```
<< LinearAlgebra`MatrixManipulation`;
<< NumericalMath`TrigFit`;
<< Graphics`Graphics`;
<< Graphics`Arrow`;
Off[General::"spell"];
Off[General::"spell1"];

(* Read Transcript Size Data *)

stream = "Desktop/mRNA_Lengths.txt";
matrix = Import[stream, "Table"];
{genes, arrays} = Dimensions[matrix] - {1, 1}
Clear[stream];

{6776, 30}

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

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

```
(* Calculate SVD *)

{eigenarrays, eigenabundances, eigengenes} = SingularValues[matrix];
eigengenes = -eigengenes;
eigenarrays = -eigenarrays;
eigengenes[[12]] = -eigengenes[[12]];
eigenarrays[[12]] = -eigenarrays[[12]];
eigengenes[[13]] = -eigengenes[[13]];
eigenarrays[[13]] = -eigenarrays[[13]];
eigengenes[[15]] = -eigengenes[[15]];
eigenarrays[[15]] = -eigenarrays[[15]];
eigengenes[[16]] = -eigengenes[[16]];
eigenarrays[[16]] = -eigenarrays[[16]];
eigengenes[[18]] = -eigengenes[[18]];
eigenarrays[[18]] = -eigenarrays[[18]];
eigengenes[[21]] = -eigengenes[[21]];
eigenarrays[[21]] = -eigenarrays[[21]];
eigengenes[[24]] = -eigengenes[[24]];
eigenarrays[[24]] = -eigenarrays[[24]];
eigengenes[[25]] = -eigengenes[[25]];
eigenarrays[[25]] = -eigenarrays[[25]];
eigengenes[[27]] = -eigengenes[[27]];
eigenarrays[[27]] = -eigenarrays[[27]];
eigengenes[[28]] = -eigengenes[[28]];
eigenarrays[[28]] = -eigenarrays[[28]];
eigenarrays = Transpose[eigenarrays];
fractions = eigenabundances^2 / Sum[eigenabundances[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, arrays}] / Log[arrays]];
entropy = N[Round[100 * entropy] / 100]

0.68
```

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

contrast = 3.5;
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[[1, 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 - 3, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] →
    Text[labelx, {b, c + 2.75}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c + 2.25}, {0, 0}, {0, 1}];
g3 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction → Identity];

(* Fit Eigengenes With an Asymmetric Parabola *)

k1 = 0.36;
k2 = 0.09;

Clear[f];
f[x_] := If[(x - 11) < 0, 0.5 * k1 * (x - 11) ^2, 0.5 * k2 * (x - 11) ^2];
inflection = Plot[arrays - f[x + 0.5], {x, 0.5, arrays - 0.5},
   PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.013]},
   DisplayFunction → Identity];
g1 = Show[{g3, inflection},
   AspectRatio → 1.05,
   DisplayFunction → Identity];
```

```
(* Fit Fractions *)

Clear[constant, λ];
f = FindFit[Table[fractions[[n]], {n, 2, 10}], constant * λ^x, {constant, λ}, x]

{constant → 0.138516, λ → 0.806993}

λ = 0.8;
constant = constant /. f;
correlation = Dot[Table[fractions[[n]], {n, 2, 10}], Table[constant * λ^n, {n, 2, 10}]] /
   Sqrt[Dot[Table[fractions[[n]], {n, 2, 10}], Table[fractions[[n]], {n, 2, 10}]]] /
  Sqrt[Dot[Table[constant * λ^n, {n, 2, 10}], Table[constant * λ^n, {n, 2, 10}]]]

0.998854

(* Create Fractions Bar Chart Displays With Fitting Graph *)

fractions[[2]]

0.115404

limit = 0.12;

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 - 15}, {a, 0, 15 - 2}];
table = Table[fractions[[arrays - a]], {a, 15, arrays - 2}];

g = BarChart[table,
   BarOrientation -> Horizontal,
   PlotRange -> {{0, limit * 1.0001}, {0.5, 15 - 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 + 1.25}, {0, 0}, {0, 1}];

f1 = NSolve[Log[x / constant] / Log[1 / λ] + 15 == 1, x][[1]]
x1 = x /. f1;
f2 = NSolve[Log[x / constant] / Log[1 / λ] + 15 == 14, x][[1]]
x2 = x /. f2;

{x → 0.00609198}

{x → 0.110812}
```

```
graph = Plot[Log[x / constant] / Log[1 / λ] + 15, {x, x1, x2},
   PlotRange → {0, 14},
   PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.016]},
   DisplayFunction → Identity];
g3 = Show[{g, graph},
   AspectRatio -> 1.25,
   PlotRange → All,
   DisplayFunction → Identity];
```

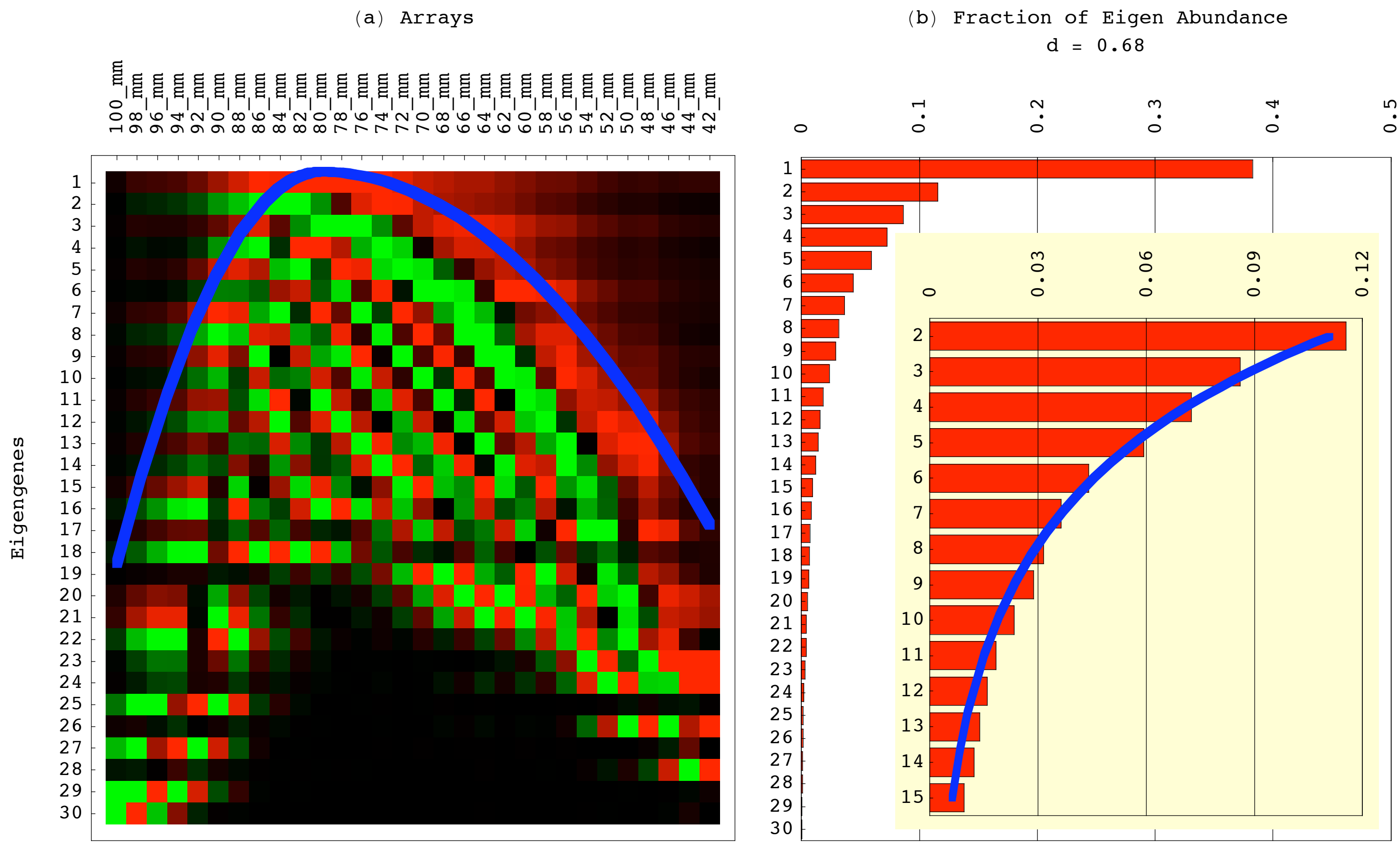
```
fractions[[1]]
```

```
0.382673
```

```
limit = 0.5;
```

```
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}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 1}];
labelx = ColumnForm[
   {"(b) Fraction of Eigen Abundance", StringJoin["d = ", ToString[entropy]], " "},
   Center];
labely = ColumnForm[{" ", " "}, 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, labely, labelx, None},
   GridLines -> {gridx, None},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] →
    Text[labely, {b - 0.1, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] →
    Text[labelx, {b, c + 2.75}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c + 2.25}, {0, 0}, {0, 1}];
g2 = Show[{g,
    Graphics[{RGBColor[1, 1, 0.8], Rectangle[{0.08, 1}, {0.49, 27.2}]}],
    Graphics[{Rectangle[{0.08, 1}, {0.49, 27.2}, g3]}]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction → Identity];
```

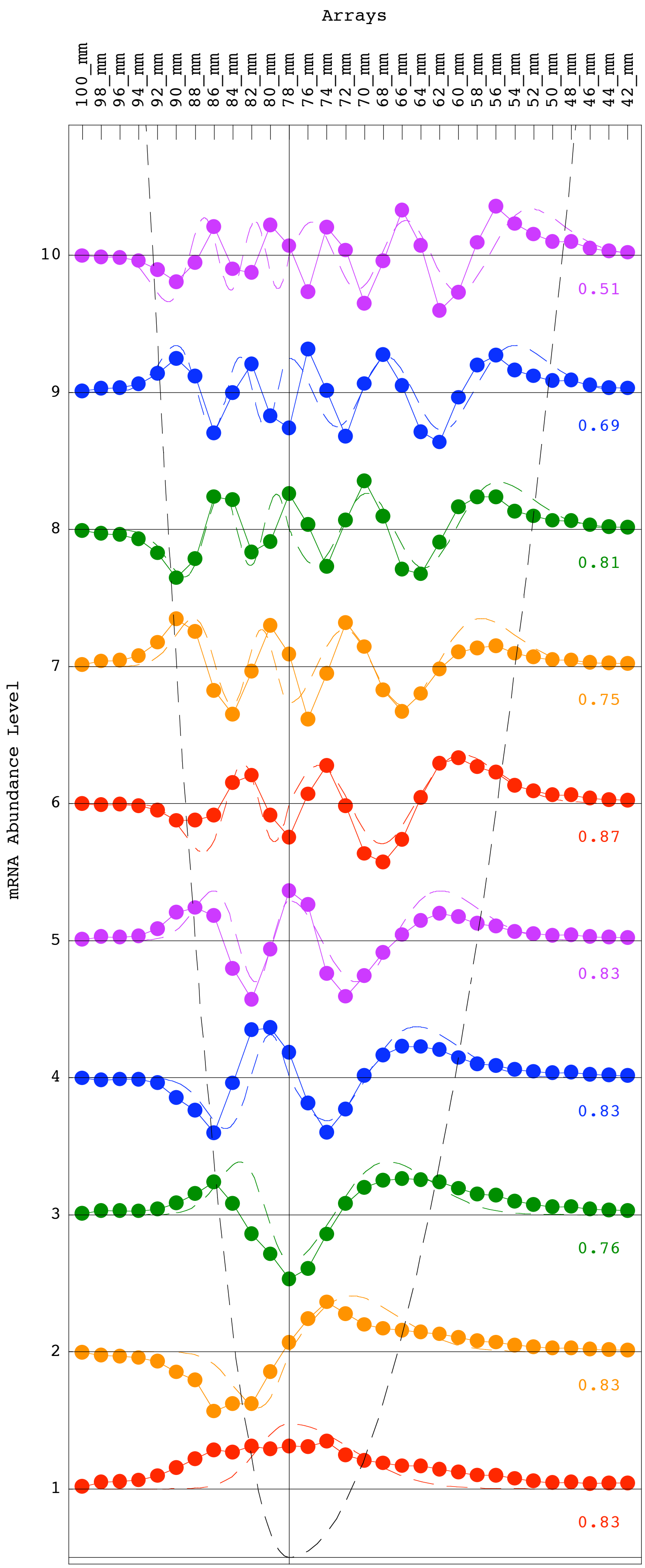
```
(* Display Eigengenes and Fractions With Fitting Graphs *)

Show[GraphicsArray[{g1, g2}],
  GraphicsSpacing -> -0.2];
```




```
(* Fit Eigengenes with Asymmetric Hermite Functions *)

k1 = 0.36;
k2 = 0.09;

Clear[g, h];
h[x_, n_, gamma_] := Exp[-gamma * x ^ 2 / 2] * HermiteH[n, Sqrt[gamma] * x] *
   Sqrt[Sqrt[gamma / Pi] / Factorial[n] / (2 ^ n)];
g[x_, n_] := If[x - 11 < 0, h[x - 11, n - 1, k1] * (1 / k1) ^ 0.25, h[x - 11, n - 1, k2] * (1 / k2) ^ 0.25];
normalization = Table[
   Sqrt[Sum[g[x, n] ^ 2., {x, 0, arrays - 1}]],
   {n, 1, 10}];
correlation = Table[
   Sum[g[x, n] * eigengenes[[n, x + 1]] / normalization[[n]], {x, 0, arrays - 1}],
   {n, 1, 10}];
meancorrelation = Round[100. * Sqrt[Sum[0.1 * correlation[[n]] ^ 2, {n, 1, 10}]]] / 100.
correlation = Round[100. * correlation] / 100.;

0.78
```

```
(* Fit Differential Equation with an Asymmetric Parabola *)

Clear[f];
f[x_] := If[(x - 11) < 0, 0.5 * k1 * (x - 11) ^ 2, 0.5 * k2 * (x - 11) ^ 2];
```

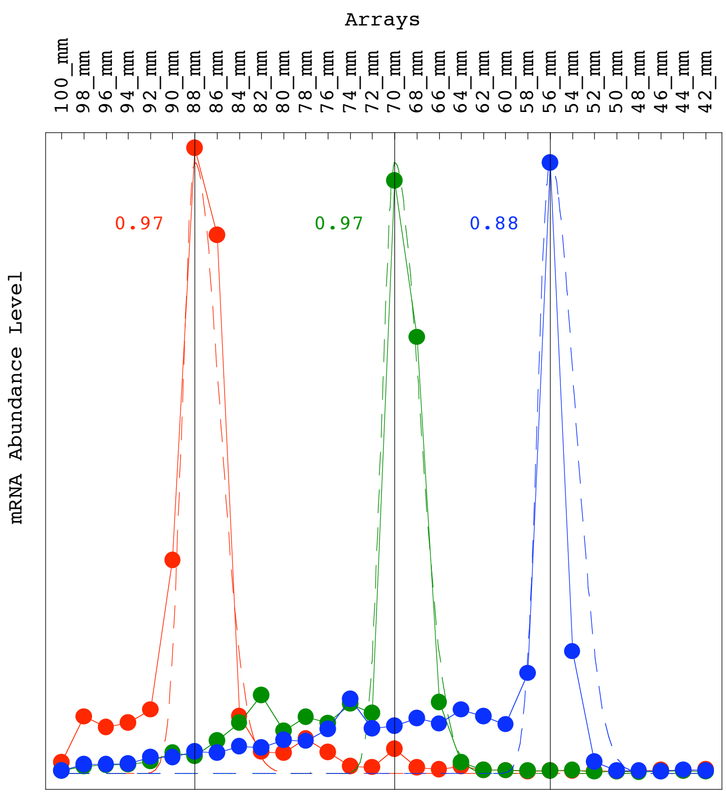
```
(* Create Selected Eigengenes Graph Display With Fitting Graphs *)

labelx = ColumnForm[{" ", "Arrays"}, Center];
labely = ColumnForm[{" ", "          mRNA Abundance Level"}, Center];
framex = Table[{a - 1, arraynames[[1, a]]}, {a, 1, arrays}];
framey = Table[{n - 0.5, n}, {n, 1, 10}];
color = {
   RGBColor[0.75, 0, 1],
   RGBColor[1, 0, 0],
   RGBColor[1, 0.5, 0],
   RGBColor[0, 0.5, 0],
   RGBColor[0, 0, 1]};

points = Table[0, {n, 1, 10}];
lines = Table[0, {n, 1, 10}];
Do[
 {coordinates = Table[{a - 1, eigengenes[[n, a]] + n - 0.5}, {a, 1, arrays}],
  points[[n]] = Table[Point[coordinates[[a]]], {a, 1, arrays}],
  lines[[n]] = Line[coordinates]},
 {n, 1, 10}]
points = Table[Graphics[{color[[Mod[n, 5] + 1]], PointSize[0.022], points[[n]]}], {n, 1, 10}];
lines = Table[Graphics[{color[[Mod[n, 5] + 1]], lines[[n]]}], {n, 1, 10}];
graphs = Table[
   Plot[g[x, n] / normalization[[n]] + n - 0.5,
    {x, 0, arrays - 1},
    PlotStyle -> {color[[Mod[n, 5] + 1]], Dashing[{0.03, 0.02}]},
    DisplayFunction → Identity],
   {n, 1, 10}];
texts = Table[
   Graphics[{color[[Mod[n, 5] + 1]], Text[correlation[[n]], {27.5, n - 0.75}]}],
   {n, 1, 10}];
inflection = Plot[f[x], {x, 11 - Sqrt[10.45 * 2 / k1], 11 + Sqrt[10.45 * 2 / k2]},
   PlotStyle -> {RGBColor[0, 0, 0], Dashing[{0.03, 0.02}]},
   DisplayFunction → Identity];

g = Show[{points, lines, graphs, texts, inflection},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    GridLines -> {{{11, RGBColor[0, 0, 0]}},
     AppendColumns[{{0, RGBColor[0, 0, 0]}}, Table[{a - 0.5, RGBColor[0, 0, 0]}, {a, 1, 10}]]
    },
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.05, 10.45},
    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 + 0.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
        Text[a_, {b_, c_}, {0., -1.}] ->
            Text[a, {b, c + 0.15}, {0, 0}, {0, 1}];
g3 = Show[g,
        AspectRatio → 2.4,
        PlotRange -> All];
```


Arrays
100_mm
98_mm
96_mm
94_mm
92_mm
90_mm
88_mm
86_mm
84_mm
82_mm
80_mm
78_mm
76_mm
74_mm
72_mm
70_mm
68_mm
66_mm
64_mm
62_mm
60_mm
58_mm
56_mm
54_mm
52_mm
50_mm
48_mm
46_mm
44_mm
42_mm
mRNA Abundance Level
10
9
8
7
6
5
4
3
2
1
0.51
0.69
0.81
0.75
0.87
0.83
0.83
0.76
0.83
0.83

```
(* Create Synthetic Data *)

(* Fit Asymmetric Gaussians to Genes and Arrays' Distributions *)

k1 = 0.36;
k2 = 0.09;

Clear[a, b, α, β, λ]
λ = 0.8;
f1 = NSolve[Sqrt[(α - k1 / 2) / (α + k1 / 2)] == λ, α];
g1 = NSolve[2 * Sqrt[(α /. f1) ^2 - β^2] == k1, β][[1]];
a = ((α /. f1) + (β /. g1))[[1]]
b = ((α /. f1) - (β /. g1))[[1]] * a / (a - ((α /. f1) - (β /. g1))[[1]])

1.62

0.02025

Clear[a, b, α, β, λ]
λ = 0.8;
f1 = NSolve[Sqrt[(α - k2 / 2) / (α + k2 / 2)] == λ, α];
g1 = NSolve[2 * Sqrt[(α /. f1) ^2 - β^2] == k2, β][[1]];
a = ((α /. f1) + (β /. g1))[[1]]
b = ((α /. f1) - (β /. g1))[[1]] * a / (a - ((α /. f1) - (β /. g1))[[1]])

0.405

0.0050625

Clear[f, x];
f[x_, a_, b_] =
  If[x < 0,
   N[Exp[-4 * a * b / (a + b) * x^2]],
   N[Exp[-a * b / (a + b) * x^2]]];
syndistribution = Table[f[x, a, b], {x, -10, 19}];
Clear[α, β, λ]

Clear[f, x, p];
f[x_, p_, a_, b_] =
  If[x < p,
    N[Exp[-4 * a * (x - p) ^2]],
    N[Exp[-a * (x - p) ^2]]] *
   If[p < 0,
    N[Exp[-4 * b * p^2]],
    N[Exp[-b * p^2]]];
synmatrix = Table[f[x, p, a, b], {p, -10, 19}, {x, -10, 19}];
{syngenes, arrays} = Dimensions[synmatrix]
Clear[α, β, λ]

{30, 30}
```

```
(* Fit the Profiles of Individual Genes With an Asymmetric Gaussian *)

ygr020c = matrix[[Position[genenames, "YGR020C"][[1, 1]]]];
ynl220w = matrix[[Position[genenames, "YNL220W"][[1, 1]]]];
ycl030c = matrix[[Position[genenames, "YCL030C"][[1, 1]]]];
ygr020c = ygr020c / Sort[ycl030c, OrderedQ[{#2, #1}] &][[1]];
ynl220w = ynl220w / Sort[ycl030c, OrderedQ[{#2, #1}] &][[1]];
ycl030c = ycl030c / Sort[ycl030c, OrderedQ[{#2, #1}] &][[1]];

correlation1 = Dot[ygr020c, Table[f[x, 7, a, b], {x, 1, 30}]] /
    Sqrt[Dot[ygr020c, ygr020c]] /
   Sqrt[Dot[Table[f[x, 7, a, b], {x, 1, 30}], Table[f[x, 7, a, b], {x, 1, 30}]]];
correlation2 = Dot[ynl220w, Table[f[x, 16, a, b], {x, 1, 30}]] /
    Sqrt[Dot[ynl220w, ynl220w]] /
   Sqrt[Dot[Table[f[x, 16, a, b], {x, 1, 30}], Table[f[x, 16, a, b], {x, 1, 30}]]];
correlation3 = Dot[ycl030c, Table[f[x, 23, a, b], {x, 1, 30}]] /
    Sqrt[Dot[ycl030c, ycl030c]] /
   Sqrt[Dot[Table[f[x, 23, a, b], {x, 1, 30}], Table[f[x, 23, a, b], {x, 1, 30}]]];
meancorrelation = Round[100 * (correlation1 + correlation2 + correlation3) / 3] / 100.
correlation1 = Round[100 * correlation1] / 100.;
correlation2 = Round[100 * correlation2] / 100.;
correlation3 = Round[100 * correlation3] / 100.;

0.94

(* Display the Profiles of Individual Genes With an Asymmetric Gaussian Fit *)

coordinates1 = Table[{n, ygr020c[[n]]}, {n, 1, arrays}];
points1 = Table[Point[coordinates1[[n]]], {n, 1, arrays}];
points1 = Graphics[{RGBColor[1, 0, 0], PointSize[0.022], points1}];
lines1 = Graphics[{RGBColor[1, 0, 0], Line[coordinates1]}];
graph1 =
  Plot[f[x, 7, a, b] / f[7, 7, a, b],
   {x, 1, 30},
   PlotStyle → {RGBColor[1, 0, 0], Dashing[{0.03, 0.02}]},
   Axes → False,
   Frame → True,
   PlotRange → All,
   DisplayFunction → Identity];

coordinates2 = Table[{n, ynl220w[[n]]}, {n, 1, arrays}];
points2 = Table[Point[coordinates2[[n]]], {n, 1, arrays}];
points2 = Graphics[{RGBColor[0, 0.5, 0], PointSize[0.022], points2}];
lines2 = Graphics[{RGBColor[0, 0.5, 0], Line[coordinates2]}];
graph2 =
  Plot[f[x, 16, a, b] / f[16, 16, a, b],
   {x, 1, 30},
   PlotStyle → {RGBColor[0, 0.5, 0], Dashing[{0.03, 0.02}]},
   Axes → False,
   Frame → True,
   PlotRange → All,
   DisplayFunction → Identity];

coordinates3 = Table[{n, ycl030c[[n]]}, {n, 1, arrays}];
points3 = Table[Point[coordinates3[[n]]], {n, 1, arrays}];
points3 = Graphics[{RGBColor[0, 0, 1], PointSize[0.022], points3}];
lines3 = Graphics[{RGBColor[0, 0, 1], Line[coordinates3]}];
graph3 =
  Plot[f[x, 23, a, b] / f[23, 23, a, b],
   {x, 1, 30},
   PlotStyle → {RGBColor[0, 0, 1], Dashing[{0.03, 0.02}]},
   Axes → False,
   Frame → True,
   PlotRange → All,
   DisplayFunction → Identity];
```

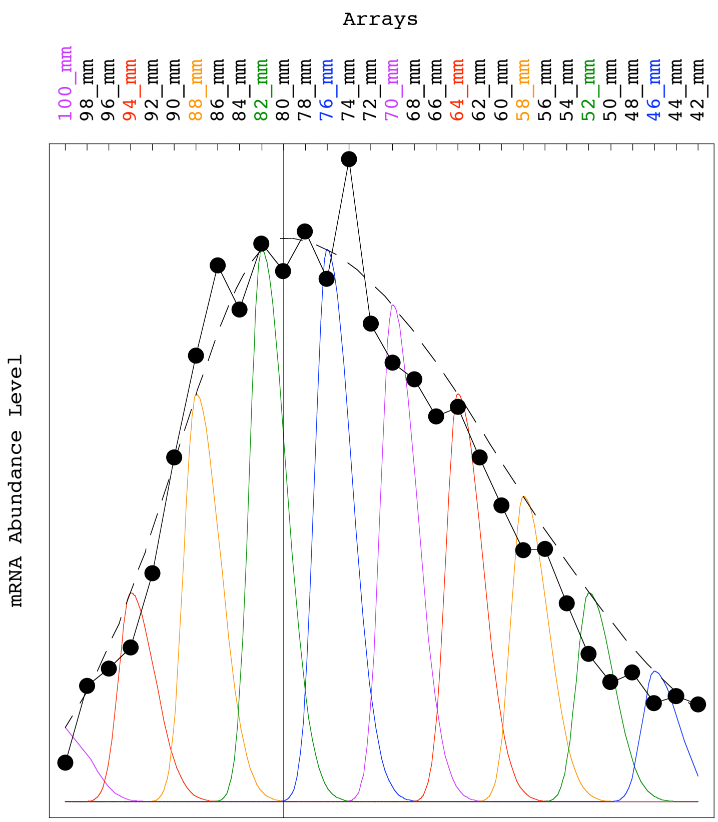
```
text1 = Graphics[{RGBColor[1, 0, 0], Text[ToString[correlation1], {4.5, 0.9}]}];
text2 = Graphics[{RGBColor[0, 0.5, 0], Text[ToString[correlation2], {13.5, 0.9}]}];
text3 = Graphics[{RGBColor[0, 0, 1], Text[ToString[correlation3], {20.5, 0.9}]}];
framex = Table[{n, arraynames[[1, n]]}, {n, 1, arrays}];
g = Show[{
    points1, lines1, graph1, text1,
    points2, lines2, graph2, text2,
    points3, lines3, graph3}, text3,
   Frame -> True,
   FrameLabel -> {None, labely, labelx, None},
   GridLines -> {{{7, RGBColor[0, 0, 0]}, {16, RGBColor[0, 0, 0]}, {23, RGBColor[0, 0, 0]}}, None},
   FrameTicks -> {None, None, framex, None},
   PlotRange → All,
   DisplayFunction → Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {m_, c_}, {1., 0.}] →
    Text[labely, {m - 1.4, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {m_, c_}, {0., -1.}] →
    Text[labelx, {m, c + 0.14}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[n_, {m_, c_}, {0., -1.}] →
    Text[n, {m, c + 0.07}, {0, 0}, {0, 1}];
g4 = Show[g,
   AspectRatio -> 1.05,
   PlotRange → All];
```



```
(* Create Synthetic Data Graph Display *)

graphs = Table[
   Plot[f[x, p, a, b],
    {x, -10, 19},
    PlotStyle -> color[[Mod[(p + 10) / 3, 5] + 1]],
    Axes → False,
    Frame → True,
    PlotRange → All,
    DisplayFunction → Identity],
   {p, -10, 19, 3}];

graph =
  Plot[f[p, p, a, b],
   {p, -10, 19},
   PlotStyle -> {RGBColor[0, 0, 0], Dashing[{0.03, 0.02}]},
   Axes → False,
   Frame → True,
   PlotRange → All,
   DisplayFunction → Identity];

(* Fit the Distribution of the Genes *)

distribution = Sum[matrix[[a]], {a, 1, genes}];
distribution = distribution / Sort[distribution, OrderedQ[{#2, #1}] &][[1]];

Clear[constant, g];
g = FindFit[Drop[distribution, {1, 10}], constant * Exp[-b * x ^ 2], {constant}, x]

{constant → 0.877522}

distribution = distribution / (constant /. g);
correlation = Dot[distribution, Table[f[p, p, a, b], {p, -10, 19}]] /
   Sqrt[Dot[distribution, distribution]] /
  Sqrt[Dot[Table[f[p, p, a, b], {p, -10, 19}], Table[f[p, p, a, b], {p, -10, 19}]]]

0.993874
```

(* Display Synthetic Data With the Distribution of the Genes *)

```
coordinates = Table[{n - 11, distribution[[n]]}, {n, 1, arrays}];
points = Table[Point[coordinates[[n]]], {n, 1, arrays}];
points = Graphics[{RGBColor[0, 0, 0], PointSize[0.022], points}];
lines = Graphics[{RGBColor[0, 0, 0], Line[coordinates]}];

framex = Table[{n - 11, arraynames[[1, n]]}, {n, 1, arrays}];
Do[framex[[n]] = {n - 11, StyleForm[arraynames[[1, n]], FontColor -> color[[Mod[(n - 1) / 3, 5] + 1]]]},
 {n, 1, arrays, 3}]
labely = "mRNA Abundance Level";
labelx = ColumnForm[{"Arrays", " ", " "}, Center];
g = Show[{graphs, graph, points, lines},
   Frame -> True,
   FrameLabel -> {None, labely, labelx, None},
   GridLines -> {{{0, RGBColor[0, 0, 0]}}, None},
   FrameTicks -> {None, None, framex, None},
   DisplayFunction → Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {m_, c_}, {1., 0.}] →
    Text[labely, {m - 1, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {m_, c_}, {0., -1.}] →
    Text[labelx, {m, c + 0.07}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[n_, {m_, c_}, {0., -1.}] →
    Text[n, {m, c + 0.08}, {0, 0}, {0, 1}];
g5 = Show[g,
   AspectRatio -> 1.05,
   PlotRange → All];
Clear[a, b];
```



```
(* Calculate SVD *)

{eigenarrays, eigenabundances, syneigengenes} = SingularValues[synmatrix];
fractions = eigenabundances^2 / Sum[eigenabundances[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, arrays}] / Log[arrays]];
entropy = N[Round[100 * entropy] / 100]

0.75

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

contrast = 3.5;
displaying = Table[
   If[contrast * syneigengenes[[i, j]] > 0,
    If[contrast * syneigengenes[[i, j]] < 1, {contrast * syneigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * syneigengenes[[i, j]] > -1, {0, -contrast * syneigengenes[[i, j]]}, {0, 1}]],
   {i, 1, arrays}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[1, 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 - 3, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] →
    Text[labelx, {b, c + 2.75}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c + 2.25}, {0, 0}, {0, 1}];
g3 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction → Identity];

(* Fit Eigengenes With an Asymmetric Parabola *)

k1 = 0.36;
k2 = 0.09;

Clear[f];
f[x_] := If[(x - 11) < 0, 0.5 * k1 * (x - 11)^2, 0.5 * k2 * (x - 11)^2];
inflection = Plot[arrays - f[x + 0.5], {x, 0.5, arrays - 0.5},
   PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.013]},
   DisplayFunction → Identity];
g1 = Show[{g3, inflection},
   AspectRatio → 1.05,
   DisplayFunction → Identity];
```

```
(* Fit Fractions *)

Clear[constant, λ];
f = FindFit[Table[fractions[[n]], {n, 2, 10}], constant * λ^x, {constant, λ}, x]

{constant → 0.193882, λ → 0.80218}

λ = 0.8;
constant = constant /. f;
correlation = Dot[Table[fractions[[n]], {n, 2, 10}], Table[constant * λ^n, {n, 2, 10}]] /
   Sqrt[Dot[Table[fractions[[n]], {n, 2, 10}], Table[fractions[[n]], {n, 2, 10}]]] /
  Sqrt[Dot[Table[constant * λ^n, {n, 2, 10}], Table[constant * λ^n, {n, 2, 10}]]]

0.999978

(* Create Fractions Bar Chart Displays With Fitting Graph *)

fractions[[1]]

0.19483

limit = 0.2;

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}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 1}];
labelx = ColumnForm[
   {"(b) Fraction of Eigen Abundance", StringJoin["d = ", ToString[entropy]], " "},
   Center];
labely = ColumnForm[{" ", " "}, 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, labely, labelx, None},
   GridLines -> {gridx, None},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] →
    Text[labely, {b - 0.04, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] →
    Text[labelx, {b, c + 2.75}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c + 2.25}, {0, 0}, {0, 1}];
```

```
f1 = NSolve[Log[x / constant] / Log[1 / λ] + 30 == 1, x][[1]]
x1 = x /. f1;
f2 = NSolve[Log[x / constant] / Log[1 / λ] + 30 == 30, x][[1]]
x2 = x /. f2;
```

```
{x → 0.000300018}
```

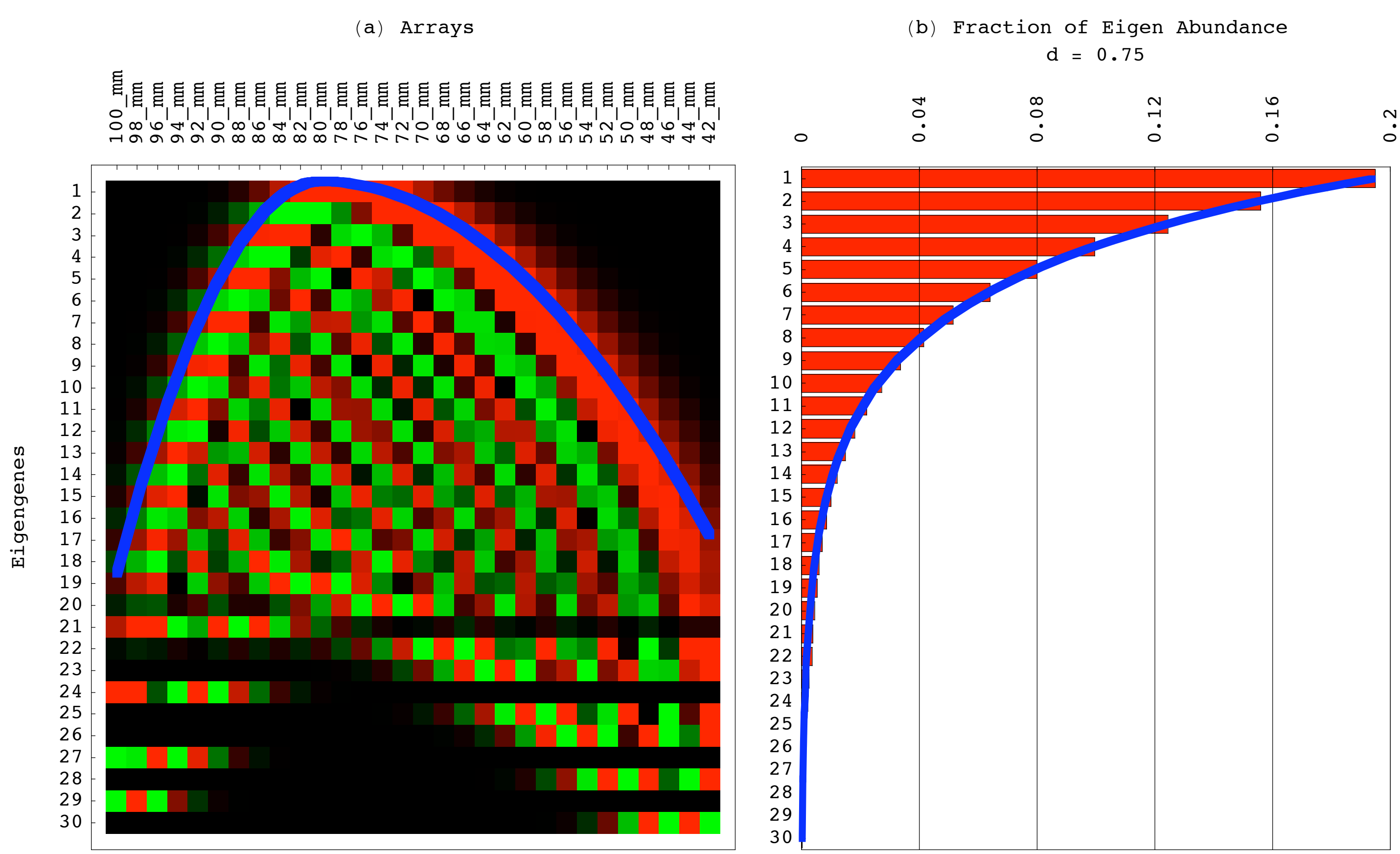
```
{x → 0.193882}
```

```
graph = Plot[Log[x / constant] / Log[1 / λ] + 30, {x, x1, x2},
   PlotRange → {0, 29},
   PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.01]},
   DisplayFunction → Identity];
```

```
g2 = Show[{
    g,
    graph},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction → Identity];
```

```
(* Display Eigengenes and Fractions With Fitting Graph *)
```

```
Show[GraphicsArray[{g1, g2}],
  GraphicsSpacing -> -0.2];
```

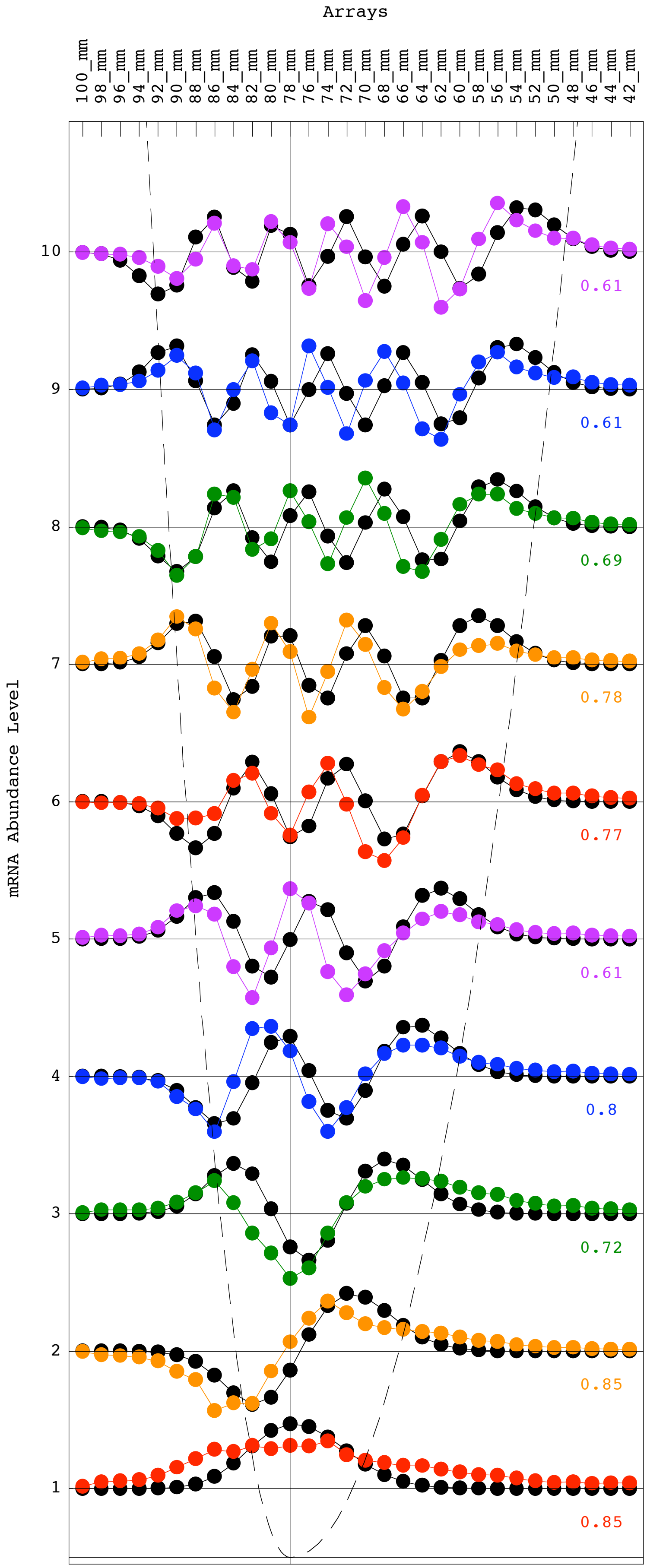



```
(* Fit Eigengenes with Synthetic Eigengenes *)
```

```
correlation = Table[Dot[eigengenes[[n]], syneigengenes[[n]]], {n, 1, 10}];
meancorrelation = Round[100. * Sqrt[Sum[0.1 * correlation[[n]]^2, {n, 1, 10}]]] / 100.
correlation = Round[100. * correlation] / 100.;
```

```
0.73
```

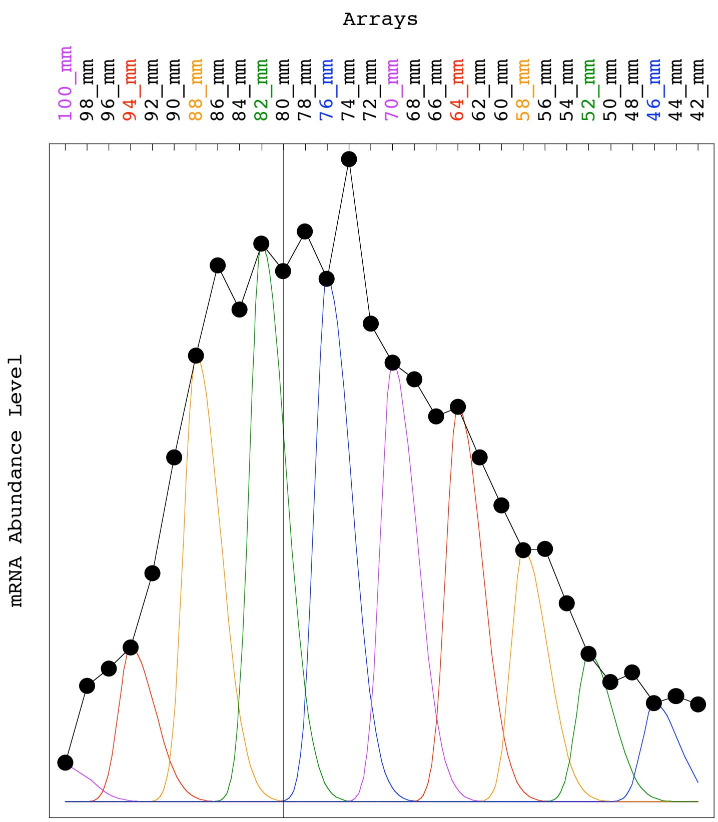
```
(* Fit Differential Equation with an Asymmetric Parabola *)

Clear[f];
f[x_] := If[(x - 11) < 0, 0.5 * k1 * (x - 11) ^ 2, 0.5 * k2 * (x - 11) ^ 2];

(* Create Selected Eigengenes With Fitting Synthetic Eigengenes Graph Display *)

labelx = ColumnForm[{" ", "Arrays"}, Center];
labely = ColumnForm[{" ", "           mRNA Abundance Level"}, Center];
framex = Table[{a - 1, arraynames[[1, a]]}, {a, 1, arrays}];
framey = Table[{n - 0.5, n}, {n, 1, 10}];

points = Table[0, {n, 1, 10}];
lines = Table[0, {n, 1, 10}];
Do[
 {coordinates = Table[{a - 1, eigengenes[[n, a]] + n - 0.5}, {a, 1, arrays}],
  points[[n]] = Table[Point[coordinates[[a]]], {a, 1, arrays}],
  lines[[n]] = Line[coordinates]},
 {n, 1, 10}]
points = Table[Graphics[{color[[Mod[n, 5] + 1]], PointSize[0.022], points[[n]]}], {n, 1, 10}];
lines = Table[Graphics[{color[[Mod[n, 5] + 1]], lines[[n]]}], {n, 1, 10}];

synpoints = Table[0, {n, 1, 10}];
synlines = Table[0, {n, 1, 10}];
Do[
 {coordinates = Table[{a - 1, syneigengenes[[n, a]] + n - 0.5}, {a, 1, arrays}],
  synpoints[[n]] = Table[Point[coordinates[[a]]], {a, 1, arrays}],
  synlines[[n]] = Line[coordinates]},
 {n, 1, 10}]
synpoints = Table[Graphics[{RGBColor[0, 0, 0], PointSize[0.022], synpoints[[n]]}], {n, 1, 10}];
synlines = Table[Graphics[{RGBColor[0, 0, 0], synlines[[n]]}], {n, 1, 10}];
texts = Table[
   Graphics[{color[[Mod[n, 5] + 1]], Text[correlation[[n]], {27.5, n - 0.75}]}],
   {n, 1, 10}];
inflection = Plot[f[x], {x, 11 - Sqrt[10.45 * 2 / k1], 11 + Sqrt[10.45 * 2 / k2]},
   PlotStyle -> {RGBColor[0, 0, 0], Dashing[{0.03, 0.02}]},
   DisplayFunction → Identity];

g = Show[{synpoints, synlines, points, lines, texts, inflection},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    GridLines -> {{{11, RGBColor[0, 0, 0]}},
     AppendColumns[{{0, RGBColor[0, 0, 0]}}, Table[{a - 0.5, RGBColor[0, 0, 0]}, {a, 1, 10}]]
    },
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.05, 10.45},
    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 + 0.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
        Text[a_, {b_, c_}, {0., -1.}] ->
            Text[a, {b, c + 0.15}, {0, 0}, {0, 1}];
g3 = Show[g,
        AspectRatio → 2.4,
        PlotRange -> All];
```



Arrays
100_mm
98_mm
96_mm
94_mm
92_mm
90_mm
88_mm
86_mm
84_mm
82_mm
80_mm
78_mm
76_mm
74_mm
72_mm
70_mm
68_mm
66_mm
64_mm
62_mm
60_mm
58_mm
56_mm
54_mm
52_mm
50_mm
48_mm
46_mm
44_mm
42_mm
mRNA Abundance Level
10
9
8
7
6
5
4
3
2
1
0.61
0.61
0.69
0.78
0.77
0.61
0.8
0.72
0.85
0.85

```
(* Create Synthetic Data With the Distribution of Genes *)

k1 = 0.36;
k2 = 0.09;
a = 0.405;
b = 0.0050625;

Clear[f, x, p];
f[x_, p_, a_, b_] =
  If[x < p,
   N[Exp[-4 * a * (x - p) ^2]],
   N[Exp[-a * (x - p) ^2]]];
synmatrix = Table[f[x, p, a, b] * distribution[[p + 11]], {p, -10, 19}, {x, -10, 19}];
{syngenes, arrays} = Dimensions[synmatrix]
Clear[α, β, λ]

{30, 30}

(* Create Synthetic Data Graph Display *)

graphs = Table[
   Plot[f[x, p, a, b] * distribution[[p + 11]],
    {x, -10, 19},
    PlotStyle -> color[[Mod[(p + 10) / 3, 5] + 1]],
    Axes → False,
    Frame → True,
    PlotRange → All,
    DisplayFunction → Identity],
   {p, -10, 19, 3}];
```

```
(* Display Synthetic Data With the Distribution of the Genes *)

coordinates = Table[{n - 11, distribution[[n]]}, {n, 1, arrays}];
points = Table[Point[coordinates[[n]]], {n, 1, arrays}];
points = Graphics[{RGBColor[0, 0, 0], PointSize[0.022], points}];
lines = Graphics[{RGBColor[0, 0, 0], Line[coordinates]}];

framex = Table[{n - 11, arraynames[[1, n]]}, {n, 1, arrays}];
Do[framex[[n]] = {n - 11, StyleForm[arraynames[[1, n]], FontColor -> color[[Mod[(n - 1) / 3, 5] + 1]]]},
 {n, 1, arrays, 3}]
labely = "mRNA Abundance Level";
labelx = ColumnForm[{"Arrays", " ", " "}, Center];
g = Show[{graphs, points, lines},
   Frame -> True,
   FrameLabel -> {None, labely, labelx, None},
   GridLines -> {{{0, RGBColor[0, 0, 0]}}, None},
   FrameTicks -> {None, None, framex, None},
   DisplayFunction → Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {m_, c_}, {1., 0.}] →
    Text[labely, {m - 1, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {m_, c_}, {0., -1.}] →
    Text[labelx, {m, c + 0.07}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[n_, {m_, c_}, {0., -1.}] →
    Text[n, {m, c + 0.08}, {0, 0}, {0, 1}];
g4 = Show[g,
   AspectRatio -> 1.05,
   PlotRange → All];
Clear[a, b];
```



```
(* Calculate SVD *)

{eigenarrays, eigenabundances, syneigengenes} = SingularValues[synmatrix];
fractions = eigenabundances^2 / Sum[eigenabundances[[a]]^2, {a, 1, arrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, arrays}] / Log[arrays]];
entropy = N[Round[100 * entropy] / 100]

0.73

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

contrast = 3.5;
displaying = Table[
   If[contrast * syneigengenes[[i, j]] > 0,
    If[contrast * syneigengenes[[i, j]] < 1, {contrast * syneigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * syneigengenes[[i, j]] > -1, {0, -contrast * syneigengenes[[i, j]]}, {0, 1}]],
   {i, 1, arrays}, {j, 1, arrays}];
framex = Table[{a - 0.5, arraynames[[1, 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 - 3, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] →
    Text[labelx, {b, c + 2.75}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c + 2.25}, {0, 0}, {0, 1}];
g3 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction → Identity];

(* Fit Eigengenes With an Asymmetric Parabola *)

k1 = 0.36;
k2 = 0.09;

Clear[f];
f[x_] := If[(x - 11) < 0, 0.5 * k1 * (x - 11)^2, 0.5 * k2 * (x - 11)^2];
inflection = Plot[arrays - f[x + 0.5], {x, 0.5, arrays - 0.5},
   PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.013]},
   DisplayFunction → Identity];
g1 = Show[{g3, inflection},
   AspectRatio → 1.05,
   DisplayFunction → Identity];
```

```
(* Fit Fractions *)

Clear[constant, λ];
f = FindFit[Table[fractions[[n]], {n, 2, 10}], constant * λ^x, {constant, λ}, x]
```

{constant → 0.213226, λ → 0.782208}

```
λ = 0.8;
constant = constant /. f;
correlation = Dot[Table[fractions[[n]], {n, 2, 10}], Table[constant * λ^n, {n, 2, 10}]] /
   Sqrt[Dot[Table[fractions[[n]], {n, 2, 10}], Table[fractions[[n]], {n, 2, 10}]]] /
  Sqrt[Dot[Table[constant * λ^n, {n, 2, 10}], Table[constant * λ^n, {n, 2, 10}]]]
```

0.998998

```
(* Create Fractions Bar Chart Displays With Fitting Graph *)

fractions[[1]]
```

0.210146

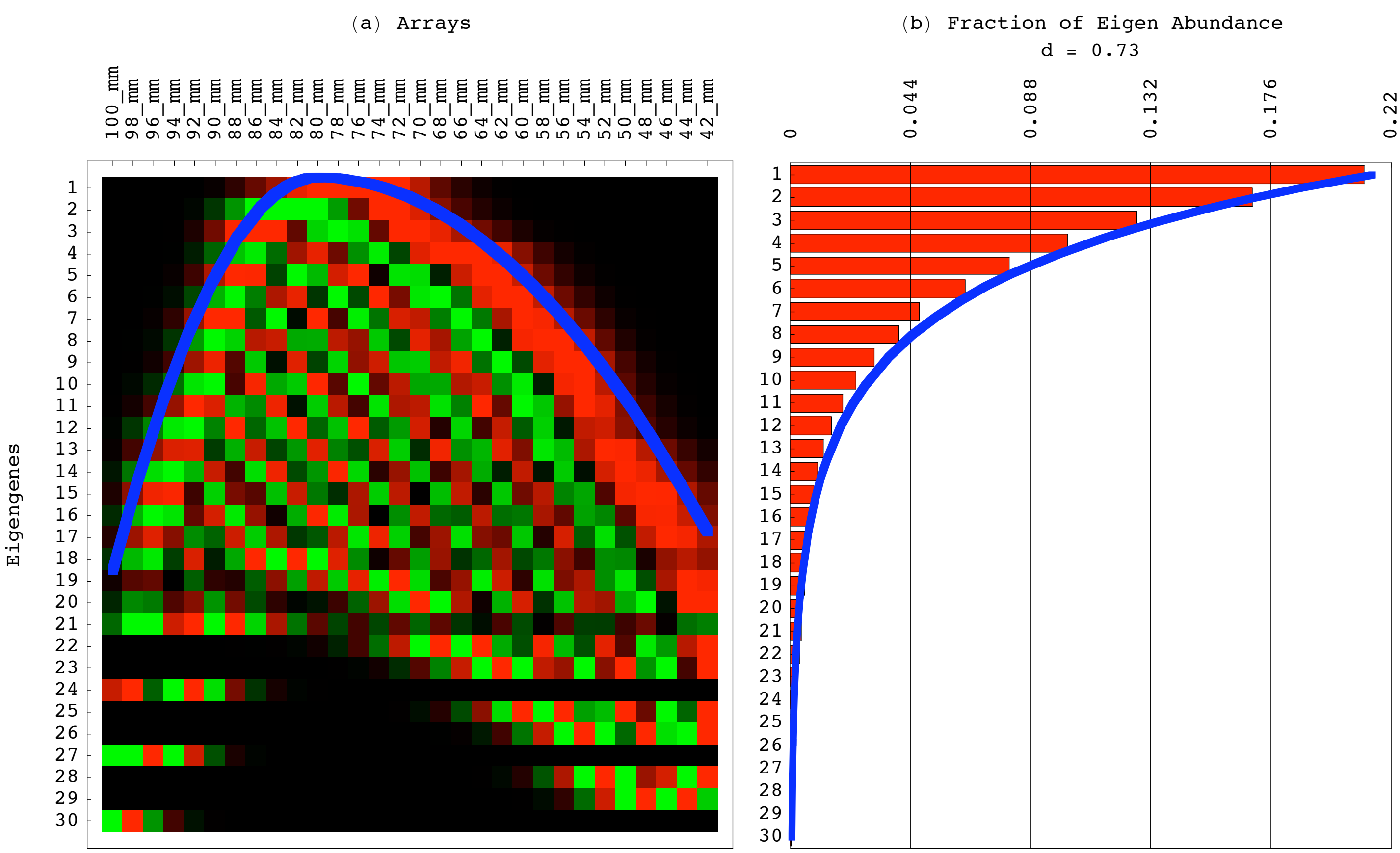
```
limit = 0.22;

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}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, arrays - a}, {a, 0, arrays - 1}];
labelx = ColumnForm[
   {"(b) Fraction of Eigen Abundance", StringJoin["d = ", ToString[entropy]], " "},
   Center];
labely = ColumnForm[{" ", " "}, 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, labely, labelx, None},
   GridLines -> {gridx, None},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] →
    Text[labely, {b - 0.04, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] →
    Text[labelx, {b, c + 2.75}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c + 2.25}, {0, 0}, {0, 1}];
```

```
f1 = NSolve[Log[x / constant] / Log[1 / λ] + 30 == 1, x][[1]]
x1 = x /. f1;
f2 = NSolve[Log[x / constant] / Log[1 / λ] + 30 == 30, x][[1]]
x2 = x /. f2;
```

```
{x → 0.000329951}
```

```
{x → 0.213226}
```

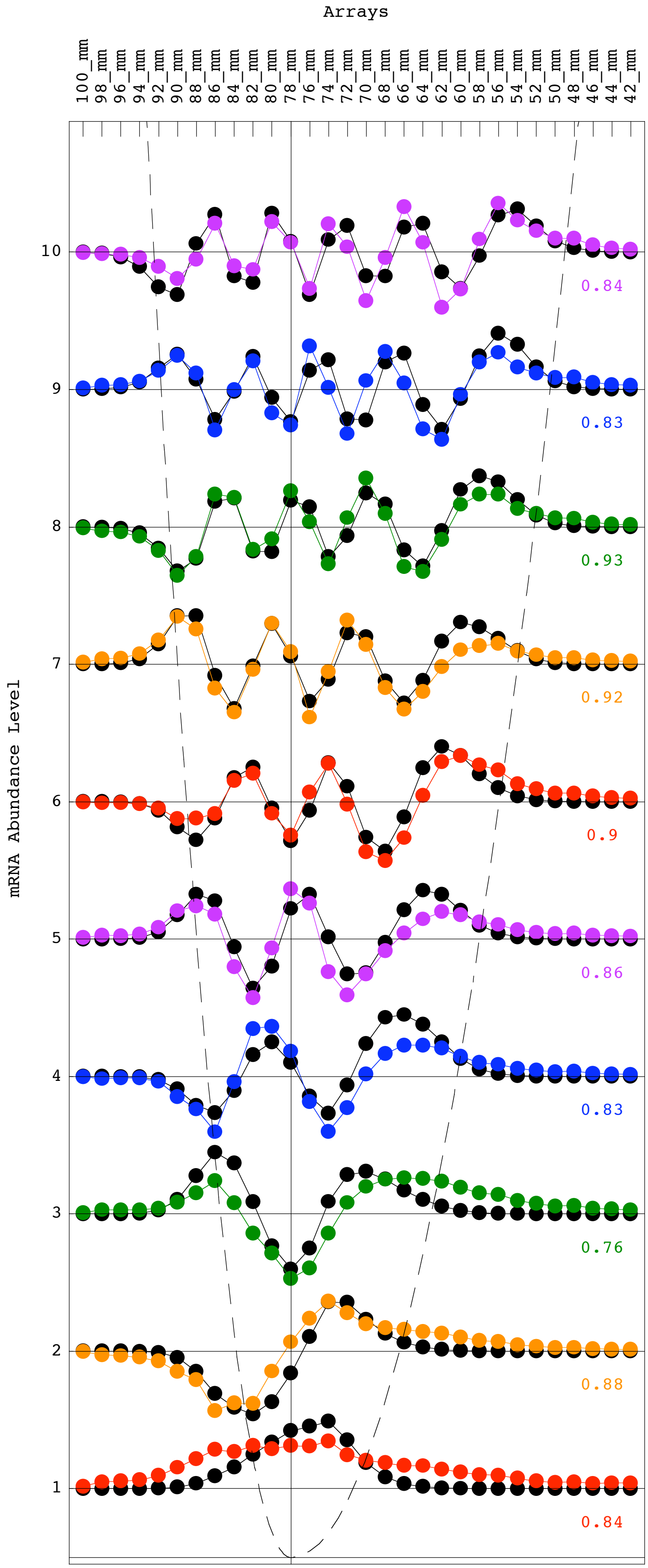
```
graph = Plot[Log[x / constant] / Log[1 / λ] + 30, {x, x1, x2},
   PlotRange → {0, 29},
   PlotStyle -> {RGBColor[0, 0, 1], Thickness[0.01]},
   DisplayFunction → Identity];

g2 = Show[{
    g,
    graph},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction → Identity];

(* Display Eigengenes and Fractions With Fitting Graph *)

Show[GraphicsArray[{g1, g2}],
  GraphicsSpacing -> -0.2];
```




```
(* Fit Eigengenes with Synthetic Eigengenes *)

correlation = Table[Dot[eigengenes[[n]], syneigengenes[[n]]], {n, 1, 10}];
meancorrelation = Round[100. * Sqrt[Sum[0.1 * correlation[[n]]^2, {n, 1, 10}]]] / 100.
correlation = Round[100. * correlation] / 100.;
```

```
0.86
```

```
(* Fit Differential Equation with an Asymmetric Parabola *)

Clear[f];
f[x_] := If[(x - 11) < 0, 0.5 * k1 * (x - 11) ^ 2, 0.5 * k2 * (x - 11) ^ 2];

(* Create Selected Eigengenes With Fitting Synthetic Eigengenes Graph Display *)

labelx = ColumnForm[{" ", "Arrays"}, Center];
labely = ColumnForm[{" ", "           mRNA Abundance Level"}, Center];
framex = Table[{a - 1, arraynames[[1, a]]}, {a, 1, arrays}];
framey = Table[{n - 0.5, n}, {n, 1, 10}];

points = Table[0, {n, 1, 10}];
lines = Table[0, {n, 1, 10}];
Do[
 {coordinates = Table[{a - 1, eigengenes[[n, a]] + n - 0.5}, {a, 1, arrays}],
  points[[n]] = Table[Point[coordinates[[a]]], {a, 1, arrays}],
  lines[[n]] = Line[coordinates]},
 {n, 1, 10}]
points = Table[Graphics[{color[[Mod[n, 5] + 1]], PointSize[0.022], points[[n]]}], {n, 1, 10}];
lines = Table[Graphics[{color[[Mod[n, 5] + 1]], lines[[n]]}], {n, 1, 10}];

synpoints = Table[0, {n, 1, 10}];
synlines = Table[0, {n, 1, 10}];
Do[
 {coordinates = Table[{a - 1, syneigengenes[[n, a]] + n - 0.5}, {a, 1, arrays}],
  synpoints[[n]] = Table[Point[coordinates[[a]]], {a, 1, arrays}],
  synlines[[n]] = Line[coordinates]},
 {n, 1, 10}]
synpoints = Table[Graphics[{RGBColor[0, 0, 0], PointSize[0.022], synpoints[[n]]}], {n, 1, 10}];
synlines = Table[Graphics[{RGBColor[0, 0, 0], synlines[[n]]}], {n, 1, 10}];
texts = Table[
   Graphics[{color[[Mod[n, 5] + 1]], Text[correlation[[n]], {27.5, n - 0.75}]}],
   {n, 1, 10}];
inflection = Plot[f[x], {x, 11 - Sqrt[10.45 * 2 / k1], 11 + Sqrt[10.45 * 2 / k2]},
   PlotStyle -> {RGBColor[0, 0, 0], Dashing[{0.03, 0.02}]},
   DisplayFunction → Identity];

g = Show[{synpoints, synlines, points, lines, texts, inflection},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    GridLines -> {{{11, RGBColor[0, 0, 0]}},
     AppendColumns[{{0, RGBColor[0, 0, 0]}}, Table[{a - 0.5, RGBColor[0, 0, 0]}, {a, 1, 10}]]
    },
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.05, 10.45},
    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 + 0.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
        Text[a_, {b_, c_}, {0., -1.}] ->
            Text[a, {b, c + 0.15}, {0, 0}, {0, 1}];
g3 = Show[g,
        AspectRatio → 2.4,
        PlotRange -> All];
```



Arrays
100_mm
98_mm
96_mm
94_mm
92_mm
90_mm
88_mm
86_mm
84_mm
82_mm
80_mm
78_mm
76_mm
74_mm
72_mm
70_mm
68_mm
66_mm
64_mm
62_mm
60_mm
58_mm
56_mm
54_mm
52_mm
50_mm
48_mm
46_mm
44_mm
42_mm
mRNA Abundance Level
10
9
8
7
6
5
4
3
2
1
0.84
0.83
0.93
0.92
0.9
0.86
0.83
0.76
0.88
0.84