RBMRB : A package to download and visualize BMRB data in R

Kumaran Baskaran

November 08, 2016

# RBMRB

[BMRB](http://www.bmrb.wisc.edu/) collects, annotates, archives, and disseminates (worldwide in the public domain) the important spectral and quantitative data derived from NMR spectroscopic investigations of biological macromolecules and metabolites. The goal is to empower scientists in their analysis of the structure, dynamics, and chemistry of biological systems and to support further development of the field of biomolecular NMR spectroscopy

[RBMRB](https://github.com/uwbmrb/RBMRB) is a library to fetch NMR chemical shift data directly from [BMRB](http://www.bmrb.wisc.edu/) into R environment as a data frame in R. This facilitates access to BMRB data for statistical analysis and data visualization. It is using the [BMRB-API](https://github.com/uwbmrb/BMRB-API) to fetch the data from BMRB database.

## Installation

RBMRB library has been developed and tested in R version 3.3.x. It requires the following R packages preinstalled

* **httr** to import data from BMRB web server(version 1.2.1 or later)
* **data.table** to format the imported data into a data frame in R (version 1.9.6 or later)
* **rjson** to deal with BMRB-API (version 0.2.15 or later)
* **ggplot2** to simulate spectra (version 2.1.0 or later)
* **plotly** for interactive graphics in simulated spectra (version 4.5.2 or later)

Users should make sure that the above packages have beed installed correctly with the required versions, before proceeding to RMBRM insallation.

Here is the instruction to install those packages. Open your R and use the following command

installed.packages(c("httr","data.table","rjson","ggplot2","plotly"))

#### Method 1

Once the necessary packages have been installed, proceed with RMRBM installation. The source file can be downloaded from [GitHub](https://github.com/uwbmrb/RBMRB/raw/master/RBMRB_2.0.tar.gz)

After downloading the source file, use the following command to install RBMRB library

install.packages("~/Downloads/RBMRB\_2.0.tar.gz",repos=NULL,type="source")

Note: provide the correct path to the downloaded file.

#### Method 2

If you have devtools library in your R, then you can install directly from GitHub.

library(devtools)  
install\_github("uwbmrb/RBMRB/RBMRB")

## Usage

RBMRB can be used in a similar way like any other library in R.

library(RBMRB)

## Data access

BMRB data can be imported in two ways

* **Entry method** Chemical shift data from single or multiple entries
* **Atom method** Chemical shift data from all entries for a given atom

### Entry method

#### fetch\_entry\_chemical\_shifts:

This function will fetch the 'Atom\_chem\_shift' loop from a NMR-STAR file for a given entry or a list of entries in CSV format. This function works on both macromolecules and metabolites data base. For metabilites entry ids should have right prefix (example 'bmse000034')

###### Examples:

df1<-fetch\_entry\_chemical\_shifts(15060)  
df2<-fetch\_entry\_chemical\_shifts(c(17074,17076,17077))  
df2<-fetch\_entry\_chemical\_shifts(c('17074','17076','17077'))  
df3<-fetch\_entry\_chemical\_shifts(c('bmse000034','bmse000035','bmse000036'))

These data frames have the following columns

colnames(df1)

## [1] "ID" "Assembly\_atom\_ID"   
## [3] "Entity\_assembly\_ID" "Entity\_ID"   
## [5] "Comp\_index\_ID" "Seq\_ID"   
## [7] "Comp\_ID" "Atom\_ID"   
## [9] "Atom\_type" "Atom\_isotope\_number"   
## [11] "Val" "Val\_err"   
## [13] "Assign\_fig\_of\_merit" "Ambiguity\_code"   
## [15] "Occupancy" "Resonance\_ID"   
## [17] "Auth\_entity\_assembly\_ID" "Auth\_asym\_ID"   
## [19] "Auth\_seq\_ID" "Auth\_comp\_ID"   
## [21] "Auth\_atom\_ID" "Details"   
## [23] "Entry\_ID" "Assigned\_chem\_shift\_list\_ID"

Sample data output

head(df1)

## ID Assembly\_atom\_ID Entity\_assembly\_ID Entity\_ID Comp\_index\_ID Seq\_ID  
## 1 1 . 1 1 20 20  
## 2 2 . 1 1 20 20  
## 3 3 . 1 1 20 20  
## 4 4 . 1 1 20 20  
## 5 5 . 1 1 21 21  
## 6 6 . 1 1 21 21  
## Comp\_ID Atom\_ID Atom\_type Atom\_isotope\_number Val Val\_err  
## 1 LEU H H 1 8.149 NA  
## 2 LEU CA C 13 56.016 NA  
## 3 LEU CB C 13 42.180 NA  
## 4 LEU N N 15 122.739 NA  
## 5 VAL H H 1 8.048 NA  
## 6 VAL CA C 13 63.412 NA  
## Assign\_fig\_of\_merit Ambiguity\_code Occupancy Resonance\_ID  
## 1 . 1 . .  
## 2 . 1 . .  
## 3 . 1 . .  
## 4 . 1 . .  
## 5 . 1 . .  
## 6 . 1 . .  
## Auth\_entity\_assembly\_ID Auth\_asym\_ID Auth\_seq\_ID Auth\_comp\_ID  
## 1 . . 20 LEU  
## 2 . . 20 LEU  
## 3 . . 20 LEU  
## 4 . . 20 LEU  
## 5 . . 21 VAL  
## 6 . . 21 VAL  
## Auth\_atom\_ID Details Entry\_ID Assigned\_chem\_shift\_list\_ID  
## 1 HN . 15060 1  
## 2 CA . 15060 1  
## 3 CB . 15060 1  
## 4 N . 15060 1  
## 5 HN . 15060 1  
## 6 CA . 15060 1

### Atom method

#### fetch\_atom\_chemical\_shifts:

This function will fetch the chemical shift data from all the entries for a given atom. The atom name should be in NMR-STAR atom nomenclature.

###### Examples:

df4<-fetch\_atom\_chemical\_shifts('CG2')  
df5<-fetch\_atom\_chemical\_shifts('C9')

These data frames have the following columns

colnames(df4)

## [1] "Entry\_ID" "Entity\_ID"   
## [3] "Comp\_index\_ID" "Comp\_ID"   
## [5] "Atom\_ID" "Atom\_type"   
## [7] "Val" "Val\_err"   
## [9] "Ambiguity\_code" "Assigned\_chem\_shift\_list\_ID"

Sample data output

head(df4)

## Entry\_ID Entity\_ID Comp\_index\_ID Comp\_ID Atom\_ID Atom\_type Val  
## 1 10001 1 1 ILE CG2 C 15.700  
## 2 10001 1 6 ILE CG2 C 17.900  
## 3 10002 1 3 ILE CG2 C 17.516  
## 4 10002 1 18 VAL CG2 C 22.278  
## 5 10002 1 19 THR CG2 C 21.957  
## 6 10002 1 26 THR CG2 C 21.779  
## Val\_err Ambiguity\_code Assigned\_chem\_shift\_list\_ID  
## 1 0.4 1 1  
## 2 0.3 1 1  
## 3 0.4 1 1  
## 4 0.4 1 1  
## 5 0.4 1 1  
## 6 0.4 1 1

## Data manipulation

There are few data manipulation functions are availbale to facilitate plotting.

#### convert\_cs\_to\_n15hsqc:

This function will reformat the chemical shift data frame into a data frame which is easy to plot the N15-HSQC spectrum from the data.

#### Examples

n15hsqc1<-convert\_cs\_to\_n15hsqc(df1)  
n15hsqc2<-convert\_cs\_to\_n15hsqc(df2)

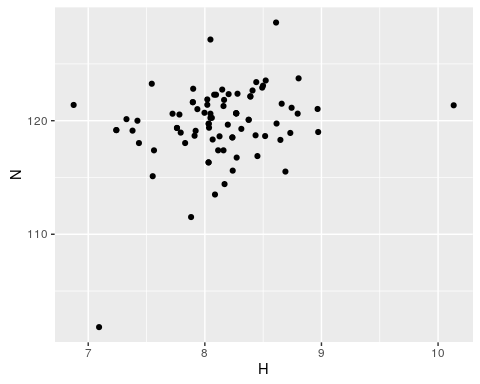
The output data frame will look like

head(n15hsqc1)

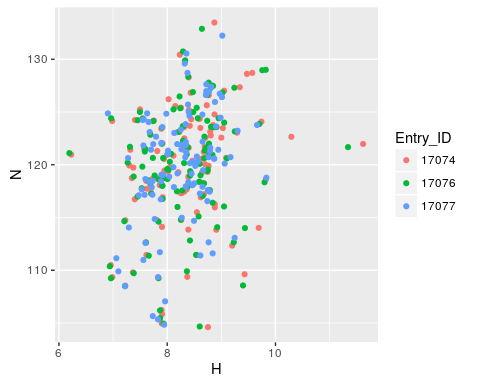
## Entry\_ID Comp\_index\_ID Entity\_ID Assigned\_chem\_shift\_list\_ID Comp\_ID\_H  
## 1 15060 101 1 1 ASP  
## 2 15060 102 1 1 ASP  
## 3 15060 103 1 1 SER  
## 4 15060 104 1 1 ASP  
## 5 15060 105 1 1 GLU  
## 6 15060 106 1 1 GLU  
## Comp\_ID\_N H N  
## 1 ASP 8.269 120.647  
## 2 ASP 8.376 120.080  
## 3 SER 8.239 115.602  
## 4 ASP 8.409 122.658  
## 5 GLU 8.269 120.647  
## 6 GLU 8.391 122.119

This data frame is easy to plot using any plotting library

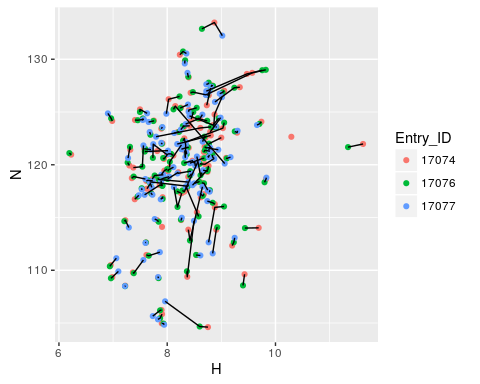
library(ggplot2)  
plt1<-ggplot(n15hsqc1)+geom\_point(aes(x=H,y=N))  
plt1



plt2<-ggplot(n15hsqc2)+geom\_point(aes(x=H,y=N,color=Entry\_ID))  
plt2



plt3<-ggplot(n15hsqc2)+geom\_point(aes(x=H,y=N,color=Entry\_ID))+geom\_line(aes(x=H,y=N,group=Comp\_index\_ID))  
plt3



#### convert\_cs\_to\_c13hsqc:

This function will reformat the chemical shift data frame into a data frame which is easy to plot the C13-HSQC spectrum from the data.

#### Examples

c13hsqc1<-convert\_cs\_to\_c13hsqc(df1)  
c13hsqc2<-convert\_cs\_to\_c13hsqc(df2)

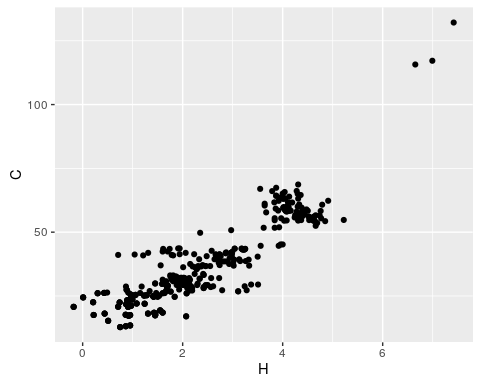
The output data frame will look like

head(c13hsqc1)

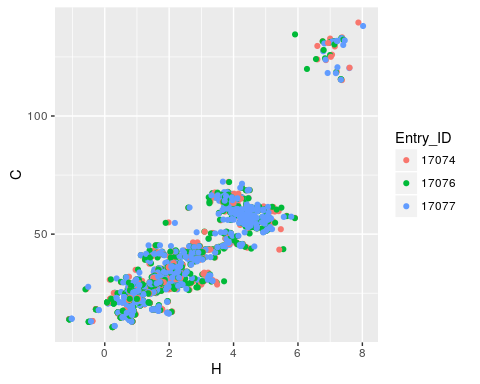
## Entry\_ID Comp\_index\_ID Entity\_ID Assigned\_chem\_shift\_list\_ID Comp\_ID\_C  
## 1 15060 101 1 1 ASP  
## 2 15060 102 1 1 ASP  
## 3 15060 103 1 1 SER  
## 4 15060 104 1 1 ASP  
## 5 15060 105 1 1 GLU  
## 6 15060 106 1 1 GLU  
## Comp\_ID\_H Atom\_ID\_C Atom\_ID\_H C H  
## 1 ASP CA HA 54.487 4.630  
## 2 ASP CA HA 54.572 4.609  
## 3 SER CA HA 58.470 4.420  
## 4 ASP CA HA 54.567 4.640  
## 5 GLU CA HA 56.521 4.271  
## 6 GLU CA HA 56.400 4.300

and the user may generate a spectrum using the following script

library(ggplot2)  
plt1<-ggplot(c13hsqc1)+geom\_point(aes(x=H,y=C))  
plt1



plt2<-ggplot(c13hsqc2)+geom\_point(aes(x=H,y=C,color=Entry\_ID))  
plt2



#### convert\_cs\_to\_tocsy:

This function will reformat the chemical shift data frame into a data frame which is easy to plot the TOCSY spectrum from the data. Note : Since both dimensions have protein chemical shifts, the columns are named as Val.x and Val.y

#### Examples

tocsy1<-convert\_cs\_to\_tocsy(df1)  
tocsy2<-convert\_cs\_to\_tocsy(df2)

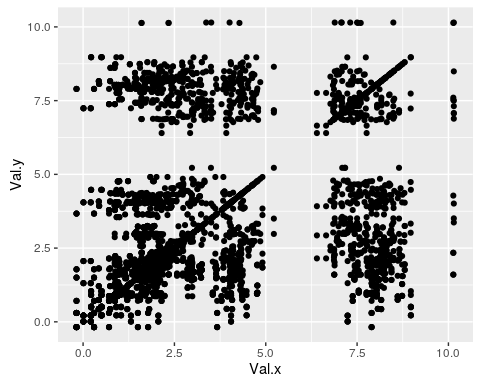
after conversion the data will look like

head(tocsy1)

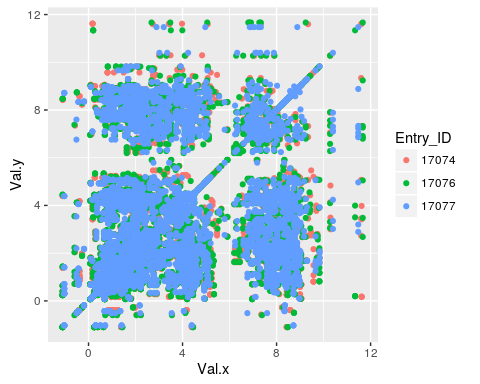
## Entry\_ID Entity\_ID Comp\_index\_ID Assigned\_chem\_shift\_list\_ID ID.x  
## 1 15060 1 100 1 915  
## 2 15060 1 100 1 915  
## 3 15060 1 100 1 916  
## 4 15060 1 100 1 916  
## 5 15060 1 101 1 919  
## 6 15060 1 101 1 919  
## Assembly\_atom\_ID.x Entity\_assembly\_ID.x Seq\_ID.x Comp\_ID.x Atom\_ID.x  
## 1 . 1 100 GLY HA2  
## 2 . 1 100 GLY HA2  
## 3 . 1 100 GLY HA3  
## 4 . 1 100 GLY HA3  
## 5 . 1 101 ASP H  
## 6 . 1 101 ASP H  
## Atom\_type.x Atom\_isotope\_number.x Val.x Val\_err.x Assign\_fig\_of\_merit.x  
## 1 H 1 3.960 NA .  
## 2 H 1 3.960 NA .  
## 3 H 1 4.000 NA .  
## 4 H 1 4.000 NA .  
## 5 H 1 8.269 NA .  
## 6 H 1 8.269 NA .  
## Ambiguity\_code.x Occupancy.x Resonance\_ID.x Auth\_entity\_assembly\_ID.x  
## 1 2 . . .  
## 2 2 . . .  
## 3 2 . . .  
## 4 2 . . .  
## 5 1 . . .  
## 6 1 . . .  
## Auth\_asym\_ID.x Auth\_seq\_ID.x Auth\_comp\_ID.x Auth\_atom\_ID.x Details.x  
## 1 . 100 GLY HA1 .  
## 2 . 100 GLY HA1 .  
## 3 . 100 GLY HA2 .  
## 4 . 100 GLY HA2 .  
## 5 . 101 ASP HN .  
## 6 . 101 ASP HN .  
## ID.y Assembly\_atom\_ID.y Entity\_assembly\_ID.y Seq\_ID.y Comp\_ID.y  
## 1 915 . 1 100 GLY  
## 2 916 . 1 100 GLY  
## 3 915 . 1 100 GLY  
## 4 916 . 1 100 GLY  
## 5 919 . 1 101 ASP  
## 6 920 . 1 101 ASP  
## Atom\_ID.y Atom\_type.y Atom\_isotope\_number.y Val.y Val\_err.y  
## 1 HA2 H 1 3.960 NA  
## 2 HA3 H 1 4.000 NA  
## 3 HA2 H 1 3.960 NA  
## 4 HA3 H 1 4.000 NA  
## 5 H H 1 8.269 NA  
## 6 HA H 1 4.630 NA  
## Assign\_fig\_of\_merit.y Ambiguity\_code.y Occupancy.y Resonance\_ID.y  
## 1 . 2 . .  
## 2 . 2 . .  
## 3 . 2 . .  
## 4 . 2 . .  
## 5 . 1 . .  
## 6 . 1 . .  
## Auth\_entity\_assembly\_ID.y Auth\_asym\_ID.y Auth\_seq\_ID.y Auth\_comp\_ID.y  
## 1 . . 100 GLY  
## 2 . . 100 GLY  
## 3 . . 100 GLY  
## 4 . . 100 GLY  
## 5 . . 101 ASP  
## 6 . . 101 ASP  
## Auth\_atom\_ID.y Details.y  
## 1 HA1 .  
## 2 HA2 .  
## 3 HA1 .  
## 4 HA2 .  
## 5 HN .  
## 6 HA .

Plotting TOCSY spectrum

library(ggplot2)  
plt1<-ggplot(tocsy1)+geom\_point(aes(x=Val.x,y=Val.y))  
plt1



plt2<-ggplot(tocsy2)+geom\_point(aes(x=Val.x,y=Val.y,color=Entry\_ID))  
plt2



#### filter\_residue:

This function will filter the data frame and remove all non standard amino acids. The data frame should contain the amino acid information in the Comp\_ID column. ####Examples

df6<-fetch\_atom\_chemical\_shifts('CG2')  
df7<-filter\_residue(df6)

## Data visualization

RBMRB library contains few functions to generate interactive visualization of BMRB data with out any data manipulation. The interactive visualizations use **plotly** library. If user has problem with plotly, then this feature may be diabled by providing an argument 'interactive=FALSE' for these functions. These interactive plots can be zoomed in and out using a mouse and will show tooltip information when you mouse over. These visualizations can be exported as a stand alone html file

#### HSQC\_15N

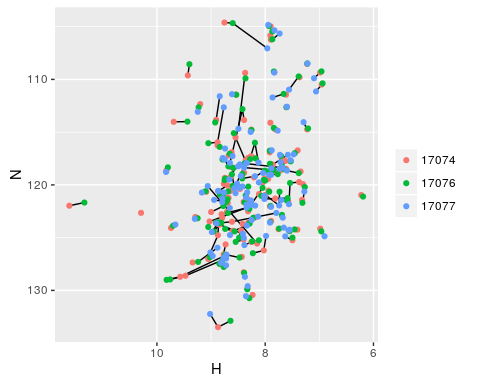
This function will simulae N15-HSQC spectrum for a given entry or list of entries.

##### Examples

These interactive visualization can be exported as single stand alone html file

spec1<-HSQC\_15N(15060)  
spec1  
spec2<-HSQC\_15N(c(17074,17076,17077),type='line')  
spec2

spec3<-HSQC\_15N(c(17074,17076,17077),type='line',interactive = F)  
spec3



#### HSQC\_13C

This function will simulae C13-HSQC spectrum for a given entry or list of entries.

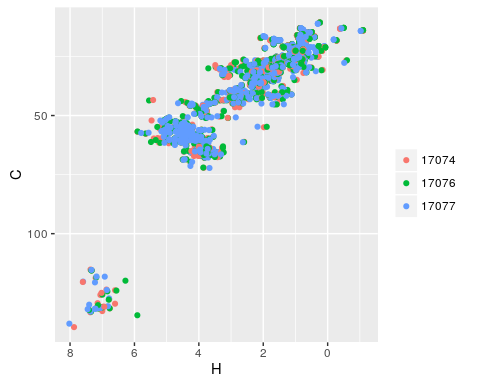
##### Examples

These interactive visualization can be exported as single stand alone html file

spec1<-HSQC\_13C(15060)  
spec1  
spec2<-HSQC\_13C(c(17074,17076,17077))  
spec2

Non interactive plot

spec3<-HSQC\_13C(c(17074,17076,17077),interactive = F)  
spec3



#### TOCSY

This function will simulae TOCSY spectrum for a given entry or list of entries.

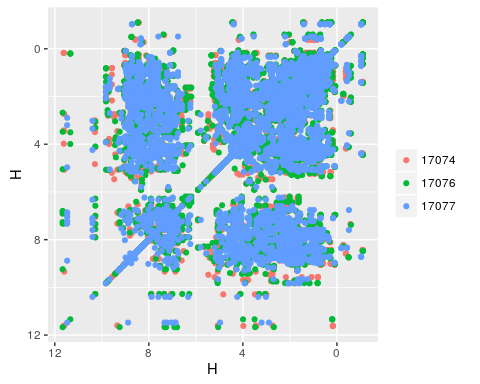
##### Examples

These interactive visualization can be exported as single stand alone html file

spec1<-TOCSY(15060)  
spec1  
spec2<-TOCSY(c(17074,17076,17077))  
spec2

Non interactive plot

spec3<-TOCSY(c(17074,17076,17077),interactive = F)  
spec3



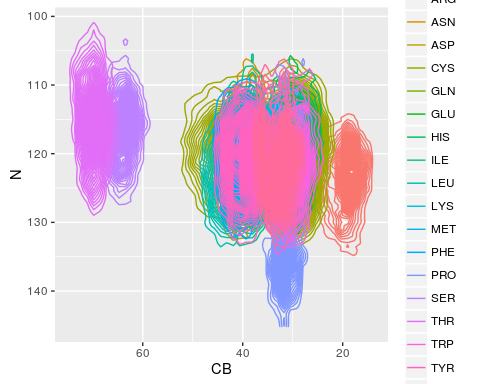
#### chemical\_shift\_corr

This function will plot the distribution of checmial shift correlation between any two atoms from the 20 standard amino acids. The distribution of a particular residue may turn on and off by clicking the residue name in the legand.

corr\_plot1<-chemical\_shift\_corr('CB','N')  
corr\_plot1  
corr\_plot2<-chemical\_shift\_corr('CA','HA\*')  
corr\_plot2

Non interactive plot

corr\_plot1<-chemical\_shift\_corr('CB','N',interactive = F)  
corr\_plot1



corr\_plot2<-chemical\_shift\_corr('CA','HA\*',interactive = F)  
corr\_plot2

