COMPUTATIONAL BIOLOGY

CS-549

PROJECT REPORT

SUBMITTED BY:

Sagardeep Mahapatra (108771077)

Biswaranjan Panda (108800494)

Neha Bhatnagar (108750977)

BRIEF RECAP:

We are using the following method to identify overlap between the long reads:

We first pick up some cluster centers from the reads and then try to cluster reads with those centers. For every read the following steps are performed:

a) Using the hash table setup in the cluster centers, we first locate a position in the read (using Rabin Karp algorithm) that might represent the start of a potential overlap with any of the cluster centers.
b) Once the position is located, we then verify some subsequent bases from that position.
c) If the subsequent bases also match with the subsequent bases of the cluster center, then there is high chance that the given read will successfully cluster.
d) If the subsequent bases do not match, go to (a) to locate the next position.

(Cluster Center)

ACTGC *GTAAC CTGAC CTGAA CTAGC* TTTCT CGCGC AAACG

May be start of an overlap (Read)

ACCCC GGGTT GC***ACT GC*** *GTA AC CTG AC CTG AA CTA GC*TTT

Verify some subsequent bases to confirm clustering

Representation of the Cluster Center:

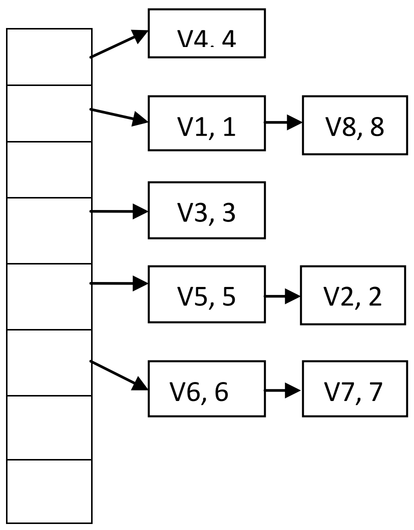
Consider the following read which is to be set up as a cluster center. We convert the read into **anchors** and for each anchor, we compute an **anchor vector** as shown below.

ACTGC	GTAAC	CTGAC	CTGAA	CTAGC	TTTCT	CGCGC	AAACG
V1	V2	V3	V4	V5	V6	V7	V8

The hash table stores all the **vector numbers** and **their positions** in the read.

struct hash *htable [NO_ANCHORS];

```
int read_number;

char sequence [READ_LENGTH];

int anchor_vector[NO_ANCHORS];

int points[NO_READS];

int counter;

struct cluster *next;
```



TESTS PERFORMED:

We incrementally test our algorithm on the following data sets:

1) Simulated reads of a small portion of E-Coli genome with no sequencing errors
   Then compare our results with the overlaps detected by minimus.

2) Simulated reads of a small portion of E-Coli genome with errors

3) Actual E-Coli long reads

TERMINOLOGIES:

In each of the following tests, we out put the following:

a) Overlaps:
   All overlaps output from the program are in the following format:

   (X1, pos1, X2, pos2)

   Implies read X1 starting from base POS1 overlaps with read X2 at base POS2

b) False Collisions:

   A false collision is an event where the start of a potential overlap is detected using the hash table setup, but the subsequent bases did not match. We output the total number of false collisions in the clustering process.

c) Total Base Comparisons:

   Represents the total number of base comparisons performed in the clustering process

LEARNING FROM THE DATA SET:

There arise a number of questions while performing clustering on the reads:

How many cluster centers should be chosen?

Since, the goal is to have a good clustering; as few centers as possible is preferable.

If N is the total number of reads, we initially choose (log N) reads and set them up as cluster centers. Then we cluster reads to these (log N) centers. If there are any leftover reads, which did not cluster with any of these cluster centers, then we again choose some more cluster centers (log X, where X = the number of leftover reads). We follow this process till we see that there is no effective clustering happening.

However, if N is large enough, log (log N) reads can be chosen as centers and results can be compared with the scenario where log N reads are chosen as centers.

Which reads should be picked as cluster centers?

We first verify all the reads and obtain information about them (as mentioned below):

Total number of reads: …
Total length of all reads: …
Length of longest read: …
Length of shortest read: …
Number of reads having length <1000: …
Number of reads having length between 1000 and 2000: …
Number of reads having length between 2000 and 3000: …
Number of reads having length between 3000 and 4000: …
…
Number of reads having length > 15000: …

The aim is to pick longer reads as cluster centers, because we assume that, longer the length of the read, greater is the probability that more reads will cluster with it. However, while clustering, we run the algorithm twice (once with choosing longer reads as centers and once with choosing random centers) and compare the results in both the cases.

What should be the anchor length?

As mentioned earlier, we convert the cluster center into anchors and compute anchor vectors for each anchor. Choosing an anchor length is determined by two things:

Impact on clustering: Anchor length has an impact on clustering because we search for an exact match (which might be a potential overlap position) based on the anchors and having a large anchor length will decrease the probability of getting an exact match. Smaller the anchor length, better are the chances of finding an exact match.

Impact on false collisions: However, having small anchor length will result in higher number of false collisions and eventually increase the total number of base comparisons.

Thus, there exists a tradeoff, and hence, we run the algorithm varying the anchor length and compare the results in each case.

How many subsequent bases much be checked in order to confirm clustering?

Currently we start by checking the subsequent 15 bases. But, in case the results are not good, we can increase the same and see the effect on the results.

Clustering on simulated reads from a small portion of E-Coli genome with no sequencing errors

We run our clustering algorithm on simulated reads from a small portion of E-Coli genome and then compare our results with the overlaps obtained from minimus.

Following information was extracted from the data set:

Total number of reads: 168
Total length of all reads: 294026
Length of the longest read: 8007
Length of the shortest read: 40
Number of reads < 1000: 73
Number of reads between 1000 and 2000: 40
Number of reads between 2000 and 3000: 25
Number of reads between 3000 and 4000: 13
Number of reads between 4000 and 5000: 6
Number of reads between 5000 and 6000: 7
Number of reads between 6000 and 7000: 1
Number of reads between 7000 and 8000: 2
Number of reads between 8000 and 9000: 1
Number of reads between 9000 and 10000: 0
…
Number of reads > 15000: 0

Parameters for the program:

Number of cluster centers chosen:

Since, the total number of reads is just 168, we start off with log 168 = 5 clusters.

Reads to select as cluster centers:

We pick longer reads to be set up as cluster centers with the assumption that more reads will cluster to them.

Anchor length: 5 bases

Number of subsequent matches verified to confirm clustering: 15 bases

Output:

Total number of clusters obtained: 6
Total number of false collisions: 472591
Total number of base comparisons: 8047919

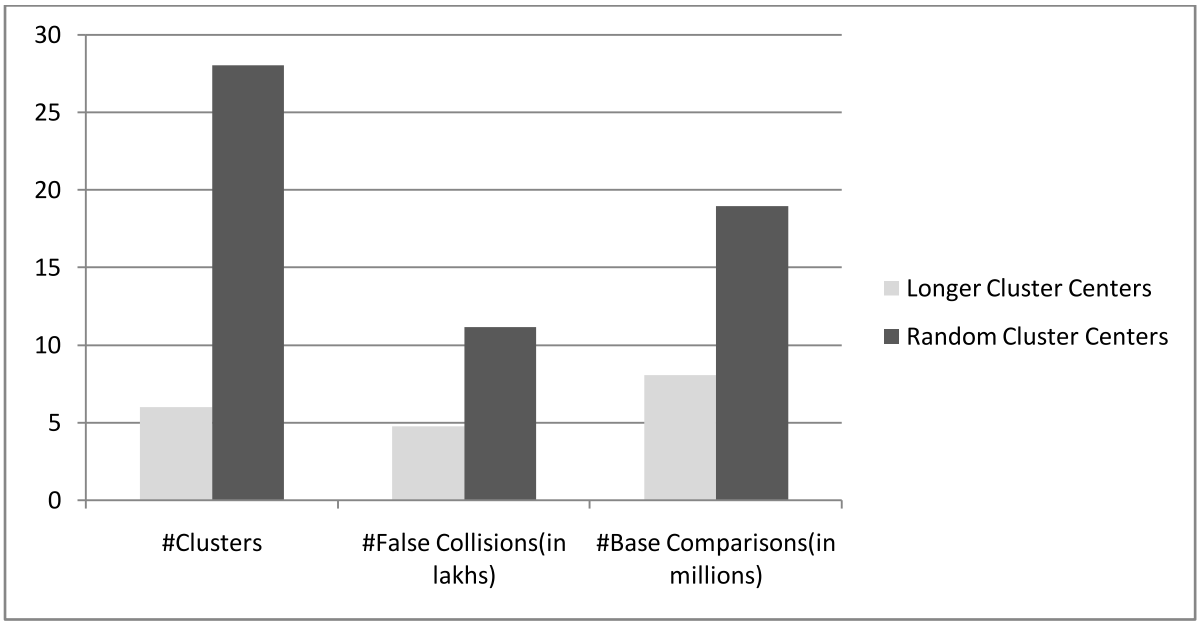
Comparing Results

We compare our results with the results obtained from minimus: (Overlaps in bold correspond to the ones that matched with the overlaps obtained from minimus)

Our results	Results obtained from minimus
(85,86) (85,87) (85,88) (85,89) (85,90) (85,91) (85,92) (85,93) (85,94) (85,25) (85,96) (85,97)	(44,85) (70,85) (73,85) (74,85) (78,85) (79,85) (81,85) (82,85) (84,85) **(85,86)**
(148,143) (148,144) (148,145) (148,146) (148,147) (148,149) (148,150) (148,151) (148,152) (148,153) (148,154) (148,155) (148,156) (148,157) (148,158) (148,159) (148,160) (148,161) (148,162) (148,163) (148,164) (148,165) (148,166)	(132,148) (133,148) (136,148) (137,148) (138,148) (139,148) (140,148) (142,148) **(143,148) (144,148) (145,148) (146,148) (147,148) (148,149) (148,150) (148,151) (148,152) (148,153) (148,154) (148,155) (148,156) (148,157) (148,158)**
(106,98) **(106,99) (106,100) (106,101) (106,102) (106,104) (106,105) (106,107) (106,108) (106,109) (106,110) (106,111)** (106,112) (106,113) (106,114) (106,115) (106,116) (106,117) (106,118) (106,119) (106,120) (106,121) (106,122) (106,123) (106,124) (106,125) (106,126) (106,127) (106,128) (106,129) (106,130) (106,131) (106,132) (106,133) (106,134) (106,135) (106,136) (106,137) (106,138) (106,139) (106,140) (106,141) (106,142)	(78,106) (90,106) (97,106) **(99,106) (100,106) (101,106) (102,106)** (103,106) **(104,106) (105,106) (106,107) (106,108) (106,109) (106,110) (106,111)**
(44,45) (44,46) (44,47) (44,48) (44,49) (44,50) (44,51) (44,52) (44,53) (44,54) (44,55) (44,56) (44,57) (44,58) (44,59) (44,60) (44,61) (44,62) (44,63) (44,64) (44,65) (44,66) (44,67) (44,68) (44,69) (44,70) (44,71) (44,72) (44,73) (44,74) (44,75) (44,76) (44,77) (44,78) (44,79) (44,80) (44,81) (44,82) (44,83) (44,84)	**(44,45) (44,46) (44,47) (44,48) (44,49) (44,50) (44,51) (44,52) (44,53) (44,54) (44,55) (44,56) (44,57) (44,58) (44,59) (44,60) (44,61) (44,62) (44,63) (44,64) (44,65) (44,66) (44,67) (44,68) (44,69) (44,70) (44,71) (44,72) (44,73) (44,74) (44,75) (44,76) (44,77) (44,78) (44,79) (44,80) (44,81) (44,82) (44,83) (44,84)** (44,85)
(7,8) (7,12) (7,13) (7,14) (7,17) (7,19) (7,21) (7,22) (7,23) (7,24) (7,25) (7,26) (7,27) (7,28) (7,29) (7,33) (7,34) (7,35) (7,36) (7,40) (7,41) (7,42) (7,43)	**(7,8)** (7,9) (7,10) (7,11) **(7,12) (7,13) (7,14)** (7,15) (7,16) **(7,17)** (7,18) **(7,19)** (7,20) **(7,21) (7,22) (7,23) (7,24) (7,25) (7,26) (7,27) (7,28) (7,29)** (7,30) (7,31) (7,32) **(7,33) (7,34) (7,35) (7,36)** (7,37) (7,38) (7,39) **(7,40) (7,41) (7,42) (7,43)** (2,7) (4,7) (5,7) (6,7)
(2,1) (2,3) (2,4) (2,5) (2,6) (2,9) (2,10) (2,11) (2,15) (2,16) (2,18) (2,20) (2,30) (2,31) (2,32) (2,37) (2,38) (2,39)	**(1,2) (2,3) (2,4) (2,5) (2,6)** (2,7) (2,8) **(2,9) (2,10) (2,11)** (2,12) (2,13) (2,14) **(2,15) (2,16)** (2,17) **(2,18)** (2,19) **(2,20)** (2,21) (2,22) (2,23) (2,24) (2,25) (2,26) (2,27) (2,28) (2,29) **(2,30) (2,31) (2,32)** (2,33) (2,34) (2,35) (2,36) (2,37) **(2,38) (2,39)** (2,40)

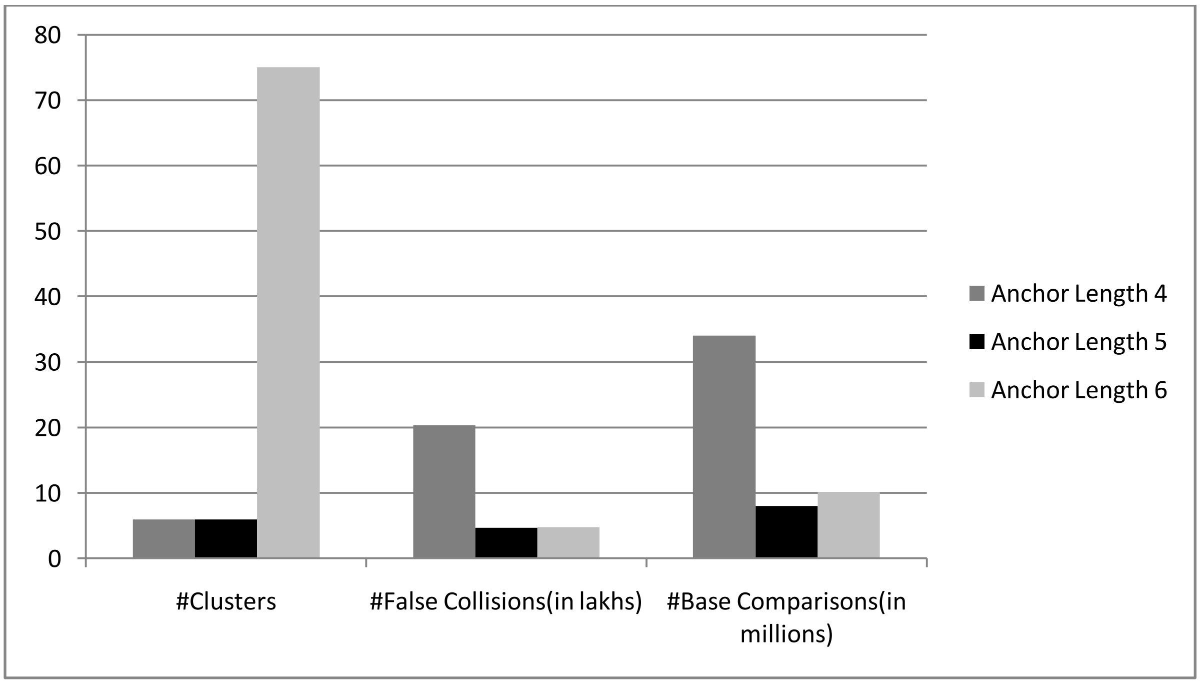
Choosing cluster centers randomly

We then select the cluster centers randomly (instead of choosing longer reads to be cluster centers) and compare our results with the previous case. All other parameters remain the same.



Varying the anchor length

We then vary the anchor length and compare the results in terms of the number of false collisions, base comparisons and number of clusters obtained in each case.



Observations

1) Clustering was better when longer cluster centers were chosen. Hence, our assumption was correct. Having longer cluster centers is more effective than choosing centers randomly, as longer the center is; greater are the chances that more reads will cluster.

2) Lowering the anchor length resulted in higher false collisions and base comparisons. On, increasing the anchor length, clustering was ineffective.

3) One more thing observed is that reads nearby each other generally clustered together.

CONCLUSION

The approach used to detect overlaps between the reads first finds an actual match of a few bases on the reads and then compares the subsequent bases to confirm clustering.

- The proposed method to detect overlaps between reads is applicable when there are no errors or only substitution errors.
- But when the sequencing errors are mostly insertion or deletion errors, the proposed algorithm does not give any results.