Fostering collaboration through improved software development practices for the ONEFlux eddy covariance data processing pipeline

Gilberto Pastorello¹, Carlo Trotta²,³, Alessio Ribeca³, Keith Beattie¹, Sy-Toan Ngo¹, Housen Chu¹, You-Wei Cheah¹, Danielle Christianson¹, Giacomo Niccoli²,³, Sigrid Dengel¹, Diego Polidori³, Peter Isaac⁴, Matthew Archer⁵, Dominic Orchard⁵, Deb Agarwal¹, Sebastien Biraud¹, Margaret Torn¹,⁶, and Dario Papale³

¹Lawrence Berkeley National Laboratory, Berkeley, United States of America  
²Euro-Mediterranean Centre on Climate Change (CMCC), Viterbo, Italy  
³University of Tuscia, Viterbo, Italy  
⁴TERN-OzFlux, Melbourne, Australia  
⁵University of Cambridge, Cambridge, United Kingdom  
⁶University of California, Berkeley, United States of America

Standardized processing of eddy covariance data is important for studies combining data from multiple sites, for validating remote sensing measurements as well as runs of ecosystem and climate models, and for applications relying on these flux data to create derived products like upscaled fluxes, among other examples. However, maintaining consistency within the software used for this processing while allowing for evolution of this code across research networks presents novel challenges in software development. The introduction of the ONEFlux (Open Network-Enabled Flux) eddy covariance data processing pipeline, originally developed within a collaboration of the AmeriFlux Management Project, the European Fluxes Database, and the ICOS Ecosystem Thematic Centre, supported the creation of consistently processed global eddy covariance data products. In particular, ONEFlux codes were used to generate the FLUXNET2015 dataset, which is widely adopted by thousands of eddy covariance data users in their work in research, ranging from soil microbiology to large scale drought effects, and also education, from basic plant biology all the way to global climate change. We are now more thoroughly instrumenting the code, and the code development process, to better address these challenges, efforts which we will describe in this presentation. In particular, we are seeking to improve software development practices to allow for more streamlined collaboration on expanding and contributing to the codebase. For instance, we are adopting planned release cycles for code updates, designing more detailed ways to incorporate and evaluate new modules, introducing data-centric testing and continuous integration, improving code performance, and adopting several other software engineering best practices more widely in the development workflows. The main goal of these changes is to lower the barriers for running ONEFlux by regional networks processing their data, while at the same time better supporting contributions from the community into the codebase. This will be critical to continue the current use of ONEFlux to generate updated versions of flux datasets by regional networks, the components of new global products.