

Spring 2019

Applying Modern Data Analysis to Challenges in Chemistry
 New Policies for Sensitive Data
 Cyberteam Focus Groups
 Managing Python Package Installations with Anaconda
 Upcoming Changes to Backups
 RMACC Symposium

Applying Modern Tools of Data Analysis to Current Challenges in Chemistry

Research Highlight · Jacquelyne Read and Matthew Sigman, Department of Chemistry

What if there were no such thing as “bad data”? In this case, we are not referring to the quality of the data, but the experimental outcome. In the field of asymmetric catalysis, a flourishing area in organic chemistry, a central goal is to develop reactions that are able to form one enantiomer (a chiral molecule which has a non-superimposable mirror image) of product in preference to the other enantiomer possible. A helpful analogy to understand the concept of enantiomers is the right and left hand—mirror images, but not identical. Enantioselective reactions have many important applications, such as the more efficient synthesis of drug molecules, which often need to be made as just one enantiomer because different enantiomers sometimes result in drastically different biological responses. Often, when enantioselective reactions are published, only data meeting or exceeding the gold standard of 95% desired enantiomer to 5% undesired enantiomer are reported. This results in useful, but non-optimal, data never being published.

The Sigman lab (<https://chem.utah.edu/directory/sigman/research-group/>) takes a different approach. We seek to make use of a wider range of data collected during the reaction optimization process because results showing low enantioselectivity and high enantioselectivity can be equally information-rich and help us learn about a given reaction. Our workflow involves collecting molecular properties (such as size, shape, and electronic nature) relevant to a reaction we seek to study. These properties are then used as parameters in a multivariable linear regression algorithm and correlated to the experimentally determined reaction selectivity. The resulting equations are applied to the prediction of molecules that should lead to higher (and sometimes lower) enantioselectivity, which are then synthesized and validated experimentally. Ultimately, a deeper understanding of the reaction can be garnered through analysis of

the statistical model. This workflow has been successfully applied to reaction properties beyond enantioselectivity, such as regioselectivity (where a reaction occurs on a given molecule) and rates of chemical processes, which will be discussed further in the examples below. Central to our technology is the calculation of molecular properties using density functional theory (DFT), which is accomplished using the computational resources of the CHPC, among others.

Applications to Asymmetric Catalysis

As a recent example, the development and analysis of an enantiodivergent, sp^3 - sp^2 Suzuki–Miyaura reaction was aided by our approach (Figure 1) [1]. Previous work on this cross-coupling reaction was recognized by the Nobel Prize in 2010 as a means to change strategies of drug synthesis such that further advances remain of high interest to the chemical community. In our studies, the outcome of

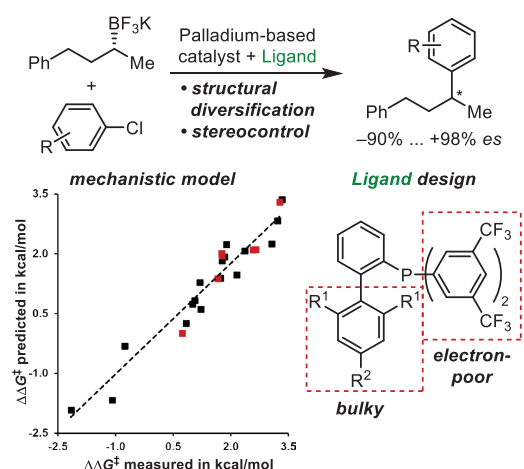


Figure 1. An enantiodivergent, sp^3 - sp^2 Suzuki–Miyaura reaction and a model describing the influence of ligand properties on product selectivity.

the reaction was shown to be strongly dependent on the structure of the phosphine ligand, which binds to the palladium metal during the reaction, used. To further exploit this observation, the structure, spatial arrangement, and electronic properties of 24 phosphine ligands were calculated using the computational resources provided by the CHPC. Through multivariable linear regression analysis of the computed parameters and the experimental data, various properties of the ligands were able to be related to the observed reaction selectivity. In addition, the model obtained through this analysis could be used to predict better ligands, and as a result was thus able to facilitate the development of optimal conditions for this highly selective Suzuki coupling reaction. For a press release on this study, see <https://unews.utah.edu/simulations-enable-choose-your-own-adventure-stereochemistry/>.

Applications to Selectivity in Heck Reactions

We have also studied another class of cross-coupling reactions, redox-relay Heck reactions, the area in which our group conducts most of our experimental research. In these reactions, a group is added into a carbon–carbon double bond, and the catalyst is able to migrate along a carbon chain towards an alcohol group. Central to the success of these transformations is the pyridine-oxazoline (PyrOx) ligand scaffold that can be made chiral, thereby providing a manner for inducing enantioselectivity. Also important is the manner in which these ligands bind to palladium, the metal catalyzing these reactions. Transition states for these reactions have been calculated using DFT to reveal the exact process through which these reactions take place. More recent computational studies have focused on parameterizing the PyrOx ligands that help impart some of this unique reactivity [2]. Structural features of these ligands were chosen to allow the size, shape, and electronic distribution to be reflected in the parameter set. This set of molecular descriptors was then applied to a Heck redox-relay transformation, shown in Figure 2, that generated two types of products. Multivariable linear regression analysis was conducted on this data and the parameter set to better understand why some PyrOx ligands gave different product ratios than others. The successful model, presented in Figure 2, was able to predict additional ligands that performed optimally in this transformation. The parameter set, built from calculations conducted on the CHPC cluster, is a great resource for the chemical community to study many reactions using this common set of ligands. A study we published this year has also used this parameterization methodology to better understand the relative stability of cobalt–PyrOx complexes in an electrochemical setting for applications toward more sustainable catalysis in the future.

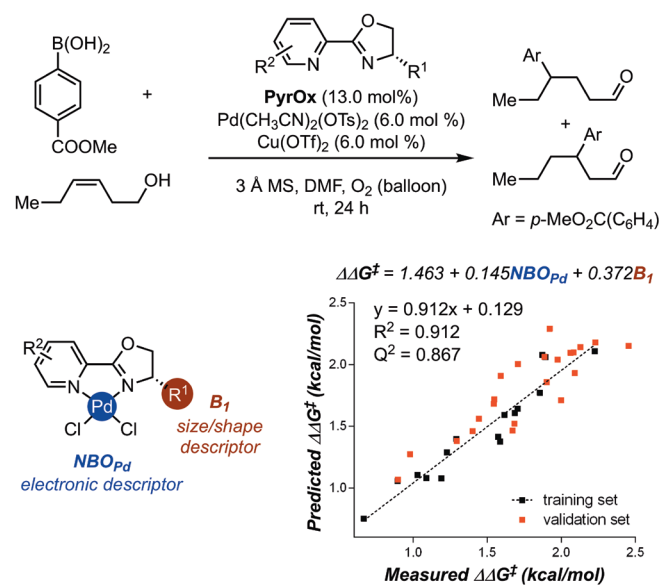


Figure 2. Molecular descriptors used to describe and predict product selectivity in a redox-relay Heck reaction.

Applications to the Development of Flow-Battery Electrolytes

The parameterization and modeling workflow has also proven useful beyond reaction or catalyst development to include important applications such as developing stable and recyclable electrolyte molecules for large-scale, renewable energy storage. Nonaqueous redox-flow batteries have the potential to offer significant improvements in how our society stores and generates energy; however, for these batteries to become widely used, organic molecules must be developed that offer high energy density, high stability and reusability, and low costs. The Sigman group in conjunction with the Minter group, here at Utah, as part of the Joint Center for Energy Storage Research (JCESR), are working to develop solutions to current obstacles in energy storage.

To aid the development of electrolyte molecules that can be used in these systems, we applied our parameterization methodology to a set of twelve acylpyridine radicals. This class of molecules appeared promising for use as electrolytes in flow batteries, but their stability needed to be significantly improved while also maintaining a favorable redox potential [3]. Many structural descriptors of these molecules as well as thermodynamic properties were calculated. The decomposition rates of these acylpyridine radicals were able to be modeled using one physical descriptor along with the electrochemical reduction potential of these molecules (Figure 3). Such a well-fit model not only allowed us to make sense of the trends in how long-lived these intermediates were, but also enabled us to predict significantly better performing acylpyridine radicals. For a press release on this study, see <https://unews.utah.edu/stabilizing-energy-storage/>.

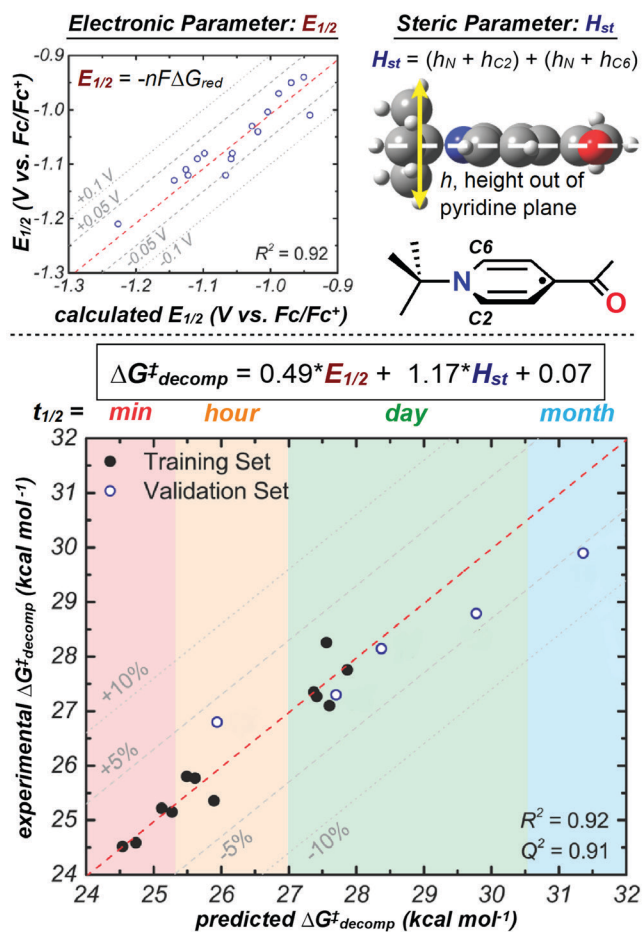


Figure 3. A model developed to predict the stability of flow-battery electrolytes.

The success of our parameterization methods in this system was important to demonstrate the utility of the methodology beyond catalysis. We are further developing this tool set as we tackle new chemical systems, and we have also begun using more sophisticated machine-learning algorithms to analyze larger and more complex data sets.

References

- [1] Zhao, S.; Gensch, T.; Murray, B.; Niemeyer, Z. L.; Sigman, M. S.; Biscoe, M. R. *Science* **2018**, *362*, 670–674.
- [2] Guo, J.-Y.; Minko, Y.; Santiago, C. B.; Sigman, M. S. *ACS Catal.* **2017**, *7*, 4144–4151.
- [3] Sevov, C. S.; Hickey, D. P.; Cook, M. E.; Robinson, S. G.; Barnett, S.; Minter, S. D.; Sigman, M. S.; Sanford, M. S. *J. Am. Chem. Soc.* **2017**, *139*, 2924–2927.

New Policies Governing the Protection of Sensitive Data

Policy · Tom Cheatham

In the new age of concern over privacy, and given the frequency of data breaches, many organizations have tightened their security posture, and the U is no different. However, often the policies that are implemented are very difficult

to enforce and people are unaware of what it requires to be compliant. This holds true for the policies governing the management of data. At the November 2018 Regents meeting, in discussions with the Utah System of Higher Education, the R345 “Information Technology Resource Security” policy (<https://higheredutah.org/policies/policy345/>) was passed. The biggest change is that sensitive data = restricted data.

Previously the U’s IT policy committee, as per Policy 4-004 (<https://regulations.utah.edu/it/4-004.php>), specifically section C dealing with Data Classification and Encryption, provided a clear distinction between the requirements for the handling of sensitive versus restricted data, the latter of which must be encrypted both at rest and in motion. At the U, sensitive data is

- FERPA
- student information (including recommendation letters and grades)
- intellectual property
- employee information
- current litigation materials
- contracts
- building and utilities documentation

Restricted data includes

- personally identifiable information (PII)
- protected health information (PHI)
- payment card information (PCI)
- financial information
- donor information

PII includes social security number, driver’s license or state ID number, passport number, visa number, alien registration number, fingerprints, other biometrics, as well as full name in combination with mother’s maiden name or date of birth or last four digits of SSN or citizenship or immigration status or ethnic or religious affiliation.

Under the new policy the same encryption at rest and in motion requirements for restricted data are now required for sensitive data at the U. To address this new policy, CHPC will employ self-encrypting drives in future storage purchases. The treatment of legacy gear is at this time unclear, but it is likely that the university rules will apply to only new equipment.

In addition to encryption, additional appropriate controls need to be implemented; some examples are the use of two-factor authentication, evaluation of any third parties given access to the data, and an inventory of active devices.

The rule also states that sensitive or restricted data should not be stored in personal devices (i.e., non-university owned) unless the user needs such data on the device in order to perform duties necessary to conduct business of the university. Any use of personal devices for such

data requires not only encryption of these devices but also approval from the appropriate Dean, Department Chair or Vice President. There are exemptions for the need of approval made for grades, letters of recommendation, retention-promotion-tenure documents, patentable research findings, etc. that are used regularly in the performance of faculty and staff duties. As stiff fines to the university (and individually) can be imposed for a lost/stolen device which does not have proof of encryption, it is critical that these devices not only be encrypted, but that the state of encryption is properly documented. CHPC has an example of the Device Encryption Attestation document (<https://www.chpc.utah.edu/documentation/Encryption%20Attestation.pdf>) which we use.

Regarding the evaluation of third parties given access to the data, note that at present the university does not have a Business Associate Agreement (BAA) with Google, so if you have patentable research data on Google, you are not compliant with the updated policy. Amazon Web Services are only compliant in specific cases (it is recommended you ask the Privacy Office if you have questions or concerns). The university does have BAAs with Azure, Box, and LabArchives, making them suitable for storage of restricted and sensitive data.

If you have any questions on the security policies, reach out to either the Information Security Office via ISO-GRC@utah.edu or to CHPC via helpdesk@chpc.utah.edu. Additional information on BAAs can be obtained on the university's Information Security and Privacy Office site (<https://uofuhealth.utah.edu/privacy-office/>).

Cyberteam Focus Groups

In 2017, the CHPC received a National Science Foundation grant to contribute to a regional team of “cyberinfrastructure facilitators” providing high performance computing (HPC) support throughout the Rocky Mountain region. One of the goals of this team is to improve our understanding of HPC users’ computing needs and knowledge of existing resources. To this end, CHPC will conduct online surveys and focus group discussions with several labs using HPC resources across a variety of research areas this year. We hope to discover whether or not labs have unmet HPC needs, and whether those needs can be met through new HPC services, improved documentation, or additional training. Following focus group discussions, participating labs will receive specific recommendations on how their computing needs could be met. If you would like to participate in a focus group please email Brett Milash at brett.milash@utah.edu or helpdesk@chpc.utah.edu.

Managing Python Package Installations with Anaconda

Technology · Wim Cardoen



Python's Success and Challenges

According to a 2018 study [1] by the Institute of Electrical and Electronics Engineers (IEEE), the Python language was again ranked the most popular programming language. Many factors have contributed to its popularity, including the clarity and expressiveness of the language and the speed with which Python code can be developed and deployed. The existence of a wide variety of Python packages (freely available at the Python Package Index [pypi.org]) is a major contributor as well.

The installation of some popular Python packages from the source (e.g. NumPy [numpy.org], SciPy [scipy.org], and Matplotlib [matplotlib.org], which form the building blocks of many scientific and data science packages) can be quite daunting: it requires a substantial knowledge of program compilation and linking. At the CHPC, we have been addressing this issue by actively maintaining several Linux Python stacks (versions of Python with many general-purpose packages, such as NumPy and SciPy, in addition to packages requested by our users). Over the years, we have experienced the following bottlenecks:

- *Curse of success*: Due to the increasing popularity of the Python language among our users, it has become a challenge to keep up with the growing Python package installation requests from individual users.
- *Package versioning*: The Python packaging system allows *one and only one* version of a Python package to be installed within a Python stack. This may lead to potential conflicts among Python packages built within the same stack (for example, package A requires version *x.y.z* of package W while package B explicitly depends on version *a.b.c* of package W).
- *External dependency issues*: Many scientific packages require the presence of external or non-default libraries. The installation of such packages from the source (e.g. `netCDF4`, `pygdal`, and `wrf-python`) by our users can be a very difficult and tedious process.

The user can create his or her own Python Virtual Environment (PyVenv) or modify the `PYTHONPATH` environment variable to circumvent the package versioning issue. The Anaconda suite also addresses the remaining issues.

Recommended Approach: The Anaconda Suite

Anaconda, Inc. (anaconda.com), a major player in the data science and consulting industry, provides a free Python stack (containing the most recent versions of Python 2.7 and 3 for Linux, macOS, and Windows). After download-

ing and installing the Anaconda suite in his or her home directory, the user can *autonomously* install or uninstall Python packages in his or her environment.

The Anaconda suite also addresses the aforementioned dependency problem: a wide array of Python packages (and their dependencies) are offered as pre-compiled objects by either Anaconda itself or by Anaconda users (in “channels”), making the user installation of Python packages substantially easier.

Due to its flexibility and ease of installation, we are now encouraging our users to maintain their own private Anaconda suite, which results in a significantly improved experience for the CHPC user as well as for the CHPC staff. We still have seen a few rare cases with conflicting Linux dependencies. If such cases occur, we assist our users.

Installing the Anaconda Suite

The Anaconda suite comes in 2 different flavors: a full-fledged distribution, *Anaconda*, and a light-weight distribution, *Miniconda*. The Anaconda distribution contains more than 100 pre-installed packages and its installation requires several gigabytes of disk space. The Miniconda distribution contains the bare essentials and its footprint is therefore an order of magnitude smaller.

To save space in home directories, we recommend our users install Miniconda and only the packages they strictly need. At the time of the installation, the user can also choose between either a Python 2.7 or a Python 3 version. In what follows, we will describe the installation process of a Python 3 Miniconda distribution. The installation of Python 2 or the Anaconda distribution is almost identical.

Installing Miniconda (with Python 3)

The installation of the Miniconda distribution can be done in the following way:

- Download the Miniconda installer from the Anaconda website:

```
wget https://repo.continuum.io/
miniconda/Miniconda3-latest
Linux-x86_64.sh
```

- Install the distribution within a directory of your \$HOME environment:

```
bash ./Miniconda3-latest-Linux-x86_64.sh
-b -p $HOME/software/pkg/miniconda3
```

The flag `-b` forces an unattended installation; the flag `-p` is followed by the directory where the Miniconda will be installed.

Making Your Miniconda Available

After a successful installation, the `python3` executable can be found in the `$HOME/software/pkg/miniconda3/bin` directory. We can make the `python3` command read-

ily available (without needing to use the absolute path from the command line) by either:

- prepending the directory `$HOME/software/pkg/miniconda3/bin` to the `$PATH` variable as follows:

```
# bash environment
export PATH=$HOME/software/pkg
miniconda3/bin:$PATH
```

```
# tcsh or csh environment
setenv PATH $HOME/software/pkg
miniconda3/bin:$PATH
```

- or creating an Lmod (<https://lmod.readthedocs.io/en/latest/>) module (the preferred approach):

```
mkdir -p $HOME/MyModules/miniconda3
cp /uufs/chpc.utah.edu/sys/installdir
python/modules/miniconda3/latest.lua
$HOME/MyModules/miniconda3
```

In order to load the module `latest.lua`, the directory `$HOME/MyModules` must be added to the LMOD `MODULEPATH` environment variable by using the following command:

```
module use $HOME/MyModules
```

The modification of the `MODULEPATH` environment variable can be made permanent by editing the file `$HOME/.custom.sh` (for users with the bash shell) and inserting the command `module use $HOME/MyModules` on the line following `#!/bin/bash`.

If, instead, the user has a tcsh or csh shell, the file `$HOME/.custom.csh` needs to be modified. The command `module use $HOME/MyModules` then needs to be placed on the line following the `#!/bin/tcsh` command.

Once we have set up the `miniconda3/latest.lua` file, we can load the `miniconda3` module with the command `module load miniconda3/latest`.

A primary advantage of the module approach is that it allows the user to “unload” the `miniconda3` module (if for instance, he or she wanted to use another version of Python) with `module unload miniconda3`. Maintaining multiple Python installations is much simpler.

Installing Additional Packages with conda install

The `conda` command, which comes with many flags and options, is the central administrating tool of the Miniconda distribution.

- The command `conda list` presents a list of the currently installed packages and their version number.
- The command `conda install [packagename]` will try to install the package *packagename*. As an example, the command `conda install matplotlib` will install the package *matplotlib* and all its dependencies.

- The command `conda uninstall [packagename]` will remove the package *packagename* from the current Python stack.

Installing Packages from Channels

The `conda install` command will fail if the package it tries to install cannot be found. Conda stores packages in channels (repositories), and the default channel contains a limited number of packages. If the package is not found, visit the Anaconda website and search for the package. If there is any channel that provides this package, it will be listed.

The `conda install` command allows users to specify a channel:

```
conda install -c [channel_name]
[packagename]
```

For example, `conda install -c conda-forge netcdf4` attempts to install the `netcdf4` package from the `conda-forge` channel.

The `conda-forge` repository is an important non-default channel. As the `conda install` command works on the basis of the “first-come, first-serve” principle for the default channels, the priority of the `conda-forge` channel, or any other, channel can be increased by adding it to the top of channel list:

```
conda config --addchannels conda-forge
```

Instead of prepending the channel (in other words, adding the repository at the top of the list to be searched), one can also append a channel (like `conda-forge`, for example) to the the repository list:

```
conda config --append channels conda-
forge
```

In order to avoid potential underlying library conflicts, we recommend trying to install as many packages as possible from the same channel.

Installing Python Packages using `pip install`

If the desired package cannot be located in any channel, the `pip install` command can be used as last resort. The drawback of this approach is the loss of the dependency and version management flexibility that `conda` provides. There is also a chance of the `pip install` command failing if it can't find the missing dependencies (e.g. libraries) for a package.

References

- [1] Stephen Cass. The 2018 Top Programming Languages. Python stays on top, and Assembly enters the Top Ten., July 2018. [Online; posted 31 July 2018].

Upcoming Backup Changes

Updates · Anita Orendt



Over the next few months, changes will be made to the backup offerings provided by CHPC. There are several motivations for these changes: the end of life of the backup agent currently being used, the aging tape backup hardware, and unsustainable growth in storage and backup requests.

The first change is the backup software being used. The current product, Arkeia, is no longer being developed and existing maintenance contracts will not be supported beyond the end of 2019. After evaluating several backup solutions, we chose Bareos, an open source product. At least initially, we will also have a support contract from Bareos GmbH & Co. KG.

In addition, we are moving away from backing up onto tape, as the CHPC tape infrastructure is aging. The cost of replacing the existing tape infrastructure, i.e., the tape libraries and servers, as well as expanding this infrastructure to keep up with the growing demand, would be costly. Instead, we are moving backups to the existing archive Ceph object storage systems, **pando** in the general environment and **elm** in the protected environment. As a note, if CHPC were to refresh the tape infrastructure, the cost would have to be passed onto the end user (currently the backup is subsidized with groups only paying for the tapes), and it works out to be nearly the same per TB cost as backing up to the Ceph space.

These changes will apply to all backups offered by CHPC: the backup of the group purchased home directory space in the general environment, that of the group spaces in the general environment, and that of both home directories and project spaces in the protected environment. However, CHPC will continue to manage the backups of the purchased home directory space in the general environment and the cost of the backups is already built into the price of the storage. Therefore, groups who have purchased home directory space (space beyond the default 50 GB home directories, which are *not* backed up by CHPC) will see no changes as CHPC will continue to handle the backup of these directories at the same interval. The same is true for the backup of both the home directory and project space in the protected environment.

However, for the group spaces in the general environment, these changes will impact groups who purchase new group spaces and wish to have these spaces backed up. In the existing model, groups were able to request that CHPC create quarterly archives of group spaces by purchasing the required tapes. The process of doing these quarterly archives to tape is very people- and time-intensive. Mov-

ing forward, CHPC will no longer be offering backup to tape as an option for either new general environment group spaces *or* for existing general environment group spaces that are not already being backed up to tape. The transition from tape-based to disk-based backup will be phased, with CHPC continuing to backup to tape the group spaces for which the owning group has already purchased tapes.

Finally, as the total project space in the protected environment continues to grow, we will need to explore moving to a backup model similar to that we are introducing for the general environment group space.

Please remember that you should always have an additional copy—or possibly multiple copies, on independent storage systems—of any crucial/critical data. While storage systems built with data resiliency mechanisms (such as RAID and erasure coding used on CHPC storage offerings) allow for multiple component failures, they do not offer any protection for large-scale hardware failures, software failures leading to corruption, or for accidental deletion or overwriting of data. Please take the necessary steps to protect your data to the level you deem necessary.

Group Space Backup Options

In order to assist groups with this transition, CHPC is working to provide information on a number of options that exist for backup of data. Many of these options can also be used for backing up your CHPC home directory if it is not being backed up as well as for backing up any local computer resources.

CHPC backup of group space to pando: CHPC will continue to provide the option of CHPC-managed quarterly archives of the entire contents of a group space to pando, provided that the group purchases the required storage. In order to do this, you will need up to twice as much space on pando, in order to allow for two copies, such that the previous copy can exist while the latest backup is being processed. The current price for pando storage is \$140/TB for 5 years. The pando space can be shared among all members of a group.

Owner backup to the U's Google Drive Space: There is a university agreement with Google (gcloud.utah.edu) that provides for unlimited storage. This space is a free option for user-managed backup of data and is one that a number of CHPC users already use, via the tool rclone, described in the next section. Please keep in mind that Google Drive is only suitable for public data; it is *not* suitable for sensitive or restricted data—see the article on sensitive data in this newsletter as well as the information found on the above link for additional details. One consideration is that

the Google Drive storage is owned by an individual, not by a group.

Owner backup to Box: This is a second no-cost option suitable for sensitive/restricted data. However, there is a file size limitation of 15 GB. In addition, if using rclone, the credentials expire and have to be reset periodically.

Owner backup to pando: This is a good choice if a group wishes not to use Google Drive, especially if only a subset of the data needs to be backed up or if a different backup frequency is desired. Again, this will require the purchase of the necessary amount of storage on pando.

Owner backup to other storage external to CHPC: Some groups have access to other storage resources, external to CHPC, whether at the university or at other sites. The tools that can be used are dependent on the nature of the storage.

Backup Tools

While CHPC anticipates making the Bareos backup utility available to the user, initially this will not be an option. There are, however, other tools available, some of which are mentioned on our Data Transfer Services web page (https://www.chpc.utah.edu/documentation/data_services.php#Parallel_Transfer_Tools), to handle the backup. Tools include:

- **Rclone** – this is a tool best suited for transfers to object storage systems such as pando, Google Drive, and Box
- **Fpsync** – a parallel version of rsync; suited for transfers between typical Linux “POSIX -like” file systems
- **Globus** – best suited for transfers to or from resources outside of CHPC

The link above is to the CHPC documentation, which provides a good starting point for using these tools.

RMACC Symposium

Registration for the Rocky Mountain Advanced Computing Consortium (RMACC) Symposium (<http://www.rmacc.org/HPCSymposium>),

which will be May 21–23 in Boulder, Colorado, is now open. The symposium features

a variety of speakers, discussions, and tutorial sessions. It also includes a student poster competition for graduate and undergraduate students with the winners receiving an all-expenses-paid trip to either PEARC19 in Chicago, IL or SC19 in Denver, CO.

The general registration fee is \$175; students can register for \$35 and postdocs for \$95. A “tutorials-only” (Thursday) option is available for \$110. Student scholarships to help with registration and travel fees are available.



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

Thank you for using CHPC resources!

Welcome to CHPC News!

If you would like to be added to our mailing list, please provide the following information via the contact methods described below.

Name:

Phone:

Email:

Department
or Affiliation:

Address:
(campus
or U.S. mail)

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 acknowledgments 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 in one of the following ways:

Electronic responses

By email: helpdesk@chpc.utah.edu

By fax: (801) 585-5366

Paper responses

By U.S. mail: 155 South 1452 East, Rm 405
Salt Lake City, UT 84112-0190

By campus mail: INSCC 405