

# Making Where Available to the Community

Geir Arne Hjelle, Ann-Silje Kirkvik, Michael Dähnn, Ingrid Fausk

**Abstract** The Norwegian Mapping Authority is an associated Analysis Center within the IVS and is currently preparing to contribute VLBI analyses to the IVS with its new *Where* software. The software is being made available to the geodetic community under an open source MIT license. You can download *Where* from GitHub and try it for yourself. *Where* is written in Python and comes with a graphical tool called *There* that can be used to analyze results from *Where* and edit sessions. Furthermore, a geodetic Python library called *Midgard* is made available. This library contains components of *Where* that are useful in a wider setting. If you are doing any geodetic data analysis in Python, *Midgard* might be useful to you.

**Keywords** VLBI, Software, Python, Open source

## 1 Introduction

The Norwegian Mapping Authority (NMA) has been developing a new software for geodetic analysis over the last years [2, 3]. That software is called *Where* and is currently being used in NMA's bid for becoming an operational Analysis Center within the International VLBI Service for Geodesy and Astrometry (IVS) [4]. In addition NMA has made *Where* available to the community as open source software [1] under the MIT license.

The *Where* software is currently able to analyze single VLBI sessions, where ambiguities have already been resolved (level 4 databases). There are plans to

---

Norwegian Mapping Authority

extend the capabilities of the software going forward, including preliminary support for SLR and GNSS analyses.

The background for why NMA has developed a new software and which models are available, as well as early benchmarks and results, are covered in [3] and [4]. This note will focus more on how you can obtain and use the software yourself.

## 2 Overview of Where

The *Where* software is mainly written in Python<sup>1</sup>. It is cross platform and can run on Linux, macOS, and Windows. *Where* is available for download at

`kartverket.github.io/where/`

*Where* can do different kinds of analyses. Each kind of analysis is defined as a pipeline consisting of separate steps or stages. Currently, only the VLBI pipeline is publicly available, but work on other pipelines based on SLR and GNSS is underway.

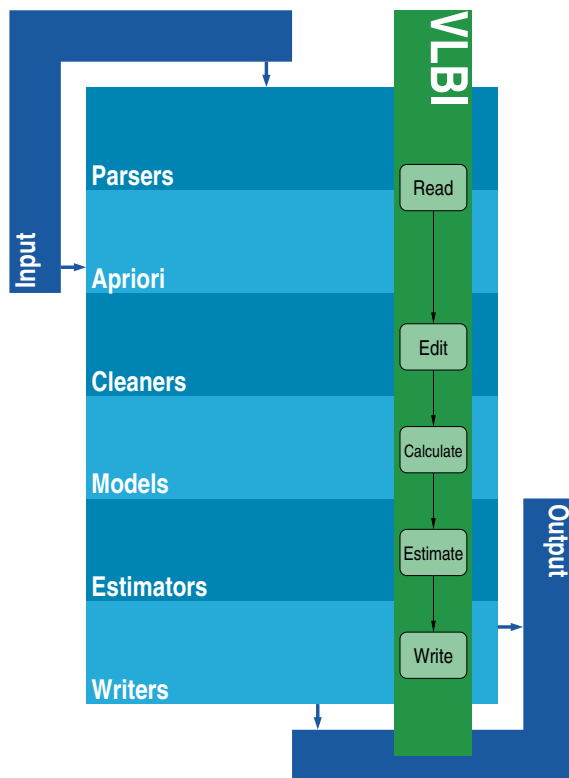
Figure 1 illustrates the VLBI pipeline in the green, vertical box that contains arrows. Pipelines are based on a few simple principles:

- Stages in a pipeline are performed sequentially.
- The output from one stage is input to the next stage.
- The state of an analysis can be inspected at each stage.

While the pipelines are tailored to one particular kind of analysis, each stage relies on components that are common to all pipelines. This is illustrated as horizontal, blue boxes in Figure 1.

---

<sup>1</sup> [www.python.org](http://www.python.org)



**Fig. 1** The VLBI pipeline. The stages *Read*, *Edit*, *Calculate*, *Estimate*, and *Write* are performed sequentially to perform a VLBI analysis. Other pipelines can be defined that may reuse the common Where components: *Parsers*, *Apriori*, *Cleaners*, *Models*, *Estimators*, and *Writers*.

This architecture makes it easy to reuse code across techniques. For instance, different pipelines can use the same ITRF component for a priori station positions, the same ocean loading model for station displacements, or the same Kalman filter estimator.

Where comes bundled with a graphical tool for looking at and analyzing results from an analysis. This tool is called There. Figure 2 shows a screenshot of There in action. More information about There will be given in Section 5 below.

In an effort to make the functionality of Where as available as possible, components that can be reused outside Where pipelines are being separated into its own package called Midgard. While Midgard has its own web page<sup>2</sup>, it is most easily installed through the

<sup>2</sup> [kartverket.github.io/midgard/](http://kartverket.github.io/midgard/)



**Fig. 2** The There graphical analysis tool. The plot shows post-fit residuals (after the *Estimate* stage) for the XE session on June 7, 2018. The colors indicate the baseline. That is, all residuals for the same baseline have the same color.

Python Package Index (PyPI)<sup>3</sup> using the standard Pip tool:

```
$ pip install midgard
Section 6 contains more information.
```

### 3 Try It Yourself

Where is available through the GitHub platform. GitHub<sup>4</sup> is a place for sharing code and working together on software development. In addition to downloading Where and using it for your own analysis, you may also contribute to the further development of the software. Get in touch<sup>5</sup> if you have any questions.

To start using Where you first need to install Python. We recommend the Anaconda Python distribution<sup>6</sup>, as it bundles Python together with a lot of useful data science tools. Once you have Python on your system follow the instructions at the Where web page<sup>7</sup> to install the program.

Once the program is installed you can run an analysis. This is done with a command of the form

```
$ where 2018 6 7 --vlbi --session=XE
```

The date, pipeline, and session must be specified. When executed, this will perform a VLBI analysis with the default settings (see Section 4 below for more infor-

<sup>3</sup> [pypi.org/project/midgard/](http://pypi.org/project/midgard/)

<sup>4</sup> [github.com](http://github.com)

<sup>5</sup> [kartverket.github.io/where/pages/contact.html](http://kartverket.github.io/where/pages/contact.html)

<sup>6</sup> [www.anaconda.com/download](http://www.anaconda.com/download)

<sup>7</sup> [kartverket.github.io/where/installing.html](http://kartverket.github.io/where/installing.html)

mation). The program will automatically download all auxiliary files needed to do the analysis.

All information about an analysis, including the configuration, a list of file dependencies, and intermediate and final results, are stored in a separate directory. While running the analysis, Where will by default print information about what it is doing to the console. The name of the analysis directory is printed at the top of this log, and the log is also stored to a file inside the analysis directory.

## 4 Changing the Analysis Parameters

Most aspects of a Where analysis are configurable. The default configuration is defined in configuration files<sup>8</sup>. You can change the default configuration for all analyses by creating a file called `where_local.conf` inside a directory called `.where` that resides in your home directory.

You can look at the configuration of an analysis by using the `where_setup` tool:

```
$ where_setup 2018 6 7 -v --session=XE
=====
VLBI XE 2018-06-07

[vlbi]
atmospheric_tides           = cm
atmospheric_tides_cmc      = False
ephemerides                 = de430
...
```

Individual parameters can be changed by specifying them on the command line when running either `where` or `where_setup`:

```
$ where 2018 6 7 -v --session=XE \
  --ephemerides=de421
```

You can also use the `-E` (`--edit`) option to open and change the configuration in an editor:

```
$ where 2018 6 7 -v --session=XE -E
```

While NMA so far has used Where to analyze 24-hour R1 and R4 sessions, the software can analyze other kinds of VLBI sessions as well by changing the configuration appropriately. To easily work with different kinds of sessions, Where supports configuration profiles. You can for example analyze the XU intensive session on June 7, 2018 as follows:

```
$ where 2018 6 7 -v --session=XU \
  --profile=intensives
```

A graphical utility for setting up and changing the configuration is in preparation. It will be integrated into There later.

## 5 Running Analyses

To run a series of analyses you can use the `where_runner` tool. It will use master files<sup>9</sup> to call Where for all sessions in a specified time interval. For example, to analyze all sessions in January 2018 use the following:

```
$ where_runner 2018 1 1 2018 1 31 --vlbi
```

If you try to rerun an analysis you have already done, Where will by default do so in the same directory as before. Where is also lazy software, so it will not even do the analysis unless either the configuration or at least one of the input files have changed.

When comparing different models or settings you will be interested in running the same analysis twice and keeping the results. The way to do this is to label each of your analyses using the `--id` option:

```
$ where 2018 6 7 -v --session=XE \
  --id=vtrf \
  --reference_frames=vtrf
```

The results from an analysis can be inspected using the There graphical tool. Starting There is usually done only by specifying that you want to look at VLBI analyses (either `-v` or `--vlbi`):

```
$ there -v
```

This opens the There application. There is organized using tabs. Along the top you will see different tabs including *Timeseries* and *Session*.

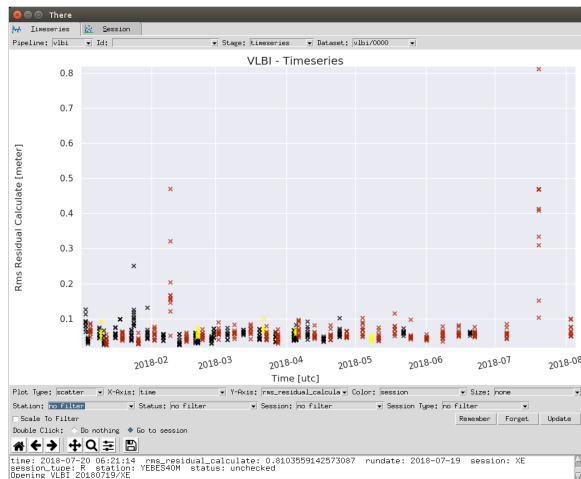
The *Timeseries* is used to get an overview over all sessions that were analyzed. It can display many different summary parameters for each session, including residuals (see Figure 3), size of session, and estimated parameters. Clicking on a point gives more information in the status bar at the bottom of the window.

You can also choose *Go to session* and then double click on a point to investigate that session more closely. This will open the *Session* tab as in Figure 4.

The dropdown menus can be used to choose everything from plot type, to which variables to show, to

<sup>8</sup> [github.com/kartverket/where/tree/master/config](https://github.com/kartverket/where/tree/master/config)

<sup>9</sup> [ivsc.gsfc.nasa.gov/program/control\\_files.html](https://ivsc.gsfc.nasa.gov/program/control_files.html)



**Fig. 3** An overview over analyzed sessions. The *Timeseries* tab shows aggregated information about many sessions. In this example you can see that a few sessions (including one in July 2018) have much higher residuals than is normal. The colors indicate session name.

filters that can be applied to the data. Each point represents one observation in the session. Clicking on a point shows information about that observation in the status bar.

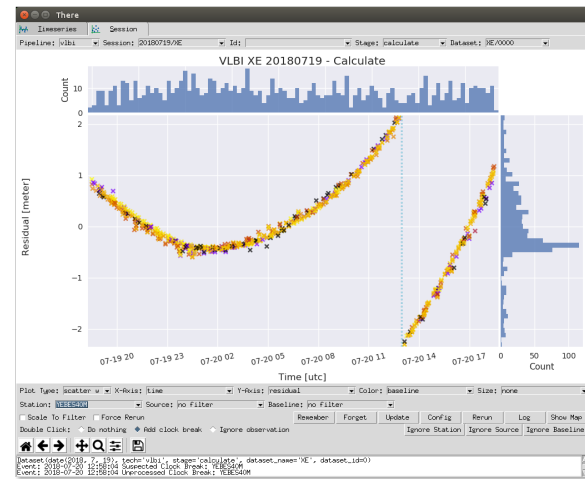
You can edit a session from within *There*. As an example, consider Figure 4. There is a clock break at around 13:00 UTC that should be edited out. *Where* has preliminary support for detecting clock breaks by itself. To mark the clock break and rerun the analysis, you do the following:

- Choose *Add clock break*.
- Filter on the station in question, in this example YEBES40M.
- Double click in the plot where the clock break should be added. You might need to zoom in to do this precisely.
- Click the *Rerun* button.

After rerunning the analysis, *There* will update with the improved residuals.

Other editing operations, like ignoring single observations or all observations for stations, baselines, or sources are available directly in the *There* interface. However, more complicated editing can be done by opening the configuration file. This is most easily done by clicking the *Config* button.

One other option for getting insight about the results of an analysis is to plot the results on a map.



**Fig. 4** Looking at the results from one analysis. The *Session* tab shows information about one particular session. The current plot shows *pre* residuals for the XE session on July 19, 2018. The results have been filtered to only show the YEBES40M station. A clock break is clearly visible in the residuals at around 13:00 UTC and is marked with a blue dashed line.

Clicking the *Show Map* button in *There* will open an interactive map in your browser. The map shows all stations and baselines in the session. Colors are used to show the size of the residuals (green is smaller), while the size of circles and width of lines represent the number of observations for a given station or baseline.

The map illustrates the geometry of the solution. As it is interactive you can also zoom and pan and, for instance, look at local conditions at a site.

## 6 Reuse in Your Own Programs

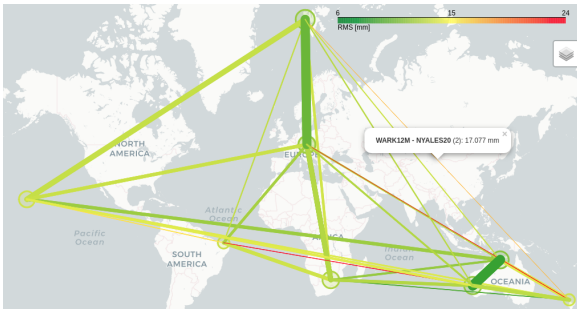
If you are using Python for programming geodetic applications or analyzing geodetic data, *Midgard* might also be interesting to you directly.

*Midgard* is a library containing more general purpose geodetic functionality. *Where* uses *Midgard* extensively, and the plan is to move all parts of *Where* that can be reused to *Midgard*. Moving common components to *Midgard* is an ongoing process. Check the web page<sup>10</sup> for the current status.

You can use *Midgard* also if you are not using *Where*. In that case, first install *Midgard* using *pip*:

```
$ pip install midgard
```

<sup>10</sup> [kartverket.github.io/midgard/](http://kartverket.github.io/midgard/)



**Fig. 5** Interactive map of one analysis. The colors indicate the size of the residuals on given stations and baselines. The size of a circle and the width of a line show how many observations were used for a station or baseline, respectively

You can then use Midgard in your own scripts. For instance, to read a Rinex file<sup>11</sup> you can do something such as:

```
>>> from midgard import parsers
>>> path = "ALGO00CAN_R_20121601000.rnx"
>>> parsers.parse_file("rinex3_obs", path)
```

More information about Midgard and examples of how to use it is available online.

## 7 Conclusions

At the time of writing, the latest available version of Where is v0.12.1. However, new development on Where is ongoing, and we plan to release version 1.0.0 within 2018. See the web page<sup>12</sup> for the latest updates.

Please get in touch<sup>13</sup> if you find Where useful or have any questions.

## References

1. Hjelle, G. A., et al., Where: A scientific python package for geodetic analysis, in preparation.
2. Kirkvik, A.-S., Norwegian mapping authority analysis center biennial report 2015–2016, in Baver, K. D., Behrend, D., Armstrong, K. L. (eds.), *International VLBI Service for Geodesy and Astrometry 2015+2016 Biennial Report*, pages 245–248, NASA/TP-2017-219021, 2017.

<sup>11</sup> [en.wikipedia.org/wiki/RINEX](http://en.wikipedia.org/wiki/RINEX)

<sup>12</sup> [kartverket.github.io/where/](http://kartverket.github.io/where/)

<sup>13</sup> [kartverket.github.io/where/pages/contact.html](http://kartverket.github.io/where/pages/contact.html)

3. Kirkvik, A.-S., et al., Where - a new software for geodetic analysis, in Haas, R., Elgered, G. (eds.), *Proceedings of the 23rd European VLBI Group for Geodesy and Astrometry Working Meeting*, 2017.
4. Kirkvik, A.-S., et al., NMA analysis center – progress report, in Armstrong, K. L., Behrend, D. Baver, K. D., IVS 2018 General Meeting Proceedings, 2018, this volume.