + 1}],
  Frame -> True,
  Axes -> False,
  FrameTicks -> {framex, framey, None, None},
  PlotRange -> {{0.5, arrays + 0.5}, {0., 100}},
  DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., 1.}] ->
  Text[a, {b, c - 16}, {0, 0}, {0, 1}];
g1 = Show[g,
  AspectRatio -> 13 / (arrays + 1),
  PlotRange -> All,
  DisplayFunction -> Identity];

```

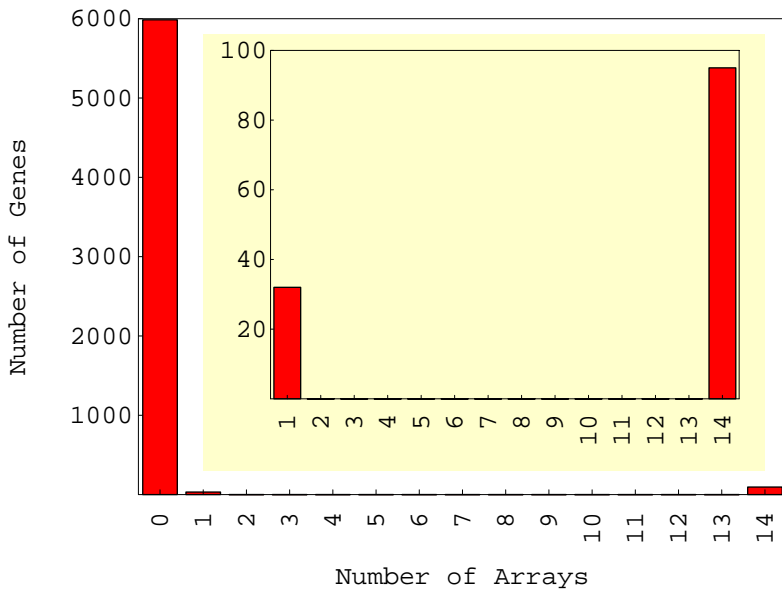
```

framex = Table[a - 1, {a, 1, arrays + 1}];
sizes = Flatten[
  Table[
    Dimensions[
      Characters[
        ToString[framex[[a]]]
      ]
    ], {a, 1, arrays + 1}];
Do[
  Do[framex[[a]] = StringJoin[" ", ToString[framex[[a]]]],
    {b, 1, size - sizes[[a]]},
    {a, 1, arrays + 1}];
framex = Table[{a, framex[[a]]}, {a, 1, arrays + 1}];
framey = {1000, 2000, 3000, 4000, 5000, 6000};
labelx = ColumnForm[{"Number of Arrays"}, Center];
labely = ColumnForm[{"Number of Genes"}, Center];
g = BarChart[numbers,
  Frame -> True,
  Axes -> False,
  FrameLabel -> {labelx, labely, None, None},
  FrameTicks -> {framex, framey, None, None},
  GridLines -> {None, None},
  PlotRange -> {{0.5, arrays + 1.5}, {0, 6000}},
  DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
  Text[labely, {b_, c_}, {1., 0.}] ->
  Text[labely, {b - 2.5, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
  Text[labelx, {b_, c_}, {0., 1.}] ->
  Text[labelx, {b, c - 700}, {0, 1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., 1.}] ->
  Text[a, {b, c - 700}, {0, 0}, {0, 1}];

(* Display Gene Position of Null Data *)

g2 = Show[{g,
  Graphics[{RGBColor[1, 1, 0.8], Rectangle[{2, 300}, {15, 5800}]}],
  Graphics[{Rectangle[{0.5, -250}, {16.5, 5750}, g1}]}],
  AspectRatio -> 10 / (arrays + 1),
  PlotRange -> All];

```



```
(* Select Genes by Number of Missing Data Points *)
```

```
matrix = AppendRows[Table[{counter[[a]]}, {a, 1, genes}], genenames, matrix];  
matrix = Sort[matrix, OrderedQ[{#1, #2}&];  
fullgenenames = TakeColumns[  
  TakeRows[matrix, {1, numbers[[1]]}],  
  {2, 2}];  
fullmatrix = TakeColumns[  
  TakeRows[matrix, {1, numbers[[1]]}],  
  {3, arrays + 2}];  
missinggenenames = TakeColumns[  
  TakeRows[matrix, {numbers[[1]] + 1, numbers[[1]] + numbers[[2]]}],  
  {2, 2}];  
missingmatrix = TakeColumns[  
  TakeRows[matrix, {numbers[[1]] + 1, numbers[[1]] + numbers[[2]]}],  
  {3, arrays + 2}];
```

```
(* Locate Array Position of Null Data *)
```

```
locator = Table[0, {numbers[[2]]}];  
Do[  
  locator[[a]] = locator[[a]] + Position[missingmatrix[[a]], Null][[1]],  
  {a, 1, numbers[[2]]}];
```

```
(* Sort Raw Data According to the Position of Missing Data Points for Each Gene *)
```

```
missingmatrix = AppendRows[locator, missinggenenames, missingmatrix];  
missingmatrix = Sort[missingmatrix, OrderedQ[{#1, #2}&];  
locator = TakeColumns[missingmatrix, {1, 1}];  
missinggenenames = TakeColumns[missingmatrix, {2, 2}];  
missingmatrix = TakeColumns[missingmatrix, {3, arrays + 2}];
```

```
(* Examine Subset of Genes with Full Data *)
```

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

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

```
0.14
```

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

```
probabilities[[2]]  
0.0311499  
  
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[probabilities[[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 Probability", StringJoin["d = ", ToString[entropy]], " "}, Center];
g = BarChart[
  Table[probabilities[[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

0.241246 Cos[ $\frac{2 \pi x}{13}$ ]

0.322334 Sin[ $\frac{2 \pi x}{13}$ ]

0.190306 Cos[ $\frac{2 \pi x}{13}$ ]

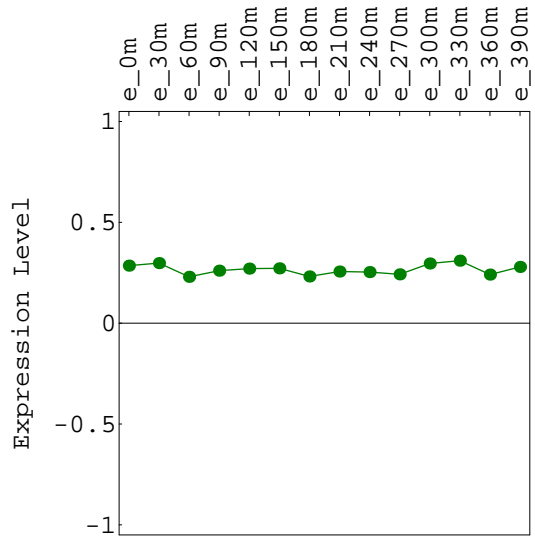
```

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];

```

(c) Arrays





```

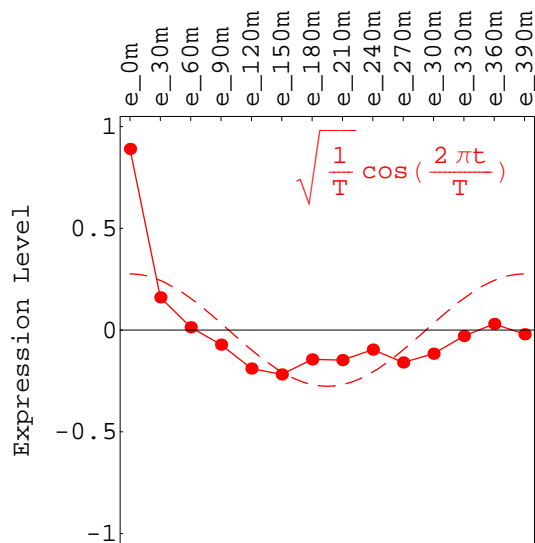
graph = Plot[
  eigengenes2,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[1, 0, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
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[" $\sqrt{\frac{1}{T}} \cos(\frac{2 \pi t}{T})$ ", {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];

```

(c) Arrays



```

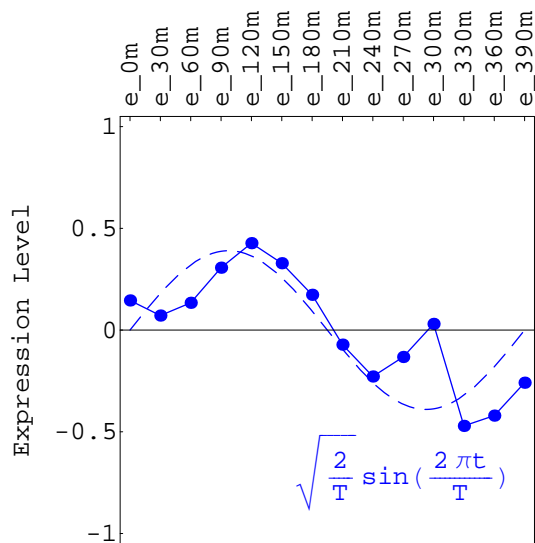
graph = Plot[
  eigengenes3,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[0, 0, 1], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
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[" $\sqrt{\frac{2}{T}} \sin(\frac{2\pi t}{T})$ ", {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];

```

(c) Arrays



```

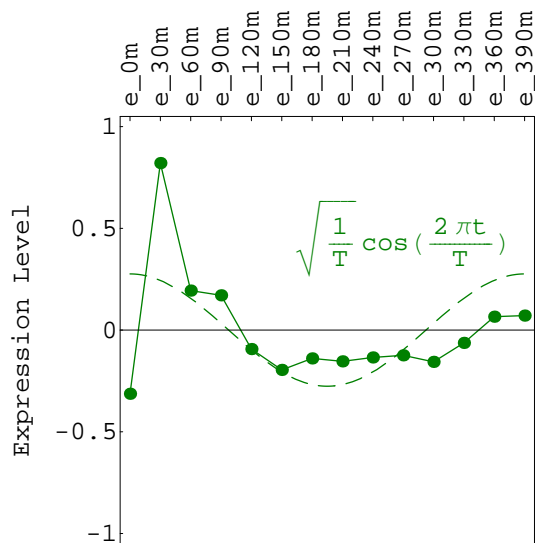
graph = Plot[
  eigengenes4,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[0, 0.5, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
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[" $\sqrt{\frac{1}{T}} \cos(\frac{2 \pi t}{T})$ ", {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];

```

(c) Arrays

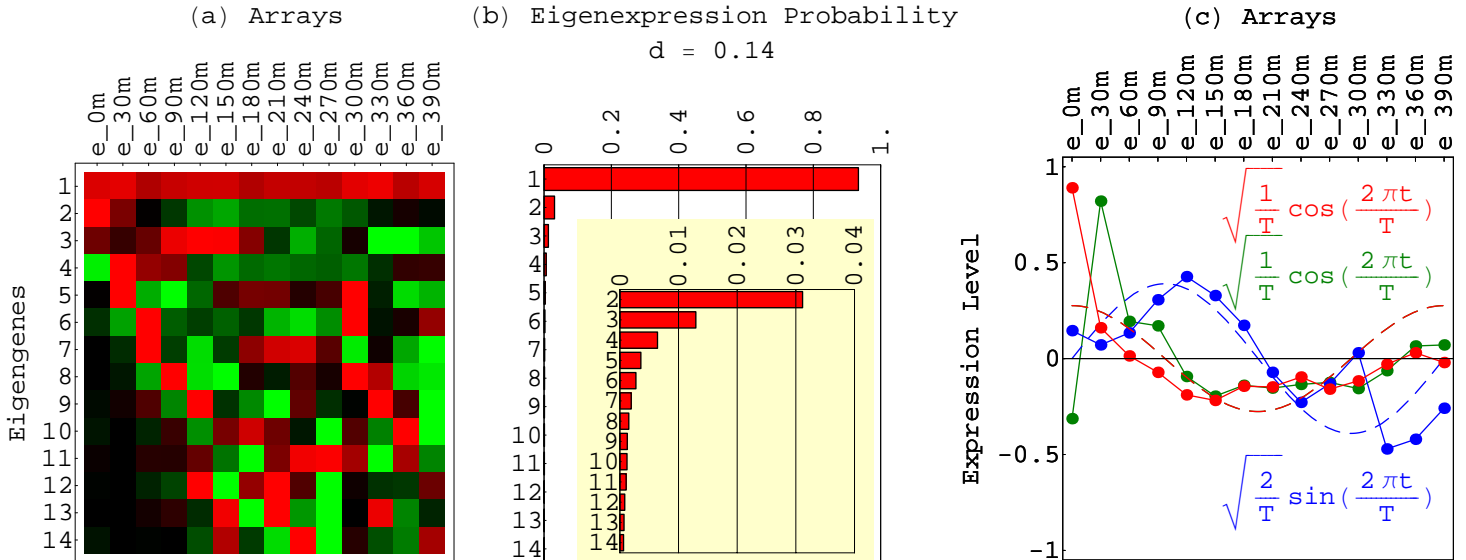


(\* Display Selected Eigengenes \*)

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

(\* Display Eigengenes, Probabilities and Selected Eigengenes \*)

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



(\* Choose Subset of Eigengenes for Estimation \*)

```
eigengenes = TakeRows[eigengenes, {1, 4}];
```

(\* Estimate Missing Data \*)

```
Do[
  missingmatrix[[a, locator[[a, 1]]]] =
  N[Round[Flatten[Dot[Dot[
    Transpose[Drop[
      Transpose[{missingmatrix[[a]]},
        {locator[[a, 1]}]}],
    PseudoInverse[Transpose[Drop[
      Transpose[eigengenes],
        {locator[[a, 1]}]}],
    eigengenes]][[locator[[a, 1]]] * 100] / 100],
  {a, 1, numbers[[2]]}];
  genenames = AppendColumns[fullgenenames, missinggenenames];
  matrix = AppendColumns[fullmatrix, missingmatrix];
  {genes, arrays} = Dimensions[matrix];

{6018, 14}
```

```
(* Normalize 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];  
probabilities = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, arrays}];  
entropy = -N[Sum[probabilities[[a]] * Log[probabilities[[a]]], {a, 1, arrays}] / Log[arrays];  
entropy = N[Round[100 * entropy] / 100]
```

```
0.14
```

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

```
probabilities[[2]]
```

```
0.0310586
```

```
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[probabilities[[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 Probability", StringJoin["d = ", ToString[entropy]], " ", Center];
g = BarChart[
  Table[probabilities[[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]

```

0.266138

0.241377 Cos[ $\frac{2\pi x}{13}$ ]

0.322557 Sin[ $\frac{2\pi x}{13}$ ]

0.188704 Cos[ $\frac{2\pi x}{13}$ ]

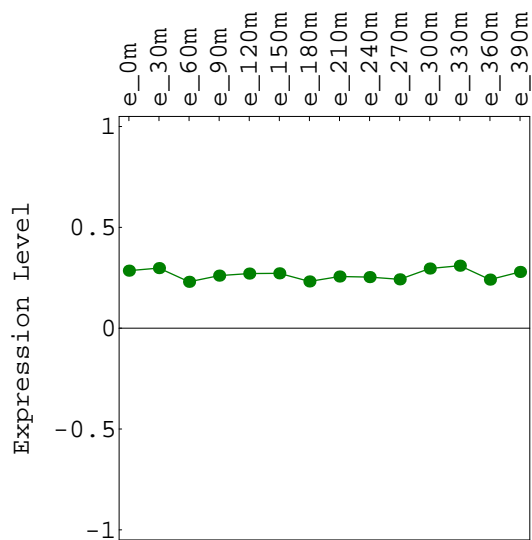
```

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];

```

(c) Arrays





```

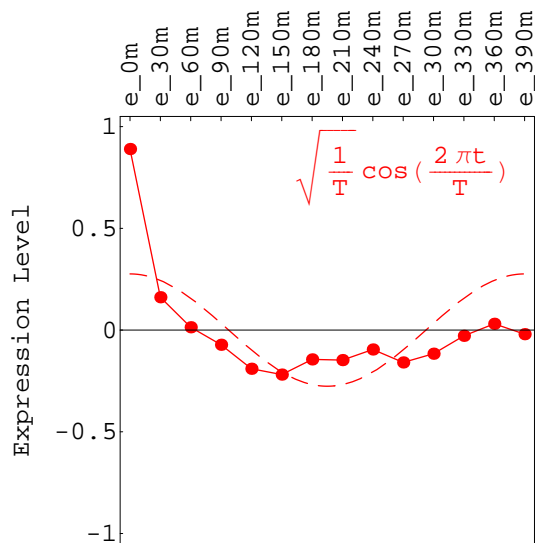
graph = Plot[
  eigengenes2,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[1, 0, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
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[" $\sqrt{\frac{1}{T}} \cos(\frac{2 \pi t}{T})$ ", {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];

```

(c) Arrays



```

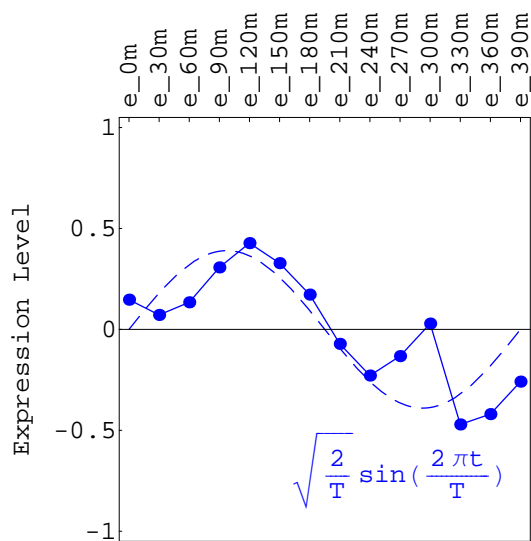
graph = Plot[
  eigengenes3,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[0, 0, 1], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
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[" $\sqrt{\frac{2}{T}} \sin(\frac{2 \pi t}{T})$ ", {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];

```

(c) Arrays

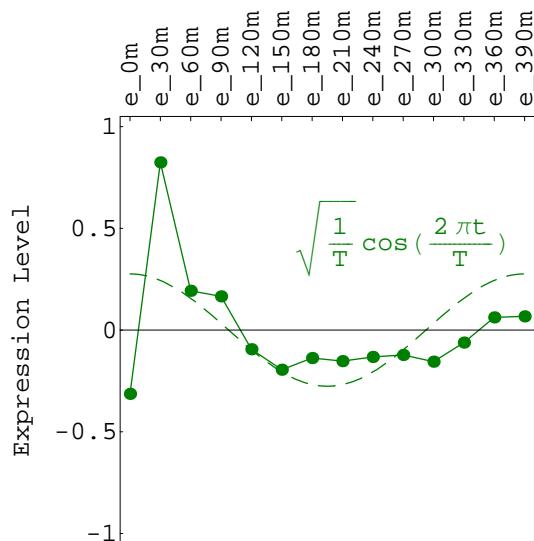


```

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[" $\sqrt{\frac{1}{T}} \cos(\frac{2 \pi t}{T})$ ", {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];

```

(c) Arrays



(\* Display Selected Eigengenes \*)

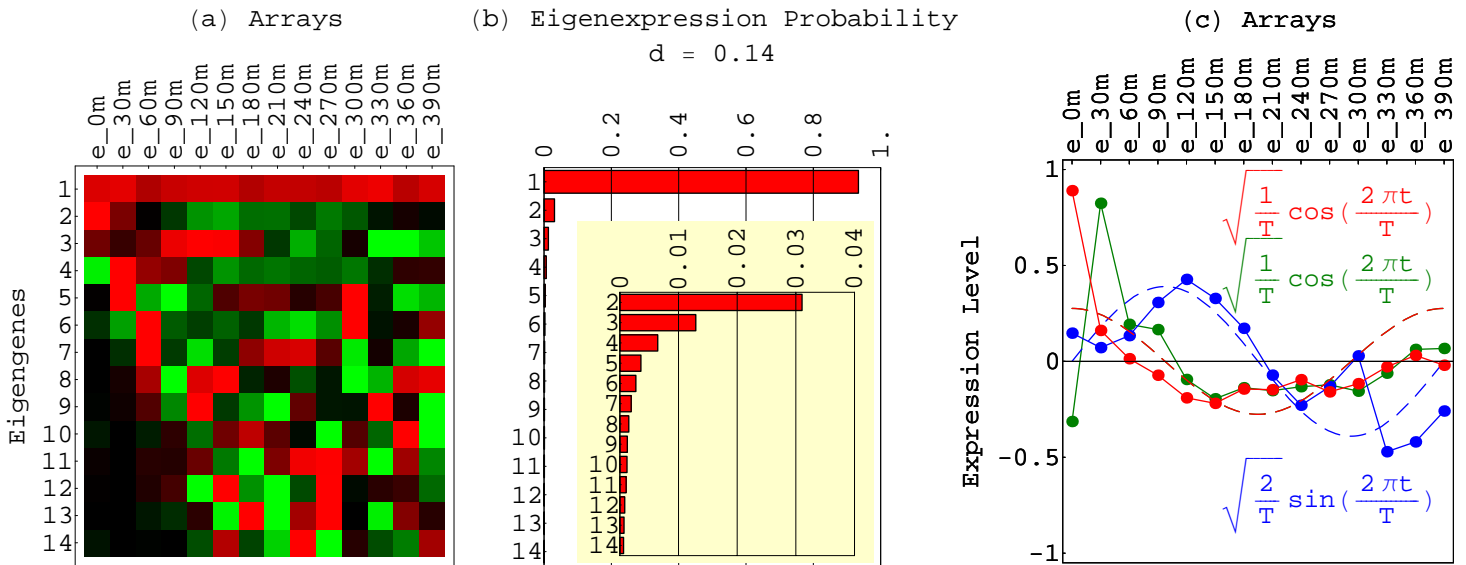
```

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

```

(\* Display Eigengenes, Probabilities 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];
probabilities = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[probabilities[[a]] * Log[probabilities[[a]]], {a, 1, arrays}] / Log[arrays];
entropy = N[Round[100 * entropy] / 100]
```

0.31

(\* Create Probabilities Bar Charts Displays \*)

```
probabilities[[2]]
```

0.0178757

```
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[probabilities[[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 Probability", StringJoin["d = ", ToString[entropy]], " "}, Center];

```

```

g = BarChart[
  Table[probabilities[[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 2D Red & Green Raster Display \*)

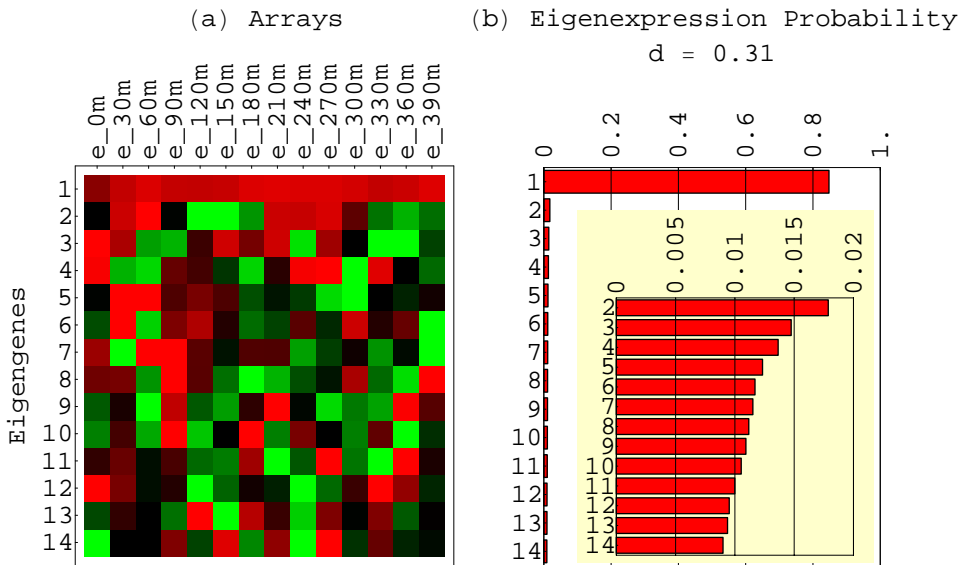
```

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];

```

(\* Display Both Eigengenes & Probabilities \*)

```
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[[2]] = -eigengenes[[2]];
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];
probabilities = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[probabilities[[a]] * Log[probabilities[[a]]], {a, 1, arrays}] / Log[arrays];
entropy = N[Round[100 * entropy] / 100]
```

0.88

(\* Create Probabilities Bar Chart Display \*)

```
probabilities[[1]]
```

0.223497

```

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 Probability", StringJoin["d = ", ToString[entropy]], " "}, Center];
g = BarChart[
  Table[probabilities[[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 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}], 0.12]
eigengenes2 = Chop[TrigFit[eigengenes[[2]], 2, {x, arrays - 1}], 0.1]

-0.135641 Cos[ $\frac{2 \pi x}{13}$ ] + 0.291085 Sin[ $\frac{2 \pi x}{13}$ ]

0.303266 Cos[ $\frac{2 \pi x}{13}$ ] + 0.124232 Sin[ $\frac{2 \pi x}{13}$ ]

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

```

```

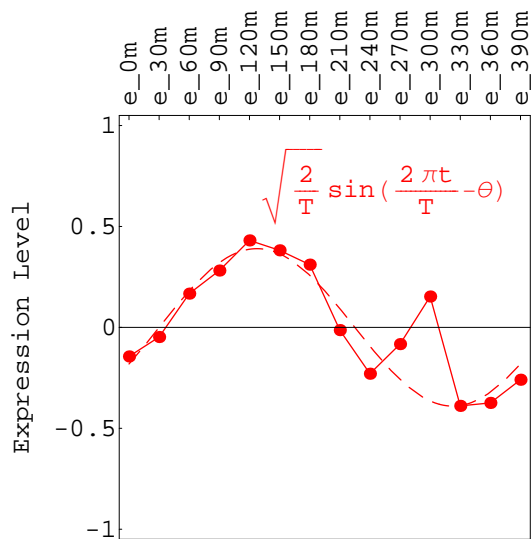
graph = Plot[
  eigengenes1,
  {x, 0, arrays - 1},
  PlotStyle -> {RGBColor[1, 0, 0], Dashing[{0.03, 0.02}]},
  DisplayFunction -> Identity];
labelx = ColumnForm[{"(c) Arrays"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, arraynames[[2, a]]}, {a, 1, arrays}];
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[" $\sqrt{\frac{2}{T}} \sin(\frac{2\pi t}{T} - \theta)$ ", {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];

```

(c) Arrays

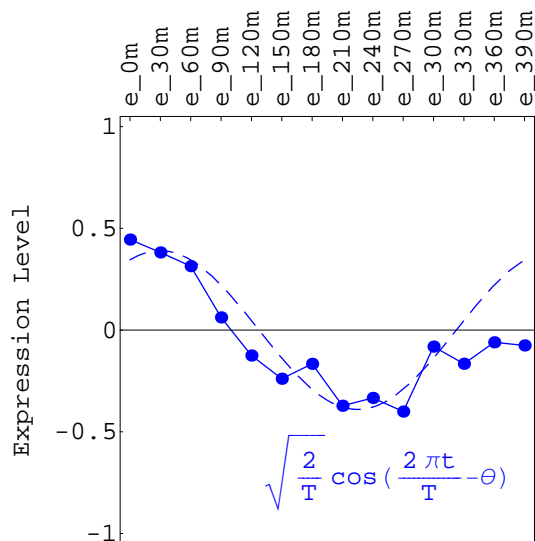


```

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[" $\sqrt{\frac{2}{T}} \cos(\frac{2\pi t}{T} - \theta)$ ", {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];

```

(c) Arrays



(\* Display Selected Eigengenes \*)

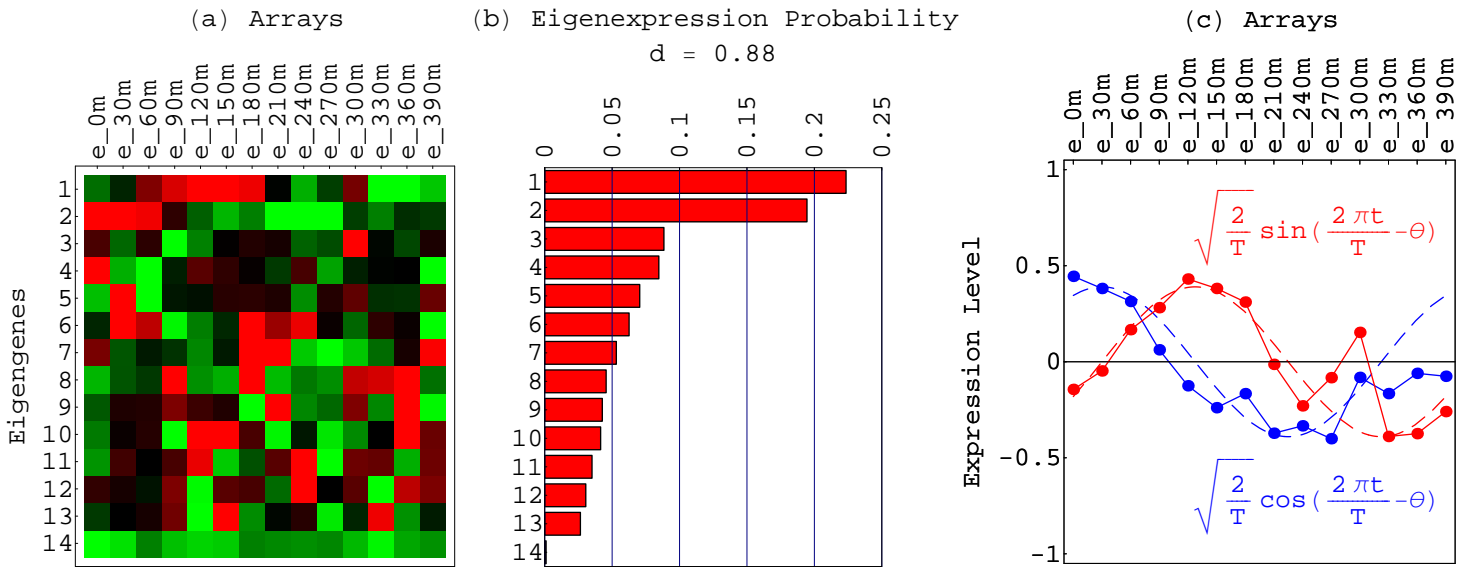
```

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

```

(\* Display Eigengenes, Probabilities 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  $|\alpha_2\rangle$ "}, Center];
labely = ColumnForm[{"Array Correlation with  $|\alpha_1\rangle$ "}, 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[" $\theta$ ", {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  $|\gamma_2\rangle$ "}, Center];
labely = ColumnForm[{"Gene Correlation with  $|\gamma_1\rangle$ "}, Center];
coordinates = Table[
{genecorrelations[[2, a]] / Sqrt[Dot[matrix[[a]], matrix[[a]]]},
genecorrelations[[1, a]] / Sqrt[Dot[matrix[[a]], matrix[[a]]]}],
{a, 1, genes}];

```

```
(* Create Parameter Graphs of 799 Cell Cycle Genes According to Projections on Eigengenes *)
```

```
stream = StringJoin[name, ":Desktop Folder:SPIE Data:Spellman_Classify_Cycle.txt"];
list = ReadList[stream, Word, RecordLists -> True, NullWords -> True];
list = Drop[list, 2];
stages = {"M/G1", "G1", "S", "S/G2", "G2/M"};
points = {points1, points2, points3, points4, points5};
radii = {radii1, radii2, radii3, radii4, radii5};
Do[{
  position = Position[list, stages[[b]]],
  table = Table[list[[position[[a, 1]], 1]], {a, 1, Dimensions[position][[1]]},
  position = Table[Position[genenames, table[[a]]], {a, 1, Dimensions[table][[1]]},
  table = Flatten[Position[position, {}]],
  Do[
    position = Drop[position, {table[[a]], table[[a]]}, {a, Dimensions[table][[1]], 1, -1}],
    points[[b]] = Table[Point[coordinates[[position[[a, 1, 1]]]], {a, 1, Dimensions[position][[1]]}],
    radii[[b]] = Table[
      Sqrt[coordinates[[position[[a, 1, 1]], 1]]^2 + coordinates[[position[[a, 1, 1]], 2]]^2,
      {a, 1, Dimensions[position][[1]]}],
    {b, 1, Dimensions[stages][[1]]}
  Dimensions[points[[1]][[1]]][[1]]
  Dimensions[points[[2]][[1]]][[1]]
  Dimensions[points[[3]][[1]]][[1]]
  Dimensions[points[[4]][[1]]][[1]]
  Dimensions[points[[5]][[1]]][[1]]

113

297

69

121

191

radii = Sort[Flatten[radii], OrderedQ[{{#1}, {#2}}]&];
N[Round[radii[[145]] * 100] / 100]
N[Round[radii[[146]] * 100] / 100]

0.49

0.5

(* 791 cell cycle genes, 113 in M/G1, 297 in G1, 69 in S, 121 in S/G2, 191 in G2/M. *)

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

```

clb2 = coordinates[Position[genenames, "YPR119W"][[1, 1]]];
cln3 = coordinates[Position[genenames, "YAL040C"][[1, 1]]];
p = Show[{
  Graphics[{RGBColor[1, 0.5, 0], PointSize[0.02], points[[5]]}],
  Graphics[{RGBColor[1, 1, 0], PointSize[0.02], points[[1]]}],
  Graphics[{RGBColor[1, 0, 0], PointSize[0.02], points[[4]]}],
  Graphics[{RGBColor[0, 0.5, 0], PointSize[0.02], points[[2]]}],
  Graphics[{RGBColor[0, 0, 1], PointSize[0.02], points[[3]]}],
  Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.6, -0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.875, -0.775}]}],
  Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-0.875, -0.75}]}],
  Graphics[{RGBColor[0, 0, 1], Text["S", {-0.8, 0.8}]}],
  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[1, 0.5, 0], PointSize[0.02], Point[clb2]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[{0.55, -0.95}, clb2,
    HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75}]}],
  Graphics[{RGBColor[0, 0, 0], Text["CLB2", {0.75, -0.95}]}],
  Graphics[{RGBColor[1, 0.5, 0], PointSize[0.02], Point[cln3]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[{0.8, 0.7}, cln3,
    HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75}]}],
  Graphics[{RGBColor[0, 0, 0], Text["CLN3", {0.85, 0.8}]}],
  Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 0.5]}],
  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];

```

(\* Create Parameter Graphs of 104 Cell Cycle Genes According to Projections on Eigengenes \*)

```

stream = StringJoin[name, ":Desktop Folder:SPIE Data:Traditional_Classify_Cycle.txt"];
list = ReadList[stream, Word, RecordLists -> True, NullWords -> True];
list = Drop[list, 2];
stages = {"M/G1", "G1", "S", "S/G2", "G2/M"};
points = {points1, points2, points3, points4, points5};
radii = {radii1, radii2, radii3, radii4, radii5};
Do[{
  position = Position[list, stages[[b]]],
  table = Table[list[[position[[a, 1]], 1]], {a, 1, Dimensions[position][[1]]},
  position = Table[Position[genenames, table[[a]], {a, 1, Dimensions[table][[1]]},
  table = Flatten[Position[position, {}]],
  Do[
  position = Drop[position, {table[[a]], table[[a]]}, {a, Dimensions[table][[1]], 1, -1}],
  points[[b]] = Table[Point[coordinates[[position[[a, 1, 1]]]], {a, 1, Dimensions[position][[1]]},
  radii[[b]] = Table[
  Sqrt[coordinates[[position[[a, 1, 1]], 1]]^2 + coordinates[[position[[a, 1, 1]], 2]]^2,
  {a, 1, Dimensions[position][[1]]},
  {b, 1, Dimensions[stages][[1]]}

```

```
Dimensions[points[[1]]][[1]]
Dimensions[points[[2]]][[1]]
Dimensions[points[[3]]][[1]]
Dimensions[points[[4]]][[1]]
Dimensions[points[[5]]][[1]]
```

19

52

8

9

15

```
radii = Sort[Flatten[radii], OrderedQ[{{#1}, {#2}}]&];
N[Round[radii[[11]] * 100] / 100]
N[Round[radii[[12]] * 100] / 100]
```

0.47

0.51

(\* 103 cell cycle genes, 19 in M/G1, 52 in G1, 8 in S, 9 in S/G2, 15 in G2/M. \*)

(\* 92 with more than 25% of normalized expression in the cell cycle subspace. \*)

```
cdc8 = coordinates[[Position[genenames, "YJR057W"]][[1, 1]]];
clb2 = coordinates[[Position[genenames, "YPR119W"]][[1, 1]]];
cln3 = coordinates[[Position[genenames, "YAL040C"]][[1, 1]]];
p = Show[{
  Graphics[{RGBColor[1, 0.5, 0], PointSize[0.02], points[[5]]}],
  Graphics[{RGBColor[1, 1, 0], PointSize[0.02], points[[1]]}],
  Graphics[{RGBColor[1, 0, 0], PointSize[0.02], points[[4]]}],
  Graphics[{RGBColor[0, 0.5, 0], PointSize[0.02], points[[2]]}],
  Graphics[{RGBColor[0, 0, 1], PointSize[0.02], points[[3]]}],
  Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.6, -0.95}]}],
  Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.875, -0.775}]}],
  Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-0.875, -0.75}]}],
  Graphics[{RGBColor[0, 0, 1], Text["S", {-0.8, 0.8}]}],
  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, 0.5, 0], PointSize[0.02], Point[cdc8]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[{0.685, 0.875}, cdc8,
    HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75}]}],
  Graphics[{RGBColor[0, 0, 0], Text["CDC8", {0.875, 0.875}]}],
  Graphics[{RGBColor[1, 0.5, 0], PointSize[0.02], Point[clb2]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[{0.55, -0.95}, clb2,
    HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75}]}],
  Graphics[{RGBColor[0, 0, 0], Text["CLB2", {0.75, -0.95}]}],
  Graphics[{RGBColor[0, 0.5, 0], PointSize[0.02], Point[cln3]}],
  Graphics[{RGBColor[0, 0, 0], Arrow[{0.825, 0.675}, cln3,
    HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75}]}],
  Graphics[{RGBColor[0, 0, 0], Text["CLN3", {0.875, 0.75}]}],
  Graphics[{RGBColor[0, 0, 0], Dashing[{0.03, 0.02}], Circle[{0, 0}, 0.5]}],
  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}];
p3 = Show[p,
  AspectRatio -> 1.,
  PlotRange -> All,
  DisplayFunction -> Identity];

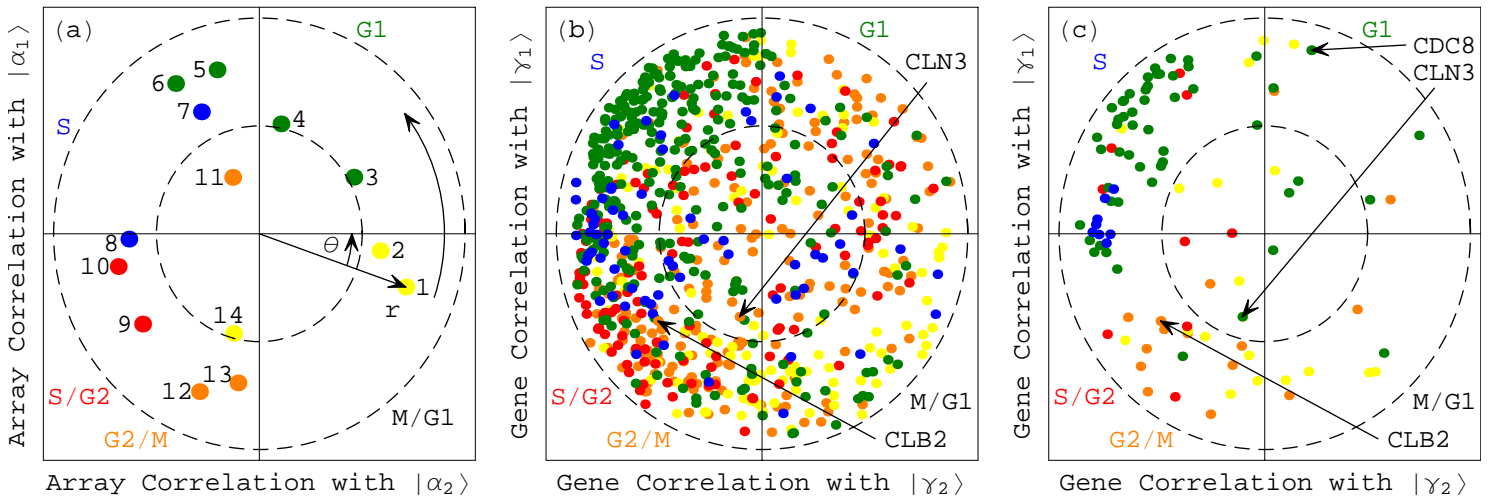
```

(\* 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]
-0.106217

```

(\* 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 = 3131;
positive1 = 3132;
sortmatrix[[negative1, 1]]
sortmatrix[[positive1, 1]]
-0.000446564
0.0000113487

```

```

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 = 1331;
positive2 = 1332;
sortmatrix[[negative2, 1]]
sortmatrix[[positive2, 1]]

-0.000803638

0.000157627

sortmatrix = AppendColumns[
  TakeRows[sortmatrix, {positive2, genes}],
  TakeRows[sortmatrix, {1, negative2}]
];
positive3 = 4687;
negative3 = 4688;
sortmatrix[[positive3, 1]]
sortmatrix[[negative3, 1]]

1.60621

-0.39346

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 = 267;  
beginph1 = 268;  
phases[[endph5]] - ph1  
phases[[beginph1]] - ph1
```

{-0.000277555}

{0.000750785}

```
endph1 = 1800;  
beginph2 = 1801;  
phases[[endph1]] - ph2  
phases[[beginph2]] - ph2
```

{-0.000142146}

{0.00018248}

```
endph2 = 3152;  
beginph3 = 3153;  
phases[[endph2]] - ph3  
phases[[beginph3]] - ph3
```

{-0.000416694}

{0.00009141}

```
endph3 = 3842;  
beginph4 = 3843;  
phases[[endph3]] - ph4  
phases[[beginph4]] - ph4
```

{-0.00119745}

{0.000090682}

```
endph4 = 4687;  
beginph5 = 4688;  
phases[[endph4]] - ph5  
phases[[beginph5]] - ph5
```

{-3.61242 × 10<sup>-6</sup>}

{0.000323438}

(\* 6018 yeast genes, 1598 in M/G1, 1533 in G1, 1352 in S, 690 in S/G2, 845 in G2/M. \*)

(\* Create Classified Sorted Data 2D 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 2D 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];

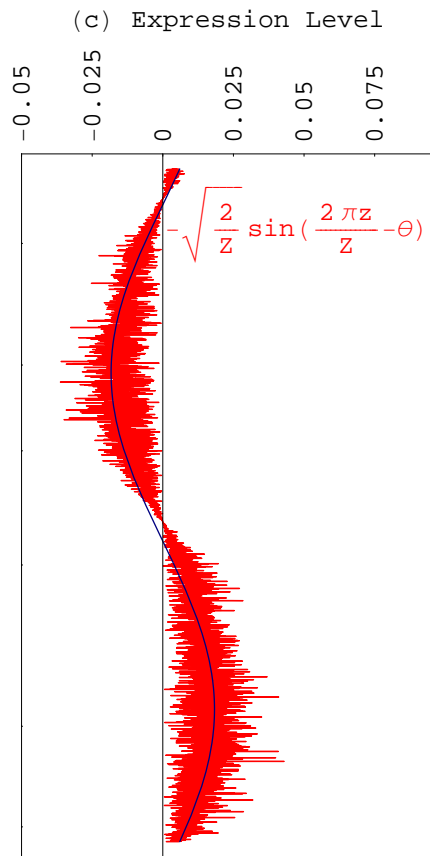
eigenarrays1 = Chop[TrigFit[eigenarrays[[1]], 1, {x, genes - 1}], 0.0025]
eigenarrays2 = Chop[TrigFit[eigenarrays[[2]], 1, {x, genes - 1}], 0.0025]

0.0039938 Cos[ $\frac{2 \pi x}{6017}$ ] - 0.0165232 Sin[ $\frac{2 \pi x}{6017}$ ]

-0.0164555 Cos[ $\frac{2 \pi x}{6017}$ ] - 0.00411512 Sin[ $\frac{2 \pi x}{6017}$ ]

zerophase = N[ArcTan[arraycorrelations[[1, 1]] / arraycorrelations[[2, 1]]] / Pi];
eigenarrays1 = -0.0183 * Sin[2 * Pi * (x + Round[zerophase * 6017 / 2]) / 6017];
eigenarrays2 = -0.0183 * Cos[2 * Pi * (x + Round[zerophase * 6017 / 2]) / 6017];

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["- $\sqrt{\frac{2}{z}} \sin(\frac{2 \pi z}{z} - \theta)$ ", {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}, {135, -genes - 135 + 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 + 1020}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
  Text[a, {b, c + 477}, {0, 0}, {0, 1}];
p1 = Show[g,
  AspectRatio -> GoldenRatio * 1.15,
  PlotRange -> All];
```



```

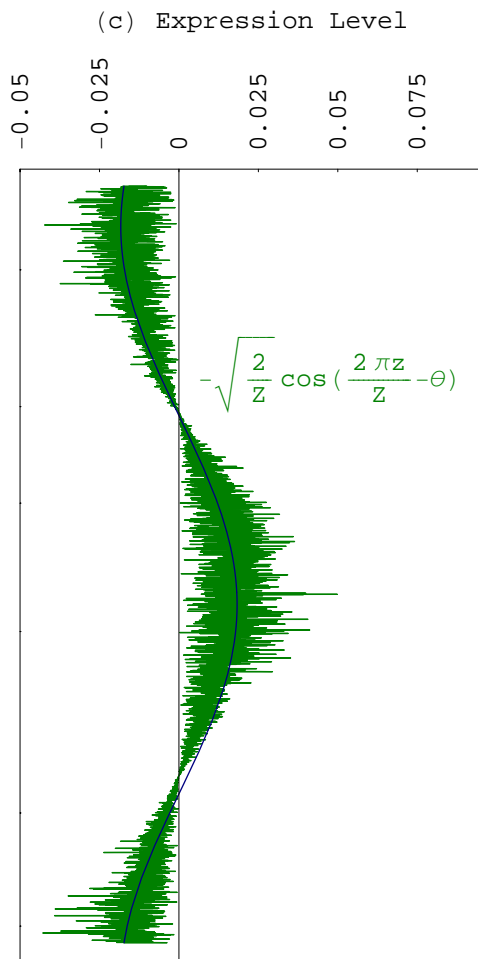
graph = ParametricPlot[{eigenarrays2, -x},
  {x, 0, genes - 1},
  PlotStyle -> {RGBColor[0, 0, 0.5]},
  DisplayFunction -> Identity];
labelx = "(c) Expression Level";
labely = ColumnForm[{" ", " ", " ", " ", " ", " "}, Center];
framex = {
  {-0.05, "-0.05 "}, {-0.025, "-0.025"}, {0, "0 "},
  {0.025, "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[" $-\sqrt{\frac{2}{z}} \cos(\frac{2\pi z}{z} - \theta)$ ", {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}, {135, -genes + 1 - 135}},
  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 + 1020}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
  Text[a_, {b_, c_}, {0., -1.}] ->
  Text[a, {b, c + 477}, {0, 0}, {0, 1}];
p2 = Show[g,
  AspectRatio -> GoldenRatio * 1.15,
  PlotRange -> All];

```



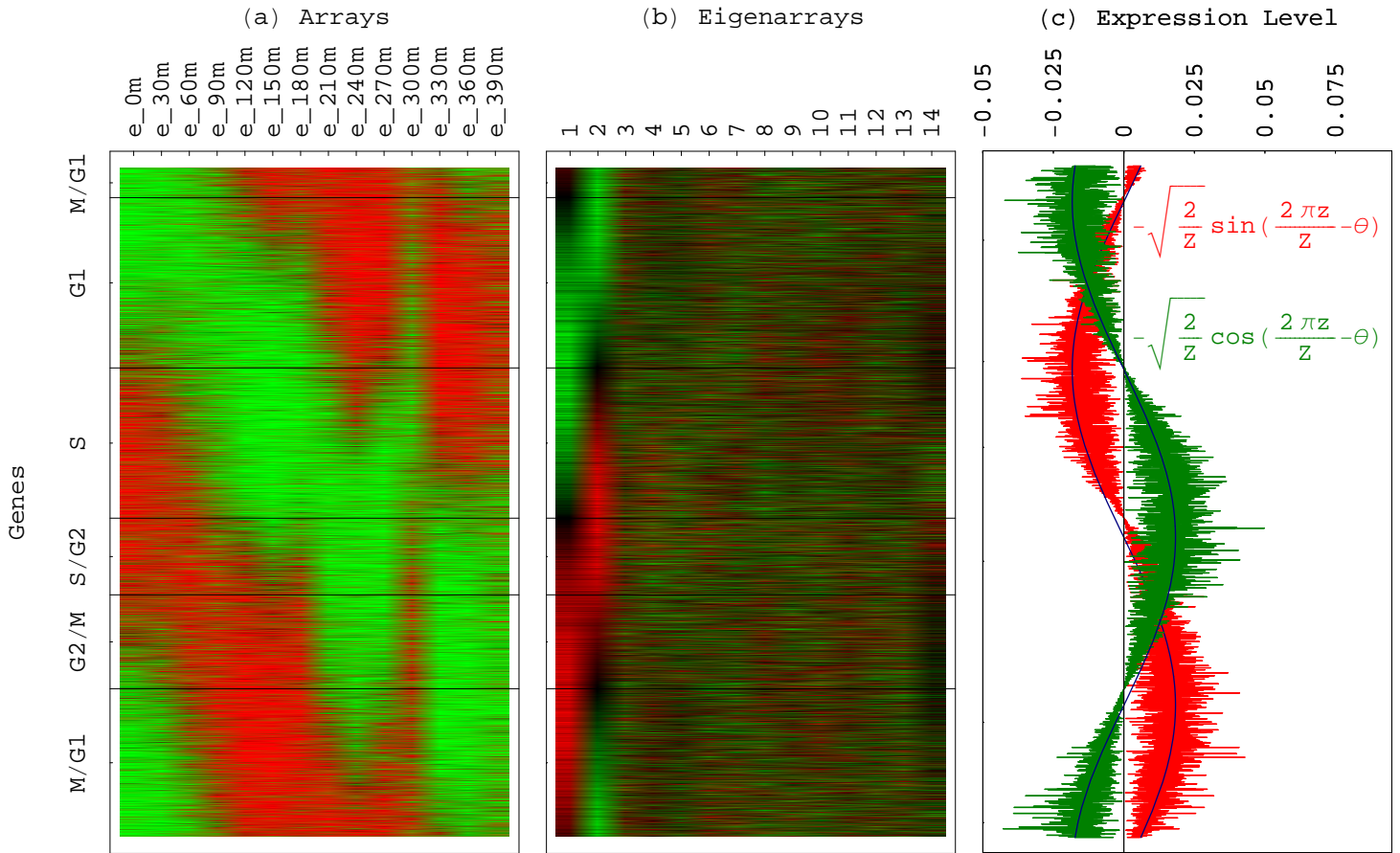


(\* Display Classified Selected Sorted Eigenarrays \*)

```
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:SPIE Data:Sort_Elutriation.txt"],
  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 2D 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 2D 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];

```

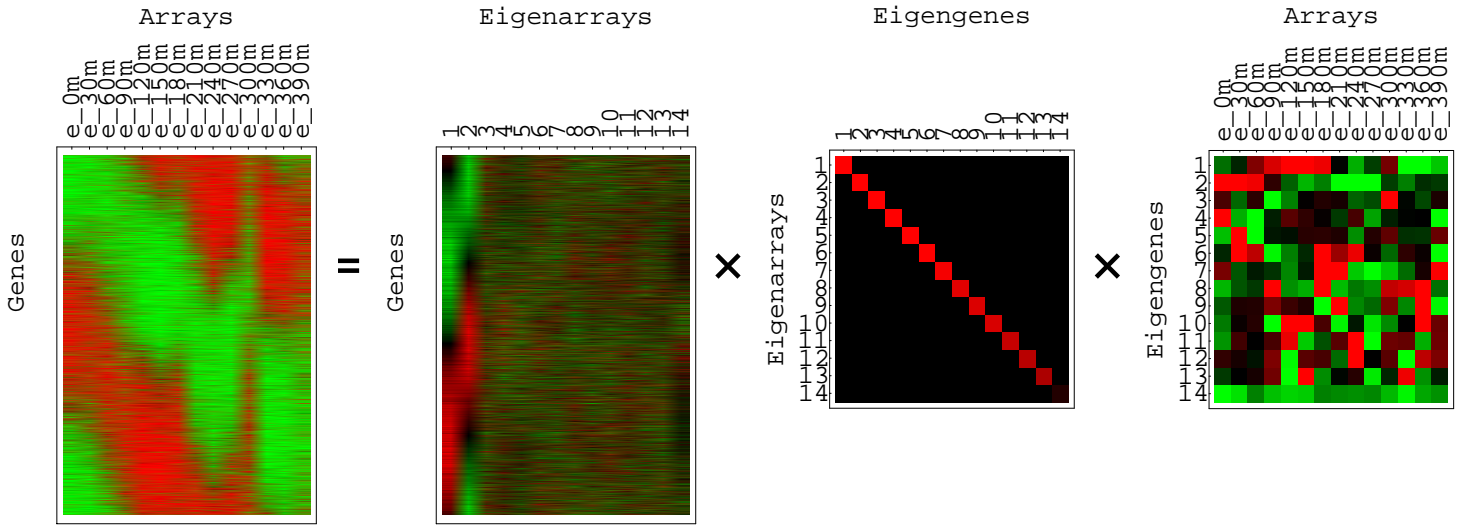
```
(* Create Eigengenes 2D Red & Green Raster Display *)
```

```
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 = 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 *)
```

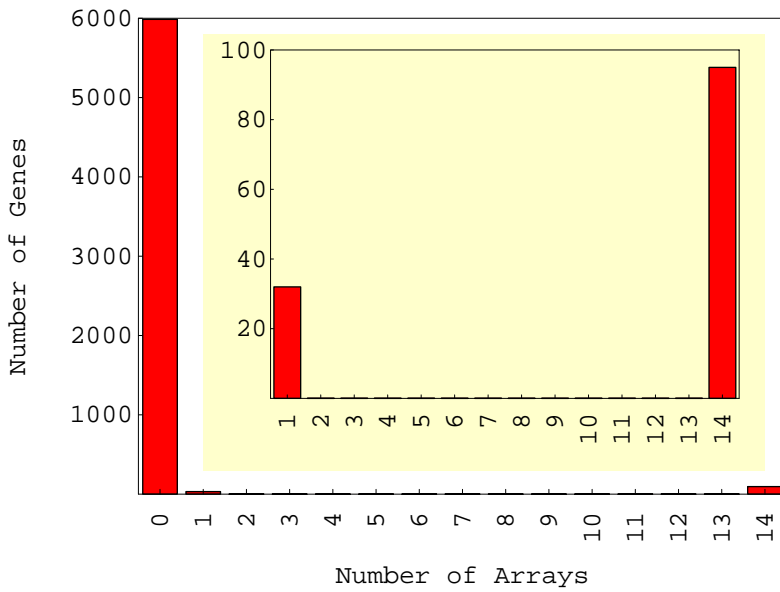
```
equal = Show[Graphics[
  Text[StyleForm["=", FontSize -> 20, FontWeight -> Bold], {0, 0}]
], DisplayFunction -> Identity];
times = Show[Graphics[
  Text[StyleForm["x", FontSize -> 20, FontWeight -> Bold], {0, 0}]
], DisplayFunction -> Identity];
```

```
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 \*)

(\* Locating Gene Position of Null Data \*)



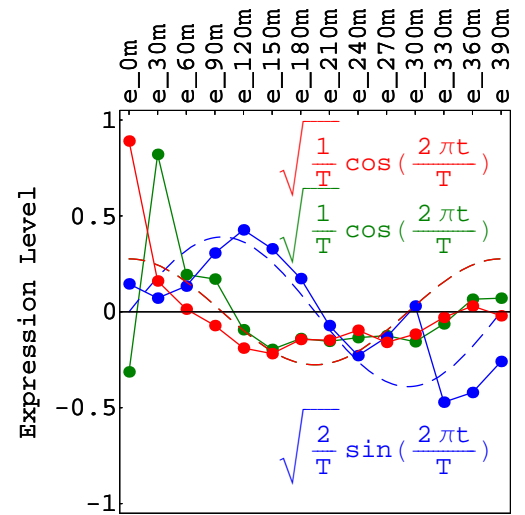
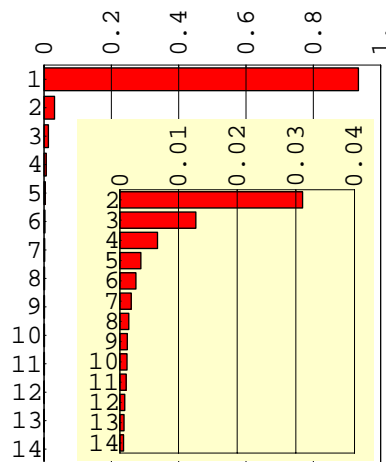
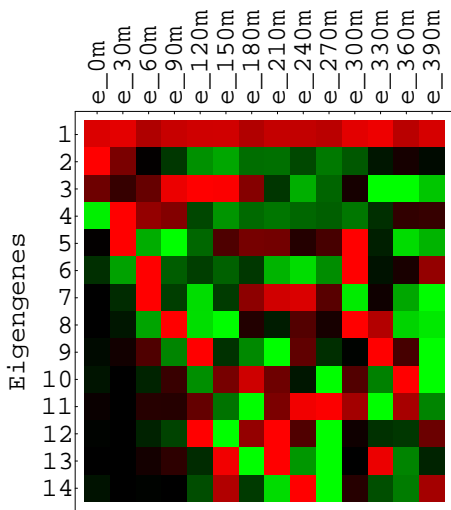
(\* Choosing Subset of Eigengenes for Estimation \*)

(a) Arrays

(b) Eigenexpression Probability

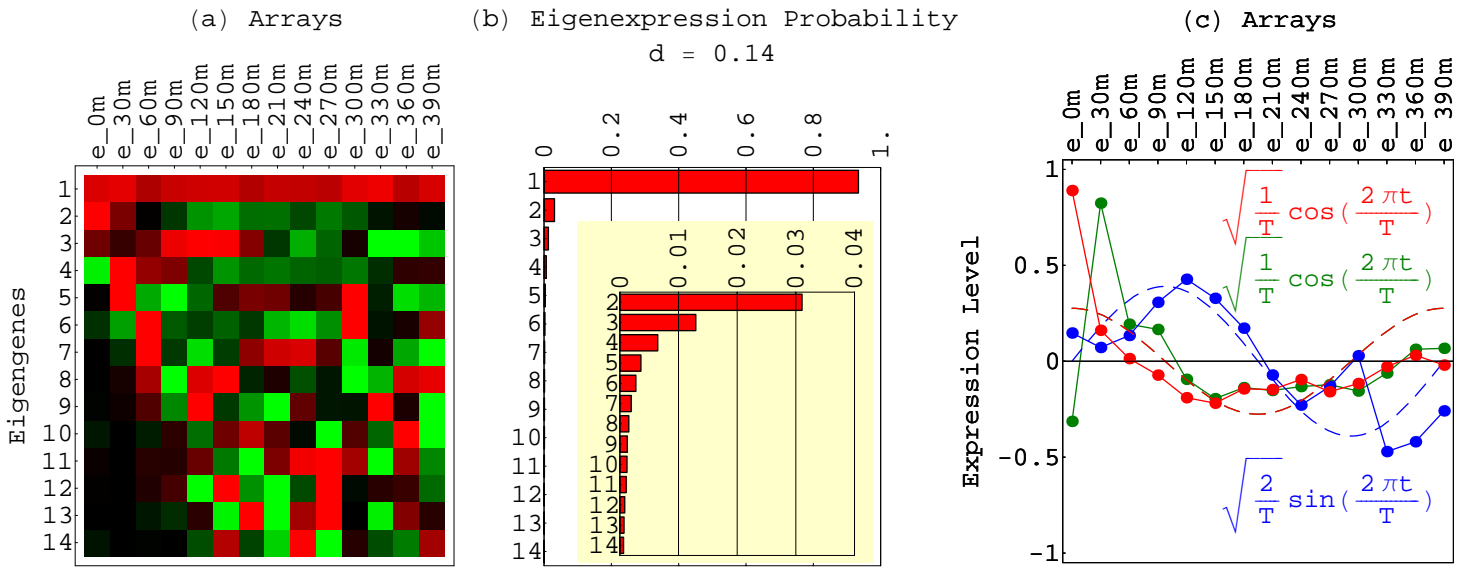
(c) Arrays

$d = 0.14$

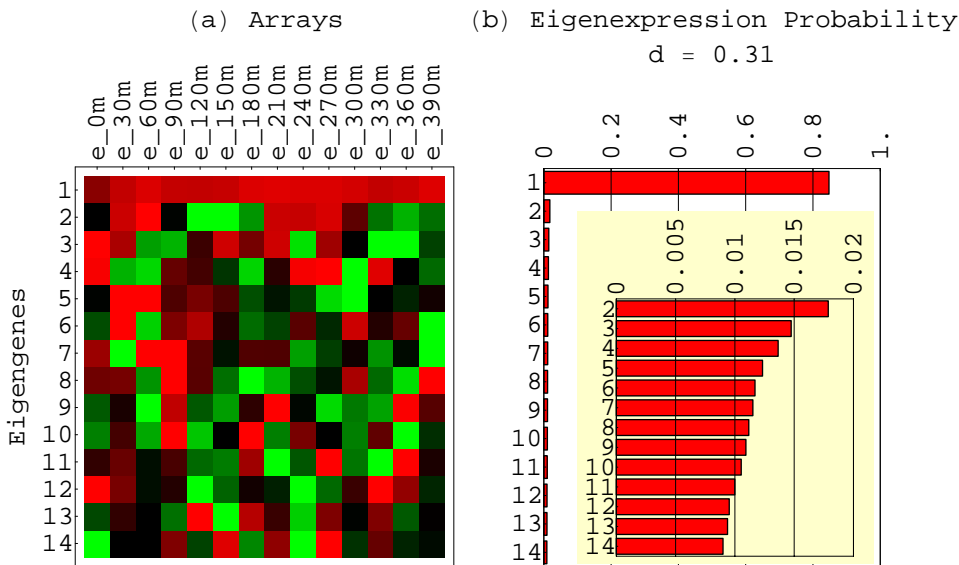




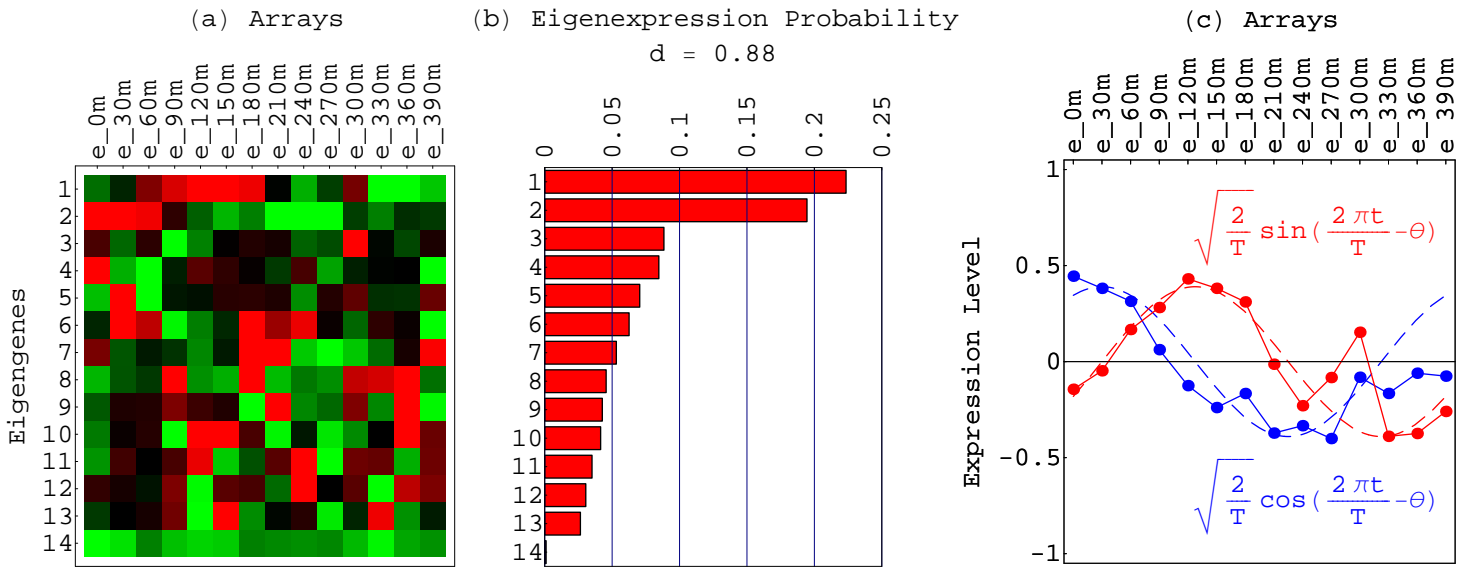
(\* Centering by Removing the Additive Constant, i.e., the Steady State \*)



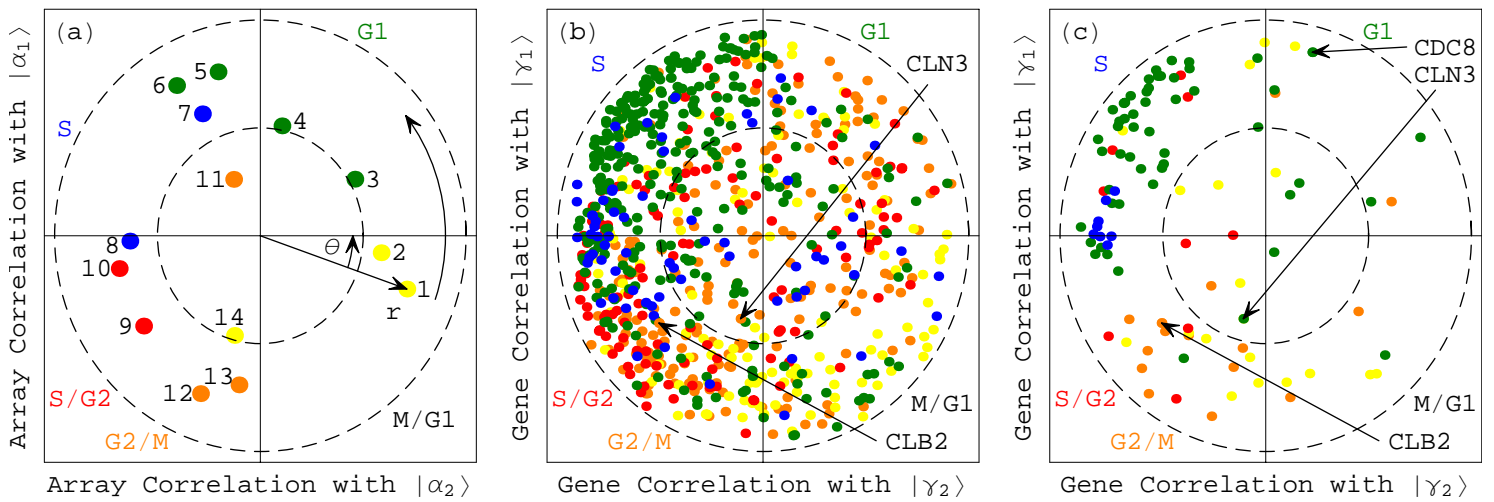
(\* Normalizing the Variances by Removing the Multiplicative Constant \*)



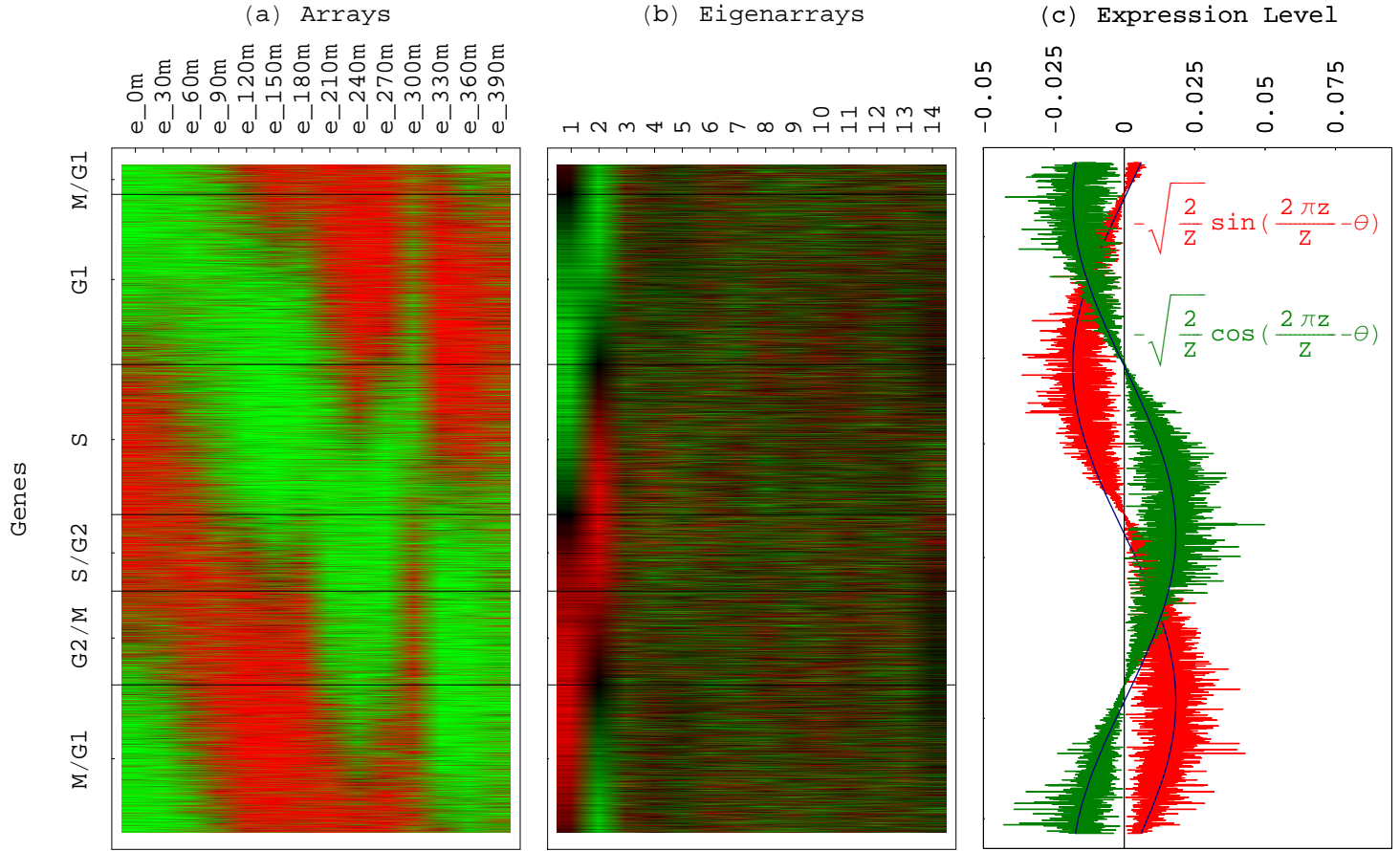
(\* Normalized Data \*)



(\* Sorting the Normalized Data \*)



(\* Classified Sorted Normalized Data \*)



(\* Summarized SVD \*)

