WALLABY Kinematic Analysis Proto-Pipeline User Guider

N. Deg, K. Spekkens on behalf of TWG5

9 March 2022

1 INTRODUCTION

This document is meant to guide someone through the use of the scripts used to generate the WALLABY Pilot Phase I kinematic models. A full description of the observations and analysis are found in Westmeier et al. (2022) and Deg et al. (2022). This proto-pipeline (herafter WKAPP) is relatively straightforward to use and modify for fields beyond the initial fields analyzed in Deg et al. (2022).

2 SOFTWARE REQUIREMENTS

There are a number of pieces of software required for WKAPP. Most of the scripts contained here are python3 scripts and require somewhat standard libraries. These include Numpy (Harris et al. 2020), Scipy (Virtanen et al. 2020), Astropy (Astropy Collaboration et al. 2013, 2018), Pandas (Reback et al. 2020), and Matplotlib (Hunter 2007). These can all be installed via pip commands.

There are two key kinematic modelling algorithms that are required for WKAPP. 3DBAROLO (Di Teodoro & Fraternali 2015) is available at 3D-BAROLO. It must be installed on the same machine as the WKAPP scripts and the path to the code will need to be defined in certain modules (see Sec. 6.

The second kinematic modelling algorithm required is FAT (Kamphuis et al. 2015). This code is available at FAT. FAT itself relies on the TiRiFiC code (Józsa et al. 2007), which is available at TiRiFiC. Due to the machines that WKAPP was developed and tested on, FAT and TiRiFiC may be installed on a different machine than where the WKAPP scripts are located. The details on how FAT is used in the context of WKAPP are found in Sec. 7.

A third piece of software required for WKAPP is MCGSuite (Spekkens et al., in prep). This code is available at MCGSuite. It also needs to be located on the same machine as the WKAPP scripts and the paths will need to be defined in some of the modules.

3 QUICK START

The workflow for a user who wished to use WKAPP to generate models like those made in Deg et al. (2022) is:

(i) Set the particular field definitions required in the files contained in the *ReleaseConfigurationOptions* folder.
(ii) Run *WallabyPreliminary.py*.
(iii) Run *Wallaby_BasicAnalysis.py*.
(iv) Run *Wallaby_BaroloAnalysis.py*.
(v) Run *Wallaby_FATAnalysis.py*.
(vi) Transfer the appropriate files/folders to a machine capable of running FAT and do the FAT analysis.
(vii) Transfer the FAT results back to the WKAPP machine and specify the location of these results.
(viii) Run *Wallaby_BasicFitComparison.py*.
(ix) Visually examine the diagnostic plots produced by the fit comparison script and make a fit success catalogue file.
(x) Run *Wallaby_ModelAveraging*.

An explanation of the overall structure and each of these individual steps is laid out in the following sections.

4 PROTO-PIPELINE OVERVIEW

For the pilot phase I observations we have generated a proto-pipeline to go from the outputs of the WALLABY source finding analysis (Westmeier et al. 2022) to the kinematic models found in Deg et al. (2022). Unlike a full pipeline, the proto-pipeline is broken into multiple steps that require user interventions. The steps in the proto-pipeline are:

(i) Pre-processing
(ii) 3DBAROLO Analysis
(iii) FAT Analysis
(iv) Fit Comparisons
(v) Model Generation

Each of these steps is controlled by one or two scripts found within the WKAPP folder.

In addition to these general steps, there are a number of field specific definitions that must be set for each run. These are found in the */ReleaseConfigurationOptions* folder. Within that folder, the files *AnalysisOptions.py* and *FolderAndFileDefs.py* must be set to import the correct field definitions. The definitions themselves are set in the *HydraField.py*, *NormaField.py*, and *NGC4636Field.py* modules. These definitions include the location of the input data files, the locations to save various files, a variety of labels, etc. Additional fields can be added by simply setting the appropriate definitions for a new field and importing them into the configuration options files. A key thing to note is that the path to the 3DBAROLO executable must be set in the field definition files. For MCGSuite, the path to the executable is set in */MCGModel/MCGModelGeneration.py*.

Once the field specific definitions are set for your machine and selected in the configuration files, the WKAPP scripts can be run to produce kinematic models in the same manner as those released in Deg et al. (2022).

5 PRE-PROCESSING

The outputs from the source-finding analysis described in Westmeier et al. (2022) are used as inputs to WKAPP. They consist of a catalogue file and a set of cubelets, masks, moment maps, and profiles for each galaxy. The cubelets themselves are in frequency units. The first step in WKAPP is converting these to velocity space.

The velocity conversion is done via *WallabyPreliminary.py*. Not only does this do the velocity conversion, using the frequency at the center of the cubelet (not at the reference point indicated by the cubelet header), it also uses 3DBAROLO to smooth the cubelets to a velocity of ~ 12 km/s. Both the smoothed and unsmoothed cubelets are saved to a location specified in the configuration files. It is possible to run WKAPP using either of these cubelets for the FAT and 3DBAROLO analysis.

Once *WallabyPreliminary.py* is run, there is an additional pre-processing script to run; *Wallaby_BasicAnalysis*. This script does a number of calculations on both the SoFiA catalogue values and on the cubelets themselves and saves them to a csv file. While there are a number of things that this script calculates (distance, mass, predicted R_{HI}, volume elements), the most important ones are the conversion of the SoFiA sizes to beams and the calculation of the S/N. Both the size and the S/N are used when choosing a sub-sample to attempt kinematic modelling.

6 3DBAROLO ANALYSIS

The next step in WKAPP is the 3DBAROLO analysis. This is done via the script *WALLABY_BaroloAnalysis.py*. This script directly runs 3DBAROLO on each cubelet that satisfies the size and S/N constraints. As such, it is necessary that 3DBAROLO be installed and the path to the executable must be specified in the particular field definition file. The script will store the outputs into the folder specified in the same definition file. It is possible to run this on either the 4 km s^{-1} or 12 km s^{-1} cubelets. However, it is recommended that the user analyzes the 12 km s^{-1} cubelets as the smoothing improves the 3DBAROLO performance.

It is worth noting here that this wrapper around 3DBAROLO adjusts the basic 3DBAROLO performance slightly. We have found that the version of 3DBAROLO used in the development of WKAPP occasionaly failed when using the 'Two Stage' mode needed to generate flat disk models (the main source of failures is that the PA of a ring or two would be correct but greater than 360°, which would affect the averaging in the second step). These issues have likely been addressed by the 3DBAROLO authors in current versions. Nonetheless, the *WALLABY_BaroloAnalysis.py* does not use the built-in 'Two Stage' mode for the flat disk models. Rather it runs 3DBAROLO once with all parameters free. It then does the geometric averaging from the outputs of the first 3DBAROLO run. This averaging corrects for issues like enforcing that the PA is between $0 - 360°$, or the possibility of some rings having poorly constrained V_{sys} measurements. Then 3DBAROLO is run a second time with the geometry values fixed to the average values from the first run.

7 FAT ANALYSIS

The FAT analysis in WKAPP is set up, but not run, via *WALLABY_FATAnalysis.py*. This script uses the field definition file to organize a folder structure suitable for a FAT analysis, as well as generating the catalogue file needed for FAT . As with the 3DBAROLO script, this can be set to use either the 4 km s^{-1} or 12 km s^{-1} resolution cubes. Unlike 3DBAROLO FAT does not show a significant difference in the flat disk models for either resolution, but it does run faster on the smoothed cubes.

The reason this script does not run FAT directly is that FAT is a relatively slow code. In the workflow used for generating the models contained in Deg et al. (2022), FAT was installed on a different machine than the machine containing the WKAPP scripts. Thus, once *WALLABY_FATAnalysis.py* was run, the folder structure was transferred to the machine where FAT was installed. Then FAT was run using the catalogue file generated by this script. Once FAT completes it's analysis for a particular field, the results must be transferred back to the research machine and the location of the results must be specified in the particular field definition file.

8 FIT COMPARISONS

Once the 3DBAROLO and FAT runs are complete for a given field, it is necessary to visually inspect the results in order to determine whether the fits are acceptable for use in the final models. To add in this task, the script *Wallaby_BasicFitComparison.py* generates diagnostic plots for each galaxy analyzed in the fitting process. These diagnostic plots show the 3DBAROLO and FAT best fitting parameters for the rotation curve, surface density, and geometric parameters. It also includes the estimates on the geometric parameters from the SoFiA analysis that made the cubelets in the first place. If either the FAT or 3DBAROLO fits fail, the script will still generate a diagnostic plot that will contain either the fit that succeeded or empty panels if both codes fail.

The user will need to examine each fit to determine whether a fit is successful or fails. For WKAPP, a key requirement is that both FAT and 3DBAROLO produce realistic models for the galaxy. During this examination, the user will need to create a CSV formatted file that contains columns labelled "NAME", "ID", "USABLE_MODEL", and "FLAGS". The file must also contain entries for every galaxy in the field. This file name must then be set in the particular field definition file as the variable **FitSuccessCatalogue**.

It is important to note that only galaxies that have a "1" in the entry for the "USABLE_MODEL" column will be included in the final model catalogue at the end of WKAPP.

The user can specify their own system of flags in the fit success file. For Deg et al. (2022), we used the system:

- Class 0: No obvious issues
- Class 1: The final model has an inclination above or below 75° and 20° respectively.
- Class 2: The uncertainty in $V_{sys} > 15$ km s^{-1}.
- Class 3: The galaxy satisfies both the Class 1 and 2 requirements.

9 MODEL GENERATION

The final step in WKAPP is generating the 'average' models. This is done via the *Wallaby_ModelGeneration.py* script. As noted above, only galaxies that have an entry of "1" in the "USABLE_MODEL" of the user created fit success catalogue CSV file will have average models generated. This script averages the 3DBAROLO and FAT fits for each successful galaxy into a final kinematic model. This model, along with a cube realization, a diagnostic plot, and the 3DBAROLO and FAT results are saved to a file/folder structure as given in the particular field definition file.

REFERENCES

Astropy Collaboration et al., 2013, A&A, 558, A33

Astropy Collaboration et al., 2018, AJ, 156, 123

Deg N., Deg N., Deg N., Deg N., Deg N., 2022, Publ. Astron. Soc. Australia, 468, 731

Di Teodoro E. M., Fraternali F., 2015, MNRAS, 451, 3021

Harris C. R., et al., 2020, Nature, 585, 357

Hunter J. D., 2007, Computing in Science & Engineering, 9, 90

Józsa G. I. G., Kenn F., Klein U., Oosterloo T. A., 2007, A&A, 468, 731

Kamphuis P., Józsa G. I. G., Oh S. . H., Spekkens K., Urbancic N., Serra P., Koribalski B. S., Dettmar R. J., 2015, MNRAS, 452, 3139

Reback J., et al., 2020, pandas-dev/pandas: Pandas 1.0.3, doi:10.5281/zenodo.3715232, `https://doi.org/10.5281/zenodo.3715232`

Virtanen P., et al., 2020, Nature Methods, 17, 261

Westmeier T., Westmeier T., Westmeier T., Westmeier T., Westmeier T., 2022, Publ. Astron. Soc. Australia, 468, 731