

APAC03

Standardization of Sequence Alignment Scores

Hilary S Booth, Ole M Nielsen,
John H Maindonald, Susan R Wilson
& Peter Maxwell

Centre for Bioinformation Science (CBiS)

Mathematical Sciences Institute & John Curtin School of Medical Research
& Australian Partnership for Advanced Computation

AUSTRALIAN NATIONAL UNIVERSITY

Email: Hilary.Booth@anu.edu.au

Comparing DNA or Protein Sequences

- Comparing protein (or DNA) sequences might be thought of (by a computer scientist) as comparing strings in a finite alphabet.
- But the assessment of their similarity is a complex biological problem, involving the structure of proteins and the function of genes.
- An alphabet of 4 letters (DNA)
- An alphabet of 20 letters (the amino acids of a protein molecule)
- How similar are two strings in the protein alphabet?

BLOSUM62 MATRIX

A R N D C Q E G H I L K M F P S T W Y V

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

$$\log\left(\frac{p_{ab}}{p_a p_b}\right)$$

Q: How reliable is this score?

The log odds in the BLOSUM62 matrix are an average taken over the entire proteome.

What if we have significantly different frequencies of residues?

CHCIVCKCDLC

CACVICRCAKC

Z-score over all permutations

- Take the (gapped) score of an alignment
- Calculate the scores for all possible permutations
- Take the mean and standard deviation of the scores
- Calculate Z-score of all permutations
- If we calculated the mean and standard deviation for all permutations, we would calculate $N!$ scores.
- The POZITIV algorithm is a fast way of calculating the (ungapped) mean and standard deviation of all possible permutations
- Two steps $O(N)$ and $O(1)$

A Diagram of All Permutations

Consider two aligned sequences in the following format:

	K	D	L	R	K	H
S	*					
K		*				
L			*			
R				*		
G					*	
H						*

where an asterix indicates a match.

A Diagram of All Permutations

Here is a permutation of the alignment:

K D L R K H

S	*				
K	*				
L		*			
R			*		
G				*	
H					*

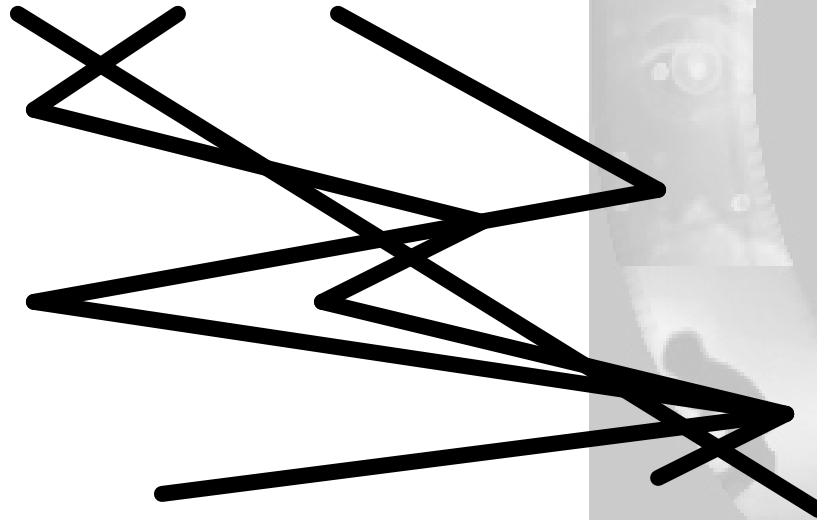
in which we swapped the order of the first two letters.

All permutations = all possible paths down the scoring matrix:

where every row and column is visited exactly once.

S
K
L
R
G
H

K D L R K H



Mean over all $N!$ paths

Using the symmetry of all paths down the matrix:

The mean over all paths is simply the mean of all of the matrix entries multiplied by the length of the paths N .

(N is the number of letters in the sequence alignment.)

Let the sum along a path be denoted by S_{path}

So that

$$S_{path} = s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N}$$

Then the variance over all paths is

$$\mathbf{s}_{perms}^2 = \mathbf{s}_{paths}^2 = \frac{1}{N!} \sum_{paths} S_{path}^2 - \mathbf{m}_{path}^2$$

Let the sum along a path be denoted by S_{path}

So that

$$S_{path} = s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N}$$

Then the variance over all paths is

$$\mathbf{s}_{perms}^2 = \mathbf{s}_{paths}^2 = \frac{1}{N!} \sum_{paths} S_{paths}^2$$


$$-m_{paths}^2$$

Easy term

Let the sum along a path be denoted by S_{path}

So that

$$S_{path} = s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N}$$

Then the variance over all paths is

$$\mathbf{s}_{perms}^2 = \mathbf{s}_{paths}^2 = \frac{1}{N!} \sum_{paths} S_{paths}^2 - \mathbf{m}_{paths}^2$$

Not so easy term

Consider term from previous slide:

$$\sum_{paths} S_{path}^2 = \sum_{paths} (s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N})^2$$


We can write this in terms of the scoring matrix $[M_{ij}]$

$$\sum_{paths} S_{path}^2 = (N-1)! \left(\sum_{i,j=1}^N s_{ij}^2 \right) + (N-2)! \left(\sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \right)$$

Consider term from previous slide:

$$\sum_{paths} S_{path}^2 = \sum_{paths} (s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N})^2$$

We can write this in terms of the scoring matrix $[M_{ij}]$

$$\sum_{paths} S_{path}^2 = (N-1)! \left(\sum_{i,j=1}^N s_{ij}^2 \right) + (N-2)! \left(\sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \right)$$


Consider term from previous slide:

$$\sum_{paths} S_{path}^2 = \sum_{paths} (s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N})^2$$

We can write this in terms of the scoring matrix $[M_{ij}]$

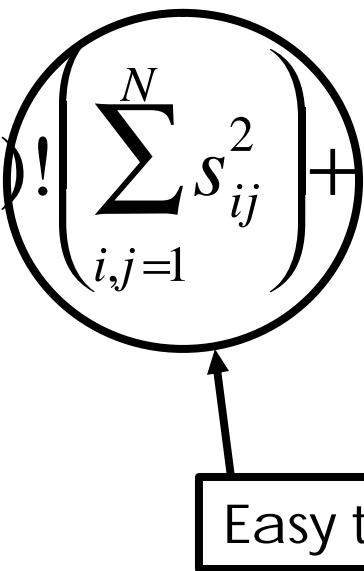
$$\sum_{paths} S_{path}^2 = (N-1)! \left(\sum_{i,j=1}^N s_{ij}^2 \right) + \underbrace{(N-2)!}_{\uparrow} \left(\sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \right)$$

Consider term from previous slide:

$$\sum_{paths} S_{path}^2 = \sum_{paths} (s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N})^2$$

We can write this in terms of the scoring matrix $[M_{ij}]$

$$\sum_{paths} S_{path}^2 = (N-1)! \left(\sum_{i,j=1}^N s_{ij}^2 \right) + (N-2)! \left(\sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \right)$$



Consider term from previous slide:

$$\sum_{paths} S_{path}^2 = \sum_{paths} (s_{i_1 j_1} + s_{i_2 j_2} + \dots + s_{i_N j_N})^2$$

We can write this in terms of the scoring matrix $[M_{ij}]$

$$\sum_{paths} S_{path}^2 = (N-1)! \left(\sum_{i,j=1}^N s_{ij}^2 \right) + (N-2)! \left(\sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \right)$$

Not so easy term

But we can calculate this term as follows:

$$\begin{aligned} & \sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \\ &= \left(\sum s_{ij} \right)^2 - \sum_i \left(\sum_j s_{ij} \right)^2 \\ & \quad - \sum_j \left(\sum_i s_{ij} \right)^2 + \sum s_{ij}^2 \end{aligned}$$

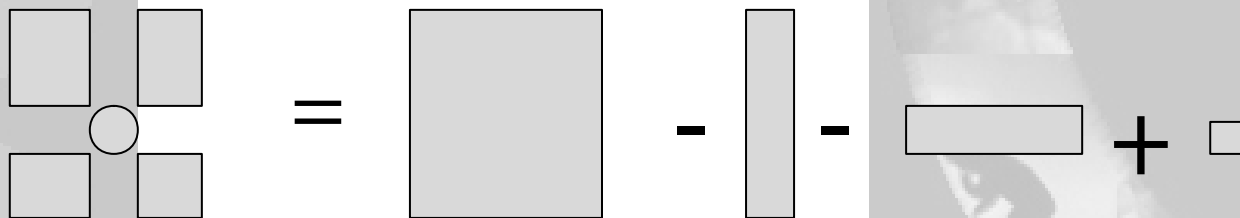
But we can calculate this term as follows:

$$\begin{aligned} & \sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right) \\ &= \left(\sum s_{ij} \right)^2 - \sum_i \left(\sum_j s_{ij} \right)^2 \\ & \quad - \sum_j \left(\sum_i s_{ij} \right)^2 + \sum s_{ij}^2 \end{aligned}$$

(which is not as bad as it looks)

Diagram of

$$\sum_{i,j=1}^N s_{ij} \left(\sum_{k \neq i, l \neq j} s_{kl} \right)$$



Collecting together similar terms further simplifies calculations

Additional economy gained by tabulating the joint frequency of residues in the two sequences in a 20 x 20 matrix.

This is an $O(N)$ calculation - just counting the frequencies of the residues

Then the calculations are done on a 20 x 20 matrix in $O(1)$ time.

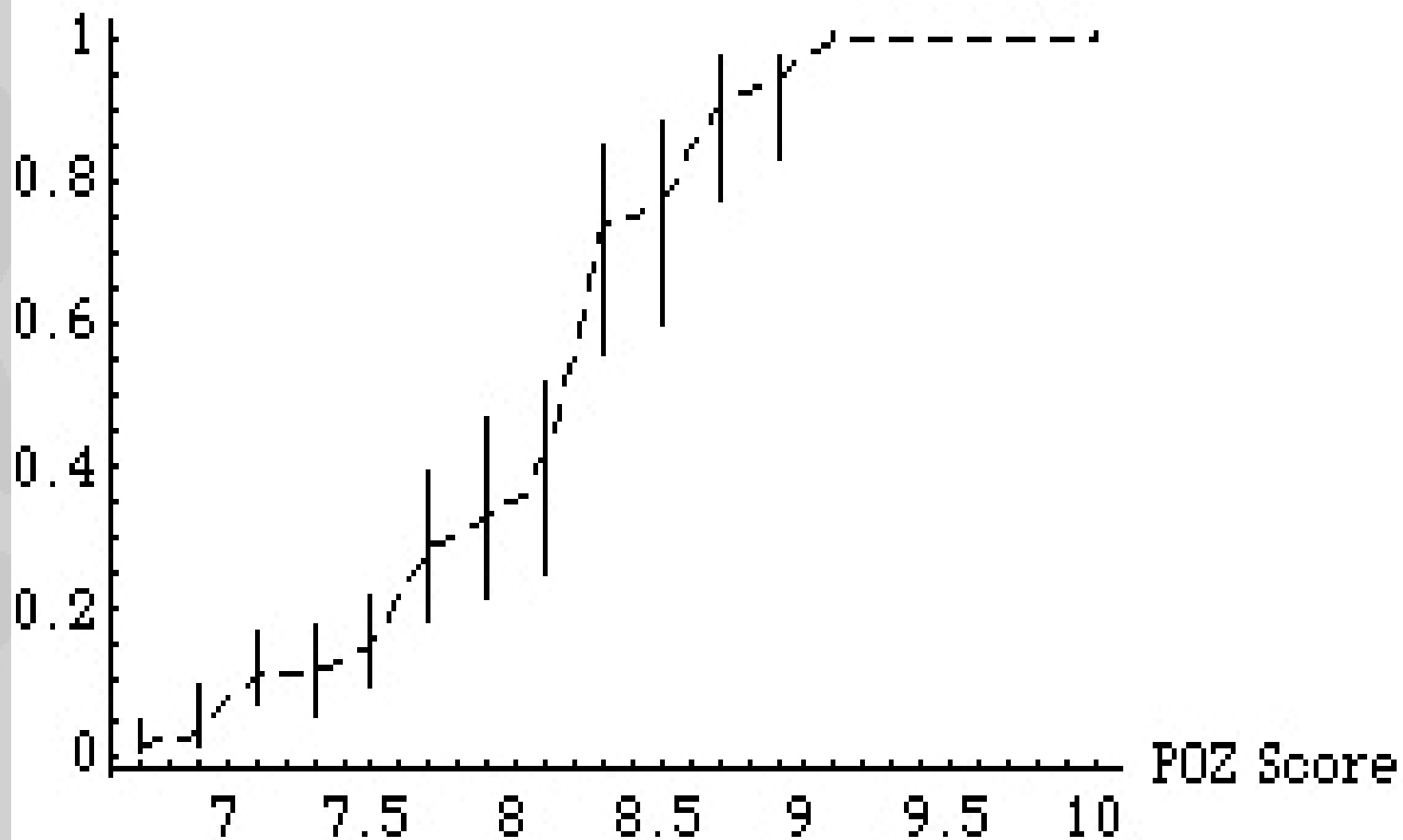
We used the POZ-score as secondary filter on sequences which had already been optimally aligned using the Smith-Waterman algorithm.

- Each Smith-Waterman alignment is given a POZ score using all combinations
- Sequences are ranked according to decreasing POZ score

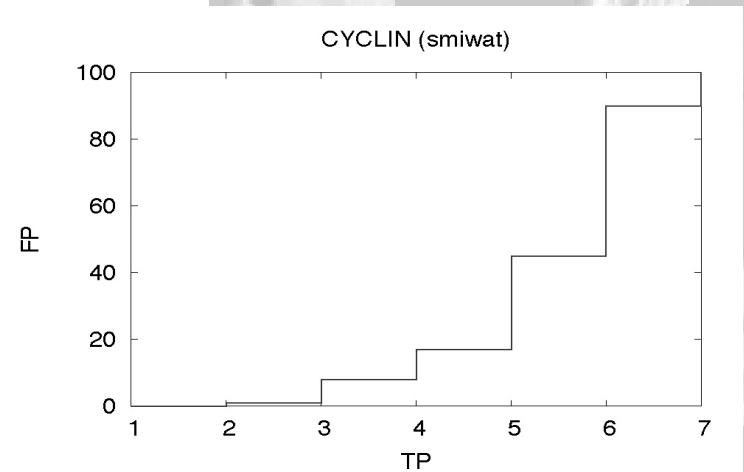
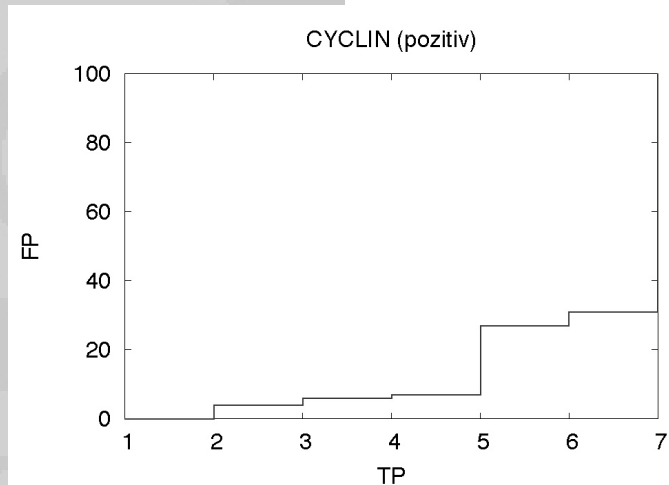
We tested the POZ-score on the Aravind data set of yeast protein sequences.

- Aravind data available from NCBI
- 6341 yeast sequences
- 104 query sequences (not repeated in data)
- 104 files listing true positive matches
- Total of approx 1200 true positives
- Approx 12 trues per query
- But varying from 2 to 140 trues
- Variable length and “quality” of match

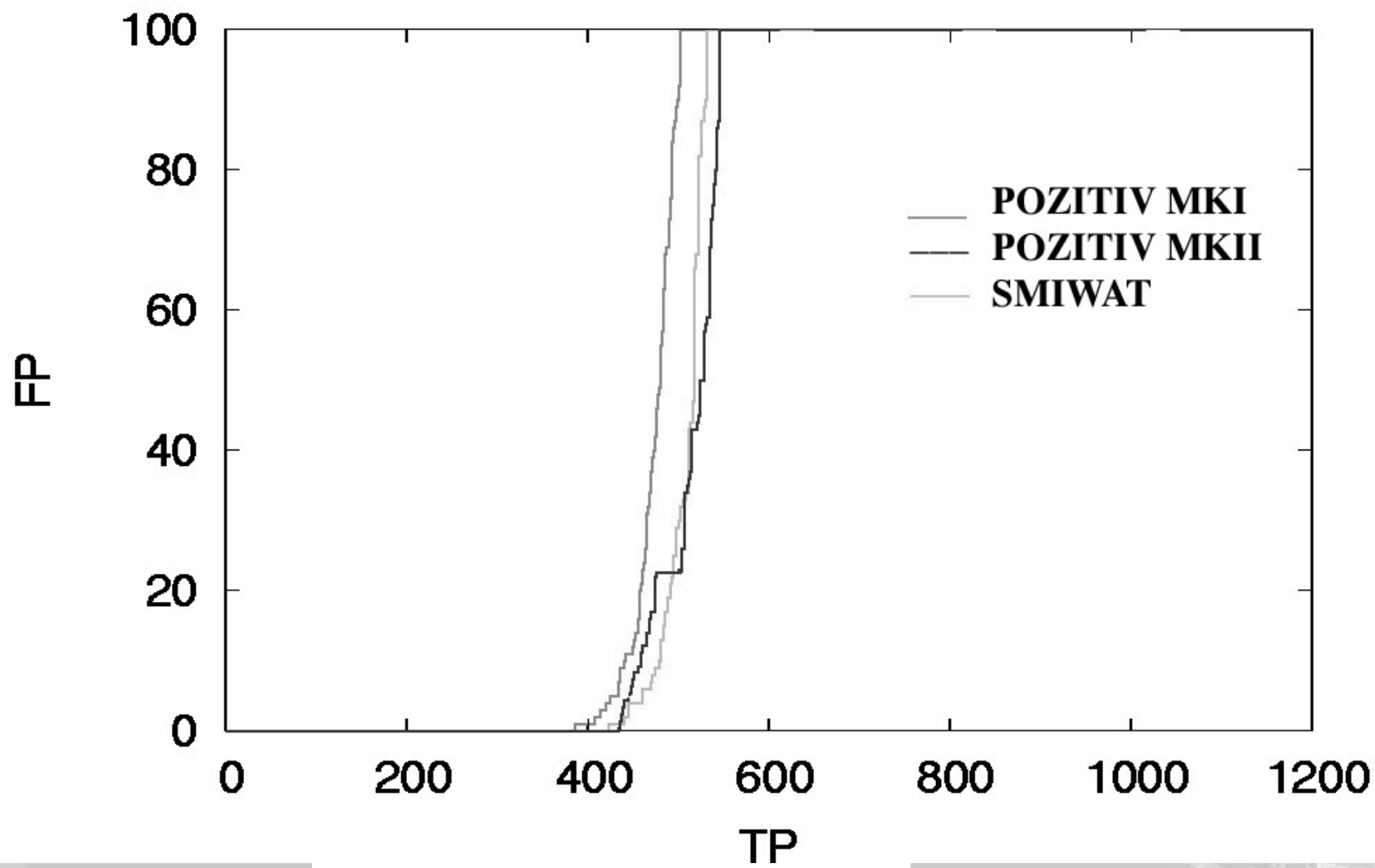
Prob T



Individual queries improved by using normalized score for example CYCLIN:



- Of the 104 queries, 15 were improved using the normalized scores
- Wouldn't expect the scores to improve on average since the BLOSUM matrices are based on averages over the whole database



Results

- Can measure distance from mean of all permutations in $O(N)$ time
- Can give a P-value based upon empirical measurements
- $P(T \mid \text{POZ} > 9.5) = 1$
- Is likely to improve search results only when composition differs from average background frequencies
- Tendency to be more reliable with long “true” matches than with short exact “partial” matches
- When testing a set of queries against a database, normalized score is therefore is not likely to improve the results “on average”
- Possibly performs better when comparing across species due to different compositions

Implementation of POZITIV

- Implemented using Python C-extensions for the computationally intensive parts of each pair-wise match and parallelism using MPI bindings for Python
- The parallelisation was done across the underlying database so that each query was matched against several database entries simultaneously.
- The POZITIV algorithm and the underlying optimised routines were released as open source in January 2003.
- See <http://datamining.anu.edu.au/software/pozitiv/> for details.

APAC

- The parallel version was run on the APAC supercomputing facility at the ANU and achieved a speedup of about 25 on 32 processors.
- Together with other optimisations we were thus able to match 104 proteins to a database containing 6341 yeast proteins in 6 minutes whereas the same problem was completely infeasible prior to optimising and parallelising.
- This is orders of magnitude faster than a similar program called "fastpairwise.py" which is part of a package known as BioPython available at URL <http://biopython.org/>. The group is currently looking into merging our technology into this package.

PyPar

- The parallelism was achieved through a Python-MPI binding called Pypar, developed at the ANU, which allows programs written in the Python programming language to run in parallel on multiple processors and communicate using the message passing interface standard MPI on the APAC Alpha server as well as Solaris and Linux platforms.
- Pypar was first published as open source on 7 Feb 2002 and is available at URL <http://datamining.anu.edu.au/~ole/pypar>.

Acknowledgements

- Peter Maxwell (CBiS)
- Sue Wilson (CBiS & MSI)
- Ole Nielsen (APAC & MSI)
- John Maindonald (CBiS & MSI)
- Andrew Butterfield (CBiS)
- Margaret Kahn (APAC)



Example

- To search yeast database using POZITIV see
- <http://cbis.anu.edu.au/~maxwell/tsearch.cgi>
- MSSNLTEEQIAEFKEAFALFDKDNNGSISSELATVMRSLGLSP
SEAEVNDLMNEIDVDGNHQIEFSEFLALMSRQLKSNDSEQELL
EAFKVFDKNGDGLISAAELKHVLT SIGEKLTD AEVDDMLREVS
DGSGEINIQQFAALLSK
- True Positive list:
536373,1420581,486337,2131118,786306,849194,1322650,1301979,5
57855,1323073,665661,1370553,536069,1301981,1323153,595536

BioInfoSummer Dec 1-5 at ANU

Newcomers to bioinformatics welcome

www.maths.anu.edu.au/events/BioInfoSummer



AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE
SUMMER SYMPOSIUM IN BIOINFORMATICS
1 - 5 DECEMBER, 2003

KEYNOTE SPEAKERS AND
INTRODUCTORY LECTURES
IN BIOINFORMATICS

AMSI TRAVEL SCHOLARSHIPS
FOR HONOURS AND PHD STUDENTS

<http://www.amsi.org.au/conferences/2003/BioInfoSummer.html>

CENTRE FOR BIOINFORMATION SCIENCE (CBIS)
AUSTRALIAN NATIONAL UNIVERSITY

Contact: BioInfoSummer@cbis.anu.edu.au