Name: \_\_\_\_\_\_\_\_\_\_\_Solutions\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BIOINF 525 Module 3

Lab #2

3/30/2017

Please complete the exercises below. Throughout the lab sessions for this module, we will use the following notation:

Plain text indicates actions that should be taken

*Italicized text indicates explanatory material*

**Bold text indicates a point where a written response is required**

Exercise 1 – Analysis of GEO data sets

*The Warburg effect is an often-studied (and often-hyped) observation that cancer cells frequently obtain more energy from glycolysis followed by fermentation to lactic acid, and less from oxidative phosphorylation, than normal tissues. Here we will examine whether Warburg-like effects can be detected in published gene expression data sets. For these purposes, we will focus on expression of three genes: PGK1 (phosphoglycerate kinase 1, a glycolytic enzyme), CS (citrate synthase, an enzyme involved in the citric acid cycle), and LDHB (lactate dehydrogenase B, which is involved in fermentation during anaerobic metabolism.*

Go to the GEO Dataset Browser at <https://www.ncbi.nlm.nih.gov/sites/GDSbrowser> and search for data set GDS5666 – this contains gene expression profiling of a mouse tumor cell line along with corresponding metastatic isolates derived from that parental population.

Click on GDS5666, and then the “Expression Profiles” button further down on the page.

In the resulting search box, type PGK1 next to the “GDS5666[ACCN]” string and search. Click on the graph thumbnail image next to the first result.

*The graph that arises includes both a raw score – “count” – and the percentile rank of transcript level for that gene in each sample.*

**Record the average percentile rank for PGK1 in each of the five subpopulations considered in this study:**

Parental: 36.67 primary explant: 80.67 lung explant: 32.33

Bone explant: 26 Liver explant: 78

Now, repeat the process for citrate synthase (CS) and lactate dehydrogenase B (LDHB). **Record the average percentile ranks for CS and LDHB in each of the samples.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Parental** | **Primary** | **Lung** | **Bone** | **Liver** |
| **CS** | 76 | 60.67 | 56.33 | 38 | 35.33 |
| **LDHB** | 2.67 | 38 | 91.33 | 45 | 58 |

**Do you see an exaggeration of a Warburg-like effect in any of the samples? If so, which one(s)?**

Yes; all of the explants seem to show an enhanced Warburg effect: They appear to have reduced TCA cycle activity (using CS as a proxy) and increased fermentation (using LDHB as a proxy). Note that since glycolysis is upstream of both pathways, PGK1 expression is not diagnostic here.

Working together with your neighbors, identify one or more GEO Data sets that would let you determine whether the fermentation/oxidative phosphorylation balance is altered either in cancer cells vs. noncancerous cells, in cells treated with a chemotherapeutic agent vs. those that are not, or in drug resistant vs. nonresistant cancer cells.

**Describe the data sets that you compare, including GDS numbers, and describe what difference you are looking for**

Here we will use data set GDS3966, which compares primary with metastatic melanomal samples. To follow the wording of the question, we hypothesize that the metastatic cells will show an increase in LDHB expression and a decrease in CS expression, reflecting a shift toward enhanced fermentation.

**Record the relevant summary statistics for comparing the populations of interest, using both the value and percentile ranking scores from the GEO data set. Do you see any changes in regulation of fermentation between the populations that you are considering? Do you come to a different conclusion depending on which statistic you use? Which one seems more reliable?**

CS:

Value Primary 605.9 +/- 149.1; metastasis 608.7 +/- 209.7

Percentile Primary 95.4 +/- 1.18, metastasis 94.7 +/- 1.6

LDHB:

Value Primary 1503.7 +/- 758.2; metastasis 2085.7 +/- 765.8

Percentile Primary 97.7 +/- 1.04; metatstasis 98.4 +/- 0.91

Thus, the Value statistic suggests an increase in fermentation in the metastatic cells (although no loss of CS expression), whereas the percentile statistic is less clear. However, the fact that LDHB and CS are both at the very high end of the expression spectrum means that any changes observed using that statistic will likely be very small. Since the samples were reliably normalized and belong to the same data set, the Value statistic is likely preferable.

**Describe a statistical test that would be appropriate for testing whether a significant difference exists between the levels of expression of the three genes that we are considering here between the two samples.**

In general, either a t-test family comparison (if a normality assumption is justified by the number and distribution of the data) or a Mann-Whitney U test (given more than ~5 samples per group) would be appropriate here.

To simplify our analysis of the balance of fermentation to TCA cycle, we act on the log2 ratios of CS/LDHB levels in each sample. Observing that the log ratios are not normally distributed, we use the Mann Whitney U test, obtaining p=0.0007907 and a confidence interval of 0.29 – 0.97, thus suggesting a roughly 50% increase in fermentation in the metastatic cells.

Exercise 2 – Analysis of high-throughput sequencing data

*For the remainder of this lab session, you will work through a series of steps that could be applied to analyze a typical Illumina sequencing experiment. Typically, you would need access to a cluster or server with all of the described software installed on it. For simplicity, today we will simulate that situation be listing off the command line that would be used at each step, and having you act on only a small subset of the data to see what occurs at each step.*

Download the bundle containing the files that we are analyzing, and unzip it in a suitable location. The files are available on the course website.

*Quality control*

Given an initial set of fastq files from a sequencing run, you might run fastQC on those files as follows.

*fastqc reads.fastq*

The fastQC plot in fastqc\_example.pdf shows a run where several QC metrics failed. Inspect the results, consulting the fastQC documentation at <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> as needed.

**What seems to be the underlying cause for the QC problems? How would you recommend preprocessing the data before proceeding?**

The per-cycle base pair compositions indicate the presence of a single, strongly over-represented sequence. The “Over-represented sequences” heading shows that this is because >60% of all reads contain either adapter or PCR primer sequences, indicating strong contamination by adapter dimers. It is thus crucial that the sequences be subjected to adapter clipping prior to alignment.

*Assuming that your samples pass basic QC, you can now begin preprocessing them. As a first step, any residual adapter sequences should be removed. For the next several steps, we’ll be considering a set of fastq files named step1.fastq, step2.fastq, etc. While these files are small, containing only 10 reads, they will allow you to visually observe the changes occurring at each step of data processing. Note that in all of the upcoming stages, we are only scratching the surface of the capabilities of the tools that you are using; you should be sure to check out the documentation and example workflows before using them for anything serious.*

If you used standard Illumina TruSeq adapters, they can be removed with the command

cutadapt –a AGATCGGAAGAGC -o step\_2.fastq step\_1.fastq

*The sequence AGATCGGAAGAGC matches the double stranded portion of the TruSeq y linker, and thus will be present at the beginning of any incidents of read-through into an adapter.*

**Inspect the differences between the step\_1.fastq and step\_2.fastq files. How many reads had adapters clipped? Look at the step\_1.fastq file in a text editor and try manually clipping adapters. How many did you find? If there was a difference, carefully examine step\_2.fastq and explain why.**

Three reads have been clipped. However, searching manually for the adapter sequence yields only 2 reads. This is because read @K00135:141:HHJ3TBBXX:1:2228:6218:48983 has only part of the adapter sequence; however, cutadapt and most similar programs will trim any portion of the end of the read that could match adapter sequence.

*Next we will remove low quality portions of the reads to improve the alignment of the parts that remain.*

**Before doing quality trimming, manually inspect the step\_2.fastq file and consult the Wikipedia page on fastq files. Given that these reads came from an Illumina machine, what quality score encoding seems to be in use here?**

This appears to be in Illumina 1.8/phred+33 encoding, as quality characters like ! can only appear in this encoding, not in phred+64.

There are a variety of software tools to remove low quality portions of reads; here we use one called Trimmomatic. We can remove low quality read tails with the command:

TrimmomaticSE –phred33 step\_2.fastq step\_3.fastq TRAILING:3 SLIDINGWINDOW:4:15

*The trimmed reads will be output to step\_3.fastq*

**How many reads were trimmed by Trimmomatic?**

Three reads were trimmed – the low quality regions of those reads should also be visually apparent when you check the reads in step\_2.fastq

Try running a nucleotide blast search against the nr database using the read labeled @K00135:141:HHJ3TBBXX:1:2228:13524:49054, both before and after trimmomatic.

**What is the E value for the top blast hit with the low quality portion of that read included? What about after quality trimming?**

The E value is 0.008 with the low quality portion included and 0.001 with the low quality portion trimmed.

Finally, we are ready to align our reads. A huge variety of tools are available for this purpose, and as they are highly application specific, you should check the literature and consult an expert for dealing with your particular application.

As the reads we are considering here are RNA-seq data, we will use a splicing-aware aligner to map them to the genome. We can run hisat2, part of the new tuxedo suite (the replacement for the famous bowtie-tophat-cufflinks combo), as follows:

hisat2 -x /path/to/fly/genome/files –U step\_3.fastq -S aligned.sam

*The command line arguments are, in order, a path to the annotation files for the organism that we are aligning to, the input reads (which can also be paired end), and the name of the output sam file.*

*The results of aligning the ten reads from step\_3.fastq to the D. melanogaster transcriptome are present in a sam file in step4.sam. The sam file format is the standard for storage of aligned reads; a very useful reference manual at* [*https://samtools.github.io/hts-specs/SAMv1.pdf*](https://samtools.github.io/hts-specs/SAMv1.pdf) *and gives all of the gory details.*

For now, we will focus on a few aspects of the sam file. First, open the sam alignment in a text editor. The lines beginning with ‘@’ are header lines, and can be skipped – they describe the alignment and the chromosomes present in the reference genome. Scroll down to the lines that start with K00135 – these are the actual aligned reads.

*The third field in each of the alignment lines tells which chromosome the sequence aligned to; a \* will be present for reads that did not align.*

**What fraction of the reads aligned successfully?**

60% of reads aligned successfully

**Choose the first of the reads that did not align successfully, run a blast search for it, and record what organism it came from. Why might this have happened?**

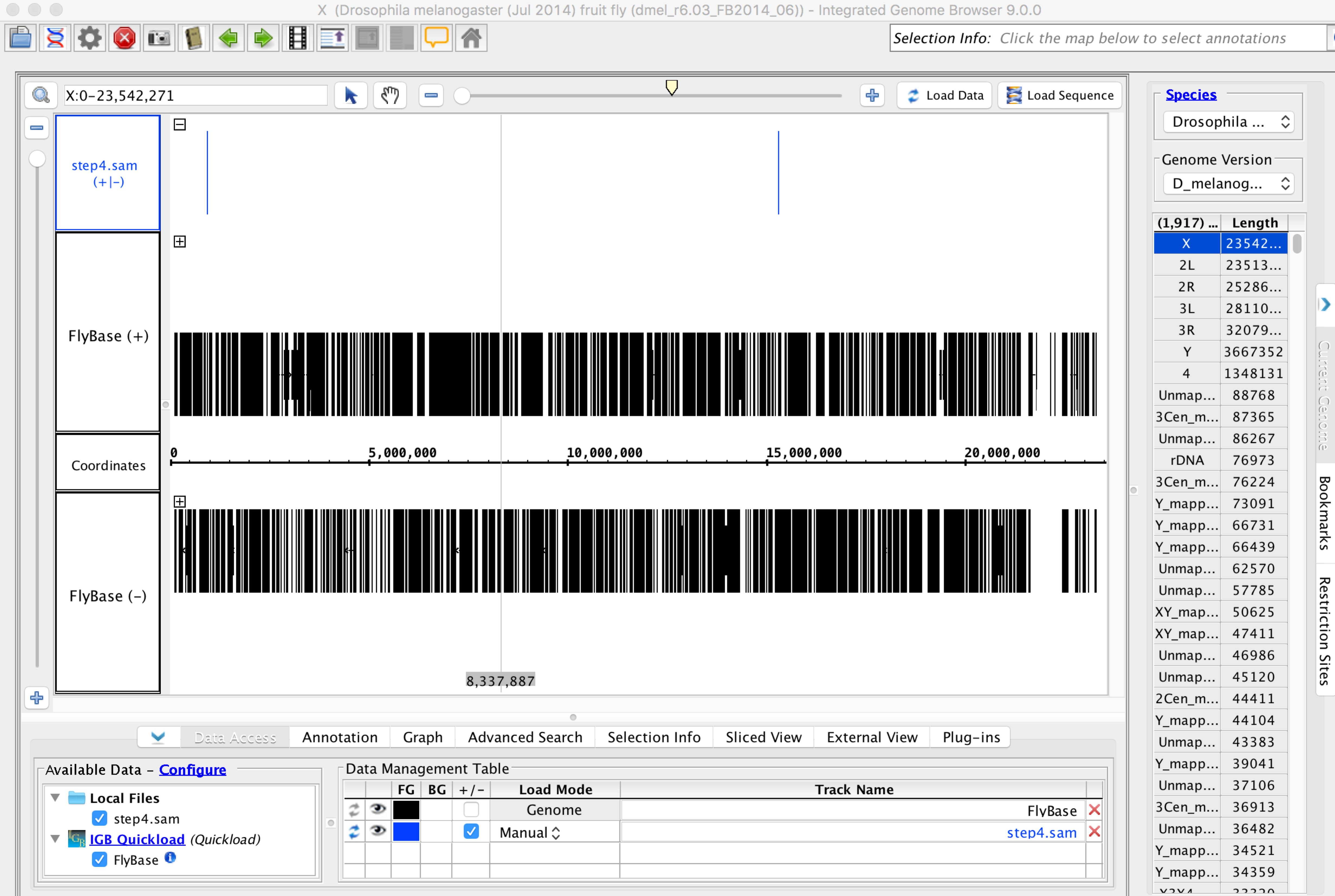
Read K00135:141:HHJ3TBBXX:1:2228:16163:48913 aligns to the genome of Staphylococcus epidermidis. As the name suggests, this is a common skin bacterium, and likely reflects contamination of the sequencing prep by someone who wasn’t careful enough putting their gloves on.

*Now we will use a genome browser to inspect the alignments and observe where the genes ended up. Note that we would typically have a LOT more reads to observe. There are tutorials for IGB available online, and I highly encourage you to try one. You should also try out IGV; preferences for one or the other vary from person to person. For now we will focus on demonstrating a few key features.*

Open IGB, and select the *D. melanogaster* genome from the menu that appears. Then to load your alignments, go to File->Load File and select step4.sam

Initially, IGB will not load the data to conserve memory. You can push the “Load Data” button in the top right of the display to actually read in the transcript locations.

Now, navigate to the alignment location of the read labeled K00135:141:HHJ3TBBXX:1:2228:17675:48966. You can do this either by clicking the location that you want to go to in the IGB window, scrolling in the IGB window, or typing a specific location (beginning with the chromosome) into the box at the top left of the interface:



Note that you can find the location on the chromosome of each alignment in the SAM file in the fourth field of the line.

**What gene did read K00135:141:HHJ3TBBXX:1:2228:17675:48966 align to? (you may need to zoom in, using the slider to the right of the location entry box shown above, in order to see anything)**

CG3699

(note that this is easiest if you look at the location of that read in step4.sam and then zoom in on that exact location)

*As a final step, we will observe some ChIP-seq data using IGB. The file chipseq.gr contains the enrichment of reads from a ChIP-seq experiment in E. coli, compared to an input sample.*

Close IGB and re-open it. This time select the E coli K12 MG1655 genome from the startup menu. Then load the chipseq.gr file into your data set. Any areas above zero in this trace show an enrichment of DNA in the antibody pulldown compared with an input sample.

**Consider for our purposes that a ‘peak’ (likely binding site) in this data is given by a region of at least 100 bp with an enrichment score above 3. Find a peak in the data, give its location, and speculate what gene might be regulated by that binding site.**

There are many possible answers. Peaks can be found either visually, or by using the Graph->Thresholding feature to automatically apply the criteria that we specified. There is, for example, a large peak at 85,513, which appears to be regulating ilvI.