The Community Inversion Framework v1.0: a unified system for atmospheric inversion studies: Supplement

Antoine Berchet[1,*], Espen Sollum[2], Rona L. Thompson[2], Isabelle Pison[1], Joël Thanwerdas[1], Grégoire Broquet[1], Frédéric Chevallier[1], Tuula Aalto[6], Peter Bergamaschi[5], Dominik Brunner[7], Richard Engelen[10], Audrey Fortems-Cheiney[1], Christoph Gerbig[8], Christine Groot Zwaaftink[2], Jean-Matthieu Haussaire[7], Stephan Henne[7], Sander Houweling[4], Ute Kartens[3], Werner L. Kutsch[13], Ingrid T. Luijkx[9], Guillaume Monteil[3], Paul I. Palmer[11], Jacob C. A. van Peet[4], Wouter Peters[9,12], Philippe Peylin[1], Elise Potier[1], Christian Rödenbeck[8], Marielle Saunois[1], Marko Scholze[3], Aki Tsuruta[6] and Yuanhong Zhao[1]

[1]Laboratoire des Sciences du Climat et de l'Environnement, CEA-CNRS-UVSQ, Gif-sur-Yvette, France
[2]Norwegian Institute for Air Research (NILU), Kjeller, Norway
[3]Dep. of Physical Geography and Ecosystem Science, Lund University, Sweden
[4]Vrije Universiteit Amsterdam, Department of Earth Sciences, Earth and Climate Cluster, Amsterdam, the Netherlands
[5]European Commission Joint Research Centre, Ispra (Va), Italy
[6]Finnish Meteorological Institute (FMI), Helsinki, Finland
[7]Swiss Federal Laboratories for Materials Science and Technology (Empa), Dübendorf, Switzerland
[8]Max Planck Institute for Biogeochemistry, Jena, Germany
[9]Meteorology and Air Quality Group, Wageningen University and Research, Wageningen, the Netherlands
[10]European Centre for Medium-Range Weather Forecasts, Reading, RG2 9AX, UK
[11]School of GeoSciences, University of Edinburgh, Edinburgh, EH9 3FF, UK
[11]Centre for Isotope Research, University of Groningen, Groningen, the Netherlands

Correspondence: antoine.berchet@lsce.ipsl.fr

Table S1: Plugin types in pyCIF

⋆ obsoperator:	- purpose:	maps data from the control space to the observation space and conversely			
	- inheritance:	- controlvect			
		- obsvect			
		- model			
	- metadata	none			
	- data	none			
	- methods:	- obsoper:	$\mathbf{x}$	$\rightarrow$	$\mathcal{H}(\mathbf{x})$
			$\mathcal{H}^*(\mathbf{y}^*)$	$\leftarrow$	$\mathbf{y}^*$
	- calls:	- controlvect.$\mathbf{\Pi}_{\mathcal{X}}^{\mathfrak{F}}$:	$\mathbf{x}$	$\leftrightarrow$	$\mathbf{x}_{\text{model}}$
		- model.$\mathbf{\Pi}_{\mathfrak{F}}^{\mathcal{F}}$:	$\mathbf{x}_{\text{model}}$	$\leftrightarrow$	model inputs
		- model.run:	model inputs	$\leftrightarrow$	model outputs
		- model.$\mathbf{\Pi}_{\mathcal{C}}^{\mathfrak{M}}$:	model outputs	$\leftrightarrow$	$\mathbf{y}_{\text{model}}$
		- obsvect.$\mathbf{\Pi}_{\mathfrak{M}}^{\mathcal{Y}}$:	$\mathbf{y}_{\text{model}}$	$\leftrightarrow$	$\mathbf{y}^{\text{o}}$
⋆ controlvect:	- purpose:	initializes the control vector (including metadata) and uncertainties and computes projections from and to the control space			
	- comments:	needs metadata from the model (e.g., resolution) and the domain to carry out projections			
	- inheritance:	- domain			
		- model			
	- metadata:	- components (e.g., fluxes, initial conditions, etc.)			
		- dimension			
		- correlation patterns			
	- data:	- $\mathbf{x}_{\text{b}}$			
		- $\mathbf{B}$ if stored, main components otherwise			
	- methods:	- $\mathbf{\Pi}_{\mathcal{X}}^{\mathfrak{F}}$:	$\mathbf{x}$	$\leftrightarrow$	$\mathbf{x}_{\text{model}}$
		- $\mathbf{\Pi}_{\mathcal{X}}^{\mathfrak{A}}$:	$\mathbf{x}$	$\rightarrow$	$\chi \equiv \mathbf{B}_{1/2}\mathbf{x}$
		- $\mathbf{\Pi}_{\mathfrak{A}}^{\mathcal{X}}$:	χ	$\rightarrow$	$\mathbf{x} \equiv \mathbf{B}_{1/2}\chi$
		- init_$\mathbf{B}$:	some data	$\rightarrow$	$\mathbf{B}$
	- calls:	- domain.resolution			
⋆ obsvect:	- purpose:	initializes the observation vector (including metadata) from the measurements and computes projections from and to the observation space			
	- inheritance:	- domain			
		- measurements			

	- metadata	- species			
		- observation type (in situ, satellite, etc.)			
		- correlations if any			
	- data	- $\mathbf{y}^{\mathrm{o}}$			
		- $\mathbf{R}$ if stored, main components otherwise			
	- methods:	- $\mathbf{\Pi}^{\mathcal{Y}}_{\mathfrak{M}}$:	$\mathbf{y}_{\mathrm{model}}$	$\leftrightarrow$	$\mathbf{y}^{\mathrm{o}}$
		- $\mathbf{\Pi}^{\mathcal{Y}}_{\mathfrak{D}}$:	$\mathbf{y}_{\mathrm{meas}}$	$\leftrightarrow$	$\mathbf{y}^{\mathrm{o}}$
		- $\mathbf{R}^{-1}\cdot()$:	$\mathbf{y}$	$\rightarrow$	$\mathbf{y}=\mathbf{R}^{-1}\mathbf{y}$
	- calls:	- domain.resolution			
⋆ model:	- purpose:	drives the transport model, prepares inputs and extracts outputs to CIF-compatible structures			
	- inheritance:	none			
	- metadata	- resolution			
		- computation mode			
		- sub-periods if any			
		- chemistry if any			
		- model-specific configuration			
		- path to fixed inputs (e.g., meteo data)			
	- data	none			
	- methods:	- run:	model inputs	$\rightarrow$	model outputs
		- $\mathbf{\Pi}^{\mathcal{C}}_{\mathcal{F}}$:	$\mathbf{x}_{\mathrm{model}}$	$\leftrightarrow$	model inputs
		- $\mathbf{\Pi}^{\mathfrak{M}}_{\mathcal{C}}$:	model output	$\leftrightarrow$	$\mathbf{y}_{\mathrm{model}}$
	- calls:	none			
⋆ simulator:	- purpose:	computes the cost function and its gradient			
	- inheritance:	observation operator			
	- metadata	none			
	- data	none			
	- methods:	simul:	χ	$\rightarrow$	$(J(\chi),\nabla_{\chi}J)$
	- calls:	- obsoperator.obsoper:	$\mathbf{x}$	$\rightarrow$	$\mathcal{H}(\mathbf{x})$
			$\delta\mathbf{y}$	$\rightarrow$	$\mathbf{H}^{*}\delta\mathbf{y}$
⋆ minimizer:	- purpose:	minimizes a function starting from a given point			
	- inheritance:	observation operator			
	- metadata	none			
	- data	none			
	- methods:	minimize:	(χ,J)	$\rightarrow$	χ_{opt}

- calls: - simulator.simul: $\chi \quad \rightarrow \quad (J(\chi), \nabla_\chi J)$