ODNP Workup Software

Ryan Barnes

13th July, 2015

Chapter 1

Introduction

This guide outlines, in very rough detail, the how to of operating the ODNP workup software with python. Right now this is fairly unorganized but I would appreciate your help in making this work towards what you need it for.

Chapter 2

Installation

2.1 Windows Install

This details how to install the necessary software on windows. This will also go through how to get the program setup and running.

2.1.1 Necessary Packages

1. Download & install 'python xy' *Make sure to do the full install instead of the defaul custom install.*
   (a) Install 'pymongo' do this by typing 'easy_install pymongo' in the commandline. Note that if the command line yells at you that easy_install is not a recognized command you need to edit the path variable to inclue the python directory in the program files x86 directory. Google is yo friend.
2. Download & install 'git'
3. Ask Ryan to share the 'workupSoftware' repository with you and follow the instructions for how to clone that repository to a directory of your choice.
4. Install vim74 *This is for me (Ryan) for when I go play on your computer. If you install this you get major brownie points :)*
5. Install notepad *This is for you and will make your life easier for when you need to edit the programs, which hopefully you wont have to do.* Or you can just use the native spyder editor which comes with python xy
6. Install latex via texlive for windows. Just run the simple install. *This takes a long ass time so start this early.*
7. Install pymongo, type "pip install pymongo" while logged in as an administrator.

Chapter 3

Guide to Running the Workup Code

Put instructions here for running the workup code. Initially go through the answers for the query statements, especially how to enter the dnp and t1 exp numbers. Then go through exceptions and how the user should use the command line to remedy the situation.

3.1 Getting Started

Here go through how to get up and running once everything is installed.

Chapter 4

Database Information

I'm using mongodb to database all the worked up data according to a hierarchical scheme. This will allow a user to search the database for entries based on certain parameters.

What I've learned after spending sometime working with the database is that organization is bloody necessary! Ultimately the key addition to the database needs to be strictly controlled, if it isn't then it's going to turn into a huge mess. The database keys that I currently am using are listed below.

Also note that when storing a value in the database use the full name of the protein and specify units for instance CheY should be written as 'Chemotaxis Y' and concentration should be '180 uM' or '1.65 mg/ml'.

I save the data in the database in this format 'data':'t1':'data':listOfData,'error':listOfError,'dim1':listOfDim1Vals,'dimNames':list
This allows me to nicely store the data without clogging the database keys with a bunch of crap.

4.1 Proper Databasing Format

Following proper format is **BLOODY FUCKING NECESSARY!!!** That being said let I outline below what proper format should be.

4.1.1 Macromolecule

Any macromolecule such as a protein or liposome should be entered in full chemical name. For instance CheY should be entered in as Chemotaxis Y, with the appropriate capitalization and spacing!

The first letter should be capitalized and the spacing between words should be adhered to.

A liposome such as DOPC should be entered is as 1,2-Dioleoyl-sn-glycero-3phosphocholine.

I realize this is a pain in the ass but this is why the workup script pulls your last entry for the databasing section of the code in hopes that you don't need to enter this in many times.

4.1.2 Concentration

4.2 Database Key Definitions

Currently the search headings include:

1. operator - this is the experimenter's name e.g. Ryan Barnes.
2. date - this is the date of the experiment
3. setType - this is the type of experiment, do not modify this as it's set by the code e.g. 'dnpExp' or 't1Exp'. Also if entering in worked up data such as a series of kSigma values or such this tag is set to 'seriesData'.

4. macroMolecule - this is the macromolecule in the sample e.g. protein (Chemotaxis Y, tau187), liposome (DOPC, DPPC). *If there is no macromolecule this is set to 'None'.*
5. bindingPartner - this is the other macromolecule in sample that is not spin labeled. *If there is no other macromolecule in the sample this is set to none.*
6. concentrationMM - this is the approximate concentration of the macroMolecule, this assumes the spin label is the same concentration.
7. concentrationMeasured - this is the measured concentration of the macromolecule either by DLS or UV-vis.
8. temperature - this is the temperature of the experiment set **in Kelvin!**.
9. solvent - this is the solvent for the experiment.
10. otherNotes - this is other notes useful for the experiment such as what the hell binding partner is or what series measurement this is part of. For instance I measure CheY in absence and presence of Urea to see what happens when I denature CheY. I say in other notes 'part of CheY denaturation experiment'.

4.3 Pulling Data from the Database

Right now this is notes to Ryan. I make a first attempt at pulling the ODNP data, specifically k_σ data from the database in the script *'returnDatabaseCheY.py'* .

This code is a mess but does what is necessary to store the k_σ data value and error in a nddata. Unfortunately the nddata set is not more help in storing the values. Currently I have to make a data matrix and an error matrix as well as dimension count lists and use these dimension count lists to dump the data and error from the database into the data and error matricies. I then take the full matricies and set this as my data for my nddata set. *You could fix the nddata so it sets the data and error value for the appropriate indecies given, currently it does not do this and is very frustrating.*

Also to note this code is functionally dynamic, so if any of the dimensions grow the code should handle it.

Another note is that the connection does not work from chembiochemGuest for whatever reason, but this brings up the point that you should house the database locally and sync the local database with the one online

This will silently fail if it over writes a value in a given matrix. You should add a check to make sure that a value doesn't already exist in the matrix position.

I think what I've learned is that it's not nice to drop everything in one matrix because you can't really handle any repeat experiment with any flexibility other than just hardcoding.

What you should do is loop through producing nddata sets given appropriate search parameters. This is the easiest way to do this for now.

Chapter 5

GUI Implementation

I barely start on this but do make mentionable progress. I save a script FileDialogueHandler.py that opens a GUI window with a button and a text line. When you click the button a file browser is opened and once you select a folder the text box displays the name of the folder.

The gui is implemented as newWorkup.py which makes use of the ODNPWorkup class in returnIntegralsDev.

I start a demo GUI for using a matplotlib widget along with Qt designer it's in the mplQtDesigner/main.py.

5.1 Reading from the database

I start a gui for browsing the database in browseDatabase/main.py.

15/06/23 I leave off with a method that can scroll the data base nicely and list what is currently available.

Chapter 6

Development

6.1 New Implementations:

1. ☐ When you save the series data all the experiment names should be saved with the corresponding data value as a separated dimension. This way you can refer back to what is what.
2. ✓ You need to wrap the bare code into functions and classes. Basically I want one function to call for doing all of the ODNP workup and another function to call for doing all of the EPR workup. The goal is to make the actual GUI code as clean and modular as possible. *this is done in returnIntegralsDev.py. You can call the code by 'import returnIntegralsDev' 'returnIntegralsDev.workupODNP(odnpName,eprName)'.*
3. ☐ You should also request that the user loads in a calibration file for the cw-EPR double integral experiment.
4. ☐ Change the entry query options to numerical, rather than, text entry. *What the fuck am I asking here?*
5. ✓ Include the EPR double integral workup as part of the software. *Now included in newWorkup.py*
6. ✓ Fix the file browser functionality so this doesn't crash so easily. *Now included in newWorkup.py*
7. ☐ Fit the T_1 raw data in the log space. This should be much more robust to weridness. Also you should plot the data in this space. *I don't think this is actually something you want to do.* What would be cool is with the GUI add a pop up window that allows you to adjust your guess for the fit.
8. ☐ Write the metadata to a file.
9. ✓ You should apodize your signal in the integrate function to boost the Signal to noise. *Right now this is automatically done by fitting the decay of the signal and multiplying by such*
10. ☐ You also need to go through the plotting from matlablike as this does not behave correcly.
11. ☐ in rb_dnp1 set a constant to a value such that you can use this as an identifier for figuring out if the experiment was run with the new version of the software. No, it shouldn't be that *'dnpconf'* should just ask for the starting power for the T_1 series.
12. ☐ Add function in returnPowers to accept or reject the first power based on the experiment time and the time spent at the first power, this also should have a tag for the T1 series.
13. ☐ Clearly write out the formalism for entering data in the database.
14. ✓ You need to change the rate fit to do a weighting based on the error of the T_1 value. *This is done*
15. ☐ Add script to take the output of the Atenbach project for EPR fitting and database this. This should save the data, the fit, and the fit parameters - specifically the correlation time and the A and G matrix. *This is not part of the workupSoftware!* Do not get ahead of yourself!
16. ✓ Make returnIntegrals work with new database module.

17. ✓ You should dump the worked up data sets (csv files and whatnot) into a directory called data.

18. ✓ You need to add a supplemental script so you can go back and update or change a given database entry, this should be located by the experiment name, pulled from the database. *This is in updateDatabaseEntry.py*

19. ✓ In database the 'setType' key should define either the experiment or the workup series…Maybe. *I added this so for dnp it says 'dnpExp' and t10 it says 't10', you can also now use this key as a label for series data sets.*

20. ✓ You should move all of the data in the database dictionaries to a data subfield in which you hold all relevant data sets. This way you could make the database search and drop something more dynamic and not depend on several if statements. *This seems to work now although it might have a few buggies.* Now the problem is to either update all of the existing database entries into the new scheme.

21. ✓ For the repeat database entry key. You should pull the repeats for the count of database items corresponding to the macromolecule, bindingpartner, and spinlabelsite. *I've changed the script that pulls the data sets from the database to do this.*

22. ✓ DATABASE: What you should do is loop through producing nddata sets given appropriate search parameters. This is the easiest way to do this for now. *This now works fairly well, I have a function to return a 1D nddata set and this seems to work ok for the moment.*

23. □ In the future you should play with using mongodb aggregate functions to make querying the database easier.

24. ✓ Sort database dictionary alphabetically before presentation.

25. / Add net magnetization plot to the output pdf! This is a nice check to see that the T1 power series makes sense. *No, I don't think this is anymore useful*

26. ✓ You need a couple of data sets from the emx odnp system to calibrate your code to. *This is done now that the EMX is updated to run the most recent code.*

27. ✓ You need to list the relevant ODNP parameters in the pdf produced. You should also make it so the T_1 value is written below the T_1 integral graph.

28. □ You should add a check when a user changes key value. This check should run through the existing keys values to make sure that the value has already been entered, if it is a new key value the program should ask if this new key value is correctly entered, the idea is to catch mistakes in spelling. *Another way would be to let the user choose from a list of options instead of having to type in the key value.* Actually I like the idea of a dynamic list for editing a key value instead of the user entering their own key values.

29. ✓ Open a file dialog so the user can pick their experiment directory and the experiment file to work up. *This is done in concept, the minimum running example is in 'fileDialog.py'. You just need to implement and wrap into 'returnIntegrals.py'*

30. □ Make it so the notebook shows the operators name, not mine. *I don't think this is going to work like it should. Something weird with the latex package calls. Right now it just says Han Lab Notebook.*

31. ✓ system calls that handle both windows and mac both in naming the file type and in compiling the pdf. *This does seem to work fairly robustly now.*

32. ✓ Make a way to reload the freshly produced pdf. Mac will work with preview (you just need to set to use preview by default). Windows should work with Sumatra pdf –> This seems to work for mac nicely, need to add windows features now. *Mac use preview and windows uses Sumatra*

33. ✓ Axis labels in the last plots generated by the code. This is important for kSigma because you want to give the appropriate units!

34. ✓ You should make it so that the database dictionary keys are pulled from the live database and fill in the key values from the default file in the directory. Or keep a copy of the directory keys and if the current keys are different than the directory keys just update the directory keys. *It looks like you will have to home roll a function to print all keys.* Keys are now pulled from the live database and the key values are filled in from the operator's last entry.

35. ✓ When entering database values you should list all possible choices for the given key. Use '.distinct('keyValue')'. *You should make it so the database values that are returned are operator specific.*

36. ✓ Add handlers for mistyped answers so program does not crash. Just say do not understand and re loop through the question in a while loop. *This still needs done for the error handling functions.* The error handling function need reworked I think. It was a good try but there is too much static code, it needs to be switched to dynamic type.

37. ✓ Change the experimental parameters queries to dynamic type like the database queries *This should be thoroughly debugged, it's not right now which might be a problem.* This actually seems to work nicely now.

38. ☐ Newton's method to find starting guess for T1 experiments. For now the t1StartingGuess works. You could also use lmfit...

39. ✓ Make a way to force an experiment time for diving up the powers files *I pull the experiment time from the Bruker output and set this as the minimum experiment time. If the experiment is successful the workup software should work fine dividing up the powers.*

40. ✓ You should calculate the experiment time from the Bruker timing function and hand this with an associated arror to the 'returnSplitPowers' function. *Right now this pulls the experiment time and uses as a lower bound for the split powers function.*

6.2 Bugs:

1. ☐ Integrate does not line up peaks correctly for the zero power measurements. This is not ok. I've noticed max_drift influences how this works however it does not correct the problem in a logical way. At the end of the day you should really re-write this so it behaves as you would expect in a manipulatable way.

2. ☐ The newWorkup.py crashes upon clicking 'run script' on windows balls... You need to figure out why this is.

3. ☐ Crashes when experiment numbers aren't set right. You should just automatically pick the experiment numbers based on the experiment title.

4. ☐ Returns powers error on r'150707_CheY_D41C_8MUrea_RT_ODNP'.

5. ✓ The EPR double integration falls apart for some reason for the 10 uM sample of 4 OHT. This could be a threshold problem, you must look into this. *This was dependent on how the peaks were picked. Switched to a new method that uses a top down search for finding the peaks and subsequent bottomup search for finding the valleys.*

6. ✓ In windows it depends how the program was launched. If mingw things run nicely if not it turns to crap. You should look into how to do all calls with python instead of using subprocess. All these problems are to do with the subprocess calls. Look into *shutil* I force it to use the cmd, the worst tool in the world but a consitent tool none the less.

7. ☐ When code soft exits in windows it crashes on the print statements. *You should test this, it should be fixed but I'm not entirely sure without debugging.* This actually still needs to be fixed but you need to use windows to recreate the problem.

8. ☐ First power missed in *150228 ÌĞCheY ÌĞN121C ÌĞNone ÌĞ181uM ÌĞYesUrea ÌĞRT ÌĞODNP* . It looks like the power picking fucntion got caught on the initial jump.

9. ✓ In the database repeat is saved as both string and int. This isn't ok - every parameter, other than the list of data should be a string. *now every key and value except the 'data' and '_id' are entered as strings.*

10. ☐ You need to move to the new figlist_var class of matlablike.py. Waiting to hear back from John on this one. *Right now the figures have no axes. This is unfortunate and needs fixing. I have a suspicion that this has to do with the order inwhich the matplotlib backends are brought in.* Wait for now as this isn't of major importance.

11. ☐ Code hard fails when the T_1 fit fails. This should at least give a warning, although adding newton's method should alleviate future failures.

12. ☐ I truly do not believe my T_1 error estimate from the covariance.

13. ☐ When working up a short experiment such as *'150211_4OHT_400uM_25pGlyH2O_250K_ODNP'* I get an error because it under estimates the maximum experiment time. For now it's just hardset to be 20 seconds longer than it should be.

14. ☐ RyanS_2015_1_30_psk1_DNP is a good reason to force experiment time for finding powers.

15. ☐ RyanS_2015_1_30_DOPC_postP188_110uM_tempo_DI_dnp crashes hard because dnpExps aren't set right.

16. ☐ in *'150128_CheY_M17C_None_202uM_RT_ODNP_REPEAT'* the T_1 powers function misses the first power. This is strange. For now I've just dropped the T_1 value.

17. ☐ The M17C set has a database entry for setType = kSigmaSeries. However there is no data stored in the database dictionary. This should not happen and if continues to will make a mess in the database.

18. ☐ There needs to be a better way to set the repeat counter. It's hard to tell what already exists in the database.

19. ☐ The way you compile the pdf is stupid. You should change this to speed up the code significantly.

20. ☐ You need to remove any instance of the database parameters file.

21. ✓ Add a check so that you don't overwrite matrix vales when you pull the values from the database. *Well it doesn't silently fail now but it still fails and I'm not sure of a good way to handle it other than writing a new matrix to hold the conflicted values.* More thinking to come.

22. ✓ For an old database entry the keys and values pulled do not contain any of the new items. This is to be expected but not ok because it does not allow you to enter information according to new keys…How to do this dynamically? *The set '141029_CheY_K91C_P2_320uM_RT_ODNP' currently has this error, I haven't fixed it so when you come to this use this set as your dummy set.* This seems to work ok now. It pulls the most recent dictionary entry and updates the experiment's database dictionary.

23. ☐ The database parameter checking should make use of the function returnDatabaseDictionary and you should hand it the necessary keys to drop!

24. ☐ You need to add a test operator and make sure the behavior of the code, primarily the database search, is changed appropriately. Implementation of high power amplifier: The initial spectrometer design incorporated a 10 W solid-state amplifier without a âĂIJblankingâĂİ switch after the amplifier to prevent excessive noise from reaching the receiver. Accordingly, the level of noise for pulsed measurements was larger than ideal. Additionally, for pulsed EPR measurements on realistic systems, the highest power amplifiers available (1 kW) must be used to achieve the most uniform excitation possible. To address these issues we have implemented a 1 kW TWT amplifier into our spectrometer. Because this amplifier is gated, the noise level after the pulses is approximately the same as thermal noise. As shown in Fig 6, the noise power after microwave pulses is reduced by 3 orders of magnitude by implementing the TWT amplifier.

    Figure 6: Noise power comparison relative to thermal noise of TWT (a) and solid-state amplifier (b). The noise power is orders of magnitude lower for the solid state amplifier because the TWT amplifier is gated.

25. ☐ When the internet connection does not exist the code hard crashes. Set a runtime warning when asking user if they want to database their information. Also set a soft crash for when code tries to access database without internet connection.

26. ☐ When the t1SeparatePhaseCycle is not set appropriately the code crashes. You should put in place a graceful crash that tells the user to correct t1SeparatePhaseCycle.

27. ✓ If the experiment exists, the database parameters file should be pulled from a search given the experiment name and nothing else. Then you should pop the dims that you don't want to see.

28. ✓ The returnIntegrals opens two connections to the database. You should make it so that only one connection is opened, and that the returndatabase parameters accepts a connection instance instead of creates it's own.

29. ✓ If you edit the zeroth item in the dynamic options menu, it breaks the loop and goes on for some reason. *This happened because you by chance set answer to False by setting it to zero.* Fixed now. This was also a problem with the experimental dictionary but is now also fixed.

30. ☐ Windows!! You need to debug that bitch!! *This is almost done, you just need to finish with the last of the sys calls.*

31. ✓ When the workup experiment crashes the database parameters dictionary is not saved. This is because it's pulling those parameters from the online dictionary and not from the file. *You should make it so you hand the returnDatabase function the name of the experiment. If the experiment name exists just hand back that database entry.* It now is.

32. ☐ You should add a method to force a time for the power series. You should print the absolute experiment time underneath the curve so the user can select a start guess and a forced time.

33. ☐ 'returnExpTimes' fails in windows when the experiment files aren't set right. You need to make it so it fails in a way that the user can fix it.

34. ✓ 131115_tempcont_dnp_9_58GHz_jss gave power series error, it looks like the time steps are really off. *This is actually just a broken experiment.* No need to adjust code to handle this. I did add functionality to the powers dividing function to drop the first number of time values with 'timeDropStart' and I also added a maximum experiment time as 'expTimeMax'. It might be worth adding debugging functionality to plot the time series and also show the time values of the spikes, this way the experimenter can pick out the values nicely.

35. ✓ 140509_200uM_OHT_ODNP gave an error because a glitch occuring early. You should add a way to throw away values that occur before a certain time. *This seems to work now.*

36. ✓ 140728_CheY_CtermP6C_400uM_ODNP_Repeat2 this throws an error because it can't line up the enhancement series and the powers file. *This seems to be resolved. I think an earlier fix to the powers picking function fixed this, mainly choosing a minimum experiment time.*

37. ☐ When code crashes due to not lining up the powers file with enhancemnet or T_1 series. The powers file is not returned in a csv.

38. ☐ You need to add wrong answer handling to the power series exception handlers.

6.3 Methods to Implement:

I start a module *'database.py'* to contain all of my database helper functions. Note I need to move any database stuff I have to this file.

1. ☐ You should wrap something to pull from mongo and dump the meta data and data to separate csv's in the same file.

2. ☐ You should also write a method to go from this csv format to the database such that you can add meta data to file

3. ✓ Wrap method of saving the nddata sets to mongo database *This is done in a very janky way right now.*

4. ✓ Method for print statement headers

5. ☐ Method to save an nddata to csv file.

6. ✓ Method to update a database entry - this has to be done off of the experiment name. Right now this is stored as barescript *'updateDatabase.py'* I move this function to 'database.py' named modDictVals()

7. ☐ Function for changing the keys and values of a dictionary variable, I can see that you'll soon be making calls to this function repeatedly.

6.4 File Organization

The way you store the style files for tex and your modules for python need some help.

6.5 Users Guide:

1. ☐ Windows installation of packages and use.
2. ☐ Mac installation of packages and use.
3. ☐ Introduction to the software.
4. ☐ Walk through of using the program.
5. ☐ Common errors, reasons and how to handle them from the command line.
6. ☐ Explanation of functions used *This is a backend thing and is not necessary for now.* Actually you just need to make appropriate doc-strings for the functions specifically in matlablike

Chapter 7

EPR Integral Workup

Is this necessary for the ODNP? It might be nice to do this as standard this way you can tell if the double integrals are the same. *This is a future idea.*