

# RNA-Seq

## Li & Homer '2010: A survey of sequence alignment algorithms for next-generation sequencing

- Fast alignment algorithms use auxiliary data indices
  - For short reads,
  - For reference genomes
  - For both
- Indexing algorithms
  - Hash tables
  - Suffix trees
  - BWT

# Hash Table indexing

- Seed-extend paradigm
  - E.g., BLAST
  - Use hash table to find locations of k-mers ( $k=11$  for DNA) and extend using variation of Smith-Waterman
- Improvements to BLAST to handle short reads against long genomes
  - Seed using non-consecutive matches, aka "spaced seed".
  - Specific locations vs. any k-mismatch hit (with a bound on  $k \leq 2$  to limit possibilities)
  - Minimum number of seeds for given read length, sensitivity requirement and memory usage.

# Hash Table Indexing

- Memory requirement is problematic
- One Solution: Two-level indexing.
  - Hash table for  $j$ -long keys ( $j < q$ )
  - To find  $q$ -long keys, first search in  $j$ -long prefix hash table, then binary search in the bucket.

# Allowing gaps

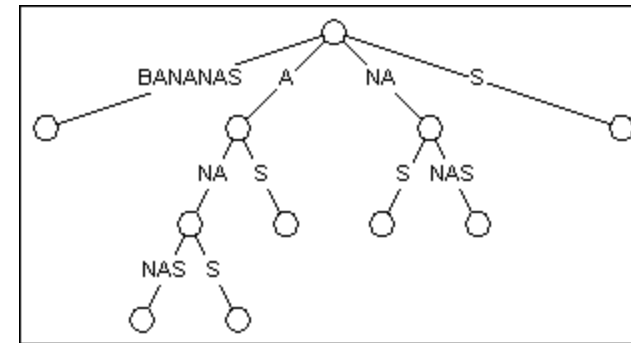
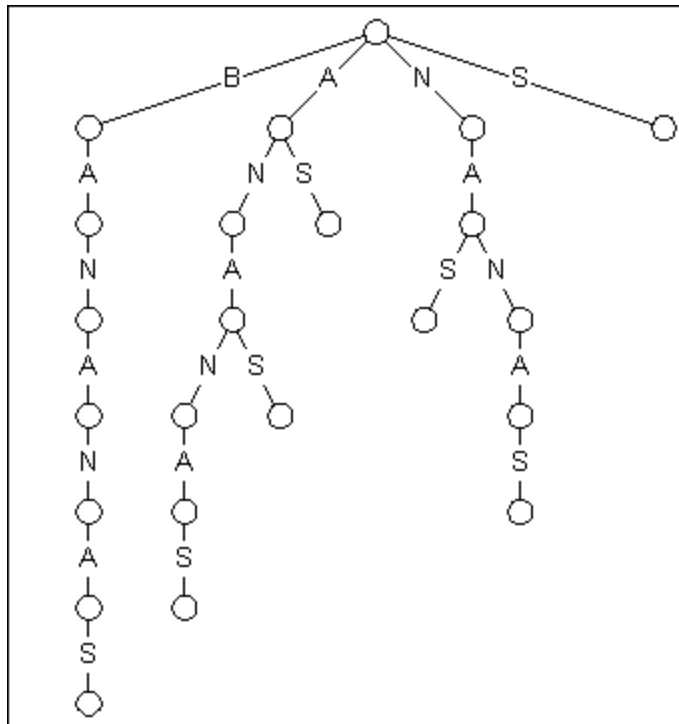
- Hash table doesn't allow for gaps in seeds
- Solution: q-gram filter (BLAT)
  - With at most  $k$  differences, the  $w$ -long query and the  $w$ -long database substring share at least  $(w+1) - (k+1)q$  common substrings of length  $q$ .
- Spaced seeds: extend on long seed match. vs. q-gram filter: initiates extension with multiple shorter seed matches.

# Improvements in Seed-Extension

- Seed extension is typically unnecessary with long spaced seeds
- Most only extend without gaps.
- Improvements to seed extension
  - Constrain dynamic programming around seeds
  - Vectorized code

# Suffix tries

- A suffix trie is a data structure that stores all suffixes of a text.



# Burrows Wheeler Transform

Transformation				
Input	All rotations	Sorted into lexical order	Taking last column	Output last column
<div> ^BANANA   </div>	<div> ^BANANA      ^BANANA  A   ^BANAN  NA   ^BANA  ANA   ^BAN  NANA   ^BA  ANANA   ^B  BANANA   ^ </div>	<div> ANANA   ^B  ANA   ^BAN  A   ^BANAN  BANANA   ^  NANA   ^BA  NA   ^BANA  ^BANANA      ^BANANA </div>	<div> ANANA   ^B  ANA   ^BAN  A   ^BANAN  BANANA   ^  NANA   ^BA  NA   ^BANA  ^BANANA      ^BANANA </div>	<div> BNN^AA   A </div>