Lawrence Berkeley National Laboratory

LBL Publications

Title

Investigating Scientific Data Change with User Research Methods

Permalink

https://escholarship.org/uc/item/87b7h27d

Authors

Paine, Drew

Ghoshal, Devarshi

Ramakrishnan, Lavanya

Publication Date

2020-08-01

DOI

10.2172/1650129

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Investigating Scientific Data Change with User Research Methods

Drew Paine, Devarshi Ghoshal, Lavanya Ramakrishnan
Data Science and Technology Department
Lawrence Berkeley National Laboratory
{pained,dghoshal,lramakrishnan}@lbl.gov

August 2020

Abstract

Scientific datasets are continually expanding and changing due to fluctuations with instruments, quality assessment and quality control processes, and modifications to software pipelines. Datasets include minimal information about these changes or their effects requiring scientists manually assess modifications through a number of labor intensive and ad-hoc steps. The Deduce project is investigating data change to develop metrics, methods, and tools that will help scientists systematically identify and make decisions around data changes. Currently, there is a lack of understanding, and common practices, for identifying and evaluating changes in datasets since systematically measuring and managing data change is under explored in scientific work. We are conducting user research to address this need by exploring scientist's conceptualizations, behaviors, needs, and motivations when dealing with changing datasets. Our user research utilizes multiple methods to produce foundational, generative insights and evaluate research products produced by our team. In this paper, we detail our user research process and outline our findings about data change that emerge from our studies. Our work illustrates how scientific software teams can push beyond just usability testing user interfaces or tools to better probe the underlying ideas they are developing solutions to address.

Keywords — user research, qualitative methods, data change, scientific software development
Report Number: LBNL-2001347

1 Introduction

Scientific advances increasingly depend on the effective processing of large scientific data being produced by experiments, observations, and simulations. However, datasets are often changing due to instrument configurations, quality assessment and quality control processes, and modifications to software pipelines. Frequently today, there is limited understanding or representation of these data changes and its effects on scientific processes, largely hindering the development of software tools.

Building software tools for complex domains requires understanding nuances of the problem space and varied ways stakeholders work. User research is a well established component of commercial software development that investigates user's practices and behaviors, generates interface designs, and evaluates problems, tools, and documentation being developed. Adopting user research techniques in scientific environments is challenging due to the emergent, evolving nature of scientific work. Products of scientific work are not well defined at the beginning of a project, and building tools to successfully support research requires working in constantly shifting environments. In the past decade, some scientific software development work has carried out user research to evaluate different types of interfaces, often through usability testing techniques, in an effort to identify problems and fix them before a release [1, 9, 12, 13]. Evaluating interfaces alone is insufficient since it leaves out whether the fundamental concepts driving a project's design efforts are aligned with their intended users wants and needs.

The Deduce [1] (Distributed Dynamic Data Analytics Infrastructure for Collaborative Environments) project is investigating data change with the goal of developing metrics, tools, and practices that will help scientists identify changes and determine how they impact an analysis. The project is supporting the identification of data change with the development of Dac-Man [2] [6]. Dac-Man is a change detection and management framework capable of comparing massive datasets at scale, in a variety different computing environments. We are using user research methods to gather foundational insights, and generate ways to convey information, about data change in scientific work along with evaluations to assess and reflect on the products of this work. By incorporating multiple user research methods our work is designed to not only evaluate a Dac-Man prototype but step back and assess the design space of our project. Our efforts enable us to help conceptualize data change in scientific work while iteratively shaping products in this project.

In this paper, we discuss the user research methods we used and the particular ways we organized this work over the course of our project. We show how our foundational data collection during resulted in design requirements for a data change calculation tool and articulated a problem space. We illustrate how conducting foundational, generative, and evaluative work produced revelations about this prototype and our initial problem space to deepen our understanding and identify areas for further investigation.

2 Background

To contextualize this paper's findings we first describe user research and examine previous work applying these methods to scientific software development. We then explain the Deduce project, its focal scientific areas, and its three research thrusts which includes the user research efforts discussed in this paper.

2.1 User research and scientific software development

User research methods support system designers by engaging focal users at all stages of a tool's development. User researchers can draw upon a variety of qualitative and quantitative methods to investigate the needs of users and evaluate the effectiveness of products [7, 15, 16]. User research ranges from foundational work to understand people's practices, generative approaches to develop design concepts and problem spaces, and evaluative methods to test and identify improvements to products being built [7]. These approaches, and the purposes underlying each, as different threads of user research combined help continuously shape and refine a problem space and the products a team is developing to address this space. Teams iterate between foundational, generative, and evaluative approaches as their understandings of problems evolve and the products they develop need refinement.

Foundational methods can include exploratory user surveys or more in-depth ethnographic inquiries with participant observation and semi-structured interviews. Ethnography in particular enables researchers to develop a deep understanding of group cultures and ways of working that can shape software development work [14]. Generative methods include developing mockups or prototypes of user interaces (or other artifacts) to develop ideas and explore a problem space. Evaluative methods include usability tests in laboratory environments or expert heuristic evaluations of interfaces. Surveys can also include an evaluative component if the research team has artifacts to solicit feedback on.

[1] http://deduce.lbl.gov/

[2] https://github.com/deduce-dev/dac-man

Some scientific software projects have incorporated user research in their development processes over the past decade to evaluate different forms of their products [1,9,12,13]. Poon et al. [12] conducted interviews with astrophysicists to evaluate whether a distributed chat tool being developed enabled these scientists to better handle telescope monitoring across geographical and temporal divides. Aragon et al. [1] used a participatory design process for an astrophysics visualization tool—observing existing work practices and interviewing science team members, then gathering feedback on each iteration of a user interface to identify mismatches between the tool's design and the scientist's ways of working. Macaulay et al.'s [9] team evaluated a piece of life sciences image categorization software and conducted foundational ethnographic research to understand the environment this tool is used in. These researchers engaged with an existing project, with clear goals and a working product, to rapidly and iteratively improve the software's graphical user interface while ensuring it meets existing user needs and helps draw more members of the community to the project. Ramakrishan et al. [13] incorporated a usability study process when building a scientific workflow programming interface to evaluate the design of the API.

The work in our paper is distinct from prior efforts by focusing on the concept of data change rather than just user interfaces, even as we evaluated our project's prototypes and artifacts. Data change problems are inherent in scientific ecosystems, but a concrete generalized conceptualization of scientific data change or tools to manage it are not well defined. We needed to learn about challenges from scientists dealing with changes in their datasets to be able to produce metrics and tools for calculating change. Our approach investigates data change from multiple perspectives, collecting foundational insights about the problem space, generating ways to present concepts to users, and evaluating our own research products.

2.2 The Deduce project

The Deduce project is exploring the challenges and methodologies for enabling usable and efficient scientific data change management. Currently scientists producing next-generation discoveries face challenges with their ability to efficiently and effectively analyze large quantities of data, let alone identify when parts of these sets have changed. Datasets are continuously expanding and changing due to instrument adjustments, software updates, and/or quality assessment and control practices. Currently data producers seldom provide enough information with releases describing what, how, or why the data changed. Scientists using these datasets employ a number of manual and ad-hoc steps to identify and assess the impacts of data changes on their unfolding work. There is presently a lack of tools, metrics, or practices that are readily available to help researchers systematically evaluate changes in their datasets. We urgently need to develop the resources to facilitate this work so that we can better support data exploration and analyses with change information a visible, integral aspect of workflows and practices. This is essential to by making data change a visible, integral aspect of workflows and practices.

The Deduce project is focused on use cases from a few primary scientific domains. Each domain captures data of varying scale in different structures, resulting in multiple forms of data change to influence our investigation. The **cosmology** use case comes from the Sloan Digital Sky Survey (SDSS), which consists of images of the sky captured through an optical telescope. SDSS releases many versions of the dataset on a yearly basis, where each version contains millions of files and terabytes of data. The **environmental sciences** use case consists of Fluxnet data that has measurements on the exchanges of carbon dioxide (CO_2), water vapor, and energy between terrestrial ecosystems and the atmosphere. These datasets are released every 7-8 years, and incur several data and metadata related changes. The **light sources** use case draws upon data from one of the beamlines at the Advanced Light Source (ALS) x-ray synchrotron facility at Lawrence Berkeley National Laboratory. Different ALS beamlines can generate terabytes of raw and derived data each day, in this case x-ray images of materials under different configuration parameters (like pressure, temperature, etc.). Currently, scientists manually look at the images to identify the impact of these changing parameters. The final use case refers to NASA's **MODIS** (Moderate Resolution Imaging Spectroradiometer) satellite data that provides measurements about Earth's changing dynamics that include land, oceans, and the atmosphere. These datasets are often updated and removed without any associated metadata that describes the changes.

2.2.1 Three research thrusts

The Deduce team is working across three main research thrusts, Figure 1. All three are necessary to help us better characterize how scientists in the multiple domains we are studying as use cases grasp data change and determine how to design and build tools to convey change. Our first thrust is the user research described in the following section. The second thrust is investigating how to detect and measure change in scientific datasets with statistical measures. Team members are employing statistical techniques from uncertainty estimation to produce quantifiable metrics of data change in multiple scientific domains. The overall goal of this thrust is to enable scientists to assess the impacts of data change on the analyses they are undertaking, providing the information necessary to determine under what conditions to re-run a past analysis [2,5]. Our user research evaluated some of this thrust's products through a survey to identify and refine requirements for such metrics (see Figure 1). The project's third thrust is developing Dac-Man, a change

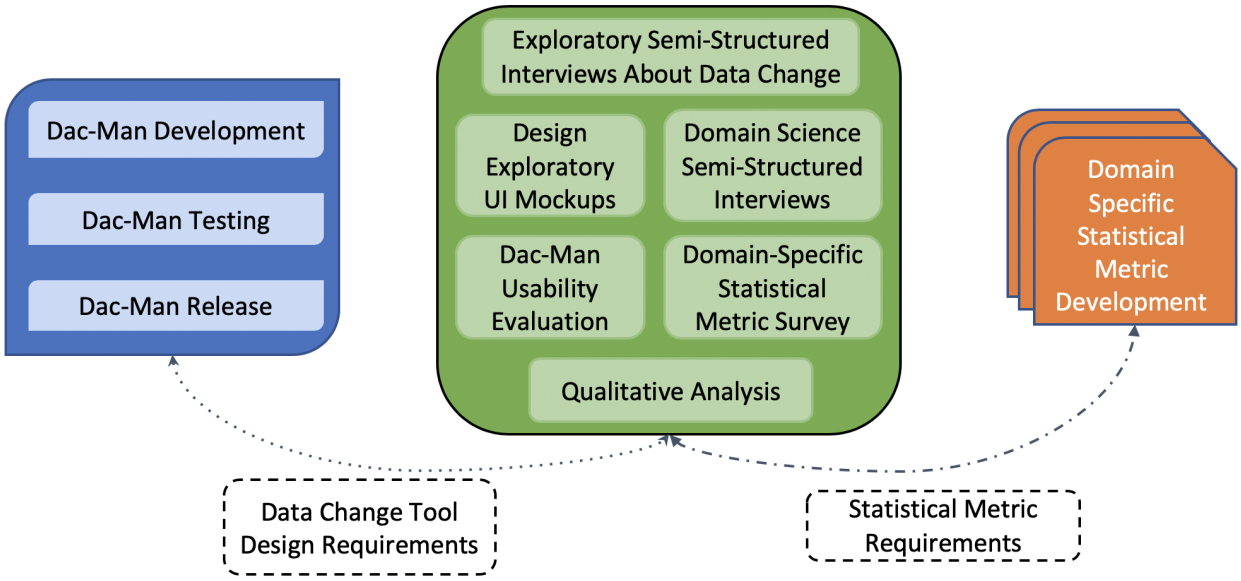


Figure 1: Overview of our user research process to iteratively investigate and characterize scientific data change.

detection and management framework capable of comparing massive datasets at scale, in a variety different computing environments. Dac-Man is designed to reveal changes in datasets at a filesystem level as well as in the data itself so that scientists can make more informed decisions about the version(s) of data to use in their analyses. The design of Dac-Man was influenced and refined through the efforts of our user research. The focus of this paper is our first thrust, explaining how we undertook user research.

3 Our User Research Approach

We employed multiple methods in our project, outlined in Figure 1, that cover the spectrum of foundational, generative, and evaluative approaches. We conducted foundational semi-structured interviews, generated concepts through exploratory user interface mockups, evaluated a prototype tool through usability tests, and explore change in an earth sciences context while evaluating statistical metrics utility to scientists through a survey. Over the span of a project particular user research methods are applicable at different moments. Our experience illustrates one case where using multiple methods helped us define and reflect on a problem space and enabled us to accommodate the diversity of our possible users and scientific domains as well as the dynamic nature of scientific environments. Incorporating multiple research methods over time and having a focus on larger concepts helps us produce rich results. Our results are more comprehensive than what would have been possible using a single method or by focusing on evaluating a particular product alone.

The approach to these methods that we illustrate may differ from the needs of other projects. Teams incorporating user research should balance different trade offs. This is an issue we examine further in related work [10].

3.1 Foundational insights from semi-structured interviews

We have conducted semi-structured interviews with domain scientists to learn about data change. This method is key a element of ethnographic inquiries where researchers are developing nuanced understandings of ways of working and group cultures. Semi-structured interviews provide the opportunity to develop detailed descriptions, hear multiple perspectives on processes, and develop holistic descriptions of problems [17].

With semi-structured interviews the interviewer crafts a set of open-ended questions to ask subjects. The interview is semi-structured rather than rigidly structured so that a consistent narrative emerges across interviews while leaving room for detailed exploratory questions throughout the conversation. This provides a mechanism to follow up and probe particular issues more deeply. Commercial software projects commonly characterize such variations of semi-structured interviews as contextual inquiry to distinguish this method from common laboratory studies [8].

3.1.1 Interview Methods

In our ethnographically informed approach, as industrial contextual inquiry approaches also attempt, we try to engage with participants in their everyday work setting and ask them to show us artifacts and products they use and produce. Seeing where and how scientists do their work enables us to gather key contextual insights that may not emerge when they answer questions on their own. Semi-structured interviews are essential in our efforts to develop a grounded understanding of different forms of scientific work and changes to data as part of this work. Scientific software teams can similarly benefit from conducting interviews since even the simplest of conversations should generate insights about the users they are working with. Interviews (and observations alongside them in a person's workspace) can reveal that a scientist is in fact not using a tool as expected, circumventing what the team thought was being used and crafting their own approach. Talking with users in a semi-structured manner is an opportunity to revisit and upend assumptions while gaining real world insights about the problems being addressed.

Our two rounds of semi-structured interviews were conducted in person or over video conference, designed to last around one hour, and recorded for later analysis. Each set of interviews was designed to help us characterize data change in different scientific domains and obtain some feedback on emerging products of the Deduce project.

Our exploratory semi-structured interviews were led by one of the project's PIs and joined by another member or two from the team while the domain science interviews were conducted solely by the first author. Organizing semi-structured interviews with a single or multiple interviewers changes the tenor of the conversation, participants are potentially more open with a single interviewer. Having multiple interviewers can lead to deeper exploration of particular issues due to varying points of views among the team. In our case having multiple people present during exploratory interviews resulted in a faster feedback loop among team members early in the project when tasks and concepts were nascent. By the time our domain science interviews were conducted the project had a variety of products in process and had developed a shared conceptualization of the problem space. This made it easier for a single interviewer to probe the concepts and outputs of the team with scientific participants.

At the conclusion of interviews we asked for sample datasets from the subject for use when developing tools and statistical metrics. These example datasets helped our team understand and assess different types of changes. We collected an older and a newer release of each dataset including: *a)* Cosmology data from the Sloan Digital Sky Survey (SDSS)[3], *b)* X-ray imaging data from two Advance Light Source (ALS)[4] beamlines, *c)* Earth science data products from the Moderate Resolution Imaging Spectroradiometer (MODIS)[5], *d)* Earth science flux data from the Fluxnet[6] network, and *e)* Earth science data from the Watershed Function Scientific Focus Area (Watershed SFA)[7] project

Exploratory interviews about data change. Our exploratory interviews were conducted with four domain scientists. These included earth scientists working with Fluxnet or MODIS data and cosmologists using SDSS data. We inquired about how subjects perceived and handled data change in their work processes, the impacts of data change on downstream products and workflows, and how they would like to view and interact with data change calculations and the analyses producing this information. Additionally, early results from statistical change analyses of MODIS data were presented to our science collaborators which resulted in identifying in insights into metrics that might be useful to certain domains. These early interviews highlighted the difference in perspectives between data producers and consumers, the impact of data change on models, both areas the team had not initially considered. These exploratory interviews generated a baseline from which our team used to develop a plan for the project resulting in development of statistical methods and tools (Dac-Man).

Domain science interviews. We interviewed twelve domain scientists to learn about their research area, data they work with as part of projects, determine versions of data products to use, expectations for how often new releases are available, an example of unexpected effects from a change in a release, and asked for feedback on user interface mockups (discussed below) and potential metrics of data change. These interviews were designed to help us better characterize data change and identify additional use cases for change management tools. We interviewed five astrophysicists, five earth scientists, and two scientists working with ALS beamline data. This round of interviews resulted in more views on data change, particularly as interviewees could probe our conceptualization through the mockups we discussed. With four of these interviewees, we followed up the semi-structured interviews with a usability test of our Dac-Man prototype (explained below). We explore findings from these interviews in more depth in [11].

[3] https://www.sdss.org/
[4] http://als.lbl.gov/
[5] https://modis.gsfc.nasa.gov/data/
[6] https://fluxnet.fluxdata.org/
[7] https://watershed.lbl.gov/

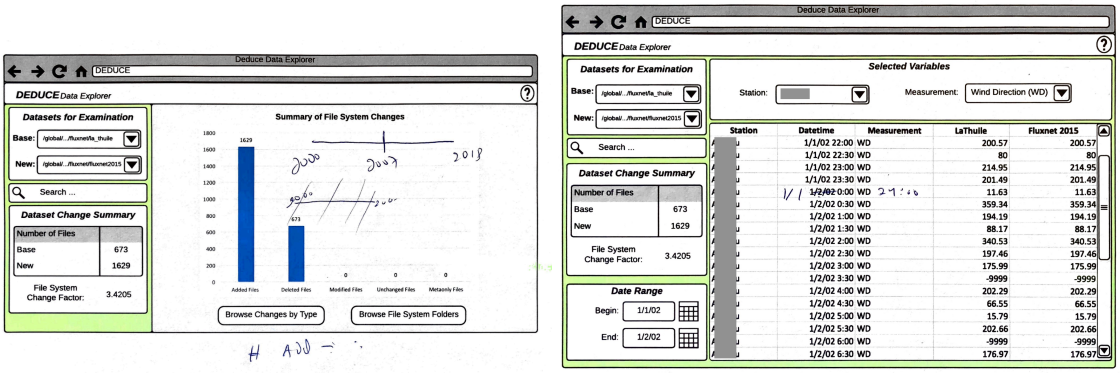


Figure 2: Example user interface mockups marked up by an interviewee. The mockups show real outputs comparing values. The participant noted that being able to visually explore changes for different types of variables like the mockup presented on the right would be useful for their data analysis work.

3.2 Generating exploratory user interface (UI) mockups

Our second user research method is generative, designing exploratory user interfaces and eliciting feedback from colleagues and scientists to explore what data change means. Generative methods enable a team to develop representations of a problem and spark conversations that lead to new understandings of a problem and/or requirements in tools. This type of method enabled our team to ask what kinds of interventions we could develop for scientists and how we could represent change in different domain datasets. We designed and incorporated UI mockups into our domain science interviews to probe our understanding of the problem space we developed, understand different domain specific approaches to identifying changes, and to figure out if a graphical user interface is something our team even needed to develop.

Science software teams similarly can use mockups to work out initial ideas for tackling a problem among the team, and to engage users in discussions about concepts being addressed. Teams can create mockups as simple as sketches on paper all the way across the spectrum to fully interactive prototypes. Determining what fidelity of mockup to employ will influence the types of insights gathered, with interactive prototypes conveying a more concrete and thus less open to reinterpretation concept. Sketches leave room for a re-envisioning of a concept during discussions with potential users. Teams should balance fidelity based on their goal and ability to incorporate feedback in their process.

Our team chose user interface mockups that are of intermediate fidelity (not simple sketches but also not actually interactive) as a way of making calculations of data changes more concrete while still leaving room for exploration of the problem in conversations with stakeholders. Mockups are a form of prototyping that can range in fidelity from simple pencil sketches on paper to high fidelity interactive designs built with various design tools [3]. We chose a middle ground, producing designs using a general website template in the LucidChart[8] tool. We incorporated some real data change calculations produced by a Dac-Man prototype or from the statistical calculations of the Deduce team. This ensured that the mockups were grounded in real scientific data, making them more tangible to our subjects. We varied the mockups to incorporate actual change calculations produced with data that a given interviewee would be familiar with (e.g. cosmology data for SDSS participants, Fluxnet data for earth scientists). Producing mockups with this middle ground fidelity demonstrated that our products are functional but still open to re-interpretation as we learn more from different participants. This helps reduce some risk with high fidelity mockups where a participant can feel like there isn't room to re-imagine the design being discussed.

We discussed initial versions of our mockups with colleagues in our data science department who build systems for scientists. These were informal 15 to 30 minute conversations where we asked about the datasets they have experience working with, the notion of data change we were tackling with the mockups, and most importantly their interpretation of the printed visuals we had. These conversations identified some confusing elements of our initial design and the likely domain specific nature of much of this type of user interface. We revised our mockups based on this informal feedback to use in our semi-structured interviews, clarifying the change information and metrics produced by Dac-Man.

Discussing mockups during our domain science interviews provided a mechanism to shift from the descriptions of interviewees work to elicit concrete feedback and reflection about data change and our representations of the concept. One scientist took the opportunity to mark up printed copies of the mockups when explaining their work and perceptions of data change, (see Figure 2). Their feedback about our conceptualization of data change and our tool's calculations resulted in new potential avenues to explore in future design work.

[8] `https://www.lucidchart.com/`

3.3 Evaluative usability tests

Following our second round of semi-structured interviews we conducted six usability tests of Dac-Man. Usability tests are an evaluative user research method. Evaluative methods help teams gauge how well their products align with the thoughts and work practices of subjects. When projects are close to releasing a product a usability evaluation can identify areas to quickly fix by organizing structured tests with well specified tasks to complete [15]. Usability testing is the user research method we have identified as most commonly completed in scientific projects as teams neared releases of a tool. We conducted usability evaluations to develop feedback for the Deduce project's third research thrust building Dac-Man so that the commands and documentation could be revised before an initial release. Science software teams should employ usability testing to verify the terminology of their tool and find issues with interfaces that would hinder adoption by users. If testing is completed early enough it is also possible to identify misalignments between a new tool and the workflows and practices scientists already use within an ecosystem, potentially adapting to fit better within such an ecosystem.

We designed our usability evaluations to be semi-structured and flexible rather than focused on strict completion of a narrow set of tasks. Our motivation was to obtain feedback about the Dac-Man prototype and assess our overall conceptualization of data change. We designed our test to explore how users understand the mechanics and significance of the Dac-Man change management framework with different types of scientific data. This included how well the tool meets some of the initial and primary use cases developed using results of our exploratory semi-structured interviews.

Our usability tests were designed to be conducted one-on-one with the interviewer and tester installing the project's code from a private BitBucket[9] repository. The tests took place in-person in the participant's office, a shared conference room, or over a Zoom video chat. Conventionally tests are conducted in a usability lab on a controlled computer, and increasingly remote testing is common with tools like User Zoom[10] to increase the validity by allowing users to test in a more natural environment. Since scientific tools and data often require particular computing environments we wanted testers to try Dac-Man in their usual workspace, whether that was their laptop or an HPC system they generally use. This is also a change from common usability testing practice where the system users interact with, along with its data, would be controlled and specially setup for the test. We wanted to have insights from more realistic scenarios as well as the opportunity to evaluate how our prototype intersected with user's own conceptualizations of change in their data. As a result, our evaluations ended up being fairly exploratory and not rigidly systematic, which we needed for our project.

We encouraged flexibility from our testers to enable them to explore our documentation, command line interfaces, and outputs conveying information about data change. We wanted participants to explore the tool's features and explain how it did and did not align with their existing work practices or products. We wanted to clarify participant's data change use cases and identify the different types of outputs that are relevant for understanding the changes in their datasets as well as aspects of the tool to fix immediately before releasing the first version. We began test sessions with a brief reminder about our project and description of our Dac-Man prototype. We asked the testers to talk aloud while working through Dac-Man 's README and when trying out commands. Subjects were also encouraged to ask questions and offer general thoughts on data change throughout.

Structuring our evaluations. Our evaluations were organized like our semi-structured interviews to take place for thirty to sixty minutes. The interviewer conducting the evaluation had a general task list to work through with participants, but we were not trying to evaluate time on task or other quantified aspects of the experience as is traditional in laboratory-based, moderated usability tests [9, 13, 15]. We first asked testers to read about the tool from the repository README then install it in their everyday computing environment (personal laptop or an HPC system). This would enable them to test Dac-Man with some of their current scientific data. We asked each tester about the datasets they brought to develop a sense of how these products were structured. We also wanted to know how they handled data change, asking about tools and practices they had developed already. After installing the tool every participant ran a simple sample test script and examined the outputs. We originally intended for the participant to try out the complete change management workflow using a sample dataset consisting of two versions of a LaTeX paper draft. The intention was to see how they experienced the commands and outputs of the tool. We ended up skipping this task to have participants try the tool using their particular domain specific data since they were interested in exploring the data they regularly work with. Finally, we asked wrap-up questions to inquire about their thoughts on the experience, the tool and its commands and outputs, and thoughts on other use cases that come to mind.

Administering usability evaluations. We recruited six participants for individual evaluation sessions. The participants were selected based on the different levels of expertise in using data science tools, types of datasets they deal with, and their roles in managing large datasets. We conducted the six usability evaluations when we were nearly ready to release a first version of Dac-Man and wanted to be able to adjust any lingering issues. This was also early enough in our

[9] `https://bitbucket.org/`
[10] `https://www.userzoom.com/`

process that longer-term efforts could be reflected on and adjusted if needed.

We asked four of our domain science interviewees to chat again shortly after that conversation to try out the prototype. We also asked two department colleagues experienced with using a wide variety of scientific data and who had previously offered feedback on our user interface mockups. Each of our testers had some familiarity with the Deduce project and had thought about data change to some extent. Each individual was asked to bring at least two versions of datasets they regularly work with to the session. We did this so that the testers would be exploring change in data they are very familiar with and to see how our tool worked with a variety of scientific data. Two of the testers worked with SDSS cosmology datasets, two with Fluxnet earth science datasets, one a biology dataset, and one with ALS beamline x-ray imaging data.

Four of our evaluations took place in person with the interviewer present. Two participants completed the evaluation remotely. One conducted the test remotely with an interviewer present over Zoom[11] video conference where their screen was shared and recorded. The other was given access to the project repository in advance since they are located elsewhere and we intended to use Zoom here as well. However, this individual worked through all of the test tasks without an interviewer present before the session began. We had to adjust our protocol and solely interview them about their experience installing and using the tool after the fact.

3.4 Survey evaluation of flux data change analyses

The Deduce project's environmental sciences use case explores how two synthesis datasets measuring exchanges of carbon dioxide, water vapor, and other variables were produced 7-8 years apart. Our team examined the two publicly available releases of this global flux data by conducting multiple statistical analyses of changes between the releases from 2007 (La Thuile) and 2015 (FLUXNET2015). These statistical analyses developed metrics and visualizations of change that offer an overview of the changes between each of these major releases. To evaluate these metrics and visualization we developed a survey to gather feedback about their utility for identifying and assessing changes.

Science software teams may choose to employ a survey for evaluative purposes like we primarily did but this method is also useful for gathering large-scale foundational insights. Surveying a community about needs can generate initial ideas that inform a subsequent round of interviews where the team can gather deeper, user specific knowledge. Our team chose a survey in this case, rather than an interview or usability test, so that participants would be free to interpret each visual as long as they desired and provide open ended feedback. We felt that this was essential since these plots are artifacts of nuanced scientific analysis work. Other science software teams may find a similar need with some sort of artifact and could consider a survey for evaluation. This works best if the community in question will engage with the team through such a mechanism.

Survey Design. Our survey design includes a set of overview questions, background on our change analysis, plots resulting from this analysis for evaluation, and wrap-up questions. The questions and plots are available in an appendix in Section A.

The overview questions asked about the La Thuile and FLUXNET2015 flux datasets respondents use in their work, familiarity with the two data releases analyzed, relationship to flux data (e.g. PI, research scientist, student), whether they have encountered change in flux data releases, how they analyze any changes currently, and whether they have any tools to help analyze change.

The background about our change analysis explained how variables between the two data releases were compared since names and methods of recording data shifted, requiring manual work to enable a comparison. We made respondents affirm that they understood this process before continuing the survey and asked for any feedback they might have.

The core of the survey was seven questions asking respondents to evaluate plots visualizing particular change analyses, see Figure 3. The analyses progressed from high level views of change across the entirety of each release, change in a specific variable throughout the data releases, narrowing to specific variables from single sites of data collection. These represented a range of perspectives on the changed data releases and illustrated different possible issues to look for when assessing a new data release. For each plot we asked for respondents key takeaways, how strongly they agreed or disagreed that the plot helped them grasp change between the data releases, and for any additional feedback on the plot. We included open ended questions along with Likert scale questions so that we could obtain detailed thoughts while also having a more abstracted measure across responses.

Finally, our wrap-up questions asked for thoughts on additional metrics or visualizations of change the respondents thought would be helpful and an opportunity to provide any additional feedback.

Survey Process. We initially designed and piloted this survey in 2018 with three scientists. These scientists were interviewed about data change and contacted about our pilot survey since they were expert users and contributors to the two data releases our team analyzed for changes. We piloted the survey by sharing a draft in a Word document. The

[11] `https://zoom.us/`

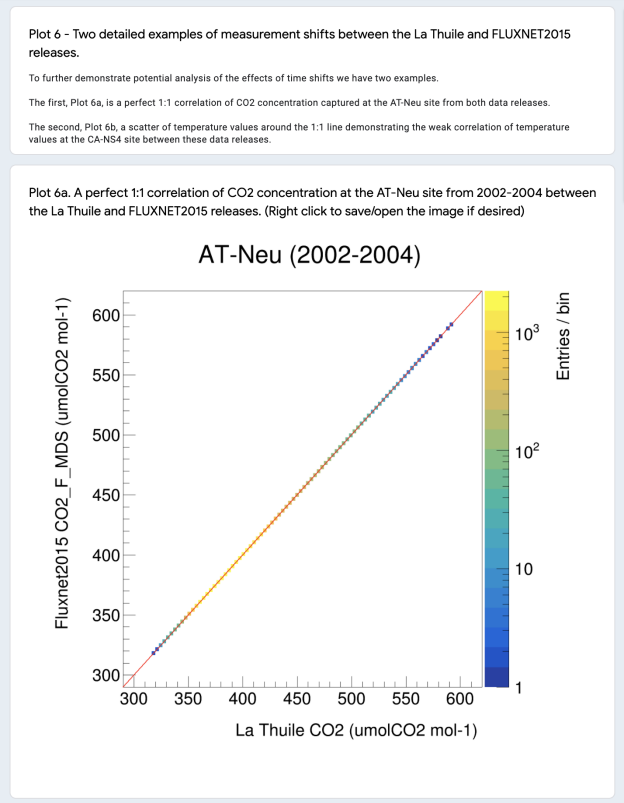


Figure 3: Example plot and description included in our survey.

feedback from this pilot made it clear that we not only needed to refine the presentation of the plots but also simplify how we presented metadata about each plot. Based on preliminary feedback it seemed respondents skimmed information about our team's analysis, repeatedly raising questions in later answers that could have been resolved by reading an overview at the beginning of our survey. This lead to our decision to require survey respondents affirm that they had read a description of our change analysis before proceeding to examine the plots.

During summer and autumn 2019 we shared a revised version of the survey with the earth science community who uses this flux data. We created a Google Form survey and distributed it via email lists to the community producing these datasets in June 2019 with follow up emails in August. We received a total of eight responses from our initial solicitation, all from PIs or other established researchers. We attempted to obtain feedback from more junior researchers by targeting a community mailing list dedicated to such members in our August follow up emails, but we did not receive any additional responses. We suspect that a summer time frame when researchers conduct field work combined with the detailed nature of the information in this type of survey may have been impediments to obtaining more feedback.

3.5 Iterative qualitative data analysis

We have iteratively analyzed our multiple forms of data using an "open coding" [4] approach. This type of approach is a mechanism for identifying emergent ideas in qualitative data by reading and re-reading the data gathered, noting down concepts and insights as "codes". Codes are shorthand for a concept which a team systematically applies to a qualitative dataset. Examples from our semi-structured interviews include "source of dataset", "expectations of change", "project organization", "temporal rhythm", "useful metrics", "desire to filter outputs" and so on. Qualitative researchers may code data by taking notes as they work through interview transcripts or use purpose-built analysis software like atlas.TI[12] or NVivo[13]. Our team coded our different forms of data on paper printouts or in Google Document copies of transcripts.

Once a dataset is open coded the researcher groups these codes into themes. These themes group together and provide a higher level view of a phenomenon emerging from the data. As themes are identified the qualitative researcher writes analytic memos describing the phenomenon using the data that was open coded. The research team discusses

[12] `https://www.atlasti.com/`

[13] `https://www.qsrinternational.com/nvivo/nvivo-products`

these memos, and the open codes or themes, to identify design opportunities and areas for further research. Through this process, we developed design requirements for the Dac-Man prototype, notions of data change in different domains, and ideas to guide our ongoing data collection. For example, after completing the six usability evaluations the first author open coded the data collected. This information was organized into themes then categorized for short or long-term changes to the tool. This draft was then discussed with the two co-authors to synthesize and refine the feedback while identifying areas for further analysis. The results of our two phases of data collection using the methods described emerged from this open coding analysis process.

4 Results

Through our project's user research we gathered a variety of findings. Foundational work from our exploratory semi-structured interviews and interface mockups helped us characterize a problem space and generate initial design requirements for a data change calculation tool. With subsequent data collection we expanded upon these initial insights and evaluated both the prototype of this tool and our depiction of this problem space. We discuss results from each methods to illustrate how our multi-method approach enabled us to evaluate our tool prototype and appraise the larger problem of scientific data change.

4.1 Determining initial design requirements from foundational work

Our exploratory interviews produced an initial set of insights about the role of data change in current scientific workflows. From this information we were able to articulate initial requirements for tools and metrics to help scientists identify and understand data change. Here we discuss findings from this round of interviews, including insights from sample datasets that we acquired, and explore the design requirements we identified from this data for building our change management framework.

4.1.1 Exploratory interview findings

Our initial exploratory interviews highlighted multiple useful pieces of information to help us begin to characterize data change as a problem in scientific work. This information also highlighted areas where technical interventions could aid this work.

Manual, ad-hoc change handling. Currently scientist's methods of handling data change and further processing are mostly manual and ad-hoc. In many cases, subjects acknowledged that they ignore data changes because it is too expensive to understand and process. Users noted that data change issues were becoming critical and more challenging to handle with increasing volumes of data. They recognized that deeper understanding of changes in their datasets is needed to further their science goals.

Multiple categories of change. Users recognized that there is a need to understand data change at multiple levels, i.e., at the level of a) collections – what files have been added, deleted or updated between two versions, b) metadata – what metadata has been added, deleted or updated c) file level – what and how has the data inside a file changed.

Variability in change measurement. The measurements used to quantify change today vary widely – from qualitative to simple statistical measures and/or in-depth relying on domain specific knowledge. For example, with the MODIS earth science data users are aware that data changes in certain geographical regions might matter more than others for the analyses they conduct.

Struggles providing adequate information. Stakeholders producing datasets struggle to provide adequate information on data change from one version of a release to another. End-users of the datasets struggle to identify and assess the data changes and its impact on their analyses as well as the code they write to produce these results. These users are interested in learning about and understanding data changes and being able to judge the impact of changes on downstream research products. For example, did a variable in two Fluxnet releases shift due to a precision change or a wholesale adjustment from miscalibration and what effect does this have on an analytical model the scientist uses.

Need for reliable tools. Interviewees mentioned that they struggled to consistently identify and examine data changes due to the lack of frameworks that detect and process change. They will often just re-process all of their data, but since this is a computationally expensive and human-intensive process they do not end up actually handling data changes too often.

Need for visual representations. Our participants noted that they need interfaces like a version control system's "diff" feature along with interfaces such as MATLAB or other visual analytics tools that would allow them to study change

using generalized tools in combination with their domain specific methods and metrics.

4.1.2 Insights from sample datasets

During our exploratory interviews we collected sample datasets where we could examine changes between two versions. These sample datasets revealed that numerous domain-specific forms of data change are inherent when producing statistical metrics. This information was utilized by our team members investigating statistical metrics of change. The initial goal was to develop metrics that are applicable to specific datasets from different domains so these real world examples supported this work. The insights gained from these interviews and sample datasets underscored that the volume and complexity of these datasets necessitates a generalized framework to handle calculations of change at large scales in domain-specific settings.

Dac-Man development work was shaped by the SDSS[14] cosmology dataset. SDSS releases are organized into hierarchical folders with key metadata embedded in folder and filenames. File types are consistent and easily able to be opened and interrogated through existing software libraries with minimal effort. This project also mirrors copies of each release across multiple computing facilities and has a need to ensure that each copy is identical. The scientists who provided this data also articulated clear questions they were looking to get answers to about potential data changes in their releases. These questions included: i) Which directories and files were identical in two versions? ii) Which directories had identical structures at a given level but had different file contents? iii) Which files differ and how they were different? iv) Were there any missing directories or files?

4.1.3 Synthesizing design requirements and developing Dac-Man

Analyzing the findings from the semi-structured interviews, the complexity of our sample scientific datasets, and the need for scalability, we synthesized a set of design requirements for our change management framework [6]. The insights from our exploratory interviews heavily emphasized the domain-specific nature of nuances to data change. They also identified specific opportunities to address general filesystem level data changes that emerge across domains. We identified the following five design requirements for Dac-Man .

Identify and classify different general types of changes. Interviewees noted various categories of changes. Data change tools need to be able to identify changes in entire collections, the metadata of files and folders, and the data values inside files.

Support handling increasing volumes and complexities of the underlying data. Interviewees emphasized that calculating data changes is often too time consuming given the large size of their datasets while also noting the difficulties they face comparing anything other than the full releases. Tools need to handle variable amounts of analysis (from two files to two whole datasets) based on a user's need at any given moment.

Allow domain-specific change analyses through plugins. Scientists indicated calculating changes needs to be possible no matter which file format they use and at large scales. Frameworks need to support plugins that define methods to calculate domain-specific changes.

Adapt to different computing environments, including desktops and multi-node HPC clusters. Our interviewees work with systems ranging from personal computers to HPCs. Dac-Man needed to function across such systems since the environments in which versions of datasets live can vary widely by domain.

Compare datasets that are not co-located on the same system. Our SDSS use case elicited the need to compare datasets between multiple disconnected, remote computing environments since large data archives are often mirrored across multiple facilities and systems. With the massive volumes of data, transferring full datasets across networks to perform a change comparison would not be easily doable or rational. This led to the requirement that change management tools need to implement efficient methodologies for identifying and capturing change information that can function across multiple computing sites.

The insights and design requirements resulted in the implementation of our Dac-Man framework that identifies changes in scientific datasets at large scales. Dac-Man (DAta Change Management [6]) allows users to efficiently identify and capture changes between multiple versions of datasets. The framework detects changes of different types and granularities while providing a command-line utility to track and capture changes on systems ranging from desktops to HPCs. Dac-Man can also be extended with plugins to support domain-specific change analyses. These design requirements were implemented in our prototype of the Dac-Man framework. Dac-Man allows users to efficiently identify and capture changes between multiple versions of datasets through a command-line utility that runs on desktops and HPCs. The framework detects changes of different types and granularities. Table 1 lists the different command-line options in

[14] `https://www.sdss.org/`

Command	Description
`dacman diff [options] OLDPATH NEWPATH`	Retrieves changes between two data versions.
`dacman compare [options] OLDPATH NEWPATH`	Computes and caches change information.
`dacman index [options] PATH`	Indexes data for speeding up change capture.
`dacman scan [options] PATH`	Scans and saves filesystem metadata and structure.

Table 1: Dac-Man command-line options.

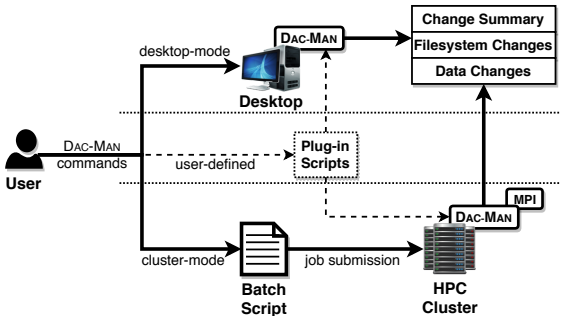


Figure 4: User interaction with Dac-Man on different platforms. On desktops, users directly use Dac-Man commands to capture changes from different versions of a dataset. On HPC clusters, users submit the Dac-Man commands through a batch script, and Dac-Man uses MPI for parallel change capture. Based on the user commands, Dac-Man generates and saves both filesystem and data changes.

Dac-Man that evolved as a result of our user research findings. Users retrieve the changes between two datasets through the command-line utility using the `dacman diff` command. By default, it allows users to retrieve changes at a filesystem level, including directories and files. It also allows users to retrieve changes in data values using the `--datachange` option. Users can scale change analysis to an HPC cluster using the MPI runtime as `--e mpi`.

Our design requirements led us to implement several optimizations in Dac-Man to speedup change detection in large scientific datasets. When a user runs `dacman diff` on two versions of a dataset, the versions are automatically indexed and the change results are saved in a cache. When new data comes in, users can explicitly index it using `dacman index`. Indexing can also be done on an HPC cluster by selecting the appropriate options through the Dac-Man command-line. This option of explicitly indexing the datasets also provides users the flexibility to compute changes in remote datasets, without the need for moving the large datasets to a single system. Users can also recompute the changes using `dacman compare`, which allows them to update the results saved in the cache when the underlying datasets change.

User research also helped us identify the usage model of the framework, allowing us to build a tool that can span from a desktop to a multi-node HPC cluster. Figure 4 describes how a user interacts with Dac-Man command-line tool on different platforms. Users can directly use the Dac-Man commands on desktops. On HPC clusters, users can submit the Dac-Man commands through a batch job script, allowing it to scale to multiple nodes using MPI. As an output, Dac-Man saves a detailed report on the changes at the filesystem, and also provides a summary of changes in the datasets. As for the data changes, the results are outputs of the data change plugins run by Dac-Man . Our usability evaluations helped us assess how well this usage model matches user expectations and identify various issues that we refined in our initial release.

4.2 Insights from exploratory user interface prototyping

Our initial set of low fidelity user interface mockups took Dac-Man outputs comparing two SDSS releases and created a web-based representation. Discussing these prototypes with data science colleagues emphasized the need to provide flexible filtering options and the ability to search through change outputs. In our initial design, we explored the use of a treemap visualization of the filesystem structures. The intention was that change information would be conveyed through different colors to give users a sense of the location of changes among their dataset's folder structure. Discussions strongly indicated that this idea was not very relatable to our users since they either did not necessarily have hierarchical folder structures or understand treemaps as a visualization. This early feedback influenced the revised designs we used in our domain scientist interviews. We shifted the focus to presenting simple plots of Dac-Man 's change calculations and/or our colleague's statistical change metrics (see Figure 2 for examples where subjects marked

up printed copies of these mockups). Discussions with these revised mockups during our interviews elicited potential things (e.g. file types, file sizes, etc.) users would want to filter and search for in the tool's outputs. Work to refine and build an interactive user interface is an ongoing effort in the Deduce project.

4.3 Findings and impacts from usability testing Dac-Man

Our evaluative user research took place after the initial version of Dac-Man was built and our team developed exploratory statistical change analyses for different domains. The six usability evaluations we conducted were designed to examine these products as well as our conceptualization of data change while also generating additional foundational insights about this issue. Many insights about the first version of Dac-Man and data change were gathered and overall we learned that this tool offers potential as a general purpose framework for calculating data change at scale.

4.3.1 Strengths of the Dac-Man prototype

Our usability evaluations affirmed that Dac-Man is useful for scientists working with large numbers of files organized in hierarchical folder structures. All six participants were able to install and run Dac-Man on their local machine, or an HPC system at NERSC[15], and use the tool to calculate changes in a dataset. Our two participants testing with SDSS data were able to install the tool and identify changes across different releases while offering useful feedback about their desire to be able to filter outputs of the tool. One SDSS subject also successfully tested Dac-Man 's support for comparing mirrors of data across two remote archives (our final design requirement). They were able to generate change comparisons for a NERSC archive and a second archive on the east coast of the US.

These evaluation sessions also supported findings from our domain science interviews and exploratory mockup discussions that scientists in every domain do not always organize their data in hierarchical folders. We recognize that such users will currently have to do some manual work to ensure data is appropriately passed to Dac-Man for comparison. This is an opportunity where designing plug-ins and templates might help facilitate such analyses.

4.3.2 Terminology and documentation requirements

The Deduce project faces an ongoing challenge of identifying the appropriate terminology for explaining different types of data change across disciplines and contexts. Our usability findings surfaced a variety of terms for clarification in both the tool's commands and outputs as well as the documentation we are producing.

User feedback identified multiple opportunities to clarify some of the terms and commands in Dac-Man . Originally the third command `compare` was named `calculate-change`. With the original name participants in the evaluations expressed confusion about the difference between this step and the final `diff` command. In another case the flag that enables a Dac-Man user to specify the location to place indexes of the datasets being compared was `-D` and described as the Deduce directory in the README. The six testers were unsure what this flag meant for their use of this tool so we renamed the flag to `-s` to refer to the staging directory. This revision helped make it clearer that the directory stores temporary outputs necessary for the tool to run.

A recurring point of confusion that annoyed testers once they understood the behavior was the way Dac-Man runs commands based on the description in our README. Dac-Man was design with four key commands to break down the steps to calculate change so that calculations can be successfully completed at large scales where some calculation steps make take significant amounts of time or be run across different computing sites and then merged together. The final command (`dacman diff`) was designed to run the three preceding commands if they had not been run before. This caused confusion and consternation among testers since our README did not explain this behavior clearly up front. Participants noted they would have just run the diff command if they were made aware of this feature. This would have been fine during these usability evaluations since testers were trying the tool with smaller datasets where scale and speed were not a concern since all four commands executed rapidly. In contrast, if a participant did have a large dataset to test then running each command individually would have illustrated the varying amounts of time each command can take in different scenarios. This finding illustrated that we need to balance our terminology and presentation of it to make it easy to run the tool in simple cases while keeping key details accessible for more complex workflows.

The documentation we had testers use during evaluation sessions had basic installation instructions as part of the README. These instructions were complicated, explaining multiple ways of setting up Python environments to help run the tool. We simplified these instructions in the final release. Confusion with terminology also highlighted the need for a quick start guide to introduce Dac-Man to users to the tool through simple use cases, with more detail available in other documentation for more complex use cases. We placed this guide within the code repository hosting this project and included a step-by-step example of how to run a change calculation with Dac-Man and illustrated the outputs a user should expect. We also produced a user guide that complements the code repository's README with more detail about the tool's design, functionality, commands, and advanced features.

[15]National Energy Research Scientific Computing Center, https://www.nersc.gov

4.3.3 Dac-Man outputs

Dac-Man was designed to output information about filesystem and data level changes. Each of our testers was able to parse the prototype's outputs about filesystem changes (additions, deletions, etc.) between two datasets. Their feedback identified additional useful information to potentially include. Suggestions included details such as file hashes, file modification times, and the usernames who made a modification.

Many of our testers had a difficult time understanding the presentation of what is change at a filesystem level versus data level in the tool's outputs. This was partially because the prototype was writing operational and debug status information to the command line instead of a log file. All of this verbose output obfuscated the important line of text that explained where the change calculation outputs were were stored in text files. This confusion prompted our team to simplify Dac-Man 's text outputs overall. First we minimized command line outputs and second we reorganized the tool to write out a log file with detailed information on Dac-Man 's execution. This ensured that key information about changes is immediately accessible while extensive details are available for the interested user.

4.3.4 Change metric

The Dac-Man prototype was also built in part to explore potential metrics of data change. The version our six subjects tried out calculated an overall change metric that was included in the tool's text outputs. This overall change metric attempted to summarize all of the changes between datasets in one numeric value as a weighted percentage. Our intention here was to explore how we could convey the magnitude of changes to users with minimal overhead. However, the usability evaluations (as well as both interviews and conversations about mockups) emphasized that this version of the metric was not helpful to scientists. Feedback underscored that a single calculated number is not comprehensible to users without context about the calculation process. The six testers and our other subjects interviewed were intrigued by the idea of a percentage change calculation. Additionally, recognized and commented that calculating this will be difficult with widely disparate datasets. Having a factor that they can use to reliably compare magnitudes of change between releases was intriguing but is challenging to design and implement and remains an area for future work since domain-specific factors are key.

4.4 Results from surveying flux data stakeholders

The responses to our final survey in 2019 produced varying insights about our flux data change analyses. Overall we received a total of eight responses from five Principal Investigators who produce flux data and three Research Scientists who use this data. All eight individuals indicated that they use at least one of the two datasets we examined as part of their own research work. Overall, positive feedback was obtained about correlation plots that looked at data changes at specific sites for single variables. Respondents also had positive views about using time series visualizations to convey change.

Across the responses to our survey questions, along with our other user research methods, we have re-framed our view on data change. Rather than see data change as a calculation to be completed and presented we find identifying and characterizing change to be an integral data analysis activity. Multiple responses indicated that a tool to help with identifying changes in datasets would be useful. To earth scientists a flexible tool oriented around producing varying plots was noted as potentially very useful.

4.4.1 Familiarity with & understanding of the two datasets

Among our eight responses each participant indicated that they were well versed in the two datasets we analyzed for changes. With the La Thuile release one person was familiar, two very familiar, and five extremely familiar with this dataset. For the FLUXNET2015 release three researchers were very familiar and five were extremely familiar. We also wanted to know if participants had encountered change in these datasets previously and seven indicated yes. One individual indicated they had not encountered changes and they had previously indicated they only use the FLUXNET2015 release so there would not be changes to identify.

The ways respondents analyzed changes between releases varied widely, ranging from a simple "don't" or "carefully" to a description of a particular approach. The approaches described included making "quick and dirty plots" to compare then re-run analyses or comparing small selections of data. The software tools used to do this were split between "none in particular" to custom analysis code written in languages from R to IDL or Python and MATLAB.

These responses underscored a finding we've seen across our methods of data collection. Scientist's practices identifying and analyzing changes are ad hoc and rely upon their ability to produce code that can surface this information. These findings also underscore the opportunity for Dac-Man to be useful in earth sciences so long as sufficient plugins are able to be built using languages that members of this community are comfortable with.

4.4.2 Feedback on change analysis plots

Feedback on the seven plots representing different aspects of change analyses were varied. Respondents indications of the usefulness of the change visualizations produced by our team depended on the type of analysis being conveyed, with more specific analyses being better understood and accepted by the individuals responding. Our overall take away that the nuances of performing change analyses is important to scientists remained unchanged and the feedback we received is emphasizing that our user interface for exploring data change must enable users to plot different variables.

The first four plots in the survey were high level overviews of change. These included one illustrating change across the entirety of the data releases, change in a single measurement across multiple physical sites, one site's changes for multiple measurements, and change in one variable at a single site. These four plots demonstrated a range of high level analyses and the survey responses indicated their utility variety widely. The first and second plots were not found to be useful and respondents were ambivalent about its potential utility even if more provenance details about the datasets were somehow included. In contrast, the third and fourth plots were perceived as more understandable and likely somewhat helpful when assessing whether a new release had changed from an old version. The open ended feedback on the overview plots raised many questions about what the change analysis really represented, whether it was realistic given the nuanced nature of many variables within the datasets, and low utility for helping the researchers continue to analyze changes.

The remaining plots in the survey were different visualizations of correlation between values for one variable at a single site when the data was raw and uncleaned. These plots illustrated the impact cleaning could have on changes, for example identifying when values for a variable were all uniformly shifted between an old and new release. This shift could throw an analysis off entirely and identifying such a change with a visualization is a way to stem any issues. The responses for these plots were positive, with respondents indicating that these visualizations help them understand changes between the La Thuile and FLUXNET2015 releases. Questions asking for comments or feedback elicited a variety of hypotheses and questions about the changes presented in the plots. Respondents noted what they thought were data processing errors they could identify from different plots and raised concerns about how they might choose to analyze and use some of this data.

Overall, the Deduce project's understanding that surfacing data change is an increasingly necessary task was supported by the limited survey responses we received. Combined with the insights from our other research methods we see the opportunity and need to advance the design of the Dac-Man framework. Ongoing work is developing a more robust plugin framework and a user interface for configuring change analyses and exploring the results. Feedback from the survey in particular highlights the utility of visualizations but a need for them to be customizable by the scientific users for their domains.

5 Conclusions

The Deduce project's user research has generated multiple types of qualitative data to impact the development of data change tools and metrics. Incorporating these multiple user research techniques in scientific environments helped us handle the challenging, emergent, and evolving nature of scientific work by refining our problem space and products. Our team has better characterized and refined our understanding of data change across multiple rounds of foundational interviews combined with generating user interface mockups and evaluating the usability of a prototype tool and utility of statistical change analyses through a survey. These insights helped our team improve its various research products while also better characterizing how scientists think about and handle change in the large datasets they use in their work.

Our findings illustrate how building useful, usable tools for science requires not only iteratively refining tools but also re-assessing and refining a team's knowledge about the evolving problems and accounting for diverse stakeholder groups. As teams building scientific software we need to flexibly but rigorously incorporate multiple user research methods to frequently gather and reflect on foundational, generative, and evaluative insights. Designing user research studies to accommodate the diversity of scientific domains is a constant process of balancing project demands, types of users, opportunities for data collection, and determinations about what, when and how the data is collected for qualitative research. We have enhanced the usability of Dac-Man and identified future development directions for the tool and the project by combining interviews, mockups, and usability evaluations. Our experiences provide a strong foundation to continue to explore rigorous, reliable user research processes that accommodate the diversity of scientific research and associated software projects in a timely way.

Acknowledgements

The authors wish to thank the members of the Deduce team, our science collaborators, and our study participants for their insights and feedback. This work is supported by the U.S. Department of Energy, Office of Science and Office of Advanced Scientific Computing Research (ASCR) under Contract No. DE-AC02-05CH11231.

References

[1] Aragon, C. R., Poon, S. S., Aldering, G. S., Thomas, R. C., and Quimby, R. Using visual analytics to develop situation awareness in astrophysics. *Information Visualization 8*, 1 (2009), 30–41.

[2] Arora, B., Dwivedi, D., Faybishenko, B., Jana, R. B., and Wainwright, H. M. Understanding and Predicting Vadose Zone Processes. *Reviews in Mineralogy and Geochemistry 85*, 1 (09 2019), 303–328.

[3] Buxton, B. *Sketching user experiences: getting the design right and the right design*. Morgan kaufmann, 2007.

[4] Charmaz, K. *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*, 2nd ed. Sage, 2014.

[5] Faybishenko, B. Detecting dynamic causal inference in nonlinear two-phase fracture flow. *Advances in Water Resources 106* (2017), 111 – 120. Tribute to Professor Garrison Sposito: An Exceptional Hydrologist and Geochemist.

[6] Ghoshal, D., Ramakrishnan, L., and Agarwal, D. Dac-man: Data change management for scientific datasets on hpc systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (Piscataway, NJ, USA, 2018), SC '18, IEEE Press, pp. 72:1–72:13.

[7] Hall, E. *Just Enough Research, 2nd Edition*. A Book Apart, New York, NY, 2019.

[8] Herzon, C., DeBoard, D., Wilson, C., and Bevan, N. Contextual inquiry, Oct 2010.

[9] Macaulay, C., Sloan, D., Xinyi, J., Forbes, P., Loynton, S., Swedlow, J. R., and Gregor, P. Usability and user-centered design in scientific software development. *Software, IEEE 26*, 1 (2009), 96–102.

[10] Paine, D., Ghoshal, D., and Ramakrishnan, L. Experiences with a flexible user research process to build data change tools. *Journal of Open Research Software* (in press).

[11] Paine, D., and Ramakrishnan, L. Surfacing data change in scientific work. In *International Conference on Information (iConference 2019)* (2019), Springer.

[12] Poon, S. S., Thomas, R. C., Aragon, C. R., and Lee, B. Context-linked virtual assistants for distributed teams: an astrophysics case study. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (1460623, 2008), ACM, pp. 361–370.

[13] Ramakrishnan, L., Poon, S., Hendrix, V., Gunter, D., Pastorello, G. Z., and Agarwal, D. Experiences with user-centered design for the tigres workflow api. In *2014 IEEE 10th International Conference on e-Science* (2014), vol. 1, pp. 290–297.

[14] Sharp, H., Dittrich, Y., and de Souza, C. R. B. The role of ethnographic studies in empirical software engineering. *IEEE Transactions on Software Engineering 42*, 8 (Aug 2016), 786–804.

[15] Sharp, H., Rogers, Y., and Preece, J. *Interaction design: beyond human-computer interaction*, 2nd ed. John Wiley and Sons Inc., 2007.

[16] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., and Diakopoulos, N. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th Edition)*, 6th ed. Pearson, 2016.

[17] Weiss, R. S. *Learning From Strangers: The Art and Method of Qualitative Interview Studies*. The Free Press, New York, NY, 1995.

Appendix A Flux Analysis Survey Questions

The survey evaluating change analyses of the La Thuile and FLUXNET2015 datasets is included as an appdenxi. This survey was distributed via Google Forms and included a human subjects consent page, overview questions, description of the analysis that generated the plots, and a series of plots with questions about their utility.

Examining change in flux data releases

Researchers from the Data Science and Technology Department at Lawrence Berkeley National Lab (LBNL) are conducting interviews with scientific researchers who work with large complex datasets to better understand how they find and select data products to use in their research, and how they assess changes between different versions of these data products. The information gathered in this research will inform the design of a dynamic data integration framework and metrics for conveying data change to support data analyses and products.

This survey is designed to help us understand how you handle data change in your work and obtain feedback on ways of analyzing change between different versions of ecosystem carbon, water, and energy flux datasets. During this survey you will be asked about your work with Flux datasets and to answer questions about our team's analyses of changes in the La Thuile and FLUXNET2015 public data releases.

You may navigate between sections at any time and results should be saved.

We estimate that this survey should take around 15-20 minutes total.

** Participation and Rights as a Research Subject **
Your participation in this research is voluntary, and you are free to quit the survey at any time. You will not be paid for your participation, nor will you be charged. There will be no direct benefit to you from taking part in this study. However, the information gain from the study will inform the design of data change analysis tools which may be able to support your research. Participation may result in a loss of privacy but your records will be kept as confidential as possible under the law. Results from the survey will be included in publications to be presented to the researchers on this project and scientific communities. Your identity will not be included in the report nor will your name be associated with any session data collected. The survey's final question gives you the option to provide your name and email for us to contact you for more information about your work with flux data. This question is entirely optional and the results will not be shared with anyone outside of our team. If asked to participate further you may freely refuse to do so.

If you have any further questions about taking part in this study you may contact Dr. Lavanya Ramakrishnan at or Dr. Deborah (Deb) Agarwal at .

Any questions you have about your rights as a research subject will be answered by the Berkeley Lab Human Subjects Committee at .

* Required

1. **I have read the above project description. I agree with the terms and hereby consent to participate in the study. If you do not agree please exit the survey now.** *
   Check all that apply.

   - [ ] Yes

Overview Questions

Please let us know a bit about your experiences with flux datasets.

2. **What flux data releases do you use in your work?** *

3. **How familiar are you with the La Thuile dataset?** *
   Mark only one oval.

	1	2	3	4	5	
Not At All Familiar	◯	◯	◯	◯	◯	Extremely Familiar

4. **How familiar are you with the FLUXNET2015 dataset?** *
   Mark only one oval.

	1	2	3	4	5	
Not At All Familiar	◯	◯	◯	◯	◯	Extremely Familiar

5. **What is your relationship to this flux data?**
   (e.g. PI, research scientist, student) *

6. **Have you encountered changes in the flux data releases you employ in your research?** *
   Mark only one oval.

   - ◯ Yes
   - ◯ No
   - ◯ Maybe
   - ◯ Unsure

7. **How do you analyze changes between flux data releases currently? Are there any software tools or techniques that you use to analyze change?**

Analysis of change in flux data releases

In the remaining sections of this survey we present plots from analyses comparing the LaThuile and FLUXNET2015 data releases.

Instructions

For this portion of the survey we would like you to examine descriptions of analyses and the resulting plots. For each analysis we have a few questions for you to provide your reactions to these plots.

Our goal is to learn how these analyses and visualizations are or are not useful to you as a scientist working with flux datasets.

Overview of our analysis approach

For this analysis, our team member compared the LaThuile and FLUXNET2015 variables listed below in Table 1.

Both datasets were downloaded from the FLUXNET fluxdata portal as CSV files.

- The La Thuile dataset was retrieved from http://fluxnet.fluxdata.org/data/la-thuile-dataset/
- The FLUXNET2015 dataset was retrieved from http://fluxnet.fluxdata.org/data/fluxnet2015-dataset/

The LaThuile and FLUXNET2015 datasets have very different structures as a result of different data processing decisions. In particular, the variable and file names were changed between each release and must be properly mapped to enable a comparison. Doing this required us to develop processing scripts so that analyses could be conducted and the plots you are reviewing created.

The data in each set consist of 30-minute data records of meteorological observations: carbon dioxide, water vapor and energy fluxes taken at the monitoring sites spanning the globe, and which were collected and standardized in the FLUXNET synthesis database.

There are a total of 426,855,663 measurements in the full intersecting dataset where there are measurement values from both the La Thuile and FLUXNET2015 datasets.

Of the 426,855,663 measurements available, 405,175,139 (94.9%) of them changed between LaThuile and FLUXNET2015 when compared as text and not numerical values.

For the La Thuile dataset we used the publicly available free, fair-use release. This dataset consists of a set of 99 unique variables (i.e., types of measurements and quality factors associated with some of them) measured at 30-minute intervals at 153 stations, with some (at least one) stations not reporting all the variables. In total 197,598,956 measurements were taken covering a time span from January 1, 1991, to January 1, 2008.

For FLUXNET2015 we used only the Tier One dataset, which is open and free for scientific and educational purposes under the same fair use policy as La Thuile. This dataset consists of 143 unique variables, which were measured at 30-minute intervals at 156 stations. In total 320,753,418 measurements were taken covering a time span from January 1, 1994, to January 1, 2015.

Table 1. LaThuile and the corresponding FLUXNET2015 variables compared between the two data releases.

La Thuile		FLUXNET2015	
Name	**Description**	**Name**	**Description**
LE_f	Latent heat flux	LE_F_MDS	Latent heat flux, gapfilled using MDS
H_f	Sensible heat flux	H_F_MDS	Sensible heat flux, gapfilled using MDS
G_f	Soil heat flux	G_F_MDS	Soil heat flux
Ta_f	Air temperature	TA_F_MDS	Air temperature, gapfilled using MDS
VPD_f	Vapor pressure deficit	VPD_F_MDS	Vapor Pressure Deficit, gapfilled using MDS
Precip_f	Precipitation	P	Precipitation
WS_f	Wind speed	WS	Wind speed
PPDF_f	Photosynthetic Photon Flux Density	PPDF_IN	Photosynthetic photon flux density, incoming
Rn_f	Net radiation	NETRAD	Net radiation
LWin	Longwave incoming radiation	LW_IN_F_MDS	Longwave radiation, incoming, gapfilled using MDS
LWout	Longwave outgoing radiation	LW_OUT	Longwave radiation, outgoing
SWin	Shortwave incoming radiation	SW_IN_F_MDS	Shortwave radiation, incoming, gapfilled using MDS
SWout	Shortwave outgoing radiation	SW_OUT	Shortwave radiation, outgoing
WD	Wind direction	WD	Wind direction
ustar	Friction velocity	USTAR	Friction velocity
Rh	Relative humidity	RH	Relative humidity
CO2	Carbon dioxide concentration	CO2_F_MDS	CO2 mole fraction, gapfilled with MDS

8. **Do you understand this analysis process?** *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe
- ◯ Unsure
- ◯ Other: _____

9. **Do you have any thoughts or feedback on the process described?**

_____

_____

_____

_____

_____

Plots to evaluate

This section contains seven plots that we would like feedback on.

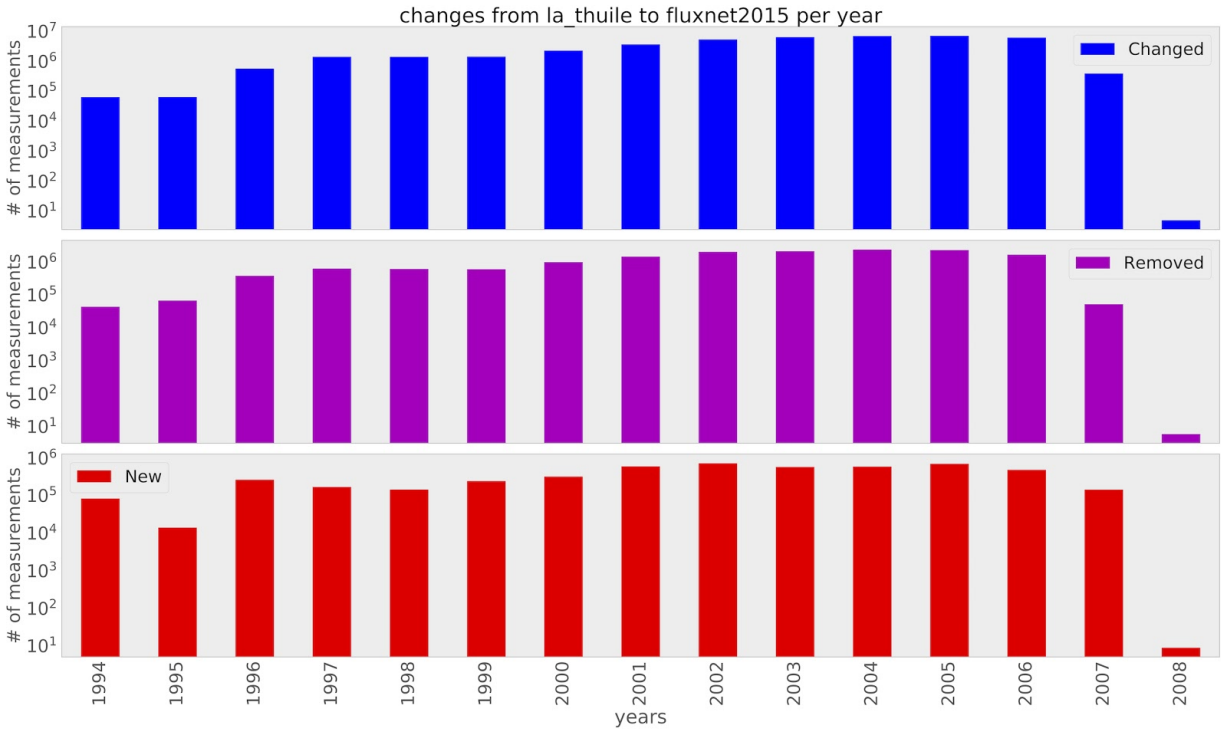
Plot 1 - Overview of changes between La Thuile and FLUXNET2015.

We grouped and plotted the intersecting set of measurements between LaThuile and FLUXNET2015 into three categories, namely:

- Changed: measurements which had values change between datasets.*
- Removed: measurements which have a value in the LaThuile dataset but not in FLUXNET2015.
- New: measurements with values only in the FLUXNET2015 dataset.

*The LaThuile dataset includes numeric values expressed as multi-decimal numbers (up to 11 decimals), while the corresponding values in the FLUXNET2015 dataset are given with only two decimals. This problem was resolved by restating the precision of the LaThuile values to that of the corresponding FLUXNET2015 ones before converting back to a string type for comparison. A total of 22,029,953 (about 5.2%) values were affected by this problem.

Plot 1. Overview of changes between La Thuile and FLUXNET2015



10. **What are your key takeaways from this plot?** *

11. **With this information how would you continue to analyze changes?** *

12. **This plot helps me assess the changes between data releases.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

13. **This type of plot would be useful when comparing two data releases.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

14. **Do you have any additional feedback about Plot 1?**

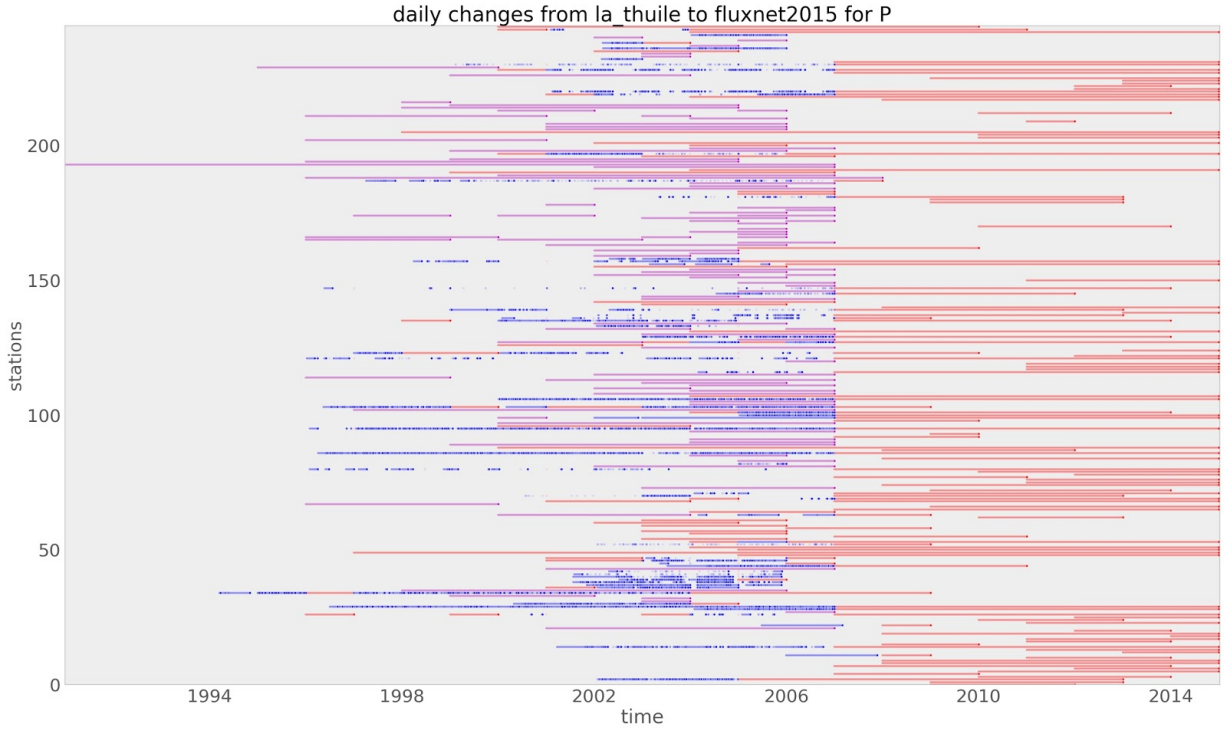
Plot 2 - Change in the Precipitation (P) measurement per station over time.

Plot 2 illustrates the changes of Precipitation (P) data for each station over time. This is an example of visualizing data changes both by type of measurement and by station.

Stations are plotted on the y-axis by numeric identifier. Years are plotted on the x-axis.

- Blue lines indicate Changed data.
- Magenta lines indicate Removed data.
- Red lines indicate New data.

Plot 2 - Change in the Precipitation (P) measurement per station over time.



15. **What are your key takeaways from this plot?** *

16. **This plot helps me assess the changes between data releases.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

17. **Do you have any additional feedback about this type of plot?**

Plot 3 - One station's changes per day for multiple measurements.

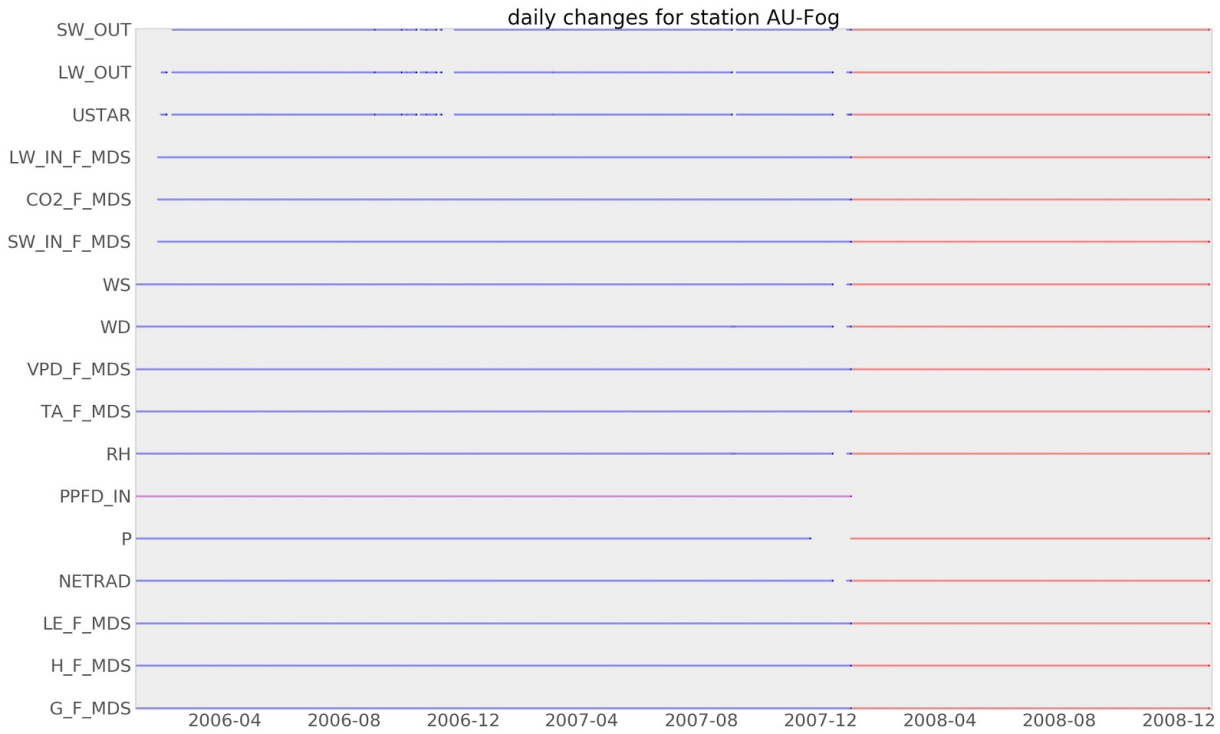
Our third analysis examined whether there were simultaneous changes in different measurements at any given time for a particular station. The purpose of this analysis was to explore whether any measurements changed together as a group or whether individual measured parameters changed independently of others.

Plot 3 illustrates this approach by visualizing changes in measurements collected at the AU-Fog station on a daily basis over a multiple year timespan.

Different variables measured are presented on the y-axis (Abbreviations on the y-axis are given earlier in Table 1.). Days are presented on the x-axis.

- Blue lines indicate the data Changed.
- Magenta lines indicate data was Removed.
- Red lines indicate the New data was added.

Plot 3 - One station's changes per day for multiple measurements.



18. **What are your key takeaways from this plot?** *

19. **This plot helps me assess the changes between data releases.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

20. **Do you have any additional feedback about Plot 3?**

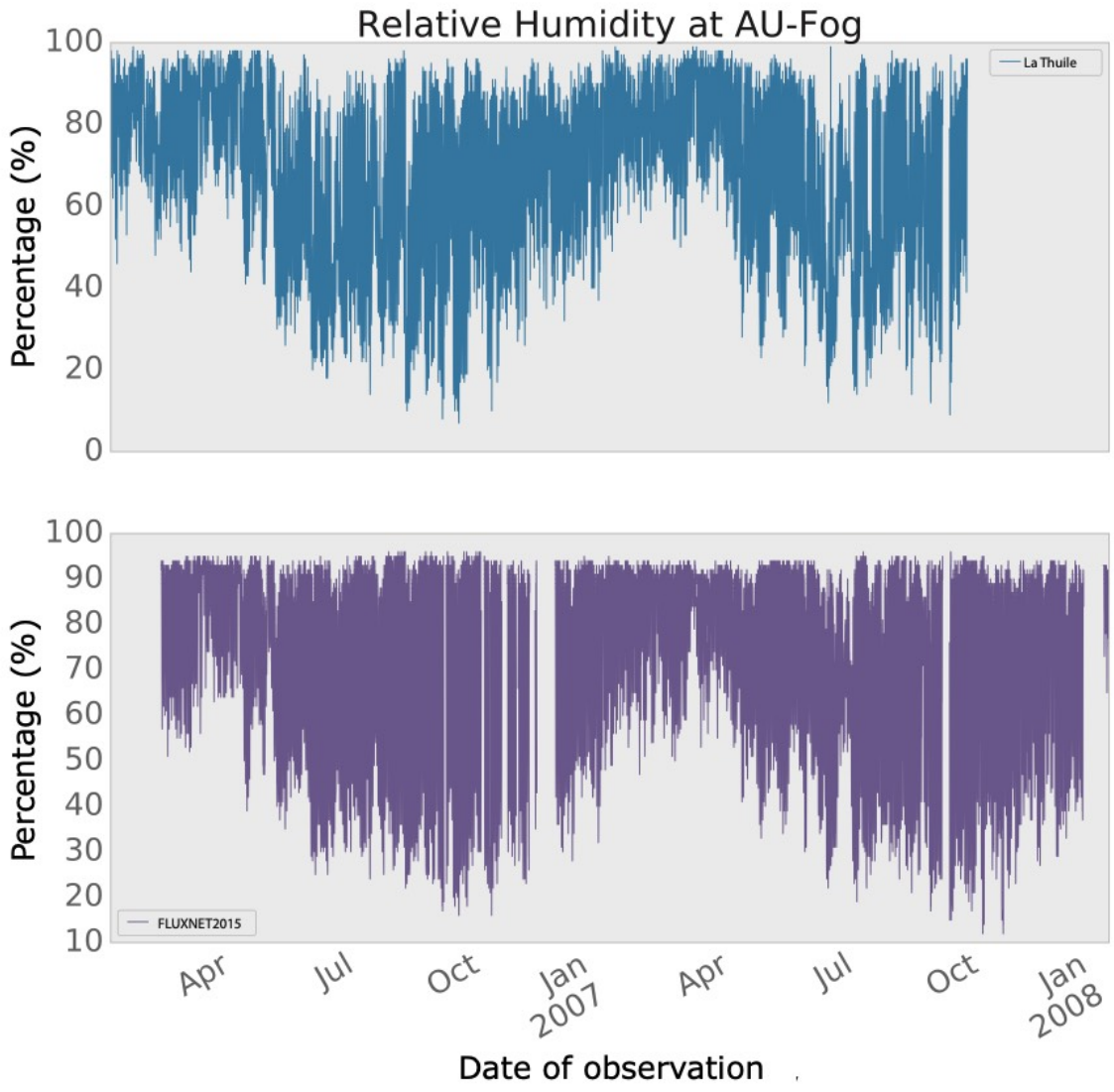
Plot 4 - Conveying different lengths, temporal patterns, and gaps in data between two releases.

Our analysis of the LaThuile and FLUXNET2015 datasets identified different lengths to each release, the presence of gaps in data, and different temporal patterns. An example of these issues is presented in Plot 4.

Plot 4 graphs a time series of relative humidity taken at the AU-Fog site to illustrate potential issues due to different release lengths, gaps in data, and varying temporal patterns.

Upper figure – La Thuile dataset (29,992 observations)
Lower figure – FLUXNET2015 (30,919 observations)

Plot 4 - Conveying different lengths, temporal patterns, and gaps in data between two releases.



21. **What are your key takeaways from this plot?** *

22. **This plot helps me assess the changes between data releases.** *

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

23. **Do you have any additional feedback about Plot 4?**

Plot 5 - Identifying shifts in measurement values between releases.

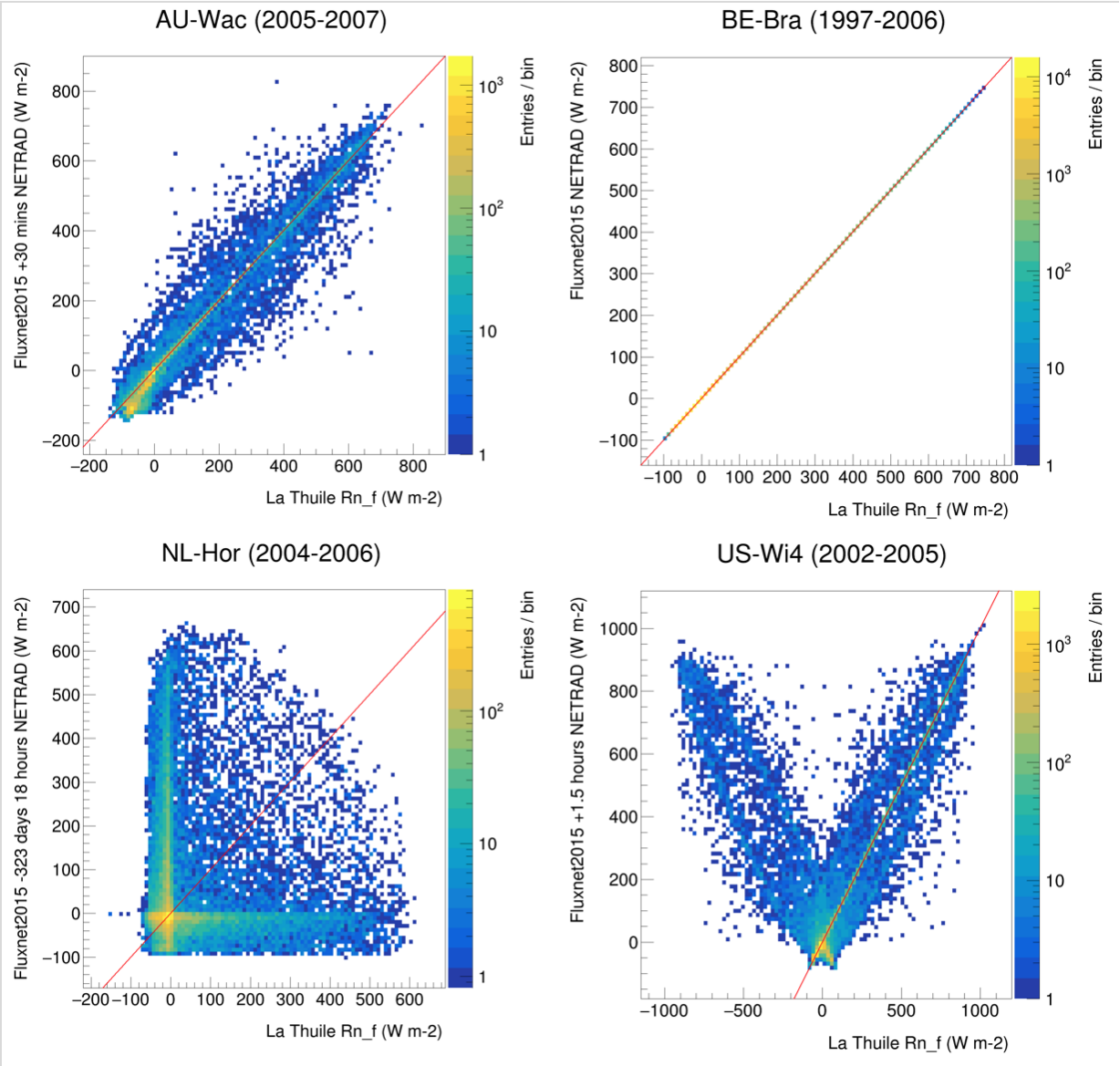
Visually examining the FLUXNET2015 and La Thuile datasets showed that this time series data might have been time shifted with respect to each other. We speculated that this may be the result of different processing decisions for each release.

To assess this hypothesis, we plotted x-y graphs for each variable, where the coordinates provided the values from the two data releases. The x-axis is generated from FLUXNET2015 releases time shifted by some quantity indicated. The y-axis is generated from La Thuile data.

For identical data releases we expected the x-y plot to exhibit a straight line with a slope of 1:1 and an intercept at 0. The spread of points around the 1:1 line indicates the difference between the corresponding values in both datasets.

Plot 5 illustrates four examples of 1:1 correlation plots for Net Radiation measurements at different monitoring stations. This example presents a subset of possible comparisons that may be possible to generate.

Plot 5 - Example plots for identifying shifts in net radiation measurement values between the two releases.



24. **What are your key takeaways from this plot?** *

25. **This plot summary helps me easily grasp the change between data releases.** *

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

26. **Do you have any additional feedback about Plot 5?**

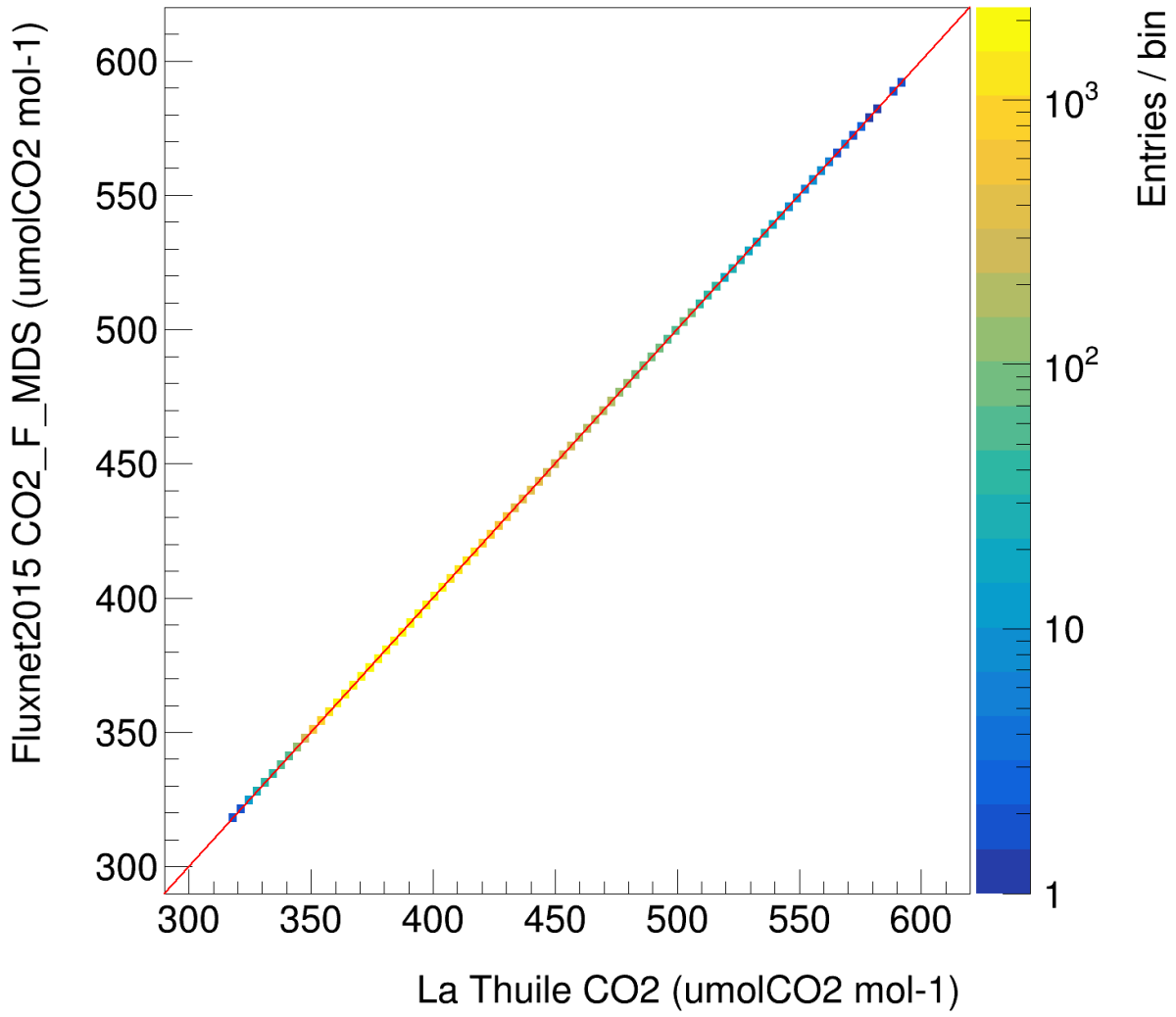
Plot 6 - Two detailed examples of measurement shifts between the La Thuile and FLUXNET2015 releases.

To further demonstrate potential analysis of the effects of time shifts we have two examples.

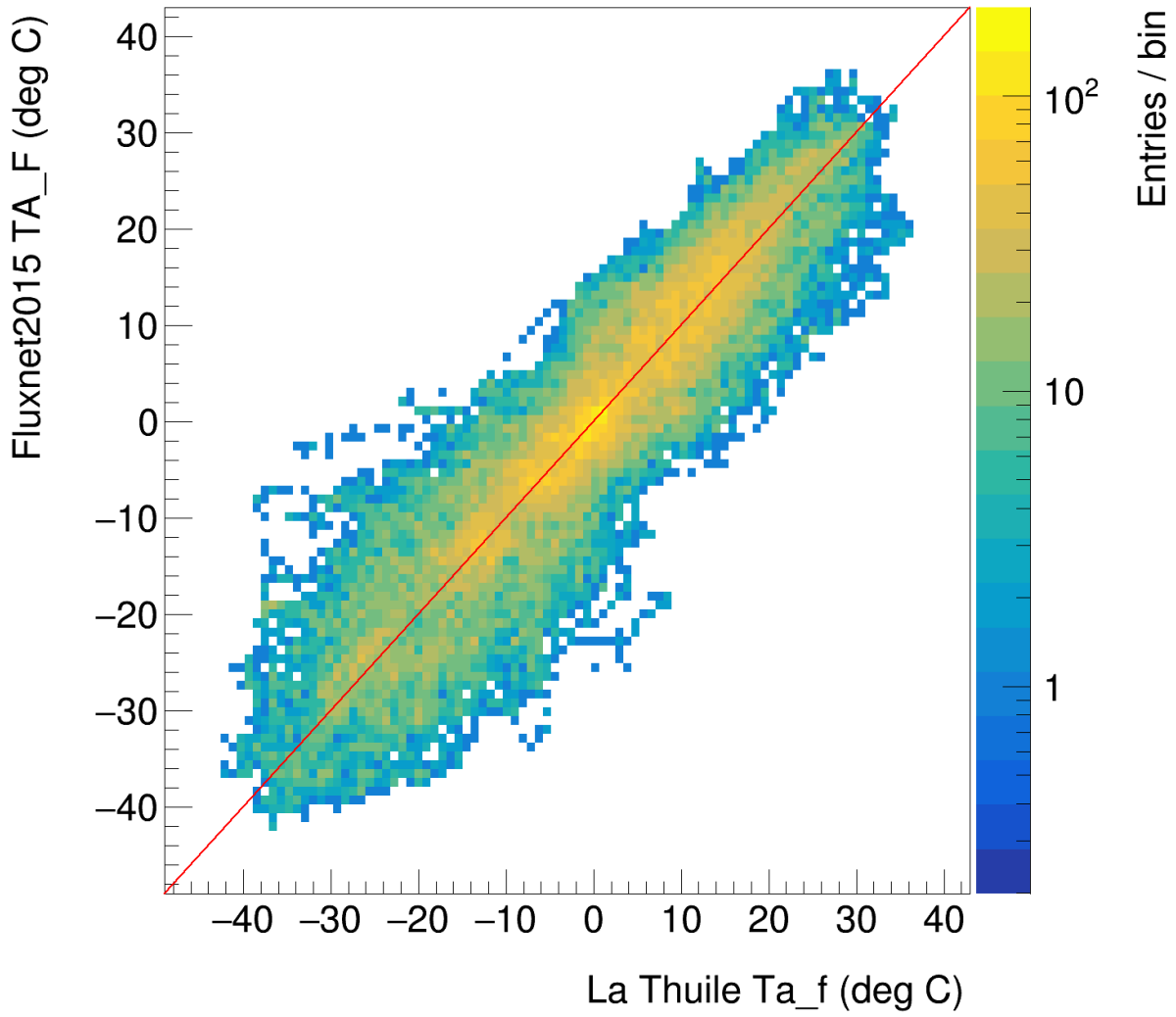
The first, Plot 6a, is a perfect 1:1 correlation of CO2 concentration captured at the AT-Neu site from both data releases.

The second, Plot 6b, a scatter of temperature values around the 1:1 line demonstrating the weak correlation of temperature values at the CA-NS4 site between these data releases.

Plot 6a. A perfect 1:1 correlation of CO2 concentration at the AT-Neu site from 2002-2004 between the La Thuile and FLUXNET2015 releases.



Plot 6b. Weak correlation of uncleaned temperature data at station CA-NS4 from 2002-2004 between the FLUXNET2015 and La Thuile releases.



27. **What are your key takeaways from these plots?** *

28. **These plots help me easily grasp the changes between data releases?** *

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

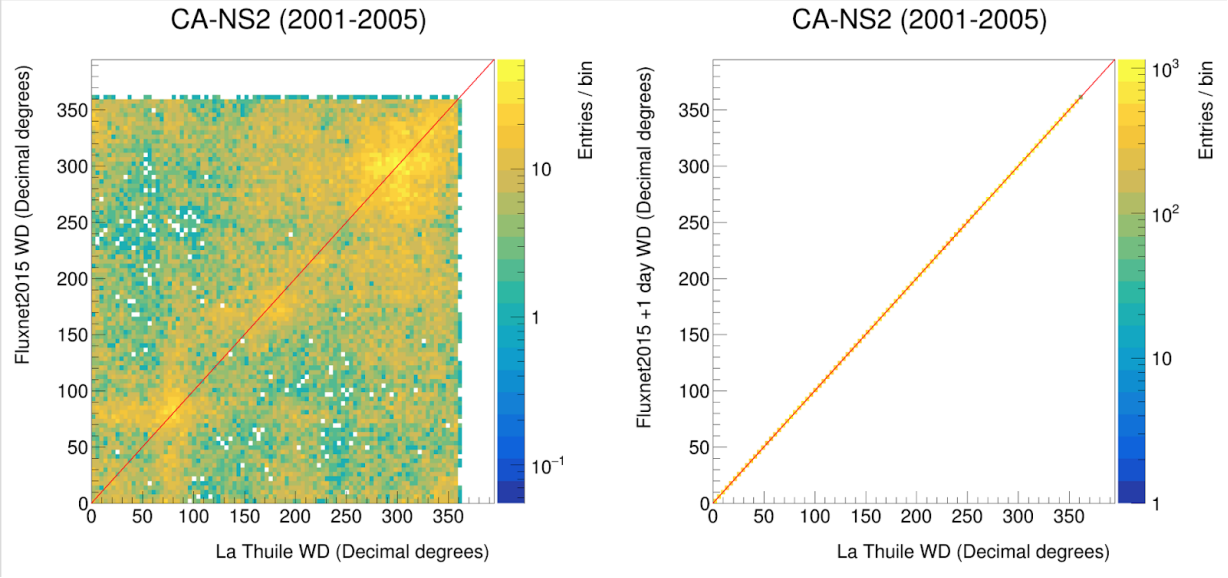
29. **Do you have any additional feedback about Plots 6a and 6b?**

Plot 7 - Visualizing the effect of an adjustment to time series values between releases.

Knowing different values may be time shifted we examined the effect of an adjustment as an example of a potential impact of changes to data processing.

Plot 7 is an example of correlation plots for the CA-NS2 site's Wind Direction values. The left figure presents the plot of no correlation between the wind direction at CA-NS2 when comparing the downloaded values of each release. The right figure presents a perfect correlation after time shifting the FLUXNET2015 data one day ahead with respect to the La Thuile time series.

Plot 7 - Visualizing the effect of an adjustment to time series values between releases.



30. **What are your key takeaways from this plot?** *

31. **Do you have any additional feedback about Plot 7?**

Wrap-up

Thank you for sharing your feedback on these plots and analyses. To wrap-up we have a few concluding questions.

32. **Are there additional metrics and plots of data change that would be helpful for your work?** *

33. **Is there anything else you think we should know?** *

34. **If you are willing to have us contact you to learn more about your work with flux data please provide your name and e-mail address. This information will remain confidential to the Deduce team and will not appear in any reports.**

Powered by
Google Forms