Affine Transformation of Negative Values for NMR Metabolomics Using the mrbin R Package

Matthias S. Klein

Department of Food Science and Technology, The Ohio State University, 2015 Fyffe Road, Columbus OH 43210

SUPPORTING INFORMATION

Guide for Calculations

The ATNV algorithm can be calculated as follows:

Step 1. Define a noise level for the data set.

Step 2. Select, for a single bin/bucket area, all negative data points in the data set.

Step 3. Find the lowest of the **negative** data points (minTMP)

Step 4. Find the lowest of the **positive** data points (maxTMP)

Step 5. If maxTMP is above the noise level, replace maxTMP by the noise level.

Step 6. For each of the negative values, replace the value as follows:

$$value = \frac{value + maxTMP - minTMP}{(maxTMP - minTMP) * maxTMP * 0.99} + maxTMP * 0.01$$

Using the atnv Function as Part of the Mrbin Workflow

The ATNV algorithm will be automatically called during running mrbin, if you select “Fix negative issues” when asked in mrbin. To start mrbin, follow these steps in R:

```
library(mrbin)
results<-mrbin()
```

Using the atnv Function on Custom Data Sets

The ATNV algorithm can be called as follows:

```
library(mrbin)
NMRdataFixed<-atnv(NMRdata = NMRdata, noiseLevels = noiseLevels)
```

This assumes having a matrix called `NMRdata` in the R workspace, containing binned NMR data, and a vector named `noiseLevels` that contains the noise levels of all spectra.

R Code for Performance Test

```
library(mrbin)
#Read and bin NMR data
results<-mrbin(silent=TRUE,parameters=list(binwidth2D=0.03,binheight=2,
 cropHSQC="Yes",reference2D=c(0.04,-0.04,-2,2),signal_to_noise2D=3,
 noiseRange2d=c(3.3,2.3,90,110),croptopRight=c(0,-1.5),croptopLeft=c(0,3.5),
 cropbottomRight=c(160,6),cropbottomLeft=c(160,10),dimension="2D",
 binMethod="Rectangular bins",binRegion=c(9.5,0.5,10,156),specialBinList=NULL,
 referenceScaling="Yes",removeSolvent="Yes",removeAreas="No",sumBins="No",
 noiseRemoval="Yes",PQNScaling="No",PQNIgnoreSugarArea="Yes",
 PQNsugarArea=c(5.4,3.35,72,100),fixNegatives="Yes",logTrafo="Yes",
 defineGroups="No",PCA="Yes",solventRegion=c(5.1,4.5),noiseThreshold=0.75,
 trimZeros="Yes",PQNminimumFeatures=40,PCAtitlelength=6,createBins="Yes",
 useAsNames="Spectrum titles",saveFiles="Yes",
 outputFileName="C:/Users/klein.663/Box Sync/OSU/Projects/mrbin/LatinSquare",
 verbose=TRUE,removeAreaList=NULL,sumBinList=NULL,Factors=NULL,
 NMRfolders=c(
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_1_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_2_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_3_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_4_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_5_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_6_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_7_2012_04_05/12/pdata/10",
  "C:/Bruker/TopSpin3.6.1/data/guest/nmr/Latin_Square_8_2012_04_05/12/pdata/10"
 )))
#Read bin data
bins<-read.csv(
 "C:/Users/klein.663/Box Sync/OSU/Projects/mrbin/LatinSquarebins.csv",
 check.names = FALSE,row.names=1)
#Create matrix for results
corr <- as.data.frame(matrix(rep(NA,ncol(bins)*8*5),ncol=5))
colnames(corr) <-  c("Spiked","Identified","Chemical shift","R","p")
#Create matrix with spike-in levels
spike<-2^(7:0)*46.875
spikeLevels<-cbind(spike,spike[c(8,1:7)],spike[c(7:8,1:6)],spike[c(6:8,1:5)],
 spike[c(5:8,1:4)],spike[c(4:8,1:3)],spike[c(3:8,1:2)],spike[c(2:8,1)])
colnames(spikeLevels)<-c("Lactate","Alanine","Creatinine","3-Hydroxybutyrate",
   "Histidine","Phosphocreatine","Betaine","Acetate")
rownames(spikeLevels)<-1:8
counter<-1
#Calculate correlation for combinations of all bins, all spike-in levels
for(j in colnames(spikeLevels)){
  for (i in 1:ncol(bins)){
      corrTMP<- cor.test(x=spikeLevels[,j],y=bins[,i],method = 'spearman')
       corr[counter,"R"] <- corrTMP$estimate
       corr[counter,"p"] <- corrTMP$p.value
       corr[counter,"Chemical shift"] <-colnames(bins)[i]
       corr[counter,"Spiked"] <-j
       counter<-counter+1
  }
}
corrFDR<-p.adjust(corr[,"p"], "BH") #Correction for multiple testing
FDRthreshold<-.1
resultTable<-corr[which(corrFDR<FDRthreshold),]
resultTable #Display significant results
```

Noise Removal

Noise removal is a method to reduce the size of data set by removing spectral areas that do not contain actual signals. In univariate data analysis approaches, this reduced number of bins can improve the statistical power of the analysis. NMR noise removal is usually based on Signal-to-Noise (SNR) ratios. Noise is defined for each individual spectrum as the average noise level in a "noise region" that is expected to be free from actual NMR signals. Noise is usually calculated as the standard deviation (or a closely related measure) of all bins in the noise region. SNR is defined as a small number, where bins being below Noise x SNR are considered noise signals. Different tools differ in how the actual noise removal is performed. In some cases, all values below SNR are set to 0, in other cases all bins that are on average less than SNR are removed. mrbin offers more flexibility and control by allowing the user to define both SNR and an additional threshold ratio T. Bins will remain in the data set if the number of values above SNR exceeds T. T can be set differently depending on the aim of the study. Large T values such as 0.75 create a "conservative" data set with fewer bins that are present consistently across most samples. This conservative data set can provide more statistical power for experiments where small changes in metabolite abundances need to be detected. Low values of T, such as 0.2, create a more exploratory data set with a larger number of bins, including metabolites that are inconsistently present. This exploratory data set may come at the cost of reduced statistical power but can be advantageous in experiments where metabolites may be completely absent in some samples, such as analyses of bacterial metabolism.

2D NMR Support

A special focus of mrbin is on the analysis of 2D NMR data, such as ^{1}H-^{13}C HSQC spectra. As can be seen in Table 1, most available software tools (and all of the free tools) for untargeted analysis do not support the use of 2D data, despite the known advantages of such data.

mrbin fully supports the use of 2D NMR data, including all steps such as binning, removal of solvents and noise, scaling, and normalization. Additionally, mrbin allows for removal of spectral areas that are known to contain no real signals. This process is referred to as spectral cropping. For HSQC spectra, spectral cropping applies to areas that are far away from a diagonal line that roughly runs from 0ppm, 0ppm to 9ppm, 160ppm (^{1}H and ^{13}C shift values). mrbin allows for automatic or user defined removal of such sparse areas, which can significantly reduce the number of bins left after noise removal, increasing statistical power of subsequent data analysis. This 2D cropping step is a novel feature and has not been reported before.

Normalization

Normalization is a step of high importance in metabolomics data sets and mrbin implements a version of PQN scaling. For 2D HSQC spectra, mrbin can automatically exclude a large part of the region containing sugar signals for the PQN scaling step. This removal was implemented to avoid an overly strong influence of sugar signals on the scaling, as sugars usually have multiple signals across a large spectral area, but still only represent one or a few metabolites. This novel feature has not been reported previously.

SUPPLEMENTAL FIGURES

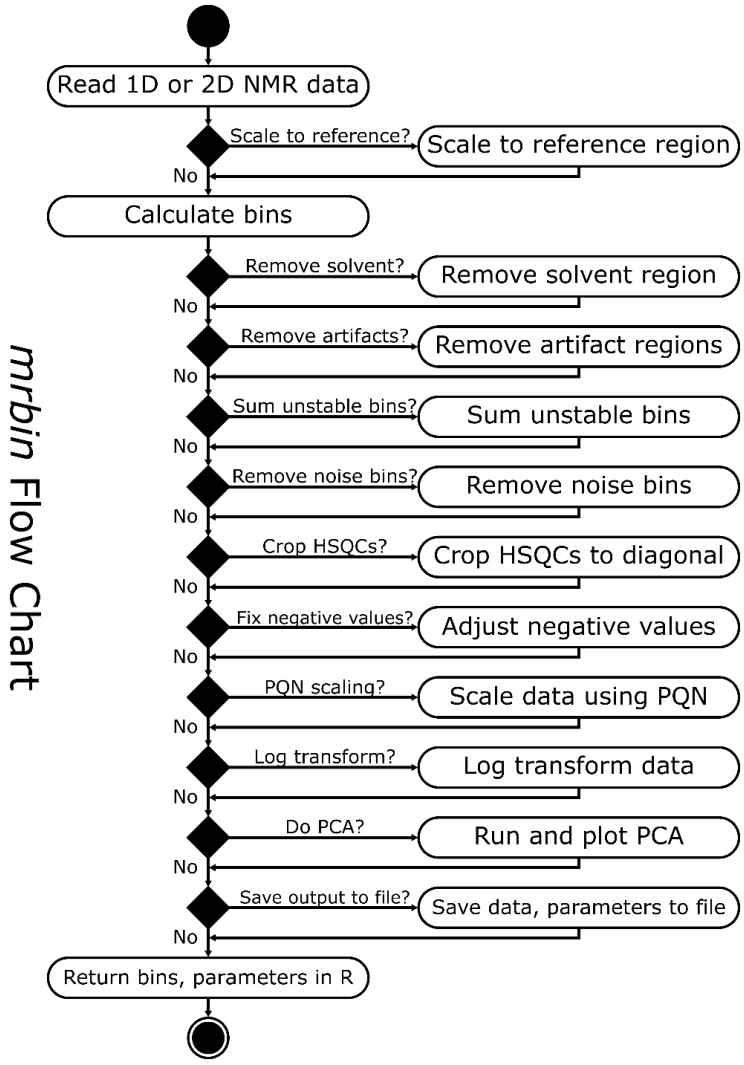

Figure S1. Flowchart of the mrbin data processing workflow.