BLAST alternatives: HMMER

uses the ‘Forward’ Hidden Markov model algorithm

pairwise alignment of query sequence and reference HMMs

- combine multiple ungapped local alignments

score directly from the resulting probabilistic alignment

- approximated (quantized) in 8 bits (0–255)
- significance scores from Gumbel extreme distribution
    - => passed to more exact alignment/score algorithm

rank scores for final output

ePCR (Schuler 1997)…

‘simulates’ PCR electronically

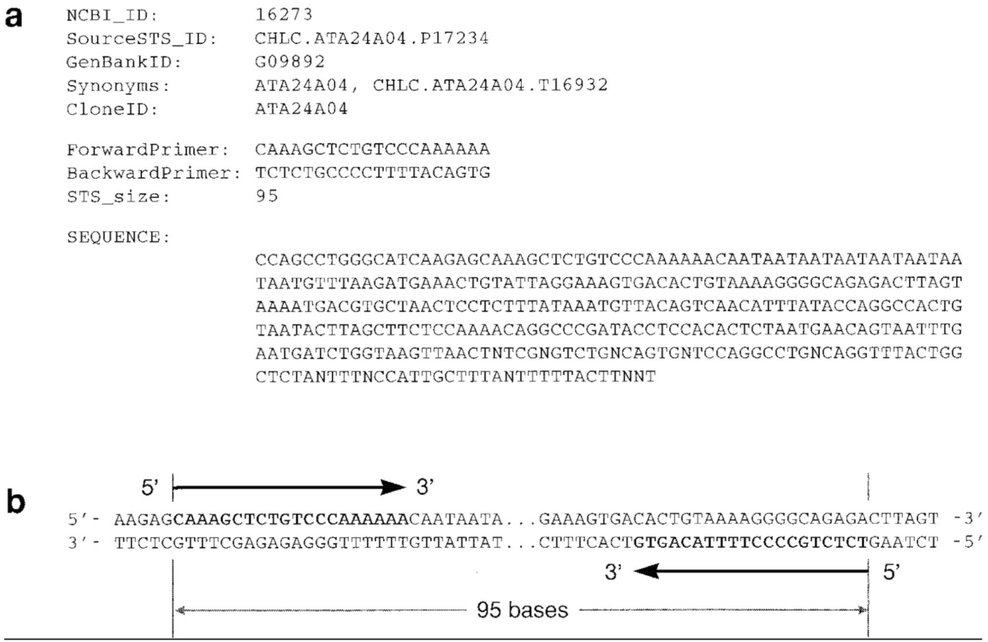
primers bind to opposite strands and face each other

- allows for some degree of primer/template mismatch
- range of distance between primers is user specified

outputs ‘amplicon’ position(s), degree of primer match

good for locating regions with conserved flanking regions

good for extracting data from poorly annotated sequence



```
a   NCBI_ID:         16273
    SourceSTS_ID:    CHLC.ATA24A04.P17234
    GenBankID:       G09892
    Synonyms:        ATA24A04, CHLC.ATA24A04.T16932
    CloneID:         ATA24A04

    ForwardPrimer:   CAAAGCTCTGTCCCAAAAAA
    BackwardPrimer:  TCTCTGCCCCTTTTACAGTG
    STS_size:        95

    SEQUENCE:
                     CCAGCCTGGGCATCAAGAGCAAAGCTCTGTCCCAAAAAACAATAATAATAATAATAATAA
                     TAATGTTTAAGATGAAACTGTATTAGGAAAGTGACACTGTAAAAGGGGCAGAGACTTAGT
                     AAAATGACGTGCTAACTCCTCTTTATAAATGTTACAGTCAACATTTATACCAGGCCACTG
                     TAATACTTAGCTTCTCCAAAACAGGCCCGATACCTCCACACTCTAATGAACAGTAATTTG
                     AATGATCTGGTAAGTTAACTNTCGNGTCTGNCAGTGNTCCAGGCCTGNCAGGTTTACTGG
                     CTCTANTTTNCCATTGCTTTANTTTTTACTTNNT
```

```
b   5'- AAGAGCAAAGCTCTGTCCCAAAAAACAATAATA...GAAAGTGACACTGTAAAAGGGGCAGAGACTTAGT -3'
    3'- TTCTCGTTTCGAGAGAGGGTTTTTTGTTATTAT...CTTTCACTGTGACATTTTCCCCGTCTCTGAATCT -5'
```

Figure 1 PCR primer sequences from a typical dbSTS record and their relationship to a query sequence that might be searched by e-PCR. (*a*) A few selected fields are shown from dbSTS record 16273 (GenBank accession no. G09892), including various names and identifiers, the sequences of the forward and reverse primers (both in 5′ → 3′ orientation), the size of the PCR product, and the sequence of the amplicon and flanking regions. (*b*) For a query sequence that is the same sense as the sequence of the dbSTS record, a successful match will include the forward primer followed by the inverse (i.e., reverse-compliment) of the reverse primer. On the other hand, if the query sequence is of the opposite sense (imagine the lower strand reversed), it will be the reverse primer followed by the inverse of the forward primer.

(Schuler 1997; https://doi.org/10.1101/gr.7.5.541)

…ePCR (Schuler 1997)…

re-PCR program is more useful than e-PCR program

works by creating a very large hash file

- ca. 3–4 times the size of the input FASTA file

optimized for ext2/ext3/ext4 file system

- does not work quickly with HFS+ or AFS (MacOS)

much faster than doing two BLAST searches

output needs to be processed with blastdbcmd or similar

freely available from NCBI (code and binary)

…ePCR (Schuler 1997)

results are dependent on the settings

- hash file creation, search settings

wildly different answers are possible

empirical settings from three primer sets (Little 2014):

- word size = 2–3 nucleotides
- discontiguous word count = 2
- indel = 0
- mismatch = 27–32% (of shortest primer)

alignment: types

pairwise: only two sequences

- useful for sequence search
- optimal solution guaranteed to be found
- alignment is, not necessarily, meaningful

multiple: more than two sequences

- the most widely used alignment type
- no guarantee that an optimal solution will be found
- alignment may be impossible (without quantum superpositioning)

alignment: local versus global

global (Needleman and Wunsch 1970)

- assumes all of the sequences are alignable
  - input = output
- alignable != homologous
- indel cost required (usually more than mismatch)

local (Smith and Waterman 1981)

- assumes parts of the sequences are alignable
  - input != output (unaligned deleted or marked)
- negative indel cost required

alignment: global

calculate differences between positions

- use substitution matrix

calculate minimum path between cells

- diagonal movements == no indel
- use indel cost for horizontal or vertical movement

find the least costly path(s) from end to start

- extract alignment

example global substitution matrix

indel (gap) cost = 1

	A	C	G	T	–
A	0	1	1	1	1
C	1	0	1	1	1
G	1	1	0	1	1
T	1	1	1	0	1
–	1	1	1	1	0

global (Needleman– Wunsch): initialization

	–	C	G	T
–	0	1	1	1
C	1	0	1	1
G	1	1	0	1
G	1	1	0	1
T	1	1	1	0

calculate differences between cells

global (Needleman– Wunsch): update

	–	C	G	T
–	0	1	1	1
C	1	0	1	1
G	1	1	0	1
G	1	1	0	0
T	1	1	1	0

```
carry+ vertical +initial
carry+ diagonal +initial
carry+horizontal+initial
```

	–	C	G	T
–	0	0+1+1	2+1+1	4+1+1
C	0+1+1	2+1+0 0+0+0 2+1+0	4+1+1 2+0+1 0+1+1	6+1+1 4+0+1 2+1+1
G	2+1+1	0+1+1 2+0+1 4+1+1	2+1+0 0+0+0 2+1+0	4+1+1 2+0+1 0+1+1
G	4+1+1	2+1+1 4+0+1 6+1+1	0+1+0 2+0+0 4+1+0	2+1+1 0+0+1 1+1+1
T	6+1+1	4+1+1 6+0+1 8+1+1	1+1+1 4+0+1 6+1+1	1+1+0 1+0+0 3+1+0

global (Needleman– Wunsch): trace back

	–	C	G	T
–	0	2	4	6
C	2	0	2	4
G	4	2	0	2
G	6	4	1	1
T	8	6	3	1

vertical
diagonal
horizontal

	–	C	G	T
–				
C		diagonal		
G			diagonal	
G			vertical	
T				diagonal

global (Needleman– Wunsch): extract alignment

	–	C	G	T
–				
C		diagonal		
G			diagonal	
G			vertical	
T				diagonal

CG–T
CGGT

alignment: local

calculate differences between positions

- use substitution matrix
    - negative penalty values, indel cost more than penalty

calculate minimum path between cells

- diagonal movements == no indel
- use indel cost for horizontal or vertical movement
- if previous cell is negative, use zero for next cell carry value

alignment = lowest cost path(s) from highest scoring cell to last diagonal element

example local substitution matrix

indel (gap) cost = -2

	A	C	G	T	–
A	1	-1	-1	-1	-1
C	-1	1	-1	-1	-1
G	-1	-1	1	-1	-1
T	-1	-1	-1	1	-1
–	-1	-1	-1	-1	1

local (Smith–Waterman): initialization

	–	C	G	C	T
–	0*	-1	-1	-1	-1
T	-1	-1	-1	-1	1
G	-1	-1	1	-1	-1
C	-1	1	-1	1	-1
A	-1	-1	-1	-1	-1

calculate differences between cells

local (Smith–Waterman): update

```
carry+ vertical +initial
carry+ diagonal +initial
carry+horizontal+initial
```

	–	C	G	C	T
–	0*	-1	-1	-1	-1
T	-1	-1	-1	-1	1
G	-1	-1	1	-1	-1
C	-1	1	-1	1	-1
A	-1	-1	-1	-1	-1

	–	C	G	C	T
–	0*	0-2-1	0-2-1	0-2-1	0-2-1
C	0-2-1	0-2-1 0+0-1 0-2-1	0-2-1 0+0-1 0-2-1	0-2-1 0+0-1 0-2-1	0-2+1 0+0+1 0-2+1
G	0-2-1	0-2-1 0+0-1 0-2-1	0-2+1 0+0+1 0-2+1	0-2-1 0+0-1 1-2-1	1-2-1 0+0-1 0-2-1
C	0-2-1	0-2+1 0+0+1 0-2+1	1-2-1 0+0-1 1-2-1	0-2+1 1+0+1 0-2+1	0-2-1 0+0-1 2-2-1
A	0-2-1	1-2-1 0+0-1 0-2-1	0-2-1 1+0-1 0-2-1	2-2-1 0+0-1 0-2-1	0-2-1 2+0-1 0-2-1

local (Smith–Waterman): path formation

	–	C	G	C	T
–	0*	0	0	0	0
C	0	0	0	0	1
G	0	0	1	0	0
C	0	1	0	2	0 0
A	0	0	0	0 0	1

vertical
diagonal
horizontal

	–	C	G	C	T
–	null	0	0	0	0
T	0	0	0	0	null
G	0	0	end	0	0
C	0	null	0	start	0
A	0	0	0	0	1

local (Smith–Waterman): extract alignment

	–	C	G	C	T
–	null	0	0	0	0
T	0	0	0	0	null
G	0	0	end	0	0
C	0	null	0	start	0
A	0	0	0	0	1

GC
GC

alignment: Which alignment is best?

final purpose of the alignment matters

e.g. phylogeny versus primer design

not simply a question of similarity versus homology

similarity = number of positions that are the same

homology = similarity due to common ancestry

the ‘true’ alignment cannot be known

estimated by reconstructing mutational events

the ‘wrong’ alignment may be more useful in many cases

alignment: Which alignment is best?

most (all?) alignment programs tested against BAliBASE

- Benchmark Alignment dataBASE
- http://www-bio3d-igbmc.u-strasbg.fr/balibase/
- a collection of 'correct' alignments
    - secondary structure based with intuitive adjustment
        - i.e. alignments that 'look' right (to someone)
    - ultimate alignment purpose is unstated
    - no guarantee that they are correct
        - many appear to be incorrect (Edgar 2010)

alignment: objective functions...

COFFEE (Notredame et al. 1998)

- measures column–by–column similarity
    - between pairwise and multiple sequence alignment
- assumes that the pairwise alignments are optimal
    - assumes a set of (arbitrary) costs
    - assumes that similarity reflects history
- does not necessarily lead to a consistent alignment

alignment: …objective functions...

Transitive Consistency Score (TCS; Chang et al., 2014)

- a rescaled extension to COFFEE
- measures column–by–column similarity
    - multiple sequence alignment against a collection of alignments
- assumes that the most common alignments are optimal
    - assumes a set of (arbitrary) costs and that similarity reflects history
- does not necessarily lead to a consistent alignment
- can be used to weight alignment quality for phylogenetic calculations
    - ‘better’ than GUIDANCE, Gblocks, trimAl

alignment: …objective functions...

sum of pairs (most commonly used)

- sum of pairwise distance between all sequences
- attempts to minimize differences in the alignment
- assumes a set of (arbitrary) costs
- assumes that similarity reflects history
- does not necessarily lead to a consistent alignment

sum of pairs

0	A	-	-	A
1	A	T	-	A
2	A	-	C	A
3	A	C	C	A

	0	1	2	3
0	0	1	1	2
1	1	0	2	2
2	1	2	0	1
3	2	2	1	0

sum = 9 (1+1+2+2+2+1)

sum of pairs

0	A	-	-	A
1	A	-	T	A
2	A	-	C	A
3	A	C	C	A

	0	1	2	3
0	0	1	1	2
1	1	0	1	2
2	1	1	0	1
3	2	2	1	0

sum = 8 (1+1+2+1+2+1)

alignment: …objective functions...

GLOCSA (Arenas Diaz et al. 2009)

- minimization of implied evolutionary steps
- additional features of the alignment to distinguish between otherwise equivalent alignments
    - mean column heterogeneity
    - distribution of indels
    - alignment size

good for phylogeny and alignment

alignment: …objective functions

POY (Varón et al. 2010)

- minimization of reconstructed evolutionary steps
    - optimal phylogeny and alignment
    - (parsimony or maximum likelihood)
- can violate the triangle inequality
- (sometimes) good for phylogeny, but not for alignment

BALi-Phy (Redelings, 2021)

- the ‘Bayesian’ analoge of POY

alignment: MUSCLE (Edgar 2004a,b)

[0] kmer distance estimation for unaligned sequences

[1] distance (UPGMA) guide tree generated

[2] pairwise global alignment down tree

    [a] consensus (profile) constructed

    [b] insertions propagated up tree

[3] K2P distances calculated

[4] back to [1] (once)

[5] pairwise global alignment down tree (like [2])

    => sum of pairs used to accept/reject realignment

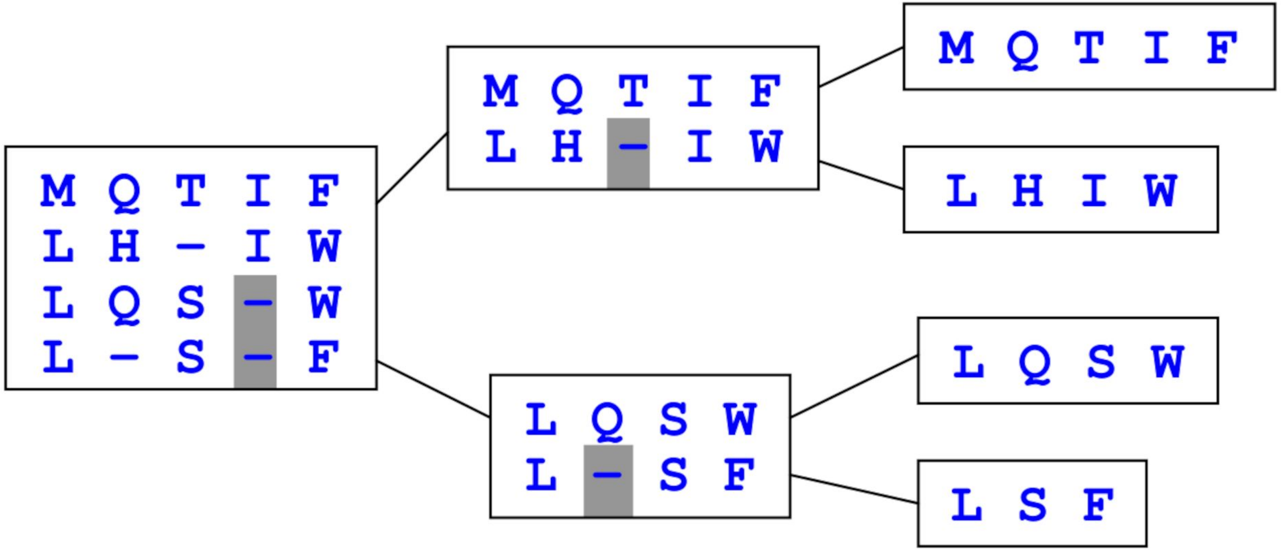
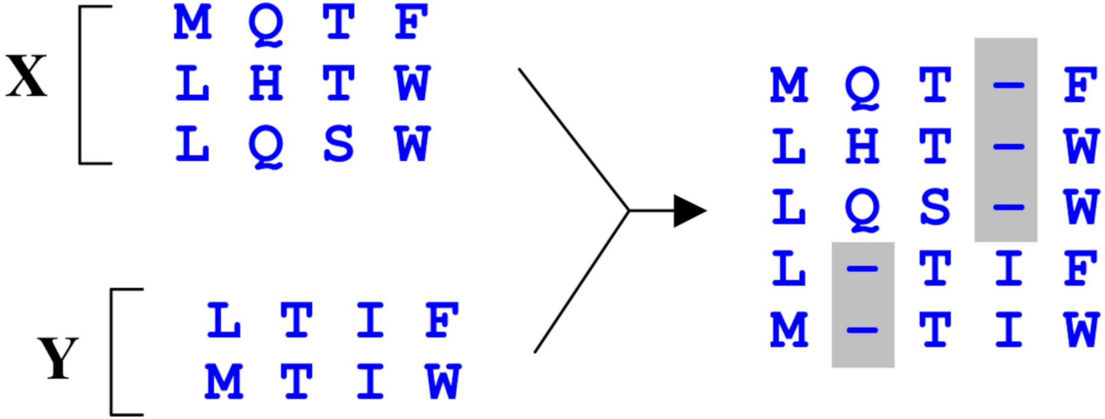


Figure 1
Progressive alignment. Sequences are assigned to the leaves of a binary tree. At each internal (i.e., non-leaf) node, the two child profiles are aligned using profile-profile alignment (see Figure 2). Indels introduced at each node are indicated by shaded background.

(Edgar 2004; https://doi.org/10.1186/1471-2105-5-113)



```
X
M Q T F
L H T W
L Q S W

Y
L T I F
M T I W

M Q T - F
L H T - W
L Q S - W
L - T I F
M - T I W
```

Figure 2

Profile-profile alignment. Two profiles (multiple sequence alignments) X and Y are aligned to each other such that columns from X and Y are preserved in the result. Columns of indels (gray background) are inserted as needed in order to align the columns to each other. The score for aligning a pair of columns is determined by the profile function, which should assign a high score to pairs of columns containing similar amino acids.

(Edgar 2004; https://doi.org/10.1186/1471-2105-5-113)

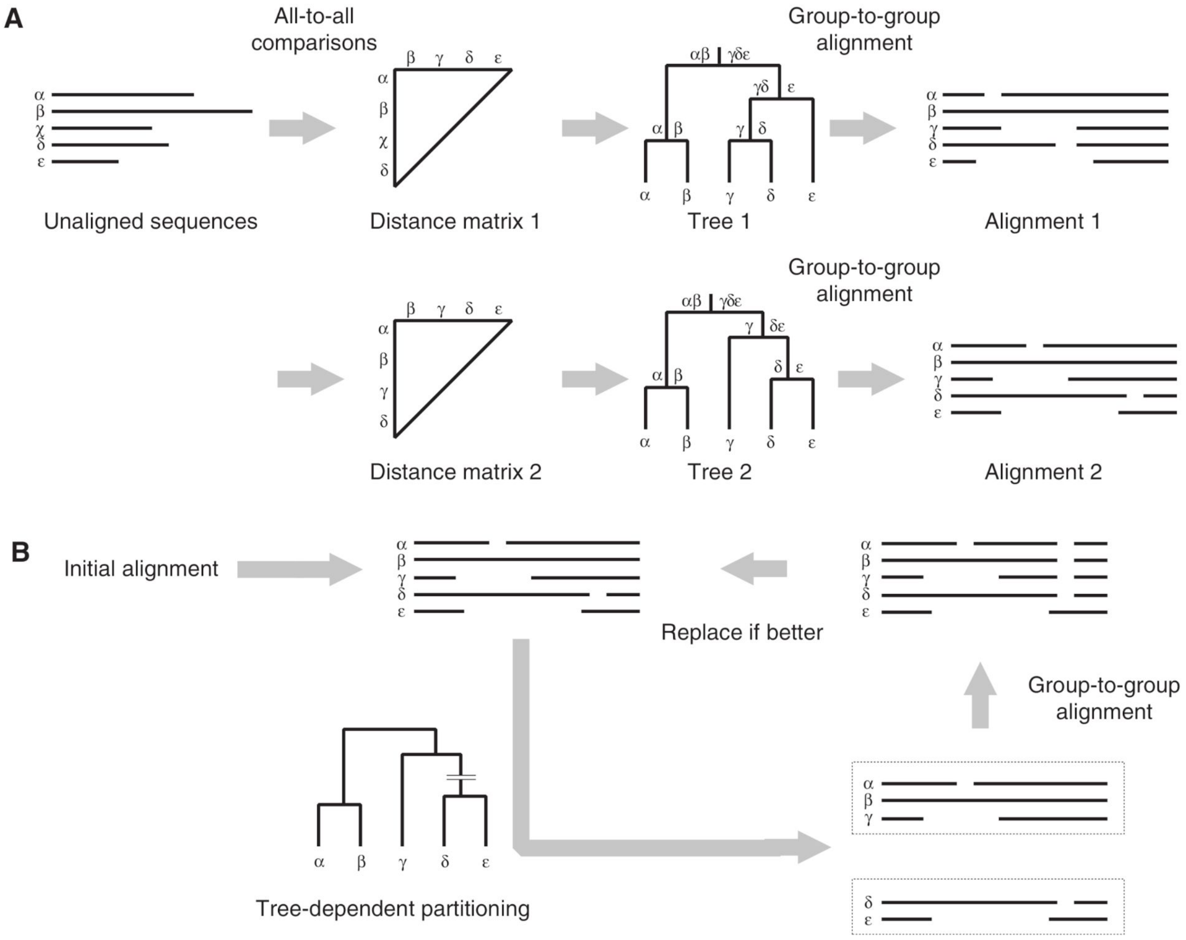


Figure I: Calculation procedures of the progressive method (**A**) and the iterative refinement method (**B**).

(Katoh & Toh 2008; https://doi.org/10.1186/1471-2105-5-113)

alignment: MAFFT (Katoh & Toh 2008)

- (too) many different algorithms available
  - uses variants of sum of pairs or COFFEE scoring
- can use local or global alignment
- can use structural pairwise alignments
  - good for low similarity sequences
- can insert sequences into a skeletal alignment
- ‘program’ is really a large shell script that dispatches to a variety of special purpose programs
  - restricts access to some algorithms by alignment size
    - can be overridden by modifying the shell script

alignment: NAST (DeSantis et al. 2006; Caporaso et al. 2010)

Nearest Alignment Space Termination (NAST)

builds a multiple sequence alignment from a template

- for each new sequence:
  - BLAST (etc.) to find most similar template sequence
  - pairwise alignment of template and new sequence
  - insert into template without introducing insertions
    - can cause local mis–alignments (or worse)

primarily used for identification (DNA barcoding, etc.)

- other better options (i.e. identification algorithms, MAFFT)

alignment: translatorX (Abascal et al. 2010)

[1] translates nucleotides to amino acids (standard tables)

[2] aligns amino acids using an external program

- can be manually edited
- can be aligned using an ‘unsupported’ program

[3] reverse translates back to the original nucleotides

- removes incomplete codons from the ends
- has difficulty with long strings of ambiguous nucleotides
- useful for difficult to align coding regions

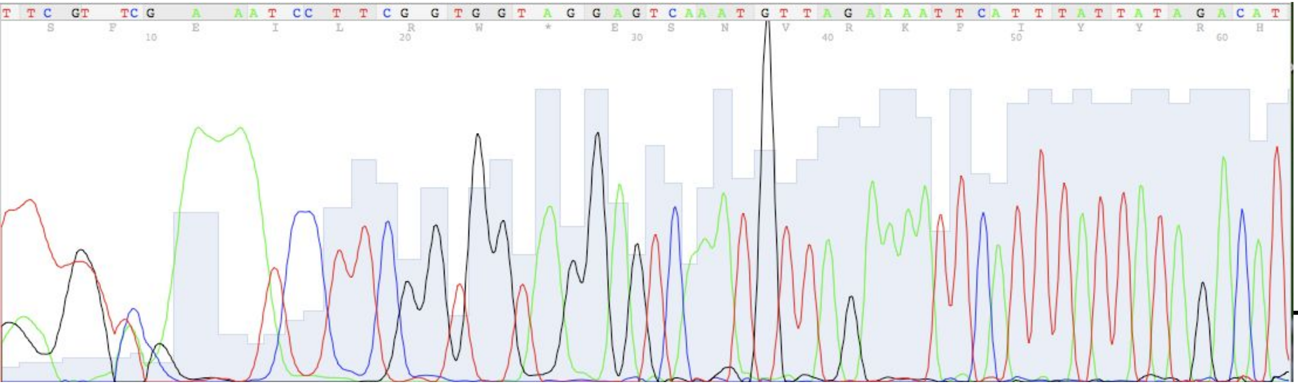
sequence quality

base–by–base error probability for base–calling programs

reflects assay bias (e.g. detection chemistry, algorithms)

allows for more efficient sequence editing and assembly

- allows for ‘poorly supervised’ automation



sequence quality: PHRED

calculates probabilities using a local window

able to distinguish between ‘good’ and ‘bad’ regions

  not able to distinguish overall ‘good’ from ‘bad’

outputs log probabilities

  e.g. $q = -10 \cdot \log_{10}(p)$ [$p = 0.001$; $q = 30$]

predicts quality by measuring peak properties

similar to linear discriminant analysis

  without assumption of normality (data are not normal)

sequence quality: Illumina base calling

model–based:

AYB (Massingham and Goldman 2012), Bustard (Illumina default), BayesCall (Kao and Song 2009), naiveBayescall (Kao and Song 2011), Onlinecall (Das and Vikalo 2012), Rolexa (Ledergerber and Dessimoz 2011), Softy (Das and Vikalo 2013), Swift (Whiteford et al. 2009), etc.

(supervised) machine learning:

Altacyclic (Erlich et al. 2008), freeIbis (Renaud et al. 2013), Ibis (Kircher et al. 2009), Optocoder (Senel et al. 2022), etc.

sequence quality: Illumina base calling

important parameters:

- cross–talk among dyes
- phasing (i.e. secondary signals) as a function of cycle
- signal decay as a function of cycle
- intensity of the previous cycle
- intensity of the current cycle
- intensity of the next cycle

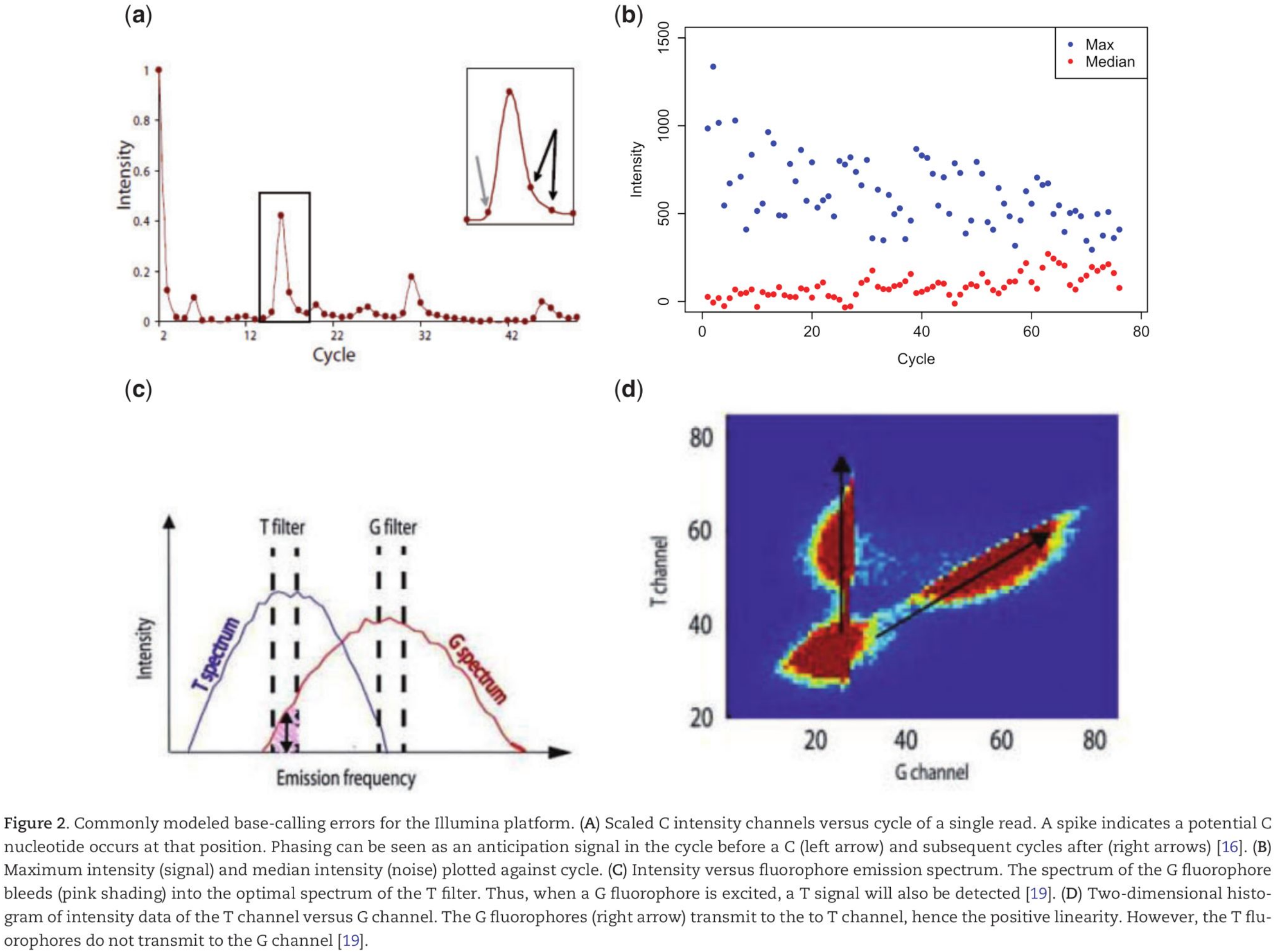


Figure 2. Commonly modeled base-calling errors for the Illumina platform. (A) Scaled C intensity channels versus cycle of a single read. A spike indicates a potential C nucleotide occurs at that position. Phasing can be seen as an anticipation signal in the cycle before a C (left arrow) and subsequent cycles after (right arrows) [16]. (B) Maximum intensity (signal) and median intensity (noise) plotted against cycle. (C) Intensity versus fluorophore emission spectrum. The spectrum of the G fluorophore bleeds (pink shading) into the optimal spectrum of the T filter. Thus, when a G fluorophore is excited, a T signal will also be detected [19]. (D) Two-dimensional histogram of intensity data of the T channel versus G channel. The G fluorophores (right arrow) transmit to the to T channel, hence the positive linearity. However, the T fluorophores do not transmit to the G channel [19].

(Cacho et al. 2016; https://doi.org/10.1093/bib/bbv088)