

# “Who Has Plots?”: Contextualizing Scientific Software, Practice, and Visualizations

DREW PAINE AND CHARLOTTE P. LEE, Human Centered Design and Engineering

---

Software is an integral element of the work of science yet it is not commonly an object of inquiry in studies of scientific infrastructures. This paper presents findings from an ethnographic study of a cosmology group’s collaborative scientific software production. We demonstrate how these cosmologists use plots to simultaneously test their software and analyze data while interrogating multiple layers of infrastructural components. We broaden perspectives on scientific software development using a sociotechnical, software studies lens to examine this work of scientific discovery as a creative and embodied, yet exacting and methodical, activity that requires a ‘human in the loop’. We offer a new reading of scientific software practices to convey how creating scientific software is often really the act of doing science itself—an intervention we believe is necessary to more successfully support scientific software sharing and infrastructure production.

## CCS Concepts

• **Human-centered computing~Computer supported cooperative work** • **Human-centered computing~Ethnographic studies** • *Human-centered computing~Scientific visualization* • Applied computing~Astronomy

## KEYWORDS

Software studies; research infrastructures; scientific software development; scientific software testing; scientific visualizations; plots; cosmology; epoch of reionization; ethnographic study.

## ACM Reference format:

Drew Paine and Charlotte P. Lee. 2017. “Who Has Plots?”: Contextualizing Scientific Software, Practice, and Visualizations. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 85 (November 2017), 21 pages. <https://doi.org/10.1145/3134720>

---

## INTRODUCTION

Most recent research on scientific software development has been undertaken in fields other than CSCW. While researchers have documented that there are significant cultural differences between scientists and software engineers (SEs) [21, 27, 45, 56], much discourse about scientific software development has prioritized identifying “missing” SE practices and teaching scientists to develop software like software engineers [42, 59, 61, 62]. CSCW and Software Studies’ sociotechnical perspectives provide an alternative way to conceptualize and examine scientific software production that can usefully understand software in scientific work as not primarily a simple implementation of algorithms and invisible labor [58], but rather as essential creative work necessary for characterizing problems and producing the knowledge to address them.

---

\* {pained, cplee}@uw.edu; Human Centered Design and Engineering, University of Washington, Seattle WA 98195

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2017 ACM. 2573-0142/November – Article 85...\$15.00 DOI: <https://doi.org/10.1145/3134720>

CSCW has long investigated collaborative scientific work—from early work by Kraut et al. [31, 32] and studies of collaboratories in the 1990s and early 2000s [16, 46] to the many recent studies of cyberinfrastructure or research/knowledge infrastructures [12, 26, 52]. CSCW’s extensive studies of scientific infrastructure projects often focus on issues of data sharing or designing organizational structures to sustain either the projects or their components over time. Additional work has explored incentives and challenges for sharing scientific software [22–24]. These approaches are impactful. At the same time, more research is needed to investigate software’s unique role in the work of knowledge discovery and to investigate how software is embedded in, supports, and is supported by a myriad of practices and artifacts ranging from printed paper plots (scientific visualizations) to complex, entangled layers of infrastructures. By investigating interactions between cosmologists using and developing software and visualizations—in particular printed plots of data—this work shows the depth (and multidirectional dependencies) and entanglements of practices and artifacts that result in new infrastructures.

Using ethnographic methods, this work investigates the software testing and data analysis work of a group of empirical cosmologists—referred to here by the pseudonym Radio group—who are engaged in building and using multiple layers of infrastructural components to meet their research goals. We are examining the following research question: *How does a cosmology research group use plots to test their data analysis software?*

CSCW scholarship examining data sharing in the sciences has emphasized that sharing requires much more than simply making some artifacts available [3, 15, 53]. In the Radio group we see practices around sharing within the group and with external collaborators using bespoke visualizations (plots) in tandem with discussions about their software, data, and evolving knowledge. The design problem of how to support the “sharing” of software transforms when we take seriously the idea of software as a coded artifact, a discursive practice, lived experience, and so on [29].

## BACKGROUND

Software has been a key element (and product) of the scientific research process for decades. Within CSCW a variety of work studies scientific software and issues of collaboration—from the production of scientific middleware software [2] to investigations of incentives for producing and sharing scientific software [22, 23], boundaries to scientific software sharing [24], and recently an investigation of oceanographic programming practices [33]. There is also an extensive body of Computer Science (CS) research investigating how scientists develop software themselves and in collaborations between scientists and software engineers [4, 21, 28, 54]. But little of this work investigates and theorizes how software expresses scientific knowledge and practices.

This CS research demonstrates that the way scientists develop software is not the same as software engineers since they face different challenges, have different goals, and have different cultures of working—but it does not try to reframe inquiries to ask how the science itself is being accomplished through software. Determining well-scoped requirements at the beginning of a project (the general ideal, but not always true, situation in software engineering) for a piece of scientific software may be difficult since the creation of the software itself is a central element of a collaborative knowledge discovery process [28, 47]. As a result the end goal for a piece of scientific software may not be known at the beginning of a project. Carver [4] suggests that software engineering “best practices” (idealized norms taught and advocated for within the community) may not be applicable to scientific software development (SSD) without significant adaptation. Kelly [28] advocates viewing the output of SSD work as the knowledge itself rather than the software as a particular product. Kelly [28] furthermore thoroughly explains how SSD is not accurately characterized by notions of end user programming or development [36]. Drawing upon Ko et al.’s [30] articulation, Kelly [28] produces a

ten point breakdown of differences, emphasizing that the end user development “viewpoint is far too software product centric and misses the important differences in how scientists work” (p.55).

### Scientific Software Testing Concerns

Testing software is necessary to ensure that it functions in a way that is actually desired by the developer. It has to be verified and validated against some set of expectations. In science, testing software directly impacts analyses and findings by shaping knowledge outputs. Among all of the CS literature on SSD one key concern and point of comparison with software engineering (SE) methodologies is how scientists do or do not go about testing the software that they produce, in particular using or not using systematic automated tools, [21, 27, 45, 56]. SE methodologies have described a variety of approaches to ongoing automated testing as part of software production, maintenance, and evolution work. These approaches include forms such as black & white box, integration, regression, and unit testing among others [19]. Across all of these approaches is a common goal of automating testing to ensure continuous coverage and to abstract the details underlying tests out of a developer’s set of immediate concerns (at least until a test fails), leaving out a discussion of the embodied testing activities software engineers must also undertake.

Within this body of literature it is noted that if scientists do use any of these SE methodologies then they employ unit, regression, and/or integration testing [45]. Sletholt et al. [56] note that testing scientific software is difficult because of emergent, iterative creation of its specifications. This inhibits black or white box testing that would require well-defined specifications or requirements. Kanewala and Bieman [27] suggest there are two issues: 1) testing challenges due to the nature of the software itself, and 2) challenges due to cultural differences between scientists and software engineering practitioners. This latter point ties back to this research community’s larger emphasis that scientific software development is not accomplished in exactly the same manner as other SE work. Heaton and Carver [21] systematically examined the literature on scientific software testing and explain that many issues come down to different understandings regarding ideas of verification and validation—the key underlying notions guiding software testing practices.

For our purposes this SSD scholarship provides a perspective on scientist’s work developing and testing software with which to compare the collaborative work we see in our study. Instead of taking software as an industrial product of scientific work we characterize it as a mutable material that operationalizes ever changing scientific methods. As a result testing practices must be re-examined with a broader lens.

### Constructing and Using Scientific Visualizations

Representation of scientific concepts and knowledge through visualizations is a long-standing focus of Science & Technology Studies (STS) scholarship [8, 34, 35, 40]—the expanse of which is summarized well in [9]. Running throughout STS studies is a theme of investigating how remote, unseeable phenomena are made visible through sensors, software, and localized, contextual practices. This STS research informs our understanding of plots in the Radio group precisely because we are interested in the embodied practices of these scientists creating and using visualizations in their day-to-day work—STS findings enable us to characterize the embodied nature of this group’s work with plots. *This helps us explain how their plot-oriented activities are a way of developing and testing software as an ecologically situated scientific product and not an industrial product to be abstracted from particular places and spaces.*

Scientific visualizations are material products resulting from the customs and practices of particular scientists doing work in a particular embodied environment. Lynch [38] explored how scientists make natural objects visible and therefore analyzable by examining “conventions” used by these researchers in their daily work that “produce features of visual displays which reflect the

disciplinary organization of scientific labour as much as they do the organization of natural objects and relationships.” Lynch and Edgerton [39] build upon this insight when looking at the aesthetic decisions of astronomy digital image processing to foreground how expressing a phenomena requires a “hands-on process” of interpretation to be able to see the physics and convey insights to other researchers. Goodwin [18] similarly emphasizes that practitioners in different disciplines develop professional vision that is “perspectival, lodged within specific social entities, and unevenly allocated” and is not a purely mental process but one embodied through situated practices in a particular relevant setting.

To understand visualizations we must examine conventions, vision, and processes in particular research contexts since particular scientific questions and hypotheses inform the creation of visualizations. Vertesi [60] explains in great depth how scientists working on NASA Mars rover missions make objects captured with remote instruments seeable through a process of “drawing as”. Vertesi emphasizes how scientists transform abstractions into materials with which to do work, materials that are necessarily bound up with the particular instruments and software produced for a project—yet she does not unpack and disentangle software production (just use) in this work. She also notes that “images on the Rover mission are drawn as a representation of a hypothesis or an interaction, to be sure, but they are also crafted in such a way as to generate or otherwise require a shared vision within the team.” Knowing this we must study how scientists draw visualizations to express hypotheses and create compositions as analytical objects for unfolding investigations using particular materials, but we must also be mindful of the need to interrogate and assess how software is produced and tested in doing so.

Finally, Passi and Jackson’s [50] recent work theorizing “data vision”—extending Goodwin [18] above—as the ability to be able to manipulate the world with data and algorithms while simultaneously mastering why, how, and when to use or adjust established methods and tools. They put forth a rule-based rather than rule-bound conceptualization of algorithmic knowledge production, emphasizing that data analysis is a craft where improvisation is necessary to successfully do work.

All together these three veins of insight help us situate the practices we see in our study yet leave a significant gap where the intricate production of software by empirical scientists to produce visualizations and knowledge is left unexamined. Passi and Jackson’s conceptualization of data vision is useful for foregrounding the craft of analysis using visualizations, but by focusing only on algorithmic learning tasks leaves a wide range of inquiry open. We see forms of data vision that are broader and more expansive than the interplay of just algorithms and data—more of a form of software or infrastructure vision—in our research site.

## RESEARCH SITE & METHODS

This paper presents findings from our episodic, longitudinal study of scientific collaboration and software development. The Radio group was one of the groups enrolled in our study and the focus of the first author’s dissertation research [47]. This work was conducted between 2011-2016 using a series of semi-structured interviews, observations of all hands meetings and teleconferences, and the collection of a myriad of artifacts including software code, design documentation, scientific publications, plots, and so on. This corpus was continually analyzed using grounded theory approaches [5, 13] in concert with a relational infrastructural perspective [57] to guide ongoing data collection and inform findings. *All names of the group, individuals, projects, and software are pseudonyms.*

### Research Site

PACM on Human-Computer Interaction, Vol. 1, No. CSCW, Article 85. Publication date: November 2017.

*Pre-print in advance of publication in the ACM digital library.*

Our research site was the Radio group at the University of Washington in Seattle, WA. The Radio group is composed of researchers studying cosmology from the Physics & Astronomy Departments. The group’s Principal Investigator is Magnus, a Professor of Physics. During our study there were three post-doctoral researchers (Brianna, Igor, and Jonah), three graduate students (Abner, Peg, and Nima), and a rotating cast of undergraduate students. Our inquiry focused on the work of Abner, Brianna, Igor, Peg, Nima, and Magnus as they were the individuals most prominently focused on producing the US EoR pipeline (see below). The Radio group were key contributors to a new multi-national radio array telescope project during our study. We refer to this as the Widefield Radio Telescope (WRT) project. We intentionally scoped our inquiry to focus on the Radio group and their software primarily—with limited interaction with US-based WRT colleagues through teleconferences—rather than studying the multi-national project as a whole. This research design was intended to enable us to deeply follow one group’s work as part of a large project to try to understand how their local work was shaped by and shaping the overarching project, rather than study the dynamics of a large project itself as most CSCW infrastructure studies do.

### ***The Widefield Radio Telescope (WRT) Project***

The WRT project is a large multi-national collaboration creating policies, instruments, data, software, and knowledge. Key to our study is understanding that the WRT as a scientific instrument is not a traditional radio telescope with a dish that points at specific areas of the sky. Instead it is designed to capture a wide field of view (e.g., the entire sky above it) at once, rather than focusing in narrowly on very bright objects, using an array of 128 ‘tiles’ of 16 dipole antennas (2,048 total antennas) that are pointed using electronic signal delays. This design affects the data produced and results in the need for entirely new software to advance scientific knowledge.

Another distinguishing element is that the WRT as a project was organized with four distinct scientific goals, each with its own organizing collaboration (only one of which came within the scope of our study). The Radio group was contributing to work studying the Epoch of Reionization (EoR) as part of the WRT’s EoR collaboration. The EoR is a period in the early history of the Universe when stars and galaxies were first lighting up that has only begun to be empirically probed within the last two decades. In current EoR research the goal is to be able to develop a better statistically rigorous measurement of the radio signal capturing this phenomenon. Producing a better statistical measurement today already requires many petabytes of data since the radio signal is exceptionally faint and easily masked with noise.

### ***The Focus of Our Inquiry: The US EoR Pipeline Software***

The WRT project’s Epoch of Reionization collaboration is producing two distinct but interoperable sets of software for doing EoR data analysis—typically called pipelines, a metaphor commonly employed in the sciences regarding software for data analysis [43]. Previously created radio astronomy software is designed to analyze bright, narrow field of view data (e.g., an observation of a star or galaxy) and often includes assumptions that limit the precision of analyses. The WRT in contrast is a wide field of view telescope and the EoR is a faint signal requiring high-precision analyses—the opposite of pre-existing data analysis software.

One of these pipelines is solely the product of the Radio group and the focus of our inquiry. We call it the US EoR pipeline. The second EoR pipeline accomplishes the same type of tasks but is implemented in a different way and was outside the scope of our study. Each pipeline is designed to exchange a variety of data products [47, 48] so that scientific methodological decisions and software functionality can be compared to achieve rigorous data analyses. Data products encapsulate observation data that is transformed through different pieces of software, capturing and portraying different scientific decisions.

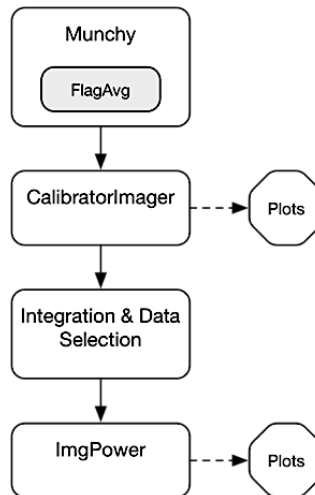


Fig. 1. A simplified diagram outlining the flow of data through the US EoR pipeline from the least analyzed at top to most at bottom. A variety of plots are output by CalibratorImager and ImgPower. Integration and data selection is a point where a human is in the loop, making scientific decisions about how the analysis unfolds.

The US EoR pipeline at its most abstract is three main components (see Figure 1): Munchy, CalibratorImager, and ImgPower. Each of these components is designed to implement algorithms and structure data in a manner that expresses the Radio group’s particular approach to EoR cosmological science that is rooted in rigorous statistical decisions—an approach that does intentionally differ from the second interoperable pipeline. This software was entirely produced by the Radio group during the period of our study. Feedback on its outputs was gathered from other WRT collaborators but these individuals were not responsible for development of the pipeline’s components.

Munchy is a series of scripts, database tables, and an outside data flagging & averaging package (FlagAvg) first organized by Abner, a PhD student in the Radio group. Munchy is designed to manage the reduction of raw telescope data using FlagAvg and to record the decisions informing executions of the overall pipeline (which settings, which pieces of data, etc.). CalibratorImager is a high-precision calibration and imaging pipeline developed by Igor, a post-doctoral researcher in the group. ImgPower is a pipeline for creating a power spectrum (a type of signal measurement) developed by Brianna, another post-doctoral researcher in the group. A fourth key facet, executed in between CalibratorImager and ImgPower, is the selection and integration of individual observations, a moment where a researcher directly shapes the analysis by including or excluding particular pieces of data—a human is intentionally in the loop and must determine how an analysis should be done. Data processed from end to end of the US EoR pipeline will be pulled from the WRT’s data archive by Munchy and processed through FlagAvg then CalibratorImager, and finally by ImgPower. In the course of the Radio group’s work intermediate data products are output then reused as elements of the pipeline software are changed, a linear path through the pipeline is not the norm.

### Data Collection & Analysis

We collected data across three episodes between 2011-2015 that resulted in over 19 hours of in-depth interviews, 30 hours of observations, and hundreds of collected artifacts such as emails, publications, code, etc. Episodically following the work was key to allowing us to see collaborative practices emerge, solidify, and grow in this group’s scientific work. Data analysis was ongoing from 2011 through 2016 to develop our understanding of the Radio group’s work. Our first data collection



episode took place between 2011-2013 when we first interviewed Magnus as a PI, conducted initial observations of all hands meetings of the local group, and conducted a round of hour long semi-structured interviews with group members (Abner, Brianna, Igor, and Peg). This episode produced a baseline understanding of the group’s organization, project work, and science. At this point in time the Radio group was building the initial versions of their US EoR pipeline software while helping bring the Widefield Radio Telescope into operation. The WRT’s “first light” as a full instrument capable of collecting data was only in July 2013. *This results in a key inflection point in not only our inquiry but the work of the Radio group itself where they shifted from instrument development to assessing its operation and doing their scientific analyses through work on software.*

The findings presented here emerge from our second and third episodes of data collection. Episode two took place between January and May 2014 while episode three followed in January and February 2015. During episode two the first author attended weekly all hands meetings, videoconferences of the Radio group and collaborators in the United States, and conducted another round of in-depth semi-structured interview with Abner, Brianna, Igor, and Peg. These interviews were designed to walk through each individual’s work to collect, process, analyze, and archive data by exploring the software they were working on, the state of the telescope, and the plots created. This round of data collection developed our understanding of the dynamics of the group’s ongoing work and helped us narrow our focus around the US EoR pipeline and its components. Plots emerged in this round of data collection as a prominent feature of the work. Finally, episode three was designed to gather reflective data from the group about the work we had seen so that we could triangulate our findings, receive updates about problems faced, and to provide us with an update on the status of their work in general. This included interviews with Abner, Brianna, and Igor for a third time, Magnus a second, and one with Nima who was a new PhD student. A few observations were conducted as well to assess whether our earlier experiences attending meetings were accurate.

Throughout our data collection we employed a grounded theory approach to analyze the data we were gathering as we worked to trace relations among different elements of the work. We open and axially coded the data that were collected and worked to develop and refine themes [5, 13, 44]. Early coding helped characterize the overall landscape of the group’s work, surface scientific terms and pieces of software to further investigate, and reveal the different individuals participating around the world on the WRT project. Over time our coding narrowed to try and tease out how this software was designed and built, the relationship of this software to the telescope and the data it produces, and the different types of plots and their roles in the work. Memos describing emerging concepts were written and shared with our research group to guide our ongoing data collection. Early findings were disseminated through publications as well [47-49]. Germane to this paper is that the collaborative use of plots emerged prominently in our meeting observations—and in our multiple rounds of interviews were continually used by members of the Radio group to explain the science, their software, the telescope, and so on. For this paper we revisited our coding, memos, and literature reviews to distill and frame the findings here.

## FINDINGS

The Radio group’s collaborative use of plots for data analysis and testing their software is a practice designed and enacted to enable qualitative evaluation of complex quantitative phenomena in a methodical and rigorous, yet creative, manner. This use of plots shapes the practices of the individual members of the group producing the pieces of software as well as their use in regular all hands meetings where data analysis and software testing work is accomplished. Plots as complex scientific artifacts fulfill a multitude of roles in the Radio group’s ongoing work in relation to their novel data analysis software. This software is grounded in complex statistical approaches that are fundamental to this realm of cosmological knowledge probing the Epoch of Reionization phenomenon—this software

is implementing and operationalizing abstractions to transform raw telescope data into forms of scientific knowledge. Plots are a material output by this software that is essential for interrogating, testing, and debugging multiple layers of infrastructural elements in this group's work, including: their software, their telescope as an operational instrument, and the state of their scientific knowledge. Plots enable the Radio group to figure out what scientific questions they are asking while in turn evaluating the answers to these questions. Our findings illustrate parts of the complex ecology of work that their software is situated in as these scientists engage in the craft of their science—the work of continually testing software and analyzing data.

### **Plots as a Material Language For Testing & Analysis**

Ongoing collaborative work in the Radio group is centered on practices creating and qualitatively examining plots as a way to interrogate and disentangle the many complex and changing elements of their research. We understand this collaborative work as an embodied form of software integration and regression testing of the US EoR pipeline that produces insight into this novel software as well as shapes scientific knowledge. This work is especially visible through the group's interactions during regular weekly meetings that are plot-oriented. Plots in the Radio group are purposefully designed to reveal different facets of the telescope as a physical instrument, the physics theory, the data collected with the operating telescope, and the particular analysis decisions enacted through the use of the various pieces of software.

The Radio group's US EoR pipeline components *CalibratorImager* and *ImgPower* can each produce a large variety of plots to facilitate the Radio group's collaborative work over time. Each plot produced by an individual in the group using this software conveys a particular scientific question as it represents the state of the software, data, and telescope at a given time. These range from various images of data and telescope functionality output by *CalibratorImager* to purpose-built two-dimensional visualizations of a power spectrum created by *ImgPower* (Figure 2 below). In particular, creating and evaluating 2D power spectrum plots continuously became central to finding issues and bugs in this data analysis software and understanding the overall state of the work. It was essential to the group's work proactively trying to break their software, analyses, and infrastructures so as to repair them.

2D power spectrum plots are vital for testing the US EoR pipeline as a set of software. They are a novel design of Magnus and the group in this scientific field that express a significant amount of scientific theory they have worked on developing—they are designed to express the Radio group's conventions of practice and way of approaching drawing this type of scientific phenomenon. For example, looking at Figure 2 the area of scientific interest to the Radio group is the “EoR Window” represented by the wedge shape above and to the left of the solid black line. Brianna designed 2D power spectrum plots to have “dirty,” “model,” and “res” or residual boxes that visualize respectively: 1) the raw image data captured by the telescope; 2) the idealized, computed model of signal on the sky; and 3) the residual signal remaining when the model is subtracted from the dirty. This design decision is rooted in the design of the data and its structures that she and Igor use in their software.



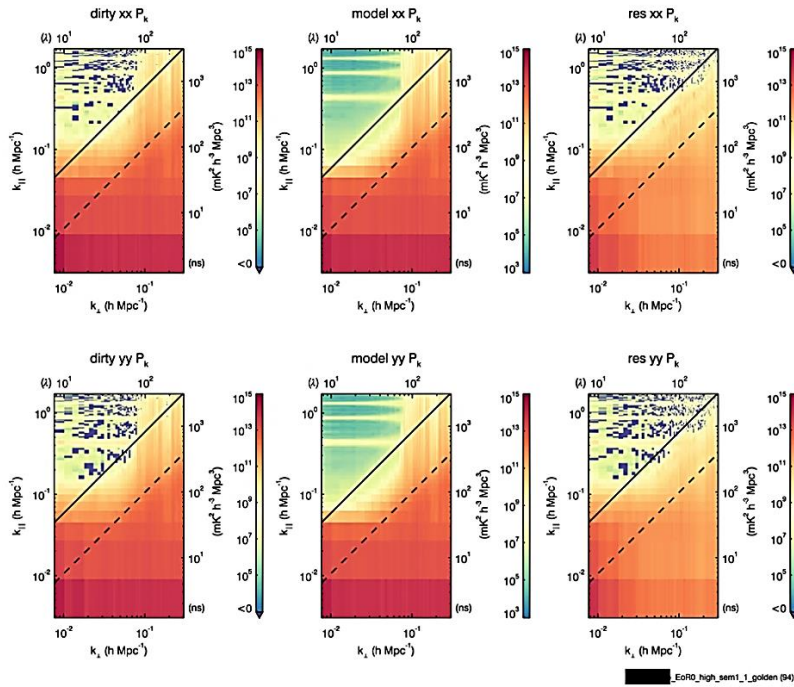


Fig. 2. An example 2D power spectrum plot that expresses the Radio group’s scientific approach to working with the Widefield Radio Telescope’s data. The area of interest scientifically is above and to the left of the solid diagonal line, known as the “EoR Window”. The goal is to drive strong unwanted noise & power signals (the reds & oranges) below this line so that faint, potential EoR signal will appear in the blue-purple to green spectrum. Metadata embedded on the plot helps the group trace a plot to the particular software version & observation data used. Image provided courtesy of the Radio group.

Scientifically the Radio group’s goal is removing unwanted spurious high-power signals (displayed in orange to red colors) and revealing faint, potential EoR signals (the purple-blue-green end of the spectrum) in the EoR Window. This is the underlying reason for their goal of adding together larger and larger quantities of data that we discuss below. To accomplish this the Radio group’s ongoing work was particularly focused on identifying, understanding, and fixing what are known in their vernacular as “systematics”. Systematics are bugs or flaws that arise from the instrument or software, or noise and other effects captured in the data. Identifying and removing systematics is essential to producing a high-precision statistical analysis since systematics drown out the sought after EoR phenomenon. Over time designing new variations of 2D power spectrum plots became a key task for group members as the basic version was no longer able to reveal enough detail for these individuals to be able to easily evaluate their scientific outputs.

Designing and selecting particular plots provides the Radio group with a material way with which to grasp the emergent outputs of the complex mathematical analyses they are implementing in software code, and to make scientific decisions about the data collected and EoR phenomenon being sought. The design and use of different medleys of plots is in part rooted in the scientific approach that Magnus advocates in his group. At multiple points in our data collection he emphasized his belief in a “qualitative” analysis of this complex instrument, data, software, and their current theoretical and empirical understanding to complement the rigorous mathematical operations encoded in the software. When we asked about the role of plots in the group’s work Magnus explained that “Um

*[long pause] plots really are our language. They're our words.*" His measured, reflective response continued, expressing that *"they're [plots] interesting 'cause they're about how we present the data via the software to ourselves, but there's also a very strong science theory side to them in that by choosing a plot you said this is what I think is important."* Using plots as a language is fundamental to how the Radio group draws their sought after natural phenomenon as an analytical object through the software they are building and enables them to construct practices for analyzing data and testing their software. Creating and using plots as a language provides a tangible material central to the ecology of their practices and informs the lived experiences of these scientists when working together as they communicate scientific choices with their software and its operations.

### ***Expressing Scientific Choices Through Plots***

The process of using plots as a language to answer particular scientific questions and/or to facilitate a given software test depends upon these scientists purposefully determining the quantity of data being worked with at any given time. The Widefield Radio Telescope captures data in approximately 2-minute intervals. The US EoR pipeline software is designed to allow researchers to add together different quantities of observation data to facilitate different analyses. Analyses can be conducted on single 2-minute observations all the way through combinations of many dozens or hundreds of hours of data. Different quantities of 2-minute observation data added together directly influences the phenomena visible in plots.

The selection of particular pieces of data to add together is an important point of this group's software testing and data analysis where a human is purposefully in the loop making a scientific decision. Including or excluding a particular 2-minute observation shapes the knowledge produced by changing what is able to be seen in output plots. For some changes to the US EoR pipeline software 2 minutes or 30 minutes of data may be sufficient to evaluate the effect, but over time larger and larger combinations became necessary to continue to identify new bugs or surface issues with the way previously understood problems were addressed. For example, three hours of data were heavily used during our second episode of data collection because fewer and fewer errors were appearing in 2, 30, and 90 minutes of data.

Understanding this we can see that using different quantities of data when running the US EoR pipeline to produce power spectrum plots results in different integration tests of this software pipeline. The plot as a product output at the end of the pipeline's operations (producing a power spectrum plot requires running data from end-to-end, but not necessarily in a linear order depending on past executions and their outputs) materially represents the full operation of the entire set of software (Munchy, CalibratorImager, and ImgPower). Running the same set of data through the software to produce a new power spectrum plot as a particular element of the software is changed enables the Radio group to regression test their entire pipeline. By producing plots using the same data before and after a change to a part of the pipeline they can rigorously evaluate how the change affects their software's operation and the scientific knowledge being produced—a form of regression test of the system to evaluate when the change results in an improvement or not. The Radio group's plots as a language enable them to systematically adapt and expand software engineering practices as part of their qualitative, scientific practices. These practices are continuously visible in the group's regular weekly all hands meetings.

### ***The Embodied Collaborative Use of Plots in Meetings***

Plots serve as a tangible, physical material for testing software and analyzing data that embodies the executions of the US EoR pipeline software and the WRT as an instrument in the Radio group's regular all hands meetings. The Radio group holds weekly all hands meetings for 1-1.5 hours so that current ongoing tasks can be discussed, issues raised, and everyone kept informed about the state of the collective ongoing research.

All hands meetings involve the members of the local group in Seattle (there are separate videoconferences for meetings with distributed collaborators). They are held in a conference room in the Physics and Astronomy Departments buildings near Magnus, Igor, and Brianna’s offices. Meetings begin with members of the group trickling in and grabbing seats while placing piles of papers, and at times laptops, on the table. Different conversations would begin among individuals until either Magnus or the scheduled meeting time arrived. Once the scheduled meeting time arrives Magnus asks for short updates from everyone. Magnus was kind and thoughtful enough to be sure to always ask us for an update about our work as well, very much working to include us as part of the group. Status updates at any given meeting varied from angst over coursework for undergraduate and early-stage graduate students to issues or concerns with software development and data analysis tasks for later-stage graduate students and post-doctoral researchers.

As these short updates end the focus of the meeting shifts to examining pressing software testing and data analysis tasks being undertaken by different individuals. Commonly Magnus would inquire “*who has plots?*” as group members would note whether they brought any with them today. Our field notes capture that across our experiences whomever had plots on a given day would proceed to spread an array of them around one side of the table (Figure 3 is a large scale analysis for Abner’s dissertation). The individual presenting these plots would then explain the issue(s) they were having with their data analysis and the software as well as potential causes they had hypothesized so far.



Fig. 3. An example of a Radio group all hands meeting analyzing 2D power spectrum plots for Abner’s dissertation work. Image provided by the Radio group.

The members of the group would proceed to inspect the different printed copies of the plots, huddling over the table and pondering what was visible in the printouts and how some phenomenon may be coming about. At times someone would circle or mark oddities in a plot or question the group about some aspect that was visible to them personally as a way to try to understand which piece of software, what part of the telescope, or what natural phenomena could be producing the visible effect. Conversations among the group could also result in members drawing on the chalkboard as they tried to explain their thinking. Individuals would frequently stand up and step back from the table a bit while pondering the array of plots spread across the table as the conversation unfolded, trying to better understand the problem at hand.

For example, during one meeting we attended in early 2014 Brianna was sharing a series of 2D power spectrum plots that she had created after making a change to `ImgPower`'s code and running the US EoR pipeline with 3 hours of data. Within her 2D plots an unexpected line was suddenly appearing that not only limited the precision of the analysis but represented some sort of potentially major flaw in the software or telescope that needed to be explained. At this point in time Brianna did not know whether the line was from an instrument systematic captured in the data or a systematic stemming from how a piece of the software was written (looking at [Figure 2](#)'s center column above, three horizontal yellow lines are visible and expected among the solid blue-green area, a fourth unexpected line appeared in this case). This became known as the “fourth line bug” or systematic [47, 48]. This emergent systematic appearing in the 2D power spectrum plots necessitated that the Radio group work to disentangle relationships among their scientific phenomena, instrument, software, and practices through these tests—elements embedded in myriad ways across multiple layers of infrastructural components.

When presenting her new plots to members of the group during this meeting, Brianna and the others proceeded to huddle around the collection arranged on the table while offering ideas as to the potential cause of this fourth line, pointing to different parts of the printed plots and discussing different individual's understandings of the work that was unfolding. The members of the group began to narrow the discussion to ponder how Igor had implemented part of the calibration approach in `CalibratorImager`, as well as pieces of the physical telescope hardware. Throughout this conversation these individuals started devising tests to run using the US EoR pipeline that would help them to try and isolate the source of this fourth line—devising “jack knives” which we discuss further below. Such tests would involve carving up data and features of their software to systematically produce outputs with different variables controlled and held static each time. Doing so would help the group determine where they might adapt their software to effectively remove this fourth line and solve (at least temporarily) this newly identified systematic. Over time, across multiple meetings and conversations among this local group, and with collaborators around the world, the Radio group determined that signal was being reflected in one type of the telescope's cables. The reflected signal was what was appearing as the fourth line and to rectify this systematic Igor had to adapt the calibration approach implemented in `CalibratorImager` to account for this physical flaw—the software had to be adjusted for the physical reality of their instrument and its flaws, commonsense for an empiricist who builds instruments yet not something a theoretician thinks about.

From this example, and the overarching practice of the Radio group using plots during meetings, we see that the plot-oriented work embodied in this group's conversations huddled around a menagerie of printed plots enables these scientists to integration and regression test their software as they analyze their data. In the case of Brianna's fourth line bug this all took place as they analyzed a 3 hour set of data, interrogating multiple layers of infrastructural components in their work (the physical instrument, scientific knowledge, data, and software). This type of process unfolds continually in the group's work as different systematics or bugs arise and new scientific questions come to the forefront. Plots as a material visualize myriad scientific decisions and enable the group to embody and express their approach to this complex work. They are engaged in a complex craft [17] where their software

and plots are part of an ecology of work that is adjusted and shaped as their knowledge expands and changes, embedded in different infrastructural components.

### **The Evolving Craft to This Plot-Driven Data Analysis**

The Radio group’s collaborative scientific work unfolds as they use their variety of plots to test their software and analyze data. Through their ongoing discussions huddled around conference tables littered with printed plots we see that the Radio group’s plot-oriented collaborative data analysis and software testing practices results in different individuals being taught about the scientific approach being implemented and collectively everyone questioning each other about different elements. This practice is by no means static, they are shaping and adjusting their materials to continuously create doable problems [17] as they interrogate different infrastructural components. In this work we can see that these scientists are constructing not just a form of “data vision” [50] but rather a software infrastructure vision as they disentangle myriad embedded relations across infrastructures when identifying and tackling systematics. These scientists intentionally adjusted their software, plots, and practices to facilitate different ways of methodically solving problems.

### ***Teasing Apart Systematics and Infrastructural Components Through Jackknife Testing***

The Radio group’s plots are designed to support many aspects of their work, and as Magnus noted for figuring out the scientific questions they are asking. A key part of the craft of their work is that the design of these plots (as well as the software that produces them) was evolved to help facilitate a physics practice for isolating variables and testing their effect called “jackknifing” or jackknife testing (the terms are used interchangeably in the group’s work, and encompass practices that complicate the basic statistical resampling notion). Jackknifing as a plot-oriented practice in the Radio group results in these scientists using their deep embodied understanding of their materials and scientific knowledge to test and evaluate different components. Understanding jackknifing in this work not only further illustrates the use of plots but also demonstrates how this scientific software development work is a complex unfolding dialogue between these scientists as expert users and developers of the US EoR pipeline software.

Long-term the members of the Radio group are focused on continually carving up their data and aspects of their software in different ways to examine different hypotheses and to question previously held assumptions as they try to identify and understand different systematics. Jackknifing is a concept that supports these scientific analysis activities—whether of the actual data produced by the telescope, the US EoR pipeline, or any other software employed by this multi-national collaboration. We first heard the term jackknifing from Brianna when talking about changes she was making to ImgPower to address a systematic that had appeared as well as prepare her software for future testing and analysis work. Asked to explain, Brianna conveyed that jackknifing is simply “*carving up your data in different ways and seeing if you see different things in the different pieces.*” Similarly, Abner explained that “*what I think of as a jackknife is choosing some parameter to test against, and so that might be by day.*” In short, jackknifing is the term these scientists give to part of their efforts to rigorously and methodically produce scientific knowledge as a dialogue between what they currently think they know and what they see emerging in their materials.

The goal of the seemingly simple concept of jackknifing is to isolate some phenomenon, but doing this requires careful thought regarding what is expected in the data being jackknifed (or the functionality of an aspect of the software being tested, if that is the focus), and what is being visualized and drawn through their practices. For example, the Widefield Radio Telescope is sensitive to faint FM radio signals bouncing off of the moon back to Earth, even though the project does not want such signal interfering in its data. If a set of observations has data with and without the moon then the group can carve it up (e.g., isolate individual 2 minute observations) based on that and



compare plots to see if there is such a signal that they need to figure out how to subtract from their analysis.

Similarly, the Radio group can make a change to one aspect of the US EoR pipeline software then execute the new and old versions with the exact same piece of data and settings then compare outputs—a form of jackknifing that is in part accomplishing a regression test. This idea of jackknifing software became a formal practice as Igor in particular faced challenges tracing his changes in CalibratorImager to effects visible in power spectrum plots. To create and carry out many types of jackknives required adapting ImgPower to facilitate the evolving practices of this group.

### *Adapting ImgPower to Create New Variations of Plots*

Constructing and evaluating different jackknives—as part of how they integration and regression test the US EoR pipeline—requires care and craft such that these scientists end up shaping and adjusting their software, plots, and analysis decisions over time as they make different choices and evaluate their effect. Brianna designed and adapted ImgPower over time in particular to facilitate creating and running different jackknife tests in this work. From the Radio group’s perspective their data analysis is on the whole better scientifically if a power spectrum is improved—e.g., less noise and unwanted signal is making it into a plot so that the sought after EoR phenomenon can emerge. This scientific orientation motivated some of Brianna’s changes to ImgPower where she created new types of power spectrum plots so that new jackknife tests could be created and sustained—in effect enabling new ways of software integration testing to facilitate longer-term regression testing of the US EoR pipeline.

By early 2015, Brianna and the Radio group were finding that it could be difficult to qualitatively see changes in 2D power spectrum plots such as the one presented in [Figure 2](#) earlier. This necessitated that she and the group construct new types of power spectrum plots. The EoR signal being sought is exceptionally faint. As noisy data or other systematics are removed from the analysis changes to the software (or differences between two sets of data) can produce very, very subtle effects in these plots.

To help suss out these subtle effects Brianna designed new variations of this base plot. She called these the “differencing” and “ratio differencing” plots. Both of these types of power spectrum plots are purposefully designed to reveal subtle effects to support and facilitate the Radio group’s qualitative testing and analysis practices—to aid these researchers as they huddle over conference tables and stare at their computer screens. Brianna’s change to her software to produce this new design of plot extends the group’s existing practices to carve up data and their knowledge in a new way so that different phenomena can be surfaced and discussed, exposing or hiding different dimensions of their evolving infrastructures. This is in part predicated upon the way she and Igor designed their internal data structures since they purposefully carve data up in key ways to ensure different statistical approaches can be accomplished properly.

“And so we have really realized that the measure of whether something makes things better is whether it makes these plots better. And often these [2D power spectrum] plots, the ones that aren't differenced, cover ten orders of magnitude, so it can be hard to see subtle changes because it's just a subtle change in a color bar that covers seven, or eight, or ten orders of magnitude. So by differencing them, we can see highlights of what those things are.”

(Brianna, post-doctoral researcher)

[Figure 4](#) below is an example of one such differencing plot. In this plot two sets of observation data have been processed through CalibratorImager and ImgPower. One set of data was subtracted from the other and the difference in signal that remains between them is visualized in red and blue (the meaning behind the colors varies depending on the particular context) such that subtle phenomena are able to visually pop out. This subtraction and use of only two color spectra (red and



blue) is designed to help the Radio group try to make subtle changes more easily visible when huddled around conference tables or computers when they are engaged in their practices of testing the US EoR pipeline software and analyzing data.

The change to ImgPower and the group’s practices to create and use differencing plots demonstrates one simple but important way they evolved their practices in their efforts to surface effects from across different infrastructures in their work. This software development and testing work is inextricably bound up with their scientific approaches to data analysis, and really even instrument design. These two newer forms of 2D power spectrum draw upon particular design decisions Brianna and Igor made for storing data internally in their software so that Brianna can foreground differences between pieces of observation data or versions of software more clearly through the construction of these plots. ImgPower as a piece of data analysis software was adapted over time with the design of the Radio group’s data and the scientific theory they are drawing upon to facilitate and support changes in their scientific methods. Perpetually adding to their milieu of plot-driven practices enables the members of the Radio group to expand their scientific knowledge, so long as they can compare the effects of changes to their earlier work. At times this can necessitate creating a new, or adapting an existing, practice as they work to develop a shared ability to interrogate their pieces of software, their telescope and the effects it inscribes in data, their scientific knowledge, and even the effects of their collaborative organizational structure.

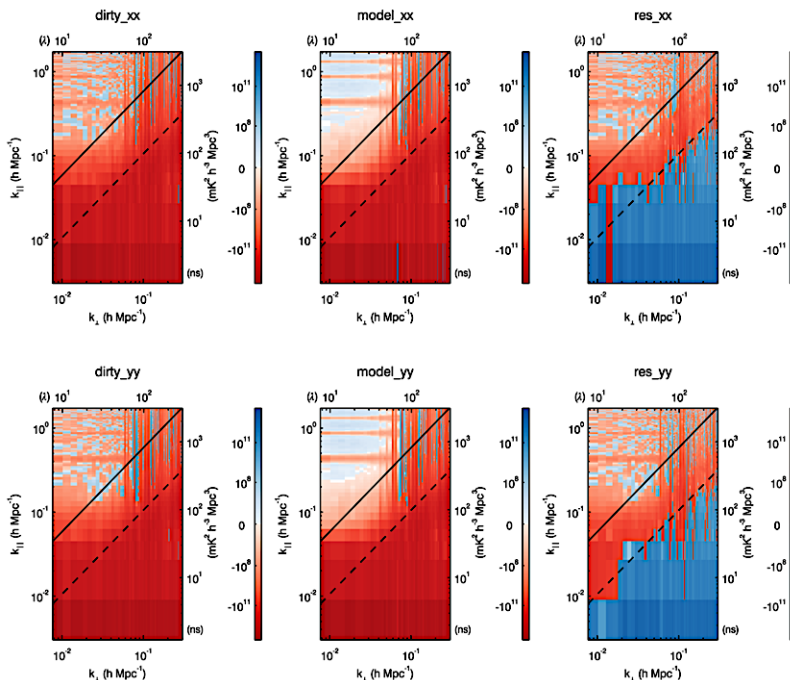


Fig. 4. An example "differencing" 2D power spectrum plot created by Brianna where one piece of data is subtracted from another and visualized. Red or blue can be either good or bad depending on the plot and the jackknife test it is designed for. Image provided courtesy the Radio group.

The Radio group’s 2D power spectrum plots visualize the different pieces of software in the US EoR pipeline (which is an infrastructure for the Radio group’s work along with the various practices) while also conveying aspects to the telescope as an operational instrument through the data it

produces (the telescope is another infrastructure in and of itself). The plots also fundamentally disclose scientific thoughts and beliefs through their construction and express the conventions of disciplinary organization to which they belong [38] and the group’s unfolding aesthetic decisions [39]. Through their work they are enacting and evolving a software infrastructure vision.

## DISCUSSION

A significant amount of CSCW research has focused on studying the work of building infrastructures to support scientific research. Our investigation revealed how scientific experts approach creating complex software and knowledge products; the Radio group’s efforts to test their software and analyze data demonstrates how this group continually interrogates multiple layered infrastructures.

### Enacting an Embodied “Infrastructure Vision” With Humans in the Loop

What we have seen through this group’s work are examples of the interrogation and evaluation of the many embedded components of infrastructures on an ongoing basis because they are expert producers and users of their software. This software is a material that stands between these scientist’s human understanding and machine processing to paraphrase Dourish [10]. The Radio group’s plot-driven software testing and data analysis practices are more than a form of “data vision” [50], they extend beyond learning to see algorithmic effects to representing the larger milieu of sets of software and instrument infrastructures. Rather than making elements of their work invisible, plots purposefully expose different phenomena in the way that they are drawn as analytical objects [60] depending on the decisions individuals make in their creation. Plots are a way to make elements of multiple infrastructures front and center in one place, they are a form of ‘scalar device’ [51] in this work. What we see in the Radio group’s ongoing efforts to identify, evaluate, and address different systematics is a way of trying to find potential breakdowns and repair them—almost but not quite reaching a point of what Sims [55] calls “anticipatory repair”. This work is the knowledge creation process that not only impacts their work today but also their future experimental and theoretical work.

These scientists do not fully know what to expect from their software and data until they begin working with its outputs using their variety of plots as materials to tangibly express different phenomena. To test their software requires more than understanding the expected behavior of the software, it requires deep scientific knowledge and understanding the vagaries of this instrument. The valid operation of the software may furthermore evolve as the quantity of data changes or assumptions coded into some algorithm shift from being valid to invalid as their scientific knowledge increases. Instead of focusing on solely implementing abstracted automated testing, the continual collaborative effort we witnessed in the Radio group is rooted in a notion from the craft of this physics research that the group refers to as “jackknifing” so as to be able to make methodical claims about their instrument, data, and especially software. They very much want to keep a human in the loop of this work (even as other facets of their work employ automated systems for operations and testing).

We can see from the Radio group’s work using plots that a lack of automation when testing their software is not necessarily a bad thing. Having a human in the loop at multiple different points of their qualitative analysis work is essential to asking and answering research questions. During their all hands meetings they are not just analyzing data and testing software collaboratively. They are teaching each other about the software, the scientific theory, and the instrument (among many things). The Radio group is ensuring effective analyses by establishing and solidifying “*mastery of the ways that worlds and tools are put together, and which worlds and tools are so combined*” as Passi and Jackson [50] describe such work. They are collectively figuring out and constructing the rules necessary to do their work using the many complex materials they are producing—the essential idea

underlying Goodwin’s [18] conceptualization of professional vision yet complicated significantly by the need to understand and be able to trace out elements of multiple layers of infrastructural components in this complex work. To think that the Radio group could or should solely or primarily automate all of their testing would be to miss the point of what the plot-oriented software testing and their infrastructural vision is to their scientific practice, and the deep embodied understanding they must collectively develop in their work.

The qualitative testing and evaluation of the US EoR pipeline software’s functionality over time is essential for these cosmologists to do their science. In addition to seeing plots as the group’s language, Magnus expressed his belief that these scientists need to be able to “understand the instrument in their bones” [47]. We would extend that idea to state that they need such an embodied feeling not only of their instrument but also the US EoR pipeline as an elaborate, scientific-theory laden set of software. Testing such software has to be a creative act in this environment. It requires remembering and grasping deeply embedded elements of multiple infrastructures as unexpected, weird issues emerge. This is a grasp that cannot necessarily be concretely captured and operationalized in an automated testing tool (although such a tool would certainly complement and reveal different facets to this infrastructure vision). They have to be able to collectively and deeply “feel” the work in order to do the science—similar to Lingel and Regan’s [37] point that software engineering work itself can be both very rational yet “deeply personal and embodied” as well. Our findings in this paper, like their study of software engineers, demonstrate that it is worthwhile to re-examine how we perceive not only software development work in different contexts but also software as an object itself.

Testing data analysis software in this context is the process of evaluating this material’s design and operation using telescope observation data to simultaneously assess multiple facets to their infrastructures, including: 1) how the designed software actually operates in concert with other materials, 2) how a new telescope is functioning as a scientific instrument, and 3) current conceptualizations of the scientific knowledge underlying these materials and the work overall. This demonstrates a different look at aspects of the work of producing and sustaining research infrastructures—foregrounding a particular material and the relations it is composed of rather than focusing on primarily the organizational concerns or the movement of data as prior CSCW studies do.

This paper’s reading of software work offers contributions to move CSCW’s investigations of software development in science forward. We provided a novel reading of the Radio group’s collaborative practices for testing their scientific software and doing data analysis using visualizations in juxtaposition with Computer Science investigations of scientific software development and testing and Science & Technology Studies characterizations of visualizations. Our investigation results in a unique analytical interpretation of scientists as producers and users of software, not as software engineers or end user developers but as skilled practitioners enacting scientific methods through software. Our findings show how artificially trying to delineate what is “software development and testing” and what is “data analysis” is counterproductive due to the intertwined embodied nature of these activities—in our field site they are all the work of “doing” science.

## CONCLUSION

Dourish [10] posits that the expressiveness of software’s code matters because “it is in how that expressiveness comes to stand between human understanding on one side and machine processing on the other, and how the temporalities, spatialities, and specificities of its expressiveness play a role in how settings of human affairs are organized in parallel with digital ones” (p.57). Continuing such work into new forms of scientific software is essential to advancing our understandings of the ways research infrastructures are enacted and adapted in the course of changing research environments as

well as complicating and reformulating traditional tropes of scientists as users and developers of software—let alone actually understanding how scientific knowledge is produced today.

Drawing upon software studies perspectives [1, 29, 41], our work here examined software as a material situated in the particular ecologies of a scientific group’s work that captures how it lives and ages [6] as their knowledge changes. We purposefully investigated software rather than just code or algorithms because they are only some of the elements of software that matter. Ensmenger [14] emphasizes that software “exists simultaneously as an idea, language, technology, and practice.” Drawing upon Kitchin and Dodge [29] we took software as “a complex, multifaceted, mutable set of relations created through diverse sets of discursive, economic, and material practices”, not just textual code to be run on commodity hardware. This is a form of engagement only beginning to gain traction among CSCW scholars [6, 7, 10, 11].

If we are to better develop nuanced sociotechnical understandings of the many ways that software matters in scientific research we must shift our investigations of scientific software and research infrastructures beyond conceptualizations of tools to offer new readings of the embodied tactics of scientists as highly skilled producers and users of software as a malleable material for knowledge production. If domain scientists are to engage with computer scientists or software engineers, it should be helpful to compare and contrast software development methods and the methods of the particular scientific discipline in order to identify challenges and opportunities. Making explicit differing assumptions and practices will be useful and often necessary when determining how to successfully support changing practices and modes of scientific inquiry through different design interventions (whether educational, technical, or some other form).

Indeed some of what this paper calls for is the application of longstanding CSCW approaches to the study of collaborative artifacts and social situations, such as paper strips for air traffic control or coordination of the London Underground [20, 25], to the study of software artifacts for science. CSCW’s scholarship has undertaken some such studies [23, 24, 26] but now can and should play a leadership role in our own and related communities (e.g. eScience, scientific software infrastructure, STS, software engineering and computer science, engineering design, etc.) that are currently (or should be) grappling with understanding and supporting science practice with and through software. CSCW has an opportunity to take a leadership role in supporting scientific computing by demonstrating how to theoretically and methodologically scope software as an object of analytic inquiry that is comprised of interrelated practices and other artifacts, such as plots, rather than as isolated code or algorithms.

## ACKNOWLEDGMENTS

This article is based on work supported by National Science Foundation grants [IIS-0954088](#) and [ACI-1302272](#) and the University of Washington. We thank our anonymous reviewers as well as Dave Randall, Megan Finn, and Daniela Rosner at the University of Washington for their feedback. Finally, we are grateful to the members of the Radio group for their participation in our study and their feedback on our findings.

## REFERENCES

- [1] David M. Berry. 2011. *The philosophy of software: code and mediation in the digital age*. Palgrave Macmillan, New York, NY.
- [2] Matthew J. Bietz, Drew Paine and Charlotte P. Lee. 2013. The work of developing cyberinfrastructure middleware projects *Proceedings of the 2013 conference on Computer supported cooperative work*, ACM, San Antonio, Texas, USA, 1527-1538.
- [3] Jeremy P. Birnholtz and Matthew J. Bietz. 2003. Data at Work: Supporting Sharing in Science and Engineering. in *GROUP*, Sanibel Island, Florida, USA.

“Who Has Plots?”: Contextualizing Scientific Software, Practice, and Visualizations  
85:19

- [4] Jeffrey C. Carver. 2012. Software Engineering for Computational Science and Engineering. *Computing in Science & Engineering*, 14 (2), 8-11. 10.1109/MCSE.2012.31
- [5] Kathy Charmaz. 2014. Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis. Sage.
- [6] Marisa Cohn. 2016. Convivial Decay: Entangled Lifetimes in a Geriatric Infrastructure *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, ACM, San Francisco, California, USA, 1509-1521.
- [7] Marisa Cohn. 2017. "Lifetime Issues": Temporal Relations of Design and Maintenance. *continent.*, 6 (1), 4-12.
- [8] Cateljine Coopmans, Janet Vertesi, Michael E Lynch and Steve Woolgar (ed.), 2014. (eds.). *Representation in scientific practice revisited*. MIT Press.
- [9] Cateljine Coopmans, Janet Vertesi, Michael Lynch and Steve Woolgar. 2014. Introduction: Representation in Scientific Practice Revisited. Coopmans, C., Vertesi, J., Lynch, M. and Woolgar, S. eds. *Representation in Scientific Practice Revisited*, MIT Press, 1-12.
- [10] Paul Dourish. 2017. *The Stuff of Bits*. The MIT Press, Cambridge, MA.
- [11] Paul Dourish and Melissa Mazmanian. 2013. Media as Material: Information Representations as Material Foundations for Organizational Practice. in *How matter matters: objects, artifacts, and materiality in organization studies*, 92-118.
- [12] Paul N. Edwards, Steven J. Jackson, Melissa K. Chalmers, Geoffrey C. Bowker, Christine L. Borgman, David Ribes, Matt Burton and Scout Calvert. 2013. Knowledge Infrastructures: Intellectual Frameworks and Research Challenges, Ann Arbor.
- [13] Robert M. Emerson, Rachel I. Fretz and Linda L. Shaw. 1995. *Writing Ethnographic Fieldnotes*. The University of Chicago Press, Chicago, IL.
- [14] N.L. Ensmenger. 2010. The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise. MIT Press, Cambridge, MA.
- [15] Ixchel Faniel and Trond Jacobsen. 2010. Reusing Scientific Data: How Earthquake Engineering Researchers Assess the Reusability of Colleagues' Data. *Computer Supported Cooperative Work (CSCW)*, 19 (3), 355-375. 10.1007/s10606-010-9117-8
- [16] Thomas Finholt. 2002. Collaboratories. in Cronin, B. ed. *Annual Review of Information Science and Technology*, American Society for Information Science, Washington, D.C., 73-107.
- [17] Joan H. Fujimura. 1996. *Crafting science: A sociohistory of the quest for the genetics of cancer*. Harvard University Press.
- [18] Charles Goodwin. 1994. Professional vision. *American anthropologist*, 96 (3), 606-633.
- [19] P. Grubb and A.A. Takang. 2003. *Software maintenance: concepts and practice*.
- [20] Christian Heath and Paul Luff. 1992. Collaboration and Control: Crisis Management and Multimedia Technology in London Underground Line Control Rooms. *Computer Supported Cooperative Work (CSCW)*, 1 (1), 69-94. 10.1007/bf00752451
- [21] Dustin Heaton and Jeffrey C. Carver. 2015. Claims about the use of software engineering practices in science: A systematic literature review. *Information and Software Technology*, 67, 207-219. <http://dx.doi.org/10.1016/j.infsof.2015.07.011>
- [22] James Howison and James D. Herbsleb. 2013. Incentives and integration in scientific software production *Proceedings of the 2013 conference on Computer supported cooperative work*, ACM, San Antonio, Texas, USA, 459-470.
- [23] James Howison and James D. Herbsleb. 2011. Scientific Software Production: Incentives and Collaboration. in *CSCW*, Hangzhou, China, ACM, 513-522. 10.1145/1958824.1958904
- [24] Xing Huang, Xianghua Ding, Charlotte P. Lee, Tun Lu and Ning Gu. 2013. Meanings and boundaries of scientific software sharing *Proceedings of the 2013 conference on Computer supported cooperative work*, ACM, San Antonio, Texas, USA, 423-434.
- [25] John A. Hughes, Dave Randall and Dan Shapiro. 1992. From ethnographic record to system design. *Computer Supported Cooperative Work (CSCW)*, 1 (3), 123-141. 10.1007/bf00752435
- [26] Marina Jirotko, Charlotte P Lee and Gary M Olson. 2013. Supporting Scientific Collaboration: Methods, Tools and Concepts. *Computer Supported Cooperative Work (CSCW)*, 22 (4-6), 667-715. 10.1007/s10606-012-9184-0
- [27] Upulee Kanewala and James M. Bieman. 2014. Testing scientific software: A systematic literature review. *Information and Software Technology*, 56 (10), 1219-1232. <http://dx.doi.org/10.1016/j.infsof.2014.05.006>
- [28] Diane Kelly. 2015. Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software. *Journal of Systems and Software*, 109, 50-61. <http://dx.doi.org/10.1016/j.jss.2015.07.027>
- [29] Rob Kitchin and Martin Dodge. 2011. *Code/Space: Software and Everyday Life*. MIT Press, Cambridge, MA.
- [30] Andrew J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman and B. Myers. 2011. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43 (3), 21.
- [31] Robert Kraut, Carmen Egidio and Jolene Galegher. 1988. Patterns of contact and communication in scientific research collaboration *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, ACM, Portland, Oregon, USA, 1-12.
- [32] Robert Kraut, Jolene Galegher and Carmen Egidio. 1986. Relationships and tasks in scientific research collaborations *Proceedings of the 1986 ACM conference on Computer-supported cooperative work*, ACM, Austin, Texas, 229-245.
- [33] Kateryna Kuksenok, Cecilia Aragon, James Fogarty, Charlotte P. Lee and Gina Neff. 2017. Deliberate Individual Change Framework for Understanding Programming Practices in four Oceanography Groups. *Computer Supported Cooperative Work (CSCW)*. 10.1007/s10606-017-9285-x

- [34] Bruno Latour. 1986. Visualization and Cognition: Thinking with Eyes and Hands. *Knowledge and Society: Studies in the Sociology of Culture Past and Present*, 6. 1-40.
- [35] Bruno Latour and Steve Woolgar. 1986. *Laboratory life: The construction of scientific facts*. Princeton University Press, Princeton, NJ.
- [36] Henry Lieberman, Fabio Paternò, Markus Klann and Volker Wulf. 2006. End-User Development: An Emerging Paradigm. in Lieberman, H., Paternò, F. and Wulf, V. eds., Springer Netherlands, 1-8.
- [37] Jessica Lingel and Tim Regan. 2014. "it's in your spinal cord, it's in your fingertips": practices of tools and craft in building software *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, ACM, Baltimore, Maryland, USA, 295-304.
- [38] Michael Lynch. 1985. Discipline and the Material Form of Images: An Analysis of Scientific Visibility. *Social Studies of Science*, 15 (1). 37-66.
- [39] Michael Lynch and Samuel Y Edgerton. 1988. Aesthetics and digital image processing: Representational craft in contemporary astronomy. *G. Fyfe & J. Law (Eds.), Picturing power: Visual depiction and social relations*. 184-220.
- [40] Michael Lynch and Steve Woolgar (ed.), 1990.^(eds.). *Representation in scientific practice*. MIT press.
- [41] Adrian Mackenzie. 2006. *Cutting code: Software and sociality*. Peter Lang Publishing.
- [42] Ben Marwick. 2015. How computers broke science — and what we can do to fix it. Retrieved January 20, 2016 from <http://theconversation.com/how-computers-broke-science-and-what-we-can-do-to-fix-it-49938>.
- [43] W. Patrick McCray. 2004. *Giant Telescopes: Astronomical Ambition and the Promise of Technology*. Harvard University Press, Cambridge, MA.
- [44] M.B. Miles and A.M. Huberman. 1994. *Qualitative data analysis: An expanded sourcebook*. Sage Publications, Incorporated.
- [45] Aziz Nanthamornphong and JeffreyC Carver. 2015. Test-Driven Development in scientific software: a survey. *Software Quality Journal*. 1-30. 10.1007/s11219-015-9292-4
- [46] Gary M. Olson and Judith S. Olson. 2000. Distance matters. *Hum.-Comput. Interact.*, 15 (2). 139-178. 10.1207/s15327051hci1523\_4
- [47] Drew Paine. 2016. Software and Space: Investigating How a Cosmology Research Group Enacts Infrastructure by Producing Software *Human Centered Design and Engineering*, University of Washington.
- [48] Drew Paine and Charlotte P. Lee. 2014. Producing Data, Producing Software: Developing a Radio Astronomy Research Infrastructure. in *IEEE 10th International Conference on e-Science (e-Science 2014)*, Guarujá, SP, Brazil, 231-238. 10.1109/eScience.2014.41
- [49] Drew Paine, Erin Sy, Ron Piell and Charlotte P. Lee. 2015. Examining Data Processing Work as Part of the Scientific Data Lifecycle: Comparing Practices Across Four Scientific Research Groups. in *iConference 2015*, Newport Beach, CA, USA. <http://hdl.handle.net/2142/73644>
- [50] Samir Passi and Steven Jackson. 2017. Data Vision: Learning to See Through Algorithmic Abstraction *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ACM, Portland, Oregon, USA, 2436-2447.
- [51] David Ribes. 2014. Ethnography of scaling, or, how to a fit a national research infrastructure in the room *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, ACM, Baltimore, Maryland, USA, 158-170.
- [52] David Ribes and Charlotte Lee. 2010. Sociotechnical Studies of Cyberinfrastructure and e-Research: Current Themes and Future Trajectories. *Computer Supported Cooperative Work (CSCW)*, 19 (3). 231-244. 10.1007/s10606-010-9120-0
- [53] Betsy Rolland and Charlotte P. Lee. 2013. Beyond trust and reliability: reusing data in collaborative cancer epidemiology research *Proceedings of the 2013 conference on Computer supported cooperative work*, ACM, San Antonio, Texas, USA, 435-444.
- [54] Judith Segal. 2009. Software Development Cultures and Cooperation Problems: A Field Study of the Early Stages of Development of Software for a Scientific Community. *Computer Supported Cooperative Work (CSCW)*, 18 (5). 581-606. 10.1007/s10606-009-9096-9
- [55] Benjamin Sims. 2017. Making Technological Timelines: Anticipatory Repair and Testing in High Performance Scientific Computing *continent.*, 81-84.
- [56] Magnus Thorstein Sletholt, Jo Hannay, Dietmar Pfahl, Hans Christian Benestad and Hans Petter Langtangen. 2011. A literature review of agile practices and their effects in scientific software development *Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering*, ACM, Waikiki, Honolulu, HI, USA, 1-9.
- [57] Susan Leigh Star and Karen Ruhleder. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research*, 7 (1). 24.
- [58] Susan Leigh Star and Anselm Strauss. 1999. Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work. *Computer Supported Cooperative Work (CSCW)*, 8. 9-30.
- [59] Victoria Stodden and Sheila Miguez. 2014. Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. *Journal of Open Research Software*, 2 (1). e21. <http://doi.org/10.5334/jors.ay>
- [60] Janet Vertesi. 2015. *Seeing like a Rover: how robots, teams, and images craft knowledge of Mars*. University of Chicago Press, Chicago, IL.



“Who Has Plots?”: Contextualizing Scientific Software, Practice, and Visualizations  
85:21

- [61] Greg Wilson. 2006. Where's the Real Bottleneck in Scientific Computing? Scientists would do well to pick up some tools widely used in the software industry. *Am Sci.* v94 i1.
- [62] Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White and Paul Wilson. 2014. Best Practices for Scientific Computing. *PLoS Biology*, 12 (1). e1001745. 10.1371/journal.pbio.1001745

Received April 2017; revised July 2017; accepted November 2017