OSNAP-CLASS MOORING PROCESSING TOOLBOX

Version 1.2

Loïc Houpert, Sam Jones, and Lewis Drysdale
[Email address]

Contents

Setup and version control

1. Git

1.1. A word on git

Users of the osnap mooring processing toolbox should request a new branch from the latest post cruise updated master from https://github.com/LewisDrysdale/m_moorproc_toolbox. This should be the last working version used and updated since the last mooring cruise and the branch should be renamed to reflect the upcoming cruise. A copy of this branch should be physically taken on the ship (i.e. *osnap/exec/$cruise/*). The data (archived from previous cruise) is not kept on git and effort should be made to make sure that the most recent version (i.e. from SAMS servers) is copied prior to the cruise/processing. Once the cruise is complete the branch should be compared and merged back towards the master and should be well commented to capture significant changes made to the code.

1.2. Git setup

If you want to set up a local copy of the project git repository to contribute to the project you should do the following steps:

1. First make sure that you have Git installed! See more details here: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git. If you are using windows you may also want to install a nice terminal (e.g. https://mobaxterm.mobatek.net)

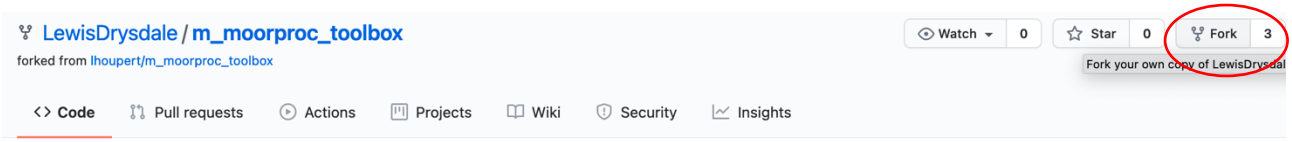
2. Open a terminal, move in your working directory (e.g. ~/Work) and then **clone the project** repository with the command:

```
git clone https://github.com/LewisDrysdale/m_moorproc_toolbox.git
```

3. Enter the git repository:

```
cd m_moorproc_toolbox
```

4. **Fork the repository** https://github.com/LewisDrysdale/m_moorproc_toolbox.git to your personal account. To do that just click the Fork button on the main repo page.



5. **Add your fork as a remote.** This remote will be named after your github username for clarity. To do that, use the command: `"git remote add your-github-username fork-url"`

For example, for a user named lhoupert, run the command:
`git remote add lhoupert https://github.com/lhoupert/m_moorproc_toolbox.git`

1.3. Check the local and remote branches

To list the local and remote branches, use the the command "`git branch –av` " . This command returns for example:

```
git branch –av
* master                          80e31f1 add microcat calib coefficients files
  remotes/origin/HEAD             -> origin/master
  remotes/origin/dy120            bf2a80b Revert "Merge pull request #3 from
LewisDrysdale/dy120"
  remotes/origin/master           80e31f1 add microcat calib coefficients files
  remotes/origin/postdy120        472a21c more upddates and tidying
```

We can see one local branch (master) and 4 remotes branches on the "origin" repository (corresponding to the repository cloned). We cannot see any branches associated with the fork directory because the local repository has not download information about it yet. To do that we need to run:

`git fetch lhoupert`

Now the command "`git branch –av`" return an updated list of branches:

```
git branch –av
* master                        80e31f1 add microcat calib coefficients files
  remotes/lhoupert/dy120        bf2a80b Revert "Merge pull request #3 from
LewisDrysdale/dy120"
  remotes/lhoupert/legacy       fe06d0c .
  remotes/lhoupert/master       5837363 remove unnecessary empty directory
  remotes/lhoupert/postdy120 472a21c more upddates and tidying
  remotes/origin/HEAD           -> origin/master
  remotes/origin/dy120          bf2a80b Revert "Merge pull request #3 from
LewisDrysdale/dy120"
  remotes/origin/master         80e31f1 add microcat calib coefficients files
  remotes/origin/postdy120      472a21c more upddates and tidying
```

If you only have a local *master* branch (as in the example above) and you need to create a local copy of the other remote branches ("legacy", "postdy120", etc…). You can do that with the one-time command:

```
git checkout –b postdy120 origin/postdy120
git checkout –b legacy    origin/legacy
```

1.4. Making changes

Before any changes, a branch should be created. **Best practices are to never commit to master branch**. It is better to work on a specific branch and then merge the branch onto the "main" or " master" branch. Your terminal should be set up to display the current active branch.

1) Update master. Before you make any changes, first checkout master (or another branch if you are working on another branch) and pull in the latest changes. In the example below we want to work on the branch postdy120 so we do:

```
git checkout postdy120
```

```
git pull
```

git pull is here a shortcut for *git pull origin postdy120*

2) Create a branch. Make a branch name that is short, descriptive, and unique. Some examples of good branch names are `fix-install`, `docs-cleanup`, and `add-travis-ci`. Some examples of bad branch names are `feature`, `fix`, and `patch`. To create the branch, do :

```
git checkout -b branch-name
```

(e.g.: `git checkout -b add-IBmoors`)

3) Make your changes and commit them. Keep commits atomic <-> each commit should represent a single unit of change. Also write helpful commit messages, so that someone can understand what the commit does just from reading the message without having to read the diff.

a. To see what file have been changes locally run the command `"git status"` in a terminal

b. To stage your changes, add the different files with the command `"git add filename [filename2 …]"`

(e.g. `"git add OSNAP_mooring_processing.docx"`)

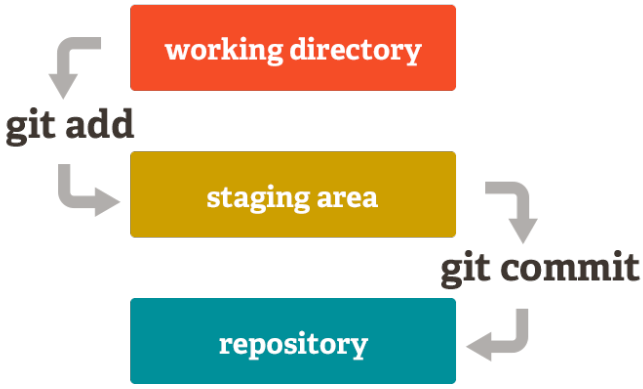


Figure 1: From https://git-scm.com/about/staging-area

c. To commit the changes to your repository run the command: `git commit -m "message explaining what the commit does"` (e.g. `git commit -m "add Git section to doc" `)

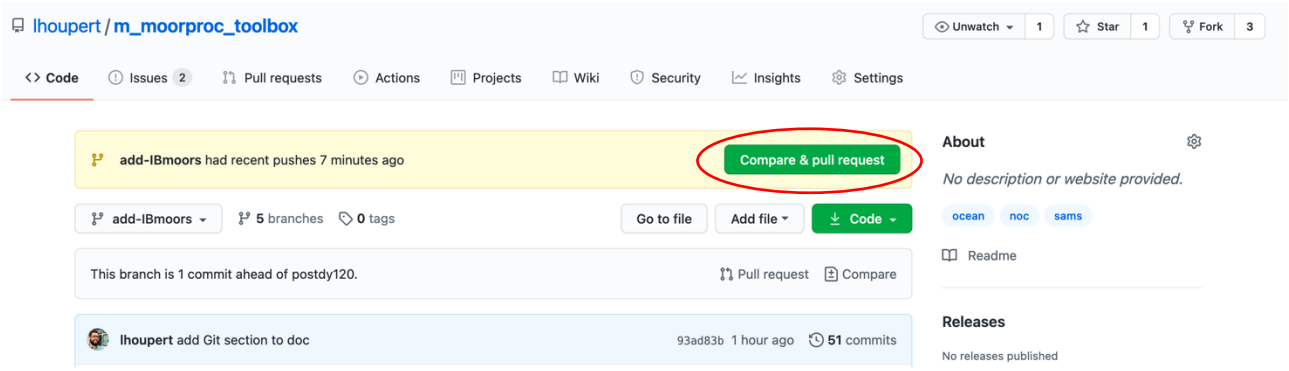
4) Push up your changes on your remote repository. Once the changes are committed, they are only applied to the local repository. To propagate the changes of the remote repository, the user needs to "push" the changes. It can easily be done on your fork repository by running the command: `git push your-github-username branch-name` (e.g. `"git push lhoupert add-IBmoors"`)

1.5. Push changes on the original remote repository

As owner of the "forked" github repository, it is easy to push the changes to it. However, because you are collaborating with other people, you will certainly want to push your changes back to the original repository so other users following changes on this repository can be aware of your modifications.

Case 1: Through a pull request (best practices - available for anyone)

Make a pull request. Once the changes are pushed on your forked repository, you can then go to your fork on GitHub. You should see a button to create a pull request from your branch. It will look something like this:

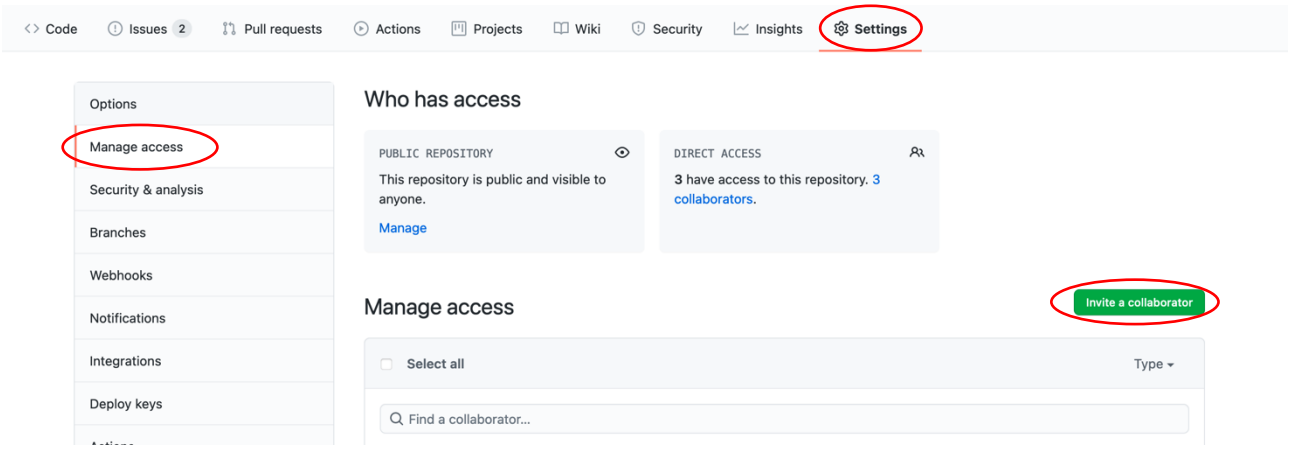


Then by selecting "comparing & pull request" , you will be presented a new page asking you more information about your request (name of the repository you want to merged your branch into, title of the request, description, etc…). You can find more information about the pull request workflow here: https://www.asmeurer.com/git-workflow/

Case 2: By pushing changes directly on the origin repository without pull request (only for authorized users).

When we started working on the postdy120 branch in November 2020, three people were working on the same files and therefore opening a "pull request" was not the most efficient way for everybody to have access to the same files in the last version of the file in real time.

Therefore, we decided to work on the same "original" repository. To do that the owner of the original repository needed first to grant "pushing rights" on his repository. This can be done by opening the settings of the repository, selecting "manage access" and "invite a collaborator":



Once the user is added as a collaborator, he/she can push changes on the "main" repository using the command: `"git push your-github-username branch-name"`

For example: `"git push origin add-IBmoors"`

1.6. More on Git

- Tutorial about Gir-workflow explained: https://www.asmeurer.com/git-workflow/
- More information on git commands: Run `"git help"` or `"git help -g"` in a `terminal` or have a look to the git website (e.g. https://git-scm.com/docs/giteveryday)
- Some notes on file manipulation with Git: https://gist.github.com/lhoupert/01daa22e0b2a4171c20a291df14e661b
- Git Cheat Sheet: https://training.github.com/downloads/github-git-cheat-sheet.pdf

2. Setup and run locally the toolbox

2.1. Get the most recent version of the toolbox

If you don't have a version of the git repository, get the last version of the code at https://github.com/LewisDrysdale/m_moorproc_toolbox. Make sure you select the correct branch (e.g. *postdy120*, *master, etc.*.), clone it or download the zip.

If you have already a git version of the repository set up following *Section 1. Git*, go in your git folder then run: `git pull origin master` (or another branch-name if you working on a specific branch)

This will update the local version of the repository with the latest version of the remote directory.

2.2. Create a branch for your development

Before modifying the code, you should create a new branch. More information are given in the

previous section, see 1.4)

2.3. Set up the toolbox in your matlab path

Once the mooring processing toolbox is set up, copy the file in Documents/exampleofstartupfile.m in you matlab working directory and rename it (e.g. startup_moor_proc.m).

Open the startup file and edits the first two variables (pathosnap and pathgit) in order to set up the matlab paths properly.

- `pathosnap` should be pointing to the directory containing the OSNAP data . For example, `pathosnap='Users/locupe/Dropbox/Work/osnap'`. This directory should contain the main mooring repositories :

```
├── Documents
├── Figures
├── cruise_data
├── data
├── exec
├── matlab_diaries
└── users
```

- `pathgit should be pointing to the mooring processing toolbox (the github repository). For example: pathgit = '/Users/locupe/Dropbox/Work/Python/Repos_perso/m_moorproc_toolbox'`

Real-time and delayed-time processing of microCAT data

1. Introduction

→ calibration of the instruments is essential to obtain accurate measurements and minimize errors in the transport calculation
→ calibration data obtain for each moored CTD during intercalibration cast: Each moored CTD is attached to a shipboard CTD rosette frame and a deep cast is carried out prior to deployment and after the recovery of the moored instruments. During the cast, 12 bottle stops of 5 mins each are performed, from the bottom to the surface, in order to cover the different range of pressure of the moored instruments. The comparison with the 1Hz-bin averaged shipboard CTD data gives calibration coefficients for each sensor (T, C, P) on each moored instrument.

Outline of the processing stages:
- Initial processing setup: creation of a data processing metadata control file.
- Raw data downloading and archiving

- Conversion from instrument format to standard Rapid Data Base (RDB) format
- Trimming of data record, basic statistics and summary plots
- Calibrating and quality control of the moored CTD data:
- Gridding of the data: creation of a lowpass filtered, regular gridded data set

NOTE: the OSNAP data structure separates executable scripts (*osnap/exec/…*) from output files (*osnap/data/moor/…*). For clarity, file extensions for scripts are coloured red, input files are coloured blue, and output files are coloured green. The output files of some scripts may become the input of a later script. Hopefully this will make navigation of this document a little easier.

2. Set up directory structure

2.1. Processing scripts:

a) If it doesn't exist, create a directory *delayed_processing_script* in *osnap/exec/$cruise/* (from the previous cruise)

2.2. Shipboard calibration files

Cruise ctd files (pre and post deployment)

a) Create the directory for the cruise data: *osnap/cruise_data/$cruise*
b) Copy the cruise data from the archive directory of the cruise data (e.g. M:\Mar_Phys\Cruises\DY078_079) ; Look at the previous year cruise to copy the same type of ctd files (*_1hz.nc, *.ros and *_align_ctm.cnv) for each caldip cast. The number of the caldip cast can be found in the cruise report or in *osnap\data\moor\proc_calib\dy078\cal_dip (cast*info.dat).*

Microcat caldip data

a) create a *cast???info.dat* file in *osnap/data/moor/proc_calib/$cruise/cal_dip/* for each caldip that summarizes informations about each caldip CTD cast (location, time) and the serial numbers and the deployment periods of the lowered microcats. The deployment period number is used later (part 6b) to create a metadata file of the mooring deployed during the deployment period.
b) create the directories that will host the raw microcat data files for each CTD cast, *osnap/data/moor/raw/$cruise/microcat_cal_dip/cast???/* (where *???* is the cast number), and move in this directory the raw data file (.hex, .cnv, .xml, .xmlcon, etc..)

1) Microcat deployment files

a) creation of a data processing control file *$moorname$info.dat* for each mooring in the directory *osnap/data/moor/proc/$moorname$*. The *$moorname$info.dat* file contains metadata for mooring position (lon, lat, waterdepth and mean magnetic deviation during the deployment), deployment period, nominal depths and serial numbers of each instrument. *see previous cruise folder for an example.
b) copy the raw SBE37 files (with .cnv files) in the directory *osnap/data/moor/raw/$cruise/microcat/*. If the raw .cnv files are not named as

$serialnumber_data.cnv, rename them.

2) Start and Set-up Matlab

a) Open in a terminal and go in the root mooring processing directory (e.g.: *OSNAP_mooring_data_processing/osnap/*). Check that in the top of the *startup.m* file a "cd" command is sending you in the cruise directory you want to process (e.g. *cd exec/pe400*). Once matlab started under the *osnap/* directory the *startup.m* file should move you in the cruise directory you are interested in. In the cruise directory *osnap/exec/$cruise/*, another *startup.m* file should be present, generating the different path related to the cruise data.

3. **Processing of the moored microcat**

a) Stage 1: Edit (paths to the data, mooring name, year of the first measurement) and run the script *osnap/exec/$cruise/stage1/microcat/mc_call_2_$cruise.m* : convert the raw data to RDB formatted file .raw for an entire mooring. The processed data are stored in *osnap/data/moor/proc/$moor/microcat/*

b) Stage 2: Check that the deployment time and recovery time are accurate in the corresponding *$osnap/data/moor/proc/$moor/moor_info.dat* file, then edit and run *osnap/exec/$cruise/ stage2/microcat/microcat_raw2use_003_with_ODO.m* . This script generates *.use* files (launching and recovery period removed) in *osnap/data/moor/proc/$moor/microcat/* . The script also creates data overview sheet including basic statistics, and produces summary plots, including 2-day low-pass plots.

4. **Lowered microcat processing after the shipboard calibration casts**

The script *osnap/exec/$cruise/stage1/microcat/mc_call_caldip_v4b.m* loads: i) the raw microcat data located in *osnap/data/moor/raw/$cruise/microcat_cal_dip/$castnber/* , ii) the shipboard CTD data *(if new calibrated ctd files are available after the cruise, the *_raw.nc and *_psal.nc have to be replaced)* files for the *$castnber* (in *osnap/data/$cruise/)*, iii) the caldip metadata file *castnnninfo.dat* file located in *osnap/data/moor/proc_calib/$cruise/cal_dip/.*

The script writes to a directory *~/osnap/data/moor/proc_calib/cal_dip/$cruise/microcat/$castnber/* which is created manually. Plots are generated for all microcat data for one CTD cast with the shipboard CTD data. Note that the raw microcat files have to be named as *serialnumber_cal_dip_data.cnv.*

The script *osnap/exec/$cruise/stage1/microcat/mc_caldip_check_$cruise.m* provides a quick quantitative comparison of Microcat cal-dip data with the SBE911 data from the CTD. Data obtained at the deepest bottle stops are used. For each instrument differences of conductivity, temperature and pressure between the instrument and the CTD sensor were calculated. The mean and standard deviation of the differences for each instrument are then presented in a table in *~/osnap/data/moor/proc_calib/cal_dip/$cruise/microcat/$castnber$/microcat_check$castnber$.log*

5. Delayed mode processing of the microcat data.

Before starting the delayed mode processing of the microcat, make sure that you have the final calibrated CTD dataset.

5.1. Set-up metadata files

Manually add entries in two metadatabases: *cruise_id.xls* lists the cruise id and unique number identifier; *microcat_calib_cruise.csv* lists the mooring name, the associated deployment and recovery cruises number. The metadata files are under *osnap/data/moor/cal_coef/*

5.2. Calculation of accurate nominal depth of the mooring instruments, update of the metadata files

a) If there are large differences between the planned instrument depth and the actual depth for most of the deployment (due to trawling etc), it is possible to calculate this from the observed pressures. Generally, we have not needed to do this stage but the following might be useful. Note this just updates the 'nominal depth' which is used for metadata and file organisation, but will not impact the final data product. The gridding algorithm pulls info directly from the pressure record for example. Run the function *ctd_instrdpth2.m* for each mooring. This will calculate accurate instrument depth from pressure record and update the *info.dat* file located in *osnap/data/moor/proc/$mooring/*

b) Manual creation of a *osnap/data/moor/cal_coef/osnapXX_deploymentdepths.dat* file, which lists all the instruments on every mooring deployed within the deployment period XX (by using the info.dat file for each mooring in *osnap/data/moor/proc/$mooring/*)

5.3. Calculation of the pre- or post- deployment conductivity and temperature calibration coefficients

o) The raw microcat data files *.raw* have to be generated before starting this process (by doing the step III for example)

a) Before processing the mooring data, the calibration coefficient has to be calculated from the comparison between the calibrated shipboard CTD and the lowered mooring CTDs, this task is achieved by running the script *osnap/exec/$cruise/delayed_processing_script/mcat_final_calibration/caldip_coefficient_calculation.m*. Several user parameters have to be edit in the beginning of the *caldip_coefficient_calculation.m* script.

b) Setup the paths to the data files in insitu_cal_osnap2.m (by adding an entry for the cruise in the if/else part). Make sure that

c) Run *caldip_coefficient_calculation.m* for each caldip cast, the user can adjust the options for the figure at the beginning of the script. The script produces plots and a table of the calibration coefficients in *osnap/data/moor/proc_calib/$cruise/cal_dip/microcat/$castnber/*. The calibration coefficients are then manually entered into 3 CSV format database in *osnap/data/moor/cal_coef/* (*microcat_cond.csv*, *microcat_temp.csv* and *microcat_pres.csv*). This process is quite arduous and ultimately we would like to automate. Note the format used in the csv tables for previous 'before-after deployment' cruise pairings and emulate. If you freeze the column and row headings ('freeze panes' in Excel) you can keep them visible while manually entering. Triple check!

d) Set-up the scripts that apply the calibration corrections (in *osnap/exec/$cruise/process/ mcat_final_calibration/microcat_apply_cal_plus_rbr_idr_osnap.m*). Particularly:

1) the string formats for the reading of the .csv files has to be edited according to the number of columns in the csv files (variable *strformat* in the header of the script).

2) the reference CTD loaded for the QC plots. The cruises used for the plots are defined by the variable *ctd_ref_cruises*, then the user has to edit the end of the script (if/else section) by adding plotting instructions specific to each cruise.

e) Once the script is set-up, *microcat_apply_cal_plus_rbr_idr_osnap.m* applies the calibration coefficients for each time-series and bad data are removed. Constant offsets and conductivity pressure correction are applied if required.

- Average trend is
- General trend: to be applied when pre - or post cruise calibration is missing for individual instruments

The end result of this stage is the processed (but not gridded) microcat data in *osnap\data\moor\proc\$mooring\microcat.*

Note that, if the script discovers a version of the processed data already exists in this directory, it will copy it, append its creation date and save it as a backup. Therefore the 'to use' data file is the one without any date appended, e.g. *rtwb1_04_2017_001.microcat*

5.4. Gridding of the data: creation of a lowpass filtered, regular gridded data set.

Update the mooring file names and microcat order field in the script \osnap\exec\$cruise\stage3\gridding\grid_osnap_mcat_data.m

Check initialisation parameters listed in that script.

Run the script

It calls the worker script: *hydro_grid_osnap_linear_interp.m* in the same directory which does the following:

- -Load microcat data
- -Fill pressure gaps
- -Convert conductivity to salinity and despike
- -Apply low-pass temporal filter
- -Close big gaps in salinity
- -Interpolate vertically
- -Save and generate plots.

Data are saved as a mat file in *~\osnap\data\moor\proc\hydro_grid\.* The gridded .mat files have the variable naming conventions listed in Table 1

5.5. merging of the several years of deployment (and if necessary moorings)

First, we have to create 2 .dat files with instrument data from EB1 in one file. The same for WB1 and WB2 combined in the same file. Note, only instruments on WB2 that are at unique depth should be included. Note 2: in the .dat file, the 3rd columns correspond to the position of the mcat in the grid file created in 4) (can be found under *~/osnap/data/moor/proc/hydro_grid/*

E.g.: *~\osnap\exec\$cruise\stage3\gridding\MCAT\ rtwb_osnap_03_2016.dat*

Info can be found in:

~\osnap\data\moor\proc\rtwb1_03_2016\rtwb1_03_2016info.dat

Second, check and update paths and parameters under SET DIRECTORIES FOR MERGE and MERGE INITIALISATION (e,g, jg_end and lastyeardata) headers in *grid_osnap_mcat_data.m*

Last, update the scripts *merge_osnap_data_west.m* and *merge_osnap_data_east.m* by copy and pasting the last merge block section. In particular: create a new deployment by copying a previous one; change all the variable name (e.g. for osnap3, changes fileID2 -> fileID3, T2 -> T3, etc…); change the figure and titles (l326-331-336-341); and change concatenation.

Output files are saved in: *~\osnap\data\moor\proc\hydro_grid_merged\.*

Table 1. The gridded and merged .mat files have the following variable naming conventions using temperature data as an example:

Time grid	
Jd_grid	julian day for ungridded data – 2 hour timesteps. NOTE the JD naming convention – currently the wrong way round!
jd	julian day for gridded data – interpolated onto 1/2-day timesteps
JG	Julian day
Time grid	
T,C,S,P	Temporal interpolated data (1/12 day)
T,S,Pf	Low pass filtered (on 1/12 day)
T,S,Pfs	Temporal interpolated filtered (1/2 day)
T,S,PGfs	Temporal interpolated (1/2 day) filtered gridded (20 m)

Merge	
T,S,Pfs	Temporal interpolated filtered (1/2 day)
T,S,PGfs	Merged temporal interpolated (1/2 day) filtered gridded (20 m)
T,SGfs2	Merged de-spiked temporal interpolated (1/2 day) filtered gridded (20 m) horizontal interpolation

Processing of moored current meter and ADCP data

1. Introduction

The purpose of this document is to describe the data processing applied to all current meter records collected from the SAMS contribution of the UK-OSNAP project, including initial quality control of the data, data editing, corrections applied, and filtering.

The current meters used on the 4 SAMS-OSNAP moorings for the period 2014-2015 consisted of Nortek Aquadopps for 3 moorings (RTEB1, RTWB1 and RTWB2) and a RDI-ADCP 75kHz on RTADCP1. The Nortek Aquadopps had recording intervals of 20 minutes, while the ADCP has a recording interval of 1 hour.

Note: ~ refers to ~*OSNAP_mooring_data_processing*

Note 2: the OSNAP data structure separates executable scripts (osnap/exec/…) from output files (osnap/data/moor/…). For clarity, file extensions for scripts are coloured red, input files are coloured blue, and output files are coloured green. The output files of some scripts may become the input of a later script. Hopefully this will make navigation of this document a little easier.

2. Nortek current meter data processing

2.1. Stage0 – Data Download (done during the cruise $CRUISE)

Raw instrument data are downloaded from the instrument after the recovery of the mooring. Record keeping of the download is done on paper and for each instrument a download sheet is completed. After the download, the data are backup and transferred to the network drive on the cruise directory *$cruise/SAMS_moorings/nortek* . Instrument setup details and data are sorted in different directory by mooring name.

Then the data (.dat, .aqd and .hdr files) are copied on the processing computer in the directory e.g *~\osnap\data\moor\raw\$cruise\nortek*

2.2. Stage1 – Conversion to standard RDB format and basic statistics

You must start by running *~\OSNAP_mooring_data_processing\osnap\ startup_$cruise.m*

A text file containing the serial numbers of the Nortek on the mooring and the filenames containing the data has to be created as follows:

~\osnap\data\moor\raw\$cruise\nortek/rteb1_03_2016_filenames.txt.

The script *~\osnap\exec\$cruise\stage1\nor/process_nors_$cruise.m* performs stage1 and stage 2 processing on nortek data. It converts nortek data from raw to rodb format for an entire mooring. The user needs to modify the following information in the beginning of the script: cruise, the mooring name and operator.

process_nors_$cruise calls *nortek2rodb_01*, which saves the files downloaded by the instrument software (stage 0) to the RDB formatted file *.raw (*in */osnap/data/moor/proc/rteb1_03_2016/nor.)*. The script *nortek2rodb_01* also create data overview sheet including basic statistics:

~\osnap\data\moor\proc\rteb1_03_2016\nor\rteb1_03_2016_Nortek_stage1.log

2.3. Stage2 – Trimming of data, summary plots

The script *process_nors_$cruise* also performs stage 2 processing on nortek data by calling the script *nortek_raw2use_02*. This script uses the raw data file *mooring_serialnum.raw* generated by stage1 and the <*mooring*>*info.dat* file:

~\osnap\data\moor\proc\rteb1_03_2016

and produces a *.use* file. It removes the launching and recovery period and produce summary and diagnostic plots (figure 1, 2), including filtered data for display and diagnostic purposes (figure3).

2.4. Stage3 – Data editing, speed of sound correction, magnetic declination correction, data filtering

The script *proc_stage3_adcp_nortek in*:

~\OSNAP_mooring_data_processing\osnap\exec\$cruise\delayed_processing_script\adcp\

is used to perform stage3 processing on current meter data, it calls the function *cm_edit_NOCS.m*. Once the data trimming procedure and an initial data control are completed, the data can be corrected for speed of sound and magnetic declination. Magnetic deviation value has to be defined in: ~\osnap\data\moor\proc\rteb1_03_2016\ *$mooring_info.dat* file by adding a line like "*MagDeviation = -19.66"*.

All Nortek current meters UK-OSNAP project use a fixed speed of sound (1500m/s), but this is still worth checking in the .cap files! The corrected sound speed is obtained by using measured values of pressure and temperature data from the Nortek and a local/regional value of salinity. For the OSNAP 2014 cruise, a mean salinity was defined for each Nortek current meter using the moored microcat and the CTD cruise profiles (Annex 2), and local salinity values are varying between 34.95 and 35.4. The corrected sound speed varies between 1491 and 1499 cm/s (Annex 3). Corrected velocities are then obtained by multiplying the uncorrected velocities with the ratio of the measured sound speed with a fixed sound speed.

Typical processing responses are:

```
Was a fixed value of sound speed of 1500 [m/s] used when the instrument was setup? y/n y
Do you want to use nearby microcat data to correct the speed of sound?   y/n n
What constant value should be used as fixed salinity (in PSU) for speed of sound correction? 35.1
Mean deployment value for the old speed of sound [m/s]: 1500.00
Mean deployment value for the new speed of sound [m/s]: 1492.82
u-component despiking:
Further manual editing required? 0=no, 1=yes: 0
v-component despiking:
Further manual editing required? 0=no, 1=yes:
```

Then the recorded velocity components are transformed into true east and north components using the local magnetic declination estimated on the median of the deployment and recovery times of each mooring from the NOAA's National Geophysical Data Centre (http://www.ngdc.noaa.gov/geomag-web/).

Then spikes from the current meter data are removed using the spike/mean ratio criterion of 10, and visual inspection can also be used to identify remaining spikes. If data gaps are less than 24 hours, they are interpolated linearly in time, otherwise flag values are inserted.

Despiked data are low pass filtered using a 40-hour Butterworth filter to remove tidal and inertial oscillations, and are then interpolated onto 12-hour time steps (figure 4).

Finally, corrected data files that have been de-spiked and interpolated onto 12 hour time steps are saved in *.edt* files in ~\osnap\data\moor\proc\rteb1_XX_XXXX\nor\.

2.5. Stage 4 – Data gridding of individual deployment

A. Create new case and run the script:

~\osnap\exec\$cruise\stage3\gridding\CM\velocity_grid_nor

C. Data are saved as a mat file in *~\osnap\data\moor\proc\velocity_grid*

Stage 5 – Gridding and merging of the several years of deployment

Create dat files e.g

~ \osnap\exec\$cruise\stage3\gridding\CM\ rteb_CM_osnap_03_2016.dat

Note that if data instruments were not available (e.g. an instrument failed) then that should not be included in the .dat file. Make sure number order of instruments is sequential!

Copy old file e.g. *CM_rteb_merging_02_2015.m* and rename then edit as follows:

- Create new moor (e.g. moor3 = 'rteb_CM_osnap_03_2016')
- Rename directories
- Edit time variables **jg_end** and **lastyeardata**
- Add a new cruise (e.g. 2c. OSNAP 3 (DY053 --> $CRUISE)), copy the relevant code and rename variables to reflect new cruise number (i.e. 3 > 4, 4 > 5 etc.)

Thoroughly check through the script for other instances that need to be changed with the addition of a new mooring.

DeepSeapHOx Data Processing

1. Stage 0 – Data Download

After download the data are backed up and transferred to the network drive, then copied onto the processing computer in the directory *osnap/data/moor/raw/$cruise/seaphox_caldip*

2. Stage 1 – Conversion to standard RDB format

The script *seaphox_call_caldip* performs stage1 processing on DeepSeapHOx data. It converts the data from raw to RDB format. The user needs to modify some information in the beginning of the script like the directory trees, the mooring name or caldip cast number. This script calls seaphox2rodb_01, which saves the file downloaded by the instrument software (stage 0) to the RDB formatted file *.raw a*nd produce summary plots and statistics for each instrument.

Download DeepSeapHOx and transfer to the processing computer:
~/osnap/data/moor/raw/$cruise/seaphox/ . For the 2017-2018 deployment of the SeapHOx 117 the data appear to have been written on several files C00000*.CSV, and several data files from the test on DY078 where still present on the memory of the DeepSeapHOx. After keeping a copy of all the original files, the data files were manually edited to make sure than only the data relevant for the 2017-2018 RTEB1 deployment were kept in this the seaphox raw directory.

If the relevant data are present in several C000*.CSV file, edit and run the script (with bash shell) *~/osnap/data/moor/raw/$cruise/seaphox/merge_sort_CSVfiles.sh.* This script will sort the lines of several CSV files generated by the SeapHOx and merge them into a single file. The name of this file, containing all the data, has to be indicated in the moor_filenames.txt file (e.g. rteb1_04_2017_filenames.txt).

Make sure that the relevant info.dat file contains the serial number of the DeepseapHOx (e.g. moor/proc/rteb1_04_2017/rteb1_04_2017 info.dat).
Run process_seaphox_$cruise.m producing timeseries for each instrument, converting raw data to RODB format and summary statistics.

Export to oceansites format [if required]

The mooring data are annually submitted to Feili Li (OSNAP) in Oceansites netcdf format. Currently we are generating this format (08/2019) though ultimately it may become BODC's remit. Loic modified a suite of scripts to do this in *osnap\exec\$cruise\export_Oceansites.*

The top-level script

osnap\exec\$cruise\export_Oceansites\convert_SAMS_mooring_to_oceansites.m

sets up directory paths and declares metadata to populate the netcdf file. This script has to be run once per mooring, note the commented out moorings on lines ~26-29. Scan through the script for any other tweaks necessary for the new mooring deployment. This script calls

osnap\exec\$cruise\export_Oceansites\rodb_to_oceansitesnetcdf

which loads the Microcat and current meter datafiles and converts them to netcdf.

Note: the script searches for known instrument IDs (Microcat = 337 or 335, Nortek = 368 or 370) to populate its arrays. This caused some indexing issues when ODO Microcats (with oxygen sensors) were added to the array for the 04_2017 mooring. Be aware that some careful checking will be required if another new instrument type is added to the moorings, including (but not limited to) lines ~56 – 83 and 86-98. This also applied to the gridding scripts (stages 4 and 5).

Data and plots are saved to

osnap\exec\$cruise\export_Oceansites\oceansites_format

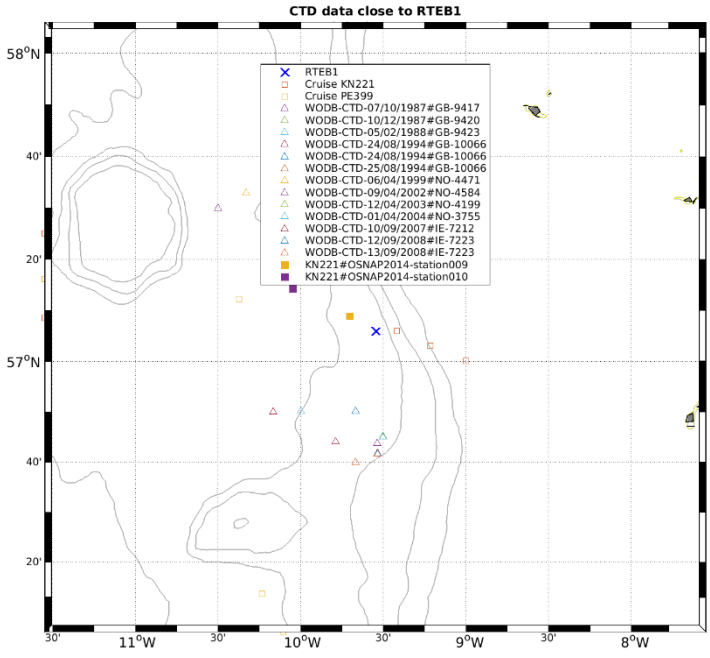
These tend to be small files which can be emailed as a .zip file or similar.

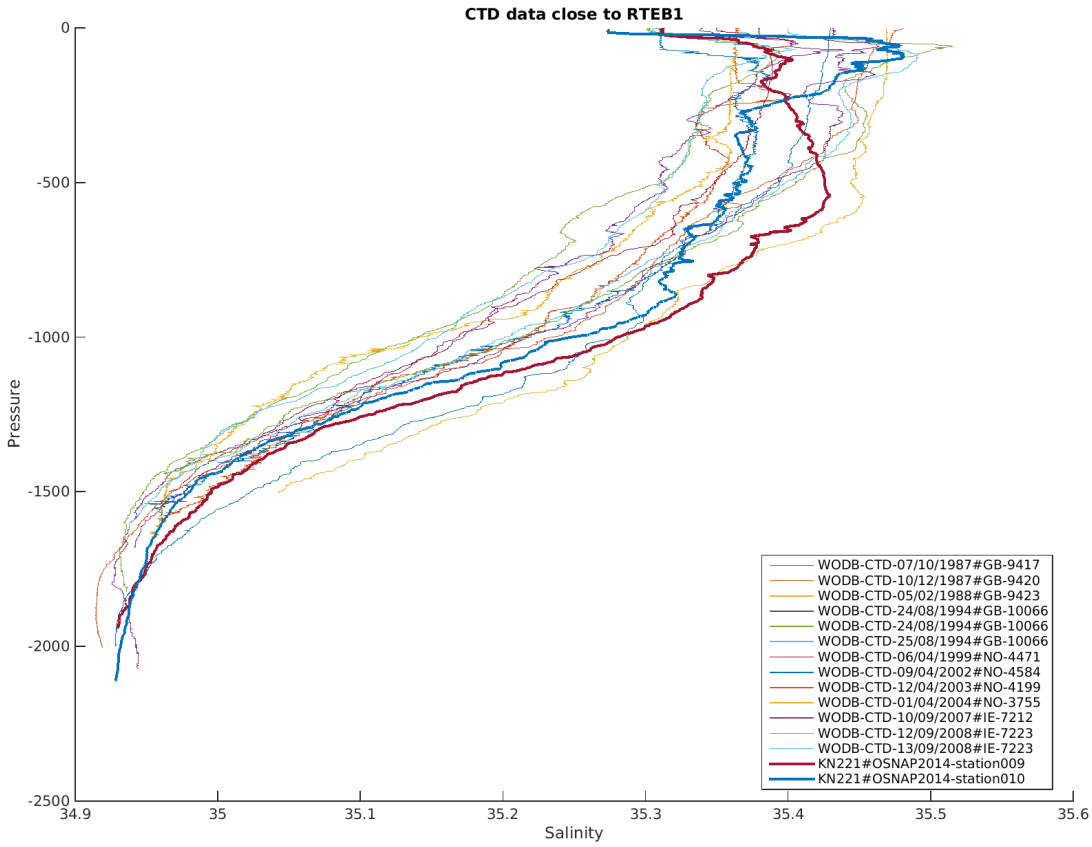
Export to NetCDF for CLASS data working group

Change the filenames in the script *osnap\exec\$cruise\export_THREDDS/Export_moor_to_NetCDF,m* to reflect the most recently processed data and run.
The

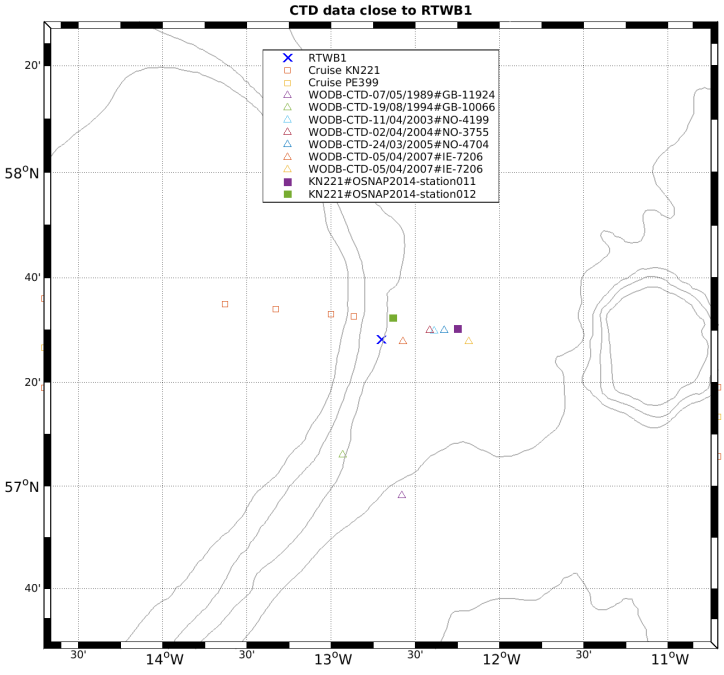
Appendix: Historical and OSNAP CTD profiles close to the moorings

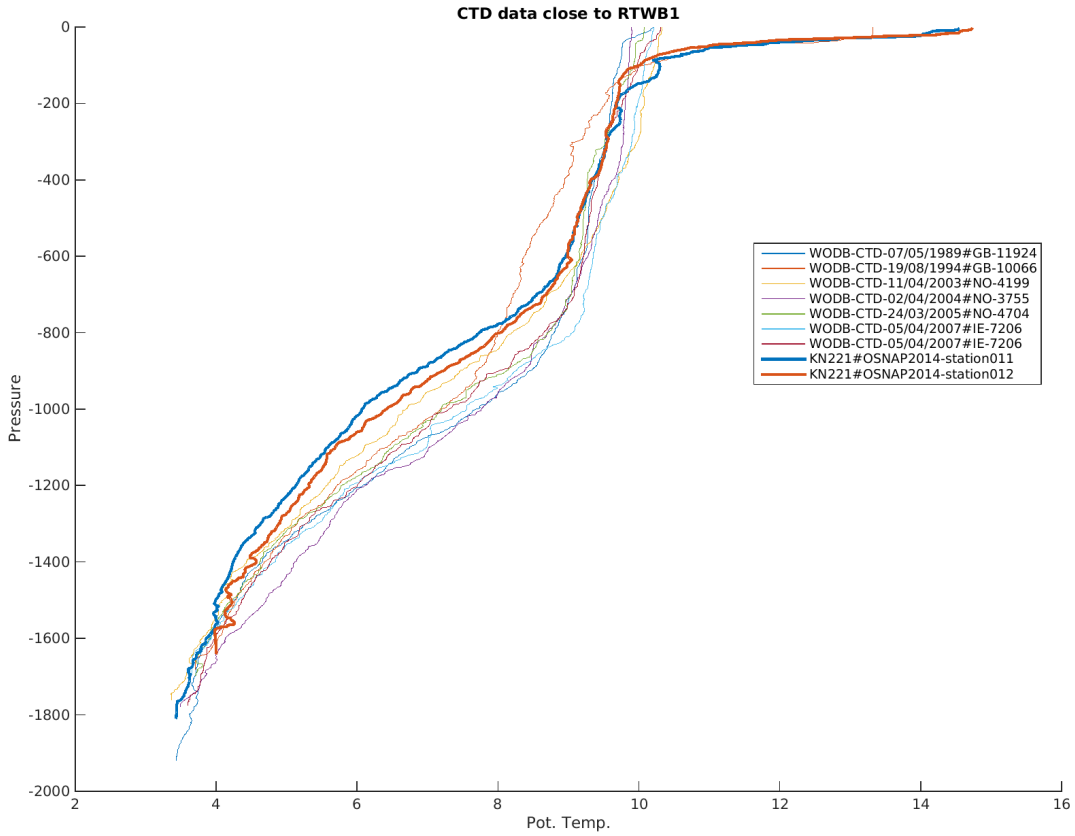
1.1. RTEB1



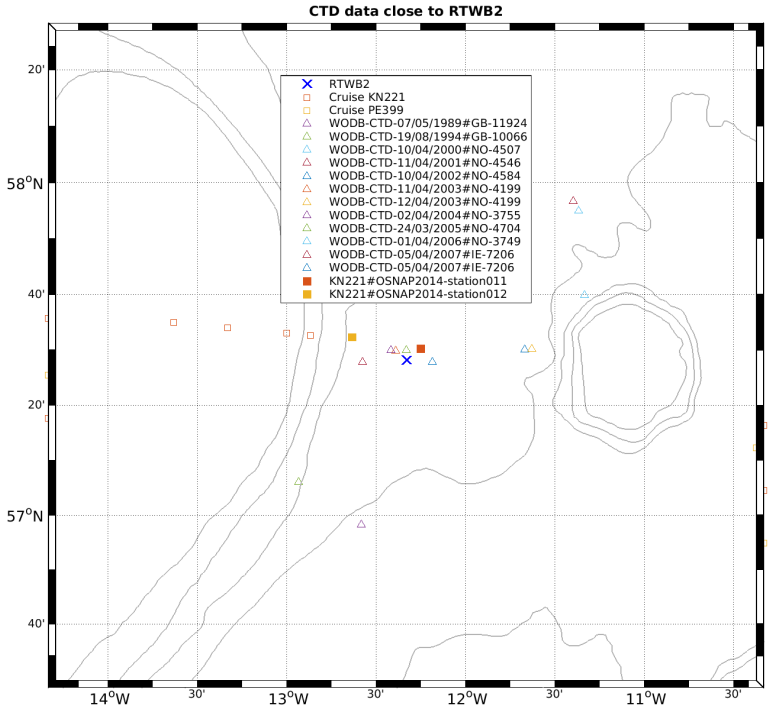

CTD data close to RTEB1
Pressure
Salinity
0
-500
-1000
-1500
-2000
-2500
34.9
35
35.1
35.2
35.3
35.4
35.5
35.6
WODB-CTD-07/10/1987#GB-9417
WODB-CTD-10/12/1987#GB-9420
WODB-CTD-05/02/1988#GB-9423
WODB-CTD-24/08/1994#GB-10066
WODB-CTD-24/08/1994#GB-10066
WODB-CTD-25/08/1994#GB-10066
WODB-CTD-06/04/1999#NO-4471
WODB-CTD-09/04/2002#NO-4584
WODB-CTD-12/04/2003#NO-4199
WODB-CTD-01/04/2004#NO-3755
WODB-CTD-10/09/2007#IE-7212
WODB-CTD-12/09/2008#IE-7223
WODB-CTD-13/09/2008#IE-7223
KN221#OSNAP2014-station009
KN221#OSNAP2014-station010

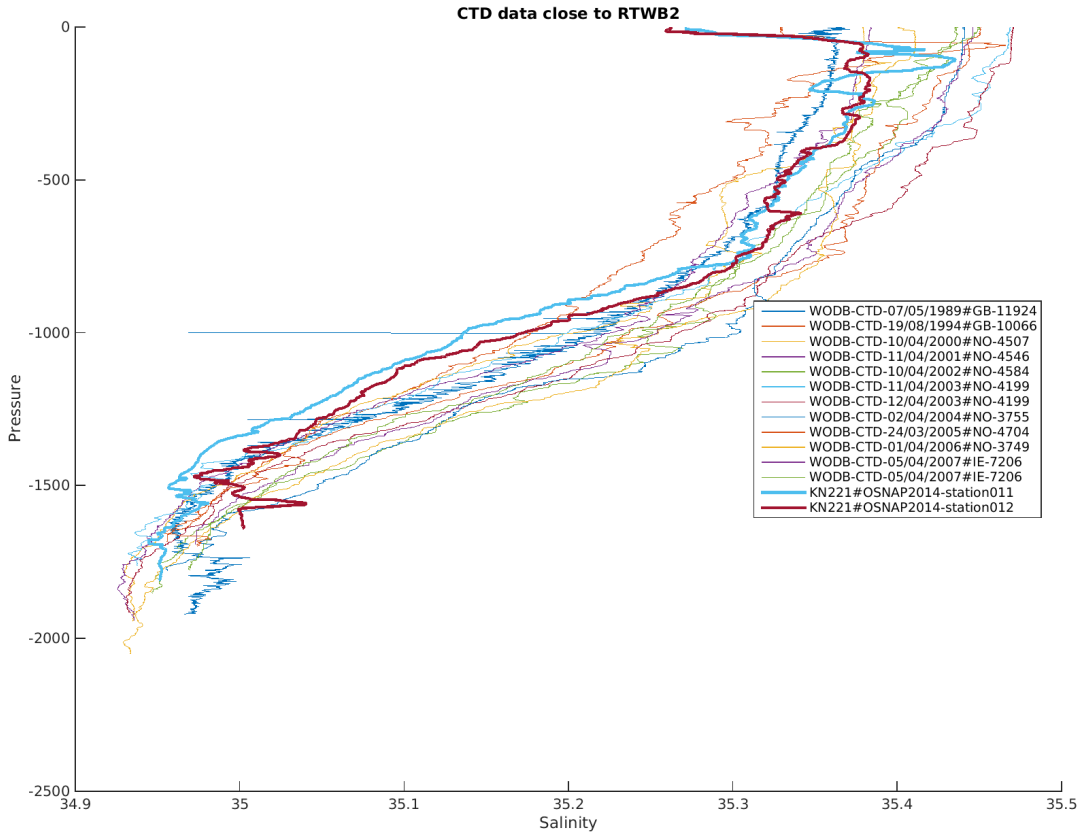
1.2. RTWB1


CTD data close to RTWB1
RTWB1
Cruise KN221
Cruise PE399
WODB-CTD-07/05/1989#GB-11924
WODB-CTD-19/08/1994#GB-10066
WODB-CTD-11/04/2003#NO-4199
WODB-CTD-02/04/2004#NO-3755
WODB-CTD-24/03/2005#NO-4704
WODB-CTD-05/04/2007#IE-7206
WODB-CTD-05/04/2007#IE-7206
KN221#OSNAP2014-station011
KN221#OSNAP2014-station012
58°N
57°N
14°W
13°W
12°W
11°W

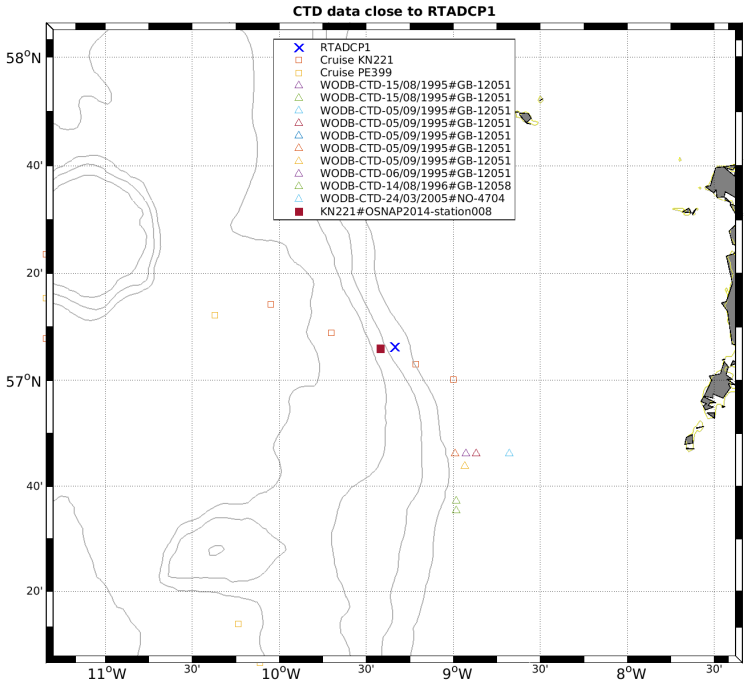

CTD data close to RTWB1
Pressure
Pot. Temp.
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
-1800
-2000
2
4
6
8
10
12
14
16
WODB-CTD-07/05/1989#GB-11924
WODB-CTD-19/08/1994#GB-10066
WODB-CTD-11/04/2003#NO-4199
WODB-CTD-02/04/2004#NO-3755
WODB-CTD-24/03/2005#NO-4704
WODB-CTD-05/04/2007#IE-7206
WODB-CTD-05/04/2007#IE-7206
KN221#OSNAP2014-station011
KN221#OSNAP2014-station012

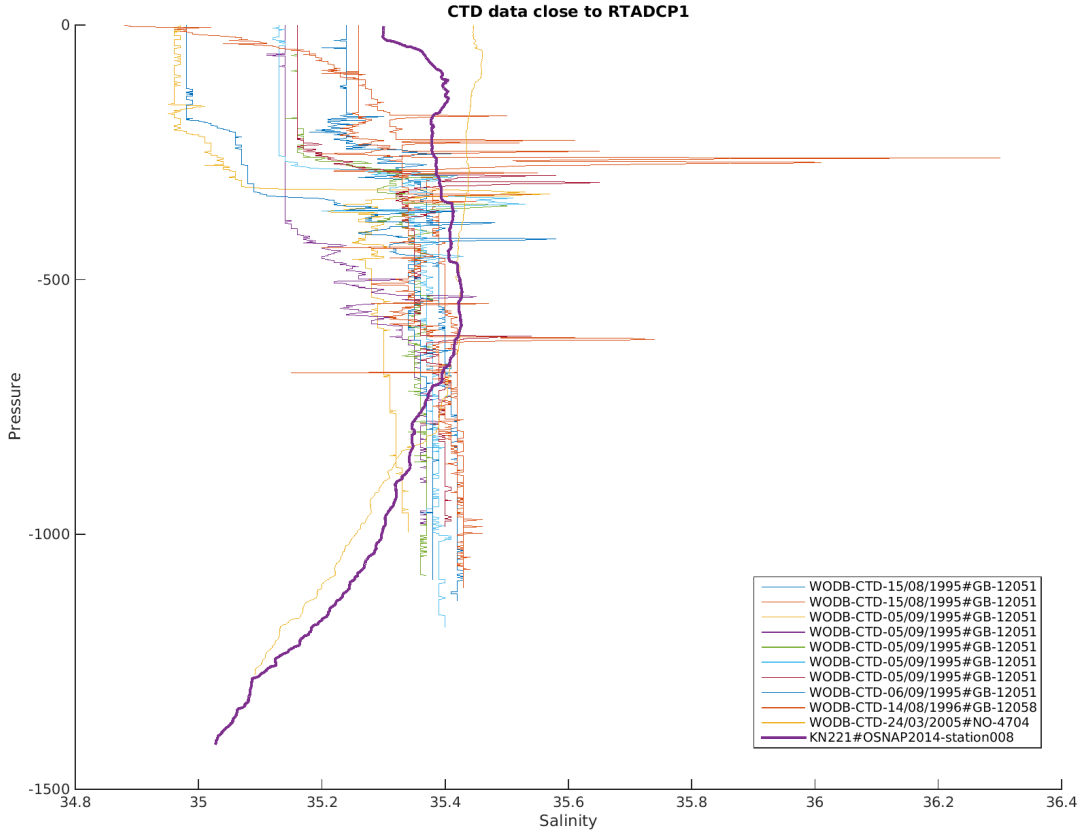
1.3. RTWB2


CTD data close to RTWB2
RTWB2
Cruise KN221
Cruise PE399
WODB-CTD-07/05/1989#GB-11924
WODB-CTD-19/08/1994#GB-10066
WODB-CTD-10/04/2000#NO-4507
WODB-CTD-11/04/2001#NO-4546
WODB-CTD-10/04/2002#NO-4584
WODB-CTD-11/04/2003#NO-4199
WODB-CTD-12/04/2003#NO-4199
WODB-CTD-02/04/2004#NO-3755
WODB-CTD-24/03/2005#NO-4704
WODB-CTD-01/04/2006#NO-3749
WODB-CTD-05/04/2007#IE-7206
WODB-CTD-05/04/2007#IE-7206
KN221#OSNAP2014-station011
KN221#OSNAP2014-station012
20'
58°N
40'
20'
57°N
40'
14°W
30'
13°W
30'
12°W
30'
11°W
30'


CTD data close to RTWB2
Pressure
Salinity
0
-500
-1000
-1500
-2000
-2500
34.9
35
35.1
35.2
35.3
35.4
35.5
WODB-CTD-07/05/1989#GB-11924
WODB-CTD-19/08/1994#GB-10066
WODB-CTD-10/04/2000#NO-4507
WODB-CTD-11/04/2001#NO-4546
WODB-CTD-10/04/2002#NO-4584
WODB-CTD-11/04/2003#NO-4199
WODB-CTD-12/04/2003#NO-4199
WODB-CTD-02/04/2004#NO-3755
WODB-CTD-24/03/2005#NO-4704
WODB-CTD-01/04/2006#NO-3749
WODB-CTD-05/04/2007#IE-7206
WODB-CTD-05/04/2007#IE-7206
KN221#OSNAP2014-station011
KN221#OSNAP2014-station012

1.4. RTADCP1


CTD data close to RTADCP1
RTADCP1
Cruise KN221
Cruise PE399
WODB-CTD-15/08/1995#GB-12051
WODB-CTD-15/08/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-06/09/1995#GB-12051
WODB-CTD-14/08/1996#GB-12058
WODB-CTD-24/03/2005#NO-4704
KN221#OSNAP2014-station008
58°N
40'
20'
57°N
40'
20'
11°W
30'
10°W
30'
9°W
30'
8°W
30'


CTD data close to RTADCP1
Pressure
0
-500
-1000
-1500
34.8
35
35.2
35.4
35.6
35.8
36
36.2
36.4
Salinity
WODB-CTD-15/08/1995#GB-12051
WODB-CTD-15/08/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-05/09/1995#GB-12051
WODB-CTD-06/09/1995#GB-12051
WODB-CTD-14/08/1996#GB-12058
WODB-CTD-24/03/2005#NO-4704
KN221#OSNAP2014-station008