Center for
HIGH PERFORMANCE COMPUTING
THE UNIVERSITY OF UTAH

Fall 2022

Improving the Quality of Healthcare Using Big-Data Analytics

Joshua J Horns, Rupam Das, Niraj Paudel, Nathan Driggs, Rebecca Lefevre, Jim Hotaling, Benjamin Brooke,
Department of Surgery, University of Utah Health

Over the last decade, healthcare research has increasingly used big-data to understand patterns of disease, efficacy of treatments, healthcare inequities, and much more. Large healthcare databases have unparalleled potential to address questions across broad segments of the population to understand rare conditions that only occur occasionally at any single institution. The Surgical Population Analysis Research Core (SPARC) is a team of data scientists, statisticians, and surgeon-scientists dedicated to improving the quality of healthcare through population-health research using big-data analytics. SPARC was founded in 2018 at the University of Utah and since that time has produced over 200 papers and abstracts and received $13,000,000 in grant funding.

Large healthcare datasets can include hundreds of millions of individuals and consist of hundreds of trillions of datapoints. Such an abundance of data requires an efficient, powerful computing center to house, curate, and analyze the data. The Center for High Performance Computing (CHPC) at the University of Utah has allowed SPARC to conduct exciting, novel research with important and immediate implications to patient care. SPARC, in partnership with CHPC, has developed extensive expertise in a number of healthcare data resources allowing the group to answer a wide variety of clinical questions: from social inequities in access to emergency surgical care, to how providers can reduce chances of complications in women giving birth.

A large proportion of SPARC's research is conducted using the IBM MarketScan database. This resource combines health insurance records for approximately 160 million people across the United States, allowing for the tracking of the progression of disease and chart patient recovery after an injury. Because MarketScan includes healthcare cost information, we can also study where patients experience the largest financial burdens. Below we highlight two recent examples of our work with MarketScan.

Healthcare Burden in Children with Anorectal Malformations

Anorectal malformations (ARM) are rare congenital anomalies that result in life-long functional impairment. ARMs are uncommon, only occurring 2-5 times in every 10,000 births, which can make it a difficult patient population to study. Surgical interventions to address ARMs, as well as on-going care afterward, can result in a significant financial burden to patients and their families and many days spent in care. Unfortunately, before this year no research team had been able to comprehensively study patterns of healthcare use and costs in these patients. Understanding how and why ARM patients are interacting with the healthcare system is the first step in creating programs designed to alleviate the financial strain these families experience.

In 2022, SPARC worked on a project led by Dr. Michael Rollins of Primary Children's Hospital attempting to quantify the number of days children with ARMs spend in the hospital and how much cost the families incur. Thanks to the computational resources at CHPC, we were able to search through over 7 trillion diagnoses from across the United States to identify 664 children born with an ARM. We followed these patients for the first five years of life and noted every hospitalization and clinic visit they made during that time as well as their associated costs.

Because of this work, we were able to estimate that children born with an ARM will, on average, spend nearly 20% of their first year of life in the hospital and ultimately make 158 healthcare visits by age five. Financially, these encounters totaled $273,000 representing an enormous economic strain on families. Due to the detailed nature of the data, we identified the first year of life as the principal time in which patients underwent surgery to address their ARM. However, we discovered that even after surgery, ARM patients were making nearly four times as many healthcare visits as children without ARMs and that this increase was particularly large in patients with more complex ARM phenotypes (Figure 1). These results help us identify patterns of healthcare use at various stages of life in an attempt to create more efficient means of delivering care to children with ARMs, thus lowering the economic burden on families.

Balancing Hospital Overcrowding and Revenue During COVID-19

In 2020, healthcare systems in the United States and around the world experienced enormous surges in hospitalizations due to the rapid spread of severe COVID-19 infections. Most hospitals suspended all elective surgical procedures in the hopes of increasing patient capacity. However, such surgeries are important, and often primary, sources of revenue for hospitals. Suspending elective surgeries may increase space to treat COVID patients, but if revenue loss becomes too substantial many facilities may not be able to recover financially and be forced to close, thus reducing overall access to healthcare. To compare the benefits in capacity-gain from canceling elective surgeries versus the economic strain it places on hospitals, SPARC conducted a nation-wide analysis of all elective surgeries to quantify the number of days these patients spent in care and the amount of money these procedures generated.

We identified over 3 billion elective surgery cases which collectively resulted in 430,000 bed-use days every month and a total revenue of $1.1 trillion. We broke down these patterns by type of surgery (i.e., those associated with neurological issues, childbirth, etc.) to better understand the types of care that would be most impacted by canceling elective surgeries (Figure 2). Though much research had been conducted during the onset of COVID into increasing hospital capacity, this was the first study to explore the balance between capacity gains and revenue loss. These data have since been used in dozens of other studies to understand the impact of COVID-19 on patients seeking specific types of care.

As SPARC continues its mission to improve healthcare by producing high-quality, novel research, we rely heavily on the data management and computer science expertise offered by CHPC. SPARC is always open to external collaboration. For more information and to get in touch, please see *https://medicine.utah.edu/surgery/research/cores/sparc*.

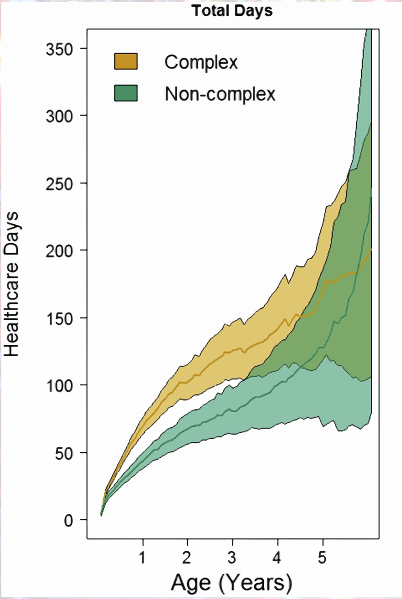


Figure 1: Average accumulated days spent interacting with the healthcare system for children with ARMs. Yellow line denotes children with more complex ARM phenotypes, green line denotes children with less complex phenotypes. Shaded regions are 95% confidence intervals. (From Rollins et al, 2021. Healthcare burden and cost in children with anorectal malformations during the first 5 years of life. The Journal of Pediatrics, 240, 122-128)

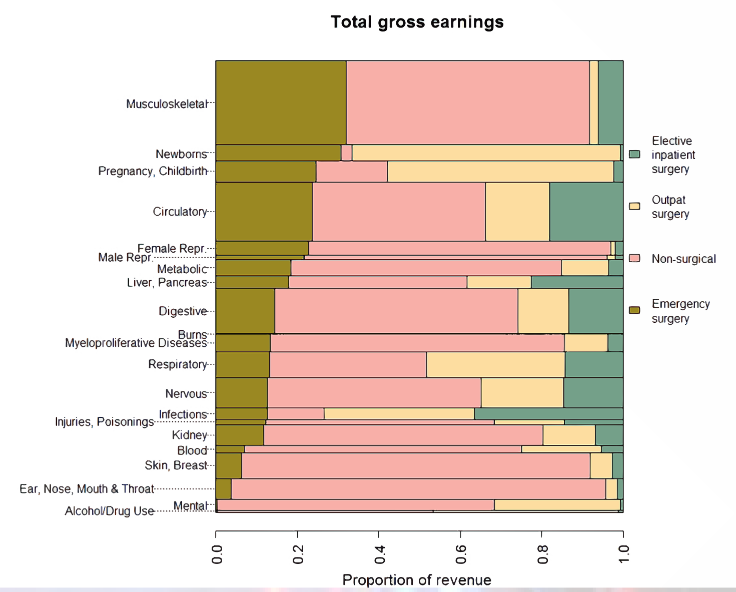


Figure 2: Financial contribution of major diagnostic categories (MDC) to gross hospital revenue. Levels are listed in descending order the percentage of each MDC category contributed by elective inpatient cases. Level width is proportional to the absolute value in US dollars. (Adapted from Tonna et al. 2020. Balancing revenue generation with capacity generation: Case distribution, financial impact and hospital capacity changes from cancelling or resuming elective surgeries in the US during COVID-19. BMC Health Services Research 20(1119)

Conda & Reproducibility

Wim Cardoen, CHPC Scientific Consultant

Conda environments can be used to make python software installations reproducible, i.e., in order to generate python environments which bear identical version numbers for their python packages. In this article, we will describe the process to set up such an environment.

As a first step, we will create an environment using the latest version of miniconda3 installed on a machine running the Ubuntu 18.04 Operating System (OS). Subsequently, the settings of this newly created conda environment will be exported into yaml file. After this we will load the settings in a new conda environment on a CHPC node running the Rocky8 OS.

Note that miniconda3 can be exchanged by anaconda3 in the process described in this article.

Creation of the Reproducible Environment

We assume that miniconda3 has been installed and has been set up in such a way that allows for the support of more than one conda environment (the default conda environment bears the name `base`). For installation instructions, see CHPC's *User Installed Python* help article.

The most common way to support multiple conda environments is to invoke `conda init` post the miniconda3 installation. Unfortunately, this procedure modifies a user's existing startup shell `.bashrc/.tcshrc` on a **permanent** basis, as it appends a block of shell code in the startup shell and forces the latest miniconda3 installation to become the default. At best, this approach may serve an individual user, but it is not apt for CHPC's cluster environment.

Therefore, we strongly recommend *CHPC's lua module template* which does not modify the startup shell and allows for both the support of multiple conda environments, allowing the user flexibility to maintain different miniconda distributions simultaneously.

In the following coding block, we create an environment that we are naming `genscience`. The command `conda activate genscience` allows one to enter the `genscience` environment. Subsequently an array of Python packages (`numpy`,..., `statsmodels`) as well as `texlive-core` will be installed in this `genscience` environment. The command `conda deactivate` forces one to leave the `genscience` environment and return to the `base` environment.

```
# Create a simple env: genscience
module load py39_4.12.0

# Create a simple env: genscience
conda create -y -n genscience

# Activate the genscience environment
conda activate genscience

#Install an array of packages
conda install -y -c anaconda numpy scipy matplotlib
conda install -y -c anaconda pandas scikit-learn scikit-learn-intelex
conda install -y -c anaconda jupyter
conda install -y -c anaconda xarray
conda install -y -c bokeh bokeh
conda install -y -c conda-forge dask
conda install -y -c conda-forge gdal
conda install -y -c conda-forge jupyterlab voila
conda install -y -c conda-forge jupyterlab-latex
conda install -y -c conda-forge scitkit-image
conda install -y -c conda-forge statsmodels
conda install -y -c conda-forge texlive-core

# Deactivate the genscience environment
conda deactivate
```

The command `conda list` displays the environments that are accessible to the miniconda3 installation. The symbol '*' is prepended to the name of the directory where the currently activated environment is stored.

```
#Conda List:
# ----------
conda list

#Find all the conda envs:
# ----------------------
base                  *  /home/sleipnir/software/pkg/mini3/py39_4.12.0
genscience               /home/sleipnir/software/pkg/mini3/py39_4.12.0/envs/
    genscience

sleipnir@ragnarok:~$ conda activate genscience
(genscience) sleipnir@ragnarok:~$ conda env list
# conda environments:
#
base                     /home/sleipnir/software/pkg/mini3/py39_4.12.0
genscience          * /home/sleipnir/software/pkg/mini3/py39_4.12.0/envs/
conda deactivate
```

Export of a Conda Environment

The information of the **genscience** environment can be easily stored in a file. The following command stores the details of the **genscience** Conda environment in the yaml file **genscience.py39.yml**.

```
# Export the genscience env into a YAML file:
# ------------------------------------------
conda env export -v -n genscience -f genscience.py39.yml
```

Reproduce an Environement Based on a YAML file

In what follows we will recreate the **genscience** environment on a different computer. In this case we will be taking the environment created on a system running an Ubuntu operating system and running it on a CHPC cluster node which is running a RockyLinux8 operating system.

After loading an existing anaconda distribution we are able to generate a Python environment based on the previously exported yaml file.

```
module use ~/EigenModules
module load myanaconda/2020.11

[u0253283@kingspeak5:~]$ conda env list
# conda environments:
#
base                     * /uufs/chpc.utah.edu/common/home/u0253283/software/pkg/anaconda3/2020.11

conda env create -n genscience -f $HOME/genscience.py39.yml

(genscience) [u0253283@kingspeak5:~]$ python3
Python 3.10.4 (main, Mar 31 2022, 08:41:55) [GCC 7.5.0] on linux
Type “help”, “copyright”, “credits” or “license” for more information.
>>> import numpy as np
>>> np.__version__
’1.22.3 ’
>>> import statsmodels
>>> statsmodels.__version__
’0.13.2 ’
```

XSEDE is now ACCESS: The XSEDE (Extreme Science and Engineering Discovery Environment) program ended Aug 31, 2022 and has been replaced by the new ACCESS (Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support) program. The computational resources have not changed – this transition only impacts the support and access to the resources. Visit the ACCESS website, https://access-ci.org/, for details about the new program.

Singularity Container Runtime is Becoming Apptainer

Martin Cuma, CHPC Scientific Consultant

Recently, it became popular and common place to deploy complex scientific applications using containers. Containers are a lightweight alternative to Virtual Machines (VMs), packaging the operation system (OS) and its software stack, while relying on the system's host OS for lower level functionality, such as the kernel and hardware drivers. Containers thus allow to run applications built with different OS on a host OS.

The most common container system is Docker, however, its focus is on virtualizing servers, and Docker is more troublesome to use in the research computing (RC) environment. And, most importantly, Docker's execution model has security implications in a shared user environment that the RC institutions run. To address the RC limitations of Docker, the Singularity container runtime was developed and became popular. In particular, running a container with Singularity maps the host file systems, user name and the environment, providing a very similar computing environment inside of the container as compared to the environment of the host. Singularity also relies on different security requirements when launching the container reducing the security concerns. Singularity also has a capability of running Docker built containers, and as such it is used by both CHPC support staff to install programs, and by CHPC users to install and run programs specific for their research. Use of Singularity at CHPC is documented at *https://www.chpc.utah.edu/documentation/software/singularity.php*.

Singularity has had a somewhat bumpy evolution over the years as the original developers sought funding for further development and struggled with keeping its code base open source. As a result of that, the Singularity brand has a corporate ownership, and open source advocates moved the project under the Linux Foundation and renamed it to Apptainer in a public announcement at the end of November 2021.

Since then, CHPC has been evaluating Apptainer and is ready to start moving to it from Singularity use. We are installing Apptainer updates as they are published and will continue supporting it. We are also considering supporting SingularityCE, the Community Edition of Singularity that is developed by SyLabs, since it retains the remote container build functionality that allows our users to build containers on CHPC systems. This feature has been removed from Apptainer.

For our users, the main difference is that the **`singularity`** module now becomes the **`apptainer`** module. The Apptainer distribution includes both the **`singularity`** and **`apptainer`** commands, therefore existing users of Singularity can simply load the **`apptainer`** module and keep using the **`singularity`** command. However, we recommend to start changing the workflows to use the **`apptainer`** command instead, as the **`singularity`** command may be removed from Apptainer in the future. From now on we will be setting up new containers that supply CHPC installed programs with the **`apptainer`** command.

As always, please report any issues you may see with Apptainer to the CHPC help desk, *helpdesk@chpc.utah.edu*. Being under the Linux Foundation, we hope that the Apptainer name and support will be stable for years to come.

Hands on Introduction to R

Wim Cardoen, CHPC Scientific Consultant

In an effort to improve the CHPC presentation schedule we are pleased to announce that we are expanding the R training session offered. With the increased interest in the use of R on CHPC resources we are moving from a single 2 hour presentation, to a series of presentations in line with the CHPC offerings for our introduction to Linux.

In the expanded version of the R training, we will cover some basic building blocks of the R programming language. We will start with some historical info on the R language. We then address the concept of the atomic data types and homogeneous vectors. Through the concept of attributes we will be able to introduce matrices, arrays, factors and datetimes. Subsequently, control structures and the concept of functions will be covered. In a subsequent section, the topic of heterogeneous vectors (list and dataframes) and IO will be addressed. If time permits we will discuss the concept of environments, libraries and a few statistical distributions. At the present time, ggplot2, debugging, profiling will not be covered. They certainly will be discussed in the near future. After each section time will be spent on exercises.

For more information about the expanded presentation series, see https://www.chpc.utah.edu/presentations/IntroR.php. Additional information on working with the R installation on the CHPC linux clusters, including information on how to install additional R packages, can be found at https://www.chpc.utah.edu/documentation/software/r-language.php.

CHPC Welcomes New Staff Members

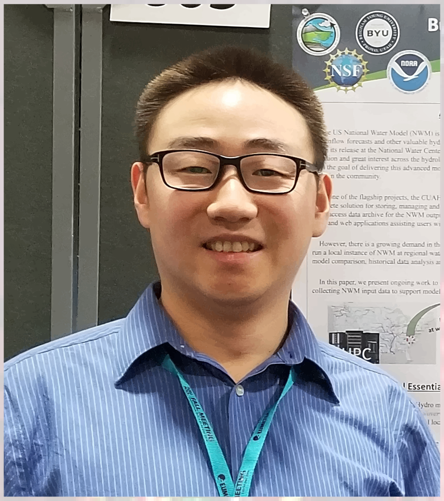
Anita Orendt, CHPC Scientific Consultant

CHPC is pleased to announce two new staff members. These additions to the CHPC staff will allow us to better support both the CHPC resources and to better provide support for our userbase. Please join us in welcoming Ben and Drew to CHPC.

Ben Dewey - System Administrator

Ben Dewey joined the CHPC staff as a system administrator this summer. Ben has been interested in working with computers since he built his first computer while in middle school. He has been taking both Computer Science and Electrical Engineering classes at the U. Previous to coming to CHPC, Ben worked as a system administrator at Conduent as well as working in IT at Ken Garff. Ben enjoys working with and testing new technologies and is excited about his new position with CHPC.

At CHPC Ben will be working with Sam Liston on the CHPC storage systems. Currently his main projects will be administration of the data transfer services such as Globus and the data transfer nodes as well as workion the migration to the Rocky Linux 8 OS on the storage compute infrastructure.



Zhiyu "Drew" Li - Scientific Consultant

Zhiyu Li (aka "Drew") received BS & MS in Geomatics from Wuhan University and PhD in Environmental Sciences from the University of Chinese Academy of Sciences. After graduation, Drew completed post-doc research under Dr. Dan Ames at Brigham Young University where he contributed to the collaborative development of an advanced hydroinformatics system - HydroShare. He then joined the CyberGIS Center (supervisor: Dr. Shaowen Wang) at the University of Illinois at Urbana-Champaign as a Research Programmer and led the development of multiple cyberGIS services and applications using XSEDE/ACCESS resources to facilitate hydrological modelings. Drew is an experienced GIS software developer with a strong connection to the field of hydrology, and he is particularly interested in developing geospatial solutions on top of advanced research cyberinfrastructure (HPC and cloud) for serving the broad water science research community.

Drew will join the existing CHPC Scientific Consulting staff. He will be working fully remote.

Upcoming CHPC downtime - Tuesday, December 6, 2022 starting at 7:30 am

This downtime will impact the general environment clusters of nothpeak, kingspeak and ash. The downtime is to move these clusters from the current to a new infiniband gateway switch. A reservation is in place to empty the slurm batch queues of jobs before the start of the downtime. Watch for an announcement with additional details.

Spring 2023 Presentation Schedule

Anita Orendt, CHPC Scientific Consultant

The spring 2023 presentation schedule has been finalized. After careful consideration, we have decided to continue to do the presentations as remote only, via zoom.

Please note that the presentations are held either 1-2pm or if they are hands-on presentations, marked with the *, they are held 1-3pm. Also note that in the Spring and Summer Semesters, the presentations are held on a Tuesday and Thursday schedule, while in the Fall Semester they are held on Monday, Wednesday, and Friday.

Due to changes with zoom, we have also changed the zoom link for the presentations. Moving forward the zoom link being used for all CHPC presentations is: *https://utah.zoom.us/j/96339929196*.

With the transition from XSEDE to ACCESS (see note on bottom of page 4 of this newsletter), it has been decided that the ACCESS program will host a limited run of the HPC Monthly Workshops developed by the Pittsburgh Supercomputing Center (PSC). The decision was also made to return to the pre-COVID satellite site model for the workshops.

The workshops run from 9am-3pm Mountain time with a 1 hour lunch break. Please note that while there is no cost associated with the workshops, attendees must register in advance. These workshops are hands on, and CHPC will have a staff member in attendance to deal with questions and any possible technical difficulties with the workshop.

The first of these was the "*GPU Programming with OpenACC*" Workshop held on Monday, November 7. Watch for further announcements of future workshops.

DATE	**PRESENTATION**	**PRESENTER(S)**
Thursday, January 19, 2023	Overview of CHPC	Anita Orendt
Tuesday, January 24, 2023	Hands on Introduction to Linux, part 1*	Anita Orendt & Brett Milash
Thursday, January 26, 2023	Hands on Introduction to Linux, part 2*	Anita Orendt & Brett Milash
Tuesday, January 31, 2023	Hands on Introduction to Linux, part 3*	Anita Orendt & Brett Milash
Thursday, February 2, 2023	Module Basics	Anita Orendt
Tuesday, February 7, 2023	Slurm and Slurm Batch Scripts	Anita Orendt
Thursday, February 9, 2023	Hands-on Introduction to Open-On-Demand*	Martin Cuma
Tuesday, February 14, 2023	Hands-On Introduction to Python, part 1*	Brett Milash & Wim Cardoen
Thursday, February 16, 2023	Hands-On Introduction to Python, part 2*	Brett Milash & Wim Cardoen
Tuesday, February 21, 2023	Hands-On Introduction to Python, part 3*	Brett Milash & Wim Cardoen
Thursday, February 23, 2023	Numpy, part 1 (Hands-On Introduction to Python, part 4)*	Wim Cardoen & Brett Milash
Tuesday, February 28, 2023	Numpy, part 1 (Hands-On Introduction to Python, part 5)*	Wim Cardoen & Brett Milash
Thursday, March 2, 2023	Introduction to Parallel Computing*	Martin Cuma
Tuesday, March 14, 2023	Introduction to Containers*	Martin Cuma
Thursday, March 16, 2023	Using Git for Version Control*	Martin Cuma
Tuesday, March 21, 2023	Hands-On Introduction to R, part 1*	Wim Cardoen
Thursday, March 23, 2023	Hands-On Introduction to R, part 2*	Wim Cardoen
Tuesday, March 28, 2023	Hands-On Introduction to R, part 3*	Wim Cardoen

New CHPC Node With AMD GPU

Martin Cuma and Anita Orendt, CHPC Scientific Consultants

Thanks to an AMD donation of a *MI100 Instinct Accelerator*, CHPC now has a compute node available to our users for testing this AMD GPU offering. This accelerator is the generation prior to the AMD Instinct MI250X GPUs found in the new Department of Energy's exascale system *Frontier* deployed earlier this year at Oak Ridge National Laboratory.

The node housing this accelerator has 64 physical CPU cores, in the form of two 32 core AMD 7513 processors, and 512 GB memory. It has been set up as a notchpeak compute node, with a slurm partition `notchpeak-eval` and account `eval`. In order to gain access for testing, please reach out to *helpdesk@chpc.utah.edu*.

The AMD ROCm platform contains the software, drivers, libraries, developer tools needed to make use of this GPU. In depth information on this platform can be found on the *AMD documentation site*. The information at this site includes guides for HIP - Heterogeneous Interface for Portability – which is AMD's dedicated GPU programming environment that can run on both AMD and Nvidia GPUs. To set up the environment to make use of the GPU we have created a module `rocm`.

For faster onboarding, AMD also provides the *Infinity Hub*, a web site which provides instructions on how to run select containerized programs on AMD GPUs. The *Infinity Hub* provides a very simple way to run these programs, including CP2K, GROMACS, LAMMPS, NAMD, PyTorch, Tensorflow or SPECFEM3D. See the Infinity Hub website for a complete list of supported applications.

For example, to run the High Performance Linpack (HPL) benchmark, we simply pull the pre-built container with the HPL from the DockerHub and execute the HPL command inside of the container:

```
$ module load apptainer
$ singularity pull rochpl.sif docker://amdih/rochpl:5.0.5_49
$ singularity run --pwd /tmp --writable-tmpfs rochpl.sif mpirun_rochpl -P 1 -Q 1 -N 64000 --NB 512
…
Final Score:    6.9187e+03 GFLOPS
```

The final HPL score of 6.92 double precision TFLOPS is reasonable in comparison to the theoretical 7.68 TFLOPS, and comparable to the performance of a Nvidia V100 GPU. The current generation MI250 GPU theoretical double precision peak is 26.62 TFLOPS and the MI250x, specific for the Frontier system, is 28.16 TFLOPS.

Please acknowledge the use of CHPC Resources

If you use CHPC computer time or staff resources, we request that you acknowledge this in technical reports, publications, and dissertations. An example of what we ask you to include in your acknowledgement is:

"A grant of computer time from the Center for High Performance Computing is gratefully acknowledged."

If you make use of the CHPC Protected Environment, please also acknowledge the NIH shared instrumentation grant:

"The computational resources used were partially funded by the NIH Shared Instrumentation Grant 1S10OD021644-01A1."

Please submit copies or citations of dissertations, reports, pre-prints, and reprints in which CHPC is acknowledged by sending to *helpdesk@chpc.utah.edu*.

The University of Utah
University Information Technology
Center for High Performance Computing
155 South 1452 East, Room 405
SALT LAKE CITY, UT 84112–0190

Center for
HIGH PERFORMANCE COMPUTING
THE UNIVERSITY OF UTAH