

01-TriDAMP-Basics

This notebook demonstrates the basic functionality of the *TriDAMP* package using simple simulated datasets. The individual steps are briefly explained. For more information regarding the implementation please refer to the *TriDAMP.nb* (available upon request) notebook.

First, both the *MathDAMP* and *TriDAMP* packages have to be loaded. Please adjust the paths to package files according to their location on your system.

```
<< "/home/baran/math/ms/MathDAMP.1.0.0/MathDAMP.m"
<< "/home/baran/math/ms/TriDAMP.1.0.0/TriDAMP.m"
```

```
MathDAMP version 1.0.0 loaded (2006/04/26)
```

```
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

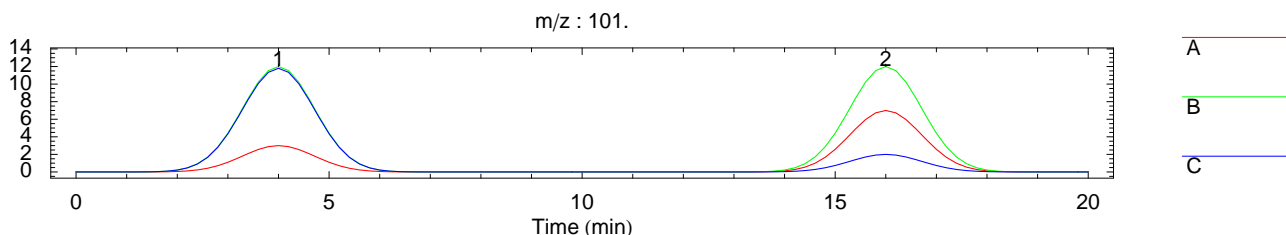
```
TriDAMP version 1.0.0 loaded (2006/05/09)
```

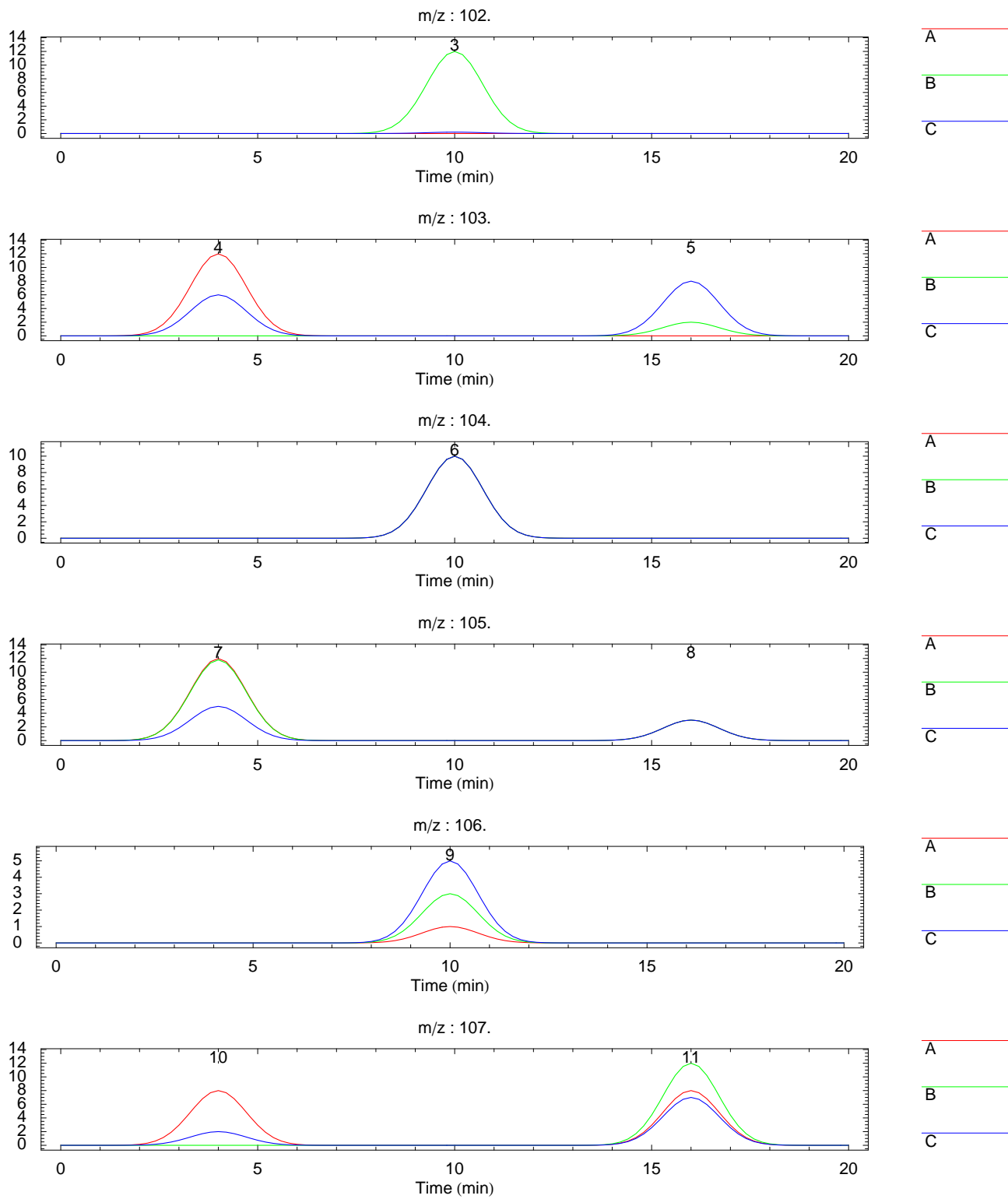
```
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Sample Data

Three simple datasets, named A, B, and C and each containing 7 dummy chromatograms, are generated. A dummy annotation table is also generated to number the peaks for easier reference and orientation on the density plots.

```
dummytimepoints = Range[0, 20, .2];
TMPGaussian[A_, x0_, w_] := A e-((#-x0)/w)2 & /@ dummytimepoints;
peakrts = {{4, 16}, {10}, {4, 16}, {10}, {4, 16}, {10}, {4, 16}};
peakamplitudes = {
  {{3, 7}, {0}, {12, 0}, {10}, {12, 3}, {1}, {8, 8}},
  {{12, 12}, {12}, {0, 2}, {10}, {11.8, 3}, {3}, {0, 12}},
  {{11.8, 2}, {.2}, {6, 8}, {10}, {5, 3}, {5}, {2, 7}}};
dummyints =
  ((Plus @@ (TMPGaussian[Sequence @@ #, 1] & /@ #)) & /@ (Transpose[#] & /@ Transpose[{#, peakrts]})) & /@
  peakamplitudes;
dummymzs = Range[101., 100. + Length[peakrts]];
antbl = MapIndexed[Join[#1, Table[ToString[#2[[1]], {2}], {1}] &,
  (Join @@ ((mzitm = #[[2]]; {mzitm, #} & /@ #[[1]]) & /@ Transpose[{peakrts, dummymzs}])]];
Off[General::"spell1"]; Off[General::"spell"];
{Aset, Bset, Cset} = {#[[1], dummymzs, dummytimepoints, {SampleName -> #[[2]]} & /@
  Transpose[{dummyints, {"A", "B", "C"}}]};
On[General::"spell1"]; On[General::"spell"];
chromplotopts = {LegendData ->
  Transpose[{{Hue[0], Hue[ $\frac{1}{3}$ ], Hue[ $\frac{2}{3}$ ]}, (SampleName /. #) & /@ {Aset, Bset, Cset}][[All, 4]]},
  PlotOptions -> {AspectRatio -> .17, PlotStyle -> {Hue[0], Hue[ $\frac{1}{3}$ ], Hue[ $\frac{2}{3}$ ]}}};
(DAMPPlotChromatogram[{Aset, Bset, Cset}, #1, Sequence @@ chromplotopts,
  AnnotationTable -> antbl] &) /@ Aset[[2]];
```

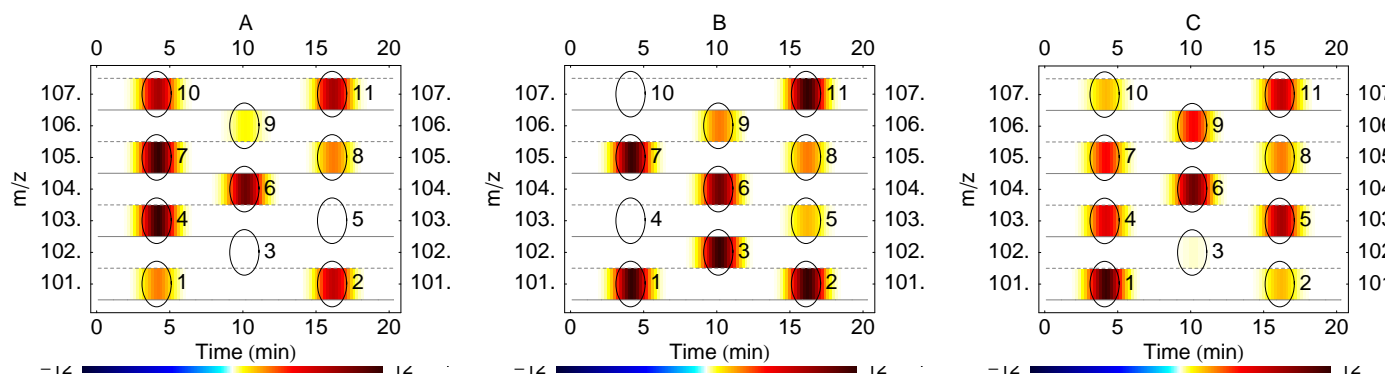




```

DAMPDensityPlot[#, MaxScale -> 12,
  PlotOptions -> {ImageSize -> 300, AspectRatio -> .8, DisplayFunction -> Identity},
  Palette -> DAMPGradientPalette[BackgroundGrayLevel -> 1],
  mzGridLineFreq -> 1, FrameTickFreqs -> {5, 1}, AnnotationTables -> {antbl},
  AnnotationOptions -> {{LabelSize -> {5, .9}, TextStyle -> DAMPTextStyle}}] & /@ {Aset, Bset, Cset};
Show[GraphicsArray[%, GraphicsSpacing -> 0], DisplayFunction -> $DisplayFunction, ImageSize -> 700];

```



3-Way Comparisons

The `TriDAMPCompare` function performs a 3-way comparison between datasets of interest and the `TriDAMPDensityPlot` function displays the results on a density plot. Reference information about the functions can be displayed by executing `?FunctionName`.

? TriDAMPCompare

`TriDAMPCompare[msdatas,options]` performs a three way comparison of `msdatas` (a list of three `msdata` datasets). The datasets are expected to be aligned. The returned dataset has a structure similar to the `msdata` format. However, instead of individual signal intensity values, the signal intensity matrix contains a list of {hue,saturation,brightness} specifications. The saturation specification is not scaled to fall between 0 and 1 (the `TriDAMPDensityPlot` function does this).

Options:

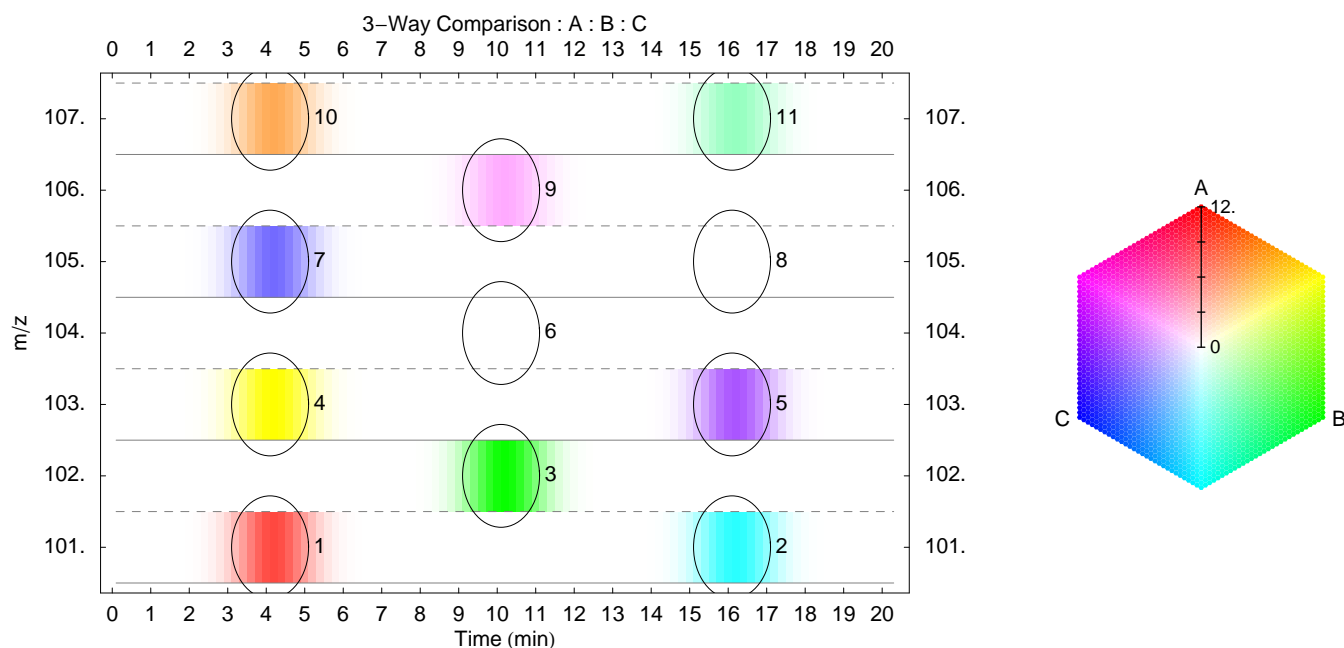
`Relative` - if set to `True`, all three corresponding signal intensities are divided by the biggest of them (or the difference between the two most distant if this is greater) prior to the three way comparison. This produces a relative three way comparison (default: `False`)

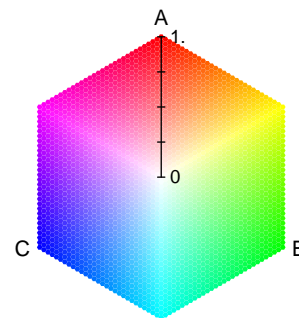
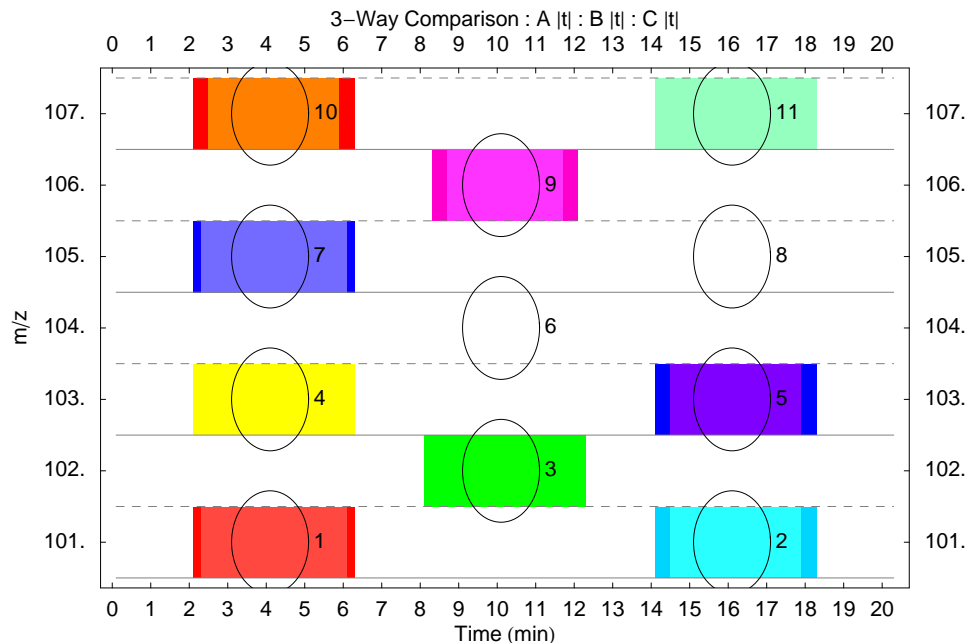
Absolute and relative 3-way comparisons are performed for the A, B, and C datasets. The results are plotted below.

```

absrslt = TriDAMPCompare[{Aset, Bset, Cset}];
relrslt = TriDAMPCompare[DAMPThreshold[#, .1] & /@ {Aset, Bset, Cset}, Relative -> True];
rsltplotopts = {PlotOptions -> {ImageSize -> 700, AspectRatio -> .45}, LegendSize -> .3, LegendOptions ->
  {Identifiers -> ((SampleName /. #) & /@ {Aset, Bset, Cset}[[All, 4]]), AnnotationTables -> {antbl},
  AnnotationOptions -> {{LabelSize -> {5, .9}, TextStyle -> DAMPTextStyle}}};
TriDAMPDensityPlot[#, Sequence@@rsltplotopts] & /@ {absrslt, relrslt};

```





The simulated data are 'perfect' – the gaussian peaks have a nonzero signal intensity over the whole time range. The relative result would therefore provide fully colored lanes if the compared datasets were not subjected to a threshold filtering prior to the 3-way comparison. The filtering is also the cause of the differently colored edges of some of the bands (only one or two of the corresponding signal intensities were leveled to 0 at those time points).

Filtering the Results

The 3-way comparison results datasets can be filtered to select only signals exceeding a certain threshold. Any custom function to filter the data can be specified also.

? TriDAMPFilter

TriDAMPFilter[threewaycomparison,criteria,threshold,options] sets to {0, 0,1} those color specifications in threewaycomparison for which the absolute values of corresponding signal intensities (or saturation values in the respective color specifications) in the criteria dataset are not equal to or greater than threshold.

Options:
 SampleNameSuffix - string to be added to the SampleName from the threewaycomparison to keep track of modifications performed on the dataset (default: "f")

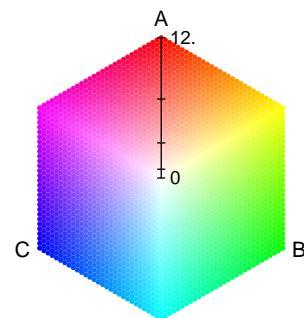
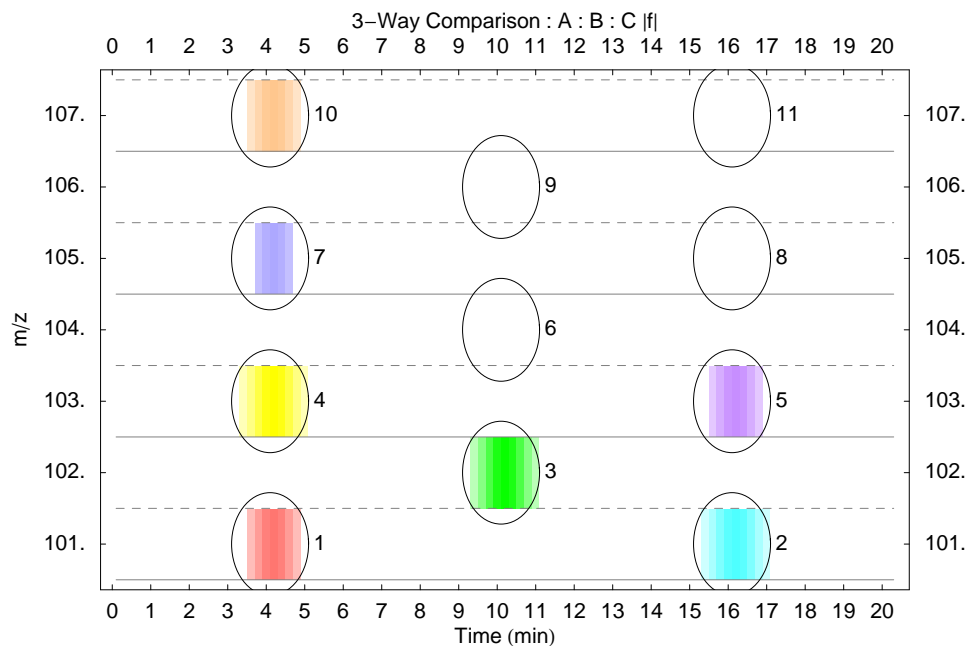
DAMPFilter[threewaycomparison,criteria,filterfunction,options] pure function filterfunction is applied to the signal intensity (or hue specification) matrix of the criteria dataset. For zero signal intensities in the result, the corresponding signal intensities in the threewaycomparison dataset are set to {0,0,1} as well.

Options:
 SampleNameSuffix - string to be added to the SampleName from the threewaycomparison to keep the track of modifications performed on the dataset (default: "f")

All options are passed further to the DAMPAppendToSampleName function (used internally), so the function's ShareBands option may be specified as well.

Below is a density plot of the absolute result from which result intensities (corresponding to saturations in the color specifications) below a threshold value of 7.1 are filtered out. Please note also that the intensity scale was modified to assign a wider saturation range to higher values. The three unlabeled tickmarks on the legend's scale axis indicate the location of the quarters of the range (3, 6, and 9 respectively).

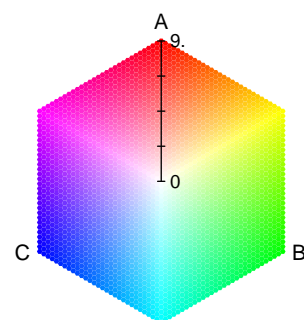
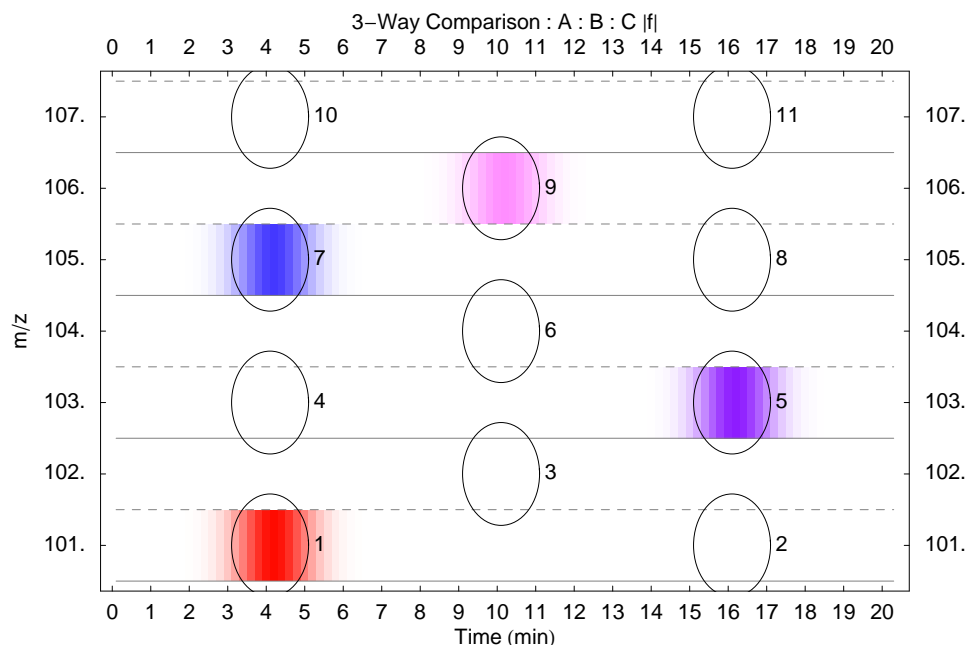
```
TriDAMPDensityPlot[TriDAMPFilter[absrslt, absrslt, 5.01],
  ScaleModifier -> (#^2 &), Sequence@@rsltplotopts];
```



As mentioned previously, a custom function can be defined to filter out results of interest. Below is an example of selecting only those results for which the signal intensities from datasets A and C are the most distant. In other words, only results from the red to blue gradient (running via pink) are selected. The hue specification borderlines are set to tolerate small deviations from this gradient (a strict definition would be $2/3 \leq \#[1] \leq 1 \mid \#[1] = 0$).

The custom filter function is expected to generate a matrix by which the original color specification matrix in the filtered dataset is multiplied. Multiplying any color specification by $\{1, 1, 1\}$ leaves the original values, multiplying by $\{0, 0, 1\}$ changes the original color to white (assuming the brightness specification in the original color is 1).

```
TriDAMPDensityPlot[TriDAMPFilter[absrslt, absrslt,
  (If[1.9/3 ≤ #[1] ≤ 1 || #[1] < .1/3, {1, 1, 1}, {0, 0, 1}] & /@# & /@# &)], Sequence@@rsltplotopts];
```



Candidate Differences

Overlaid chromatograms for the candidate differences from a result of interest can be generated in a descending order of significance using the `TriDAMPPlotCandidates` function. The function is very similar to *MathDAMP*'s `DAMPPlotCandidates`. `DAMPPlotCandidates` is actually used internally and all options passed to `TriDAMPPlotCandidates` are passed further to `DAMPPlotCandidates`.

```

TriDAMPPlotCandidates[{Aset, Bset, Cset}, absrslt, PlotCount -> 8,
  PlotVicinity -> 3 {-1, 1}, DropVicinity -> 3, PlotChromatogramOptions -> {LegendData ->
    Transpose[{{Hue[0], Hue[ $\frac{1}{3}$ ], Hue[ $\frac{2}{3}$ ]}, (SampleName /. #) & /@ {Aset, Bset, Cset}][All, 4]}],
  PlotOptions -> {AspectRatio -> .25}, AnnotationTable -> antbl}];

```

