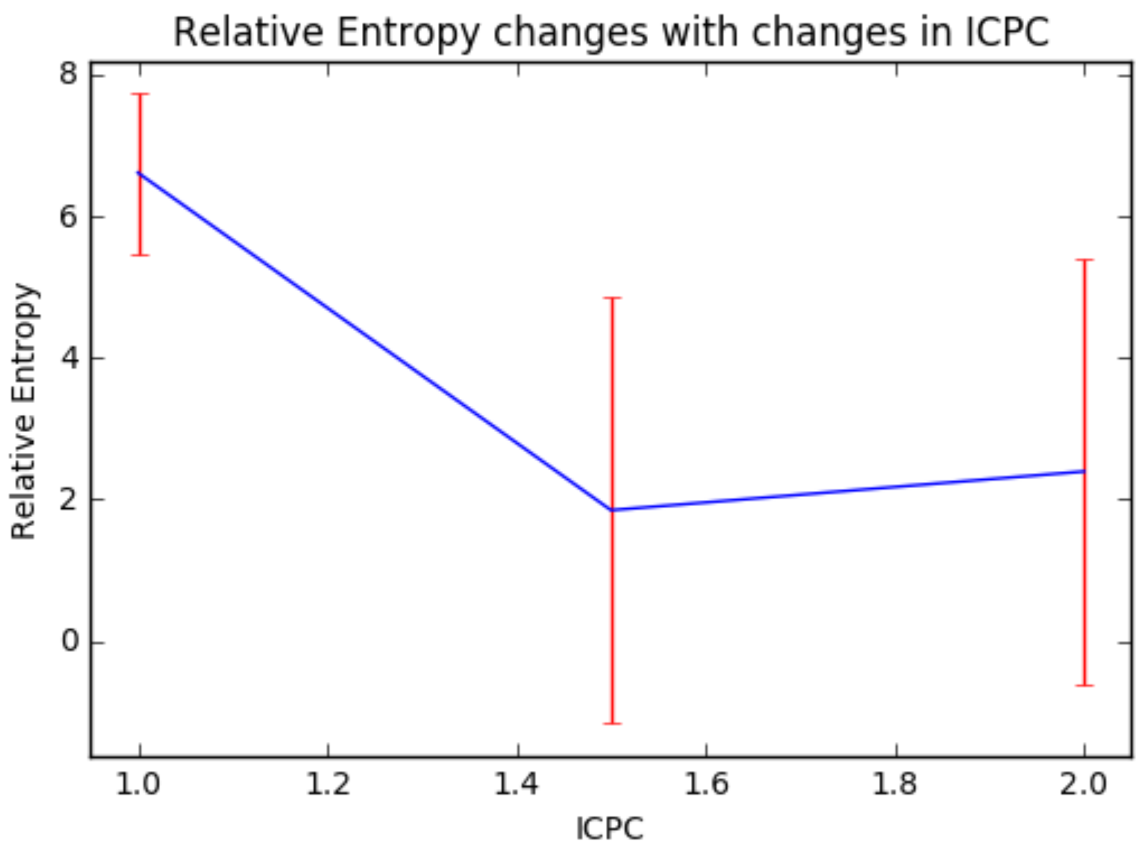
Evaluation Report

In this mini project, we generated parameter ICPC, ML and sequences with specific sampled position. Applying the motif finder to these sequences with different combination of parameters, we generated 70 PWM and recorded the starting sites of motif as well. In order to assess the motif finder, we compared the experimental finding with theoretical results by several methods.

Relative Entropy

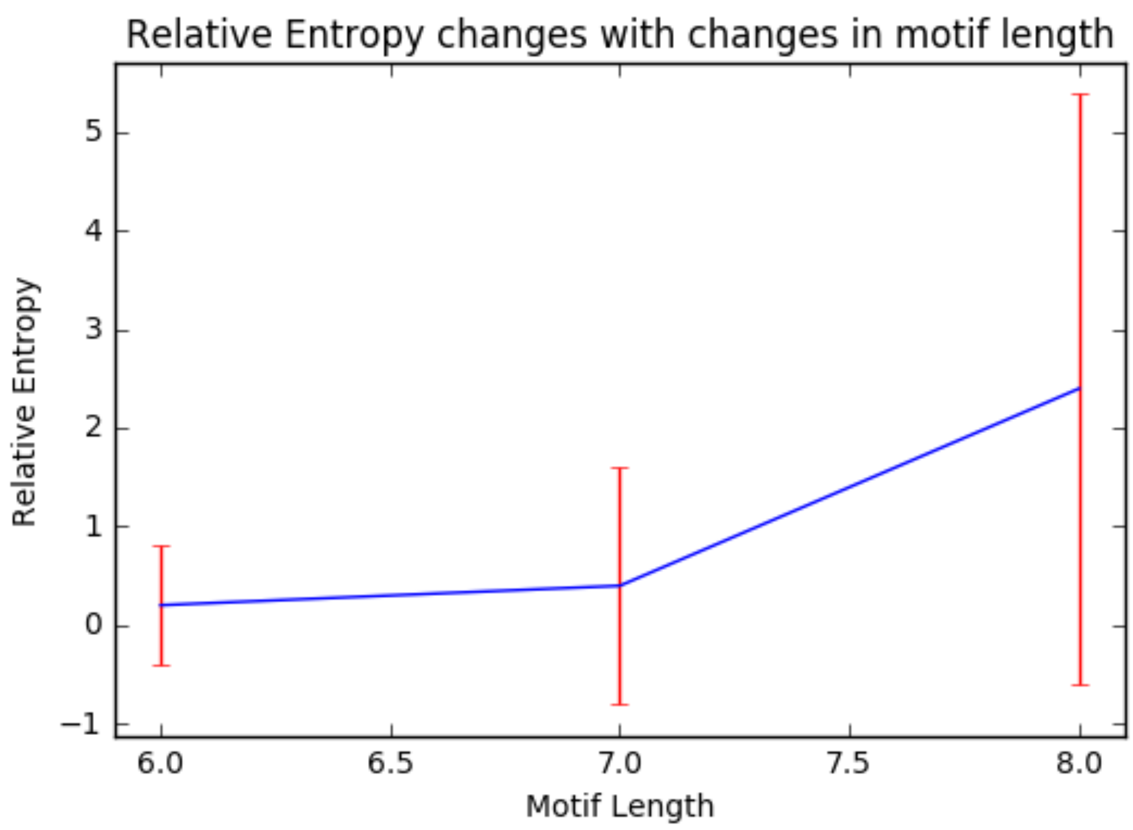
The first method is to evaluate how relative entropy between experimental PWM and theoretical PWM with respect to changes in different parameters, ICPC, motif length and sequence count. Since we have cases where the probabilities in the actual PWM is 0 but our generated PWM is non-zero, we cannot just use the relative entropy formula, $D_{KL}(P/E) = \sum_i P(i) log \frac{P(i)}{E(i)}$ (P is the predicted PWM and E is the experimental PWM). Instead, we calculated a pseudo relative entropy value where we added the relative entropy value calculated while ignoring the cases where the actual probability is zero and a weighted count of the number of these cases.

```
In [1]: import analysis
        analysis.rel_icpc()
```



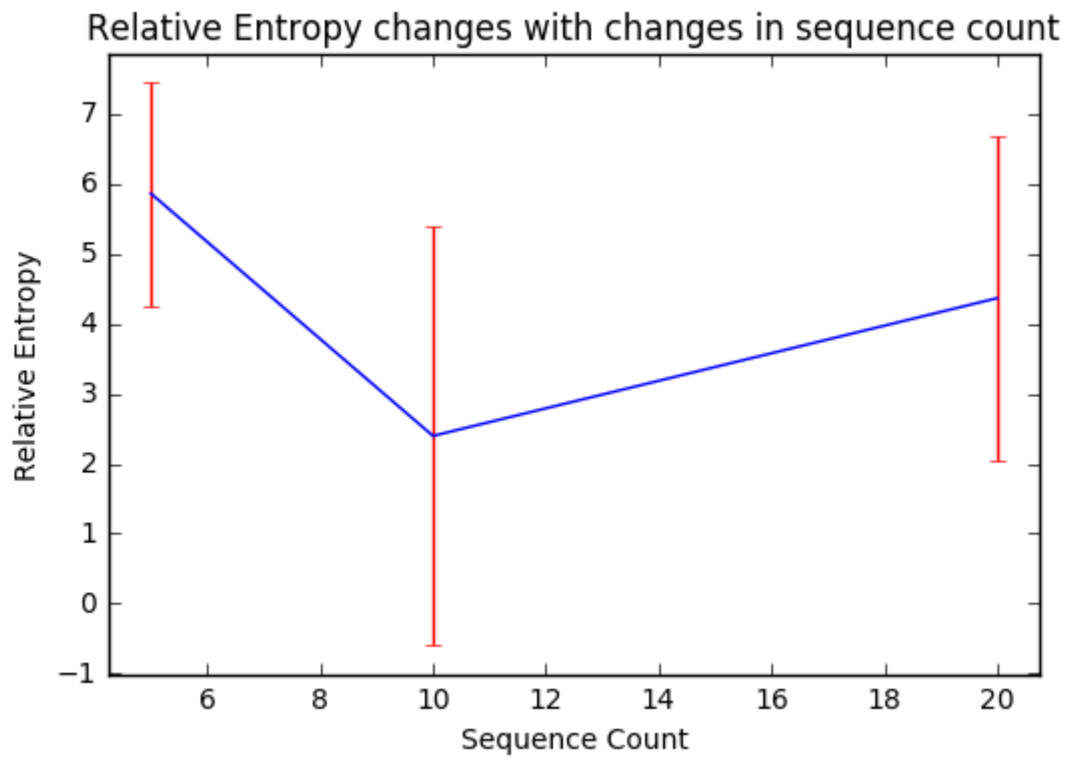
From this figure, we can observe relative entropy and standard deviation decreasing with an increase in ICPC, which means the experimental PWMs are more steady (shorter error bar) and more similar to the theoretical one. Thus, in the ICPC ranging from 1.0-2.0, the larger ICPC is, the better motif finding program performs in terms of PWM.

```
In [2]: analysis.rel_ml()
```



In this figure, our motif finder seems to have a better performance when the motif length is shorter. This is likely because it is difficult for the motif finder to randomly find longer sequences.

```
In [3]: analysis.rel_sc()
```



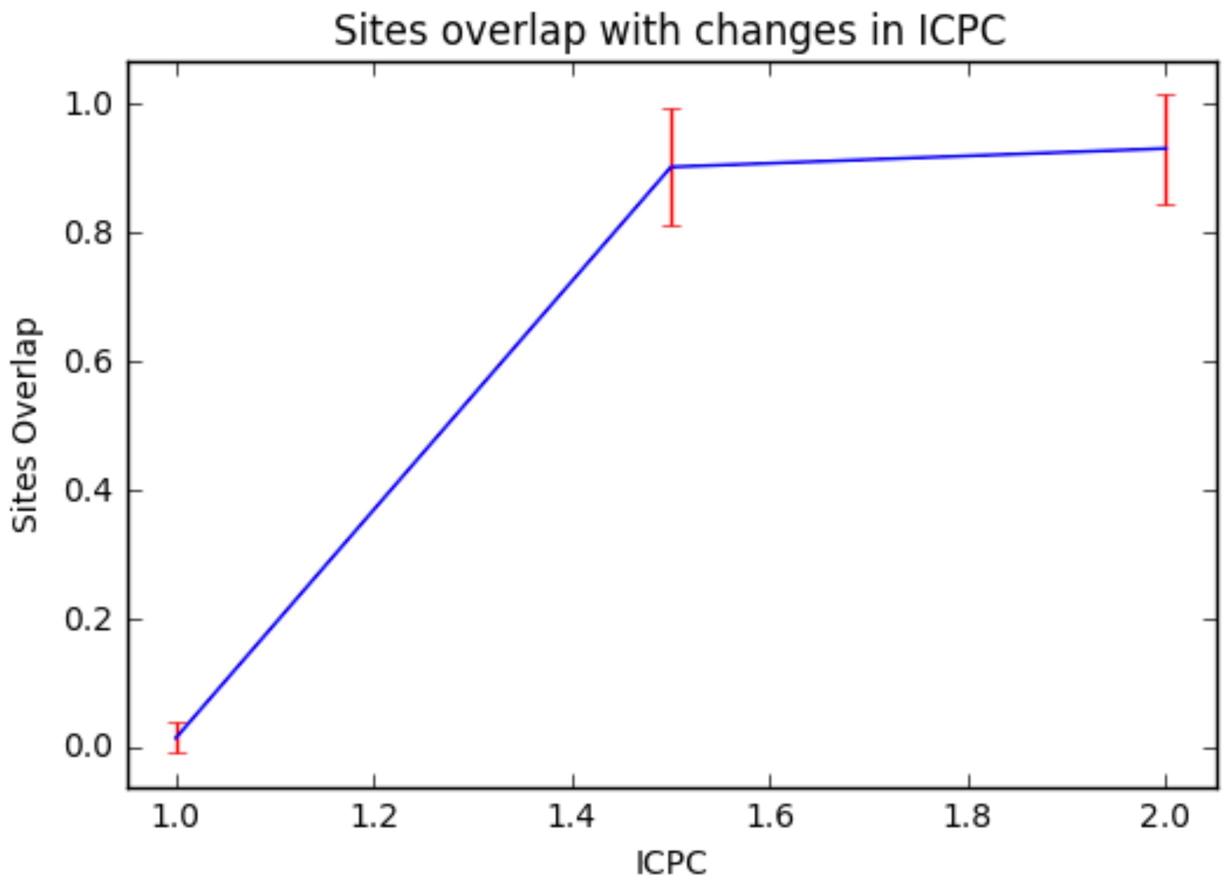
As can be seen, 10 is the best among the three sequence count according to relative entropy changes with changes in sequence count. More sequences inputting seems to have better accuracy but increases the difficulty as well. Thus, appropriate sequence count should be found both considering these two factors.

Sites Overlap

The second method focuses on the overlapped between experimental motif sites and theoretical motif sites. We calculated the site overlap proportion, for example, if the motif length is 8, the experimental starting site is 5, while the predicted starting site is 7, the overlap proportion should be 6/8. The following plots show the change in the amount of overlaps resulted from different ICPC, motif length and sequence counts.

In [4]:

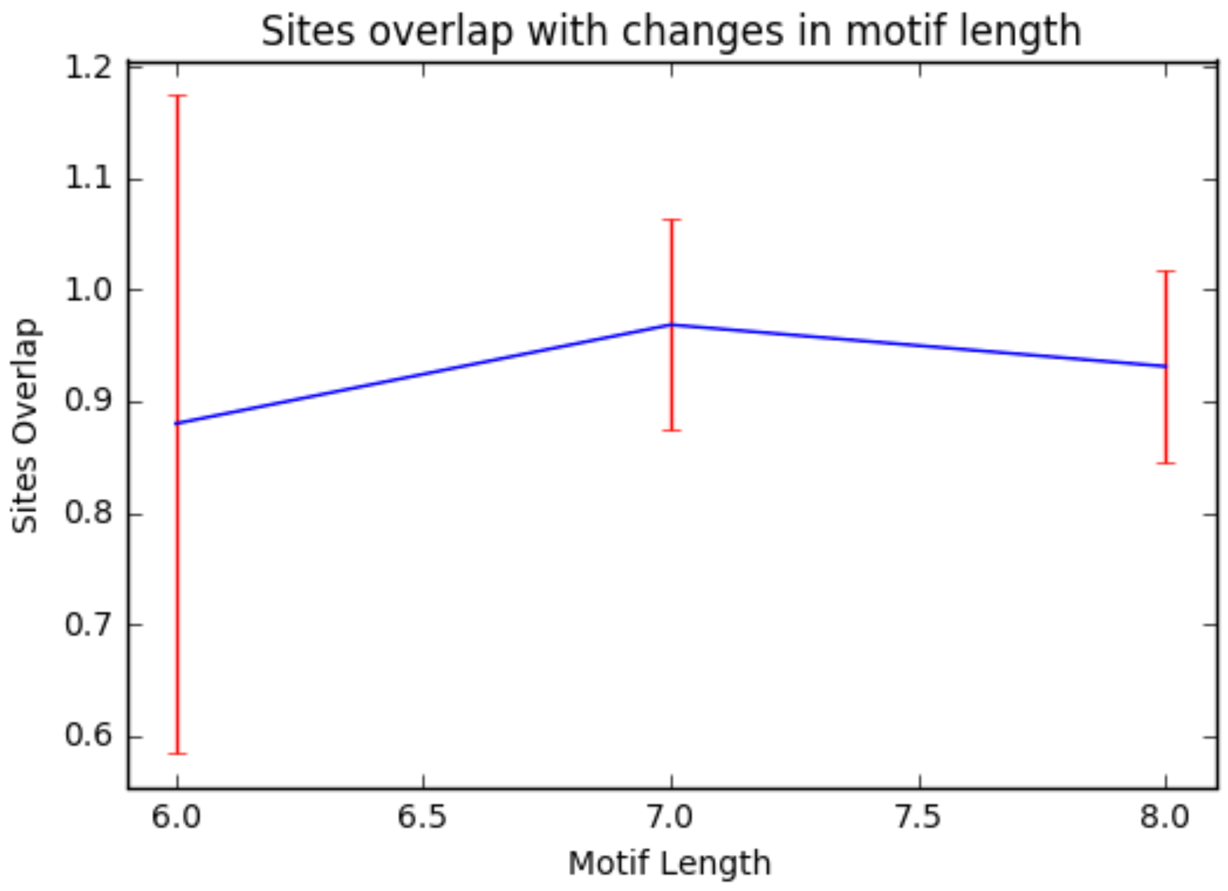
```
analysis.sites_icpc()
```



As observed, an increased in ICPC led to us being able to identify the motif better. As can be seen, when the ICPC is too low, we are nearly unable to locate any of the motifs.

In [5]:

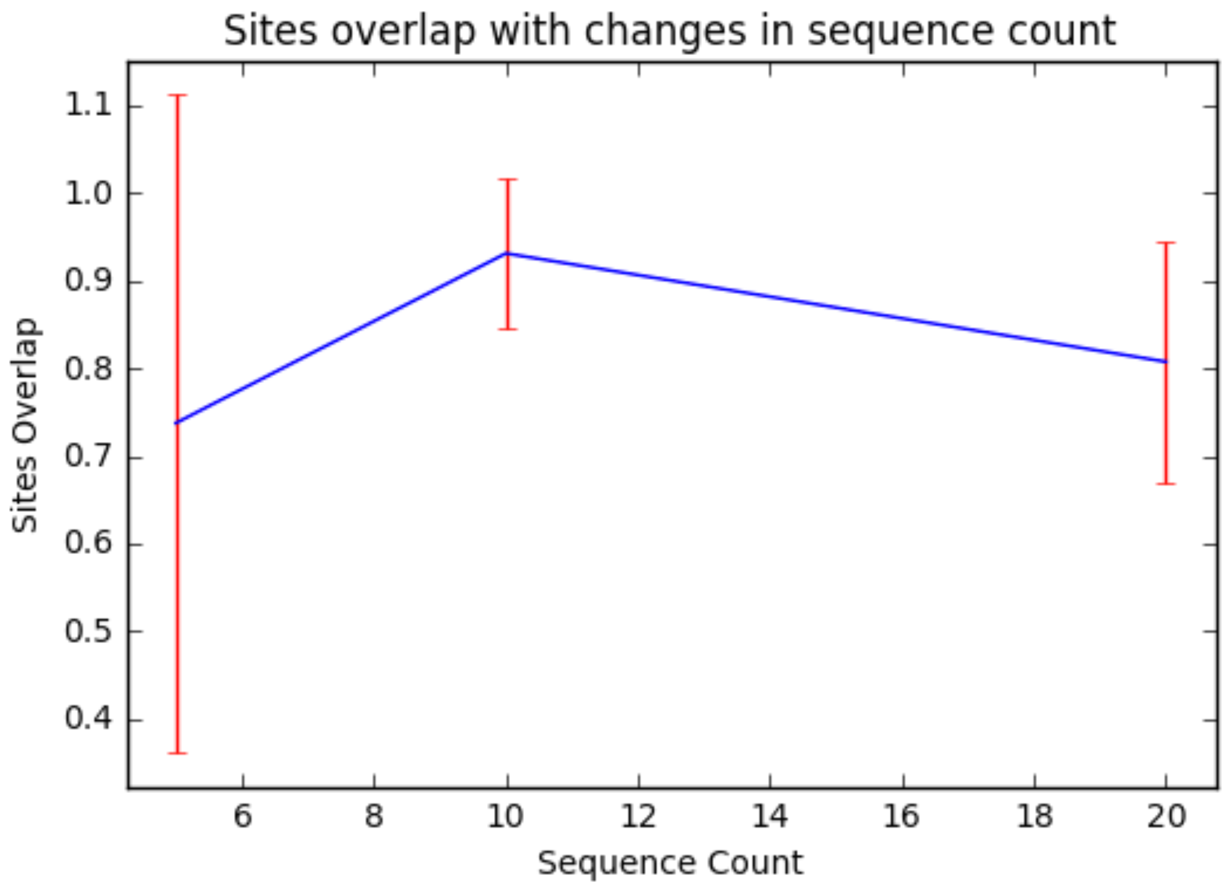
```
analysis.sites_ml()
```



As can be seen here, motif length equal to 7.0 seems to make the motif finding result better based on sites overlap which is the same in term of relative entropy.

In [6]:

```
analysis.sites_sc()
```



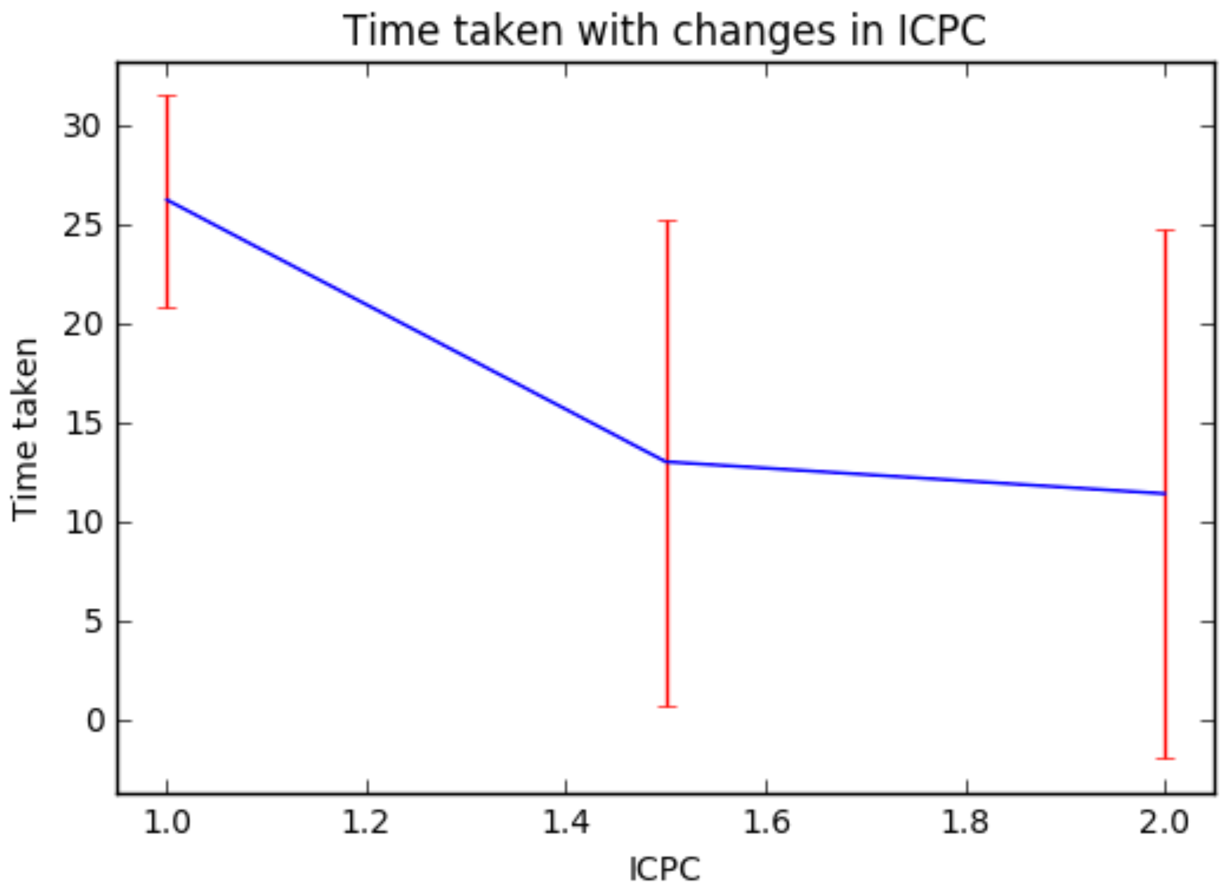
In terms of sites overlap with change in sequence count, the sequence account 10 fits the motif finder best which is also consistent with the analysis result based on relative entropy.

Time taken

Our third measure is to plot the time taken against changes in the various parameters same with the other two methods. And the time unit in the plots is second(s).

In [7]:

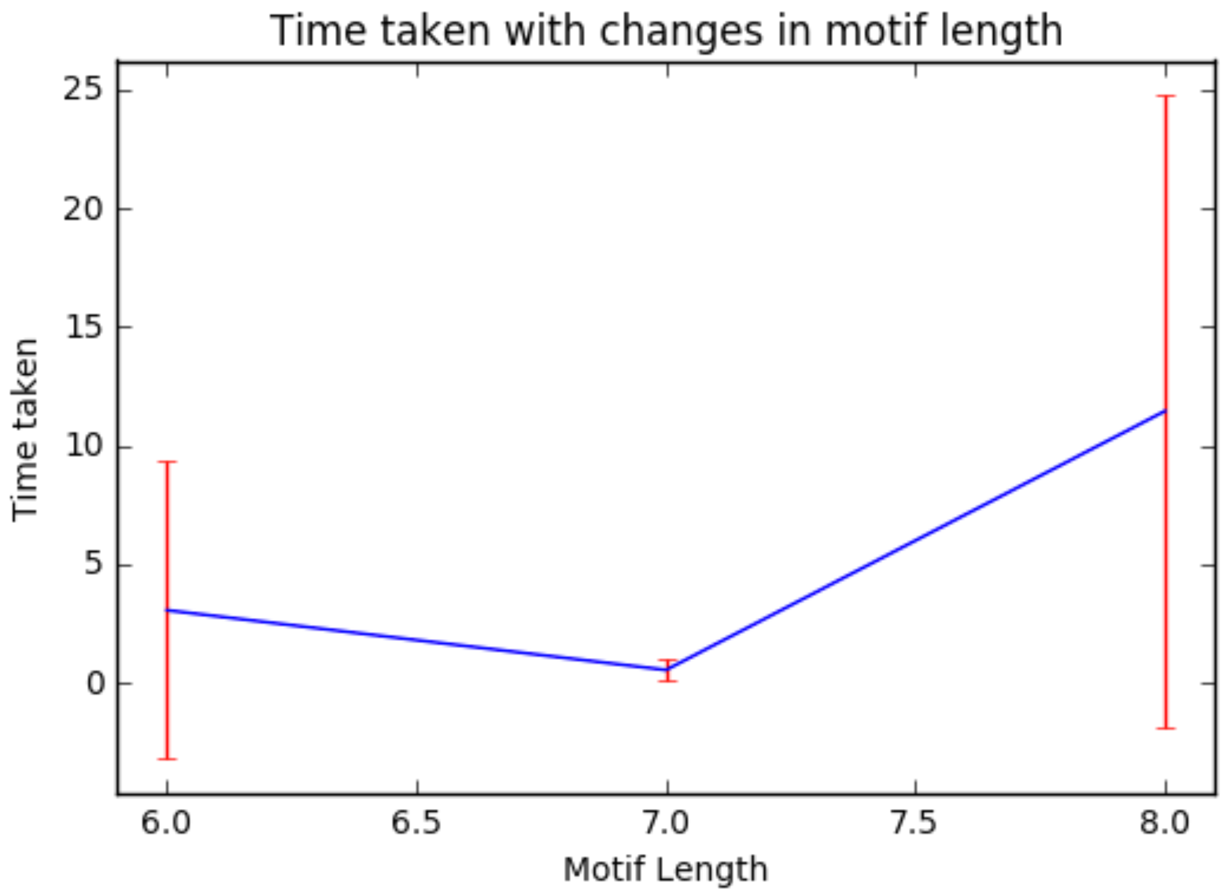
```
analysis.time_icpc()
```



With an increase in the information content, we can observe that the time taken to find each motif is going down. This is expected as with a greater ICPC, the algorithm will be more likely to alight on a satisfactory motif sooner. As our algorithm exits as soon as the current motif's score crosses a certain threshold, the higher information content allows us to return earlier.

In [8]:

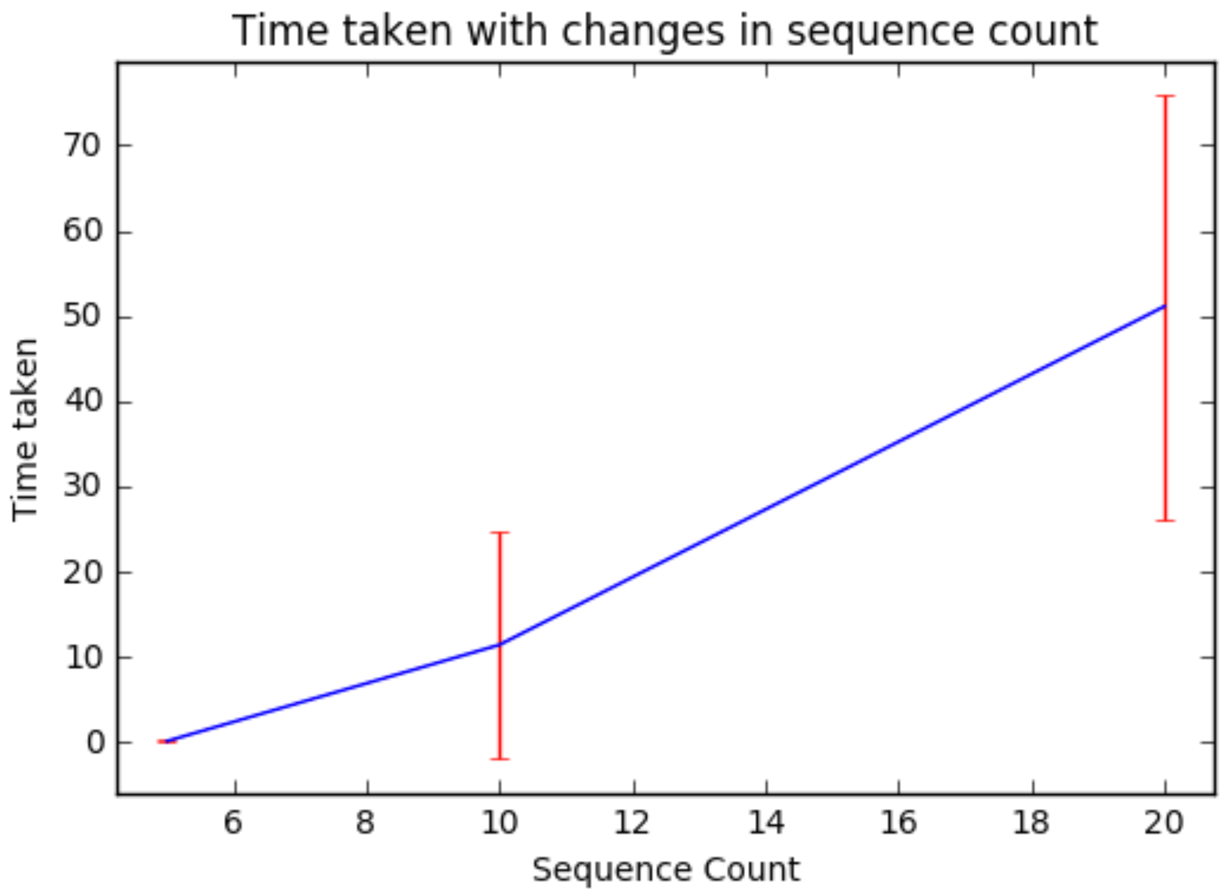
```
analysis.time_ml()
```



As can be seen, the longer the motif length, the longer the time taken for the algorithm to complete. This fits our expectation as well, since the algorithm now needs to check one more spot in each motif that it is generating.

In [9]:

```
analysis.time_sc()
```



As can be quite clearly seen here, the time taken increases as the number of sequences we are looking at increase. Since the algorithm is random, the more sequences there are, the harder it is for it to randomly find a shared motif between all the sequences.

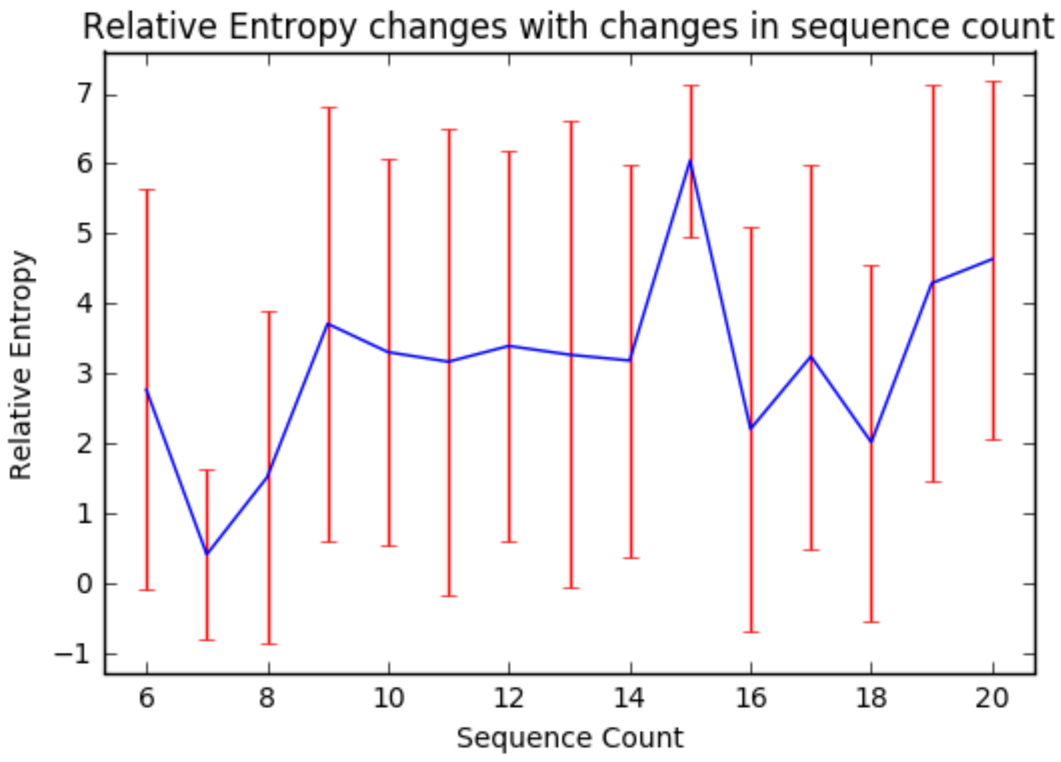
Conclusion

Taking all the three factor, relative entropy of PWM, sites overlap, runtime into consideration, the larger ICPC contributes to find motif more efficiently and accurately, while for motif length and sequence count, we need to find an appropriate value based on the performance on the difficulty, accuracy and runtime of motif finding. But, we only have three points in every plot which decreases the statistical power of our analysis dramatically. Increasing the informational content seems to have a better understanding of the regular pattern. However, a finer resolution is required in order to draw scientific conclusions.

Extra analysis

We thus generated more data sets with a finer resolution for the sequence count parameter to see if a clearer trend emerges. The results are as follows:

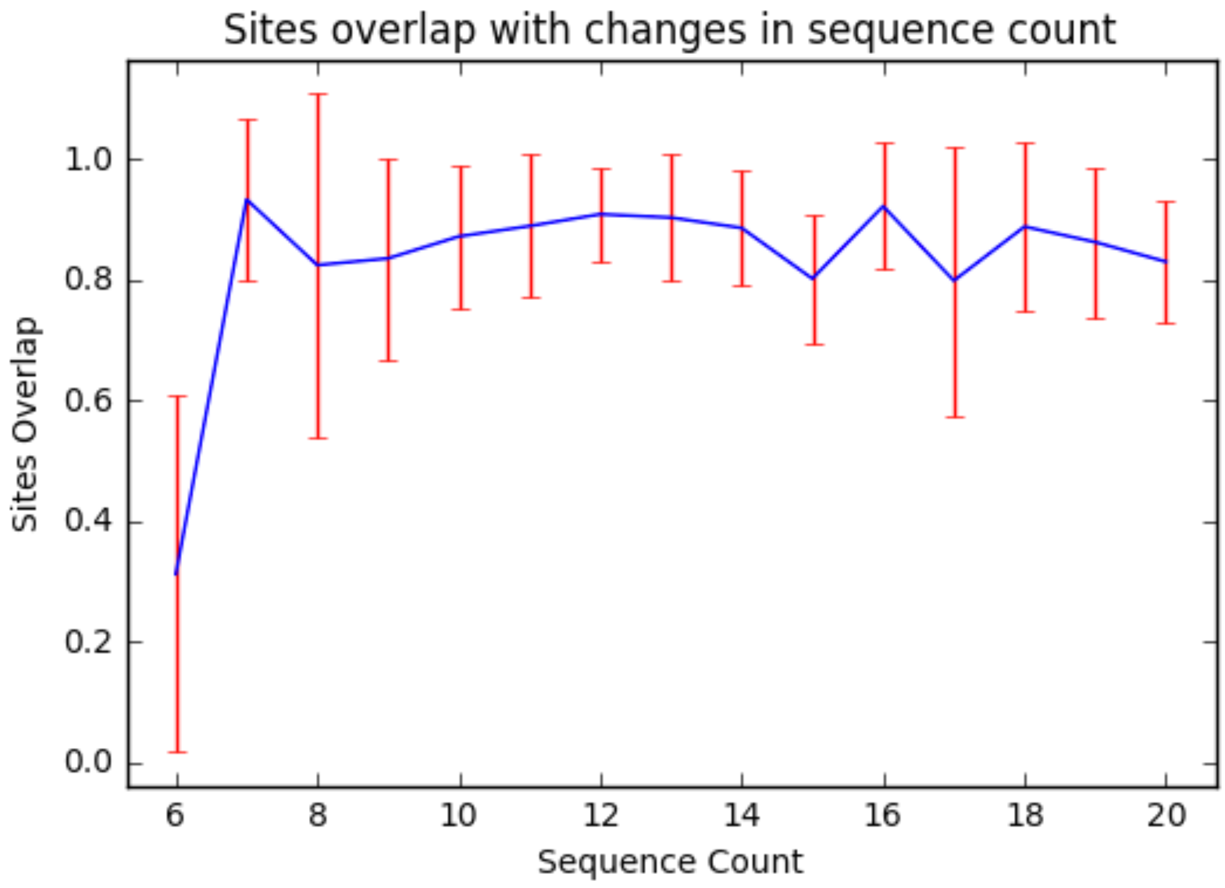
```
In [10]: analysis.rel_sc_better_filename()
```



As can be seen, even after having more data points, increasing the number of sequences does not really help us in getting a better result.

In [20]:

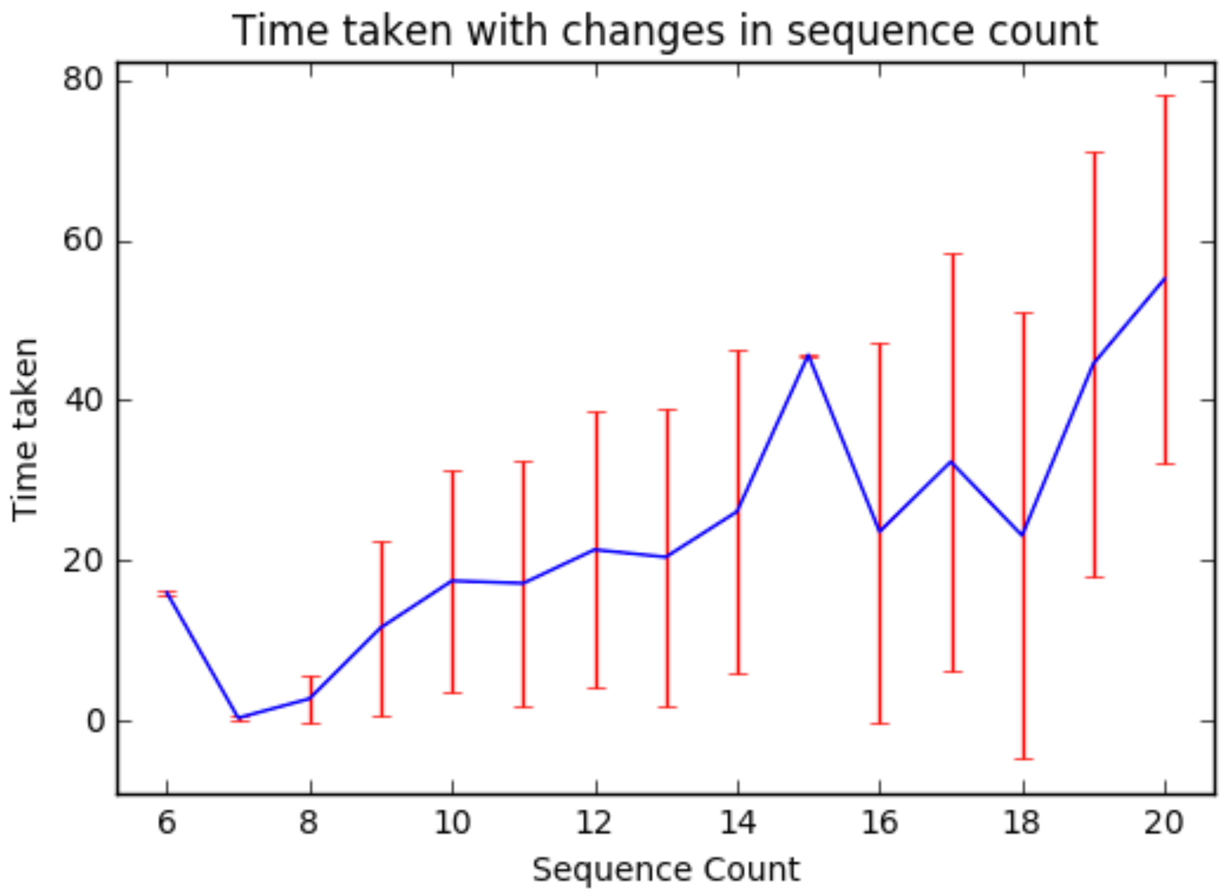
```
analysis.sites_sc_better_filename()
```



This chart seems to suggest that a minimum of 7 sequences is required before we will get a good result for sequences that are 500 nucleotides long. Any further increase does not seem to change the percentage of site overlap between the predicted motif and the actual motif. However, this seems to be incorrect as referring to the previous chart that is plotting this, with 5 sequences, we were able to get an overlap of about 0.75. This seems to suggest that the poor performance with 6 sequences is just bad luck due to randomness.

In [21]:

```
analysis.time_sc_better_filename()
```



From this chart, we can quite clearly see that increasing sequence counts causes the time taken to increase.