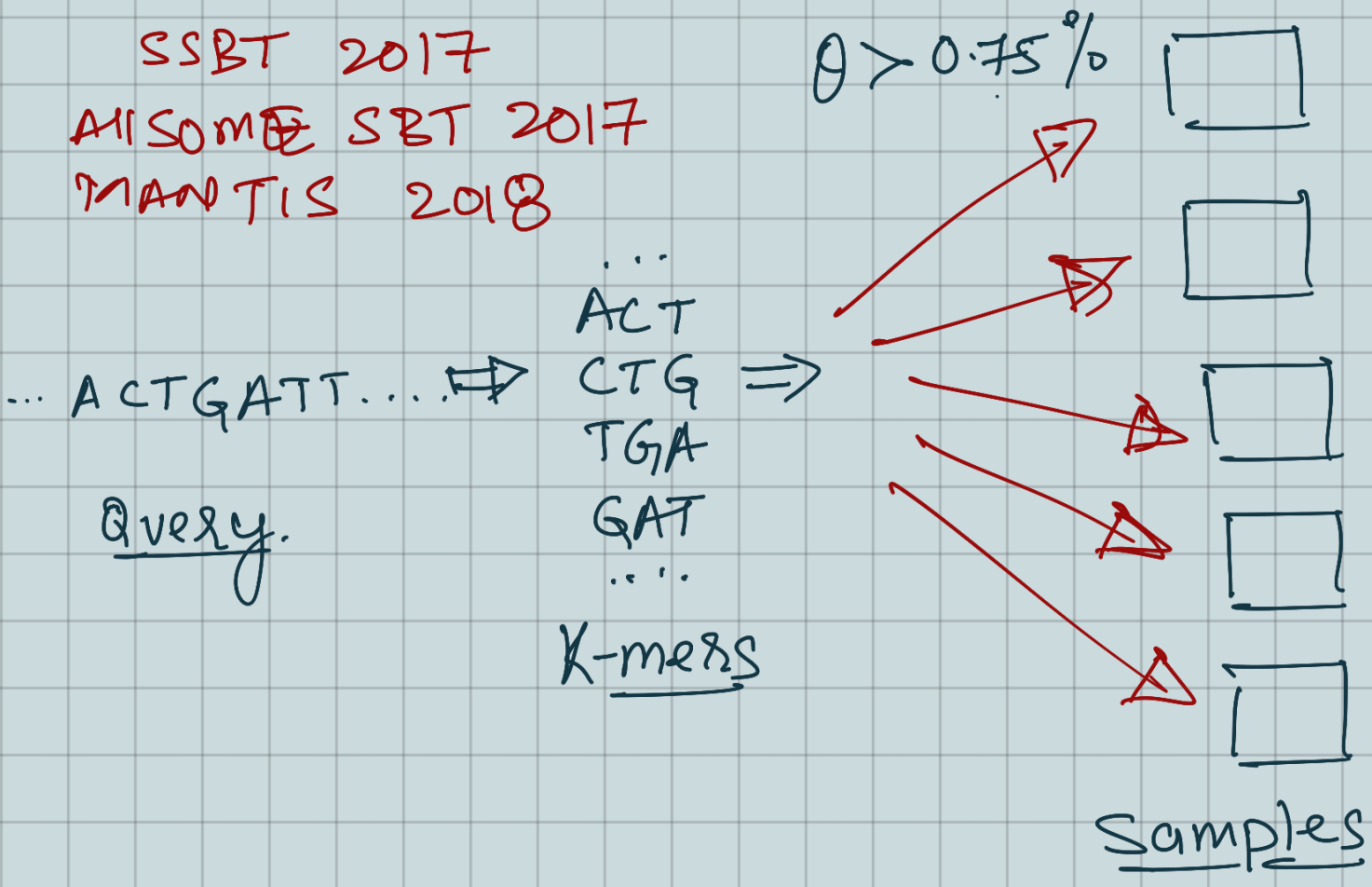
Sample discovery problem:

SBT 2016
SSBT 2017
AllSomeE SBT 2017
MANTIS 2018



→ Decompose each sample into K-mers
→ If more than θ-fraction K-mers from a query appear in a sample then there is a high chance that query appears in that sample.

→ SBT, SSBT, AllSomeE SBT
→ Based on Bloom filter.

→ Bloom Filter is used to represent K-mer contents.

Mantis [Inverted index].

S1	S2	S3	S4
	ACTG	ACTG	
ACTT			
		CTTG	CTTG
	TTTC	TTTC	
	GCGT	GCGT	GCGT
	AGCC	AGCC	

→ Another layer of indirection

↓

Map K-mers to Samples.

K-mer	Samples
ACTG	S_2 S_3
ACTT	S_1
CTTG	S_3 S_4
TTTC	S_2 S_3
GCGT	S_2 S_3 S_4
AGCC	S_2 S_3.

K-mer	ID
ACTG	0
ACTT	10
CTTG	1
TTTC	0
GCGT	11
AGCC	0

→

IDs to Samples.

0 → S_2 S_3

1 → S_3 S_4

10 → S_1

11 → S_2 S_3 S_4.

↓

	S_1	S_2	S_3	S_4
0 →	0	1	1	0
1 →	0	0	1	1
10 →	1	0	0	0
11 →	0	1	1	1

Scalability Challenge:

→ Mantis index provides fast query, and scales well up to thousands of input experiments/samples.

→ But we really want to index on the order of 10^5 - 10^6 samples.

Key observation:

→ K-mer grow at worst linearly.
→ Color classes grow super-linearly.

* Need a fundamentally better color class encoding.

Key idea:-

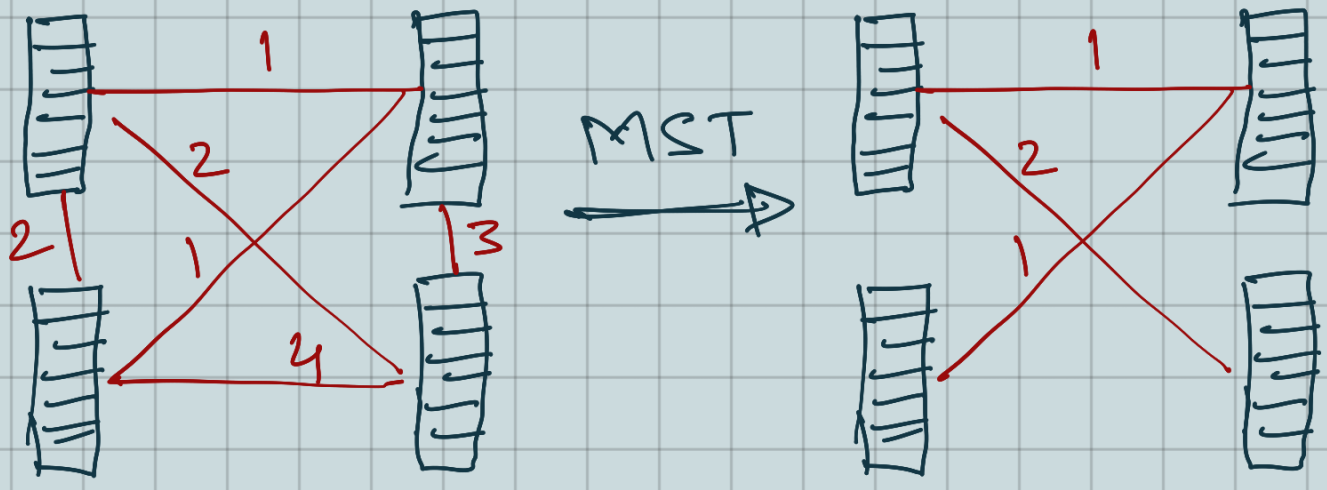
→ Exploit coherence between rows of the color class matrix

→ If we know a k-mer's color vector, we probably know a lot about its neighbor's color vectors.

Build a color class Graph:

→ Each color class is a vertex.

→ Every pair of color classes is connected by an edge whose weight is the hamming distance between the color class vectors.



Unfortunately :-

→ There are many color classes (full graph too big)

→ They are high dimensional (millions of samples)

→ Neighbor search is very hard (LSH scheme seem to work poorly).

Implicitly Represent the colored dBG.

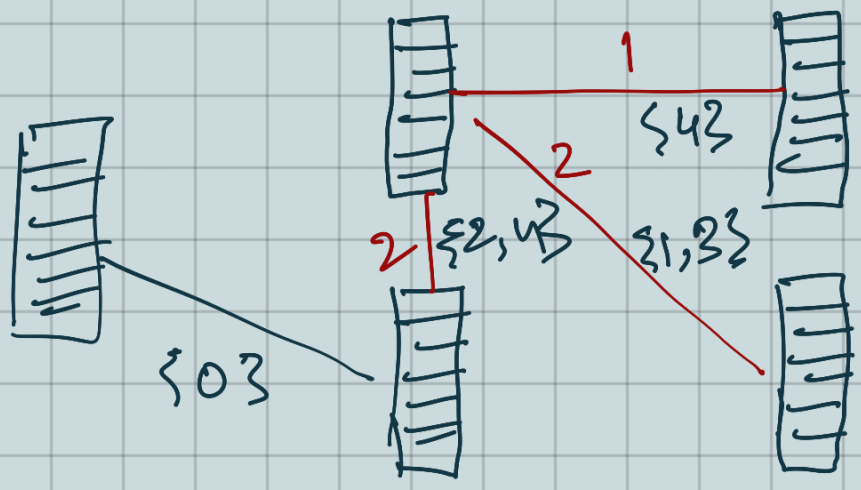
→ K-mer form the dBG.
→ Can explicitly query neighbors.

→ use deBruijn Graph as an Efficient guide for near-neighbor search in the space of color classes

→ dBG helps to impose biological relationships over color-class vectors.

→ MST derived from dBG-based color class graph is very close to the MST from complete graph.

MST efficiently encodes related color classes



→ Along the edge from each c node to its parent p, store $\delta(p,c)$

→ the positions of the bits whose parity we would have to flip to obtain the child from the parent.

→ To reconstruct a node's bit vector, walk from the node to the root, flipping the parity of the positions you encounter on each edge