**Campagne**

**PIRATA-FR31**

**N/O THALASSA**

**16 avril 2021**

**J. Grelet - al**

**Instruction décrivant le Système d’Information, les méthodes d’acquisition et de traitement des données en mer et au laboratoire**

Sommaire

[1. Principe 4](#_Toc69595786)

[2. Description 4](#_Toc69595787)

[2.1. Cartographie du réseau sur le Thalassa: 4](#_Toc69595788)

[2.2. PC d’acquisition CTD sous Windows: 5](#_Toc69595789)

[2.2.1. Login 5](#_Toc69595790)

[2.2.2. Liste des logiciels installés : 5](#_Toc69595791)

[2.2.3. Stratégie de montage des partages Windows 5](#_Toc69595792)

[2.2.4. Montage des disques réseaux depuis un client Windows : 6](#_Toc69595793)

[2.2.5. Synchronisation automatique de l’heure : 6](#_Toc69595794)

[2.3. PC-Linux de collecte et de traitement des données 6](#_Toc69595795)

[2.3.1. Login US191-Linux-2: 6](#_Toc69595796)

[2.3.2. Adresse IP 6](#_Toc69595797)

[2.3.3. Client NTP 7](#_Toc69595798)

[2.3.4. Connections à distance 7](#_Toc69595799)

[2.3.5. Montage des disques réseaux depuis un client Linux : 7](#_Toc69595800)

[2.4. Création de l'arborescence du système d'information de la campagne 8](#_Toc69595801)

[2.5. Structure du dossier PIRATA-FR31 sous mission 8](#_Toc69595802)

[2.5.1. Le script d’initialisation pour bash : PIRATA-FR31/local/etc/skel : 9](#_Toc69595803)

[2.5.2. Configuration d’une session de traitement sous Linux 9](#_Toc69595804)

[2.5.3. Les utilitaires de traitement et sauvegardes : PIRATA-FR31/local/sbin 10](#_Toc69595805)

[2.5.4. Les répertoires sous data-raw (données brutes) 10](#_Toc69595806)

[2.5.5. Les répertoires sous data-processing (données traitées) 10](#_Toc69595807)

[2.5.6. Exemple pour les répertoires de donnée de la CTD 11](#_Toc69595808)

[2.5.7. Le répertoire CTD/ascii 12](#_Toc69595809)

[2.5.8. Le répertoire CTD/data 12](#_Toc69595810)

[2.5.9. Les répertoires DOCUMENTS: 13](#_Toc69595811)

[3. Acquisition et traitement des instruments scientifiques 15](#_Toc69595812)

[3.1. La Bathysonde CTD Seabird 15](#_Toc69595813)

[3.1.1. Description 15](#_Toc69595814)

[3.1.2. Configuration du PC d’acquisition CTD-CTD 16](#_Toc69595815)

[3.1.3. Traitements des données: 16](#_Toc69595816)

[3.1.4. Les alias de traitements: 20](#_Toc69595817)

[3.2. SADCP 25](#_Toc69595818)

[3.3. OSEA 25](#_Toc69595819)

[3.4. Hermes 26](#_Toc69595820)

[3.5. Sondeur scientifique EK80 et ER60 27](#_Toc69595821)

[4. Sauvegarde: 28](#_Toc69595822)

[5. Traitements automatisés 28](#_Toc69595823)

[5.1. Crontab 28](#_Toc69595824)

[5.2. Synchro.sh 29](#_Toc69595825)

[5.3. process-all.sh 31](#_Toc69595826)

[5.4. python-plots.sh 32](#_Toc69595827)

[5.5. Tracés de la route du TSG 35](#_Toc69595828)

[6. Traitements complémentaires 37](#_Toc69595829)

[6.1. Datagui: 37](#_Toc69595830)

[6.2. Tracés de la route (fichiers kml Google Earth) 38](#_Toc69595831)

[6.2.1. Copie des fichiers sur le serveur Nginx en local : 38](#_Toc69595832)

[6.2.2. Copie des fichiers sur le serveur web à l’ifremer : 39](#_Toc69595833)

[6.2.3. Lancement du script de création du fichier kml (web): 40](#_Toc69595834)

[6.2.4. Transfert des images sur le site web de l’Ifremer 42](#_Toc69595835)

[6.2.5. Lancement du script de création du fichier kml en local 44](#_Toc69595836)

[6.2.6. Fichier de configuration config.toml 44](#_Toc69595837)

[6.2.7. Tracé de la route et des opérations 45](#_Toc69595838)

[7. De retour au laboratoire 46](#_Toc69595839)

[8. Mise à disposition des scripts et documents 47](#_Toc69595840)

[9. Log-book 47](#_Toc69595841)

[10. Légendes 48](#_Toc69595842)

[11. Suivi des versions de ce document 48](#_Toc69595843)

# Principe

Lors d’une campagne océanographique, nous mettons en œuvre les instruments de mesure du laboratoire ou prêtés et utilisons ceux qui sont disponibles à bord. Les systèmes d’acquisition ainsi que les formats de données sont généralement hétérogènes. Afin de simplifier le traitement et la validation des données, nous avons mis en place ces dernières années, un système d’information permettant du structurer et hiérarchiser l’accès aux informations, système pouvant être adapté quel que soit le navire utilisé et dupliqué de retour à terre pour finaliser les traitements au laboratoire.

# Description

Pour la campagne en cours, ce document décrit en détail la structure du Système d’Information (SI) qui a été mise en place en début de mission, SI basé sur le dépôt Git CRDAP pour « Cruise Repository for Data Acquisition and Processing » disponible sur IRD GitLab : <https://git.outils-is.ird.fr/US191/CRDAP>.

Vous trouverez ci-dessous, la description du SI utilisé lors de la campagne PIRATA-FR31 sur le N/0 Thalassa, réalisée du 23 février au 19 avril 2021 à partir de Brest à Brest.

Les données de la CTD + LADCP sont enregistrées localement sur le PC d’acquisition puis copiées sur le disque réseau par le programme Matlab « ctddSeaProcessing »

Un mini PC Intel (NUC) sous Linux (Ubuntu 20.04 LTS) est utilisé pour récupérer et copier automatiquement plusieurs fois par jours les données acquises par les systèmes du bord. Les scripts de traitement sont lancés automatiquement plusieurs fois par jour afin de générer les fichiers de données résultants. 3 sauvegardes des données sont réalisées sur des disques durs indépendants par 3 PC différents.

## Cartographie du réseau sur le Thalassa:

Sur le Thalassa, le réseau local 192.168.x.x est subdivisé en sous réseau virtuel (vlan) 192.168.5x.x

**Vlan :**

* Exploitation : 192.168.51.x
* Acquisition : 192.168.52.x
* Sondeurs : 192.168.53.x
* Navigation : 192.168.54.x
* Vidéos : 192.168.55.x
* Engins : 192.168.56.x
* Cabines : 192.168.57.x
* Wifi : 192.168.58.x
* Thalassa : 134.246.11.1
* Passerelle : 192.168.x.202

**Adresses IP fixes:**

* tl-photopcs 192.168.51.100
* tl-techsas 192.168.52.1
* tl-osea     192.168.52.4
* tl-sippican     192.168.52.5
* tl-adcp38       192.168.52.7
* tl-adcp150      192.168.52.8
* tl-hermes      192.168.52.20
* tl-ferrybox     192.168.52.27
* tl-casino       192.168.52.30
* tl-casinolh     192.168.52.31
* tl-casinolp     192.168.52.32
* tl-ek80         192.168.53.1
* tl-ek60tvo      192.168.53.6
* tl-hermes       192.168.53.20
* pc-ctd 192.168.51.219
* us191-linux-2 192.168.51.217

## PC d’acquisition CTD sous Windows:

### Login

Après démarrage du PC d'acquisition,

login: ctd

passwd: us191imago

### Liste des logiciels installés :

* Seasave et Sbe-processing Seabird V7.26.7
* BBTalk et WinADCDP RDI
* Perl ActiveState, utilisé pour lancer manuellement les scripts de traitement de la CTD
* Matlab R2020b : traitement LADCP, pressure stability et traitement automatique des données CTD et LADCP
* Putty, Xming ou ssh sous WSL (Windows Sub-system for Linux) pour ouvrir une session sur le PC Linux-2 puis lancer une console Linux sous terminal X
* Meinberg 4.2.8p5 pour la synchronisation sur le serveur de temps NTP

### Stratégie de montage des partages Windows

Sur la Thalassa, tous les utilisateurs intègrent le groupe scientifique. La configuration du gestionnaire de partage réseau « samba » qui permet de monter les systèmes de fichiers Unix comme des partages SMB Windows, fait que tous les utilisateurs de ce groupe peuvent écrire sur le partage \\tl-nas\mission, mais que seul le propriétaire du fichier peut l’effacer. Le réglage par défaut de l’explorateur Windows n’affichant pas le propriétaire, la gestion des fichiers réseau devient vite difficile à gérer.

La solution adoptée lors des campagnes PIRATA est de monter sur tous les PC des utilisateurs le partage mission à partir de l’alias « \\tl-nas » sous le compte pirata, password thazard, soit \\tl-nas\mission

La deuxième solution est de lancer manuellement le script mount-thalassa.bat qui se trouve sous

[\\tl-nas\mission\PIRATA-FR31\local\sbin](file:///\\tl-nas\mission\PIRATA-FR31\local\sbin) après l’avoir copié sur le bureau du PC, voir paragraphe 2.2.4 ci-dessous.

### Montage des disques réseaux depuis un client Windows :

Les différents PC d’acquisition utiliseront le partage [\\tl-nas\mission](file:///\\tl-nas\mission) avec le compte pirata.

Pour effectuer un montage manuel, explorateur Windows, menu outils -> connecter un lecteur réseau :

* Lecteur M : ou S :
* Dossier [\\tl-nas\mission](file:///\\tl-nas\mission)
* Cocher la case « Se reconnecter à l’ouverture de session »
* Cocher la case « Se connecter à l’aide d’informations d’identification différentes »
* Utiliser le compte pirata, password thazard

Pour effectuer un montage automatique, copier sur le bureau du PC le fichier mount-thalassa.bat se trouvant sous \\tl-nas\mission\PIRATA-FR31\local\sbin, **puis exécuter ce fichier après chaque redémarrage du PC** afin de monter automatiquement les disques réseau.

Pour le montage des partages réseaux, nous utiliserons le compte **pirata**, password : **thazard**

Extrait du fichier :

net use M: \\tl-nas\mission thazard /USER:thalassa\pirata /PERSISTENT:YES

net use N: \\tl-nas\echanges thazard /USER:thalassa\pirata /PERSISTENT:YES

net use P: \\tl-nas\p thazard /USER:thalassa\pirata /PERSISTENT:YES

net use Y: \\tl-nas\data thazard /USER:thalassa\pirata /PERSISTENT:YES

net use Q: \\tl-nas\homedir thazard /USER:thalassa\pirata /PERSISTENT:YES

### Synchronisation automatique de l’heure :

Le serveur NTP se trouve à l’adresse IP : 192.168.51.13, DNS tl-aceb

Accéder au fichier de configuration du logiciel Meinberg :

Menu démarrer -> Tous les programmes -> Meinberg -> Network time protocol -> Edit NTP configuration,

puis rajouter la ligne suivante :

# End of generated ntp.conf --- Please edit this to suite your needs

pool tl-aceb

server 192.168.51.13 iburst

Re-démarrer le service ntp :

Menu démarrer -> Tous les programmes -> Meinberg -> Network time protocol -> Service control -> Restart NTP service.

## PC-Linux de collecte et de traitement des données

### Login US191-Linux-2:

Pour ouvrir une session sous Linux :

login: science

passwd: antea

### Adresse IP

* Hostname : US191-Linux-2
* OS : Ubuntu 20.04
* adresse IP lors de la campagne : 192.168.51.217

Attention : L’adresse IP est attribuée dynamiquement et peut changer d’une campagne à une autre. Le PC étant utilisé habituellement sans écran, via une connexion ssh, celle-ci ne peut aboutir si l’on ne connait pas son adresse IP !

Si tel est le cas, une fois le PC en route et branché sur le réseau, utiliser le logiciel nmap pour scanner son adresse IP sur le réseau. Sinon, plus simple, brancher un câble vidéo HDMI sur l’écran du PC Casino et démarrer une session graphique. Ouvrir un terminal et lancer la commande ifconfig et noter l’adresse IP :

[US191-Linux-2:science]~ > ifconfig eno1

eno1 Link encap:Ethernet HWaddr f4:4d:30:63:d9:3b

inet addr:192.168.51.217 Bcast:192.168.57.255

### Client NTP

Editer le fichier /etc/ntp.conf, mettre la ligne suivante:

> sudo vi /etc/ntp.conf

# thalassa

pool tl-aceb

server 192.168.51.13 iburst

puis redémarrer le service :

> sudo service ntp restart

### Connections à distance

Depuis le PC d’acquisition, alias sshl:

jgrelet@BRSCLTPO0062:/mnt/c/Users/jgrelet$

ssh -l science 192.168.51.217

science@192.168.51.217's password:

Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-136-generic x86\_64)

…

Last login: Fri Apr 16 14:35:39 2021 from 192.168.57.43

Trying to source /mnt/campagnes/PIRATA-FR31/local/etc/skel/.bashrc.PIRATA-FR31

source /mnt/campagnes/PIRATA-FR31/local/etc/skel/.bashrc.PIRATA-FR31

Ok...

Yes, seems good !!!

[US191-Linux-2:science]~

On peut également utiliser Putty associé avec Xming (serveur X) ou VNC (prise en main à distance).

### Montage des disques réseaux depuis un client Linux :

Dans le fichier /etc/fstab :

//tl-nas/mission /mnt/campagnes cifs username=pirata,password=thazard,uid=1002,gid=1002,iocharset=utf8,sec=ntlm,auto 0 0

//tl-nas/science /mnt/science cifs username=pirata,password=thazard,uid=1002,gid=1002,iocharset=utf8,sec=ntlm,auto 0 0

//tl-nas/data /mnt/data cifs username=pirata,password=thazard,uid=1002,gid=1002,iocharset=utf8,sec=ntlm,auto 0 0

//tl-nas\homedir /mnt/q cifs username=pirata,password=thazard,uid=1002,gid=1002,iocharset=utf8,sec=ntlm,auto 0 0

Vérification avec la commande df

[US191-Linux-2:science]~

> df -h

Filesystem Size Used Avail Use% Mounted on

udev 7.8G 0 7.8G 0% /dev

tmpfs 1.6G 2.4M 1.6G 1% /run

/dev/sda2 212G 55G 147G 27% /

/dev/sdb1 1.9T 1.6T 251G 87% /media/science/WD 2T

//tl-nas/mission 5.0T 286G 4.8T 6% /mnt/campagnes

//tl-nas/data 6.9T 4.0T 3.0T 58% /mnt/data

//tl-nas/echanges 300G 136G 165G 46% /mnt/echanges

//tl-nas\homedir 1.0T 595G 430G 59% /mnt/q

//tl-media\films 7.3T 4.6T 2.3T 67% /mnt/films

//tl-nas/sondeurs 20T 7.0T 14T 35% /mnt/sondeurs

Montage manuel:

$ sudo mount -t cifs -o username= <user>, password=<password, uid=<user>, gid=<group>, iocharset=utf8, sec=ntlm,auto //tl-nas/mission /mnt/campagnes

## Création de l'arborescence du système d'information de la campagne

Sous Linux, se placer dans le répertoire /mnt/campagnes, puis installer le dépôt CRDAP dans le sous répertoire PIRATA-FR31 avec la commande suivante :

$ git clone --recurse-submodules https://git.outils-is.ird.fr/US191/CRDAP.git PIRATA-FR31

Créer un lien symbolique sous la racine /m

$ cd /

$ sudo ln –s /mnt/campagnes /m

## Structure du dossier PIRATA-FR31 sous mission

* [\\tl-nas\mission](file:///\\tl-nas\mission): partage samba réservé aux scientifiques pour la campagne.
* Peut être monté également avec l’alias \\tl-nas\mission



### Le script d’initialisation pour bash : PIRATA-FR31/local/etc/skel :

* les scripts d'initialisation du bash
* les alias de traitements
* les fonctions bash pour automatiser les traitements

### Configuration d’une session de traitement sous Linux

Le principe de cette opération est d’obtenir dans son environnement shell l’ensemble des alias nécessaires aux traitements et à la mise en forme des données. De retour au laboratoire, pour travailler sur un disque dur externe ou sur un partage réseau (nfs ou samba), il suffit de modifier les variables $DRIVE et $CRUISE pour accéder de nouveau aux scripts et données.

Dans le fichier .bashrc, définir les variables d'environnement pour la campagne:

export DRIVE=/m

export CRUISE=PIRATA-FR31

Puis en fin de script, le fichier à « sourcer »:

# Source CRUISE definitions

if( -f ${DRIVE}/local/etc/skel/.bashrc.${CRUISE} ) then

source ${DRIVE}/local/etc/skel/. bashrc.${CRUISE}

endif

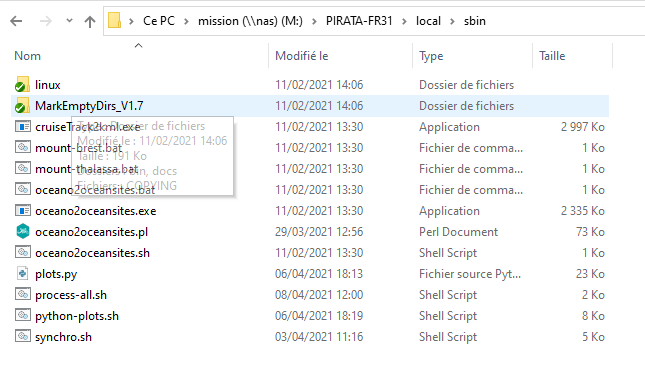
Pour passer de la campagne PIRATA-FR31 à PIRATA-FR30, il suffit de remplacer la déclaration de :

export CRUISE=PIRATA-FR31

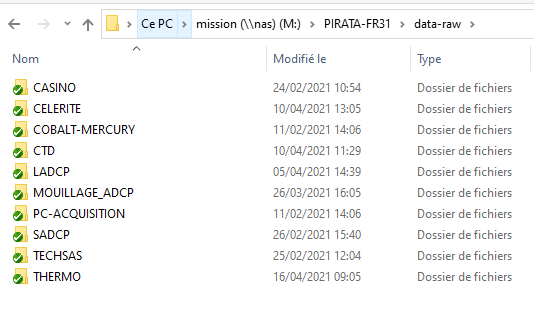
par :

export CRUISE=PIRATA-FR30

### Les utilitaires de traitement et sauvegardes : PIRATA-FR31/local/sbin



### Les répertoires sous data-raw (données brutes)



### Les répertoires sous data-processing (données traitées)



Structure du répertoire data-processing:

SADCP => ADCP de coque pour le traitement avec CASCADE ou CODAS

SOLEX => Données Météo, sondeur et chalut acquises par SOLEX

COLCOR => Les données du Thermosalinographe SBE21 transmises en temps réel

CELERITE => Profils des sondes Sippican XBT

CTD => Profils de la bathysonde

THERMO => Les données du Thermosalinographe SBE21 acquises avec Seasave

CASINO => Le journal de bord CASINO (pas utilisé lors de la campagne)

COBALT-MERCURY => Les données météo acquises par la station BATOS

CHIMIE => Les résultats des analyses

ctdSeaProcessing => le répertoire des scripts de traitement CTD et LADCP

### Exemple pour les répertoires de donnée de la CTD

Pour chaque répertoire, on trouve généralement l'arborescence suivante:

netcdf => contient les fichiers NetCDF issus des traitements

OS\_PIRATA-FR31\_CTD.nc => CTD capteurs primaires

OS\_PIRATA-FR31-ALL\_CTD.nc => CTD capteurs primaires et secondaires

OS\_PIRATA-FR31-ALL\_BTL.nc => Bouteilles capteurs primaires et secondaires

ascii => répertoire des fichiers d’entête et d’extraction des données ASCII

odv => répertoire de la collection ODV

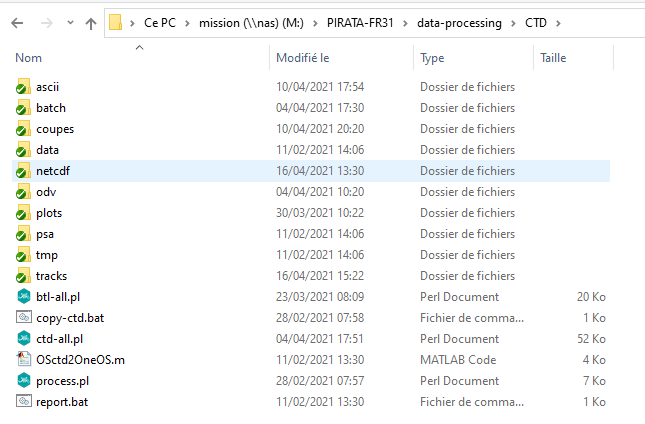
netcdf => répertoire des fichiers d’extraction Netcdf au format OceanSITES

coupes => les coupes ou sections

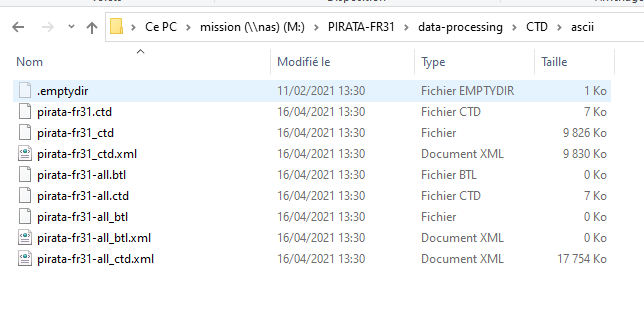
plots => les tracés réduits

tracks => la route du navire avec les stations CTD, fichier Google Earth KML

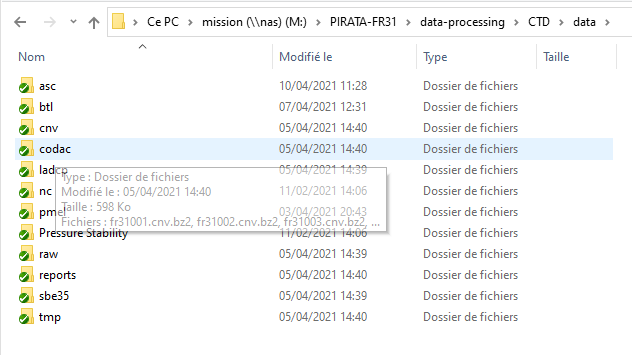
psa => les fichiers de configuration de SBE-dataprocessing



### Le répertoire CTD/ascii



### Le répertoire CTD/data



data une copie des données présentes dans data-raw, données qui peuvent être corrigées. Les fichiers sont classés par répertoires suivant les étapes du traitement. Le répertoire nc contient un fichier NetCDF par cast.

raw une copie des données présentes dans data-raw, données qui peuvent être corrigées. Les fichiers sont classés par répertoire suivant les étapes du traitement.

ladcp les données CTD acquises toutes les secondes pour le LADCP

codac les fichiers réduits envoyés à CORIOLIS

pmel les fichiers réduits envoyés au PMEL aux points de mouillages ATLAS

cnv les fichiers ascii, up (ucsp\*), down (dcsp\*) et up-down (csp\*)

btl les fichiers des prélèvements bouteilles

asc les fichiers d’entête et données profils descendant uniquement

nc le répertoire nc contient un fichier NetCDF par cast

reports les fichiers de configuration par station

sbe35 les fichiers du capteur de référence SBE35 acquis lors de la fermeture des bouteilles

Pour les autres instruments, nous retrouverons une structure quasiment identique.

### Les répertoires DOCUMENTS:

0 – COURRIER Les échanges de mails pour la préparation de la campagne

0 – DOSSIERS Les dossiers de préparation, rapports IRD et UMS, etc

0 – EQUIPEMENTS Les dossiers techniques

FORMULAIRES : Les documents pour les saisies manuscrites du déroulement des opérations

ENREGISTREMENTS : Les documents manuscrits du déroulement des opérations copies scannées ou feuilles Excel saisies

INSTRUCTIONS : Les documents décrivant les instructions et protocoles d'utilisation pour la physique et la chimie



Sous 0-Dossiers :



Et sous 0 – EQUIPEMENTS :



# Acquisition et traitement des instruments scientifiques

## La Bathysonde CTD Seabird

### Description

Pour la campagne PIRATA-FR31, nous avions à notre disposition 2 bathysondes SBE911+ de l’US191 IMAGO, S/N 1263 (S/N 1209 en rechange) ainsi qu’un carrousel (rosette de prélèvement) de 24 bouteilles de 8 litres chacune. La CTD était équipée d’un capteur de température primaire S/N 6083, d’un capteur de température secondaire S/N 6086, d’un capteur de température de référence SBE35 S/N 102, d’un capteur de conductivité S/N 4509, d’un capteur de conductivité secondaire S/N 4510, d’un capteur d’oxygène primaire S/N 3261, d’un capteur d’oxygène secondaire S/N 3265, d’un fluorimètre Wetlabs ECO-FL 4707, d’un fluorimètre Chelsea S/N 088-056 et d’un transmissiomètre Wetlabs CTS 1827R, (utilisation de la Deck Unit S/N 1050) .

Lors de cette campagne, comme lors des 3 campagnes précédentes, nous avons utilisé un capteur de précision SBE35, S/N 0102. Ce capteur est fixé sur le châssis de la bathysonde à proximité des capteurs SBE03 de la CTD SBE09. Il est relié à la CTD SBE09 et au carrousel SBE32 par un câble en Y. Ce capteur réalise une série de mesures à chaque déclenchement des bouteilles, soit 8 mesures toutes les 1.1 secondes dans la configuration standard. Il sert donc de référence pour suivre le bon fonctionnement des capteurs de température SBE03 de la CTD SBE09. Les données sont récupérées manuellement après chaque profil avec le logiciel SeaTerm et une colonne TE35 est rajoutée dans les fichiers finaux BTL (Ascii et Netcdf).

L’acquisition des données est réalisée depuis le PC IRD-US191-LS1 (HP 840G3) de l’US191 sous Windows 10 avec Seasave Version 7.26.7. Les données des LADCP sont déchargées localement sur le PC d’acquisition.

Le script Matlab « ctdSeaProcessing » permet de réaliser automatiquement l’ensemble de la copie des fichiers bruts sur le réseau puis le traitement des profils CTD et LADCP.

### Configuration du PC d’acquisition CTD-CTD

Liaison Deck-unit : USB/HPIB NI

En début de mission, nous avons eu un problème de driver avec le Multiplexeur USB 3.0  10 ports Startech, il a été remplacé par 2 multiplexeur 4 USB ABIX. Point à revoir au laboratoire avant la prochaine mission.

* COM7: Deck-unit
* COM5 : GPS NMEA en provenance de CINNA diffusion 2 (GGA à 9600 bds)
* COM3 : Modem carousel/SBE35
* COM6: Serial data out (répétiteur passerelle brassé sur prise RJ45)
* COM2 : LADCP Master
* COM4 : LADCP Slave

### Traitements des données:

#### Principe:

Actuellement, il existe un script Perl générique pour chaque type d’instrument. La configuration des scripts est décrite dans un fichier externe « config.ini » se trouvant dans le répertoire data-processing.

Ce fichier config.ini est à modifier en début de campagne afin de renseigner correctement les attributs globaux :

#### Les scripts Perl de traitement par répertoire:

Les traitements des stations CTD sont réalisés avec l’alias « pctd »

CTD -> pctd

Idem pour les données de célérité (XBT)

XBT -> pxbt

LADCP -> pladcp

TSG -> ptsg

…

L’alias « pall » permet de réaliser l’ensemble des traitements. Est lancé automatiquement 3 fois par jour depuis la crontab, après la synchronisation.

#### Le fichier de configuration générique « config.ini »

Ce fichier est à préparer, vérifier, voir modifier à chaque début de campagne en fonction de la configuration des instruments de mesures.

[global]

# with wsl, mount share as:

# sudo mount -t drvfs '\\tl-nas\mission' /mnt/m

author = jgrelet IRD march 2021 PIRATA-FR31 cruise

debug = 0

echo = 1

[cruise]

cycle\_mesure = PIRATA-FR31

plateforme = THALASSA

callsign = FNFP

imo = 9070307

mmsi = 227307000

context = AMMA

timezone = GMT

format\_date = DMY

processing\_code= 1A

begin\_date = 23/02/2021

end\_date = 19/04/2021

institute = IRD

pi = BOURLES

creator = Jacques.Grelet@ird.fr

[xml]

# dtd = local or public

dtd = public

dtdLocalPath = /m/PIRATA-FR31

#encoding = ISO-8859-1

encoding = UTF-8

[ctd]

cruisePrefix = fr31

# profile name : cspxxxyy

stationPrefixLength = 3

acquisitionSoftware = SEASAVE

acquisitionVersion = 7.22.3

processingSoftware = SBE-PROCESSING

processingVersion = 7.22.3

type = SBE911+

sn = 09P-1263

title\_summary = CTD profiles processed during PIRATA-FR31 cruise

comment = CTD profiles with pre-cruise coefficients, temperature, salinity and oxygen from primary and secondary sensors

split = ETDD,2,PRES,3,DEPH,4,TEMP,5,CNDC,7,TUR3,13,FLU2,14,FLU3,15,DOX2,16,PSAL,18,DENS,20,SVEL,22,NAVG,24

header = PRFL PRES DEPH ETDD TEMP PSAL DENS SVEL DOX2 FLU2 FLU3 TUR3 NAVG

format = %05d %6.1f %6.1f %10.6f %7.4f %7.4f %6.3f %7.2f %6.2f %6.3f %6.3f %7.4f %4d

odv\_hdr = PRES TEMP PSAL DENS SVEL DOX2 FLU2 FLU3 TUR3

odv\_unit = [db] [C] [Psu] [kg/m3] [m/s] [micromole/kg] [milligram/m3] [milligram/m3] [%]

[ctd-all]

split = ETDD,2,PRES,3,DEPH,4,TE01,5,TE02,6,CND1,7,CND2,8,DOV1,9,DOV2,10,DVT1,11,DVT2,12,TUR3,13,FLU2,14,FLU3,15,DO12,16,DO22,17,PSA1,18,PSA2,19,DEN1,20,DEN2,21,SVEL,22,NAVG,24

#split = ETDD,2,PRES,3,DEPH,4,TE01,5,TE02,6,CND1,7,CND2,8,DOV1,9,DOV2,10,DVT1,11,DVT2,12,FLU2,13,TUR3,14,DO12,15,DO22,16,NAVG,17,PSA1,18,PSA2,19,DEN1,20,DEN2,21,SVEL,22

header = PRFL PRES DEPH ETDD TE01 TE02 PSA1 PSA2 CND1 CND2 DEN1 DEN2 SVEL FLU2 FLU3 TUR3 DO12 DOV1 DVT1 DO22 DOV2 DVT2 NAVG

format = %05d %6.1f %6.1f %10.6f %7.4f %7.4f %7.4f %7.4f %7.5f %7.5f %6.3f %6.3f %7.2f %6.3f %6.3f %6.2f %7.3f %6.4f %+7.5f %7.7g %6.6g %+7.7g %4d

#format = %05d %4d %6.1f %10.6f %7.4f %7.4f %7.4f %7.4f %7.5f %7.5f %6.3f %6.3f %7.2f %6.3f %6.2f %7.3f %7.3f %6.4f %6.4f %+7.5f %+7.5f %4d

[btl]

type = SBE32 standard 24 Niskin bottles

sn = unknown

title\_summary = Water sample during PIRATA-FR31 cruise with 20 levels

comment = CTD bottles water sampling with temperature, salinity and oxygen from primary and secondary sensors

split = BOTL,1,month,2,day,3,year,4,PSA1,5,PSA2,6,DO11,7,DO12,9,DO21,8,DO22,10,Potemp090C,11,Potemp190C,12,ETDD,13,PRES,14,DEPH,15,TE01,16,TE02,17,CND1,18,CND2,19,DOV1,20,DOV2,21,DVT1,22,DVT2,23,TUR3,24,FLU2,25,FLU3,26

header = PRFL BOTL PRES DEPH ETDD TE35 TE01 TE02 PSA1 PSA2 CND1 CND2 DO11 DO12 DO21 DO22 DOV1 DOV2 DVT1 DVT2 FLU2 FLU3 TUR3

format = %05d %2d %6.1f %6.1f %10.6f %7.6g %7.4f %7.4f %7.4f %7.4f %7.5f %7.5f %7.4f %8.4f %7.3f %8.3f %8.5f %8.5f %+7.5f %+7.5f %+6.4f %+6.4f %6.2f

[xbt]

cruisePrefix = fr31

stationPrefixLength = 3

acquisitionSoftware = WinMK21

acquisitionVersion = 2.10.1

processingSoftware =

processingVersion =

type = SIPPICAN

sn = 01150

title\_summary = XBT profiles processed during PIRATA-FR31 cruise

comment = Extract from .edf files

[thermo]

cruisePrefix = fr31

stationPrefixLength = 3

acquisitionSoftware = COLCOR

acquisitionVersion =

processingSoftware =

processingVersion =

type = SBE21

sn = 3153

calDate = 23/01/2015

externalType = SBE3S

externalSn = 2546

externalCalDate = 23/01/2015

depth\_intake = 4

title\_summary = Thermosalinograph data acquired with Seabird SBE21 instrument and reduce with THECSAS/COLCOR

comment = Extract from .colcor files

[ladcp]

cruisePrefix = FR31

# profile name : cspxxxyy

stationPrefixLength = 3

acquisitionSoftware = BBTALK

acquisitionVersion = 3.04

processingSoftware = MATLAB-VISBECK

processingVersion = 10.16.2

type = WH150-WH300

sn = 23909-24584

title\_summary = IFM-GEOMAR/LDEO Matlab LADCP-Processing system adapted by FM/JG

comment = Extract from .lad files

[sadcp]

type = OS150 - OS38

snOS150 = 57568

sn0S38 = 57001

Les sections [ctd] [ctd-all], [btl] décrivent la structure des données à extraire dans les fichiers Seabird.

Chaque script de traitement va lire les données qui se trouvent dans le sous répertoire data et va les formater pour ensuite les sauvegarder aux formats suivants:

#### PROFILS (CTD)

ascii/PIRATA-FR31.ctd => entêtes des profils

ascii/PIRATA-FR31\_ctd => fichier ASCII, 2 lignes d'entête, matrice n ligne x m

colonnes, chaque entête de profil

est identifié par la profondeur -1 en valeur décimale

pour le jour julien, latitude et longitude

ascii/PIRATA-FR31\_ctd.xml => Même fichier avec un entête XML plus complet (méta-data)

odv/PIRATA-FR31\_ctd\_odv.txt => Fichier ASCII au format ODV (Ocean Data View)

netcdf/OS\_PIRATA-FR31\_CTD.nc => L'ensemble des profils au format NetCDF OceanSITES

Les fichiers « PIRATA-FR31-all » utilisés pour la calibrations de l’oxygène

Les fichiers « PIRATA-FR31.btl » avec extensions « btl » pour la calibrations de l’oxygène

Exemple d’alias pour réaliser les traitements CTD, voir paragraphe « LES ALIAS DE TRAITEMENT », ci dessous :

$ CTD

$ btl

$ btlnc

#### TRAJECTOIRES (METEO, VENT)

ascii/PIRATA-FR31.mto => l'ensemble des données au format JJ/MM/YY HH:MM:SS

DD°MMM.SSS E

ascii/PIRATA-FR31\_mto => le même fichier avec en valeur décimal le jour julien,

latitude et longitude

ascii/PIRATA-FR31\_mto.xml => Même fichier avec un entête XML plus complet

netcdf/OS\_PIRATA-FR31\_MTO.nc => L'ensemble des données au format NetCDF OceanSITES

ascii/PIRATA-FR31.tsg => l'ensemble des données au format JJ/MM/YY HH:MM:SS

DD°MMM.SSS E

ascii/PIRATA-FR31\_tsg => le même fichier avec en valeur décimal le jour julien,

latitude et longitude

ascii/PIRATA-FR31\_tsg.xml => Même fichier avec un entête XML plus complet

ascii/PIRATA-FR31\_tsgqc => Fichier ASCII utilisable avec TSG-QC

netcdf/OS\_PIRATA-FR31\_TSG.nc => L'ensemble des données au format NetCDF OceanSITES

Les fichiers peuvent ensuite être visualisés avec les logiciels Matlab R2012b ou datagui ou avec TSG-QC, disponibles sur les dépôts subversion suivants :

<https://git.outils-is.ird.fr/grelet/datagui> branche v1.0

https://git.outils-is.ird.fr/grelet/TSG-QC

### Les alias de traitements:

Le traitement peut être également réalisé à la demande en utilisant les alias suivants, regroupés pour mémoire, dans le fichier d'alias sous PIRATA-FR31/local/etc/skel/.bashrc.PIRATA-FR31.

# .bashrc.PIRATA-FR31

# script d'init de l'environnement sous bash pour cygwin ou linux

echo "source ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}"

# prompt du shell

export PS1='[\[\033[32m\]\h:\[\033[31m\]\u\[\033[00m\]]\[\033[35m\]\w\[\033[00m\]\n> '

# chemin d'acces aux donnes

export DATA=${DRIVE}/${CRUISE}

# chemin des scripts locaux

export LOCAL=${DRIVE}/${CRUISE}/local

# plot profiles and sections

export NC\_DIR=netcdf

export PROF\_DIR=plots/python

export SECT\_DIR=coupes/python

# rajoute les chemins des scripts et du repertoire courant

export PATH=$PATH:${LOCAL}/sbin:/usr/local/netcdf-3.6.2/bin:.

#if [ "$(expr substr $(uname -s) 1 5)" == "Linux" ]; then

#fi

export ROSCOP\_CSV=$DRIVE/$CRUISE/local/code\_roscop.csv

export OCEANO2OCEANSITES\_INI=$DRIVE/$CRUISE/data-processing/pirata-fr31.ini

export OCEANO2OCEANSITES\_PL='oceano2oceansites.pl'

export IMG2HTML='img2html.rb'

# for go program

# alias

alias ssht='ssh -l science thalassa'

alias sshl='ssh -l science 192.168.51.217' # pc traitement sous Linux

# alias to list functions (typeset -F or declare -F)

alias functions='typeset -F'

# alias des commandes

alias ncdump='ncdump -p5'

alias matlab='matlab -nodesktop -nosplash'

# alias des repertoires pour PIRATA-FR31

alias DATAP='cd ${DATA}/data-processing'

alias DATAR='cd ${DATA}/data-raw'

alias FR31='cd ${DATA}'

#alias CTD='cd /c/seasoft/${CRUISE}'

alias CTD='cd ${DATA}/data-processing/CTD'

alias ADJ='cd ${DATA}/data-ajustage/CTD'

alias CODAC='cd ${DATA}/data-processing/CTD/codac'

alias BTL=CTD

alias XBT='cd ${DATA}/data-processing/CELERITE'

alias THERMO='cd ${DATA}/data-processing/THERMO'

alias TSG=THERMO

alias CASINO='cd ${DATA}/data-processing/CASINO'

alias BATOS='cd ${DATA}/data-processing/BATOS'

alias COLCOR='cd ${DATA}/data-processing/COLCOR'

alias TECHSAS='cd ${DATA}/data-processing/TECHSAS'

alias LADCP='cd ${DATA}/data-processing/LADCP'

alias LDEO='cd ${DATA}/data-processing/LADCP/v10.16.2/${CRUISE}'

alias MODELS='cd ${DATA}/data-processing/MODELS'

alias CASCADE='cd ${DATA}/data-processing/SADCP'

alias OS150='cd ${DATA}/data-processing/SADCP/OS150'

alias OS38='cd ${DATA}/data-processing/SADCP/OS38'

alias SADCP=CASCADE

alias CODAS='cd ${DATA}/data-processing/SADCP/CODAS'

alias RADIOSONDAGE='cd ${DATA}/data-processing/RADIOSONDAGE'

alias RSM=RADIOSONDAGE

alias plog='cat ${DRIVE}/${CRUISE}/local/logs/process.log'

alias viplog='gvim ${DRIVE}/${CRUISE}/local/logs/process.log'

alias slog='cat ${DRIVE}/${CRUISE}/local/logs/synchro.log'

alias vislog='gvim ${DRIVE}/${CRUISE}/local/logs/synchro.log'

# CTD avec tous les capteurs primaires et l'option --top

alias ctd='perl ctd-all.pl --cycle\_mesure=${CRUISE} --institut=IRD --plateforme="THALASSA" --sn=09P-1263 --type=SBE911+ --pi=BOURLES --begin\_date=23/02/2021 --end\_date=20/04/2021 data/asc/fr310??.hdr --echo --dtd=local --top --all'

alias ctdnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd ascii/pirata-fr31\_ctd.xml --output=netcdf/OS\_${CRUISE}\_CTD.nc'

# CTD avec tous les capteurs primaires et secondaires

alias ctdall='perl ctd-all.pl --cycle\_mesure=${CRUISE} --institut=IRD --plateforme="THALASSA" --sn=09P-1263 --type=SBE911+ --pi=BOURLES--begin\_date=23/02/2021 --end\_date=20/04/2021 data/asc/fr310??.hdr --echo --dtd=local --xml --ctd\_all'

alias ctdallnc='$OCEANO2OCEANSITES\_PL --echo --nodtd --short ascii/pirata-fr31-all\_ctd.xml --output=netcdf/OS\_${CRUISE}-ALL\_CTD.nc'

alias ctdgo='oceano2oceansites -e --config=../pirata-fr31.toml --files=data/cnv/fr31\*.cnv'

alias ctdgoall='oceano2oceansites -e -a --config=../pirata-fr31.toml --files=data/cnv/fr31\*.cnv'

# bouteilles

alias btl='perl btl-all.pl --echo --dtd=local data/btl/fr31\*.btl'

alias btlnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd ascii/pirata-fr31-all\_btl.xml --output=netcdf/OS\_${CRUISE}-ALL\_BTL.nc'

# XBT (CELERITE)

alias xbt='perl xbt-edf.pl --cycle\_mesure=${CRUISE} --institute=IRD --plateforme=THALASSA --sn=unknown --type=SIPPICAN --pi=BOURLES --begin\_date=23/02/2021 --end\_date=20/04/2021 --echo --dtd=local data/\*.edf --all'

alias xbtnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd ascii/pirata-fr31\_xbt.xml --output=netcdf/OS\_${CRUISE}\_XBT.nc'

# LADCP

alias ladcp='perl ldeo-ladcp.pl --cycle\_mesure=${CRUISE} --institute=IRD --plateforme=THALASSA --sn=12817 --type=WH300 --pi=BOURLES --begin\_date=23/02/2021 --end\_date=20/04/2021 --echo --dtd=local profiles/FR31\*.lad --ascii --xml'

alias ladcpnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd ascii/pirata-fr31\_adcp.xml --output=netcdf/OS\_${CRUISE}\_ADCP.nc'

alias ladcpodv='perl all\_ldeo-ladcp.pl --cycle\_mesure=${CRUISE} --institute=IRD --plateforme="THALASSA" --type=WH300 --sn=12818 --pi=BOURLES --begin\_date=23/02/2021 --end\_date=20/04/2021 profiles/\*.lad --echo --local --all'

# SADCP (traite les fichiers de nav \*.NR2 contenant les trames CADCP)

# se placer sous SADCP/nav pour lancer les scripts

alias cadcp2all='perl cadcp2all.pl ../data/\*.N2R'

alias cadcp2ang='perl cadcp2ang.pl'

alias cpsadcpraw38='\cp -rupv /z/Mission\_Courante/EQUIPEMENTS/OS38/DONNEES/\* /m/${CRUISE}/data-raw/SADCP/OS38'

alias cpsadcpraw150='\cp -rupv /z/Mission\_Courante/EQUIPEMENTS/OS150/DONNEES/\* /m/${CRUISE}/data-raw/SADCP/OS150'

alias cpsadcp38='\cp -rupv /z/Mission\_Courante/EQUIPEMENTS/OS38/DONNEES/\*.LTA /z/Mission\_Courante/EQUIPEMENTS/OS38/DONNEES/\*.STA /m/${CRUISE}/data-processing/SADCP/CASCADE/OS38/data'

alias cpsadcp150='\cp -rupv /z/Mission\_Courante/EQUIPEMENTS/OS150/DONNEES/\*.LTA /z/Mission\_Courante/EQUIPEMENTS/OS150/DONNEES/\*.STA /m/${CRUISE}/data-processing/SADCP/CASCADE/OS150/data'

# THERMO et BATOS (sans le VENT) extrait des fichiers temps reels Colcor

alias thermo='perl thermo-colcor.pl --echo --local --all data/\*.COLCOR'

alias thermonc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd ascii/pirata-fr31\_tsg.xml --output=netcdf/OS\_${CRUISE}\_TSG.nc'

alias tsg='thermo'

alias tsgnc='thermonc'

alias ctd-tsg='perl ctd-tsg-spl.pl ../CTD/data/asc/fr31???.hdr --echo'

alias meteonc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd pirata-fr31\_mto.xml'

alias mto=meteo

alias mtonc=meteonc

# CASINO extrait les donnees TSG et METEO des fichiers csv

#alias casino='perl casino.pl --cycle\_mesure=${CRUISE} --institute=IRD --plateforme="THALASSA" --pi=BOURLES --begin\_date=23/02/2021 --end\_date=21/04/2021 --echo --local data/\*.csv --all'

alias casino='perl casino.pl --cycle\_mesure=${CRUISE} --institute=IRD --plateforme=THALASSA --pi=BOURLES --date\_debut=23/02/2021 --date\_fin=20/04/2021 --echo --local data/2021\*.csv --all'

alias casinonc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd --data\_type=trajectory ascii/pirata-fr31\_mto --output=netcdf/OS\_${CRUISE}\_MTO.nc'

alias casinosndnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd --data\_type=trajectory ascii/pirata-fr31\_snd --output=netcdf/OS\_${CRUISE}\_SND.nc'

alias casinotsgnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd --data\_type=trajectory ascii/pirata-fr31\_tsg --output=netcdf/OS\_${CRUISE}\_TSG.nc'

alias casinofboxnc='$OCEANO2OCEANSITES\_PL --echo --short --nodtd --data\_type=trajectory ascii/pirata-fr31\_fbox --output=netcdf/OS\_${CRUISE}\_FBOX.nc'

# BATOS

alias batos='perl batos-pirata-fr31.pl --echo --local --ascii --xml'

alias batosnc='$OCEANO2OCEANSITES\_PL --short --nodtd --echo ascii/pirata-fr31\_mto.xml --output=netcdf/OS\_${CRUISE}\_MTO.nc'

# procédure de backup sur disque externe iomega

alias backup='sh /m/${CRUISE}/local/sbin/backup\_pirata-fr31.sh'

# gestion HTML des images DD

alias msgcol='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/MSGCOL -t "${CRUISE} - Meteosat couleur" -n 1'

#alias msgcola='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/MSGCOL/ATLANTIQUE -t "ATLANTIQUE - Meteosat couleur" -n 1'

alias sstamse='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/SSTAMSE -t "${CRUISE} - SST AMSE" -n 1'

alias sstmetop='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/SSTMETOP -t "${CRUISE} - SST METOP" -n 1'

alias windcdc='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/WINDCDC -t "${CRUISE} - WindCDC" -n 2'

alias windws='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/WINDWS -t "${CRUISE} - Windsat RSS" -n 2'

alias iwvamsr='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/IWVAMSR -t "${CRUISE} - IW ASR2 " -n 1'

alias wspamsr='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/WSPAMSR -t "${CRUISE} - Wind speed ASR2" -n 1'

alias sstostia='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/SSTOSTIA -t "${CRUISE} - SST OSTIA" -n 2'

alias windascat='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/WINDASCAT -t "${CRUISE} - Wind ASCAT" -n 2'

alias asstreynolds='$IMG2HTML -d /m/${CRUISE}/data-processing/PRODUCTS/ASSTREYNOLDS -t "${CRUISE} - SST anomaly Reynolds" -n 2'

alias mercator1='$IMG2HTML -d /m/${CRUISE}/data-processing/MERCATOR\_MODEL/SST -t "${CRUISE} - Mercator SST" -n 2'

alias mercator2='$IMG2HTML -d /m/${CRUISE}/data-processing/MERCATOR\_MODEL/SSS -t "${CRUISE} - Mercator SSS" -n 2'

alias mercator3='$IMG2HTML -d /m/${CRUISE}/data-processing/MERCATOR\_MODEL/COURANTS -t "${CRUISE} - Mercator Surface current" -n 2'

#alias mercator4='$IMG2HTML -d /m/${CRUISE}/data-processing/MERCATOR/PSY4/SST -t "${CRUISE} - Mercator PSY4 SST" -n 2'

#alias mercator5='$IMG2HTML -d /m/${CRUISE}/data-processing/MERCATOR/PSY4/SSS -t "${CRUISE} - Mercator PSY4 SSS" -n 2'

#alias mercator6='$IMG2HTML -d /m/${CRUISE}/data-processing/MERCATOR/PSY4/SURFACE\_CURRENT -t "${CRUISE} - Mercator PSY4 Surface current" -n 2'

# ces alias peuvent etre executes en une fois avec la fonction products

alias models=products

alias cpm='\cp -rupv /m/${CRUISE}/data-processing/PRODUCTS /z/${CRUISE}'

# alias de synchronisation des données bord vers mission

alias synchro='sudo -u science bash /mnt/campagnes/${CRUISE}/local/sbin/synchro.sh'

# alias des traitement mission realisees via la crontab, on peut egalement

# utiliser la fonction bash 'pall'

alias process='sudo -u science bash /mnt/campagnes/${CRUISE}/local/sbin/process-all.sh'

# alias pour generer les fichiers globaux SADCP LTA et STA

alias vmdas='sh vmdas.sh'

# fonctions utilitaires

function tidy

{

perltidy -i=2 -bt=2 -b $1

}

function cruisetrk

{

CTD

cd tracks

$LOCAL/sbin/linux/cruiseTrack2kml-linux-amd64 -config local.toml -output pirata-fr31-local.kml

$LOCAL/sbin/linux/cruiseTrack2kml-linux-amd64 -config config.toml -output pirata-fr31.kml

}

# les fonctions pour les traitements globaux

# list of functions: declare -F or typeset -F

function pctd

{

echo ""

echo "CTD processing:"

echo "---------------"

CTD

ctd

ctdnc

echo "BTL processing:"

echo "---------------"

btl

btlnc

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc -t CTD -p -k PRES TEMP PSAL DOX2 FLU2 -g -c k- b- r- m- g- -g -o $PROF\_DIR

}

function ptsg

{

echo ""

echo "TSG processing:"

echo "---------------"

TSG

tsg

tsgnc

ctd-tsg

}

function pxbt

{

echo ""

echo "XBT processing:"

echo "---------------"

XBT

xbt

xbtnc

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc -t XBT -p -k DEPTH TEMP DENS SVEL -c k- b- k- g- -g -o plots

}

function pladcp

{

echo ""

echo "LADCP processing:"

echo "-----------------"

LADCP

ladcp

ladcpnc

plots.py $NC\_DIR/OS\_${CRUISE}\_ADCP.nc -t ADCP -p -k DEPTH EWCT NSCT -c k- r- b- -g -o $PROF\_DIR

}

function pcasino

{

echo ""

echo "CASINO processing:"

echo "------------------"

CASINO

casino

casinonc

casinosndnc

casinotsgnc

casinofboxnc

}

function products

{

echo ""

echo "Products processing:"

echo "--------------------"

msgcol

sstamse

sstmetop

windcdc

windws

iwvamsr

wspamsr

sstostia

windascat

mercator1

mercator2

mercator3

}

function pall

{

pctd

pxbt

pladcp

ptsg

pcasino

#products

cruisetrk

}

echo "Ok..."

## SADCP

.

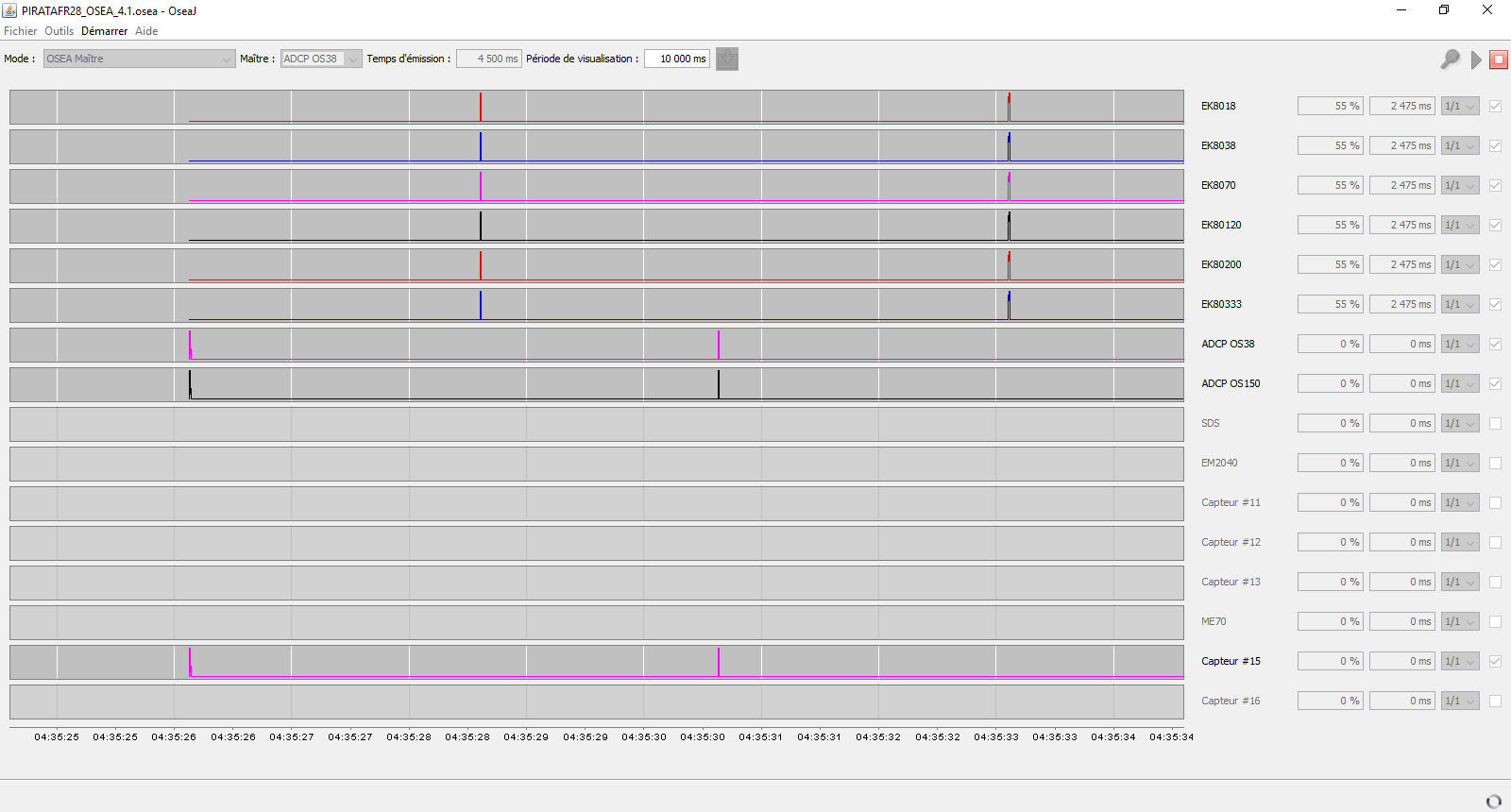
## OSEA

On utilise la configuration définie au cours de la campagne PIRATA-FR31.

La période est de 4500 ms. L’ADCP est maitre. Les sondeurs ping au bout de 2500 ms.

Lors de la campagne FR27, la période était de 3000 ms, avec ping des sondeurs au bout de 1500 ms. Depuis le montage des ADCP sous la gondole, la portée est passée de 1000 m à 1500 m, il faut donc laisser au minimum 2 secondes pour l’ADCP avant de démarrer les sondeurs. En début de mission, l’ADCP était systématiquement parasité par les sondeurs à partir de 1000m.

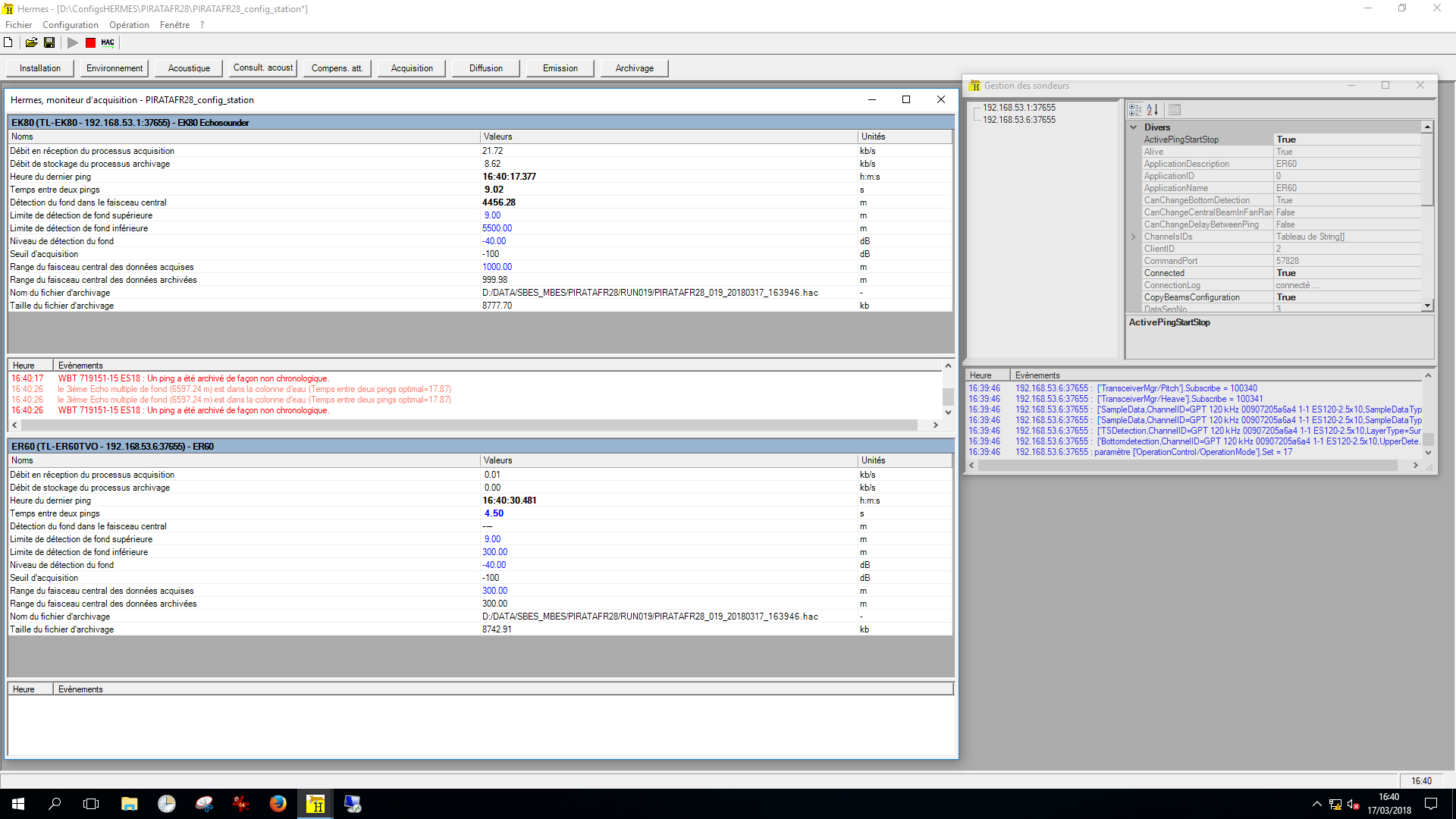
Après divers essais, nous avons gardé cette configuration pour les campagnes suivantes.



## Hermes

Après divers tests et échanges avec le Lemar/Ifremer, nous avons gardé pour la campagne, la configuration station qui, avec une recherche de fond jusqu'à 5500m, permet aux ADCP de réaliser un profil toutes les 4500 ms et aux sondeurs toutes les 9000 ms environ.

Pour le reste de la campagne, la célérité a été fixée à 1500 m/s (variables environnements) afin de garder la même référence, corrigée des tables de Carter, pour la détermination de la bathymétrie aux points de mouillages.

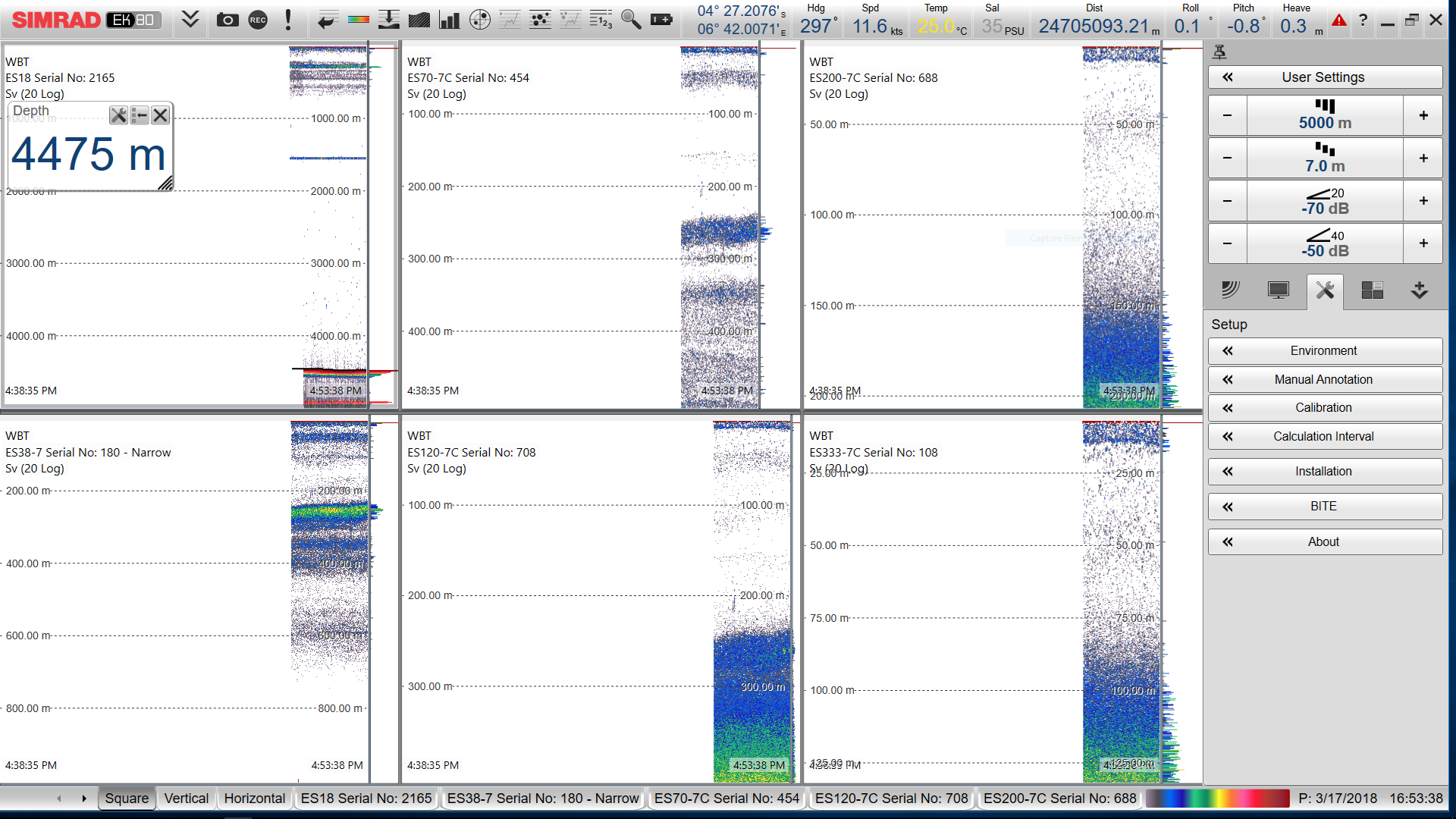


## Sondeur scientifique EK80 et ER60

Pour la configuration et le lancement de l’acquisition des sondeurs EK80/ER60, se reporter aux documents se trouvant sous :

M:\PIRATA-FR31\Documents\PROTOCOLES\ACOUSTIQUE

* INSTR\_UTILISATION\_EK60.doc (G. Roudaut 2009)



# Sauvegarde:

La sauvegarde est réalisée avec le logiciel Syncback depuis le PC d’acquisition sous Windows sur 2 disques externes et avec rsync sur le PC Linux.:

* 4 **To :** backup journalier (PC LS1)
* WD 1To: backup rsync (PC Linux-2)
* LaCie **2To :** backup journalier (PC J.Grelet)

# Traitements automatisés

## Crontab

La copie des données acquises par Thecsas est réalisée automatiquement par le PC sous Linux IRD-US191-Linux-2.

L’automatisation est réalisée via la crontab du compte science. Pour modifier son contenu, utiliser la commande crontab –e et crontab –l pour la visualiser.

$ crontab -l

# .---------------- minute (0 - 59)

# | .------------- hour (0 - 23)

# | | .---------- day of month (1 - 31)

# | | | .------- month (1 - 12) OR jan,feb,mar,apr ...

# | | | | .----- day of week (0 - 7) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat

# | | | | |

# \* \* \* \* \* command to be executed

# all processing cruise 3 times per day (local time: 8, 16, 22h)

SHELL=/bin/sh

CRUISE=PIRATA-FR31

5 9,11,17,22 \* \* \* /mnt/campagnes/${CRUISE}/local/sbin/synchro.sh > /mnt/campagnes/${CRUISE}/local/logs/synchro.log 2>&1

30 9,11,17,22 \* \* \* /mnt/campagnes/${CRUISE}/local/sbin/process-all.sh > /mnt/campagnes/${CRUISE}/local/logs/process.log 2>&1

0 8,12,21 \* \* \* rsync -av --exclude '.git' /mnt/campagnes/${CRUISE}/ /media/science/WD\ 2T/campagnes/${CRUISE}/ > /mnt/campagnes/${CRUISE}/local/logs/backup.log 2>&1

Le script synchro.sh récupère les données sur le réseau ([\\tl-nas\data\DONNEES](file:///\\nas\data\DONNEES)) et les copies dans les répertoires de data-raw et de data-processing. Lors de la mission, vérifier régulièrement les fichiers de log pour voir d’éventuels problème de droits en lecture.

## Synchro.sh

Ce script, lancé par la crontab, va récupérer les fichiers des équipements du bord acquis par TECHSAS et les copier dans notre SI.

Le résultat des courses est enregistré dans le fichier de log sous :

\\tl-nas\mission\PIRATA-FR31\local\logs\synchro.log

#!/bin/bash

#

# Mission PIRATA-FR31 N/O Thalassa fevrier 2021 J.Grelet - P.Rousselot

# script de synchronisation des donnees acquises par TECHSAS sur mission data-raw et data-processing

# > ssh -l jgrelet 192.168.57.200

# [IRD-US191-Linux-2:science]/mnt

# > df -h

# Filesystem Size Used Avail Use% Mounted on

# ...

# //tl-nas/mission 5.0T 45G 5.0T 1% /mnt/campagnes

# //tl-nas/science 5.0T 4.8G 5.0T 1% /mnt/science

# //tl-nas\homedir 1020G 292G 729G 29% /mnt/q

# //tl-nas/data 5.0T 627G 4.4T 13% /mnt/data

# /m -> /mnt/campagnes

# /q -> /mnt/q

# lancer:

# > sudo bash /mnt/campagnes/PIRATA-FR31/local/sbin/synchro.sh

# pb de droits avec montage samba, copier sous :

# /usr/local/sbin et lance par la crontab de root

# 0 7,22 \* \* \* /usr/local/sbin/synchro.sh > /mnt/campagnes/PIRATA-FR31/local/logs/synchro.log 2>&1

# repertoires source

export SOURCE=/mnt/data

export SONDEURS=/mnt/sondeurs

# disque de destination

export DRIVE=/m

# repertoire de destination

export DEST=$DRIVE/PIRATA-FR31

# nom utilise par genavir a bord pour la campagne

export CRUISE=PIRATAFR31

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "Debut de synchro : `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>" sync

echo " "

# copie SADCP vers data-raw et data-processing

echo "# copie SADCP vers data-raw et data-processing"

#/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS38/DONNEES/\* $DEST/data-raw/SADCP/OS38

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS38/ANNEXES/\* $DEST/data-raw/SADCP/OS38/ANNEXES

/bin/cp -rupv $SONDEURS/$CRUISE/ADCP/OS38/\* $DEST/data-raw/SADCP/OS38

/bin/cp -rupv $SONDEURS/$CRUISE/ADCP/OS38/\*.[L-S]TA $DEST/data-processing/SADCP/OS38/data

# OS150

#/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS150/DONNEES/\* $DEST/data-raw/SADCP/OS150

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS150/ANNEXES/\* $DEST/data-raw/SADCP/OS150/ANNEXES

/bin/cp -rupv $SONDEURS/$CRUISE/ADCP/OS150/\* $DEST/data-raw/SADCP/OS150

/bin/cp -rupv $SONDEURS/$CRUISE/ADCP/OS150/\*.[L-S]TA $DEST/data-processing/SADCP/OS150/data

# LOCH RDI DVL600

/bin/cp -rupv $SONDEURS/$CRUISE/LOCH/\* $DEST/data-raw/SADCP/DVL600

/bin/cp -rupv $SONDEURS/$CRUISE/LOCH/\*.[L-S]TA $DEST/data-processing/SADCP/DVL600/data

# copie TECHSAS ARCHIV\_NETCDF vers data-raw

echo "# copie TECHSAS ARCHIV\_NETCDF vers data-raw"

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/THS/\*.ths $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/THS

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/NAV/\*.nav $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/NAV

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/GPS/\*.gps $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/GPS

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/FBOX/\*.fbox $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/FBOX

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/MET/\*.met $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/MET

# copie TECHSAS ARCHIV\_NMEA vers data-raw et data-processing

echo "# copie TECHSAS ARCHIV\_NMEA vers data-raw et data-processing"

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/meteo/\*.met $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/METEO

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/COLCOR/\*.COLCOR $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/COLCOR

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/COLCOR/\*.COLCOR $DEST/data-raw/THERMO

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/COLCOR/\*.COLCOR $DEST/data-processing/THERMO/data

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/thsal/\*.sal $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/SBE21

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/FYBOX/\*.FYBOX $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/DONNEES//FYBOX

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/GILLA/\*.gill $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/DONNEES//GILLA

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/.sonde18/\*.snd $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/SONDE18

# copie CASINO vers data-raw et data-processing a partir de M:\PIRATAFR26\DONNEES\_BORD\CASINO

echo "# copie CASINO vers data-raw et data-processing"

#/bin/cp -ruv $SOURCE/DONNEES/$CRUISE/CASINO/DONNEES/\* $DEST/data-raw/CASINO

#/bin/cp -ruv $SOURCE/DONNEES/$CRUISE/CASINO/DONNEES/CSV/\*.csv $DEST/data-processing/CASINO/data

/bin/cp -ruv $DRIVE/CASINO/\* $DEST/data-raw/CASINO

/bin/cp -ruv $DRIVE/CASINO/CSV/\*.csv $DEST/data-processing/CASINO/data

# copie XBT .edf vers data-raw et data-processing

echo "# copie XBT .edf de data-raw et data-processing"

#/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/CELERITE/\* $DEST/data-raw/CELERITE

#/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/CELERITE/\*.edf $DEST/data-processing/CELERITE/DATA

/bin/cp -rupv $DRIVE/CELERITE/\* $DEST/data-raw/CELERITE

/bin/cp -rupv $DRIVE/CELERITE/\*.edf $DEST/data-processing/CELERITE/DATA

echo " "

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "Fin de synchro : `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

## process-all.sh

Ce script va réaliser l’extraction, le traitement des données et l’archivage sous différents formats (ASCII, XML, ODV, NetCDF) une fois la collecte effectuée par le script synchro.sh

#!/bin/bash

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "Begin process: `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo " "

shopt -s expand\_aliases

export HOME=/home/science

export DRIVE=/mnt/campagnes

export CRUISE=PIRATA-FR31

echo "Trying to source ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}"

if [ -f ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE} ]; then

. ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}

echo "Yes, seems good !!!"

else

echo "Can't source file !!! check your network, hard drive and/or ENV variables !!!"

fi

# check alias for debug

#alias

echo ""

echo "CTD processing:"

echo "---------------"

CTD

ctd

ctdnc

ctdall

ctdallnc

btl

btlnc

echo ""

echo "TSG processing:"

echo "---------------"

TSG

tsg

tsgnc

ctd-tsg

python scatter.py

echo ""

echo "XBT processing:"

echo "---------------"

XBT

xbt

xbtnc

echo ""

echo "LADCP processing:"

echo "-----------------"

LADCP

ladcp

ladcpnc

echo ""

echo "CASINO processing:"

echo "------------------"

CASINO

casino

casinonc

casinosndnc

casinotsgnc

casinofboxnc

# plot profiles and sections

# python-plots.sh

echo ""

echo "SADCP cat files:"

echo "----------------"

SADCP

cat\_all\_files.sh

echo ""

echo "GoogleEarth cruisetrack processing:"

echo "------------------------------------"

CTD

cd tracks

$LOCAL/sbin/linux/cruiseTrack2kml-linux-amd64 -config local.toml -output pirata-fr31-local.kml

$LOCAL/sbin/linux/cruiseTrack2kml-linux-amd64 -config config.toml -output pirata-fr31.kml

echo " "

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "End of process : `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

## python-plots.sh

Ce script shell permet l’automatisation des tracés des profils et sections CTD, LADCP et XBT.

Il est lancé à intervalle régulier par la crontab.

Ces sorties graphiques sont réalisées avec le programme Python plots.py sous local/sbin

Ce programme est générique et les options sont automatisées par l’utilisation d’options passées en arguments sur la ligne de commande.

Pour le tracé des sections, il est possible de fournir par anticipation le dernier numéro de profil afin que la section soit mise à jour au fur et à mesure de la réalisation des stations CTD/LADCP ou des profils XBT.

Contenu du du script shell python-plot.sh :

#!/bin/bash

shopt -s expand\_aliases

export HOME=/home/science

export DRIVE=/mnt/campagnes

export CRUISE=PIRATA-FR31

echo "Trying to source ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}"

if [ -f ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE} ]; then

. ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}

echo "Yes, seems good !!!"

else

echo "Can't source file !!! check your network, hard drive and/or ENV variables !!!"

fi

# plot profiles and sections

export NC\_DIR=netcdf

export PROF\_DIR=plots/python

export SECT\_DIR=coupes/python

echo ""

echo "Python plots processing:"

echo "------------------------"

CTD

echo "CTD profiles:"

# all profiles

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --profiles -k PRES TEMP PSAL DOX2 FLU2 -g -c k- b- r- m- g- -g -o $PROF\_DIR

echo "CTD sections:"

# section 10W

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 1N-10W\_10S-10W -k PRES TEMP --xaxis LATITUDE -l 5 28 --yscale 0 250 250 2000 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 1N-10W\_10S-10W -k PRES PSAL --xaxis LATITUDE -l 5 28 --yscale 0 250 250 2000 --yinterp 10 --clevels=20 --autoscale 33 37 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 1N-10W\_10S\_10W -k PRES DENS --xaxis LATITUDE -l 5 28 --yscale 0 250 250 2000 --yinterp 10 --clevels=20 --autoscale 20 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 1N-10W\_10S-10W -k PRES DOX2 --xaxis LATITUDE -l 5 28 --yscale 0 250 250 2000 --yinterp 10 --clevels=20 --autoscale 0 250 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 1N-10W\_10S-10W -k PRES FLU2 --xaxis LATITUDE -l 5 28 --yscale 0 250 --yinterp 10 --clevels 20 --autoscale 0 2 -o $SECT\_DIR

# point fixe 0-10W

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-10W -k PRES TEMP --xaxis TIME -l 33 48 --yscale 0 200 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-10W -k PRES PSAL --xaxis TIME -l 33 48 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 33 37 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-10W -k PRES DENS --xaxis TIME -l 33 48 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 20 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-10W -k PRES DOX2 --xaxis TIME -l 33 48 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 0 250 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-10W -k PRES FLU2 --xaxis TIME -l 33 48 --yscale 0 200 --yinterp 10 --clevels 20 --autoscale 0 2 -o $SECT\_DIR

# section 23W

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 2S-4N\_23W -k PRES TEMP --xaxis LATITUDE -l 50 78 --yscale 0 250 250 2000 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR --exclude 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 2S-4N\_23W -k PRES PSAL --xaxis LATITUDE -l 50 78 --yscale 0 250 250 2000 --yinterp 10 --clevels=20 --autoscale 33 37 -o $SECT\_DIR --exclude 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 2S-4N\_23W -k PRES DENS --xaxis LATITUDE -l 50 78 --yscale 0 250 250 2000 --yinterp 10 --clevels=20 --autoscale 20 30 -o $SECT\_DIR --exclude 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 2S-4N\_23W -k PRES DOX2 --xaxis LATITUDE -l 50 78 --yscale 0 250 250 2000 --yinterp 10 --clevels=20 --autoscale 0 250 -o $SECT\_DIR --exclude 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append 2S-4N\_23W -k PRES FLU2 --xaxis LATITUDE -l 50 78 --yscale 0 250 --yinterp 10 --clevels 20 --autoscale 0 2 -o $SECT\_DIR --exclude 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

# point fixe 0-23W

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W -k PRES TEMP --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W+profiles -k PRES TEMP --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR --display

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W -k PRES PSAL --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 33 37 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W+profiles -k PRES PSAL --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 33 37 -o $SECT\_DIR --display

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W -k PRES DENS --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 20 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W -k PRES DOX2 --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 0 250 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W+profiles -k PRES DOX2 --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels=20 --autoscale 0 250 -o $SECT\_DIR --display

plots.py $NC\_DIR/OS\_${CRUISE}\_CTD.nc --type CTD --sections --append point-fixe\_0-23W -k PRES FLU2 --xaxis TIME -l 54 69 --yscale 0 200 --yinterp 10 --clevels 20 --autoscale 0 2 -o $SECT\_DIR

XBT

echo "XBT profiles:"

# all profiles

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc --type XBT --profiles -k DEPTH TEMP DENS SVEL -c k- b- k- g- -g -o plots

echo "XBT sections:"

# sections

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc --type XBT --sections --append CANARIES\_1-30N-10W -k DEPTH TEMP --xaxis LATITUDE -l 2 17 --yscale 0 250 250 900 --xinterp 15 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc --type XBT --sections --append 10S-20S\_10W -k DEPTH TEMP --xaxis LATITUDE -l 18 28 --yscale 0 250 250 900 --xinterp 10 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc --type XBT --sections --append 10S-10W\_0-0 -k DEPTH TEMP --xaxis LATITUDE -l 29 39 --yscale 0 250 250 900 --xinterp 11 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc --type XBT --sections --append 0-10W\_0-23W -k DEPTH TEMP --xaxis LONGITUDE -l 39 61 --yscale 0 250 250 900 --xinterp 22 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR -e 59

plots.py $NC\_DIR/OS\_${CRUISE}\_XBT.nc --type XBT --sections --append 4N-23W\_CANARIES -k DEPTH TEMP --xaxis LATITUDE -l 62 75 --yscale 0 250 250 900 --yinterp 10 --clevels=30 --autoscale 0 30 -o $SECT\_DIR --display

LADCP

echo "LADCP profiles:"

# all profiles

plots.py $NC\_DIR/OS\_${CRUISE}\_ADCP.nc --type ADCP --profiles -k DEPTH EWCT NSCT -c k- r- b- -g -o $PROF\_DIR

echo "LADCP sections:"

# sections

plots.py $NC\_DIR/OS\_${CRUISE}\_ADCP.nc --type ADCP --sections --append 1N-10W\_10S-10W -k DEPTH EWCT NSCT -l 5 28 --xaxis LATITUDE --yscale 0 250 250 2000 --yinterp 20 --clevels 30 --autoscale -150 150 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_ADCP.nc --type ADCP --sections --append point-fixe\_0-10W -k DEPTH EWCT NSCT -l 33 48 --xaxis TIME --yscale 0 500 --yinterp 20 --clevels 30 --autoscale -150 150 -o $SECT\_DIR

plots.py $NC\_DIR/OS\_${CRUISE}\_ADCP.nc --type ADCP --sections --append 2S-4N\_23W -k DEPTH EWCT NSCT -l 50 78 --xaxis LATITUDE --yscale 0 250 250 2000 --yinterp 20 --clevels 30 --autoscale -150 150 -o $SECT\_DIR --exclude 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

plots.py $NC\_DIR/OS\_${CRUISE}\_ADCP.nc --type ADCP --sections --append point-fixe\_0-23W -k DEPTH EWCT NSCT -l 54 69 --xaxis TIME --yscale 0 500 --yinterp 20 --clevels 30 --autoscale -150 150 -o $SECT\_DIR

Les options du programme disponible avec l’aide : plots.py -h

> plots.py -h

usage:

python plots.py -t <TYPE> -s (SECTIONS) <OPTIONS> ... | -p (PROFILES) <OPTIONS> ...

PROFILES:

python plots.py netcdf/OS\_PIRATA-FR31\_CTD.nc -t CTD -p -k PRES TEMP PSAL DOX2 FLU2 -g -c k- b- r- m- g-

python plots.py netcdf/OS\_PIRATA-FR31\_XBT.nc -t XBT -p -k DEPTH TEMP DENS SVEL -c k- b- k- g- -g

python plots.py netcdf/OS\_PIRATA-FR31\_ADCP.nc -t ADCP -p -k DEPTH EWCT NSCT -c k- r- b- -g

SECTIONS:

python plots.py netcdf/OS\_PIRATA-FR31\_CTD.nc -t CTD -s -k PRES TEMP -l 5 28 --xaxis LATITUDE --yscale 0 250 250 2000 --xinterp 24 --yinterp 200 --clevels=30 --autoscale 0 30

python plots.py netcdf/OS\_PIRATA-FR31\_CTD.nc -t CTD -s --append 1N-10W\_10S\_10W -k PRES PSAL -l 5 28 --xaxis LATITUDE --yscale 0 250 250 2000 --xinterp 24 --yinterp 100 --clevels=15 --autoscale 34 37

python plots.py netcdf/OS\_PIRATA-FR31\_ADCP.nc -t ADCP -s --append point-fixe\_0-10W -k DEPTH EWCT NSCT -l 33 45 --xaxis TIME --yscale 0 500 --xinterp 20 --yinterp 50 --clevels 15 --autoscale -150 150

python plots.py netcdf/OS\_PIRATA-FR31\_XBT.nc -t XBT -s DEPTH TEMP -xaxis LATITUDE

python plots.py netcdf/OS\_PIRATA-FR31\_XBT.nc -t XBT -s DEPTH TEMP -xaxis TIME -l 29 36

This program read CTD NetCDF file and plot parameters vs PRES

positional arguments:

files netcdf file to parse

optional arguments:

-h, --help show this help message and exit

-a APPEND, --append APPEND

string to append in output filename

-t {CTD,XBT,ADCP}, --type {CTD,XBT,ADCP}

select type instrument CTD, XBT or LADCP

-p, --profiles, --profile

plot profiles

-k KEYS [KEYS ...], --keys KEYS [KEYS ...]

select physical parameters key(s), (default: None)

-l LIST [LIST ...], --list LIST [LIST ...]

select first and last profile, default (none) is all

-e [EXCLUDE [EXCLUDE ...]], --exclude [EXCLUDE [EXCLUDE ...]]

give a list of profile(s) to exclude

-c COLORS [COLORS ...], --colors COLORS [COLORS ...]

select colors, ex: k- b- r- m- g-

-g, --grid add grid

-s, --sections, --section

plot sections

--xaxis {LATITUDE,LONGITUDE,TIME}

select xaxis for sections

--yscale [YSCALE [YSCALE ...]]

select vartical scale for sections, ex: 0 2000 or 0 250 250 2000

--xinterp XINTERP horizontal interpolation points

--yinterp YINTERP vertical interpolation step, none plot raw data

--clevels CLEVELS contour levels

--autoscale [AUTOSCALE [AUTOSCALE ...]]

None: use NetCDF valid min and max

True: use min(Z) and max(Z)

[min, max]: define manually min and max

--display, --display\_profiles

display profiles number on top axes

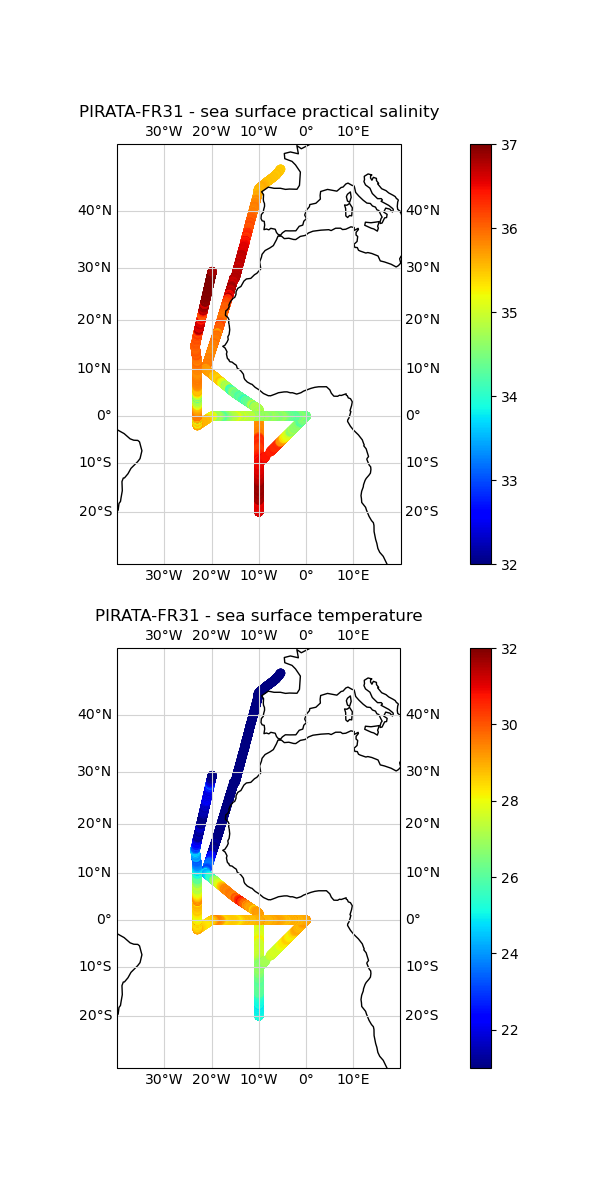
-o OUT, --out OUT output path, default is plots/

-d, --debug display debug informations

J. Grelet IRD US191 - March 2021 / April 2021

## Tracés de la route du TSG

Le trace de la route est réalisé manuellement par le programme Python scatter.py (il y a problème avec conda lorsqu’il est inclus dans Python-plots.sh, à suivre).



# Traitements complémentaires

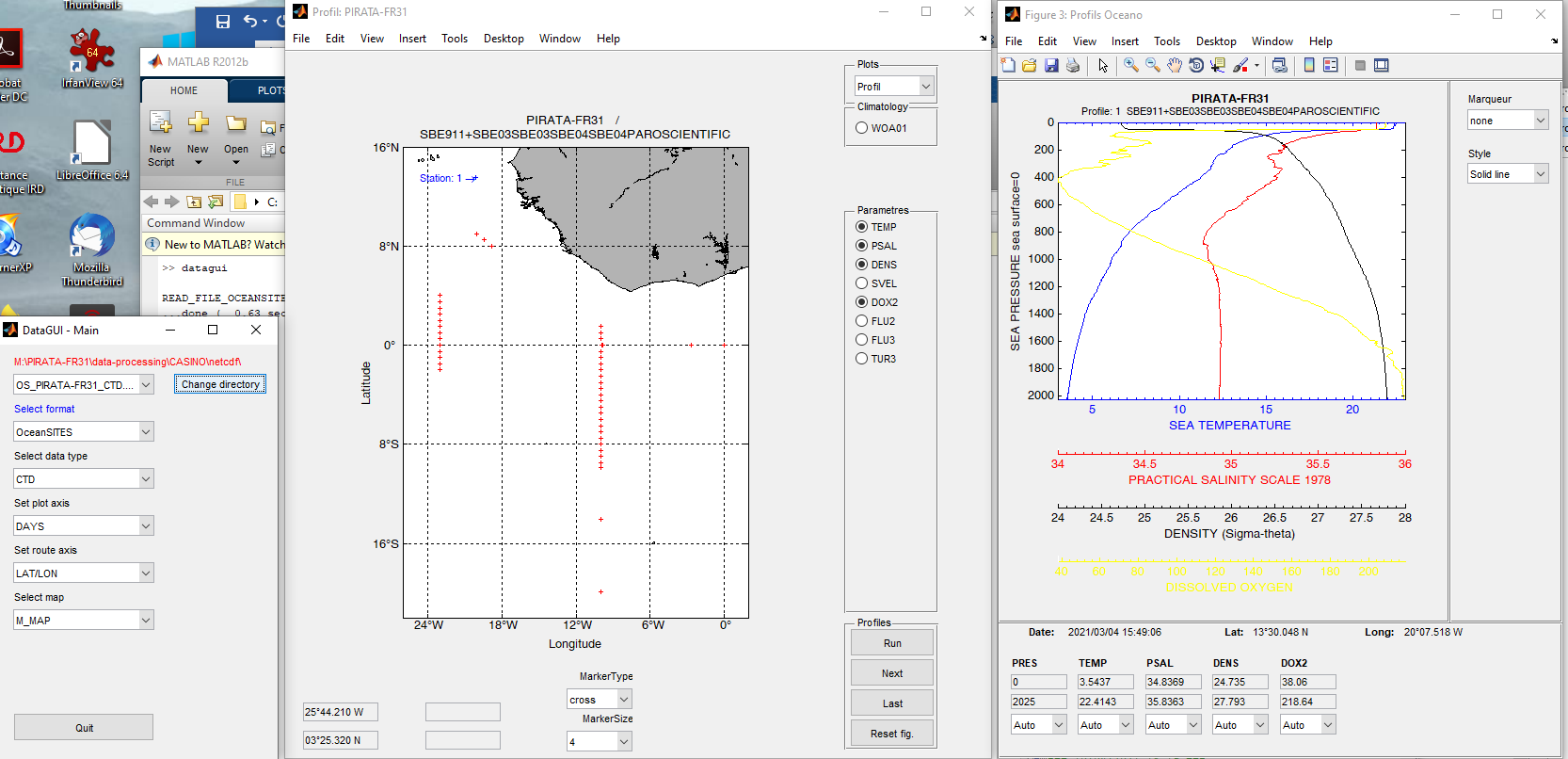
## Datagui:

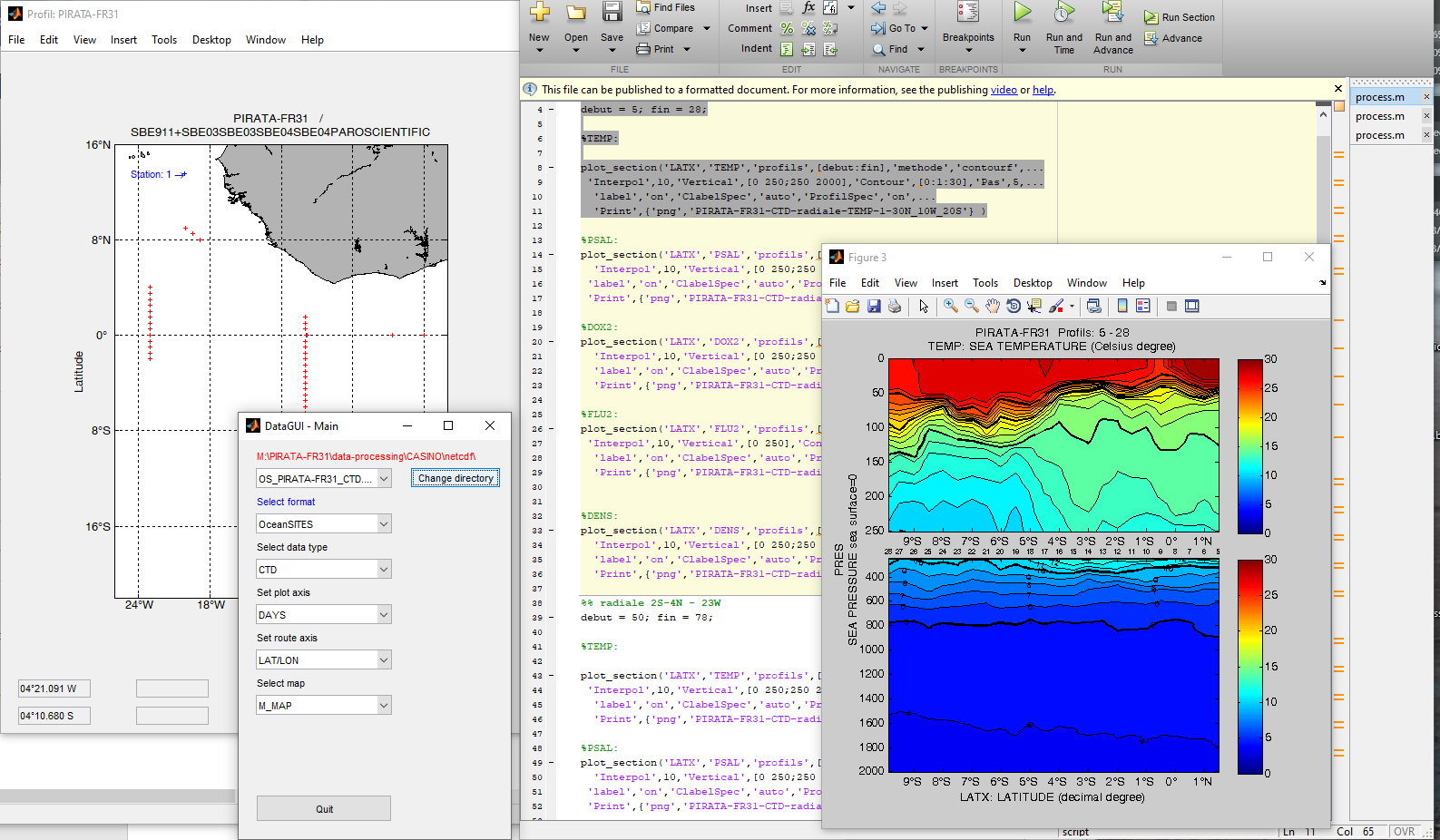
Lancer datagui sous Matlab, charger le fichier, visualiser les profils.

Pour les sections, se placer dans le répertoire CTD/coupes, LADCP/coupes et XBT/coupes et une fois les données chargées en mémoire, lancer le script process.m qui va lancer différentes impressions (fichiers .jpg) à l'aide de la fonction plot\_section.

Modifier les variables de début et fin de section afin de mettre à jour la section avec les dernières stations réalisées et copier/coller/modifier les appels à plot\_section pour réaliser le tracé de nouvelles sections.

Taper help ou doc plot\_section pour plus d'info.





## Tracés de la route (fichiers kml Google Earth)

Est réalisé à intervalle régulier depuis le répertoire M:\PIRATA-FR31\data-processing\CTD\tracks :

1. Le script Matlab cruistrack.m permet de tracer la position des stations CTD, des profils XBT ainsi que la route du navire lue depuis le fichier du Thermosalinographe. Pour cette campagne, le script extrait les données de positions directement depuis les fichiers NetCDF.
2. Le programme cruiseTrack2kml.exe permet de générer un fichier .kml que l’on pourra ouvrir avec Google Earth.

### Copie des fichiers sur le serveur Nginx en local :

Créer des liens symboliques sous la racine sur serveur NGINX. Le serveur Web ne permet pas de « browser » le contenu d’un répertoire, il faut donc entrer l’URL complète pour visualiser une image depuis son navigateur.

[US191-Linux-2:science]/var/www/html

> ll /var/www/html/PIRATA-FR31/

CELERITE -> /mnt/campagnes/PIRATA-FR31/data-processing/CELERITE/plots/

CTD -> /mnt/campagnes/PIRATA-FR31/data-processing/CTD/plots/

THERMO -> /mnt/campagnes/PIRATA-FR31/data-processing/THERMO/plots/

Démarrer, stopper, redémarrer, le serveur web:

$ sudo service nginx <start|stop|restart>

### Copie des fichiers sur le serveur web à l’ifremer :

Pour que les images des profils soient accessibles sur le web, il est nécessaire de les copier régulièrement dans le répertoire /home/wwwinter/www/htdocs/ird/us191/cruises/pirata-fr31 avec la commande Unix scp depuis le PC Linux de traitement dans les sous répertoires CTD, TSG et XBT, et en accord avec la définition des variables ctdPlots, xbtPlots et tsgPlots définies dans le fichier config.toml ci-dessous.

La machine Linux PC-giga (IP 134.246.159.164) de la classe 134.246.159 est accessible en ssh depuis la classe C privée du réseau scientifique du bord (192.168.51) grâce à la mise en place de règle de routage au niveau du service informatique de l’Ifremer.

> pwd

/m/PIRATA-FR31/data-processing/CTD/tracks

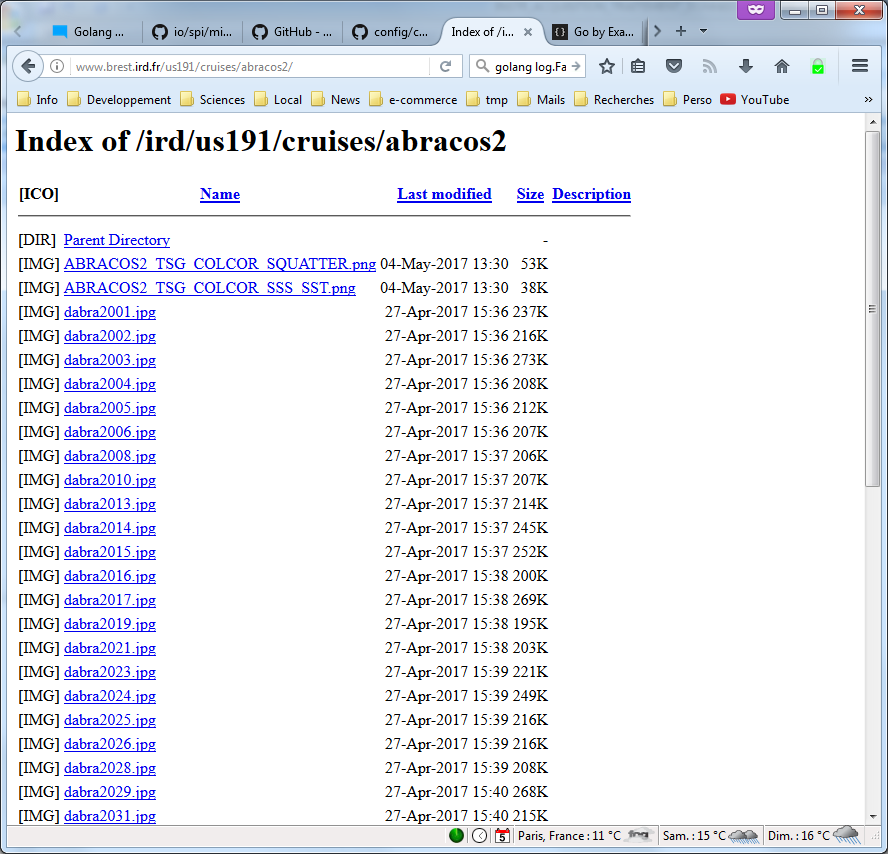
> scp ../plots/downcast/ds??.jpg [jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/pirata-fr31](mailto:jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/abracos2)/CTD

> scp ../../COLCOR/plots/\*.png [jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/pirata-fr31/TSG](mailto:jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/abracos2/TSG)

> scp ../../CELERITE/plots/\*.jpg [jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/pirata-fr31](mailto:jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/abracos2)/XBT

On peut vérifier la présence des fichiers en se connectant à l’URL :

[http://www.brest.ird.fr/us191/cruises/pirata-fr31/](http://www.brest.ird.fr/us191/cruises/abracos2/)



A changer pour FR31

### Lancement du script de création du fichier kml (web):

[ntird-us191-jg4:NTIRD-US191-JG4+jgrelet]/m/PIRATA-FR31/data-processing/CTD/tracks

> $GOBIN/cruiseTrack2kml.exe

cruiseTrack2kml, version 0.3.2 J.Grelet IRD - US191 IMAGO

Saturday, 17-Mar-18 07:35:03 GMT

Cruise: PIRATA-FR31

Ship: THALASSA

CtdPlots: http://www.brest.ird.fr/us191/cruises/pirata-fr31/CTD/ds%s.jpg

TsgPlots: http://www.brest.ird.fr/us191/cruises/sargasse/TSG/PIRATA-FR31\_TSG\_COLCOR\_SCATTER.png

CtdFile: m:/PIRATA-FR31/data-processing/CTD/ascii/pirata-fr31.ctd

XbtFile: m:/PIRATA-FR31/data-processing/CELERITE/ascii/pirata-fr31.xbt

TsgFile: m:/PIRATA-FR31/data-processing/THERMO/ascii/pirata-fr31.gps

KmlFile: PIRATA-FR31.kml

File: m:/PIRATA-FR31/data-processing/THERMO/ascii/pirata-fr31.gps

Fields:[LATITUDE LONGITUDE]

Vars: map[LONGITUDE:{3 float64} LATITUDE:{2 float64}]

SkipLine: 2

File: m:/PIRATA-FR31/data-processing/CTD/ascii/pirata-fr31.ctd

Fields:[PRFL BEGIN\_DATE BEGIN\_TIME END\_DATE END\_TIME LAT LAT\_S LON LON\_S PMAX BOTTOM\_DEPTH]

Vars: map[PRFL:{1 int} END\_DATE:{4 string} LON:{8 string} LON\_S:{9 string} PMAX:{10 float64} BOTTOM\_DEPTH:{11 float64} BEGIN\_DATE:{2 string} BEGIN\_TIME:{3 string} END\_TIME:{5 string} LAT:{6 string} LAT\_S:{7 string}]

SkipLine: 1

File: m:/PIRATA-FR31/data-processing/CELERITE/ascii/pirata-fr31.xbt

Fields:[PRFL BEGIN\_DATE BEGIN\_TIME LAT LAT\_S LON LON\_S PMAX PROBE]

Vars: map[LON:{6 string} PROBE:{9 string} BEGIN\_DATE:{2 string} BEGIN\_TIME:{3 string} LAT\_S:{5 string} PMAX:{8 float64} PRFL:{1 int} LAT:{4 string} LON\_S:{7 string}]

SkipLine: 2

TSG mark: 405

CTD mark: 10

XBT mark: 62

Cliquer 2 fois sur le fichier pirata-fr31.kml pour lancer l’application Google Earth.

Ci-dessous, un exemple de lecture d’un fichier kml lors de la campagne ABRACOS2

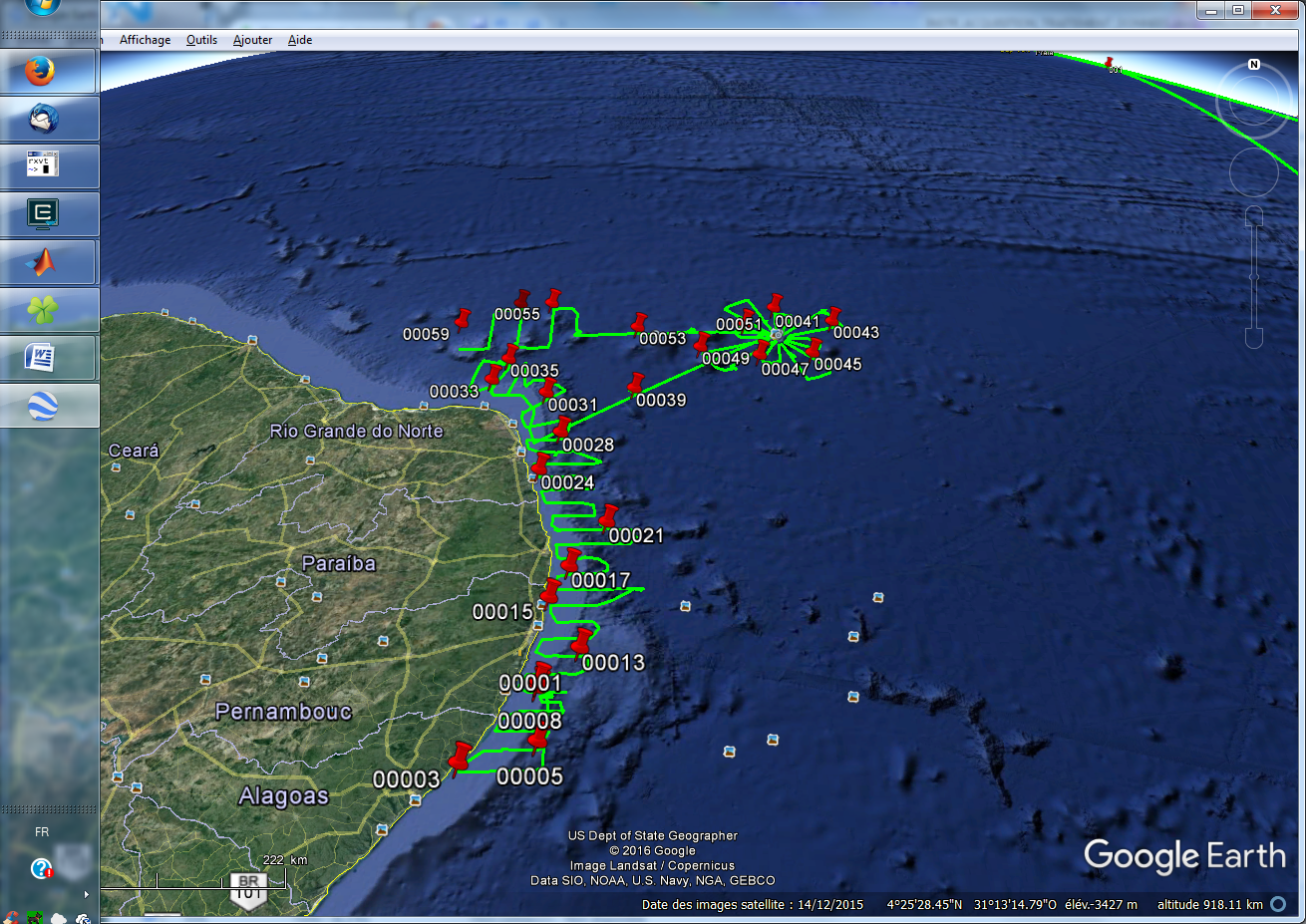


Figure 6: Affichage de la route sous Google Earth

Puis sélectionner une station pour faire apparaitre le profil de la CTD :

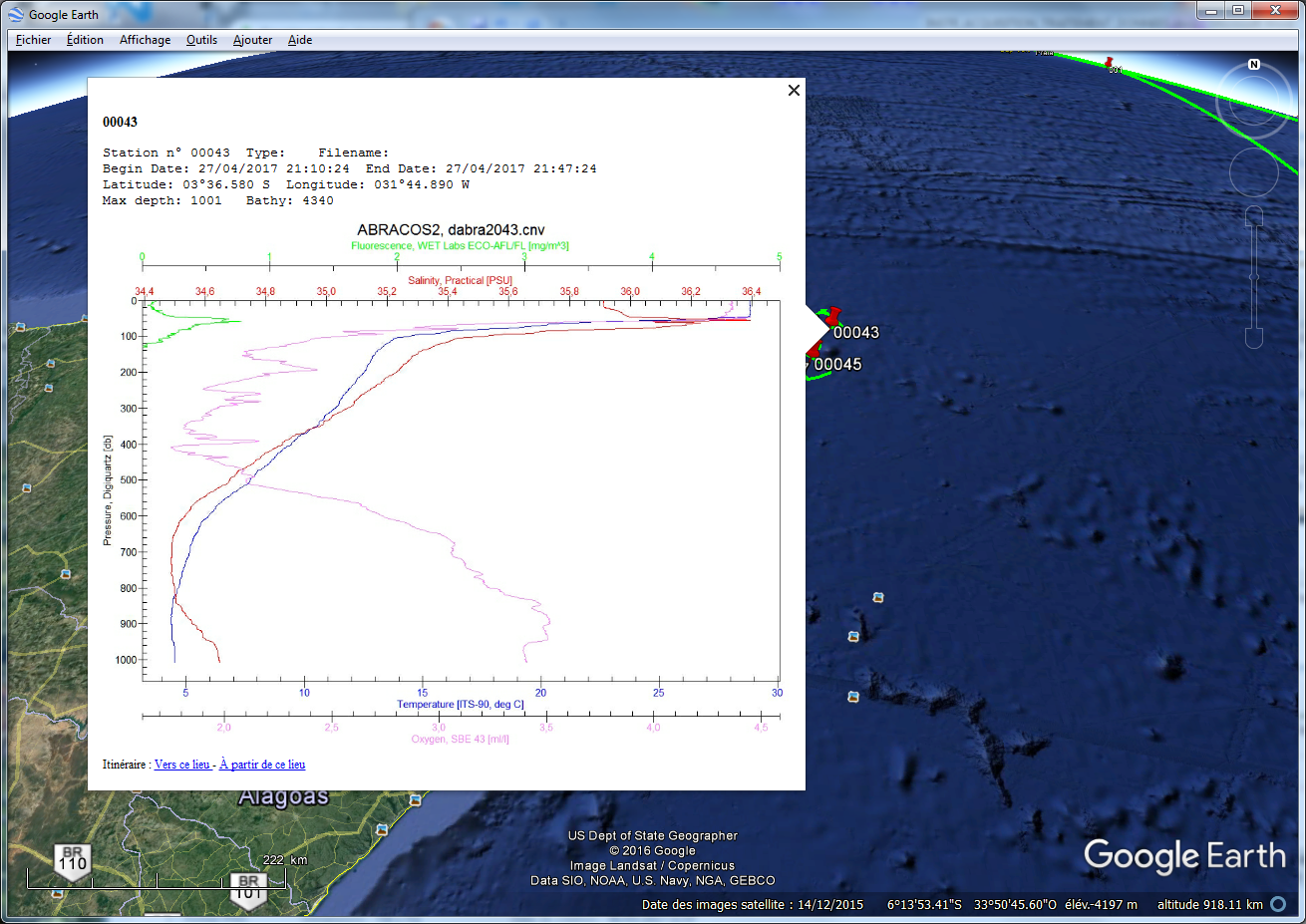


Figure 7: Affichage d'un profil CTD sous Google Earth

### Transfert des images sur le site web de l’Ifremer

Se placer dans le répertoire : /m/PIRATA-FR31/data-processing/CTD/tracks

Nous allons copier les fichiers des profils sur le PC Linux dans un répertoire qui est monté en lecture sur le serveur web de l’Ifremer.

L’adresse est la suivante : [jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises](mailto:jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises)

Les fichiers seront copiés avec la commande scp :

> scp ../plots/downcast/ds\*.jpg [jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/pirata-fr31/CTD](mailto:jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/sargasse/CTD)

Afin de ne copier que les derniers fichiers sur le serveur distant et ainsi économiser de la bande passante, utiliser la commande suivante pour ne copier que les fichiers des stations 5, 6 et 7 :

> scp ../plots/downcast/ds0[5-7].jpg jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/

ds05.jpg 100% 135KB 134.7KB/s 00:00

ds06.jpg 100% 145KB 144.5KB/s 00:00

ds07.jpg 100% 117KB 116.8KB/s 00:00

L’image de la SST/SSS le long de la route est obtenue avec le script scatter.py ou datagui, dans ce cas, charger le fichier Netcdf du TSG, sélectionner les variables SSTP et SSPS puis cliquer sur « Traj color » .

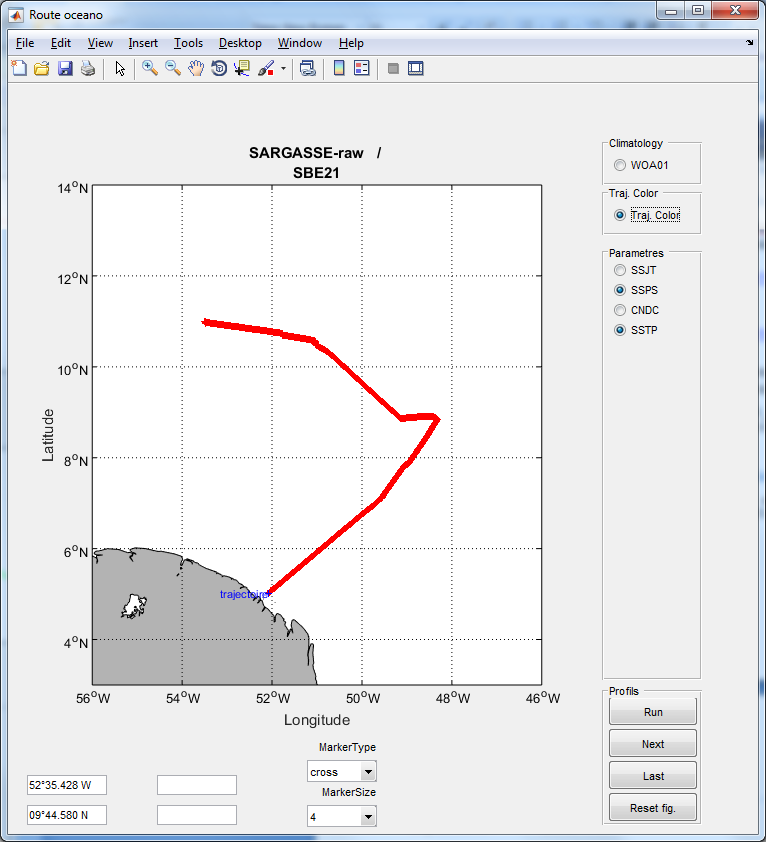


Figure 8: Tracé de la route du Thermosalinographe

Sauvegarder l’image au format .png afin que le nom corresponde avec le champ TsgPlots du fichier de config .toml, soit par exemple :

TsgPlots: http://www.brest.ird.fr/us191/cruises/pirata-fr31/TSG/PIRATA-FR31\_TSG\_COLCOR\_SCATTER.png

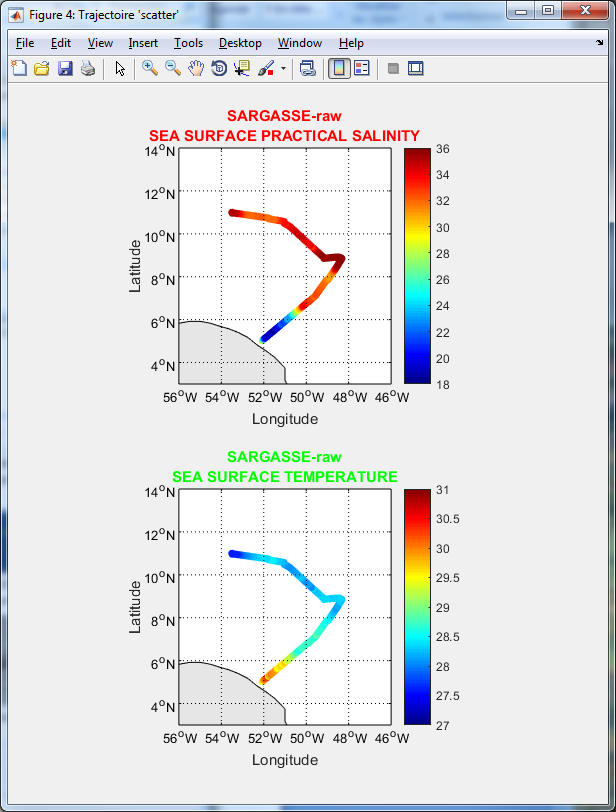


Figure 9: Thermosalinographe, trajectoire colorée pour SST et SSS

> scp ../../THERMO/plots/PIRATA-FR31\_TSG\_COLCOR\_SCATTER.png [jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/pirata-fr31/TSG](mailto:jgrelet@134.246.159.164:/home/wwwinter/www/htdocs/ird/us191/cruises/sargasse/TSG)

### Lancement du script de création du fichier kml en local

> cruiseTrack2kml.exe -config local.toml -output pirata-fr31-local.kml cruiseTrack2kml, version 0.3.1 J.Grelet IRD - US191 IMAGO

Sunday, 25-Jun-17 08:15:13 GFT

Cruise: PIRATA-FR31

Ship: ANTEA

CtdPlots: http://192.168.81.172/PIRATA-FR31/ds%s.jpg

TsgPlots: http://192.168.81.172/PIRATA-FR31/PIRATA-FR31\_TSG\_COLCOR\_SCATTER.png

CtdFile: ../ascii/pirata-fr31.ctd

XbtFile: none

TsgFile: ../../THERMO/ascii/pirata-fr31-raw.gps

KmlFile: pirata-fr31.kml

File: ../../THERMO/ascii/pirata-fr31-raw.gps

Fields:[LATITUDE LONGITUDE]

Vars: map[LATITUDE:{2 float64} LONGITUDE:{3 float64}]

SkipLine: 2

File: ../ascii/pirata-fr31.ctd

Fields:[PRFL BEGIN\_DATE BEGIN\_TIME END\_DATE END\_TIME LAT LAT\_S LON LON\_S PMAX BOTTOM\_DEPTH]

Vars: map[END\_TIME:{5 string} LAT:{6 string} LAT\_S:{7 string} LON:{8 string} LON\_S:{9 string} PMAX:{10 float64} BEGIN\_DATE:{2 string} BEGIN\_TIME:{3 string} END\_DATE:{4 string} BOTTOM\_DEPTH:{11 float64} PRFL:{1 int}]

SkipLine: 1

Sargasse-local.kml

TSG mark: 88

CTD mark: 5

**Notes** : En début de mission, il faut créer un répertoire PIRATA-FR31 sur le serveur web NGINX puis créer les 3 liens suivants avec la commande ln –s <source< <dest>

Soit :

> cd /var/www/html/PIRATA-FR31

> ln –s /mnt/campagnes/PIRATA-FR31/data-processing/CELERITE/plots/ CELERITE

> ln –s /mnt/campagnes/PIRATA-FR31/data-processing/CELERITE/plots/ CTD

> ln –s /mnt/campagnes/PIRATA-FR31/data-processing/CELERITE/plots/ THERMO

On vérifie le résultat avec la commande ls :

[US191-Linux-2:science]/var/www/html/PIRATA-FR31

> ls -la

CELERITE -> /mnt/campagnes/PIRATA-FR31/data-processing/CELERITE/plots/

CTD -> /mnt/campagnes/PIRATA-FR31/data-processing/CTD/plots/

THERMO -> /mnt/campagnes/PIRATA-FR31/data-processing/THERMO/plots/

### Fichier de configuration config.toml

Le fichier de configuration du programme, à modifier suivant la campagne et les fichiers utilisés :

> cat config.toml

# this is a configuration file for cruiseTrack2kml program

cruise = "PIRATA-FR31"

ship = "ANTEA"

ctdPlots = "http://www.brest.ird.fr/us191/cruises/pirata-fr31/CTD/ds%s.jpg"

ctdPrefix = 2

xbtPlots = "http://www.brest.ird.fr/us191/cruises/pirata-fr31/XBT/PIRATA-FR31-%s\_XBT.png"

xbtPrefix = 2

tsgPlots = "http://www.brest.ird.fr/us191/cruises/pirata-fr31/TSG/PIRATA-FR31\_TSG\_COLCOR\_SCATTER.png"

sizePlots = 700

stationNumber = true

tsgSkip = 2

tsgSplit = "LATITUDE,2,float64,LONGITUDE,3,float64"

ctdSkip = 1

ctdSplit = "PRFL,1,int,BEGIN\_DATE,2,string,BEGIN\_TIME,3,string,END\_DATE,4,string,END\_TIME,5,string,LAT,6,string,LAT\_S,7,string,LON,8,string,LON\_S,9,string,PMAX,10,float64,BOTTOM\_DEPTH,11,float64"

xbtSkip = 2

xbtSplit = "PRFL,1,int,BEGIN\_DATE,2,string,BEGIN\_TIME,3,string,LAT,4,string,LAT\_S,5,string,LON,6,string,LON\_S,7,string,PMAX,8,float64,PROBE,9,string"

[windows]

tsgFile = "../../THERMO/ascii/pirata-fr31-raw.gps"

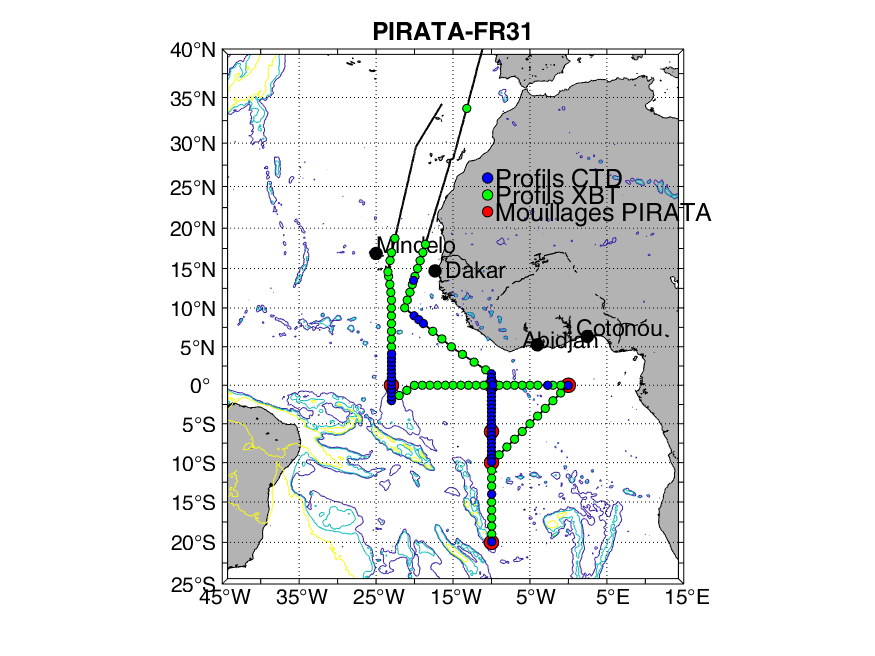
ctdFile = "../ascii/pirata-fr31.ctd"

xbtFile = "none"

### Tracé de la route et des opérations

Sous data-processing/CTD/tracks et XBT/tracks, on peut exécuter les scripts Matlab cruistrk\_xbt.m et cruistrk\_ctd.m pour tracer la route (fichier du TSG) et la position des stations CTD, profils XBT et mouillages PIRATA.

Editer le script et adapter les paramètres en fonctions des coordonnées géographiques. Utile pour le rapport de mission.



# De retour au laboratoire

Le contenu du disque de backup est copié sous le répertoire campagne sur serveur tera10, système de fichiers dédié « ird-campagnes »

Pour avoir accès aux données et scripts de traitement, l’utilisateur doit modifier ses variables d’environnement $DRIVE et $CRUISE, afin de les faires pointer vers le bon répertoire de la campagne :

Ouverture d’une session, terminal Unix :

jeu mar 3 14:40:41 CET 2011

source /mnt/campagnes/PIRATA-FR31/local/etc/skel/.cshrc.PIRATA-FR31

Ok...

[IRD-US191-GB-BXi5-4200:science]~

> echo $DRIVE

/mnt/campagnes

[IRD-US191-GB-BXi5-4200:science]~

> echo $CRUISE

PIRATA-FR31

[IRD-US191-GB-BXi5-4200:science]~

> CTD

[IRD-US191-GB-BXi5-4200:science]/mnt/campagnes/PIRATA-FR31/data-processing/CTD

# Mise à disposition des scripts et documents

Un article décrivant l’utilisation de la sonde 911plus et des LADCP est disponible sous l’espace collaboratif de l’US191 IMAGO à l’adresse suivante :

<http://www.ird.fr/us191/spip.php?article27>

A la fin de chaque campagne, deux archives y sont déposées, l’une contenant les différentes instructions et formulaires (enregistrements) utilisés, la deuxième regroupant la totalité des scripts utilisés pour l’acquisition et la validation des données.

Se placer sous /mnt/campagnes/PIRATA-FR31/data-processing par exemple. Ligne de commande utilisée pour générer l’archive des scripts :

find . \( -name "\*.pl" –o -name "\*.m" -o -name "\*.cnt" -o -name "\*.sh" -o -name "\*.psa" -o –name "\*.bat" -o -name "\*.con" -o -name "\*.batch" -o -name "\*.CMD" \) | xargs tar zcvf scripts-PIRATA-FR31.tgz

# Log-book

Il est conseillé de tenir et de compléter au jour le jour un Log-book que l’on mettra sous M:\PIRATA-FR31\Documents\ENREGISTREMENTS\PIRATA-FR31\_SYNTHESE\_STATIONS.xls

à partir des informations contenues dans les fichiers d'entête de la CTD.

Pour cela, faire un cat des fichiers PIRATA-FR31.ctd et PIRATA-FR31.btl se trouvant dans le répertoire CTD. Il est possible de copier l’ensemble d’une colonne en éditant les fichiers avec VIM et en sélectionnant la colonne graphiquement en gardant la touche ATL appuyée.

Compléter la colonne "Commentaires" en notant tous les informations utiles au post-traitement, problèmes techniques, changement de capteurs, erreurs de manipulation, etc....

Y reporter également toutes les opérations réalisées au cours de la mission : numéro de série des capteurs, changement de capteurs, etc.

# Légendes

[Figure 1: Antéa, Interconnection informatique 4](#_Toc483922755)

[Figure 2: Antéa, Synoptique CINNA 5](#_Toc483922756)

[Figure 3: Gestionnaire de sites Filezilla 22](#_Toc483922757)

[Figure 4: Copie d'un fichier COLCOR avec Filezilla 22](#_Toc483922758)

[Figure 5: OSEA, configuration petits fonds (inshore) **Erreur ! Signet non défini.**](#_Toc483922759)

[Figure 6: OSEA, configuration grands fonds (Offshore) 26](#_Toc483922760)

[Figure 7: Affichage de la route sous Google Earth 30](#_Toc483922761)

[Figure 8: Affichage d'un profil CTD sous Google Earth 31](#_Toc483922762)

# Suivi des versions de ce document

|  |  |  |  |
| --- | --- | --- | --- |
| **Rédacteur** | | **Approbateur** | |
| Nom : | Jacques Grelet | Nom : |  |
| Fonction : |  | Fonction : |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Dates** | **Versions** | **Chapitres concernés** | **Commentaires et modifications** |
| mars 2011 | 01 | tous | Mise sous forme pour PIRATA -FR21 |
| juillet 2012 | 02 | tous | Mis à jour pour PANDORA |
| mars 2013 | 03 | tous | Mis à jour pour PIRATA-FR23 |
| mai 2013 | 04 | tous | Mis à jour pour PIRATA-FR23 |
| Nov 2013 | 05 | tous | Mis à jour pour EPURE4 |
| février 2014 | 06 | tous | Mis à jour pour AMOP |
| mai 2014 | 07 | tous | Mis à jour pour PIRATA-FR24 |
| avril 2015 | 08 | tous | Mis à jour pour PIRATA-FR25 |
| juillet 2015 | 09 | tous | Mis à jour pour CASSIOPEE |
| avril 2016 | 10 | tous | Mis à jour pour PIRATA-FR26 |
| sept 2016 | 11 | tous | Mis à jour pour LAPEROUSE |
| Avril 2017 | 12 | tous | Mis à jour pour ABRACOS2 |
| Juin 2017 | 13 | tous | Mis à jour pour PIRATA-FR27 |
| Mars 2018 | 14 | tous | Ecriture complète pour PIRATA-FR31 |

|  |  |
| --- | --- |
| **Relecteur** | **Date** |
|  |  |