CWDP User Manual and Reference Guide

CFOSAT Wind Data Processor

Version 1.0
Date: 10/09/2021

Zhen Li, Anton Verhoef, Ad Stoffelen
KNMI, De Bilt, the Netherlands

Change record			
Version	Date	Author	Remarks
1.1	06-12-2017	Zhen Li	Initial draft
1.2	12-12-2017	Anton Verhoef	Review
1.3	15-12-2017	Zhen Li	Revision
0.9	21-03-2018	Anton Verhoef	For pre-check beta release
1.0	18-05-2021	Zhen Li, AV	For v1.0 release, pre-checked
1.0.01	23-08-2021	Anton Verhoef	Modified according to DRR RIDs
1.0.02	10-09-2021	Anton Verhoef	Extra modification after DRR RIDs

Table of contents

1. Introduction

CFOSAT (Chinese-French Oceanic SATellite) is a joint mission of the Chinese (CNSA) and French (CNES) space agencies with the goal to monitor the ocean surface winds and waves and to provide information on related ocean and atmospheric science and applications. It was launched on 29 October 2018. This document is the User Manual of the CFOSAT Wind Data Processor (CWDP) software package. It is organized as follows: This introduction section gives general information about CWDP. Section 2 contains the CWDP output specifications. Section 3 describes how to install, compile, and link the CWDP software. Section 4 contains the processing details.

1.1 Aims and scope

The CFOSAT Wind Data Processor (CWDP) is a software package written mainly in Fortran 90. The parts and libraries for handling HDF5, NetCDF, and BUFR format data have been partly written in C. CWDP is a wind processor for Ku-band rotating fan-beam scatterometer. It is an adaption from the PenWP (Pencil beam Wind Processor) [1]. The difference between PenWP and CWDP is that PenWP is for rotating pencil-beam scatterometers (SeaWinds on QuikSCAT, OSCAT on OceanSat-2 and ScatSat-1, RapidScat on the International Space Station, and HSCAT on the HY-2 satellite series), whereas CWDP is for rotating fan-beam scatterometers. CWDP generates surface winds from Level-2A (L2A) data. The L2A data which are the input to this wind processor are produced by the L2A processor [2]. The L2A processor includes two steps: L1B to L2A conversion and L2A HDF to L2A BUFR conversion; both are part of the CWDP package. CWDP uses L2A BUFR data to retrieve ocean surface winds. It has the ability to perform wind retrieval, ambiguity removal using the Two-Dimensional Variational Ambiguity Removal (2DVAR) method, and Multiple Solution Scheme (MSS). CWDP also supports the output with selected winds, such as the first rank or the closest rank wind, for research purpose. The output L2B format of CWDP is BUFR, which can be converted to NetCDF format. The content of the L2B output consists of wind vectors which represent surface winds within the ground swath organised in WVCs. There are four wind solutions in the output, which are ranked by their respective Maximum Likelihood Estimator (MLE) distance using the NOAA SeaWinds BUFR template, see [2] for the BUFR format. CWDP also needs Numerical Weather Prediction (NWP) model winds inputs as a first guess for the Ambiguity Removal step and they are provided in GRIB format from e.g. ECMWF.

1.2 CWDP workflow

Figure 1 is the general scheme of CWDP. The NWP model GRIB input gives land-sea mask, sea surface temperature and first guesses winds over the globe. The processing steps are:

1. Pre-processing. The input BUFR file is decoded and the radar backscatter (σ^0) values are written into the data structure of CWDP. Atmospheric attenuations are computed. The Ku-band radiation is attenuated by the atmosphere. The attenuation is based on a climatology of water vapor, determined as a function of location and time of the year [3]. A look-up table (LUT) with monthly climatological attenuations was provided by NOAA and it is delivered with CWDP in the file data/atm_attn_360_180_12.dat. The attenuations are the same as used for QuikSCAT. The attenuation correction is added to the beam $\sigma°$ in dB and the two-way nadir looking values (without incidence angle correction) are stored in the BUFR output data. The output file name is defined. Some quality control on the input data is done.

2. Collocation with NWP data. The NWP GRIB data are read and the values for land fraction, sea surface temperature and first guess winds are interpolated and stored along with the information in each WVC.

3. Inversion. The Normalized Radar Cross Sections ($\sigma°$) together with measurement geometries (azimuth angle and incidence angle) and beam parameters (frequency and polarization) are converted into a set of ambiguous wind vector solutions in this step using the Geophysical Model Function (GMF). Here the GMF NSCAT-4DS is used [4]. Details of the inversion approach can be found in [5]. The MLE is used to assign a probability to each wind vector. The normal scheme allows 4 solutions at most, but in the Multiple Solution Scheme (MSS) the maximum number of solutions is 144. References [6] and [7] give the inversion theory and the description of the

full probability density function of the vector wind (MSS).

4. Quality control. Solutions that lie far away from the GMF are likely to be contaminated by, e.g., sea ice or confused sea state. During Quality Control these solutions are identified and flagged.

5. Ambiguity Removal. This procedure identifies the most probable solution using some form of external information. In this step, the selected wind in the output can be defined as 2DVAR solution, 1st rank solution, or closest rank solution. CWDP uses a two-Dimensional VARiational scheme (2DVAR) with no MSS as default (MSS needs to be set if required). A cost function is minimized that consists of a background wind field and all solutions with their probability, using meteorological balance, mass conservation and continuity as constraints. For ambiguity removal, MSS mode, 1st rank mode, and closest rank mode are all available in CWDP. Further details on the algorithms of 2DVAR can be found in [8], [9], and [10].

6. Quality Monitoring. The last step is to output quality indicators to an ASCII monitoring file and to write the results in a BUFR format output file. It was originally designed for the SeaWinds wind product. Detail of the monitoring flag can be found in [9]. Steps 2 and 6 of the processing chains are rather trivial. The real work is done in steps 1, 3, 4, and 5.

The L2A file format specification can be found in [2], and the specification of CWDP wind retrieval product can be found in section 2 of this document.

CWDP is based on generic genscat routines. Genscat is the base for all scatterometer wind processors within the OSI SAF. It contains a library of generic code for all scatterometers: wind inversion, ambiguity removal, support routines (read and write BUFR, GRIB, HDF5, NetCDF, etc.), low-level routines (date and time handling routine, error handler, numerical conversions, random number generation, etc.). These routines have been discussed in [5]. Genscat is delivered together with the CWDP processor.

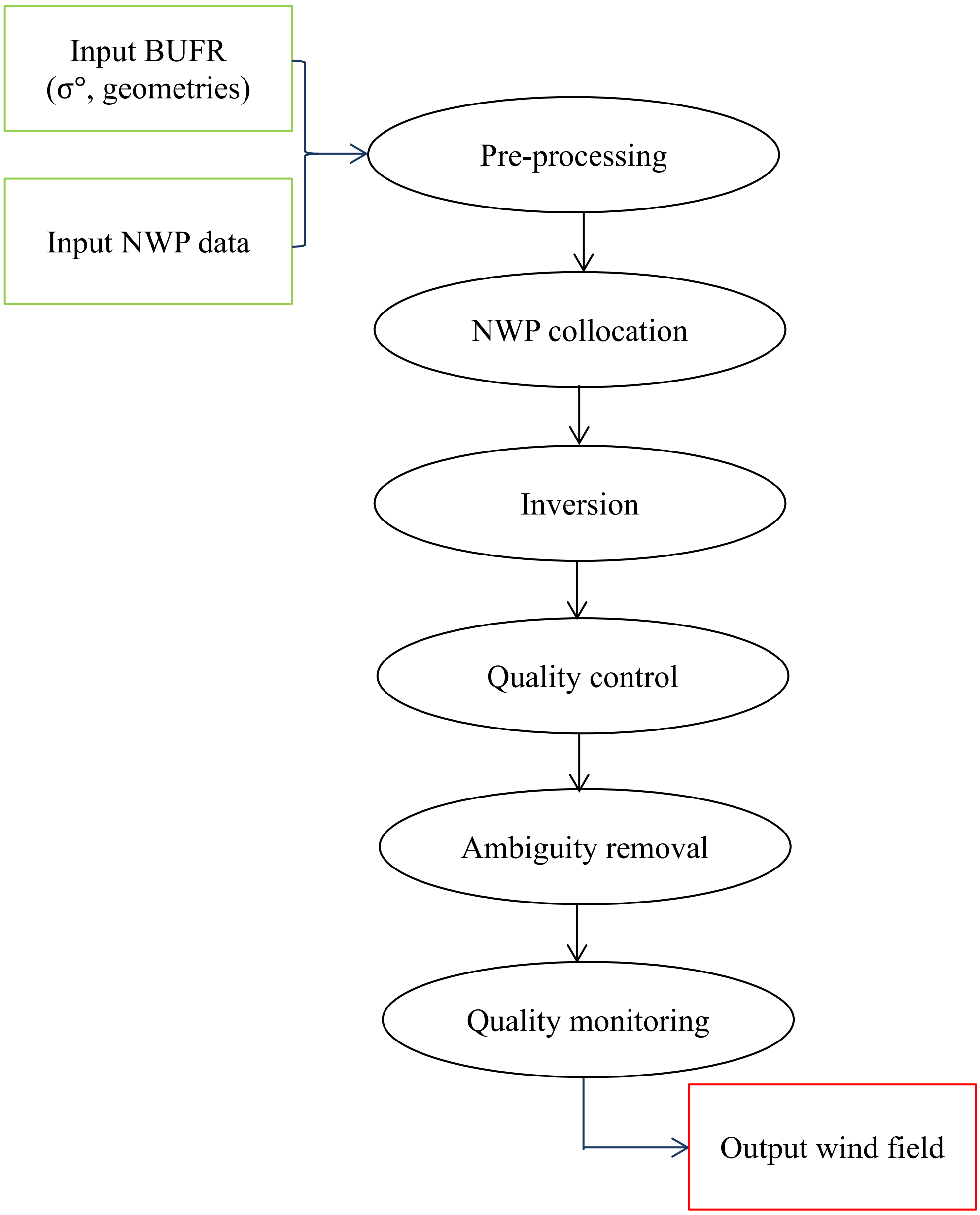

Figure 1 CWDP processing scheme.

1.3 Conventions

Names of physical quantities (e.g., wind speed components *u* and *v*), modules (e.g. *BufrMod*), subroutines and identifiers are printed italic.

Names of directories and subdirectories (e.g. `cwdp/src`), files (e.g. `cwdp.F90`), and commands (e.g. `cwdp -f input`) are printed in Courier. Software systems in general are addressed using the normal font (e.g. CWDP, genscat).

Hyperlinks are printed in blue and underlined (e.g. https://scatterometer.knmi.nl/).

2. CWDP output wind product specification

The CWDP input product is CFOSAT L2A data, and its specification is in [2]. This section is the specification of the CWDP output wind product, which has two formats – BUFR and NetCDF.

2.1 Output wind product specification (L2B)

The output wind product is generated in BUFR format which can be converted into NetCDF format afterwards. The retrieved wind vectors are the instantaneous mean surface winds at 10 m height, arranged in WVCs with resolution of 25×25 km^2 (this resolution is defined in the L1B simulation and it can be changed to 50×50 km^2). The output wind product contains up to 4 ambiguities (solutions) for every WVC. The wind solutions are ordered by decreasing probability and the data contains an index which points to the selected wind vector after ambiguity removal. The BUFR format is described in the appendix in [2].

NetCDF output description is in Table 1:

Table 1 Output wind product NetCDF format description.

NetCDF format is organized in (row_no, wvc_no, no_ambig).

No. of rows = maximum number of rows in one orbit

No. of WVCs = maximum number of WVCs in one row

(Note: the number of WVCs is defined as 48 in L2A data which might have some empty WVCs, and here the 'No. of WVCs' is the actual maximum number of WVCs, the WVCs without data are excluded.)

No. of ambig = maximum number of ambiguities in one WVC

PARAMETER	DESCRIPTION	UNIT	TYPE	DIM1	DIM2	DIM3
WVC_row_time	The time when satellite pass by a row.	-	Character	No. of rows	No. of WVCs	-

lat	The latitude of the center of a WVC in a row.	Deg	Float32	No. of rows	No. of WVCs	-
lon	The longitude of the center of a WVC in a row.	Deg	Float32	No. of rows	No. of WVCs	-
wvc_index	The WVC index of a WVC in a row.	-	Integer	No. of rows	No. of WVCs	-
nr_ambigs	Number of ambiguities of a WVC in a row	-	Integer	No. of rows	No. of WVCs	-
selected_ambigs	The index of selected ambiguities of a WVC	-	Integer	No. of rows	No. of WVCs	-
model_speed	Model speed of a WVC in a row	m/s	Float32	No. of rows	No. of WVCs	-
model_dir	Model direction of a WVC in a row	Deg	Float32	No. of rows	No. of WVCs	-
wvc_quality_flag	WVC quality flags	-	Integer	No. of rows	No. of WVCs	-
selected_wind_speed	Selected wind speed	m/s	Float32	No. of rows	No. of WVCs	-
selected_wind_direction	Selected wind direction	Deg	Float32	No. of rows	No. of WVCs	-
bs_distance	The MLE cone distance of selected wind solution	-	Float32	No. of rows	No. of WVCs	-
wind_speed_ambigs	Wind speed of up to 4 ambiguities	m/s	Float32	No. of rows	No. of WVCs	No. of ambig
wind_dir_ambigs	Wind direction of up to 4 ambiguities	Deg	Float32	No. of rows	No. of WVCs	No. of ambig
ambigs_distance	The MLE cone distance of up to 4 ambiguities	-	Float32	No. of rows	No. of WVCs	No. of ambig

2.2 System requirements

CWDP is designed for Unix (Linux) based computer platforms with a Fortran compiler and a C compiler (Table 2). CMake version 3.6 or higher (https://cmake.org/) is required in order to compile the ECMWF ecCodes library. Also, some of the scripts in the CWDP package are written in ksh (Korn shell scripting) and bash (Bash shell scripting), hence ksh and bash need to be installed as well.

To install the CWDP package, approximately 1G of disk space is needed. The output needs about 17 MB disk space to store one orbit of BUFR output data and 5 MB disk space to store one orbit of NetCDF output data. For near-real time processing, keeping a rolling archive of a couple of days, typically 20 GB of disk space is needed. A minimum of 16 GB RAM and a minimum CPU speed of 3.5 GHz are recommended.

Most scatterometer wind processors for ASCAT, OSCAT, and HSCAT instruments are used in the OSI SAF near real time wind processing at KNMI where one orbit input file should be processed and dissemination within typically 10 minutes. To allow some margin we request the complete CFOSAT processing (L1B NetCDF -> L2A HDF5, L2A HDF5 -> L2A BUFR, L2A BUFR -> L2B BUFR) to be done within 5 minutes. This is subdivided in

- 90 seconds for L1B NetCDF -> L2A HDF5 (cfosat_l1b_l2a)
- 30 seconds for L2A HDF5 -> L2A BUFR (cfosat_hdf2bufr)
- 180 seconds for L2A BUFR -> L2B BUFR (cwdp)

These times need to be feasible on a system having the minimal hardware requirements for CPU speed and RAM.

Table 2 Platforms and compilers on which CWDP has been tested.

Platform	**Fortran compiler**	**C compiler**
Fedora workstation Linux	GNU gfortran	GNU gcc
Fedora workstation Linux	Portland compiler	GNU gcc
Virtual server Red Hat Enterprise Linux	GNU gfortran	GNU gcc

3. Installing CWDP

CWDP is written in Fortran 90 (with a few low-level modules in C) and is designed to run on a computer system under Linux or Unix. It needs a Fortran 90 compiler and a C compiler for installation. CWDP comes along with a complete make system for compilation. When compiled, CWDP requires about 180-300 MB disk space and sufficient disk space and write permissions should be available.

L2A BUFR data are needed as input to run CWDP, so the L2A processor also needs to be installed in the same directory. The installation of L2A processor is described in [2].

The `genscat` for CFOSAT is delivered together with `CWDP`.

1. Copy the software package (`CWDP<version>.tar.gz`) in the chosen directory from which CWDP is to be run. After unzipping and untarring, the CWDP package is extracted into subdirectories `cwdp` and `genscat`, which are located at the directory where the tar file was located. The CWDP package includes `genscat`, the L2A converter (`cfosat_l1b_l2a`), the BUFR converter (`cfosat_hdf2bufr`) and CWDP (`cwdp`).
2. Go to directory `genscat/`:
   a) `make clean`
   b) set compiler options at the command line by:
   `. ./use_gfortran.bsh` (type 'dot' 'space' 'dot' 'slash' before the script name)
   `./Set_Makeoptions`
   Several other compilers can be chosen such as `use_pgf90.bsh`, etc.
   c) `make`
3. Go the directory `cwdp/` and type `make`
   The '`Makefile`' in this directory is for the entire software package, which is used to generate executable files (`cfosat_l1b_l2a, cfosat_hdf2bufr,`

cwdp).

cfosat_l1b_l2a is the directory for L2A processor.

cfosat_hdf2bufr is the directory for BUFR converter.

cwdp is the directory for wind retrieval.

Now CWDP is ready for use.

4. In the directory cwdp/test_data/, run script test_cwdp, a BUFR output file containing CFOSAT winds should be generated from the BUFR input file and the ECMWF GRIB data in this directory.
   L2A BUFR data can be converted to NetCDF format with
   /genscat/tools/bufr2nc_cfosat_L2A/Bufr2Nc
   L2B BUFR data can be converted to NetCDF format with
   /genscat/tools/bufr2nc_cfosat_L2B/Bufr2Nc

The executable files cfosat_l1b_l2a, cfosat_hdf2bufr, cwdp can be (re)compiled separately by changing directory into cwdp/l1b_l2a, cwdp/hdf2bufr, or cwdp/src respectively and typing make clean followed by make in these directories.

3.1 Directories and files

All the codes of CWDP are stored in a file named as CWDP.tar.gz. This file should be placed in the directory from which CWDP is to be run. After unzipping and untarring, the CWDP package is extracted into subdirectories cwdp and genscat, which are located at the directory where the tar file was located.

Subdirectories cwdp and genscat each contain some files and subdirectories, see Table 4 and Table 5. There will be several directories created during the process for storing output. After compilation, object codes of modules and routines will be generated after compilation.

Instructions for the installation of ecCodes library and HDF5 library can be found in [1].

Table 3 Contents of the directory cwdp.

Name	Contents

Name	Contents
execs	Link to CWDP executables, shell script for running cfosat_l1b_l2a, cfosat_hdf2bufr, cwdp, shell script for converting BUFR to NetCDF.
data	Look-up tables and antenna gain file needed in several processing steps
l1b_l2a	Code for converting L1B to L2A data (L2A processor)
hdf2bufr	cfosat_hdf2bufr for converting HDF5 to BUFR for L2A data.
src	Source code for CWDP program and supporting routines
test_data	nwp_list_files ECMWF wind model data.

Table 4 Contents of directory genscat.

Name	Contents
ambrem	Ambiguity removal routines
ambrem/twodvar	KNMI 2DVAR ambiguity removal routines
icemodel	Ice screening routines
inversion	Inversion and quality control routines
support	General purpose routines sorted in subdirectories
support/BFGS	Minimization routines needed in 2DVAR
support/eccodes	BUFR and GRIB file handling routines
support/Compiler_Features	Compiler specific routines, mainly command line handling
support/convert	Conversion between wind speed/direction and *u* and *v*
support/datetime	Date and time conversion routines
support/ErrorHandler	Error handling routines
support/file	File handling routines
support/hdf5	HDF5 handling routines
support/netcdf	NetCDF handling routines
support/num	Numerical definitions and number handling routines
support/singletonfft	FFT routines needed in minimization
support/sort	Sorting routines
tools/bufr2nc_cfosat_L2B	Convert BUFR to NetCDF for L2B data
tools/bufr2nc_cfosat_L2A	Convert BUFR to NetCDF for L2A data

3.2 Identify compiler

Some environment variables for identifying the compiler need to be set before compilation. The list of these variables is listed in Table 6. They can be set by using one of the use_*script in genscat. Further information of this part is in [1] (section

2.5).

Table 5 Environment variables for compilation and linking.

Variable	Function
`$GENSCAT_F77`	Reference to Fortran 77 compiler
`$GENSCAT_F90`	Reference to Fortran 90 compiler
`$GENSCAT_CC`	Reference to C compiler
`$GENSCAT_LINK`	Reference to linker for Fortran objects
`$GENSCAT_CLINK`	Reference to linker for C objects
`$GENSCAT_SHLINK`	Reference to linker for shared objects

Table 6 Properties of the use_*scripts.

Script	Fortran compiler	C compiler	Remarks
`use_g95`	`g95`	`gcc`	GNU compilers by Andy Vaught
`use_gfortran`	`gfortran`	`gcc`	GNU-GCC compiler collection
`use_ifort`	`ifort`	`icc`	Intel Fortran and C compilers
`use_pgf90`	`pgf90`	`gcc`	Portland Fortran compiler

3.3 Environment variables

A number of environment variables need to be set for CWDP (Table 8). Note that CFOSAT BUFR data are written in a specific BUFR format for CFOSAT. It is part of the official WMO BUFR Table D list of common sequences.

The `$GRIB_DEFINITION_PATH` variable is necessary for a proper functioning of the GRIB decoding software.

The variables `$LUT_FILENAME_KU_HH` and `$LUT_FILENAME_KU_VV` point CWDP to the correct binary Ku band GMF lookup tables (LUT) at HH and VV polarization, respectively. They should contain a file name including a valid path. NSCAT lookup tables are delivered with CWDP in big endian and little endian binary formats, the `<platform>` part in the paths should be set to `big_endian` or `little_endian` depending on your computer platform type.

The variable `$LUTSDIR` points CWDP to a directory containing some look up tables that are used to normalize the inversion residuals and to compute atmospheric attenuations for the Ku band radar data. The necessary tables are delivered with CWDP.

Table 7 Environment variables for CWDP.

Name	Value
$LUT_FILENAME_KU_HH	cwdp/data/little_endian/nscat4ds_250_73_51_hh.dat cwdp/data/big_endian/nscat4ds_250_73_51_hh.dat
$LUT_FILENAME_KU_VV	cwdp/data/little_endian/nscat4ds_250_73_51_vv.dat cwdp/data/big_endian/nscat4ds_250_73_51_vv.dat
$GRIB_DEFINITION_PATH	genscat/support/eccodes/definitions
$LUTSDIR	cwdp/data

4. Processing details

4.1 Command line options

The command line options for the L2A processor have been described in [2]. The CWDP main program is started from directory `cwdp/src` with command

`cwdp -f <input file> [-nwpfl <file>] [options/modes]`

with < > indicating obligatory input, and [ ] indicating non-obligatory input. The following command line options are available.

`-f <input file>`	Process a BUFR input file (L2A data) with name `input file`. BUFR files can be combined into one BUFR with `cat` command.
`-nwpfl <file>`	Read in a list of GRIB files in the file named `file`. Those GRIB files include land/sea GRIB file and SST GRIB file.

Some optional options:

`-filter`	Switch on to exclude the side WVCs of the swath due to their poor wind retrieval result. If it switched on, then only WVCs from number 2 to 41 (42 WVCs per row in total) are included in the retrieval. The WVCs at the most outer of the swath are excluded.
`-noinv`	Switch off inversion (default is switched on).
`-icemodel`	Switch on ice screening (default is switched off). This is not implemented yet for the current CWDP version. Use SST data to screen for ice.
`-noamb`	Switch off ambiguity removal (default is switched on).
`-nowrite`	Do not produce BUFR output (default is switched on).

-noc Perform σ^0 calibration (NWP Ocean Calibration, default is switched off). A calibration of the σ^0 values is performed, i.e., the backscatter values are changed in order to obtain better calibrated winds.

-mss Use the Multiple Solution Scheme for Ambiguity Removal. MSS is switched off as default. Note that in the NOAA BUFR format, only 4 out of these 144 wind solutions will be written.

-armeth <meth> Choose ambiguity removal method. Valid methods are: `1strank` - the wind solution with the lowest distance to the GMF (residual) is selected, `bgclosest` - the wind solution closest to the background model wind is selected, `2dvar` - 2DVAR. The default is `2dvar`.

-mon Switch on the monitoring function. The monitoring results are written in an ASCII file with the name `<name of BUFR output file>.mon`. By default, no monitoring file is produced.

-verbosity <L> Set the verbosity level to L (default is 0). If the verbosity level is -1 or smaller, no output is written to the standard output except error messages. If the verbosity level equals 0 only some top level processing information is written to output. If the verbosity level is 1 or greater, also additional information is given.

-mononly Write the monitoring file without any processing.

-properties Write some properties of the last row of the input file. The data acquisition date and time of the last row are written to a small ASCII output file `properties.txt`.

-writeonly Write all data to BUFR output without processing. This mode is useful to copy an input file to BUFR output without processing.

The output will be written into BUFR file with the following name definition: `INSTR_YYYYMMDD_HHMMSS_SAT_ORBIT_srv_o_SMPL_ovw_l2.bufr`, where

- INSTR is the instrument, here is rfsca.
- YYYYMMDD_HHMMSS is the acquisition date and time (UTC) of the first data in the file.
- SAT is the satellite (6 characters), here it is cfosat.
- ORBIT is the orbit number (5 digits) of the first data in the file.
- SMPL is the WVC sampling (cell spacing), 250 for 25 km.
- If the above convention results in identical input and output file names, the extension '~' is added to the output file name.

Example output file name:

rfsca_20190827_163348_cfosat_04583_o_250_ovw_l2.bufr

4.2 Test data and test programs

Directory cwdp/execs contains the script cwdp_run which sets all necessary environment variables and then run the cwdp executable with all command line options provided to the script.

Directory cwdp/test_data contains ECMWF GRIB files and a CFOSAT level 2a file which can be used to test the wind retrieval. The script test_cwdp will run CWDP to accomplish this.

Directory cwdp/execs also contains an example script called cwdp_l1b_l2a_retrieval_deliver. This can be used to run the whole chain of wind processing from L1B NetCDF input data, via L2A HDF data and L2A BUFR data to L2B BUFR output data. GRIB files from a Numerical Weather Prediction model like ECMWF containing forecast winds are needed as well. The directories which are set in the script need to be adapted to your local environment.

Figure 2 shows a test run of wind retrieval with NOC of all the WVCs with 25 km resolution. Figure 3 is the same wind retrieval result as in Figure 2 but compared with ECMWF wind data, which shows good statistics.

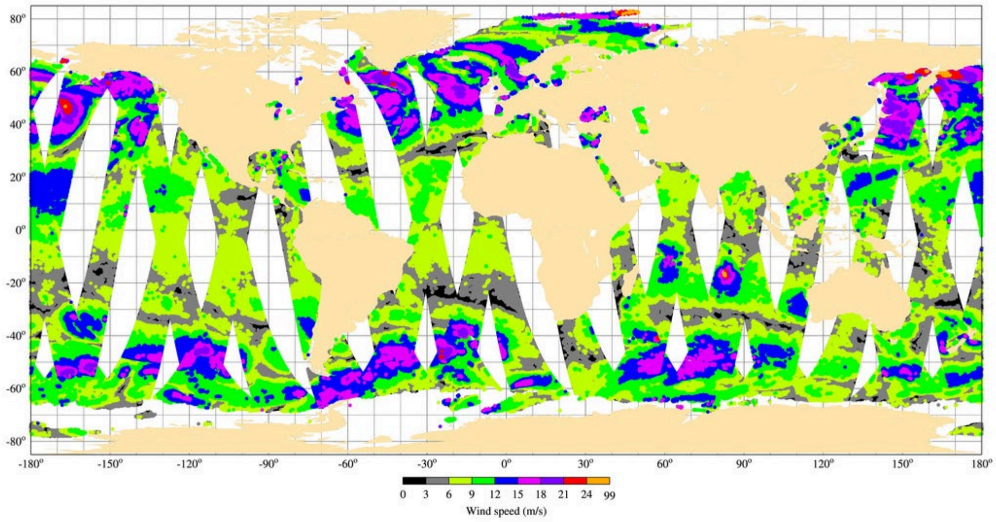

Figure 2 Global coverage test of one day (20181219), Wind retrieval result (2DVAR), 25 km product with all WVCs.

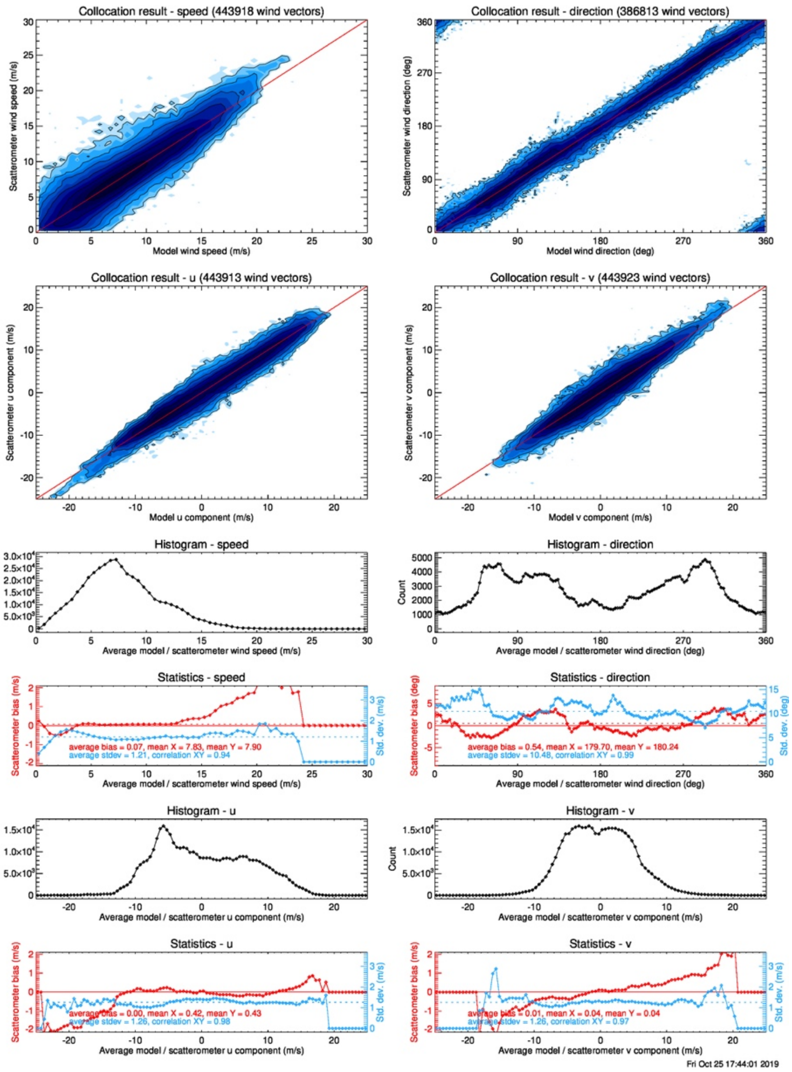

Figure 3 2DVAR wind solution (one day 20181219) with all WVCs.

4.3 Documentation

Directory `cwdp/doc` contains documentation of CWDP. Other information can be found in the comments in scripts, Makefiles and source codes.

References

[1] A. Verhoef, J. Vogelzang, J. Verspeek, and A. Stoffelen, “PenWP User Manual and Reference Guide,” de Bilt, the Netherlands, 2017.

[2] Z. Li, A. Verhoef, and A. Stoffelen, “CWDP L2A processor Specification and User Manual,” SAF/OSI/CDOP3/KNMI/TEC/MA/319, 2021.

[3] F. J. Wentz, “Final Delivery: Climatology of 14-GHz Atmospheric Attenuation,” Pasadena, 1996.

[4] See https://scatterometer.knmi.nl/nscat_gmf/ and references thereon

[5] A. Verhoef, J. Vogelzang, J. Verspeek, and A. Stoffelen, “PenWP Top Level Design”, 2017.

[6] A. Stoffelen and M. Portabella, “On Bayesian Scatterometer Wind Inversion,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1–11, 2006.

[7] M. Portabella and A. Stoffelen, “A probabilistic approach for SeaWinds data assimilation A probabilistic approach for SeaWinds data assimilation 1 A probabilistic approach for SeaWinds data assimilation,” *Met. Soc*, vol. 130, no. 596, pp. 127–152, 2004.

[8] J. Vogelzang, “Two-dimensional variational ambiguity removal (2DVAR),” de Bilt, the Netherlands, 2013.

[9] J. de Vries, A. Stoffelen, and J. Beysens, “Ambiguity Removal and Product Monitoring for SeaWinds,” de Bilt, the Netherlands, 2005.

[10] J. Vogelzang, “2D variational ambiguity removal,” de Bilt, the Netherlands, 2013.

Appendix A: Acronyms

2DVAR	2 Dimensional Variational ambiguity removal (2DVAR)
BUFR	Binary Universal Form for the Representation of data
CFOSAT	China-France Oceanography SATellite
CWDP	CFOSAT Wind Data Processor
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
genscat	generic scatterometer software routines
GMF	Geophysical Model Function
HDF5	Hierarchical Data Format version 5
L1B	Level 1-B
L2A	Level 2-A
L2B	Level 2-B
LUT	Look-up Table
MSS	Multiple Solution Scheme
NetCDF	Network Common Data Form
NWP	Numerical Weather Predictio
PenWP	Pencil beam Wind Processor
2DVAR	Two-Dimensional Variational Ambiguity Removal
WVC	Wind Vector Cell

Appendix B: Third party software

Directory `genscat/support/hdf5/hdfgroup` contains the Hierarchical Data Format (HDF) library which was downloaded from http://www.hdfgroup.org/. This software is copyright © The HDF Group. The library is automatically compiled with genscat.

Directory `genscat/support/netcdf/unidata` contains the Network Common Data Form (NetCDF) library which was downloaded from http://www.unidata.ucar.edu/. This software is copyright © UCAR/Unidata. The library is automatically compiled with genscat.

Directory `genscat/support/eccodes/ecmwf` contains the ECMWF ecCodes library which was downloaded from https://www.ecmwf.int. This software is copyright © ECMWF. The library is automatically compiled with genscat.