sequence search: …algorithms…

simulated PCR (e.g. rePCR)

- query using primer pairs (conserved flanking regions)
  - or mixtures of primers
- primers bind to opposite strands and face each other
- allows for some degree of primer/template mismatch
- range of distance between primers is user specified
- returns ‘amplicon’ position(s), degree of primer match

sequence search: …algorithms

machine learning (e.g. SDN2GO)

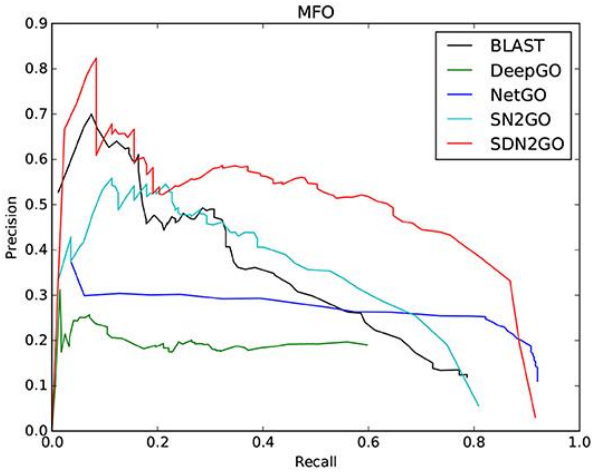
query DNA twice* (both orientations)

complex models of reference sequences

used to find features, annotations, or metadata

e.g. splice sites, GO function, taxon

typically no alignment is output



(Cai et al. 2020; https://doi.org/10.3389/fbioe.2020.00391)

sequence search: pairwise alignment

SEQHP (Goad and Kanehisa 1982; https://doi.org/10.1093/nar/10.1.247)

- alignment of query (both orientations) to all references
  - ‘global’ alignment (all positions aligned)
  - ‘local’ alignment (core most similar positions aligned)
- compute similarity
  - edit distance (a.k.a. p–distance, raw distance)
  - distance from ‘average’ similarity matrices
  - return the most similar sequences/fragments
- effective, but slow

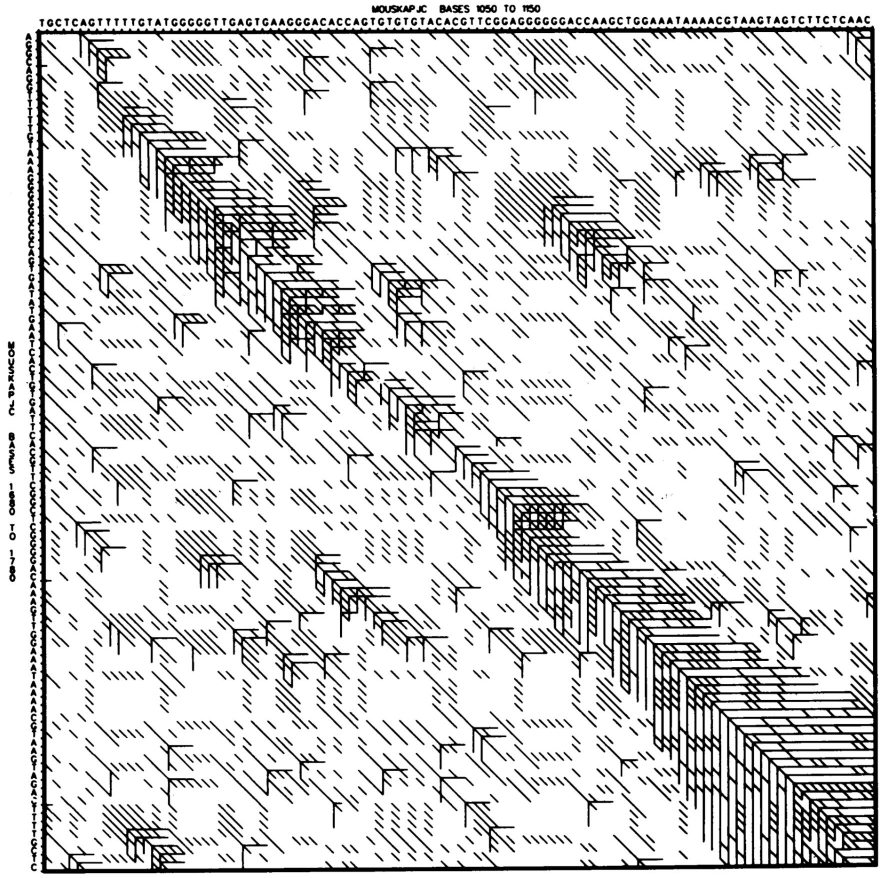


Figure 2. A forward path matrix for the comparison of bases 1050 to 1150 and 1680 to 1780 within the 5495 base segment of the mouse immunoglobulin κ light chain gene complex (Ref. 9).

(Goad and Kanehisa 1982; https://doi.org/10.1093/nar/10.1.247)

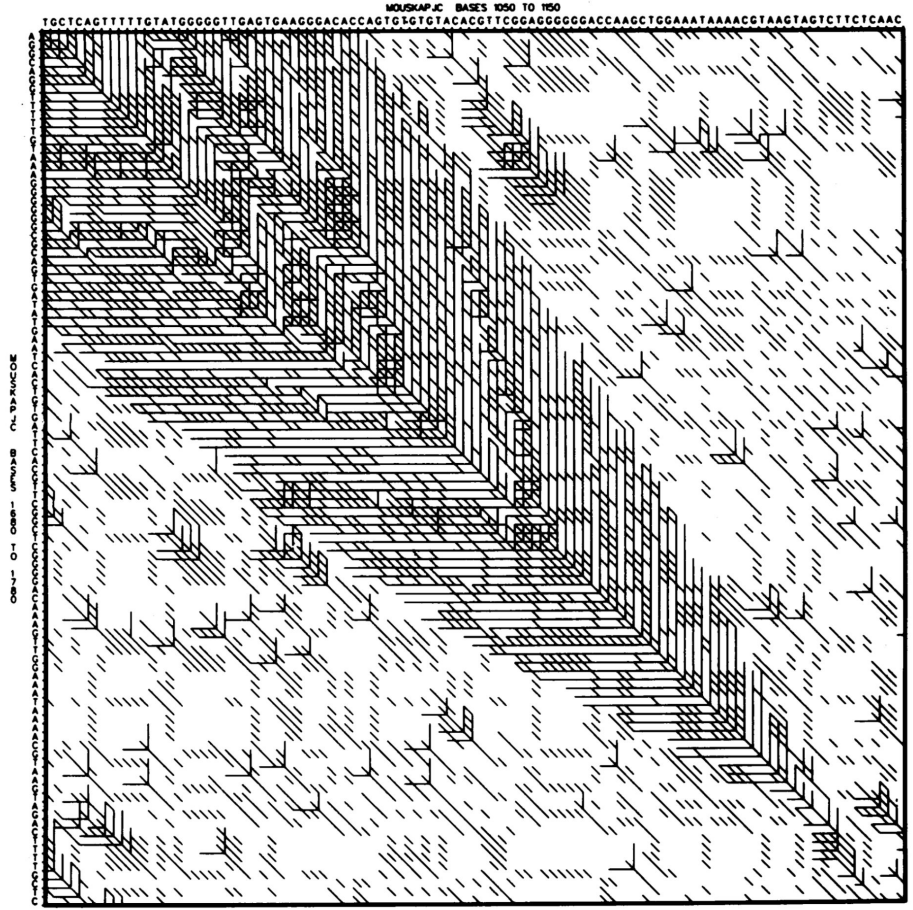


Figure 3. The corresponding reverse path matrix.

(Goad and Kanehisa 1982; https://doi.org/10.1093/nar/10.1.247)

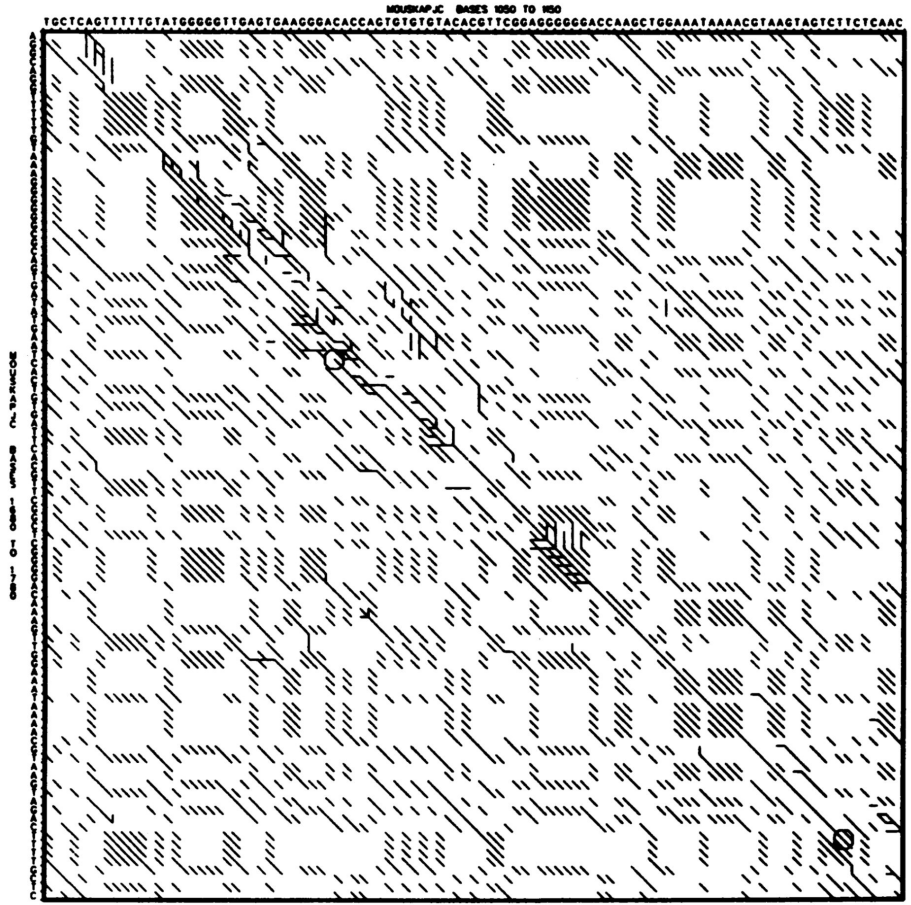


Figure 4. The logical product of the two path matrices.

(Goad and Kanehisa 1982; https://doi.org/10.1093/nar/10.1.247)

sequence search: FAST[P|A]...

Lipman and Pearson (1985; https://doi.org/10.1126/science.2983426)

Pearson and Lipman (1988; https://doi.org/10.1073/pnas.85.8.2444)

the ‘descendant’ of SEQHP, the ‘ancestor’ of BLAST

search for ‘similar’ sequences by filtering out dissimilar sequences first

- use a pre–computed table of kmer by sequence position
- compute offsets to find clustered similarities (diagonals)

estimate sequence similarity (number of shared kmers)

- pick best segments
- join segments, recalculate similarity with substitutions

local alignment of the highest–scoring segments

sequence search: ...FAST[P|A]...

	kmer 0	kmer 1	kmer 2	kmer 3	kmer 4	kmer 5	...
query	1, 28	20, 22	15	5	2, 10	12	...
sequence 0	5, 32	24, 26	19	9	6, 14	16	...
sequence 1	1	–	28, 13	–	–	–	...
sequence 2	–	1	–	–	–	7, 11	...
sequence 3	4, 31	23, 25	18	–	5, 13	15, 32	...

kmer by sequence position

sequence search: ...FAST[P|A]...

	kmer 0	kmer 1	kmer 2	kmer 3	kmer 4	kmer 5	...
query	0, 0	0, 0	0	0	0, 0	0	...
sequence 0	4, 4	4, 4	4	4	4, 4	4	...
sequence 1	0, –	–	13, –	–	–	–	...
sequence 2	–	-19	–	–	–	-5, –	...
sequence 3	3, 3	3, 3	3	–	3, 3	3, –	...

kmer offsets

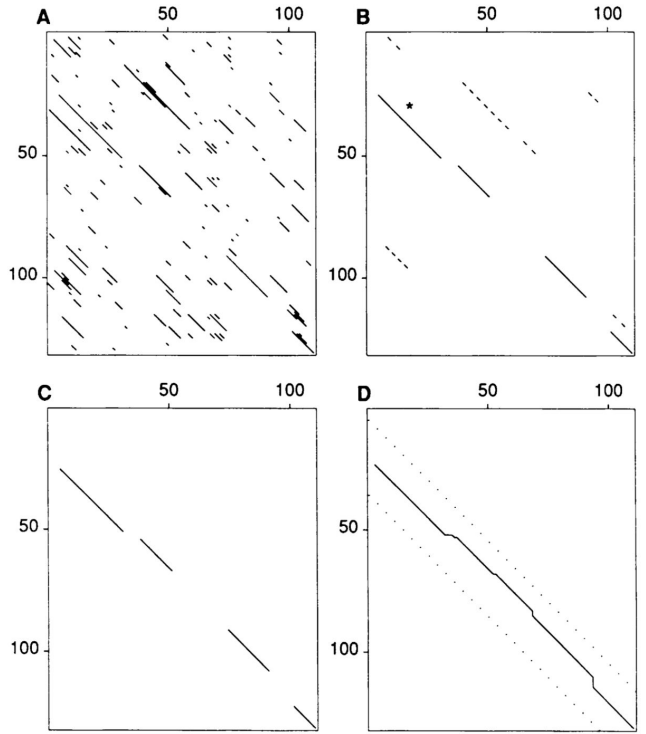


FIG. 1. Identification of sequence similarities by FASTA. The four steps used by the FASTA program to calculate the initial and optimal similarity scores between two sequences are shown. (*A*) Identify regions of identity. (*B*) Scan the regions using a scoring matrix and save the best initial regions. Initial regions with scores less than the joining threshold (27) are dashed. The asterisk denotes the highest scoring region reported by FASTP. (*C*) Optimally join initial regions with scores greater than a threshold. The solid lines denote regions that are joined to make up the optimized initial score. (*D*) Recalculate an optimized alignment centered around the highest scoring initial region. The dotted lines denote the bounds of the optimized alignment. The result of this alignment is reported as the optimized score.

(Pearson and Lipman 1988; https://doi.org/10.1073/pnas.85.8.2444)

aa substitution matrices...

derived from ‘curated’ multiple sequence alignments

highly sample dependent

assume alignment is correct

assume sequences are homologous

PAM (Dayhoff, Schwartz, and Orcutt 1978)

- Point Accepted Mutation
- PAMx: x = x mutations per 100 amino acids
- derived from alignments of protein ‘families’

…aa substitution matrices

BLOSUM (Henikoff and Henikoff 1992)

- BLOcks of Amino Acid SUbstitution Matrix
- BLOSUMx: x = sequence merge threshold
  - higher numbers = more similar sequences
- derived from indel free alignment segments
- original BLOSUM62 has a mathematical error
  - use corrected version
  - (except error version works better)

…aa substitution matrices

BLOSUM62

	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	-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	-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

E	L	V	I	S	I	S	D	E	A	D
|	|	|	|	|				|		
E	L	V	I	S	L	I	V	E	S	-
5	4	4	4	4	2	-2	-3	5	1	0

= 29 - 5

= 24

nt substitution matrices…

formulae can be used, but generally are not

- JC69 (Jukes and Cantor 1969), K80 (K2P; Kimura 1980), HKY85 (Hasegawa, Kishino, and Yano 1985), T92 (Tamura 1992), GTR (Tavaré 1986)...

nucleotide matrices not usually based on empirical data

- equal weights commonly used

…nt substitution matrices

+5/-4 (match vs. mismatch)

```
A  T  G  C  C  T  G  C  A  C  G  C
|  |  |  |     |  |  |  |     |  |
A  T  G  C  A  T  G  C  A  T  G  C
5  5  5  5 -4  5  5  5  5 -4  5  5
```

= 50 - 8

= 42

42/60 = 70%

+1/-2 (match vs. mismatch)

```
A  T  G  C  C  T  G  C  A  C  G  C
|  |  |  |     |  |  |  |     |  |
A  T  G  C  A  T  G  C  A  T  G  C
1  1  1  1 -2  1  1  1  1 -2  1  1
```

= 10 - 4

= 6

6/12 = 50%

sequence search: …FAST[P|A]...

assessment of statistical significance

- using RDF or RDF2

permute (shuffle) reference sequences

- the entire database or just similar sequences (faster)
- number of positions and base composition constant
  - order jumbled

recalculate similarity

sequence search: …FAST[P|A]

p–values are not calculated (distribution not normal)

z–values used instead

z = (similarity - mean of permuted similarity) / standard deviation of permuted similarity

z > 3 = possibly significant (0.0013%)

z > 6 = probably significant

z > 10 = significant

Table 3. Statistical significance (z value) of protein similarity scores. Protein sequences from the searches discussed in examples 1, 2, and 3 were compared with the best related and unrelated library sequences found. Z values [(score − mean score)/standard deviation] were calculated for the initial score from the mean and standard deviation of the database initial scores (initial scan), and for the initial (I) and optimized (O) scores from the mean and standard deviation of scores against randomly permuted versions of the database sequence in question. In the latter case, 50 comparisons (ktup = 1) were made with shuffled sequences.

Query sequence	Library sequence matched		Initial scan	Randomized	
	Identifier	Protein		I	O
OKBO2C (bovine cyclic AMP kinase)	TVBY8	Yeast cell cycle control	11.3	11.1	24.9
	TVFV-R	Src	11.0	10.4	23.6
	TVMS M	Mos	9.3	7.6	10.1
ANRT (rat angiotensinogen)	ITHU	Alpha-1 antitrypsin	10.0	8.3	25.8
	G1HUNM	Human Ig heavy chain (V-2)	6.8	8.1	7.8
	XHHU3	Antithrombin	5.3	5.1	24.1
	ITHUC	Alpha-1 antichymotrypsin	5.0	3.9	15.5
	TVMV-S	PDGF-related sis	4.6	3.8	3.8
VHVUNH (snowshoe hare bunyavirus nucleoprotein)	ORBPL	Lambda replication protein	4.4	4.0	2.5
	GHRB	Rabbit Ig gamma C region	4.3	4.8	3.6

(Lipman and Pearson 1985; https://doi.org/10.1126/science.2983426)

sequence search: FAST[P|A] vs. BLAST

BLAST has larger default word size (== faster)

- can be less ‘sensitive’ (ignores low similarly sequences)

BLAST focuses on most informative kmers

improved treatment of indels; tolerates

- ‘small’ indels
- ‘short’ segments of mismatch

different methods of calculating significance

- BLAST has a ‘significance’ of match
  - (assuming that search settings are correct)

sequence search: ‘BLAST’ flavors…

Altschul et al. (1990, 1997), Camacho et al. (2009)

- the ‘original’ source of algorithms
- 1997 overcomes non–matching regions (e.g. indels)
- freely available from NCBI (binary and code)

Kent (2002; BLAT)

- faster in some circumstances (e.g. large batches)
- designed to search against whole genomes
- freely available from the author (binary and code)

sequence search: …‘BLAST’ flavors…

Gish (2006; WU-BLAST)

- same algorithm as NCBI BLAST, but faster
- different default settings => different results
- binary (possibly available), but code is not distributed

Edgar (2010; USEARCH)

- kmer–based algorithm with alignment
- up to 100× faster
- ‘freemium’ model (32–bit version free; 64–bit version paid)
- binary (possibly available), but code is not distributed

sequence search: …‘BLAST’ flavors…

Panagiotis and Sahinidis (2011; GPU-BLAST)

- requires a Nvidia (CUDA) graphics processor
- up to 10× faster
- protein sequences only
- database must fit into the GPU memory
- freely available from the authors (binary and code)

sequence search: …‘BLAST’ flavors…

Zhao and Chu (2014; G-BLASTN)

- requires a Nvidia (CUDA) graphics processor
- up to 14× faster
- nucleotide sequences only
  - inefficient on short sequences (> 1,000,000 bp)
- database must fit into the GPU memory
- freely available from the authors (binary and code)

sequence search: …‘BLAST’ flavors

Buchfink et al. (2021; DIAMOND)

- up to 2500× faster
- protein sequences only
- uses a reduced amino acid alphabet (11)
- freely available from the authors (binary and code)

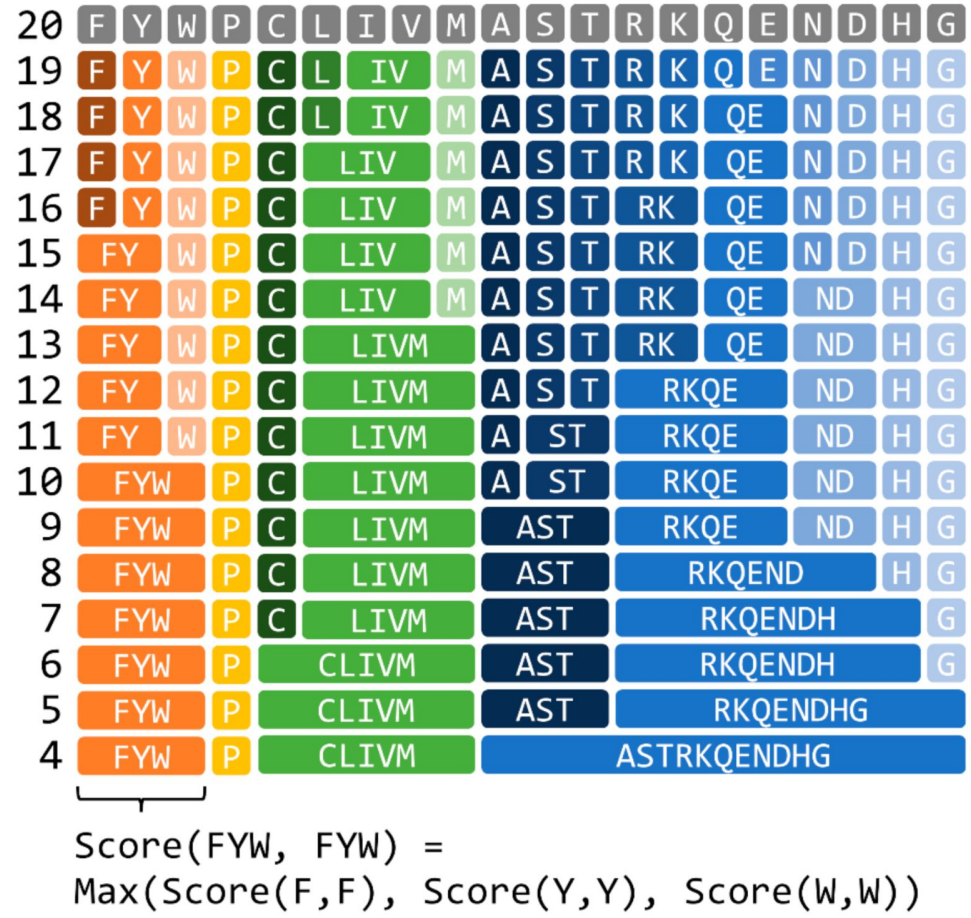


Figure 1. Reduced amino acid alphabet generated using the method proposed by Murphy et al. [9].

(Takabatake et al. 2021; https://doi.org/10.3390/genes12091455)

sequence filtration

masks (removes) repetitive or low complexity elements

- usually replaced with ‘N’ or ‘X’
- sometimes used to remove vector sequence

can improve sequence searches

- fewer false matches, faster

dust (nt) and seg (aa), and xnu (aa) are commonly used

can change percent similarity value

run by default with BLAST

BLAST: the algorithm

filter sequence to remove low complexity elements

reduce query to kmers

look for sequences with matching kmers

    exact or close matches

extend matches in both directions (local alignment)

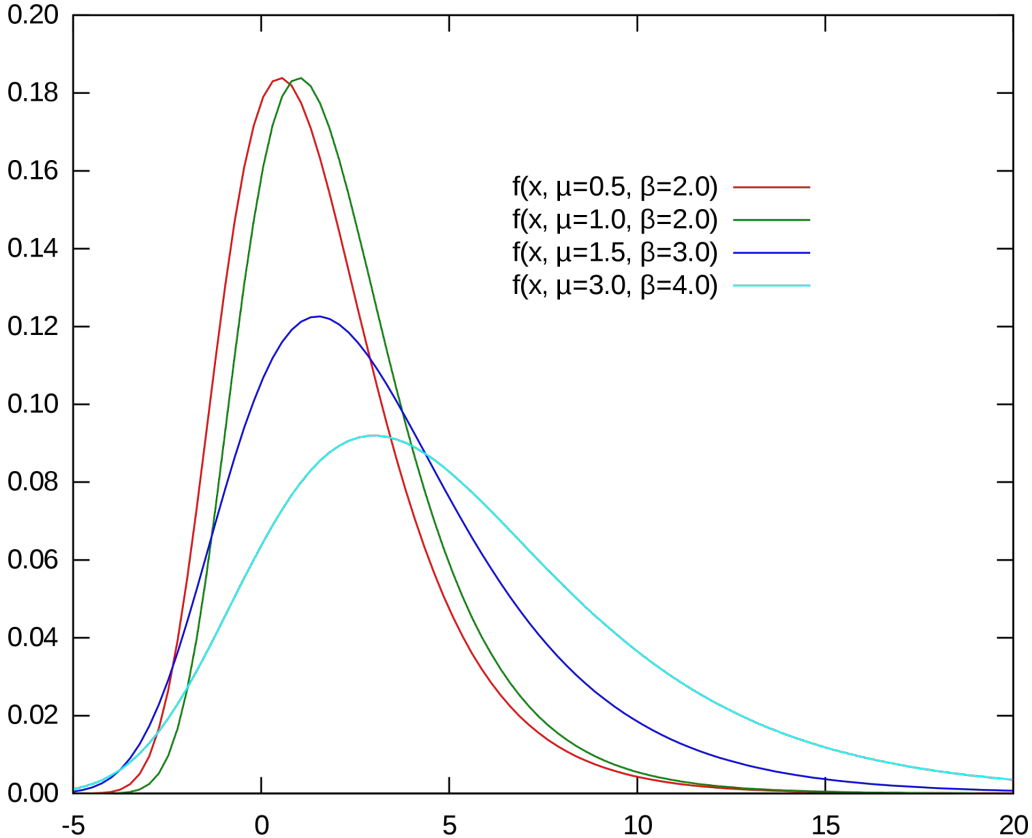
    score segments

examine high scoring (statistically significant) segments

combine segments

construct full local alignment

Gumbel extreme value



(https://commons.wikimedia.org/w/index.php?curid=4787663)

BLAST: significance

Karlin and Altschul (1990), Karlin et al. (1990)

Gumbel extreme value distribution used

- replaces sequence permutations (much faster)
- modified to fit properties of the database
  - sequences, indel scores, and the substitution matrix

minimum–maximum global alignment score can be calculated for any pair of sequences

random sequences follow the Gumbel extreme value distribution

BLAST: output…

raw score

similarity (using matrix or scoring settings)

bit score

$S' = ((\text{matrix size} \cdot \text{score}) - \ln(\text{database size}))/\ln(2)$

E value

$E = \text{queryLength} \cdot \text{subjectLength} \cdot 2^{-S'}$

BLAST: …output

raw score

- dependent on substitution matrix

bit score

- comparable across substitution matrices
- dependent on database composition
- dependent on search settings

E value

- chance that query/subject alignment score is due to chance alone
- 0.05 could be considered statistically ‘significant’
- 0.001 is usually considered statistically ‘significant’

BLAST search types

‘program’	query input	query search	database input	database search
BLASTn	DNA	DNA	DNA	DNA
BLASTx	DNA	AA	AA	AA
tBLASTx	DNA	AA	DNA	AA
tBLASTn	AA	AA	DNA	AA
BLASTp	AA	AA	AA	AA

BLAST alternatives: USEARCH

compare query to references in a prioritized order

number of kmers shared between query and reference

count kmers that are unique within the database

use 20^k > aa database size or 4^k > nt database size

align query to reference then calculate similarity

accept if more than threshold

give up search after a set number of failures to match kmers

fast, but (relatively) insensitive

BLAST alternatives: DIAMOND

reduce amino acid alphabet to 11, filter sequence

- KREDQN|C|G|H|ILV|M|F|Y|W|P|STA

uses a two–stage search (coarse– and fine–filtering)

- different kmer sizes and numbers depending on settings
- kmers selected from a large protein family database

reduce query to non–contiguous kmers

look for sequences with matching kmers

extend matches in both directions (local alignment)

- score segments

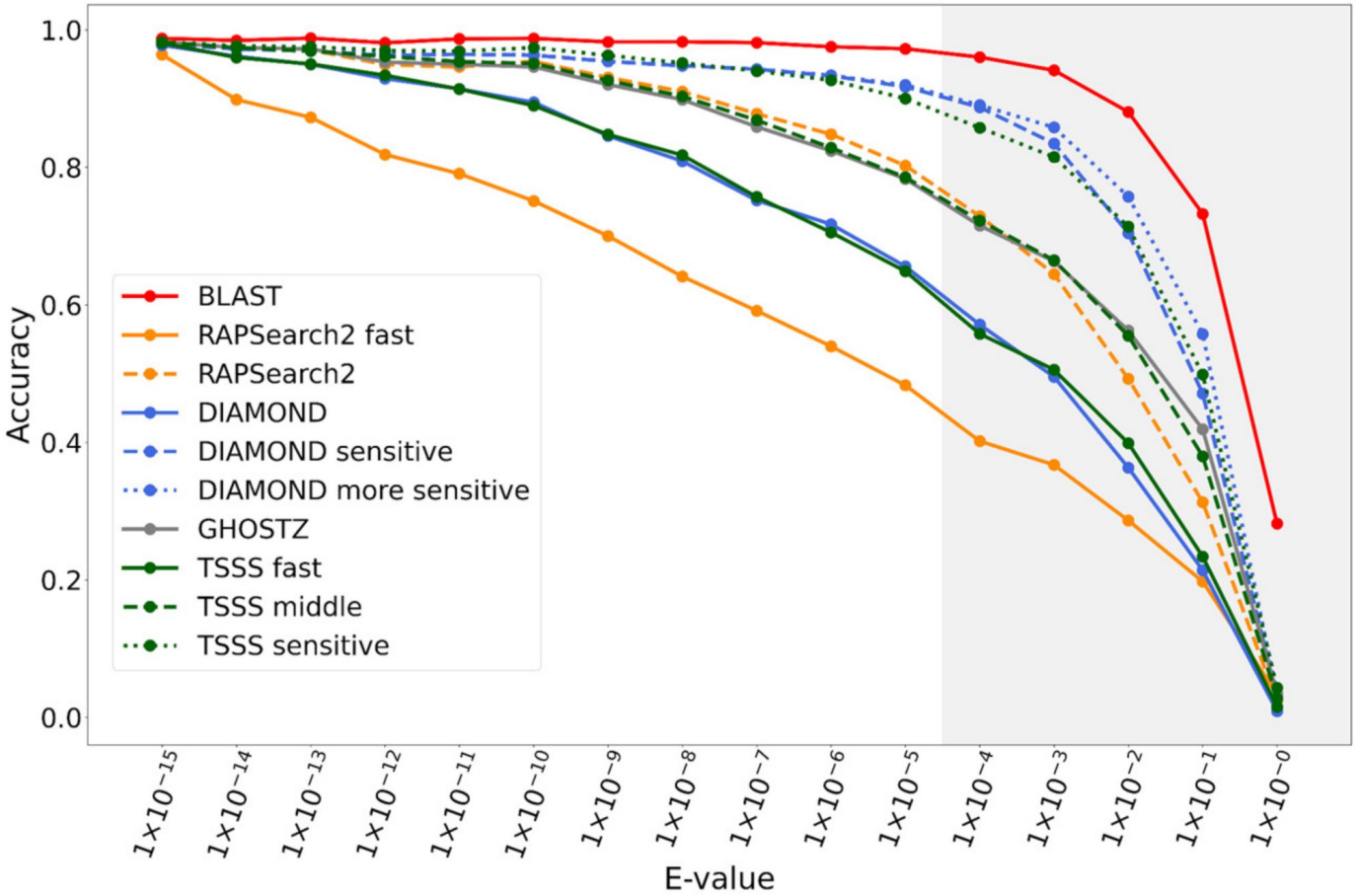


Figure 6. Accuracy of each method according to E-value.

(Takabatake et al. 2021; https://doi.org/10.3390/genes12091455)

BLAST alternatives: HMMER

query compared to hidden Markov reference alignments

20 amino acid + 2 indel parameters per alignment position

- constructed from alignments or a BLOSUM matrix
- position specific scoring is (potentially) more powerful

sequence score computed from a sample of alignments

- represents uncertainty in alignment

Markov Chain (a.k.a. MC)

a ‘random walk’ approach to estimating solutions

i.e. evaluate many related near optimal solutions

assuming:

sooner or later an ‘optimal’ solution will be found

near optimal solutions are frequent

suboptimal solutions are less frequent

the ‘average’ of suboptimal solutions is useful

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

I. INTRODUCTION

THE purpose of this paper is to describe a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. Classical statistics is assumed, only two-body forces are considered, and the potential field of a molecule is assumed spherically symmetric. These are the usual assumptions made in theories of liquids. Subject to the above assumptions, the method is not restricted to any range of temperature or density. This paper will also present results of a preliminary two-dimensional calculation for the rigid-sphere system. Work on the two-dimensional case with a Lennard-Jones potential is in progress and will be reported in a later paper. Also, the problem in three dimensions is being investigated.

* Now at the Radiation Laboratory of the University of California, Livermore, California.

II. THE GENERAL METHOD FOR AN ARBITRARY POTENTIAL BETWEEN THE PARTICLES

In order to reduce the problem to a feasible size for numerical work, we can, of course, consider only a finite number of particles. This number N may be as high as several hundred. Our system consists of a square† containing N particles. In order to minimize the surface effects we suppose the complete substance to be periodic, consisting of many such squares, each square containing N particles in the same configuration. Thus we define d_{AB}, the minimum distance between particles A and B, as the shortest distance between A and any of the particles B, of which there is one in each of the squares which comprise the complete substance. If we have a potential which falls off rapidly with distance, there will be at most one of the distances AB which can make a substantial contribution; hence we need consider only the minimum distance d_{AB}.

† We will use the two-dimensional nomenclature here since it is easier to visualize. The extension to three dimensions is obvious.

(Metropolis et al. 1953; https://doi.org/10.1063/1.1699114)

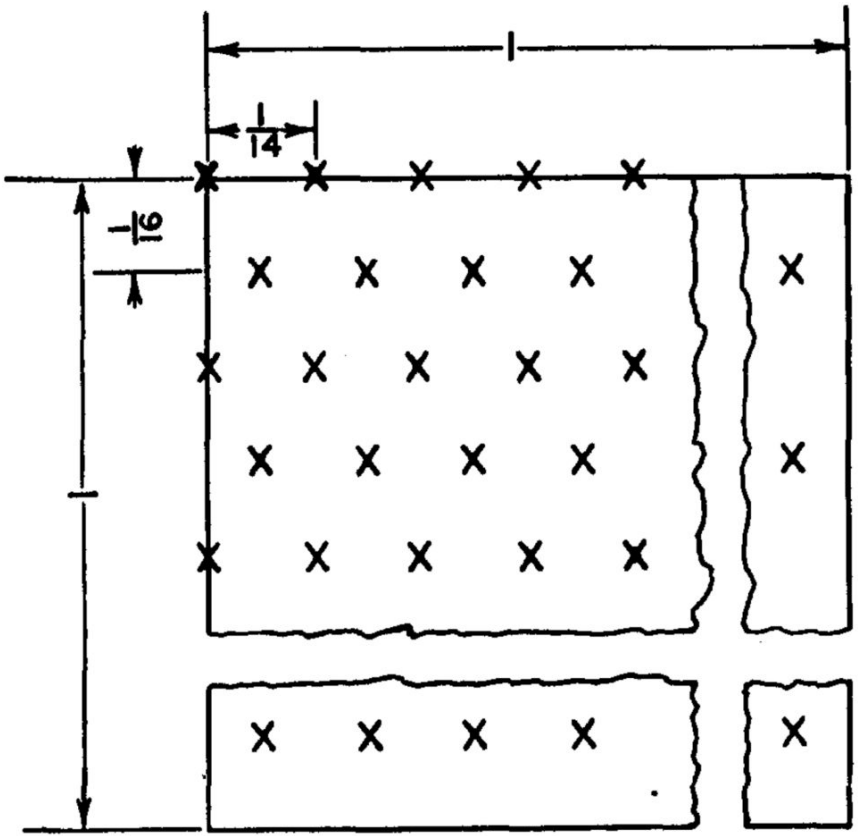


FIG. 2. Initial trigonal lattice.

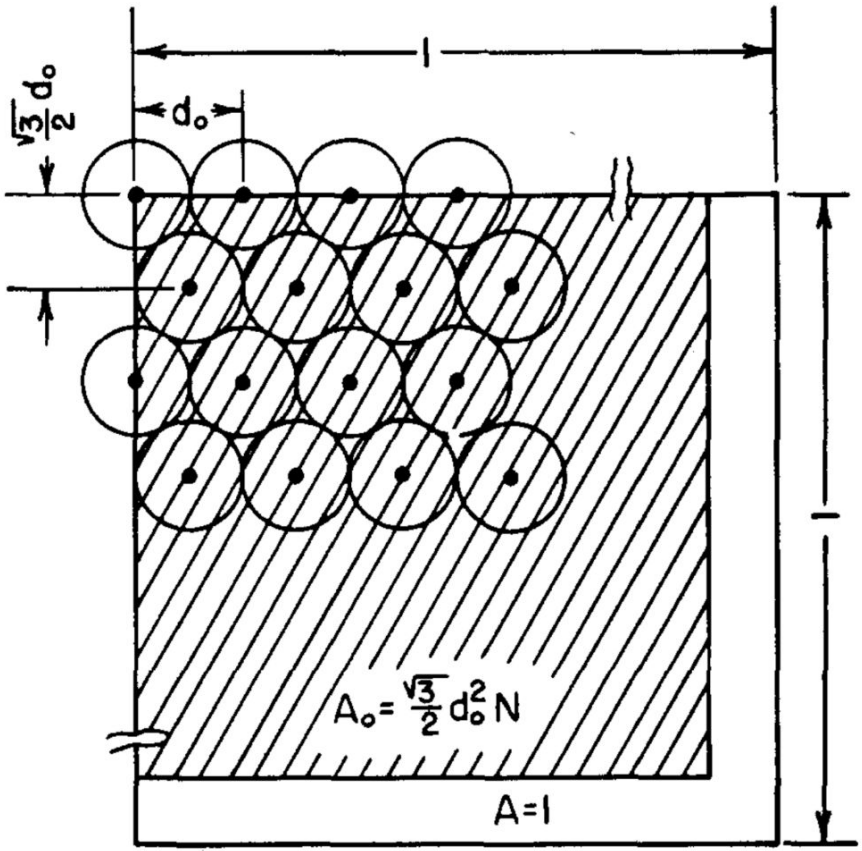


FIG. 3. The close-packed arrangement for determining A_0.

(Metropolis et al. 1953; https://doi.org/10.1063/1.1699114)

Metropolis et al. (1953)…

generate a configuration; calculate the score

randomly perturb the configuration; calculate the score

if new score is better than old score

    => accept the new configuration

if new score is worse than old score

    pick a random number

        if random ≥ score => keep old configuration

        if random < score => accept the new configuration

…Metropolis et al. (1953)

repeat perturbation and accept/reject many times

optimal is the ‘average’ over all accepted configurations

- or a sample of accepted configurations (e.g. every xth)
- or select the best configuration(s)

Metropolis–Hastings

assume that the chain represents a valid statistical sample

although the dependency is clear

perturbation amount tuned so the chain moves forward

chain must be long enough to find (an) optimal solution(s)

the beginning of the chain may be discarded (‘burn in’)

originally called a ‘Monte Carlo method’, later called ‘Markov Chain Monte Carlo’ => MCMC a.k.a. $(MC)^2$

Metropolis Coupled Markov Chain Monte Carlo [a.k.a. $(MC)^3$]

run several simultaneous chains

one chain is ‘unheated’ [i.e. like normal $(MC)^2$]

the other chains are ‘heated’ (i.e. greater perturbation)

at a low frequency configurations are traded between chains using the standard $(MC)^2$ acceptance formula

only configurations included in the unheated chain are used in later calculations

Hidden Markov Model (HMM)

Markov chain: ‘average’ optimal configuration

- i.e. an ‘exact’ calculation from configuration samples

hidden Markov model is the inverse

- configuration is a static input (usually a sequence)
- measurement is the ‘average’ model output
- can be described as a Bayesian approach