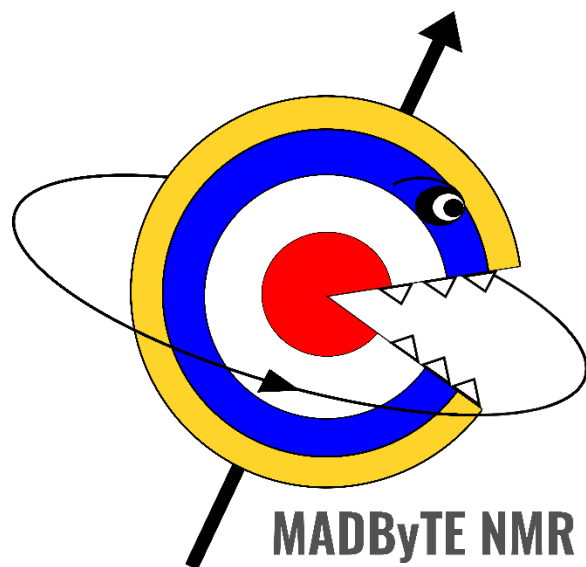


MADByTE User Manual

Manual Version 12.4.2020



©2020 Joseph M. Egan + MADByTE NMR

Contents

1: Setup	4
1.01: Installation	4
Installation Prerequisites	4
Downloading MADByTE:	4
Windows Installation	5
Mac Installation	5
Launching MADByTE	5
Scripting with MADByTE	6
1.02: Prerequisites and Data Processing.....	6
Bruker and Topspin Processing.....	7
MestreNova Processing	7
2: MADByTE Comparisons	8
2.01: How MADByTE Works.....	8
2.02: Getting Started.....	8
2.03: Running MADByTE	9
Running MADByTE on a Subset of Data.....	9
Select NMR Data Directory	9
Select NMR Data Type	10
Select Project Directory	10
Select Solvent.....	11
2.04: Processing Parameters and Filtration Cutoffs	11
Hppm_Error	11
Cppm_Error.....	11
Consensus_Error	12
Similarity_Ratio	12
3: Networks.....	13
3.01: Generating Networks.....	13
Network Structure	13

Types of Networks	14
3.02: Viewing and Manipulating Networks.....	14
3.03: Network Viewer Plugin	15
3.04: Customizing Your Network	16
3.05: Bioactivity Mapping	17
Bioactivity Data Format	17
Mapping the Bioactivity to the Network	18
4: Dereplication.....	20
4.01: Using the Dereplication Library	20
4.02: Establishing a Dereplication Library.....	21
Submit a Peak List as a Reference for Dereplication	22
Add New Data into the Dereplication Library Directly	22
4.03: Other Dereplication Options – Integration with SMART NMR	23
5.0: Advanced Documentation	24
5.01: MADByTE Input Data and Output Files.....	24
Per-Sample Outputs.....	24
Experiment Specific Outputs.....	25
6.0: Acknowledgments.....	26
Appendix A: Quick Start Tutorial.....	27
Appendix B: Directory Navigation Using a Terminal or Console.....	28

1: Setup

1.01: Installation

Installation Prerequisites

MADByTE is a program constructed using the Python programming language, that makes use of many smaller packages common to many informatics pipelines. This makes it impractical for us to provide installation executables. To install MADByTE, it is highly recommended that users first install Anaconda, which is a python distribution that makes it much easier to install the dependencies that MADByTE needs to run. Installation instructions depend heavily on the presence of an Anaconda installation.

To install Anaconda, go to <https://www.anaconda.com/products/individual> and select your operating system. Detailed instructions for installing Anaconda can be found on their website: <https://docs.anaconda.com/anaconda/install/>. We recommend using the 64-bit graphical installer and using all the default settings during installation.

Downloading MADByTE:

Please download and decompress the MADByTE program from the GitHub repository, found through a permanent link at <https://github.com/liningtonlab/madbyte/releases>. If you are familiar with Git, you can clone the repository as you normally would with any GitHub repo. However, if you are unfamiliar with Git, you can download the zip file and extract the files to your computer directly. If you choose to do this, the MADByTE directory name will be “madbyte-[version_number_here]” when you extract it, but you can rename it as you see fit. However, in future steps, you will need to navigate to this directory – so remember where you extract it and what it is called. An expanded step by step download guide can be found in the tutorial section of www.MADByTE.org.

MADByTE requires the directory structure established by the download and extraction of the code. You may store your data in directories outside of this main directory, but if MADByTE fails to launch, it is likely that the directory structure of the source code has been altered.

Please ensure that the version you are downloading is the latest release for the best experience. If you are interested in helping to expand on MADByTE or encounter any issues, you may fork the repo and provide pull requests. You may also open an issue on GitHub (<https://github.com/liningtonlab/madbyte/issues>) or email us directly at madbyte.nmr@gmail.com and we will be happy to work with you.

Windows Installation

To Install MADByTE on Windows:

- Run the script “madbyte_install.bat”
 - This will create an anaconda environment named “madbyte” and install the needed dependencies.
 - Optional (Desktop Shortcut): right click on “MADByTE.bat” and click copy, then on your desktop right-click “Paste Shortcut” to have an easy to access desktop shortcut.
- Alternatively, if you are comfortable with terminal interactions, you may follow the installation directions in the Mac/Linux section. This allows you to customize some options if you wish.
- If you have installed Anaconda in a custom location (I.E. not using the default path), the installation and launch scripts will not work, as they use the defaults from Anaconda. Instead, you can follow the instructions in the Mac installation section.

Mac Installation

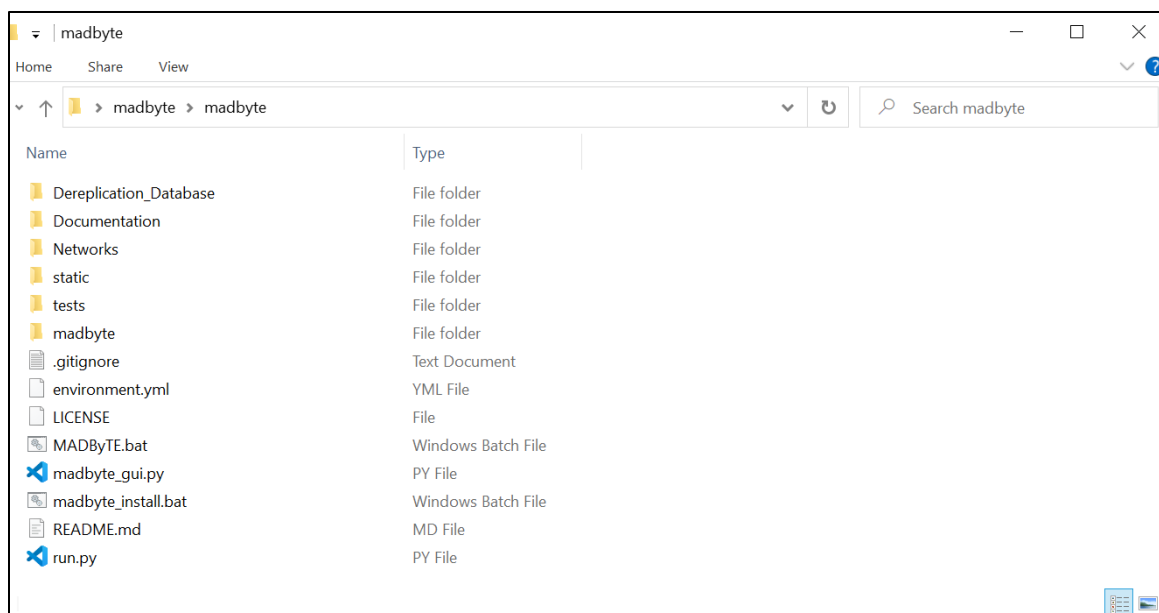
MacOS installation requires a slightly more manual approach that requires use of a console or terminal window. Once Anaconda has been set up, you may use the anaconda prompt (included through the anaconda installation) to navigate to the root “madbyte-[version_number_here]” directory and enter the following command:

```
conda env create -f environment.yml
```

If you are unfamiliar with terminal interactions, [we’ve added a quick explanation in the appendix of this manual.](#)

Launching MADByTE

To launch MADByTE, navigate to the root “madbyte” directory same directory (**folder where the README.md file is located**). It should look like this:



Next, open a terminal and run the following commands:

- `conda activate madbyte`
 - This tells anaconda to activate the MADByTE environment which houses all of the dependencies we just installed.
- `python madbyte_gui.py`
 - This launches MADByTE!

On Windows, the convenient launch script “MADByTE.bat” allows you to simply execute the madbyte.bat script to launch the GUI. You can easily create a desktop shortcut for this file by following the instructions in the [Windows Installation](#) section.

Scripting with MADByTE

As MADByTE is build using Python scripts, it can easily be integrated into your own Python pipeline. An example of how to use the MADByTE API is in “run.py”.

1.02: Prerequisites and Data Processing

MADByTE uses the peak picked data from Bruker’s Topspin or MestreNova to construct the features used in the comparison. This means that the user must perform all the necessary processing – linear prediction, phasing, peak picking – that they wish before running MADByTE.

Users must peak pick their data themselves, using methods they are comfortable with. MADByTE does not peak pick, as this is outside the scope of the utility.

Once your data processing is complete, simply follow the export steps listed below.

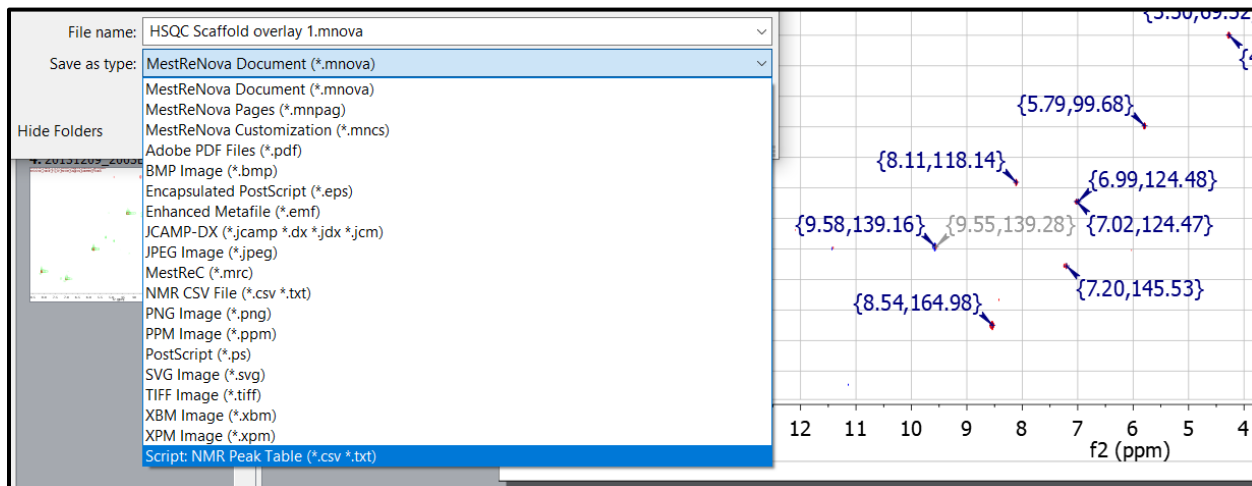
Bruker and Topspin Processing

MADByTE uses peak picked data directly from Bruker without the need to organize it in a special directory – however, it must be exported from Bruker correctly. The recommended export is to run the command ‘convertpeaklist.txt’ from the input bar in topspin. This command exports all the information contained in the peaks tab of a given experiment out as a .txt file within the pdata folder for that experiment without needing to create new directories for your data.

MestreNova Processing

MestreNova exporting is slightly more complicated than that of the Topspin method but gives the users access to a platform-independent method of processing their data.

Once the data has been peak picked, simply go to file, export as, and select Script: NMR Peak Table and format the name as: **[Sample_Name_Here]_HSQC.csv** or **[Sample_Name_Here]_TOCSY.csv**, respectively. Place files in a directory named for the sample.



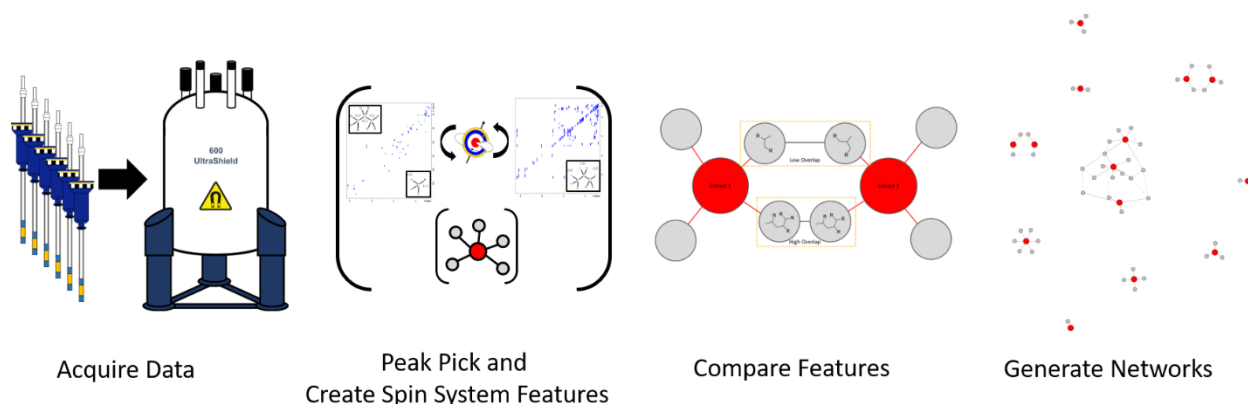
A suggested data directory structure would be:

- MADByTE Input Files
 - [Experimental_Set_Name]
 - [Sample_Name_1]
 - [Sample_Name_1_HSQC.csv]
 - [Sample_Name_1_TOCSY.csv]

- [Sample_Name_2]
 - [Sample_Name_2_HSQC.csv]
 - [Sample_Name_2_TOCSY.csv]

2: MADByTE Comparisons

2.01: How MADByTE Works



MADByTE works by taking the peak picked information and attempting to derive spin systems with carbon correlations. In essence, it is a way to use the scalar coupling information provided by TOCSY/COSY and linking it to the HSQC heteroatom information. This way, if proton resonances fluctuate due to changing substituents, you still match the core scaffold pieces. Additionally, solvent conditions, matrix contribution, etc. all influence chemical shift which would cause simple matching to fail. However, by using an orthogonal viewpoint, these shifts can be perturbed and still provide extremely important information.

2.02: Getting Started

Once the NMR data has been peak picked and exported using the proper output formats, MADByTE is ready to be run. There are a few considerations to consider that will affect the outcome of the processing, listed at the end of this section.

2.03: Running MADByTE

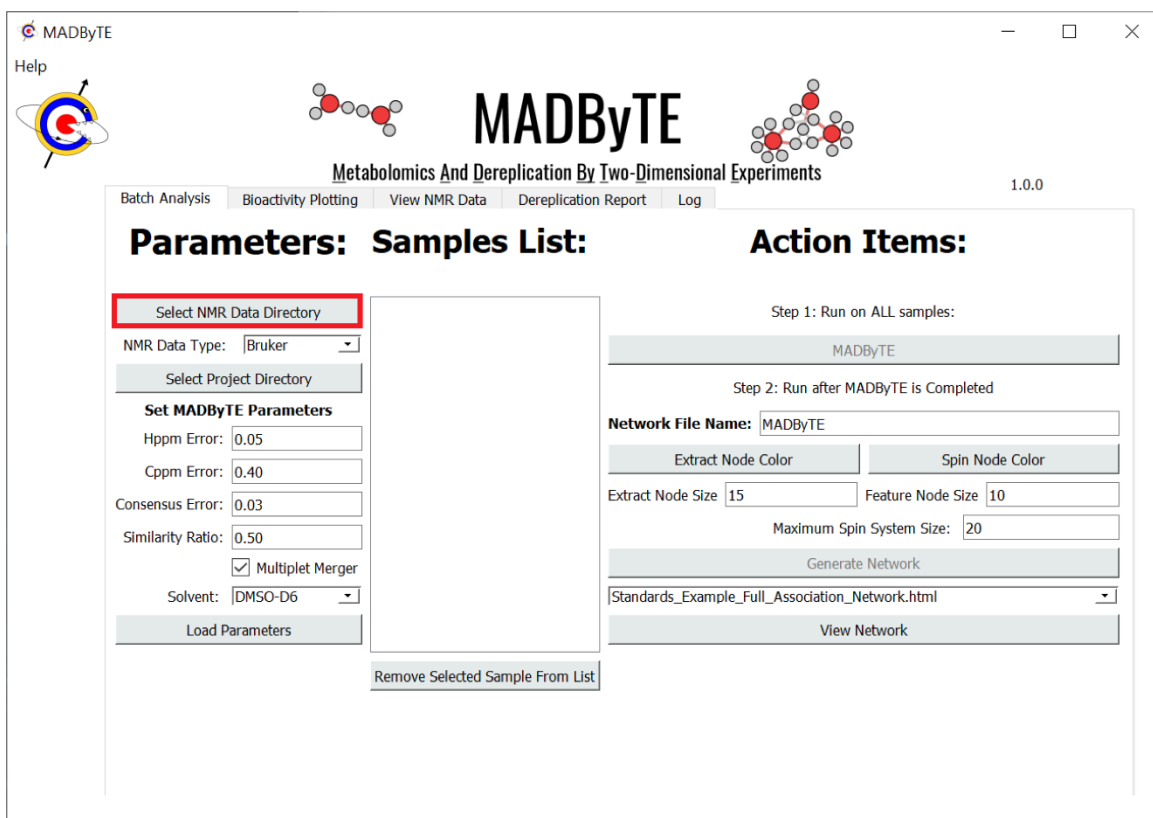
When running MADByTE, you must have already constructed the peak picked lists as outlined above. If you have not done that, do not read further.

Running MADByTE on a Subset of Data

To run MADByTE, simply follow the on-screen prompts in the GUI. Each MADByTE processing is treated as completely independent, so if you want to simply add more samples into an older experiment file, or re-process data with new cutoffs, simply re-select the NMR data directory.

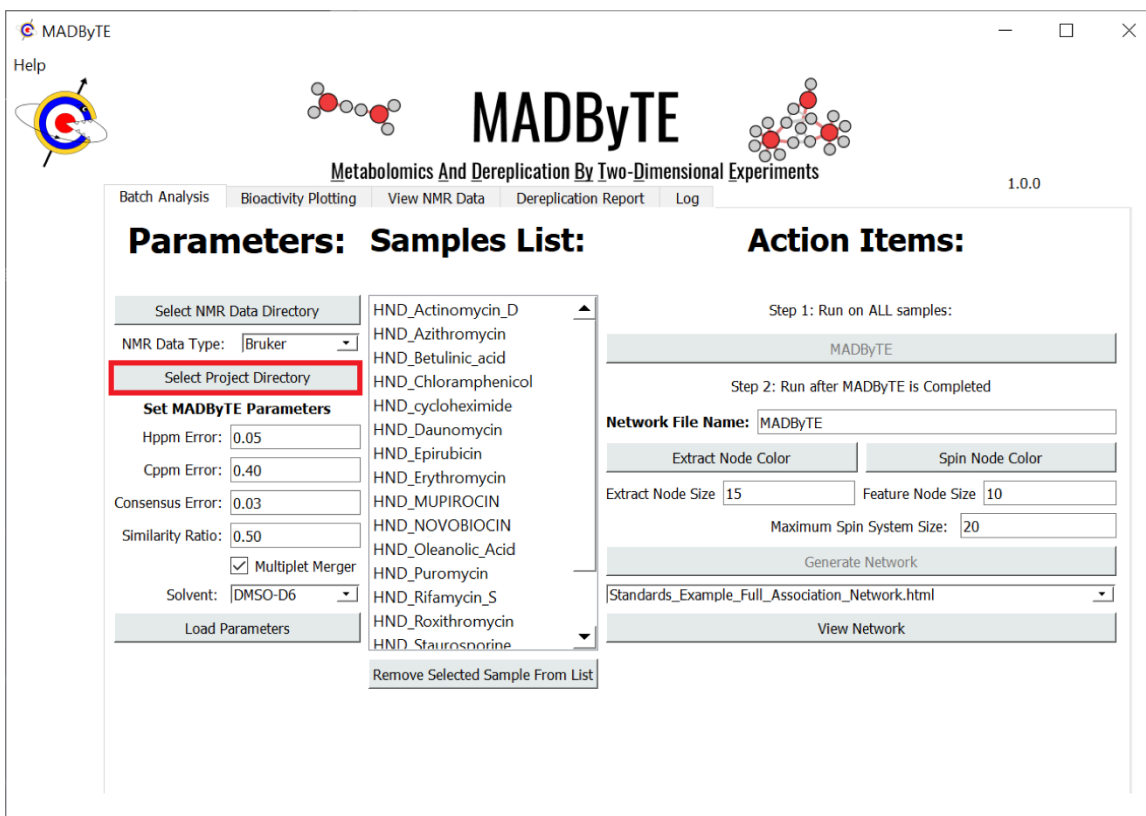
As a suggestion, create 'contextual' groupings to process. For instance, you may have data on 150+ extracts, but if they're all from different organisms and different chromatographic separations, they may not share any notable metabolites.

Select NMR Data Directory



This is where your RAW data is kept. In the case of Bruker, this will be your NMR data folder that contains all the directories topspin uses. In the case of MestreNova, this will be the directory you formatted to contain your peak picked lists (See above section – [MestreNova Data](#)).

When you select the directory, you will see the sample list populated:



If you have a sample in this list that you do not wish to run, simply highlight it and select 'remove sample'. If you accidentally remove a sample you wanted to keep, simply re-select the NMR data directory.

Select NMR Data Type

MADByTE does its best to be automated, but you must select the data type to process. In the drop-down list, simply select your data type. Currently, we support MestreNova peak picked lists and Topspin peak picking outputs. We are working towards implementing JEOL and NMRpipe data, but these are not yet fully developed.

Select Project Directory

This is the most important section to remember to keep track of. When MADByTE is finished processing, all the data derived, the graphs you will eventually generate, and the correlation matrix will be stored in this folder. So, give it a descriptive name and remember you can always delete folders later.

As a suggestion, if you were running a batch of 20 extract prefractions that were hits in an assay against MRSA, create a folder that describes that like "MADByTE_Analysis_Of_MRSA_Hits_date"

If you are using MADByTE to layer networks – see [bioactivity layering](#) - or if you are hoping to view the NMR data in the MADByTE plotting tool, you will need to select this data directory to manipulate the data.

Select Solvent

MADByTE is largely solvent agnostic, but the filtration steps on HSQC and TOCSY data are performed using the solvent as a guide. Peaks which are found in the ^1H chemical shifts of the selected solvent (and water region, if applicable) are not processed due to problems with T1 noise.

2.04: Processing Parameters and Filtration Cutoffs

There are a few important cutoffs defined in the GUI which will have a series of downstream effects in the processed data.

Hppm_Error

The Hppm_Error is defined as how far two points can be from one another in the ^1H dimension and be considered the same point. This is in reference to **both** the construction of the spin systems, as well as the matching of the spin systems to each other. Therefore, the tighter the restrictions, the better the data will agree in the final networks – however, restrictions set too tight will reduce the likelihood of matches in the final network.

To establish a basis for comparison, the maximum value of a scalar coupling was taken into consideration and assumed to be observable in an imperfect HSQC. Such that, you still see multiplet structure in an HSQC correlation. If this were the case, then a value of around 16Hz would be the maximum scalar coupling according to the standard Karplus plot. When converting the Hz to ppm for a 600MHz magnet, we find that a value can shift as much as 0.026ppm. Considering that this can happen in either direction (or in both directions in the case of a triplet), we double this threshold to be 0.05ppm. This is an estimate and works well for most applications. To establish your cut-offs, simply use the suggested equation below:

$$2 \times \left(\frac{\text{Maximum Hz Splitting Allowed (Hz)}}{\text{Carrier Frequency (MHz)}} \right) = \text{Hppm_Error}$$

And using a 600MHz magnet as a basis for this example:

$$2 \times \left(\frac{16\text{Hz}}{600\text{MHz}} \right) = 0.05\text{ppm} = \text{Hppm_Error}$$

Cppm_Error

Using the same logic as the above Hppm_Error, the Cppm_Error is how far off a carbon resonance can be to be considered the same. This restriction is more important when looking for analogs, as small changes in the scaffold can have a bigger effect on the Cppm values than one would expect. However, as a

suggestion, the same equation above can be used to determine the Cppm_Error cutoff but given that the gyromagnetic ratio for ¹³C is ¼ that of ¹H, the carrier frequency must be adjusted.

$$2 \times \left(\frac{\text{Maximum Hz Splitting Allowed (Hz)}}{\text{Carrier Frequency (MHz)}} \right) = \text{Cppm_Error}$$

Using an example frequency of 125Hz, we find that our values should be

$$2 \times \left(\frac{125\text{Hz}}{150} \right) = 1.6\text{ppm} = \text{Cppm_Error}$$

However, practically, this is a very wide window. So, for many of the applications that MADByTE was piloted on, we use 0.4ppm as the error window.

Consensus_Error

The Consensus_Error parameter is how close a resonance must be in the HSQC and TOCSY to be considered the same proton resonance. In a perfect world, this would be an exact match. However, running TOCSY pulse sequences tends to heat the sample a little, which can cause minor perturbations in the chemical shifts. As a generalizable parameter, we find that working with between 0.03ppm and 0.05 is a good starting point.

Similarity_Ratio

The similarity ratio is used after the calculation of the correlation matrix. This means that when all of the spin systems are defined, it compares the membership of each list to each other list based on the Hppm_Error and the Cppm_Error. These ratios are stored in the correlation matrix and used to network things together. The way the networking step works it fetches all correlations where the similarity score (value in the correlation matrix) is higher than the Similarity_Ratio cutoff.

So, considering a simple case:

Spin_System_1 = [(1.00,18.0),(2.15,24.6),(8.4,127.5)]

Spin_System_2 = [(1.04,18.2),(2.18,24.8),(8.6,129.0)]

Using a Cppm_Error of 0.4ppm and an Hppm_Error of 0.05, a pairwise comparison would yield:

	(1.00,18.0)	(2.15,24.6)	(8.6,129.0)
(1.04,18.2)	MATCH	NO MATCH	NO MATCH
(2.18,24.8)	NO MATCH	MATCH	NO MATCH
(8.4,127.5)	NO MATCH	NO MATCH	NO MATCH

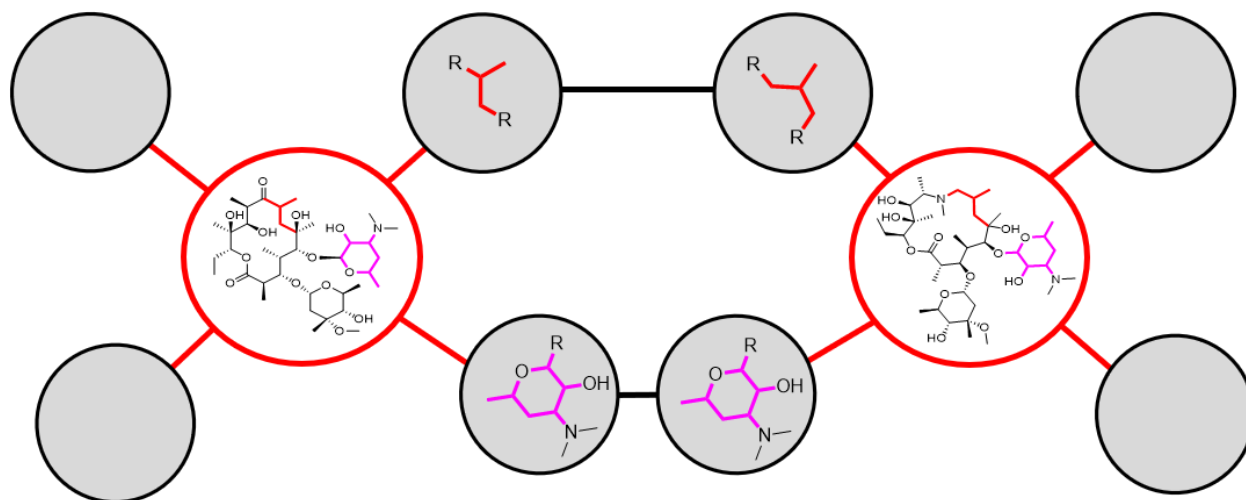
This means that 2/3 of the resonances match, yielding a similarity ratio between the spin systems of 0.66 (or 66%). If the similarity ratio was set to 0.5 (50%), the network would contain an edge between these two spin systems.

3: Networks

3.01: Generating Networks

Network Structure

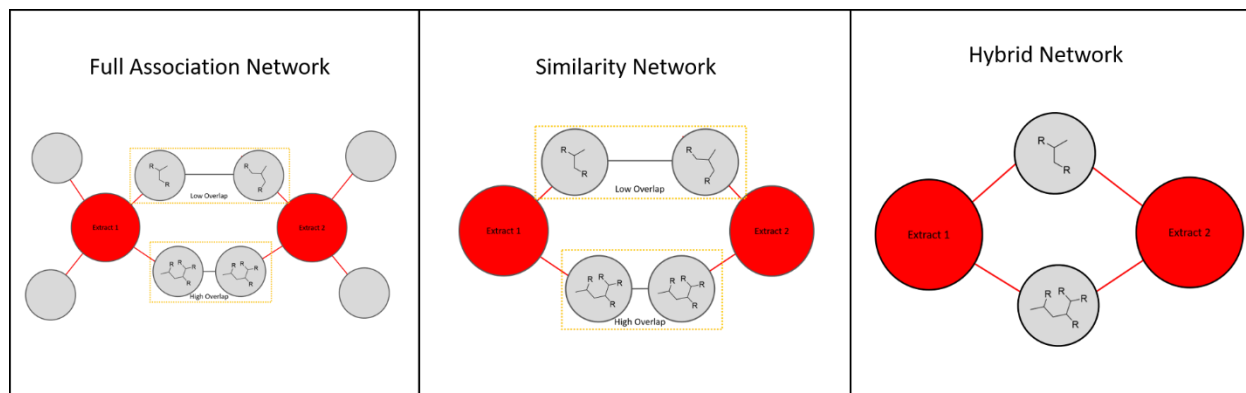
MADByTE Networks are collections of sample nodes and spin system feature nodes, tied together by their comparison scores. MADByTE looks for common proton and carbon resonances in the constructed spin systems and gauges the amount of overlap between two spin system features. If the overlap is greater than the similarity cutoff, then the spin systems are connected to each other.



In most networks, the strength of the connection is equal to the amount of overlap between the two spin systems. This means that if you have the same chemical motif conserved across several molecules (such as the desoamine moiety in azithromycin and erythromycin), the overlap will be very high, and the nodes should have more weight between them.

Types of Networks

MADByTE outputs three different types of networks, each with their own purpose.



Full Association Network

The Full Association network is a comparison network that renders all nodes for every spin system detected in every sample. After this, it draws edges between nodes displaying spectral overlap that is greater than the similarity ratio cutoff, and the weight between them is proportional to the similarity ratio between them.

Similarity Network

The similarity network is a reduced complexity mapping of the shared nodes **only**. If an extract or compound does not have any spin systems shared with the rest of the data set, neither its spin systems nor its extract ID node will be rendered. This enables a rapid viewpoint of the most conserved chemical motifs within the sample subset. As the comparison is still between nodes with >1 resonance, single resonance connections are maintained, as long as the requirement for overall similarity ratio between the spin systems is satisfied.

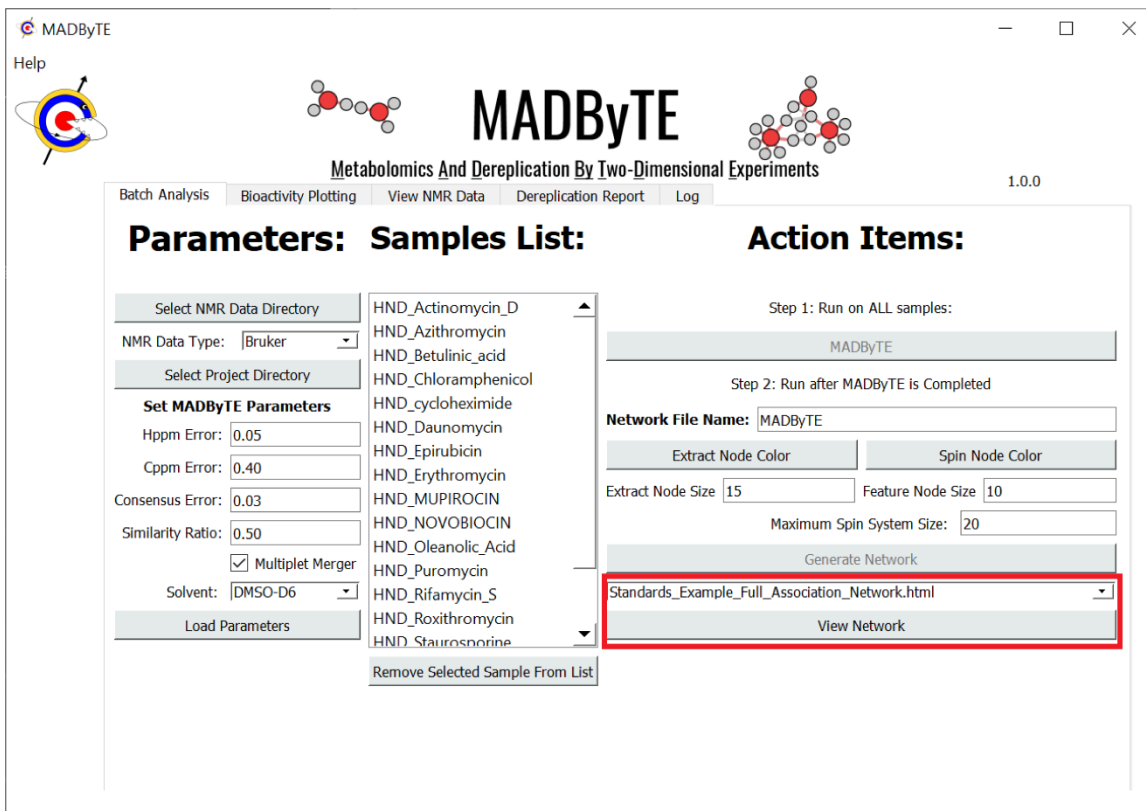
Hybrid Network

The Hybrid Network is a rendering of the Similarity Network that combines the connected nodes into their representative shared chemistry. The result will be a list of all points seen in all the similar nodes. This view drops nodes if there is only a single shared resonance between them. This is done because MADByTE was built to take advantage of spin systems, and single point connections do not satisfy this requirement.

3.02: Viewing and Manipulating Networks

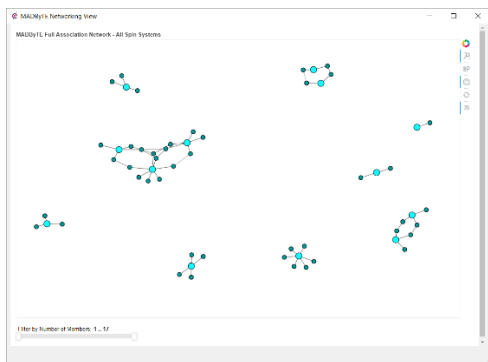
Networks can be viewed directly in the MADByTE system by navigating to the main “Batch Analysis” screen and selecting the network from the drop-down list and selecting ‘view network’.

- If the dropdown list is empty, simply select the project directory you wish to query, and the networks contained in that folder will populate the drop-down list.
- In most cases, the most recent network file will be at the top of the list, but this cannot be guaranteed across all platforms.



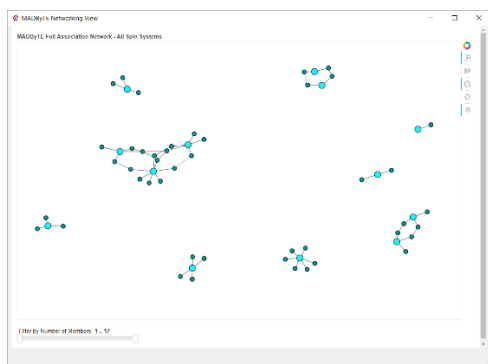
3.03: Network Viewer Plugin

MADByTE's native network viewer renders the HTML files generated by MADByTE in a pop-up window for easy navigation (Shown below – left). To filter the network by spin system size, simply drag the bar at the bottom of the screen to represent your requirements and the GUI updates automatically (Shown below – right).



3.04: Customizing Your Network

Networks generated by MADByTE can be adjusted based on your personal preferences. Color and size of the nodes can be adjusted simply by changing the values in the GUI. The standard layout is shown on the bottom left, and an alternate size and color scheme is shown on bottom right.



Extract Node Size: How large the extract ID nodes should be – default value of 15.

Feature Node Size: How large the spin system feature nodes should be – default value of 10.

Maximum Spin System Size: How large a spin system feature can be (in terms of the number of resonances allowed) and still be displayed. In some cases of exceptional overlap, large spin systems are constructed based on the Hppm and Cppm error values. If this happens regularly, we suggest shrinking these error values and running the analysis again.

3.05: Bioactivity Mapping

MADByTE contains a module for mapping bioactivity evaluations directly onto the sample ID nodes as a way to prioritize samples that show high bioactivity and shared structural motifs. To map the data correctly, there are a few things that must be considered.

Bioactivity Data Format

MADByTE allows for the layering of a single assay result on top of the sample ID nodes, but to do this it requires that the sample names match the NMR sample names. As an example, if sample “JE_Erythromycin” is the name of the NMR sample, it must also be named “JE_Erythromycin” in the bioactivity file. This means that dose responses **are not** factored in at this time.

The data must be saved as a CSV in the following format:

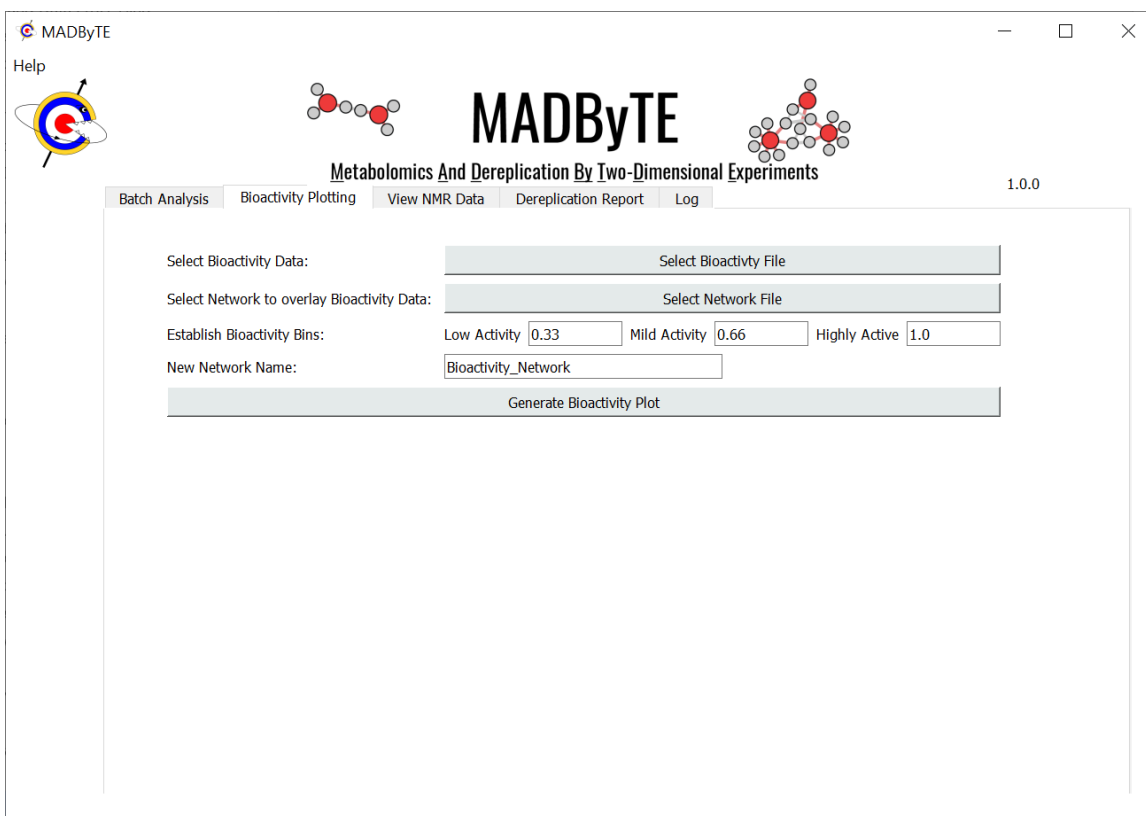
	A	B
1	Sample	Bioactivity_Score
2	HND_Azithromycin	1
3	HND_Betulinic_acid	0
4	HND_Chloramphenicol	0
5	HND_Daunomycin	0
6	HND_Doxorubicin	0
7	HND_Epirubicin	0
8	HND_Erythromycin	1
9	HND_Mupirocin	0
10	HND_Novobiocin	0
11	HND_Oleanolic_Acid	0
12	HND_Puromycin	0.5
13	HND_Rifamycin_S	0
14	HND_Staurosporine	0
15	HND_Thiamphenicol	0.5
16	HND_Roxithromycin	1

Typical Bioactivity Score Sheet For MADByTE Bioactivity Layering

Notice that the scale here is from 0-1, meaning that you must normalize your data to fit a linear scale. It **does not need to be** from 0-1, but the values will be important in the next step.

Mapping the Bioactivity to the Network

To map your bioactivity profile to your NMR data, simply navigate to the bioactivity plotting tab of the GUI and walk through the steps outlined.



The Bioactivity Plotting Window

Select the Bioactivity File

The bioactivity profile must be in CSV format, as outlined above.

Select Network File

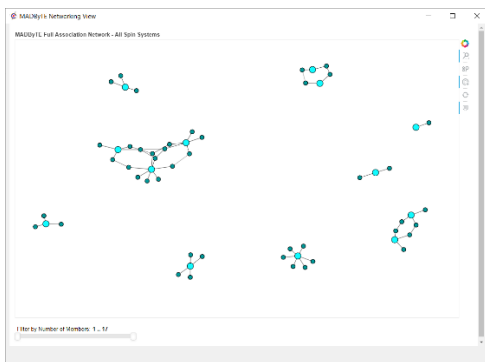
The Network File should be the graphml of the network that you wish to overlay. The bioactivity overlay can work with any of the networks generated by MADByTE.

Establish Bioactivity Bins

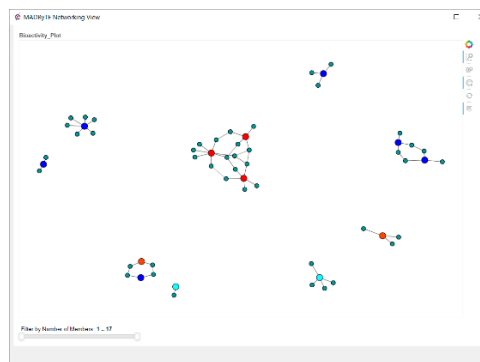
By default, MADByTE assumes a scale of bioactivity that has been normalized from 0-1. However, if your bioactivity is measured on a different scale, simply redefine the bins according to the bioactivity you wish to highlight. You can generate many networks that overlay different types of bioactivity data by simply renaming the file and regenerating the network.

Name the Network

The resulting network will be saved as an HTML file that contains the new bioactivity highlights.



A Typical Network From MADByTE



Bioactivity Color Coding Using The Bioactivity Integration Option

The “low” bioactivity compounds are color coded as blue, the ‘mild’ bioactivity compounds are color coded as orange, and the ‘highly’ bioactive compounds are color coded as red. In this instance, there were two compounds for which the bioactivity data is missing. For these compounds, the default color coding is retained.

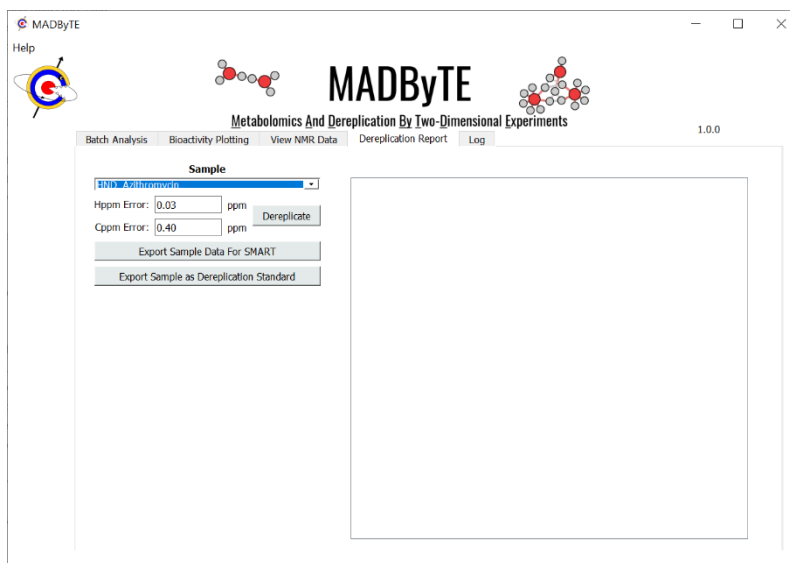
4: Dereplication

Dereplication, or predicting known molecular entities present in a mixture using reference data is possible with MADByTE using the Dereplication module. Using a database established by the user, molecular entities familiar to the user can be stored as reference files and compared against any data that has been processed by MADByTE, eliminating the need for isolation under certain circumstances.

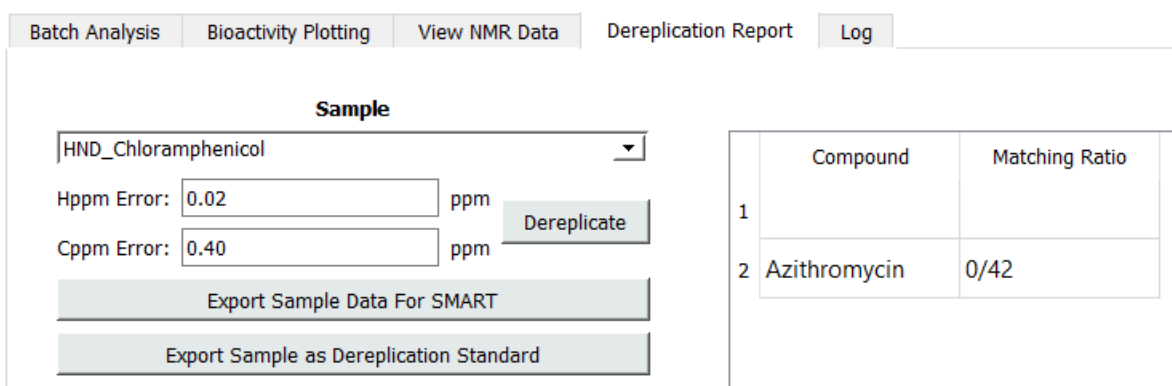
4.01: Using the Dereplication Library

To use dereplication, navigate to the Dereplication Report tab on the MADByTE GUI. There, you will see a sample drop down list populated by the samples you have processed through MADByTE previously.

- Select the sample you wish to query against the dereplication database
- Fill in the error margins you wish to use.
- Click Dereplicate and the results will be populated in the window to the right



Dereplication Report Tab



Dereplication Output and Options

4.02: Establishing a Dereplication Library

A major hurdle to effective dereplication is the lack of a curated and open centralized repository for data to compare against. Furthermore, a centralized server may be a security risk for some applications or users who wish to keep their data private and on a local machine. To facilitate dereplication under these circumstances, MADByTE offers the users a way to establish a local dereplication library which they themselves control and allow for curation.

Users can submit peak lists created by themselves, or from simulated data by storing them in the directory called Dereplication_Database within the MADByTE program folder. Once a sample has been processed by MADByTE, the system will be able to query the sample for signals that match reference compounds within user definable margins.

Submit a Peak List as a Reference for Dereplication

1. Peak pick the HSQC data for your pure compounds.
2. Save the peak picked data as a csv file with the format:
 - 2.1. H_ppm, C_ppm, Identity
 - 2.1.1. Report H_ppm to 2 decimal places
 - 2.1.2. Report C_ppm to 2 decimal places
 - 2.1.3. List the ID of the compound in the third column throughout the list
 - 2.2. Convert the csv to json
 - 2.3. Save the file as: DDF_[name of compound here].json
3. Place the file in the folder called Dereplication_Database, and you're done!
4. If you have legacy data from old publications, literature, or an in-house archive, simply format it as a json table and use the same naming convention in step 2.3.

Several sample compounds are included as a default database to get started.

Add New Data into the Dereplication Library Directly

MADByTE has an integrated method to directly submit sample data into the dereplication library. Simply process the data using the MADByTE analysis, and then navigate to the Dereplication Report tab in the MADByTE GUI.

Select the sample you wish to submit into the dereplication library and click 'Export Sample as Dereplication Standard'. This will output a list of the peak picked data used for MADByTE analysis directly into the Dereplication_Database folder and add it to the list of compounds to be searched against during the next dereplication event.

Batch Analysis Bioactivity Plotting View NMR Data Dereplication Report Log

Sample

HND_Chloramphenicol

Hppm Error: 0.02 ppm

Cppm Error: 0.40 ppm

Dereplicate

Export Sample Data For SMART

Export Sample as Dereplication Standard

	Compound	Matching Ratio
1		
2	Azithromycin	0/42

How to export the sample as a dereplication database file

Batch Analysis Bioactivity Plotting View NMR Data Dereplication Report Log

Sample

HND_Chloramphenicol

Hppm Error: 0.02 ppm

Cppm Error: 0.40 ppm

Dereplicate

Export Sample Data For SMART

Export Sample as Dereplication Standard

	Compound	Matching Ratio
1		
2	Azithromycin	0/42
3	HND_Chloramphenicol	8/8

Running the sample again after submitting a dereplication database file yields new results in the window

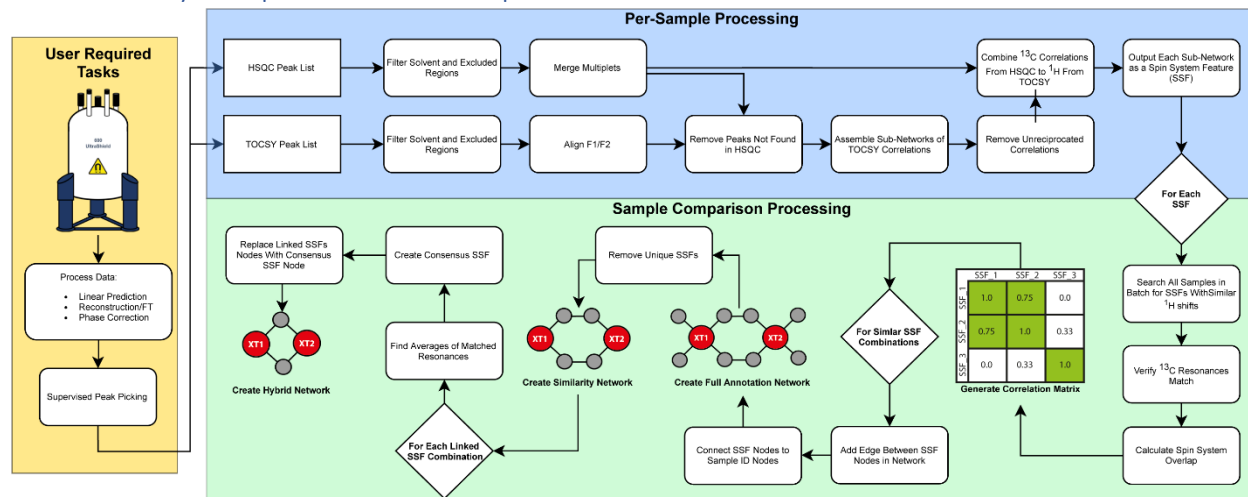
4.03: Other Dereplication Options – Integration with SMART NMR

Although there is no current standard for submission of NMR data into a centralized server, some groups have constructed fantastic resources that streamline dereplication protocols. One of these, known as SMART (Small Molecule Accurate Recognition Technology) is a webserver constructed and maintained by a team at USCD. SMART allows for users to drag and drop their NMR data and use an artificial intelligence-based tool to evaluate the spectra and return like molecules. To facilitate this as a dereplication option, we have integrated an export option into MADByTE. Simply click the “Export Sample For SMART” button and a SMART compatible peak list will be put in the data output from MADByTE allowing you to search SMART for molecules of interest.

Recommendation: MADByTE was designed for use on extract data – or mixture data- and when outputting a file for SMART, this system merely translates the peak lists into a SMART compatible format. There is no pre-processing done on these data, including solvent filtration or noise removal. Since SMART works best with pure compound data, we recommend using this option on datasets which are reasonably pure (I.E. – few compounds in the mixture). Therefore, users should be reserved about expectations from the results from SMART when submitting very complex data.

5.0: Advanced Documentation

5.01: MADByTE Input Data and Output Files



MADByTE Processing Steps and Network Outputs

Per-Sample Outputs

MADByTE has a number of output files which may be of use on a per sample basis. These output files can be compared to find out why a resonance was not considered, or why larger than expected spin systems have formed:

TOCSY Peak List: Input data formatted as described in relevant sections.

HSQC Peak List: Input data formatted as described in relevant sections.

Sample_Spin_Systems.json: A json formatted list of the Spin Systems found in the sample. Each coordinate pair is represented as a list containing tuples [(proton1,carbon1),(proton2,carbon2)....]

HSQC_DATA.json: A json formatted output of the HSQC data after multiplet merger and filtration.

HSQC_Pre_Decoupling.json: A formatted version of the HSQC data before multiplet merger.

HSQC_Preprocessed.json: A json formatted version of the HSQC input data.

TOCSY_Data.json: A json formatted output of the TOCSY data after alignment and filtration.

TOCSY_di_processed.graphml: (omitted from flowchart above) A resulting subnetwork of the TOCSY proton correlations which have been filtered according to MADByTE parameters. These TOCSY subnetworks are the spin systems features which MADByTE uses.

TOCSY_di_raw.graphml: (omitted from flowchart above) An initial subnetwork of the TOCSY proton correlations which have **not** been filtered for reciprocity or membership requirements. This is handy for

finding proton correlations which are assumed to be valid when looking at the data, but fail the filtering done by MADByTE.

TOCSY_Pre_Decoupling.json: A json formatted version of the TOCSY information after removal of self-correlations and solvent filtered zones, but before matching to HSQC information.

TOCSY_Preprocessed.json : A json formatted version of the TOCSY input data.

Experiment Specific Outputs

Correlation_Matrix.json/csv: an output of the correlation matrix calculated by MADByTE. CSV is for easy viewing in excel, json is used by MADByTE and comparison utilities.

Network Files: three networks are generated per MADByTE experiment, the differences are described in [Types of Networks](#). Each network is output as an HTML for easy viewing, and this is the network displayed when viewed using the MADByTE gui. The graphml files can be opened using other network analysis platforms (Gephi/cytoscape) for easier manipulation depending on workflow.

Spin_System_Master.json: A collection of every spin system identified in the sample subset including where it was detected and the membership of the spin system feature.

6.0: Acknowledgments

This work was supported by NSERC Discovery, NIH U41-AT008718, and NIH F31-AT010098. We thank Dr. Andrew Lewis and Dr. Eric Ye for assistance with NMR experiment selection and data acquisition necessary to build this utility.

This work was also heavily supported by Bruker, who readily replied to technical inquiries and offered advice in processing automation and experimental considerations.

We also thank the NMR and Natural Products community who volunteered to alpha test MADByTE.

MADByTE Alpha Testers

Laura Flores Bocanegra

George Peterson

Claire Fergusson

Trevor Clark

Timothy Bergeron

Rafael Reher

Kyo Bin Kang

Appendix A: Quick Start Tutorial

For this example, please download the sample data from <https://zenodo.org/record/3825107>. Next, unzip the data into a location you feel works well. Remember, the raw data and processed data should be kept separate. These data are already processed and peak picked, allowing us to skip the spectrometer side of things.

Next, launch MADByTE.

Under Parameters, select the *NMR Data Directory* (folder with the sample data in it) and a *Project Directory*. You can create a project directory with any name, so for this, we're simply going to create a directory called "Example_1". You'll notice that by selecting the NMR data folder, we can see the experiment directories of each of the compounds, if you wish to remove one, simply highlight it and select remove sample.

Under parameters, we can see some adjustable tolerances. We are going to leave these at default, except that we will adjust the *Similarity Ratio* to 0.51 (This means there must be >50% similarity to network).

Next, under action items, select MADByTE to run the analysis. If you have the terminal visible, you should be able to see the process working in the background.

Once completed a popup will appear saying the process has completed. At this point, all comparison is done between the samples, but we don't have a visual result. Let's fill out *Network File Name* to be "Example_1", and you may select node colors for both the extract and spin nodes, or you can leave them default if you wish. For this dataset, the prefix "HND_" signals to the system that these data are standards, so color assignments for the central node are re-written to teal. Once you are satisfied, simply click *Generate Network* and your data will be processed into the three network styles.

If you hover over the nodes, you can see exactly what resonances are in each spin system, and if that spin system is connected to any other spin system in your sample list. If you wanted to adjust the similarity cutoff, you can simply enter a new similarity cutoff and hit *Load Parameters* to read in the new values. You are not required to run a new MADByTE analysis to create a new network, but you are required to re-run if you adjust the Hppm or Cppm cutoff values, as these are used for multiple backend calculations.

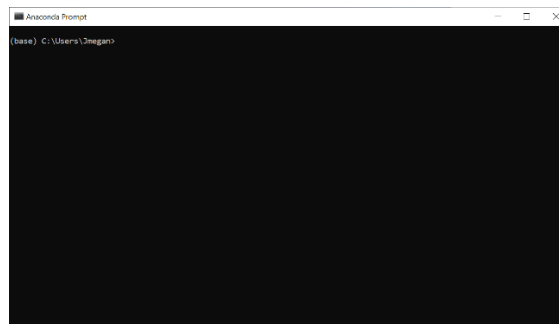
Appendix B: Directory Navigation Using a Terminal or Console

There are many types of consoles or terminals available, depending on your operating system.

Some examples include:

- Windows:
 - Command Prompt (if enabled)
 - Windows PowerShell
- Mac:
 - Terminal
- Third Party:
 - Anaconda Prompt

For this example, we will use Anaconda prompt, as it is installed along with Anaconda. When you open Anaconda Prompt, you should see a window similar to the one below.



Directory navigation is relatively simple once you know the commands. You can either move up or down one directory at a time using the command 'cd' followed by the folder you wish to navigate to – but that directory MUST be in that folder. In the example below, we navigate into a folder named 'madbyte' on the desktop. First, we navigate to the desktop using 'cd desktop' and then into the folder on the desktop by using 'cd madbyte'. If this was the master directory for MADByTE, then we could install or run MADByTE using the commands explained elsewhere in the 1.01: Installation and Launching MADByTE sections.

```
(base) C:\Users\Jmegan>cd Desktop
(base) C:\Users\Jmegan\Desktop>cd madbyte
(base) C:\Users\Jmegan\Desktop\madbyte>
```

If you accidentally navigated into the wrong directory, you can go up a directory by using 'cd ..' (example shown below).

```
(base) C:\Users\Jmegan\Desktop\madbyte>cd Wrong_Folder  
(base) C:\Users\Jmegan\Desktop\madbyte\Wrong_Folder>cd ..  
(base) C:\Users\Jmegan\Desktop\madbyte>
```