8/8 - Excellent!
###Topic 2:
Why is piping directly between programs faster than writing each consecutive output to the disk? Explain using information about computer hardware.

###Answer:

i) If piping from process A to B, then B can start working on the data that A has processed, before A is finished with all the data.

ii) Piping also avoids writing the intermediate product to disk (storage) and instead the information is kept in the computers memory. Keeping information in memory is faster than putting it in storage because memory is physically closer to the CPU (central processing unit), which processes the information, and because memory has parallel high bandwidth connections to the CPU while storage is connect by a serial cable that is much slower. Finally, traditional hard drives rely on magnetism and moving parts while memory is non-moving and relies on voltage, which means things can happen at nearly the speed of light.
Well said. 2/2
###Topic 3:
Try different filtering options for the GBS data (see http://prinseq.sourceforge.net/manual.html for options) and plot some QC graphs. Which options you would choose to implement if this was your data and why?

###Answer:
I would first trim the tag sequence off (-trim_left 5). Next I would trim off low quality reads at the ends of sequences(-trim_qual_left 10 -trim_qual_right 10). I would then trim off poly A/T tails (-trim_tail_left 5 -trim_tail_right 5). I would then filter out sequences with a high percentage of Ns (-ns_max_p 1). To remove long stretches of repeated bases I would then set a minimum entropy score (-lc_method entropy -lc_threshold 70). Finally, I would remove low quality sequences (-min_qual_mean 25). This leaves us with a pretty nice looking output, although there are still 3 sequences with poly A/T tails of ~10bp that I can't seem to get rid of!

perl ~/programs/prinseq-lite-0.20.4/prinseq-lite.pl -fastq GBS12_brds_Pi_197A2_100k_R1.fastq -fastq2 GBS12_brds_Pi_197A2_100k_R2.fastq -log log1 -out_good GBS_filter1 -trim_left 5 -trim_qual_left 10 -trim_qual_right 10 -trim_tail_left 5 -trim_tail_right 5 -ns_max_p 1 -lc_method entropy -lc_threshold 70 -min_len 30 -min_qual_mean 25

2/2 - good although because it is GBS data I wouldn't worry about the poly As as these reads are from the genome vs mRNA

###Topic 4:
I want to reduce the percent of incorrectly mapped reads when using BWA. What setting or settings should I change in BWA?

###Answer:
There are a number of setting you could change in bwa-mem to reduce the proportion of incorrectly mapped reads. For instance, you could increase the minimum seed length (-k) so that incorrect mappings are less likely to be seeded in the first place. You could also lower the band width (-w), preventing the creation of large gaps. Or you could decrease the Z-dropoff (-d), which prevents poor alignments from forming during the Smith-Waterman algorithm. Another option is to lower the length of seed that triggers re-seeding (-r). Perhaps the most obvious choice is to change the SW scoring matrix: e.g., you could increase the mismatch penalty (-B), the gap open penalty (-O), or the gap extension penalty (-E). Then with a reasonably large clipping penalty (-L) incorrect reads will be removed.

Well thought out. 2/2

###Topic 5:
Quantify the assembly metrics for your first assembly that you ran without any options (sa_assembly21). Pick two or three different sets of parameters to run. Compare the resulting assemblies with one another and discuss which ones seemed to have improved the assembly and why that might be.

###Answer:
I played around with a few setttings (-min_contig_lgth, -scaffolding, -ins_length) but the one that had the largest impact was coverage cutoff (-cov_cutoff). Increasing minimum coverage obviously reduces the number of sequences assembled and the total length of the assembled region. However, an intermediate cutoff maximizes the average length of a contig and the n50 (and hence improves assembly most). An intermediate coverage cutoff is presumably helping by removing errorenous contigs.

2/2 - good the high coverage cut-off results in a bad assembly because it is much higher than the expected coverage cut off so few regions will actually meet that high coverage level

-cov_cutoff 0

File	Number of sequences	Average length	Total length	Median	min	max	n50
contigs.fa	89110	75.059645382112	6688565	65	41	967	74

-cov_cutoff 10

File	Number of sequences	Average length	Total length	Median	min	max	n50
contigs.fa	2615	1063.75564053537	2781721	534	41	11869	2330

-cov_cutoff 100

File	Number of sequences	Average length	Total length	Median	min	max	n50
contigs.fa	83	292.012048192771	24237	164	41	2217	566