Assignment 3 sample solution.

Task 1

1. For a diploid the main copy number would have to be 2. The loss states would be 0 and 1 copy. In order to detect copy number gains, we could at least need a state with copy number 3 or higher. Any copy number is possible for a gain, but a HMM can only support a finite number of states. **(1 mark)**
2. There could be any number of copy gains in a given genome. In order to detect these accurately, states would have to be supplied by the model: 3, 4, 5, … etc. However, the complexity of the Viterbi algorithm scales quadratically with the number of states, so adding additional states is a large computational burden. One might be better off to have rough states (loss, normal, gain), and then analyse after HMM classification in more detail. **(0.5 marks)**
3. The data in the figure shows two clearly defined peak, each looking roughly normally distributed. The width of the peaks can be used to parametrise a probability function for emissions: For example, the emissions for the 2 copy state would be maximal at a ratio of 1, and only few emissions are to be observed lower than 0.8 and larger than 1.2. Probabilities can be obtained directly from the density function or by sampling the function in a discrete fashion. One should be mindful to not introduce 0 probabilities for emissions, however, as it could make the Viterbi algorithm unsolvable.
   Once the general emission spectrum has been determined for 2 copies, the same parameters can be scaled to other copy number states (centrering the same curve around 0.5, 1.5, 2, … etc).
   The peaks centering around 1 and 0.5 is a general property of any read depth profile. One might expect additional peaks forming at 1.5, 2 etc if such copy number gains are present. Specific to this data is the width of the peaks. It is influenced by the sequencing process (uniformity of read depth) and the bin size (larger bins leading to less variance). Finally, the height difference for the different peaks (here ratio 0.5 and ratio 1) can be used to estimate how much of the genome is of copy number 1 and 2 respectively and set the transition probabilities respectively. However, since the 0.5 peak is likely due to chrX, no meaningful transition probabilities for other chromosomes could be deduced. **(1.5 marks)**
4. The data shown for Task 3 would not be easy to classify by a HMM with states for CP 0,1,2,3,4, since its data point are in between those states. This would cause the model to have unfavourable emission probabilities in any state. Further, The different segments of changed copy number between 90-140Mbp cannot be differentiated by the above HMM method and would likely be classified as CP1 all across.
   One could imagine setting copy number states to the levels appropriate to the levels observed in this data, but this would not be a generally applicable method anymore. What would happen on other chromosomes, which have segments at different copy number yet again? What would happen for an entirely different data set? **(1 mark)**

Task 2

Sample solution by Kinsey Reeves. See LMS for code.

The algorithm implemented as described in Olshen et al's paper[1] segments copy number changes into different intervals along the binned data. It recursively cuts segments until no segment can find a z value greater than the threshold Z. An appropriate threshold value must be chosen such that it picks up large enough segments whilst ignoring small variations due to random noise.
The complexity of CBS is $O(n^2 \log(n))$. This can be explained by a normal binary tree depth traversal (segmenting) giving the log(n) and for each segment we must traverse it $n*n$ times. Even though we segment, the size of each segment we have split is still n. On every recursive split we once again search through all i and j values. This is a common implementation of a divide and conquer method. The amount of segmenting will be lower as the threshold is raised. In the tumor example provided it only does very few segments for z=10. (3 found 1 discarded). As we lower our Z threshold our runtime will increase as we will pick up segments which deviate from the median at a higher precision. The n^2 will also be practically faster than O runtime as $n>j>i$ for each iteration.

Injection:
A lot of students argued for log n recursions of the method.
This is intuitively and practically sensible. In reality however, each bin could end up in its own segment, making for a total of n segments.

My implementation can be divided into 3 steps. First, the CBS algorithm splices segments and then recursively searches them for further segments with a sufficient Z value. If it does not find one, it will output this whole segment into a staging array. Secondly, from this array segments are discarded if their average absolute log ratio value is less than 0.1. Finally, we must check for contiguous segments in the output as some segments are circularised and therefore spliced out from others.
Once segments are spliced we output the average log ratio, start and end points and the segment number. The final output is required but not in the spec, as it shows that although a segment may not be contigous it has the same log ratio. We can see that the example in Figure 1 shows us the 3 segments called, however one has been split into two as it is a non contigious segment. The log ratios are reported as the average log ratio of a contiguous segment. We see these reported in Figure 2.

CHR	START	END	RATIO	SEG NO.
5	0	94500000	-0.14	2
5	94500000	94550000	-1.19	1
5	94550000	124600000	-1.68	0
5	124600000	147950000	-0.92	1

Figure 1 - Outputs from CBS

Note: The last column is not in the provided spec. But it is necessary to see the groupings of the segments. These numbers are arbitrary and are to show that this segment was found together with the same z value.
The log ratio reported is the average calculated ratio after contiguous segmentation occurs, see lines 2 and 4 having different ratios. This excludes 150000000 to the end shown in Figure 2. This

[1] Olshen AB, Circular binary segmentation for the analysis of array-based DNA copy number data. Department of Epidemiology and Biostatistics

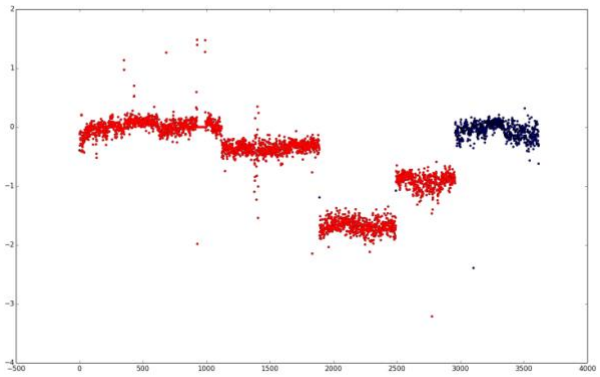
was chosen at it helps cull unecessary segments. Taking the average from two segments which are't adjacent didn't make sense.

Task 3

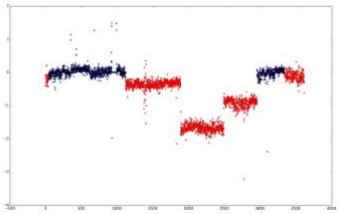
Sample solution by Haonan Li.

4 Task 3

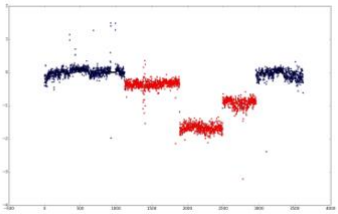
1. Figure 2 visualize the reported CNVs. The result of Z equals 10, 1 and 8 shown in Figure 2(a), 2(b), 2(c) separately. In these figures, the x axis represent the index of bins, the x coordinate times 5000 and we can get the corresponding positions in reference. The y axis is the log ratio of each bin.



(a) Z= 10



(b) Z= 1



(c) Z= 8

Figure 2: Visualization of CNV. (The red part is reported CNVs)

2. For $Z = 10$, as shown in Figure 2(a), we find that no all of the identified CNVs real. The first part, x form 0 to 1800 (0-90Mbp) is not a real CNV, in fact, only x from 1000 to 1800 (50-90Mbp) should be a CNV. In addition, x from 3500 to 3600 (175Mbp-180Mbp) might be false negatives. The average logR is lower than -0.1 and this part does not been detected. To increase sensitivity, we can decrease the Z-threshold, this will lead to more segmentations of the bins meanwhile decreasing the average range of each bins. See Figure 2(c) and 2(b).
3. Biology analyze of three large CNVs in the data:
   a. 50-90M bp

$$logR = -0.3$$

$$observed\ copy\ number = 2^{1-0.3} = 1.6$$

assume $clonality = c$, with $CN = x$, there is:

$$xc + 2(1 - c) = 1.6$$

possible (x, c) pairs:

$$x = 1, c = 0.4$$

$$x = 0, c = 0.2$$

b. 90-125Mbp

$$logR = -1.3$$

$$observed\ copy\ number = 2^{1-1.3} = 0.8$$

assume $clonality = c$, with $CN = x$, there is:

$$xc + 2(1 - c) = 0.8$$

possible (x, c) pairs:

$$x = 0, c = 0.6$$

c. 125-140Mbp

$$logR = -0.8$$

$$observed\ copy\ number = 2^{1-0.3} = 1.1$$

assume $clonality = c$, with $CN = x$, there is:

$$xc + 2(1 - c) = 1.1$$

possible (x, c) pairs:

$$x = 1, c = 0.9$$

$$x = 0, c = 0.45$$

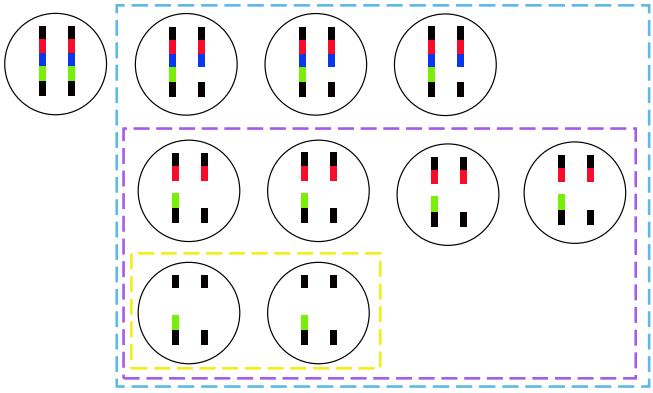
Based on the computed result above, the first and third changes could be both single or double copy losses of the DNA and the second change must be double copy losses of the DNA. Assume three CNVs happens independently on two clones. There are four possible cases for these three CNVs, show in Table 4

case#	x_1	c_1	x_2	c_2	x_3	c_3
1	0	0.2	0	0.6	0	0.45
2	0	0.2	0	0.6	1	0.9
3	1	0.4	0	0.6	0	0.45
4	1	0.4	0	0.6	1	0.9

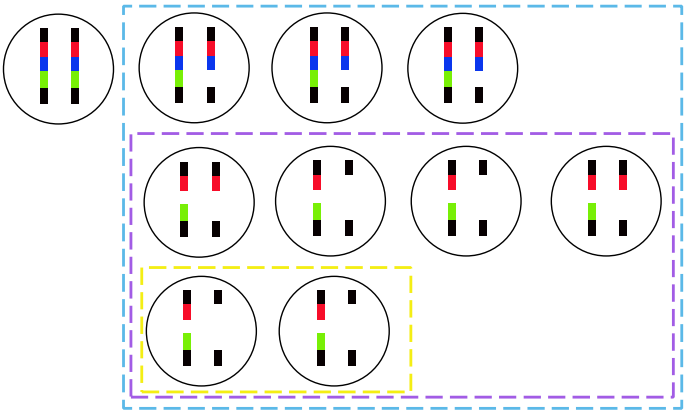
From the table we can find the first and second changes may happened in different cells because $c_1 + c_2$ no larger than 1, while there must be some cells contains the second and third changes because $c_2 + c_3$ always larger than 1. For the first and third changes, they may happened in different cell cause there are case that $c_1 + c_3$ less than 1.

As discussed in lecture, BAF could be computed from original BAM file to substantiate our theory.

Actually, there are too many possible clonal structures and possible distribution in the overall population of cells. Figure 3 shows two possible results. From the figure we find subclone must exist in these cells. For example, in Figure 3(a), 90 percent cells lose part 3 (the third change) on one chromosome and within these cell, some of them lose part 1 and 2 (first and second changes).



(a) Totally 10 cells. 1 (10%) of total is good. 3 (90%) of total lose the third changes on one chromsome (within azure square). 6 (60%) of total lose second changes on both chromosomes (within purple square) and 2 (20%) of total lose first part on both chromosomes(within yellow square).



(b) Totally 10 cells. 1 (10%) of total is good. 3 (90%) of total lose the third changes on one chromsome. 6 (60%) of total lose second changes on both chromosomes and 4 (40%) of total lose first part on one chromosome.

Figure 3: Possible Clonal structure and its distribution, The red, blue and green part represent first, second and third changes separately. The proportion of each kind of cells represent the proportion in overall population of cells.