Data-Intensive Analysis for High Energy Physics (DIANA/HEP)

Peter Elmer (Princeton University)
Kyle Cranmer (New York University)
Mike Sokoloff (University of Cincinnati)
Brian Bockelman (University of Nebraska-Lincoln)

June 27, 2014

1 Overview and Objectives

Advanced software plays a fundamental role for large scientific projects - from designing the experimental instruments to acquiring, reducing, and analyzing the resulting data. In such projects, success requires large-scale collaboration; software is the glue which enables teams of researchers to work together to exploit accelerators, telescopes and other large scientific instruments. Building the requisite software is technically challenging because the computing technologies (processors, storage, networks) are evolving and data volumes are increasing rapidly, requiring ever more sophisticated data analysis methods. Individual projects, experiments and researchers are typically organized to succeed at specific goals. Significant organizational, funding and scheduling barriers exist, especially for university researchers, to creating and sustaining software for an entire research community. Our team of physicists and computer scientists will reach across these traditional boundaries to build the next generation of community analysis libraries within the particle and nuclear physics community. We aim to meet not only the technical challenges to support physics discovery in the next decades, but also to develop these libraries in ways that build bridges to other scientific communities and explore new software-enabled collaborative analysis methods.

The quest to understand the fundamental building blocks of nature, and their interactions, is one of the longest running and most ambitious of human endeavors. Facilities such as the Large Hadron Collider (LHC), where we do our research, represent a huge step forward in our ability to answer these questions. The discovery of the Higgs boson, the observation of exceedingly rare decays of B mesons, and exclusion of countless theories beyond the Standard Model (SM) of particle physics demonstrate that these experiments deliver results. However, the most interesting fundamental physics questions remain wide open, amongst them: What is the dark matter which pervades the universe? Does space-time have additional symmetries or extend beyond the 3 spatial dimensions we know? What is the mechanism stabilizing the Higgs mass from enormous quantum corrections? Are neutrinos, whose only SM interactions are weak, their own anti-particles? Can the theories of gravity and quantum mechanics be reconciled? Experiments that address these questions in the next ten years will collect exabyte-scale data samples.

We will build more powerful analysis tools to provide sophisticated and efficient data reduction techniques, methods for statistical combinations of multiple data sets within a single experiment, and a high-level framework for collaborative efforts that extend beyond the boundaries of an individual experiment. The first motivation is that software performance must grow faster than the size of the data sets themselves to provide the reach required to make scientific breakthroughs - especially as hardware budgets are not expected to keep up. Providing this level of performance presents a major challenge both to the general scientific community [1] and to our own area of physics [2, 3]. Emerging computing architectures nominally address the issues of exponential growth in computing capacity (Moore's Law) and power consumption, but our community software is currently unable to fully exploit these technologies. Similarly, the reach of data-intensive experiments is limited by how fast data can be accessed and digested by the computational resources; both technology and increasing data volume require new computational models [4]. A second important motivation for this work is that the sophistication of analyses required to find, and eventually interpret, physics beyond the SM in ever larger data sets demands new tools and techniques.

The Principal Investigators for this project have had leadership roles in a variety of complementary software development efforts in three LHC experiments (ATLAS, CMS, LHCb) and the Open Science Grid (OSG). The PIs are responsible for several of the most widely used community libraries and developments in the overarching analysis architecture used by the field. Building on past successes creating and deploying both experiment-level and community software elements, our team is uniquely well positioned to make major improvements to a key community framework.

The ROOT toolkit [5] is the *de-facto* home for most community analysis software developed in particle physics and related fields. Begun at CERN in 1995, it provides a sophisticated data format and serialization technology (used, as an example, for 200 PB of LHC data) as well as key software tools for data modeling, likelihood fitting, statistics and multivariate data analysis. It also has a broader range of functionalities, not strictly tied to the data-intensive aspects of our science, including interactive C++ analysis, histogramming, graphics (2D and 3D), math libraries (matrix algebra), image manipulation, and tools for distributed computing. Despite many pioneering and innovative features, the components are widely seen as too coupled, and also limited by design decisions taken 20 years ago. Given the challenges from technology evolution and analysis complexity, we are at a point in the software lifecycle where large changes are needed, much like when ROOT replaced an earlier generation of FORTRAN-based tools (PAW/HBOOK). ROOT is, however, the starting point for any plan to enable the our community to exploit exabyte-scale data sets.

Therefore, we propose to create the *Data-Intensive Analysis* (**DIANA**) project to build on and improve these community libraries, move other existing software elements into community libraries, and develop additional new tools. We envisage three interrelated areas of activity:

- **Performance:** we will greatly increase parallelism and eliminate CPU- and IO-bottlenecks to achieve higher processing rates necessary to efficiently and expeditiously analyze large volumes of data. We will address some key design and implementation issues from the early days of ROOT which impact not only performance, but also the manageability of the software.
- **Interoperability:** we will reposition these key libraries to better interoperate with the larger scientific software ecosystem, transitioning the field to a more sustainable path where *new ideas and software developed elsewhere can be more easily used in particle physics* and our best products can be evaluated by other fields. We will create modular versions of the libraries that work both within the traditional ROOT framework as well as within other frameworks such as Hadoop MapReduce, Apache Spark, Mathematica, Python and R.
- **Collaborative Analysis:** we will provide new tools that build on the concept and emerging practices in particle physics that data analysis is a collaborative activity, involving many individuals working within a given experiment, working in different experiments and even between the experimental and theory communities. This will involve directly integrating into the analysis tool suite the ability to capture elements of analysis workflows needed to satisfy best practices in data preservation, analysis archival, reproducibility, and open access.

Our vision for DIANA extends beyond the work the project team will do itself. We will collaborate with other physicists and data scientists from universities, national, and international labs as well as the private sector to catalyze a broad effort. Equally importantly, *we will provide training and professional opportunities* related to developing and maintaining high-quality software to high energy physicists, data scientists, and those whose interests straddle the fields. We will judge this project a success if both (i) the tools we develop provide the performance and sophistication required to enable great physics, and (ii) our colleagues find new and powerful ways of working with others in the field and develop greater synergies with colleagues in other fields.

2 Physics Motivation

The DIANA team does research in particle physics, and our interest in developing data-intensive analysis software stems from our desire to do, and enable, great science. The understanding of physics at the shortest and longest distances is built on two related foundations: (i) the Standard Model (SM) of particle physics, a quantum field theory of quarks, leptons, and their interactions, and (ii) ΛCDM, a theory of cosmology based on Einstein's theory of gravity, also called general relativity, describing the Universe at the largest scales. Here, Λ denotes a cosmological constant associated with dark energy and CDM denotes the cold dark matter which, along with ordinary SM matter, fills the Universe. An abundance of experimental results provides precise verifications of these theories, but also exposes where they are incomplete.

For instance, today's best fits of the ΛCDM parameters from Planck satellite data indicate that SM matter accounts for only 4.9% of the mass-energy budget of the Universe, while dark matter accounts for 26.8%. While we know something about dark matter at macroscopic scales, we know nothing about its microscopic, quantum nature, *except* that its particles are not found in the Standard Model and they lack electromagnetic and SM nuclear interactions. This has motivated a large number of SM extensions, called beyond the Standard Model (BSM) physics, using the formalism of quantum field theory. Constraints for BSM physics come from "conventional" HEP experiments plus others searching for the dark matter particles either directly and indirectly.

One of the great successes of big bang cosmology is its ability to accurately predict the abundances of the light elements produced in the early Universe. However, the dominance of matter over anti-matter, which relates to the fundamental processes of leptogenesis and baryogenesis, is not well understood and requires BSM physics. This motivates a diverse set of experiments with and quarks and neutrinos, how the latter mix with each other, and CP violation, one of the necessary conditions for creating this matter/antimatter asymmetry.

The Higgs boson discovery in 2012 [6,7] answered the last open question of the Standard Model. It experimentally established the mechanism for electroweak symmetry breaking and confirmed that fundamental particles acquire their masses via their interactions with the Higgs field. It also emphasizes the importance of searching for BSM physics. To understand why, consider the muon's gyromagnetic ratio, g_μ. Through a tour de force, this fundamental constant of nature has been has been experimentally measured by the "g-2" experiment to be $g_\mu = 2.0023318416 \pm 0.000000001$; its value has been calculated with similar precision by theorists. Results like this give us confidence in quantum field theory because they critically rely on so-called quantum corrections. In this case, corrections to the initially predicted value of $g_\mu = 2$ are very small. In contrast, the same logic leads to quantum corrections for the mass of the Higgs boson that are very, very large – much larger than the measured mass itself, $m_H = 125.5$ GeV. This mystery, the so-called naturalness or hierarchy problem, has driven much of theoretical particle physics for the last two decades. Proposed solutions generically predict new particles. Some theories, such as supersymmetry, address both the hierarchy problem and provide credible dark matter candidates.

The LHC was designed to search for the Higgs boson and for BSM physics – goals in the realm of discovery science. ATLAS and CMS are optimized to observe and measure the direct production and decay of massive particles. They have now begun to measure the properties of the Higgs boson to see how well they accord with SM predictions. In parallel, they are searching for particles predicted by supersymmetry and other BSM theories. Despite an impressive level of activity, an even larger number of theoretical scenarios are not addressed explicitly, or are left altogether unexplored experimentally. This drives our interest in developing protocols for providing model-independent summaries of experimental data that can be compared to a wide variety of models. The consumers of these data could be the original experimentalists (as new models become available), phenomenologists developing new models, or "consolidators" such as the Heavy Flavor Averaging Group [8], the CKM Fitter [9,10], and UT Fit [11,12] collaborations, who systematically compare data from many measurements and multiple experiments with underlying models.

Where ATLAS and CMS were designed to study high mass particles directly, LHCb was designed to study heavy flavor physics where quantum influences of very high mass particles are manifest in lower energy phenomena. Its primary goal is to look for BSM phyics in CP violation (asymmetries in the decays of particles and their corresponding antiparticles) and rare decays of beauty and charm hadrons. As an example of how one can relate flavor physics to extensions of the SM, Isidori, Nir, and Perez [13] have considered model-independent BSM constraints from measurements of mixing and CP violation. They assume the new fields are heavier than SM fields and construct an effective theory. Then, they "analyze all realistic extensions of the SM in terms of a limited number of parameters (the coefficients of higher dimensional operators)." They determine bounds on an effective coupling strength Λ of the interaction in the limit that the dimensionless couplings

c_{ij} between flavors are unity. The critical point of their results is that kaon, B_d, B_s, and D^0 mixing and CPV measurements provide powerful constraints that are complementary to each other and often constrain BSM physics more powerfully than direct searches for high mass particles.

Summary of Physics Motivation: The Particle Physics Project Prioritization Panel (P5) issued their *Strategic Plan for U.S. Particle Physics* [14] in late May. It was very quickly endorsed by the High Energy Physics Advisory Panel and submitted to the DOE and the NSF. The section titles in "The Science Drivers" chapter underscore the breadth of important questions to be addressed, and the last identifies computing as a key element for making progress: (i) Use the Higgs boson as a new tool for discovery; (ii) Pursue the physics associated with neutrino mass; (iii) Identify the new physics of dark matter; (iv) Understand Cosmic acceleration: dark energy and inflation; (v) Explore the unknown: new particles, interactions, and physical principles; (vi) Enabling R&D and computing. We want to enable as much of this great science as possible.

3 Research Program

DIANA will provide analysis tools needed for data intensive research in physics. In the process it will build on results from three prior NSF-funded R&D efforts (see Section **??**) and integrate related products into community software. We will start with the existing ROOT toolkit [15]. This is widely used in particle and nuclear physics. It has more than 20000 users; nearly 7800 subscribe to the RootTalk forum. It is, however, at an important point in the software lifecycle. The original authors have retired (Rene Brun) or moved on to other projects (Fons Rademakers). Technology evolution and physics needs require significant change, but as Rene himself notes: "the main challenges in working with long-established software like ROOT are the large inertia that has been developed and the large amount of data" [16]. Lacking a standard definition of sustainability [17], our operational definition is "software that continues to serve its users". To continue to serve its users, ROOT must evolve profoundly. Historical pressures of data volume, computational needs, and early adoption of C++ led to significant "in-house" software development. Hence, ROOT's components are coupled, and this reinforces the model in which HEP must build everything itself. Barriers exist to use critical pieces (e.g. the IO libraries) in other contexts and to use non-ROOT libraries efficiently with ROOT. However, ROOT has engaged most of the community as the de-facto common software. In addition to core contributions from the CERN ROOT team, it has attracted major contributions from the community (xrootd, TMVA, RooFit, RooStats, etc.). It has built an agile process for integrating bug fixes and community feedback. ROOT must become more powerful through better performance and additional functionalities. DIANA aims even higher. Some elements of ROOT are intimately connected to how we do data-intensive science (data formats, statistics tools), others less so (histograms, graphics, etc.). To become more sustainable, ROOT must evolve into a more loosely coupled distribution of reusable software elements able to interoperate with more widely used scientific frameworks. On top of these changes, we will build an exciting new framework for collaborative analysis to further connect our communities and others.

DIANA will form a collaborative partnership with the CERN ROOT team, and will provide specific and significant deliverables. We will also partner with other individuals and groups to catalyze additional software elements that contribute to our vision for this software. The Research Program we plan has three broad areas of effort: *performance*, *interoperability*, and *collaborative analysis*. These areas are linked in the technical sense that the same common analysis software code libraries will be changed for each. But they are also linked in a more important sense that ultimately we build our multi-step analysis workflows from the ensemble of pieces we describe.

Performance: As data volumes grow and analysis techniques increase in complexity, the clock time required to process data often constrains researchers. Both CPU and IO bottlenecks appear in various phases of data analysis ranging from Grid-based bulk data reduction stages to sophisticated fits of large, complex data sets to highly interactive exploratory work. These bottlenecks limit how quickly physicists are able to examine the data, pose new questions, and iterate. As an example, consider a time-dependent amplitude analysis of the decay $D^0 \to K_S^0\pi^-\pi^+$ [18], very similar to the unbinned maximum likelihood fit discussed in Section **??**. A single fit of BaBar's 400K event sample ran overnight using 20 older CPUs executing under OpenMP, equivalent to $\approx$ 100 CPU hours on a single, modern processor. LHCb anticipates collecting more than 100 times as many of these events in less than a decade. A single fit to the final data sample would require 8000 hours on a single CPU or a day on an nVidia C2070 GPU. As a physics result requires hundreds or thousands of fits to real and simulated data sets to determine systematic as well as statistical uncertainties, completing such analyses expeditiously will require significantly better performance. One goal of DIANA is to address these CPU and IO bottlenecks in a more comprehensive way.

For CPU-bound problems, one focus of DIANA will be to adapt existing ROOT packages to take advantage of GPUs, other many-/multi-core technologies, and core-level vectorization. A related focus will be helping other developers of data-intensive analysis tools adapt their codes similarly, and move them into the ROOT umbrella to encourage wider use.

As an example of adapting an existing package in ROOT, we will collaborate with Wouter Verkerke, at NIKHEF in Amsterdam, to integrate GooFit [19] as the underlying function evaluation engine of RooFit [20,21] for parallelized execution. He was one of the original RooFit developers; he both maintains it and adds features. He will manage the user interface which defines functionality while we will integrate lower level technologies to speed up execution. As discussed in Section **??**, GooFit executes six times more quickly on a single core, provides multi-core functionality via OpenMP, and many-core functionality on nVidia GPUs. Cincinnati is already funded to continue work on GooFit for use on architectures like Intel's Xeon Phi and multi-node systems with many HPC GPU boards. As that technology is developed, DIANA will integrate it into RooFit.

Similarly, we will collaborate with Jonas Rademacher and his group at Bristol University to adapt the GooFit function evaluation engine for use with MINT, their interface for the MINUIT fitting package. Together, we will move this into the ROOT library. We will also continue to collaborate with Marco Gersabeck and his group at the University of Manchester, not only to extend the functionality of GooFit (now as part of RooFit), but also to incorporate GPU code they are developing for model-independent statistical tests into ROOT.

To maintain exponential growth rates in computing capacity, vendors not only deploy multi-core technologies, but also they are expanding use of vector registers within individual cores. For example, each Intel's Xeon Phi coprocessor core can perform sixteen concurrent calculations with 16 single precision floats. Software that is not organized such that the compiler can use these extended registers may only achieve 1/16 of the theoretical performance for this chip! Previous studies [22] have indicated this is a real issue for particle physics software; in that study, the average number of instructions executed per clock cycle is around 0.5.

The ROOT APIs for iterating over data, developed 20 years ago, provide access to the data from files only one "event" at a time. (An "event" in ROOT typically corresponds to the data products associated to a single particle interaction.) This serial access prevents many opportunities for vectorization, even if the transforms applied are amenable to vectorization. We will develop new APIs for ROOT that will allow user code to execute in parallel when performing binning when only simple transforms are applied. In other cases, multiple levels of filtering are required to distill a data set; we will develop a prototype system for caching events that have survived one level of selection until enough are ready to make vector processing efficient at the next level.

Similarly, we will target the TMVA multivariate analysis/machine learning package [23] which now sits in ROOT. These same techniques will be applied more generally throughout the software artifacts produced by DIANA. In addition to vectorizing code inside existing ROOT packages, and those we add to ROOT, we will collaborate with developers from the large HEP experiments to design algorithms and classes that can be used more widely.

For IO-bound problems, DIANA will provide core improvements to ROOT's RIO library. RIO offers a highly specialized file format for the ROOT user community - it provides serialization for arbitrary C++ objects, achieves high levels of compression, provides random access, provides both row-oriented and column-oriented storage, and can even do partial object deserialization (allowing the user to efficiently read out only a single attribute of an object). The file format and corresponding libraries are highly optimized to the community's use cases - critical, as "minor" decreases in file size and/or processing time save significant disk, tape, and CPU resources. CMS alone manages more than 100 PB of data, including replicas on disk and tape. We will improve RIO's performance by an order of magnitude via technical improvements to the existing IO path, a new "fast path" for certain objects, and by providing tools to help guide users' design choices.

Bottlenecks reading data usually result from latencies or the CPU cost of decompressing or deserializing bytes read from files into objects in memory. Various C++ features require RIO to edit the deserialized object in memory before it is valid (for example, updating a pointer to another object). These alterations can be a major performance hit. There is no indication to users of the "deserialization cost" of a particular object; accordingly, minor changes to the object can cause major performance decreases. Non-trivial objects can hit CPU bottlenecks in decompression or deserialization before hitting system IO or memory bandwidth limitations. Our first deliverable for improving IO performance will be a toolkit to evaluate the deserialization speed of a given object using a few heuristics (similar in spirit to `lint` for evaluating C source code). We also plan on providing tools for visualizing ROOT file format layout and a module for SystemTap [24] which will allow us to log and correlate RIO API calls with block IO activity in the kernel.

Next, DIANA will speed up RIO by revisiting several early design and implementation choices. One example will be to switch from always writing data in big-endian format to allow reading and writing both big- and little-endian; big-endian was a historical choice and it incurs a performance penalty on the currently dominant x86 architecture. For maintainability we will add performance regression testing for RIO (currently missing) to monitor the performance across ROOT releases.

In RIO, even data consisting only of simple types cannot be mapped trivially into memory after decompression. Regardless of data type, the full complexity of loading arbitrary objects - including checks for schema evolution - is always used. DIANA will provide a new, "fast path" deserialization engine for objects needing no changes after deserialization to be valid objects. Extra copies of data in-memory occur - costing CPU time and cache hits - due to features added over the years, like the "TTreeCache" used to hide IO latencies; we will eliminate these for most objects.

Finally, the currently-sequential CPU cost of unpacking data also limits the parallelization achievable for the computational portion of a program per Amdahl's Law. We have observed 5-15% sequential behavior for common data structures and up to 50% for very complex ones. We will also address a number of issues (both in API and implementation) preventing the data unpacking and decompression itself from being parallelized.

To achieve the order-of-magnitude objective, we plan produce a 50% speedup from the code cleanup, a 100% speedup from improved data structure design using our toolkit, a 100% speedup from the "fast path", and 50% speedup from allowing decompression to use multiple threads.

Interoperability: Currently there is enormous momentum in the scientific software ecosystem largely centered around Python and R. They have large numbers of modular libraries and the low barriers to entry. These are also the languages of choice for the nascent field of data science. ROOT currently provides similar functionality. However the barrier-to-entry for someone outside of the ROOT community to integrate specific parts of ROOT is quite high. Although basic bindings exist to use ROOT from Python and to call R functions, interoperability is more than interfacing.

For example, Noel Dawe, a graduate student frustrated with ROOT's level of interoperability, created the `rootpy` project. His project page [25] summarizes the situation nicely: "The scientific Python community also offers a multitude of powerful packages such as SciPy, NumPy, matplotlib, scikit-learn, and PyTables, but a suitable interface between them and ROOT has been lacking." Noel has made great progress, driven by his own needs, but the feature set is not complete. (He is writing his thesis and will begin a postdoc soon.) We will work with Noel to finish this project so it can be distributed with ROOT, including making changes in ROOT to better support `rootpy`. In addition, `rootpy` uses NumPy a Python extension for large arrays in-memory. We will also work with the developers of Blaze [26], the new version of NumPy that (much like ROOT) permits access to datasets that exceed the available memory, to integrate rootpy, ROOT, and Blaze.

The RIO library is extremely valuable to our community. However, its complex API and implementation language (C++) limit its reuse in other data processing frameworks, e.g. those written in Java. This limitation is self-enforcing: the community tends to maintain and use HEP-specific frameworks (e.g. PROOF in ROOT) where industry frameworks (Hadoop MapReduce or IPython.parallel) at least some superior features. The few attempts to utilize popular "Big Data" frameworks either converted the data to a different format (costly in terms of disk space at large scale and difficult for making careful performance comparisons) or run a separate C++ process (which loses too much performance to be practical). We will develop a "RIO interoperability toolkit" to provide simple, straightforward language bindings in Java and demonstrate their usefulness by integrating them with Apache Hadoop MapReduce and Apache Spark.

Having RIO interoperate with other languages and processing frameworks is half the problem; the other half is improving the ROOT analysis framework's ability to pull data from other file formats. This has been previously been shown by others through integration between ROOT and SQL servers. We will write new prototype plugins to explore the use of HDF5 and Google ProtoBufs as a file format for HEP data; in particular we believe Google ProtoBufs to be promising as they are widely used throughout industry and have many features similar to RIO.

Collaborative Analysis: Collaboration is a core principle of modern high energy physics. The field is composed of experimental collaborations of more than $3,000$ scientists distributed around the world. This aspect of the field famously motivated the invention of hypertext and with it the world wide web. DIANA will establish the infrastructure for a higher-level of collaborative analysis – one which builds on the successful patterns used for the Higgs discovery, one which enables a deeper communication between the theoretical community and the experiments, and one that is built from the ground up with openness, reproducibility, and provenance in mind.

The Higgs discovery highlights an emerging high-level collaborative analysis model of growing importance in particle physics. The Higgs boson decays in several ways, leading to roughly 20 different signatures. Each of these is the topic of research for a team of 10-200 physicists distributed around the world. Those teams use RooFit and HistFactory to create statistical models of the data for their specific signature. To discover the Higgs and measure its properties, each of these statistical models were brought together into a coherent combined statistical model. While combined analyses like this have been performed for many years, there was an enormous shift in the workflow when RooStats introduced the concept of the *workspace*, a technology that leverages ROOT's serialization technology. This allowed groups to work much more independently and share their statistical models in a flexible way for future combined analysis. In 2011, this technology was used to combine results across experiments with an ATLAS+CMS Higgs combination. Similar workspace-enabled multi-collaboration combinations are ongoing for Higgs and other physics topics.

The complexity of statistical models has grown at an extraordinary rate since the introduction of the *workspace*. Figure 1 shows the growth in complexity of statistical models used in the ATLAS Higgs search. The need to combine multiple analyses is not specific to Higgs physics, but also includes searches for supersymmetry, top physics, and B-physics. We expect the same approach will become more common in searches for dark matter and in neutrino physics as well.

DIANA will address the computational challenges presented by the growth of complexity of statistical models resulting from this collaborative approach. In addition to the specific strategies

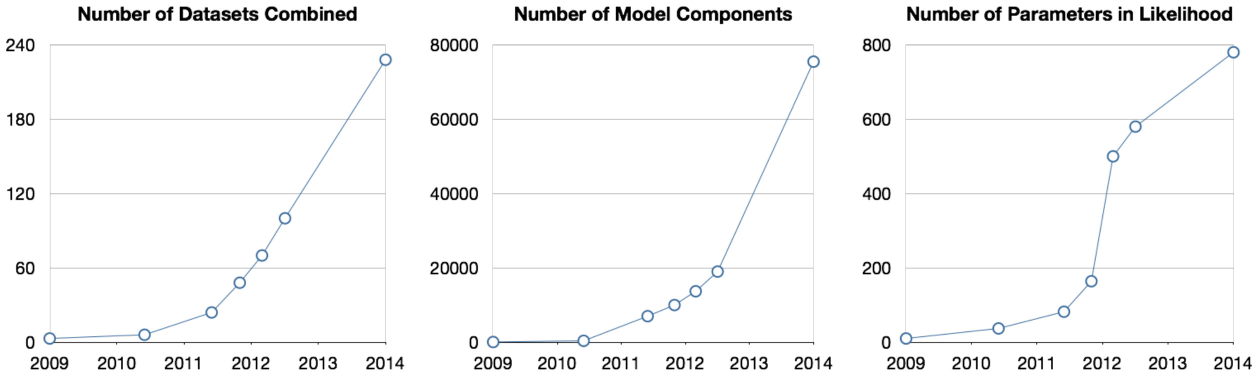

Figure 1: Evolution of the complexity of the statistical models used in the ATLAS Higgs search.

of improving performance via vectorization, parallelization, emerging architectures, and improved performance monitoring, we will integrate the existing statistical tools with numerical algorithms from applied statistics, mathematics, and data science. Examples of algorithmic integrations include optimization algorithms like L-BFGS, Bayesian sampling algorithms like the affine-invariant sampler developed by NYU applied mathematicians and astrophysicists in the form of emcee [27,28], and the low-level code changes needed to implement the automatic differentiation approach utilized in deep learning to avoid numerical precision deficiencies from finite difference algorithms [29].

The Higgs discovery also highlights a level of collaboration across communities. In Ref [30, 31] the theoretical particle physics community endorsed using RooFit/RooStats workspaces as a means of communicating the results of new physics searches. In 2013, the first likelihood scans based on this technology were published to HepData, given digital object identifiers (DOIs), and integrated into the INSPIRE (the HEP literature system) [32–34]. Talks by theorists at CERN workshops referred to this as a "leap forward" in communicating results of experiments [35, 36] and the data themselves have been cited multiple times by the theoretical community.

A second major activity is a broad effort to integrate analysis tools with the platforms being developed for preservation, open access, and provenance tracking. For example, HepData is a trusted repository for our community; however, the process of uploading data products is cumbersome and its formats do not match those used by the experiments. Furthermore, HepData currently lacks a machine-readable meta-data layer. This leads to an unstructured set of histograms associated to a particular paper. DIANA will work with HepData to provide a more direct connection to what is being used by the experiments. For example, we will work with HepData to adopt as a meta-data layer the HistFactory XML schema already being used by the statistical modeling tools. This will both encourage more groups to provide this type of supplementary material, and also improve the quality of provenance information needed for reproducibility and reduce the burden to resource-limited services like HepData.

A final component of this broad effort is to develop the necessary hooks to capture provenance and related meta-data for common analysis workflows. The DASPOS study group which is working to identify these common workflow patterns will produce its final report in 2015. The software libraries which would capture much of the necessary information are precisely those described above. DIANA can thus naturally play the role of follow-on project to implement the required specific hooks and utilities. We anticipate that this would include APIs in the analysis suite that are aware of Digital Object Identifiers (DOIs), utilities to export the provenance information to web standards such as PROV, and utilities to aid in the testing vital to data and software preservation.

4 Professional Development

Sustainable scientific software requires expert design, careful implementation, and useful documentation at the developer and user levels. In many cases, it also requires expert domain knowledge. Traditionally, too much HEP software has been written by "physicists-who-program", frequently by physics Ph.D. students and post-docs as part of required service work on their experiments. The result is often poorly documented spaghetti code [37]. Because HEP data analysis campaigns last for periods from many years to decades, and involve hundreds to thousands of developers and analysts, *providing training, professional opportunities, and recognition* for developing and maintaining innovative and efficient software is critical to the scientific success of the experiments.

While the most tangible products of DIANA/HEP will be community software libraries, developing a pipeline of broad scientists with deep expertise in data science as well as domain science will be even more important in the longer term. We address this issue in three ways. At the first level, we will *train students* to use and develop tools for data-intensive analysis. As discussed in Section 5, we will produce course materials that introduce large numbers of high school students to entry-level data science in the context of HEP. Undergraduate and graduate DIANA Fellows, as well as other students supported by the project, will learn how to develop and maintain community software. At the next level, the post-docs and research scientists employed by the project will be given *professional opportunities* to take ownership in major software projects. The NYU post-doc will be embedded with "research engineers" at the NYU Center for Data Science, part of the cross-institutional effort supported the Moore and Sloan foundations, This individual will participate both in the Reproducibility and Open Science working group and the Careers working group. The two PIs who already work full-time on software development (Bockelman, Elmer) will advance to the next level of leadership, providing both enhanced *professional opportunities* and *explicit recognition* for the quality and importance of their work to date. Finally, to support the development of career paths in this area, we will open a discussion regarding standards for citation of software products and software-enabled techniques from research publications. We will engage experiments, conferences (in whose peer-reviewed proceedings much HEP software research is defacto published today) and journals to develop such common standards, building on emerging best practices. The ensemble of the educational and professional opportunities provided by DIANA will help assure the sustainability of the software developed as part of this project, and will also help create a culture of sustainable HEP software.

5 Education, Outreach and Broader Impact

We will use three mechanisms to train the community to develop and use data-intensive tools:

- Each year, 4 DIANA Graduate Fellows will each spend 3 months intensively developing tools in conjunction with collaborating institutions. Similarly, a DIANA Undergraduate Fellow will work 10 - 12 weeks during the summer, either developing or using data-intensive tools.
- DIANA PIs, post-docs, and students will work with collaborating HEP groups to transform their high school "Masterclass" exercises into self-contained educational units.
- We will educate the HEP community how to best use the data-intensive analysis tools we develop through experiment-specific programs and community-wide short courses.

Collaborating groups will define potential projects and prepare abstracts at least six months in advance for both the Graduate and Undergraduate programs. Applications will be judged by the PIs. DIANA will provide travel and subsistence support for Graduate Fellows, but the students' Ph.D. advisors must pay their base stipends. DIANA will pay Undergraduate Fellows stipends, as well as travel and subsistence support commensurate with that of NSF-funded REU programs. All personnel who volunteer to serve as mentors will participate in formal mentor-mentee training.

An outstanding problem in developing the next generation of data scientists is attracting large and diverse populations to consider the field. The International Particle Physics Outreach

Group [38] and the U.S. QuarkNet program [39] have collaborated with HEP experiments to create a number of activities that give students students the opportunity to analyze simulated and real data. About $64,700$ high school students from around the world have participated in "Masterclass" activities over the past 10 years. Students are introduced to the basic concepts of particle physics and very quickly given data to analyze, usually in pairs. After several hours looking at individual events, building distributions, and drawing tentative conclusions, students from 3 or 4 schools compare their results in video conferences. We will adapt these materials for use outside the Masterclass framework in both high school and university courses. This will require polishing the introductory material, making connections to Common Core and Next Generation Science Standards more explicit, and re-designing the analysis exercises to provide feedback to the students (and their teachers). In addition to developing web-based versions of these materials, we will develop stand-alone tablet applications. We will employ an excellent high school physics teacher (Jeff Rodriguez) and a very experienced iPad app developer (Robert Clair) as consultants to provide supplementary expertise. QuarkNet will collaborate with us on this project. Our goal is to introduce students to the excitement of discovering patterns in data so they become more likely to consider educational and professional trajectories that require this type of critical thinking.

HEP collaborations typically provide opportunities for members to learn software tools. For example, the week-long CMS Data Analysis School (CMSDAS) [40] pairs software experts with new collaborators to build and run end-to-end examples of real analysis applications. Other collaboration have similar programs. In addition to our own experiments (CMS, ATLAS, LHCb) we will offer to participate in other experiments' analysis schools and provide short courses that are open to the community. For example, Rolf Andreassen, then a post-doc on the Cincinnati PIF grant, gave short courses at CERN (March, 2013) and at Fermilab (September, 2013) on GooFit and elements of CUDA. Participants came from all four LHC experiments: LHCb (14), CMS (13), ATLAS (6), and ALICE (1), representing 8 US institutions as well as institutions in the UK, Brazil, Italy, Germany, Russia, Switzerland, Poland, and Venezuela.

6 Project Plan: Deliverables, Milestones, & Assessment

Planning a software project of this size over 4 years requires incorporation of feedback from users and collaboration with multiple independent groups. Exact details will necessarily evolve. We thus describe the general process we expect to play out over the course of this project.

Software Engineering and Deliverables: Our research program (Section 3) includes specific deliverables from both the DIANA team and from collaborations with other groups. These are summarized in the Management and Coordination Plan with institutional responsibilities and a basic timeline. Software is "delivered" not only because code has been written, but after undergoing an iterative software engineering process that tests, validates, documents and verifies that the characteristics meet the requirements of a user community. The software engineering process for DIANA builds on the PIs experience from large HEP experiments (BaBar, CMS, etc.), OSG and ROOT itself, which have successfully engaged thousands of users and developers over periods of 10-20 years. Standard technical tools will be used, including an automated code integration and testing system, a bug reporting system, forums for discussion, project web pages, git/github.com, etc. These will likely be built on the existing systems used by ROOT, CMS, etc. The more difficult aspect for this type of project is obtaining continuous feedback from a large, diverse community. The core philosophy is "Release early, release often" [41], with an agile development model. We will use a combination of nightly integration builds, monthly "development" releases and twice per year "production" releases to engage and integrate feedback from users continuously. (See also Section 5 and the Management and Coordination Plan.) This research program will create several software artifacts and incorporate them into larger projects; this will be done under the GNU Lesser General Public License (LGPL) [42] where possible. Contributions to open source projects with

with LGPL-incompatible licenses will use the project's preferred open source license.

Milestones: The underlying heartbeat for this process will be the release cycle. The experiments in general adopt "production" software releases for widespread use. Thus key milestones will be "delivery" of software (as defined above) into production releases. Figure 2 shows a rough timeline of deliverables (labels correspond to a table in the Management and Coordination Plan). Exact dates for production releases of ROOT and other elements will be agreed with our partners early in the project and revisited periodically.

Metrics: Measuring the success of DIANA requires clear metrics. Performance improvements are the most straightforward. DIANA will work with users and experiments in Year 1 to establish a realistic benchmark set for CPU and IO bound analysis activities. These will be included with the software itself and used to measure progress over time. Adoption of releases where DIANA has made core contributions can be done in two ways. Adoption by HEP and nuclear experiments will be done by simple survey. Adoption by individual users can be measured by monitoring software downloads, broken down by release version. To measure increased use of the interoperability options we provide, we will offer and measure downloads of individual ROOT components (e.g. RIO, RooFit/RooStats, rootpy, the binding toolkit for Apache MapReduce and Spark) independently from the full ROOT distribution. Collaborative Analysis will require a number of metrics: for example, we will work with HepData to build standard tools to upload data products and arrange at the same time to gather statistics about the uploads. Lastly, the impact of DIANA can also be measured from the source code repositories, both as lines of code and as new packages added to the ROOT distribution.

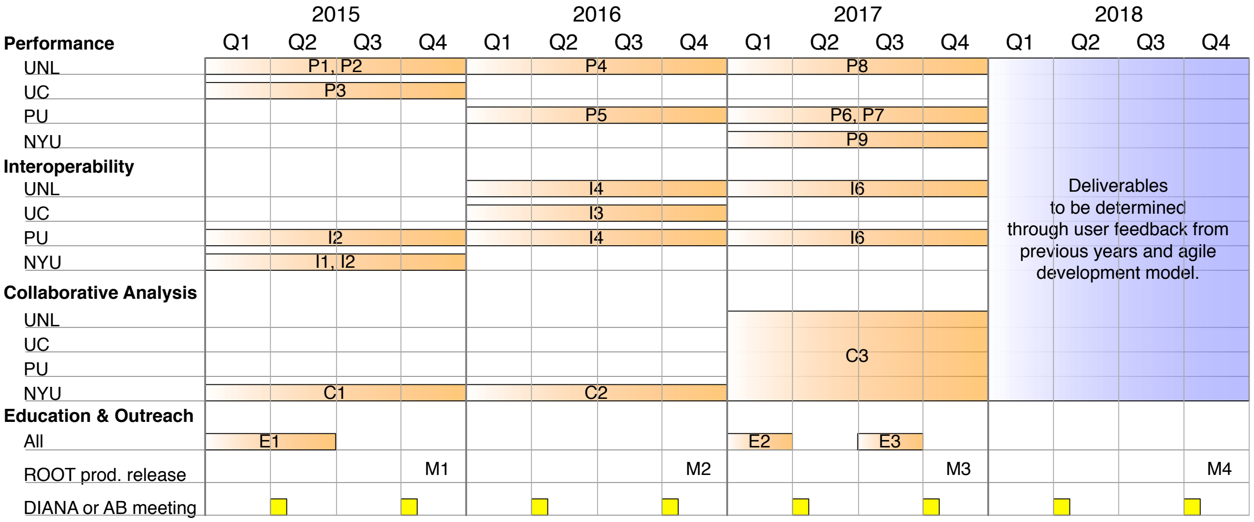

Figure 2: Activities associated to specific deliverables for DIANA; activity bars are indicative of the rough timeline for accomplishing each item, actual delivery will be accomplished through agile development using a monthly development cycle.

Assessing Sustainability Metrics & A Community Engagement Survey: DIANA will also contribute to R&D on software sustainability metrics. Sustainable software is inextricably linked to community engagement. A key metric for sustainability is the ratio of external community contributors to the core internal ones. Accordingly, we will track code contributions, bug reports, and feature requests from community and core developers. We will design and conduct community engagement surveys in Years 1 and 3 to understand the attitudes and incentives associated to community contributions and career paths in scientific software development. The results of this survey will be cross-correlated with actual behavior, traditional metrics like citation, and "altmetrics" associated to data and code as research products. The survey design will be coordinated with the NYU Moore-Sloan Data Science Environment. The results will be documented in Year 4.

A rise in citations by the HEP community to libraries from the broader scientific ecosystem, or to DIANA libraries from outside of HEP, will indicate success for our interoperability efforts. For example, a literature search in INSPIRE shows that the primary document for the ROOT-based TMVA machine learning library has 367 citations, while scikit-learn only has 5 – though outside of HEP scikit-learn is widely cited. Similarly, a rise in the number of code and data as research products given DOIs, and citations to those products, will indicate success for our collaborative analysis activities.

Advisory Board: We propose to create an Advisory Board (AB) to help DIANA meet its goals. The AB will ensure that the efforts of DIANA are 1) aligned with the needs of our core community, 2) directed towards improved interoperability with the broader scientific software ecosystem, and 3) are enhancing the practices of sustainable scientific software development. The AB membership and role is described in the project management plan. Through yearly meetings, the Advisory Board will provide a broader perspective and specific feedback on our progress and planning.

7 Synergies with Already Funded Activities

Cranmer currently serves as co-convener of the ATLAS statistics forum, and is a core developer of the RooStats project. These connection ensure that the community contributions to the primary statistics libraries will be coordinated with the long-term vision of DIANA. He also serves as an Advisory board member to both INSPIRE (the literature system for HEP) and HepData (a HEP-specific data repository). These connections will lead to a cohesive vision for how the analysis tools interface with both the literature system and the primary data repository.

Cranmer is a member of the ICFA study group on Data Preservation in High Energy Physics (DPHEP) and participates in the DASPOS project. The DASPOS effort is focused on exploration, prototyping, and experimentation of strategies for the various elements of preservation. We anticipate that the findings of the DASPOS exploratory efforts will evolve into specific recommendations for implementation within the context of the DIANA effort. Cranmer also serves as the co-lead of the Reproducibility & Open Science working group of the Moore-Sloan Data Science Environment (M-S DSE) at NYU. That group's efforts will inform the HEP-specific work described in Section 3, and our work will provide concrete examples for M-S DSE to consider.

Bockelman currently serves on the USCMS Project Execution Team and on the OSG executive team. As the Technology Area Coordinator for the OSG, he manages the teams responsible for the day-to-day maintenance of the software stack and guiding the medium-term evolution of the OSG's technology platform. Further, the OSG software distribution is a good example of work to keep software artifacts sustainable: we've had to maintain a build and test platform, keep a predictable release schedule, and support a wide, diverse community of more than 100 sites.

Elmer serves as the CMS Deputy Offline Software Coordinator. He will work in a complementary fashion within CMS (with 3500 collaborators) to validate and provide feedback on the integration of the new and evolved tools with the CMS software stack, as well as engage specific individuals and groups doing analysis. He also serves on the USCMS Project Execution Team as co-responsible (with Bockelman) for the Technologies and Upgrades area, which includes the small, but very important, USCMS-funded effort for improving ROOT (0.25 FTE, Philippe Canal/FNAL).

Cincinnati enjoys NSF support for its flavor physics research and its development of GPU-friendly algorithms that run efficiently on many-/multi-core architectures. Much of the proposed works complements the already-funded work. For example, where our PIF grant provides resources to extend and develop new algorithms for emerging architectures, DIANA will provide resources to deploy them as part of ROOT. With respect to our work on LHCb, our service responsibility in core computing is maintaining the conditions databases. As DIANA improves ROOT's RIO library, we will incorporate the new file formats for use by the experiment. Similarly, we will encourage use of DIANA's tools for visualizing ROOT file formats to optimize the design of LHCb's "data summary tape" (DST) formats. These range from "full" DST for storing all raw and reconstructed data to "micro" for storing raw data and a limited set of reconstructed objects to a proposed "turbo" format with only some reconstructed data for a limited number of tracks in the selected events. In terms of collaborative analysis, we will teach the LHCb experiment how to use RooStats *workspaces* to share data with other experiments. Already, for example, LHCb and CMS are working together to combine their analyses of $B_{s,(d)} \to \mu^+\mu^-$ to most powerfully measure the rates of these decays.

8 Summary

High quality, sustainable software is key to making break-through scientific discoveries with the exabytes of data HEP experiments will record in the next decade. It allows teams to work together productively. DIANA will build more powerful analysis tools which provide sophisticated and efficient data reduction techniques, methods for statistical combinations of multiple data sets, and a high-level framework for collaborative efforts crossing the boundaries of individual experiments. We will exploit the improved performance promised by emerging architectures, though rarely achieved. We will work inside the ROOT umbrella as this is the *de facto* home for analysis tools in our field, but we will reposition these key libraries to better interoperate with the larger scientific software ecosystem and we will add new functionality.

The PI's have had and continue to have leadership roles in many complementary software development efforts. The tools we provide will change how our community does data analysis, with new processor architectures, new software-enabled techniques, and new opportunities. Engaging the full community and a broad mix of expertise is required. This is beyond the means for any single PI, but the DIANA team together has the right synergies to accomplish these goals. We have assembled a highly qualified Advisory Board to keep us focused on our goals: tools that are aligned to the needs of HEP community, that interoperate well with the broader scientific software ecosystem, and that are sustainable.

We will also provide training for the next generation of data scientists. The ROOT-based data analysis units we develop for high school students will introduce them to the field. The 20 DIANA Fellows we support and the 2 Ph.D. students we mentor will gain valuable experience working at the interface with physics. The 4 post-doctoral data scientists working on the DIANA team will define and lead the evolution of ROOT so it is ready for the next generation of experiments. The community training we provide will ensure that the tools we develop are used effectively.

In conclusion, the sustainable software we develop will explicitly provide the performance and sophistication required for experiments at the LHC to search for and (we hope) to discover BSM physics. It will also enable other HEP and nuclear physics experiments to fully exploit their datasets. We will train our own cohort group of data scientists, and we will collaborate with others to form a coherent community of developers and users. The science excites us. We're ready to go.

References

[1] Samuel H. Fuller and Editors; Committee on Sustaining Growth in Computing Performance; National Research Council Lynette I. Millett. *The Future of Computing Performance: Game Over or Next Level?* The National Academies Press, 2011.

[2] L. A. T. Bauerdick, S. Gottlieb, G. Bell, K. Bloom, T. Blum, D. Brown, M. Butler, A. Connolly, E. Cormier, P. Elmer, M. Ernst, I. Fisk, G. Fuller, R. Gerber, S. Habib, M. Hildreth, S. Hoeche, D. Holmgren, C. Joshi, A. Mezzacappa, R. Mount, R. Pordes, B. Rebel, L. Reina, M. C. Sanchez, J. Shank, P. Spentzouris, A. Szalay, R. Van de Water, M. Wobisch, and S. Wolbers. Planning the Future of U.S. Particle Physics (Snowmass 2013): Chapter 9: Computing. *ArXiv e-prints*, January 2014.

[3] Avery, Paul and Habib, Salman (co-Chairs) and Others. Computing in High Energy Physics. `http://science.energy.gov/~/media/hep/pdf/files/Banner%20PDFs/Computing_Meeting_Report_final.pdf`, March 2014.

[4] M. Butler, R. Mount, and M. Hildreth. Snowmass 2013 Computing Frontier Storage and Data Management. *ArXiv e-prints*, November 2013.

[5] Fons Rademakers and Rene Brun. Root: an object-oriented data analysis framework. *Linux J.*, page 6.

[6] G. Aad et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys.Lett.*, B716:1–29, 2012.

[7] Serguei Chatrchyan et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys.Lett.*, B716:30–61, 2012.

[8] Heavy Flavor Averaging Group home page. `http://www.slac.stanford.edu/xorg/hfag/index.html`.

[9] J. Charles et al. CP violation and the CKM matrix: Assessing the impact of the asymmetric B factories. *Eur.Phys.J.*, C41:1–131, 2005.

[10] CKM Fitter home page, updated results and plots available at. `http://continuum.io/blog/blaze`.

[11] Marco Ciuchini, G. D'Agostini, E. Franco, V. Lubicz, G. Martinelli, et al. 2000 CKM triangle analysis: A Critical review with updated experimental inputs and theoretical parameters. *JHEP*, 0107:013, 2001.

[12] web site of the UT$_{fit}$ Collaboration. `http://www.utfit.org/UTfit/`.

[13] Gino Isidori, Yosef Nir, and Gilad Perez. Flavor Physics Constraints for Physics Beyond the Standard Model. *Ann.Rev.Nucl.Part.Sci.*, 60:355, 2010.

[14] Particle Physics Project Prioritization Panel. Building for Discovery: Strategic Plan for U.S. Particle Physics in the Global Context. `http://science.energy.gov/~/media/hep/hepap/pdf/May%202014/FINAL_DRAFT2_P5Report_WEB_052114.pdf`.

[15] ROOT home page. `http://root.cern.ch/drupal/`.

[16] Interview with Rene Brun. `http://ph-news.web.cern.ch/content/interview-rene-brun`.

[17] Daniel S. Katz, Sou-Cheng T. Choi, Hilmar Lapp, Ketan Maheshwari, Frank Löffler, Matthew Turk, Marcus D. Hanwell, Nancy Wilkins-Diehr, James Hetherington, James Howison, Shel Swenson, Gabrielle Allen, Anne C. Elster, G. Bruce Berriman, and Colin C. Venters. Summary of the first workshop on sustainable software for science: Practice and experiences (wssspe1). *CoRR*, abs/1404.7414, 2014.

[18] P. del Amo Sanchez et al. Measurement of $D^0 - \overline{D}^0$ mixing parameters using $D^0 \to K^0_S\pi^+\pi^-$ and $D^0 \to K^0_S K^+K^-$ decays. *Phys.Rev.Lett.*, 105:081803, 2010.

[19] R.E. Andreassen et al. Implementation of a Thread-Parallel, GPU-Friendly Function Evaluation Library. *IEEE Access*, 2, 2014.

[20] Wouter Verkerke and David P. Kirkby. The RooFit toolkit for data modeling. *eConf*, C0303241:MOLT007, 2003.

[21] RooFit home page. `https://root.cern.ch/drupal/content/roofit`.

[22] M J Kortelainen, P Elmer, G Eulisse, V Innocente, C D Jones, and L Tuura. The evolution of cms software performance studies. *Journal of Physics: Conference Series*, 331(4):042013, 2011.

[23] Andreas Hoecker, Peter Speckmayer, Joerg Stelzer, Jan Therhaag, Eckhard von Toerne, and Helge Voss. TMVA: Toolkit for Multivariate Data Analysis. *PoS*, ACAT:040, 2007.

[24] SystemTap Home Page. `https://sourceware.org/systemtap/wiki`.

[25] rootpy home page. `http://www.rootpy.org`.

[26] Blaze home page. `http://continuum.io/blog/blaze`.

[27] Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1):65–80, 2010.

[28] Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. emcee: The mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):pp. 306–312, 2013.

[29] Louis B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science*. Springer, Berlin, 1981.

[30] S. Kraml, B.C. Allanach, M. Mangano, H.B. Prosper, S. Sekmen, et al. Searches for New Physics: Les Houches Recommendations for the Presentation of LHC Results. *Eur.Phys.J.*, C72:1976, 2012.

[31] F. Boudjema, G. Cacciapaglia, K. Cranmer, G. Dissertori, A. Deandrea, et al. On the presentation of the LHC Higgs Results. 2013.

[32] ATLAS Collaboration. Data from Figure 7 from: Measurements of Higgs boson production and couplings in diboson final states with the ATLAS detector at the LHC. 10.7484/INSPIRE-HEP.DATA.RF5P.6M3K.

[33] ATLAS Collaboration. Data from Figure 7 from: Measurements of Higgs boson production and couplings in diboson final states with the ATLAS detector at the LHC. 10.7484/INSPIRE-HEP.DATA.26B4.TY5F.

[34] ATLAS Collaboration. Data from Figure 7 from: Measurements of Higgs boson production and couplings in diboson final states with the ATLAS detector at the LHC. 10.7484/INSPIRE-HEP.DATA.A78C.HK44.

[35] Jeremy Bernon. `https://indico.cern.ch/event/313725/session/1/contribution/21/4/material/slides/0.pdf`.

[36] Sabine Kraml. `https://indico.cern.ch/event/313725/session/1/contribution/22/0/material/slides/0.pdf`.

[37] R.W. Conway and D. Gries. *An Introduction to Programming: A Structured Approach Using PL-1 and PL-C.* Little, Brown, third edition, 1979.

[38] International Particle Physics Outreach Group home page. `http://ippog.web.cern.ch/`.

[39] QuarkNet home page. `http://quarknet.fnal.gov/`.

[40] S. Malik, F. Hoehle, K. Lassila-Perini, A. Hinzmann, R. Wolf, et al. Maintaining and improving of the training program on the analysis software in CMS. *J.Phys.Conf.Ser.*, 396:062013, 2012.

[41] Eric S. Raymond. *The Cathedral and the Bazaar.* O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1st edition, 1999.

[42] GNU Lesser General Public License, version 2.1. `https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html`.