SFU CO-OPERATIVE EDUCATION

SFU Science & Environment Co-op

Employer - Project Release and Evaluation

ph. 778-782-4716 fax. 778-782-3031

Please read and evaluate the work report/project below, then return the form and report/project to the student for submission to SFU Co-op. Your feedback is a key part of the student's learning. A co-op coordinator will also review the project and discuss your feedback with the student.

Student Name _____ Major _____Student #_____

Work Term **Fall** **Spring** **Summer** 201__

Company/Organization: _____ Department: _____

Supervisor's Name: _____ Title/Position: _____

Supervisor's Phone:_____ E-mail: _____

Title of Report/Project: _____

Type of Report/Project: _____

Evaluation: (*We would appreciate your evaluation if possible, especially for reports/projects of the following types: technical report, literature review, oral presentation or scientific poster.*)

	excellent	**good**	**satisfactory**	**unsatisfactory**
Quality of Content				
Authority & accuracy				
Analytical content				
Thoroughness				
Quality of Presentation				
Literary Quality				
Grammar & spelling				
Clarity				
Style				
Organization/structure				
Level of learning demonstrated				
Overall Grade				

Comments: *(We welcome your comments on the student's report/project, even if an evaluation was not completed above.)*

_____

_____

_____

_____

Supervisor's Consent:

I have reviewed the work report/project, and the report/project is (please check ONE box only):

❒ *Approved for use as reference material for other co-op students. (This assists students researching possible future co-op roles in your organization.*)

❒ *Confidential and may be reviewed only by Co-op staff (e.g., report contains proprietary information that should not be released).*

Supervisor's Signature_____ Date _____

The information on this form is collected under the authority of the University Act (R.S.B.C. 1996, c.468, s.27(4)(a). It is related directly to and required by the SFU Science & Environment Co-op Program to inform the student of the program regulations and to formally contract the student to abide by such regulations. The Form will be filed in the student's personal file. If you have any questions about the collection and use of this information, please contact the Program Manager, Science & Environment Co-op Program 778-782-4123.

SFU CO-OPERATIVE EDUCATION

CO-OP WORK REPORT

SFU Science & Environment Co-op
8888 University Drive, Room P9447
Burnaby, BC V5A 1S6
Tel: 778-782-4716 • Fax: 778-782-3031
www.sfu.ca/coop

Organization:	
Supervisor's Name:	
Student's Job Title:	

Report Title:

Student's Name:	
Degree (BSc, BA, MSc, etc.):	
Major:	
Work Term # (1, 2, 3, etc.):	
Semester (e.g., Fall 2016):	

Type of Report:

Technical	Lit. Review	Interview	Presentation
Poster	Reflective	Article/Blog Post	Other

Table of contents

Summary

During my time at BCCRC IT I supported researchers by installing software and providing assistance in user account management. I took the opportunity not only to understand their business needs from the IT perspective, but also from their own perspective by involving myself in research experiences within the company. I provide a technical report that includes good IT practices around user account management, data management and network security; an example of a software installation guide from the command line; results from the research experience I involved myself in and further how doing research at BCCRC allowed me to comprehend researchers' needs.

Section 1 IT Support

Section 1.1 Good practices for user account management

Employees within BCCRC require the creation, extension, deletion and modification of their accounts. Every user is assigned a unique account within BCCRC. This account allows them to log in on any computer within the company's network and services such as email, booking rooms, requesting software, etc. Every three months, user's password expires and a password change must take place. IT is responsible for creating, modifying, extending and deleting any given user's account.

Below are key aspects of what good practices for user account management encompasses.

1.1.1 Account Extension and modification

It is important to ensure we receive a request from the user's manager. No one other than the manager can request for a user's account to be created, extended, deleted or modified in any way. This protects the company from just anyone changing permissions. Users have access to their group's files only. If access to files outside their scope is needed a request with a valid reason must be send to IT prior to approving and modifying.

1.1.2 Password reset

As mentioned previously, passwords must change every three months. The purpose of this is to minimize the potential for outsiders to hack into user's account and obtain confidential data, patient's information, research results, along with other sensitive information. In the event that a user forgets their password, we must receive a request from the user's manager to reset their password.

1.1.3 Local Administrator Policy

As a general rule of thumb, users are not granted administrator rights as this compromises all computers connected to the network. In the event that a user requires administrator rights for their work, not only must IT receive a request from their manager, but also they must prove this is to satisfy a business need.

Section 1.2 Workshops and online trainings [1]

The following subsections display some of the things I learned by taking workshops and online trainings within BCCRC. Whereas the main target of the workshops were researchers, I distilled valuable information for CRC IT.

1.2.1 Best Practices for Research Data Management: Using Dataverse for Research Data

In this workshop I learned about the importance of keeping good data management practices from two perspectives: the researcher perspective and the IT security perspective. I will focus on the IT security perspective in this report.

This perspective focuses mostly on Data Storage and Security. Good practices include that each user ensures to have at least three copies of their files and data, two of which should be stored on two different media, and the third one to be kept offsite. From the IT perspective, it is of interest to ensure the following aspects are covered:

Network security

It is important to know who has access to the network and also to ensure the network is

protected against potential threats.

Physical security

Knowing who has access to the computers, files and how is data being transported.

Computer security

Ensuring anti-virus software is installed and up to date in all network computers, as well as ensuring that computers are protected against power surges. Use of password and data encryption in all computers and to provide means for researchers to access their data in private and secure settings are crucial steps.

Backing up data

Back ups are performed on a regular basis, depending on the needs of the research group and the means available and can be done on a daily, weekly, biweekly, monthly, every six months or on a yearly basis.

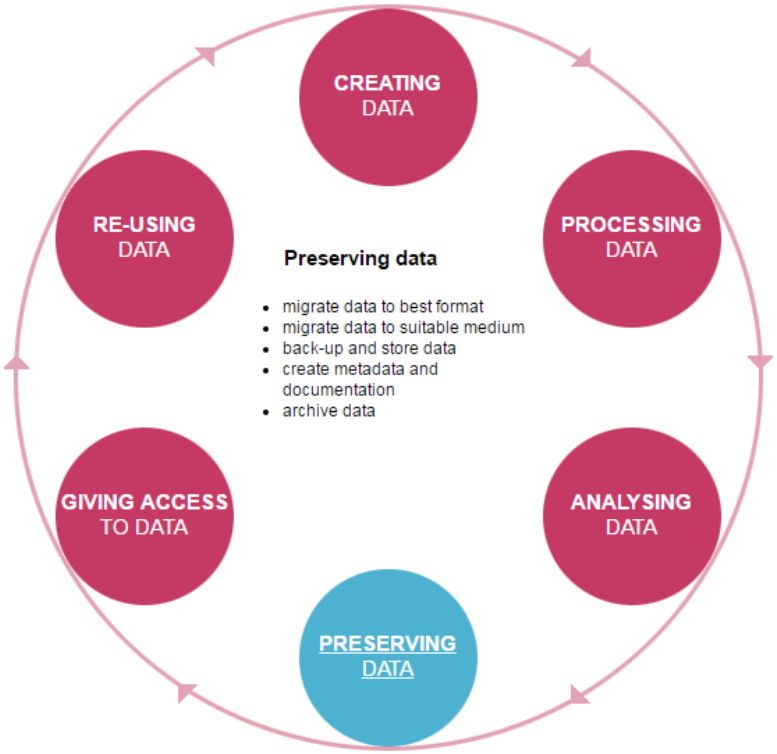

Figure 4. Data management diagram. In this stage the focus is good practices for data preservation [2]

1.2.2 Introduction to the Genome Sciences Centre

What is the Genome Sciences Centre?

" Today the Genome Sciences Centre specializes in large-scale, ultra high-throughput sequencing applications. With more than 16 sequencing instruments and a state of the art data and computation centre, the GSC decodes cancer genomes, epigenomes & transcriptomes, and is poised to apply this knowledge for the direct benefit of cancer patients.

Recognizing the diverse applications of genomics approaches, the Genome Sciences Centre also provides technology platform access to Genome Canada and Genome BC projects working in the areas of human health, the environment, forestry, agriculture, and aquaculture.

The Genome Sciences Centre has played a leading role in many large scale genomics initiatives."[3]

I talked to Lance Bailey, Systems Coordinator at BCGSC, about good practices to keep the network safe and this is the advice he gave me:

> "1) Do not allow local administrator rights to researchers, make it a condition of connecting to the network.
> 2) Have sound reasons why local administrator rights are a bad thing with [anonymous] examples.
> 3) Be responsive to the software installation requests. It is often the waiting for software that people hate not the inability to do the install themselves.
> 4) Clear expectations with the researchers – they can't install their own software but they also cannot expect to nickel and dime the installer with a thousand tweaks of the install over the course of weeks. "

I have learned throughout my stay at CRC IT how important it is to be responsive to software requests. Despite Lance's advice, there are a number of workers at BCCRC who

requiret administrator rights. We grant these rights when there is a business need. Whereas this allows BCCRC employees to do their work as desired, it also compromises every computer on the network. It is thus important to be careful around who and why we give these rights.

1.2.3 A Hands-on Introduction to ORCA

One solution GSC IT implemented to the problem of giving researchers administrator rights was creating ORCA (genOmics Research Container Architecture). ORCA is a "platform for bioinformatics analysis. It is suited for those wishing to conduct self-serve analysis using their own existing data".[4]

It is a private and secure environment, and it allows GSC IT to never grant administrator rights to researchers while also satisfying their software needs.

The way to access ORCA is by secure shelling using a standard SSH client with corresponding credentials. It is command line-based and includes libraries and applications from Homebrew-Science. I had an opportunity to use it during Hackseq to download data and software.

Advantages of using ORCA as a researcher are: I do not need to install software on my computer and as such I save setup time. All software I needed for Bioinformatics analysis is found there. Some disadvantages can present when there is no familiarity with command line use.

Advantages of ORCA from the IT perspective: it provides full control of the applications installed on every business computer, it is private and secure. Disadvantages of ORCA include its physical limitations, as well as its maintenance costs in time and money.

Section 1.3 Software maintenance

Upon receiving an IT form (see Appendix B for a sample) and scheduling an appointment with the user, we can then proceed to install requested software. The procedure varies depending on the operating system and provided it is standard installation, this process is quite straightforward.

In the event that we are asked to install customized software, we describe in the notes section of the OTRS the steps we took so that future IT staff can revise it and use it as a guide.

1.3.1 Licences and Policies

It is important that CRC IT ensures all policies are in place and no licence agreement is being violated. This requires taking special care of requesting purchase receipts and invoices as well as verifying that those users who request software as UBC staff or for UBC assets are eligible for software as per UBC Licence Agreement. Examples of this include Matlab and Solidworks.

1.3.2 Installing on Mac, Windows, vs Ubuntu

Most installations are performed remotely.
From Windows we access all CRC computers via Remote Desktop Connection. Nearly all installations for Windows are straightforward and require following a wizard.

From a Mac, standard installations are done remotely via Apple Remote Desktop. Installations can be done either via a dmg file or via terminal.

From a Linux it is done via secure shelling. All installations are done from the terminal.

There are a number of research groups at BCCRC that require customized software to perform data analysis or make heavy use of Bioinformatics tools. Most of these tools are open source, and as such provide a higher level of difficulty to build and/or debug.

1.3.3 An example of open source software installation done on a MacOS X

Here is an example of a customized installation for a BCCRC computer.

Pre-requisites:

Ensure a working Xcode version that is compatible with Mac OS. I installed on Mac OSX El Capitan, and the Xcode version this computer has is 7.3.1. Ensure you have the Command Line Tools installed. To install, open a terminal and type

```
$ xcode-select --install
```

Once the installation has been completed, type

```
$ gcc --version
$ whereis gcc
```

You should find it in /usr/bin. We will need gcc 4.7.0 or later for STAR.

Install Homebrew.[5]

Open a terminal, copy and paste

```
$ /usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

To check the installation was done successfully, type on the terminal

```
$ brew update

$ brew doctor
```

Once this has terminated, type

```
$ brew install wget
```

```
$ brew install git
```

we will need these later for STAR.

Install bedtools

```
$ brew install homebrew/science/bedtools
```

Install SAMtools

```
$ brew install homebrew/science/samtools
```

Install Cufflinks

```
$ brew install homebrew/science/cufflinks
```

Install Filezilla

```
$ brew tap caskroom/cask

$ brew cask install filezilla
```

Install TextWrangler

```
$ brew cask install textwrangler
```

Install STAR (a.k.a RNA-seq aligner)

At the time of installation, there was no homebrew working version of STAR for Mac OS X El Capitan, otherwise you could simply do

```
$ brew install homebrew/science/rna-star
```

Instead

```
$ git clone https://github.com/alexdobin/STAR.git
```

A download file with the name STAR will appear. Copy that and paste it in the /opt directory. Next

```
$ cd /opt/STAR/bin/MacOSX_x86_64

$ chmod a + x ./STAR
```

Install bedGraphToBigWig

```
$ git clone git://github.com/timpalpant/Ruby-Genomics.git
```

Once downloaded, copy the file in the /opt directory

```
$ cd /opt/Ruby-Genomics/ext/ucsc-binaries

$ chmod a + x ./bedGraphToBigWig
```

If interested in installing the whole Ruby-Genomics package, once you copied in /opt

```
$ sudo gem install bundler

$ password

$ cd Ruby-Genomics

$ bundle install
```

Test it

```
$ rake
```

Section 2. Research Experiences at BCCRC

Section 2.1 Hackseq 2016 Work Report

About Hackseq

" Hackseq is a Vancouver-based Hackathon (software development 'hacking' marathon) focused on open genomics. We want to bring individuals with diverse backgrounds together to collaborate on scientific questions and problems.

Our philosophy is open-source, open-notebook, open science."[6]

Project

Given a command line tool with a number of parameters and a target metric to be optimized, implement a tool that finds the values for those parameters and also maximizes a given target metric.
In particular, consider genome sequence assembly, which often has a variety of parameters related to expected coverage of the reads and heuristics to remove read errors and collapse heterozygous variation.
Currently, the optimization process is done manually resulting in tedious and unnecessary long running time.
The purpose of this project is to design and implement a tool to automate this process and generate a report of the result.

Project Lead

Shaun Jackman, Graduate Student, BC Cancer Agency Genome Sciences Centre

Participants

Craig Glastonbury, Daisi Huang, Hamid Younesy, Jasleen Grewal, Lisa Bang, Veera Manikandan Rajagopal, Y. Brian Lee, Laura Gutierrez Funderburk.

Project Outline

1. Identify functions and data sets to optimize
    1. A 200kpb bacterial artificial chromosome (BAC) that can be opti- mized in minutes for development using ABySS
    2. A real genome assembly problem that can be optimized in a few hours using ABySS
2. Evaluate optimizers for usability and speed
    1. OPAL by Dominique Orban – Optimization of algorithms with OPAL
    2. Spearmint by Michael Gelbart – Predictive Entropy
    3. Search for Multi- objective Bayesian Optimization
    4. ParOpt by Stefan Seemayer which uses scipy.optimize and Nelder- Mead optimization
    5. Python packages such as scikit-optimize
    R packages, mco and optimix
3. Generate a report of the results of the optimization
    1. Generate plots of target metric vs parameters
    2. Draw the Pareto frontier of the target metric and a second metric of interest (contiguity and correctness) likely in R using ggplot
4. Write a short report of our experience
    1. Post on GitHub pages
    2. Possibly submit to a preprint server (bioRxiv, PeerJ, Figshare)
    3. Possibly submit for peer review, such as F1000Research Hackathon

Motivation

Black-box optimization lies at the intersection of diverse fields, including statistics (experimental design), optimization and machine learning, just to name a few.

My involvement in this project

See Appendix A to see my Github page.

We were initially interested in optimizing $N50$ given a parameter $16 \leq k \leq 50$.
I learned how to perform basic assembly steps using ABySS, how to use BWA and BWA-SW to align reads and contigs to a reference genome, and used IGV to visualize these alignments.

Indexed the reference file

```
$ bwa index chr3.fa
```

Aligned the reads to the reference using BWA

```
$ bwa mem -t2 chr3.fa 200k.fq > 200k.sam
```

Inspected the reads

```
$ gunzip -c 200k.fq.gz | head
```

Assembled the reads into contigs using ABySS. I fixed a parameter $k = 42$ during this assembly

```
$ gunzip -k 200k.fq.gz
$ mkdir k42/
$ ln -s ../200k.fq  k42/
$ abyss-pe -C k42 name=HS0674 k=42 v=-v in="200k.fq" contigs 2>&1 | tee abyss.log
```

The assembly runs in three stages: assemble contigs without paired-end information, align the paired-end reads to the initial assembly, and merge contigs joined by paired-end information

```
$ abyss-pe name=HS0674 k=42 in="200k.fq.gz" unitigs -n
$ abyss-pe name=HS0674 k=42 in="200k.fq.gz" pe-sam -n
$ abyss-pe name=HS0674 k=42 in="200k.fq.gz" contigs -n
```

Once the assembly has completed, the *N*50 of the assembly for *k*=42 is

n	N:500	L50	min	N80	N50	N20	E-size	Max	Sum	Name
488	130	31	502	653	1436	2513	1685	5160	14096	k42/HS0674-contigs.fa

After iterating this process for different values of k= 38,48, 20 I obtained

n	N:500	L50	min	N80	N50	N20	E-size	Max	Sum	Name
126	58	14	554	2259	4335	8744	5456	13375	197029	K38/HS0674-contigs.fa
174	0	0	0	0	0	0	0	0	0	k48/HS0674-contigs.fa

931	75	15	503	1619	3438	7316	4308	9007	180548	K20/HS0674-contigs.fa

Here is the $N50$ of the assembly

k	$N50$
20	3438
38	3438
42	1436
48	0

Manually attempting different values of k and without an adequate searching algorithm makes this search time consuming and tedious. Shaun had manually attempted linear and grid search, and was interested in exploring Bayesian and amoeba methods of optimization.

Shaun distributed the different packages and open-source tools to be tested as per the project outline. He and I worked on Spearmint, an open source tool that performs Bayesian Optimization, along with MongoDB.

Shaun Jackman implemented code for a single parameter k that optimized $N50$. I assisted by debugging and later extended this code for two parameters k, s. After 20 iterations this is what I obtained through Spearmint:

Iteration	k	s	$N50$
1	16	200	3269
2	16	2000	1263
3	17	2000	1640
4	22	200	7794
5	27	200	19308
6	28	200	19356
7	29	290	17858
8	30	1100	9467
9	31	2000	12520
10	32	200	14700

11	37	200	16442
12	37	2000	16442
13	41	200	16451
14	41	1982	16451
15	41	2000	16451
16	43	1170	16449
17	43	1997	16449
18	45	200	8594
19	45	245	8594
20	45	2000	6308

Thus, there is a maximum $N50 = 19356$ for $k = 28, s = 200$

Conclusions

Spearmint works best with continuous functions, and the data sets and functions we attempted are discrete. Unfortunately there are no open-source tools whose purpose is to optimize discrete functions. We were able to make Spearmint work with not too many hassles by restricting the maximum number of parameters to optimize to 16 (max Spearmint can take is 99), and providing discrete functions and/or step functions while being cautious around the values that these parameters took on.

Open questions

Optimizing $N50$ with respect of k, s.

Plotting results.

Finding the $N50$ of real human genome assembly

Section 4. Conclusions and remarks

Making use of the open source software that the Bioinformatics department at BCCRC uses during my research experience and having a hands-on experience doing research within the company not only motivated me to learn more from my surroundings, but also gave me extra tools to understand the importance of fostering good practices within IT. Such practices include and are not limited to: responsiveness, excellent interpersonal and communication skills, implementation of network security practices, just to name a few. In particular, it strengthened my understanding of what it means to ensure the network is protected and allowed me to see the consequences of not taking proper care of data and user account management.

Section 5. Acknowledgements

Thanks to Jana Makar, Roland Santos, Scott Baker, Lance Bailey and Eugene Barsky for providing online training and answering my questions on best IT practices for user account and data management.

Thanks to Shaun Jackman for believing in me. Having a sense that my research project leader believed in my capacity to discover made a huge difference in my learning experience.

Special thanks to my employers and supervisor: Dubravko Pajalic, Ken Ho and Alexander Mogutnov for being supportive of me during the hardest times. Thanks to my co op coworkers Samin Semsarilar and Benjamin Hoffman for fostering a fun learning environment.

References

[1]. All workshops were taken through Westgrid:

* Best Practices for Research Data Management - Using Dataverse for Research Data:

https://www.westgrid.ca/events/best_practices_research_data_management_using_dataverse_research_data

* Introduction to the BC Cancer Agency Canada's Michael Smith Genome Sciences Centre:

https://www.westgrid.ca/events/introduction_genome_sciences_center

* A Hands-On Introduction to ORCA (the genOmics Research Container Architecture):

https://www.westgrid.ca/events/handson_introduction_orca

[2]. Data Lifecycle http://www.data-archive.ac.uk/create-manage/life-cycle

[3]. Genome Sciences Centre website http://www.bccrc.ca/dept/cmsgsc

[4]. ORCA website http://www.bcgsc.ca/services/orca

[5]. Homebrew official website http://brew.sh/

[6]. Hackseq 2016 website http://www.hackseq.com/

Appendix A. Hackseq results

See https://github.com/lfunderburk/hackseq2016_Spearmint for more details on the results

Appendix B. IT forms

BC Cancer Agency Research IT Support Request Form

One request/computer per form

Please use "BC Cancer Agency Research IT Equipments Move Request Form" for moving request.

Part I – General Information

Ticket #			
Department		Room #:	
Phone		Requested by:	

Description of the work (With computer name if you have it):

Purpose of the request:

Part II - Software Installation

Software Package	**Cost[1]**	**Installation required**
Microsoft Office	$380	
Adobe Acrobat Professional	$100	
Sophos Antivirus	Covered by CRC license	
Endnote	Covered by CRC license	
Additional software installation Please specify and provide name of the license holder, serial number and other information needed for installation		

Part III - Hardware Installation

Item	**Cost**	**Item required**
Network patch cable up to 15'	$20	
New Network Drop	$250	
Existing Network Drop	*Please specify drop number and location (e.g. 13-110-05)*	

Part IV – Cost Centre (Required)

Cost Centre:

Part V – Desired Completion Date

Desired Completion Date:

_____ _____

Lab Manager/Supervisor Signature **Print Name of Lab Manager/Supervisor**

Date:

[1]Price may slightly vary +/- due to US dollar exchange rate