Elements: Improving tools based on data-description standards for gigabyte-scale data sets

April 4, 2019

Amy Roberts

University of Colorado Denver

Project Summary

Elements: Improving tools based on data-description standards for gigabyte-scale data sets

Overview: The PI proposes to improve a set of data-analysis tools that require a description of the data format - rather than requiring a particular data format - to handle the gigabyte-scale datasets common in nuclear physics. The long-term goal of this work is to increase the accessibility of science analysis for both active researchers and science students.

In pursuit of this goal, this proposal has three objectives: (1) Improve existing software that provides access to data in any user-described format so that it works well with gigabyte-scale datasets. (2) Build community adoption and support of these tools. (3) Increase access to analysis of scientific data sets for undergraduate students, early-career researchers, and scientists who do not have access to dedicated software support. These objectives will be met as follows:

(1) Improving standards-based analysis software: Scientists who wish to analyze custom-format data can already do so with existing open-source software, e.g. kaitai-struct. However, this software has poor performance for files larger than a gigabyte, limiting its usefulness to the nuclear physics community. The PI proposes to improve this software to allow responsive analysis of gigabyte-scale datasets in python. This work is feasible thanks to libraries built and maintained by the IRIS-HEP collaboration that provide an intuitive interface to data structures optimized for rapid access to large, event-based data sets.

(2) Building community: The PI proposes yearly workshops to bring together developers and scientists for training, user feedback, and development of this software.

(3) Increasing Access: To fully participate in nuclear physics research, students must have extensive scientific computing skills. Mentor networks help some students through this maze but are not equally available. The PI requests funding for undergraduate students to develop and test documentation and training materials for the standards-based analysis tools and for fundamental scientific computing concepts.

Intellectual merit: This work supports the science effort of the Super Cryogenic Dark Matter Search, an experiment that seeks to better understand the nature of dark matter. This work supports additional high-priority science in the nuclear physics community such as the origin of the elements in the universe, fundamental symmetry testing, and radiation monitoring for national security.

Broader impacts: The PI proposes this work because it would be immediately useful to her own work on dark matter and because software that can provide easy access to any user-described data format would meet an urgent need of communities dealing with gigabyte-scale data. The PI requests funds to develop and field-test documentation for this software to maximize its usefulness to the community.

Right now, the necessity for local development of analysis code restricts participation in science analysis. With a small set of well-documented community tools based on data-description standards, along with open-access material introducing the computing concepts required to use those tools, the PI hopes to increase research access to a much wider group of individuals and increase the time expert scientists can spend doing science.

Contents

1 Overview

The PI proposes to improve existing tools that support analysis for any format of data through a data-description language. The existing tools work well for kilobyte-scale data sets but are inconveniently slow for gigabyte-scale data.

The PI also proposes work intended to develop a community of users and contributors for this general-purpose data-access software. This includes investing in user and developer documentation, building example analyses, and implementing integration tests to make it easier for the community to trust and contribute to the software. The PI also proposes yearly workshops designed to bring scientists and developers together to work on analysis.

1.1 User Stories

S has a data set that would be valuable to include in an analysis that combines different types of carbon scattering data. Because the data was taken with a polarized beam, it would provide a valuable constraint to her global fit. But she can't include it because the only code that can read the data won't compile on her machine. She's not familiar enough with Fortran to fix it and the original author died a few years ago. She could still write a paper, but the results aren't all that interesting without this data.

P would like to add an additional detector to his setup – this would allow him to set a stronger limit on a reaction rate that effects how stars create heavy elements. A colleague lent him a digitizer to instrument his extra detector, but the analysis code he has is for CAEN instruments and the loaner digitizer is from XIA. After burning a weekend trying and failing to adapt his code to handle the XIA data, he decides to try the experiment without the extra detector. He might get a useful constraint on the integrated cross section, and maybe if the funding for his postdoc comes through they'll have time to sort out the code.

All Q wants to do is look at the data from a detector used to search for dark matter. He got one of the files onto his computer, but when he opened it in Word it looked like a bunch of garbage symbols. Three frustrating days later, he now knows the data is "binary" and he's downloaded "source code" that should allow him to look at the data. When none of the commands in the readme file make sense, he asks his professor for help, who tells him to talk to the postdoc, who apparently doesn't respond to email. It takes almost an entire summer, but Q does eventually figure out how to compile the code, run the code, and look at a single event.

Each of these stories features a scientist trying to access data in different, binary formats. And in each case, software that provides access to their data already exists - but the usability of that software is poor enough to create a significant barrier to science. In each of these cases, the problems the researchers face are solvable, for someone expert in scientific software. But such experts are relatively rare, and many scientists have to solve their data-access problems themselves.

1.2 Deliverables

The PI proposes to improve the scientific-software ecosystem to lower the barrier to access custom, binary-format data by developing tools that provide access to that data based on a user-provided description. Scientists with custom-format data could then - by providing a description of that data - gain access to analysis tools that are supported by a broader community.

The goal of the proposed work is to significantly increase the accessibility of scientific data analysis on custom-format binary data. To accomplish this goal, the proposed work will produce the following deliverables:

Data-access software that provides convenient and quick access to gigabyte-scale datasets in any binary format. This software will require the user to provide a description of the format.

User documentation and examples that allow scientists to easily use the data-access software for their own analyses.

Training materials for foundational computing skills to ensure that individuals with little to no background in scientific computing can successfully work through an analysis tutorial and make progress towards their own analysis.

Contribution documentation, tests, and interface documentation that makes it easy for the community to contribute to the project development and documentation.

A fledgling community focused on creating, documenting, and supporting common tools that are useful for the entire community.

The software and related documentation will all be developed openly on gitlab. Installation instructions for both end-users and developers will be included and heavily tested and will include instructions for users without administration privileges so that scientists working on computing clusters are supported.

Releases of the software will be granted a DOI through Zenodo; all publications will be released on preprint servers.

Community development will be fostered through yearly workshops, heavy prioritization of documentation development and testing, and the support of an online forum for community discussion.

1.3 Broader Impacts

The fundamental goal of the proposed work is to make data easily accessible to individuals who want to answer science questions.

The immediate target audience for this work are active scientists, early-career researchers, and undergraduate students who are participating in research that require access of data for which there is not a good program. This software will reduce the time the community spends on software development.

The intent of this work is also to make access to science more equitable.

- High quality documentation makes the software easier to use for scientists with all levels of computing backgrounds
- Contribution documentation and tutorials increase the pool of contributors
- An online resource that provides findable materials on basic concepts in scientific computing makes it possible for novices to use the software

These aspects of the work increase science access in every case to individuals without existing background in computing. They also increase science access to access to limited lab or institutional resources:

- Researches at undergraduate-only institutions often struggle to train undergraduates to work in a custom computing environment unless they have funding for a postdoc
- Lab staff and a pool of experienced graduate students and postdocs can help with undergraduate and graduate student training, but these resources are not available at many institutions.

- Even at institutions with significant resources for training and mentorship, access to these reasons is not always equitable.

Large numbers of students are getting their education at community colleges and four year institutions. Complex, poorly-documented software infrastructure that gates scientific work is an unnecessary burden on faculty at these institutions and poses a higher barrier to students because they are less likely to have access to informal computing support structure than their peers at research institutions.

The proposed work will create the kernel of an ecosystem that is usable, well-documented for both novices and experts, and enjoys the support of a community that is accessible online. The PI believes that this infrastructure is a minimum requirement for equitable access to science analysis.

Undergraduate Education The Physics Department at CU Denver maintains an undergraduate-only program committed to providing students with hands-on research experience at this urban campus. The Department operates in close collaboration with the Physics Department at Metropolitan State University of Denver, and students from both institutions participate in the CU Denver groupâĂŹs activities, broadening the reach of the research program beyond a single institution. Currently, there are seven undergraduates in the PI's group majoring in Physics, Chemistry, Mechanical Engineering, and Computer Science.

2 Intellectual Merit

The long-term goal of this work is to significantly increase access to science data in the nuclear physics community and other communities that work with gigabyte-scale, custom-format, binary data sets.

One way to solve this problem is to improve existing software that provides access to data based on a user-provided description of that data. Scientists can then gain access to analysis tools that are supported by a broader community by providing a description of their data in a standard format. The work the PI proposes consists of (1) software development, (2) extensive documentation development, and (3) yearly workshops focused on bringing together developers and scientists to work on science analysis. The PI proposes to integrate this work so that every year there is a release of usable software and documentation that can recieve extensive testing at the yearly workshop. The yearly workshops will be an opportunity for helping scientists use the tools for their analyses and will also offer an opportunity for the community to determine the roadmap for the upcoming year.

The proposed work is feasible becuase it builds off existing infrastructure. Multiple data-description language standards exist and have robust communities that provide support for describing data with their language and tools that allow access to data [**?**, **?**, **?**, **?**]. In addition, work done by DIANA/HEP has created flexible libraries that provide high-energy physicists tools that are compatible with the python scientific ecosystem [**?**, **?**]. The PI proposes to combine Kaitai Struct with the DIANA/HEP awkward-array library - this will bring the speed and convenience the awkward-array library enjoys with gigabyte-scale data sets to any scientist who describes their data per the Kaitai Struct rules. Both Kaitai Struct and awkward-array are specifically designed to be extensible and have extensive documentation for developers who wish to interface with their software.

Leveraging the considerable existing infrastructure makes it feasible to develop a science-ready analysis tool compatible with any data format within a year. This helps community engagement with this project, as within the three years of this grant three workshops could be held that bring scientists together with developers to work on science analysis. This provides invaluable testing and

an opportunity to involve the community in the continued development of the tool. The intended result is a community analysis tool that is usable, aligns well with the needs of the community, and enjoys broad adoption.

2.1 Impact

The example analyses will initially focus on current collaborators: XIA and SuperCDMS. Therefore the proposed work is expected to directly improve the time-to-analysis and accessibility for users of these data sets.

More broadly, the PI expects the following impacts on the communities that work with gigabyte-scale data sets:

- Reduce redundant time spent on writing and wrangling custom data-access software.
- Significantly increase the involvement of students in the science-analysis phase of research.
- Improve the reproducibility of science results.
- Increase the equity of access to scientific data analysis.

These impacts rely not just on the initial software product, but also on the usability of the documentation and the breadth of community adoption. The PI has therefore included work on documentation, stipends for students to test and improve the documentation, end-to-end testing, and community workshops in the plan of work.

2.2 Long-term research goals

The PI is a member of the Super Cryogenic Dark Matter Search (SuperCDMS) and serves as the data quality technical coordinator for the upcoming experiment at SNOLAB.

The PI, working together with the SuperCDMS anaysis coordinator and team leaders, has identified that a critical piece of the data quality monitoring will be the ability for beginning graduate students and postdocs to do preliminary science analysis of the data within a week of its collection. This requirement demands a substantial improvement in the analysis infrastructure.

The PI's group has, together with the computing infrastructure team at the Stanford Linear Accelerator, implemented a web-based analysis environment. This provides access to the SuperCDMS data and software with unprecedented ease - and there is interest in improving the existing analysis tools to make data analysis easier and better-documented for new collaboration members.

The proposed work would deliver software that could improve the analysis infrastructure and move SuperCDMS towards software that is supported by a broader community. Right now the complexity of the software is a significant limiting factor in SuperCDMS' ability to release data sets that might be of interest to high school and undergraduates; moving to community-supported software infrastructure is a necesesary prerequisite to sharing our data more broadly.

The proposed work provides potential benefit to SuperCDMS by prototyping an analysis environment. The PI's involvement in SuperCDMS significantly benefits the proposed work because she has access to gigabyte-scale data sets and event-based analyses that are at the high end of requirements for many experimental nuclear physics analyses. Thus developing prototype SuperCDMS analysis code is a nice proving ground for the target user and provides a nice complement to the smaller-data XIA analysis proposed as the other primary test analysis.

The PI also enjoys an active research program using SuperCDMS data to study the behavior of low-energy deposition in solids. The training required for students to be able to look at data can take several semesters. The PI sees building a variety of training materials as essential to increasing undergraduate access to dark matter science. And well-documented analysis software would immediately benefit students if that same documentation helped them with SuperCDMS data analysis. In addition, students who spend time developing and improving software libraries for community software will be doing work that has visibility for potential employers.

3 Plan of Work

The timing of the proposed work is driven by the proposed, yearly workshops that focus on (1) teaching scientists how to use the tools to access their data, (2) working with scientists to perform their analyses in the python environment, (3) identifying improvements needed for the software to be easy to learn and useful in analysis, and (4) bringing developers into close contact with the science community using their tools. Each workshop will result in an updated roadmap for the software.

Thus, the workshops - and software releases that include testing, documentation, and example analyses - are the primary milestones of the proposed work.

The work for each yearly cycle can be broken down into the following categories: development of basic computing skills learning material; development of the data-access library; planning and execution of the workshop; and a community-driven update of the roadmap. See Table **??** for details on who will perform this work.

The minimum requirements of the work determine the work plan and are the following:

1. If students or staff move on to other positions, their replacements should be able to get up to speed in a month or less.
2. Someone with no domain knowledge but reasonable persistence should be able to run the example analysis within a week.
3. Someone with no domain knowledge but reasonable persistence should be able to analyze their own data within a month.
4. A scientist who uses the access-data library to obtain a science result should know how to cite the software.
5. A scientist experiencing trouble using the software should be able to determine how to get help quickly (within five minutes of searching).
6. A scientist who wishes to improve the code should be able to quickly determine how to contact the developers and how to change, test, and push the code.

The scope of the proposed software is relatively modest: copy an existing framework and adapt it so that it stores data in awkward-array structures rather than slower, dictionary structures. The development and testing of this code will take time - but reference code for similar work exists, there is robust community support, and there is a developer guide that gives specific instructions for developers who wish to extend the existing Kaitai Struct code in this way. The proposed work is feasible because it connects two libraries that are both designed for this purpose.

The majority of the proposed work is in making this software easy to use for scientists, and making it easy for the community to participate in the direction and development of the software. This requires robust documentation for both users and developers. Users will require installation instructions, instructions for using the library, and guidance on how to adapt the examples for their own analysis needs. Developers will need additional documentation: instructions for changing the code and testing the code, and instructions and guidance on contributing their changes to the project.

To meet the minimum requirements, the workplan involves creating of initial documentation and an automated testing suite by the Professional Research Associate and example analysis created by the Master's student.

Undergraduates will begin either by working on new scientific computation skills or improving or adding to material of already-developed scientific computation skills. Students who join the lab currently have two first projects to chose from: working through an introductory lab on water simulation, or working through an example analysis of gamma-spectroscopy data. The students try to perform their work using existing documentation; the PI provides guidance when this is inadequate. This provides an opportunity for students to design improvements and learn the basics of contributing to a code repository. In addition to providing valuable training for the students, this process identifies gaps in the documentation that are often invisible to experts.

This initial work is expected to result in improvements and additions to web-accessible tutorial materials. In addition, this training will provide a foundation that will allow the students to attempt the following actions:

- Successfully follow a simple example analysis using the data-access library.
- Successfully follow a tutorial to make and share a change to the library documentation.
- Successfully follow a tutorial to make, test, and share a change to the library source code.

Work	Who	Notes
Roadmap & workplan development	PI, PRA, Master's student	
Basic skills materials	Undergrads	
Data-access library development		
code dev	Professional Research Assistant	this includes testing, user documentation, and dev documentation
contributing	Professional Research Assistant	this includes automated testing and contribution guidelines and instructions
example analyses	Master's student	
documentation testing	Undergraduates, Master's student	documentation testing may feed back into additional skill documetation
Workshop		
Recruiting	PI	
Organization	Professional Research Assistant, Undergraduates	
Pre-workshop analysis coordination	PI, Master's student, PRA	
Community-driven update of roadmap	PI, PRA, Community	

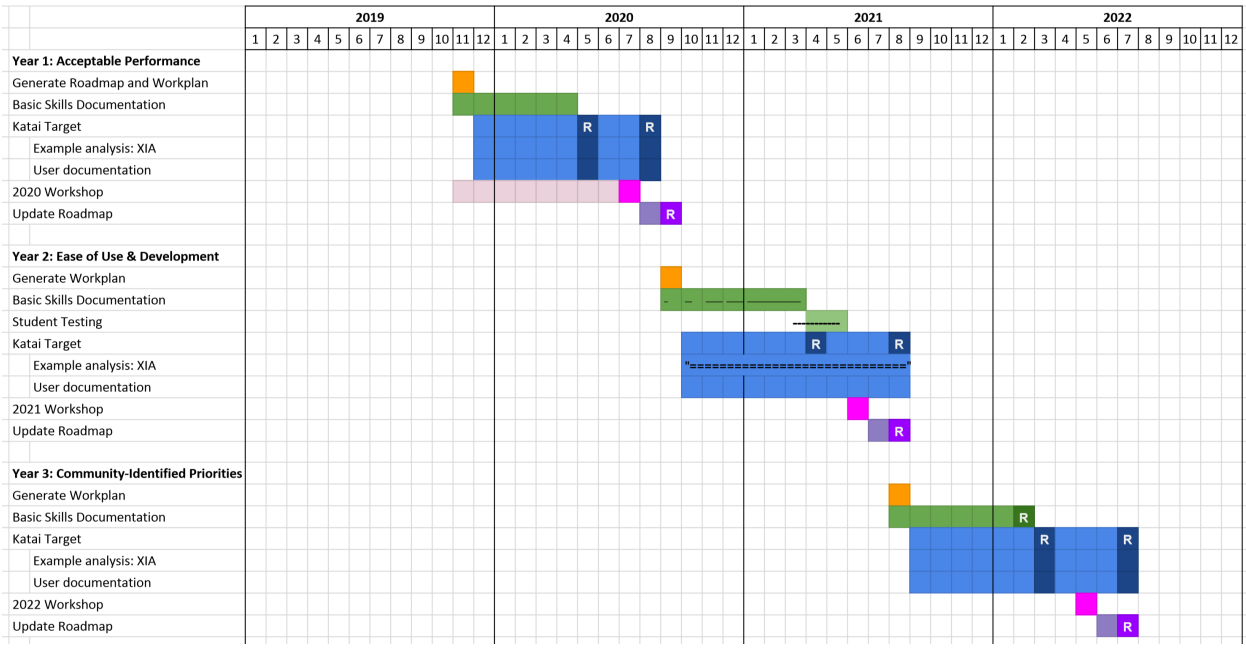

Figure 1: Schedule for the proposed work.

4 Solicitation-specific review criteria

4.1 Science-driven

This project serves the immediate needs of researchers in the dark matter community and the experimental nuclear physics community by providing a common toolset for analyzing data in any format.

The PI expects that

- Multiple research projects across the NSF directorate will be more productive because they can use existing, documented tools rather than building their own. The PI intends to estimate this impact with citations from scientific papers.
- Increased involvement of undergraduate researchers in science analysis due to improved documentation and an extended support network. The PI intends to measure this through undergraduate involvement in her own lab, community surveys, and tracking community forum data.
- Several example analyses will be publicly released, with accompanying documentation and support information for pre-requisite computing skills. The intent is for these educational materials to be accessible to someone with no domain knowledge. The PI believes these training materials will be an equitable training resource.

4.2 Innovation

A common limitation of data-analysis software that is entirely home-grown is that it does not scale as data grows and changes - a human has to update or rewrite the code if the requirements change substantially.

This library makes heavy use of existing libraries. The benefit is that as those libraries improve and scale to larger data sets, this software inherits that improvement.

In both cases, significant human effort is needed to adapt the software to changing data and needs. But by leveraging well-supported, open-source libraries, the burden is shifted away from an individual scientist and towards an active community that is highly motivated to solve similar problems. This library serves as sugar to allow scientists with many different data formats to take advantage of these popular libraries.

The danger to this approach is the same - if these libraries lose community support or focus on very different problems then this library will lose relevance over time. To mitigate this risk, the PI is focusing on integrating with a library supported by the IRIS-HEP collaboration and with pandas, which enjoys extreme popularity in the data science community.

4.3 Close collaboration among stakeholders

The PI proposes to engage both cyberinfrastructure experts and the experimental nuclear physics community by (1) working closely with pilot experiments to build software that works effectively for scientists analyzing event-based data, (2) holding yearly workshops intended to foster interaction between the scientists using the software and cyberinfrastructure developers, and (3) attending conferences that will allow outreach to the scientific community (for example, the Low Energy Community Meeting) and the cyberinfrastructure community (for example, CHEP).

The PI has working relationships with scientists in the SuperCDMS collaboration and the XIA corporation, both of which are interested in exploring the proposed software as solutions to analysis needs.

In addition, the IRIS-HEP collaboration is interested in this work as it would extend their awkward-array library to a broader audience. Awkward array was developed as part of DIANA/HEP [**?**] and that effort will continue with IRIS-HEP [**?**] . Collaborating with IRIS-HEP gives us access to experienced cyberinfrastructure developers who have focused on developing software suitable for terabyte-scale data.

4.4 Building on existing, recognized capabilities

The proposed work builds on existing capabilities and communities in several ways:

- **The PI proposes to use already-existing data description languages.** The languages Katai Struct and the Data Format Description Langauage both have active communities and tools that work with data when provided a description. Kaitai Struct is the target for the proposed work because (1) it is more human-readable than than the XML-based DFDL, and (2) Katai Struct generates code libraries that allow users to load their data into the programming environment of their choice; DFDL currently works by providing an XML or JSON equivalent of the binary data. While this is a powerful approach because any language with an XML or JSON parser can now read the data, it also produces a secondary data file that is an order of magnitude larger than most binary files. This makes DFDL, in its current state, unusable for scientists with gigabyte-scale data sets as it would make the required storage space for analysis prohibitively expensive.

- **The PI proposes to use already-existing infrastructure for the data-analysis library.** Scientists who would like to avoid writing custom software to read their binary data can already use the Kaitai Struct compiler to generate libraries to read their data in python, C++, and a multitude of other languages. The advantage is that there is substantial support documentation and an active community available for troubleshooting. The disadvantage is that the current Kaitai Struct python compiler stores the data in a structure that does not provide adequate speed performance for gigabyte-scale data sets. By improving the existing Katai Struct compiler software, we can build a science-ready analysis library and scientists can benefit from the existing community support and documentation.

- **Use a supported and optimized data structure** for the improvements to the Kaitai Struct compiler. The "awkward-array" library was developed by DIANA/HEP and is now supported by IRIS-HEP and is part of a set of libraries designed to provide flexible data-analysis tools for the high-energy physics community. The awkward-array data structure is optimized for fast queries on an event-based data set and as such is ideal for the majority of nuclear physics data. By choosing this data structure as the target, we bring the optimized and convenient analysis environment of awkward-array to any scientist who describes their data with the Katai Struct language.

- **Provide analysis tools for the python environment and training materials that take advantage of the python ecosystem.** Python is a popular analysis environment in the field of big-data and has enjoyed significant adoption in the scientific community; enough so that python support is compiled in the dominant high-energy physics software, ROOT, by default. By providing a python library for data analysis, scientists can make use

of a full ecosystem that supports data analysis: numpy for convenient array manipulation; scipy for fitting; matplotlib for producing publication-quality figures; and even numba for easy compilation of code that needs to run fast. This entire environment is easily installed - even for users without administration privileges - through the Anaconda Python distribution. There are many free and paid programming environments that are availble, notably the Jupyter environment. Code written in this environment is particularly nice as a tutorial because it is rendered nicely on github, gitlab, and interactive notebooks can be opened in one click through binder. By providing a small set of introductory documentation, scientists can benefit from the effort the python community has put in to lower the barrier for use.

4.5 Project plans, and system and process architecture

Architecture of the software: The architecture of the Kaitai Struct compiler that targets an awkward-array data structure will follow that of the existing Katai Struct software. Implementing a Kaitai Struct compiler for a new language requires

1. Writing a "runtime library" that provides a standard stream interface in the target language. For example, one of the functions every language needs to have defined is a method that returns the size of the file or string stream. By writing a "size" function for the language of interest that follows the Katai Struct API, the code generation becomes simpler.
2. Writing a "compiler" that translates Kaitai Struct concepts, implemented in Scala, into the target language.
3. Writing a test runner for the new language.

The proposed work targets the python environment, and there is already a Kaitai Struct python compiler. The "compiler" for the python implementation, however, stores data in native-python data structures that provide inconveniently slow access to standard queries on large, gigabyte-scale data sets.

However, the changes that need to be implemented to instead store the data in the faster awkward-array data structure are restricted to the compiler code. The runtime library provides a convenient interface for reading data from a file or stream - this code only cares about the file system interface and does not need to change. The python test interface will need to be updated as the access syntax for the data will change slightly.

Although there is opportunity for improving the speed of the data load, this development will instead focus on adhering to the existing format and style of Kaitai Struct. The goal is to make the existing Kaitai Struct community useful to scientists who work with gigabyte-scale data sets; waiting for a few minutes for the data to load is not ideal but is typical of many locally-built solutions. We can address the more-critical issue of rapid data queries while staying well within the existing framework of Kaitai Struct and intend to do so for the initial implementation of the software.

If we find that data-load times are a significant issue for the nuclear physics community then we will consider more substantial changes to the Kaitai Struct compiler and runtime libary.

Architecture of the user documentation: User documentation should make it possible for users with little to no domain knowledge to use the data-access library for science. Documentation for the use of the library will be stored as text files in the repository with the code. The files

will be written in markdown syntax to improve their readability; this will also render them nicely on cloud-based repository hosts such as github and gitlab. The following documentation will be provided:

1. How to get help with questions or issues about the library.
2. How to install the library and its dependencies.
3. An overview explaining what the user will need to provide (data and a description of the data) and what the library will provide (software to read that data).
4. A tutorial walking through the use-case of a scientist looking at simple data with a custom format.
5. Links to additional resources detailing more complex data formats and more complex analyses.
6. Citation guidelines.

Architecture of the developer documentation. Documentation intended to facilitate development of the code will be stored in the repository alongside the code. Text files referenced in the top-level README file will detail, for every repository,

1. How to install, develop, and test the code for individuals who wish to make changes.
2. How to contribute changes back to the project. This will provide instructions on the version control practices used by the repository maintainers and instructions for implementing the tests required for changes to be considered for merging with the main code base.

Architecture of the basic scientific computing skills documentation: Documentation of basic computational skills and concepts will have several possible forms: (1) Text and images, (2) tutorial videos, (3) jupyter notebooks, (4) printable images that illustrate a focused concept, and (5) links to recommended resources such as Software Carpentry tutorials.

All materials will be licensed with a permissive, open-source license such as CC-BY or MIT. The source for all the materials will be publicly available through a public host such as github or gitlab and will be archived on a content-tracker such as the Open Science Framework or Figshare. Videos will be released on YouTube and licensed CC-BY.

All materials will be disseminated using a static site generated by Antora. Antora is specifically designed for documentation and allows a user to specify a set of repositories containing text files formatted in the Asciidoc markdown language to build a single, searchable documentation site. Because Antora generates a static site, free hosting services are readily available. This solution allows my students to focus on creating material to explain core concepts and practice interacting with version control rather than spending time wrestling with web development.

The topics students choose to document are largely student-led, with some guidance from the PI. Spring 2019 marks the inception of this project, and the concepts chosen by students for illustration have focused on (1) tutorial-format guide for installing python and running a basic python-based analysis of gamma spectroscopy data, (2) instructions for using a docker container to simplify installation of a complex software environment, and (3) a poster explaining what an executable file is.

Documentation that will be provided in this format alongside the scientific computing resources will include

1. instructions on where to get help with the material and how to provide feedback and and file bug reports
2. instructions for those who wish to contribute to the documentation
3. instructions for the deployment of the documentation

Engineering processes: A primary goal of the proposed work is to build software that can be supported and maintained by the community. A primary risk of the proposed work is staff turnover and associated loss of knowledge and onboarding time.

The software design, development, documentation, and testing work together to make it easy for the community to contribute to the software development - a goal that mitigates turnover risk as well.

The design process of the software will start with project documentation that describes (1) use cases, (2) requirements, (3) assumptions, (4) key decisions, and (5) definitions. Such documentation is particularly useful for programmers who lack experience in experimental nuclear physics data analysis and makes it easier for skilled experts to contribute to the project. This documentation also serves as a way to focus community discussions into defining a minimum useful scope for the software.

The development of all software in the PI's lab is done using version control software (git) and a central "repository server" that everyone interacts with. Cloud-based servers such as github and gitlab are used because they are easy to use, provide robust backups, and also serve as a platform for dissemination and collaboration.

The PI's approach to version control and software releases prioritizes (1) easy-to-get, working code and (2) rapid updates. This is implemented by building end-to-end tests and configuring automatic test running triggered by any changes to the repository. Rather than insisting on a specific release cycle, developers are encouraged to put their changes on the public, master branch if their code is non-breaking. Code on the public, master branch MUST pass all tests. Semantic versioning will be used to alert users to breaking changes.

Work that breaks tests MUST be maintained on a separate "branch" that is publicly available but that will only be available to users by explicit action. Instructions for developing code on such a branch will be included in the contributions documentation.

The key to this type of development is building simple end-to-end tests, implementing an automated testing framework, and investing heavily in documentation and testing of that documentation up front. This development strategy works well for the proposed project because the software goal is already well-defined and the initial plan for implementation - leverage the existing Kaitai Struct framework as much as possible - is clear. In addition, this development strategy is well-supported by community solutions and the organization of the PI's lab:

- end-to-end testing of python code - and even tutorial notebooks in jupyterlab format - enjoy a thriving ecosystem in Python.
- many automated testing frameworks are designed specifically to support developers using cloud-based repository hosts such as github and gitlab; many are freely available to open-source projects.
- novices are always on hand to test documentation. The PI maintains a group of approximately six students, many of whom have minimal experience with scientific computing. Giving

them goals such as: work through this example analysis and obtain a similar plot exposes conceptual gaps and problems with the documentation while providing excellent training for the student. In addition, students have responded well to the opportunity to make a substantive contribution to the lab.

In summary, the proposed code development will be strongly tied to documentation, automated testing, and documentation testing.

Trustworthiness

End-to-end tests of the software will include tests where the outcome is known.

Provenance

All releases will be archived with their own DOIs on Zenodo. Guidance to cite the version of the software used will be included in the top-level README of all releases and posted on all related websites.

Reproducibility

End-to-end tests of the software will completely define all inputs, including a reference data set, and compare the output to reference products such as histograms. The PI acknowledges that this is in no way a complete test of reproducibility but feels that it will be a useful starting point.

Instructions for system setup will be included in the documentation. Full or partial system specifications are required for automatic test running and will be versioned together with the rest of the code. The most popular of these, Docker, will be used. More complete system specifications as provided by Nix and Guix will be considered if need arises for more complete reproducibility.

Usability

For the proposed software, usability consists of several scenarios:

Can a scientist easily use this code to do data analysis on a custom-format data set?	
Is the scientist aware that this software exists? Can the scientist find the software easily even if all they recall is a vague description?	Publications citing DOIs hosted on Zenodo with published preprints; well-indexed project website; open development on github, gitlab; indexed on python library repositories where appropriate
Can the scientist install the software on their analysis computer easily, with or without root access?	Installation documentation; testing of installation documentation by novices; review of systems that need installation support at workshops
Does the software apply itself well to this particular analysis need? Can the scientist see how to use the software in their analysis?	Example analyses; testing of example analyses by novices; creation of a "now you try" document to test effectiveness.
Does the scientist know where to get help or discuss issues with the software?	All documentation and code will contain a header directing users to the project forum and repository issue tracking.
Can an interested individual contribute to the development of the software easily?	
Is there a clear description of the requirements and purpose of the software so that developers can decide if they'd like to participate?	Requirements documentation and a description of use cases will precede all programming work and will be versioned with the rest of the code.
Is it clear where to get help or discuss the code?	All documentation will link to the forum and the repository issue tracker. The forum will be configured for web crawlers to maximize its discoverability.
Are there clear and complete instructions on testing the software locally?	Local and remote testing infrastructure is as high in priority as the initial development of the code; documentation will be written as part of the first efforts. Students will test these local development instructions. Instructions may reference the scientific computing documentation.
Are there clear and complete contribution instructions?	Contribution instructions will be developed with high priority once there is an initial passing test, even before there is functioning code. Students will test these contribution instructions. Instructions may reference the scientific computing documentation.

Adaptability

The proposed software intends to support scientific analysis of gigabyte-scale data for the coming decade.

The benefits of the proposed software, compared to the current, group-driven methods, are increased access to scientific analysis through ease of use, quality documentation, and community support. And improved return on invested maintenance time, since effort on a common set of tools can

benefit many scientists.

Another advantage of the proposed software is that it leverages existing, well-supported projects to deliver science-ready software. Adaptations to changing data needs and opportunities can potentially come from efforts outside nuclear physics.

There is always the possibility that these dependencies could be abandoned. Archives of all dependent software will be made as a safeguard against this. Other risk mitigation the PI will pursue is significant investment in automated testing and documentation, particularly interface documentation.

4.6 Deployment and user outreach:

A key component of the user outreach will consist of annual workshops designed to promote hands-on use of the software and close collaboration with the software developers. To maximize the effectiveness of these workshops,

- Participants will be contacted in advance to begin early coordination of the analysis they're interested in doing with the data-access library
- Communication before and after the workshop will be encouraged through the maintenance of an open forum
- Prototype software will be released along with installation documentation, an example analysis, and contribution documentation prior to the workshop and tested by novices
- Discussion of the community roadmap will be integrated into the workshop and a new release of the roadmap will follow each workshop
- The conference will be registered on the Open Science Framework, providing an archived record of material prepared by participants.

Deployment of this software will use common open-source channels such as github, gitlab, and python package indexes such as PyPI and Conda. Github and Gitlab both provide static page serving for projects.

Project communication will use the built-in issue tracking provided by repository hosts and open forum software such as Discourse. Live-chat is not an anticipated need, but if it becomes clear this would be useful then the PI will consider using freely-available chat services such as Slack, Zulip, or Gitter.

Deployment of the software will also happen through academic channels such as preprint servers, project releases on the Open Science Framework, and/or Figshare.

4.7 Acceptance and evaluation:

Community adoption of this software is central to the success and broader impact of this project.

The simplest evaluation metrics of community adoption and use will be (1) citations of software in peer-reviewed scientific papers and (2) the number of contributions to the code from developers outside the PI's group.

Additional metrics that may provide useful information could include: (a) number of downloads from the Zenodo or gitlab site and (b) quantity of interactions on the issue tracker and forum.

The PI intends to use interviews to understand how the software meets (or not) the needs of the community. The yearly workshops will provide an ideal setting for such discussions. In addition, a standing feedback survey will be linked from the project page to capture responses from the people who have the time and willingness to share.

This feedback will directly inform decisions about priorities and have an official outlet in the release of an annual roadmap. Discussion and contribution to the roadmap will be open to the community and will begin at the yearly workshop.

4.8 Deliverables

4.9 Metrics

The simplest evaluation metrics of community adoption and use will be (1) citations of software in peer-reviewed scientific papers and (2) the number of contributions to the code from developers outside the PI's group.

Additional metrics that may provide useful information could include: (a) number of downloads from the Zenodo or gitlab site, (b) quantity of interactions on the issue tracker and forum, and (c) materials provided by data-sharing platforms that educate scientists on data-description languages.

Another informal metric of community support is how many scientists apply to the yearly workshop who are not actively solicited by the group and individual interviews with scientists using the software.

Year 1

An initial release of the improved software and the XIA library is made. Both have basic testing coverage, are tested automatically upon commit, and have initial guidelines for contribution. An initial release of the skills documentation has been made.

- A whitepaper is published describing data description languages and giving use-cases
- No citations of the software library are expected from peer-reviewed science results in year 1.
- Between 15 and 20 scientists register for the Data Access workshop. These scientists are expected to predominantly come from the nuclear physics community. The PI expects all attendees to register after specific invitation.
- Between 2 and 5 scientists register projects with the Open Science Framework or similar platform to work on analysis of their data collaboratively
- An initial release of the improved software and the XIA library is made. Both have basic testing coverage, are tested automatically upon commit, and have initial guidelines for contribution.
- Inexperienced students who try to follow the analysis tutorial are able to find answers to some of their questions but most are expected to be unable to complete the tutorial without expert assistance
- Interviews with scientists attending the workshop show that out of ten scientists: approximately two find the software useful as-is for their analysis work; approximately five would find the software useful but do not plan to use it because of solvable issues; and approximately three do not plan to use the software either because it is not useful to them or because their issues with the software cannot be easily addressed.

Year 2

The highest-priority improvements as identified by the community roadmap are released. The contribution guidelines and instructions are well-tested. The most common failings of the skills documentation have been addressed.

- At least one peer-reviewed science result cites the data-access library.
- Between 15 and 20 scientists register for the Data Access workshop. Some diversity of discipline is expected, although most are expected to predominantly come from the nuclear physics community. The PI expects one or two attendees to find out and register for the conference from someone outside the group and for most to register after specific invitation.
- Between 3 and 7 scientists register projects with the Open Science Framework or similar platform to work on analysis of their data collaboratively
- There has been at least one request for an improvement or feature from the community that has been either fixed by my team or another contributor.
- Inexperienced students who try to follow the analysis tutorial get stuck on these issues less frequently. Most are able to complete the tutorial without expert assistance but are unable to make significant progress on the "now you try" analysis tutorial without expert assistance.
- Interviews with scientists attending the workshop show that out of ten scientists: approximately four find the software useful as-is for their analysis work; approximately four would find the software useful but do not plan to use it because of solvable issues; and approximately two do not plan to use the software either because it is not useful to them or because their issues with the software cannot be easily addressed.

Year 3

The highest-priority improvements identified on the road map for the core library are released. The contribution guidelines and instructions are well-tested. The basic-skills documentation has resources or recommends resources that address most of the questions that arise when inexperienced students try to follow the analysis tutorial.

- At least five peer-reviewed science results cite the data-access library.
- Between 15 and 25 scientists register for the Data Access workshop. Some diversity of discipline is expected, although most are expected to predominantly come from the nuclear physics community. The PI expects at least three attendees to find out and register for the conference from someone outside the group and for most to register after specific invitation.
- Between 5 and 10 scientists register projects with the Open Science Framework or similar platform to work on analysis of their data collaboratively
- There has been at least five requests for an improvement or feature from the community that has been either fixed by my team or another contributor.
- Inexperienced students can independently find answers to most of their questions when following the analysis tutorial. Most students still need expert help on the "try your own" analysis tutorial but find it easier to formulate their questions

- Interviews with scientists attending the workshop show that out of ten scientists: approximately six find the software useful as-is for their analysis work; approximately two find the software useful as-is but would very much like to see one or several non-trivial issues addressed; and approximately two do not plan to use the software either because it is not useful to them or because their issues with the software cannot be easily addressed.

4.10 Sustained and sustainable impacts

The goal of the proposed work is to significantly increase the accessibility of scientific data analysis on custom-format binary data.

The proposed work will not eliminate the need for developing custom software within institutions and groups - but the project will provide a common toolset that will work for the majority of gigabyte-scale, event-based analysis despite the lack of a uniform data format. The success of this project is expected to

- Reduce redundant time spent on writing custom data-access software
- Reduce experts' time spent fighting with existing software during analysis
- Significantly increase the involvement of students in the science-analysis phase of research
- Improve the reproducibility of science results
- Allow software development to occur in an environment where credit for this often-invisible work is possible

The success of this project depends on significant community involvement; the PI believes that the flexible nature of the proposed software, its immediate need for many in the community, and the community-building work of the grant can achieve this goal.

The plan for long-term sustainability of this project is twofold. (1) The scope of the proposed software is small and the PI expects this software to be in a mature state by the end of these funds. Therefore the project is expected to require less effort from the core development team. (2) It is expected that the work proposed will build a broad community of users and developers who will continue to support the software because it is directly useful to their science goals.

In addition to hoping that the community developed during the course of this funding will be able to maintain the software at a usable level, the PI will investigate possible support from organizations like NumFOCUS as well as the Open Science Framework and Figshare.

5 Results from Prior NSF Support

The PI has no completed NSF grants, but is a co-PI on the current SuperCDMS NSF grant entitled: "Collaborative Research: The SuperCDMS SNOLAB Experiment (NSF-1809769)." The award amount was split between multiple institutions and the 3-PI contingent at CU Denver was allocated $340,000 over three award years between 8/15/2018âĂŤ7/31/2021.

This PI received 0.5 months of summer salary per year and travel funds to attend collaboration meetings.

5.0.1 Intellectual merit

This grant supports students and scientists working on an experiment that addresses one of the most fundamental problems of modern science, the nature of dark matter. The SuperCDMS SNOLAB experiment will achieve world-leading sensitivity for dark matter searches in the 1–10 GeV/c^2 mass range.

The PI's group is funded primarily for contributions to the data acquisition and data quality systems.

5.0.2 Accomplishments

This grant has been recently awarded and there are not yet any publications.

A critical need facing the collaboration as we move to larger data sets is transitioning our analysis platform to computing clusters where we can submit jobs to batch queues. This requires a reworking of existing analysis software, which is primarily MatLab-based and cannot be run at SLAC.

The group began testing and documenting the installation requirements of prototype python software and has since developed the first isolated build environment, allowing reliable installation of the analysis tools across platforms.

Josh Elsarboukh has worked closely with SLAC computing division to successfully deploy this analysis environment via a web interface. This work represents an unprecedented ease of access within the collaboration and has made it possible for test facilities working on crucial R&D and calibration efforts to efficiently analyze their data.

5.0.3 Broader impacts

The SuperCDMS experimental and R&D efforts advance phonon-mediated detectors and new active veto concepts, which have already found many applications in cosmology, astronomy and industry.

Efforts to improve the accessibility, maintainability, and reproducibility of the analysis environment are goals that many scientific experiments share; these efforts further support the development of a scientific ecosystem that's badly needed. Our efforts have already helped the ATLAS collaboration set up a similar environment and we are currently working to set up a similar service with ComputeCanada.

This grant supports the training of undergraduate researchers. The CU Denver Physics Department maintains an undergraduate-only program committed to providing students with hands-on research experience at this urban campus. Currently the PI's lab employs seven undergraduate students, all of whom are involved in developing documentation for scientific computing skills relevant to SuperCDMS. Current students are from physics, mechanical engineering, chemistry, and computer science majors and all learn basic computing skills that are essential for scientific research. CU Denver enjoys a diverse student population and the PI actively seeks to represent the student population in her lab and to ensure that her lab provides a space for students to thrive. The PI has adopted a code of conduct and published her interview process and expectations. The PI is able to cast a broad recruiting net by teaching introductory physics and participating in college-sponsored open houses. In addition, the PI works with students to identify conferences that are of particular interest and helps them secure funding for attendance. Recent examples are the 2018 National Organization for the Professional Advancement of Black Chemists and Chemical Engineers (NOBCChE) conference, and the 2018 Out in STEM (oSTEM) conference. The PI recognizes the

value of the conferences as a place to practice crucial scientific communication skills and also as a place for underrepresented students to build their identities as scientists.