```
(* © Orly Alter 2009 *)
```



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

```
(* Tensor HOSVD Modeling of DNA Microarray Data *)
```

```
(* Combine Data from Two Different DNA Microarray Platforms *)
```

```
(* Read Relative mRNA Expression Data from the First DNA Microarray Platform *)

stream = "Desktop/96/Data/24_SC18_R.txt";
matrix = Import[stream, "Table"];
{spots, arrays} = Dimensions[matrix] - {3, 1}
Clear[stream];
```

```
{4771, 24}
```

```
genenames = TakeRows[
   TakeColumns[matrix, {1, 1}],
   {4, spots + 3}];
arraynames = TakeColumns[
   TakeRows[matrix, {1, 3}],
   {2, arrays + 1}];
matrix = TakeColumns[
   TakeRows[matrix, {4, spots + 3}],
   {2, arrays + 1}];
matrix = ToExpression[matrix];

(* Convert Data to Log2 of Relative mRNA Expression *)

matrix = N[Log[2, matrix]];
matrix = ReplaceAll[matrix, N[Log[Null] / Log[2]] -> "Null"];

(* Average Data from Multiple Probes of the Same Gene *)

Do[If[genenames[[a]] == genenames[[a + 1]],
  If[genenames[[a]] == genenames[[a + 2]],
   If[genenames[[a]] == genenames[[a + 3]],
    Do[{count = Count[{matrix[[a, b]], matrix[[a + 1, b]], matrix[[a + 2, b]], matrix[[a + 3, b]]}, "Null"],
      If[count == 4,
       matrix[[a, b]] = "Null",
       matrix[[a, b]] = ReplaceAll[(
           matrix[[a, b]] + matrix[[a + 1, b]] + matrix[[a + 2, b]] + matrix[[a + 3, b]]
          ) / (4 - count), "Null" -> 0]]},
     {b, 1, arrays}]]],
  matrix[[a]] = matrix[[a]]],
 {a, 1, spots - 3}]
```

```
Do[If[genenames[[a]] == genenames[[a + 1]],
  If[genenames[[a]] == genenames[[a + 2]],
   If[genenames[[a]] == genenames[[a + 3]],
    {matrix = Drop[matrix, {a + 1, a + 3}],
     genenames = Drop[genenames, {a + 1, a + 3}]}]]],
 {a, spots - 3, 1, -1}]
{spots, arrays} = Dimensions[matrix]
```

```
{4762, 24}
```

```
Do[If[genenames[[a]] == genenames[[a + 1]],
  If[genenames[[a]] == genenames[[a + 2]],
   Do[{count = Count[{matrix[[a, b]], matrix[[a + 1, b]], matrix[[a + 2, b]]}, "Null"],
     If[count == 3,
      matrix[[a, b]] = "Null",
      matrix[[a, b]] = ReplaceAll[(
          matrix[[a, b]] + matrix[[a + 1, b]] + matrix[[a + 2, b]]
         ) / (3 - count), "Null" -> 0]]},
    {b, 1, arrays}]],
  matrix[[a]] = matrix[[a]]],
 {a, 1, spots - 2}]
Do[If[genenames[[a]] == genenames[[a + 1]],
  If[genenames[[a]] == genenames[[a + 2]],
   {matrix = Drop[matrix, {a + 1, a + 2}],
    genenames = Drop[genenames, {a + 1, a + 2}]}]],
 {a, spots - 2, 1, -1}]
{spots, arrays} = Dimensions[matrix]
```

```
{4558, 24}
```

```
Do[If[genenames[[a]] == genenames[[a + 1]],
  Do[{count = Count[{matrix[[a, b]], matrix[[a + 1, b]]}, "Null"],
    If[count == 2,
     matrix[[a, b]] = "Null",
     matrix[[a, b]] = ReplaceAll[(
         matrix[[a, b]] + matrix[[a + 1, b]]
        ) / (2 - count), "Null" -> 0]]},
   {b, 1, arrays}],
  matrix[[a]] = matrix[[a]]],
 {a, 1, spots - 1}]
Do[If[genenames[[a]] == genenames[[a + 1]],
  {matrix = Drop[matrix, {a + 1}],
   genenames = Drop[genenames, {a + 1}]}],
 {a, spots - 1, 1, -1}]
sc18 = AppendColumns[arraynames, matrix];
```

```
(* Read Relative mRNA Expression Data from the Second DNA Microarray Platform *)
```

```
stream = "Desktop/96/Data/72_STL_R.txt";
matrix = Import[stream, "Table"];
{spots, arrays} = Dimensions[matrix] - {3, 1}
Clear[stream];
```

```
{8540, 72}
```

```
genenames = TakeRows[TakeColumns[matrix, {1, 1}],
   {4, spots + 3}];
arraynames = TakeColumns[TakeRows[matrix, {1, 3}],
   {2, arrays + 1}];
matrix = TakeColumns[TakeRows[matrix, {4, spots + 3}],
   {2, arrays + 1}];
matrix = ToExpression[matrix];
```

```
(* Convert Data to Log2 of Relative mRNA Expression *)
```



```
matrix = N[Log[2, matrix]];
matrix = ReplaceAll[matrix, N[Log[Null] / Log[2]] -> "Null"];
```

```
(* Average Data from Multiple Probes of the Same Gene *)

Do[If[genenames[[a]] == genenames[[a + 1]],
  Do[{count = Count[{matrix[[a, b]], matrix[[a + 1, b]]}, "Null"],
    If[count == 2,
     matrix[[a, b]] = "Null",
     matrix[[a, b]] = ReplaceAll[(
         matrix[[a, b]] + matrix[[a + 1, b]]
       ) / (2 - count), "Null" -> 0]]},
   {b, 1, arrays}],
  matrix[[a]] = matrix[[a]]],
 {a, 1, spots - 1}]
Do[If[genenames[[a]] == genenames[[a + 1]],
  {matrix = Drop[matrix, {a + 1}],
   genenames = Drop[genenames, {a + 1}]}],
 {a, spots - 1, 1, -1}]
stl = AppendColumns[arraynames, matrix];

(* Combine Data from the Two Different DNA Microarray Platforms *)

matrix = N[AppendRows[
    AppendColumns[{{"YORF"}, {"YORF"}, {"YORF"}}, genenames],
    TakeColumns[sc18, {1, 12}],
    TakeColumns[stl, {1, 36}],
    TakeColumns[sc18, {13, 24}],
    TakeColumns[stl, {37, 72}]]];
{genes, arrays} = Dimensions[matrix] - {3, 1}
Clear[sc18, stl];
```

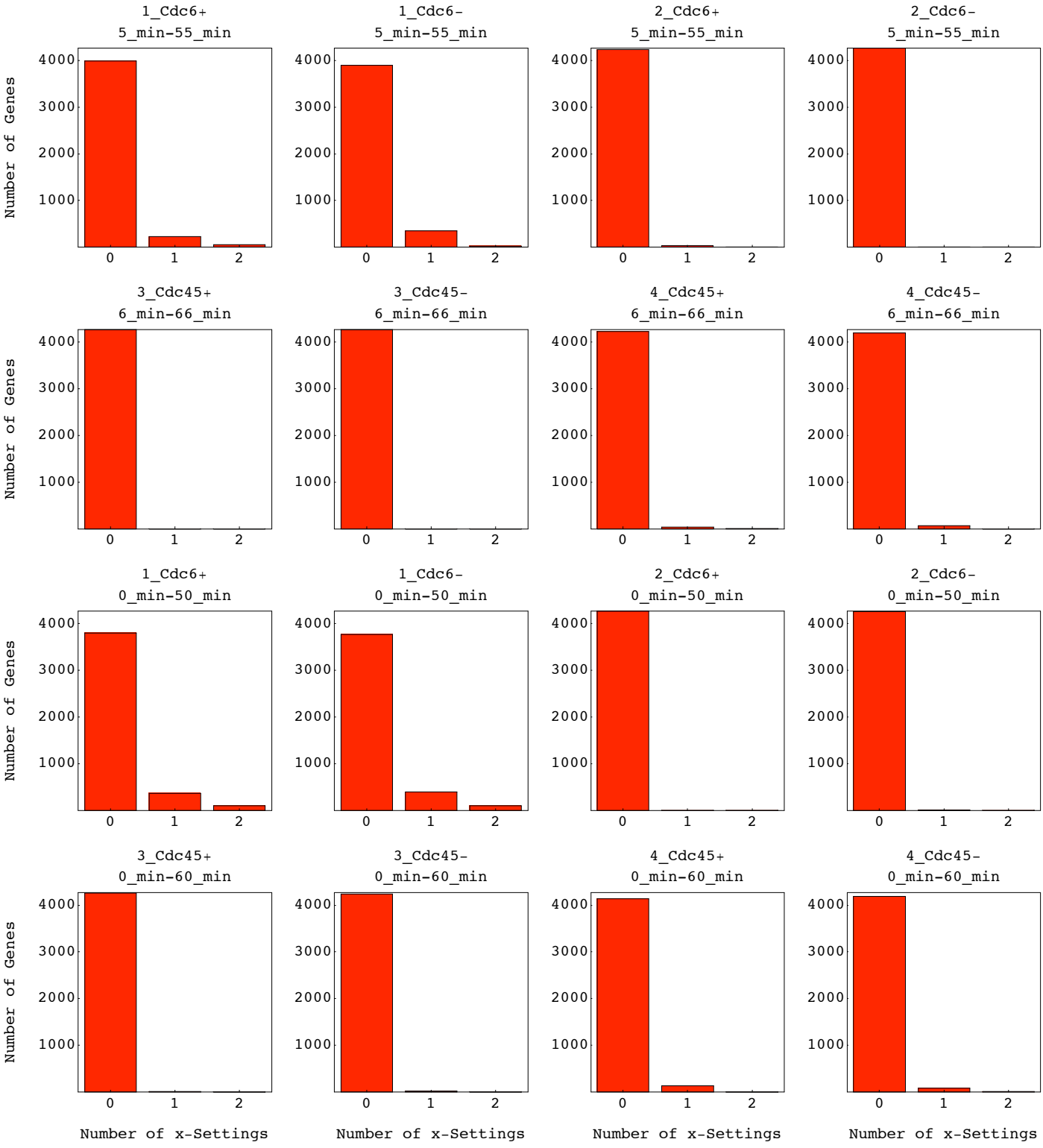
```
{4270, 96}
```

```
genenames = TakeRows[
   TakeColumns[matrix, {1, 1}],
   {4, genes + 3}];
arraynames = TakeColumns[
   TakeRows[matrix, {1, 3}],
   {2, arrays + 1}];
matrix = TakeColumns[
   TakeRows[matrix, {4, genes + 3}],
   {2, arrays + 1}];
matrix = ToExpression[matrix];

tensor = {
   TakeColumns[matrix, {1, 6}], TakeColumns[matrix, {7, 12}],
   TakeColumns[matrix, {13, 18}], TakeColumns[matrix, {19, 24}],
   TakeColumns[matrix, {25, 30}], TakeColumns[matrix, {31, 36}],
   TakeColumns[matrix, {37, 42}], TakeColumns[matrix, {43, 48}],
   TakeColumns[matrix, {49, 54}], TakeColumns[matrix, {55, 60}],
   TakeColumns[matrix, {61, 66}], TakeColumns[matrix, {67, 72}],
   TakeColumns[matrix, {73, 78}], TakeColumns[matrix, {79, 84}],
   TakeColumns[matrix, {85, 90}], TakeColumns[matrix, {91, 96}]};
{yarrays, zgenes, xarrays} = Dimensions[tensor]
```

```
{16, 4270, 6}
```

```
tensornames = {
   TakeColumns[arraynames, {1, 6}], TakeColumns[arraynames, {7, 12}],
   TakeColumns[arraynames, {13, 18}], TakeColumns[arraynames, {19, 24}],
   TakeColumns[arraynames, {25, 30}], TakeColumns[arraynames, {31, 36}],
   TakeColumns[arraynames, {37, 42}], TakeColumns[arraynames, {43, 48}],
   TakeColumns[arraynames, {49, 54}], TakeColumns[arraynames, {55, 60}],
   TakeColumns[arraynames, {61, 66}], TakeColumns[arraynames, {67, 72}],
   TakeColumns[arraynames, {73, 78}], TakeColumns[arraynames, {79, 84}],
   TakeColumns[arraynames, {85, 90}], TakeColumns[arraynames, {91, 96}]};
```

```
(* Estimate Missing Data Using SVD *)

(* Count Missing Data *)

counter = Table[0, {yarrays}];
numbers = Table[0, {yarrays}];
gnumbers = Table[0, {yarrays}];
framex = Table[{a, a - 1}, {a, 1, 3}];
framey = {1000, 2000, 3000, 4000};

Do[{
  matrix = tensor[[t]],
  counter[[t]] = Table[Dimensions[Position[matrix[[a]], Null]][[1]], {a, 1, genes}],
  positions = Table[0, {a, 1, xarrays + 1}],
  Do[
   positions[[a]] = Flatten[Position[Flatten[counter[[t]]], a - 1]],
   {a, 1, xarrays + 1}],
  numbers[[t]] = Flatten[Table[Dimensions[positions[[a]]],
     {a, 1, 3}]],
  labelx1 = If[t > 12, "Number of x-Settings", " "],
  labelx2 = ColumnForm[{" ", tensornames[[t, 2, 1]],
     StringJoin[tensornames[[t, 3, 1]], "-", tensornames[[t, 3, 6]]]}, Center],
  labely = If[Mod[t, 4] == 1, "Number of Genes", " "],
  g = BarChart[numbers[[t]],
    Frame -> True,
    Axes -> False,
    FrameLabel -> {labelx1, labely, labelx2, None},
    FrameTicks -> {framex, framey, None, None},
    GridLines -> {None, None},
    PlotRange -> {{0.5, 3 + 0.5}, {0, zgenes}},
    DisplayFunction -> Identity],
  g = FullGraphics[g],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labely, {b_, c_}, {1., 0.}] ->
     Text[labely, {b - 1, c}, {0, 0}, {0, 1}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labelx1, {b_, c_}, {0., 1.}] ->
     Text[labelx1, {b, c - 600}, {0, 1}, {1, 0}],
  gnumbers[[t]] = Show[g,
    AspectRatio -> 0.9,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]

(* Display Missing Data Counts *)

g = Show[GraphicsArray[Table[{GraphicsArray[Table[
        gnumbers[[a + 4 * b]], {a, 1, 4}]], GraphicsSpacing -> -0.275]},
     {b, 0, 3}]], GraphicsSpacing -> -0.1575,
   PlotRange -> {{0, 1}, {0, 1}}];
Clear[gnumbers];
```


1_Cdc6+
5_min-55_min
1_Cdc6-
5_min-55_min
2_Cdc6+
5_min-55_min
2_Cdc6-
5_min-55_min
3_Cdc45+
6_min-66_min
3_Cdc45-
6_min-66_min
4_Cdc45+
6_min-66_min
4_Cdc45-
6_min-66_min
1_Cdc6+
0_min-50_min
1_Cdc6-
0_min-50_min
2_Cdc6+
0_min-50_min
2_Cdc6-
0_min-50_min
3_Cdc45+
0_min-60_min
3_Cdc45-
0_min-60_min
4_Cdc45+
0_min-60_min
4_Cdc45-
0_min-60_min
Number of Genes
Number of x-Settings
1000
2000
3000
4000
0
1
2

```
(* Sort Genes by Number of Arrays with Missing Data *)

fullgenenames = Table[0, {yarrays}];
missinggenenames1 = Table[0, {yarrays}];
missinggenenames2 = Table[0, {yarrays}];
fullmatrix = Table[0, {yarrays}];
missingmatrix1 = Table[0, {yarrays}];
missingmatrix2 = Table[0, {yarrays}];
Do[{
  matrix = AppendRows[Transpose[{counter[[t]]}], genenames, tensor[[t]]],
  matrix = Sort[matrix, OrderedQ[{#1, #2}] &],
  fullgenenames[[t]] = TakeColumns[
    TakeRows[matrix, {1, numbers[[t, 1]]}],
    {2, 2}],
  fullmatrix[[t]] = TakeColumns[
    TakeRows[matrix, {1, numbers[[t, 1]]}],
    {3, xarrays + 2}],
  missinggenenames1[[t]] = TakeColumns[
    TakeRows[matrix, {numbers[[t, 1]] + 1, numbers[[t, 1]] + numbers[[t, 2]]}],
    {2, 2}],
  missingmatrix1[[t]] = TakeColumns[
    TakeRows[matrix, {numbers[[t, 1]] + 1, numbers[[t, 1]] + numbers[[t, 2]]}],
    {3, xarrays + 2}],
  missinggenenames2[[t]] = TakeColumns[
    TakeRows[matrix,
     {numbers[[t, 1]] + numbers[[t, 2]] + 1,
      numbers[[t, 1]] + numbers[[t, 2]] + numbers[[t, 3]]}],
    {2, 2}],
  missingmatrix2[[t]] = TakeColumns[
    TakeRows[matrix,
     {numbers[[t, 1]] + numbers[[t, 2]] + 1,
      numbers[[t, 1]] + numbers[[t, 2]] + numbers[[t, 3]]}],
    {3, xarrays + 2}]},
 {t, 1, yarrays}]
Clear[counter];

(* Sort Genes by Position of Arrays with Missing Data *)

locator1 = Table[0, {yarrays}];
locator2 = Table[0, {yarrays}];
Do[{
  locator1[[t]] = Table[0, {numbers[[t, 2]]}],
  Do[
   locator1[[t, a]] = locator1[[t, a]] + Flatten[Position[missingmatrix1[[t, a]], Null]],
   {a, 1, numbers[[t, 2]]}],
  locator2[[t]] = Table[0, {numbers[[t, 3]]}],
  Do[
   locator2[[t, a]] = locator2[[t, a]] + Flatten[Position[missingmatrix2[[t, a]], Null]],
   {a, 1, numbers[[t, 3]]}],
  missingmatrix1[[t]] = AppendRows[locator1[[t]], missinggenenames1[[t]], missingmatrix1[[t]]],
  missingmatrix1[[t]] = Sort[missingmatrix1[[t]], OrderedQ[{#1, #2}] &],
  locator1[[t]] = TakeColumns[missingmatrix1[[t]], {1, 1}],
  missinggenenames1[[t]] = TakeColumns[missingmatrix1[[t]], {2, 2}],
  missingmatrix1[[t]] = TakeColumns[missingmatrix1[[t]], {3, xarrays + 2}],
  missingmatrix2[[t]] = AppendRows[locator2[[t]], missinggenenames2[[t]], missingmatrix2[[t]]],
  missingmatrix2[[t]] = Sort[missingmatrix2[[t]], OrderedQ[{#1, #2}] &],
  locator2[[t]] = TakeColumns[missingmatrix2[[t]], {1, 2}],
  missinggenenames2[[t]] = TakeColumns[missingmatrix2[[t]], {3, 3}],
  missingmatrix2[[t]] = TakeColumns[missingmatrix2[[t]], {4, xarrays + 3}]},
 {t, 1, yarrays}]
```

```
(* Calculate SVD Before Missing Data Estimation *)

fulleigengenes = Table[0, {yarrays}];
fullfractions = Table[0, {yarrays}];
fullentropy = Table[0, {yarrays}];
Do[{{eigenarrays, eigenexpressions, fulleigengenes[[t]]} = SingularValues[fullmatrix[[t]]],
  fullfractions[[t]] = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, xarrays}],
  fullentropy[[t]] =
   -N[Round[100 * Sum[fullfractions[[t, a]] * Log[fullfractions[[t, a]]],
         {a, 1, xarrays}] / Log[xarrays]] / 100]},
 {t, 1, yarrays}]

(* Estimate Missing Data Using the Most Significant Eigengene *)

Do[{
  Do[
   missingmatrix1[[t, a, locator1[[t, a, 1]]]] =
    N[Round[Flatten[Dot[Dot[
            Transpose[Drop[
              Transpose[{missingmatrix1[[t, a]]}],
              {locator1[[t, a, 1]]}]],
            PseudoInverse[Transpose[Drop[
               Transpose[{fulleigengenes[[t, 1]]}],
               {locator1[[t, a, 1]]}]]]],
           {fulleigengenes[[t, 1]]}]][[locator1[[t, a, 1]]]] * 100] / 100],
   {a, 1, numbers[[t, 2]]}],
  Do[Do[
    missingmatrix2[[t, a, locator2[[t, a, b]]]] =
     N[Round[Flatten[Dot[Dot[
             Transpose[Drop[Drop[
                Transpose[{missingmatrix2[[t, a]]}],
                {locator2[[t, a, 2]]}], {locator2[[t, a, 1]]}]],
             PseudoInverse[Transpose[Drop[Drop[
                 Transpose[{fulleigengenes[[t, 1]]}],
                 {locator2[[t, a, 2]]}], {locator2[[t, a, 1]]}]]]],
            {fulleigengenes[[t, 1]]}]][[locator2[[t, a, b]]]] * 100] / 100],
    {b, 1, 2}],
   {a, 1, numbers[[t, 3]]}],
  fullmatrix[[t]] = AppendRows[fullgenenames[[t]], fullmatrix[[t]]],
  missingmatrix1[[t]] = AppendRows[missinggenenames1[[t]], missingmatrix1[[t]]],
  missingmatrix2[[t]] = AppendRows[missinggenenames2[[t]], missingmatrix2[[t]]],
  If[Dimensions[missingmatrix1[[t]]] == {2},
   tensor[[t]] = fullmatrix[[t]],
   tensor[[t]] = AppendColumns[fullmatrix[[t]], missingmatrix1[[t]]]],
  If[Dimensions[missingmatrix2[[t]]] == {2},
   tensor[[t]] = tensor[[t]],
   tensor[[t]] = AppendColumns[tensor[[t]], missingmatrix2[[t]]]],
  tensor[[t]] = TakeColumns[
    Sort[tensor[[t]], OrderedQ[{#1, #2}] &],
    {2, xarrays + 1}]},
 {t, 1, yarrays}]
Clear[fullmatrix, missingmatrix1, missingmatrix2, locator1, locator2, numbers];

(* Calculate SVD After Missing Data Estimation *)

eigengenes = Table[0, {yarrays}];
fractions = Table[0, {yarrays}];
entropy = Table[0, {yarrays}];
Do[{{eigenarrays, eigenexpressions, eigengenes[[t]]} = SingularValues[tensor[[t]]],
  fractions[[t]] = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, xarrays}], entropy[[t]] =
   -N[Round[100 * Sum[fractions[[t, a]] * Log[fractions[[t, a]]], {a, 1, xarrays}] / Log[xarrays]] / 100]},
 {t, 1, yarrays}]
```

```
(* Display SVD Before and After Missing Data Estimation *)

(* Create Bar Chart Display of Fractions Before Missing Data Estimation *)

g2 = Table[0, {yarrays}];
gridx = Table[a, {a, 0, 1, 0.2}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
framex[[6]] = {1, "1"};
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, xarrays - a}, {a, 0, xarrays - 1}];
Do[{
  labelx = ColumnForm[
    {"(b) Eigenexpression Fraction",
     DisplayForm[RowBox[{SubsuperscriptBox["d", t, ′], " = ", ToString[fullentropy[[t]]]}]], " "},
    Center],
  g = BarChart[
    Table[fullfractions[[t, xarrays - a]], {a, 0, xarrays - 1}],
    BarOrientation -> Horizontal,
    PlotRange -> {{0, 1.0001}, {0.5, xarrays + 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 + 0.95}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  g2[[t]] = Show[g,
    AspectRatio -> 1.35,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]
```

```
(* Create Bar Chart Display of Fractions After Missing Data Estimation *)

g5 = Table[0, {yarrays}];
gridx = Table[a, {a, 0, 1, 0.2}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
framex[[6]] = {1, "1"};
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, xarrays - a}, {a, 0, xarrays - 1}];
Do[{
  labelx = ColumnForm[
    {"(e) Eigenexpression Fraction",
     DisplayForm[RowBox[{SubsuperscriptBox["d", t, ""], " = ", ToString[entropy[[t]]]}]], " "},
    Center],
  g = BarChart[
    Table[fractions[[t, xarrays - a]], {a, 0, xarrays - 1}],
    BarOrientation -> Horizontal,
    PlotRange -> {{0, 1.0001}, {0.5, xarrays + 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 + 0.95}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  g5[[t]] = Show[g,
    AspectRatio -> 1.35,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]
```

```
(* Create 2 D Red & Green Raster Display of x-Eigengenes Before Missing Data Estimation *)

g1 = Table[0, {yarrays}];
contrast = 3.5;
framey = Table[{a + 1 - 0.5, xarrays - a}, {a, 0, xarrays - 1}];
labely = "x-Eigengenes";
labelx = ColumnForm[{"(a)  x-Settings", " ", " "}, Center];
Do[{
  displaying = Table[
    If[contrast * fulleigengenes[[t, i, j]] > 0,
     If[contrast * fulleigengenes[[t, i, j]] < 1, {contrast * fulleigengenes[[t, i, j]], 0}, {1, 0}],
     If[contrast * fulleigengenes[[t, i, j]] > -1, {0, -contrast * fulleigengenes[[t, i, j]]}, {0, 1}]],
    {i, 1, xarrays}, {j, 1, xarrays}],
  framex = Table[{a - 0.5, tensornames[[t, 3, a]]}, {a, 1, xarrays}],
  g = Show[
    Graphics[
     RasterArray[
      Table[
       RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
       {i, xarrays, 1, -1}, {j, 1, xarrays}]]],
    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 - 1.2, c}, {0, 0}, {0, 1}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labelx, {b_, c_}, {0., -1.}] ->
     Text[labelx, {b, c + 1}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  g1[[t]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]
```

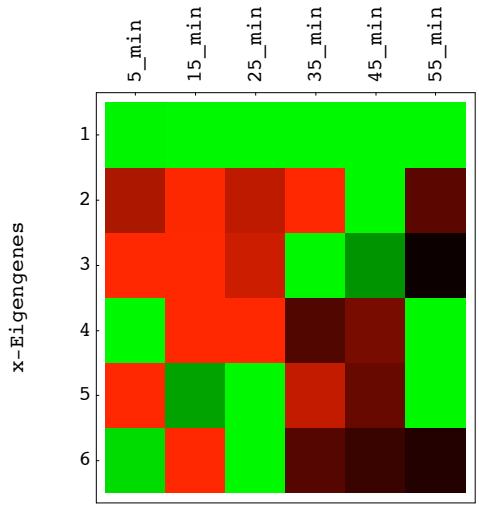
```
(* Create 2 D Red & Green Raster Display of x-Eigengenes After Missing Data Estimation *)

g4 = Table[0, {yarrays}];
contrast = 3.5;
framey = Table[{a + 1 - 0.5, xarrays - a}, {a, 0, xarrays - 1}];
labely = "x-Eigengenes";
labelx = ColumnForm[{"(d) x-Settings", " ", " "}, Center];
Do[{
  displaying = Table[
    If[contrast * eigengenes[[t, i, j]] > 0,
     If[contrast * eigengenes[[t, i, j]] < 1, {contrast * eigengenes[[t, i, j]], 0}, {1, 0}],
     If[contrast * eigengenes[[t, i, j]] > -1, {0, -contrast * eigengenes[[t, i, j]]}, {0, 1}]],
    {i, 1, xarrays}, {j, 1, xarrays}],
  framex = Table[{a - 0.5, tensornames[[t, 3, a]]}, {a, 1, xarrays}],
  g = Show[
    Graphics[
     RasterArray[
      Table[
       RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
       {i, xarrays, 1, -1}, {j, 1, xarrays}]]],
    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 - 1.2, c}, {0, 0}, {0, 1}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labelx, {b_, c_}, {0., -1.}] ->
     Text[labelx, {b, c + 1}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  g4[[t]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]
```

```
(* Create Graph Display of Selected x-Eigengenes Before Missing Data Estimation *)

g3 = Table[0, {yarrays}];
p = Table[0, {3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
framey = {-1, -0.5, 0, 0.5, 1};
labelx = "(c) x-Settings";
labely = ColumnForm[{" ", "Expression Level"}, Center];
Do[{
  framex = Table[{a - 1, tensornames[[t, 3, a]]}, {a, 1, xarrays}],
  Do[{
    coordinates = Table[{a - 1, fulleigengenes[[t, n, a]]}, {a, 1, xarrays}],
    points = Table[Point[coordinates[[a]]], {a, 1, xarrays}],
    line = Line[coordinates],
    g = Show[
      {Graphics[{color[[n]], PointSize[0.022], points}],
       Graphics[{color[[n]], Thickness[0.0044], line}]},
      Frame -> True,
      FrameLabel -> {None, labely, labelx, None},
      FrameTicks -> {None, framey, framex, None},
      PlotRange -> {-1.05, 1.05},
      GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
      DisplayFunction -> Identity],
    g = FullGraphics[g],
    g[[1, 2]] = g[[1, 2]] /.
      Text[labely, {b_, c_}, {1., 0.}] ->
       Text[labely, {b - 1.25, 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, -1}, {0, 1}],
    p[[n]] = Show[g,
      AspectRatio -> 1.05,
      PlotRange -> All,
      DisplayFunction -> Identity]},
   {n, 1, 3}],
  g3[[t]] = Show[{p[[3]], p[[2]], p[[1]]},
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]
Clear[fullentropy, fullfractions, fulleigengenes];
```

```
(* Create Graph Display of Selected x-Eigengenes After Missing Data Estimation *)

g6 = Table[0, {yarrays}];
p = Table[0, {3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
framey = {-1, -0.5, 0, 0.5, 1};
labelx = "(f) x-Settings";
labely = ColumnForm[{" ", "Expression Level"}, Center];

Do[{
  framex = Table[{a - 1, tensornames[[t, 3, a]]}, {a, 1, xarrays}],
  Do[{
    coordinates = Table[{a - 1, eigengenes[[t, n, a]]}, {a, 1, xarrays}],
    points = Table[Point[coordinates[[a]]], {a, 1, xarrays}],
    line = Line[coordinates],
    g = Show[
      {Graphics[{color[[n]], PointSize[0.022], points}],
       Graphics[{color[[n]], Thickness[0.0044], line}]},
      Frame -> True,
      FrameLabel -> {None, labely, labelx, None},
      FrameTicks -> {None, framey, framex, None},
      PlotRange -> {-1.05, 1.05},
      GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
      DisplayFunction -> Identity],
    g = FullGraphics[g],
    g[[1, 2]] = g[[1, 2]] /.
      Text[labely, {b_, c_}, {1., 0.}] ->
       Text[labely, {b - 1.25, 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, -1}, {0, 1}],
    p[[n]] = Show[g,
      AspectRatio -> 1.05,
      PlotRange -> All,
      DisplayFunction -> Identity]},
   {n, 1, 3}],
  g6[[t]] = Show[{p[[3]], p[[2]], p[[1]]},
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {t, 1, yarrays}]
Clear[entropy, fractions, eigengenes];

(* Display x-Eigengenes, Fractions and Selected x-Eigengenes *)

Do[
 Show[GraphicsArray[{
    {GraphicsArray[{g1[[t]], g2[[t]], g3[[t]]}, GraphicsSpacing -> -0.15]},
    {GraphicsArray[{g4[[t]], g5[[t]], g6[[t]]}, GraphicsSpacing -> -0.15]}}],
  GraphicsSpacing -> 0,
  PlotLabel -> ColumnForm[{" ", tensornames[[t, 2, 1]],
     StringJoin[tensornames[[t, 3, 1]], "-", tensornames[[t, 3, 6]]]}, Center],
  PlotRange -> {{0.032, 0.968}, {0, 0.775}}],
 {t, 1, yarrays}]
Clear[g1, g2, g3, g4, g5, g6];
```

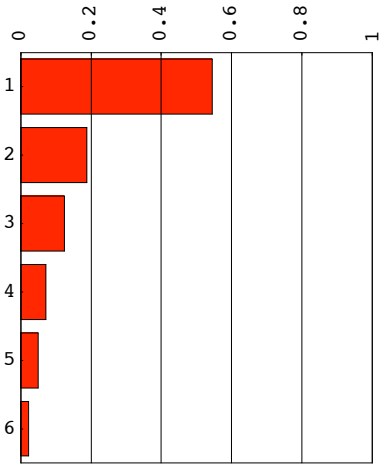
1_Cdc6+
5_min-55_min

(a) x-Settings

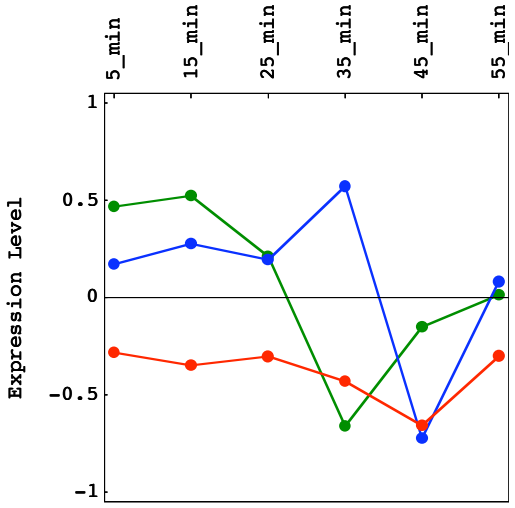


(b) Eigenexpression Fraction

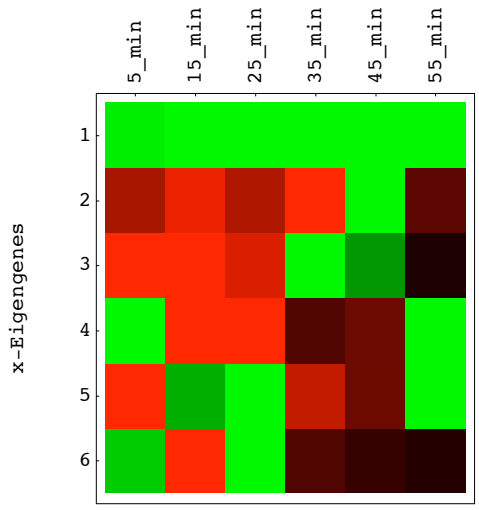
$d_1' = 0.74$



(c) x-Settings

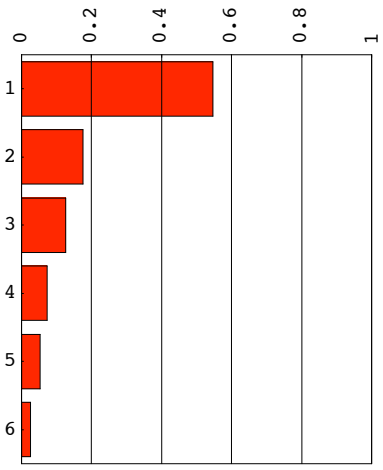


(d) x-Settings

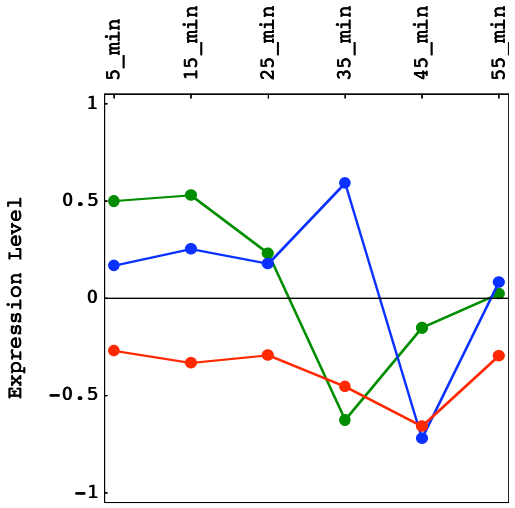


(e) Eigenexpression Fraction

$d_1 = 0.75$

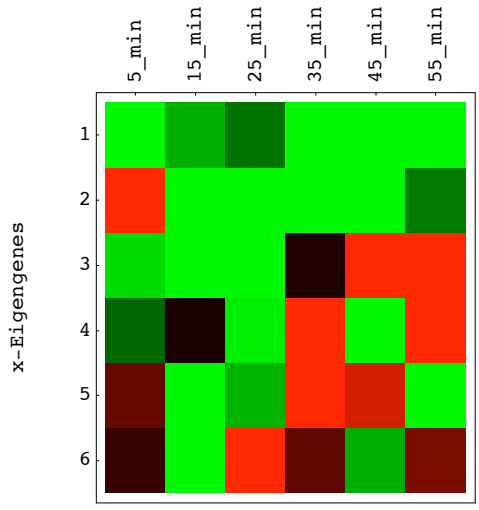


(f) x-Settings



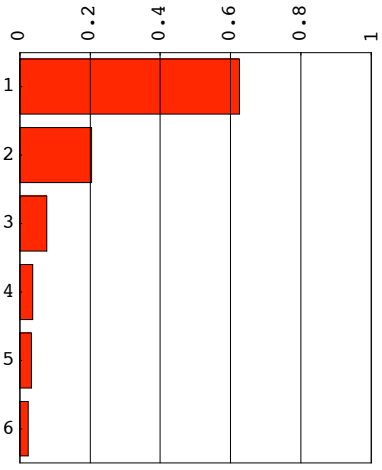
1_Cdc6-
5_min-55_min

(a) x-Settings

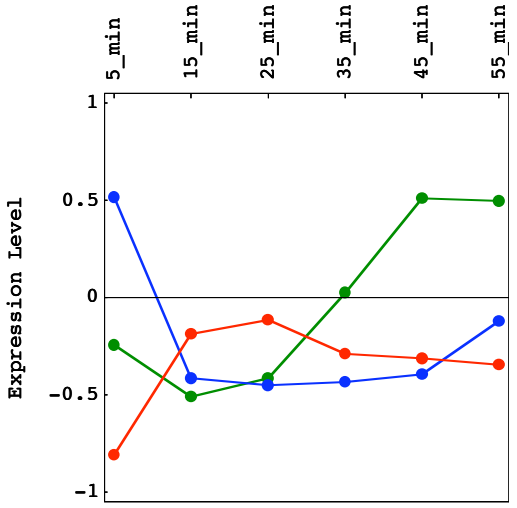


(b) Eigenexpression Fraction

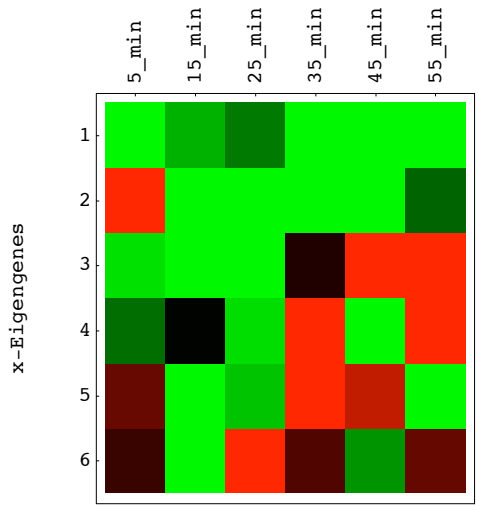
$d_2' = 0.64$



(c) x-Settings

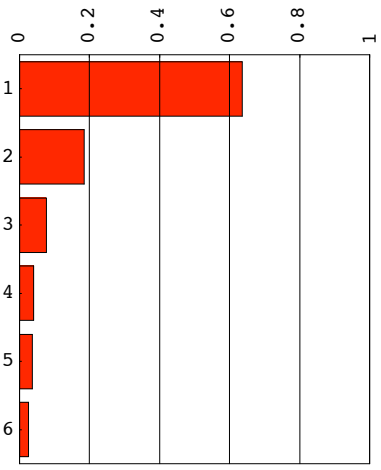


(d) x-Settings

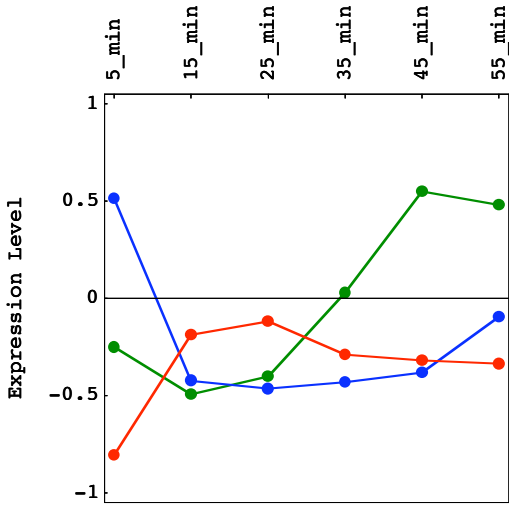


(e) Eigenexpression Fraction

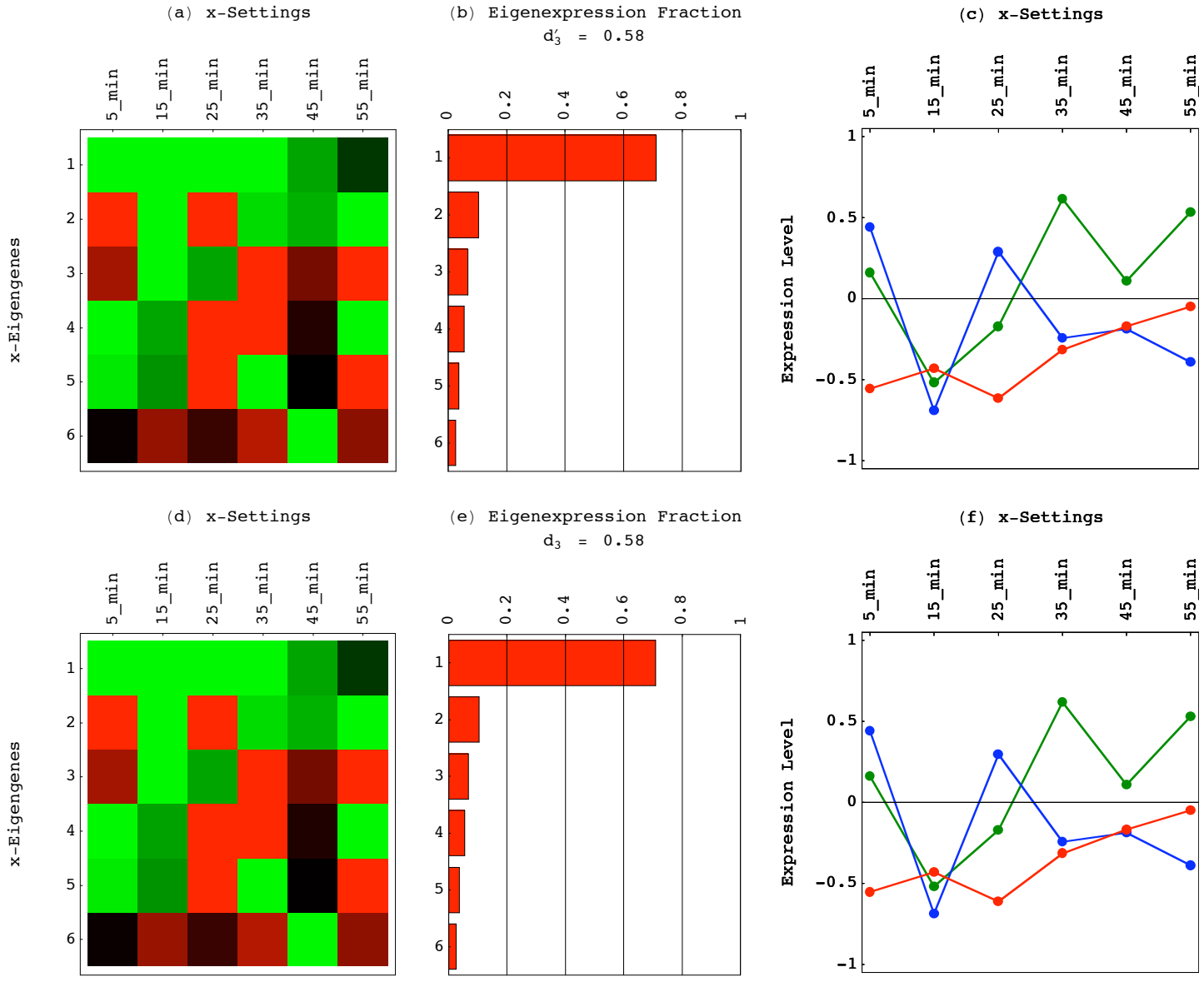
$d_2 = 0.64$



(f) x-Settings

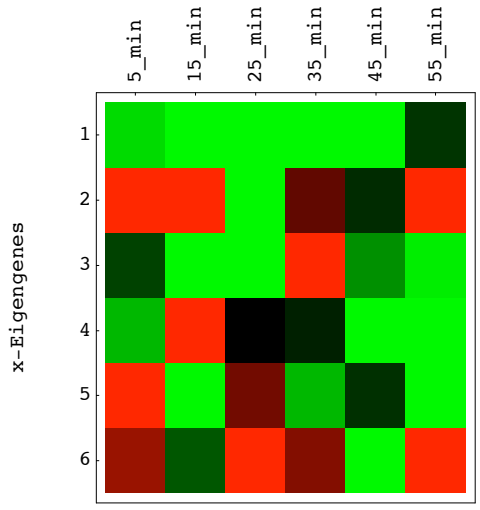


2_Cdc6+
5_min-55_min



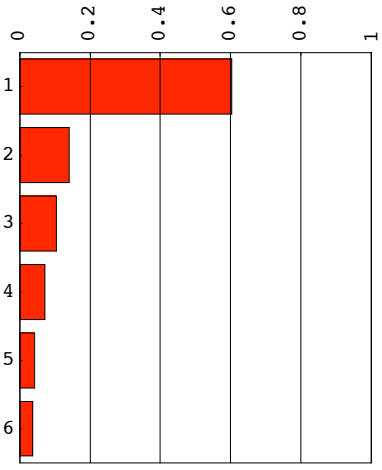
2_Cdc6-
5_min-55_min

(a) x-Settings

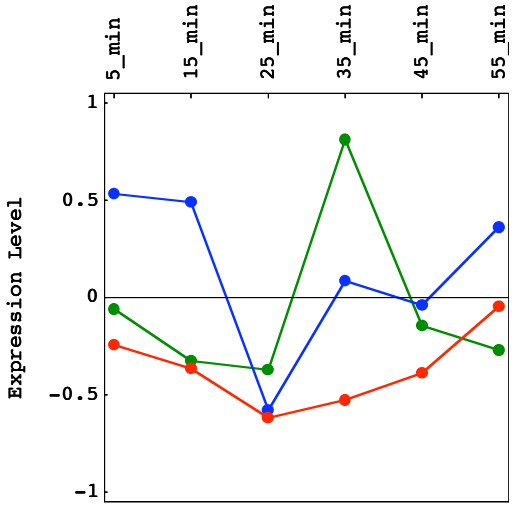


(b) Eigenexpression Fraction

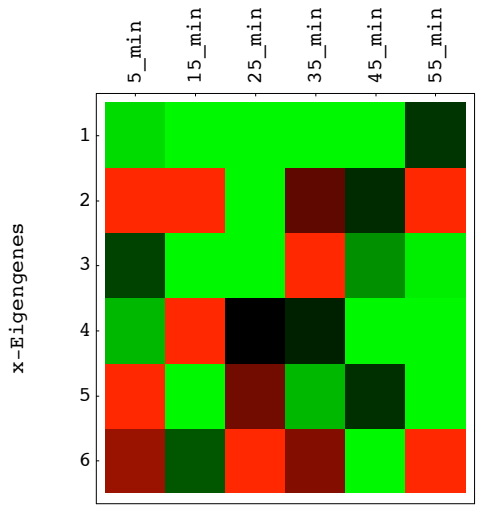
$d_4' = 0.7$



(c) x-Settings

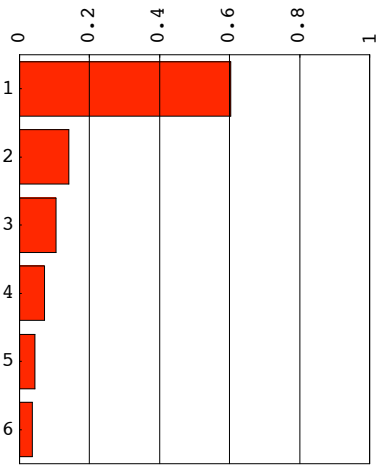


(d) x-Settings

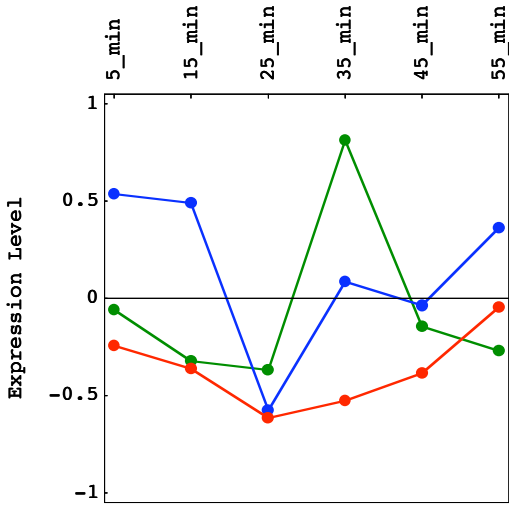


(e) Eigenexpression Fraction

$d_4 = 0.7$

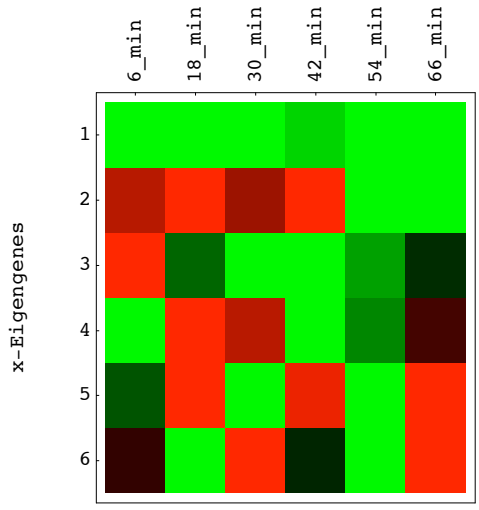


(f) x-Settings



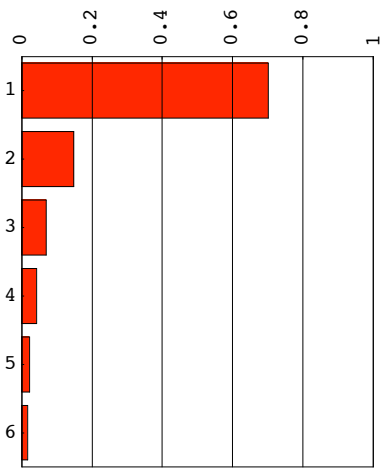
3_Cdc45+
6_min-66_min

(a) x-Settings

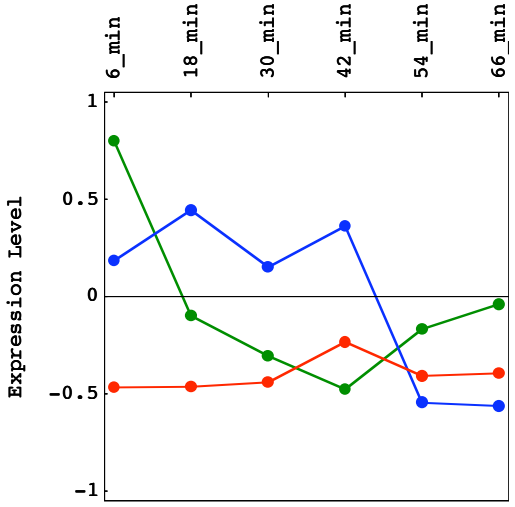


(b) Eigenexpression Fraction

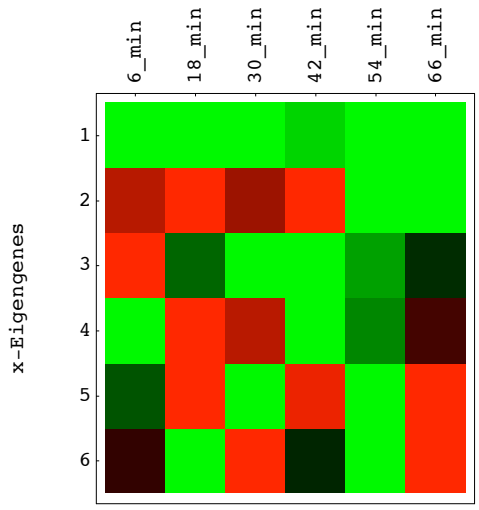
$d'_5 = 0.56$



(c) x-Settings

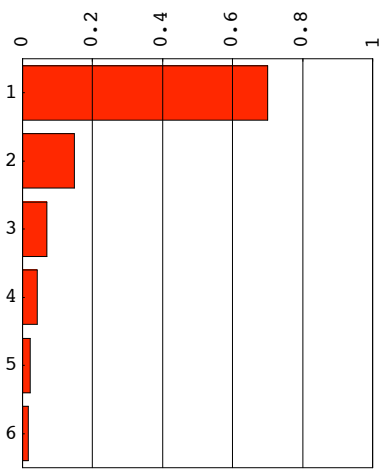


(d) x-Settings

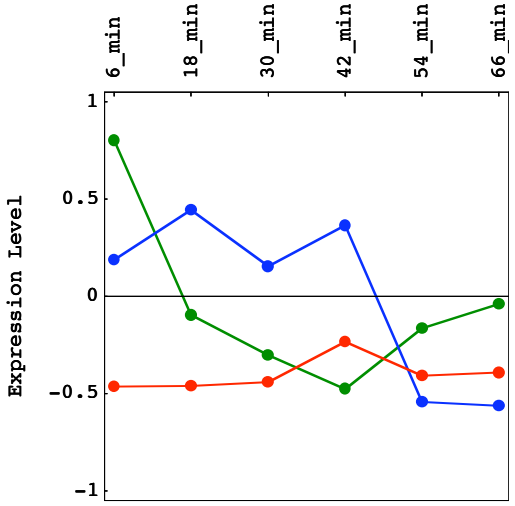


(e) Eigenexpression Fraction

$d_5 = 0.56$

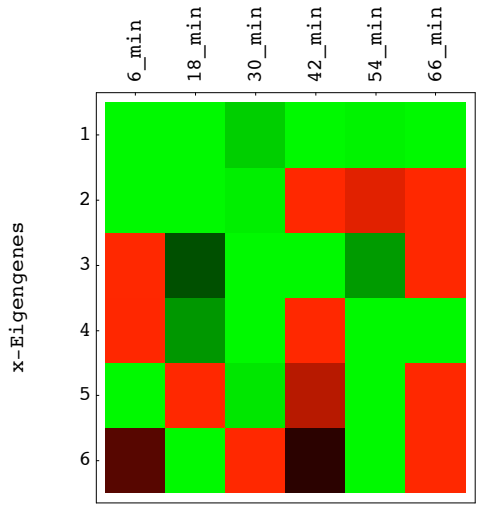


(f) x-Settings



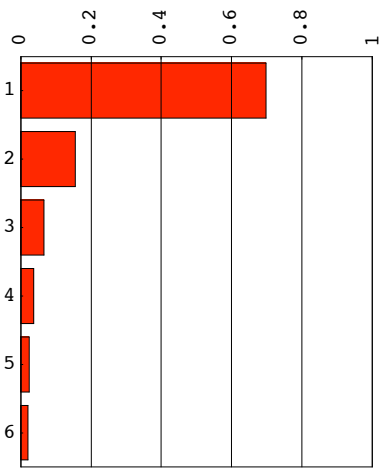
3_Cdc45-
6_min-66_min

(a) x-Settings

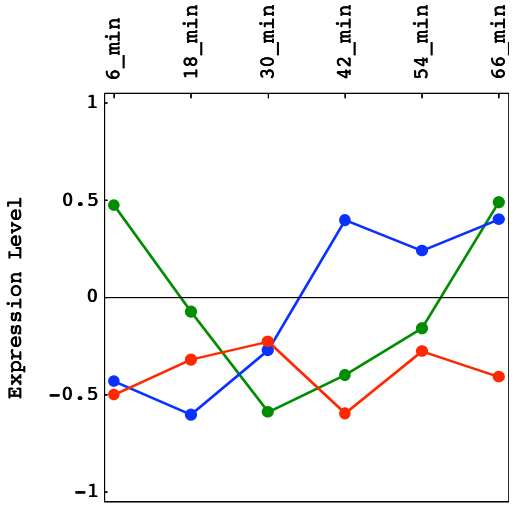


(b) Eigenexpression Fraction

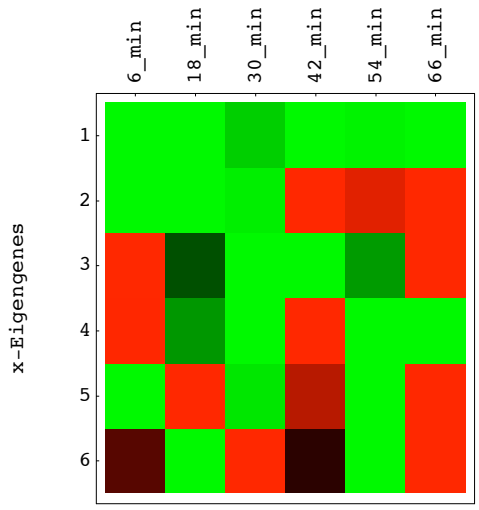
$d'_6 = 0.56$



(c) x-Settings

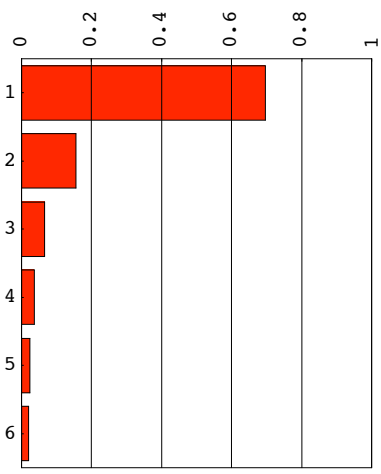


(d) x-Settings

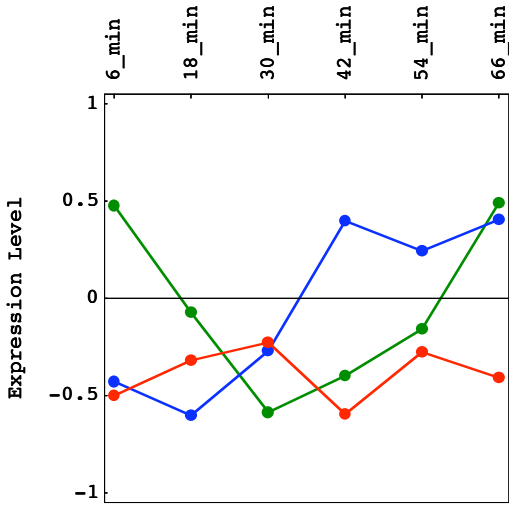


(e) Eigenexpression Fraction

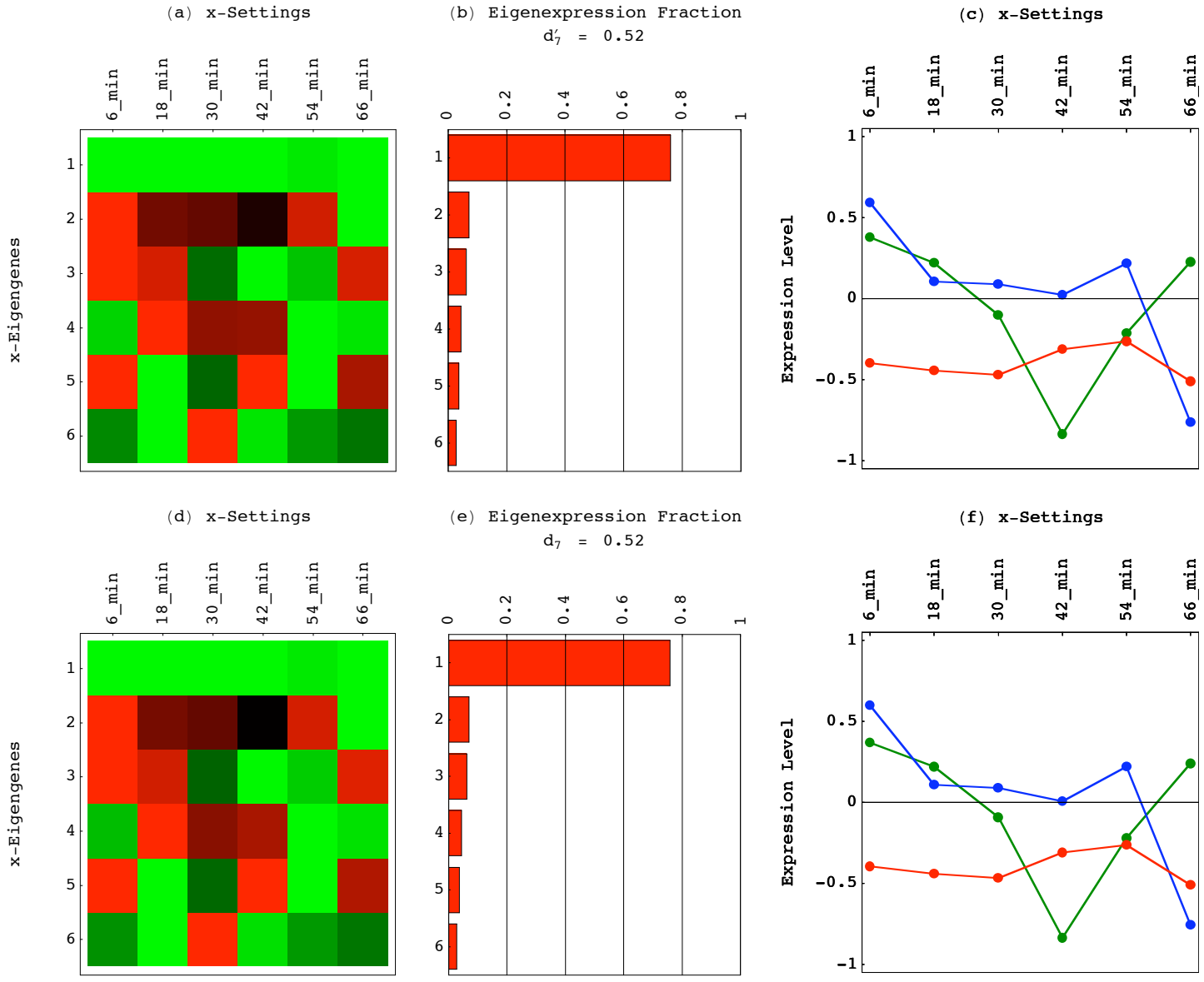
$d_6 = 0.56$



(f) x-Settings

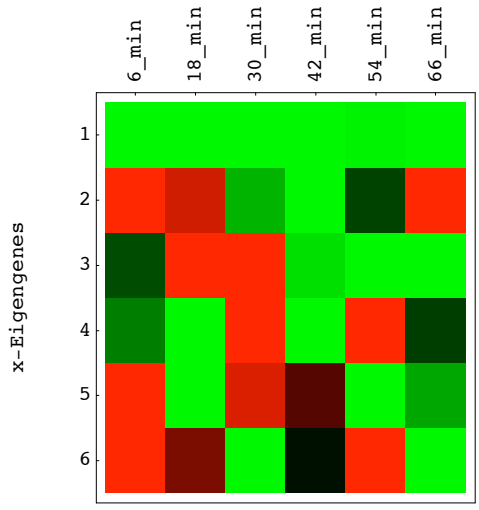


4_Cdc45+
6_min-66_min



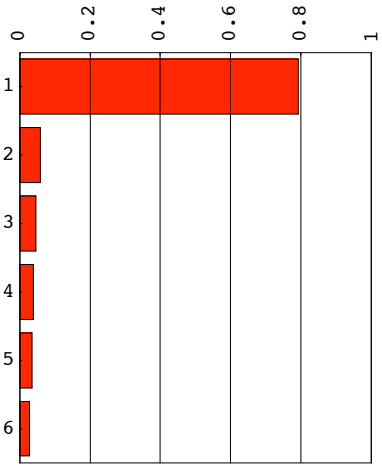
4_Cdc45-
6_min-66_min

(a) x-Settings

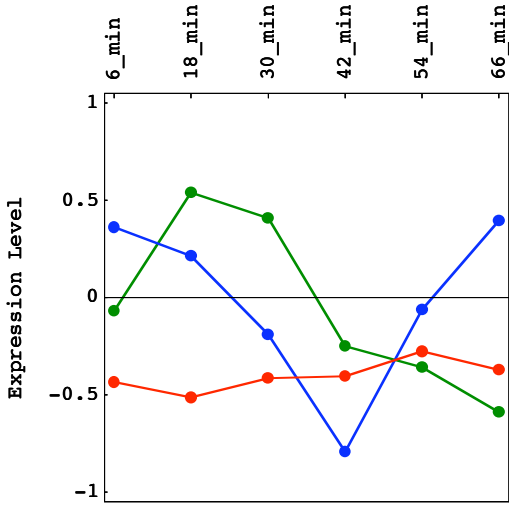


(b) Eigenexpression Fraction

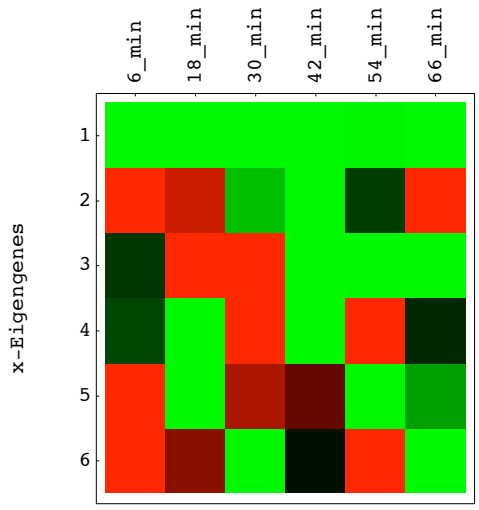
$d'_8 = 0.47$



(c) x-Settings

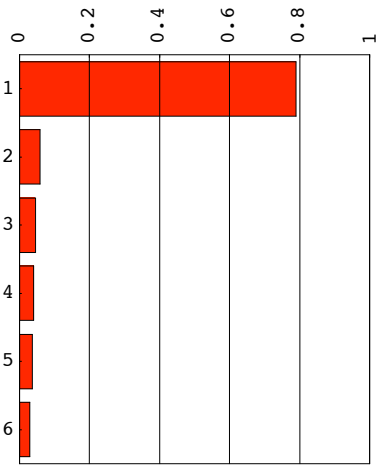


(d) x-Settings

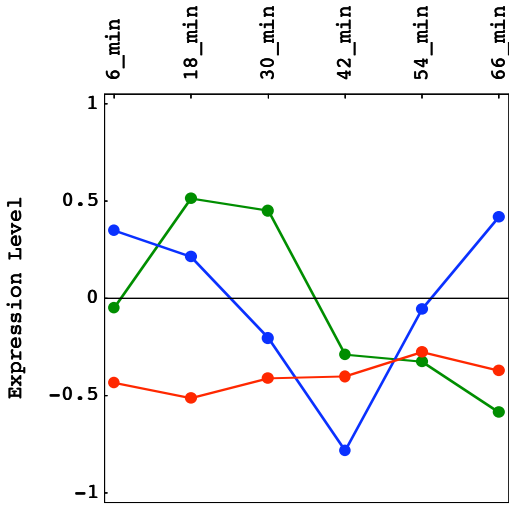


(e) Eigenexpression Fraction

$d_8 = 0.47$

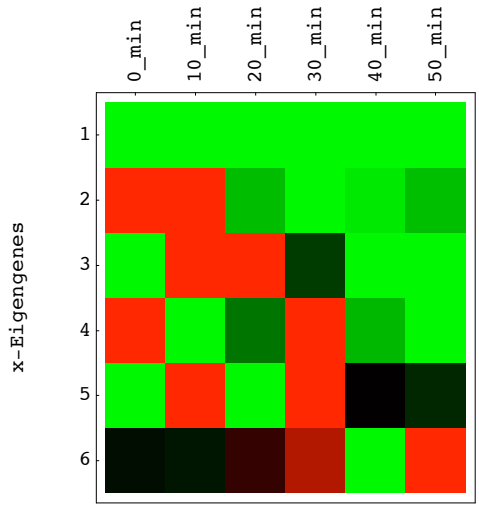


(f) x-Settings



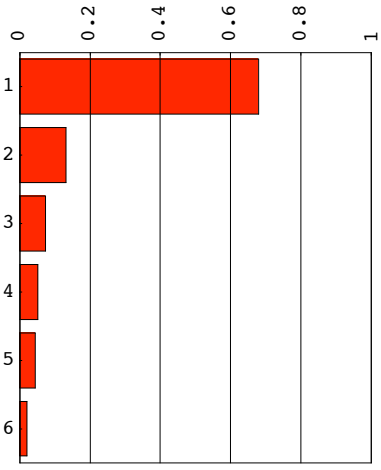
1_Cdc6+
0_min-50_min

(a) x-Settings

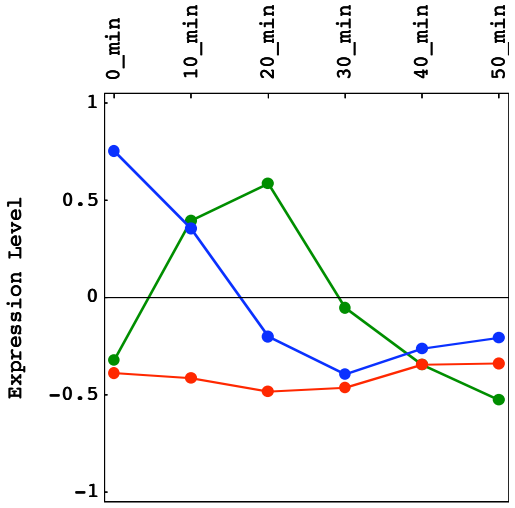


(b) Eigenexpression Fraction

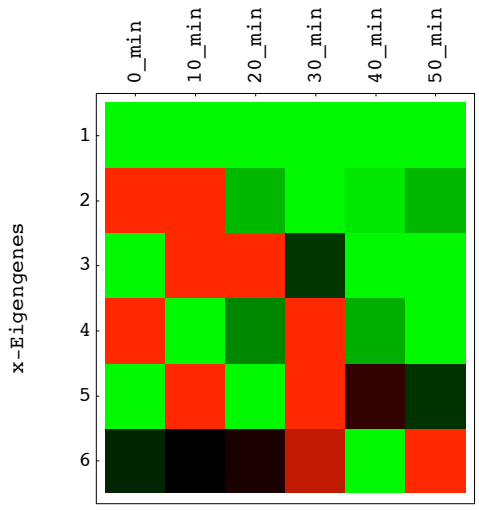
$d'_9 = 0.61$



(c) x-Settings

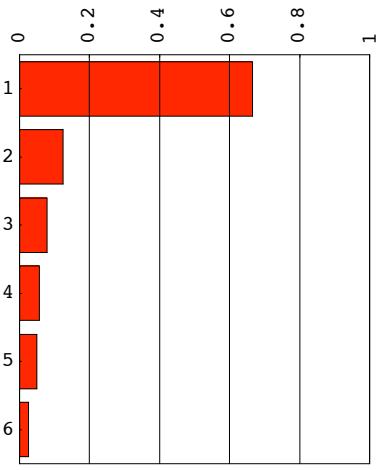


(d) x-Settings

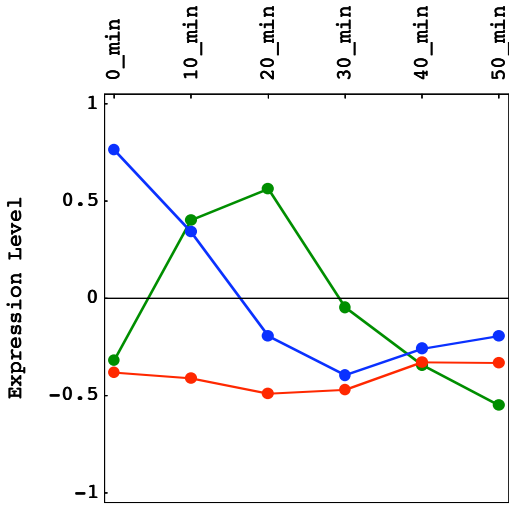


(e) Eigenexpression Fraction

$d_9 = 0.63$

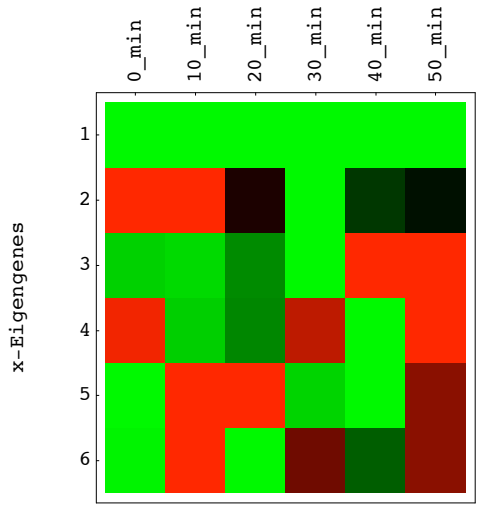


(f) x-Settings



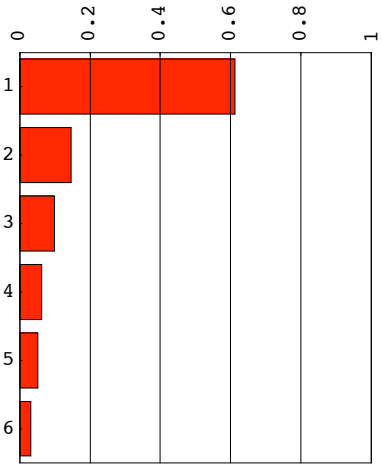
1_Cdc6-
0_min-50_min

(a) x-Settings

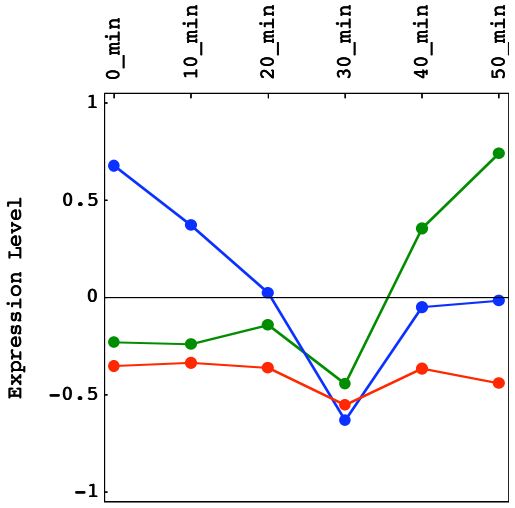


(b) Eigenexpression Fraction

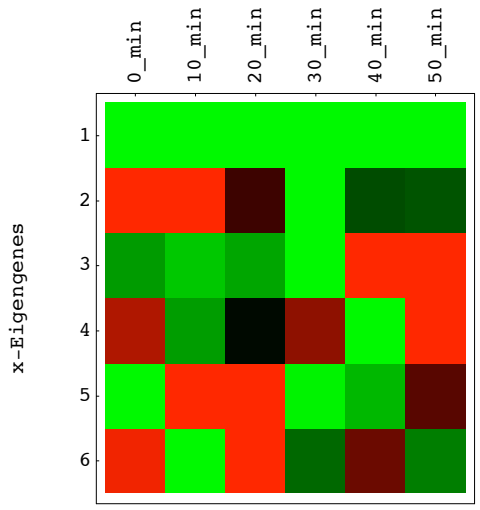
$d'_{10} = 0.69$



(c) x-Settings

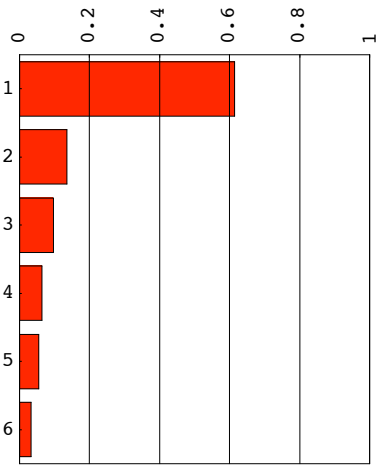


(d) x-Settings

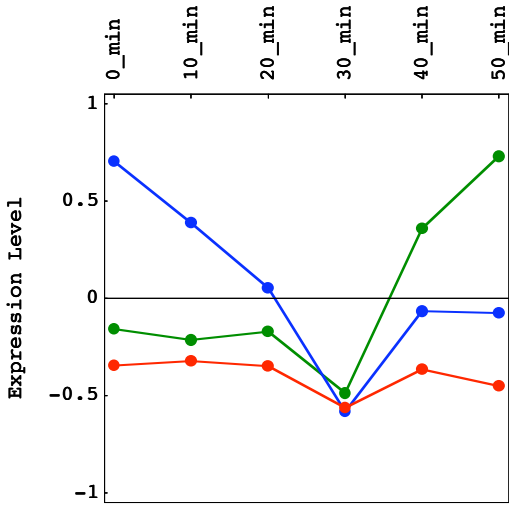


(e) Eigenexpression Fraction

$d_{10} = 0.7$

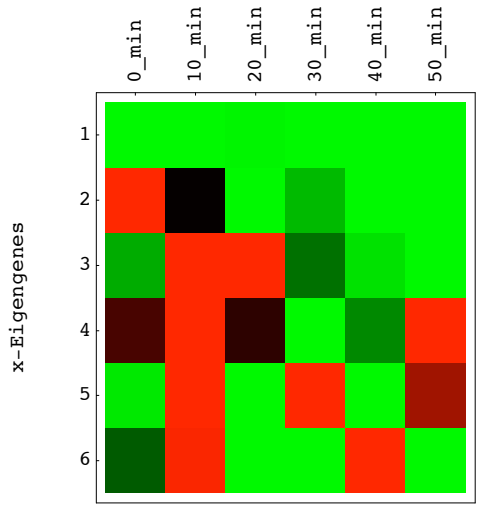


(f) x-Settings

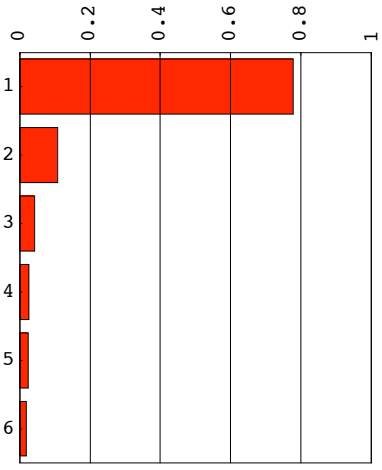


2_Cdc6+
0_min-50_min

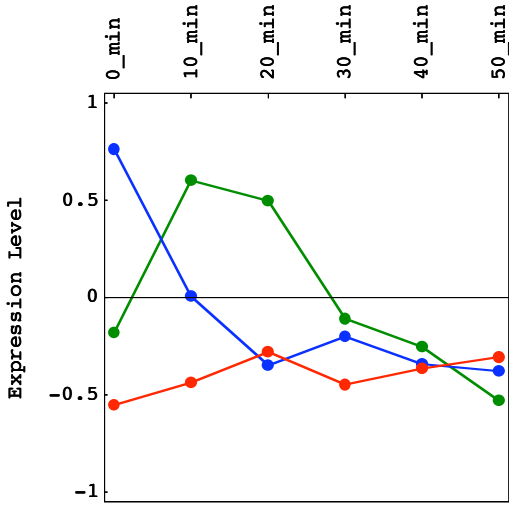
(a) x-Settings



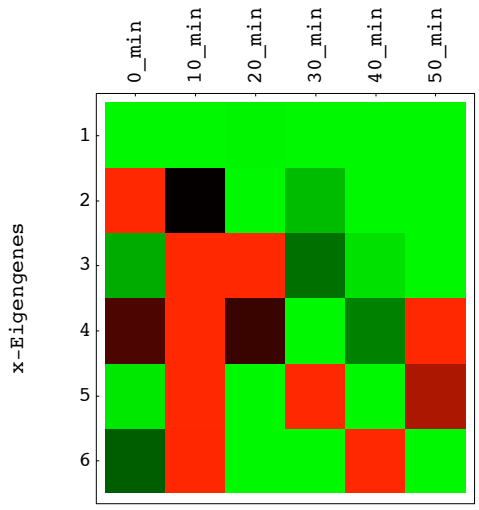
(b) Eigenexpression Fraction
d'_{11} = 0.47



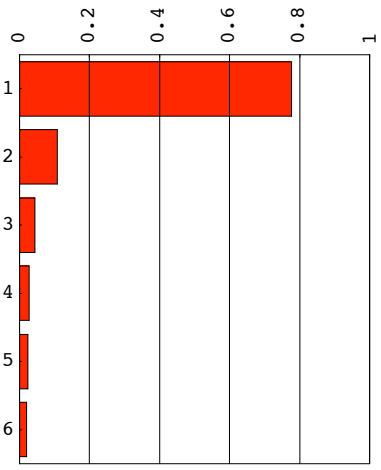
(c) x-Settings



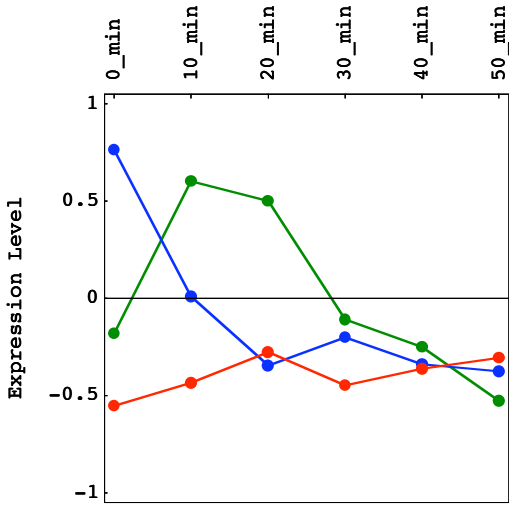
(d) x-Settings



(e) Eigenexpression Fraction
d_{11} = 0.47

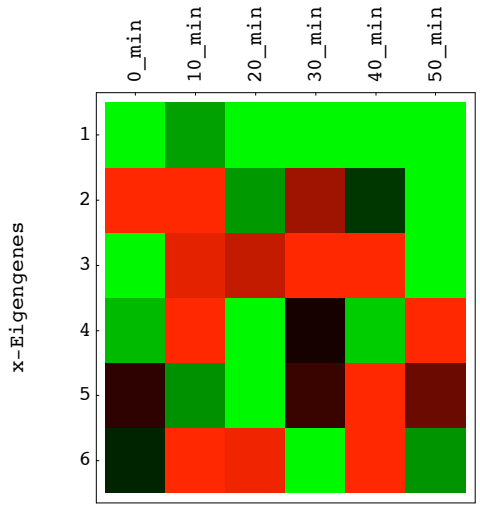


(f) x-Settings



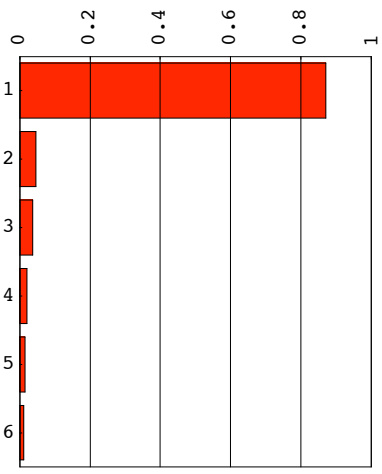
2_Cdc6-
0_min-50_min

(a) x-Settings

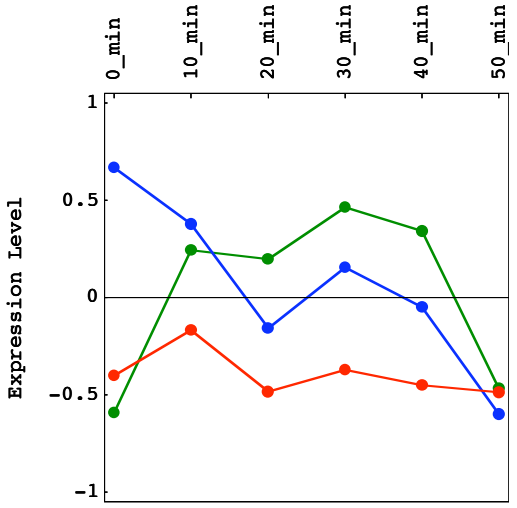


(b) Eigenexpression Fraction

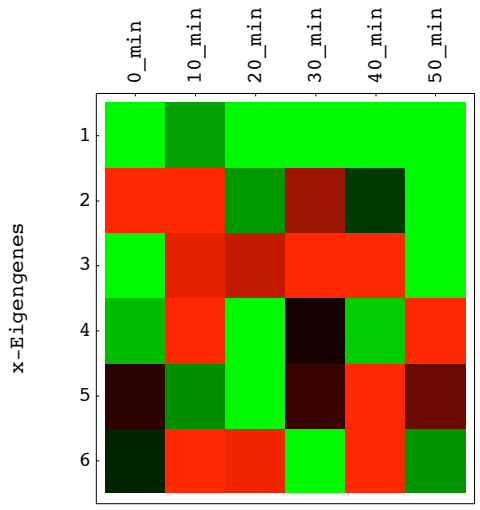
d'_{12} = 0.32



(c) x-Settings

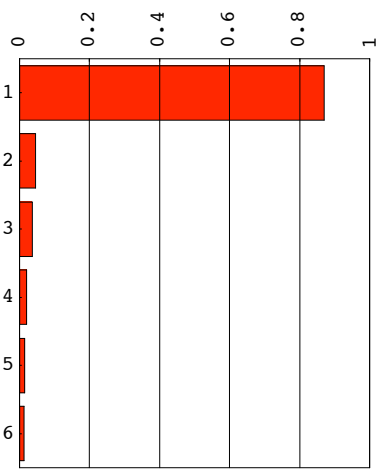


(d) x-Settings

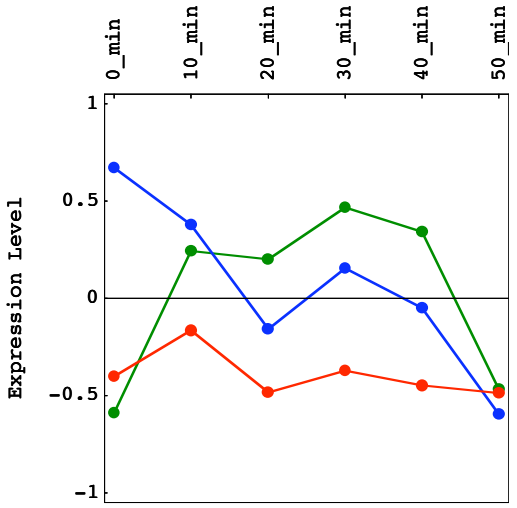


(e) Eigenexpression Fraction

d_{12} = 0.32

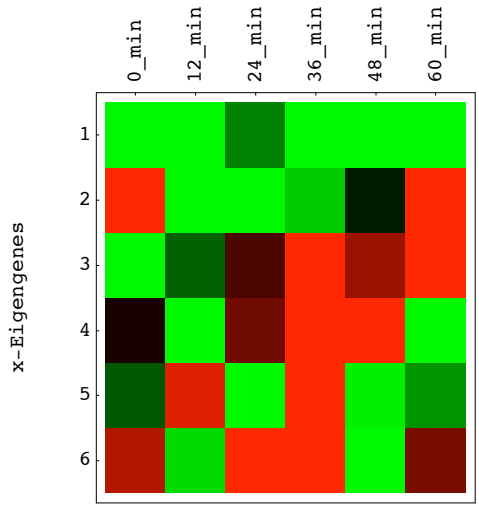


(f) x-Settings

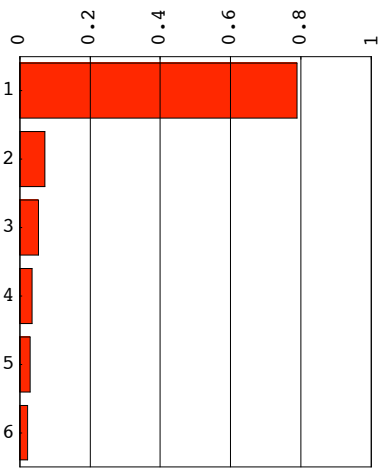


3_Cdc45+
0_min-60_min

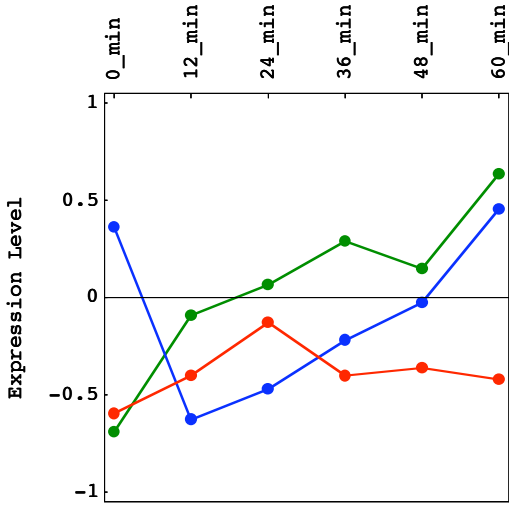
(a) x-Settings



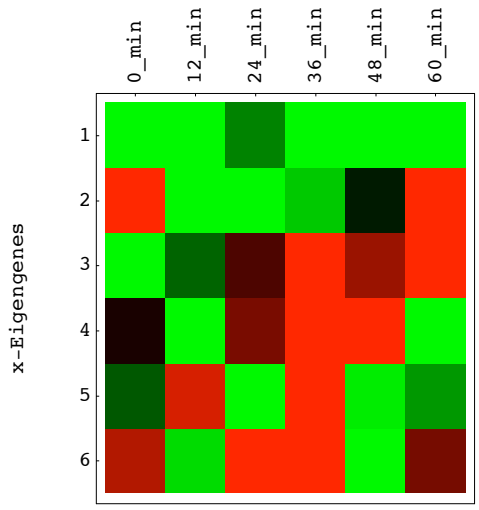
(b) Eigenexpression Fraction
d'_{13} = 0.47



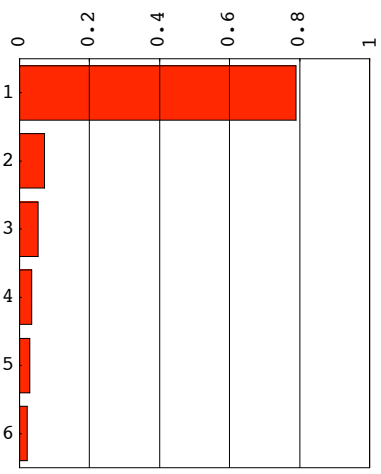
(c) x-Settings



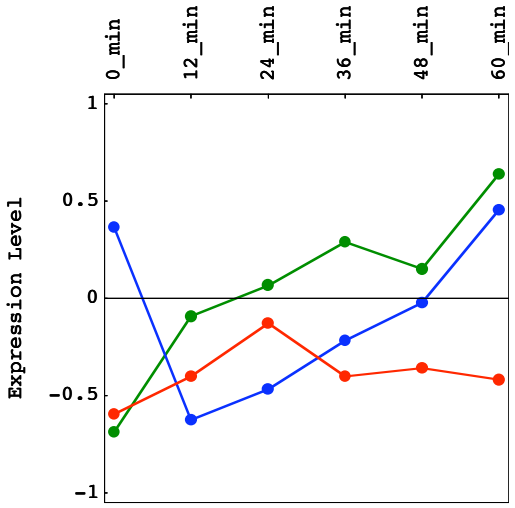
(d) x-Settings



(e) Eigenexpression Fraction
d_{13} = 0.47

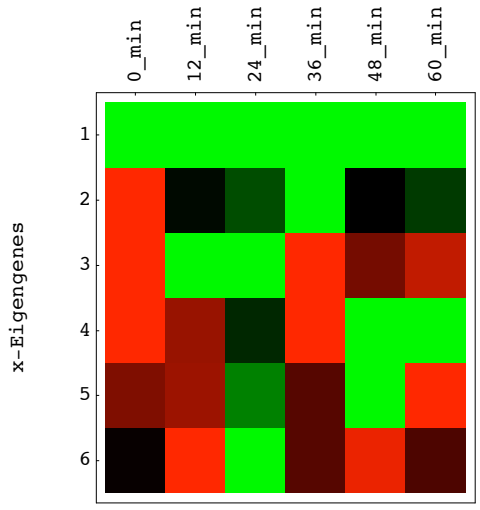


(f) x-Settings

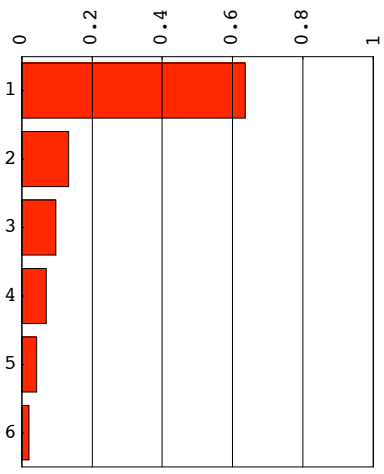


3_Cdc45-
0_min-60_min

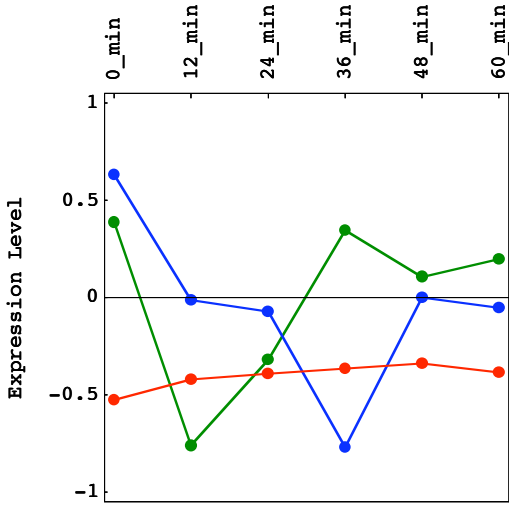
(a) x-Settings



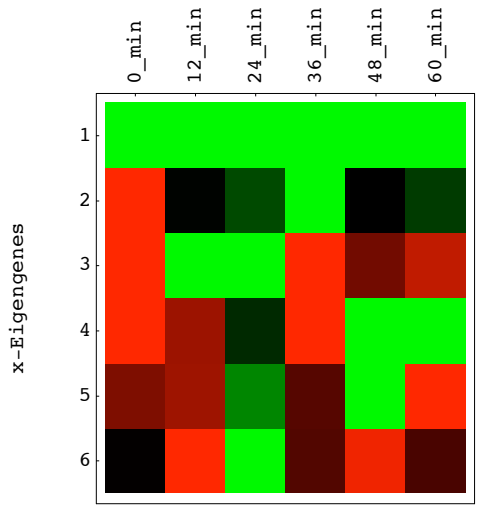
(b) Eigenexpression Fraction
d'_{14} = 0.66



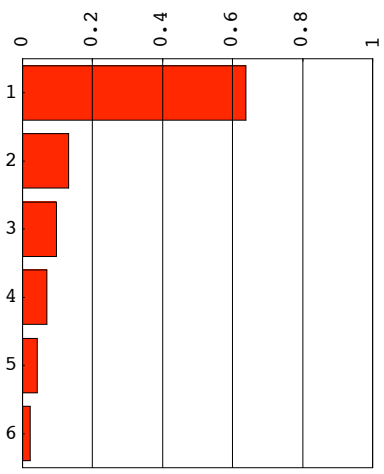
(c) x-Settings



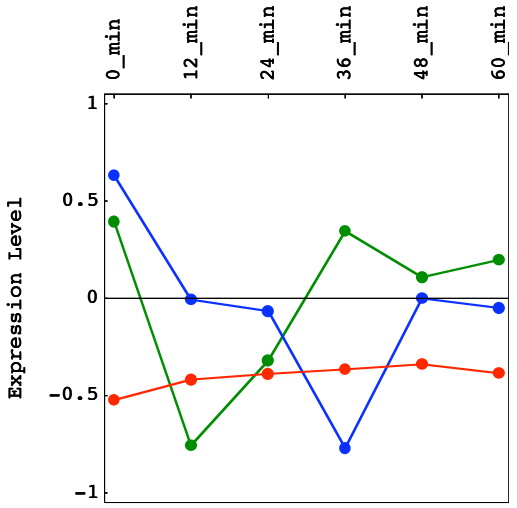
(d) x-Settings



(e) Eigenexpression Fraction
d_{14} = 0.66

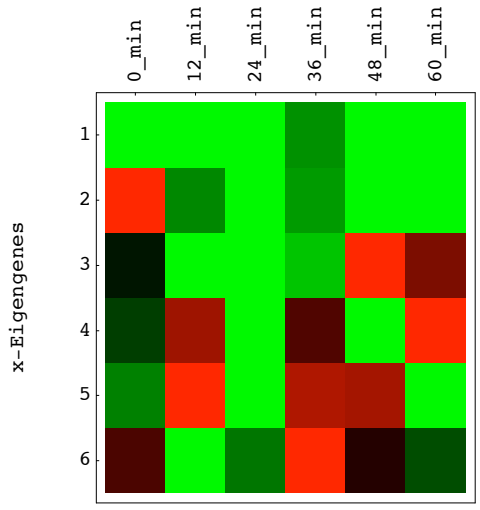


(f) x-Settings

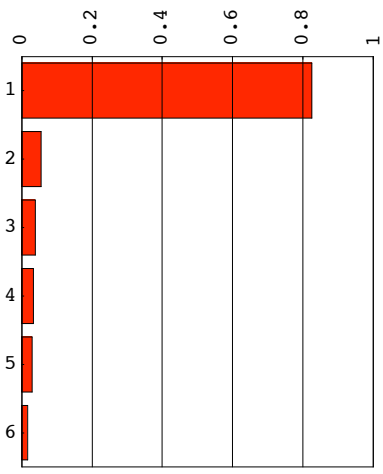


4_Cdc45+
0_min-60_min

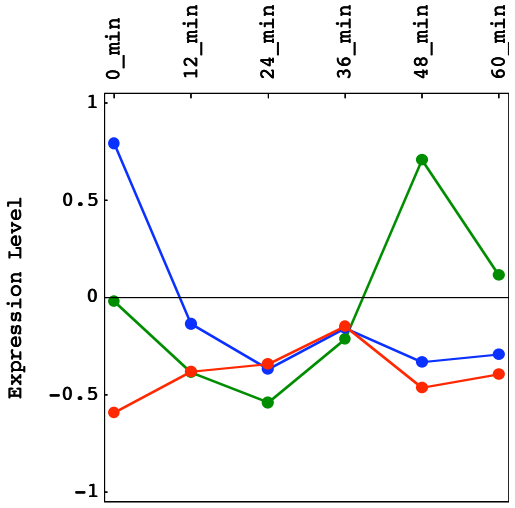
(a) x-Settings



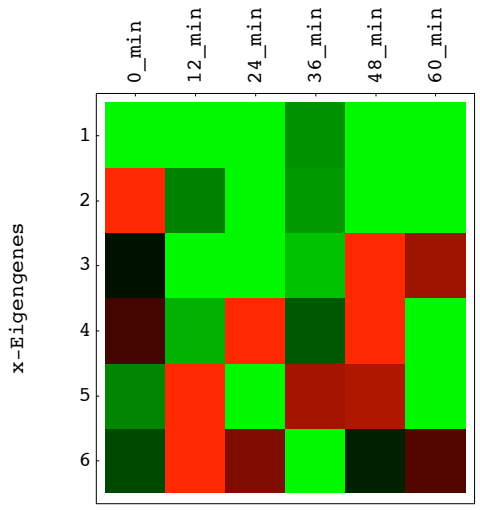
(b) Eigenexpression Fraction
d'_{15} = 0.41



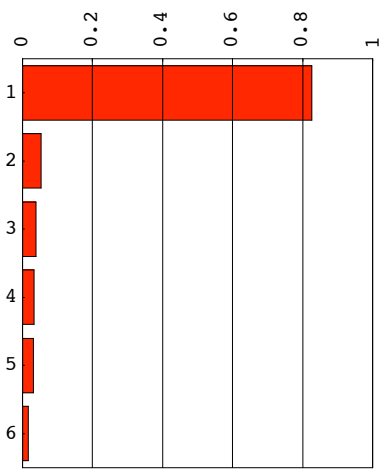
(c) x-Settings



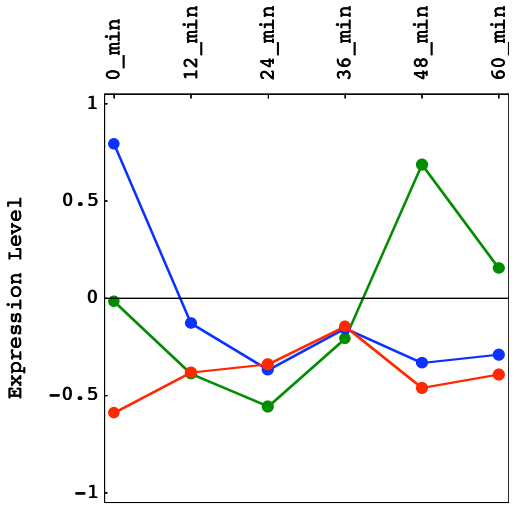
(d) x-Settings



(e) Eigenexpression Fraction
d_{15} = 0.41

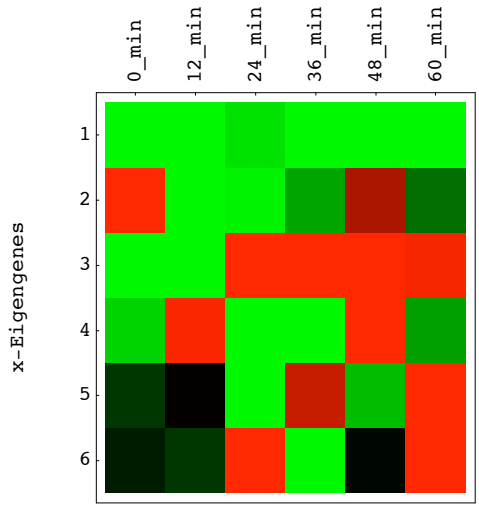


(f) x-Settings



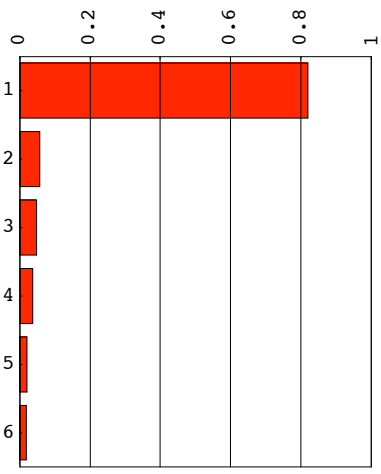
4_Cdc45-
0_min-60_min

(a) x-Settings

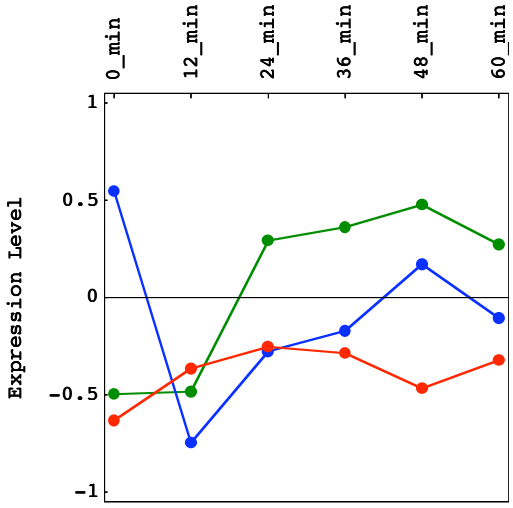


(b) Eigenexpression Fraction

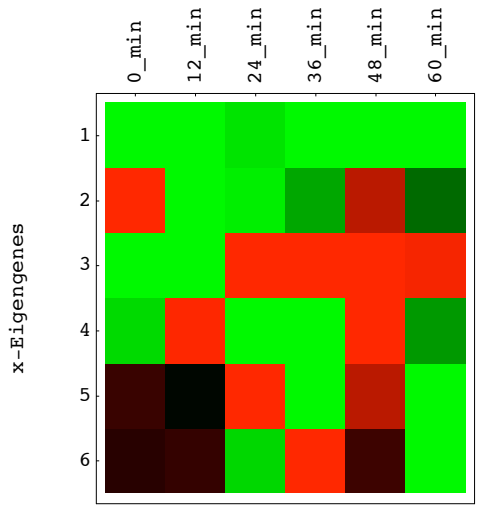
$d'_{16} = 0.42$



(c) x-Settings

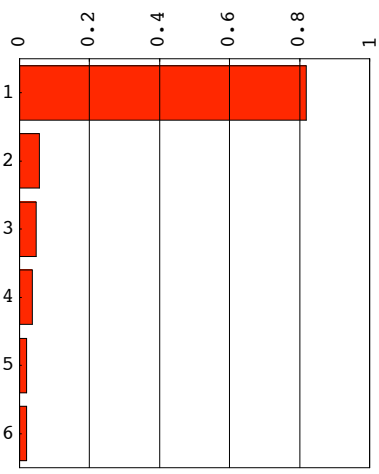


(d) x-Settings

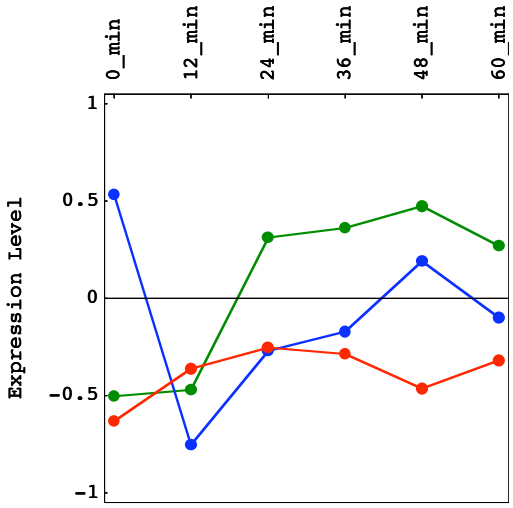


(e) Eigenexpression Fraction

$d_{16} = 0.42$



(f) x-Settings



```
(* Structure the Data Tensor *)
```

```
(* Combine Data from Two Different Hybridization Batches of Each Time Course *)

tensor = {
   AppendRows[tensor[[1]], tensor[[9]]],
   AppendRows[tensor[[2]], tensor[[10]]],
   AppendRows[tensor[[3]], tensor[[11]]],
   AppendRows[tensor[[4]], tensor[[12]]],
   AppendRows[tensor[[5]], tensor[[13]]],
   AppendRows[tensor[[6]], tensor[[14]]],
   AppendRows[tensor[[7]], tensor[[15]]],
   AppendRows[tensor[[8]], tensor[[16]]]};
{yarrays, zgenes, xarrays} = Dimensions[tensor]
tensornames = {
   AppendRows[tensornames[[1]], tensornames[[9]]],
   AppendRows[tensornames[[2]], tensornames[[10]]],
   AppendRows[tensornames[[3]], tensornames[[11]]],
   AppendRows[tensornames[[4]], tensornames[[12]]],
   AppendRows[tensornames[[5]], tensornames[[13]]],
   AppendRows[tensornames[[6]], tensornames[[14]]],
   AppendRows[tensornames[[7]], tensornames[[15]]],
   AppendRows[tensornames[[8]], tensornames[[16]]]};
```

```
{8, 4270, 12}
```

```
(* Convert Data from Log2[Red/Green] to Log2[Signal/Reference] *)

tensor[[3]] = -tensor[[3]];
tensor[[4]] = -tensor[[4]];
tensor[[7]] = -tensor[[7]];
tensor[[8]] = -tensor[[8]];

(* Normalize Each Array to Zero Average and Unit Variance *)

product1 = 1;
tensor = Transpose[tensor, {1, 3, 2}];
Do[{
  tensor[[a, b]] = tensor[[a, b]] - Sum[tensor[[a, b, c]], {c, 1, zgenes}] / zgenes,
  product1 = product1 * Sqrt[Dot[tensor[[a, b]], tensor[[a, b]]]],
  tensor[[a, b]] = tensor[[a, b]] / Sqrt[Dot[tensor[[a, b]], tensor[[a, b]]]]
 }, {a, 1, yarrays}, {b, 1, xarrays}]
tensor = Transpose[tensor, {1, 3, 2}];
product1 = product1^(1 / xarrays / yarrays)
```

```
57.3395
```

```
(* Detect Experimental Artifacts by Using HOSVD *)
```

```
(* Compute HOSVD x-Eigengenes *)
```

```
mode2 = Flatten[tensor, 1];
xarraynames = {2, 4, 6, 8, 10, 12, 1, 3, 5, 7, 9, 11};
{eigenarrays, eigenexpressions, xeigengenes} = SingularValues[mode2];
eigenarrays = Transpose[eigenarrays];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, xarrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, xarrays}] / Log[xarrays]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.73
```

```
(* Create Fractions Bar Chart Display *)
```

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

```
0.492475
```

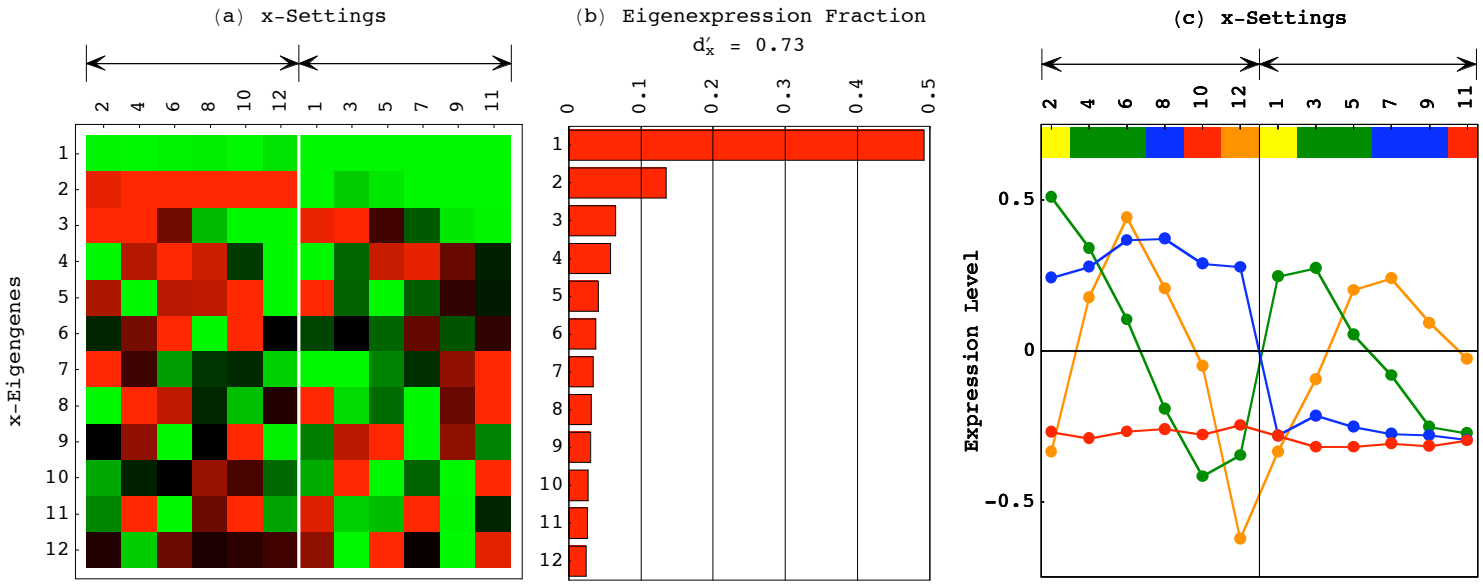
```
gridx = Table[a, {a, 0, 0.5, 0.1}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, xarrays - a}, {a, 0, xarrays - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d'x = ", ToString[entropy]], " "},
   Center];
g = BarChart[
   Table[fractions[[xarrays - a]], {a, 0, xarrays - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.5 * 1.0001}, {0.5, xarrays + 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 + 0.9}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[{g},
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

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

```
arrows[x1_, x2_, x3_, y_, h_] :=
  Graphics[{
    Arrow[{x1, y}, {x2, y}, HeadCenter -> 0.5, HeadLength → length, HeadWidth -> 0.75],
    Arrow[{x2, y}, {x1, y}, HeadCenter -> 0.5, HeadLength → length, HeadWidth -> 0.75],
    Arrow[{x2, y}, {x3, y}, HeadCenter -> 0.5, HeadLength → length, HeadWidth -> 0.75],
    Arrow[{x3, y}, {x2, y}, HeadCenter -> 0.5, HeadLength → length, HeadWidth -> 0.75],
    Line[{{x1, y - h}, {x1, y + h}}],
    Line[{{x2, y - h}, {x2, y + h}}],
    Line[{{x3, y - h}, {x3, y + h}}]}];
```

```
length = 0.035;
contrast = 3.6;
displaying = Table[
   If[contrast * xeigengenes[[i, j]] > 0,
    If[contrast * xeigengenes[[i, j]] < 1, {contrast * xeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * xeigengenes[[i, j]] > -1, {0, -contrast * xeigengenes[[i, j]]}, {0, 1}]],
   {i, 1, xarrays}, {j, 1, xarrays}];
framex = Table[{a - 0.5, xarraynames[[a]]}, {a, 1, xarrays}];
framey = Table[{a + 1 - 0.5, xarrays - a}, {a, 0, xarrays - 1}];
labely = "x-Eigengenes";
labelx = ColumnForm[{"(a) x-Settings", " ", " "}, Center];
g = Show[
   Graphics[
    RasterArray[
     Table[
      RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
      {i, xarrays, 1, -1}, {j, 1, xarrays}]]],
   AspectRatio -> 1,
   Frame -> True,
   FrameTicks -> {None, framey, framex, None},
   FrameLabel -> {None, labely, labelx, None},
   GridLines -> {{{6, {RGBColor[1, 1, 1], Thickness[0.0066]}}}, None},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {b - 1.5, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 0.9}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[{g, arrows[0, 6, 12, 14, 0.5]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Create Selected x-Eigengenes Graph Display *)

cycle[x_, y_, z_] := {
   Graphics[{RGBColor[1, 1, 0], Rectangle[{x - 0.25, y}, {x + 0.5, z}]}],
   Graphics[{RGBColor[0, 0.5, 0], Rectangle[{x + 0.5, y}, {x + 2.5, z}]}],
   Graphics[{RGBColor[0, 0, 1], Rectangle[{x + 2.5, y}, {x + 3.5, z}]}],
   Graphics[{RGBColor[1, 0, 0], Rectangle[{x + 3.5, y}, {x + 4.5, z}]}],
   Graphics[{RGBColor[1, 0.5, 0], Rectangle[{x + 4.5, y}, {x + 5.5, z}]}],
   Graphics[{RGBColor[1, 1, 0], Rectangle[{x + 5.5, y}, {x + 6.5, z}]}],
   Graphics[{RGBColor[0, 0.5, 0], Rectangle[{x + 6.5, y}, {x + 8.5, z}]}],
   Graphics[{RGBColor[0, 0, 1], Rectangle[{x + 8.5, y}, {x + 10.5, z}]}],
   Graphics[{RGBColor[1, 0, 0], Rectangle[{x + 10.5, y}, {x + 11.25, z}]}]};

p = Table[0, {n, 1, 4}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0], RGBColor[1, 0.5, 0]};
labelx = ColumnForm[{"(c) x-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, xarraynames[[a]]}, {a, 1, xarrays}];
framey = {-0.5, 0, 0.5};
```

```
Do[{
  coordinates = Table[{a - 1, xeigengenes[[n, a]]}, {a, 1, xarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, xarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.75, 0.75},
    GridLines -> {{{5.5, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
    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 + 0.28}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 4}]
g3 = Show[{
    cycle[0, 0.64, 0.74],
    arrows[-0.25, 5.5, 11.25, 0.95, 0.05],
    p[[4]], p[[3]], p[[2]], p[[1]]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Display x-Eigengenes, Fractions and Selected x-Eigengenes *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.16];
```



```
(* Compute HOSVD y-Eigengenes *)

mode3 = Flatten[Transpose[tensor, {3, 1, 2}], 1];
yarraynames = Flatten[Transpose[tensornames, {3, 1, 2}], 1][[xarrays + 1]];
{eigenarrays, eigenexpressions, yeigengenes} = SingularValues[mode3];
yeigengenes[[1]] = -yeigengenes[[1]];
eigenarrays[[1]] = -eigenarrays[[1]];
yeigengenes[[2]] = -yeigengenes[[2]];
eigenarrays[[2]] = -eigenarrays[[2]];
eigenarrays = Transpose[eigenarrays];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, yarrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, yarrays}] / Log[yarrays]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.89
```

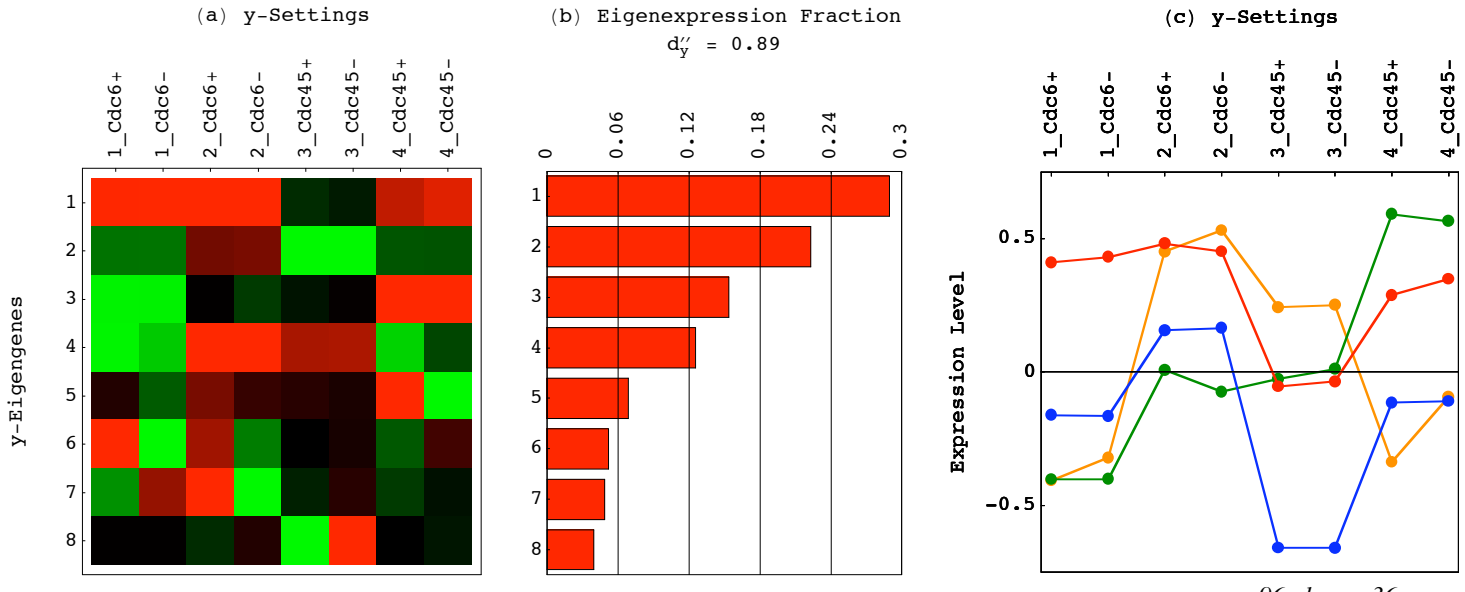
```
(* Create Fractions Bar Chart Display *)

fractions[[1]]
```

```
0.289817
```

```
gridx = Table[a, {a, 0, 0.3, 0.06}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, yarrays - a}, {a, 0, yarrays - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d''y = ", ToString[entropy]], " "},
   Center];
g = BarChart[
   Table[fractions[[yarrays - a]], {a, 0, yarrays - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.3 * 1.0001}, {0.5, yarrays + 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 + 1.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] →
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

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

contrast = 2.4;
displaying = Table[
   If[contrast * yeigengenes[[i, j]] > 0,
    If[contrast * yeigengenes[[i, j]] < 1, {contrast * yeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * yeigengenes[[i, j]] > -1, {0, -contrast * yeigengenes[[i, j]]}, {0, 1}]],
   {i, 1, yarrays}, {j, 1, yarrays}];
framex = Table[{a - 0.5, yarraynames[[a]]}, {a, 1, yarrays}];
framey = Table[{a + 1 - 0.5, yarrays - a}, {a, 0, yarrays - 1}];
labely = "y-Eigengenes";
labelx = ColumnForm[{"(a) y-Settings", " ", " "}, Center];

g = Show[
   Graphics[
    RasterArray[
     Table[
      RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
      {i, yarrays, 1, -1}, {j, 1, yarrays}]]],
   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 - 1.25, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 1.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create y-Eigengenes Graph Display *)

p = Table[0, {n, 1, 4}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0], RGBColor[1, 0.5, 0]};
labelx = ColumnForm[{"(c) y-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, yarraynames[[a]]}, {a, 1, yarrays}];
framey = {-0.5, 0, 0.5};
Do[{
  coordinates = Table[{a - 1, yeigengenes[[n, a]]}, {a, 1, yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.75, 0.75},
    DisplayFunction -> Identity],
  g = FullGraphics[g],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labely, {b_, c_}, {1., 0.}] ->
     Text[labely, {b - 1.6, 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, -1.0}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 4}]
g3 = Show[{p[[4]], p[[3]], p[[2]], p[[1]]},
   DisplayFunction -> Identity];

(* Display y-Eigengenes and Fractions *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.15];
```



```
(* Rotate y-Eigengenes *)

pattern = {1, 1, 1, 1, 1, 1, 1, 1};
x1 = Dot[pattern, yeigengenes[[1]]] / Dot[pattern, yeigengenes[[2]]];
ArcTan[x1] / Pi
```

```
-0.312755
```

```
pattern = {1, -1, 1, -1, 0, 0, 0, 0};
x2 = Dot[pattern, yeigengenes[[6]]] / Dot[pattern, yeigengenes[[7]]];
ArcTan[x2] / Pi
```

```
0.346019
```

```
pattern = {0, 0, 0, 0, 1, -1, 1, -1};
x3 = Dot[pattern, yeigengenes[[5]]] / Dot[pattern, yeigengenes[[8]]];
ArcTan[x3] / Pi
```

```
-0.250671
```

```
yeigengenes = {
   -(x1 * yeigengenes[[1]] + yeigengenes[[2]]) / Sqrt[x1^2 + 1],
   (x1 * yeigengenes[[2]] - yeigengenes[[1]]) / Sqrt[x1^2 + 1],
   yeigengenes[[3]],
   yeigengenes[[4]],
   -(x3 * yeigengenes[[5]] + yeigengenes[[8]]) / Sqrt[x3^2 + 1],
   (x2 * yeigengenes[[6]] + yeigengenes[[7]]) / Sqrt[x2^2 + 1],
   (x2 * yeigengenes[[7]] - yeigengenes[[6]]) / Sqrt[x2^2 + 1],
   (x3 * yeigengenes[[8]] - yeigengenes[[5]]) / Sqrt[x3^2 + 1]};
```

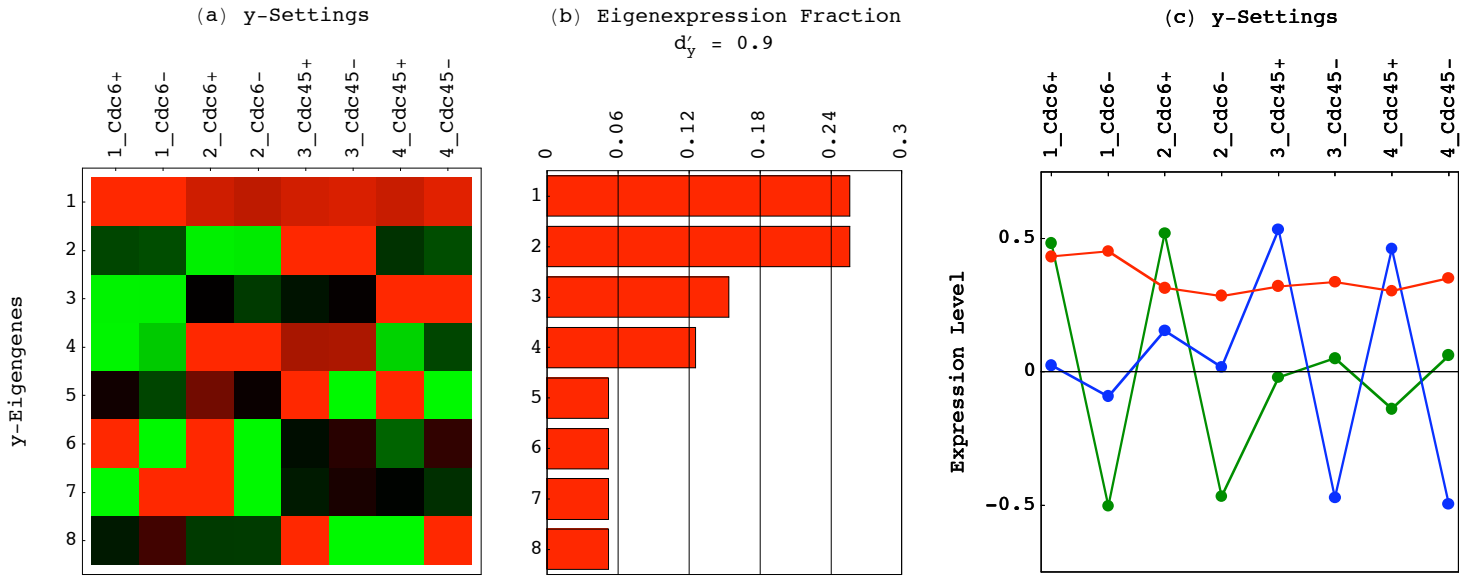
```
(* Approximate Fractions and Entropy *)

fractions[[1]] = (fractions[[1]] + fractions[[2]]) / 2;
fractions[[2]] = fractions[[1]];
fractions[[5]] = Sum[fractions[[a]], {a, 5, 8}] / 4;
Do[fractions[[a]] = fractions[[5]], {a, 6, 8}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, yarrays}] / Log[yarrays]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.9
```

```
(* Create Approximate Fractions Bar Chart Display *)

gridx = Table[a, {a, 0, 0.3, 0.06}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, yarrays - a}, {a, 0, yarrays - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d'y = ", ToString[entropy]], " "},
   Center];
g = BarChart[
   Table[fractions[[yarrays - a]], {a, 0, yarrays - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.3 * 1.0001}, {0.5, yarrays + 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 + 1.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];

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

contrast = 2.4;
displaying = Table[
   If[contrast * yeigengenes[[i, j]] > 0,
    If[contrast * yeigengenes[[i, j]] < 1, {contrast * yeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * yeigengenes[[i, j]] > -1, {0, -contrast * yeigengenes[[i, j]]}, {0, 1}]],
   {i, 1, yarrays}, {j, 1, yarrays}];
framex = Table[{a - 0.5, yarraynames[[a]]}, {a, 1, yarrays}];
framey = Table[{a + 1 - 0.5, yarrays - a}, {a, 0, yarrays - 1}];
labely = "y-Eigengenes";
labelx = ColumnForm[{"(a) y-Settings", " ", " "}, Center];
g = Show[
   Graphics[
    RasterArray[
     Table[
      RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
      {i, yarrays, 1, -1}, {j, 1, yarrays}]]],
   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 - 1.25, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 1.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Create Rotated y-Eigengenes Graph Display *)

p = Table[0, {n, 1, 6}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0],
   RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
labelx = ColumnForm[{"(c) y-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, yarraynames[[a]]}, {a, 1, yarrays}];
framey = {-0.5, 0, 0.5};
```

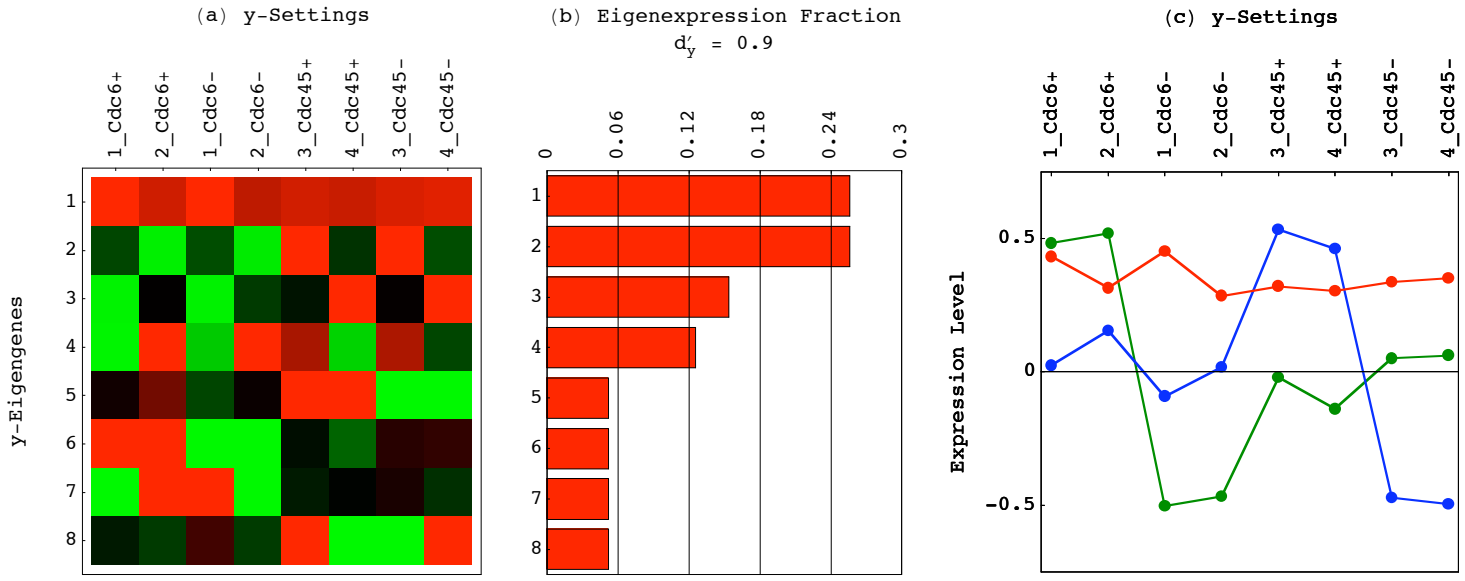
```
Do[{
  coordinates = Table[{a - 1, yeigengenes[[n, a]]}, {a, 1, yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.75, 0.75},
    DisplayFunction -> Identity],
  g = FullGraphics[g],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labely, {b_, c_}, {1., 0.}] ->
     Text[labely, {b - 1.6, 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, -1.0}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 6}]
g3 = Show[{p[[6]], p[[5]], p[[1]]},
   DisplayFunction -> Identity];

(* Display Rotated y-Eigengenes and Approximate Fractions *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.15];
```



```
(* Rearrange y-Settings *)

ryeigengenes = AppendRows[
   TakeColumns[yeigengenes, {1}], TakeColumns[yeigengenes, {3}],
   TakeColumns[yeigengenes, {2}], TakeColumns[yeigengenes, {4}],
   TakeColumns[yeigengenes, {5}], TakeColumns[yeigengenes, {7}],
   TakeColumns[yeigengenes, {6}], TakeColumns[yeigengenes, {8}]];
ryarraynames = {yarraynames[[1]], yarraynames[[3]], yarraynames[[2]], yarraynames[[4]],
   yarraynames[[5]], yarraynames[[7]], yarraynames[[6]], yarraynames[[8]]};

(* Create Approximate Fractions Bar Chart Display With Rearranged y-Settings *)

gridx = Table[a, {a, 0, 0.3, 0.06}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, yarrays - a}, {a, 0, yarrays - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d'y = ", ToString[entropy]], " "},
   Center];
g = BarChart[
   Table[fractions[[yarrays - a]], {a, 0, yarrays - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.3 * 1.0001}, {0.5, yarrays + 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 + 1.5}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create Rotated y-Eigengenes 2 D Red & Green Raster Display With Rearranged y-Settings *)

contrast = 2.4;
displaying = Table[
   If[contrast * ryeigengenes[[i, j]] > 0,
    If[contrast * ryeigengenes[[i, j]] < 1, {contrast * ryeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * ryeigengenes[[i, j]] > -1, {0, -contrast * ryeigengenes[[i, j]]}, {0, 1}]],
   {i, 1, yarrays}, {j, 1, yarrays}];
framex = Table[{a - 0.5, ryarraynames[[a]]}, {a, 1, yarrays}];
framey = Table[{a + 1 - 0.5, yarrays - a}, {a, 0, yarrays - 1}];
labely = "y-Eigengenes";
labelx = ColumnForm[{"(a) y-Settings", " ", " "}, Center];
g = Show[
   Graphics[
    RasterArray[
     Table[
      RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
      {i, yarrays, 1, -1}, {j, 1, yarrays}]]],
   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 - 1.25, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 1.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Create Rotated y-Eigengenes Graph Display With Rearranged y-Settings *)

p = Table[0, {n, 1, 6}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0],
   RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
labelx = ColumnForm[{"(c) y-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, ryarraynames[[a]]}, {a, 1, yarrays}];
framey = {-0.5, 0, 0.5};
```

```
Do[{
  coordinates = Table[{a - 1, ryeigengenes[[n, a]]}, {a, 1, yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    GridLines -> {None, {{0, RGBColor[0, 0, 0]}}},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.75, 0.75},
    DisplayFunction -> Identity],
  g = FullGraphics[g],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labely, {b_, c_}, {1., 0.}] ->
     Text[labely, {b - 1.6, 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, -1.0}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 6}]
g3 = Show[{p[[6]], p[[5]], p[[1]]},
   DisplayFunction -> Identity];

(* Display Rotated y-Eigengenes and Approximate Fractions With Rearranged y-Settings *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.15];
```



```
(* Reconstruct the Data Tensor in the Subspace of Biological Variation by Using HOSVD *)
```

```
(* Reconstruct the x-Eigengenes and the y-Eigengenes *)

xeigengenes = Drop[xeigengenes, {5, 12}];
xeigengenes = Drop[xeigengenes, {2, 2}];

yeigengenes = Drop[yeigengenes, {7, 8}];
yeigengenes = Drop[yeigengenes, {2, 4}];
```

```
(* Reconstruct the Core Tensor *)

core = Transpose[tensor, {3, 1, 2}];
core = Transpose[core, {2, 3, 1}];
core = Dot[core, Transpose[xeigengenes]];
core = Transpose[core, {3, 1, 2}];
core = Dot[core, Transpose[yeigengenes]];
core = Transpose[core, {2, 3, 1}];
```

```
(* Reconstruct the Data Tensor *)

tensor = Transpose[core, {3, 1, 2}];
tensor = Dot[tensor, yeigengenes];
tensor = Transpose[tensor, {1, 3, 2}];
tensor = Dot[tensor, xeigengenes];
tensor = Transpose[tensor, {2, 1, 3}];
Dimensions[tensor]
```

```
{8, 4270, 12}
```

```
(* Average Duplicated Time Courses *)

tensor = {
   (tensor[[2]] + tensor[[4]]) / 2.,
   (tensor[[6]] + tensor[[8]]) / 2.,
   (tensor[[1]] + tensor[[3]] + tensor[[5]] + tensor[[7]]) / 4.};
{yarrays, zgenes, xarrays} = Dimensions[tensor]
tensornames =
  ReplaceAll[ReplaceAll[ReplaceAll[
     Transpose[Drop[Transpose[
        {tensornames[[2]],
         tensornames[[6]],
         tensornames[[1]]},
        {2, 1, 3}], {1}], {2, 1, 3}],
     "1_Cdc6+" -> "Cdc6+/45+"], "3_Cdc45-" -> "Cdc45-"], "1_Cdc6-" -> "Cdc6-"];
tensornames[[3, 2]] = xarraynames;
```

```
{3, 4270, 12}
```

```
(* Normalize Each Array to Zero Average and Unit Variance *)

product2 = 1;
tensor = Transpose[tensor, {1, 3, 2}];
Do[{
  tensor[[a, b]] = tensor[[a, b]] - Sum[tensor[[a, b, c]], {c, 1, zgenes}] / zgenes,
  product2 = product2 * Sqrt[Dot[tensor[[a, b]], tensor[[a, b]]]],
  tensor[[a, b]] = tensor[[a, b]] / Sqrt[Dot[tensor[[a, b]], tensor[[a, b]]]]
 }, {a, 1, yarrays}, {b, 1, xarrays}]
tensor = Transpose[tensor, {1, 3, 2}];
product2 = product2 ^ (1 / xarrays / yarrays)
```

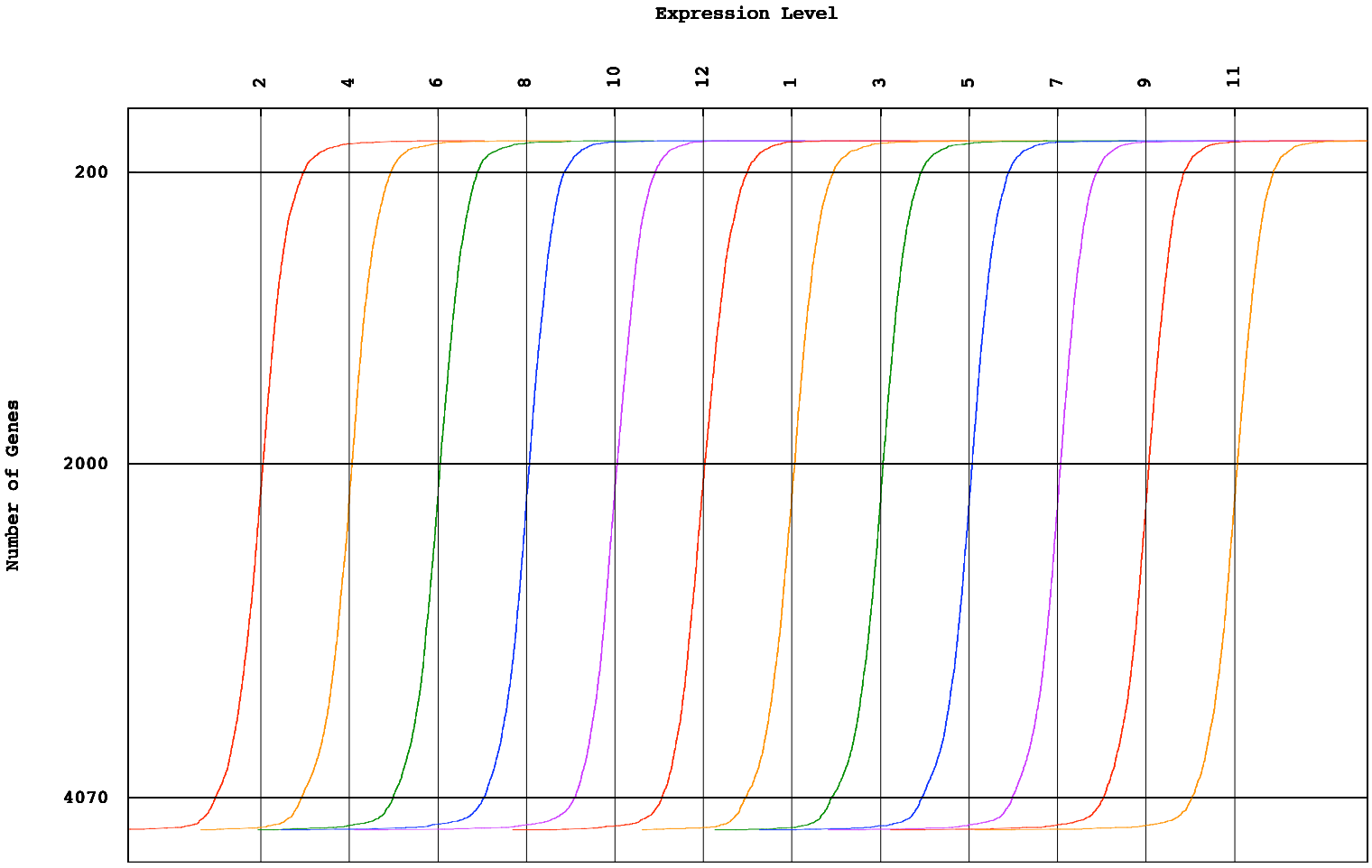
```
0.479619
```

```
(* Associate Time Points with Cell Cycle Phases of Control Time Courses *)

(* Display Sorted Control Time Courses Data *)

(* Center, Normalize and Sort Control Time Courses Data *)

arraypatterns = Transpose[tensor[[3]]];
dimensions = Dimensions[arraypatterns][[1]]
```

```
12
```

```
average = Table[1, {a, 1, zgenes}];
average = N[average / Sqrt[Dot[average, average]]];
arraypatterns = arraypatterns - N[Outer[Times, Dot[arraypatterns, average], average]];
Do[arraypatterns[[a]] = arraypatterns[[a]] / Sqrt[Dot[arraypatterns[[a]], arraypatterns[[a]]]],
 {a, 1, dimensions}]
Do[arraypatterns[[a]] = Sort[arraypatterns[[a]], OrderedQ[{{#2}, {#1}}] &],
 {a, 1, dimensions}]

(* Create Sorted Control Time Courses Data Display *)

p = Table[0, {a, 1, dimensions}];
color = {RGBColor[0.75, 0, 1], RGBColor[1, 0, 0], RGBColor[1, 0.5, 0],
   RGBColor[0, 0.5, 0], RGBColor[0, 0, 1]};
labelx = "Expression Level";
labely = "Number of Genes";
framex = Table[{0.05 * a, xarraynames[[a]]}, {a, 1, dimensions}];
framey = {{-200, " 200"}, {-2000, "2000"}, {-zgenes + 200, "4070"}};
Do[{coordinates = Table[
     If[arraypatterns[[n, a]] + 0.05 * n < -0.025, -0.025,
      If[arraypatterns[[n, a]] + 0.05 * n > 0.675, 0.675,
       arraypatterns[[n, a]] + 0.05 * n]],
     {a, 1, zgenes}],
   coordinates = Table[{coordinates[[a]], -a + 1}, {a, 1, zgenes}],
   line = Line[coordinates],
   g = Show[Graphics[{color[[Mod[n, 5] + 1]], line}],
     Frame -> True,
     FrameLabel -> {None, labely, labelx, None},
     FrameTicks -> {None, framey, framex, None},
     GridLines -> {{{0.05 * n, RGBColor[0, 0, 0]}}, {{-200, RGBColor[0, 0, 0]},
        {-2000, RGBColor[0, 0, 0]}, {-zgenes + 200, RGBColor[0, 0, 0]}}},
     PlotRange -> {{-0.025, 0.675}, {200, -zgenes + 1 - 200}},
     DisplayFunction -> Identity],
   g = FullGraphics[g],
   g[[1, 2]] = g[[1, 2]] /.
     Text[labely, {b_, c_}, {1., 0.}] ->
      Text[labely, {b - 0.052, c}, {0, 0}, {0, 1}],
   g[[1, 2]] = g[[1, 2]] /.
     Text[labelx, {b_, c_}, {0., -1.}] ->
      Text[labelx, {b, c + 420}, {0, -1}, {1, 0}],
   g[[1, 2]] = g[[1, 2]] /.
     Text[a_, {b_, c_}, {0., -1.}] ->
      Text[a, {b, c}, {0, -1}, {0, 1}],
   p[[n]] = Show[g,
     AspectRatio -> 1 / GoldenRatio,
     PlotRange -> All,
     DisplayFunction -> Identity]
  }, {n, 1, dimensions}];
```

```
(* Display Sorted Control Time Courses Data *)

Show[Table[p[[a]], {a, 1, dimensions}],
  DisplayFunction -> $DisplayFunction];
```




```
(* Associate with Cell Cycle Phases *)

(* Compute Significance of Association with Cell Cycle Phases *)

stream = "Desktop/96/Data/Gene_Annotations.txt";
geneannotations = Import[stream, "Table"];
geneannotationnames = geneannotations[[1]];
geneannotations = Drop[geneannotations, {1}];
Dimensions[geneannotations][[1]]
Clear[stream];
```

```
4270
```

```
expression = AppendRows[
   ReplaceAll[TakeColumns[geneannotations, {7}],
    {"M/G1" -> "Y", "G1" -> "N", "S" -> "N", "S/G2" -> "N", "G2/M" -> "N", "None" -> "N"}],
   ReplaceAll[TakeColumns[geneannotations, {7}],
    {"M/G1" -> "N", "G1" -> "Y", "S" -> "N", "S/G2" -> "N", "G2/M" -> "N", "None" -> "N"}],
   ReplaceAll[TakeColumns[geneannotations, {7}],
    {"M/G1" -> "N", "G1" -> "N", "S" -> "Y", "S/G2" -> "N", "G2/M" -> "N", "None" -> "N"}],
   ReplaceAll[TakeColumns[geneannotations, {7}],
    {"M/G1" -> "N", "G1" -> "N", "S" -> "N", "S/G2" -> "Y", "G2/M" -> "N", "None" -> "N"}],
   ReplaceAll[TakeColumns[geneannotations, {7}],
    {"M/G1" -> "N", "G1" -> "N", "S" -> "N", "S/G2" -> "N", "G2/M" -> "Y", "None" -> "N"}]];
annotationnames = {"M/G1", "G1", "S", "S/G2", "G2/M"};
annotations = Dimensions[annotationnames][[1]];

most = 200;
numbers = Table[Count[TakeColumns[expression, {n}], {"Y"}], {n, 1, annotations}];
parallelnumbers = Table[
  Count[
   TakeRows[
    TakeColumns[
     Sort[
      AppendRows[TakeColumns[tensor[[3]], {a}], TakeColumns[expression, {n}]], OrderedQ[{{#2}, {#1}}] &],
     {2}],
    {1, most}],
   {"Y"}], {n, 1, annotations}, {a, 1, xarrays}];
parallelprobability = Table[
   Sum[N[Binomial[numbers[[n]], b] * Binomial[zgenes - numbers[[n]], most - b] /
       Binomial[zgenes, most]], {b, parallelnumbers[[n, a]], most}],
   {n, 1, annotations}, {a, 1, xarrays}];
antinumbers = Table[
  Count[
   TakeRows[
    TakeColumns[
     Sort[
      AppendRows[TakeColumns[tensor[[3]], {a}], TakeColumns[expression, {n}]], OrderedQ[{{#1}, {#2}}] &],
     {2}],
    {1, most}],
   {"Y"}], {n, 1, annotations}, {a, 1, xarrays}];
antiprobability = Table[
   Sum[N[Binomial[numbers[[n]], b] * Binomial[zgenes - numbers[[n]], most - b] /
       Binomial[zgenes, most]], {b, antinumbers[[n, a]], most}],
   {n, 1, annotations}, {a, 1, xarrays}];
```

```
(* Tabulate Significance of Association with Cell Cycle Phases *)

ScientificForm[TableForm[Transpose[
   AppendColumns[
    AppendColumns[
     Transpose[AppendColumns[{{"Parallel"}}, {{"Association"}}, Transpose[{xarraynames}]]],
     Transpose[AppendColumns[{{" "}}, {{" "}}, Table[{" "}, {a, 1, xarrays}]]]],
    AppendRows[
     Table[{" "}, {a, 1, annotations}],
     Transpose[{annotationnames}],
     parallelprobability]]]], 2]
```

Parallel

Association	M/G1	G1	S	S/G2	G2/M
2	1.1×10^{-6}	6.8×10^{-2}	9.4×10^{-2}	$4.\times10^{-1}$	3.4×10^{-1}
4	4.6×10^{-4}	2.2×10^{-14}	2.3×10^{-1}	$6.\times10^{-1}$	1.
6	3.6×10^{-1}	7.5×10^{-40}	$2.\times10^{-5}$	7.9×10^{-1}	1.
8	9.8×10^{-1}	2.2×10^{-14}	6.9×10^{-10}	2.4×10^{-3}	9.9×10^{-1}
10	9.8×10^{-1}	8.7×10^{-1}	$2.\times10^{-5}$	7.3×10^{-8}	2.2×10^{-3}
12	3.6×10^{-1}	1.	9.4×10^{-2}	2.1×10^{-2}	2.2×10^{-15}
1	1.2×10^{-4}	7.8×10^{-1}	2.3×10^{-1}	$4.\times10^{-1}$	$8.\times10^{-1}$
3	1.7×10^{-3}	$1.\times10^{-4}$	4.4×10^{-1}	$4.\times10^{-1}$	9.7×10^{-1}
5	$2.\times10^{-1}$	3.9×10^{-15}	2.6×10^{-3}	$4.\times10^{-1}$	9.9×10^{-1}
7	3.6×10^{-1}	6.6×10^{-16}	$2.\times10^{-5}$	2.3×10^{-1}	9.9×10^{-1}
9	9.8×10^{-1}	$1.\times10^{-4}$	3.2×10^{-6}	2.4×10^{-3}	6.6×10^{-1}
11	9.8×10^{-1}	6.6×10^{-1}	$2.\times10^{-5}$	1.9×10^{-4}	4.9×10^{-1}

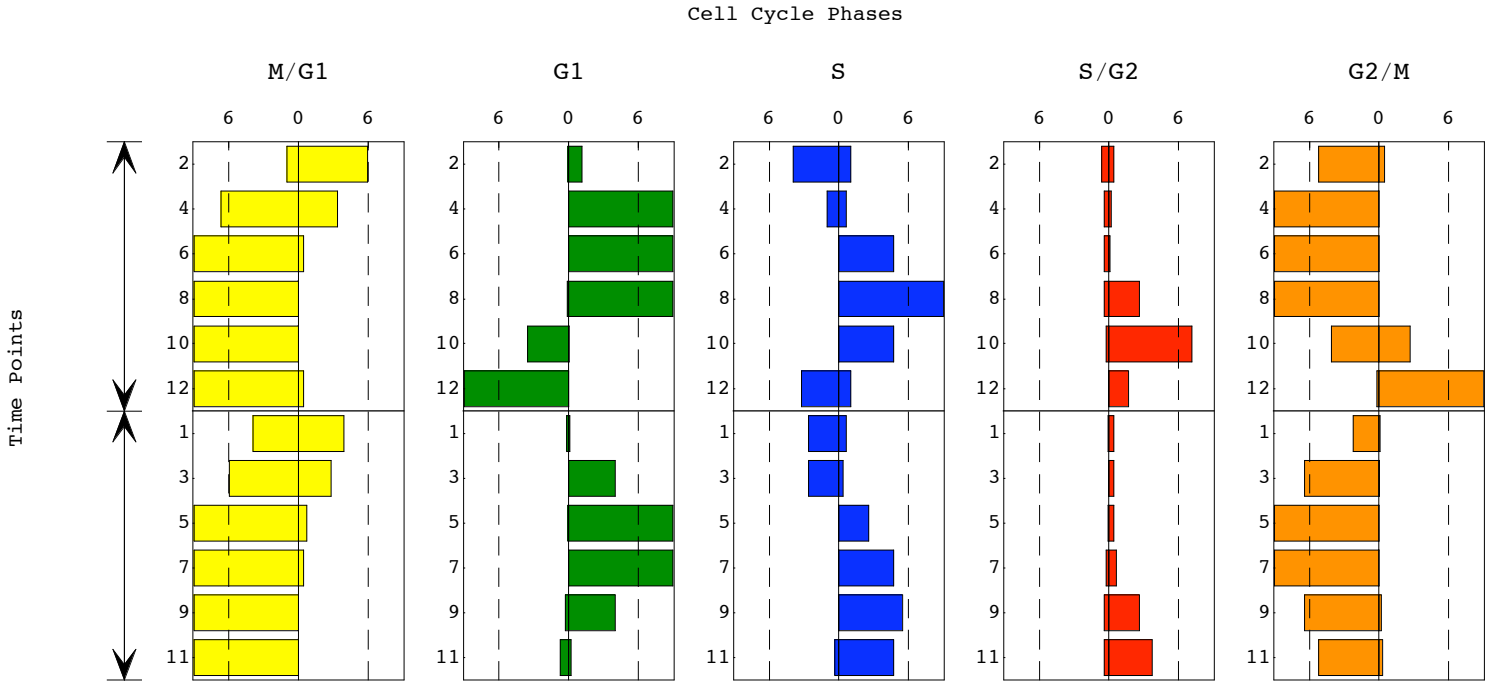
```
ScientificForm[TableForm[Transpose[
   AppendColumns[
    AppendColumns[
     Transpose[AppendColumns[{{"Antiparallel"}}, {{"Association"}}, Transpose[{xarraynames}]]],
     Transpose[AppendColumns[{{" "}}, {{" "}}, Table[{" "}, {a, 1, xarrays}]]]],
    AppendRows[
     Table[{" "}, {a, 1, annotations}],
     Transpose[{annotationnames}],
     antiprobability]]]], 2]
```

Antiparallel

Association	M/G1	G1	S	S/G2	G2/M
2	9.7×10^{-2}	8.7×10^{-1}	1.1×10^{-4}	2.3×10^{-1}	6.5×10^{-6}
4	$2.\times10^{-7}$	9.9×10^{-1}	9.4×10^{-2}	$4.\times10^{-1}$	2.9×10^{-16}
6	1.7×10^{-16}	9.7×10^{-1}	1.	$4.\times10^{-1}$	4.9×10^{-26}
8	1.1×10^{-25}	7.8×10^{-1}	1.	$4.\times10^{-1}$	1.5×10^{-10}
10	6.3×10^{-27}	2.8×10^{-4}	9.2×10^{-1}	$6.\times10^{-1}$	8.1×10^{-5}
12	1.3×10^{-11}	1.6×10^{-36}	5.8×10^{-4}	9.2×10^{-1}	6.6×10^{-1}
1	1.2×10^{-4}	6.6×10^{-1}	2.6×10^{-3}	7.9×10^{-1}	5.8×10^{-3}
3	1.1×10^{-6}	9.4×10^{-1}	2.6×10^{-3}	9.2×10^{-1}	$4.\times10^{-7}$
5	1.7×10^{-13}	8.7×10^{-1}	1.	7.9×10^{-1}	2.6×10^{-11}
7	9.3×10^{-20}	9.4×10^{-1}	1.	$6.\times10^{-1}$	2.6×10^{-11}
9	4.7×10^{-22}	5.3×10^{-1}	1.	$4.\times10^{-1}$	$4.\times10^{-7}$
11	4.7×10^{-22}	1.9×10^{-1}	4.4×10^{-1}	$4.\times10^{-1}$	6.5×10^{-6}

```
(* Create Bar Chart Display of Significance of Association with Cell Cycle Phases *)

color = {RGBColor[1, 1, 0], RGBColor[0, 0.5, 0], RGBColor[0, 0, 1],
   RGBColor[1, 0, 0], RGBColor[1, 0.5, 0]};
arrowcolor = {RGBColor[0, 0, 0], RGBColor[1, 1, 1], RGBColor[1, 1, 1],
   RGBColor[1, 1, 1], RGBColor[1, 1, 1]};
gridx = {{Log[10, 10^-6], {RGBColor[0, 0, 0], Dashing[{0.06, 0.04}]}}, {0, RGBColor[0, 0, 0]},
   {-Log[10, 10^-6], {RGBColor[0, 0, 0], Dashing[{0.06, 0.04}]}}};
framex = {{Log[10, 10^-6], "6"}, {0, "0"}, {-Log[10, 10^-6], "6"}};
labelx = Table[StyleForm[annotationnames[[a]], FontSize -> 12], {a, 1, 5}];
framey = Table[{xarrays - n + 1, xarraynames[[n]]}, {n, 1, xarrays}];
p = Table[0, {a, 1, annotations}];

Do[{paralleltable = Table[If[
     -Log[10, parallelprobability][[n, xarrays - a + 1]] > 9,
     9, -Log[10, parallelprobability][[n, xarrays - a + 1]]],
    {a, 1, xarrays}],
  antitable = Table[If[Log[10, antiprobability][[n, xarrays - a + 1]] < -9,
     -9, Log[10, antiprobability][[n, xarrays - a + 1]]], {a, 1, xarrays}],
  g = StackedBarChart[
    paralleltable, antitable,
    BarOrientation -> Horizontal,
    BarStyle -> color[[n]],
    Axes -> False,
    Frame -> True,
    FrameTicks -> {None, framey, framex, None},
    FrameLabel -> {None, None, labelx[[n]], None},
    GridLines -> {gridx, {{6.5, {RGBColor[0, 0, 0], Thickness[0.0044]}}}},
    PlotRange -> {{Log[10, 10^-9] * 1.01, -Log[10, 10^-9] * 1.01}, {0.5, xarrays + 0.5}},
    DisplayFunction -> Identity],
  g = FullGraphics[g],
g[[1, 2]] = g[[1, 2]] /.
    Text[labelx[[n]], {b_, c_}, {0., -1.}] ->
     Text[labelx[[n]], {b, c + 1}, {0, -1}, {1, 0}],
  p[[n]] = Show[{g,
     Graphics[{arrowcolor[[n]], Arrow[{-15, 0.5}, {-15, 6.5},
        HeadCenter -> 0.5, HeadLength → 0.1, HeadWidth -> 0.75]}],
     Graphics[{arrowcolor[[n]], Arrow[{-15, 6.5}, {-15, 0.5},
        HeadCenter -> 0.5, HeadLength → 0.1, HeadWidth -> 0.75]}],
     Graphics[{arrowcolor[[n]], Arrow[{-15, 6.5}, {-15, 12.5},
        HeadCenter -> 0.5, HeadLength → 0.1, HeadWidth -> 0.75]}],
     Graphics[{arrowcolor[[n]], Arrow[{-15, 12.5}, {-15, 6.5},
        HeadCenter -> 0.5, HeadLength → 0.1, HeadWidth -> 0.75]}],
     Graphics[{arrowcolor[[n]], Line[{{-13.5, 0.5}, {-16.5, 0.5}}]}],
     Graphics[{arrowcolor[[n]], Line[{{-13.5, 6.5}, {-16.5, 6.5}}]}],
     Graphics[{arrowcolor[[n]], Line[{{-13.5, 12.5}, {-16.5, 12.5}}]}]},
    PlotRange -> All,
    AspectRatio -> 2,
    DisplayFunction -> Identity]},
 {n, 1, annotations}]
```

```
(* Display Significance of Associations with Cell Cycle Phases *)

labelx = "Cell Cycle Phases";
labely = "Time Points";
Show[
  Table[Graphics[{
     Rectangle[{-9.09 * 1.01 + 15 * a, 0.5}, {9.09 + 15 * a, 12.5}, p[[a]]]}], {a, 5, 1, -1}],
  AspectRatio -> 0.45,
  Frame -> True,
  FrameTicks -> False,
  FrameStyle -> {{RGBColor[1, 1, 1]}, {RGBColor[1, 1, 1]}},
  FrameLabel -> {None, labely, labelx, None}];
```



```
(* Compute Tensor HOSVD of Averaged Data Tensor *)
```

```
(* Compute HOSVD x-Eigengenes *)
```

```
mode2 = Flatten[tensor, 1];
{eigenarrays, eigenexpressions, xeigengenes} = SingularValues[mode2];
dimension = Dimensions[eigenexpressions][[1]];
xeigengenes[[2]] = -xeigengenes[[2]];
eigenarrays[[2]] = -eigenarrays[[2]];
xeigengenes[[3]] = -xeigengenes[[3]];
eigenarrays[[3]] = -eigenarrays[[3]];
eigenarrays = Transpose[eigenarrays];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, dimension}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, dimension}] / Log[dimension]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.31
```

```
(* Create Fractions Bar Chart Display *)
```

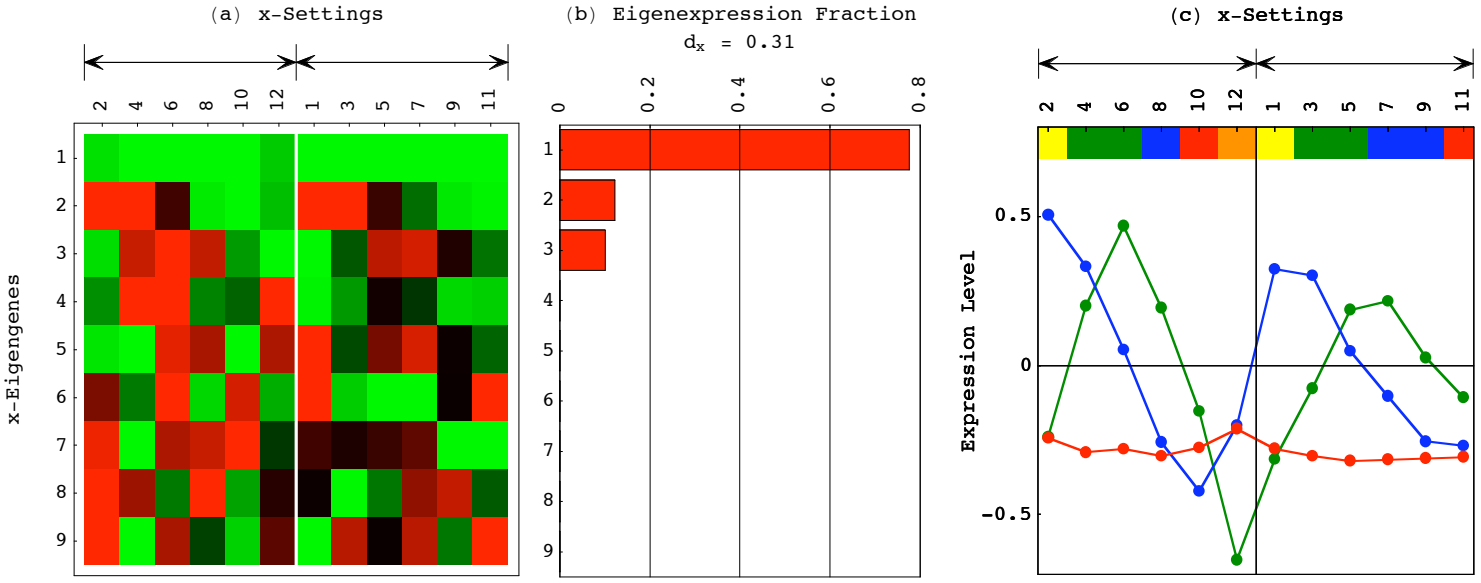
```
fractions[[1]]
```

```
0.775266
```

```
gridx = Table[{a, RGBColor[0, 0, 0]}, {a, 0, 0.8, 0.2}];
framex = Table[{a, a}, {a, 0, 0.8, 0.2}];
framey = Table[{a + 1, dimension - a}, {a, 0, dimension - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d_x = ", ToString[entropy]], " "},
   Center];
g = BarChart[Table[fractions[[dimension - a]], {a, 0, dimension - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.8 * 1.0001}, {0.5, dimension + 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 + 0.7}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[{g},
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

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

contrast = 3.6;
displaying = Table[
   If[contrast * xeigengenes[[i, j]] > 0,
    If[contrast * xeigengenes[[i, j]] < 1, {contrast * xeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * xeigengenes[[i, j]] > -1, {0, -contrast * xeigengenes[[i, j]]}, {0, 1}]],
   {i, 1, dimension}, {j, 1, xarrays}];
framex = Table[{a - 0.5, xarraynames[[a]]}, {a, 1, xarrays}];
framey = Table[{a + 1 - 0.5, dimension - a}, {a, 0, dimension - 1}];
labely = "x-Eigengenes";
labelx = ColumnForm[{"(a) x-Settings", " ", " "}, Center];
g = Show[
   Graphics[
    RasterArray[
     Table[
      RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
      {i, dimension, 1, -1}, {j, 1, xarrays}]]],
   AspectRatio -> 1,
   Frame -> True,
   FrameTicks -> {None, framey, framex, None},
   FrameLabel -> {None, labely, labelx, None},
   GridLines -> {{{6, {RGBColor[1, 1, 1], Thickness[0.0066]}}}, None},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {b - 1.5, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 0.7}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[{g, arrows[0, 6, 12, 10.5, 0.375]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Create Selected x-Eigengenes Graph Display *)

p = Table[0, {n, 1, 3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0], RGBColor[1, 0.5, 0]};
labelx = ColumnForm[{"(c) x-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, xarraynames[[a]]}, {a, 1, xarrays}];
framey = {-0.5, 0, 0.5};
```

```
Do[{coordinates = Table[{a - 1, xeigengenes[[n, a]]}, {a, 1, xarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, xarrays}],
  line = Line[coordinates],
  g = Show[{Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.7, 0.8},
    GridLines -> {{{5.5, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
    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 + 0.3}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 3}]
g3 = Show[{
    cycle[0, 0.695, 0.795],
    arrows[-0.25, 5.5, 11.25, 1.0125, 0.05],
    p[[3]], p[[2]], p[[1]]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Display x-Eigengenes, Fractions and Selected x-Eigengenes *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.16];
```



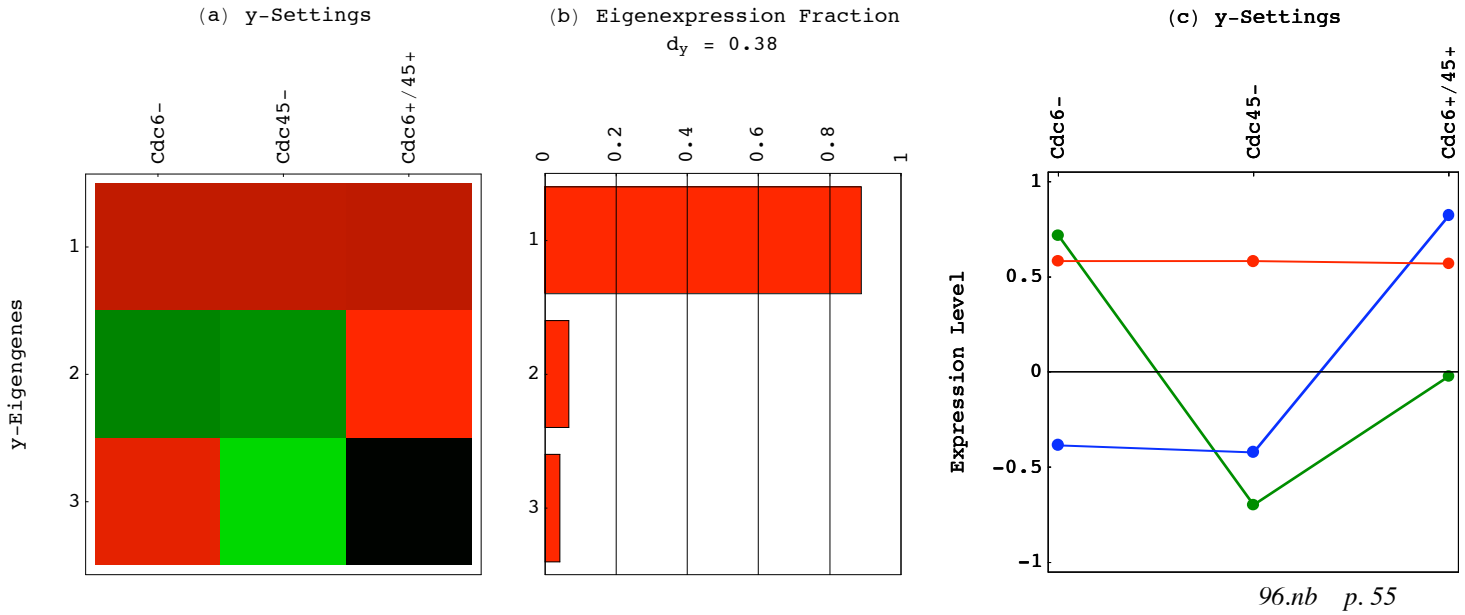
```
(* Compute HOSVD y-Eigengenes *)

mode3 = Flatten[Transpose[tensor, {3, 1, 2}], 1];
yarraynames = Flatten[Transpose[tensornames, {3, 1, 2}], 1][[xarrays]];
{eigenarrays, eigenexpressions, yeigengenes} = SingularValues[mode3];
yeigengenes[[1]] = -yeigengenes[[1]];
eigenarrays[[1]] = -eigenarrays[[1]];
yeigengenes[[2]] = -yeigengenes[[2]];
eigenarrays[[2]] = -eigenarrays[[2]];
eigenarrays = Transpose[eigenarrays];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, yarrays}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, yarrays}] / Log[yarrays]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.38
```

```
(* Create Fractions Bar Chart Display *)

gridx = Table[a, {a, 0, 1.0, 0.2}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
framex[[6]] = {1, "1"};
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, yarrays - a}, {a, 0, yarrays - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d_y = ", ToString[entropy]], " "},
   Center];
g = BarChart[
   Table[fractions[[yarrays - a]], {a, 0, yarrays - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 1.0001}, {0.5, yarrays + 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 + 0.6}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

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

contrast = 1.2;
displaying = Table[
   If[contrast * yeigengenes[[i, j]] > 0,
    If[contrast * yeigengenes[[i, j]] < 1, {contrast * yeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * yeigengenes[[i, j]] > -1, {0, -contrast * yeigengenes[[i, j]]}, {0, 1}]],
   {i, 1, yarrays}, {j, 1, yarrays}];
framex = Table[{a - 0.5, yarraynames[[a]]}, {a, 1, yarrays}];
framey = Table[{a + 1 - 0.5, yarrays - a}, {a, 0, yarrays - 1}];
labely = "y-Eigengenes";
labelx = ColumnForm[{"(a) y-Settings", " ", " "}, Center];
g = Show[
   Graphics[
    RasterArray[
     Table[
      RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
      {i, yarrays, 1, -1}, {j, 1, yarrays}]]],
   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 - 0.5, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 0.65}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[g,
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create y-Eigengenes Graph Display *)

p = Table[0, {n, 1, 3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
labelx = ColumnForm[{"(c) y-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, yarraynames[[a]]}, {a, 1, yarrays}];
framey = {-1, -0.5, 0, 0.5, 1};
Do[{
  coordinates = Table[{a - 1, yeigengenes[[n, a]]}, {a, 1, yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], 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 - 0.5, c}, {0, 0}, {0, 1}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labelx, {b_, c_}, {0., -1.}] ->
     Text[labelx, {b, c + 0.7}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 3}]
g3 = Show[{p[[3]], p[[2]], p[[1]]},
   DisplayFunction -> Identity];

(* Display y-Eigengenes and Fractions *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.15];
```



```
(* Compute HOSVD Eigenarrays *)
```

```
mode1 = Transpose[Flatten[
    Transpose[tensor, {1, 3, 2}], 1]];
{genes, arrays} = Dimensions[mode1]
zarraynames = Transpose[Flatten[
     Transpose[tensornames, {1, 3, 2}], 1]][[2]];
```

```
{4270, 36}
```

```
{zeigenarrays, eigenexpressions, eigengenes} = SingularValues[mode1];
eigengenes[[2]] = -eigengenes[[2]];
zeigenarrays[[2]] = -zeigenarrays[[2]];
eigengenes[[3]] = -eigengenes[[3]];
zeigenarrays[[3]] = -zeigenarrays[[3]];
eigengenes[[8]] = -eigengenes[[8]];
zeigenarrays[[8]] = -zeigenarrays[[8]];
zeigenarrays = Transpose[zeigenarrays];
dimension = Dimensions[eigenexpressions][[1]];
fractions = eigenexpressions^2 / Sum[eigenexpressions[[a]]^2, {a, 1, dimension}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, dimension}] / Log[dimension]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.49
```

```
(* Create Fractions Bar Chart Display *)
```

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

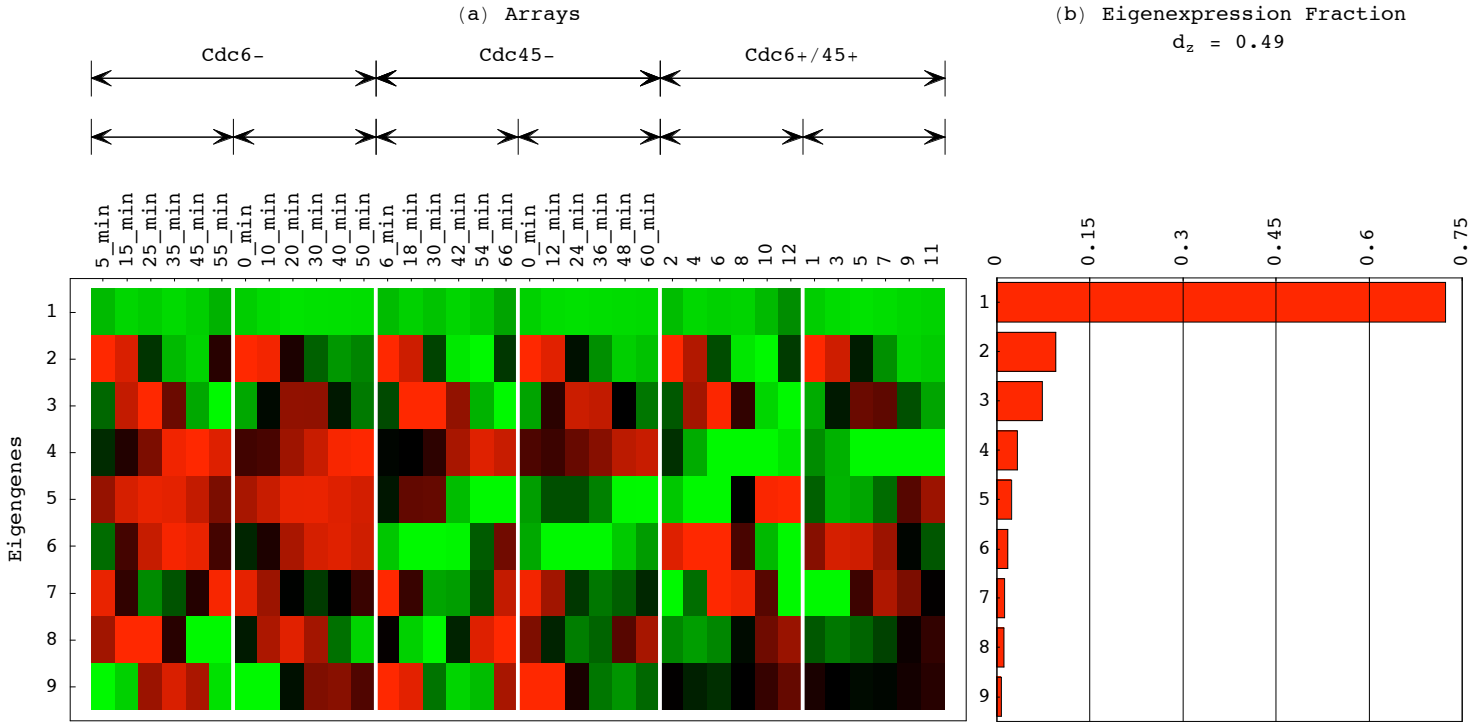
```
0.723663
```

```
gridx = Table[a, {a, 0, 0.75, 0.15}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{a + 1, dimension - a}, {a, 0, dimension - 1}];
labelx = ColumnForm[
   {"(b) Eigenexpression Fraction", StringJoin["d_z = ", ToString[entropy]], " "},
   Center];
g = BarChart[
   Table[fractions[[dimension - a]], {a, 0, dimension - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.75 * 1.0001}, {0.5, dimension + 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 + 3.8}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.35,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

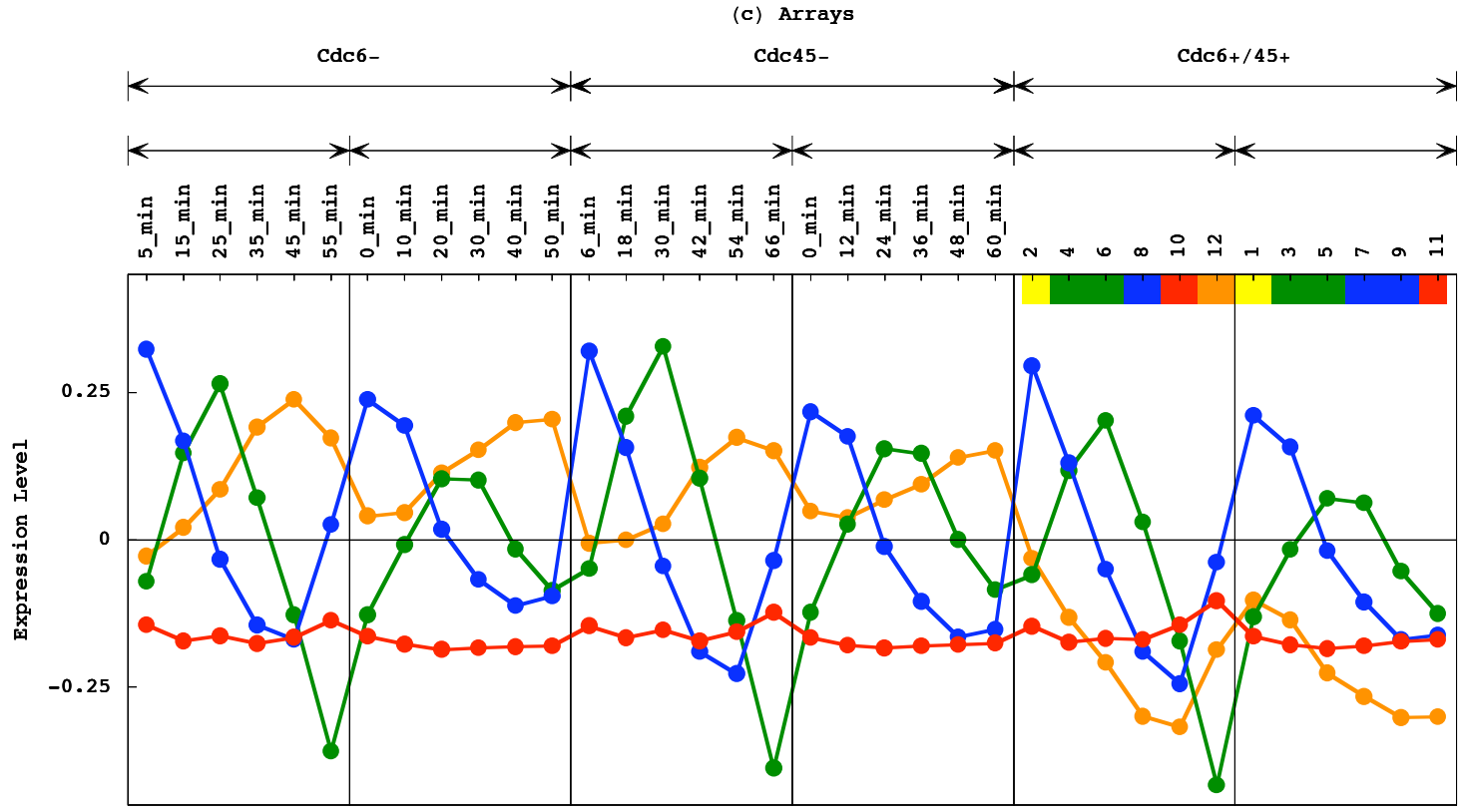
```
(* Create Eigengenes 2 D Red & Green Raster Display *)

length = 0.02;
contrast = 4.8;
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, dimension}, {j, 1, xarrays * yarrays}];
framex = Table[{a - 0.5, zarraynames[[a]]}, {a, 1, xarrays * yarrays}];
framey = Table[{a + 1 - 0.5, dimension - a}, {a, 0, dimension - 1}];
labely = "Eigengenes";
labelx = ColumnForm[{"(a) Arrays", " ", " "}, Center];
gridx = Table[{a, {RGBColor[1, 1, 1], Thickness[0.0044]}}, {a, 6, 30, 6}];
g = Show[{
    Graphics[
     RasterArray[
      Table[
       RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
       {i, dimension, 1, -1}, {j, 1, xarrays * yarrays}]]],
    Graphics[{RGBColor[0, 0, 0], Text[yarraynames[[1]], {6, 14}]}],
    Graphics[{RGBColor[0, 0, 0], Text[yarraynames[[2]], {18, 14}]}],
    Graphics[{RGBColor[0, 0, 0], Text[yarraynames[[3]], {30, 14}]}]},
   AspectRatio -> 1,
   Frame -> True,
   FrameTicks -> {None, framey, framex, None},
   FrameLabel -> {None, labely, labelx, None},
   GridLines -> {gridx, None},
   PlotRange → {-0.225, 9.225},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {b - 1.6, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 4}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[{g,
    arrows[0, 6, 12, 12.25, 0.375],
    arrows[12, 18, 24, 12.25, 0.375],
    arrows[24, 30, 36, 12.25, 0.375],
    arrows[0, 12, 24, 13.5, 0.375],
    arrows[12, 24, 36, 13.5, 0.375]},
   AspectRatio -> 1.05 / 1.5,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create Selected Eigengenes Graph Display *)

length = 0.0125;
p = Table[0, {n, 1, 6}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1],
   RGBColor[0, 0.5, 0], RGBColor[1, 0.5, 0], RGBColor[1, 0, 0], RGBColor[0, 0, 1]};
labelx = ColumnForm[{"(c) Arrays"}, Center];
gridx = Table[{a, RGBColor[0, 0, 0]}, {a, 5.5, 29.5, 6}];
framex = Table[{a - 1, zarraynames[[a]]}, {a, 1, xarrays * yarrays}];
labely = ColumnForm[{" ", " ", "Expression Level"}, Center];
framey = {-0.25, 0, 0.25};
Do[{
  coordinates = Table[{a - 1, eigengenes[[n, a]]}, {a, 1, xarrays * yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, xarrays * yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.011], points}],
     Graphics[{color[[n]], Thickness[0.0022], line}],
     Graphics[{RGBColor[0, 0, 0], Text[yarraynames[[1]], {5.5, 0.82}]}],
     Graphics[{RGBColor[0, 0, 0], Text[yarraynames[[2]], {17.5, 0.82}]}],
     Graphics[{RGBColor[0, 0, 0], Text[yarraynames[[3]], {29.5, 0.82}]}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, "           "},
    GridLines -> {gridx, {{0, RGBColor[0, 0, 0]}}},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {{-0.5, 35.5}, {-0.45, 0.45}},
    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 + 0.4}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05 / 2.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 4}]
g3 = Show[{
    cycle[24, 0.4, 0.45],
    arrows[-0.5, 5.5, 11.5, 0.66, 0.025],
    arrows[11.5, 17.5, 23.5, 0.66, 0.025],
    arrows[23.5, 29.5, 35.5, 0.66, 0.025],
    arrows[-0.5, 11.5, 23.5, 0.77, 0.025],
    arrows[11.5, 23.5, 35.5, 0.77, 0.025],
    p[[4]], p[[3]], p[[2]], p[[1]]},
   AspectRatio -> 1.05 / 2,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

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

Show[GraphicsArray[{g1, g2}],
  GraphicsSpacing -> -0.3];
```




```
Show[GraphicsArray[{g3}]];
```



```
(* Compute Core and Subtensors *)
```

```
(* Compute Core Tensor *)

core = Transpose[tensor, {3, 1, 2}];
core = Dot[Transpose[zeigenarrays], core];
core = Transpose[core, {2, 3, 1}];
core = Dot[core, Transpose[xeigengenes]];
core = Transpose[core, {3, 1, 2}];
core = Dot[core, Transpose[yeigengenes]];
core = Transpose[core, {2, 3, 1}];
```

```
(* Compute Subtensors and Fractions *)

fractions = Sort[Abs[Flatten[core]], OrderedQ[{{#2}, {#1}}] &];
dimension = Dimensions[fractions][[1]];
subtensors = Table[Flatten[
    If[
     Position[core, fractions[[a]]] == {},
     {-1, Position[core, -fractions[[a]]]},
     {1, Position[core, fractions[[a]]]}]
   ], {a, 1, 8}];
signs = Flatten[TakeColumns[subtensors, {1}]];
subtensors = AppendRows[
   TakeColumns[subtensors, {3, 4}],
   TakeColumns[subtensors, {2}]];
fractions = fractions^2 / Sum[fractions[[a]]^2, {a, 1, dimension}];
entropy = -N[Sum[fractions[[a]] * Log[fractions[[a]]], {a, 1, dimension}] /
     Log[dimension]];
entropy = N[Round[100 * entropy] / 100]
```

```
0.23
```

```
(* Identify and Display Significant Subtensors *)
```

```
(* Identify Significant Subtensors *)

r[z_, x_, y_] := Position[subtensors, {z, x, y}][[1, 1]];
rfractions = {
   fractions[[r[1, 1, 1]]],
   fractions[[r[2, 2, 1]]],
   fractions[[r[3, 3, 1]]],
   fractions[[r[4, 1, 2]]],
   fractions[[r[5, 1, 3]]],
   fractions[[r[6, 1, 3]]],
   fractions[[r[7, 3, 2]]],
   fractions[[r[8, 3, 3]]]};
```

```
(* Create Significant Fractions Bar Charts Display *)

rsubtensors = {
   {"  1", "1", "1"},
   {"  2", "2", "1"},
   {"  3", "3", "1"},
   {"  4", "1", "2"},
   {"  5", "1", "3"},
   {"  6", "1", "3"},
   {"  7", "3", "2"},
   {"  8", "3", "3"}};
```

```
rsigns = {
   signs[[r[1, 1, 1]]],
   signs[[r[2, 2, 1]]],
   signs[[r[3, 3, 1]]],
   signs[[r[4, 1, 2]]],
   signs[[r[5, 1, 3]]],
   signs[[r[6, 1, 3]]],
   signs[[r[7, 3, 2]]],
   signs[[r[8, 3, 3]]]};

rfractions[[5]]
```

```
0.0108561
```

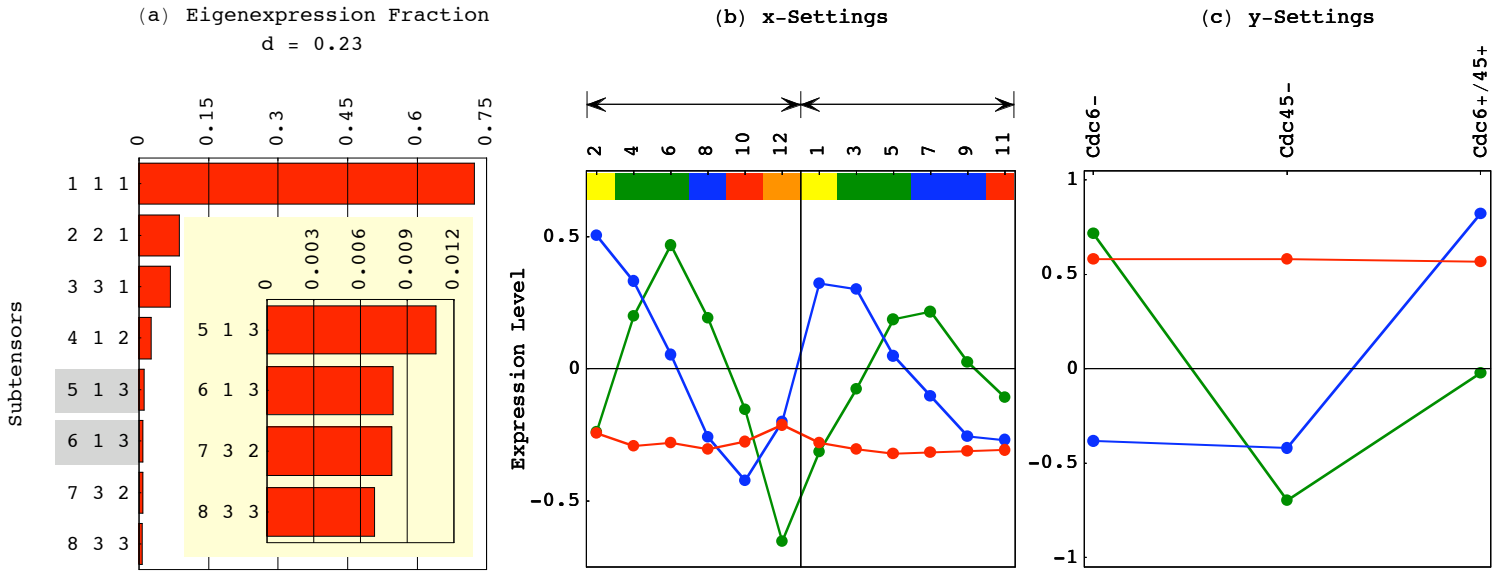
```
gridx = Table[a, {a, 0, 0.012, 0.003}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 5}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 5}];
framey = Table[{8 - a + 1, StringJoin[
     " ", ToString[rsubtensors[[a, 1]]],
     " ", ToString[rsubtensors[[a, 2]]],
     " ", ToString[rsubtensors[[a, 3]]]]},
   {a, 5, 8}];
table = Table[rfractions[[9 - a]], {a, 1, 4}];
g = BarChart[table,
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.012 * 1.0001}, {0.5, 4 + 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}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.3,
   PlotRange -> All,
   DisplayFunction -> Identity];

rfractions[[1]]
```

```
0.722771
```

```
gridx = Table[a, {a, 0, 0.75, 0.15}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{8 - a + 1, StringJoin[
     " ", ToString[rsubtensors[[a, 1]]],
     " ", ToString[rsubtensors[[a, 2]]],
     " ", ToString[rsubtensors[[a, 3]]]]},
   {a, 1, 8}];
labelx = ColumnForm[
   {"(a) Eigenexpression Fraction", StringJoin["d = ", ToString[entropy]], " "}, Center];
labely = ColumnForm[{"Subtensors", " "}, Center];
g = BarChart[
   Table[rfractions[[8 - a]], {a, 0, 8 - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.75 * 1.0001}, {0.5, 8 + 0.5}},
   AspectRatio -> 1,
   Axes -> False,
   Frame -> True,
   FrameTicks -> {None, framey, framex, None},
   FrameLabel -> {None, labely, labelx, "    "},
   GridLines -> {gridx, None},
   DisplayFunction -> Identity];
```

```
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {b - 0.22, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 1.3}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[{
    Graphics[Table[{GrayLevel[0.1 * rsigns[[8 - a]] + 0.9],
       Rectangle[{0, a + 0.55}, {-0.18, a + 1.4}]}, {a, 0, 7}]],
    g,
    Graphics[{RGBColor[1, 1, 0.8], Rectangle[{0.1, 0.75}, {0.72, 7.35}]}],
    Graphics[{Rectangle[{0.035, 0.75}, {0.7, 7.2}, g2]}]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Create Selected x-Eigengenes Graph Display *)

length = 0.035;
p = Table[0, {n, 1, 3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0], RGBColor[1, 0.5, 0]};
labelx = ColumnForm[{"(b) x-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, xarraynames[[a]]}, {a, 1, xarrays}];
framey = {-0.5, 0, 0.5};
Do[{
  coordinates = Table[{a - 1, xeigengenes[[n, a]]}, {a, 1, xarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, xarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.75, 0.75},
    GridLines -> {{{5.5, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
    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 + 0.5}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 3}]
g2 = Show[{
    cycle[0, 0.64, 0.74],
    arrows[-0.25, 5.5, 11.25, 1, 0.05],
    p[[3]], p[[2]], p[[1]]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create Rotated y-Eigengenes Graph Display *)

p = Table[0, {n, 1, 3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
labelx = ColumnForm[{"(c) y-Settings"}, Center];
labely = ColumnForm[{" ", " "}, Center];
framex = Table[{a - 1, yarraynames[[a]]}, {a, 1, yarrays}];
framey = {-1, -0.5, 0, 0.5, 1};
Do[{
  coordinates = Table[{a - 1, yeigengenes[[n, a]]}, {a, 1, yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], 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 - 0.5, c}, {0, 0}, {0, 1}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labelx, {b_, c_}, {0., -1.}] ->
     Text[labelx, {b, c + 0.7}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 3}]
g3 = Show[{p[[3]], p[[2]], p[[1]]},
   DisplayFunction -> Identity];

(* Display Fractions, Selected x-Eigengenes and Rotated y-Eigengenes *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.1375];
```



```
(* Identify and Display Significant and Unique Subtensors *)
```

```
(* Identify Significant and Unique Subtensors *)

r[z_, x_, y_] := Position[subtensors, {z, x, y}][[1, 1]];
rfractions = {
   fractions[[r[1, 1, 1]]],
   fractions[[r[2, 2, 1]]],
   fractions[[r[3, 3, 1]]],
   fractions[[r[4, 1, 2]]],
   fractions[[r[5, 1, 3]]] + fractions[[r[6, 1, 3]]],
   fractions[[r[7, 3, 2]]],
   fractions[[r[8, 3, 3]]]};

(* Create Significant and Unique Fractions Bar Charts Display *)

rsubtensors = {
   {"  1", "1", "1"},
   {"  2", "2", "1"},
   {"  3", "3", "1"},
   {"  4", "1", "2"},
   {"5+6", "1", "3"},
   {"  7", "3", "2"},
   {"  8", "3", "3"}};
rsigns = {
   signs[[r[1, 1, 1]]],
   signs[[r[2, 2, 1]]],
   signs[[r[3, 3, 1]]],
   signs[[r[4, 1, 2]]],
   1,
   signs[[r[7, 3, 2]]],
   signs[[r[8, 3, 3]]]};

rfractions[[4]]
```

```
0.0266241
```

```
gridx = Table[a, {a, 0, 0.028, 0.007}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 5}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 5}];
framey = Table[{7 - a + 1, StringJoin[
     " ", ToString[rsubtensors[[a, 1]]],
     " ", ToString[rsubtensors[[a, 2]]],
     " ", ToString[rsubtensors[[a, 3]]]]},
   {a, 4, 7}];
table = Table[rfractions[[8 - a]], {a, 1, 4}];
g = BarChart[table,
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.028 * 1.0001}, {0.5, 4 + 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}, {0, -1}, {0, 1}];
g2 = Show[g,
   AspectRatio -> 1.3,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

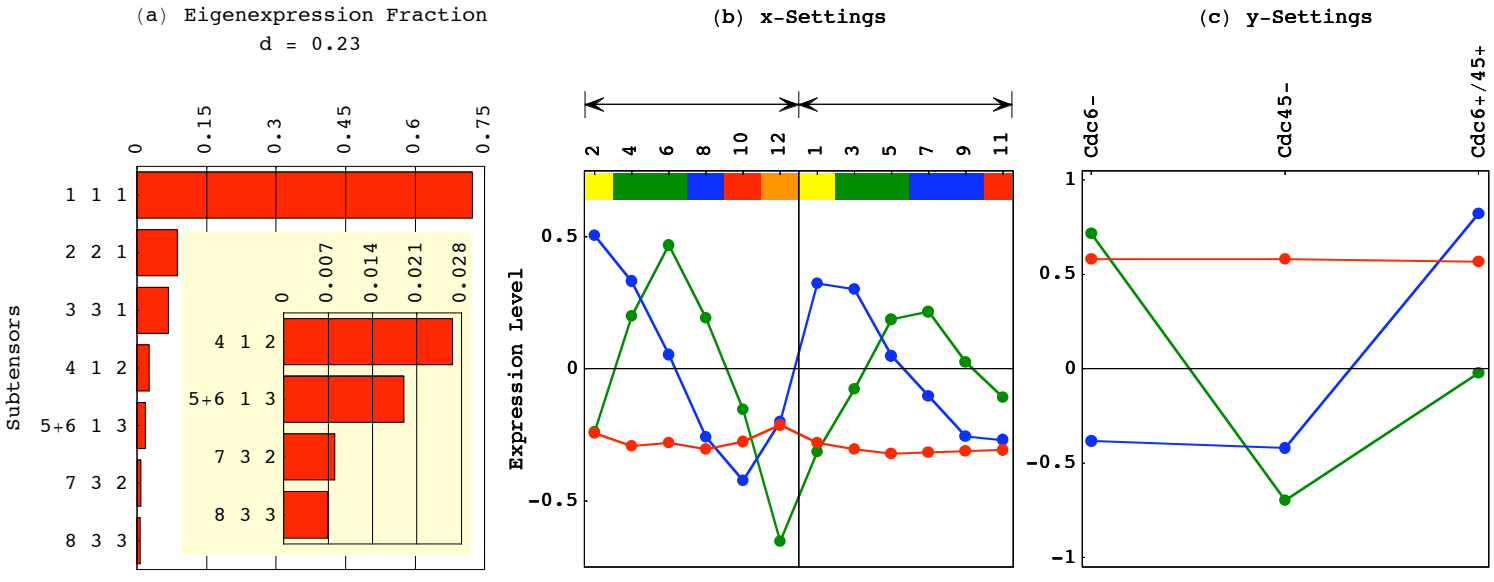
```
rfractions[[1]]
```

```
0.722771
```

```
gridx = Table[a, {a, 0, 0.75, 0.15}];
framex = Table[{gridx[[a]], gridx[[a]]}, {a, 1, 6}];
gridx = Table[{gridx[[a]], RGBColor[0, 0, 0]}, {a, 1, 6}];
framey = Table[{7 - a + 1, StringJoin[
     " ", ToString[rsubtensors[[a, 1]]],
     " ", ToString[rsubtensors[[a, 2]]],
     " ", ToString[rsubtensors[[a, 3]]]]},
   {a, 1, 7}];
labelx = ColumnForm[
   {"(a) Eigenexpression Fraction", StringJoin["d = ", ToString[entropy]], " "}, Center];
labely = ColumnForm[{"Subtensors", " "}, Center];
g = BarChart[
   Table[rfractions[[7 - a]], {a, 0, 7 - 1}],
   BarOrientation -> Horizontal,
   PlotRange -> {{0, 0.75 * 1.0001}, {0.5, 7 + 0.5}},
   AspectRatio -> 1,
   Axes -> False,
   Frame -> True,
   FrameTicks -> {None, framey, framex, None},
   FrameLabel -> {None, labely, labelx, "    "},
   GridLines -> {gridx, None},
   DisplayFunction -> Identity];
g = FullGraphics[g];
g[[1, 2]] = g[[1, 2]] /.
   Text[labely, {b_, c_}, {1., 0.}] ->
    Text[labely, {b - 0.22, c}, {0, 0}, {0, 1}];
g[[1, 2]] = g[[1, 2]] /.
   Text[labelx, {b_, c_}, {0., -1.}] ->
    Text[labelx, {b, c + 1.3}, {0, -1}, {1, 0}];
g[[1, 2]] = g[[1, 2]] /.
   Text[a_, {b_, c_}, {0., -1.}] ->
    Text[a, {b, c}, {0, -1}, {0, 1}];
g1 = Show[{g,
    Graphics[{RGBColor[1, 1, 0.8], Rectangle[{0.1, 0.75}, {0.72, 6.35}]}],
    Graphics[{Rectangle[{0.075, 0.75}, {0.72, 6.2}, g2]}]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create Selected x-Eigengenes Graph Display *)

length = 0.035;
p = Table[0, {n, 1, 3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0], RGBColor[1, 0.5, 0]};
labelx = ColumnForm[{"(b) x-Settings"}, Center];
labely = ColumnForm[{" ", "Expression Level"}, Center];
framex = Table[{a - 1, xarraynames[[a]]}, {a, 1, xarrays}];
framey = {-0.5, 0, 0.5};
Do[{
  coordinates = Table[{a - 1, xeigengenes[[n, a]]}, {a, 1, xarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, xarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], line}]},
    Frame -> True,
    FrameLabel -> {None, labely, labelx, None},
    FrameTicks -> {None, framey, framex, None},
    PlotRange -> {-0.75, 0.75},
    GridLines -> {{{5.5, RGBColor[0, 0, 0]}}, {{0, RGBColor[0, 0, 0]}}},
    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 + 0.5}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 3}]
g2 = Show[{
    cycle[0, 0.64, 0.74],
    arrows[-0.25, 5.5, 11.25, 1, 0.05],
    p[[3]], p[[2]], p[[1]]},
   AspectRatio -> 1.05,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create Rotated y-Eigengenes Graph Display *)

p = Table[0, {n, 1, 3}];
color = {RGBColor[1, 0, 0], RGBColor[0, 0, 1], RGBColor[0, 0.5, 0]};
labelx = ColumnForm[{"(c) y-Settings"}, Center];
labely = ColumnForm[{" ", " "}, Center];
framex = Table[{a - 1, yarraynames[[a]]}, {a, 1, yarrays}];
framey = {-1, -0.5, 0, 0.5, 1};
Do[{
  coordinates = Table[{a - 1, yeigengenes[[n, a]]}, {a, 1, yarrays}],
  points = Table[Point[coordinates[[a]]], {a, 1, yarrays}],
  line = Line[coordinates],
  g = Show[
    {Graphics[{color[[n]], PointSize[0.022], points}],
     Graphics[{color[[n]], Thickness[0.0044], 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 - 0.5, c}, {0, 0}, {0, 1}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[labelx, {b_, c_}, {0., -1.}] ->
     Text[labelx, {b, c + 0.7}, {0, -1}, {1, 0}],
  g[[1, 2]] = g[[1, 2]] /.
    Text[a_, {b_, c_}, {0., -1.}] ->
     Text[a, {b, c}, {0, -1}, {0, 1}],
  p[[n]] = Show[g,
    AspectRatio -> 1.05,
    PlotRange -> All,
    DisplayFunction -> Identity]},
 {n, 1, 3}]
g3 = Show[{p[[3]], p[[2]], p[[1]]},
   DisplayFunction -> Identity];

(* Display Fractions, Selected x-Eigengenes and Rotated y-Eigengenes *)

Show[GraphicsArray[{g1, g2, g3}],
  GraphicsSpacing -> -0.1375];
```



```
(* Associate Selected Eigenarrays and Superpositions of Eigenarrays *)
```

```
(* Read Gene and ARS Annotations *)

stream = "Desktop/96/Data/Gene_Annotations.txt";
geneannotations = Import[stream, "Table"];
geneannotationnames = geneannotations[[1]];
geneannotations = Drop[geneannotations, {1}];
Dimensions[geneannotations][[1]]
Clear[stream];
```

```
4270
```

```
stream = "Desktop/96/Data/ARS_Annotations.txt";
arsannotations = Import[stream, "Table"];
arsannotations = Drop[arsannotations, {1}];
arss = Dimensions[arsannotations][[1]]
Clear[stream];
```

```
325
```

```
(* Annotate Genes with ARSs at Their 3' Ends *)

arsat3end = Table["", {genes}];
Do[{
  Do[arsat3end[[a]] = StringJoin[arsat3end[[a]], If[geneannotations[[a, 9]] == arsannotations[[b, 2]],
      If[geneannotations[[a, 10]] < geneannotations[[a, 11]],
       If[arsannotations[[b, 3]] - geneannotations[[a, 11]] < 100,
        If[geneannotations[[a, 11]] < arsannotations[[b, 4]],
         If[geneannotations[[a, 10]] < arsannotations[[b, 3]],
          arsannotations[[b, 1]], ""], ""], ""],
       If[geneannotations[[a, 11]] - arsannotations[[b, 4]] < 100,
        If[geneannotations[[a, 11]] > arsannotations[[b, 3]],
         If[geneannotations[[a, 10]] > arsannotations[[b, 4]],
          arsannotations[[b, 1]], ""], ""], ""]], ""]],
   {b, 1, arss}],
  If[arsat3end[[a]] == "", arsat3end[[a]] = "None"]},
 {a, 1, genes}]
```

```
(* Annotate Genes with ARSs at Their 5' Ends *)

arsat5end = Table["", {genes}];
Do[{
  Do[arsat5end[[a]] = StringJoin[arsat5end[[a]], If[geneannotations[[a, 9]] == arsannotations[[b, 2]],
      If[geneannotations[[a, 11]] < geneannotations[[a, 10]],
       If[arsannotations[[b, 3]] - geneannotations[[a, 10]] < 100,
        If[geneannotations[[a, 10]] < arsannotations[[b, 4]],
         If[geneannotations[[a, 11]] < arsannotations[[b, 3]],
          arsannotations[[b, 1]], ""], ""], ""],
       If[geneannotations[[a, 10]] - arsannotations[[b, 4]] < 100,
        If[geneannotations[[a, 10]] > arsannotations[[b, 3]],
         If[geneannotations[[a, 11]] > arsannotations[[b, 4]],
          arsannotations[[b, 1]], ""], ""], ""]], ""]],
   {b, 1, arss}],
  If[arsat5end[[a]] == "", arsat5end[[a]] = "None"]},
 {a, 1, genes}]

(* Annotate Genes that Overlap ARSs *)

arsoverlap = Table["", {genes}];
Do[{
  Do[arsoverlap[[a]] = StringJoin[arsoverlap[[a]],
     If[geneannotations[[a, 9]] == arsannotations[[b, 2]],
      If[geneannotations[[a, 10]] < geneannotations[[a, 11]],
       If[arsannotations[[b, 3]] - geneannotations[[a, 10]] ≥ 0,
        If[arsannotations[[b, 4]] - geneannotations[[a, 11]] ≤ 0,
         arsannotations[[b, 1]], ""],
        If[arsannotations[[b, 4]] - geneannotations[[a, 11]] ≥ 0,
         arsannotations[[b, 1]], ""], ""],
       If[arsannotations[[b, 3]] - geneannotations[[a, 11]] ≥ 0,
        If[arsannotations[[b, 4]] - geneannotations[[a, 10]] ≤ 0,
         arsannotations[[b, 1]], ""],
        If[arsannotations[[b, 4]] - geneannotations[[a, 10]] ≥ 0,
         arsannotations[[b, 1]], ""],
        ""]
      ], ""]],
   {b, 1, arss}],
  If[arsoverlap[[a]] == "", arsoverlap[[a]] = "None"]},
 {a, 1, genes}]

(* Save ARS Annotations *)

ars = AppendColumns[
   {{"ARS_at_3'<100", "ARS_at_5'<100", "Overlaps_ARS"}},
   Transpose[AppendRows[{arsat3end, arsat5end, arsoverlap}]]];
```

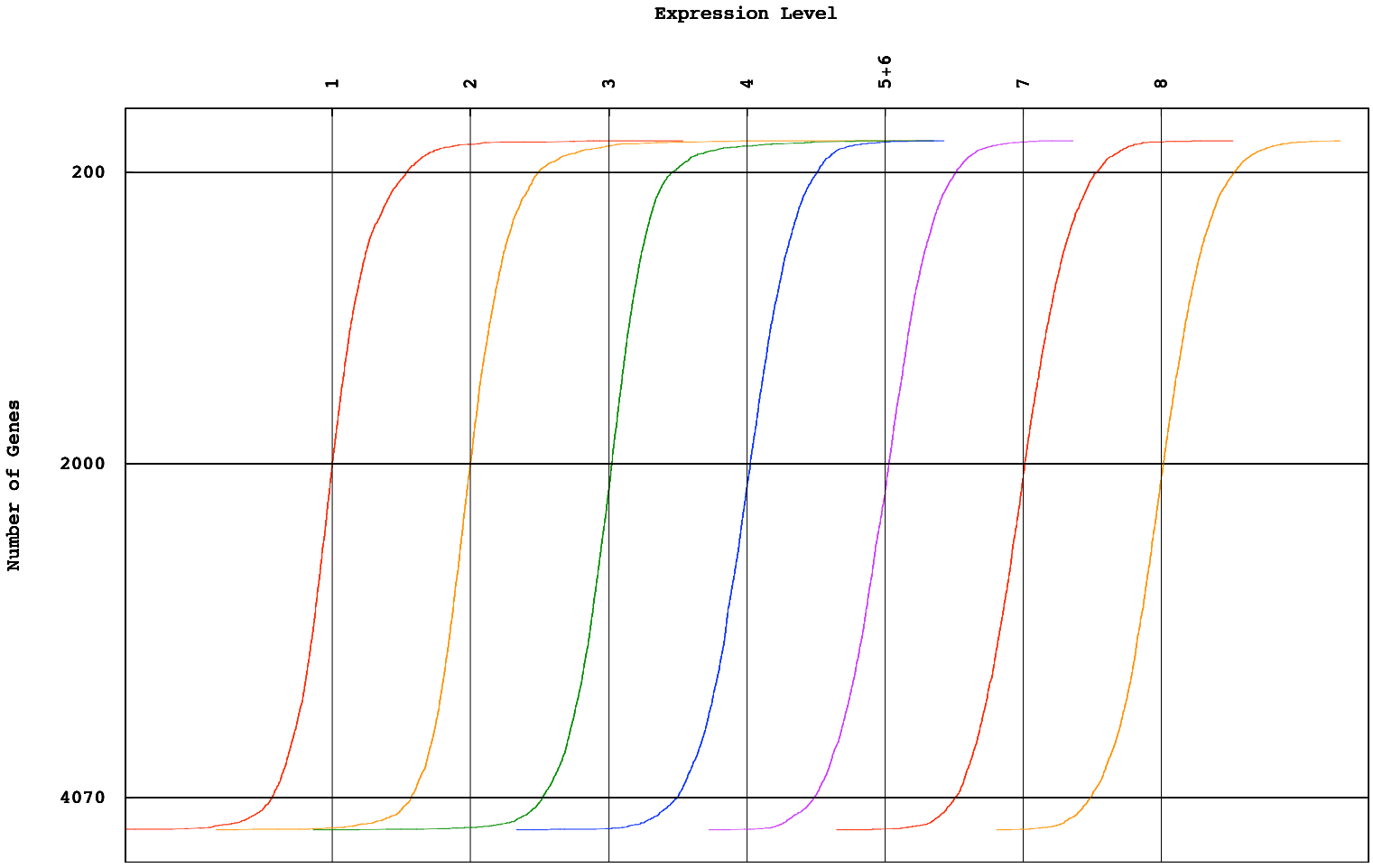
```
(* Annotate Genes Significant in Superpositions of Subtensors *)

zeigenarray[z1_, z2_, x_, y_] := (
    signs[[r[z1, x, y]]] * Sqrt[fractions[[r[z1, x, y]]]] * TakeColumns[zeigenarrays, {z1}] +
     signs[[r[z2, x, y]]] * Sqrt[fractions[[r[z2, x, y]]]] * TakeColumns[zeigenarrays, {z2}]
   ) / Sqrt[fractions[[r[z1, x, y]]] + fractions[[r[z2, x, y]]]];
rzeigenarrays = AppendRows[
   TakeColumns[zeigenarrays, {1, 4}],
   zeigenarray[5, 6, 1, 3],
   TakeColumns[zeigenarrays, {7, 8}]];

(* Annotate Genes Underexpressed in the 4 th Subtensor, Overexpressed in the 5 th and 6 th *)

intersection456 = N[Round[Sqrt[
       rfractions[[4]] * TakeColumns[rzeigenarrays, {4}]^2 +
        rfractions[[5]] * TakeColumns[rzeigenarrays, {5}]^2 +
        rfractions[[6]] * TakeColumns[rzeigenarrays, {6}]^2] * 100000] / 100000];
Do[intersection456[[a]] =
  If[rzeigenarrays[[a, 4]] < 0.007,
   If[rzeigenarrays[[a, 5]] > -0.007,
    If[rzeigenarrays[[a, 6]] > -0.007, intersection456[[a]],
     {"N"}], {"N"}], {"N"}], {a, 1, zgenes}]

(* Annotate Genes Overexpressed in the 4 th Subtensor, Underexpressed in the 6 th and 7 th *)

intersection457 = N[Round[Sqrt[
       rfractions[[4]] * TakeColumns[rzeigenarrays, {4}]^2 +
        rfractions[[5]] * TakeColumns[rzeigenarrays, {5}]^2 +
        rfractions[[7]] * TakeColumns[rzeigenarrays, {7}]^2] * 100000] / 100000];
Do[intersection457[[a]] =
  If[rzeigenarrays[[a, 4]] > -0.007,
   If[rzeigenarrays[[a, 5]] < 0.007,
    If[rzeigenarrays[[a, 7]] < 0.007, intersection457[[a]],
     {"N"}], {"N"}], {"N"}], {a, 1, zgenes}]

(* Save Significance in Subtensor Superpositions Annotations *)

intersection = AppendColumns[
   {{"Histone_Enriched", "ARS_at_3'_Enriched"}},
   AppendRows[intersection456, intersection457]];

(* Save Annotations with Selected Eigenarrays and Superpositions of Eigenarrays *)

geneannotations = AppendColumns[{geneannotationnames}, geneannotations];
geneannotations = AppendRows[
   TakeColumns[geneannotations, {1, 8}],
   ars,
   TakeColumns[geneannotations, {9, 15}],
   AppendRows[geneannotations,
    AppendColumns[{{"1", "2", "3", "4", "5+6", "7", "8"}},
     N[Round[rzeigenarrays * 100000] / 100000]]],
   intersection];

stream = OpenWrite[
   "Desktop/96/Data/Eigenarrays.txt",
   PageWidth -> Infinity];
Write[stream, OutputForm[TableForm[
    geneannotations,
    TableSpacing -> {0, 1}]]];
Close[stream];
```

```
(* Display Sorted Selected Eigenarrays and Superpositions of Eigenarrays *)

(* Center, Normalize and Sort Selected Eigenarrays and Superpositions of Eigenarrays *)

arraypatterns = Transpose[rzeigenarrays];
arraypatternnames = {"1", "2", "3", "4", "5+6", "7", "8"};
dimensions = Dimensions[arraypatterns][[1]]
```

7

```
average = Table[1, {a, 1, zgenes}];
average = N[average / Sqrt[Dot[average, average]]];
arraypatterns = arraypatterns - N[Outer[Times, Dot[arraypatterns, average], average]];
Do[arraypatterns[[a]] = arraypatterns[[a]] / Sqrt[Dot[arraypatterns[[a]], arraypatterns[[a]]]],
 {a, 1, dimensions}]
Do[arraypatterns[[a]] = Sort[arraypatterns[[a]], OrderedQ[{{#2}, {#1}}] &],
 {a, 1, dimensions}]

(* Create Sorted Selected Eigenarrays and Superpositions of Eigenarrays Display *)

p = Table[0, {a, 1, dimensions}];
color = {
   RGBColor[0.75, 0, 1],
   RGBColor[1, 0, 0],
   RGBColor[1, 0.5, 0],
   RGBColor[0, 0.5, 0],
   RGBColor[0, 0, 1]};
labelx = "Expression Level";
labely = "Number of Genes";
framex = Table[{0.05 * a, arraypatternnames[[a]]},
   {a, 1, dimensions}];
framey = {{-200, "200"}, {-2000, "2000"}, {-zgenes + 200, "4070"}};
Do[{
   coordinates = Table[
     If[arraypatterns[[n, a]] + 0.05 * n < -0.025, -0.025,
      If[arraypatterns[[n, a]] + 0.05 * n > 0.425, 0.425,
       arraypatterns[[n, a]] + 0.05 * n]],
     {a, 1, zgenes}],
   coordinates = Table[{coordinates[[a]], -a + 1}, {a, 1, zgenes}],
   line = Line[coordinates],
   g = Show[
     Graphics[{color[[Mod[n, 5] + 1]], line}],
     Frame -> True,
     FrameLabel -> {None, labely, labelx, None},
     FrameTicks -> {None, framey, framex, None},
     GridLines -> {{{0.05 * n, RGBColor[0, 0, 0]}}, {{-200, RGBColor[0, 0, 0]},
        {-2000, RGBColor[0, 0, 0]}, {-zgenes + 200, RGBColor[0, 0, 0]}}},
     PlotRange -> {{-0.025, 0.425}, {200, -zgenes + 1 - 200}},
     DisplayFunction -> Identity],
   g = FullGraphics[g],
   g[[1, 2]] = g[[1, 2]] /.
     Text[labely, {b_, c_}, {1., 0.}] ->
      Text[labely, {b - 0.032, c}, {0, 0}, {0, 1}],
   g[[1, 2]] = g[[1, 2]] /.
     Text[labelx, {b_, c_}, {0., -1.}] ->
      Text[labelx, {b, c + 420}, {0, -1}, {1, 0}],
   g[[1, 2]] = g[[1, 2]] /.
     Text[a_, {b_, c_}, {0., -1.}] ->
      Text[a, {b, c}, {0, -1}, {0, 1}],
   p[[n]] = Show[g,
     AspectRatio -> 1 / GoldenRatio,
     PlotRange -> All,
     DisplayFunction -> Identity]
  }, {n, 1, dimensions}];
```

```
(* Display Sorted Selected Eigenarrays and Superpositions of Eigenarrays *)

Show[Table[p[[a]], {a, 1, dimensions}],
  DisplayFunction -> $DisplayFunction];
```



```
(* Sort Time Points and Genes in the Common Cell Cycle Subtensor Space *)

(* Create Parameteric Plot of HOSVD-Sorted Time Points *)

x = 2;
y = 3;
coordinate[z_] := xeigengenes[[z]];
phase = ArcTan[coordinate[y] / coordinate[x]] +
   (Sign[coordinate[x]] + Table[{1}, {a, 1, xarrays}]) * Pi / 2.;
radius = Flatten[Sqrt[(coordinate[y]^2 + coordinate[x]^2) /
     Sum[coordinate[a]^2, {a, 1, 3}]]];
xcoordinate = Flatten[-radius * Cos[phase]];
ycoordinate = Flatten[-radius * Sin[phase]];
points = Table[Point[{xcoordinate[[a]], ycoordinate[[a]]}], {a, 1, xarrays}];
textcorrections = 0.1 * {{-1, 0}, {1, 0}, {1, 0}, {-1, 0}, {-0.75, 1}, {-0.75, 1},
    {-1, 0}, {-1, 0}, {0, 1}, {-1, 0}, {-0.75, 0.75}, {0.75, -1}};
texts = Table[Text[xarraynames[[a]], {xcoordinate[[a]], ycoordinate[[a]]} +
     textcorrections[[a]]], {a, 1, xarrays}];

p = Show[{
    Graphics[{RGBColor[1, 1, 0], PointSize[0.035],
      points[[1]], points[[7]]}],
    Graphics[{RGBColor[0, 0.5, 0], PointSize[0.035],
      points[[2]], points[[3]], points[[8]], points[[9]]}],
    Graphics[{RGBColor[1, 0, 0], PointSize[0.035],
      points[[5]], points[[12]]}],
    Graphics[{RGBColor[0, 0, 1], PointSize[0.035],
      points[[4]], points[[10]], points[[11]]}],
    Graphics[{RGBColor[1, 0.5, 0], PointSize[0.035], points[[6]]}],
    Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.2, -1.12}]}],
    Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.9, -0.75}]}],
    Graphics[{RGBColor[0, 0.5, 0], Text["G1", {0.8, 0.8}]}],
    Graphics[{RGBColor[0, 0, 1], Text["S", {-0.4, 1}]}],
    Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-1.1, -0.36}]}],
    Graphics[texts],
    Graphics[{RGBColor[0, 0, 0], Text["(a)", {-1.1, 1.15}]}],
    Graphics[{RGBColor[0, 0, 0], Text["θ=π/2", {0.2, 1.12}]}],
    Graphics[{RGBColor[0, 0, 0], Text["θ=0", {1.12, -0.12}]}],
    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]}],
    Graphics[{RGBColor[0, 0, 0], Arrow[{-1.25, 0}, {1.25, 0},
       HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}],
    Graphics[{RGBColor[0, 0, 0], Arrow[{0, -1.25}, {0, 1.25},
       HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}]},
   AspectRatio -> 1,
   PlotRange -> {{-1.25, 1.25}, {-1.25, 1.25}},
   Frame -> True,
   FrameTicks -> False,
   FrameLabel -> {None, None, None, None},
   GridLines -> {None, None},
   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}];
s1 = Show[p,
   AspectRatio -> 1.0,
   PlotRange -> All,
   DisplayFunction -> Identity];
```

```
(* Create Parameteric Plot of HOSVD-Sorted Cell Cycle-Regulated Genes *)

x = 2;
y = 3;
coordinate[z_] := TakeColumns[zeigenarrays, {z}]; phase = ArcTan[coordinate[y] / coordinate[x]] +
  (Sign[rfractions[[x]] * coordinate[x]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
radius = Flatten[Sqrt[coordinate[y]^2 + coordinate[x]^2] /
    Sqrt[Sum[coordinate[a]^2, {a, 1, 3}]]];
xcoordinate = Flatten[-radius * Cos[phase]];
ycoordinate = Flatten[-radius * Sin[phase]];

points = Table[0, {5}];
radii = Table[0, {5}];
stages = {"M/G1", "G1", "S", "S/G2", "G2/M"};
Do[{
  position = Flatten[Position[Drop[Transpose[geneannotations][[7]], 1], stages[[a]]]],
  points[[a]] = Table[Point[{xcoordinate[[position[[b]]]], ycoordinate[[position[[b]]]]}],
    {b, 1, Dimensions[position][[1]]}],
  radii[[a]] = Table[radius[[position[[b]]]], {b, 1, Dimensions[position][[1]]}]},
 {a, 1, 5}]
Dimensions[Flatten[points]][[1]]
Position[Table[If[Sort[Flatten[radii], OrderedQ[{{#2}, {#1}}] &][[b]] < 0.5, 0, 1],
    {b, 1, Dimensions[Flatten[radii]][[1]]}], 0][[1, 1]] - 1
```

```
576
```

```
528
```

```
Do[Print[Dimensions[points[[a]]][[1]]], {a, 1, 5}]
```

```
84
```

```
211
```

```
52
```

```
88
```

```
141
```

```
(* 576 cell cycle genes: 84 in M/G1, 211 in G1, 52 in S, 88 in S/G2, 141 in G2/M. *)

Do[Print[Position[Table[If[Sort[radii[[a]], OrderedQ[{{#2}, {#1}}] &][[b]] < 0.5, 0, 1],
      {b, 1, Dimensions[radii[[a]]][[1]]}], 0][[1, 1]] - 1], {a, 1, 5}]
```

```
73
```

```
201
```

```
48
```

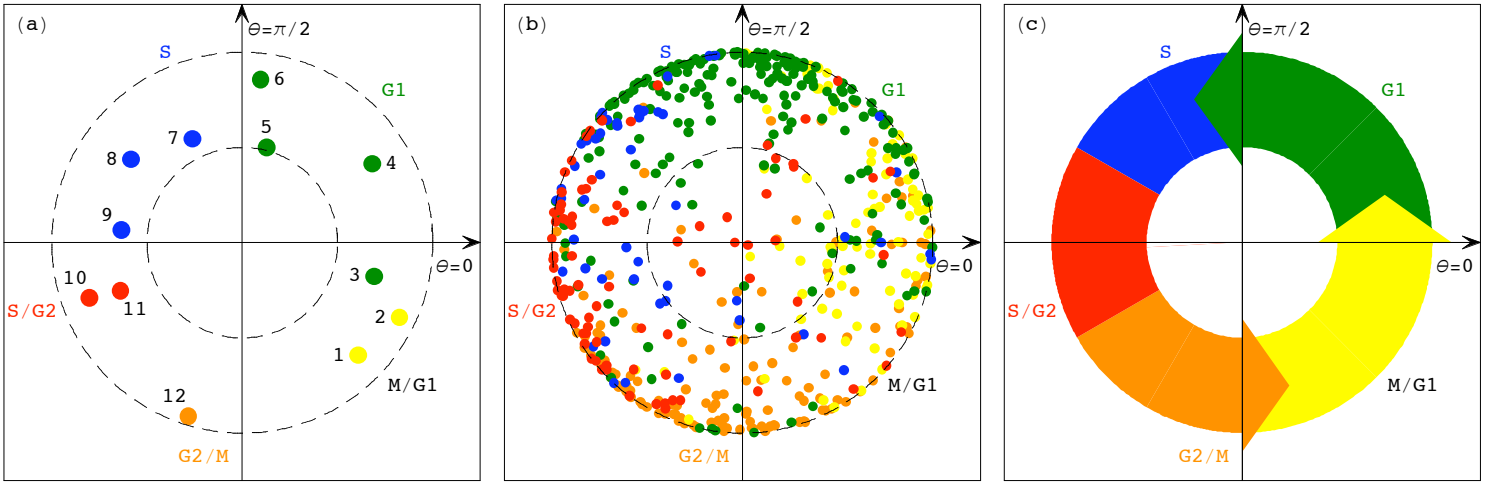
```
75
```

```
131
```

```
(* 528 genes with at least 25 % of normalized expression in the cell cycle subspace: *)

(* 73 in M/G1, 201 in G1, 48 in S, 75 in S/G2, 131 in G2/M. *)
```

```
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[0, 0.5, 0], PointSize[0.02], points[[2]]}],
    Graphics[{RGBColor[0, 0, 1], PointSize[0.02], points[[3]]}],
    Graphics[{RGBColor[1, 0, 0], PointSize[0.02], points[[4]]}],
    Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.2, -1.12}]}],
    Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.9, -0.75}]}],
    Graphics[{RGBColor[0, 0.5, 0], Text["G1", {0.8, 0.8}]}],
    Graphics[{RGBColor[0, 0, 1], Text["S", {-0.4, 1}]}],
    Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-1.1, -0.36}]}],
    Graphics[{RGBColor[0, 0, 0], Text["(b)", {-1.1, 1.15}]}],
    Graphics[{RGBColor[0, 0, 0], Text["θ=π/2", {0.2, 1.12}]}],
    Graphics[{RGBColor[0, 0, 0], Text["θ=0", {1.12, -0.12}]}],
    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]}],
    Graphics[{RGBColor[0, 0, 0], Arrow[{-1.25, 0}, {1.25, 0},
       HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}],
    Graphics[{RGBColor[0, 0, 0], Arrow[{0, -1.25}, {0, 1.25},
       HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}]},
   AspectRatio -> 1,
   PlotRange -> {{-1.25, 1.25}, {-1.25, 1.25}},
   Frame -> True,
   FrameTicks -> False,
   FrameLabel -> {None, None, None, None},
   GridLines -> {None, None},
   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}];
s2 = Show[p,
   AspectRatio -> 1.0,
   PlotRange -> All,
   DisplayFunction -> Identity];

(* Create Schematic Plot of HOSVD Cell Cycle 2 D Subspace *)

R[theta_] = {{Cos[theta], Sin[theta]}, {-Sin[theta], Cos[theta]}};
x = Transpose[{{0.4, 0}, {1.1, 0}, {0.75, 0.25}}];
polypoints[theta_] = Transpose[Dot[R[theta], x]];
```

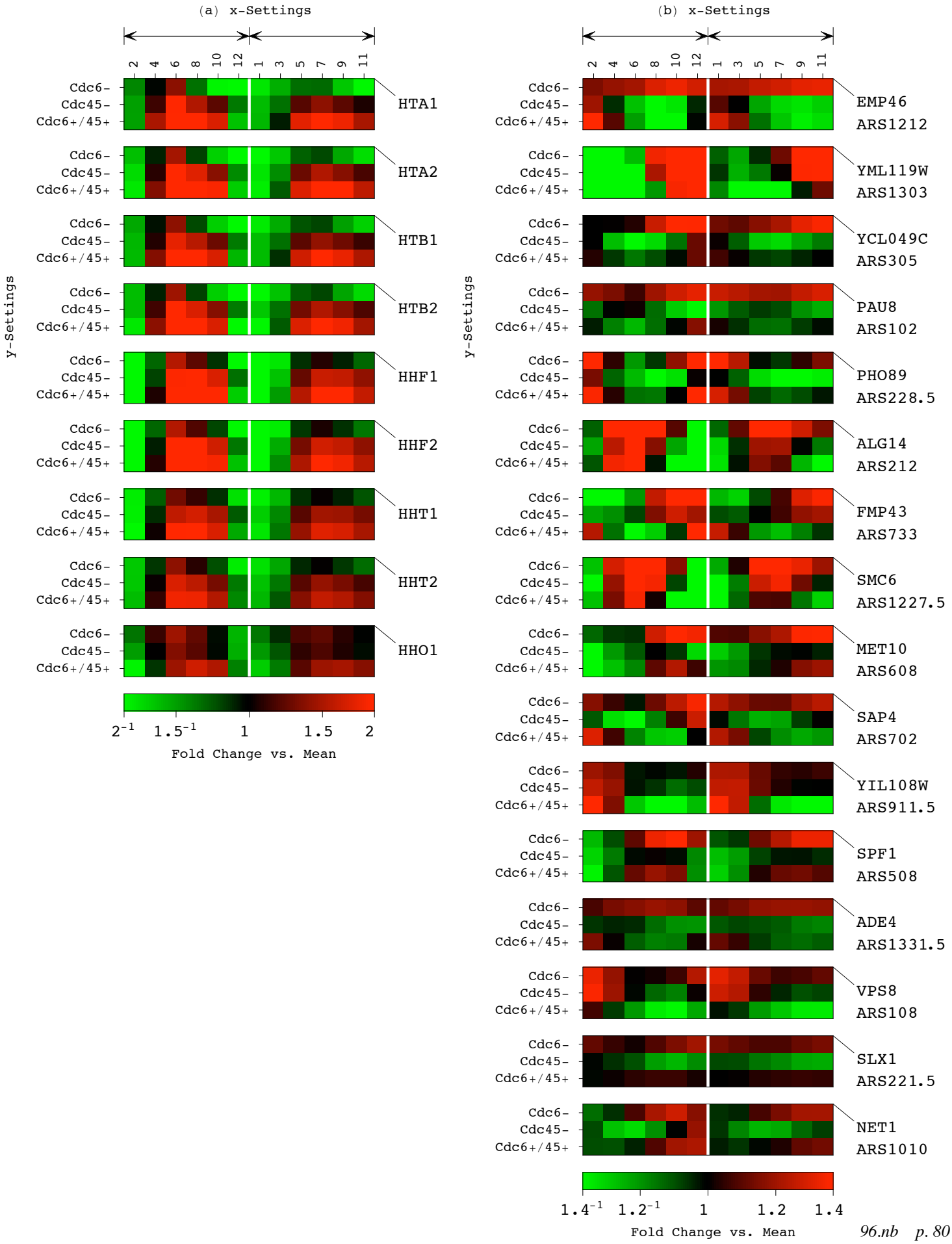
```
p = Show[{
    Graphics[{RGBColor[1, 0.5, 0], Text["G2/M", {-0.2, -1.12}]}],
    Graphics[{RGBColor[0, 0, 0], Text["M/G1", {0.9, -0.75}]}],
    Graphics[{RGBColor[0, 0.5, 0], Text["G1", {0.8, 0.8}]}],
    Graphics[{RGBColor[0, 0, 1], Text["S", {-0.4, 1}]}],
    Graphics[{RGBColor[1, 0, 0], Text["S/G2", {-1.1, -0.36}]}],
    Graphics[{RGBColor[0, 0, 0], Text["(c)", {-1.1, 1.15}]}],
    Graphics[{RGBColor[0, 0, 0], Text["Θ=π/2", {0.2, 1.12}]}],
    Graphics[{RGBColor[0, 0, 0], Text["Θ=0", {1.12, -0.12}]}],
    Graphics[{RGBColor[1, 0.5, 0], Disk[{0, 0}, 1, {-5 * Pi / 6, -Pi / 2}]}],
    Graphics[{RGBColor[1, 1, 0], Disk[{0, 0}, 1, {-Pi / 2, 0}]}],
    Graphics[{RGBColor[0, 0.5, 0], Disk[{0, 0}, 1, {0, Pi / 2}]}],
    Graphics[{RGBColor[0, 0, 1], Disk[{0, 0}, 1, {Pi / 2, 5 * Pi / 6}]}],
    Graphics[{RGBColor[1, 0, 0], Disk[{0, 0}, 1, {5 * Pi / 6, 7 * Pi / 6}]}],
    Graphics[{RGBColor[1, 1, 1], Disk[{0, 0}, 0.5]}],
    Graphics[{RGBColor[1, 0.5, 0], Polygon[polypoints[Pi / 2]]}],
    Graphics[{RGBColor[1, 1, 0], Polygon[polypoints[0]]}],
    Graphics[{RGBColor[0, 0.5, 0], Polygon[polypoints[3 * Pi / 2]]}],
    Graphics[{RGBColor[0, 0, 0], Arrow[{-1.25, 0}, {1.25, 0},
       HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}],
    Graphics[{RGBColor[0, 0, 0], Arrow[{0, -1.25}, {0, 1.25},
       HeadCenter -> 0.5, HeadLength -> 0.035, HeadWidth -> 0.75]}]},
   AspectRatio -> 1,
   PlotRange -> {{-1.25, 1.25}, {-1.25, 1.25}},
   Frame -> True,
   FrameTicks -> False,
   FrameLabel -> {None, None, 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}];
s3 = Show[p,
   AspectRatio -> 1.,
   PlotRange -> All,
   DisplayFunction -> Identity];
Clear[x];

(* Display HOSVD-Sorted Time Points and Genes in the Common Cell Cycle Subtensor Space *)

Show[GraphicsArray[{s1, s2, s3}],
    GraphicsSpacing -> 0];
```



```
(* Display Selected Genes *)

(* Create Selected Genes Raster Displays *)

contrast = 1 / (Log[2, 2] / product1 / product2);
genelist1 =
  {{"YDR225W", "HTA1" },
   {"YBL003C", "HTA2"},
   {"YDR224C", "HTB1" },
   {"YBL002W", "HTB2"},
   {"YBR009C", "HHF1"},
   {"YNL030W", "HHF2"},
   {"YBR010W", "HHT1"},
   {"YNL031C", "HHT2"},
   {"YPL127C", "HHO1"}};
dimension1 = Dimensions[genelist1][[1]];
genelist1 = Table[genelist1[[n]], {n, dimension1, 1, -1}];
color1 = Table[0, {a, 1, dimension1}];
Do[{gene = Transpose[tensor, {2, 1, 3}][[Position[genenames, genelist1[[n, 1]]][[1, 1]]]],
  average = Sum[Flatten[gene][[a]], {a, 1, yarrays * xarrays}] / yarrays / xarrays,
  gene = gene - Table[average, {yarrays}, {xarrays}],
  displaying = Table[If[contrast * gene[[i, j]] > 0,
     If[contrast * gene[[i, j]] < 1, {contrast * gene[[i, j]], 0}, {1, 0}],
     If[contrast * gene[[i, j]] > -1, {0, -contrast * gene[[i, j]]}, {0, 1}]],
    {i, yarrays, 1, -1}, {j, 1, xarrays}],
  color1[[n]] = Table[RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
    {i, 1, yarrays}, {j, 1, xarrays}]},
 {n, 1, dimension1}]

contrast = 1 / (Log[2, 1.4] / product1 / product2);
genelist2 =
  {{"YLR080W", "EMP46", "ARS1212"},
   {"YML119W", "YML119W", "ARS1303"},
   {"YCL049C", "YCL049C", "ARS305"},
   {"YAL068C", "PAU8", "ARS102"},
   {"YBR296C", "PHO89", "ARS228.5"},
   {"YBR070C", "ALG14", "ARS212"},
   {"YGR243W", "FMP43", "ARS733"},
   {"YLR383W", "SMC6", "ARS1227.5"},
   {"YFR030W", "MET10", "ARS608"},
   {"YGL229C", "SAP4", "ARS702"},
   {"YIL108W", "YIL108W", "ARS911.5"},
   {"YEL031W", "SPF1", "ARS508"},
   {"YMR300C", "ADE4", "ARS1331.5"},
   {"YAL002W", "VPS8", "ARS108"},
   {"YBR228W", "SLX1", "ARS221.5"},
   {"YJL076W", "NET1", "ARS1010"}};
dimension2 = Dimensions[genelist2][[1]];
genelist2 = Table[genelist2[[n]], {n, dimension2, 1, -1}];
color2 = Table[0, {a, 1, dimension2}];
Do[{gene = Transpose[tensor, {2, 1, 3}][[Position[genenames, genelist2[[n, 1]]][[1, 1]]]],
  average = Sum[Flatten[gene][[a]], {a, 1, yarrays * xarrays}] / yarrays / xarrays,
  gene = gene - Table[average, {yarrays}, {xarrays}],
  displaying = Table[
    If[contrast * gene[[i, j]] > 0,
     If[contrast * gene[[i, j]] < 1, {contrast * gene[[i, j]], 0}, {1, 0}],
     If[contrast * gene[[i, j]] > -1, {0, -contrast * gene[[i, j]]}, {0, 1}]],
    {i, yarrays, 1, -1}, {j, 1, xarrays}],
  color2[[n]] = Table[
    RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
    {i, 1, yarrays}, {j, 1, xarrays}]},
 {n, 1, dimension2}]
```

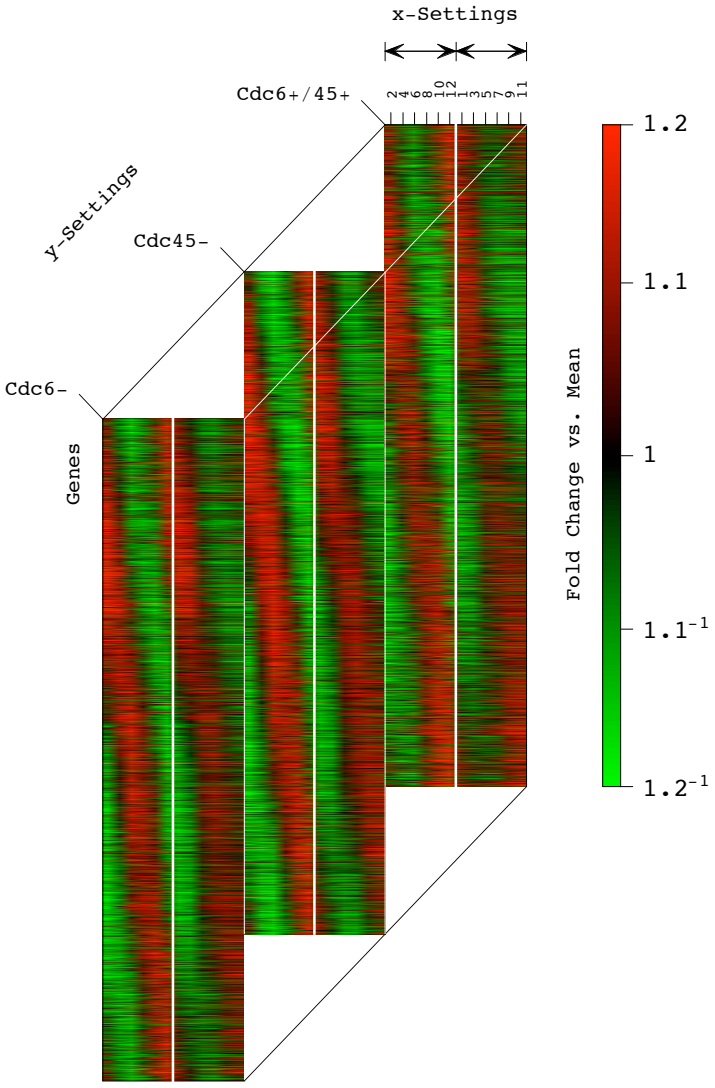
```
length = 0.0125;
colorscale = Table[
   If[k > 60, RGBColor[(k - 60) / 60, 0, 0], RGBColor[0, (60 - k) / 60, 0]],
   {j, 1, 1}, {k, 1, 120}];
legend1 = {
  Graphics[RasterArray[colorscale, {{0, 5 + 4 * 7}, {12, 6 + 4 * 7}}]],
  Graphics[Line[{{0, 5 + 4 * 7}, {0, 6 + 4 * 7}, {12, 6 + 4 * 7}, {12, 5 + 4 * 7}, {0, 5 + 4 * 7}}]],
  Graphics[Line[{{-6 + 6, 5 + 4 * 7}, {-6 + 6, 4.65 + 4 * 7}}]],
  Graphics[Line[{{Log[2, 1 / 1.5] * 6 + 6, 5 + 4 * 7}, {Log[2, 1 / 1.5] * 6 + 6, 4.65 + 4 * 7}}]],
  Graphics[Line[{{6, 5 + 4 * 7}, {6, 4.65 + 4 * 7}}]],
  Graphics[Line[{{Log[2, 1.5] * 6 + 6, 5 + 4 * 7}, {Log[2, 1.5] * 6 + 6, 4.65 + 4 * 7}}]],
  Graphics[Line[{{6 + 6, 5 + 4 * 7}, {6 + 6, 4.65 + 4 * 7}}]],
  Graphics[Text[StyleForm["2^-1", FontSize -> 12], {-6 + 6, 4.45 + 4 * 7}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1.5^-1", FontSize -> 12], {Log[2, 1 / 1.5] * 6 + 6, 4.45 + 4 * 7}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1 ", FontSize -> 12], {6, 4.45 + 4 * 7}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1.5 ", FontSize -> 12], {Log[2, 1.5] * 6 + 6, 4.45 + 4 * 7}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["2 ", FontSize -> 12], {6 + 6, 4.45 + 4 * 7}, {0, 1}, {1, 0}]],
  Graphics[Text["Fold Change vs. Mean", {6.25, 2.5 + 4 * 7}]]};
genes1 = {
   Table[Graphics[RasterArray[color1[[n - 7]], {{0, 3 + 4 * n}, {12, 6 + 4 * n}}]], {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[{RGBColor[1, 1, 1], Thickness[0.0033], Line[{{6, 3 + 4 * n}, {6, 6 + 4 * n}}]}],
    {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[Line[{{0, 3 + 4 * n}, {0, 6 + 4 * n}, {12, 6 + 4 * n}, {12, 3 + 4 * n}, {0, 3 + 4 * n}}]],
    {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[Line[{{0, 6.5 + 4 * n - a}, {-0.2, 6.5 + 4 * n - a}}]],
    {a, 1, yarrays}, {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[Text[yarraynames[[a]], {-0.5, 6.5 + 4 * n - a}, {1, 0}]],
    {a, 1, yarrays}, {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[Line[{{12, 6 + 4 * n}, {13, 5 + 4 * n}}]], {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[Text[StyleForm[genelist1[[n - 7, 2]], FontSize → 12], {13, 4.5 + 4 * n}, {-1, 0}]],
    {n, 1 + 7, dimension1 + 7}],
   Table[Graphics[Line[{{a - 0.5, 6 + 4 * dimension2}, {a - 0.5, 6.3 + 4 * dimension2}}]], {a, 1, xarrays}],
   Table[Graphics[Text[xarraynames[[a]], {a - 0.5, 6.4 + 4 * dimension2}, {0, -1}, {0, 1}]],
    {a, 1, xarrays}],
   Graphics[Text["(a) x-Settings", {6.25, 10 + 4 * dimension2}]],
   arrows[0, 6, 12, 8.5 + 4 * dimension2, 0.5],
   Graphics[Text["y-Settings", {-5.5, 53.5}, {0, -1}, {0, 1}]]};
```

```
legend2 = {
  Graphics[RasterArray[colorscale, {{22, 5}, {34, 6}}]],
  Graphics[Line[{{22, 5}, {22, 6}, {34, 6}, {34, 5}, {22, 5 + 4 * 0}}]],
  Graphics[Line[{{-6 + 28, 5}, {-6 + 28, 4.65}}]],
  Graphics[Line[{{Log[2, 1 / 1.2] * 6 / Log[2, 1.4] + 28, 5}, {Log[2, 1 / 1.2] * 6 / Log[2, 1.4] + 28, 4.65}}]],
  Graphics[Line[{{28, 5}, {28, 4.65}}]],
  Graphics[Line[{{Log[2, 1.2] * 6 / Log[2, 1.4] + 28, 5}, {Log[2, 1.2] * 6 / Log[2, 1.4] + 28, 4.65}}]],
  Graphics[Line[{{6 + 28, 5}, {6 + 28, 4.65}}]],
  Graphics[Text[StyleForm["1.4^-1", FontSize -> 12], {-6 + 28, 4.45}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1.2^-1", FontSize -> 12],
    {Log[2, 1 / 1.2] * 6 / Log[2, 1.4] + 28, 4.45}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1 ", FontSize -> 12], {28, 4.45}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1.2 ", FontSize -> 12],
    {Log[2, 1.2] * 6 / Log[2, 1.4] + 28, 4.45}, {0, 1}, {1, 0}]],
  Graphics[Text[StyleForm["1.4 ", FontSize -> 12], {6 + 28, 4.45}, {0, 1}, {1, 0}]],
  Graphics[Text["Fold Change vs. Mean", {28.25, 2.5}]]};
genes2 = {
   Table[Graphics[RasterArray[color2[[n]], {{22, 3 + 4 * n}, {34, 6 + 4 * n}}]], {n, 1, dimension2}],
   Table[Graphics[{RGBColor[1, 1, 1], Thickness[0.0033], Line[{{28, 3 + 4 * n}, {28, 6 + 4 * n}}]}],
    {n, 1, dimension2}],
   Table[Graphics[Line[{{22, 3 + 4 * n}, {22, 6 + 4 * n}, {34, 6 + 4 * n}, {34, 3 + 4 * n}, {22, 3 + 4 * n}}]],
    {n, 1, dimension2}],
   Table[Graphics[Line[{{22, 6.5 + 4 * n - a}, {21.8, 6.5 + 4 * n - a}}]],
    {a, 1, yarrays}, {n, 1, dimension2}],
   Table[Graphics[Text[yarraynames[[a]], {21.5, 6.5 + 4 * n - a}, {1, 0}]],
    {a, 1, yarrays}, {n, 1, dimension2}],
   Table[Graphics[Line[{{34, 6 + 4 * n}, {35, 5 + 4 * n}}]], {n, 1, dimension2}],
   Table[Graphics[Text[StyleForm[
       ColumnForm[{" ", genelist2[[n, 2]], genelist2[[n, 3]]}, Left],
       FontSize -> 12], {35, 4.5 + 4 * n}, {-1, 0}]], {n, 1, dimension2}],
   Table[Graphics[Line[{{a + 21.5, 6 + 4 * dimension2}, {a + 21.5, 6.3 + 4 * dimension2}}]],
    {a, 1, xarrays}],
   Table[Graphics[Text[xarraynames[[a]], {a + 21.5, 6.4 + 4 * dimension2}, {0, -1}, {0, 1}]],
    {a, 1, xarrays}],
   Graphics[Text["(b) x-Settings", {28.25, 10 + 4 * dimension2}]],
   arrows[22, 28, 34, 8.5 + 4 * dimension2, 0.5],
   Graphics[Text["y-Settings", {16.5, 53.5}, {0, -1}, {0, 1}]]};

(* Display Selected Genes Raster Displays *)

g = Show[{legend1, genes1, legend2, genes2},
   AspectRatio -> 1.3,
   PlotRange → {{-6.5, 39.5}, {2, 75}}];
```


(a) x-Settings
2 4 6 8 10 12 1 3 5 7 9 11
y-Settings
Cdc6-
Cdc45-
Cdc6+/45+
HTA1
HTA2
HTB1
HTB2
HHF1
HHF2
HHT1
HHT2
HHO1
2^-1 1.5^-1 1 1.5 2
Fold Change vs. Mean
(b) x-Settings
EMP46 ARS1212
YML119W ARS1303
YCL049C ARS305
PAU8 ARS102
PHO89 ARS228.5
ALG14 ARS212
FMP43 ARS733
SMC6 ARS1227.5
MET10 ARS608
SAP4 ARS702
YIL108W ARS911.5
SPF1 ARS508
ADE4 ARS1331.5
VPS8 ARS108
SLX1 ARS221.5
NET1 ARS1010
1.4^-1 1.2^-1 1 1.2 1.4
Fold Change vs. Mean

```
(* Display Sorted Averaged Data Tensor *)

(* Center Data Tensor *)

ctensor = Transpose[Flatten[Transpose[tensor, {1, 3, 2}], 1]];
average = N[Table[1, {a, 1, xarrays * yarrays}] / Sqrt[xarrays * yarrays]];
ctensor = ctensor - N[Outer[Times, Dot[ctensor, average], average]];

(* Sort Data Tensor *)

x = 2;
y = 3;
phase = -ArcTan[TakeColumns[zeigenarrays, {y}] / TakeColumns[zeigenarrays, {x}]] +
   (Sign[TakeColumns[zeigenarrays, {x}]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
stensor = AppendRows[phase, ctensor];
stensor = Sort[stensor, OrderedQ[{#2, #1}] &];
stensor = {
   TakeColumns[stensor, {2, xarrays + 1}],
   TakeColumns[stensor, {xarrays + 2, 2 * xarrays + 1}],
   TakeColumns[stensor, {2 * xarrays + 2, 3 * xarrays + 1}]};
Clear[x, y];

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

length = 0.025;
contrast = 1 / (Log[2, 1.2] / product1 / product2);
displaying = Table[
   If[contrast * stensor[[i, j, k]] > 0,
    If[contrast * stensor[[i, j, k]] < 1, {contrast * stensor[[i, j, k]], 0}, {1, 0}],
    If[contrast * stensor[[i, j, k]] > -1, {0, -contrast * stensor[[i, j, k]]}, {0, 1}]],
   {i, 1, yarrays}, {j, zgenes, 1, -1}, {k, 1, xarrays}];
colortensor[i_] := Table[
   RGBColor[displaying[[i, j, k, 1]], displaying[[i, j, k, 2]], 0],
   {j, 1, zgenes}, {k, 1, xarrays}];
gtensor = {
   Graphics[RasterArray[colortensor[1], {{0, -18}, {24, 36}}]],
   Graphics[RasterArray[colortensor[2], {{24, -6}, {48, 48}}]],
   Graphics[RasterArray[colortensor[3], {{48, 6}, {72, 60}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{24, -18}, {0, -18}, {0, 36}, {48, 60},
       {72, 60}, {72, 6}, {24, -18}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{72, 60}, {24, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{24, -6}, {24, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{48, -6}, {48, 48}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0044], Line[{{12, -18}, {12, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0044], Line[{{36, -6}, {36, 48}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0044], Line[{{60, 6}, {60, 60}}]}],
   Table[Graphics[Line[{{24 * a - 24, 24 + 12 * a}, {24 * a - 28, 26 + 12 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[yarraynames[[a]], {24 * a - 29, 26.5 + 12 * a}, {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Settings", {-2, 53}, {0, 0}, {1, 1}]],
   Table[Graphics[Line[{{2 * a - 1, 60}, {2 * a - 1, 61}}]], {a, 25, 36}],
   Table[Graphics[
     Text[StyleForm[xarraynames[[a]], FontSize -> 7], {2 * a + 47, 61.75}, {0, -1}, {0, 1}]], {a, 1, 12}],
   arrows[48, 60, 72, 66, 0.75],
   Graphics[Text["x-Settings", {60, 69}]],
   Graphics[Text["Genes", {-5, 31.5}, {0, 0}, {0, 1}]]};
```

```
(* Display Data Tensor *)

colorscale = Table[
   If[j > 60, RGBColor[(j - 60) / 60, 0, 0], RGBColor[0, (60 - j) / 60, 0]],
   {j, 1, 120}, {k, 1, 1}];
g = Show[{
    gtensor,
    Graphics[Text["Fold Change vs. Mean", {80, 31.5}, {0, 0}, {0, 1}]],
    Graphics[RasterArray[colorscale, {{85, 6}, {88, 60}}]],
    Graphics[Line[{{85, 6}, {88, 6}, {88, 60}, {85, 60}, {85, 6}}]],
    Graphics[Line[{{88, 60}, {90, 60}}]],
    Graphics[Line[
      {{88, Log[2, 1.1] * 27 / Log[2, 1.2] + 33}, {90, Log[2, 1.1] * 27 / Log[2, 1.2] + 33}}]],
    Graphics[Line[{{88, 33}, {90, 33}}]],
    Graphics[Line[
      {{88, Log[2, 1 / 1.1] * 27 / Log[2, 1.2] + 33}, {90, Log[2, 1 / 1.1] * 27 / Log[2, 1.2] + 33}}]],
    Graphics[Line[{{88, 6}, {90, 6}}]],
    Graphics[Text[StyleForm["1.2", FontSize -> 12], {91, 60}, {-1, 0}, {1, 0}]],
    Graphics[Text[StyleForm["1.1", FontSize -> 12],
      {91, Log[2, 1.1] * 27 / Log[2, 1.2] + 33}, {-1, 0}, {1, 0}]],
    Graphics[Text[StyleForm["1", FontSize -> 12], {91, 33}, {-1, 0}, {1, 0}]],
    Graphics[Text[StyleForm["1.1^-1", FontSize -> 12],
      {91, Log[2, 1 / 1.1] * 27 / Log[2, 1.2] + 33}, {-1, 0}, {1, 0}]],
    Graphics[Text[StyleForm["1.2^-1", FontSize -> 12], {91, 6}, {-1, 0}, {1, 0}]]
   }, PlotRange → {{-20, 105}, {-19, 71}},
   AspectRatio → 1.5];
```




```
(* Save Averaged and Sorted Data Tensor *)

arraynames = Table[StringJoin[
    yarraynames[[IntegerPart[a / (xarrays + 0.5) + 1]]], "_", ToString[zarraynames[[a]]]],
   {a, 1, xarrays * yarrays}];
arraynames = AppendRows[{{"Angular_Distance", "SGD_YORF"}}, {arraynames}];

(* Sort Data Tensor *)

x = 2;
y = 3;
phase = -ArcTan[TakeColumns[zeigenarrays, {y}] / TakeColumns[zeigenarrays, {x}]] +
   (Sign[TakeColumns[zeigenarrays, {x}]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
stensor = AppendRows[phase, genenames,
   N[Round[Transpose[Flatten[Transpose[tensor, {1, 3, 2}], 1]] * 100000 * product1 * product2] / 100000]];
stensor = Sort[stensor, OrderedQ[{#2, #1}] &];
phase = N[Round[TakeColumns[stensor, {1}] / Pi * 100000] / 100000];
stensor = AppendRows[phase, TakeColumns[stensor, {2, xarrays * yarrays + 2}]];
averaged = AppendColumns[arraynames, stensor];
Clear[x, y];
stream = OpenWrite["Desktop/96/Data/Averaged_Data.txt", PageWidth -> Infinity];
Write[stream, OutputForm[TableForm[averaged, TableSpacing -> {0, 1}]]];
Close[stream];
```

```
(* Display Reformulated HOSVD of the Data Tensor  *)

(* Display Data Tensor *)

(* Sort Data Tensor *)

space = 18;
x = 2;
y = 3;
phase = -ArcTan[TakeColumns[zeigenarrays, {y}] / TakeColumns[zeigenarrays, {x}]] +
   (Sign[TakeColumns[zeigenarrays, {x}]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
stensor = AppendRows[phase, Transpose[Flatten[Transpose[tensor, {1, 3, 2}], 1]]];
stensor = Sort[stensor, OrderedQ[{#2, #1}] &];
stensor = {
   TakeColumns[stensor, {2, xarrays + 1}],
   TakeColumns[stensor, {xarrays + 2, 2 * xarrays + 1}],
   TakeColumns[stensor, {2 * xarrays + 2, 3 * xarrays + 1}]};
Clear[x, y];

(* x-Center Data Tensor *)

average = N[Table[1, {a, 1, xarrays}] / Sqrt[xarrays]];
ctensor = stensor - N[Outer[Times, Dot[stensor, average], average]];

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

length = 0.00625;
contrast = 125;
displaying = Table[
   If[contrast * ctensor[[i, j, k]] > 0,
    If[contrast * ctensor[[i, j, k]] < 1, {contrast * ctensor[[i, j, k]], 0}, {1, 0}],
    If[contrast * ctensor[[i, j, k]] > -1, {0, -contrast * ctensor[[i, j, k]]}, {0, 1}]],
   {i, 1, yarrays}, {j, zgenes, 1, -1}, {k, 1, xarrays}];
colortensor[i_] := Table[
   RGBColor[displaying[[i, j, k, 1]], displaying[[i, j, k, 2]], 0],
   {j, 1, zgenes}, {k, 1, xarrays}];
gtensor = {
   Graphics[RasterArray[colortensor[1], {{0, -18}, {24, 36}}]],
   Graphics[RasterArray[colortensor[2], {{24, -6}, {48, 48}}]],
   Graphics[RasterArray[colortensor[3], {{48, 6}, {72, 60}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{24, -18}, {0, -18}, {0, 36}, {48, 60},
       {72, 60}, {72, 6}, {24, -18}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{72, 60}, {24, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{24, -6}, {24, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{48, -6}, {48, 48}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011], Line[{{12, -18}, {12, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011], Line[{{36, -6}, {36, 48}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011], Line[{{60, 6}, {60, 60}}]}],
   Table[Graphics[Line[{{24 * a - 24, 24 + 12 * a}, {24 * a - 28, 26 + 12 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[yarraynames[[a]], {24 * a - 29, 26.5 + 12 * a}, {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Settings", {-2, 53}, {0, 0}, {1, 1}]],
   Table[Graphics[Line[{{2 * a - 1, 60}, {2 * a - 1, 61}}]], {a, 25, 36}],
   Table[Graphics[
     Text[StyleForm[xarraynames[[a]], FontSize -> 7], {2 * a + 47, 61.75}, {0, -1}, {0, 1}]], {a, 1, 12}],
   arrows[48, 60, 72, 66, 0.75],
   Graphics[Text["x-Settings", {60, 69}]],
   Graphics[Text["Genes", {-5, 31.5}, {0, 0}, {0, 1}]]};
```

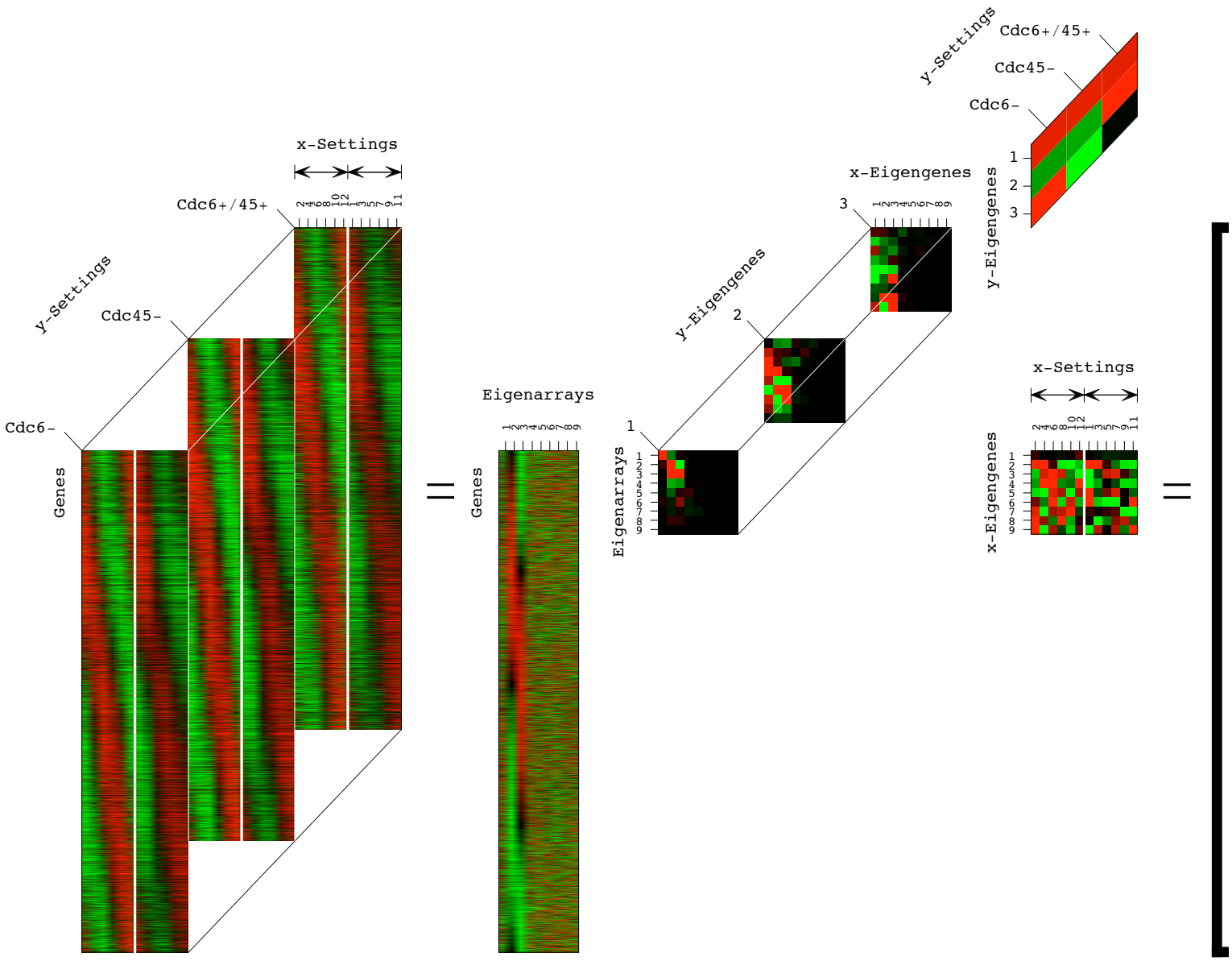
```
(* Display Eigenarrays *)

(* Sort Eigenarrays *)

x = 2;
y = 3;
phase = -ArcTan[TakeColumns[zeigenarrays, {y}] / TakeColumns[zeigenarrays, {x}]] +
   (Sign[TakeColumns[zeigenarrays, {x}]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
matrix = AppendRows[phase, zeigenarrays];
matrix = Sort[matrix, OrderedQ[{#2, #1}] &];
matrix = TakeColumns[matrix, {2, Dimensions[zeigenarrays][[2]] + 1}]; Clear[x, y];

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

contrast = 75;
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, zgenes, 1, -1}, {j, 1, Dimensions[zeigenarrays][[2]]}];
colorz = Table[RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
   {i, 1, zgenes}, {j, 1, Dimensions[zeigenarrays][[2]]}];
gzeigenarrays = {
   Graphics[RasterArray[colorz, {{space + 76, -18}, {space + 94, 36}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{space + 76, -18}, {space + 76, 36}, {space + 94, 36},
       {space + 94, -18}, {space + 76, -18}}]}],
   Table[Graphics[Line[{{2 * a + space + 75.5, 36}, {2 * a + space + 75.5, 36 + 1}}]], {a, 1, 9}],
   Table[Graphics[Text[StyleForm[a, FontSize -> 7], {2 * a + space + 75.5, 37.75},
      {0, -1}, {0, 1}]], {a, 1, 9}],
   Graphics[Text["Eigenarrays", {space + 85, 42}]],
   Graphics[Text["Genes", {space + 71, 31.5}, {0, 0}, {0, 1}]]};

(* Display Core *)

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

contrast = 5;
displaying = Table[
   If[contrast * core[[i, j, k]] > 0,
    If[contrast * core[[i, j, k]] < 1, {contrast * core[[i, j, k]], 0}, {1, 0}],
    If[contrast * core[[i, j, k]] > -1, {0, -contrast * core[[i, j, k]]}, {0, 1}]],
   {i, 1, yarrays}, {j, Dimensions[core][[2]], 1, -1}, {k, 1, Dimensions[core][[3]]}];
colorcore[i_] := Table[RGBColor[displaying[[i, j, k, 1]], displaying[[i, j, k, 2]], 0],
   {j, 1, Dimensions[core][[2]]}, {k, 1, Dimensions[core][[3]]}];
gcore = {
   Graphics[RasterArray[colorcore[1], {{2 * space + 94, 27}, {2 * space + 112, 36}}]],
   Graphics[RasterArray[colorcore[2], {{2 * space + 118, 39}, {2 * space + 136, 48}}]],
   Graphics[RasterArray[colorcore[3], {{2 * space + 142, 51}, {2 * space + 160, 60}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{2 * space + 94, 36}, {2 * space + 94, 27},
       {2 * space + 112, 27}, {2 * space + 160, 51}, {2 * space + 160, 60}, {2 * space + 142, 60},
       {2 * space + 94, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{2 * space + 142, 51}, {2 * space + 160, 60}}]}],
   Graphics[{RGBColor[0, 0, 0], Line[{{2 * space + 118, 39}, {2 * space + 142, 51}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{2 * space + 118, 39}, {2 * space + 136, 48}}]}],
   Graphics[{RGBColor[0, 0, 0], Line[{{2 * space + 112, 36}, {2 * space + 118, 39}}]}],
   Table[Graphics[Line[{{2 * space + 70 + 24 * a, 24 + 12 * a}, {2 * space + 66 + 24 * a, 26 + 12 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[a, {2 * space + 65 + 24 * a, 26.5 + 12 * a}, {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Eigengenes", {2 * space + 108, 53}, {0, 0}, {1, 1}]],
   Table[Graphics[Line[{{2 * space + 141 + 2 * a, 60}, {2 * space + 141 + 2 * a, 61}}]], {a, 1, 9}],
   Table[Graphics[Text[StyleForm[a, FontSize -> 7],
      {2 * space + 141 + 2 * a, 61.75}, {0, -1}, {0, 1}]], {a, 1, 9}],
   Graphics[Text["x-Eigengenes", {2 * space + 151, 66}]],
   Table[Graphics[Line[{{2 * space + 94, 26.5 + a},
       {2 * space + 92.25, 26.5 + a}}]], {a, 1, 9}],
   Table[Graphics[Text[StyleForm[a, FontSize -> 7],
      {2 * space + 91.25, 36.5 - a}, {1, 0}]], {a, 1, 9}],
   Graphics[Text["Eigenarrays", {2 * space + 85, 31.5 - 1}, {0, 0}, {0, 1}]]};
```

```
(* Display x-Eigengenes *)

(* x-Center x-Eigengenes *)

matrix = xeigengenes;
average = N[Table[1, {a, 1, xarrays}] / Sqrt[xarrays]];
matrix = matrix - N[Outer[Times, Dot[matrix, average], average]];

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

length = 0.00625;
contrast = 3.5;
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, Dimensions[xeigengenes][[1]], 1, -1}, {j, 1, xarrays}];
colorx = Table[
   RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
   {i, 1, Dimensions[xeigengenes][[1]]}, {j, 1, xarrays}];
gxeigengenes = {
   Graphics[RasterArray[colorx, {{3 * space + 160, 27}, {3 * space + 184, 36}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{3 * space + 160, 27}, {3 * space + 160, 36},
       {3 * space + 184, 36}, {3 * space + 184, 27}, {3 * space + 160, 27}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011], Line[{{3 * space + 172, 27}, {3 * space + 172, 36}}]}],
   Table[Graphics[Line[{{3 * space + 159 + 2 * a, 36}, {3 * space + 159 + 2 * a, 37}}]], {a, 1, 12}],
   Table[Graphics[Text[StyleForm[xarraynames[[a]], FontSize -> 7], {3 * space + 159 + 2 * a, 37.75},
      {0, -1}, {0, 1}]], {a, 1, 12}],
   arrows[3 * space + 160, 3 * space + 172, 3 * space + 184, 42, 0.75],
   Graphics[Text["x-Settings", {3 * space + 172, 45}]],
   Table[Graphics[Line[{{3 * space + 160, 26.5 + a}, {3 * space + 158.25, 26.5 + a}}]], {a, 1, 9}],
   Table[Graphics[Text[StyleForm[a, FontSize -> 7], {3 * space + 157.25, 36.5 - a}, {1, 0}]], {a, 1, 9}],
   Graphics[Text["x-Eigengenes", {3 * space + 151, 31.5}, {0, 0}, {0, 1}]]};

(* Display y-Eigengenes *)

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

contrast = 1.5;
displaying = Table[
   If[contrast * yeigengenes[[i, j]] > 0,
    If[contrast * yeigengenes[[i, j]] < 1, {contrast * yeigengenes[[i, j]], 0}, {1, 0}],
    If[contrast * yeigengenes[[i, j]] > -1, {0, -contrast * yeigengenes[[i, j]]}, {0, 1}]],
   {i, yarrays, 1, -1}, {j, 1, yarrays}];
colory[x_, y_, dx_, dy_] := Table[{
    RGBColor[displaying[[i, j, 1]], displaying[[i, j, 2]], 0],
    Polygon[{
      {x + (j - 1) * dx, y + 0.5 * (j - 1) * dx + (i - 1) * dy},
      {x + j * dx, y + 0.5 * j * dx + (i - 1) * dy},
      {x + j * dx, y + 0.5 * j * dx + i * dy},
      {x + (j - 1) * dx, y + .5 * (j - 1) * dx + i * dy},
      {x + (j - 1) * dx, y + 0.5 * (j - 1) * dx + (i - 1) * dy}}]},
   {i, 1, yarrays}, {j, 1, yarrays}];
gyeigengenes = {
   Graphics[colory[3 * space + 160, 60, 8, 3]],
   Graphics[{RGBColor[0, 0, 0], Line[{{3 * space + 160, 60},
       {3 * space + 160, 69}, {3 * space + 184, 81}, {3 * space + 184, 72}, {3 * space + 160, 60}}]}],
   Table[Graphics[Line[{{3 * space + 160, 58.5 + 3 * a}, {3 * space + 158.25, 58.5 + 3 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[a, {3 * space + 157.25, 70.5 - 3 * a}, {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Eigengenes", {3 * space + 151, 60}, {0, 0}, {0, 1}]],
   Table[Graphics[Line[{{3 * space + 155.5 + 8 * a, 66.75 + 4 * a},
       {3 * space + 151.5 + 8 * a, 68.75 + 4 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[yarraynames[[a]], {3 * space + 150.5 + 8 * a, 69.25 + 4 * a},
      {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Settings", {3 * space + 145.5 - 2, 79.75}, {0, 0}, {1, 1}]]};
```

```
(* Display Second Subtensor *)

subtensor = Outer[Times, yeigengenes[[1]], Flatten[TakeColumns[rzeigenarrays, {2}]], xeigengenes[[2]]];

(* Sort Subtensor *)

x = 2;
y = 3;
phase = -ArcTan[TakeColumns[zeigenarrays, {y}] / TakeColumns[zeigenarrays, {x}]] +
   (Sign[TakeColumns[zeigenarrays, {x}]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
stensor = AppendRows[phase, Transpose[Flatten[Transpose[subtensor, {1, 3, 2}], 1]]];
stensor = Sort[stensor, OrderedQ[{#2, #1}] &];
stensor = {
   TakeColumns[stensor, {2, xarrays + 1}],
   TakeColumns[stensor, {xarrays + 2, 2 * xarrays + 1}],
   TakeColumns[stensor, {2 * xarrays + 2, 3 * xarrays + 1}]};
Clear[x, y];

(* x-Center Subtensor *)

average = N[Table[1, {a, 1, xarrays}] / Sqrt[xarrays]];
ctensor = stensor - N[Outer[Times, Dot[stensor, average], average]];

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

length = 0.00625;
contrast = 350;
displaying = Table[
   If[contrast * ctensor[[i, j, k]] > 0,
    If[contrast * ctensor[[i, j, k]] < 1, {contrast * ctensor[[i, j, k]], 0}, {1, 0}],
    If[contrast * ctensor[[i, j, k]] > -1, {0, -contrast * ctensor[[i, j, k]]}, {0, 1}]],
   {i, 1, yarrays}, {j, zgenes, 1, -1}, {k, 1, xarrays}];
colortensor[i_] := Table[
   RGBColor[displaying[[i, j, k, 1]], displaying[[i, j, k, 2]], 0],
   {j, 1, zgenes}, {k, 1, xarrays}];
gsubtensor1 = {
   Graphics[RasterArray[colortensor[1], {{5.625 * space + 189, -18}, {5.625 * space + 213, 36}}]],
   Graphics[RasterArray[colortensor[2], {{5.625 * space + 213, -6}, {5.625 * space + 237, 48}}]],
   Graphics[RasterArray[colortensor[3], {{5.625 * space + 237, 6}, {5.625 * space + 261, 60}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{5.625 * space + 213, -18}, {5.625 * space + 189, -18},
       {5.625 * space + 189, 36}, {5.625 * space + 237, 60}, {5.625 * space + 261, 60},
       {5.625 * space + 261, 6}, {5.625 * space + 213, -18}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{5.625 * space + 261, 60}, {5.625 * space + 213, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{5.625 * space + 213, -6}, {5.625 * space + 213, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{5.625 * space + 237, -6}, {5.625 * space + 237, 48}}]}],
   Graphics[
    {RGBColor[1, 1, 1], Thickness[0.0011], Line[{{5.625 * space + 201, -18}, {5.625 * space + 201, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011],
     Line[{{5.625 * space + 225, -6}, {5.625 * space + 225, 48}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011],
     Line[{{5.625 * space + 249, 6}, {5.625 * space + 249, 60}}]}],
   Table[Graphics[Line[{{5.625 * space + 165 + 24 * a, 24 + 12 * a},
       {5.625 * space + 161 + 24 * a, 26 + 12 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[yarraynames[[a]], {5.625 * space + 160 + 24 * a, 26.5 + 12 * a}, {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Settings", {5.625 * space + 187, 53}, {0, 0}, {1, 1}]],
   Table[Graphics[Line[{{5.625 * space + 188 + 2 * a, 60}, {5.625 * space + 188 + 2 * a, 61}}]], {a, 25, 36}],
   Table[Graphics[Text[StyleForm[xarraynames[[a]], FontSize -> 7], {5.625 * space + 236 + 2 * a, 61.75},
      {0, -1}, {0, 1}]], {a, 1, 12}],
   arrows[5.625 * space + 237, 5.625 * space + 249, 5.625 * space + 261, 66, 0.75],
   Graphics[Text["x-Settings", {5.625 * space + 249, 69}]],
   Graphics[Text["Genes", {5.625 * space + 184, 31.5}, {0, 0}, {0, 1}]]};
```

```
(* Display Third Subtensor *)

subtensor = Outer[Times, yeigengenes[[1]], Flatten[TakeColumns[rzeigenarrays, {3}]], xeigengenes[[3]]];

(* Sort Subtensor *)

x = 2;
y = 3;
phase = -ArcTan[TakeColumns[zeigenarrays, {y}] / TakeColumns[zeigenarrays, {x}]] +
   (Sign[TakeColumns[zeigenarrays, {x}]] + Table[{1}, {a, 1, zgenes}]) * Pi / 2.;
stensor = AppendRows[phase, Transpose[Flatten[Transpose[subtensor, {1, 3, 2}], 1]]];
stensor = Sort[stensor, OrderedQ[{#2, #1}] &];
stensor = {
   TakeColumns[stensor, {2, xarrays + 1}],
   TakeColumns[stensor, {xarrays + 2, 2 * xarrays + 1}],
   TakeColumns[stensor, {2 * xarrays + 2, 3 * xarrays + 1}]};
Clear[x, y];

(* x-Center Subtensor *)

average = N[Table[1, {a, 1, xarrays}] / Sqrt[xarrays]];
ctensor = stensor - N[Outer[Times, Dot[stensor, average], average]];

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

length = 0.00625;
contrast = 350;
displaying = Table[
   If[contrast * ctensor[[i, j, k]] > 0,
    If[contrast * ctensor[[i, j, k]] < 1, {contrast * ctensor[[i, j, k]], 0}, {1, 0}],
    If[contrast * ctensor[[i, j, k]] > -1, {0, -contrast * ctensor[[i, j, k]]}, {0, 1}]],
   {i, 1, yarrays}, {j, zgenes, 1, -1}, {k, 1, xarrays}];
colortensor[i_] := Table[
   RGBColor[displaying[[i, j, k, 1]], displaying[[i, j, k, 2]], 0],
   {j, 1, zgenes}, {k, 1, xarrays}];
gsubtensor2 = {
   Graphics[RasterArray[colortensor[1], {{6.75 * space + 261, -18}, {6.75 * space + 285, 36}}]],
   Graphics[RasterArray[colortensor[2], {{6.75 * space + 285, -6}, {6.75 * space + 309, 48}}]],
   Graphics[RasterArray[colortensor[3], {{6.75 * space + 309, 6}, {6.75 * space + 333, 60}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{6.75 * space + 285, -18},
       {6.75 * space + 261, -18}, {6.75 * space + 261, 36}, {6.75 * space + 309, 60},
       {6.75 * space + 333, 60}, {6.75 * space + 333, 6}, {6.75 * space + 285, -18}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{6.75 * space + 333, 60}, {6.75 * space + 285, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{6.75 * space + 285, -6}, {6.75 * space + 285, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Line[{{6.75 * space + 309, -6}, {6.75 * space + 309, 48}}]}],
   Graphics[
    {RGBColor[1, 1, 1], Thickness[0.0011], Line[{{6.75 * space + 273, -18}, {6.75 * space + 273, 36}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011], Line[{{6.75 * space + 297, -6}, {6.75 * space + 297, 48}}]}],
   Graphics[{RGBColor[1, 1, 1], Thickness[0.0011], Line[{{6.75 * space + 321, 6}, {6.75 * space + 321, 60}}]}],
   Table[Graphics[Line[{{6.75 * space + 261 + 24 * a - 24, 24 + 12 * a},
       {6.75 * space + 261 + 24 * a - 28, 26 + 12 * a}}]], {a, 1, 3}],
   Table[Graphics[Text[yarraynames[[a]], {6.75 * space + 232 + 24 * a, 26.5 + 12 * a}, {1, 0}]], {a, 1, 3}],
   Graphics[Text["y-Settings", {6.75 * space + 259, 53}, {0, 0}, {1, 1}]],
   Table[Graphics[Line[{{6.75 * space + 260 + 2 * a, 60}, {6.75 * space + 260 + 2 * a, 61}}]], {a, 25, 36}],
   Table[Graphics[Text[StyleForm[xarraynames[[a]], FontSize -> 7],
      {6.75 * space + 308 + 2 * a, 61.75}, {0, -1}, {0, 1}]], {a, 1, 12}],
   arrows[6.75 * space + 309, 6.75 * space + 321, 6.75 * space + 333, 66, 0.75],
   Graphics[Text["x-Settings", {6.75 * space + 321, 69}]],
   Graphics[Text["Genes", {6.75 * space + 256, 31.5}, {0, 0}, {0, 1}]]};
```

```
(* Display Reformulated Core Tensor *)

rcore = {Sqrt[rfractions[[2]]], Sqrt[rfractions[[3]]]};
contrast = 5;
colorrcore = Table[If[contrast * rcore[[i]] < 1, RGBColor[contrast * rcore[[i]], 0, 0],
    RGBColor[1, 0, 0]], {i, 1, 2}];
grcore = {
   Graphics[RasterArray[{{colorrcore[[1]]}}, {{9 * space + 338, 31}, {9 * space + 348, 36}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{9 * space + 338, 31}, {9 * space + 348, 31},
       {9 * space + 348, 36}, {9 * space + 338, 36}, {9 * space + 338, 31}}]}],
   Graphics[RasterArray[{{colorrcore[[2]]}}, {{9 * space + 338, 24}, {9 * space + 348, 29}}]],
   Graphics[{RGBColor[0, 0, 0], Line[{{9 * space + 338, 24}, {9 * space + 348, 24},
       {9 * space + 348, 29}, {9 * space + 338, 29}, {9 * space + 338, 24}}]}]};

(* Display Reformulated HOSVD *)

equal1 = Graphics[Text[StyleForm["=", FontSize -> 40], {72 + 0.5 * space, 31.5}]];
equal2 = Graphics[Text[StyleForm["=", FontSize -> 40], {3.5 * space + 184, 31.5}]];
left1 = Graphics[{Thickness[0.0044], Line[{{4 * space + 189, -18},
      {4 * space + 184, -18}, {4 * space + 184, 60}, {4 * space + 189, 60}}]}];
hdots1 = Graphics[Text[StyleForm["⋯,", FontSize -> 40], {4.625 * space + 189, 31.5}]];
coma = Graphics[Text[StyleForm[",", FontSize -> 40], {6 * space + 261, 31.5}]];
hdots2 = Graphics[Text[StyleForm[",⋯", FontSize -> 40], {7.5 * space + 333, 31.5}]];
right1 = Graphics[{Thickness[0.0044], Line[{{8.125 * space + 333, -18},
      {8.125 * space + 338, -18}, {8.125 * space + 261 + 72 + 5, 60}, {8.125 * space + 333, 60}}]}];
text1 = Graphics[Text[StyleForm["Subtensors", FontSize -> 12], {6.0625 * space + 261, 75}]];
text2 = Graphics[Text[StyleForm["Subtensors", FontSize -> 12], {8.5 * space + 338, 31.5}, {0, 0}, {0, 1}]];
left2 = Graphics[{Thickness[0.0044], Line[{{8.75 * space + 343, -18},
      {8.75 * space + 338, -18}, {8.75 * space + 338, 60}, {8.75 * space + 343, 60}}]}];
vdots1 = Graphics[Text[StyleForm["⋮", FontSize → 40], {9 * space + 343, 36 + 5}]];
vdots2 = Graphics[Text[StyleForm["⋮", FontSize → 40], {9 * space + 343, 24 - 6}]];
right2 = Graphics[{Thickness[0.0044], Line[{{9.25 * space + 343, -18},
      {9.25 * space + 348, -18}, {9.25 * space + 348, 60}, {9.25 * space + 343, 60}}]}];
g = Show[{
    gtensor, equal1,
    gzeigenarrays, gcore, gxeigengenes, gyeigengenes, equal2,
    left1, hdots1, gsubtensor1, coma, gsubtensor2, hdots2, right1, text1,
    text2, left2, vdots1, grcore, vdots2, right2},
   PlotRange -> All,
   AspectRatio -> 0.4];
```


x-Settings
Cdc6+/45+
y-Settings
Cdc45-
Cdc6-
Genes
Eigenarrays
Genes
y-Eigengenes
x-Eigengenes
Eigenarrays
y-Settings
Cdc6+/45+
Cdc45-
Cdc6-
y-Eigengenes
x-Settings
x-Eigengenes