Computational Genomics Project 3 – Kinsey Reeves

Task 1- HMM's to predict Copy number variation

1.

The states in the HMM would be most probable at 2 with other states at 1, 0, 3 as a mutated cell could lose one or both copies of its DNA and have other parts inserted into it in either one chromosome or both. Our emission probabilities would change as would our transition probabilities having 2 being the most likely state and deviations away from it with higher probabilities.

2.

Increasing the sensitivity of a HMM will increase the complexity as there will be more states to traverse. The complexity of the viterbi algorithm is O(nm^2) where n is the length of the sequence and m is the number of states. For every step in the sequence, we must check a state against all others to calculate the transition probabilities. Increasing the sensitivity will generate more states and will therefore increase the complexity / run time.

3.

The emission probabilities are the probability that a certain segment is a copy number state. We must associate each peak in our distribution to a copy number. Here, the large peak in figure 1 will be CP2 as it is a diploid organism and having a read on both chromosomes present is the most likely situation in a normal cell. So the most likely state for any read would be CP2 at the highest peak in the data. Using Figure 1, we could create two probability distributions based on log ratio of read depth. One at Cp1 and one at Cp2.

The highest peak in general will be the normal read depth. In a normal cell the highest peak will indicate the normal read depth as it is the most probable. I.e. the peak on diploid data should correspond to the copy number of 2 due to two copies each from each side of the chromosome. Other peaks will show when insertions or deletions have occurred. E.g. a peak associated with cp1 will be a deletion and cp3 an insertion. This data shows us that the bins with ratios between 0.8 and 1.2 are most common. The most common mutation is a loss of about half of the chromosome / genomic data indicated by the other peak at Cp1.

We can derive the transition probabilites for a detection HMM by seeing how likely a sequence is to be in a certain copy number state in a normal setting and then in a mutated setting. In this example, we can see that it's more probable to be in Cp2. So, our state transition probablities would be low from Cp2 to Cp1, and then more probable from Cp1 back to Cp2. It would also be high for Cp2 to Cp2 and low for Cp1 to Cp1.

Non clonal data:

We can't train a strong model to fit data if we don't have strict training data in the first place. Using data from clonal cells means all the cells are the same and have the same chromosomes. In a population where cells having different ratios of missing or present copy number segments means a segment isn't in a CN state or not. It could be in both as there are some cells with this segment present and others without it. Yes we could represent states as non integer, meaning a segment could be represented as in two states at the same time with different probabilities.

Task 2. Circular Binary Segmentation Algorithm Implementation

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.

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

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

Task 3 – Analysis of the results

1.

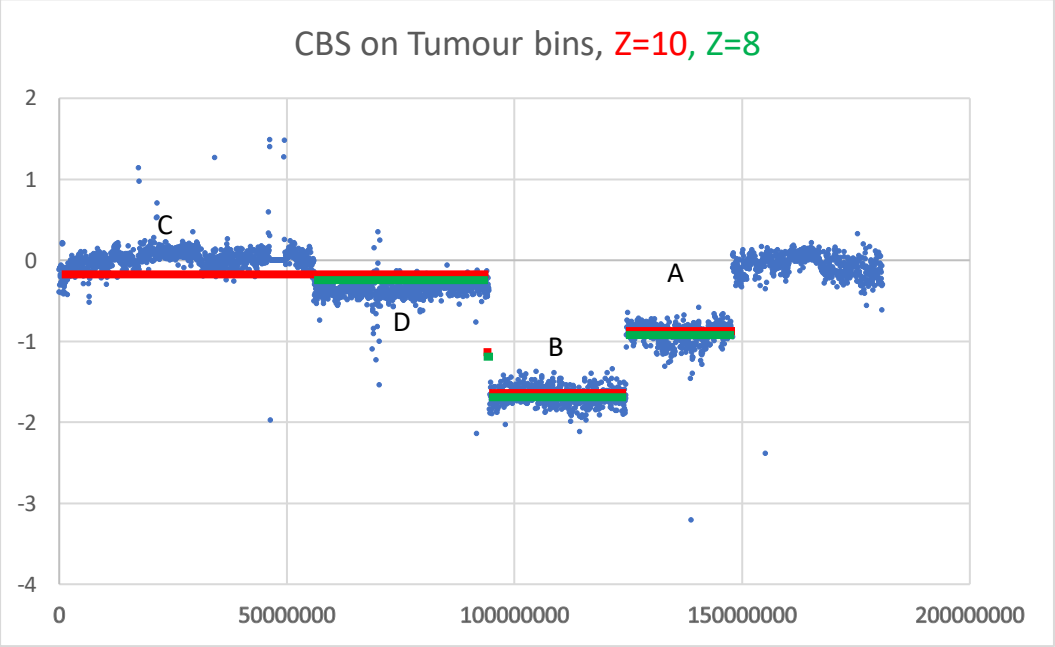

Figure 2 - CBS outputs overlayed on binned log ratios of read depth generated in excel.

We see above that the above datapoints correspond to the data provided in figure 1. However, we see that it may have wrongly called some copy number change segments using our Z = 10. There is a segment from the start to B and from the end of A to the end which is wrongly called. Half of it is culled due to having a log ratio < 0.1. Using Z =8 seems to much more closely fit the obvious segments.

The segment between 0 and 94.5mbp (C) is called incorrectly as it is visually edivent that there is unlikely to be a copy number change in the area between 0 and 56mbp. However, it has taken the average of the whole segment from 0 to 94.5mbp, due to not having Z low enough, and therefore the average has been lowered to over 0.1 from the region at D.

From here the two most obvious segments are called at A and B. However, looking at figure 1, the sement which begins (having only a single bin) at 94.5mbp to 94.5mbp and continues at 124mbp has been called this way due to the way CBS circularises the DNA. As the end of A is joined to the start of B prior to it being segmented, and this point fits more closely to the average of A than it does B. This point may actually have the same copy number as segment A, or it may be random noise.

Using a Z value of 10, we see that it gives us a false positive of a cnv in the region from 0 to ~50mbp. This is due to the region being calculated on the average value, which is lowered between 56mbp and 94 mbp (D). We can see that lowering the Z value to 8 seems to fit the 3 more obvious segments at D, B and A and rightly does not classify C as a CNV. This parameter would suit better the data provided as it picks up on the obvious changes in the data.

Median 1st third		Cp1	Cp2	Cp3
133590		50-90mbp	90-125mbp	125-140mbp
	Read depth(avg)	109061	50398	74691
	% present	82%	38%	56%
1-(RD/Med)	% lost	18%	62%	44%

Figure 3 - Estimated segment presence in total cell population

We can see that these 3 segments were lost in varying amounts in the cells throughout the sample. A greater absolute log value would mean that a greater amount of DNA was lost in this case. These events are not always independent. We can't say how often these mutations occur together however it is likely that a mutation caused one loss of DNA, then a further mutation from a cell in this population caused a further loss in DNA. So some cells would be missing only Cp2, then descendents of these cells may be missing Cp2 and Cp1.

Figure 3 shows us the estimated presence of the DNA region in the population of cells. These values were calculated as the proportion of the average binned read depth out of the median of the first third of the data read depth. We see that 18% of cells were missing CP1, 62% Cp2 and 44% Cp3. From this, we can deduce that the loss of these 3 segments are not independent as .62+.44+.8 > 1 and must have occurred in the same cells sometimes. Cp2 must have occurred on both copies of the chromosome in some cells as the proportion is above 50%. i.e. as we have a diploid organism, losing a Cp2 on a single chromosome in every cell would mean Cp2 is at 50%, as it is greater than this, the copy must have been lost from both chromosomes in some cells. We can see this more clearly in Figure 4. This could also be possible with Cp1 and Cp3 however we can't be positive from the data.

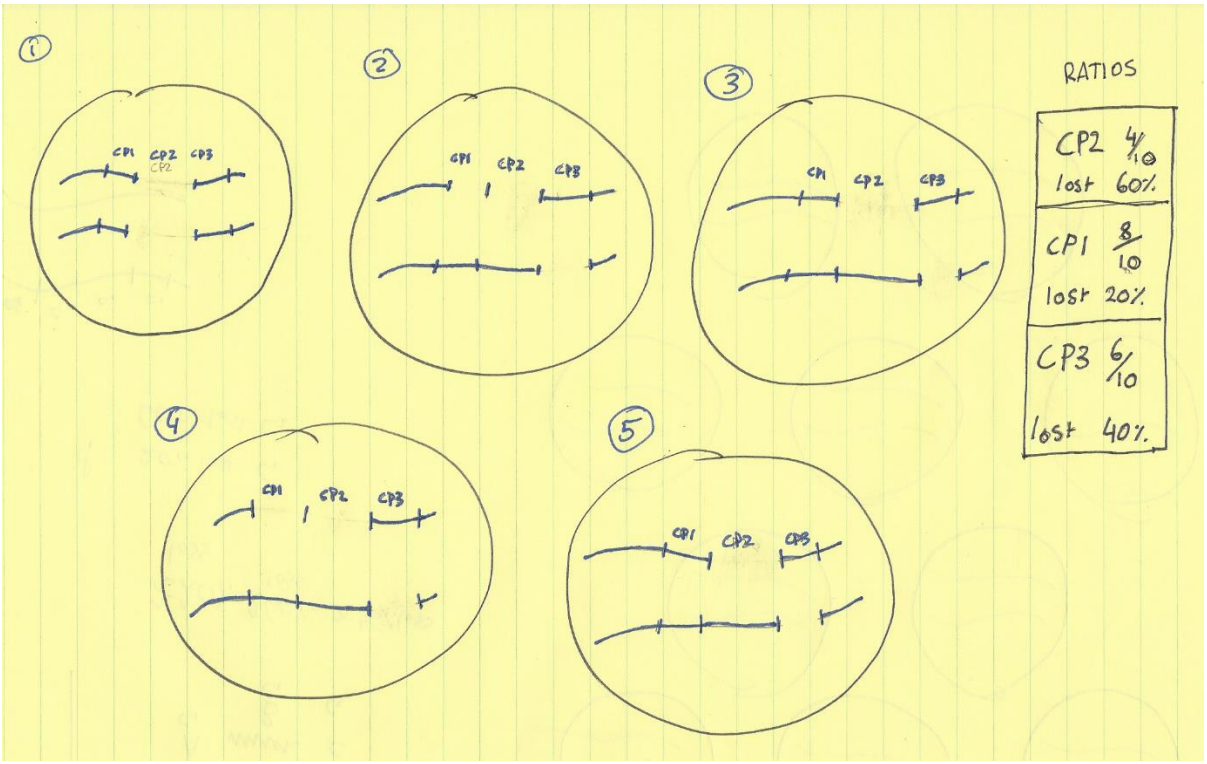

Figure 4 - Diagram of a cell population giving the same results

Figure 4 shows us a drawing of a population of 5 cells. Where the presence of copy number segments is meant to resemble similar ratios to those shown in Figure 3 and 2 with rounded aproximatations. We see that in each cell we have at least one copy of CP2 mising with some having both. As these are cancer cells, the inheritance of missing segments can occur together. I.e. an already mutated cell mutates again

and loses both segments of DNA. Cell 3 mutating to cell 2 and 4 in figure 4 would be an example of this. This is why we see regions in our graphs look like steps, as it is possibly mutations happening ontop of mutations. However we cannot deduce this with certainty as all we are provided with are the log ratios of the read depths.