LDA classification for protein mapping

The code is written entirely in Python 3.8

Input files:

1. Spreadsheet with NMR data of test protein
2. Text file with amino acid sequence of the test protein (*fasta* file)
3. Text file with BMRB entry IDs of the proteins for training set

Output files:

1. Figure showing classification probabilities for each spin system in the input spreadsheet
2. Excel spreadsheet with the matrix of classification probabilities

Formatting the input files

1. The spreadsheet containing NMR data of the protein under investigation can be a Microsoft Excel file (extension *.xlsx) or LibreOffice file (extension *.ods), and formatted as follows:

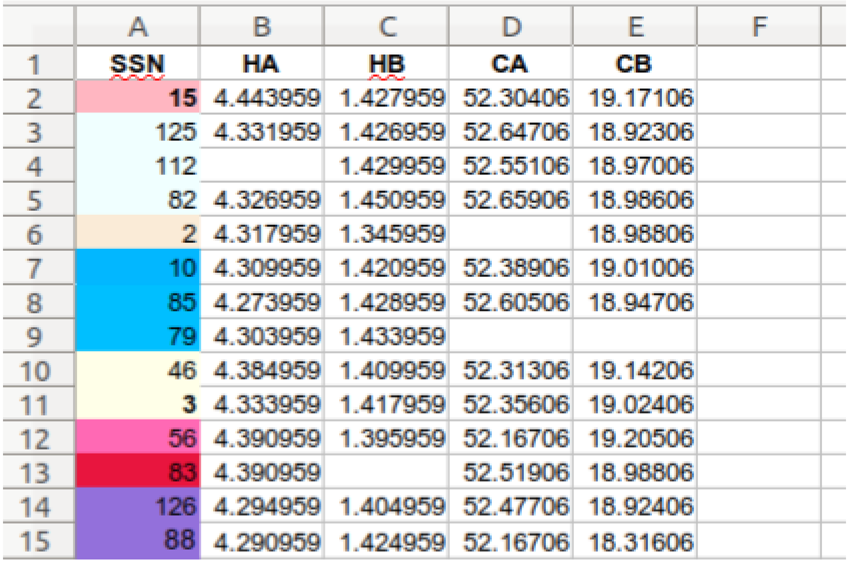
➢ Type 'SSN' for Spin Systems Numbers as a header of the first column (cell A1). Then, starting from cell A2, place all the spin systems numbers in this column. These numbers can be in any order.

➢ Headers for the next columns go according to chemical shifts that will be used for the classification. There can be any number of chemical shifts and they can be placed in any order. Don't leave empty columns in between. Names of chemical shifts should be abbreviated according to the following table:

Chemical shift	*Header*
Hα	HA
Hβ	HB
Cα	CA
Cβ	CB
H^N	H
N	N
CO	C

➢ Copy the chemical shift values for each spin system in the corresponding column. If there are missing chemical shift values, then those cells must be left empty.

➢ The following Figure gives an example of what the spreadsheet may look like:



	A	B	C	D	E	F
1	SSN	HA	HB	CA	CB	
2	15	4.443959	1.427959	52.30406	19.17106	
3	125	4.331959	1.426959	52.64706	18.92306	
4	112		1.429959	52.55106	18.97006	
5	82	4.326959	1.450959	52.65906	18.98606	
6	2	4.317959	1.345959		18.98806	
7	10	4.309959	1.420959	52.38906	19.01006	
8	85	4.273959	1.428959	52.60506	18.94706	
9	79	4.303959	1.433959			
10	46	4.384959	1.409959	52.31306	19.14206	
11	3	4.333959	1.417959	52.35606	19.02406	
12	56	4.390959	1.395959	52.16706	19.20506	
13	83	4.390959		52.51906	18.98806	
14	126	4.294959	1.404959	52.47706	18.92406	
15	88	4.290959	1.424959	52.16706	18.31606	

Example of input spreadsheet with NMR data on test protein.

Empty cells represent missing chemical shift values.

2. The *fasta* text file must contain only the amino acid sequence of the test protein, without a header or any extra information, and can be split into more than one line. For example:

```
MDVFMKGLSKAKEGVVAAAEKTKQGVAEAAGKTKEGVLYVGSKTKEGVVHGVATVAEKT
EQVTNVGGAVVTGVTAVAQKTVEGAGSIAAATGFVKKDQLGKNEEGAPQEGILEDMPVDP
DNEAYEMPSEEGYQDYEPEA
```

3. The text file containing the BMRB entry IDs of the proteins that will comprise the training set must have 1 entry ID per line, and it may contain any number of entry IDs. The file must be without a header or any extra information. The following list is an example of the content of one such file:

```
6436
6869
11526
15176
15179
15180
15201
15225
15430
```

Executing the code

To use the code you must first get it from GitHub by either downloading it from

https://github.com/gugumatz/LDA-for-mapping-IDPs

or by synchronizing the GitHub repository with a directory in your computer and pulling the files. Example files will be downloaded along with the python code.

To execute the code, open a terminal in the same directory where the code is and type the following command line:

```
python3 ./LDA_code.py <test-protein>.xlsx <fasta>.txt <IDs>.txt
```

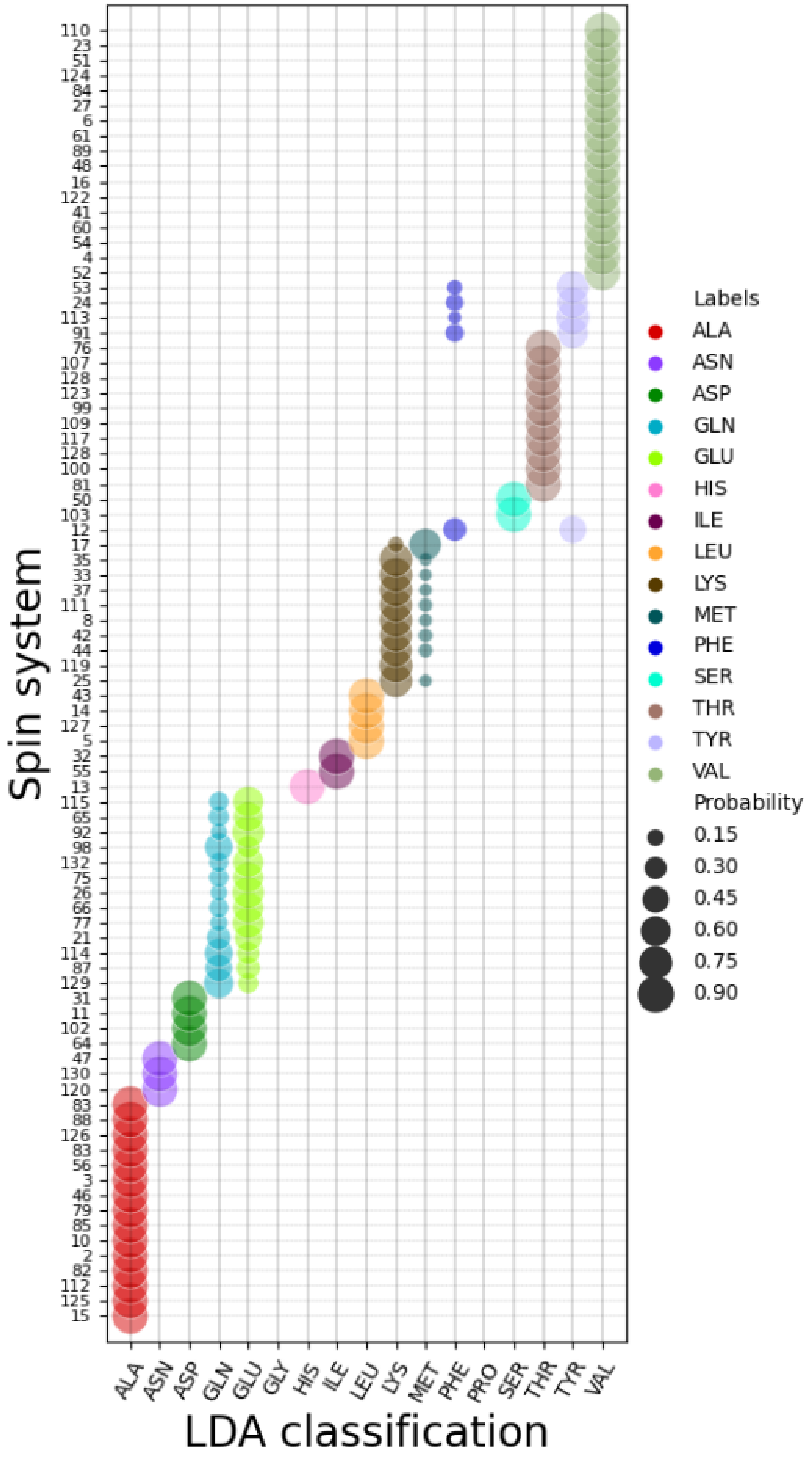
The input files should be typed in the order shown above: first the spreadsheet (here in *.xlsx* format but it can also be *.ods*) with NMR data of the test protein, followed by the *fasta* text file with the amino acid sequence and lastly the BMRB entry IDs text file for the training set.

Depending on the specifications of the computer where you run the code, the size of the data you input and your internet connection (for downloading the training set), the code takes between 5 and 30 seconds of run-time.

Reading the output files:

1. A figure will pop up after the code finishes executing. This figure summarizes the results, showing the LDA classification probabilities for each of the spin systems given as input in the spreadsheet.
   An example figure is shown on the next page. Spin systems numbers are shown on the vertical y-axis and amino acid residue types on the horizontal x-axis. Each amino acid type is represented by a different marker color in the plot, while marker sizes represent classification probabilities.

   If there is a single marker aligned with a given spin system, it means that LDA found a probability greater than 90% that this spin system is of the appropriate amino acid type. On the contrary, if more than 1 marker is aligned with a spin system, it means that classification probabilities are split among many amino acid types.



Example output figure. Marker size represents classification probabilities given by the label to the right, and marker colors represent amino acid types. Colors are transparent to enable the visualization of overlapping markers.

Important: the code trains the LDA model only with those residue types present in the amino acid sequence in the *fasta* file. Probabilities below 10% are automatically deleted from the figure to ease visualization.

2. The second output file is a spreadsheet named “Probabilities.xlsx” containing the same information as the output figure but in the form of a table. The first column of the table holds the spin systems numbers and in subsequent columns the probabilities for each of the amino acid types present in the training set (which are drawn from the *fasta* text file).