Accelerating all-all Protein Sequence Matching

Pratyush Maini, Nikhil Yadala
Under the Guidance of
Prof. Jim Larus, Stuart Byma, Sahand Kashani

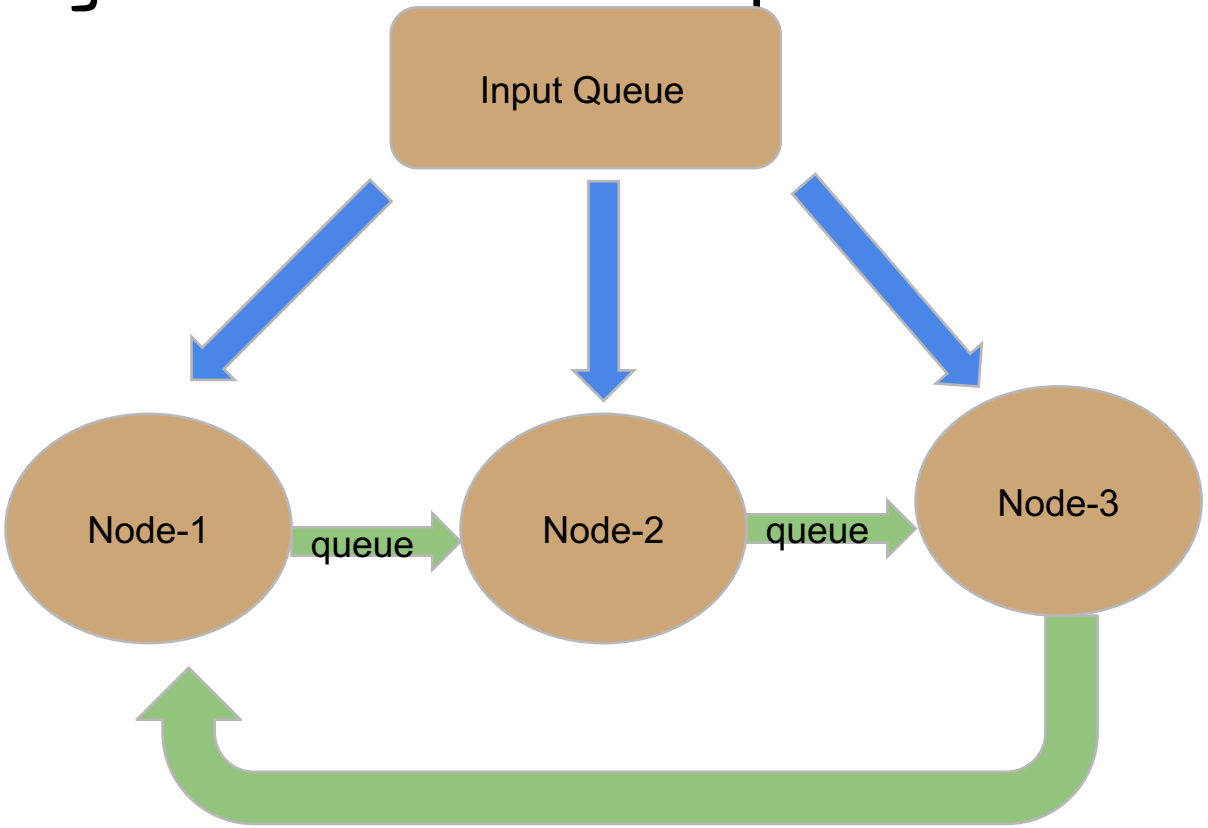
Motivation

1. Dataset size increasing tremendously.
2. Cost of Sequencing Reducing drastically.
3. All pair protein matching has many important applications.
4. About half of the inferred protein matches are not used in final orthology inference

Existing methods

- Smith Waterman - Dynamic Programing based
- By constructing Phylogenetic trees
- Generative models- Hidden markov model based
- Variations and Mixtures of these

Clustering in Persona Tensorflow



Striped & Banded Smith-Waterman

So what is Smith Waterman Algorithm?

1. Local Sequence Alignment
2. Tailored for nucleic acids/ protein sequences
3. $O(n^2)$ Algorithm
4. Aligns two sequences by matches/mismatches (also known as substitutions), insertions, and deletions

Why Local Alignment?

Two proteins may share, a small stretch of high protein similarity, but may be very different outside that region.

Global Alignments would result in high number of insertions/deletions -- mismatches outside the region

Hence a **lower score**

Which is undesirable

Working of Smith-Waterman

1. **Determine the substitution matrix and the gap penalty scheme.**
   Match
   Mismatch
   Gap open
   Gap Extension

Below diagonal: BLOSUM 62
Above diagonal: BLOSUM 62 - PAM 160

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
	0	-1	1	0	2	1	1	2	1	2	0	0	2	4	1	5	1	2	-2	5	C
		2	0	-2	0	-1	0	0	0	1	0	0	0	1	0	-1	-1	1	1	-1	S
C	9		2	-1	-1	-1	0	0	0	0	0	0	-1	0	-1	1	0	1	1	3	T
S	-1	4		2	-2	-1	-1	0	0	-1	-1	-1	1	1	0	-1	0	0	2	1	P
T	-1	1	5		2	-1	-2	-2	-1	0	0	1	1	0	0	1	0	1	1	2	A
P	-3	-1	-1	7		2	0	-1	-2	0	1	1	0	0	-1	0	-1	1	2	4	G
A	0	1	0	-1	4		3	-1	-1	0	0	1	-1	0	-1	0	-1	0	0	0	N
G	-3	0	-2	-2	0	6		2	-1	-1	-1	0	-1	0	0	0	0	2	1	3	D
N	-3	1	0	-2	-2	0	6		1	0	0	2	2	1	-1	0	0	2	2	4	E
D	-3	0	-1	-1	-2	-1	1	6		0	-2	0	1	1	-1	0	0	1	3	3	Q
E	-4	0	-1	-1	-1	-2	0	2	5		2	-1	0	1	0	-1	0	1	2	2	H
Q	-3	0	-1	-1	-1	-2	0	0	2	5		-1	-1	0	-1	1	0	1	3	-4	R
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8		1	-2	-1	1	1	2	3	1	K
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5		-2	-1	-1	0	1	2	4	M
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5		-1	1	0	0	1	3	I
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5		-1	0	-1	1	2	L
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4		0	1	2	4	V
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4		-1	-2	1	F
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4		-1	2	Y
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		-1	W
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

Working of Smith-Waterman

2. Initialize the scoring matrix.

Dimensions: (1+length_seq1)*(1+length_seq2)

Zero Initialisation: $A_{i0} = 0$ & $A_{0j} = 0$.

The extra first row and first column make it possible to align one sequence to another at any position, and setting them to 0 makes the terminal gap free from penalty.

Scoring Matrix

	0	A	C	G	T	A	T	G	C
0	0	0	0	0	0	0	0	0	0
A	0	2	0	0	0	2	0	0	0
C	0	0	4	2	1	0	1	0	2
G	0	0	2	6	4	3	2	3	1
A	0	2	1	4	5	6	4	3	2
A	0	2	1	3	3	7	5	4	3
C	0	2	4	2	2	5	6	4	6
C	0	0	2	3	1	4	4	5	6
C	0	0	2	1	2	3	3	3	7
T	0	0	0	1	3	2	5	3	5
T	0	0	0	0	3	2	4	4	4
G	0	0	0	2	1	2	2	6	4
C	0	0	2	0	1	0	1	4	8

Working of Smith-Waterman

3. Scoring.

Left to right,

Top to bottom

$$H_{ij} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ \max_{k \geq 1} \{H_{i-k,j} - W_k\}, \\ \max_{l \geq 1} \{H_{i,j-l} - W_l\}, \\ 0 \end{cases} \qquad (1 \leq i \leq n, 1 \leq j \leq m)$$

where

$H_{i-1,j-1} + s(a_i, b_j)$ is the score of aligning a_i and b_j,

$H_{i-k,j} - W_k$ is the score if a_i is at the end of a gap of length k,

$H_{i,j-l} - W_l$ is the score if b_j is at the end of a gap of length l,

0 means there is no similarity up to a_i and b_j.

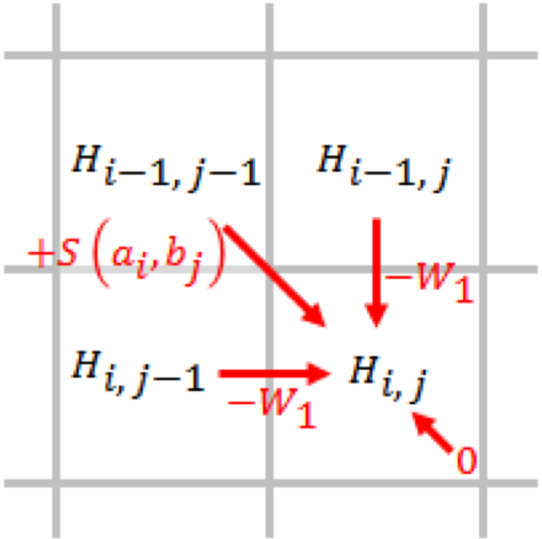
$O(m^2n)$ Algorithm

Working of Smith-Waterman

3. Scoring.

Left to right,

Top to bottom



O(mn) Algorithm

Gap penalty = Constant/ Affine Gap Penalty (A + BL)

Working of Smith-Waterman

4. Traceback.

Starting at the element with the highest score, traceback based on the source of each score recursively, until 0 is encountered.

The segments that have the highest similarity score based on the given scoring system is generated in this process.

Existing Implementation - SWPS3

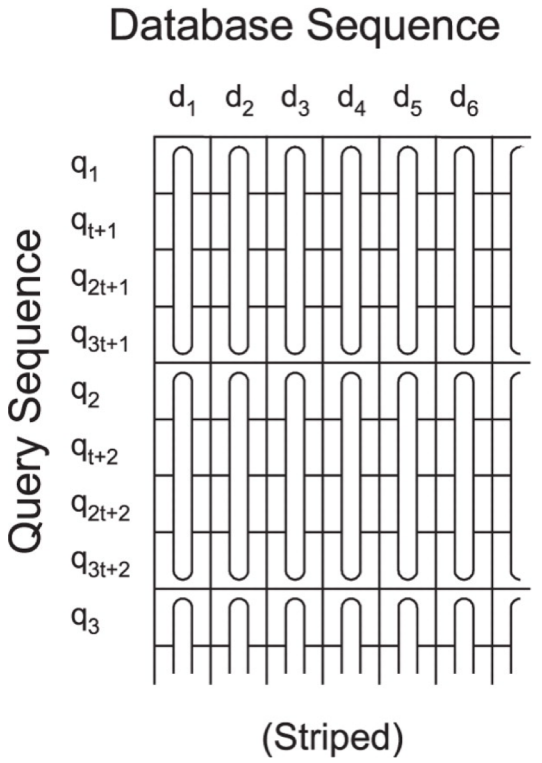
1. Vectorized implementation of the Smith-Waterman
2. Compatible with the current PERSONA framework
3. Slow computation in standard O(mn) method.
4. Lags significantly as larger sequences approach.

Optimisation with Striped Smith-Waterman (SSW)

Striped Query Profile:

When calculating $H_{i,j}$ the value from the scoring matrix $W(q_i, d_j)$ is added to $H_{i-1,j-1}$. To avoid the lookup of $W(q_i, d_j)$ for each cell, a Query profile is pre-calculated.

Then the calculation of $H_{i,j}$ requires just an addition of the pre-calculated score to the previous $H_{i,j}$.



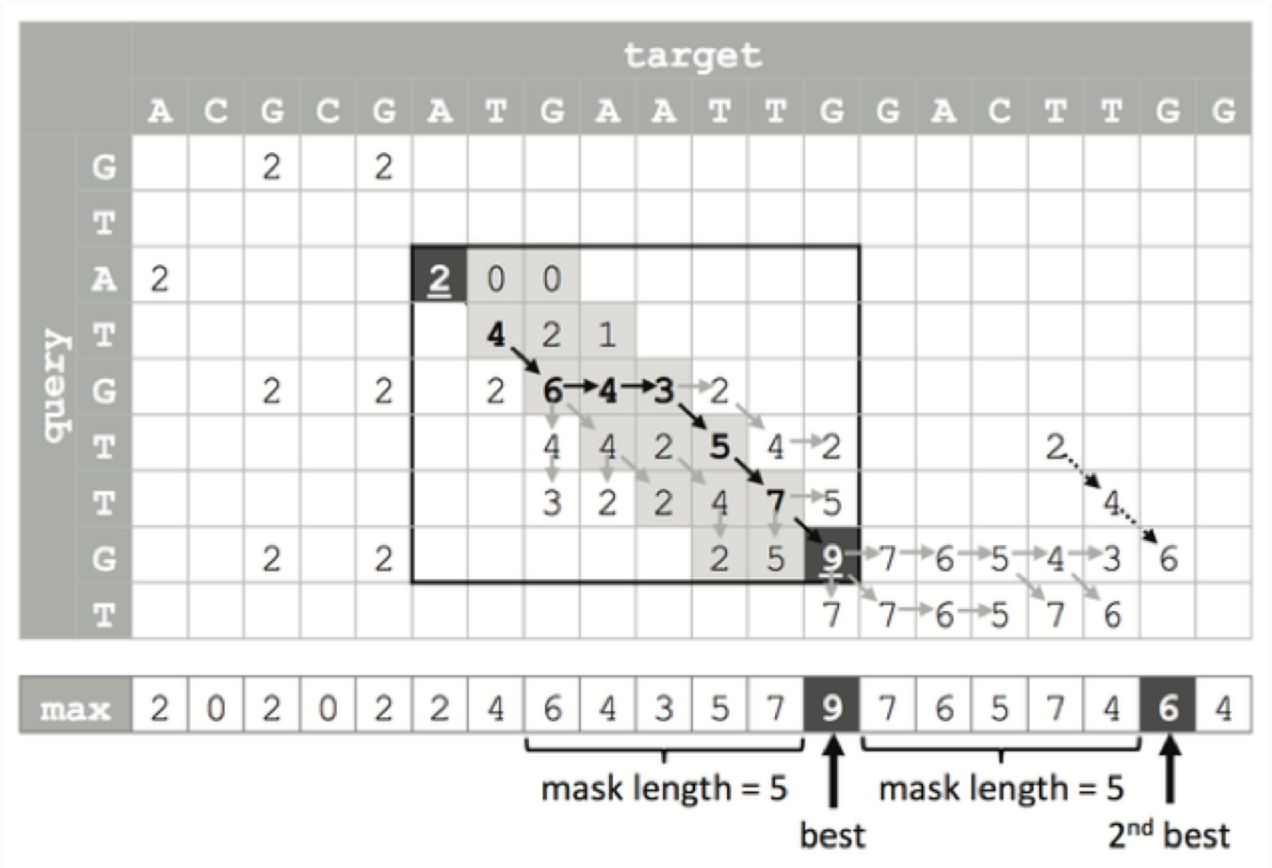
Working of SSW

1. Record the optimal alignment ending positions.
2. Store the maximal score of each column in a “max” array
3. Record the complete column with maximal score.
4. Get the optimal starting position using a reversed SW by calculating a much smaller scoring matrix.
5. Generate detailed alignment by a banded SW on a very small band characterised by the starting and ending positions.

Step 4, 5 on much smaller matrix -- less costly

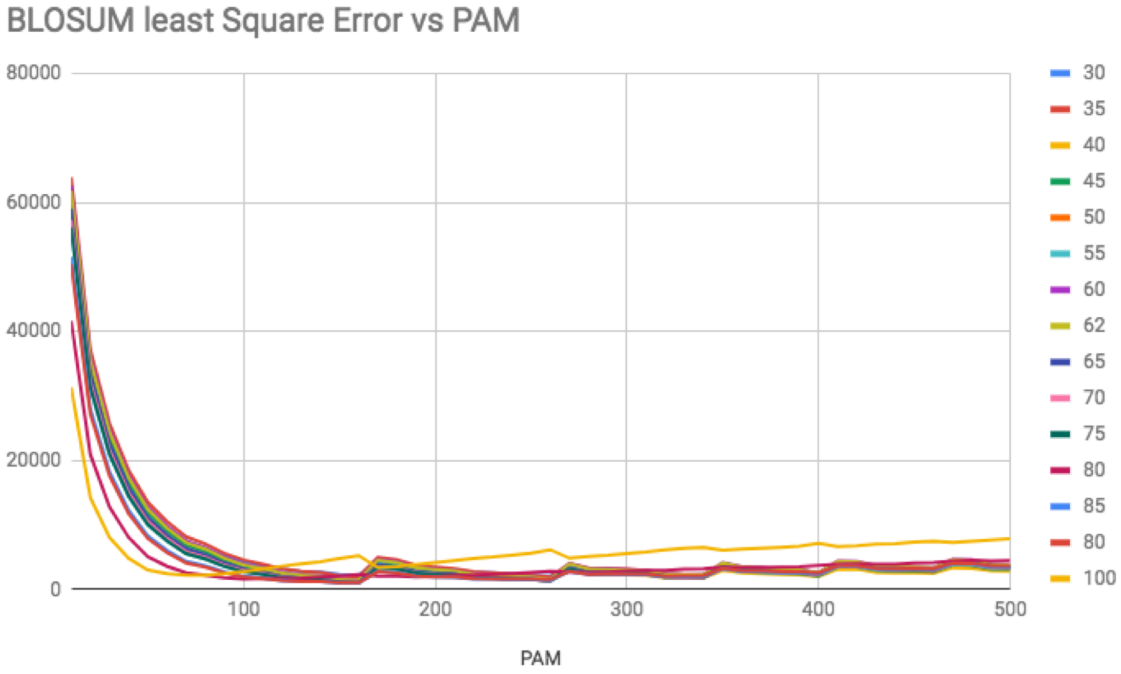
Match = 2
Mismatch = -1
Gap open = -2
Gap Ext. = -1

Illustration of alignment traceback and suboptimal alignment score determination.



Finding the Right Substitution Matrix

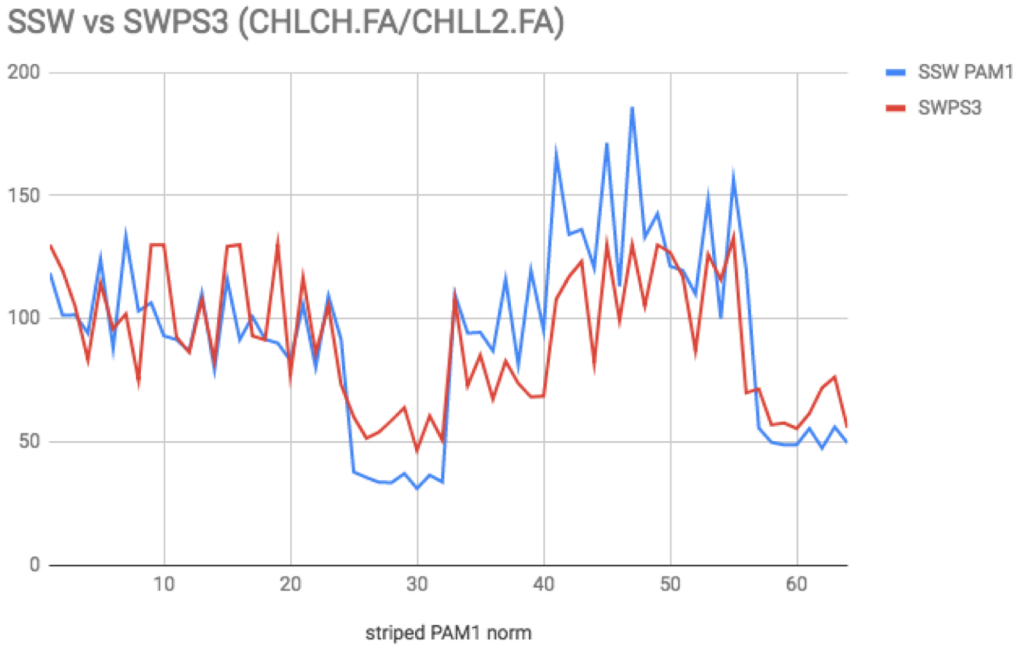
1. PAM260 and PAM160 matrix work the best.
2. Peaks emerge following PAM220 matrix.
3. Also, may have correlations with the fact that they align closely with BLOSUM.
4. Further interpretation of results needed



Results

1. **All pairs Matching**
   Striped Smith-Waterman vs SWPS3

	Small	**Large**
SSW	0.011 sec	202 sec
SWPS3	0.096 sec	2520 sec



Results

2. Over the Persona Framework:

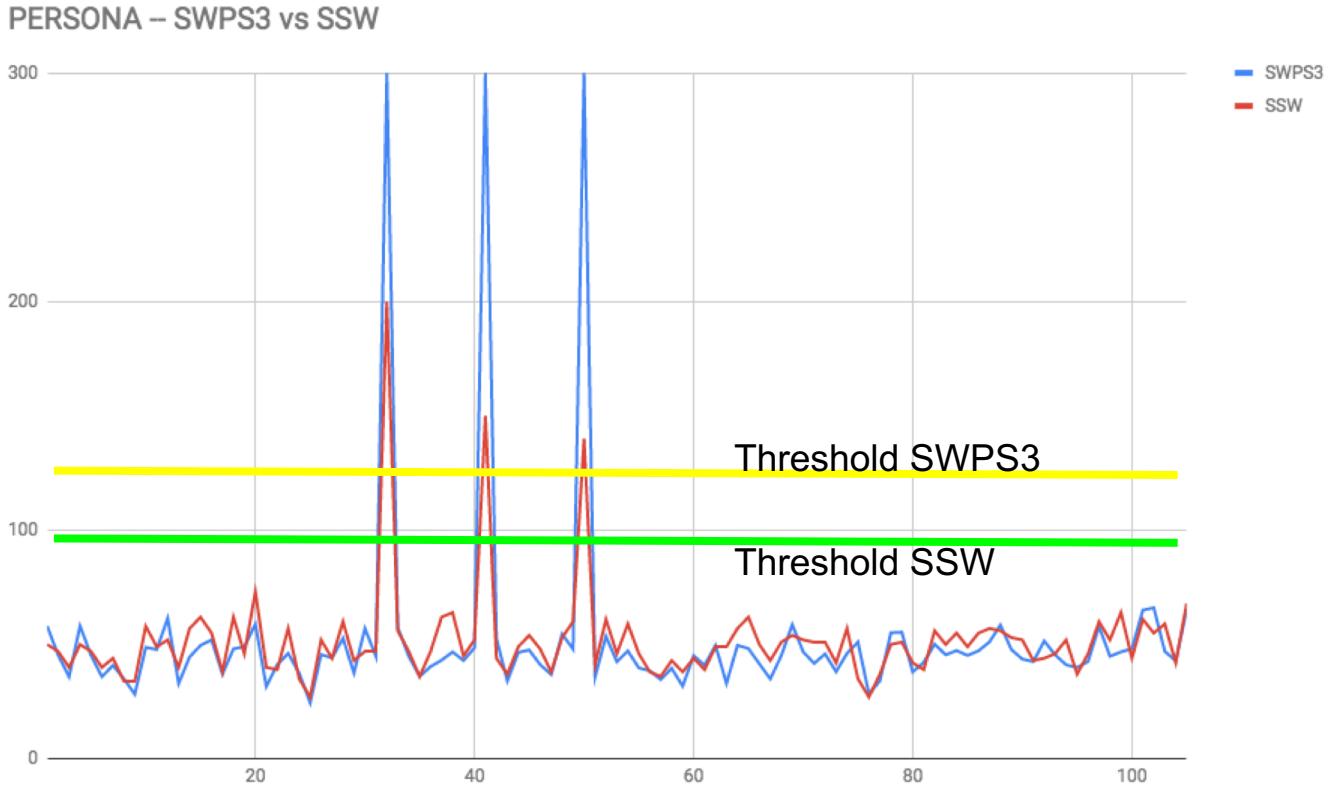
a. **Accuracy:** Results must obey the actual desired values irrespective of the speed-up
b. **Timing Constraints:** Speed-up a function of the fraction of time spent in alignment score computation

Results

a. **Accuracy:**

100 % match over Small Dataset

~99.9 % match over Large Dataset



Results

b. Timing:

I. **Small Dataset:** SSW = 0.607 sec, SWPS3 = 0.761 sec

II. **Large Dataset:** SSW = 530.36 sec, SWPS3 = 468.78 sec

Many optimisations possible :-)

--Calculates optimal and sub-optimal score
--Initialises substitution matrix for each pair
--Denormalizes each sequence every time it’s compared with another sequence.
--Banded Alignment not needed just for clustering
--Reverse Alignment not needed for clustering

Minhash

SW substitution matrices

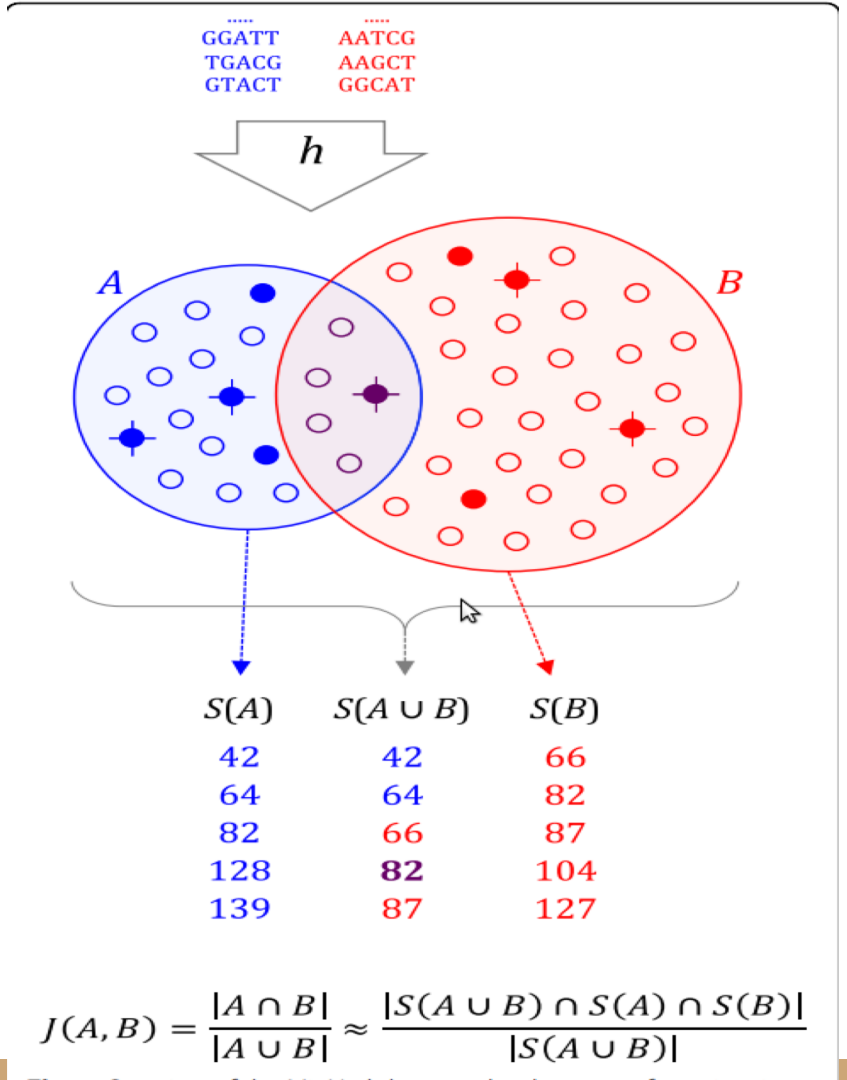
- BLOSUM (**BLO**cks **SU**bstitution **M**atrix)
- Point Accepted Mutation:
  - Single amino acid mutation (like SNV) that happens evolutionarily
  - Data is collected from the phylogenetic trees
  - PAM-1 is inferred
  - Markov chain model of PAMs give PAM-N

$$M_n = M_1^n$$

$$\mathrm{PAM}_n(i,j) = log\frac{f(j)M_n(i,j)}{f(i)f(j)} = log\frac{f(j)M^n(i,j)}{f(i)f(j)} = log\frac{M^n(i,j)}{f(i)}$$

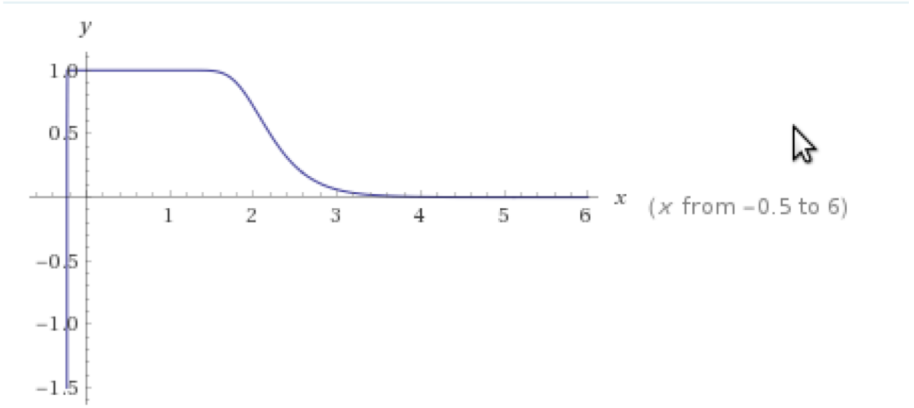
minhash

Jaccard:



How to choose the K-mer length

$$P(K \in X) = 1 - \left(1 - |\Sigma|^{-k}\right)^{n}$$



Plot of $1 - \left(1 - 20^{-x}\right)^{500}$

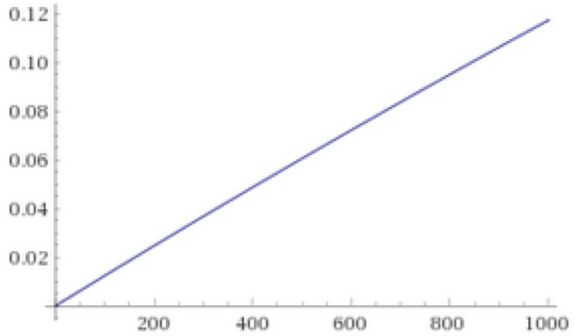
$$k' = \left\lceil \log_{|\Sigma|}(n(1-q)/q) \right\rceil$$

Till what lengths of protein is K=3 a good measure

Input interpretation:

plot	$1-\left(1-\frac{1}{20^3}\right)^x$	$x = 0$ to 1000

Plot:



Sketching and Final minhash distance

- We consider S sketches (s minimum hashed Kmers)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

If d is the probability of a single substitution, expected number of mutations in a k-mer = $\lambda = kd.$

Probability that no mutation will occur is e^{-kd}

Jaccard is an approximation of this value, so $D = -\frac{1}{k} \ln \frac{2j}{1+j}$

P value - Can we trust all these approximations.

$$r = \frac{P(K \in X)P(K \in Y)}{P(K \in X) + P(K \in Y) - P(K \in X)P(K \in Y)}$$

$$p(x; s; w; m) = 1 - \sum_{i=0}^{x-1} \frac{\binom{w}{i}\binom{m-w}{s-i}}{\binom{m}{s}}$$

Convergence of hypergeometric cdf into binomial

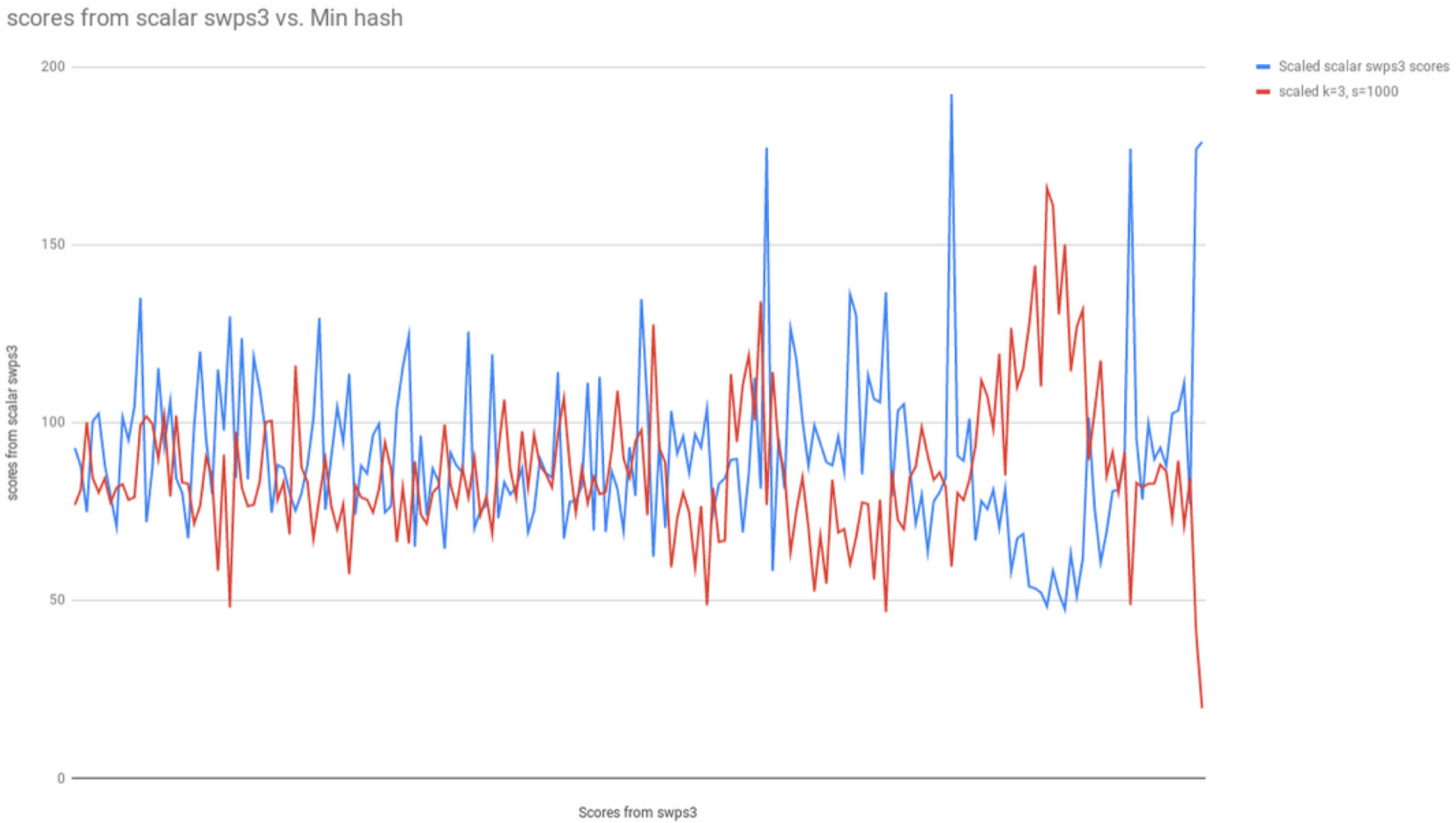
Table 1: The probability functions of the hypergeometric distribution with $n = 5$ and $r/N = 0.4$ for various values of N and of the binomial distribution with $n = 5$ and $p = 0.4$.

	$\binom{r}{y}\binom{N-r}{5-y}/\binom{N}{5}$				
	$N = 10$	$N = 20$	$N = 100$	$N = 1000$	
y	$r = 4$	$r = 8$	$r = 40$	$r = 400$	$\binom{5}{y}(.4)^y(.6)^{5-y}$
0	.0238	.0511	.0725	.0772	.0778
1	.2381	.2554	.2591	.2592	.2592
2	.4762	.3973	.3545	.3465	.3456
3	.2381	.2384	.2323	.2306	.2304
4	.0238	.0542	.0728	.0764	.0768
5	.0000	.0036	.0087	.0101	.0102

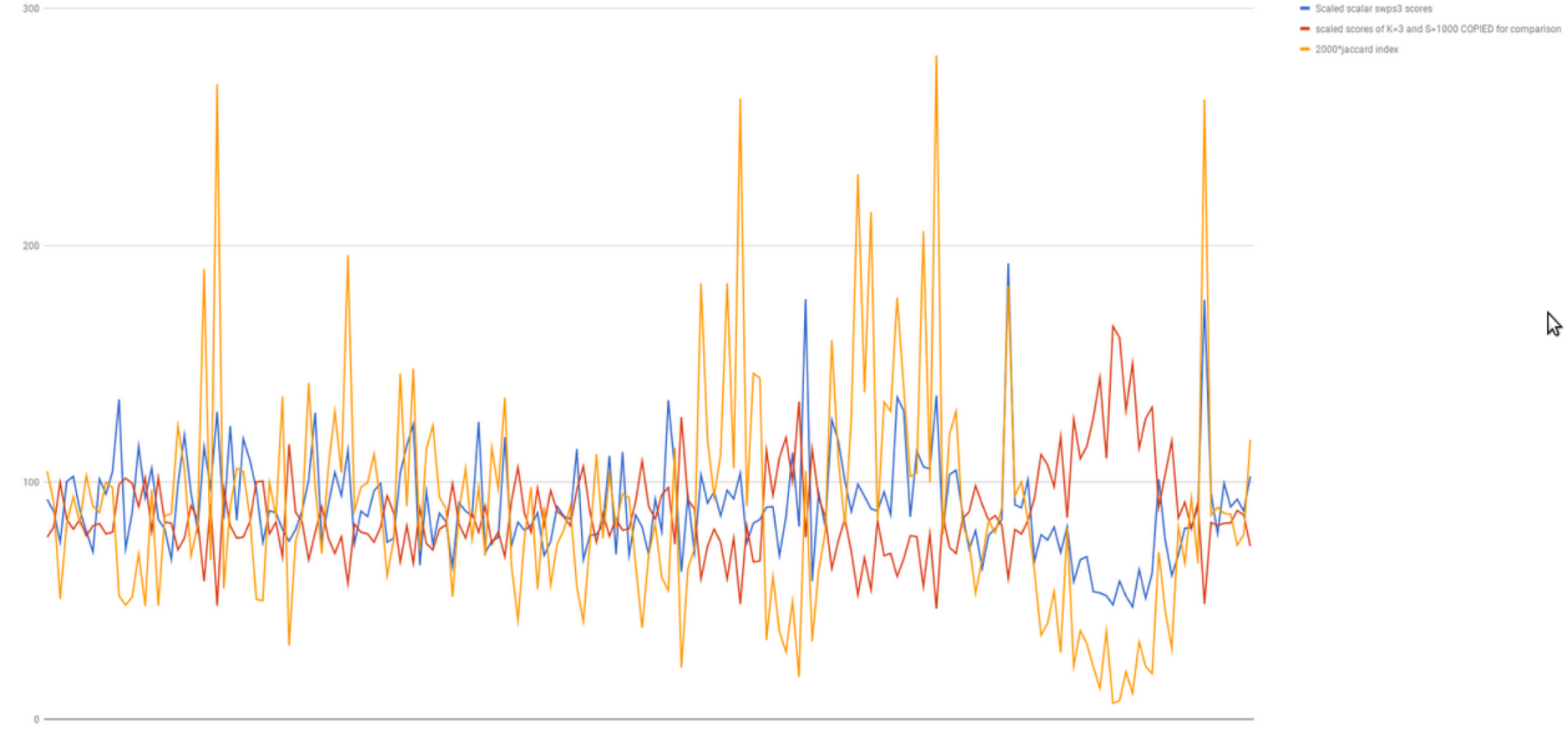
Probability of atleast x matches, given sketch s, r

$$p(x; s; r) = 1 - \sum_{i=0}^{x-1} \binom{s}{i} r^i (1-r)^{s-i}$$

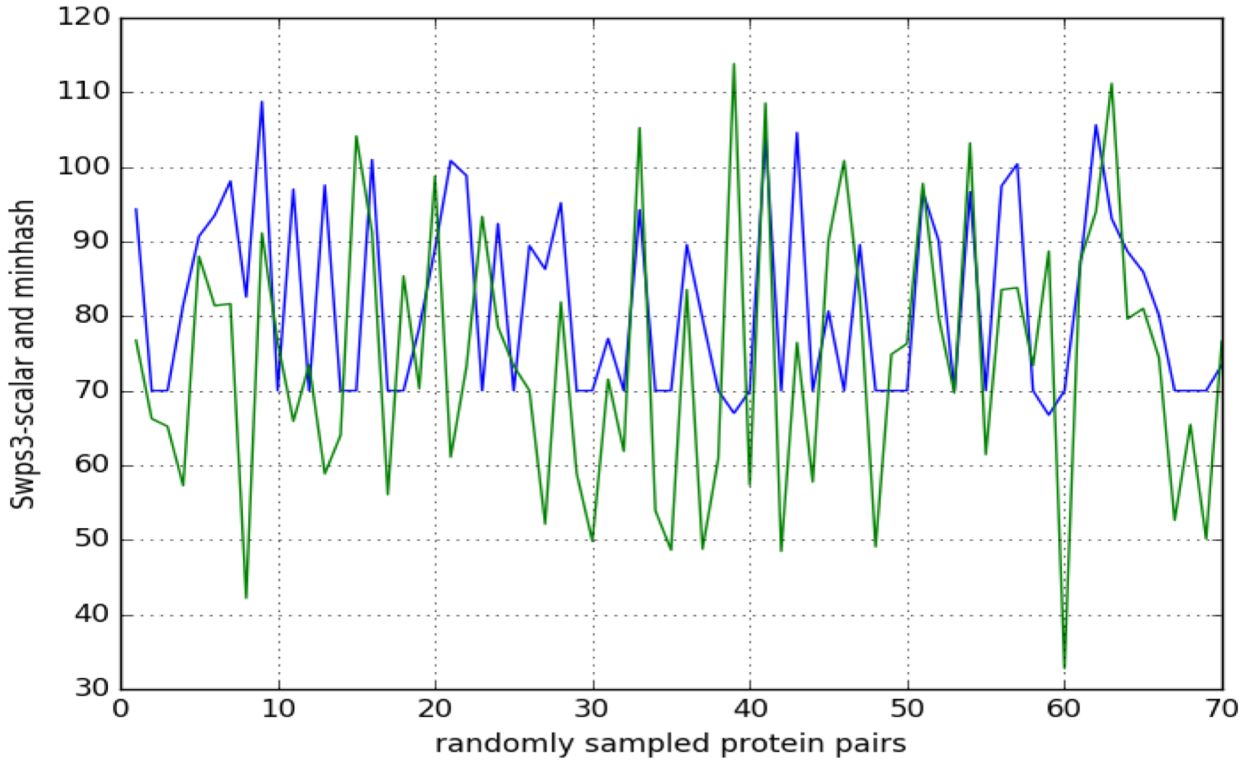
RESULTS?


scores from scalar swps3 vs. Min hash
200
150
100
50
0
scores from scalar swps3
Scores from swps3
Scaled scalar swps3 scores
scaled k=3, s=1000

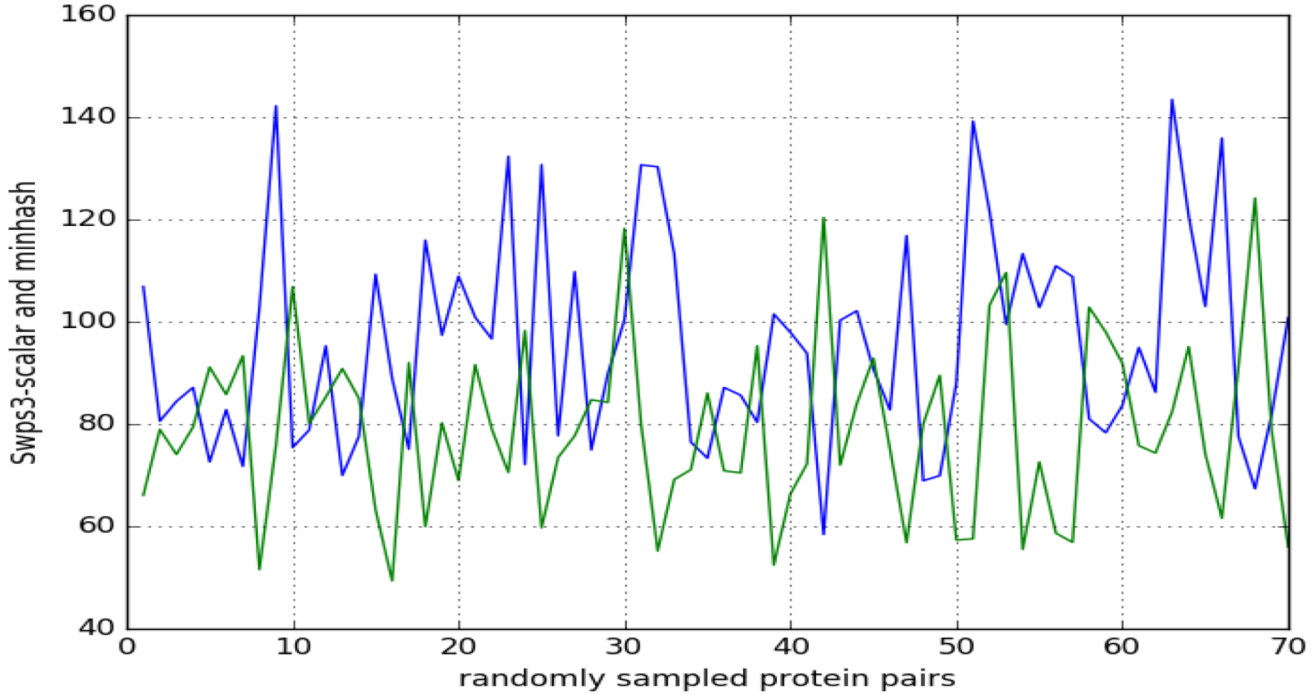
scores from scalar swps3 vs. minhash scores vs direct jaccard of k=3,s=1000



swps3-scalar vs minhash-k=4,s=1000 for bacteria data



swps3-scalar vs minhash-k=3,s=1000 for bacteria data



Time Complexity -

- For small dataset - with 8 proteins per fa, 0.008256 seconds
- For large dataset - with 2000 proteins per fa, 48.62 seconds

In the complete clustering framework
After tuning the parameters to match the number of clusters formed,
Total number of matches in small dataset with swps3 = 100%
Total number of matches in large dataset wrt swps = 70%

It misses 30% of matches (why? Due to abnormal peaks in both swps3 and ssw )

But, why is it slow- 3 times slower :(

- Currently, the coverage calculation is done by swps3
- Sketch of a pair is being calculated every time (But i observed 30-40 % decrease in time if we calculate sketch apriori for every chunk)

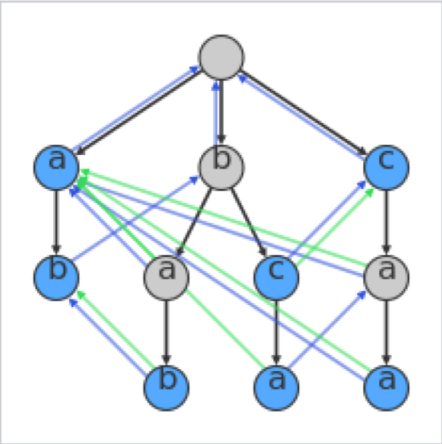
We can approximate coverage with P values.

Relation between sketch size and PAM number

S = PAM * Kmer

Because k =1 corresponds to 1% change :P

Aho - Corasick algorithm for exact matching



A visualization of the trie for the dictionary on the right. Suffix links are in blue; dictionary suffix links in green. Nodes corresponding to dictionary entries are highlighted in blue.

Reducing the clustering problem to inexact - AH

Helpful with bigger sequences
Once an edit distance K is fixed, we can compute the total number alignments possible in O(k,min(m,n)) time, (but an O(max(m,n))) pre-processing time.

THANK YOU