

# Genome Variant Analysis 2020

General Introduction to the Zoom edition

# Background

- Doctorate from The Ohio State University
  - Epigenetic modifications and cell signaling in ovarian cancer
- Postdoctoral work with Jeffrey Barrick at UT Austin
  - Microbial evolution, and synthetic biology
- Self taught computational biologist
  - 8<sup>th</sup> year teaching this course
- Commonality:
  - Next-generation Sequencing

# Disclaimers up front

- Spelling – Has never been a skill I possess, I'm sure there are errors left in the tutorials. Let me know if any interfere with learning.
- Typos – Will likely be your biggest problem in using the commands provided, and the biggest problem in your own work. Try to type the commands out to get practice, but remember you can copy paste.
- Names – Normally I apologize up front for not being able to remember people's names. This year I apologize if I mispronounce your name, please do correct me if it bothers you.
- Royal "we" – Several of these tutorials were originally produced by former instructors or TA's and reference "we" or "your instructor(s)". I thank them all for their help, and take responsibility for anything that goes wrong.

# Zoom Disclaimers

- Please change your zoom name to start with a M or P depending on if you are using a Mac or PC.
- Feedback will be more important than ever, email or before/after class.
- Don't be shy asking questions or wanting me to look at your screen.
- Headphones – I'll have them on the whole time even if not sitting directly in front of screen.
- Dogs – I have dogs they are awesome and usually lazy during the day. Very possible at least one of them will decide to bark while I am talking. He's very sorry he just can't help it.
- Birds – I do NOT have a pet bird. I DO have a husky who does a pretty good bird call and imitation of a screaming hyena. He's also sorry.

# Class Expectations

- Participant goals:
  1. Learn how to analyze my data, and have it fully analyzed by the end of the class.
  2. Maybe learn how to analyze NGS data in general.
- Teaching goals:
  1. Teach the fundamentals of NGS variant analysis.
  2. Provide context and exposure multiple types of data.
  3. Use example commands to familiarize you with variety of programs.
  4. Provide resources to enable you to do analysis you haven't thought of yet.

# Where to start?

- Many say “don’t know where to start” their data analysis once they have their data files.
- Ideally should have “started” weeks-months ago in planning experiments.
  - If you fall in this category don’t worry you are in a common situation.
  - Later presentations will explain differences in sequencing libraries and how to make the best choices going forward.

# General notes about analysis

- There is no one “perfect” way to analyze next-gen data.
  - Results in many, nearly equivalent or redundant analysis types and tools.
  - Not all types of analysis are necessary, or even appropriate.
- Wet lab methods used to generate data may limit available/appropriate analysis methods.
  - Not-surprisingly, planning experiments from the beginning based on what answers/analysis you want can save time and money.

# Introductions:

- Who are you?
- What lab are you in?
- What organism do you work with (can be as broad as virus, bacteria, plant, animal, human, etc)?
- 1-2 sentences on what type of variant analysis you are interested in
  - Type of reference genome
  - Mixed populations vs clonal isolates
  - Association vs causation studies



# Set up recommendations

The image shows a web browser window displaying a Wikis page from The University of Texas at Austin. The page title is "Linux and Lonestar 5 Setup -- GVA2020" and it is part of the "Genome Variant Analysis Course 2020". The page content includes a table of contents with sections like Overview, Objectives, and Tutorial. The Tutorial section lists steps such as logging into Lonestar5, setting up profiles, and understanding TACC. Below the browser window, a terminal window shows the installation of cutadapt. The terminal output includes the command `cutadapt --help` and its output, which describes the tool's purpose and usage. The terminal also shows a command `cutadapt -a GATCGGAAGAGCGGTTCGTGAGGAGACAGGCAGAAATGAAGAAGAGCTCA -o output.fastq input.fastq` and an error message: `cutadapt: error: no such option: -v`.

Linux and Lonestar 5 Setup -- GVA2020

Created by Daniel Edward Deatherage, last modified about 4 hours ago

- Overview:
- Objectives:
- Example things you will encounter in the course:
  - Code blocks
  - Warnings
  - Info boxes
  - Tip boxes
- Tutorial:
  - Logging into lonestar5
  - Setting up your lonestar profile
  - Setting up other shortcuts:
  - Understanding what your .bashrc file actually does.
  - Editing files
  - How should we name files and folders?
  - Stringing commands together and controlling their output
  - Understanding TACC
    - Diagram of Lonestar5 directories: What connects to what, how fast, and for how long.
    - Understanding "jobs" and compute nodes.
      - Further sbatch reading
      - Using launcher\_creator.py
  - Running a job
  - Interrogating the launcher queue

```
ded@ded:~/GVA_fastqc_tutorial$ cutadapt --help
the filtering criterion in order for it to be
filtered. Default: any
--interleaved      Read and write interleaved paired-end reads.
--untrimmed-paired-output=FILE
                   Write second read in a pair to this FILE when no
                   adapter was found in the first read. Use this option
                   together with --untrimmed-output when trimming paired-
                   end reads. Default: output to same file as trimmed
                   reads
--too-short-paired-output=FILE
                   Write second read in a pair to this file if pair is
                   too short. Use together with --too-short-output.
--too-long-paired-output=FILE
                   Write second read in a pair to this file if pair is
                   too long. Use together with --too-long-output.
tacc:/scratch/01821/ded/GVA_fastqc_tutorial$ cut
cut
tacc:/scratch/01821/ded/GVA_fastqc_tutorial$ cutadapt -v
cutadapt version 1.14
Copyright (C) 2010-2017 Marcel Martin <marcel.martin@scilifelab.se>

cutadapt removes adapter sequences from high-throughput sequencing reads.

Usage:
  cutadapt -a ADAPTER [options] [-o output.fastq] input.fastq

For paired-end reads:
  cutadapt -a ADAPT1 -A ADAPT2 [options] -o out1.fastq -p out2.fastq in1.fastq in2.fastq

Replace "ADAPTER" with the actual sequence of your 3' adapter. IUPAC wildcard
characters are supported. The reverse complement is *not* automatically
searched. All reads from input.fastq will be written to output.fastq with the
adapter sequence removed. Adapter matching is error-tolerant. Multiple adapter
sequences can be given (use further -a options), but only the best-matching
adapter will be removed.

Input may also be in FASTA format. Compressed input and output is supported and
auto-detected from the file name (.gz, .xz, .bz2). Use the file name '-' for
standard input/output. Without the -o option, output is sent to standard output.

Citation:
Marcel Martin. Cutadapt removes adapter sequences from high-throughput
sequencing reads. EMBnet.Journal, 17(1):10-12, May 2011.
http://dx.doi.org/10.14806/ej.17.1.200

Use "cutadapt --help" to see all command-line options.
See http://cutadapt.readthedocs.io/ for full documentation.

cutadapt: error: no such option: -v
tacc:/scratch/01821/ded/GVA_fastqc_tutorial$ packet_write_wait: Connection to 129.1
14.62.196 port 22: Broken pipe
```

To: Everyone  
Type message here...

# Computers computers computers

- Millions of reads, 100s of bp long, mapping to millions-billions of base long references.
- Mac/linux computers are your friend.
- Windows is getting better with powershell, but many analysis programs are only available for linux/mac.
- TACC is a time machine that lets you get stuff done much faster
  - This is where we will start the class.