**All positions in my answers are 0 indexed (unlike the table in the article)**

1. **Build a hash table for the provided list of sequences “seqs” (same as in the paper). Print the table.**

AA 1,18

AC 0,8 1,4 1,10

GT 0,0 0,26 0,28 1,0 1,36 2,18

AG 0,14 1,34

CC 0,20 1,30 2,4 2,6

CA 1,2 1,8 1,20 1,26 1,32 2,20 2,22

CG 0,4

GG 0,24 0,30 1,16 1,28 2,0

AT 1,12 2,2

GA 0,2 0,16 1,14 1,24

TG 0,12 1,6 2,8

TA 2,24

TC 0,6 0,10 0,18 1,22 1,40 2,10

CT 0,22 1,38 1,42 2,12 2,14 2,16

1. **Calculate the sorted list of hits for the provided query sequence. Each hit should be a tuple of (index, shift, offset). Print the number of hits and the first and last hit in your list.**

Number of hits: 23

First hit: (0, 4, 8)

Last hit: (2, 20, 22)

1. **Calculate the maximum match for the provided query sequence, and print the alignment (the two sequences with the right offset, such that the characters match).**

**GTCAACTGCAACATGAGGAACATCGACAGGCCCAAGGTCTTCCT**

**||||||||**

**TGCAACAT**

1. **Using the TAIR10 genome, how many sequences does the genome file contain and what is the total sequence length? Does this match your expectation on the Arabidopsis genome?**

The genome contains: 5 sequences

The total sequence length is: 119146348

The actual Arabidopsis genome is larger; these are only the first 5 chromosomes.

1. **Build a hash table for the Arabidopis chromosomes. Use k = 15. How many keys (different 15-mers) does your hash contain?**

The hashtable contains 7531450 k-mers of size 15 ( only K-mers of size < 15 are removed)

**6. For the query sequences in the FASTA file athal\_query.fasta, find the maximum hit(s) in the genome. Report your findings (if any).**

The first query was the only one that had a match with more than 1 k-mer in the database:

Longest match first query:

[(2, 160637, 160650), (2, 160637, 160665), (2, 160637, 160680), (2, 160637, 160695), (2, 160637, 160710), (2, 160637, 160725), (2, 160637, 160740), (2, 160637, 160755), (2, 160637, 160770), (2, 160637, 160785), (2, 160637, 160800), (2, 160637, 160815), (2, 160637, 160830), (2, 160637, 160845), (2, 160637, 160860), (2, 160637, 160875), (2, 160637, 160890), (2, 160637, 160905), (2, 160637, 160920), (2, 160637, 160935), (2, 160637, 160950), (2, 160637, 160965), (2, 160637, 160980), (2, 160637, 160995), (2, 160637, 161010), (2, 160637, 161025), (2, 160637, 161040), (2, 160637, 161055)]

**7. What is the algorithm trying to optimize?**

The algorithm tries to optimize for the longest possible (local) match of (a part of) the query in the database without mismatches. Possible mismatches pre-/proceeding the match are not taken into consideration.

**8. What did you learn about the time complexity of the search part of the algorithm.**

I think because a lookup in the hashmap has O(1) the complexity will probably come close to n \* m where n is the number of k-mers in the query and m is the number of times that k-mer occurs in the database.

**9. What is the main difference in time complexity of the SSAHA algorithm and BLAST**

SSAHA is three to four orders of magnitude faster than BLAST, as stated in the article. I expect this is due to the use of a hashtable.

**10. How many k-mers with an overlap of p bases does a reference sequence of length m contain?**

I have seen that the number of k-mers = (n - k+1) but I am not sure how to incorporate the overlap in this equation

**Optional:**

**SSAHA2 Query 0 forward: aligns with my Query 0 forward**

[(2, 160637, 160650), (2, 160637, 160665), (2, 160637, 160680), (2, 160637, 160695), (2, 160637, 160710), (2, 160637, 160725), (2, 160637, 160740), (2, 160637, 160755), (2, 160637, 160770), (2, 160637, 160785), (2, 160637, 160800), (2, 160637, 160815), (2, 160637, 160830), (2, 160637, 160845), (2, 160637, 160860), (2, 160637, 160875), (2, 160637, 160890), (2, 160637, 160905), (2, 160637, 160920), (2, 160637, 160935), (2, 160637, 160950), (2, 160637, 160965), (2, 160637, 160980), (2, 160637, 160995), (2, 160637, 161010), (2, 160637, 161025), (2, 160637, 161040), (2, 160637, 161055)]

**SSAHA2 Query 1 complement: aligns with my Query 1 complement**

[(3, 43141, 43155), (3, 43141, 43170), (3, 43141, 43185), (3, 43141, 43200), (3, 43141, 43215), (3, 43141, 43230), (3, 43141, 43245), (3, 43141, 43260), (3, 43141, 43275), (3, 43141, 43290), (3, 43141, 43305)]

**SSAHA2 Query 2 complement: aligns with my Query 2 complement**

[(3, 2726909, 2726910), (3, 2726909, 2726925), (3, 2726909, 2726940), (3, 2726909, 2726955), (3, 2726909, 2726970), (3, 2726909, 2726985), (3, 2726909, 2727000), (3, 2726909, 2727015), (3, 2726909, 2727030), (3, 2726909, 2727045), (3, 2726909, 2727060), (3, 2726909, 2727075), (3, 2726909, 2727090), (3, 2726909, 2727105), (3, 2726909, 2727120), (3, 2726909, 2727135), (3, 2726909, 2727150), (3, 2726909, 2727165), (3, 2726909, 2727180), (3, 2726909, 2727195), (3, 2726909, 2727210), (3, 2726909, 2727225), (3, 2726909, 2727240), (3, 2726909, 2727255), (3, 2726909, 2727270), (3, 2726909, 2727285), (3, 2726909, 2727300), (3, 2726909, 2727315), (3, 2726909, 2727330), (3, 2726909, 2727345), (3, 2726909, 2727360), (3, 2726909, 2727375), (3, 2726909, 2727390), (3, 2726909, 2727405), (3, 2726909, 2727420), (3, 2726909, 2727435), (3, 2726909, 2727450), (3, 2726909, 2727465), (3, 2726909, 2727480), (3, 2726909, 2727495), (3, 2726909, 2727510)]