CMSC423: Chapter 9
Suffix arrays

Brief recap

- Suffix trees – a way to organize all suffixes of a string
- Can be constructed in O(n)
- Enable searches in O(n)

Suffix arrays

- Suffix trees are expensive > 20 bytes / base
- Suffix arrays: lexicographically sort all suffixes

ATG 4
ATCATG 1
CATG 3
G 6
TCATG 2
TG 5

- Stop and think: Why sort the suffixes?

- Stop and think: How much memory is needed to store a suffix array?

Suffix arrays

- Size of suffix array = O(n) integers (only store suffix number)

ATG	4
ATCATG	1
CATG	3
G	6
TCATG	2
TG	5

- Can quickly find the correct suffix through binary search

Binary search

- Is TCA in the string?

ATG 4		ATG 4		ATG 4	
ATCATG 1	TCA?	ATCATG 1		ATCATG 1	
CATG 3		CATG 3	TCA?	CATG 3	
G 6		G 6		G 6	
TCATG 2		TCATG 2		TCATG 2	TCA!!!
TG 5		TG 5		TG 5	

- With N = len(text), M = len(pattern)
  What is the runtime of the binary search?

Binary search

- Is TCA in the string?

ATG 4		ATG 4		ATG 4
ATCATG 1	TCA?	ATCATG 1		ATCATG 1
CATG 3		CATG 3	TCA?	CATG 3
G 6		G 6		G 6
TCATG 2		TCATG 2		TCATG 2 TCA!!!
TG 5		TG 5		TG 5

- With N = len(text), M = len(pattern)
  What is the runtime of the binary search?

- # of comparisons X time needed to compare
- O(log N) X O(M) = O(M log N)

Making suffix arrays practical

- O(log N + M) is possible with some preprocessing
- LCP array – longest common prefixes of (certain) pairs of suffixes

- Stop and think: Given a suffix tree, can you construct a suffix array? (and what is the algorithm and runtime)
- Stop and think: Given a suffix tree, can you construct the LCP array for all pairs of suffixes adjacent in the suffix array?

ATG 4
ATCATG 1
CATG 3
G 6
TCATG 2
TG 5

LCP(4,1); LCP(1,3); LCP(3,6); LCP(6,2); LCP(2,5)

Next: the Burrows Wheeler Transform