The SDAP data assimilation platform in brief

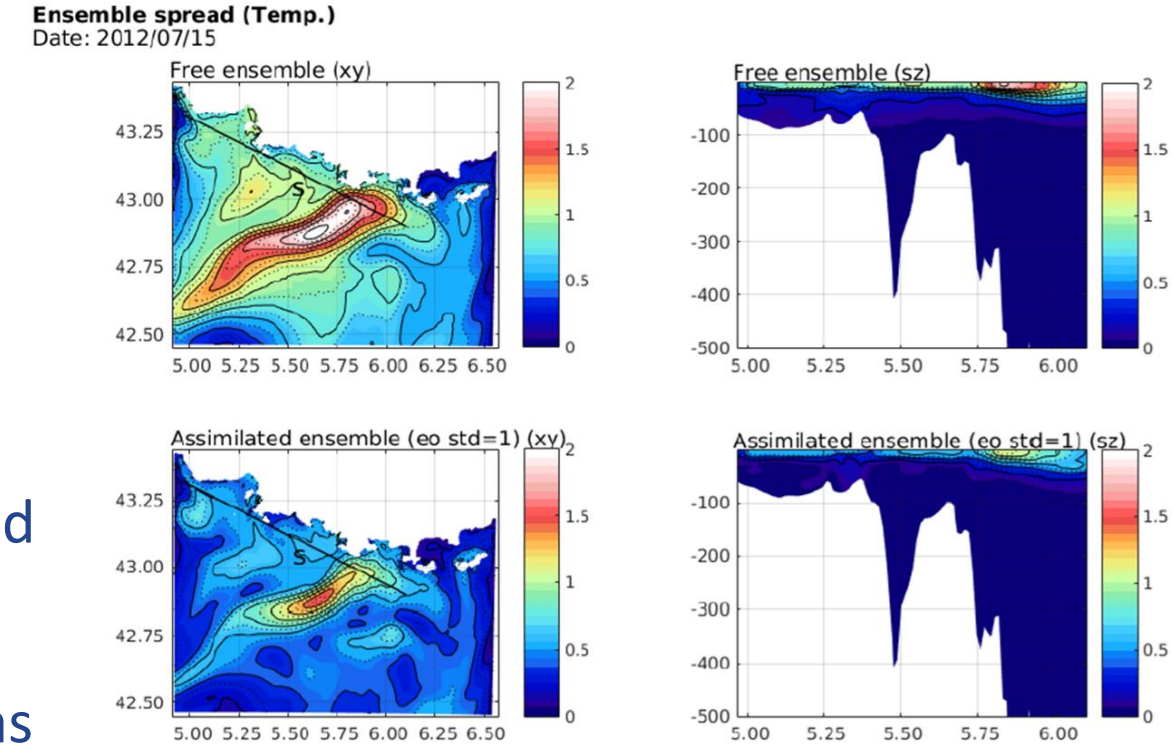
Pierre De Mey-Frémaux (Emeritus DR CNRS, LEGOS, Toulouse, France)
and collaborators

Talk DOI: 10.5281/zenodo.22829207

SIROCCO Day and Training Course
Toulouse, 21-25 September 2026

SDAP purpose and main features

- Provide a flexible platform to carry out multivariate assimilation of geophysical data in a numerical model, and interpret the results.
- The program is multi-geometry (finite differences or finite elements), multi-algebra (plug-in analysis kernels), multi-model (simple standardized interface) and multi-grid (e.g., parent + child).
- The program supports almost uniquely Ensemble assimilation approaches such as the Stochastic Ensemble Kalman Filter.
- The program has been integrated with the SYMPHONIE model, and results have been published; however, SYMPHONIE has evolved since then, and the changes are progressively being incorporated.



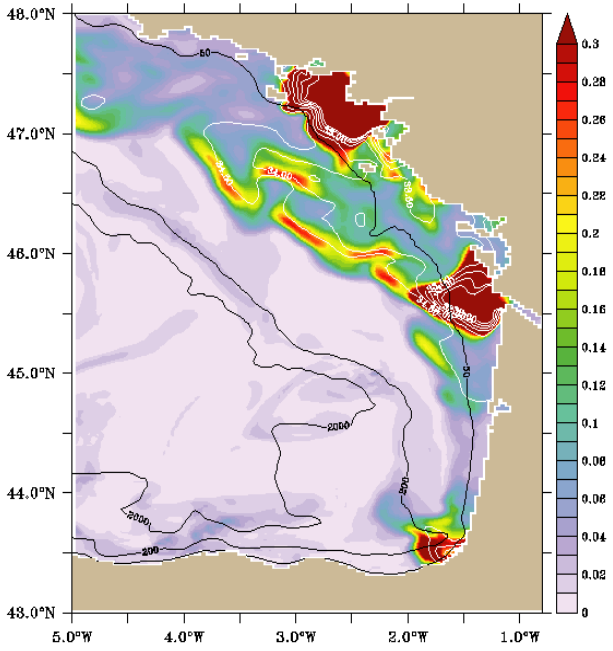
Assimilating SST in SYMPHONIE r208 (500m/40 levels) with a SEnKF in the Cassis upwelling area *(Han et al., 2020)*

CODE AI — free .io HUMAN MADE

Impact of SEnKF assimilation of SST on river plumes in Bay of Biscay

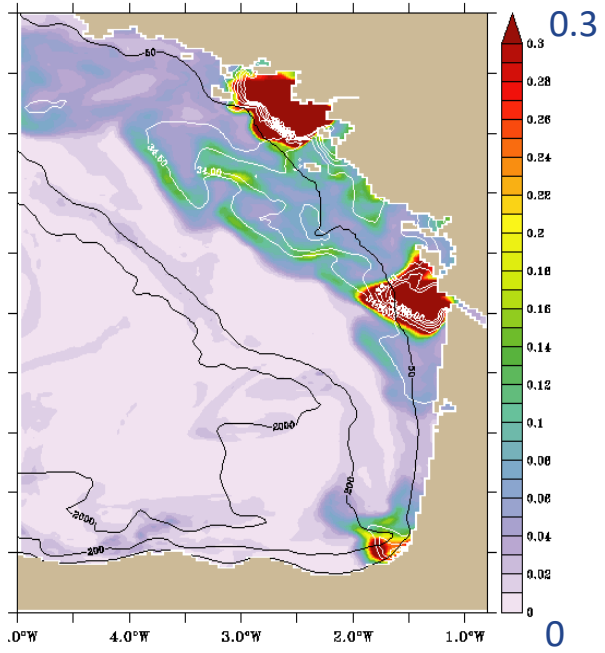
Assim(SST)
Ens. spread in SSS
before assimilation

March 20



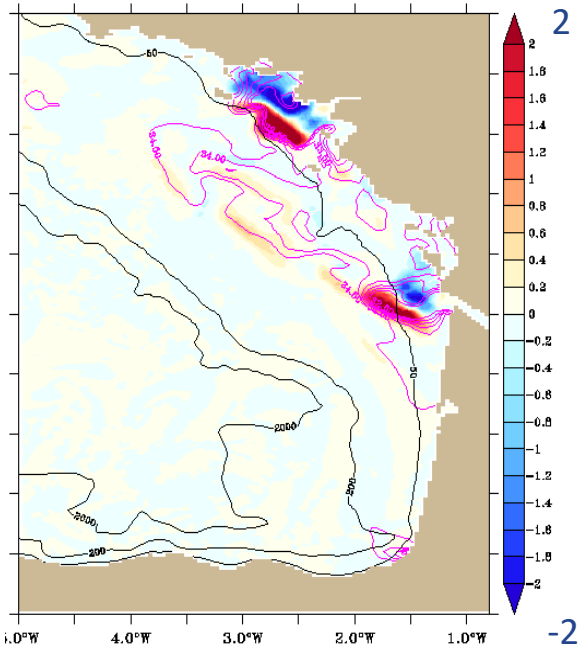
Assim(SST)
Ens. spread in SSS
after assimilation

March 20



Assim(SST)
Correction in SSS
Member #20

March 20



➔ Freshwater fronts: spread reduction by EnKF
➔ Freshwater river plumes: spread reduction + coastward displacement

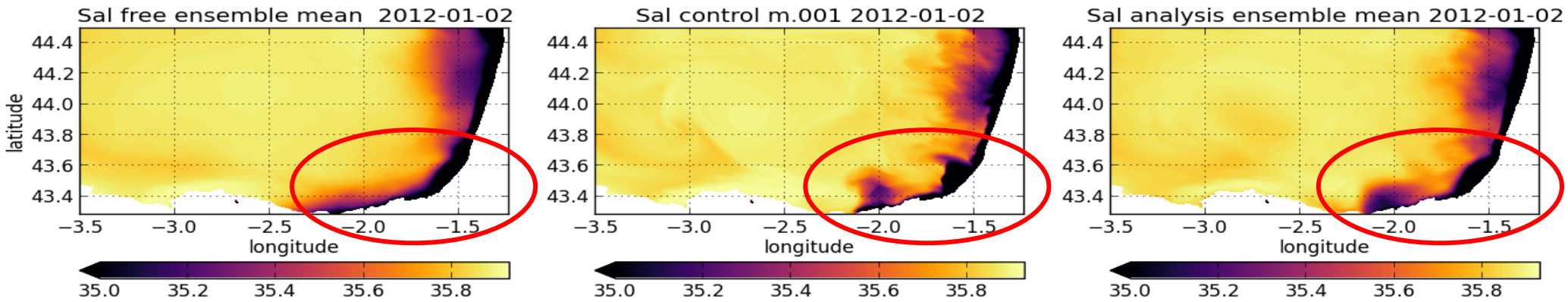
(Ayoub and De Mey-Frémaux, 2016)

Simultaneous downscaling and SEnKF assimilation (SST, SSH)

Simple downscaling from CMEMS IBI36

"Truth" (control member)

Assimilated solution



Landes Plateau SSS, 2 Jan 2012

Added value of assimilation:

➔ Improvement of salinity while it was not observed

➔ Small scales of internal shelf around Adour plume

➔ Improvement of mesoscale (hardly visible)

(Ghantous et al., 2019)

SDAP is more than just assimilation… Ensemble & probabilistic metrics available in diagnostic tools:

Metric	Function	SDAP 1.7 Implementation
State-space Ensemble spread f(range)	Calculate the Ensemble variance over a given grid.	`olax --nc=evar`
Multigrid representers and analysis	Calculate multigrid Ensemble representers; calculate an analyzed correction for assimilation with a EnKF step.	`olax --rep and --analysis`
State-space trends f(range)	Calculate the state-space trends at forecast ranges within the current cycle between the forecast (Ensemble average or specific member) and the initial state for a given grid. Optionally compare the trends with those from a previous run.	`scrumcat --sstrend`
Data-space basic statistics	Calculate statistics between Ensemble and noisified CONTROL_OBS or real observations for a given grid.	`scrumcat --basic`
Data-space differential grid statistics	Compare the Norm of the Innovation Vector (NIV) between all pairs of grids over the common part of the data vector.	`scrumcat --xgrid`
Norm of innovation vector (data-space)	Calculate the L2 distance between Ensemble average and CONTROL_OBS or real observations for a given grid.	`scrumcat --basic`
Data-space probabilistic skill score	Calculate the Continuous Ranked Probability Score between Ensemble and noisified CONTROL_OBS or real observations for a given grid.	`scrumcat --crps`
Data-space probabilistic binary score	Calculate the Brier Score of a binary event between Ensemble and noisified CONTROL_OBS or real observations for a given grid.	`scrumcat --brs`
Talagrand Diagrams (Rank Histograms; data-space)	Examine how the CONTROL_OBS or real observations project onto the classes of the Ensemble for a given grid.	`scrumcat --RH`
Array-space statistics (data-space and array-space)	Calculate empirical consistency analysis array-space criteria ArMCA1, ArMCA2.0, ArMCA2.1.	`scrumcat --ArMCA`

https://docs.google.com/document/d/1CRCVtBK3Z9uRXIOGHdUJDhvQrsESZtCQD4Hv2006z3c/edit?usp=drive_link

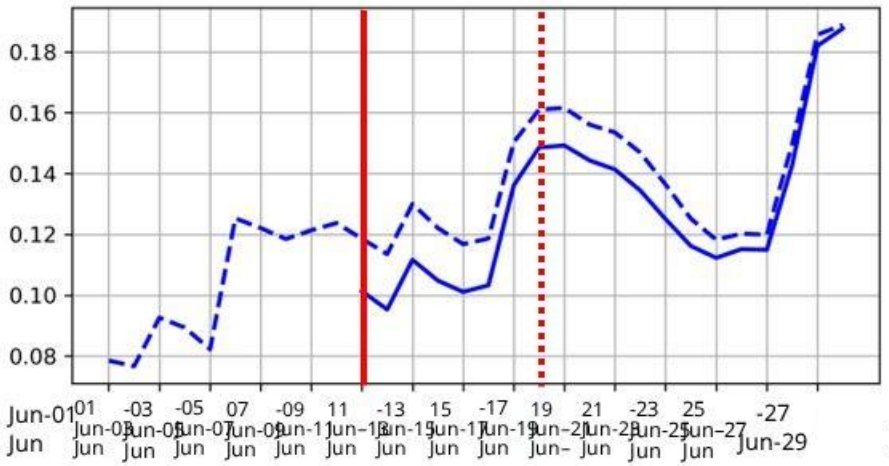
SLA assimilation (solid line) vs. free run (dashed line) – coastal upwelling, June 2017

Cantabrian/ Galician upwelling

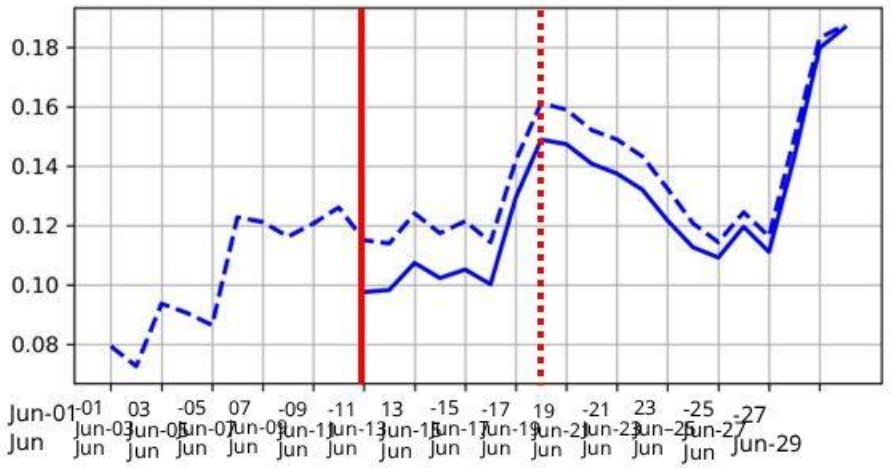
CRPS: Continuous Ranked Probability Score (lower = better)

Coastal strip

Parent SST CRPS



Child SST CRPS



Assimilation on June 11, 2017

- Two nested grids:
  - parent = regional 1/36°
  - child = coastal 1/108°
- Wind errors considered
- NEMO community model

Significant improvement in upwelling forecasts on both nested grids when:

- SLA assimilation is performed on both grids in a single step (using dual-grid Ensemble members and covariances)
- the grids are coupled in both directions (2-way).

Dual-scale/dual-grid coupled assimilation allows for the proper consideration of all SLA scales, including the finest ones.

(De Mey-Frémaux et al., 2026, in prep.)

SDAP: Examples of workflows & pipelines (bash, Python, etc.)

ASSIMILATION OF REAL OR PSEUDO-OBSERVATIONS WITH SEnKF

1. Get Ensemble(s) onto working machine with a format that SDAP can understand
2. Generate CONTROL_OBS for QRTB (same Ensemble) **(Skip if assimilating real observations!)**
3. Generate OLA files and samples libraries containing randomized observations across the Ensemble
4. Run analysis for member *m*, providing state update
5. Restart model from previous restart + state update for member *m*
6. Go to 4. until done for all *m*'s
7. Produce graphics.

RUN DATA-SPACE OR STATE-SPACE DIAGNOSTIC TOOL ON ENSEMBLE

1. Get Ensemble(s) onto working machine with a format that SDAP can understand
2. Generate CONTROL_OBS for QRTB (same Ensemble) **(Skip if using real observations!)**
3. Run **`scrumcat`** diagnostic tool(s) of your choice
4. Produce graphics.

MORE WORKFLOWS…

- SEQUOIA (in-core) data assimilation workflow
- SPRUCE (scripted) data assimilation workflow
- Even more workflows available – e.g. see **`olax`** and **`scrumcat`** Usage.

SDAP practical usage

- Git clone from public repository:
  - https://gitlab.in2p3.fr/sirocco/sdap/sdap-dev
- Git branches (**`git branch`**↵), including:
  - **`master`** (stable release, currently v1.7)
  - Feature branches (unstable).
- Configure and compile:
  - **`git branch the_branch_you_want`**↵
  - **`make config`**↵ then choose among list of preset configurations
  - Edit **`sdap.conf`** for Fortran compiler & flags, locations of BLAS, LAPACK, MPI, NETCDF libraries, and other configuration stuff
  - **`make`**↵ makes commands **`beluga`**, **`olax`**, **`scrumcat`**
  - Entering commands without argument returns Usage.
- In the ULIB sandbox, the code is yours to change and adapt, without risk of interference with the 'static' SDAP libs. E.g.: "assimilation physics", definitions of uncertainties, numerical model interface, additional metrics, Python scripts, etc.

SDAP-SYMPHONIE 2026 Interface Refresh

`feat_symphint` feature branch of SDAP:

- Unstable
- Reuses some elements of previous implementations: de Nicola, Ghantous, Han
- Workflow is script-based, with separate executables for SYMPHONIE and SDAP (separate MPI setups)
- *!!!'Free' SYMPHONIE Ensembles are needed to be able to use interface!!!*

DONE, TESTED:

- Load a modern SYMPHONIE grid (tested with S271 & S362)
- Various grid-related updates in ULIB

MISSING, CRITICAL:

- Load SYMPHONIE ensembles (1-2 Pierre*months, more if complications!)
  - Accept SYMPHONIE restart file format (NetCDF)
  - See if can use previously-defined options to restart timestepping scheme
  - Activate **`scrumcat`** tool in configuration
- Need full-size case to validate everything, incl. the attached modeller(s) (& ultimate publication)

MISSING, OPTIONAL:

- Handle grid differences between members due to exposed sandbars, deep convection, etc.
  → mechanism already exists, check if OK.