PROJECT SUMMARY

Overview:

The open-source glue package allows scientists to explore relationships within and across related datasets, by making it easy for them to make multi-dimensional linked visualizations of datasets, select subsets of data interactively or programmatically in 1, 2, or 3 dimensions, and see those selections propagate live across all open visualizations of the data (e.g. graphs, maps, diagnostics charts). A unique feature of glue is that datasets from different sources can be linked to each other, using user-defined mathematical relationships between sets of data components, which makes it possible to carry out selections across datasets. Glue, written in Python, is designed from the ground-up for multidisciplinary work, and it is currently helping researchers make discoveries in geoscience, genomics, astronomy, and medicine. It is also giving insights into data from outside academia, including open data provided by governments and cities.

To become sustainable in the long term, glue development needs to become a community-driven effort. Through tutorial and developer workshops, coding sprints, and strategic collaborations with researchers in several disciplines and experienced open source developers, the glue team will help user communities extend glue by developing new functionality useful within particular fields of research. The team will help users contribute the most widely-needed functionality back to glue, and will recruit active contributors to participate in core glue development. As the community grows, glue development will be guided to focus on several major features useful to the broad research community, including: support for very large datasets, support for running glue fully in the browser (inside Jupyter notebooks and Jupyter Lab), and improved interoperability with third-party tools.

Intellectual Merit:

Having selections propagate between multiple views of a dataset is referred to as "brushing and linking" or "linked-views." While this paradigm has been used well outside of science, it is surprisingly under-used by scientists. Commercial linked-view tools can be brought to bear on scientific data when it is all within one data file, in tabular form. But, when investigations rely on high-dimensional images (e.g. in astronomy or medicine), no extant tool can be used - as none handles linked-view selection in 2D images and 3D volumes. Furthermore, scientists' data is often spread across multiple files, and being able to visualize and interact with multiple datasets simultaneously is crucial for scientists. Cognitive and visualization research has long shown that humans are especially adept at seeing change, which is why real-time data exploration using linked views of high-dimensional data offers such an effective way to look for trends, outliers and salient subsets, even in very large data sets. Glue is, at present, the only software package that allows for real-time exploration of high-dimensional data spread across multiple files, and as such offers unparalleled potential for insight.

Broader Impacts:

The range of glue's use in science extends from exploring astronomical observations to analyzing radiological images of the brain to exploring patterns in multi-wavelength satellite images. But, as glue's capabilities expand, so will its potential for impact outside of scientific research. Glue is easy enough to use that: it is a good tool for teaching the fundamentals of data exploration; and it is at the right level to offer effective exploration of online, open, public data. In particular, the proposed work includes enhancing glue's capacity to produce dynamic web pages that will make it possible for learners and members of the general public to share their visual explorations of interesting datasets. Bringing glue, the first free linked-view tool for exploring high-dimensional data, to a broad audience will have long-lasting impact on the way scientists, learners, and the public think about exploring data, and it will serve as a seminal model when high-dimensional linked-view data tools are developed in the future.

TABLE OF CONTENTS

For font size and page formatting specifications, see PAPPG section II.B.2.

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

TABLE OF CONTENTS

For font size and page formatting specifications, see PAPPG section II.B.2.

	Total No. of Pages	Page No.* (Optional)*
Cover Sheet for Proposal to the National Science Foundation		
Project Summary (not to exceed 1 page)		
Table of Contents	1	
Project Description (Including Results from Prior NSF Support) (not to exceed 15 pages) **(Exceed only if allowed by a specific program announcement/solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)**	0	
References Cited		
Biographical Sketches (Not to exceed 2 pages each)	2	
Budget (Plus up to 3 pages of budget justification)	5	
Current and Pending Support	1	
Facilities, Equipment and Other Resources	1	
Special Information/Supplementary Documents (Data Management Plan, Mentoring Plan and Other Supplementary Documents)	3	
Appendix (List below.) **(Include only if allowed by a specific program announcement/ solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)**		
Appendix Items:		

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

Project Description

1. Introduction

Rapid increases in the volume and variety of data available to scientists offer unprecedented opportunities for insight along with often unparalleled challenges for interactive data exploration and analysis. Machine-learning algorithms, while extremely powerful, are only part of the solution, and a paradigm shift is needed in the way scientists **interactively explore** the vast amounts of high-dimensional and heterogeneous datasets being produced.

We have developed a new open-source software package named ***glue*** that allows scientists to explore relationships within and across related datasets, making it easy for them to make **multi-dimensional linked visualizations of datasets**, select subsets of data interactively or programmatically in 1, 2, or 3 dimensions, and to see those selections propagate live across all open visualizations of the data (e.g., graphs, maps, diagnostics charts, etc.). Our aim is **not** to develop a single visualization tool that will solve all problems in science, but rather to focus on providing an environment that enables advanced linking and visualization of datasets while also providing a way to interface with existing visualization solutions. In this way, *glue* can fit into the ecosystem of tools already commonly used by scientists to **provide new and powerful ways of exploring their data**.

The two most critical unique features of "glue" explain its name. First, the program allows users to "**glue" together different data sets** in its exploratory visualization environment, without ever requiring a merged file. Shared attributes—for example coordinates in images, maps, or tables—are linked (glued) via a GUI, as either an identity (as in "latitude" in map file A is the same as the column called "lat" in file B), or as a mathematical expression (as in "latitude" in image C is ="time" in file A, multiplied by 15). This process eliminates much of the data munging common in data science today, but the approach is not common in other modern tools. Second, selections of subsets of data within displays, made interactively or algorithmically are linked amongst displays, effectively **"glueing" displays** of information together visually in real-time.

A live linked-view approach to data visualization, often called "brushing and linking," was first proposed by statistician John Tukey in 1977 as the best way to explore relationships in high-dimensional data. Tukey's ideas, inspired by the challenges facing particle physicists, were instantiated in the domain-agnostic "DataDesk" program in the mid-1980's. DataDesk was initially Mac-only because it required a mouse to select subsets on-screen. So, only a few lucky Mac-wielding scientists (including PI Goodman) were early adopters of DataDesk, and most researchers continued to write scripts based on *a priori* assumptions, missing out on the the advantage of real-time exploration of large data sets. Once mice became ubiquitous and data sets grew large, the power of brushing and linking became obvious in business analytics, where the approach took off (e.g. Spotfire, Tableau). Meanwhile, though, scientists mostly kept writing scripts.

With glue, we seek to bring real-time linked-view data exploration to researchers whose data sets are inherently high-dimensional. Nearly all scientists we teach to use *glue* were not previously aware of the brushing and linking paradigm, and they are immediately converted to this mode of thinking and exploration once they realize that it is possible. The *glue* package is designed from the ground-up for **multidisciplinary** data analysis, and it has been used by researchers to explore astronomical, medical, genomic, geoscience, civic, and commercial data sets. The extant software has largely been written by two lead developers employed through NASA contracts to PI Goodman, with just under 10% of the code having been contributed by others. With a user base moving from hundreds to thousands, we request funding that will allow us to: 1) build a **strong open source user and developer community** in order to ensure the long-term sustainability of glue; and 2.) allow us to **implement missing critical features** in *glue* that will be enhance its utility across more fields, in turn widening the developer pool.

2. Overview of current functionality

2.1. Graphical user interface

At present *glue* has one main (desktop-based) graphical user interface (**Figure 1**). This interface consists of a *canvas* area in which users can add various data *viewers*, each of which is effectively a new window on the canvas, in order to create an environment best suited to a particular problem (North & Shneiderman, 2000). On the left is a sidebar containing a representation of all loaded datasets, as well as controls for the data viewers once these are created. Loaded datasets can be of heterogeneous formats, and they never need to be merged to be used in concert.

In a typical workflow, users load datasets, glue shared attributes of datasets together in a GUI dialog, drag data sets onto the main canvas to create viewers, and explore the data by making selections in different viewers. Selections made in one view propagate live to all other linked views, instantiating the **brushing and linking** paradigm. The full data sets are shown in gray by default while selections are highlighted in colors that can be customized by the user (see red and blue highlighting in Figure 1).

Data from different sources are combined together on-the-fly, which extends the brushing and linking logic to work across datasets.

New "derived" data components (variables) can also be created within glue, using any combination of data already loaded. For example, a user might load in a table listing the longitudes, latitudes, and depths of earthquake hypocenters (origins). A user interested in visualizing the global three-dimensional distribution of hypocenters on a cartesian grid can define new components, x, y, and z, via a spherical to cartesian conversion. Setting up such *derived* components is easily done in a window that allows a user to construct any arbitrary mathematical expression (using both functions and arithmetic operators).

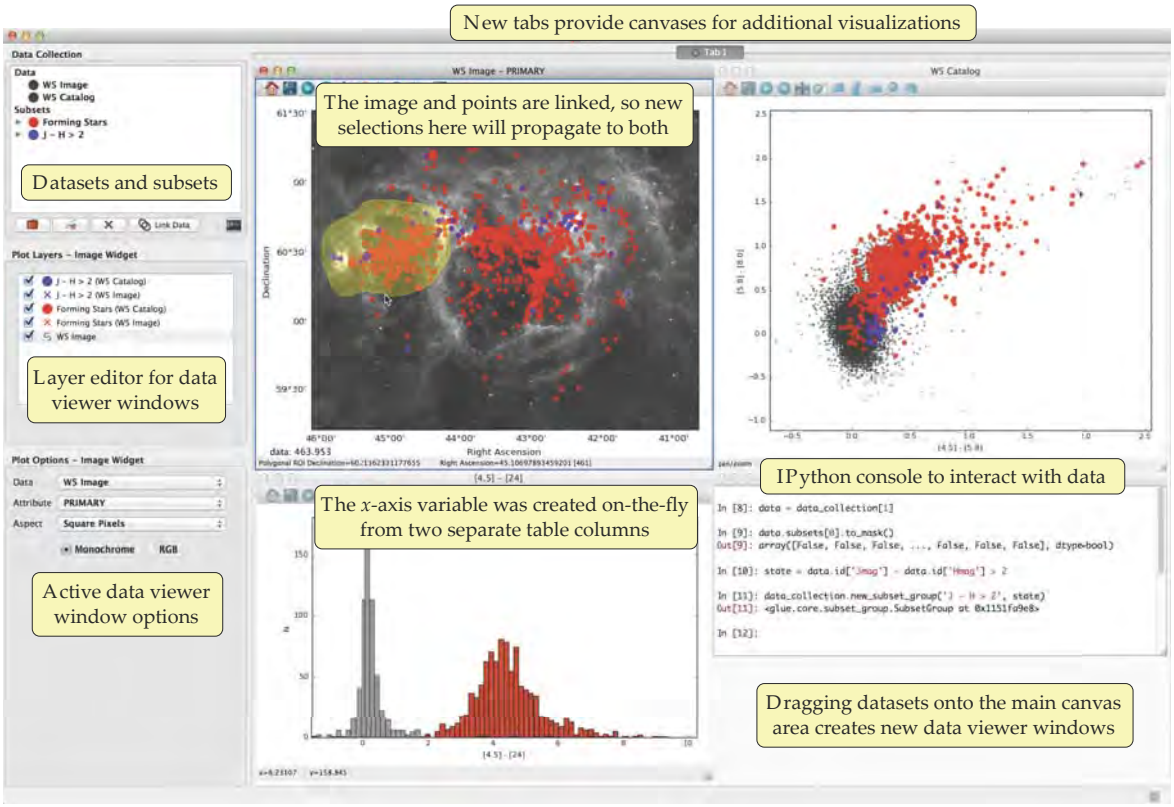

Figure 1: The main *glue* application window (with caption notes shown with a yellow background).

The standard distribution of *glue* offers **scatter plots**, **histograms**, and **image viewers**, and many more specialized and domain-specific plots are available as plug-ins. The image viewer displays 2D images, as well as 3D "cubes" using a slider that adjusts which plane of the cube is shown in the 3D view. In addition **3D scatter plots** and **volume-rendering viewers** are also available (see Figure 2). The rightmost panel of Figure 2 shows data from the famous 1854 cholera epidemic in Soho (Snow 1855), presently featured in PI Goodman's online HarvardX course on Prediction.

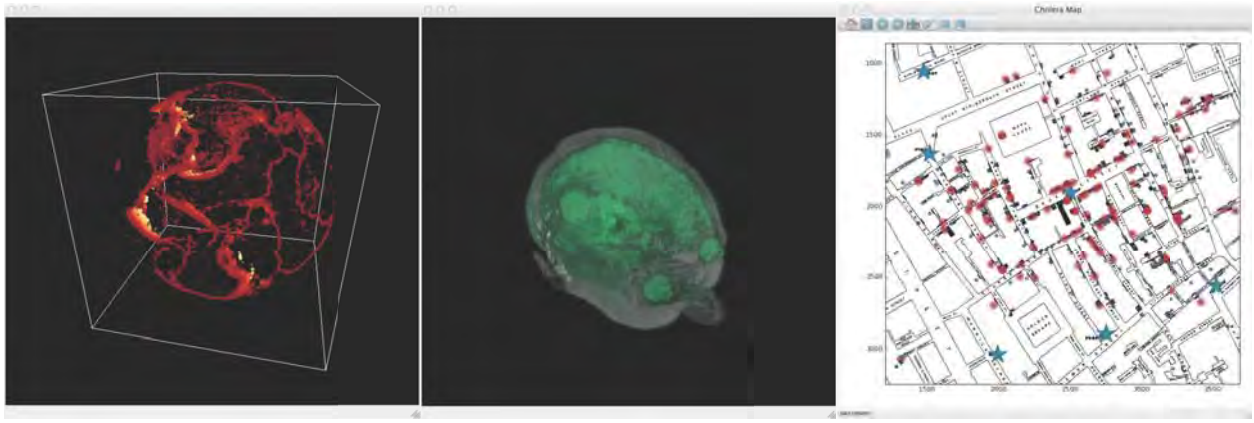

Figure 2: Examples of data viewers in *glue*. From *left* to *right*: 3D scatter plot visualization of earthquake locations (color-coded by depth under the surface), 3D volume rendering of medical/radiology data showing two datasets simultaneously, and a custom data viewer showing the now famous map by John Snow of the 1854 cholera outbreak in the neighborhood around Broad Street, London (Snow 1855). All deaths are shown as small blue circles, a subset of deaths selected in another viewer is shown as red circles, and the location of water pumps are shown as stars.

2.2. Python interface

Using *glue* through the graphical user interface (GUI) offers a wide range of functionality, but an important feature for researchers is the ability to directly access all the data being visualized and be able to interact with it in a non-graphical way. In particular, researchers may want to select subsets in a very specific and reproducible way (for example depending on complex models) that would be difficult to emulate graphically. By writing glue in Python and leveraging existing frameworks (Qt) to build its GUI. we have made it possible for the user to interact with the Python layer in the following ways:

- As shown in Figure 1, a *glue* user can launch a **built-in IPython terminal** (Perez & Granger 2007) right on the main canvas, which can be used to access any of the underlying data objects and components currently loaded, along with existing viewers. From this terminal, users can also create new components (variables), modify or create new selections, and load or create more datasets.
- The *glue* application can also be launched directly **from an existing interactive Python session**, and data objects can be passed to the *glue* application. While and after using *glue*, users can then access any information about the state of *glue* from the original Python session from which they launched *glue*.
- **Python scripts** can be used to set up various data objects, links, add any customization, and then launch *glue*. Users can programmatically extend the interface in various ways to suit their data analysis workflow. By way of examples, users can define their own: data loaders (such as for domain-specific file formats); colormaps; data viewers; and linking functions. Many of these can be defined without any knowledge of how to build a GUI, and without an advanced knowledge of Python.

These ways of interacting with *glue* from Python make it easy for scientists to create a **more reproducible workflow** for data analysis and to customize and extend *glue* to be an environment best-suited to their needs. Sharing custom data-loaders, analysis tools, and plot types has already been, and will continue to be, a great way for less full-time developers to contribute to the open-source *glue* code base.

3. User and Developer Community

No commercial packages (see Section 5) offering brushing and linking support the **high-dimensional data formats (2D and 3D images and cubes)** that are so important in science and medical imaging. The initial motivation behind *glue* was to create an open-source, extensible, alternative to these packages that would support graphical selection and linked-views of high-dimensional data, and would be easy for scientists to use. As development and prototyping began, requests for extended features started to pour in, and the scope of the *glue* project quickly became broader. Initial development was carried out in 2011-12 by Chris Beaumont and Thomas Robitaille at Harvard, as part of the Seamless Astronomy program led by PI Goodman. After Beaumont received his PhD in 2013, he worked on *glue* as a full-time developer at Harvard until the end of 2014 when he took up a data-scientist position at Counsyl. Since 2015, Robitaille, who is also a leader on the *astropy* project, has been *glue's* lead developer. Robitaille is now funded via NASA contracts awarded to PI Goodman to develop *glue* for use with data from the upcoming *James Webb Space Telescope* (successor to the *Hubble Space Telescope*). Robitaille is a key leader of the glue effort, and he will lead the technical coordination (commits) of the open-source contributions to glue that the work proposed here will generate.

At the start of the glue project, our small team (then entirely at Harvard) carried out formal user studies to motivate the design of the interface. This work was led by PI Borkin, who specializes in visualization and human-computer interaction, and it yielded the present interface and design philosophy behind *glue*.

From very early on, we started building a community of users by setting up a user mailing list, and we quickly found that users across different fields of science were interested in using *glue*. The package was then adapted to make it easy for researchers to develop new functionality for *glue* with a minimal understanding of how to write GUIs, which allowed them to easily support their own data files and even create custom data visualizations. As a result, the line between users and developers became blurred (since even normal users were able to customize the *glue* environment). This has proved to be a good strategy for ultimately getting more people involved in the development of *glue* itself.

Over time, the community of users has grown to include people over a wide range of subjects – including astronomy, geographic information systems (GIS), genomics, medical science, and even data scientists in private companies (such as Yelp). As discussed Section 9, we estimate that there are now at least several hundred *glue* users. The core development has been carried out by Beaumont and Robitaille. To ensure the long-term sustainability of the project, **we clearly need to expand the contributor/developer base**.

4. Advancing research capabilities across domains [SI²]

In this section we give a short list of example applications of *glue*, demonstrating its broad applicability.

4.1. Geographical Sciences: Remote Sensing and Imagery

Satellite images are often taken at many wavelengths – for example, the Landsat 8 satellite includes two instruments that provide in total nine shortwave bands and two longwave thermal bands. Each pixel in the resulting image therefore includes 11-dimensions (wavelengths) worth of intensity information. Analyzing such data typically also involves computing vegetation indices – for example the Normalized Difference Vegetation Index (NDVI) which is an arithmetic combination of visible and near-infrared bands. In collaboration with Robin Wilson (University of Southampton, England) we have developed a plugin named **glue-geospatial** that adds support to *glue* for common file formats used in geographical sciences. With this plugin, it is trivial to load into *glue* and display the multi-band images, compute these indices, then interactively or algorithmically select values based on these indices, and see where these regions are located spatially.

4.2. Astronomy: Investigating the structure of our Galaxy

Glue is particularly well suited to datasets in astronomy (the PI's original discipline) that include multiple images of the sky and catalogs of point sources. The *glue* package has been used in dozens of astronomical studies, but to highlight one example of a recent prominent study, consider the work by Zucker et al. (2015) that used glue to identify a set of large-scale filamentary structures, known as "Bones," that offer a completely new way to trace the structure our Galaxy. These structures can be seen both in infrared images that show the dust as dark regions where the background light is absorbed, as well as at radio wavelengths, where the gas emits radiation (at wavelengths that depend on the velocity of the gas). Using *glue*, Zucker et al. were able to link together the 2D infrared (plane-of-the-sky) and 3D radio (velocity-resolved) images, find salient features that exist in both, and use the results to identify features which best define the so-called "skeleton" of our Milky Way Galaxy, in 3D.

4.3. Medicine: Radiology and Segmentation

We are collaborating with team members Bruce Rosen and Jayashree Kalpathy-Cramer at the Martinos Center for Biomedical Imaging at Massachusetts General Hospital (Charlestown, MA) to ensure that *glue* will be used most effectively in medical research. One area of investigation is the use of *glue* to visualize data from 3D radiology scans (such as MRI or CT scans) in order to help visualize and analyze the morphology of brain tumors. Current software solutions often rely on inefficient manual identification of tumors in 2D slices, one slice at a time. With *glue,* researchers and clinicians alike can use more sophisticated selection and linking methods in order to isolate tumors and/or important anatomical features and help place them in context in 3D visualizations of a wide variety of scan modalities. We are in the process of developing a **glue-medical** plugin that: 1) understand several common file formats (including so far DICOM, NIFTI, and NRRD) and coordinate systems commonly used in radiology scans; and 2.) makes it possible to load in pre-existing selections/segmentations, as well as export new selections and load them into other software used in the medical sciences, such as 3D Slicer.

4.4. Science Education: WorldWide Telescope, glue, and CODAP

Thanks to PI Goodman's leadership in the NSF-supported WorldWide Telescope (WWT) computer & outreach programs, we are engaged in discussions about glue with members of the Concord Consortium (CC), an organization well known for its work on incorporating technology into STEM education. (Goodman is PI of a WWT-based outreach program, for which Chad Dorsey, CEO of CC, serves as advisory board Chair). Given that our glue team has already succeeded in using WWT's HTML5 API to include a WWT viewer inside glue's desktop interface, we are confident that other HTML5 tools with APIs can be turned into glue viewers as well. One such tool is the NSF-funded Common Online Data Analysis Platform (CODAP), developed by William Finzer of CC. CODAP's functionality is similar to glue for tabular data, and it can display information on map base layers. But CODAP is limited to data sets with fewer than about 500 rows and cannot handle images (2D) or volumetric (3D) data. We discuss our planned collaboration with CC further in Sections 6 and 13.

4.5. Beyond science

glue can and has already been used to explore a wide range of open datasets, such as civic, national, or international open datasets. Private sector companies with data science divisions are also interested. For example, we received out-of-the-blue requests for support from data scientists at Yelp who were using *glue* to analyze their data.

In our own most recent "non-scientific" application of glue, we have set up a tiny antenna connected to a Raspberry Pi on a rooftop at Harvard for the purpose of monitoring air traffic in real-time. For less than $100, this sensor system gives us the 3D position and other parameters (including vertical speed, ground speed, heading, and so on) of all aircraft within roughly 100 miles of Boston. The aircraft data can be visualized in glue in conjunction with 2D and 3D topological, political, and weather maps. We plan to

share and use this data feed and visualizations of it as a: 1) relevant everyday-life example that laypeople, including K16 learners, will understand; and 2) a testbed for ingesting live-stream data to glue.

5. Comparison to alternative or existing elements [SI²]

The aim of *glue* is not to provide a fully generic visualization tool, but instead to provide an efficient way to *link multi-dimensional datasets* together and allow the user to visualize these using a variety of viewers, including viewers that they may develop themselves. Other software packages allow brushing and linking *within a single dataset*, but none offers links amongst multiple data sets, or links to/from the 3D data viewers so critical in Astronomy, Medicine, and Geosciences. Examples include:

- **DataDesk**: first released in 1986 and developed by Paul Velleman, DataDesk. targeted at tabular data (does not support gridded data such as images or data cubes). Brushing and linking supported, but data from different sources cannot be utilized simultaneously. Commercial, $799 for a single license.
- **Spotfire**: first released in 1996, provides the ability to construct different views of the same dataset using histograms, scatter plots, maps, line plots, and a 3D scatter plot viewer. Does not support gridded data formats (images and cubes). Does not allow linking of different datasets together. Desktop, $650/year; cloud version $2000/year.
- **Tableau**: launched 2003 (based on "Polaris," see Stolte, Tang & Hanrahan 2002). Linked views of tabular data. Similar to Spotfire in that it provides a desktop and cloud version. Good support for big data, but – as is also the case for Spotfire and DataDesk – the number of input data formats is limited. No 3D visualization. License for (non-pro, non-student) desktop version, $999.
- **TOPCAT**: first released in 2003. Domain-specific (astronomy) package for tabular data. Histograms and 2D and 3D scatter plots. Brushing and linking within a dataset (Taylor 2005). Non-tabular data or arbitrary linking of different datasets not supported, but basic links can be made to image-based data via a message-passing hub (see "SAMP," in Section 7). Runs in pure Java. Free.

While these solutions are well suited to the case of data of common visualizations of data from a single table, *glue* includes a number of **unique and advanced features** (Borkin et al. 2017; Qian et al. 2017[1]):

1. **Linking of multiple datasets via functions that connect attributes of one data set to another.**
2. **Loading and linking images and 3D data**: *glue* supports reading in images, sets of images (where each image is a component of a single dataset), data cubes, and higher-dimensional objects, and allows these to be linked with other datasets, including tables (Qian et al. 2017).
3. **Python integration**: *glue* includes the ability to drop down into a Python terminal at any point in the data exploration, and also allows users to easily customize *glue*.

Finally, *glue* is **open source**, **openly developed,** and **free,** so researchers can easily join the effort to help develop it within their fields.

One open-source library worth mentioning here is *bokeh*, which is a recently developed Python and javascript library for advanced (1D and 2D) visualization in the browser that supports ways of selecting data in visualizations. However, *bokeh* is just a framework for the actual visualization and not an application, and thus does not deal for example with loading data or setting up links between datasets. It is therefore not an alternative to *glue* but rather a library we can (and plan to) use to provide a version of *glue* that works in the browser (see Section 6.2).

[1] Preprints of these new submissions to IEEE Trans. on Vis. & Comp. Graphics are available upon request. A key table presented in the articles, contextualizing glue's features with respect to related software, is also on the glue web site.

6. Research and development agenda [SI²]

Our proposed work falls into two main categories: development of new and critical features (this section), and work to expand the community of contributors/developers in order to ensure the long term sustainability of the project (Section 7).

The development essential to tackle modern scientific data challenges and workflows across fields cannot be **carried out by a single developer** – both because no single person could have the needed expertise, and because code written by only one person ("hero") is usually only well-understood by that person, making it hard to sustain beyond the end of the hero's involvement and/or funding. Building on the modular architecture of glue, we seek to use SSE funding to engage with researchers and open source developers from (at least) the Astronomy, Medicine, Geospatial and Science Education communities, a to collaborate on developing specific modules. Expanding the community both of users and developers in this multi-disciplinary way should make *glue* more sustainable in the long term.

6.1. Support for "Big Data"

Objectives: develop a **data abstraction layer** that separates the data access and computation, e.g., of histograms, from the visualization. Develop proof of concept data objects, including ones that use a web API to **access remote data**. Implement more efficient ways to **display larger numbers of points**, and **optimize** the calculation of data **selections** for large datasets. Finally, implement support for **live** and **time-dependent** data.

The data linking and selection paradigm in *glue* works well for catalogs with ~10^6 rows and gridded datasets with ~10^9 pixels/voxels. However, as datasets go beyond these limits, some components of *glue* slow down enough that interactive data analysis becomes difficult. There are several bottlenecks in the performance, which we describe in the following sections, along with our proposed solutions.

Loading of large datasets: In many fields of science (as well as in industry), it is becoming increasingly common to have datasets that exceed the size of the available physical memory on a computer, and may in some cases even be too large for users to have the entire data stored on local computers. Two extreme examples supported by NSF are: the *Large Synoptic Survey Telescope* (LSST; Tyson, 2002), which will map a large fraction of the night sky every few days, producing unprecedented data volumes (15Tb per night); and the Daniel K. Inouye Solar Telescope (DKIST), which will produce 70Tb of multi-wavelength imaging data per day (Wampler & Goodrich, 2009).

To give *glue* the ability to access data from local or remote files and data stores, we will develop a data abstraction layer that will separate the data access and computation (such as computing histograms or fixed-resolution buffers) from the remainder of *glue*. With this in place, we will be able to implement data objects that interact behind the scenes with either servers that host large datasets (and provide significant computational capabilities) or data objects that access local data in an efficient way. Data objects loaded into the application will all look the same, despite some being backed by files, some by databases, and some by remote data servers. For reference, this kind of seamless data access interface is demonstrated in Tableau, and no changes to glue's data access GUI would be required by this development.

As an example, the *yt* package (Turk et al., 2010; previously funded by an NSF SSE award), provides a common data interface to many file formats that can efficiently store multi-resolution simulation data. Robitaille will work with the *yt* developers (including John ZuHone at the Harvard-Smithsonian Center for Astrophysics) to provide a *glue* data object that leverages the power of *yt* behind the scenes, and allows the user to seamlessly interact with these large datasets without ever loading the whole dataset into memory. Similarly, we will work on a data object backed by the *blaze* package, which provides extremely fast data access to different data sources such as databases.

Fast visualization: Another bottleneck when it comes to big data is in computing and displaying visualizations in "real time." For example, *matplotlib* (Hunter, 2007), which is currently used for the 2D scatter plot visualization becomes slow once ~10^6 points or more are shown, severely limiting the size of the largest catalogs that can be visualized. We plan to explore working around this limitation in two ways: first by switching to using an OpenGL-backed viewer (which we are already using for the 3D data viewers), and second by working on smart ways to display the data, for example by avoiding showing all the data at the same time (e.g., Fekete, 2002; Cui et al., 2006; Ellis, 2008; Shneiderman, 2008). In regions of high point density, we could for example group neighboring points together on-the-fly to avoid ever showing more than ~10^6 individual points. Alternatively, we could use a package such as datashader (which essentially computes density plots on-the-fly) to render these kinds of plots.

Fast linking/selection: Finally, provided that the data loaders and visualizations can keep up with the volume of data, a final bottleneck is the computation of selections. For instance, if a user draws a polygon to select points from a 3D scatter plot viewer, or voxels from a 3D volume rendering in an arbitrary direction, *glue* needs to be able to compute very efficiently which elements of the subset fall into the selection. While these operations are conceptually simple, the large number of elements in the dataset make it a challenge for selection to update fast enough for exploration to still feel interactive. Computing and updating selections can be sped up using the same approach and infrastructure as for the data abstraction layer – if the raw data are stored on a remote server with significant computational capabilities, the abstraction layer could also include ways to compute selections remotely. Harvard's Odyssey cluster and Research Computing Environment (available to PI Goodman at no direct cost to NSF) offers an excellent platform for testing many of these ideas about data abstraction layers.

Live data/sensor networks: Another example of challenging data to deal with is *live* data, for example data from sensor networks. We therefore plan to extend the data abstraction described above to allow for real-time data streams, so that the data viewers and any data/selections within them change in real time (*glue* already supports having time axes in data, but does not support data that is being updated *live*). This will be done by allowing data objects to notify *glue* whenever the underlying data changes, forcing the viewers and selections to be updated. In addition, we plan to implement the ability to record and replay this type of data so that the user can replay the changes over time using a simple data slider. The aircraft-monitoring data mentioned in Section 4.5 will be used to test glue's ability to handle live streams.

6.2. Interactivity in the browser

Objectives: develop ***bokeh*-powered data viewers** that can then be used to provide a **Jupyter notebook**-based data exploration environment, and make it possible to **export fully interactive websites** from *glue*.

At the core of the *glue* package is a Python library that deals with loading and representing datasets and data components, as well as data linking and selections. This library is well isolated from the GUI code, which means that the core library can easily be reused for other applications. Leveraging this modularity will facilitate the creation of a browser-based variant of *glue*. Rather than develop a complete web application from scratch, we propose to take advantage of the existing Jupyter Project infrastructure. The Jupyter Project has developed the **Jupyter notebook,** a widely used application to create notebooks in the browser for Python and other programming languages. Users can add *cells* which can contain code, headings, text, figures, and so on, and the code can be directly executed from the browser. Recently, the Jupyter Project have released early versions of a new interactive environment for the browser – **Jupyter Lab** – that allows multiple widgets to be arranged in a browser window, in the same conceptual way as multiple data viewers can be arranged in *glue*.

Implementing *glue* inside Jupyter notebooks and Jupyter Lab would have the following benefits:

1. It would be possible to **decouple** the computer on which the data are stored and the computation is done from the computer on which *glue* is used. Thus, a user with any computer with a browser could then use *glue* interactively and tap into significant computational resources.
2. Having a web-based version of *glue* would open the door to having a way for **multiple users** in different locations to collaborate on the data exploration in real time.
3. Thanks to the notebook, researchers would be able to record an accurate record of their data analysis or computation and share it with other scientists, thus providing **full reproducibility**.

The new *glue* data viewers inside Jupyter notebooks and Labs *glue* will leverage *bokeh* for visualizations and selections *and* connect to the core *glue* code, enabling linking of datasets, creation of new variables, and all the rest of the *glue* functionality. Leveraging existing open source technologies (Jupyter, *bokeh*) will allow fast integration with *glue* as well as re-using an environment that will be familiar to many users.

In addition to allowing *glue* to be used in notebooks to carry out new analysis, we also want to expand the **export capabilities** of the desktop application. For some simple visualizations, *glue* already offers limited javascript export. Static graphs can be exported to plotly, and graphs with brushing and linking can be exported as javascript to a custom prototype tool built by the extended *glue* community, called *d3po*. PI Goodman has highlighted *d3po* to great effect in recent presentations, as well as in the *"Paper" of the Future*, available online (Goodman et al. 2014), which demonstrates several interactive technologies expected to revolutionize scholarly publishing. Our plan now is to extend *d3po*-like functionality to more of *glue*'s features (e.g., images, 3D plots), so that interactive plots of any kind can be readily shared on the web, including in future scholarly communication.

We propose to enable export in more general ways by generating a *Flask* or *Django* web app that includes the data as well as a page with viewers powered by the *bokeh* library. Ideally, this kind of interactive visualization could be hosted by scholarly journals in conjunction with publications. PI Goodman wrote *The Paper of the Future* as part of her work on the 2014 AAS Publications (Modernization) Task Force, and Robitaille became the AAS Journals' Software Editor in 2016, so the time is ripe to develop enhance *glue's* web publishing abilities. AAS is committed to piloting interactive graphics attached to data, just like the kind glue can generate, and we expect other publishers will follow suit with these added-value products.

6.3. Interoperability with widely used tools [SI²]

Objectives: develop **new** data **viewers** that embed **existing tools such as the WorldWide Telescope** inside *glue*. In addition, develop plugins that allow **communication with other tools**

In addition to making it possible to use *glue* in Jupyter notebooks and leveraging existing technologies for fast data access, there are a number of other ways in which we can build on existing infrastructure to complement the functionality in *glue.*

Leveraging third-party tools inside *glue*: One example of a third-party tool for which we want to develop an integration layer is WorldWide Telescope project (WWT; now an open source project, first developed at Microsoft Research), which allows advanced visualization of astronomical datasets on the celestial sphere and GIS-style data on the Earth. An HTML5 version of WWT (with an external API) is available and can be embedded inside a Qt widget and made into a *glue* data viewer, which would be able to show datasets and subsets. We would then directly benefit from all the functionality built-in to WWT with minimal developer effort. Early tests of this approach have already been successful, and we plan (see Broader Impact) to test a similar approach using the CODAP STEM outreach visualization tool.

Connections to other software: In research, users often have tools that are familiar and do a specific task well, so it is important that we make it easy for *glue* to communicate with other popular tools. There is no unique way to accomplish this communication, but there are several promising avenues to explore. In Astronomy, the *simple application messaging protocol* (SAMP; Taylor et al., 2015) allows applications to

communicate with each other, including transferring information about datasets and data subsets. A number of astronomical tools already implement communication using SAMP – for instance WWT (Goodman et al. 2012), the DS9 (Joye & Mandel 2003, Joye 2006) package for image and catalog visualization, and the TOPCAT (Taylor 2005) catalog exploration tool. We plan to develop a SAMP connector for *glue*. And, we will seek out similar application-linking opportunities beyond Astronomy, where the same general concept of allowing data and subsets to be sent from/to *glue* can also be extended to any package with an API, for example the *Blender* package for 3D visualization (Roosendaal & Selleri, 2004, Naiman 2016), or the CODAP tool (xx reference to be addedxx).

7. Building a sustainable user and developer community [SI²]

Objectives: expand the user base, train users to develop custom visualizations for *glue*, and recruit and train new developers for the projects, in order to achieve long term sustainability.

The ensure that *glue* is **sustainable in the long term**, beyond the SI2-SSE funding period, and that new functionality will continue to be developed and maintained, we need to expand the community of *glue* contributors and core developers

The core *glue* package currently has only one funded core developer (Robitaille), with a few additional developers each helping with development plugin packages. At this point in time, funding from the SI2-SSE program is therefore crucial to systematically **recruit**, **train and fund more developers** to help carry out the work described in Section 6, and to make the project more sustainable. Concretely, we plan to:

- Hold **workshops** to train new contributors and work together on projects. We have held tutorials in the past and have found that the modular nature of *glue* means that starting developing components of *glue* via the plugin infrastructure is an efficient way to get people involved, because writing plugins does not often require knowledge of the whole *glue* code base. The workshops will be a combination of tutorials and hack/project time, where people can team up and develop new functionality for *glue*.
- Fund consulting **developers** with experience with (for example) *yt*, the Jupyter Project, science education, and domain science areas, so that they can help implement new features and get involved in the development of *glue*. We will work with developers as consultants, either directly or through organizations such as the Concord Consortium or NumFOCUS (which supports sustainable development and represents a number of open-source projects relevant to our work on *glue*, including *Astropy*, *yt*, and Project Jupyter). Note that we are not asking for a tremendous amount of money to pay consulting developers. Instead, we are relying on a model that has worked well in other open-source efforts, where some organizations or individuals contribute development at no cost, because the products of the development serve their own interests. The requested consulting funding will be used strategically, only to pay for development we have no way of acquiring via opportunistic collaboration.
- Fund several **undergraduate students** at Northeastern University to work, under the supervision of PI Borkin, via the Co-op program. This unique program affords undergraduates the opportunity to work full-time on a project for six months and gain real-world research and job experience. Students will focus on user interaction efficiency (Borkin's particular expertise), domain-specific plugins, STEM education, or other areas related to the *glue* project, as their talents best suit. The Co-op students will learn invaluable skills relating to software development and contributing to an open source project while developing useful functionality for *glue*.

With this highly-leveraged plan, we hope to not simply implement the functionality described in Section 6, but also to recruit people interested in working on *glue* in the longer term. By finding new contributors/developers in different fields of science and science education, we also open up more possibilities in terms of applying for funding across different scientific disciplines, beyond the funding period for the SI2-SSE program.

8. Project, outreach, and education plan [SI²]

We propose to carry out the work described above over three years, using the following breakdown:

Year 1: We (primarily Robitaille) will design the **abstract data layer** needed to implement support for **big data** and **remote data**, and will then begin to implement data objects for specific data sources. We (Robitaille, with ZuHone+) will investigate mechanisms to **speed up** and potentially **offload** to remote servers **heavy computations** related to making selections in large datasets, and will work on ways to efficiently plot data so that we are no longer limited by the restrictions on the number of data points that the current viewers can display. We (primarily Goodman) will run two in-person and one online **workshop** to **increase our user and developer base**, and we (all) will also work one-on-one with a few researchers in different fields and **collaboratively develop new functionality** to make *glue* suitable for their field. Robitaille will work with developers from new fields as they become available, and Goodman will be responsible for strategy and decision-making on the balance of paid vs. volunteer contributions.

Year 2: We intend to spend a larger fraction of this year **helping/encouraging** new members of the user and **developer community** to develop features, rather than developing all new functionality ourselves (under the NASA contract that pays Robitaille). By progressively moving to a **community coordination** role (Goodman, Robitaille) rather than a pure development role, we will be setting up the project for **long term sustainability**. Specific projects in Year 2 will include a focus on CODAP-glue integration that will shape the nature of a consulting agreement to be carried out in Year 3. We anticipate that the CODAP work will relate to development anticipated for Year 3, re:glue's functionality as an HTML5 API, and *vice versa* (functionality of HTML5 APIs within *glue).* Borkin and Co-op students will work with the broader glue collaboration on how to best (functionally and visually) integrate newly-developed functionality into the larger glue package, and to consider any needed revisions to glue's user-facing interface.

Year 3: We will collaborate with interested contributors on developing Jupyter notebook and Jupyter Lab-based interfaces to *glue*, as well as writing **documentation** (Robitaille) showing how to do the full **data exploration** in the **notebook**. In parallel, we will work collaboratively on new data viewers, based on **WWT, CODAP,** and other **domain-specific priorities** that arise in Years 1 and 2 (likely medical). Finally, by Year 3, we will complete work with developers in fields where glue is widely used in order to develop connections between *glue* and their domain-specific software (e.g. via SAMP in Astronomy).

PI Goodman (Harvard), and PI Borkin (Northeastern) will work together with lead developer Robitaille (funded by NASA) to coordinate the user and developer community for *glue,* and provide management for the *glue* project (including ensuring that the objectives set here are met). Northeastern University will fund undergraduate students to work on *glue* through their Co-op program, while Harvard will contract with developers as consultants to carry out work that cannot be accomplished by students or volunteers.

Over the three year grant period, PIs Goodman and Borkin will also publish at least two articles on *glue*'s approach and utility (cf. Borkin et al. 2017; Qian et al. 2017). Each year, Harvard will be responsible for hosting ~3 *glue* workshops, in order to continue to recruit new users and developers. As opportunities arise, some of these workshops may be targeted (e.g. a special meeting of medical imaging experts), or hosted in conjunction with existing conferences, to avoid additional travel on the part of participants.

9. Usage and development metrics, and measures of success [SI²]

Measuring the exact usage of an open-source project is not easy, since it is in principle possible for anyone to download the package and never get in touch with us. Furthermore, there are no trivial download statistics, because there are a number of venues for installing *glue,* some of which are beyond our control (for example the Debian apt-get package manager, and the Anaconda Python Distribution). However, we can make the following observations about *glue* usage to date:

Our first two papers explicitly about glue are being submitted IEEE Journal xx now (Borkin et al. 2017, Qian et al. 2017). Those will give an opportunity to track citations in the future, but domain end-users are not very likely to cite a visualization research article. To enhance reproducibility for research done with *glue*, we have started assigning DOIs to each *glue* release (via Zenodo). We will also be advertising these release DOIs as a means of citation, so perhaps they will be a useful usage metric in the future.

Contributor/developer metrics are easier to track. So far, **23 individuals have contributed to the *glue* code** and 93.7% of the commits in the main repository have been made by Chris **Beaumont** and Thomas **Robitaille** (the former and current lead developers respectively), 4.2% of the commits have been made by the next four contributors, and 2.1% of commits have been made by the remaining 17 contributors.

Glue Usage Indicators

- The Python Package Index and Anaconda Cloud statistics show each major release of glue (about 4/year) being downloaded between **1,500 and 2,000 times.**
- The newest introductory **video** (*Multi-dimensional linked data exploration with Glue*) on the glue website has been viewed over **4,500 times** in a little under a year.
- The *glue* repository on GitHub has been **starred over 250 times**. Since GitHub account holders can star a repository, this is an indication of high-end user and/or developer interest.
- At least **100 researchers** have been **trained** in person by members of our team, at in-person tutorials held in conjunction with glue workshops.
- We are contacted **several times/year** by "power" users from disciplines where we have not worked, and from the private sector, showing that people are finding out about *glue* on their own and using it effectively.

Our **targets metrics** by the end of the award period for **usage** are: more than **200 subscribers** on the user mailing list and more than **30 citations** in scholarly publications to the DOIs for the *glue* releases, spread over at least 3 different fields of science. For **development**, our targets are: adding at least **5 new contributors** who *cumulatively* contribute more than 5% (and ideally more than 10%) of the **new** code to the **core** *glue* package over the 3 years of the award, and adding at least **3 plugins** for *glue* created and led by other researchers outside the core team. Recent experience suggests these are conservative goals.

10. Engineering process (development to release) and license [SI²]

When the project was first funded (in late 2013), NASA and Harvard negotiated to assure that *glue* would remain open-source, under a **3-clause BSD-license.** The full *glue* code base is hosted on GitHub, where contributors typically open a *pull request* that contains a set of changes that are then be reviewed by the lead developers of *glue*. A large number of *unit tests* are included to ensure that the code is stable over time, and all tests are automatically run whenever a user proposes a contribution, to make it easy for the developers to determine whether a certain set of changes will break the package. This is a workflow adopted by many other popular scientific packages, including *numpy* (van der Walt et al., 2011), *scipy*, *matplotlib* (Hunter, 2007), and *astropy* (Astropy Collaboration, 2013). Adopting the same workflow as these packages will make it easier to recruit new contributors who are already familiar with the workflow.

Several versions of *glue* have already been released to the community. Every few months we make a new major release that includes new functionality (e.g. 0.7). In between these we typically make minor releases (e.g. 0.7.2) that fix bugs and issues with the current major release. We typically create a tar file of the release which is uploaded to the Python Package Index (the standard place to release Python packages). We have worked with developers at Continuum Analytics to ensure that *glue* is made available by default to the conda tool in the Anaconda Python Distribution, currently a very popular mechanism for setting up Python on any platform. Every time a new release of *glue* is made, it is then made available for conda. In addition, we work via Linux package managers to ensure that *glue* is available to Linux users, for example in the default package repository for Debian-based Linux systems. We will continue to make

regular releases over the period of the SSE grant, to make it easy for scientists to install and update *glue* as new features become available. As part of the upgrade and installation procedures, we will collect **feedback** on new developments to help ensure that we meet the objectives outlined in previous sections.

11. Security, trustworthiness, reproducibility and usability [SI²]

Security: the main interface for *glue* is a desktop application, and as such, there is no danger of someone gaining access to another user's data or session (unless the computer itself is compromised). For the proposed Jupyter implementation, the security aspect will be the same as when using the Jupyter notebook for other types of analysis (that is, *glue* does not introduce any additional security concerns): whoever sets up the Jupyter server should ensure that it is password-protected and runs over HTTPS (both of which are supported by the Jupyter server).

Trustworthiness: *glue* currently includes many tests that are run every time a change or proposed change is made. We write regression tests for every bug that we find, and unit tests for every feature that we add, to make sure that features do not suddenly break and that the code is as trustworthy as possible.

Reproducibility motivates: our Python interface to *glue;* *glue* in the Jupyter notebook; and issuing DOIs for *glue* code versions. In addition, we allow users to save and restore complete sessions in the desktop application, so that the whole environment can be recorded.

Usability. We regularly work directly with users to make sure that the interface remains intuitive and usable. User feedback will continue to be collected both in workshops, and in formal usability studies carried out by PI Borkin and her Co-op students at Northeastern. Borkin's research focuses largely on usability, so changes made to *glue*'s interface will be documented and explained in formal publications.

12. Adaptability to new technologies and requirements [SI²]

The *glue* package is written in a very modular way, designed to facilitate constant modernization. So, as technologies, especially visualization libraries, evolve, we have and will continue to respond to new opportunities by evolving *glue*. The 3D viewers now in *glue* are based on a Python library (*VisPy*) that did not exist when the project was started. Our planned development of browser-based viewers (Section 6.2) is essentially an initiative on our part to adapt to the fact that significant numbers of researchers (via Jupyter) and educators (via WWT and CODAP) have started to move more of their work to the browser.

Similarly, the data abstraction layer described in Section 6 will allow us to adapt to new data access technologies and requirements without modifying the main *glue* application. In fact, the only change that would require a complete redesign of *glue* will be a move away from Python by the scientific community. Given that the popularity of Python is still on the rise, we do not realistically expect this to happen within the next decade or even two. Eventually, though, a more popular language will supplant Python in science, and when that happens, the concepts developed and tested as part of *glue* will still be valuable, and can be reimplemented using new technologies.

13. Broader impacts

PI Goodman has had success in bringing modern scientific ideas and practices to the public through her work on WorldWide Telescope-related programs, including the NSF-funded WWT Ambassadors (WWTA) STEM outreach program, discussed under "Results from Prior Support," below.

Our *glue* team and the Concord Consortium (members of which Goodman has gotten to know thanks to discussions re:WWTA) share deep interest in promulgating the use of linked views in K12 Data Science curricula. So, as part of the work proposed here, we plan to work together to prototype a web-based data science curriculum based on the concepts embodied in glue, using the technologies offered by the CODAP and/or Jupyter platforms. As mentioned in Section 4, we plan to collaborate directly with CODAP's creator, William Finzer of Concord Consortium, on experiments to determine what

combination(s) of: "*glue* in CODAP" (via the *glue* API); "CODAP in *glue*" (via CODAP's API); and/or CODAP and *glue* in Jupyter notebooks/labs makes the most sense, in both research and educational contexts. We cannot sensibly carry out these experiments until Year 2 (see schedule in Section 8), by which time we will have implemented more *glue* web functionality via Jupyter variants. Thus, we are only asking for funding here to prototype a data science curriculum based on *glue* and CODAP. The planned collaboration will begin with discussion and experimentation in Year 2, and in Year 3, we will test rapid prototypes on volunteer high school students already involved in WWTA and Concord Consortium projects. One or more of the Co-op college students mentored by PI Borkin will participate in the prototyping and testing. Assuming all goes well, the goal of our collaboration by the end of Year 3 will be to propose a broader project aimed at honing, deploying, and testing a high-school-level data science curriculum to NSF (e.g. DRK12) and/or other foundations, based on results from the prototype testing. The goal of the proposed K12 curriculum would be to increase high-school students' ability to gain insight from data using linked-view data visualization in concert with basic statistics.

As described above, there are a wide range of data sets we can use as examples in prototyping, and part of our collaboration with the Concord Consortium will include seeing which data sets engage learners most. We expect that the three-dimensional aircraft-tracking data sets discussed in Section 4 will offer maximum engagement, but we will test the popularity and clarity of astronomy, geospatial and medical imaging examples as well.

The widespread use of glue across many fields of study already gives it "broad" impact, but we expect, given how *important but rarely-taught* data science skills are in current K12 curricula, that the educational work we will pilot as part of this SSE grant may have the longest-lasting broad impact.

(As some reviewers may notice on PI Goodman's bio, she is also very involved at present in creating modular content and interactive tools for the edX online learning platform. For those reviewers reference, the answer is "yes, it's likely that the Year 3 proposal would include piloting a fully online version of the glue-CODAP data science curriculum.")

14. Results from Prior NSF Support (Goodman)

Three inter-related recent projects where PI Alyssa Goodman is PI or Co-PI are: ***Thinking Spatially about the Universe–A Physical and Virtual Laboratory for Middle School Science***, DRL-1503395, PI Alyssa Goodman (Harvard), Co-PI Julia Plummer (Penn State), 7/1/15-6/30/18, $901,115; ***Building an Understanding of Astronomical Sizes and Scales with WorldWide Telescope***, DUE-1140440, PI Edwin Ladd (Bucknell), Co-PIs Alyssa Goodman (Harvard) & Katharyn Nottis (Bucknell), 9/1/12-8/31/17; $199,961; and ***EAGER: A Prototype WorldWide Telescope Visualization Lab Designed in the Web-based Inquiry Science Environment***, IIS-1254535, PI Alyssa Goodman, (Harvard) Co-PI Susan Sunbury (Smithsonian), 9/1/12-8/31/14, $149,658.00. Accomplishments from these projects are discussed below.

14.1 Intellectual Merit

The goal of the *Seamless Astronomy* group at the Harvard-Smithsonian Center for Astrophysics is to advance the use and development of software (e.g. *glue*) in accelerating the pace of research and in improving educational outcomes. The group is led by PI Goodman, who also founded the WorldWide Telescope Ambassadors (WWTA) Program (Goodman et al., 2012), which uses the WorldWide Telescope (WWT) software to improve STEM teaching and learning. WWTA was originally supported by Microsoft Research, with whom Goodman worked closely from 2009-15 in order to extend WWT's capabilities. Since 2012, WWTA has received NSF support through the projects listed above. Recent WWTA-team publications all present research designed to **optimize strategies for teaching concepts involving high-dimensional spatial reasoning**. In our work with 6th-8th grade students (Udomprasert et al., 2012, 2013, 2014), one key finding is that the *order* in which physical and virtual models of the Sun-Earth-Moon system are shown to students can be adjusted to effect learning outcomes, based on age and other

demographic factors. In our WWTA work at the college level (Gingrich et al., 2015; Nottis et al., 2015; Ladd et al., 2015, 2016), we have developed, tested, and published the "Size, Scale, and Structure Concept Inventory (S3CI) for Astronomy," which presents a *new, vetted, way to evaluate student understanding of size, scale, and structure concepts in the astronomical context.*

14.2 Broader Impacts

The curricular materials developed under the projects discussed above are all online at many well-curated websites. (A web search for "WorldWide Telescope Ambassadors" will give a sense of the work's penetration and availability). The free WWT software (enhanced by Goodman's work on WWTA, above) has wide distribution (~20 million downloads), can run in a web browser, and is available on GitHub. Several other large organizations beyond those participating in the NSF-funded research listed above are now also using the materials created. Downloads by teachers of the Moon Phases curriculum number in the thousands, and content from Goodman's group's WorldWide Telescope creations is often highlighted in mass media (including NOVA). WWTA's educational tour on extrasolar planets won the tour-making "Grand Prize" in at the 2017 American Astronomical Society meeting. And, as part of the NASA-sponsored "Bringing the Universe to America's Classrooms" program, Goodman's group has begun working with WGBH to distribute NASA content highlighted in WGBH productions using WWT, and WWTA curricula, online.

14. Results from Prior NSF Support (Borkin)

As a new Assistant Professor, Michelle Borkin has no prior PI or Co-PI grants to report, but her PhD work on "Flow Simulation with Visualization of Simulated and Experimental Data Applied to Biophysics and Astrophysics", was supported by an NSF Graduate Research Fellowship for $30,000 per twelve-month fellowship year, from 9/1/2010 - 9/1/2014. Fellow ID number 2010103380.

14.1 Intellectual Merit

The goal of Borkin's PhD dissertation research was to develop novel interdisciplinary visualization techniques for multidimensional data in the imaging sciences. The research resulted in novel 2D and 3D visualizations of biophysical and astronomical data (Arce et al., 2011, Borkin et al., 2011, Goodman et al., 2014, Lipşa et al., 2012), visualization techniques and tools for computer science provenance data (Borkin et al., 2013), and theory on the perception and cognition of visualizations (Borkin et al., 2013, 2016).

14.2 Broader Impacts

The results of this highly interdisciplinary research have been applied in subsequent research projects, and research papers associated with this fellowship have high citation counts. The novel results have also been disseminated to the public through presentations at scientific research conferences as well as through courses at Harvard University. The data and code developed as part of the perception and cognition research is available through the MassVis project website.

References

Arce, H. G., Borkin, M. A., Goodman, A. A., Pineda, J. E., and Beaumont, C. N., 2011. A bubbling nearby molecular cloud: COMPLETE shells in Perseus. The Astrophysical Journal 742, no. 2 (2011): 105.

The Astropy Collaboration: Robitaille, T.P., Tollerud, E.J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A.M., Kerzendorf, W.E. and Conley, A., 2013. Astropy: A community Python package for astronomy. Astronomy & Astrophysics, 558, A33 pp. 1-9

Beaumont, C.N., Goodman, A.A., Kendrew, S., Williams, J.P. and Simpson, R., 2014. The Milky Way Project: Leveraging citizen science and machine learning to detect interstellar bubbles. The Astrophysical Journal Supplement Series, 214:3 pp. 1-18

Borkin, M., Gajos, K., Peters, A., Mitsouras, D., Melchionna, S., Rybicki, F., Feldman, C., & Pfister, H., 2011. Evaluation of Artery Visualizations for Heart Disease Diagnosis. IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2011), 17, 12, 2479-2488.

Borkin, M., Vo, A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., & Pfister, H., 2013. What Makes a Visualization Memorable? IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2013), 19, 12, 2306-2315.

Borkin, M., Bylinskii, Z., Kim, N., Bainbridge, C., Yeh, C., Borkin, D., Pfister, H., & Oliva, A., 2016. Beyond Memorability: Visualization Recognition and Recall. IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2015), 22, 1, 519-528.

Borkin, M., Robitaille, T., Beaumont, C., Xuran Qian, P., Munzner, T., Goodman, A., 2017. Glue: A Linked Data Visual Exploration Tool. IEEE Transactions on Visualization and Computer Graphics, in-prep for submission.

Cui, Q., Ward, M.O., Rundensteiner, E.A. and Yang, J., 2006. Measuring data abstraction quality in multiresolution visualizations. Visualization and Computer Graphics, IEEE Transactions on, 12(5), pp. 709-716.

Ellis, G., 2008. Random Sampling as a Clutter Reduction Technique to Facilitate Interactive Visualisation of Large Datasets (Doctoral dissertation, Lancaster University).

Fekete, J.D. and Plaisant, C., 2002. Interactive information visualization of a million items. In IEEE Symposium on Information Visualization, 2002. INFOVIS 2002, pp. 117-124.

Gingrich, E.C., Ladd, E.F., Nottis, K.E.K., Udomprasert, P. and Goodman, A.A., 2015, November. The Size, Scale and Structure Concept Inventory (S3CI) for Astronomy. In Astronomical Society of the Pacific Conference Series, Vol. 500, p. 269.

Goodman, A., Fay, J., Muench, A., Pepe, A., Udomprasert, P., Wong, C. 2012. WorldWide Telescope in Research and Education, in Astronomical Data Analysis Software and Systems XXI, ASP Conference Proceedings Vol 461. Edited by P. Ballester, D. Egret, and N.P.F. Lorente.

Goodman, A.A., 2012. Principles of high-dimensional data visualization in astronomy. Astronomische Nachrichten, Vol.333, Issue 5-6, 333, pp. 505-514. doi:10.1002/asna.201211705

Goodman, A. A. , Peek, J. , Accomazzi, A., Beaumont, C., Borgman, C.L., Chen H., Crosas M., Erdmann C., Muench A. , Pepe A., Wong, C., 2014, The "Paper" of the Future, Authora article id. 8762

Goodman, A., Alves, J., Beaumont, C., Benjamin, R., Borkin, M., Burkert, A., Dame, T., Jackson, J., Kauffmann, J., Robitaille, T., & Smith, R., 2014. The Bones of the Milky Way. Astrophysical Journal, 797, 53.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Computing in science and engineering, 9(3), pp.90-95.

Joye, W.A. and Mandel, E., 2003. New features of SAOImage DS9. In Astronomical data analysis software and systems XII, Vol. 295, pp. 489-492.

Joye, W.A., 2006. New features of SAOImage DS9. In Astronomical Data Analysis Software and Systems XV, Vol. 351, pp. 574-576.

Ladd, E.F., Gingrich, E.C., Nottis, K.E.K., Udomprasert, P. and Goodman, A.A., 2015, November. Combining Real World Experiences with WorldWide Telescope Visualization to Build a Better Parallax Lab. In Astronomical Society of the Pacific Conference Series, Vol. 500, p. 191

Ladd, E.F., Udomprasert, P., Nottis, K.E.K., and Goodman, A. A. 2016. Building a Three-Dimensional Universe from the Classroom: Multiperspective Visualization for Non-Science Undergraduates. Education and New Developments Conference, June 2016, Ljubljana, Slovenia.

Lipşa, D., Laramee, R. S., Cox, S. J., Roberts, J. C., Walker, R., Borkin, M., & Pfister, H., 2012. Visualization for the Physical Sciences. Computer Graphics Forum, 31, 2317.

Naiman, J.P., 2016. AstroBlend: An astrophysical visualization package for Blender. Astronomy and Computing, 15, pp. 50-60.

North, C. and Shneiderman, B., 2000. Snap-together visualization: can users construct and operate coordinated visualizations?. International Journal of Human-Computer Studies, 53(5), pp. 715-739.

Nottis, K.E.K., Ladd, E.N., Goodman, A. and Udomprasert, P., 2015. Initial Development of a Concept Inventory to Assess Size, Scale, and Structure in Introductory Astronomy. US-China Education Review, 5(11), pp. 689-700.

Pérez, F. and Granger, B.E., 2007. IPython: a system for interactive scientific computing. Computing in Science & Engineering, 9(3), pp.21-29.

Roosendaal, T. and Selleri, S., 2004. The Official Blender 2.3 guide: free 3D creation suite for modeling, animation, and rendering. No Starch Press.

Borkin, M., Yeh, C., Boyd, M., Macko, P., Gajos, K., Seltzer, M., & Pfister, H., 2013, Evaluation of Filesystem Provenance Visualization Tools. IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2013), 19, 12, 2476-2485.

Shneiderman, B., 2008, June. Extreme visualization: squeezing a billion records into a million pixels. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 3-12). ACM.

Snow, J., 1855. On the mode of communication of cholera. John Churchill.

Stolte, C., Tang, D. and Hanrahan, P., 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. IEEE Transactions on Visualization and Computer Graphics, 8(1), pp.52-65.

Taylor, M.B., 2005, December. TOPCAT & STIL: starlink table/VOTable processing software. In Astronomical Data Analysis Software and Systems XIV, Vol. 347, pp. 29-33.

Taylor, M.B., Boch, T. and Taylor, J., 2015. SAMP, the Simple Application Messaging Protocol: Letting applications talk to each other. Astronomy and Computing, 11, pp.81-90.

Tukey, J.W.: 1977, Exploratory Data Analysis, Addison-Wesley, Reading, p. 688

Turk, M.J., Smith, B.D., Oishi, J.S., Skory, S., Skillman, S.W., Abel, T. and Norman, M.L., 2010. yt: A multi-code analysis toolkit for astrophysical simulation data. The Astrophysical Journal Supplement Series, 192(1), pp. 9-16.

Tyson, J.A., 2002, December. Large synoptic survey telescope: overview. InAstronomical Telescopes and Instrumentation (pp. 10-20). International Society for Optics and Photonics.

Udomprasert, P.S., Goodman, A.A. and Wong, C., 2012, August. WWT Ambassadors: WorldWide Telescope for Interactive Learning. In Connecting People to Science: A National Conference on Science Education and Public Outreach, Vol. 457, pp. 149-154.

Udomprasert, P.S., Goodman, A.A. and Wong, C., 2013, January. WorldWide Telescope Ambassadors: A Year 3 Update. In Communicating Science: A National Conference on Science Education and Public Outreach, Vol. 473, p. 137.

Udomprasert, P., Goodman, A., Sunbury, S., Zhang, Z.H., Sadler, P., Dussault, M., Block, S., Lotridge, E., Jackson, J. and Constantin, A.M., 2014, July. Visualizing Moon Phases with WorldWide Telescope. In Astronomical Society of the Pacific Conference Series, Vol. 483, p. 297.

Van Der Walt, S., Colbert, S.C. and Varoquaux, G., 2011. The NumPy array: a structure for efficient numerical computation. Computing in Science & Engineering, 13(2), pp.22-30.

Wampler, S. and Goodrich, B., 2009, September. A scalable data handling system for atst. In Astronomical Data Analysis Software and Systems XVIII, Vol. 411, pp. 527-530.

Xuran Qian, P., Goodman, A., Robitaille, T., Xu Cai, M., and Michelle A. Borkin, 2017. 3D Linked Visualization for High-dimensional Data in Glue. IEEE Transactions on Visualization and Computer Graphics, in-prep for submission.

Zucker, C., Battersby, C. and Goodman, A., 2015. The Skeleton of the Milky Way. The Astrophysical Journal, 815(1), p.23.

Biographical Sketch for Alyssa A. Goodman, February 2017

Professional Preparation

Massachusetts Institute of Technology, Cambridge, MA; ScB in Physics, 1984
Harvard University, Cambridge, MA; AM in Physics, 1986 and PhD in Physics, 1989
University of California at Berkeley, Berkeley, CA; President's Fellowship, 1989-92

Appointments

2017-	Co-Director for Science, Radcliffe Institute for Advanced Study (2017-19 term)
2015–	Robert Wheeler Willson Professor of Applied Astronomy, Harvard University
2016–2017	Edward, Frances, and Shirley B. Daniels Fellow, Radcliffe Institute
1999–2015	Professor of Astronomy, Harvard University
1995–	Research Associate, Smithsonian Astrophysical Observatory
2008–2011	Scholar-in-Residence, WGBH Boston (sabbatical & *pro bono* consulting)
2008–2010	Core Member, Harvard Initiative in Innovative Computing
2005–2008	Founding Director, Harvard Initiative in Innovative Computing
2001–2002	Visiting Fellow, Yale University (sabbatical)
1996–1999	Associate Professor of Astronomy, Harvard University
1995–1997	Head Tutor, Harvard University Astronomy Department
1992–1996	Assistant Professor of Astronomy, Harvard University
1989–1992	President's Fellow, University of California, Berkeley

Products *(*=most closely related to this proposal)*

*Goodman, A. A. 2012, *Principles of High-Dimensional Data Visualization in Astronomy*, Astronomische Nachrichten, Vol.333, Issue 5-6, p. 505-514 (also arXiv:1205.4747), dx.doi.org/10.1002/asna.201211705

Goodman, A.A., Alves, J., Beaumont, C.N., Benjamin, R.A., Borkin, M.A., Burkert, A., Dame, T.M., Jackson, J., Kauffmann, J., Robitaille, T., Smith, R.J. 2014, *The Bones of the Milky Way*. Astrophysical Journal, 797, 53. doi:10.1088/0004-637X/794/1/1

*Goodman, A. A., Pepe, A., Blocker, A., Borgman, C. L., Cranmer, K., Crosas, M., DiStefano, R., Gil, Y., Groth, P., Hedstrom, M., Hogg, D., Kashyap, V., Mahabal, A., Siemiginowska, A. and Slakovic, A. 2014, *10 Simple Rules for the Care and Feeding of Scientific Data,* PLoS Comp Biol, dx.doi.org/10.1371/journal.pcbi.1003542

*Goodman, A. & Wong, C. 2015, *WorldWide Telescope.* Sky & Telescope April 2015, http://www.shopatsky.com/sky-telescope-april-2015-digital-issue

Pepe, A., Goodman, A., Muench, A., Crosas, M., Erdmann, C. 2014 *How Do Astronomers Share Data? Reliability and Persistence of Datasets Linked in AAS Publications and a Qualitative Study of Data Practices among US Astronomers*. PLoS ONE. 9, 4798. dx.doi.org/10.1371/journal.pone.0104798

Rice, T.S., Goodman, A.A., Bergin, E.A., Beaumont, C., Dame, T.M., 2016, *A UNIFORM CATALOG OF MOLECULAR CLOUDS IN THE MILKY WAY,* Astrophysical Journal, 822, 52. (also arXiv:160202791) doi:10.3847/0004-637X/822/1/52

*Robitaille, T, Beaumont, C., Qian, P., Borkin, M., & Goodman, A. (2017). glueviz v0.10: multidimensional data exploration [Data set]. Zenodo. http://doi.org/10.5281/zenodo.293197

*Sanders, N. E., Faesi, C. & Goodman, A. A. 2013, *A New Approach to Developing Interactive Software Modules through Graduate Education*, Journal of Science Education and Technology, doi:10.1007/s10956-013-9474-4

Zook, M…Goodman, A., et al. 2017, *Ten Simple Rules for Responsible Big Data Research,* PLoS Comp Biol, in press.

Zucker, C., Battersby, C., and Goodman, A.A. 2015, *The Skeleton of the Milky Way*, Astrophysical Journal, 815, 23. doi:10.1088/0004-637X/815/1/23

Synergistic Activities

Initiative in Innovative Computing (IIC) & Seamless Astronomy In AG's IIC Directorship, she built a new institution at Harvard, hosting dozens of researchers, whose mission was to address and answer scientific questions that are unanswerable without bringing domain scientists and computer scientists into closer collaboration. IIC [iic.harvard.edu] efforts brought a host of new courses, scholars, and events to Harvard, many of which/whom have now become part of the Institute for Advanced Computational Science [iacs.seas.harvard.edu]). Since the IIC work, AG has served on several related committees, including the National Academy's Board on Research Data and Information, while also leading the "Seamless Astronomy" group at the Center for Astrophysics [projects.iq.harvard.edu/seamlessastronomy]. Seamless Astronomy initiatives (including *ADS Labs*, the *Astronomy Dataverse*, *Authorea*, and the *glue* visualization package) are all aimed at streamlining the research process. Goodman is presently serving on the Steering Committee of the new pan-Harvard *Data Science Initiative*.

Scientific Visualization For the past decade, AG has taught data visualization in her course known as the *Art of Numbers* [artofnumbers.org], and in graduate data visualization workshops as well. AG founded the *Astronomical Medicine* effort at the IIC, which created now-deployed software tools for visualizing three-dimensional astronomy data using tools developed for medical research. This work also led to the first 3D PDF to be published in *Nature*. AG gives many invited presentations on visualization, including as a keynote speaker at Edward Tufte's "Advanced Visualization" workshop in 2015. AG led the efforts that initiated the open-source multi-dimensional data-exploration software project now known as glue [glueviz.org], and is the PI of the NASA funding for that project.

WorldWide Telescope/WWT Ambassadors From 2007-2015, AG served as the lead astronomical consultant/collaborator in the creation of the WorldWide Telescope (WWT) program from Microsoft Research. In 2015, she helped WWT transition to an Open Source effort under the auspices of the AAS, and she now helps AAS staff integrate WWT into Research, Publication and Teaching projects in the US and beyond. In 2008, while on sabbatical at WGBH, AG helped create, and starred in, an episode of *"Fetch! with Ruff Ruffman,"* which explained multi-wavelength observations, using WWT, and was nominated for an Emmy. In 2009, AG founded the WorldWide Telescope Ambassadors program [wwtambassadors.org], which recruits and trains volunteers to use WWT in educational settings worldwide.

Online Learning AG has taken a leadership role in the creation and deployment of interactive online learning modules. In 2013 all of the graduate students in AG's Harvard graduate-level class on the Interstellar Medium created (15) interactive online tools, released (including on edX) for use by the public, and described in Sanders, Faesi & Goodman, 2013 (see Products, above). AG was the local host for the 2013 dotAstronomy5 [dotastronomy.com/events/five/] meeting that brought together a worldwide community of researchers, teachers, and software developers interested in using the web more effectively for astronomy research and education. AG is currently leading the creation of a new, highly modular, online edX course about the how humanity strives to predict its own future, called PredictionX.

Social Media and Talks AG [@aagie] tweets about data science, astronomy, education, and data visualization, and she maintains several online sites about similar topics [e.g. alyssagoodman.tumblr.com]. AG serves as the principal faculty advisor to the Communicating Science Conferences [comscicon.com], which train graduate students to improve science communication through many channels, including online. AG is a popular public speaker, and all of her presentations are online at tinyurl.com/AGtalks.

Biographical Sketch: Thomas P. Robitaille

(a) Professional Preparation

University of St Andrews, St Andrews, United Kingdom, MPhys Astrophysics, 2005

University of St Andrews, St Andrews, United Kingdom, PhD Astrophysics, 2008

Harvard College Observatory, Cambridge, MA, NASA Spitzer Postdoctoral Fellowship

(b) Appointments

2015 - now: Scientific software development consultant

2011 - 2015: Max Planck Research Group Leader

Max Planck Institut für Astronomie, Heidelberg, Germany

(c) Products (* = most closely related to the proposal)

* Robitaille, Thomas; Beaumont, Chris; Qian, Penny; Borkin, Michelle & Goodman, Alyssa. (2017), glueviz v0.10: multidimensional data exploration [software]. Zenodo. http://doi.org/10.5281/zenodo.293197

* The Astropy Collaboration: Robitaille, Thomas; Tollerud, Erik J.; Greenfield, Perry; Droettboom, Michael; Bray, Erik; Aldcroft, Tom; Davis, Matt; Ginsburg, Adam; Price-Whelan, Adrian M.; Kerzendorf, Wolfgang E.; Conley, Alexander; Crighton, Neil; Barbary, Kyle; Muna, Demitri; Ferguson, Henry; Grollier, Frédéric; Parikh, Madhura M.; Nair, Prasanth H.; Unther, Hans M.; Deil, Christoph; Woillez, Julien; Conseil, Simon; Kramer, Roban; Turner, James E. H.; Singer, Leo; Fox, Ryan; Weaver, Benjamin A.; Zabalza, Victor; Edwards, Zachary I.; Azalee Bostroem, K.; Burke, D. J.; Casey, Andrew R.; Crawford, Steven M.; Dencheva, Nadia; Ely, Justin; Jenness, Tim; Labrie, Kathleen; Lim, Pey Lian; Pierfederici, Francesco; Pontzen, Andrew; Ptak, Andy; Refsdal, Brian; Servillat, Mathieu; Streicher, Ole (2013), Astropy: A community Python package for astronomy, Astronomy & Astrophysics, Volume 558, A33, http://dx.doi.org/10.1051/0004-6361/201322068

* Robitaille, Thomas; Cruz, Kelle; Greenfield, Perry; Jeschke, Eric; Juric, Mario; Mumford, Stuart; Prescod-Weinstein, Chanda; Sosey, Megan; Tollerud, Erik; VanderPlas, Jake; Ford, Jes; Foreman-Mackey, Dan; Jenness, Tim; Aldcroft, Tom; Alexandersen, Mike; Bannister, Michele; Barbary, Kyle; Barentsen, Geert; Bennett, Samuel; Boquien, Médéric; Campos Rozo, Jose Ivan; Christe, Steven; Corrales, Lia; Craig, Matthew; Deil, Christoph; Dencheva, Nadia; Donath, Axel; Douglas, Stephanie; Ferreira, Leonardo; Ginsburg, Adam; Goldbaum, Nathan; Gordon, Karl; Hearin, Andrew; Hummels, Cameron; Huppenkothen, Daniela; Jennings, Elise; King, Johannes; Lawler, Samantha; Leonard, Andrew; Lim, Pey Lian; McBride, Lisa; Morris, Brett; Nunez, Carolina; Owen, Russell; Parejko, John; Patel, Ekta; Price-Whelan, Adrian; Ruggiero, Rafael; Sipocz, Brigitta; Stevens, Abigail; Turner, James; Tuttle, Sarah; Yanchulova Merica-Jones, Petia; Yoachim, Peter (2016), Python in Astronomy 2016 Unproceedings, Zenodo. http://doi.org/10.5281/zenodo.56793

* Robitaille, Thomas; Barmby, Pauline; Mumford, Stuart; Jeschke, Eric; Persson, Magnus; Kendrew, Sarah; Aldcroft, Tom; Archibald, Anne; Barbary, Kyle; Barentsen, Geert; Beckmann,

Ricarda; Bray, Erik; Craig, Matthew; Crawford, Steve; Crighton, Neil; Cruz, Kelle; Deil, Christoph; Dencheva, Nadia; Droettboom, Mike; Geers, Vincent; Ginsburg, Adam; Gomez, Haley; Gomez, Edward; Greenfield, Perry; Hagen, Alex; Hayes, Laura; Ishida, Emille; Karr, Jennifer; Kerzendorf, Wolfgang; Koepferl, Christine; Leyder, Jean-Christophe; Martin-Carrillo, Antonio; McCully, Curtis; Mechtley, Matt; Moolekamp, Fred; Moorhead, Althea; Moss, Vanessa; Perez-Suarez, David; Ryan, Daniel; Saroff, David; Servillat, Mathieu; Sipocz, Brigitta; Smethurst, Rebecca; Smith, Britton; Sosey, Megan; Stevens, Abigail; Terrón, Victor; Teuben, Peter; Tollerud, Erik; Turner, James; van Elteren, Arjen; Villaume, Alexa; Villforth, Carolin; Zuntz, Joe (2015), Python in Astronomy 2015 Unproceedings. Zenodo. http://doi.org/10.5281/zenodo.19330

* Goodman, A.A., Alves, J., Beaumont, C.N., Benjamin, R.A., Borkin, M.A., Burkert, A., Dame, T.M., Jackson, J., Kauffmann, J., Robitaille, T., Smith, R.J. 2014, The Bones of the Milky Way. Astrophysical Journal, 797, 53. doi:10.1088/0004-637X/794/1/1

Robitaille, T. P. (2011), HYPERION: an open-source parallelized three-dimensional dust continuum radiative transfer code, Astronomy & Astrophysics, Volume 536, A79, 17

(d) Synergystic activities

Leadership of Open Source Projects: Robitaille is one of the coordinators and lead developers for the Astropy package and has been since the start of the project in 2011. Astropy is a project to create a core package for Astronomy and encourage interoperability of more specialized packages. Almost 200 individuals have contributed to this project so far, and the project includes between 10 and 20 core developers at any one time. This software is being used by major observatories in Astronomy, including for the data processing for the upcoming James Webb Space Telescope.

Development of Open Source projects: Robitaille has created a number of open source packages, including Hyperion (a radiative transfer code for astronomy), reproject (an image reprojection/regridding package), and dozens of other small Python packages. In addition, he has contributed to many collaborative open source projects in the Scientific Python ecosystem, such as Astropy, Glue, Numpy, SciPy, and Matplotlib.

Scientific Editorial work: in addition to development effort, Robitaille currently works as a Scientific Editor for software papers for the American Astronomical Society, and was one of the co-authors of a new policy that now allows authors to submit papers describing software (without necessarily having novel scientific results).

Conference organization: Robitaille has chaired the scientific organization of a number of scientific conferences. Most notably, he started a new conference series entitled Python in Astronomy, the two first installments of which took place in Leiden (2015) and Seattle (2016), and the upcoming meeting in 2017 taking place in Leiden. The aim of this new conference series is to bring together researchers, developers, educators and users that have a common interest in using Python for Astronomy research, and get them to share knowledge and work together on projects during the conference (in which formal talks are only a small fraction). This conference series has been extremely successful, with typical over-subscription rates of 4:1 or more.

MICHELLE A. BORKIN

Northeastern University
ATTN: Michelle Borkin, 202 WVH
360 Huntington Avenue
Boston, MA 02115-5000

617.373.6355 [office]
m.borkin@northeastern.edu
http://www.ccs.neu.edu/home/borkin/

PROFESSIONAL PREPARATION

- Harvard College, Cambridge, MA, Astronomy and Astrophysics & Physics, B.A., 2006.
- Harvard University, Cambridge, MA, Applied Physics, M.S., 2011.
- Harvard University, Cambridge, MA, Applied Physics, Ph.D., 2014.
- University of British Columbia, Vancouver, BC, Canada, Computer Science, Postdoctoral Research Fellow, 2014–2015.

APPOINTMENTS

2015-present Assistant Professor, College of Computer and Information Science, Northeastern University

RELATED PRODUCTS

- Chris Beaumont, Thomas Robitaille, Michelle Borkin, & Alyssa Goodman. 2014. **glueviz v0.4: multidimensional data exploration**. http://www.glueviz.org/ DOI: 10.5281/zenodo.13866 http://dx.doi.org/10.5281/zenodo.13866 .
- Alyssa A. Goodman, João Alves, Christopher N. Beaumont, Robert A. Benjamin, Michelle A. Borkin, Andreas Burkert, Thomas M. Dame, James Jackson, Jens Kauffmann, Thomas Robitaille, & Rowan J. Smith. **"The Bones of the Milky Way"**, Astrophysical Journal, 797, 53-66, 2014.
- Chris Beaumont, Thomas Robitaille, Alyssa Goodman, & Michelle Borkin. **"Multidimensional Data Exploration with Glue"**, Proceedings of the 12th Python in Science Conference, 8-12, 2013.
- Michelle A. Borkin, Chelsea S. Yeh, Madelaine Boyd, Peter Macko, Krzysztof Z. Gajos, Margo Seltzer, & Hanspeter Pfister. **"Evaluation of Filesystem Provenance Visualization Tools"**, IEEE Transactions on Visualization and Computer Graphics, 19, 12, 2476-2485, 2013. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6634189
- Dan R. Lipşa, Robert S. Laramee, Simon J. Cox, Jonathan C. Roberts, Rick Walker, Michelle A. Borkin, & Hanspeter Pfister. **"Visualization for the Physical Sciences"**, Computer Graphics Forum, 31, 2317-2347, 2012. http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2012.03184.x/full

OTHER SIGNIFICANT PRODUCTS

- Michelle A. Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S. Yeh, Daniel Borkin, Hanspeter Pfister, & Aude Oliva. **"Beyond Memorability: Visualization Recognition and Recall"**, IEEE Transactions on Visualization and Computer Graphics, 22, 1, 519-528, 2016.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7192646
- Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, & Hanspeter Pfister. **"What Makes a Visualization Memorable?"**, IEEE Transactions on Visualization and Computer Graphics, 19, 12, 2306-2315, 2013.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6634103
- Michelle A. Borkin, Krzysztof Z. Gajos, Amanda Peters, Dimitrios Mitsouras, Simone Melchionna, Frank J. Rybicki, Charles L. Feldman, & Hanspeter Pfister. **"Evaluation of Artery Visualizations for Heart Disease Diagnosis"**, IEEE Transactions on Visualization and Computer Graphics, 17, 12, 2479-2488, 2011.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6065015&tag=1
- Alyssa A. Goodman, Erik W. Rosolowsky, Michelle A. Borkin, Jonathan B. Foster, Michael Halle, Jens Kauffmann, & Jaime E. Pineda. **"A role for self-gravity at multiple length scales in the process of star formation"**, Nature, 457, 63-66, 2009.
  http://www.nature.com/nature/journal/v457/n7225/full/nature07609.html
- Daniel Wigdor, Hao Jiang, Clifton Forlines, Michelle Borkin, & Chia Shen. **"The WeSpace: The Design, Development, and Deployment of a Walk-Up and Share Multi-Surface Visual Collaboration System"**, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 1237-1246, 2009.
  http://dl.acm.org/citation.cfm?id=1518886

SYNERGISTIC ACTIVITIES

- Guest speaker and faculty supporter of NUWIT (Northeastern Women in Technology).
- WISTEM (Women in Science, Technology, Engineering, and Mathematics) mentor through the Harvard College Women's Center, 2009-2014.
- Star Family Prize for Excellence in Advising at Harvard (Nominee, 2013)
- Recipient of a National Defense Science and Engineering Graduate (NDSEG) Fellowhip 2010, and a National Science Foundation (NSF) Graduate Research Fellowship 2010.
- Paper reviewer for the proceedings of IEEE Information Visualization (InfoVis), EuroVis, and ACM SIGCHI, as well as IEEE Transactions on Visualization and Computer Graphics.