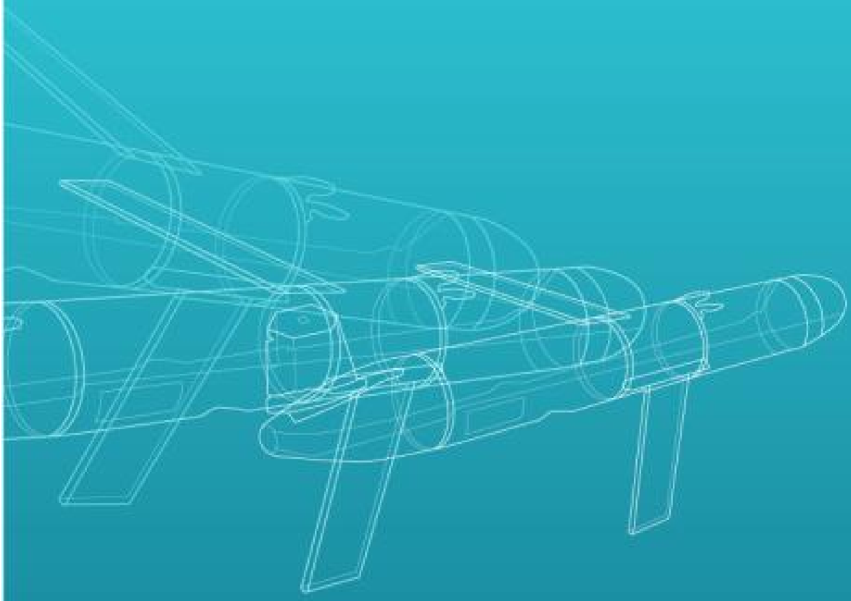
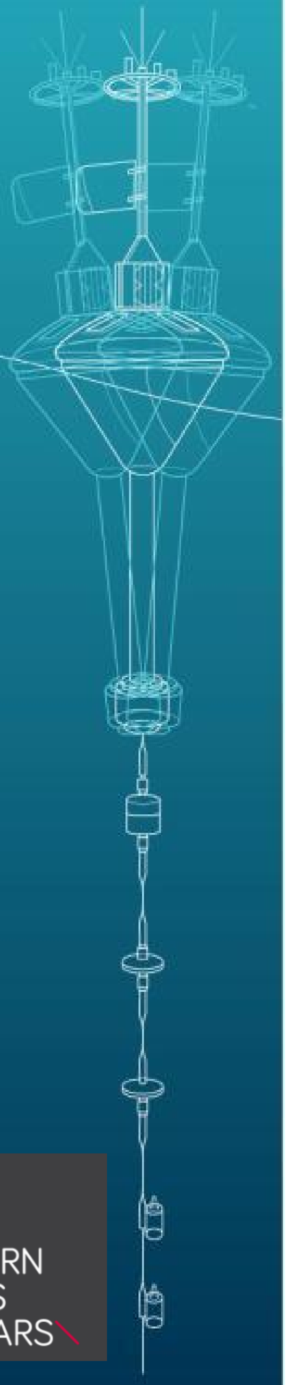




# Glider Toolbox

version 1.3.1

SOCIB DATA CENTER FACILITY

***Manual***

SOCIB Balearic Islands Coastal Observing and Forecasting System

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

GOVERN ILLES BALEARS

The glider toolbox (GTB) is a set of MATLAB/Octave scripts and functions developed at SOCIB to manage the data collected by a glider fleet. They cover the main stages of the data management and data processing both in real and delayed time modes. This document is a guide for users to start running the glider toolbox in glider data. It describes the main concepts of the GTB necessary to understand the usability of the tool from a user point of view.

# 1. Overview

The Coastal Ocean Observing and Forecasting System (SOCIB for Sistema de Observación Costero y de Predicción de las Islas Baleares) is a multi-platform distributed and integrated system that provides streams of oceanographic data and modelling services to support operational oceanography in a European and international framework (see www.socib.eu). Among the observing facilities, glider observations play a critical role in the understanding of water transport between the Mediterranean sea and the Atlantic ocean. The glider toolbox was developed to support our glider operations, mainly glider data management and processing. A large effort was made to produce final data that follow international standard. Because of the success of the GTB in our internal operations, we share this tool with the international community. We believe that the external users will benefit of the functionalities included in this toolbox and we will have the opportunity for improvement from the input of the glider community.

SOCIB glider data is publicly available via various visualization and dissemination tools. As examples, data is available using thredd services. Our data is organized by glider, mission, processing level and year of the deployment (see list of deployments). Our thredds and web sites include a set of visualization tools to help users accessing metadata and data of the NetCDF files in a friendly manner. Users can also visualize deployments using our dapp application at http://apps.socib.es/dapp/. This set of tools are helpful to understand the results from the glider toolbox.

## 2. Requirements

- Linux (recommended) or Windows OS
- Matlab 2012, Matlab 2014 or Octave (Matlab is recommended).
- C++ compiler
- Imagemagick
- External libraries
    - mexnc and snctools (Commit #4053):
      NetCDF library preferences ( Sourfoge/mexcdf). Octave users may install octcdf instead of snctools.
    - General Polygon Clipper library (GPC library) by Alan Murta
    - m_map 1.4 library: Mapping toolbox (m_map1.4.tar.gz)
        - GSHHS high-resolution coastline database v2.0 (download and follow instructions)
    - seawater 3.3 library: CSIRO Sea Water Library (seawater_ver3_3.1.zip)
    - m2html: inline html documentation generator
- Webb Research package for Slocum database
  (github: kerfoot/slocum for linux or coolcloud.mote/slocum for windows)
    - dba2asc: converts Slocum binary data to ascii format
    - dba_merge: merge ascii navigation and science files
    - dba_sensor_filter and dba2_time_filter: filter data based on sensor or timestamps
    - dba2_orig_matlab: converts ascii Slocum files to Matlab/Octave data files
- Matlab jar libraries:
    - Data base drivers 9.4 (postgresql-9.4.1212.jre6.jar). MATLAB interface with the PostgreSQL JDBC (Java Database Connectivity) driver. An Octave alternative would be needed. It may be installed from the distribution repositories as a package libpg-java (renamed to libpostgresql-jdbc- java in Debian repositories on March 2012).
    - NetCDF 4.2 library (netcdfAll-4.2.jar from Unidata)

# 3. Installation

- Clone repository at https://github.com/socib/glider_toolbox.
  This creates a directory glider_toolbox with the following structure:
  - bin: binary scripts supporting the GTB
  - config: contains configuration files.
  - ext_lib: contains external libraries and binaries.
  - glider_data: contains files for local processing when using default configuration
  - m: contains Matlab/Octave code
  - documentation: contains schematics and manuals
- Download and uncompress external libraries, binary files and matlab jars. Users can run the gtb_install_extlibs under the bin folder. This script downloads and uncompresses these files automatically to the right locations.
  - External libraries to ext_lib/lib resulting in the following folders under ext_lib/lib: m2html, m_map, mexcdf/mexnc, mexcdf/snctools, seawater.
  - GPC library under m/mex_tools/gpcl (gpc.c and gpc.h). Notice that the installation script downloads the files under ext_lib and creates a soft link to m/mex_tools.
  - Webb Research binaries to ext_lib/bin
- Optionally, you can add the GSHHS high-resolution coastline database. The installation script included this database in the installation.
- Download jar libraries and configure Matlab. Users that run the installation script can find the jar files under ext_lib/matlab
  - Copy jar files to $MATLAB_ROOT/java/jarext
  - Add path to $MATLAB_ROOT/toolbox/local/classpath.txt
    - $matlabroot/java/jarext/netcdfAll-4.2.jar
    - $matlabroot/java/jarext/postgresql-9.4.1212.jre6.jar
- Setup Mex libraries from Matlab. Notice that GPC library must be added before this setup. Run matlab glider_toolbox and call the following commands.

```
>> cd m
>> mex -setup  (select 0 to answer prompt question)
>> configGliderToolboxPath
>> setupMexPosixtime
>> setupMexSFTP
>> exit
```

- Create GTB library documentation: Run **make doc** from the glider_toolbox. Additionally, an online version of the of the documentation is available at http://socib.es/users/glider/glider_toolbox/doc/.
- Edit configuration files for real time (config/configMainRT.txt) and delayed mode (config/configMainDT.txt).
    - Local and public paths (default writes data at glider_data under the installation path)
    - Database information (default does not use database but the deployment text files under the config directory)
    - Dockserver (required for real time)

# 4. Quick Start

In this section we explain a few examples to guide users for a quick start processing glider data with GTB. It assumes that the the glider toolbox is installed in /path/to/glider_toolbox as explained in the previous section. This is a practical description of the processing. The explanation of the processing scripts and function is in the Processing section. For more details and processing options refer to next sections. Example of figures that are produced by the GTG are in Appendix A and B. As mentioned in the introduction of this document, users can find examples of produced netCDF at:

http://thredds.socib.es/thredds/catalog/auv/glider/catalog.html.

Follow the steps that are described below as guidelines to process data in real time or delayed time using default configurations. Section C contains an overview for ways to process in a more customized way. These examples are explained in the context of linux users but window users could perform the equivalent steps.

## a. Real time processing

- Add dockserver information to configMainRT.txt under config folder. Add the following lines

```
dockservers.active                  = 1
dockservers.remote_base_dir  = /var/opt/gmc/gliders
dockservers.remote_xbd_dir    = from-glider
dockservers.remote_log_dir     = logs
dockservers.server(1).host       = 192.168.0.1
                    (set your the ip of your remote dockserver)
dockservers.server(1).user       = localuser
                    (set your the name of the user accessing your remote dockserver)
dockservers.server(1).pass       = yourSuperSecretPwd
                    (set the password for the user to access your remote dockserver)
```

- Set the deployment description file. Edit the deploymentRT.txt in the config folder.

**Note**:

In this example, the GTB looks for data under /var/opt/gmc/gliders/myGlider/20171101 in the remote dockserver which is a path built from dockservers.remote_base_dir and deployment_list(1).deployment_start. 20171101 corresponds to 7.3700e+05 which is also the day 305 of the year. The path where the data is downloaded from the remote dockserver and copied to /path/to/glider_toolbox/glider_data/binary in the local drive. In addition, the format of the names of SBD and TBD files should follow:

myglider-2017-305-X-0.sbd and myglider-2017-305-X-0.tbd

- Required parameters

  |  |  |
  |---|---|
  | deployment_list(1).deployment_id | = 1 |
  | deployment_list(1).deployment_name | = NOV2017_TERESA_GFMR0064 |
  | deployment_list(1).deployment_start | = 7.3700e+05 |
  | (days since Jan 0, 0000, this example corresponds to 01-Nov-2017) |  |
  | deployment_list(1).deployment_end | = NaN |
  | deployment_list(1).glider_name | = myglider |
  | deployment_list(1).glider_serial | = 518 |
  | deployment_list(1).glider_model | = Slocum G2 Deep |
  | deployment_list(1).glider_instrument_name | = myinstrument |
  | deployment_list(1).glider_deployment_code | = 0002 |

- Some optional parameters. These are used to populate the global attributes.

  |  |  |
  |---|---|
  | deployment_list(1).project | = MARS2015 |
  | deployment_list(1).abstract | = Some description |
  | deployment_list(1).project_url | = http://myproject.org/ |
  | deployment_list(1).principal_investigator | = John Smith |
  | deployment_list(1).principal_investigator_email | = john.smith@glider.edu |
  | deployment_list(1).author | = Pierre Lemoine |
  | deployment_list(1).author_email | = plemoine@glider.edu |
  | deployment_list(1).institution | = myInstitution |

- Set the DB access to false. Edit configMainRT.txt and add or modified the following line:
  - db_access.active  = 0

- Run the processing script main_glider_data_processing_rt
  - cd /path/to/glider_toolbox
  - matlab

  ```
  >> cd m
  >> main_glider_data_processing_rt
  >> exit
  ```
- Check results
  - cd /path/to/glider_toolbox
  - cd glider_data
  - The directory should contain:
    - binary
      - myglider-2017-305-3-0.sbd
      - myglider-2017-305-3-0.tbd
      - …
    - ascii
      - myglider-2017-305-3-0-sbd.dba
      - myglider-2017-305-3-0-tbd.dba
      - ...
    - figures
      - chlorophyll.png
      - ctd_profiles.png
      - current_map.png
      - density.png
      - flntu_profiles.png

- oxygen_concentration.png
- oxygen_profiles.png
- oxygen_saturation.png
- salinity.png
- temperature.png
- temperature_salinity.png
- turbidity.png

■ netcdf

- dep0001_myglider_myinstrument_L0_2017-11-01_data_rt.nc
- dep0001_myglider_myinstrument_L1_2017-11-01_data_rt.nc
- dep0001_myglider_myinstrument_L2_2017-11-01_data_rt.nc

## b. Delayed time processing

- Set the deployment description file. Edit the deploymentDT.txt in the config folder.

  - Required parameters

```
deployment_list(1).deployment_id                = 1
deployment_list(1).deployment_name              = NOV2017_TERESA_GFMR0064
deployment_list(1).deployment_start             = 7.3700e+05
            (days since Jan 0, 0000, this example corresponds to 01-Nov-2017)
deployment_list(1).deployment_end               = NaN
deployment_list(1).glider_name                  = myglider
deployment_list(1).glider_serial                = 518
deployment_list(1).glider_model                 = Slocum G2 Deep
deployment_list(1).glider_instrument_name       = myinstrument
deployment_list(1).glider_deployment_code       = 0002
```

  - Some optional parameters. These are used to populate the global attributes.

```
deployment_list(1).project                          = MARS2015
deployment_list(1).abstract                         = Some description
deployment_list(1).project_url                      = http://myproject.org/
deployment_list(1).principal_investigator           = John Smith
deployment_list(1).principal_investigator_email = john.smith@glider.edu
deployment_list(1).author                           = Pierre Lemoine
deployment_list(1).author_email                     = plemoine@glider.edu
deployment_list(1).institution                      = myInstitution
```

- Copy the glider data from the DBD and EDB files from the glider compact flash to the glider_data directory. In this example, the GTB looks at for data /path/to/glider_toolbox/glider_data. The format of the names of DBD and EBD files should be as follow:

  myglider-2017-305-X-0.dbd and myglider-2017-305-X-0.ebd

  Users can use rename_dbd_files under /path/to/glider_toolbox/bin to rename the files (rename_dbd_files binary/*.DBD and rename_dbd_files binary/*.EBD)

- Set the DB access to false. Edit configMainDT.txt and add or modified the following line:
  - db_access.active  = 0

- Run the processing script main_glider_data_processing_dt
  - cd /path/to/glider_toolbox
  - matlab

    ```
    >> cd m
    >> main_glider_data_processing_dt
    >> exit
    ```
- Check results
  - cd /path/to/glider_toolbox
  - cd glider_data
  - The directory should contain:
    - binary
      - myglider-2017-305-3-0.dbd
      - myglider-2017-305-3-0.ebd
      - …
    - ascii
      - myglider-2017-305-3-0-dbd.dba
      - myglider-2017-305-3-0-ebd.dba
      - ...
    - figures
      - chlorophyll.png
      - ctd_profiles.png
      - current_map.png
      - density.png
      - flntu_profiles.png
      - oxygen_concentration.png
      - oxygen_profiles.png
      - oxygen_saturation.png
      - salinity.png
      - temperature.png
      - temperature_salinity.png
      - turbidity.png
    - netcdf
      - dep0001_myglider_myinstrument_L0_2017-11-01_data_dt.nc
      - dep0001_myglider_myinstrument_L1_2017-11-01_data_dt.nc
      - dep0001_myglider_myinstrument_L2_2017-11-01_data_dt.nc

## c. Custom processing

Real time and delayed time processing can be customized for more specific needs (see configuration section). Customization is mainly related to the configuration of the location and name format of the product files and the use of a database. The configTemplate.txt describes the parameters that could be configured. These configurations can be defined in the configMainRT.txt for real time processing or configMainDT.txt for delayed time processing. In addition, users can use the gliderDataProcessing function with the option ‘config’ to define their own configuration files for specific processing. The processing steps using deployment text file definitions are similar as the ones described above. The use of the database is described in the Deployment and Metadata section.

# 5. Processing

The processing steps are described in the diagrams in the documentation folder and in the socib documentation at http://socib.es/users/glider/glider_toolbox/. These documents and diagrams described the details of the software and processing steps. In this section we describe the different ways of processing data. For a practical usability point of view, refer to the previous section (Quick start).

There are two ways to process glider data for each data mode (real or delayed time): Processing scripts and processing function. In both cases, users edit configuration text files to set specific configurations. Contrary to previous versions of the GTB, editing the Matlab software is not required for GTB versions v1.3.0 (and up). Overall, users can start processing data with the default configuration. In this case, the data working directory is in glider_data folder at the installation path as described in the Quick Start section. The only required configuration is for real time mode to indicate the access configuration of the dockserver.

## a. Processing Scripts (real and delayed time)

Users may process glider data by running the main_glider_data_processing_***X***t scripts where ***X*** defines “r” or “d” for real or delayed time respectively. These scripts use the configMain***X***T.txt configuration files to overwrite the default configurations set by the Matlab functions. These configuration files are located under the config folder in the installation path of the GTB. Dockserver configuration of configMainRT.txt is required for real time mode. Real and delayed time processing may need access to database information to populate the global variables of the final products as explained in the metadata section. In this case, the configuration of the database access in the configMainRT.txt and configMainDT.txt is also needed. Users that are interested in processing data in custom folders must edig these two configuration files appropriately. They can see the configTemplate.txt files for a list of configurables.

### b. Processing Functions

The GTB recently includes a set of processing functions that allows its users to input custom information for specific runs. The gliderDataProcessing function allows users to process a list of deployments and the deploymentDataProcessing function is used for processing a single deployment. These functions are used for both, real and delayed time modes. A configuration file must be input. Users can input the default configuration files in the config folder for real time (configMainRT.txt) and delayed time (configMainDT.txt) or set an arbitrary configuration file for a specific processing. This approach allows for a more flexible processing and will replace the processing scripts in future releases. In fact, the gliderDataProcessing function can be used in the same way as the processing scripts. For more information refer to the documentation of the code.

# 6. Configuration files

Previous versions of the GTB required editing Matlab configuration functions to customize the configuration to the needs of the users. The current approach of the GTB version 1.30 (and up) differs in the sense that the configuration files are set to a default configuration that allows the user to process data in a basic mode without the need to edit any code. For a more complex setup, users add or edit text files that are read by the processing scripts and functions to overwrite the default values. With this approach, different types of data processing are performed by running the GTB using different configuration files. Configuration files may live in any place in the disk but it is recommended to use the config folder in the installation path for consistency. The configMainRT.txt and configMainDT.txt are used by the real and delayed time mode processing scripts as explained in the Processing section. Custom files may be created in the config folder with no arbitrary name format. These configuration files can be run using the processing functions that are described in the section above.

Any parameter that is defined in the configuration file will overwrite the default values that are assigned by the GTB code. The configTemplate.txt describes the possible configurations. Users can use the “#” symbol to write comments in their own configuration files. All of the configuration parameters are set to default values except for the database and dockserver information. Configuration of the dockserver access is required for real time processing. The configuration of the database is only required for users that use a database to retrieve the metadata information.

# 7. Deployment Metadata

Metadata of the deployment may be input in three different ways:

- database
- deployment files
- matlab variables

The fields must be defined by the deployment as shown in the table 1. The required fields of the structure are deployment_id, deployment_name, deployment_start, deployment_end, glider_name, glider_serial and glider_model. Additionally, other fields may be used to overwrite values of the global attributes of the NetCDF metadata (see code documentation for more details).

| Structure field | Database field |
|---|---|
| deployment_id | deployment_id |
| deployment_name | deployment_name |
| deployment_start | deployment_initial_date |
| deployment_end | deployment_end_date |
| glider_name | platform_name |
| glider_serial | instrument_serial |
| glider_model | instrument_model |

**Table1 - Deployment information**

## a. Metadata Database

The GTB provides tools to retrieve the deployment metadata from a database. Users must enable the database access by setting db_access.active to 1 and defining the database access in the configuration file (configMain***X***T.txt). By default, a table (named deployment) in a database (named instrumentation) is query to retrieve the keywords that match the fields of the deployment structure as described by table 1.

Potentially, users can use any database, table and field names by editing the configRTDeploymentInfoQueryDB.m and configDTDeploymentInfoQueryDB.m.

## b. Metadata Deployment Text Files

Deployment metadata can be input by using deployment files. The processing scripts use the deploymentRT.txt and deploymentDT.txt files in the config folder. The deploymentDataProcessing and gliderDataProcessing processing functions accept any arbitrary deployment file as an input. All the information must be set in the deployment files and the "#" symbol is used to write comments. Users may refer to the deploymentRT.txt and deploymentDT.txt files as an example.

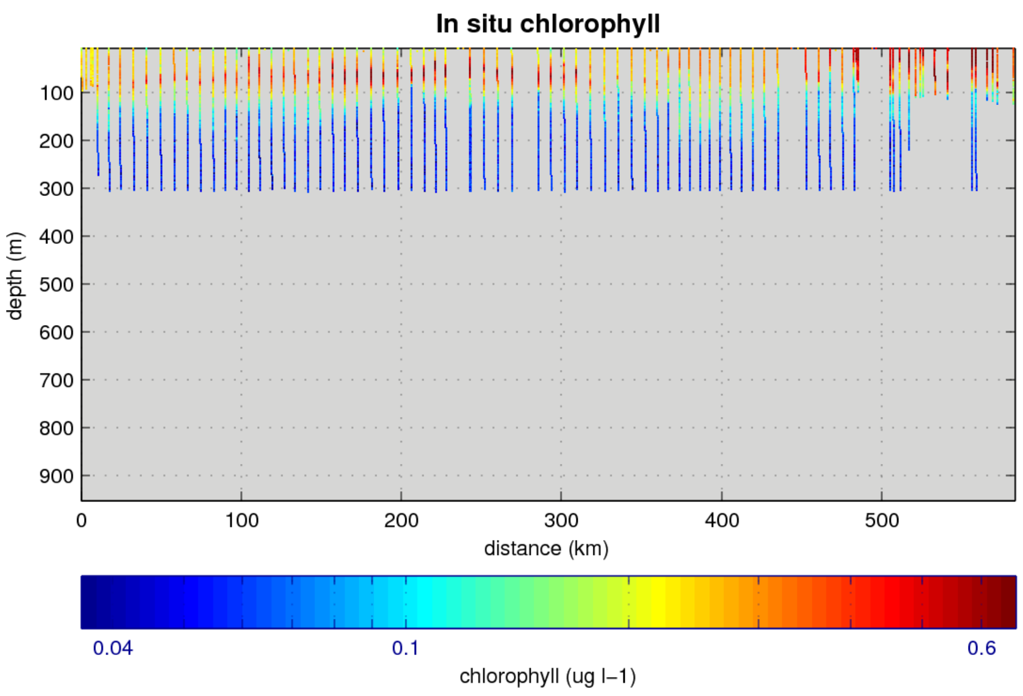
The main_glider_data_processing_dt and main_glider_data_processing_rt processing scripts read the deployment files when the database access is disabled (db_access.active is set to 0). The gliderDataProcessing processing function accepts any arbitrary deployment file but the database access must also be disabled for the function to use the deployment file. Below we present an example of the definition of two deployments:

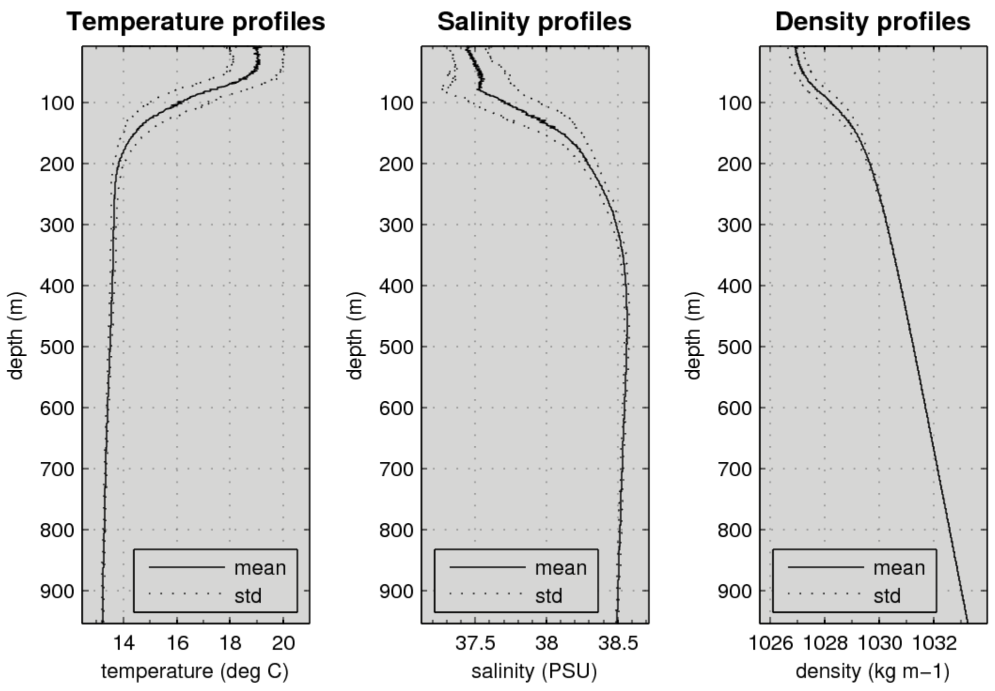
```
deployment_list(1).deployment_id             = 9
deployment_list(1).deployment_name           = NOV2017_TERESA_GFMR0064
deployment_list(1).deployment_initial_date = 7.3700e+05
deployment_list(1).deployment_end_date       = NaN
deployment_list(1).platform_name             = teresa
deployment_list(1).instrument_serial         = 518
deployment_list(1).instrument_model          = Slocum G2 Deep

deployment_list(2).deployment_id             = 10
deployment_list(2).deployment_name           = NOV2017_SDEEP04_GFMR0065
deployment_list(2).deployment_initial_date = 7.3700e+05
deployment_list(2).deployment_end_date       = NaN
deployment_list(2).platform_name             = sdeep04
deployment_list(2).instrument_serial         = 520
deployment_list(2).instrument_model          = Slocum G2 Deep
deployment_list(2).abstract                  = Deployment for scientific tests
deployment_list(2).author                    = John Smith
```

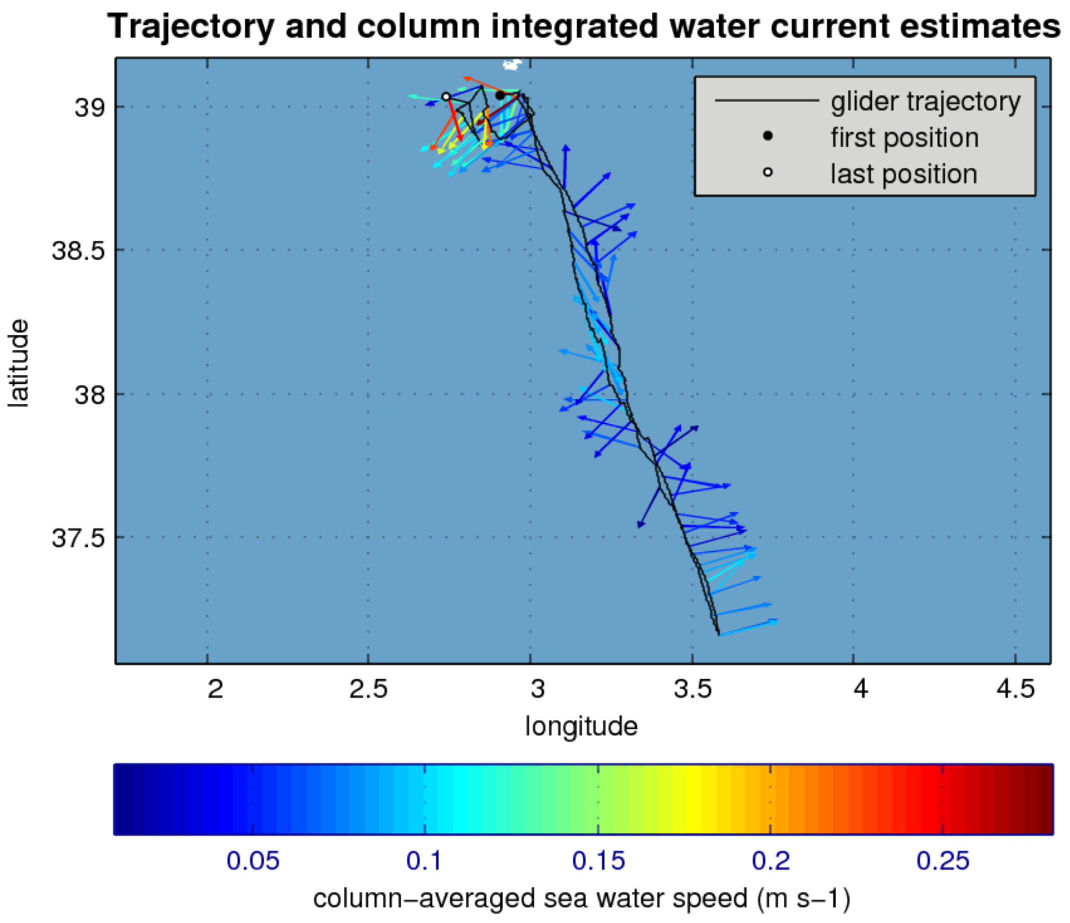
## c. Metadata Deployment Structure

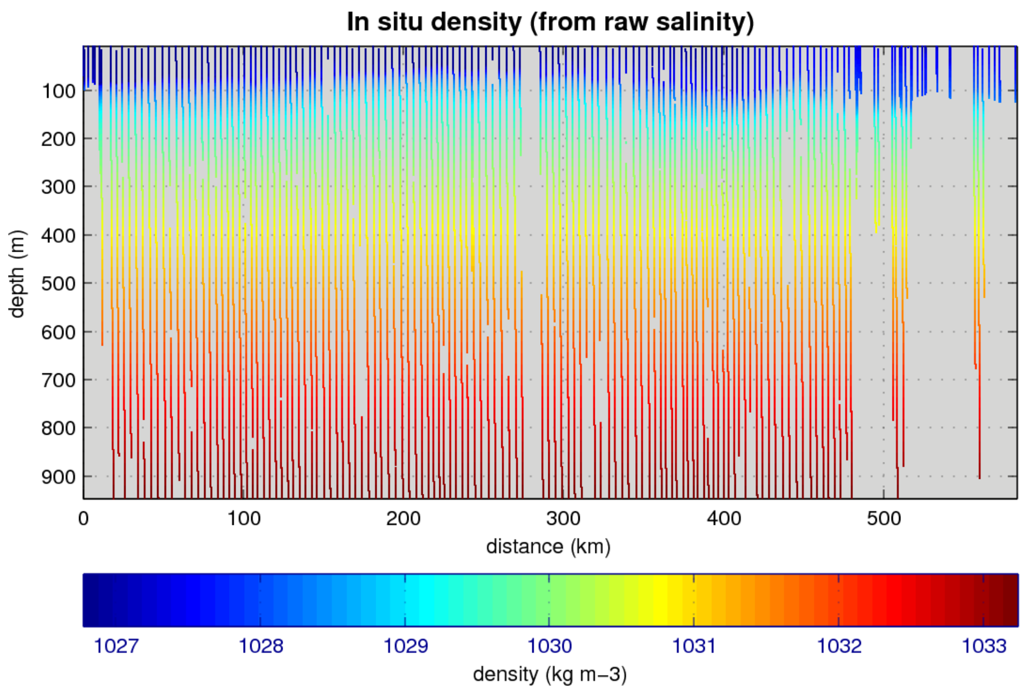
The deployment metadata may be input to the gliderDataProcessing deploymentDataProcessing functions as structures. Notice that processing scripts do not provide this option. The gliderDataProcessing function takes an array of structures and calls the deploymentDataProcessing function for each element of the array that represents a single deployment. As for the other options to define the deployments, the required fields are described in table 1 and additional fields may be used to overwrite global attributes of the NetCDF outputs.

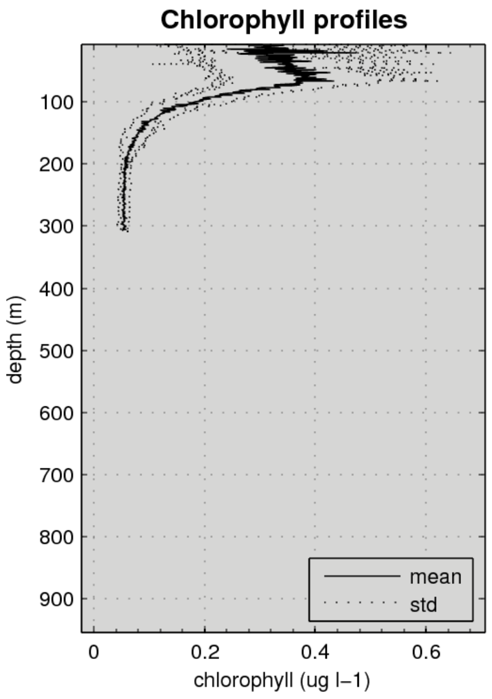
# 8. APPENDIX A: Real time figures

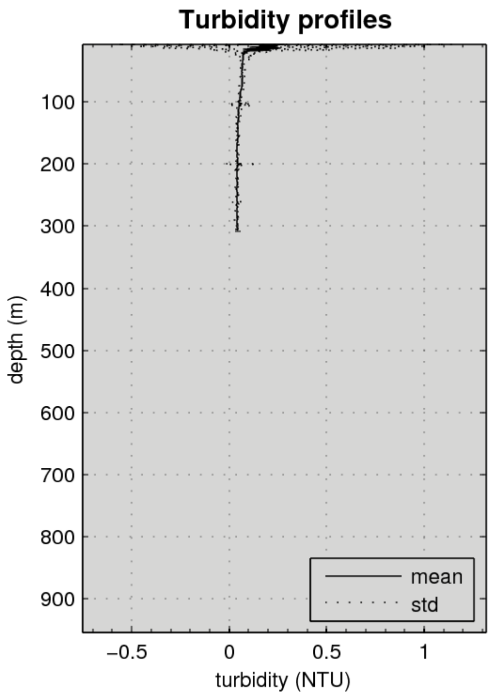


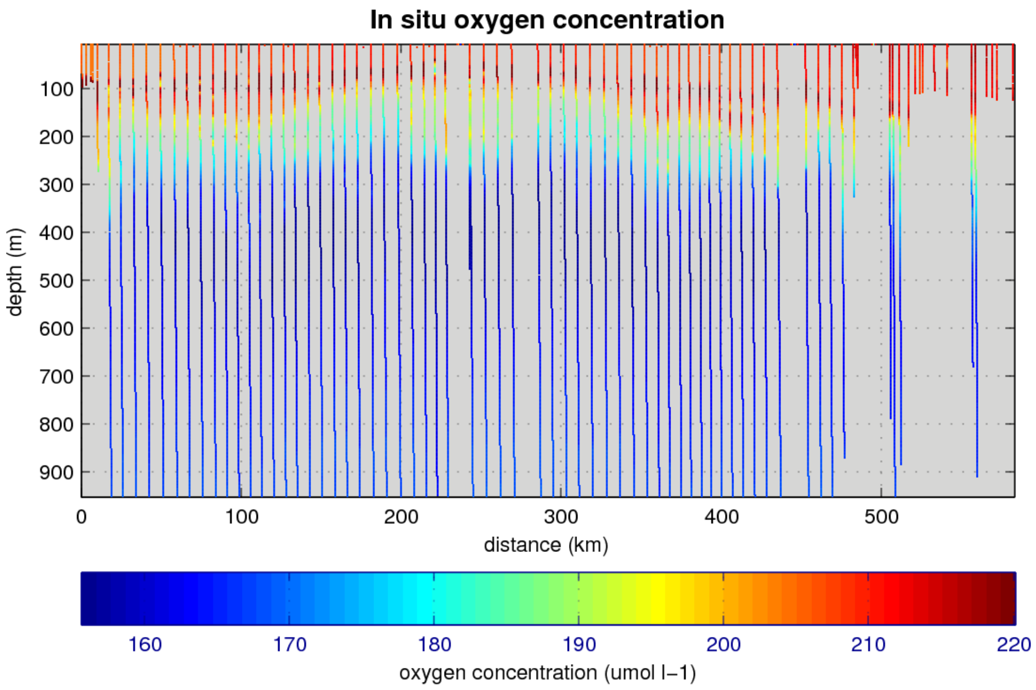


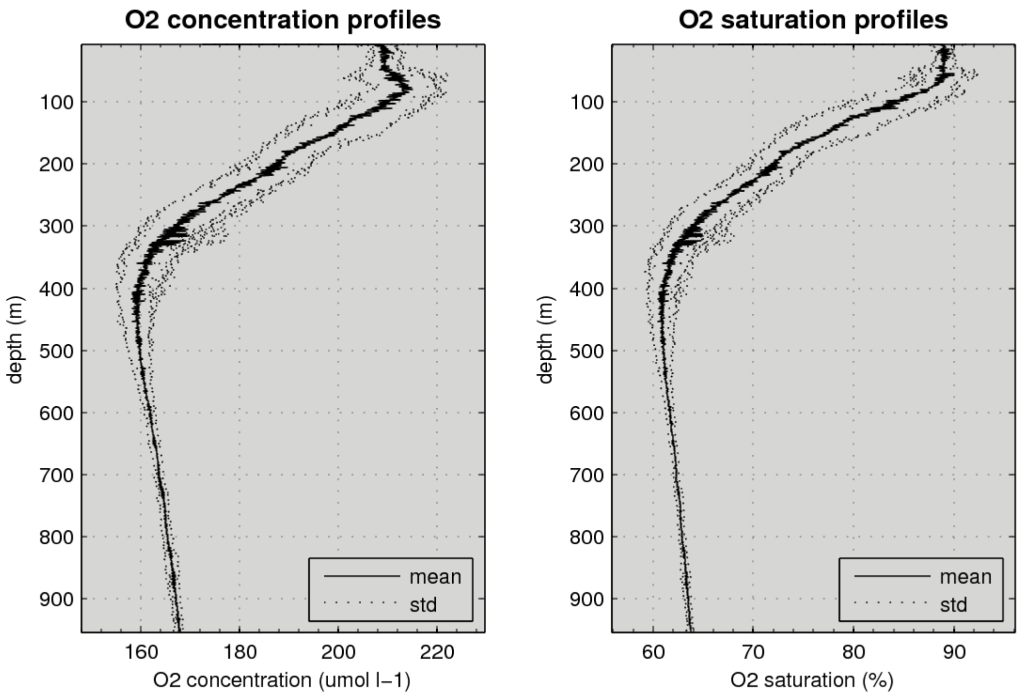

Trajectory and column integrated water current estimates
glider trajectory
first position
last position
latitude
longitude
39
38.5
38
37.5
2
2.5
3
3.5
4
4.5
0.05
0.1
0.15
0.2
0.25
column–averaged sea water speed (m s–1)

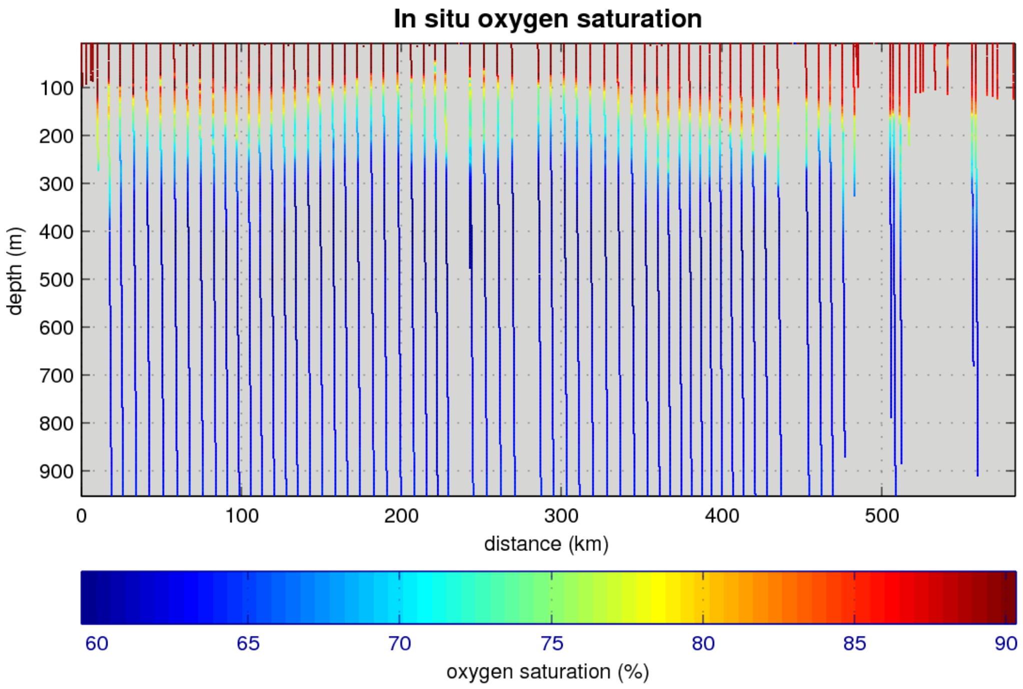

In situ density (from raw salinity)
depth (m)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
distance (km)
1027
1028
1029
1030
1031
1032
1033
density (kg m–3)

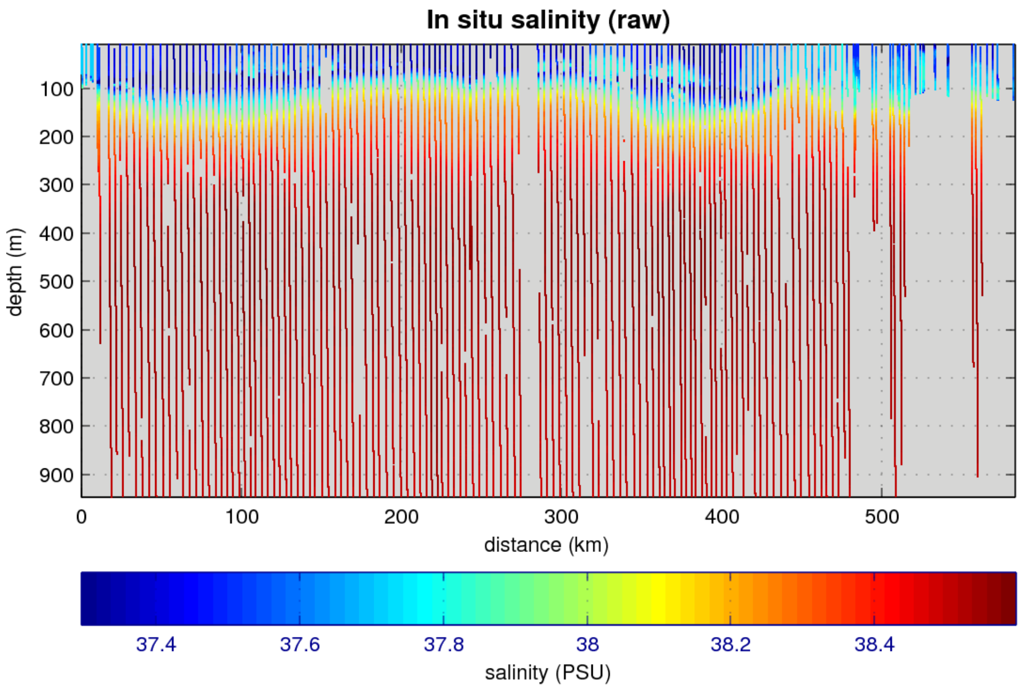

Chlorophyll profiles
depth (m)
100
200
300
400
500
600
700
800
900
0
0.2
0.4
0.6
chlorophyll (ug l−1)
mean
std

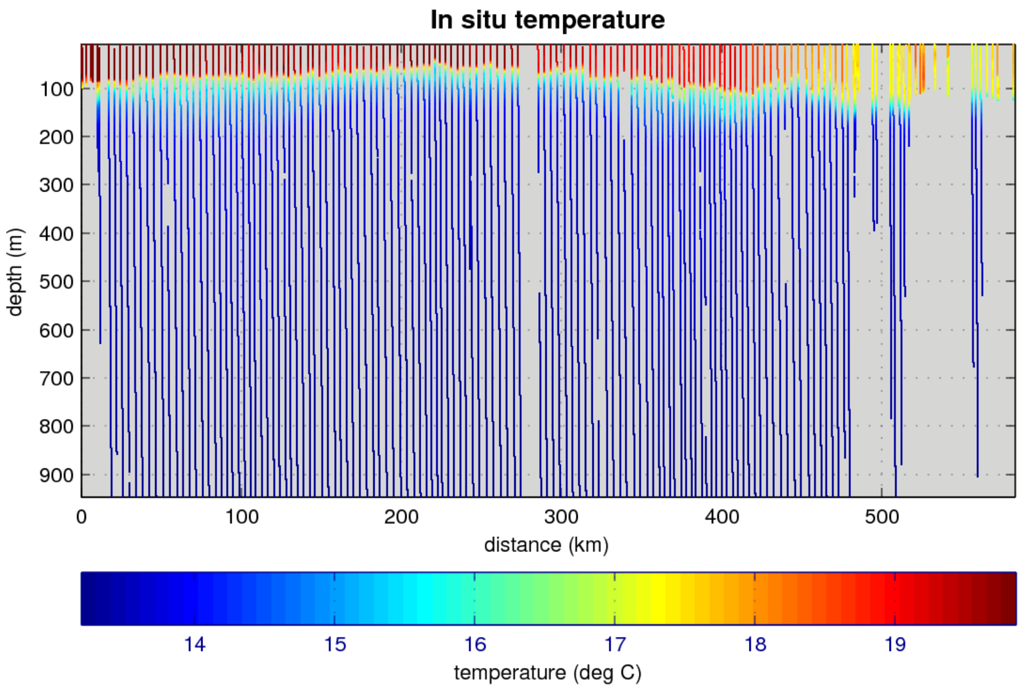

Turbidity profiles
depth (m)
100
200
300
400
500
600
700
800
900
−0.5
0
0.5
1
turbidity (NTU)
mean
std

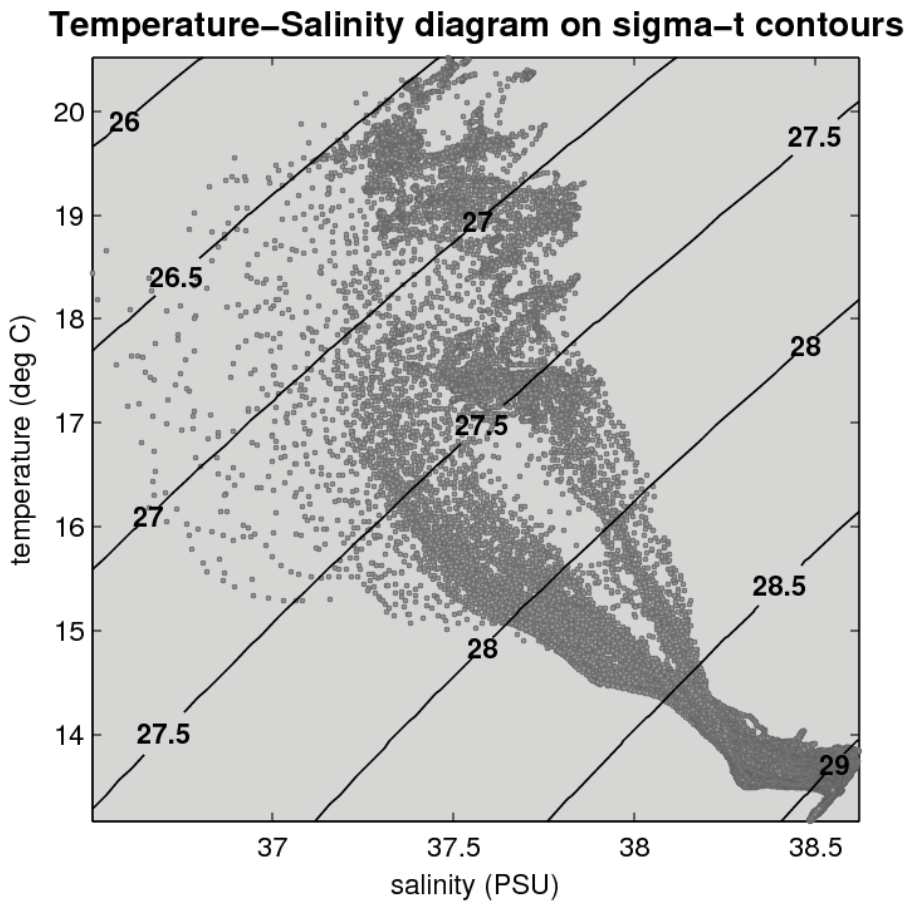

In situ oxygen concentration
depth (m)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
distance (km)
160
170
180
190
200
210
220
oxygen concentration (umol l−1)

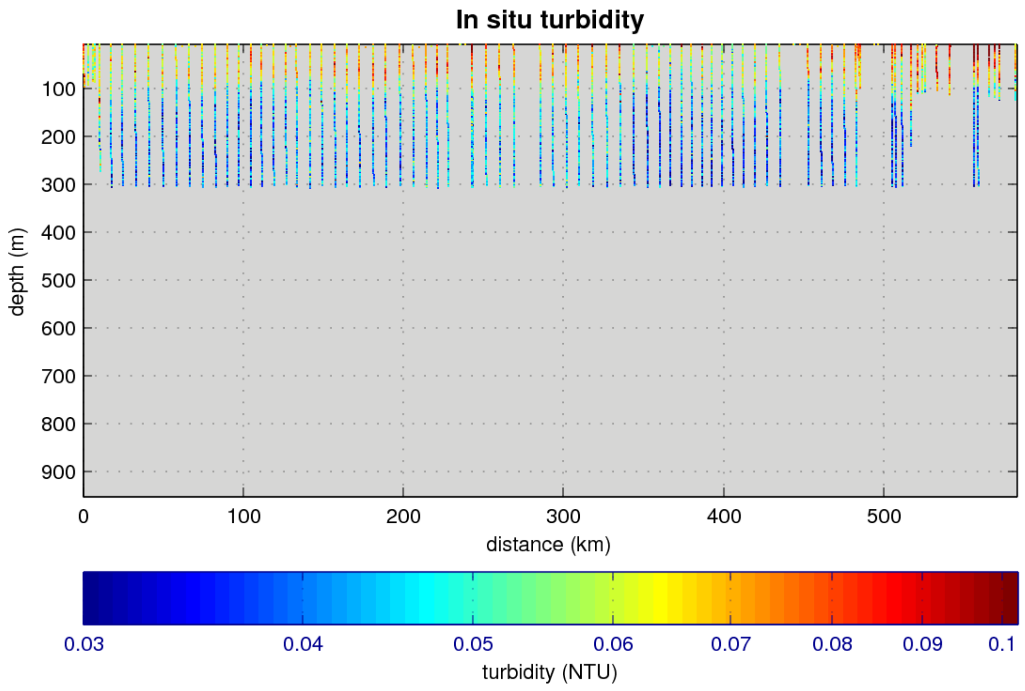

O2 concentration profiles
depth (m)
O2 concentration (umol l−1)
mean
std
O2 saturation profiles
O2 saturation (%)


In situ oxygen saturation
depth (m)
distance (km)
oxygen saturation (%)

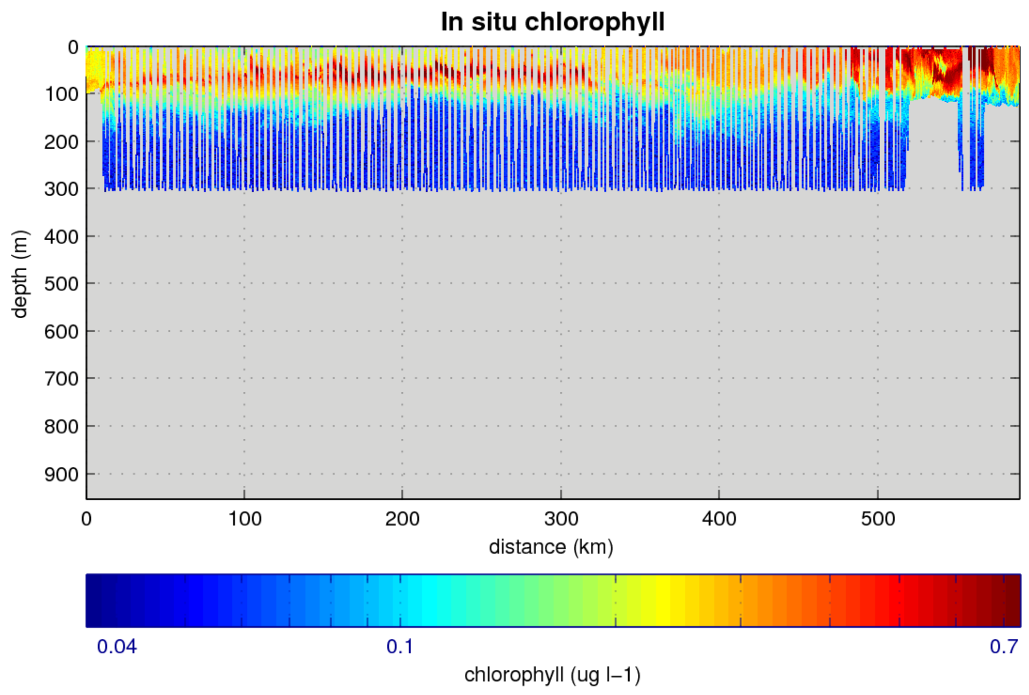

In situ salinity (raw)
depth (m)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
distance (km)
37.4
37.6
37.8
38
38.2
38.4
salinity (PSU)

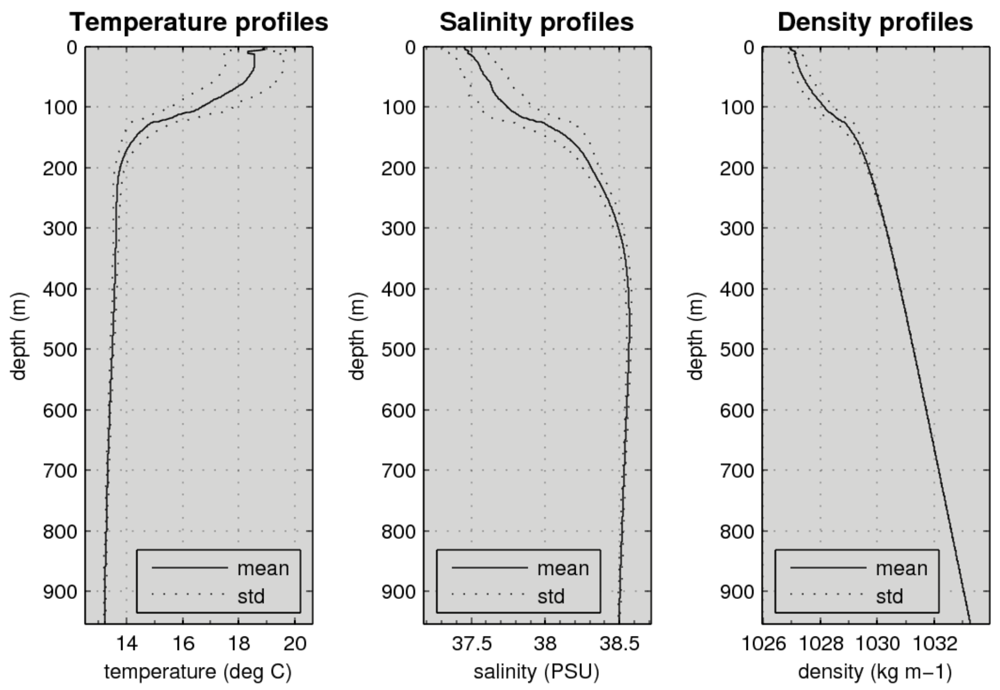

In situ temperature
depth (m)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
distance (km)
14
15
16
17
18
19
temperature (deg C)

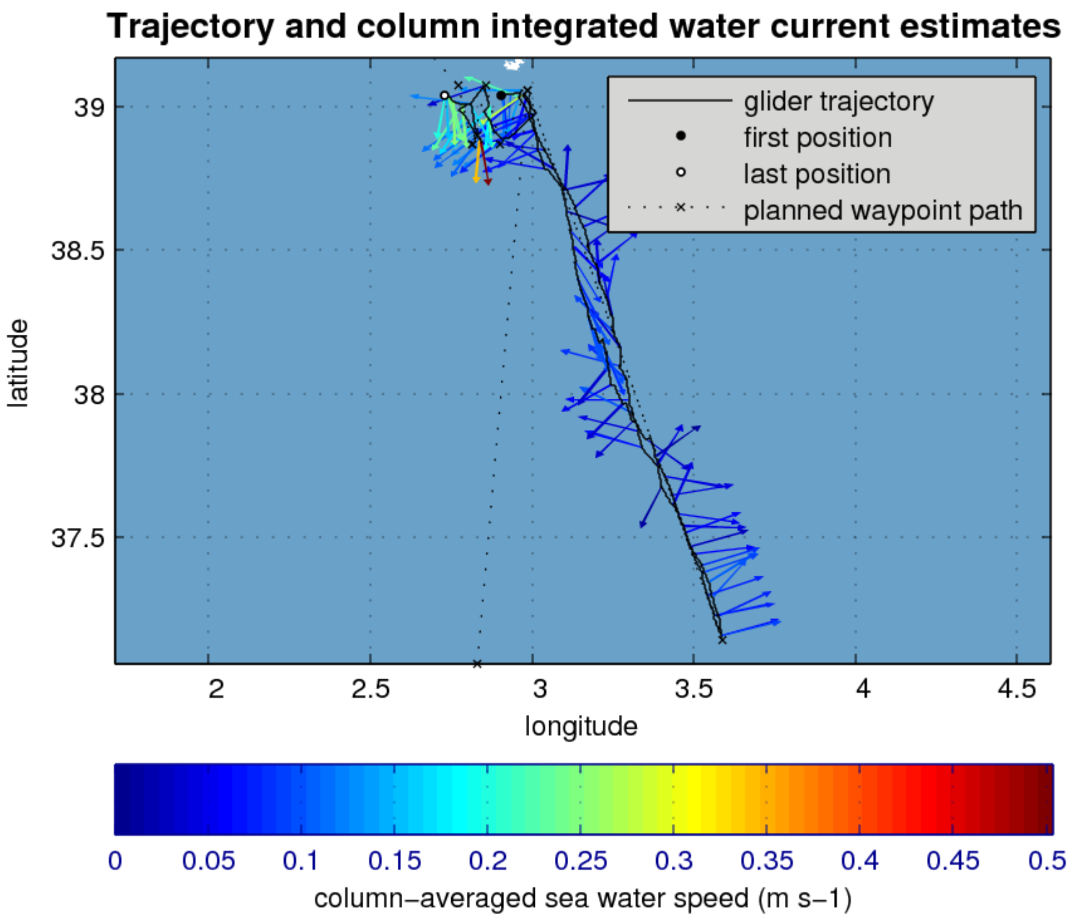

Temperature–Salinity diagram on sigma–t contours
temperature (deg C)
salinity (PSU)
20
19
18
17
16
15
14
37
37.5
38
38.5
26
26.5
27
27.5
28
28.5
29


In situ turbidity
depth (m)
distance (km)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
turbidity (NTU)

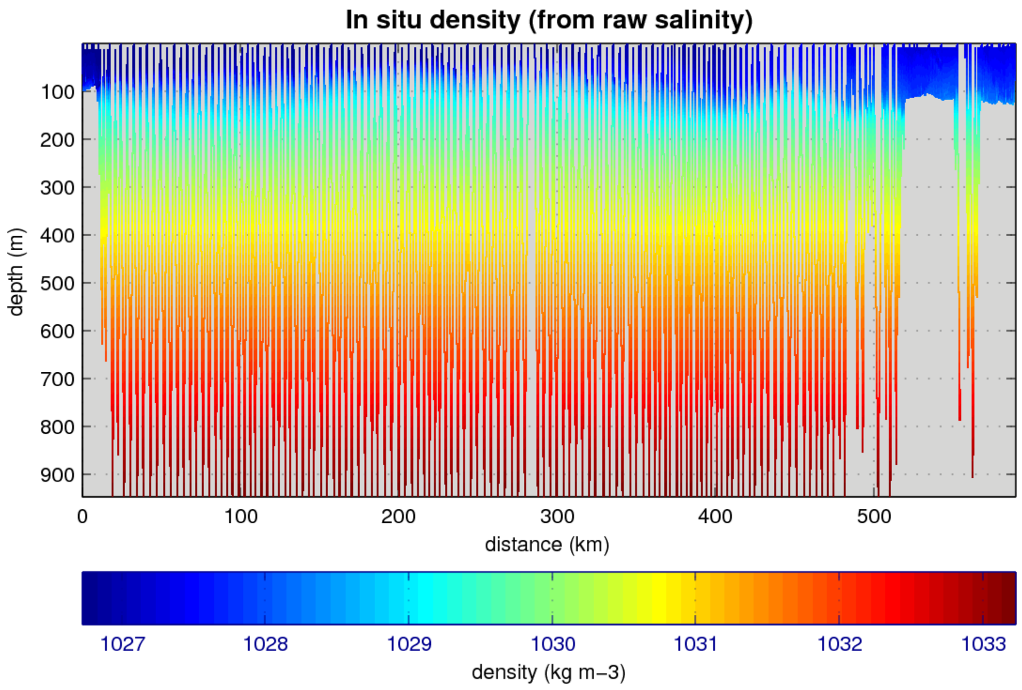
# 9. APPENDIX B: Delayed time figures



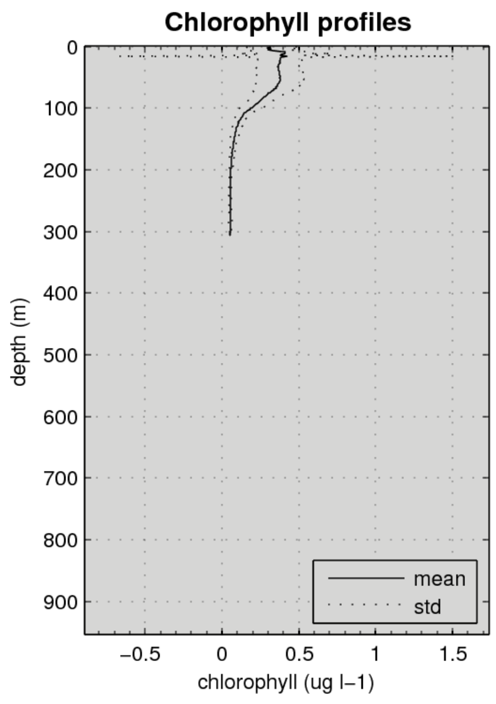


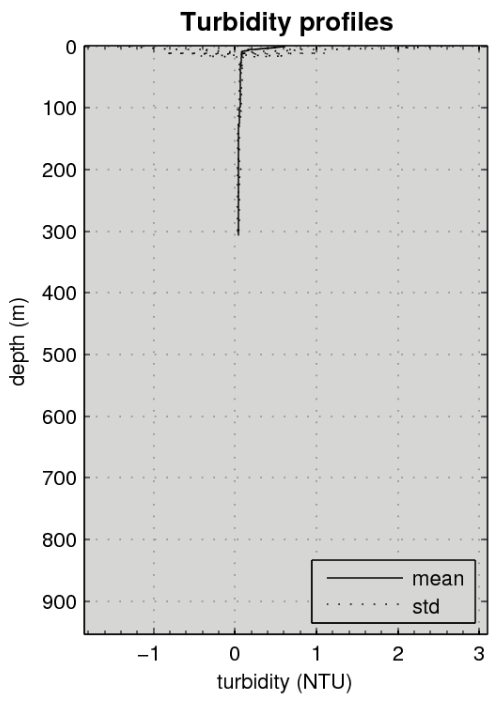


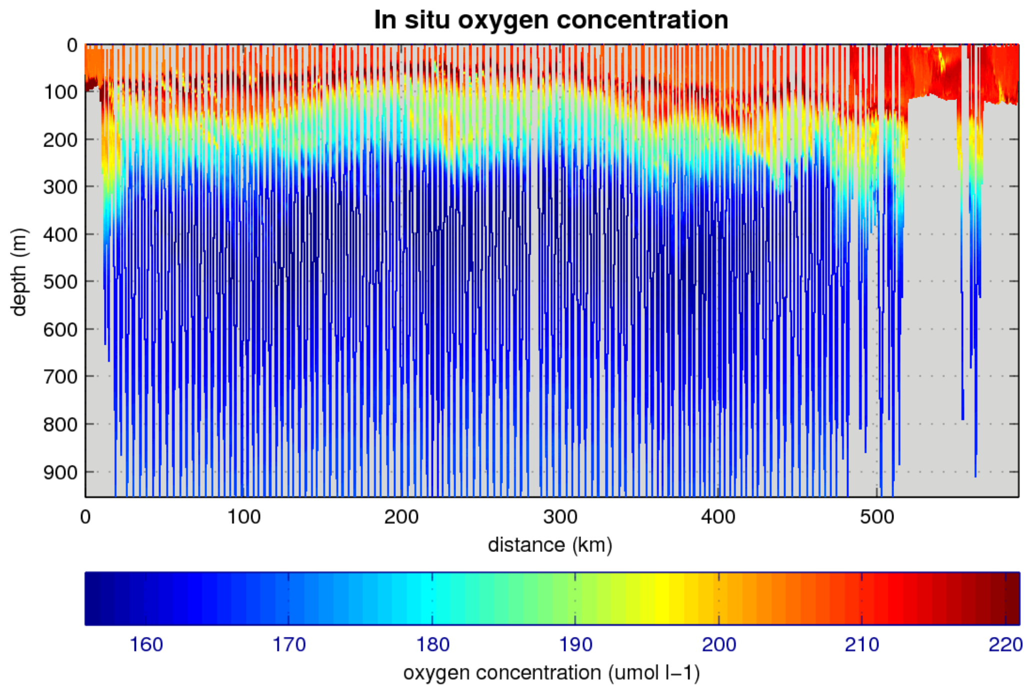
Trajectory and column integrated water current estimates
glider trajectory
first position
last position
planned waypoint path
latitude
39
38.5
38
37.5
longitude
2
2.5
3
3.5
4
4.5
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
column–averaged sea water speed (m s–1)

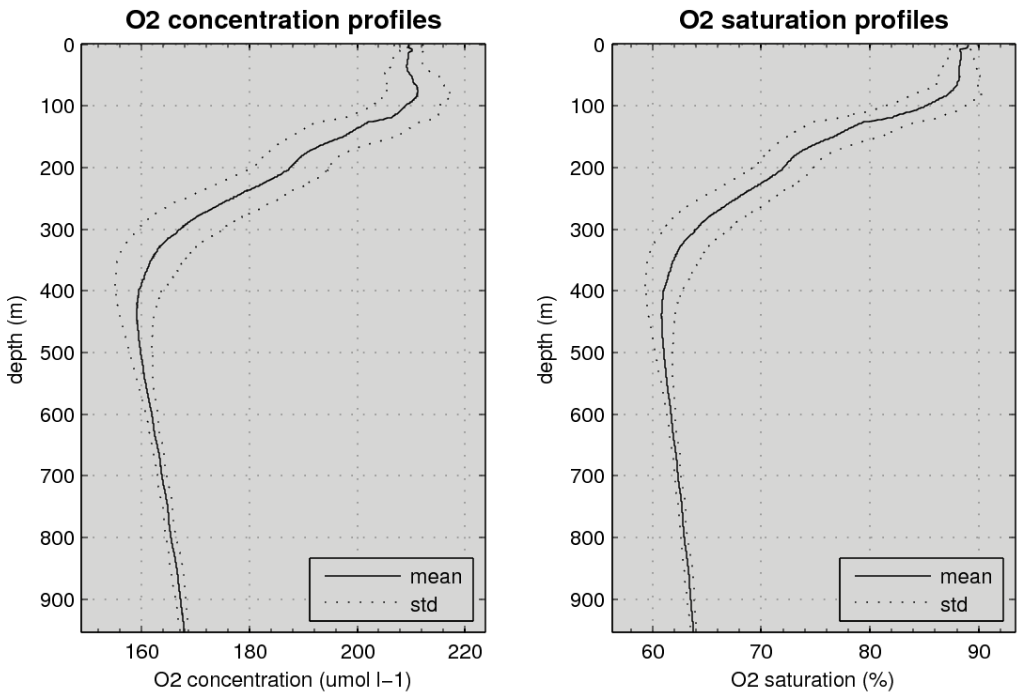


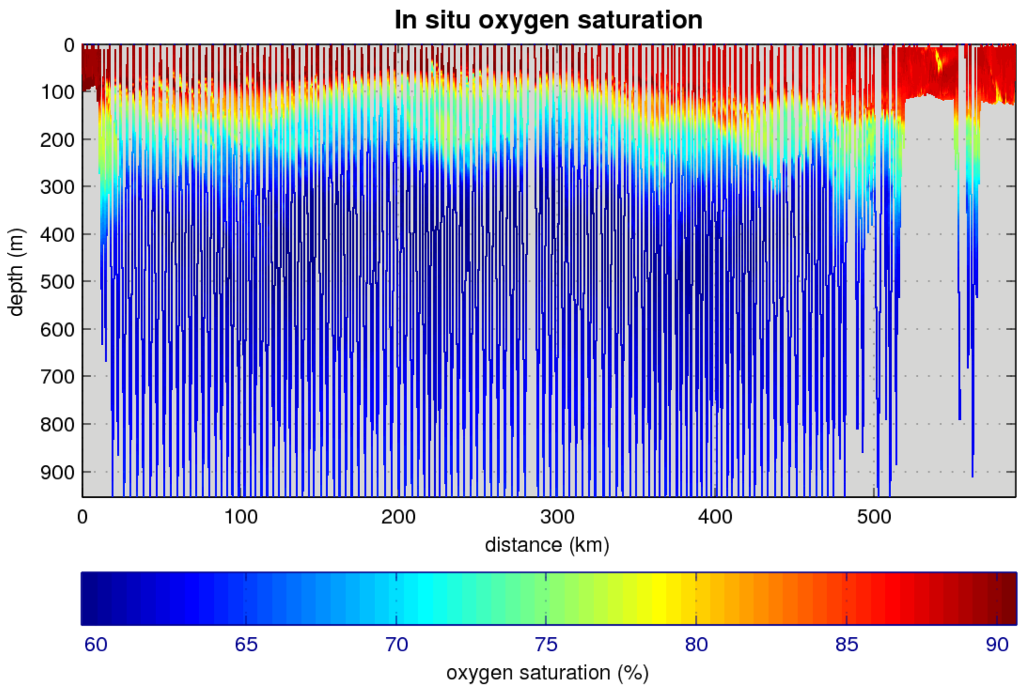
In situ density (from raw salinity)
depth (m)
100
200
300
400
500
600
700
800
900
distance (km)
0
100
200
300
400
500
1027
1028
1029
1030
1031
1032
1033
density (kg m–3)

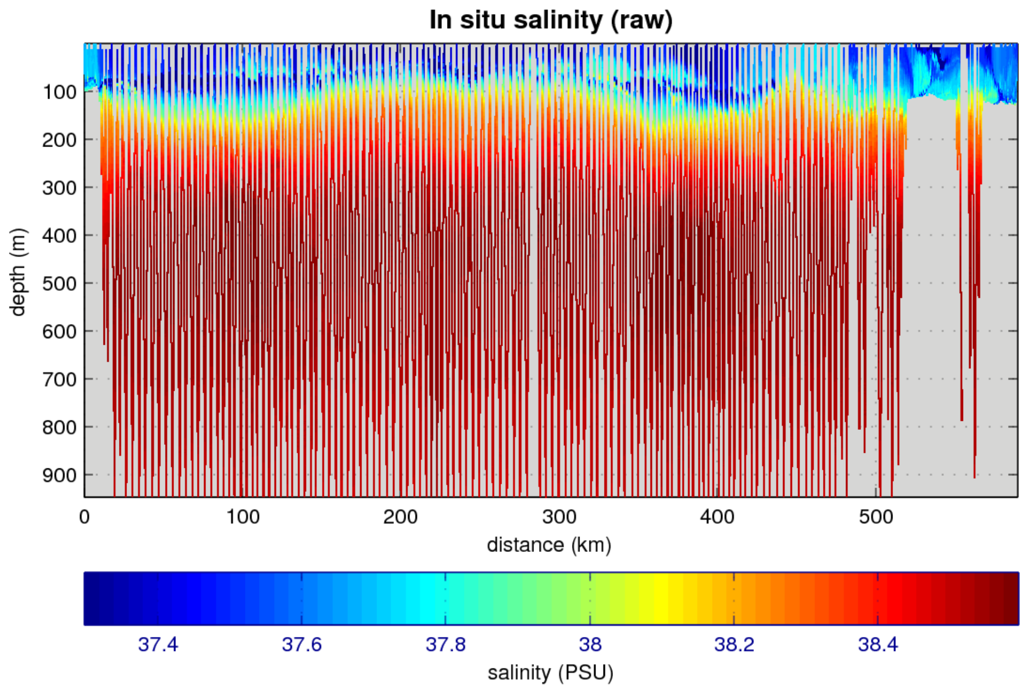

Chlorophyll profiles
depth (m)
chlorophyll (ug l−1)
mean
std

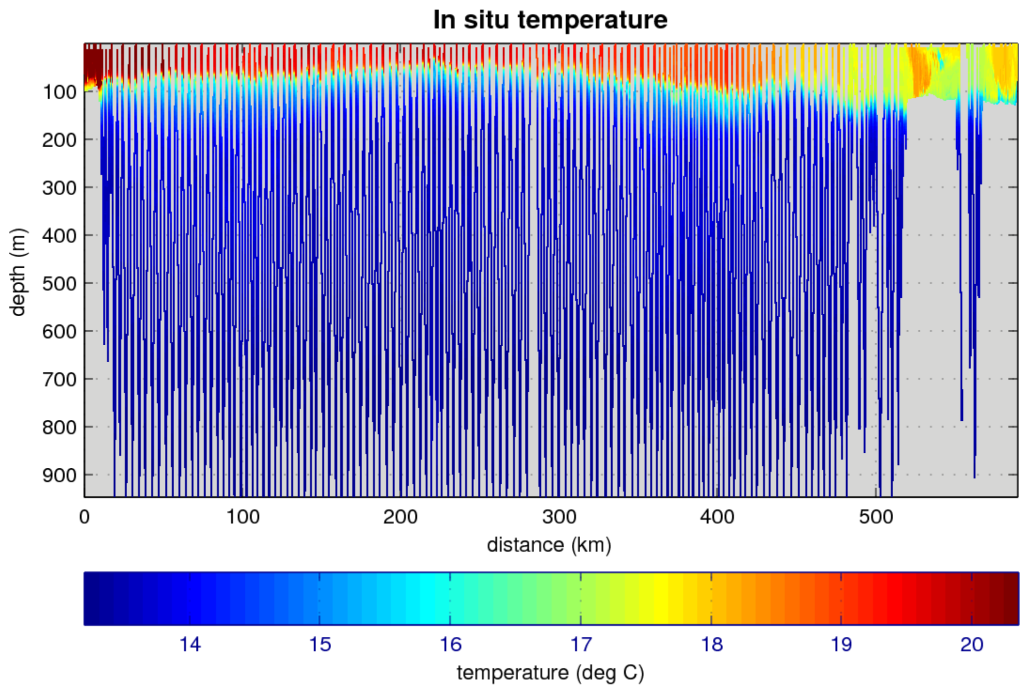

Turbidity profiles
depth (m)
turbidity (NTU)
mean
std

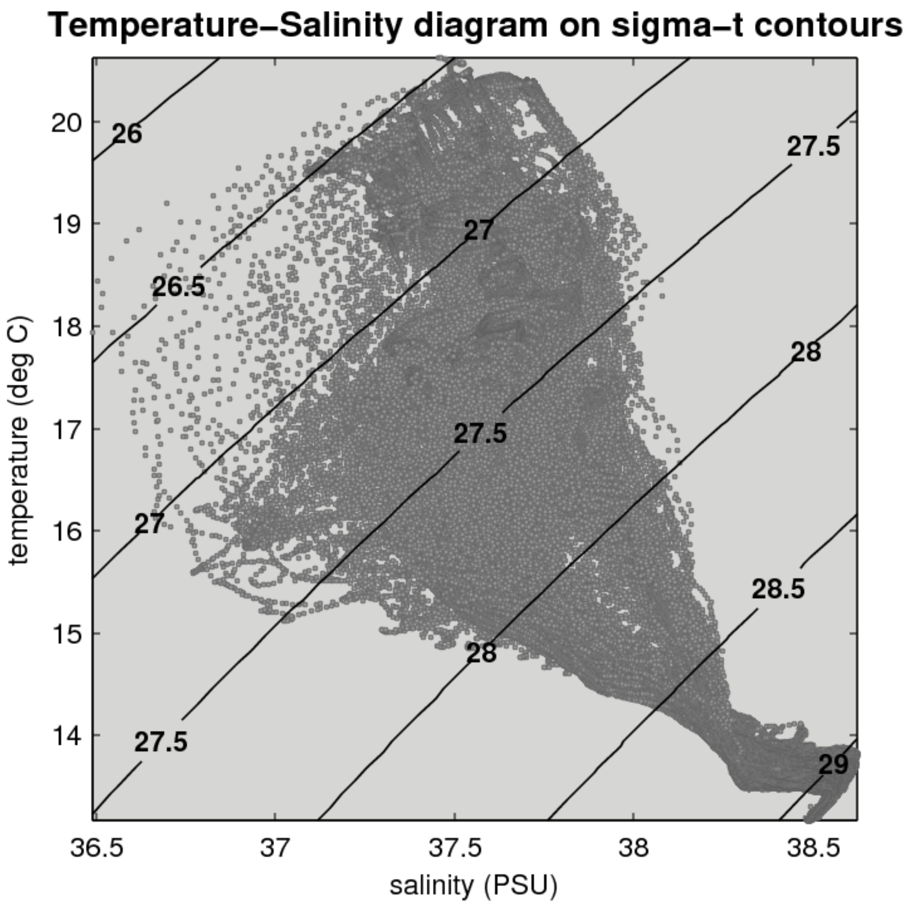

In situ oxygen concentration
depth (m)
distance (km)
oxygen concentration (umol l−1)

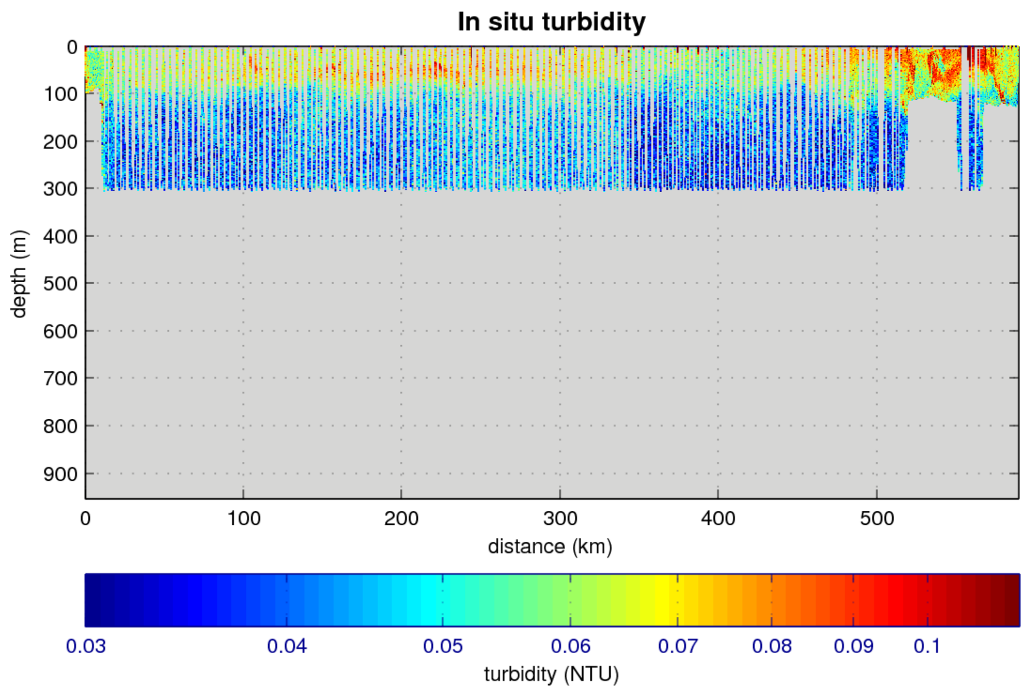

O2 concentration profiles
O2 saturation profiles
depth (m)
mean
std
O2 concentration (umol l−1)
O2 saturation (%)


In situ oxygen saturation
depth (m)
distance (km)
oxygen saturation (%)


In situ salinity (raw)
depth (m)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
distance (km)
37.4
37.6
37.8
38
38.2
38.4
salinity (PSU)


In situ temperature
depth (m)
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
distance (km)
14
15
16
17
18
19
20
temperature (deg C)


Temperature–Salinity diagram on sigma–t contours
temperature (deg C)
salinity (PSU)
20
19
18
17
16
15
14
36.5
37
37.5
38
38.5
26
26.5
27
27.5
28
28.5
29


In situ turbidity
depth (m)
distance (km)
turbidity (NTU)
0
100
200
300
400
500
600
700
800
900
0
100
200
300
400
500
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1