# Table of Contents

1 Table of Contents  
1.1 Background .......................................................... 3  
1.2 Installation .......................................................... 6  
1.3 Usage ................................................................. 10  
1.4 Help ................................................................. 20  
1.5 License ............................................................... 21
A tool for the optimization of compound mixtures in 1D $^1$H NMR ligand affinity screens.
Background

Introduction

Ligand affinity screening by nuclear magnetic resonance (NMR) spectroscopy is a versatile tool that is routinely used to support drug discovery and functional genomics research. The real power of NMR ligand affinity screens arises from its ability to directly detect protein-ligand binding events under native or near-native sample conditions. The most common NMR screening approaches (i.e. line-broadening, STD-NMR, and WaterLOGSY) are focused on detecting changes in the one-dimensional (1D) \(^1\)H spectrum of a ligand upon binding to a protein. Hundreds of compounds can be analyzed in a day by coupling these benefits with an automated sample changer, software to optimize probe tuning and other parameters, and rapid NMR data collection of with a cryogenic probe.

To improve the efficiency of NMR ligand affinity screens, compounds are usually evaluated in mixtures. Using mixtures provides two significant benefits: (1) a larger number of compounds can be screened in a shorter amount of time; and (2) the amount of protein required for the entire screen is reduced. However, there are also some drawbacks. As the size of a mixture increases, the probability that a mixture might contain more than one compound competing for the same binding site increases, thus weakening the NMR observable binding event for each compound. Also, compounds in the same mixture may react or interact with each other, thus chemically changing the compounds or causing problems with solubility or aggregation. Finally, the NMR signals from the multiple compounds may overlap leading to ambiguity in analyzing the binding results and the necessity for rescreening with individual compounds. This last problem can be overcome by creating mixtures without peak overlaps, but this task becomes challenging when one is working with a screen composed of hundreds or thousands of compounds. However, a previous study showed that mixtures with minimal peak overlap could be efficiently created by using a simulated annealing algorithm.

NMRmix is a freely available, open-source software solution for the generation of mixtures with minimal peak overlap. NMRmix was written in Python and utilizes the Qt framework with PyQt bindings to build a graphical user interface. NMRmix utilizes a combination of 1D \(^1\)H peak lists for each compound and an overlap range to define whether overlaps occur. The overlap scoring function considers the proportion of peaks in a compound that are overlapped as well as the intensity of the peaks. The software has customizable parameters, downloadable peak list data from the BMRB or HMDB, interactive simulated spectra views, graphs of statistics, and an easily useable interface. Additionally, NMRmix outputs regions of interest (ROIs) in a readable format that can be used to automate the analysis of NMR ligand affinity screening data.
Defining Overlaps

To determine whether an overlap occurs in a mixture, each $^1$H NMR signal in each compound is assigned a spectral range defined by a tolerance $\delta$ (in ppm) added to and subtracted from its chemical shift value $c$ ($c \pm \delta$ in ppm). This user-defined overlap range effectively represents the spectral region belonging to each peak of a compound. An overlap in a mixture occurs when the overlap range of a peak for one compound overlaps with the overlap range of a peak from a different compound. Overlaps are not registered when evaluating peaks from the same compound. The overlap range can be set independently for each peak in a compound, or a single global value for all peaks can be used.

Scoring Overlaps

The goal of generating mixtures with no overlaps is frequently not achievable, owing to the number of compounds in each mixture and the distribution of $^1$H NMR chemical shifts in the compound library. Overlaps that do occur should not represent a significant proportion of the peaks belonging to any one compound in the mixture to minimize the ambiguity in identifying compounds in a mixture. Therefore, instead of using the total number of overlaps to optimize the mixtures, NMRmix uses a scoring function based on the proportion of peaks in each compound that are overlapped:

$$S_C = k \frac{N_O}{N_T}$$

where $S_C$ represents the overlap score for the compound, $k$ represents the score scaling function, $N_O$ represents the number of peaks in the compound that are overlapped, and $N_T$ represents the total number of peaks in the compound. In this approach, the penalty associated with an overlap is lower in a compound with a large number of peaks when compared with a compound that has only a few or just one peak.

NMRmix also has the ability to use peak intensities as a factor in evaluating the optimization of mixtures. In most NMR ligand-detect screening approaches (e.g., line-broadening, STD, or WaterLOGSY), identifying a binding event depends on monitoring changes in the line width, intensity, or position of peaks belonging to a compound. The presence of nearby strong signals can hinder the detection of such changes in weak signals. To minimize this problem, NMRmix offers an optional scoring function ($S_C^*$) for use in optimizing mixtures:

$$S_C^* = k \frac{I_O}{I_T}$$

where $S_C^*$ represents the modified overlap score for the compound, $k$ represents the score scaling
function, \( I_0 \) represents the sum of all of the intensities of the overlapped peaks in the compound, and \( I_T \) represents the sum of all of the intensities of all of the peaks in the compound.

**Optimizing Mixtures**

The optimization of mixtures in NMRmix occurs through simulated annealing. In brief, a small number of compounds from different mixtures are swapped during each step of the annealing process. The number of compounds being swapped (the mixing rate) is a user-defined parameter that can be specified prior to optimization. After each annealing step, the new arrangement of compounds in the mixtures is evaluated based on a total overlap score, which is either accepted or rejected. This process continues until the maximum number of annealing steps (set by the user) is reached or stops before if none of the mixtures contains overlaps.

The total overlap score (\( S_T \)) used to evaluate each arrangement of compounds in the mixtures is simply the sum of all of the overlap scores for each compound (\( S_C \) or \( S_C^* \)) in their respective mixtures:

\[
S_T = \sum_{i=1}^{n} S_C
\]

During each step of the annealing process, if the total overlap score for the new arrangement of mixtures is less than or equal to the total overlap score of the current arrangement of mixtures, the new arrangement is automatically accepted; however, to minimize the possibility that the total overlap score for the mixtures becomes trapped in a local minimum due to a particular arrangement of compounds in the mixtures, each step of the simulated annealing process in which the total overlap score increases is evaluated according to a modified Boltzmann probability:

\[
P = e^{-\frac{\Delta S_T}{k_T M S_T}}
\]

where \( P \) is the probability of acceptance, \( \Delta S_T \) is the difference between the total overlap score of the new set of mixtures and that of the current set of mixtures, \( k_T \) is the temperature scaling factor, \( k_S \) is the score scaling factor, \( M \) is the mixing rate, and \( T \) is the step temperature. To facilitate the calculation of a probability of acceptance, each annealing step is associated with a “temperature” value that decreases on each successive step. At the beginning of the annealing process, when the temperature is higher, the probability for accepting a more overlapped arrangement of compounds is greater. Even a new arrangement at a temperature of 10,000 that increases the total overlap score by 20,000 (equivalent to two completely overlapped compounds) still has an 8.2% chance of acceptance. Each subsequent step of the annealing process lowers the temperature, thus lowering the base acceptance probability. In most cases, once the temperature reaches approximately 100 (using default parameters), the probability of acceptance becomes zero.

Fig. 1.2: Temperature dependence of the Boltzmann acceptance probability as a function of \( \Delta S_T \), the difference between the total overlap score of the new set of mixtures and that of the current set of mixtures, with \( \Delta S_T \) values: (blue) 500; 1000 (green); 2500 (yellow); 5000 (orange); 7500 (red); 10000 (violet); 15000 (gray); and 20000 (maroon). The calculations of acceptance probabilities used the default values for the temperature scaling factor (25,000), score scaling factor (10,000), and mixing rate (2).
Summary

NMRmix is a powerful, freely available, open-source tool for generating mixtures of small molecules with minimal NMR peak overlap. The optimization of mixtures is accomplished by using a simulated annealing algorithm previously described (Arroyo 2013). The user-friendly GUI facilitates easy mixture optimization and data analysis, and NMRmix only requires information about the compound library and a source for reference 1D $^1$H peak lists to get started. Additionally, NMRmix introduces the concept of intensity scoring, which penalizes overlaps that occur on the most intense peaks instead of treating overlaps of all peaks equally. After optimization, the resulting mixture table and ROI list can be exported to an easily readable CSV format. The availability of the ROI list in an easily readable format can also facilitate automation of data analysis for NMR-based ligand screening. The ranges of the ROI list can be easily extracted through scripting and imported into various NMR analysis tools as integration regions to automatically quantitate and compare spectra and identify hits. Future versions of NMRmix could include other non-NMR criteria, such as reactivity, solubility, aggregation, or structural similarity, into the score for optimizing mixtures. Additionally, NMRmix could be adapted toward optimizing mixtures for other nuclei used for NMR ligand affinity screens such as $^{19}$F-based screens.

Related References


Installation

Obtaining NMRmix

NMRmix is available from the NMRFAM website as zipped python source files. An example compound library comprising of metabolites found in the BMRB can also be downloaded and directly imported into NMRmix.

NMRFAM Virtual Machine

In an effort to minimize the effort involved with installing software, NMRFAM has created a Linux virtual machine which includes most of the software developed at NMRFAM. A virtual machine is essentially like running another computer, which may be a completely different operating system (i.e. Linux, Windows, etc.), on your computer. The NMRFAM Virtual Machine can be run using a VM software, such as the Oracle VirtualBox.

Download the NMRFAM Virtual Machine here.

Warning: While the NMRFAM Virtual Machine is the easiest way to install all the various NMRFAM software, including NMRmix, running a virtual machine on your computer will have to share resources between your computer’s operating system and the virtual machine’s operating system. For that reason, you’ll want to use a computer that has at least two CPUs or CPU cores and a significant amount of RAM (>4GB).
Dependencies

NMRmix relies on the use of several packages to work. The following represent the minimum version of each required dependency. Testing of NMRmix was done with these versions. There is no guarantee that a different version will work.

Software Dependencies

- Python 2.7.11 or Python 3.5.1
- RDKit 2015.09.02
- Qt 5.1
- Cairo 1.14.6
- PyCairo 1.10.0
- Pango 1.38.1 [OPTIONAL]
- PyQt 5.5.1

Python Packages

- Numpy 1.10.4
- Matplotlib 1.5.1
- BeautifulSoup 4.4.1
- Pillow 3.1.1
- Unicodecsv 0.14.1 [ONLY NEEDED FOR PYTHON 2 BUILD]

Mac OS X Installation (Homebrew)

The easiest approach to install NMRmix on Mac OS X would be to use the `install_mac.sh` script found in the NMRmix folder. This script may take up to an hour or so to complete due to the installation of some of the necessary dependencies.

Execute this script by typing the following into the terminal from the NMRmix directory to install NMRmix using Python 3:

```
./install_mac_py3.sh
```

Or type the following to install using Python 2:

```
./install_mac_py2.sh
```

However, if you would like to manually work through the installation process that the script performs, follow the instructions below. The installation of NMRmix on Mac OS X benefits from the use of a package manager (i.e. Homebrew, MacPorts, Anaconda, etc.) to download and install the necessary packages. The instructions here will focus on the use of Homebrew to install the necessary packages, but the installation procedures should be fairly similar on other package managers. If you have successfully installed NMRmix using another package manager, please email Jaime Stark your instructions, and it will be added to the documentation.

The `install_mac.sh` installs the Homebrew package manager and uses it to install the dependencies necessary for running NMRmix. At the end of the script, the script calls a python script (`setup_mac.py`) that copies the NMRmix program to your Applications directory and then sets up a symbolic link that allows NMRmix to be started from the terminal with a simple command.
Warning: There may be problems installing NMRmix using this script or the method described below if you already use another package manager. If you have concerns about this issue, email Jaime Stark or try using the NMRFAM Virtual Machine.

Installing Homebrew

Open a terminal and type the following command:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/˓→master/install)"
```

This will probably ask you to install the Xcode command line tools, if you don’t already have Xcode installed. While the Xcode command line tools should be able to be installed at this point, the Homebrew installation may still fail. If so, try installing the Xcode command line tools first by typing the following:

```
xcode-select --install
```

If the installation of both the Xcode command line tools and Homebrew successfully completes, type the following command to check for any problems:

```
brew doctor
```

If there are any issues, you will want to refer to the documentation for Homebrew to help troubleshoot.

Installing Packages

As you might have guessed, all commands using the Homebrew package manager start with `brew`. For example to get information about installing Python 3, which is necessary for NMRmix, type the following command:

```
brew info python3
```

This command will show a short overview of the Python package, description, dependencies, and installation arguments.

**Note:** Normally, you could install both of the Python 2 and Python 3 dependencies without much issue. However, this is not currently true for the RDKit package. It can only be install for either Python 2 or Python 3, not both. So for the purposes of NMRmix, you will need to decide whether to use the Python 2 or Python 3 version.

To install Python 3, type the following command:

```
brew install python3
```

Or type the following to install Python 2:

```
brew install python
```

**Why Install Python?**

Python 2.7 is already preinstalled by Apple in Mac OS X. However, this version is not frequently updated, and often utilizes outdated packages. The version installed by package managers like Homebrew are much easier to update and manage.
This will begin by installing all the software dependencies for python first, and then it will install python. Each package that is necessary for NMRmix can be installed in this way. Type the following commands into a terminal window in this order to install all of the necessary packages, for the Python 3 build:

```
brew install python3
brew install fontconfig
brew install py3cairo
brew tap rdkit/rdkit
brew install rdkit --with-pycairo --with-python3
```

Or type the following to install all the packages necessary, for the Python 2 build:

```
brew install python
brew install fontconfig
brew install py2cairo
brew install libsvg-cairo
brew install gtk
brew install pyqt5
brew tap rdkit/rdkit
brew install rdkit --with-pycairo
```

The installation may take a while (an hour or more), especially for the installation of pyqt5 and their dependencies.

Following the successful installation of these packages, several python packages will also need to be installed. To install Python packages, the PIP package manager will be used, which works similarly to Homebrew and was installed along with Python. Type the following commands into the terminal for the Python 3 build:

```
pip3 install numpy
pip3 install matplotlib
pip3 install PyQt5
pip3 install cairocffi
pip3 install beautifulsoup4
pip3 install Pillow
```

Or type the following for the Python 2 build:

```
pip install numpy
pip install matplotlib
pip install cairocffi
pip install beautifulsoup4
pip install Pillow
pip install unicodecsv
```

These commands should install each python package and their dependencies.

## Installing NMRmix

The NMRmix python script can be executed from the NMRmix folder. However, we can make NMRmix easier to execute from the terminal with the following commands. First, we will make the NMRmix python script executable. In terminal, change the directory to the folder containing the NMRmix program and then type the following command:

```
chmod +x NMRmix/nmrmix/NMRmix.py
```

Next, copy the the NMRmix directory to the Applications directory with the following command:

```
cp -r NMRmix /Applications
```
Finally, create a symlink that allows NMRmix to be executed easily within terminal using the following command:

```
ln -s /Applications/NMRmix/nmrmix/NMRmix.py /usr/local/bin/nmrmix
```

To start NMRmix, type the following into the terminal:

```
nmrmix
```

### Linux Installation

The easiest approach to install NMRmix on Linux-based (Ubuntu, Debian, or Red Hat) systems would be to use the `install_linux.py` script found in the NMRmix folder.

Execute this script by typing the following into the terminal from the NMRmix directory:

```
python install_linux.py [INSTALLATION DIR]
```

where the path to the directory where you would like to install NMRmix replaces the [INSTALLATION DIR]. This script will download the appropriate dependencies, copies the NMRmix program into the installation directory, and sets up a symlink to allow NMRmix to be started from the terminal with a simple command:

```
nmrmix
```

If you have any issues with this installation process, please email Jaime Stark or try using the NMRFAM Virtual Machine.

### Windows Installation

Coming Soon!

### Usage

#### Starting NMRmix

Assuming that NMRmix was installed using the installer scripts, open a terminal window and type:

```
nmrmix
```

If NMRmix was installed manually without the scripts, but all the necessary dependencies are present, then the program can be started like any other python program. Open a terminal window, change the current directory to the location of the NMRmix source files, and type:

```
python NMRmix.py
```

The NMRFAM splash screen should pop up, and NMRmix should start. On the resulting NMRmix title screen, press **Continue** to start using NMRmix.

#### Setting Default Parameters

The default user-defined parameters can be set and saved for future uses of NMRmix. The parameters are saved as a simple text file (`parameters.txt`) that is located in a directory called `.nmrmix`, which can be found under the
user’s home directory. This file and folder is only created if the default parameters are changed and saved. If for any reason NMRmix is unable to create this directory/file, saving the default parameters will not be possible. Instead, each parameter will need to be modified manually, if necessary, for each execution of the NMRmix program.

To set the default preferences, click the Set Default Preferences button on the NMRmix title window. The Default Preferences window should appear with four tabs: Directories/Scoring; Optimizing; Refining; and Display/Statistics.

**Directories/Scoring Parameters**

- **Working Directory** For a description of this parameter, see Setting Working and Peaklist Directories.
- **Local Peak List Directory** For a description of this parameter, see Setting Working and Peaklist Directories.
- **Overlap Range** For a description of this parameter, see
- **Score Scaling** For a description of this parameter, see
- **Use Intensity Scoring** For a description of this parameter, see
- **Autosave Results** For a description of this parameter, see

**Optimizing Parameters**

- **Starting Mixture Number** For a description of this parameter, see
- **Max Mixture Size** For a description of this parameter, see
- **Extra Mixtures** For a description of this parameter, see
- **Cooling Rate** For a description of this parameter, see
- **Start Temp** For a description of this parameter, see
- **Final Temp** For a description of this parameter, see
- **Max Steps** For a description of this parameter, see
- **Mix Rate** For a description of this parameter, see
- **Iterations** For a description of this parameter, see
- **Restrict by Group** For a description of this parameter, see
- **Randomize Initial Mixture** For a description of this parameter, see

**Refining Parameters**

- **Use Refinement** For a description of this parameter, see
- **Cooling Rate** For a description of this parameter, see
- **Start Temp** For a description of this parameter, see
- **Final Temp** For a description of this parameter, see
- **Max Steps** For a description of this parameter, see
- **Mix Rate** For a description of this parameter, see
Display/Statistics Parameters

Optimization Progress Bar Update (steps) During the optimization of the mixtures, a progress bar is used to show the number of annealing steps that have been completed. On most computers each step can happen fairly quickly. This parameter defines how often to draw an update to the progress bar. For example, the default is set to 50 steps, which indicates that the progress bar will update every 50 annealing steps. With some lower end computers, it may be useful to draw the updates less frequently (larger step size).

Simulated Peak Width Fraction (half-width at half-maximum) The simulated spectra in NMRmix draws each peak using a Lorentz distribution. While the resolution and line width of NMR peaks is often dependent upon magnet strength, the simulated spectra uses a default line width for drawing. This parameter only changes the cosmetic look of the simulated peaks and does NOT affect the optimization process.

Saving Default Parameters

Close This button closes the Default Preferences window without saving any of the changes made. This means that NMRmix will use the parameters that were loaded upon startup, whether these are the factory defaults or the parameters read from an existing parameters file.

Reset This button resets all of the parameters to the factory default parameters. While this updates all the default parameter values for this open instance of NMRmix, these values are not saved as the default values in the .nmrmix/parameters.txt file. To save these values, press the Save button.

Restore This button resets all of the parameters to the values found in the .nmrmix/parameters.txt file. This updates all the default parameter values for this open instance of NMRmix. Since these values are read directly from the parameters.txt file, there is no need to save unless changes are made.

Save This button will write the all of the parameters in the Default Parameters window to the .nmrmix/parameters.txt file. If the .nmrmix directory does not exist, NMRmix will create it and place the parameters.txt file within. A window will pop-up indicating whether the save was successful.

If saving fails (likely due to permissions issues), these current parameters will be the defaults for the currently open instance of NMRmix. In this case, subsequent launches of NMRmix will not use the updated parameters as they were not saved.

The Compound Library Window

The compound library window is where the user will identify the compounds that will be used to make mixtures and associate those compounds to a peak list. Additional information regarding grouping and/or structure can also be added. While the adding of compounds can be done manually within NMRmix, it is often easier to generate a compound library file (see The Compound Table) externally using a spreadsheet program like Excel then import that library into NMRmix.

Setting Working and Peaklist Directories

Working Directory Setting the Working Directory determines where NMRmix will save results and figures. Additionally, it also sets the initial directory that NMRmix search to open the compound library file.

Local Peaklist Directory Setting the Local Peaklist Directory indicates where to find and save peak lists. Local peaklist files for each compound must be included in this folder. Even when using online sources for peaklists like BMRB and HMDB, the peaklists will be downloaded to the folder selected here.
## 1.3. Usage

### NMRmix Library Import

**Working Directory**

/Users/Jerkr/Desktop

**Local Peak List Directory**

/Users/Jerkr/Desktop/Peaklists

<table>
<thead>
<tr>
<th>Active</th>
<th>Identifier</th>
<th>Compound Name</th>
<th>BMRR_ID</th>
<th>HMDB_ID</th>
<th>Peaklist File</th>
<th>Peaklist Format</th>
<th>Group</th>
<th>PubChem CID</th>
<th>KEGG ID</th>
<th>SMILES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P1-A1</td>
<td>Tris(N,N,N',N'-tetrahydroxyl)</td>
<td>bme000070</td>
<td>HMD00719</td>
<td>P1-A1.csv</td>
<td>BMRR_ID = D20</td>
<td>5565</td>
<td>C07308</td>
<td>CD(CD(CD(CO))C)C</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>P1-A2</td>
<td>D(-)-Malic acid</td>
<td>bme000073</td>
<td>HMD00180</td>
<td>P1-A2.csv</td>
<td>BMRR_ID = D20</td>
<td>3255</td>
<td>CO2O</td>
<td>C(C(=O)(C(=O)O))C</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>P1-A3</td>
<td>Lactate</td>
<td>bme000079</td>
<td>HMD000740</td>
<td>P1-A3.csv</td>
<td>BMRR_ID = D20</td>
<td>11335</td>
<td>C07064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>P1-A4</td>
<td>Lactate</td>
<td>bme000082</td>
<td>HMD000740</td>
<td>P1-A4.csv</td>
<td>BMRR_ID = D20</td>
<td>11335</td>
<td>C07064</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>P1-A5</td>
<td>D-Isoprenaline</td>
<td>bme000116</td>
<td>HMD00247</td>
<td>P1-A5.csv</td>
<td>BMRR_ID = D20</td>
<td>5760</td>
<td>C0064</td>
<td>CD(CD(CD(CO))C)C</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>P1-A6</td>
<td>Sucrose</td>
<td>bme000118</td>
<td>HMD000219</td>
<td>P1-A6.csv</td>
<td>BMRR_ID = D20</td>
<td>9089</td>
<td>C00099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>P1-A7</td>
<td>Glutaric acid</td>
<td>bme000406</td>
<td>HMD00061</td>
<td>P1-A7.csv</td>
<td>BMRR_ID = D20</td>
<td>119</td>
<td>C00034</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>P1-A8</td>
<td>L-Aspartic acid</td>
<td>bme000560</td>
<td>HMD00112</td>
<td>P1-A8.csv</td>
<td>BMRR_ID = D20</td>
<td>5603</td>
<td>C00073</td>
<td>CD(CD(CD(CO))C)C</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>P1-A9</td>
<td>L-Aspartic acid</td>
<td>bme000560</td>
<td>HMD00112</td>
<td>P1-A9.csv</td>
<td>BMRR_ID = D20</td>
<td>5603</td>
<td>C00073</td>
<td>CD(CD(CD(CO))C)C</td>
<td></td>
</tr>
</tbody>
</table>

**Peak List Import**

- **Open Library**: Select library to open.
- **Add Compound**: Add new compound.
- **Delete Compound**: Remove selected compound.
- **Delete All**: Remove all compounds.
- **Quit NMRmix**: Exit program.
- **Continue To Import Peak Lists**: Proceed with peak list import.
The Compound Table

The compound table represents the information for each compound that will be used to create optimal mixtures. Each row of the compound table represents a single compound. Compounds can be added to the compound table one-by-one within NMRmix through the use of Add Compound and editing the individual fields. However, the easier way to add compounds is to create a library file and then import it into NMRmix. The library file is a CSV (Comma Delimited Values) file with a specific set of column headers that can be opened and edited within a spreadsheet editor like Excel. For a description of these columns, see Compound Library Fields.

Note: The values that exist within the compound table when peak lists are imported will be exported as a new library file when NMRmix outputs the optimization results (see The NMRmix Results). This will allow you to maintain the state of the library used for each use of NMRmix.

Example Library Files

- An empty compound library file with all of the correct headers that can be edited in spreadsheet software to add your own compounds.
  Download Empty Library File
- An example compound library file that contains all of the metabolites from the BMRB that have $^1$H chemical shifts.
  Download BMRB Library File

Open Library Opens a file dialog window to select the library file to import. The file must be in the comma delimited format (CSV). Upon opening the library file, the contents of the compound table will be erased and replaced with the contents of the newly imported library file.

Add Compound Add an empty row to the end of the compound table where details of a new compound can be added.

Delete Compound Deletes the currently selected row. The currently selected row will be highlighted.

Delete All Removes all compounds from the compound table.

Once all of the information for the compounds to be mixed has been added, pressing Continue to Import Peak Lists will open the window to Importing Peaklists. Any compound that has an empty identifier/compound name or has a duplicate identifier will pop up an error message. Compounds without any peak list information will fail during the peak list import.

Compound Library Fields

Active Indicates whether the compound should be included for import and use in the mixtures. Active compounds will have peak lists imported and will be available in later components of NMRmix. Inactive compounds will not have their peak lists imported and will be ignored from the rest of the NMRmix process. Setting a compound to ignore allows for a compound to remain in the library, but be easily excluded from the compound currently being screened. This is useful in cases when there is no more stock solution of a compound or a screen only wants to screen particular types of compounds.

Identifier The identifier is as unique string of characters that represents the compound. Ideally, the identifier should be kept to a low number of characters (< 10 characters) and should represent the compound beyond just one screen. Typical examples for a unique ID may be database identifiers, plate-well designations, inventory catalog numbers, etc. This is a required field.

Compound Name The name of the compound being screened. This is a required field but does not need to be unique.
**BMRB ID** The BMRB ID allows for compounds with 1H peak lists in the BMRB to be downloaded. The BMRB ID should follow the format of bmseXXXXXX, where the XXXXX represents the six digit BMRB ID number. This field is optional if another peak list source exist.

**HMDB ID** The HMDB ID allows for compounds with 1H peak lists in the HMDB to be downloaded. The HMDB ID should follow the format of hmdbXXXXX, where the XXXX represents the five digit HMDB ID number. This field is optional if another peak list source is available.

**Peaklist File** The filename, including extension, of the file containing the peak list data. These files should be located in the folder indicated for the Local Peaklist Directory (see Setting Working and Peaklist Directories). This field is optional if another peak list source is available through BMRB or HMDB.

**Peaklist Format** Indicates where to get the peak list data for the compound and what format the data is in.

- **BMRB ID** Must have BMRB ID field filled. It initially search for bmrbXXXXXX.str file in Local Peaklist Directory. If it exists, it will use that file, otherwise it will attempt to find the peak list associated with the BMRB ID online, and download the bmrbXXXXXX.str file to the Local Peaklist Directory. This option may require an internet connection.

- **HMDB ID** Must have HMDB ID field filled. It initially search for hmdbXXXXX.txt file in Local Peaklist Directory. If it exists, it will use that file, otherwise it will attempt to find the peak list associated with the HMDB ID online, and download the hmdbXXXXX.txt file to the Local Peaklist Directory. This option may require an internet connection.

- **TOPSPIN** The file indicated in the Peaklist File field is in the Bruker TopSpin (v3) format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

- **VNMR** The file indicated in the Peaklist File field is in the Agilent VnmrJ format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

- **MNOVA** The file indicated in the Peaklist File field is in the Mestrelab Mnova format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

- **ACD** The file indicated in the Peaklist File field is in the ACD NMR Processor format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

- **NMRSTAR** The file indicated in the Peaklist File field is in the NMRSTAR (v3) format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

- **HMDB** The file indicated in the Peaklist File field is in the HMDB format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

- **USER** The file indicated in the Peaklist File field is in the User-Defined format. If this option is chosen, the Peaklist File field must be filled with a valid file name.

**Group** Adding a group name to a compound indicates that the compound belongs to that group. This is useful in cases where some compounds of one group are not allowed to be considered in mixtures that contain a compound of another group. One common group classification is the solvent used to create the stock solution of the compound. It may be desirable to not mix compounds dissolved in D₂O with compounds dissolved in DMSO.

Adding a group to a compound does not automatically keep different groups from mixing. The effects of grouping are not turned on until the option is selected when Setting Ignore Regions or restricting by group in Optimization Settings.

Note that any compound that does not have a group entered in this field when grouping is turned on is considered a member of the N/A group.

**PubChem CID** This field allows a PubChem Compound ID (CID) to be associated with the compound. Currently, this value does not have any functionality in NMRmix, but may be used in future versions to populate information about the compound.
KEGG ID  This field allows a KEGG COMPOUND ID to be associated with the compound. Currently, this value does not have any functionality in NMRmix, but may be used in future versions to populate information about the compound, such as biochemical pathways.

SMILES  The simplified molecular-input line-entry system (SMILES) 1D representation of the compound. This used to provide 2D molecular structure representation in the NMRmix user interface.

Notes  Allows for general text notes to be stored for each compound. This field is not currently used in NMRmix.

Importing Peaklists
The peaklist import window shows the progress of importing the peak lists from the locations specified in the compound library file.

To begin the import process, click the **Start Import** button (left figure). After clicking this button, the import process should begin. The log window will show the importing results for each compound in the compound library (center figure). If the peaklist was successfully imported, the log output will be blue or black. If the compound was set to be not active in the compound library and thus ignored, the log output will be orange. Finally, if the peaklist failed to import for a compound, the log output will be red. The importing process can be stopped by pressing the **Stop Import** button. Upon completing the import process, a short summary of the number of compounds that succeeded, were ignored, or failed will appear, followed by a list of the compound unique identifiers that failed (right figure).

Pressing the **Cancel** button will back out of the peaklist import window returns to the compound library window. This allows you to fix any compounds that failed to import by checking the filename, file format, or verifying BMRB/HMDB IDs.

Pressing **Continue** will accept the current results of the import process, close the window, and continue with *The Peaklist Statistics Window*.

**Warning:** Compounds that fail to import are treated as if they are inactive in the library. They are set to inactive in the compound library file that is exported in the NMRmix results. Cancel out of the peaklist import window to fix these issues, if necessary.

**Note:** A log file will be generated for this import process when NMRmix results are created.

**Peaklist Formats**

**ACD NMR Processor**

The ACD NMR Processor format
Download Example ACD Peaklist

**HMDB**

The HMDB format
Download Example HMDB Peaklist

**Mestrelab Mnova**

The Mestrelab Mnova format
Download Example Mnova Peaklist

**NMRSTAR (v3)**

The NMRSTAR format
Download Example NMRSTAR Peaklist

**Bruker TopSpin (v3)**

The Bruker TopSpin format
Download Example TopSpin Peaklist

**Agilent VnmrJ**

The Agilent VnmrJ format
Download Example VnmrJ Peaklist

**User-Defined**

The User-Defined format
Download Example Mnova Peaklist
The Peaklist Statistics Window

Reviewing Statistics

stats

Editing Compound Information and Peaklists

editing

Setting Ignore Regions

Ignore regions.

The Mixtures Window

Look at all those mixtures.

The Mixtures Table

Mixtures table

Scoring Settings

score settings

1.3. Usage
Optimization Settings

Optimize settings

Refinement Settings

Refine settings

The Mixtures Spectrum View

Mixtures spectrum view

Manually Setting Mixtures

Manually Setting Mixtures

Importing Previous Mixtures

Import

Optimizing Mixtures

Make better mixtures.

Optimization Process

Here we are optimizing/refining.

View Optimization Results

Here you can look at the results.

The NMRmix Results

The NMRmix Results

Help

Video Tutorials

Video tutorials demonstrating the capabilities of NMRmix are available on Youtube.
Frequently Asked Questions (FAQ)

Can peaklists of other nuclei (i.e. $^{13}\text{C}$ or $^{19}\text{F}$) be used in NMRmix?

Not yet. This is a planned enhancement for future versions of NMRmix.

Contact

If you have any problems or questions with the software, or if you have some suggestions to improve NMRmix, please email Jaime Stark.

License

NMRmix: A tool for the optimization of compound mixtures in 1D $^1\text{H}$ NMR ligand affinity screens

Copyright (C) 2016 National Magnetic Resonance Facility At Madison, University of Wisconsin - Madison

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

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. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see GNU Licenses.