autofit Documentation

Release 1.0

Nathan Seifert

August 01, 2013

This is currently a work in progress!

Contents:

CHAPTER

ONE

INTRODUCTION TO AUTOFIT

Welcome to the Autofit documentation!

1.1 Authorship

The original implementation and idea for Autofit was developed in the lab of Brooks Pate at the University of Virginia. However, the modern Python-based implementation has primarily been developed by Steve Shipman, of New College of Florida, and Ian Finneran, a former undergraduate student of Steve and now a graduate student in Geoff Blake's group at Caltech. General maintenance, support, marketing, etc – as well as tender loving care outside of programming – has been primarily done by the Pate group at UVa.

1.2 Motivation

1.2.1 Scientific

In short, Autofit is an automated triples fitting tool for the purpose of identifying unknown target spectra in a broadband rotational spectrum. Since the implementation of high speed digital electronics has become routine in broadband microwave spectroscopy, new tools were required in order to decouple spectra of low abundance molecules out of broadband scans that are increasingly becoming more sensitive and spectrally dense. Some of the most dense spectra taken in the Pate lab have dynamic ranges of over 20000:1, and with line densities of over 1 MHz^{-1}. These kinds of line densities make it unfeasable to manually fit spectra using traditional visual cues, especially for species such as isotopologues which are often components of the dense "weeds" seen in deep-averaged rotational spectra.

1.2.2 Computational

Originally, Autofit was developed as a script for PTC's Mathcad. Although Mathcad is easy to use, it is sadly proprietary, not used by the majority of physical chemists, and is rather unstable (and extremely slow!) for computation with large data sets. Python is a natural choice for applications such as autofit, since it is well supported by the scientific community, it is open source (and the majority of useful scientific and mathematical libraries are as well), it is lightweight and completely portable, and the language itself has a remarkably small learning curve for scientists with respect to other languages such as FORTRAN and C. Additionally, it is relatively easy to scale computations across multiple CPUs or CPU cores using Python, which is a necessity for making Autofit an efficient tool.

1.3 What Autofit Can Do for You!

As stated in the motivation, Autofit is an automated triples fitter for broadband rotational spectra. As a program, it merely serves as a generator and processor of input and output files for Herb Pickett's legendary SPCAT/SPFIT program package, which can be found on the JPL website. Since Autofit is built exclusively in Python, the only functional requirements are compatible binaries for SPCAT and SPFIT and a Python interpreter for whatever platform you want to run Autofit on. Here is, in a jiffy, what Autofit does:

- Given a set of ab-initio or guess rotational constants and an experimental spectrum, Autofit will fit a A/B/C rotational constant triplet to every possible set of 3 lines possible within an error window.
  - For example, if you want to fit the 2_{02}- 1_{01}transition, which is predicted to be at 10 GHz, with an error window of 300 MHz, Autofit will check every experimental line between 9.7 and 10.3 GHz.
  - In this example (and in general), Autofit will do this for the three transitions you choose, and then output a sorted list of A/B/C triples ranked by goodness of fit (which is calculated by fitting an additional set of transitions to build up an effective fit)
- For a given A/B/C triple output by Autofit, Autofit can take that triple and refine the fit with distortion and/or additional transitions in the fit.
- For a given set of experimental rotational constants for a parent species, and an ab-initio structure for the parent species, Autofit can search for isotopologues separately and automatically by scaling predicted constants via the ab initio geometry.
- In principle, Autofit can scale to an arbitrary number of CPUs or cores. It does this by dividing the total list of triples to check into N segments, where N is the number of CPUs/cores being used in the calculation.
  - As it stands, Autofit can fit around 35-50 triples a second per core on a typical last-generation Intel CPU (Ivy Bridge). So for a 4-core (8 core via Hyperthreading) system, approximately 250 triples a second can be fit. A typical Autofit run of 10 million triples therefore takes around 10-12 hours.
- Although this is currently in development, Autofit will soon be integrated into a GUI where choosing transitions to check for Autofit and analyzing Autofit results will be possible, therefore removing the requirement of using an additional program (such as JB95 or PGOPHER) for checking Autofit results by hand.

1.4 What Autofit Can't Do for You!

Rotational spectroscopists, wipe the sweat off your brow – Autofit is not here to make your job obsolete. Rather, Autofit has been designed to speed up the purely mechanical (and often mind-numbingly painful) process of fitting spectra. For instance:

- Autofit does not fit nuclear quadrupole hyperfine. If you run Autofit on a molecule, such as one containing nitrogen, you will often find multiple fits that are roughly close to each other in RMS error. This is due to the fact that Autofit finds the same rigid rotor spectrum, but fits to different resolved hyperfine components.
- Internal rotation is not considered at all. For fitting a rigid rotor spectrum with Autofit, you will usually get at least two fits: one with the A state fit, which will fit to the rigid rotor model closely, and a higher energy E state fit. More than likely you will get fits where A and E transitions are mixed, albeit at a higher RMS error.
- Autofit does NOT fit distortion! You can enter in distortion at the beginning for SPCAT to consider, but it will be treated as a static variable in the predictions. There is functionality to take an Autofit result and then go and fit distortion by adding additional lines to the fit, but this is effectively an automated tool for doing "final" fits in SPFIT. None of it is done automatically during the autofitting process.

And most importantly, though this point is becoming less important as we continue to refine the program to make it easier to use:

- Autofit does not making fitting rigid rotor spectra a mindless activity. Some knowledge, or at least common sense, about what rotational transitions are good choices as triples fitting parameters is required. Mindlessly choosing transitions from the predicted list will almost never give you good results. In the Pate group, we have a saying that has become true over and over again: **Autofit does not fail you, only you fail Autofit!**

However, fear not – there are a number of helpful tips and tricks available in this documentation that will ease your Autofit journey. For a trained rotational spectroscopist, use your best judgement – what combinations of lines do you typically use to get a good rough fit on A, B and C?

1.5 Licensing

As it stands currently (as of August 2013), Autofit is licensed with the GPLv2 license. This is temporary, as it is likely we will move to a more liberal license in the future. You can find the actual terms of the GPLv2 license here.

1.6 Support

Currently, maintenance and support is managed by Nathan Seifert of the Pate group. He is happy to take any questions you might have on the program, both in terms of scientific and computational support. You can contact him at **nas3xf[at]virginia[dot]edu**.

CHAPTER

TWO

GETTING STARTED WITH AUTOFIT

Okay, so hopefully the introduction hasn't turned you off from the idea of fitting spectra automatically (though if you're reading this I'm sure you're still quite interested!). One of the motivations of writing this documentation is not only to teach a new user how to use Autofit, but also to provide a detailed repository of information and verification for those who are still skeptical of the method. Rotational spectroscopy is uniquely beautiful amongst the spectroscopic techniques, and the amount of rich chemical information we can gather from a simple microwave spectrum is often incredible. However, we as a scientific community have gotten to the point where the absolute amount of information we can gather from a single spectrum is beyond the capabilities of efficient analysis using manual fitting techniques.

This is where Autofit comes in. We have been using this program in the Pate lab for a couple of years now, and we have attributed much of our speed, efficiency and success to intelligent use of Autofit. This would be an apt time to say that "the proof is in the pudding," but it is more instructive for us to educate and have Autofit prove to you its incredible potential as a essential tool for broadband rotational spectroscopy, hopefully alongside Pickett's CALPGM suite and the vast majority of tools on Zbiginew Kisiel's PROSPE.

2.1 Prerequisites for Autofit

- Python 2.7.5 – Currently, as standard with most scientific applications in Python, Python 3.3 is NOT supported, though in reality this is due to the required additional libraries listed below not necessarily supporting 3.3.
  - Python 2.7.5 for Windows can be downloaded here.
- Numpy / Scipy
  - Numpy is available here.
  - Scipy is also available here.
- Matplotlib, available here.
- Easygui, available here.

For the purposes of creating/moving files, Autofit uses *bash shell* commands. Therefore, Autofit will *not* work in the standard Windows command prompt. You need to use a bash shell emulator. We recommend either of the two listed below:

- Cygwin, available here. NOTE: There are a lot of options in the Cygwin install, but Autofit will run out of the box in a default Cygwin installation.
- The mingw32 distribution included in Git for Windows, available here. This is a nice package in case you want to keep up to date with our Github repository for the latest updates to Autofit.

If you want to use the experimental GUI programs currently being developed, you will also need PyQT4, binaries of which can be obtained here.

- SPCAT / SPFIT. There are 64-bit binaries available on the JPL repository but they pop up these annoying windows that will spam your computer when you run Autofit (in fact they'll pop up each time you fit a triple, so 100-250 times a second!). **Instead, we recommend using the 32-bit binaries available on our Github repository, listed in the next section.**

2.2 Obtaining Autofit

You can find the latest version of Autofit and the experimental GUI at our Github repository.

If you're comfortable with using Git, you can clone our repository by entering in the command:

```
git clone https://github.com/pategroup/bband_scripts.git
```

Otherwise, here are direct links to the most useful programs in the repository for the average user:

- Autofit, stand-alone: Current version: v15
- SPCAT.exe available here.
- SPFIT.exe available here.

There are many other useful scripts available in the repository, including some basic workup scripts for broadband spectra and the new experimental fitting GUI/Autofit analysis program. Feel free to browse the repository and see if anything catches your eye (the README files in the root directory and in the autofit directory give details about what each script in the repository is).

In addition, there are SPCAT/SPFIT binaries that are tested (and working) in x86 (32-bit only!) Linux, as well as a tested but older version of autofit (v9) for Linux available in the /autofit root of the Github repository. If you are running 64-bit Linux, these binaries will NOT work unless you have installed the proper libraries (e.g. ia32libs in Ubuntu) for running 32-bit binaries in 64-bit Linux!!!

In terms of best practices:

- Include SPCAT.exe / SPFIT.exe in a new folder with the Autofit program. Autofit will create new directories for each job you start with it, so make sure you have permissions to create directories.
- When you start a new job in Autofit, Autofit requires that the job name (and hence the directory name) is unique. So if you already have a directory in your autofit folder called "/foobar" from some old run and you want to start a new job called foobar, make sure to delete the old /foobar folder.
- Autofit is completely CPU dependent and not all that reliant on RAM availability, so if you have a computer that likes to overheat when you work the CPU hard, either avoid running Autofit on all available cores or find a better cooling solution.

2.3 Notes on "Out of the Box" Performance with Autofit

- Running Autofit in an environment such as IPython or Spyder is probably doomed for failure. Always run in a simple bash shell by using the command:

  ```
  python prog_A_vX.py
  ```

- Since ALL of the testing done on Autofit has been on either Windows 7 or Ubuntu/Debian, functionality for Autofit on OS X is unknown. There have been some preliminary suggestions that it does NOT work out of the box on OS X. This could be due to the fact that the OS X terminal is not bash, but rather tsch (unless this has changes). Changing the terminal to bash could alleviate these issues.

- We find that the best out of the box performance is gained by installing a very simple Python environment and avoiding use of distributions such as Enthought. It could very well work just fine on Enthought or Anaconda, but installing the packages listed above to create a very minimal Python environment tends to work 100% of the time. Again, feel free to e-mail us if you have issues with this.

CHAPTER

THREE

AUTOFIT: THE FIRST TIME IS ALWAYS YOUR BEST TIME

This is filler for later!

3.1 Blah blah

CHAPTER FOUR

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*