Multi-string matching

Given string $T$

$m := |T|$

Given pattern string $P_1, \ldots, P_k$

$n := |P_1| + \ldots + |P_k|$

Do any pattern strings appear in $T$?

$|\mathcal{L}| = O(1)$

KMP once per pattern; $O(km + n)$
Approach 1:
Build a data structure for $P_1, P_2, \ldots, P_k$. Scan $T$.

Suppose a pattern was a prefix of $T$.

Build a trie over patterns.
Do a "search" for $T$.
Stop when you hit a pattern's node.
Add suffix links:
A link from w’s node to the node for longest proper suffix of w.

Work your way down the trie. If you can’t progress, take suffix link and try.
again. Skip current character if cannot progress from root.

Can build trie + suffix links in $O(n)$ time.

$O(m)$ time searching $T$.

$O(n + m)$ total.

(Add some stuff to list all instances of patterns in $O(n + m + z)$ time.

Aho-Corasick)}
For computational genomics, we'd rather have a structure over a fixed $T$.

**Observation:** If $x$ is a substring of $w$, then $x$ is a prefix of a suffix of $w$. 
Suffix trie of $T$ is a trie over all suffixes of $T$.

If any node represents $x$, $x$ is a prefix of some suffix, therefore a substring of $T$.

Common to take a sentinel character $\$ \# and build suffix trie for $T$.\#
Suffix trie:
Bad case: $T = a^m b^m D$.  

$m$ copies of a chain of $m$ 6's.  

$\Theta(m^2) = \Theta(1+T_2)$ nodes.
**Suffix tree**

A Patricia suffix trie, over $T$. Has $m+1$ leaves. All internal nodes have $\geq 2$ children $\Rightarrow \leq m$ internal nodes. So $\Theta(m)$ nodes.

Instead of writing whole substrings at links, just write beginning & ending positions of $T$, $\Theta(m)$ space.
Can build in \( \Theta(nm) \) time (complicated)

Can search all patterns in \( \Theta(n) \) so \( \Theta(n + m) \) total
Alternative: Suffix array

Store suffixes in an array sorted in alphabetical order.

To search for \( P \):
- Binary search for subarray starting with first character of \( P \).
- Recursively search rest of \( P \) in that subarray.
$\Theta(m)$ space.

Search for $P$ in $O(\|P\| \log m)$ time.

Might be faster in practice.
Final Exam: Mon. Dec. 12th
Cumulative:
No labeling components.
But do know what BFS, DFS do.
No Borůvka.
6-7 questions
eval. utdallas.edu

↑

please