

```
*****
*   Description of Folder Contents   *
*****
```

```
+-----+
| Python |
+-----+
```

-- Particle Simulator --

Project Description:

This code is from a course in scientific computing with Python. It simulates the velocity and energy of gas particles in a closed system. Further details about the project are provided in ./Python/Project Description.pdf.

This information can also be obtained at the original course website:

http://secant.cs.purdue.edu/cs190c:project3_09

A brief analysis of the simulation results is provided in ./Python/Brian Denton - CS190C Project 3.pdf.

The folder ./Python/Provided Code/ contains the code provided by the instructor to begin the project. I authored all other code.

A Windows batch file and Linux bash script are provided in the ./Python folder to run this simulation. An installation of the matplotlib module is required for plotting figures.

-- NGS_CCDF --

Project Description:

The NGS_CCDF project does the computations to generate a complementary CDF (CDF) of base call qualities and depth-of-coverage of reads from next generation sequencing (NGS) data. This project is primarily written in C++ so it is in the Cpp folder. But there is also a Python script in that folder that provides some validation of the results from the C++ code.

```
+-----+
| Cpp    |
+-----+
```

-- FedEx --

Project Description:

This code is from a course in data structures and algorithms. It simulates the arrival and departure of planes and packages from a FedEx terminal. Further details of the project are provided in ./Cpp/FedEx/FedEx.pdf.

The operating characteristics of the simulation are determined by the frequency of package and plane arrival, how long it takes to load packages onto a plane, and the time required for equipment maintenance. Parameters for these variables can be set in

./Cpp/FedEx/GlobalConstants.h.

After the simulation completes summary statistics are provided describing the number of packages processed during the simulation, the minimum, maximum, and average time a package spends in the system before it is loaded onto a plane.

The folder ./Cpp/FedEx/Provided Code/ contains the code provided by the instructor to begin the project. I authored all other code.

A Windows batch file and Linux bash script are provided in the ./Cpp/FedEx folder to compile and run this simulation. These files assume an installation of the g++ compiler.

-- NGS_CCDF --

Project Description:

This program provides the computations to generate a complementary CDF (CCDF) of base call qualities and depth-of-coverage of reads from next generation sequencing (NGS) data. The program takes an NGS pileup and exome data as inputs and generates frequency counts for the number of reads at each genomic location that surpass specified base call quality thresholds. These frequency counts are then plotted using R.

Usage: NGS_CCDF example.pileup example.exome.bed example.csv

example.pileup

This is a text pileup file (e.g. output from the Broad Institute's Genome Analysis Toolkit). Each line in the pileup file represents a particular genomic position. The pileup contains five columns providing the chromosome number, position number, the reference base, the reads covering that position, and the phred base quality score for each of the reads.

An annotated example follows:

CHROM	POS	REF	READS	QUAL
22	16076052	C	CCCCCCCCC	;BD%HH>IF

example.exome.bed

This is a text file that has three columns listing the chromosome, start position, and stop position of every exon.

An annotated example follows:

CHROM	START	STOP
1	123231	123252

example.csv

This is the output text file from the NGS_CCDF application. Each row corresponds to a row in the pileup. This file contains seven columns corresponding to the chromosome, genomic position, how many reads at that position had a phred quality score Q>10, how many reads had Q>20, how many reads had Q>30, how many reads had Q>40, and a boolean indicating whether or not that position is in an exon.

An annotated example follows:

CHROM	POS	Q>10	Q>20	Q>30	Q>40	InExon
1	123231	24	24	24	12	TRUE

A Python script is also provided to do an independent validation of the NGS_CCDF results. The Python script implements a less sophisticated, but also less efficient version of the NGS_CCDF algorithm. Because it takes a more direct approach I use this as the gold standard for checking small subsets of my results to confirm my program works. Spoiler alert: It works.

A Windows batch file and Linux bash script are provided in the ./Cpp/NGS_CCDF folder to compile and run this application and to call R to generate the plot. These files assume an installation of the g++ compiler. It is also assumed R is installed on your system and can be called from the command line.

```
+-----+
| Java                                     |
+-----+
```

Project Description:

This code is from an introductory Java programming course. It implements a version of the Minesweeper game.

Further details of the program are available in ./Java/Program Description.pdf. This information is also available at <http://web.ics.purdue.edu/~cs180/Spring2009Web/>

A Windows batch file and Linux bash script are provided in the ./Java folder to compile and run this simulation. These files assume an installation of the javac compiler.