Open source PySAMOSA SAR Satellite altimetry retracking software

PySAMOSA
SAR Altimetry Retracking

Florian Schlembach, Marcello Passaro*

* Deutsches Geodätisches Forschungsinstitut (DGFI-TUM), Technical University of Munich

OSTST 2023
Forum-only, Puerto Rico,7th-11th of November 2023

Motivation

- To date no open source software implementation is publicly available for L2 processing of satellite altimetry data of the Sentinel-3 and Sentinel-6 MF mission
- No reference implementation of the EUMETSAT baseline product is available
- Efforts to improve the baseline L2 processing requires a reimplementation of the baseline processing chain, which is never the same as the baseline

Objectives of the publication of the PySAMOSA software

- Allow other researcher
  - To retrack data by their own
  - Extend the baseline processor
  - To reproduce results from the literature, e.g. [3,4,5]
- Support open science

PySAMOSA

- Is a Python-based software for (retracking)

processing **open ocean** and **coastal** waveforms from SAR satellite altimetry

- Measures sea surface heights, wave heights, and wind speed for the oceans and inland waters
- Uses the SAMOSA2 model as described in [1,2]
- Supports the Sentinel-3 and Sentinel-6 MF mission
- Retracks Level-1b (L1b) data to extract the retracked variables SWH, range, and Pu
- Both open ocean and coastal waveforms can be retracked
- Available with a LGPLv3 open source license

Retracking **open ocean** waveforms is based on

- The specification documents from the official EUMETSAT baseline (S3 [1], S6-MF [2]).

For coastal waveforms, the two retracking algorithms are implemented:

- SAMOSA+ [3]
- CORAL [4,5]

In addition, FF-SAR-processed S6-MF data can be retracked using the zero-Doppler beam of the SAMOSA2 model and a specially adapted α_p LUT table, created by the ESA L2 GPP project [7]. The application of the FF-SAR-processed data has been validated in [5].

PySAMOSA

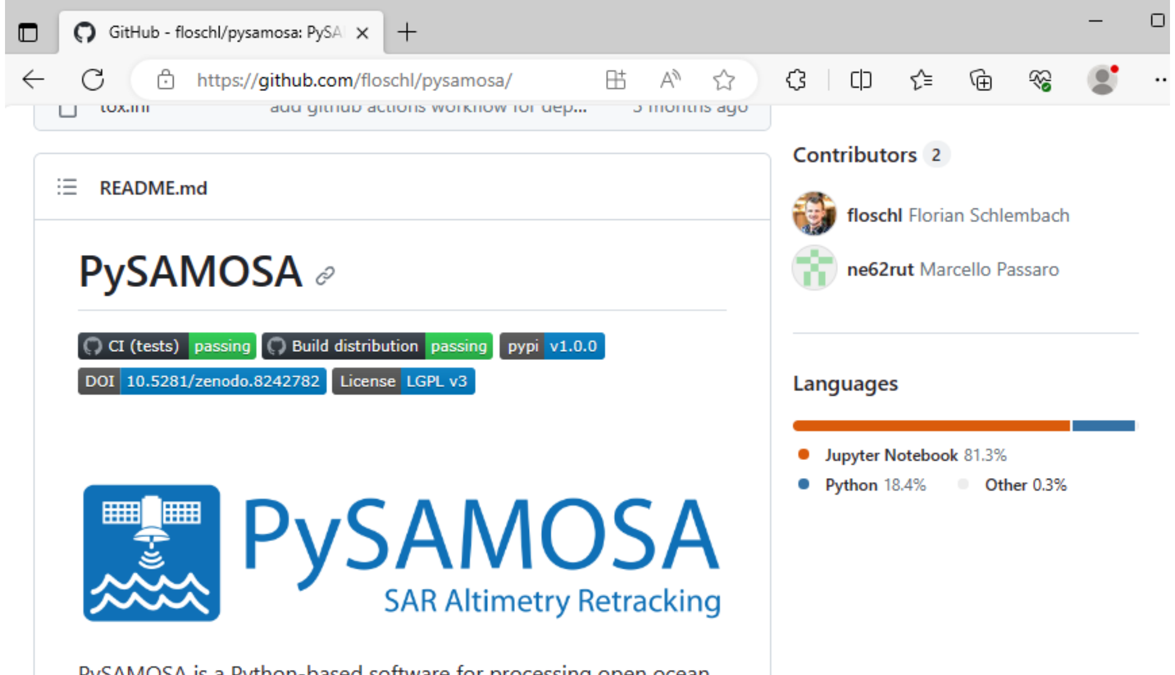
Available and easy-to-use as an official PyPI package

```
$ pip install pysamosa
```

- The source code of PySAMOSA can be found on GitHub

http://github.com/floschl/pysamosa

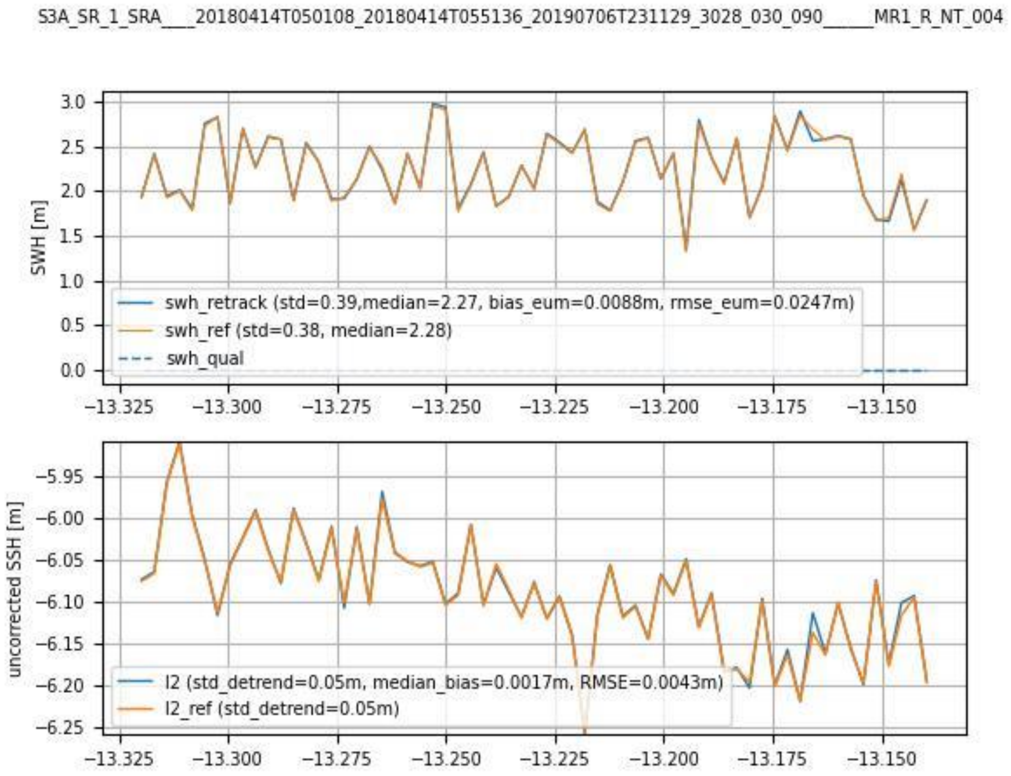
with a getting-started, tips, and a minimal-working code example and jupyter notebook to retrack open ocean data etc.



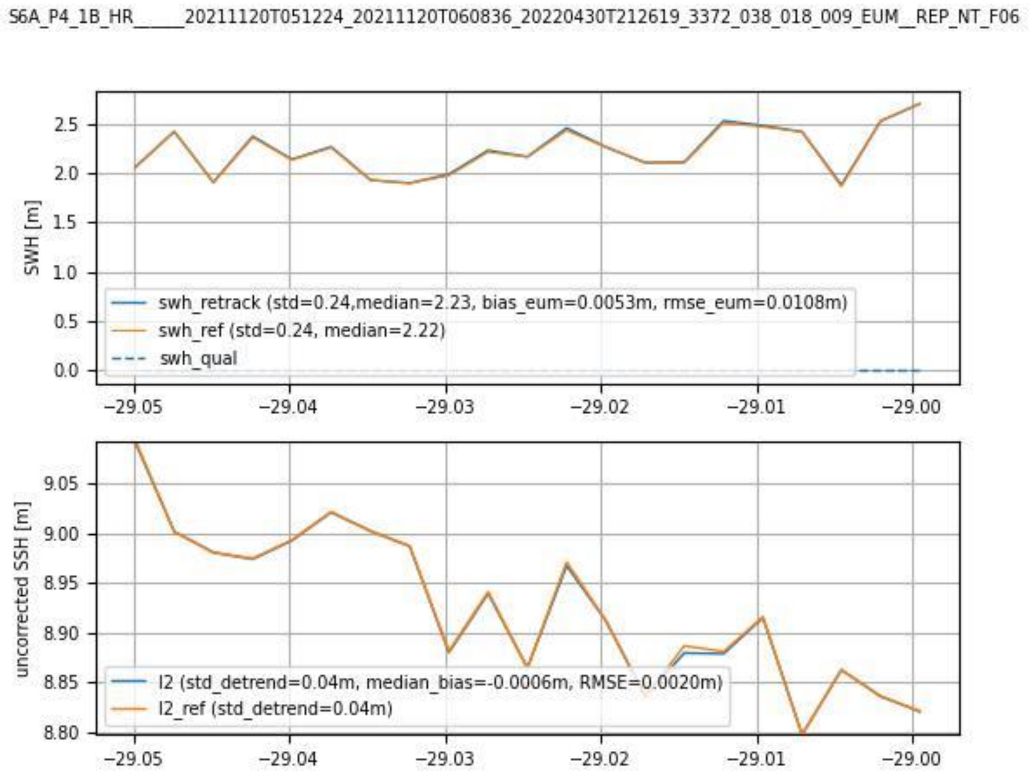
Validation

The implementation is compared against the EUMETSAT baseline and SAMOSA+ in [4,5].
Two short retracked open ocean segments (of the given code sample) for the Sentinel-3 and Sentinel-6 MF missions are shown here

Sentinel-3A



Sentinel-6 MF



Future Work

There are currently no plans from the authors to continue the development of PySAMOSA.

However, the publication of the software was done to support open science. It is intended to be a community-based project. Contributions to this project are very welcome and greatly appreciated.

Possible developments of this project are:

Retracking-related

- Align CS-2 retracking with the CS-2 baseline processing chain, validate against SAMpy developed as part of the ESA Cryo-TEMPO project
- Implement evolutions of the EUMETSAT's baseline processing chain [6], e.g. the numerical retracking planned for Q3/2023

Software-related

- Create notebook for a coastal retracking demo
- Create richer documentation (readthedocs)

Conclusion

- A first open source multi-mission SAR altimetry retracking processor was published
- PySAMOSA is now publicly available on GitHub

http://github.com/floschl/pysamosa

- or as an official PyPI package on pypi.org

https://pypi.org/project/pysamosa/

- With this release, we contribute to open science supporting the scientific research community. We welcome contributions to PySAMOSA.

PySAMOSA
SAR Altimetry Retracking

References

[1] SAMOSA Detailed Processing Model: Christine Gommenginger, Cristina Martin-Puig, Meric Srokosz, Marco Caparrini, Salvatore Dinardo, Bruno Lucas, Marco Restano, Américo, Ambrózio and Jérôme Benveniste, Detailed Processing Model of the Sentinel-3 SRAL SAR altimeter ocean waveform retracker, Version 2.5.2, 31 October 2017, Under ESA-ESRIN Contract No. 20698/07/I-LG (SAMOSA), Restricted access as defined in the Contract, Jérôme Benveniste (Jerome.Benvensite@esa.int) pers. comm.

[2] EUMETSAT. Sentinel-6/Jason-CS ALT Level 2 Product Generation Specification (L2 ALT PGS), Version V4D; 2022. https://www.eumetsat.int/media/48266.

[3] Dinardo, Salvatore. 'Techniques and Applications for Satellite SAR Altimetry over Water, Land and Ice'. Dissertation, Technische Universität, 2020. https://tuprints.ulb.tu-darmstadt.de/11343/.

[4] Schlembach, F.; Passaro, M.; Dettmering, D.; Bidlot, J.; Seitz, F. Interference-Sensitive Coastal SAR Altimetry Retracking Strategy for Measuring Significant Wave Height. Remote Sensing of Environment 2022, 274, 112968. https://doi.org/10.1016/j.rse.2022.112968.

[5] Schlembach, F.; Ehlers, F.; Kleinherenbrink, M.; Passaro, M.; Dettmering, D.; Seitz, F.; Slobbe, C. Benefits of Fully Focused SAR Altimetry to Coastal Wave Height Estimates: A Case Study in the North Sea. Remote Sensing of Environment 2023, 289, 113517. https://doi.org/10.1016/j.rse.2023.113517.

[6] Scharroo, R.; Martin-Puig, C.; Meloni, M.; Nogueira Loddo, C.; Grant, M.; Lucas, B. Sentinel-6 Products Status. Ocean Surface Topography Science Team (OSTST) meeting in Venice 2022. https://doi.org/10.24400/527896/a03-2022.3671.

[7] ESA L2 GPP Project: FF-SAR SAMOSA LUT generation was funded under ESA contract 4000118128/16/NL/AI.

Spare Slides

Minimal working example (1/2)

- From https://github.com/floschl/pysamosa/blob/main/README.md

```
from pathlib import Path
import numpy as np

from pysamosa.common_types import L1bSourceType
from pysamosa.data_access import data_vars_s3, data_vars_s6
from pysamosa.retracker_processor import RetrackerProcessor
from pysamosa.settings_manager import get_default_base_settings, SettingsPreset


l1b_files = []
# l1b_files.append(Path('S6A_P4_1B_HR______20211120T051224_20211120T060836_20220430T212619_3372_038_018_009_E
l1b_files.append(Path.cwd().parent / '.data' / 's6' / 'l1b' / 'S6A_P4_1B_HR______20211120T051224_20211120T060

l1b_src_type = L1bSourceType.EUM_S6_F06
data_vars = data_vars_s6

# configure coastal CORAL retracker
pres = SettingsPreset.CORALv2
rp_sets, retrack_sets, fitting_sets, wf_sets, sensor_sets = get_default_base_settings(settings_preset=pres, l

rp_sets.nc_dest_dir = l1b_files[0].parent / 'processed'
rp_sets.n_offset = 0
rp_sets.n_inds = 0  #0 means all
rp_sets.n_procs = 6  #use 6 cores in parallel
rp_sets.skip_if_exists = False
```

Minimal working example (2/2)

- From https://github.com/floschl/pysamosa/blob/main/README.md

```
additional_nc_attrs = {
    'L1B source type': l1b_src_type.value.upper(),
    'Retracker preset': pres.value.upper(),
}

rp = RetrackerProcessor(l1b_source=l1b_files, l1b_data_vars=data_vars['l1b'],
                        rp_sets=rp_sets,
                        retrack_sets=retrack_sets,
                        fitting_sets=fitting_sets,
                        wf_sets=wf_sets,
                        sensor_sets=sensor_sets,
                        nc_attrs_kw=additional_nc_attrs,
                        bbox=[np.array([-29.05, -29.00, 0, 360])],
                        )

rp.process()  #start processing

print(rp.output_l2)  #retracked L2 output can be found in here
```

A running minimal working example for retracking is shown in `notebooks/retracking_example.ipynb`.

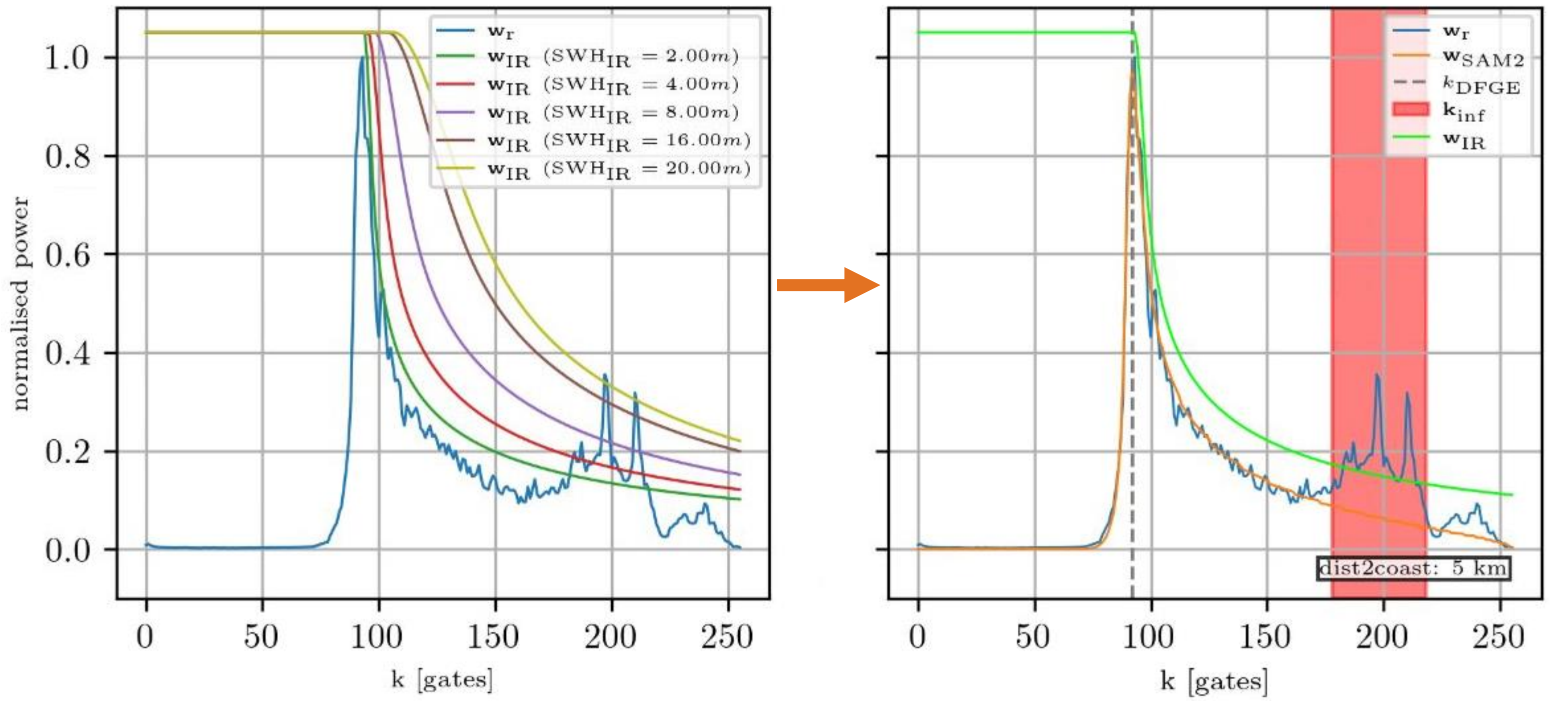
CORALv2 Coastal Retracker

Schlembach F., Passaro M., Dettmering D., Bidlot J., Seitz F.: **Interference-sensitive coastal SAR altimetry retracking strategy for measuring significant wave height**. Remote Sensing of Environment, 274, 112968, 10.1016/j.rse.2022.112968, 2022

Adaptive Interference Masking (AIM)

→ senses and masks interference within the trailing edge

Generation of a single-look SAMOSA model $\mathbf{w}_{\mathrm{SAM2}}$ to produce the interference reference waveform $\mathbf{w}_{\mathrm{IR}}(\mathrm{SWH}_{\mathrm{IR}})$



detected interference gates

$$\mathbf{k}_{\mathrm{inf}} = \mathrm{True}(\mathbf{w}_{\mathrm{r}} > \mathbf{w}_{\mathrm{IR}})$$

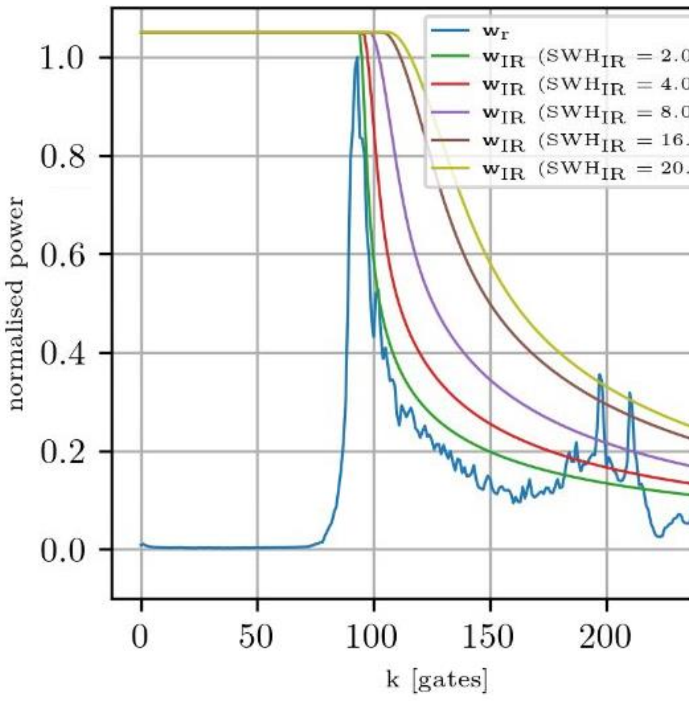
→ AIM detects interference gates and excludes them from fitting procedure → **quality** of SWH estimate is improved.

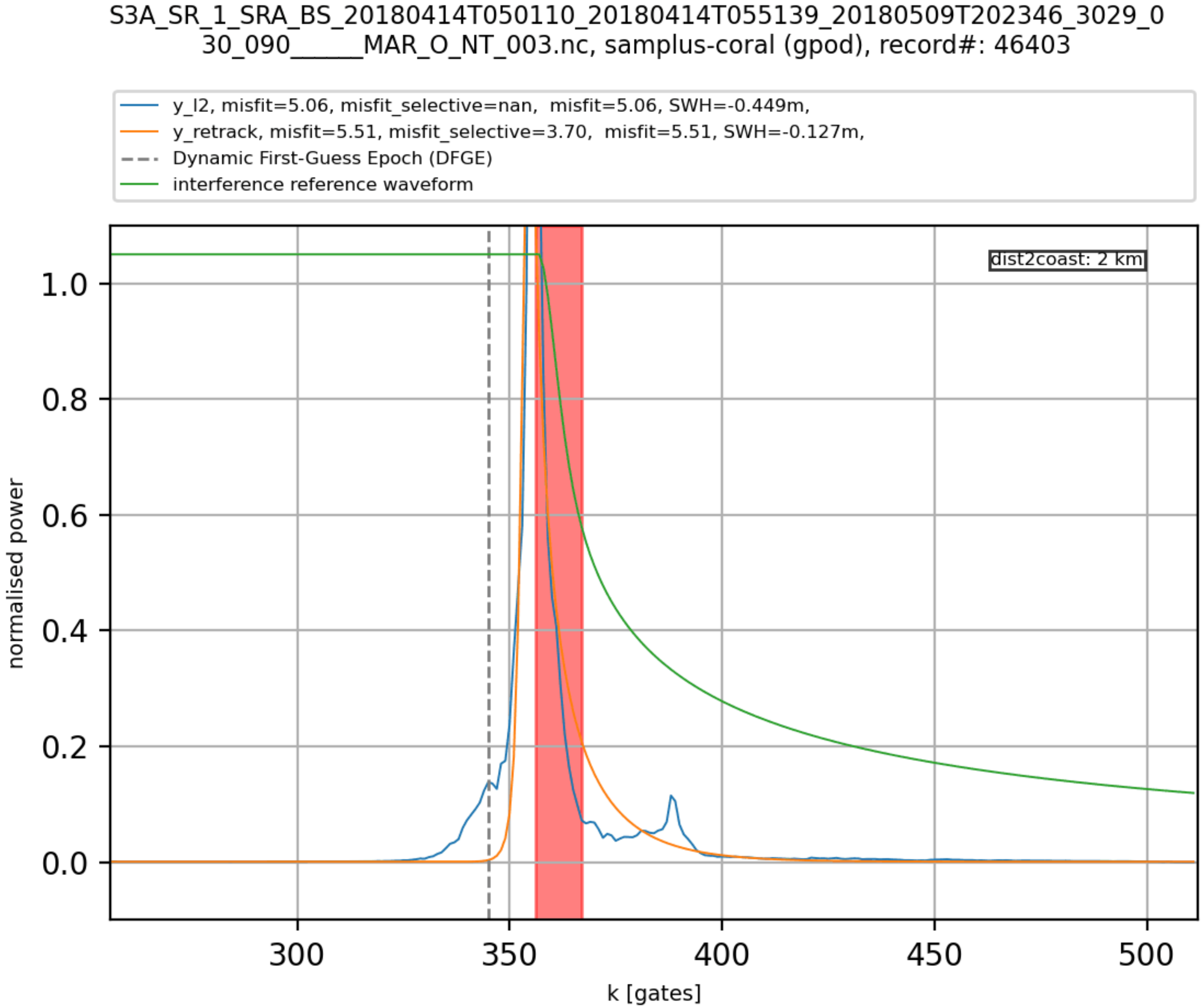
CORALv2 Coastal Retracker

Adaptive Interference Masking (AIM)

→ senses and masks interference within the

Generation of a single-look SAMOSA model $\mathbf{w}_{S}$

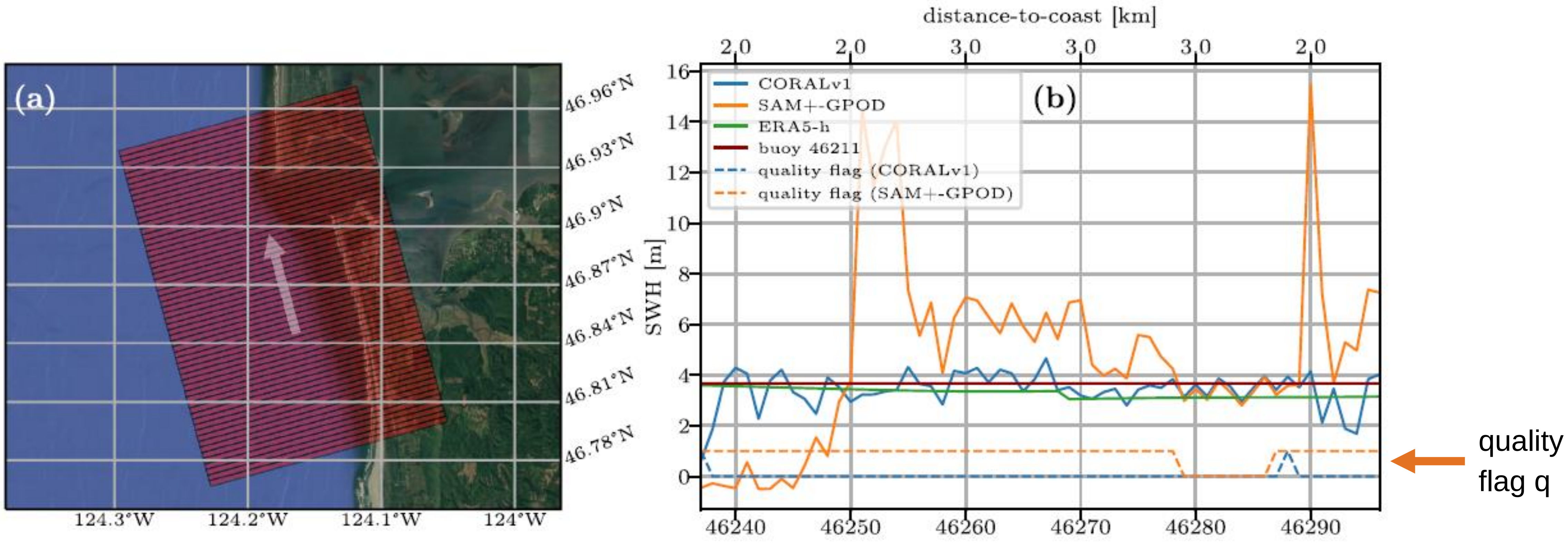




→ AIM detects interference gates and excludes

CORALv2 Coastal Retracker

Retracking waveforms with strong coastal interference by CORALv2 in comparison with SAMOSA+



→ **quantity** of the SWH estimates is increased significantly