

---

# **Robbie**

***Release 2.2.0***

**ADACS**

**Oct 18, 2022**



## CONTENTS:

<b>1</b>	<b>Description</b>	<b>1</b>
1.1	Dependencies . . . . .	1
1.2	Credit . . . . .	2
1.3	Links . . . . .	2
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Nextflow . . . . .	3
2.2	Robbie scripts . . . . .	3
2.3	Docker containers . . . . .	3
2.4	Alternative when not using containers . . . . .	4
<b>3</b>	<b>Setting up robbie on a new hpc or cluster</b>	<b>5</b>
<b>4</b>	<b>Quickstart</b>	<b>7</b>
4.1	images.txt . . . . .	7
4.2	robbie.nf . . . . .	7
4.3	nextflow.config . . . . .	7
4.4	-profile . . . . .	8
<b>5</b>	<b>Visualisation</b>	<b>9</b>
5.1	Running locally with Docker . . . . .	9
5.2	Running on a cluster with Singularity . . . . .	9
5.3	Visualising transients . . . . .	10
5.4	Visualising transient Epoch's . . . . .	10
5.5	Transient candidate selection . . . . .	10
<b>6</b>	<b>Workflow</b>	<b>11</b>



## DESCRIPTION

Robbie automates the process of cataloguing sources, finding variables, and identifying transients.

The workflow was described initially in [Hancock et al. 2018](#) however the current workflow is shown below:

- Preprocessing:
  - Convolve all images to a common psf (optional)
  - Create background and noise maps (if they are not found)
  - Correct astrometry using fitswarp (optional)
- Variable/persistent source detection:
  - Stack the warped images to form a mean image
  - Source find on the mean image to make a reference catalogue
  - Priorized fit this catalogue into each of the individual images
  - Join the epoch catalogues to make a persistent source catalogue
  - Calculate variability stats and generate a light curve for each source
- Transient candidate identification:
  - Use the persistent source to mask known sources from the individual epochs
  - Source find on the masked images to find transients
  - Concatenate transients into a single catalogue, identifying the epoch of each detection

See [workflow](#) for a diagram of how Robbie works.

## 1.1 Dependencies

Robbie relies on the following software:

- [AegeanTools](#)
- [fits\\_warp](#)
- [Stils/TOPCAT](#)
- [Nextflow](#)
- [SWarp](#)

All dependencies except for Nextflow will be installed in the docker image.

## 1.2 Credit

If you use Robbie as part of your work, please cite [Hancock et al. 2018](#), and link to this repository.

This project relies in part on software development provided by the [ADACS](#) merit allocation program for 2022A.

## 1.3 Links

You can obtain a docker image with the Robbie dependencies installed at [DockerHub](#)

## INSTALLATION

Robbie relies on 3 core technologies to run:

- nextflow
- python
- docker or singularity containers (optional)

### 2.1 Nextflow

Nextflow can be installed in a few ways:

- By following instructions at [nextflow.io](https://nextflow.io)
  - `wget -qO- https://get.nextflow.io | bash`
- By using a packaged manager such as [Conda](#):
  - `conda install -c bioconda nextflow`
- If you are using Pawsey you can just run:
  - `module load nextflow`

### 2.2 Robbie scripts

The Robbie scripts are all bundled as a python package which can be downloaded and installed via github.

Using pip: `pip install https://github.com/ADACS-Australia/Robbie.git`

Using Conda: `conda install git pip` (and then run the above)

### 2.3 Docker containers

#### 2.3.1 Robbie base container

The best way to use Robbie is via a docker container that has all the software dependencies installed. Ensure docker is running, then build the container using:

```
docker build -t paulhancock/robbie-next -f docker/Dockerfile .
```

or by pulling the latest build from [DockerHub](#) via

```
docker pull paulhancock/robbie-next
```

Please see *Quickstart* on how to run Robbie once the setup is complete.

### 2.3.2 Robbie visualisation container

To construct the Docker image for visualisation of the Robbie results, run the following:

```
./build_docker.sh
```

within the `robbie_viewer_server` directory. Please see *Visualisation* on how to visualise the results of Robbie using the aforementioned Docker image.

## 2.4 Alternative when not using containers

If cannot, or chose not to, use containers then you will need to ensure that the Robbie scripts are available on your local machine.

Python dependencies can be installed via:

```
pip install -r requirements.txt
```

[Stils/TOPCAT](#) needs to be downloaded and available on your machine. Once the `.jar` file has been copied to your machine you need to edit `nextflow.config` and set the following parameter so that Robbie knows how to invoke stils:

```
params.stilts = java -jar /path/to/topcat-full.jar -stilts
```

Alternatively the above can be set from the command line using the `--stilts` argument.

[SWarp](#) can be installed via source or rpm as per the website description. After installing you should be able to invoke SWarp using the command line `SWarp`. If this is not the case, and you need a different command to use SWarp then edit the following parameter so that Robbie knows how to invoke SWarp:

```
params.swarp = <swarp command>
```

Alternatively the above can be set from the command line using the `--swarp` argument.



## SETTING UP ROBBIE ON A NEW HPC OR CLUSTER

The file `nextflow.config` contains all the information about how to run on different environments which are referred to as executors. The default executor is `local` which means to run on the current machine, which is what you would do on your desktop/laptop. In order to work on an hpc cluster you'll need to set the `executor` to be `slurm` or `pbs` or whatever it is your hpc uses to schedule jobs. The Pawsey hpc clusters all use `slurm`.

Nextflow allows you to set 'profiles' within the `nextflow.config` which will let a user easily choose a whole bunch of settings without needing to update the configuration file. These profiles can be selected by using `-profile <name>` from the command line when invoking your `Robbie.nf` script.

Currently, Robbie has been set up with profiles for Magnus, Zeus, and Garrawarla. The Magnus and Zeus profiles need to be selected manually using `-profile Mangus` or `-profile Zeus`. The Garrawarla profile doesn't need to be specified as it will automatically be set if Robbie sees that you are running on a machine with a hostname that starts with `garrawarla` (eg, one of the login nodes).

If you want to use Robbie on a different cluster or hpc, you'll need to create a profile in the `nextflow.config` file. The best way to do this is to copy/paste from a different profile, and then refer to the [Nextflow documentation](#) to understand what each parameter is doing, and what needs to be changed for your system.



## QUICKSTART

Robbie now uses Nextflow to manage the workflow and can be run on a local system or a supercomputing cluster. You can use a container via singularity, docker, or the host's software. The current development cycle tests Robbie using singularity on an HPC with the Slurm executor - other setups *should* work but haven't been extensively tested.

### 4.1 images.txt

Before running Robbie, you will need to create a text file that contains the paths to each image to be processed. By default, this text file is called `images.txt` and is in your current directory. You can also give Robbie a custom file name and location using `--image_file <path>/<file_name>.txt`. For example, if there is a directory named "images" containing the `.fits` files:

```
ls images/* > images.txt
```

will populate `images.txt` with the image paths relative to the parent directory.

### 4.2 robbie.nf

This file describes the workflow and can be inspected but shouldn't be edited directly. For an explanation of the command line arguments, run

```
robbie.nf --help
```

### 4.3 nextflow.config

This file is the configuration setup and contains all the command line arguments' default values. You can change these defaults by copying the `nextflow.config` and editing the relevant params. You can then use your custom config via:

```
nextflow -C my.config run robbie.nf
```

The `-C my.config` directs Nextflow to use *only* the configuration described in `my.config`. If you use `-c`, you can supply a config file that is supplementary to `nextflow.config`, for example `my.config`. If both config files have the same parameters defined, `my.config` takes precedence otherwise the parameters are merged.

## 4.4 -profile

If you're running Robbie on your local machine, you should use the `-profile local` option to use the Robbie docker image. For example:

```
nextflow -C my.config run robbie.nf -profile local
```

If you're running Robbie on a supercomputing cluster (HPC), you should use the relevant cluster profile (`-profile zeus` or `-profile magnus`) to assure you're using the cluster's job queue (such as Slurm). If there isn't a profile for your cluster (check in `nextflow.config`), you may have to make your own.

## VISUALISATION

### 5.1 Running locally with Docker

To start the Docker container containing the Bokeh server, run the following script in the main Nextflow directory:

```
./run_robbie_viewer.sh
```

This will run the viewer using the images output from Robbie within the default `results` directory. If your output directory is different to the default, you can add either the relative or absolute path as an optional argument:

```
./run_robbie_viewer.sh -p path_to_dir
```

When plotting large images, it is recommended to also specify an RA and DEC position, as well as a size in coordinate units, to cutout a portion of the image for plotting. For example, if we want to plot an image with centre position of RA 335°, DEC -15° and size of 5°:

```
./run_robbie_viewer.sh -p path_to_dir -c 335,-15,5
```

### 5.2 Running on a cluster with Singularity

The Robbie Viewer is available on Pawsey as a part of the SHPC (Singularity Recipe HPC). To install it, we will load the SHPC module, install the viewer as a module and then load it:

```
module load shpc/0.0.53
shpc install cjproud/robbie_viewer
module load cjproud/robbie_viewer/latest/module
```

Now, the viewer will be available on our path and we can run it as normal:

```
./run_robbie_viewer.sh -p path_to_dir -c RA,DEC,PAD
```

## 5.3 Visualising transients

Visualising different transient candidates can be done in multiple ways. For example, the transient candidate can be selected using the table, sky plot or variables plot as shown below:

## 5.4 Visualising transient Epoch's

Cycling through Epoch's for each transient candidate is just as easy, for example you can use either the Epoch slider or select each Epoch in the Peak Flux vs. Epoch graph:

## 5.5 Transient candidate selection

Bokeh has multiple ways to interact with the data shown in the plots and table. To select multiple transient candidates, one option is to hold `shift` and click on the table entries. Once we zoom out, we can see all the selected transients on each plot:

The Box select tool can also be used to select transient candidates. After drawing the bounding box for selection, the transient candidates are highlighted in the other plots as well as the table below:

## WORKFLOW

The workflow was described initially in [Hancock et al. 2018](#) however the current workflow is shown below:

- Preprocessing:
  - Convolve all images to a common psf (optional)
  - Create background and noise maps (if they are not found)
  - Correct astrometry using fitswarp (optional)
- Variable/persistent source detection:
  - Stack the warped images to form a mean image
  - Source find on the mean image to make a reference catalogue
  - Priorized fit this catalogue into each of the individual images
  - Join the epoch catalogues to make a persistent source catalogue
  - Calculate variability stats and generate a light curve for each source
- Transient candidate identification:
  - Use the persistent source to mask known sources from the individual epochs
  - Source find on the masked images to find transients
  - Concatenate transients into a single catalogue, identifying the epoch of each detection

Items with a (?) are optional processing steps that can be turned on/off. Items in a cylinder are the final outputs of the processing workflow.