Genome assembly and annotation

GastroPak Microbial Bioinformatics Workshop

DR CHRISTOPHER QUINCE

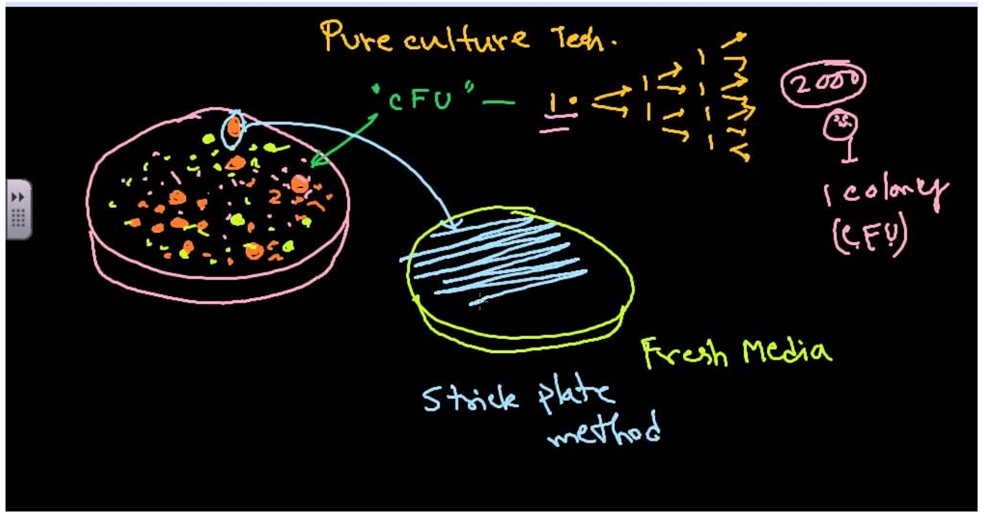
Earlham/Quadram Group Leader

Overview

- Brief introduction to microbial genome assembly
- Assembly evaluation
- Microbial genome annotation

Obtaining samples for genome sequencing

- Microbial genome sequencing usually proceeds from cultured isolates
- Streak or spread plate method
- We typically know what species should be present as a single homogenous strain
- Good practice to check with Sanger sequencing of 16S rRNA
- Need to consider DNA extraction protocol but typically can obtain high quality DNA
- Can also perform 'sweep metagenomics' by sequencing whole plate prior to isolation
- Multiple strains same species



Earlham Institute

Single-cell microbial genome sequencing

- FACs sorting of individual cells
- Sequencing following multiple displacement amplification (MDA)

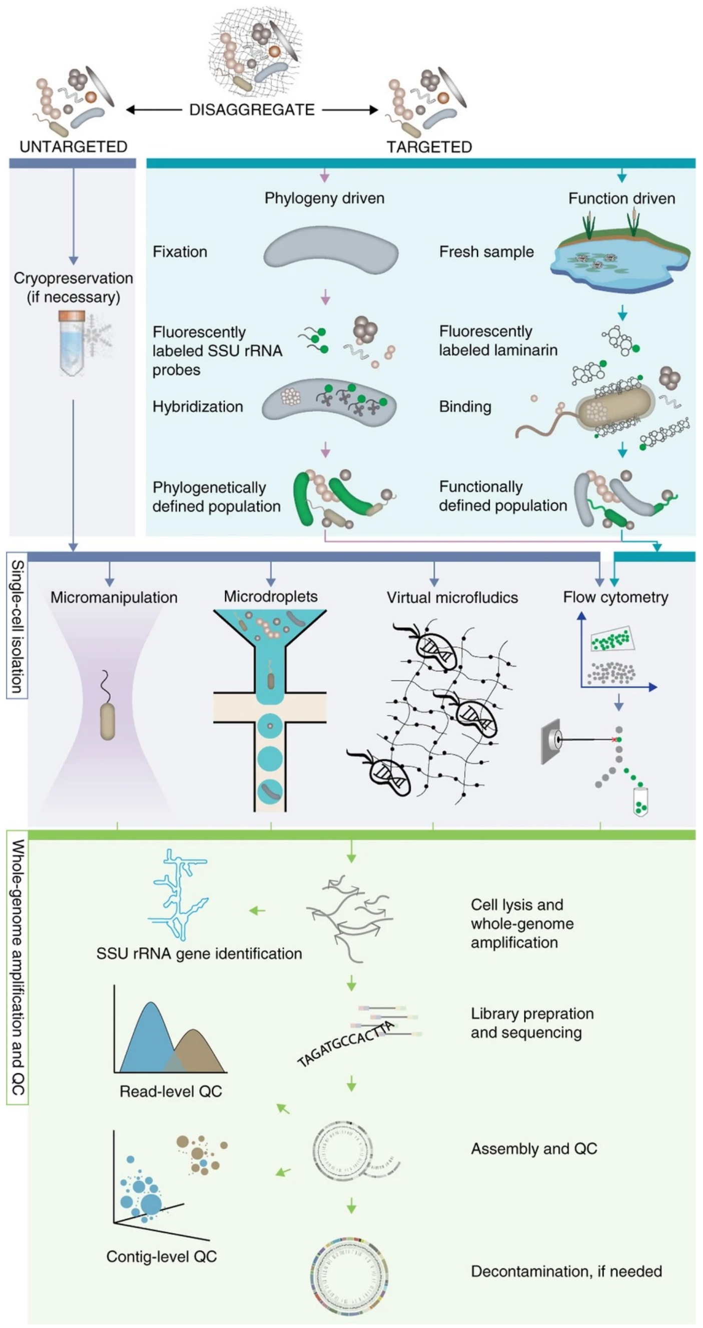
The trajectory of microbial single-cell sequencing

Tanja Woyke, Devin F R Doud & Frederik Schulz

Nature Methods **14**, 1045–1054 (2017) | Cite this article

9091 Accesses | **90** Citations | **57** Altmetric | Metrics

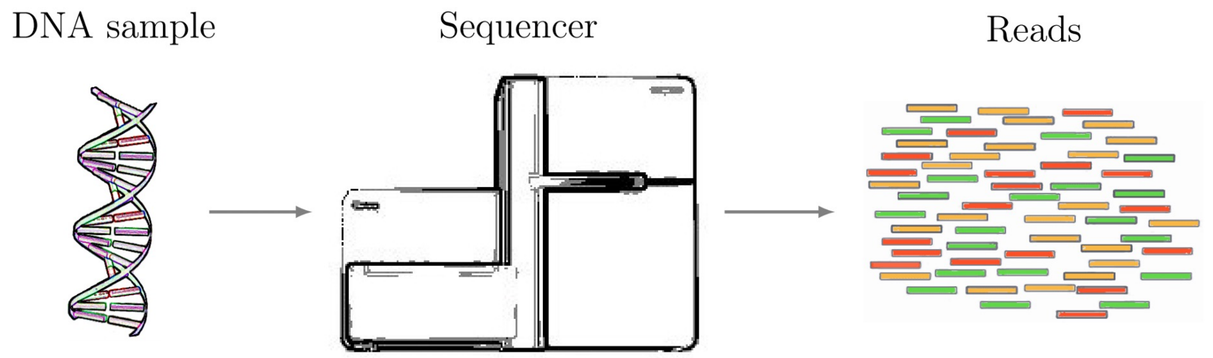
https://www.nature.com/articles/nmeth.4469/



UNTARGETED
DISAGGREGATE
TARGETED
Phylogeny driven
Function driven
Cryopreservation (if necessary)
Fixation
Fresh sample
Fluorescently labeled SSU rRNA probes
Fluorescently labeled laminarin
Hybridization
Binding
Phylogenetically defined population
Functionally defined population
Single-cell isolation
Micromanipulation
Microdroplets
Virtual microfludics
Flow cytometry
Whole-genome amplification and QC
SSU rRNA gene identification
Cell lysis and whole-genome amplification
TAGATGCCACTTA
Read-level QC
Library prepration and sequencing
Assembly and QC
Contig-level QC
Decontamination, if needed

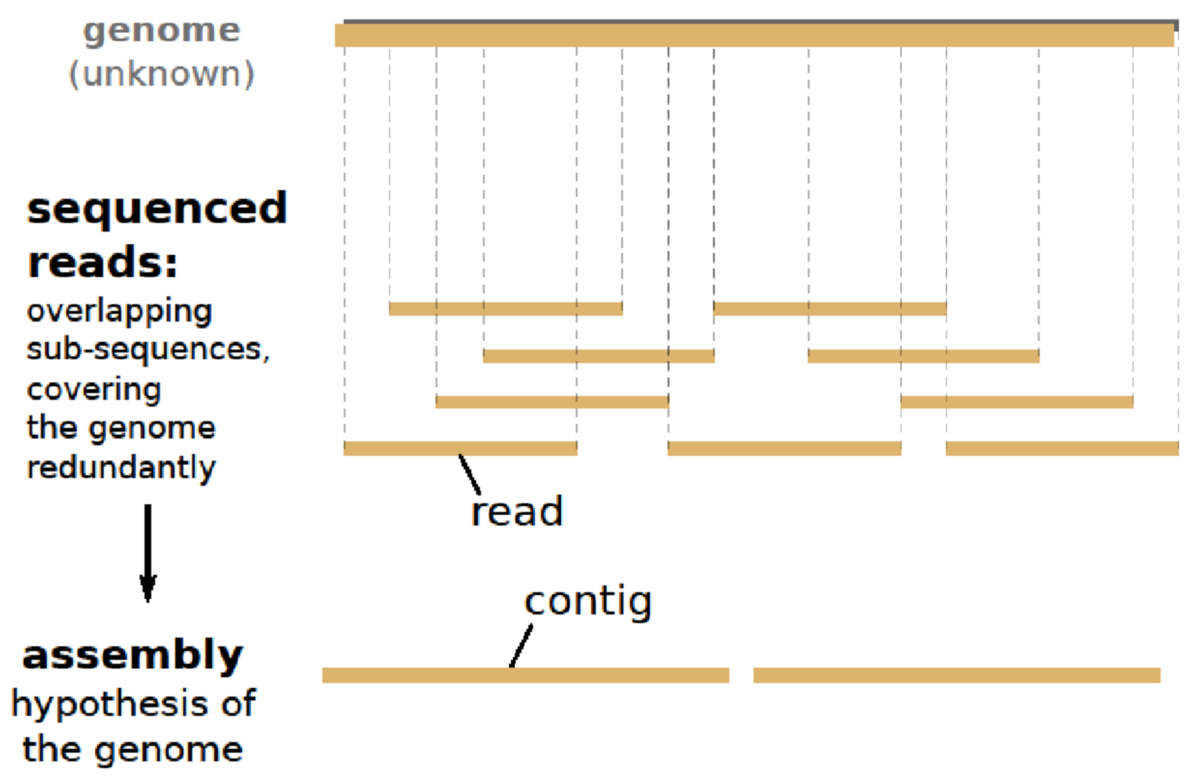
The genome assembly problem

Shotgun genomics DNA is physically broken and replicated into millions of small fragments



Assembly methods

- Overlap-Layout-Consensus
  - Overlap/String graph
- de Bruijn graph



Credit: R. Chikhi

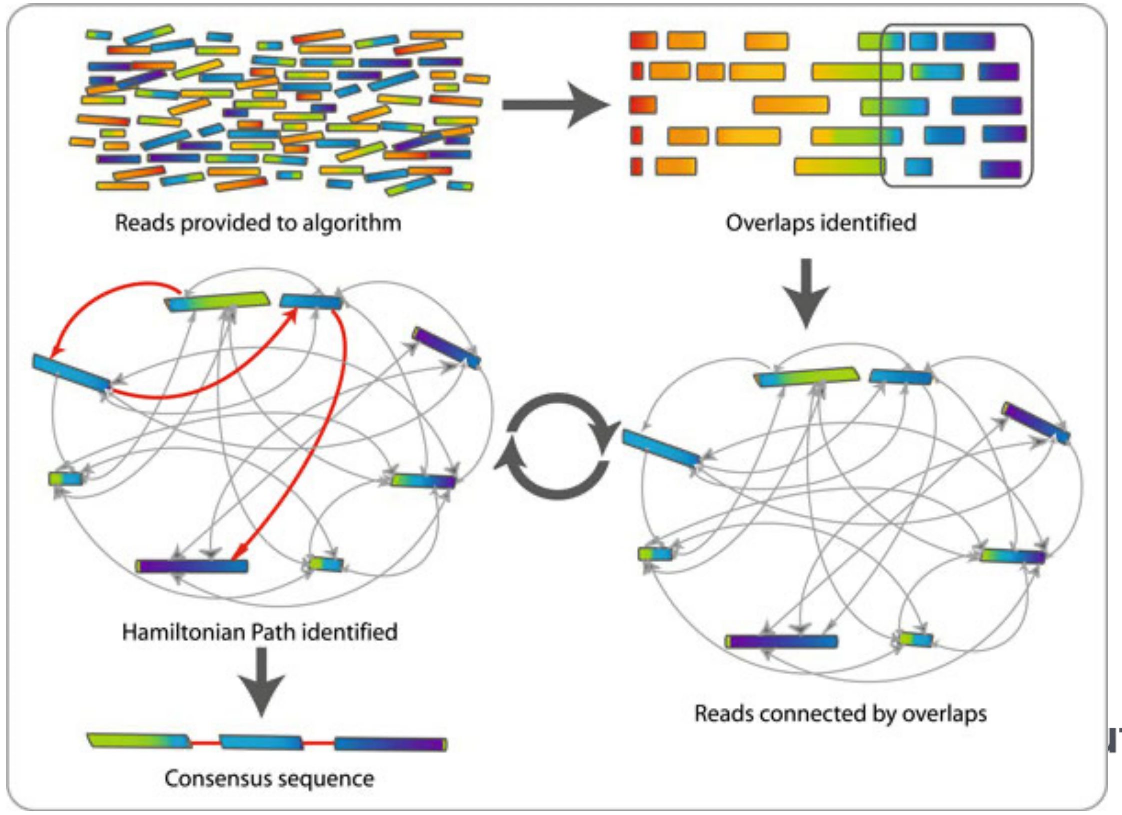
Earlham Institute

Short-read metagenomics assembly

- Assembly consists of looking for overlaps between short reads

Read 1: ACTCGTCGTG
Read 2: GTGCGATTCC → ACTCGTCGTG / GTGCGATTCC → Contig: ACTCGTCGTGCGATTCC

- Two methods for detecting read overlaps, Overlap Layout Consensus (OLC) and de Bruijn graph assemblers
- OLC is more intuitive:
  - Overlap: reads are compared to one another to identify overlapping regions (MinHash)
  - Layout: reads are positioned based on overlaps
  - Create graph with nodes (reads) and edges (overlaps)
  - Hamiltonian path visits every vertex once
  - Consensus: multiple sequence alignment is carried out to produce a consensus sequence
- Scales badly with read number – why?


Reads provided to algorithm
Overlaps identified
Hamiltonian Path identified
Reads connected by overlaps
Consensus sequence

Detecting sequence overlaps

- Requires pairwise comparisons scales as N^2
- Comparison requires alignment of two reads
- Why is alignment needed?
- This is computationally expensive (see below)
- Speed up with minimizer methods

```
ACTGGGCAATGACTC
ATTGCTCACCATCCG
```

```
ACTGGGCAATGACTC
       ATTG-CTCACCATCCG
```

Edit distance two
One overlap one substitution
Overlap 8 bp

Earlham Institute

K-mers

- k-mers are substrings of length k contained within a biological sequence e.g. trimers
- Any sequence of length L will contain L - k + 1 *k-mers*
- How many unique k=4 tetramers are there?

```
Sequence: ATCGATCAC
3-mer #0: ATC
3-mer #1:  TCG
3-mer #2:   CGA
3-mer #3:    GAT
3-mer #4:     ATC
3-mer #5:      TCA
3-mer #6:       CAC
```

https://bioinfologics.github.io/post/2018/09/17/k-mer-counting-part-i-introduction/ Bernardo Clavijo

K-mers (continued)

- Often convenient to consider ‘canonical’ kmers where a kmer and its reverse complement are equivalent e.g. ACTG == CAGT
- This is because a DNA sample and hence sequenced reads (almost) always contains sequence from both strands
- How many k = 3 and k = 4 canonical kmers are there?
- Oten use odd values of k to avoid palindromic k-mers
- Why are k-mers useful:
  - Enable efficient sequence comparison without alignment
  - Scalable short read assembly

```
Sequence: ATCGATCAC
3-mer #0: ATC
3-mer #1:  TCG
3-mer #2:   CGA
3-mer #3:    GAT
3-mer #4:     ATC
3-mer #5:      TCA
3-mer #6:       CAC
```

Offset	0	1	2	3	4	5	6
3-mer	ATC	TCG	CGA	GAT	ATC	TCA	CAC
Reverse Complement	GAT	CGA	TCG	ATC	GAT	TGA	GTG
Canonical	ATC	CGA	CGA	ATC	ATC	TCA	CAC

Canonical k-mer	**Total count**
ATC (incl. GAT)	3
CGA (incl. TCG)	2
TCA	1
CAC	1
Sum	**7**

de Bruijn graph assembly

Steps:

1. Identify kmers in reads
2. Take all (k-1)-mers from the set of k-mers, e.g. ATG, TGC-> AT, TG, GC

3. Construct a multi-graph with nodes being k-1-mers; draw an edge between two k-1 mers only if the two k-1 mers are taken from the same read. **Ex:** GCT & CTA

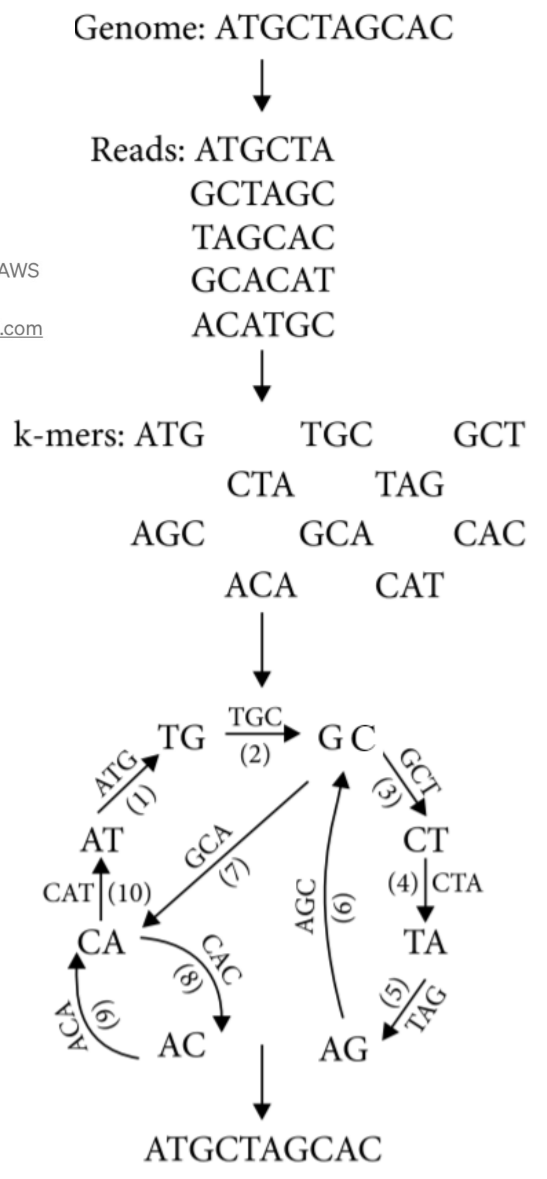
4. Graph constructed this way is guaranteed to have a Eulerian trail, follow the trail and connect the nodes to form our original sequence

- Efficient storage space
  - fixed word length
  - Independent of sequencing depth
- How to choose k? -> iterative kmers

Uditha Maduranga

1.1K Followers

Computer Science and Engineering Graduate | AWS Community Builder | Former Intern — Software Engineer at Sysco LABS | maduranga95@gmail.com

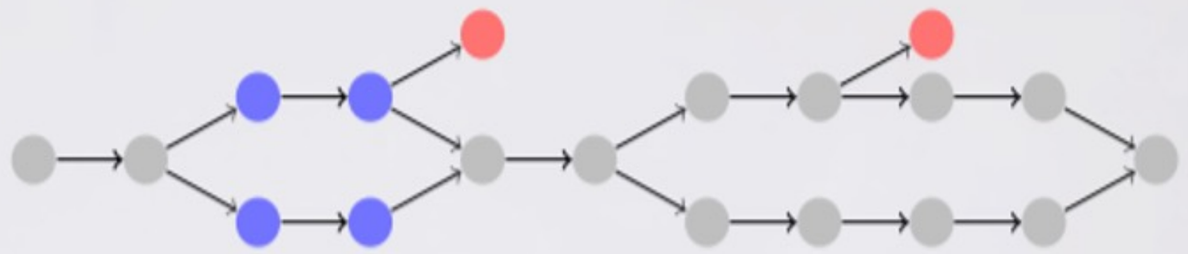


Earlham Institute

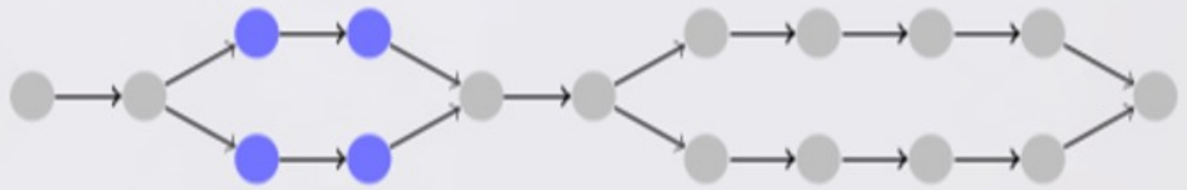
https://medium.com/towards-data-science/genome-assembly-using-de-bruijn-graphs-69570efcc270

Short read assemblers

1) de Bruijn **graph** construction



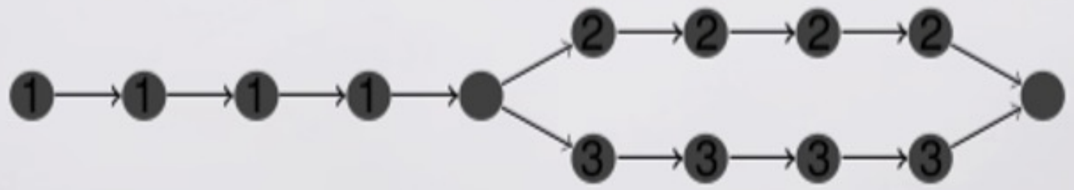
2) Likely sequencing errors are removed.



3) Variations (e.g. SNPs, similar repetitions) are removed.

→ **Collapses strains**

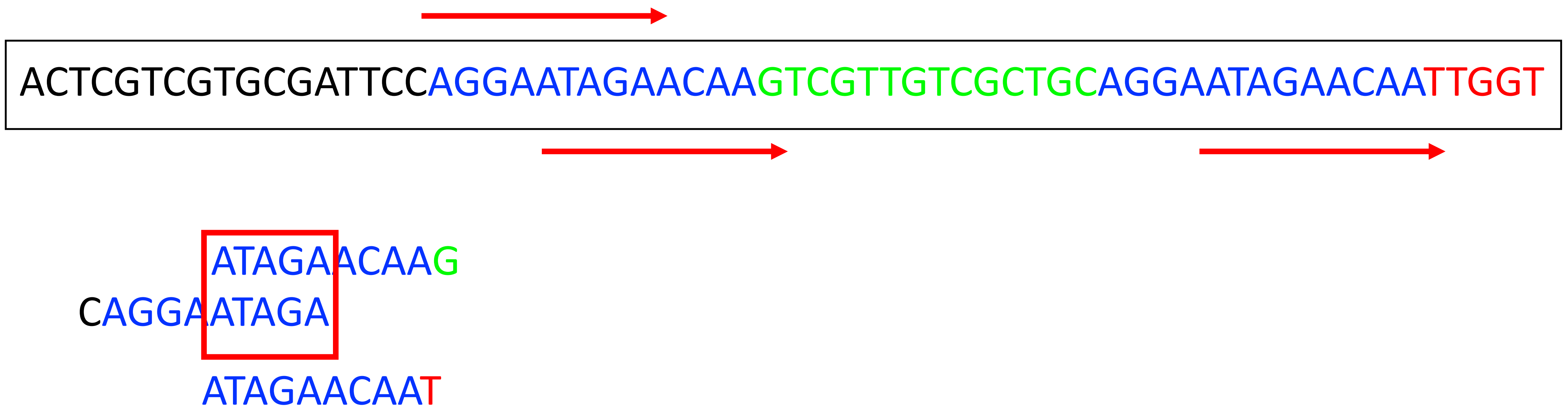
4) **Simple paths** (i.e. contigs) are returned.



5) Extra steps: repeat-resolving, scaffolding

Genome repeats

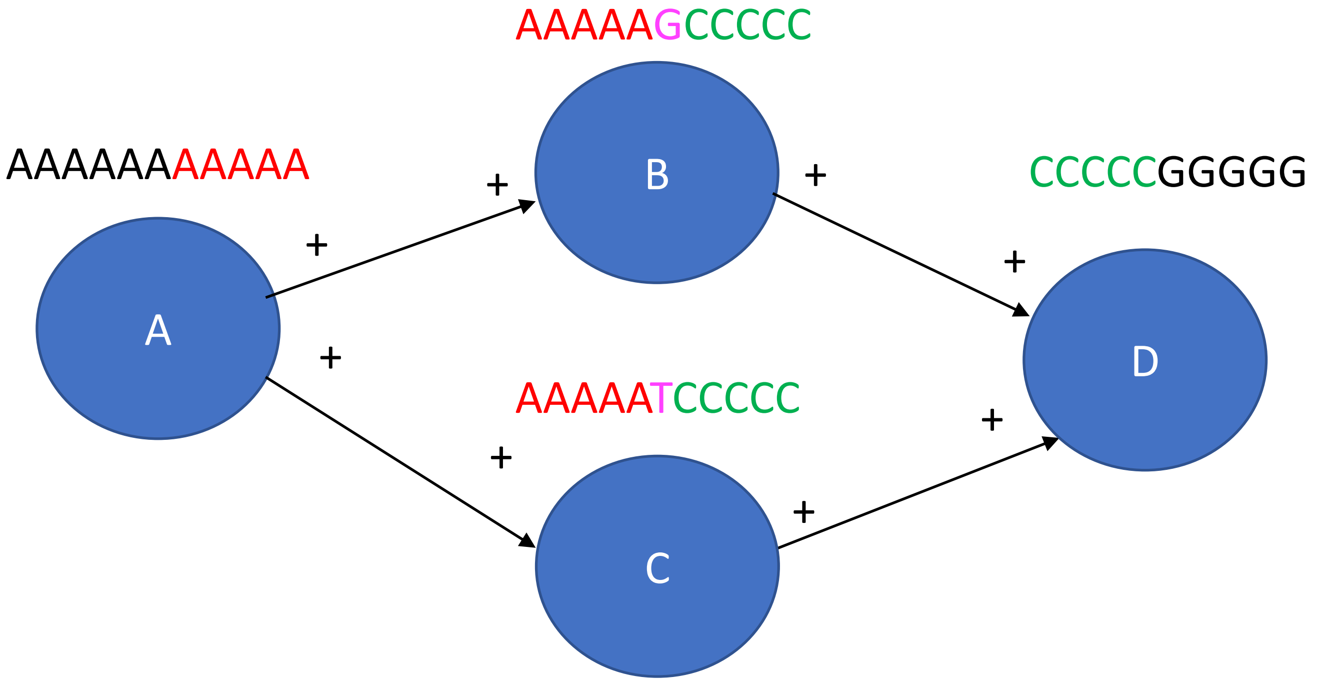
- Difficulties arise because of ‘repeat’ regions on genomes that exceed read length


ACTCGTCGTGCGATTCCAGGAATAGAACAAGTCGTTGTCGCTGCAGGAATAGAACAATTGGT
ATAGAACAAG
CAGGAATAGA
ATAGAACAAT

Assembly graph represents uncertainty

- Produced by all assemblers following compaction (de Bruijn graphs) or removal of transitive reads (string graphs)
- Nodes represent sequence (unitigs) and edges overlaps
- Represents fundamental uncertainty in possible paths and hence sequences

A->B->C = AAAAAAAAAAGCCCCCGGGGGG
A->C->D = AAAAAAAAAATCCCCCGGGGGG



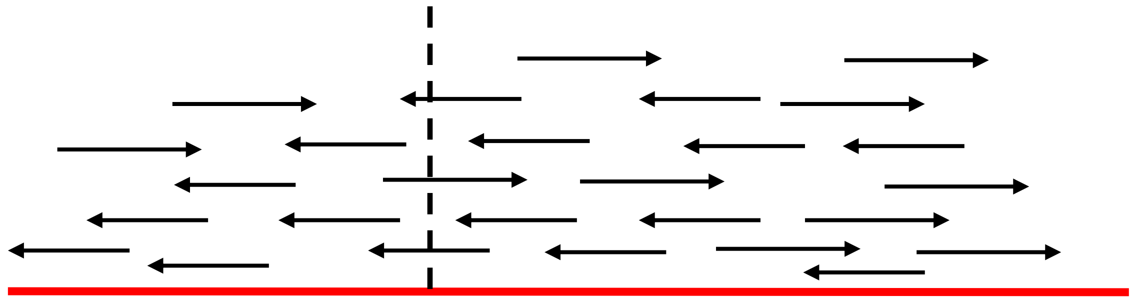
Earlham Institute

Assembly evaluation

- Evaluate an assembly by looking at the following statistics:
  - The size of the assembly: sum of contig lengths
  - N50: Given a set of contigs, the N50 is defined as the sequence length of the shortest contig at 50% of the total assembly length
  - Example: consider 9 contigs with lengths 2,3,4,5,6,7,8,9,10; their sum is 54, half of the sum is 27, and the size of the genome also happens to be 54. 50% of this assembly would be 10 + 9 + 8 = 27 (half the length of the sequence): N50=8
- What factors effect assembly quality? Key one is average depth of coverage:

Depth of coverage

$$c_n = 3$$



n

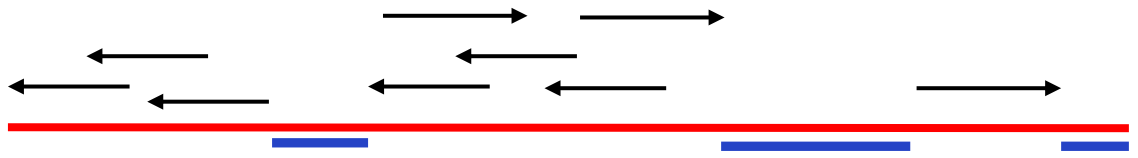
Average depth of coverage of genome

$$\langle c \rangle = \frac{RN}{L}$$

- R is read length
- N is number of reads
- L is genome length

Breadth of coverage

- At low average coverage depths then genome is fragmented through chanceType equation here.
- Bread of coverage is fraction of genome covered by at least one read or equivalent fraction of bases with $c_n > 0$



$$b = \frac{\sum_{n=1}^{L} c_n = 0}{L}$$

Lander-Waterman statistics

- Poisson statistics allow us to estimate breadth of coverage from average coverage depth:

$$\mathrm{b}= 1 - \exp(-\langle c\rangle)$$

- At an average coverage of 5 then 99.3% of genome will be sequenced
- The expected number of contigs:

$$\mathrm{g}= N\exp(-\langle c\rangle)$$

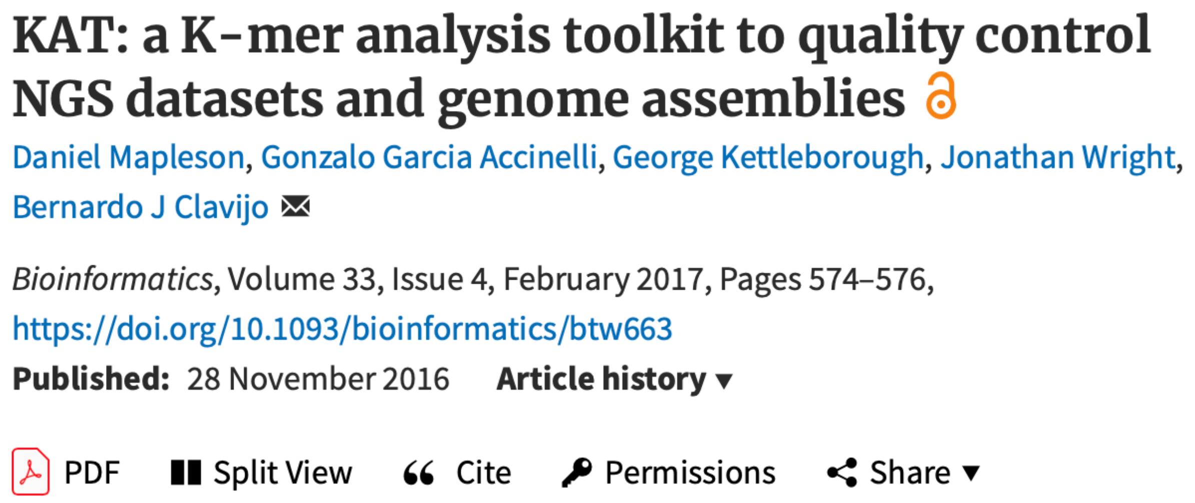
- Coverage depth of 5 with 300 bp reads and 5 Mbp genome what is the expected number of contigs?
- This is an best case prediction due to repeats

Genome assembly software

- For short reads typically use use iterative k-mer de Bruijn graph assembler:
  - SPAdes: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3342519/
  - Megahit
- For long reads OLC:
  - Nanopore: Flye or Canu
  - HiFi PacBio: hifiasm

Contamination

- Potential issue is presence of sequence from organisms other than the target genome
- Can explore this very simply by looking at k-mer frequency spectrum which for bacteria should be unimodal
- Slightly more sophisticated blob plots

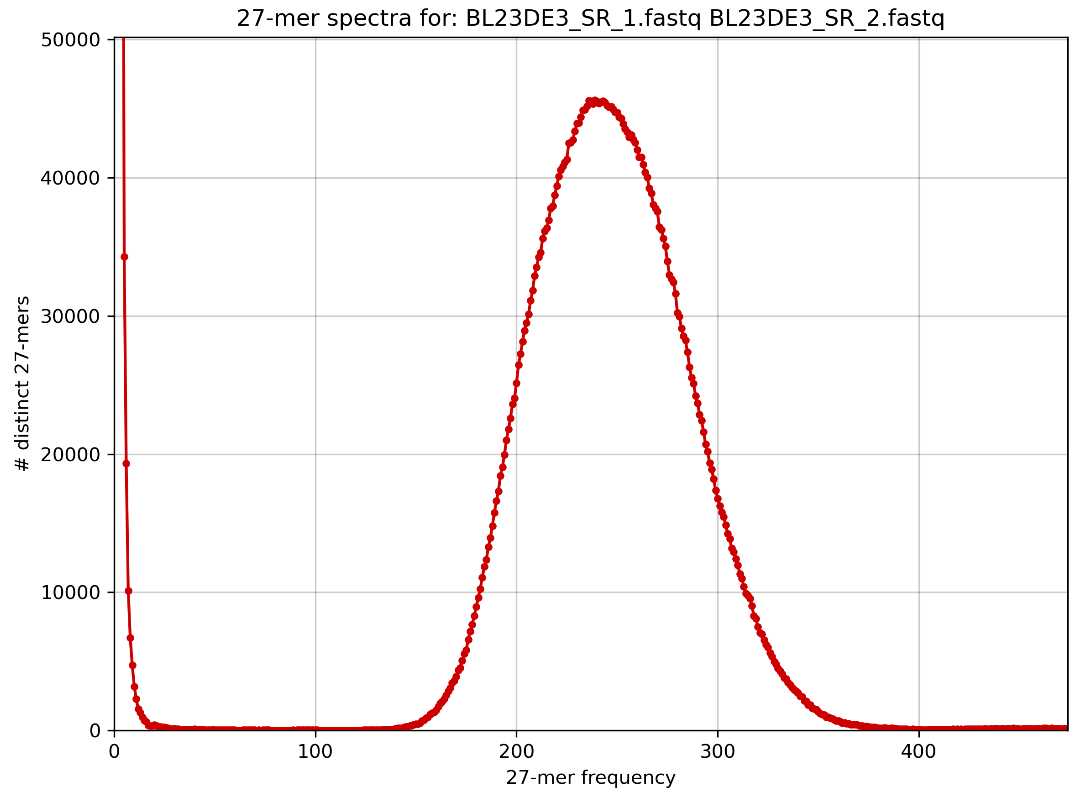


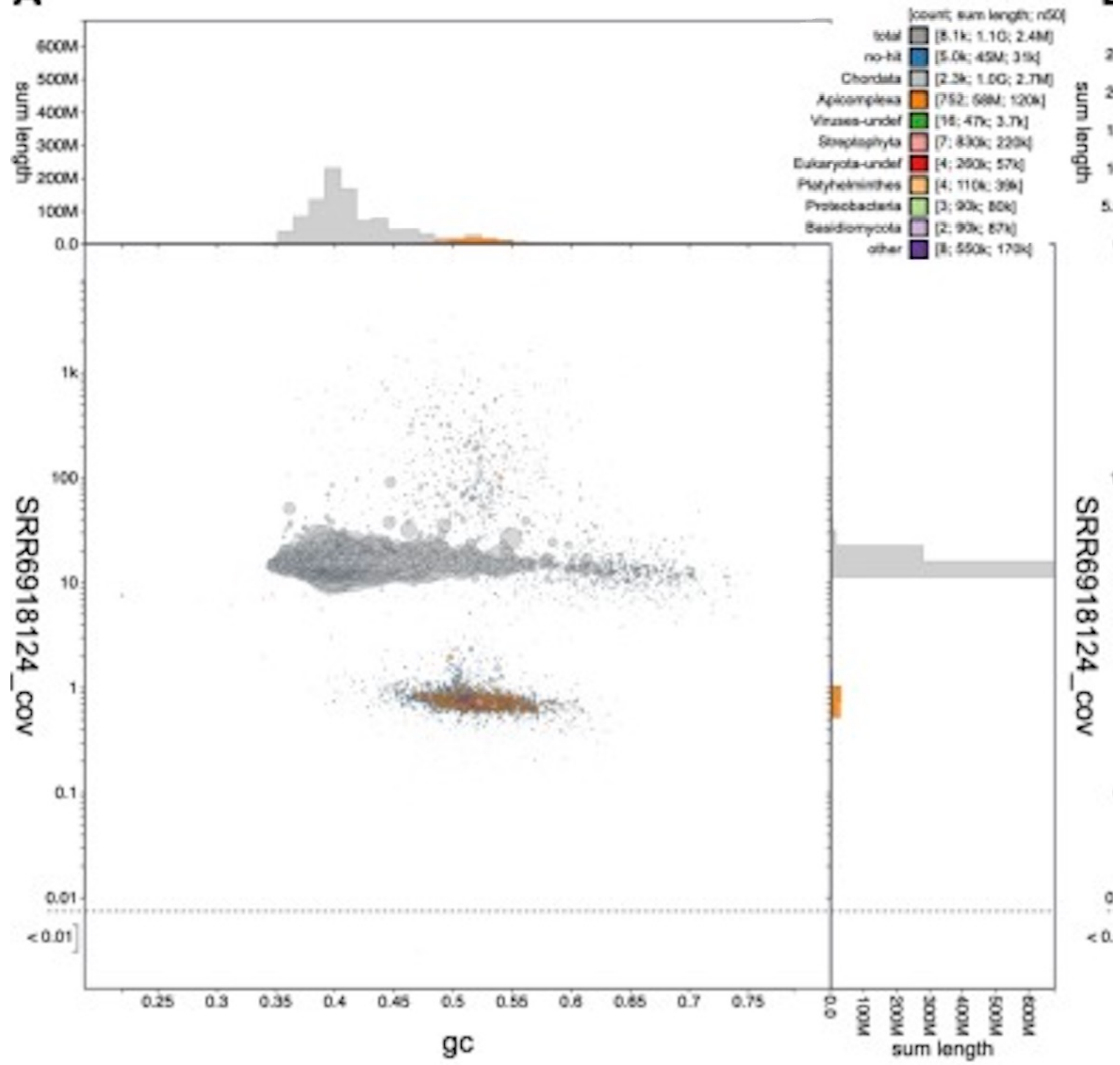
KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies

Daniel Mapleson, Gonzalo Garcia Accinelli, George Kettleborough, Jonathan Wright, Bernardo J Clavijo

Bioinformatics, Volume 33, Issue 4, February 2017, Pages 574–576, https://doi.org/10.1093/bioinformatics/btw663

Published: 28 November 2016 **Article history** ▾





G3 (Bethesda). 2020 Apr; 10(4): 1361–1374.
Published online 2020 Feb 18. doi: 10.1534/g3.119.400908

PMCID: PMC7144090
PMID: 32071071

BlobToolKit – Interactive Quality Assessment of Genome Assemblies

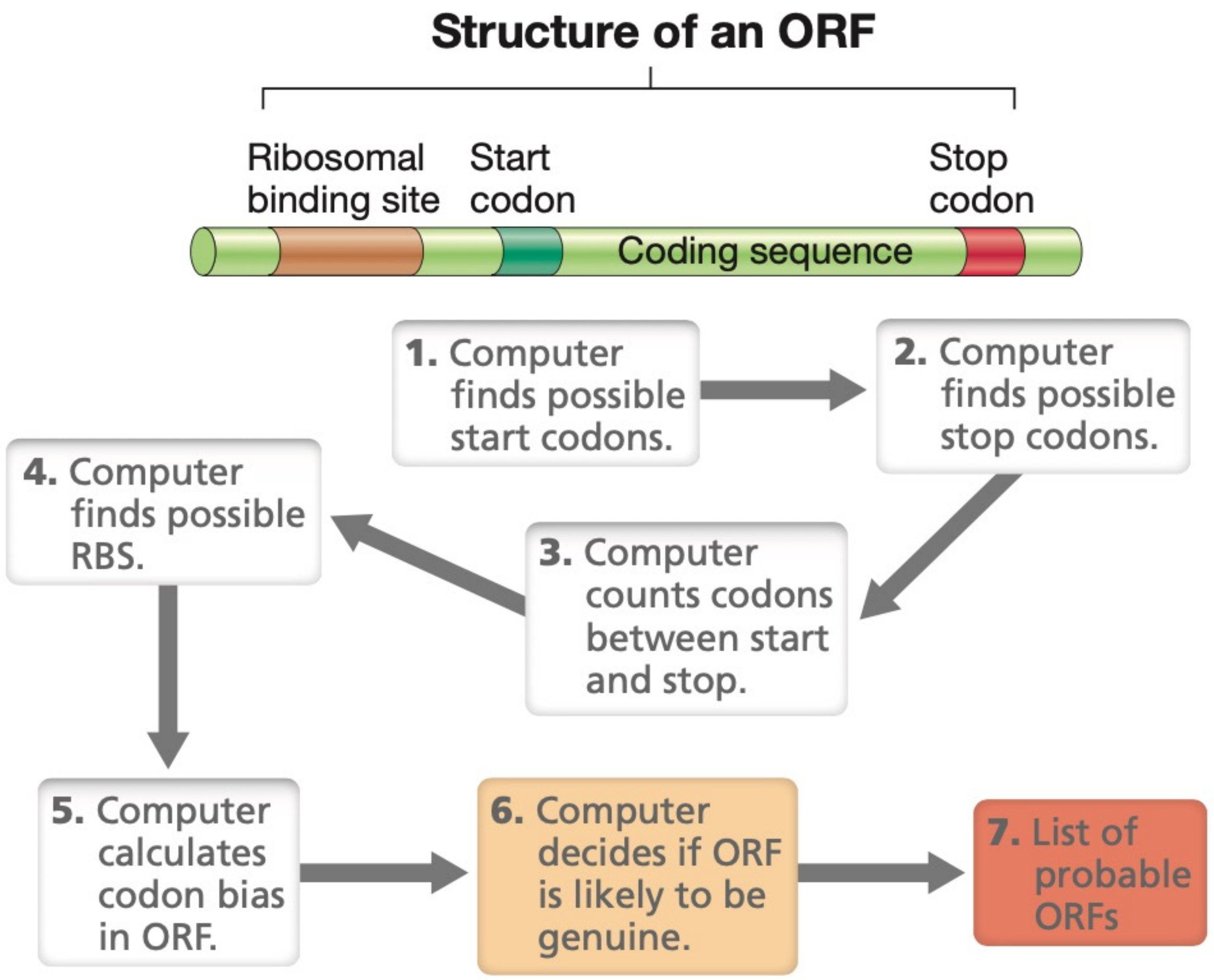
Richard Challis,[*,†,1] Edward Richards,[‡] Jeena Rajan,[‡] Guy Cochrane,[‡] and Mark Blaxter[*†]

Author information Article notes Copyright and License information Disclaimer

Earlham Institute

Genome annotation – Identifying ORFs

- Steps in bacterial genome annotation:
    - Identify open reading frames (ORFs) in contigs
- Typical program for this is Prodigal
- Does require the genetic code to be known
- Prodigal will try and guess code but can get this wrong for exotic microbes

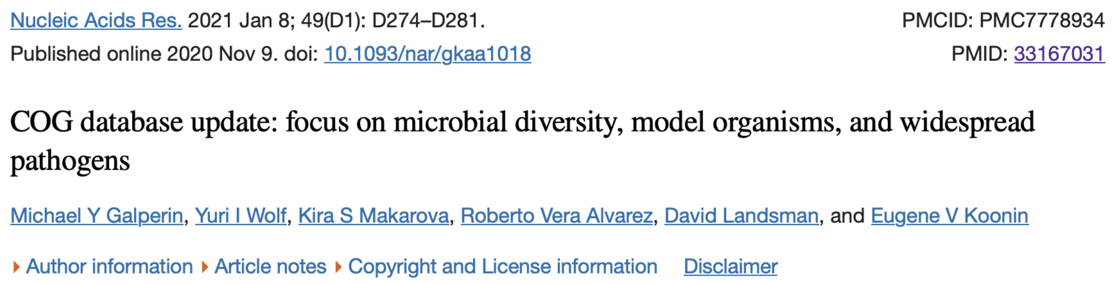

Structure of an ORF
Ribosomal binding site
Start codon
Stop codon
Coding sequence
1. Computer finds possible start codons.
2. Computer finds possible stop codons.
3. Computer counts codons between start and stop.
4. Computer finds possible RBS.
5. Computer calculates codon bias in ORF.
6. Computer decides if ORF is likely to be genuine.
7. List of probable ORFs

Annotating protein coding ORFs to functions

- Annotating ORFs to potential functions is done by searching against databases
- Essentially based on ‘sequence homology’
- We assume that similar sequences perform similar functions
- Search can be performed either with amino acids or nucleotides
    - Pros and cons?
- Complication is need to align sequences

Reference database example

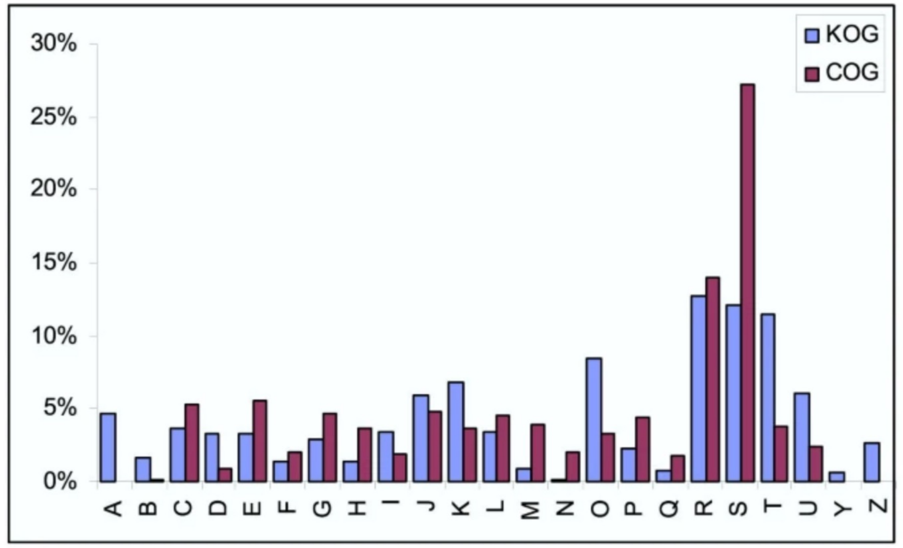
- Database is a set of sequences with labels (annotations)
- Labels can be any feature of interest e.g. Clusters of orthologous genes (COGs) http://www.ncbi.nlm.nih.gov/COG/
- Each COG consists of individual orthologous genes, share common ancestor through speciation, with a particular function:
  - e.g. COG0001 Glutamate-1-semialdehyde aminotransferase
- Current COG based on on complete genomes of 1187 bacteria and 122 archaea
- Database consists of 4,877 COGs that include 3,236,575 unique genes mapped to 3,455,867 genomic loci, which represents 73.5% of the 4,401,819 proteins (4,110,746 bacterial and 291,073 archaeal proteins) encoded by the covered species
- Return to databases in later metagenomics lectures


Nucleic Acids Res. 2021 Jan 8; 49(D1): D274–D281. PMCID: PMC7778934
Published online 2020 Nov 9. doi: 10.1093/nar/gkaa1018 PMID: 33167031

COG database update: focus on microbial diversity, model organisms, and widespread pathogens

Michael Y Galperin, Yuri I Wolf, Kira S Makarova, Roberto Vera Alvarez, David Landsman, and Eugene V Koonin

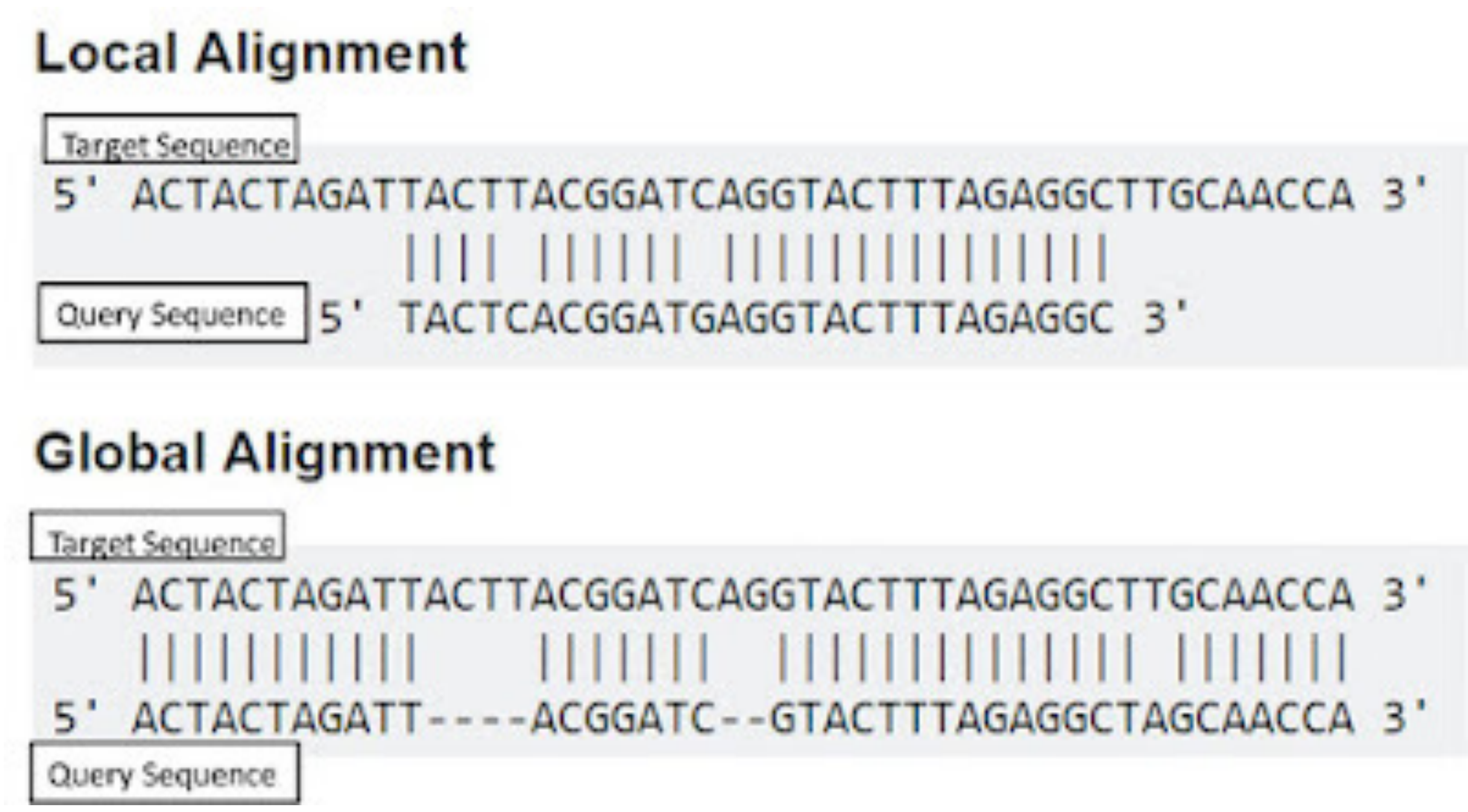
Author information Article notes Copyright and License information Disclaimer



Functional classification of prokaryotic (COGs) and eukaryotic (KOGs) clusters of orthologs. Designations of functional categories: A, RNA processing and modification (not used for prokaryotic COGs), B, chromatin structure and dynamics, C, energy production and conversion, D, cell cycle control and mitosis, E, amino acid metabolism and transport, F, nucleotide metabolism and transport, G, carbohydrate metabolism and transport, H, coenzyme metabolism, I, lipid metabolism, J, translation, K, transcription, L, replication and repair, M, cell wall/membrane/envelope biogenesis, N, Cell motility, O, post-translational modification, protein turnover, chaperone functions, P, Inorganic ion transport and metabolism, Q, secondary metabolites biosynthesis, transport and catabolism, T, signal transduction, U, intracellular trafficking and secretion, Y, nuclear structure (not applicable to prokaryotic COGs), Z, cytoskeleton (not applicable to prokaryotic COGs); R, general functional prediction only (typically, prediction of biochemical activity), S, function unknown. The numbers were obtained after subtracting the COGs that consisted entirely of proteins from unicellular eukaryotes from the COG collection.

Sequence alignment

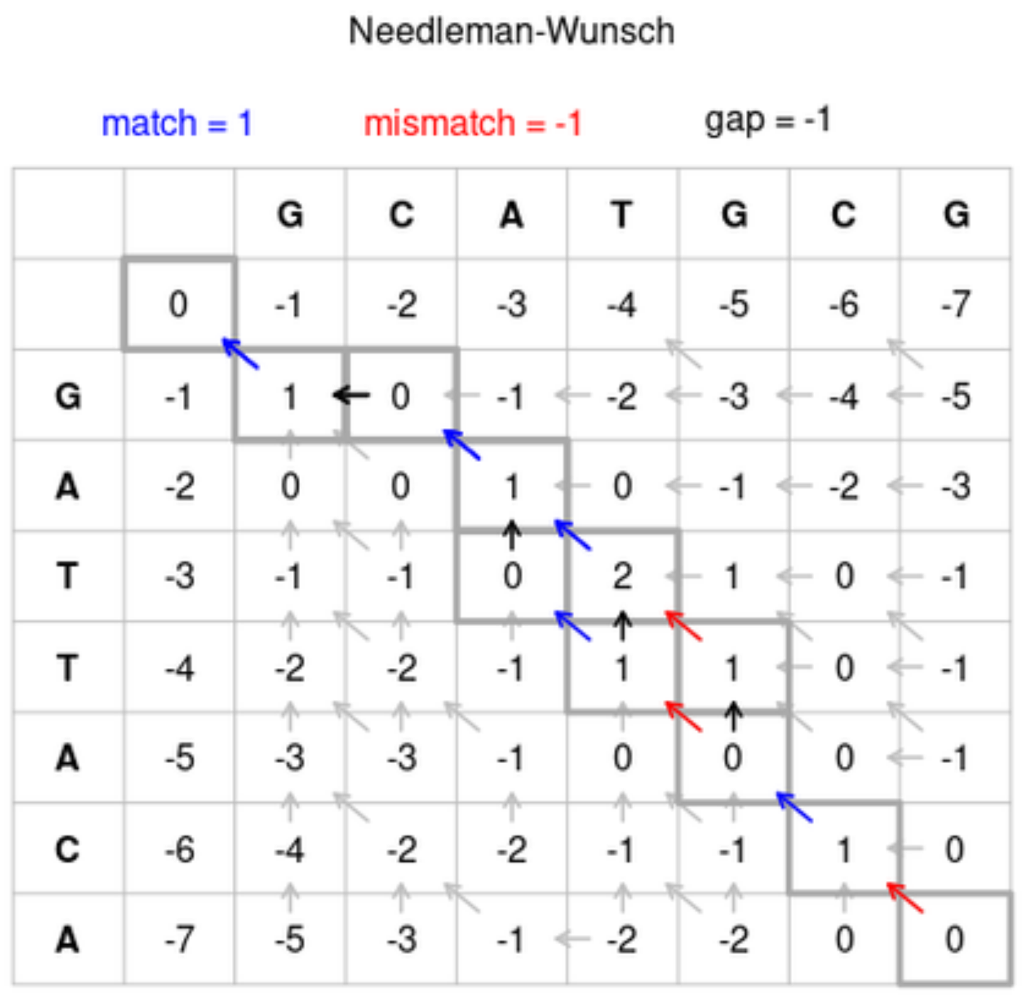
- Alignment can be:
  - Global
  - Local
- Global matches whole sequence
  - search ORF against a database of ORFs
- Local matches part of query to reference database
  - search ORF against a genome


Local Alignment
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
Query Sequence
5' TACTCACGGATGAGGTACTTTAGAGGC 3'
Global Alignment
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
Query Sequence

Earlham Institute

Sequence alignment (cont.)

- Sequence alignment can be performed exactly given a scoring system:
  - E.g. Match (+1), mismatch (-1) or indel (-2)
- Needleman-Wunsch (Smith-Waterman) algorithm finds global (local) optimal alignment for any scoring system
- These are forms of dynamic programming and scale with L^2 where L is the sequence length
- Simple scoring suitable for nucleotides
- Amino acids have different properties and frequencies motivates complex scoring matrices e.g. BLOSUM62 and PAM

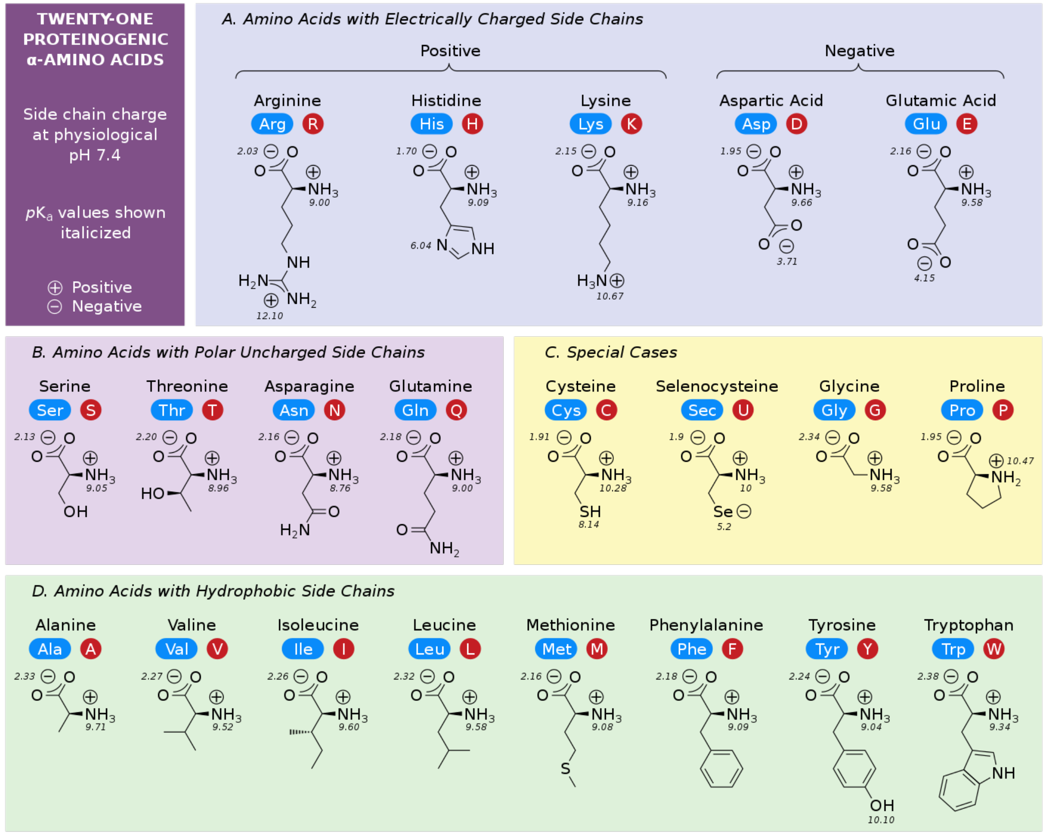


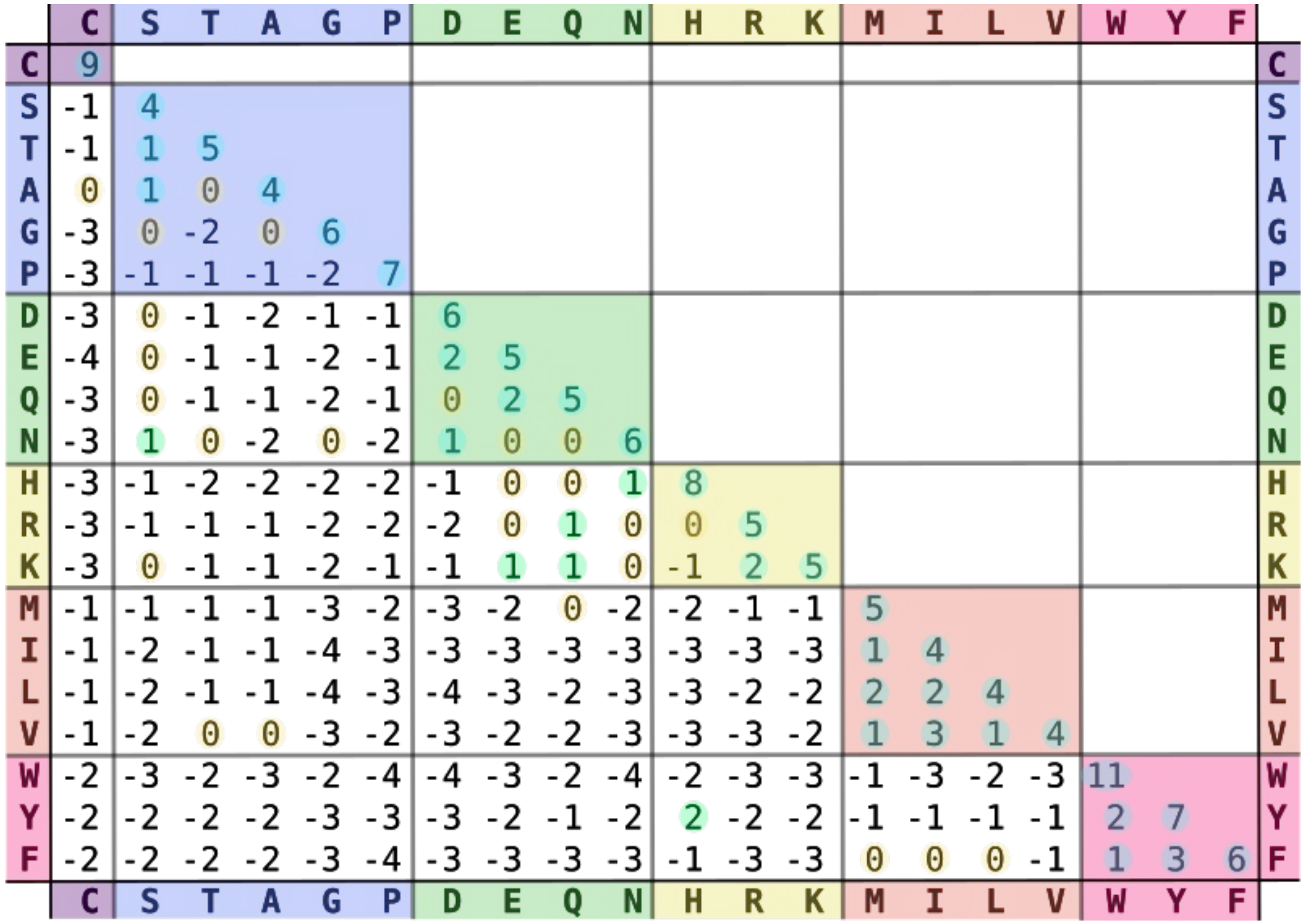
Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

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

BLOSUM: Blocks Substitution Matrix





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

Practical alignment software

- Practical alignment algorithms use a heuristic to approximate exact alignments
- BLAST (basic local alignment search tool) – Altschul et al. 1990
  - Uses seeds similar to protein k-mers to quickly find matches then extends them
  - Differerent modes, blastn, blastp, blastx
  - Highly sensitive can find very distant hits but relatively slow
  - Uses BLOSUM matrices
  - Still useful because of website where whole of NCBI is indexed: https://blast.ncbi.nlm.nih.gov/Blast.cgi
  - Or when database is small
- DIAMOND (Buchfink et al. 2015)
  - Drop-in replacement for BLAST based on double indexing
  - Similar accuracy but 10,000 times faster
- Mapping software sacrifice sensitivity for speed, accurate but only within certain level of similarity
  - E.g. BWA or Minimap
- Hidden Markov models are sensitive but lack precision for very similar sequences
  - E.g HMMER

Measures of hit quality used by BLAST

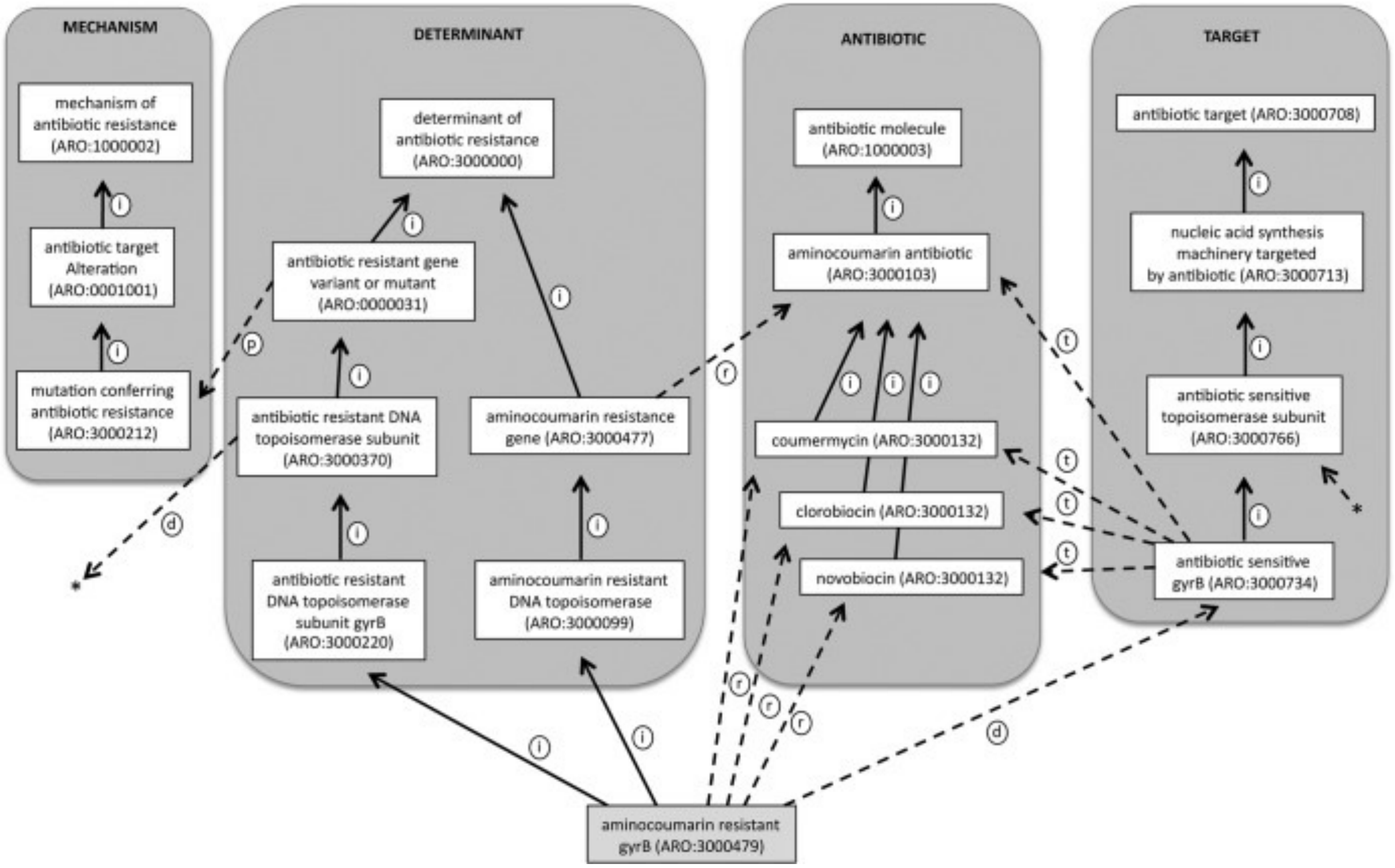
- Percent identity: no. of matches divided by length of alignment
- Bigger database more matches by chance
- Bit-score: Log2 of the database size necessary to find a match this good by chance
    - Larger is better
- E-value: no. of sequences expected to match with this quality when searching against a random database of the same size
    - Smaller is better

$$E = \frac{mn}{2^b}$$

E: e-value
b: bit-score
m: query sequence length
n: database size

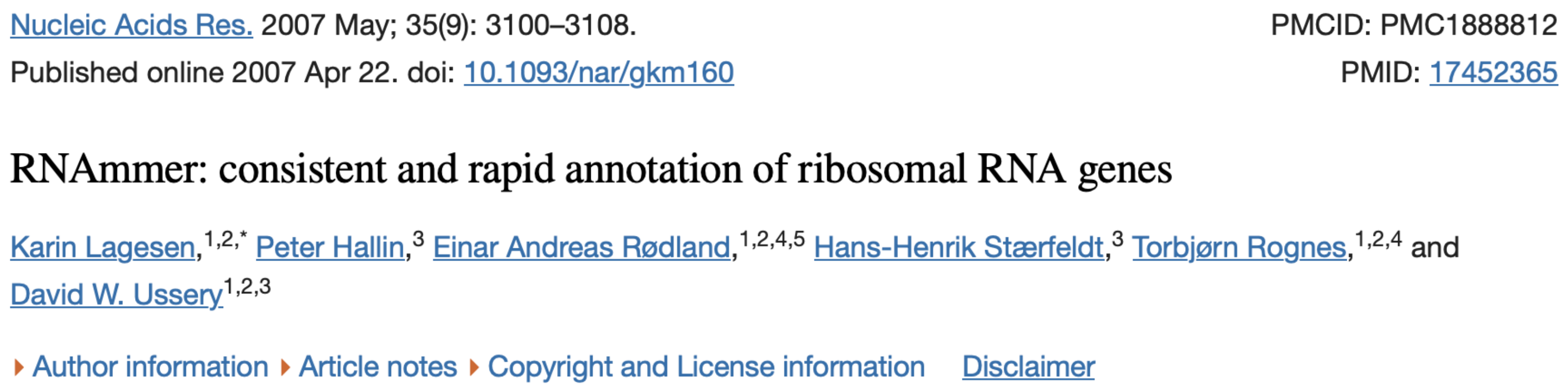
Annotation of anti-microbial resistance genes

- Need database of ARGs many examples: e.g. The Comprehensive Antibiotic Resistance Database (CARD)
- CARD (2020 update): 5,148 Reference Sequences mapped to AMR ontology https://card.mcmaster.ca
- Uses BLAST but with custom bit score threshold for each gene


MECHANISM
mechanism of antibiotic resistance (ARO:1000002)
antibiotic target Alteration (ARO:0001001)
mutation conferring antibiotic resistance (ARO:3000212)
DETERMINANT
determinant of antibiotic resistance (ARO:3000000)
antibiotic resistant gene variant or mutant (ARO:0000031)
antibiotic resistant DNA topoisomerase subunit (ARO:3000370)
aminocoumarin resistance gene (ARO:3000477)
antibiotic resistant DNA topoisomerase subunit gyrB (ARO:3000220)
aminocoumarin resistant DNA topoisomerase (ARO:3000099)
ANTIBIOTIC
antibiotic molecule (ARO:1000003)
aminocoumarin antibiotic (ARO:3000103)
coumermycin (ARO:3000132)
clorobiocin (ARO:3000132)
novobiocin (ARO:3000132)
TARGET
antibiotic target (ARO:3000708)
nucleic acid synthesis machinery targeted by antibiotic (ARO:3000713)
antibiotic sensitive topoisomerase subunit (ARO:3000766)
antibiotic sensitive gyrB (ARO:3000734)
aminocoumarin resistant gyrB (ARO:3000479)
i
p
d
r
t
*

Earlham Institute

Identifying RNA genes



Nucleic Acids Res. 2007 May; 35(9): 3100–3108. PMCID: PMC1888812

Published online 2007 Apr 22. doi: 10.1093/nar/gkm160 PMID: 17452365

RNAmmer: consistent and rapid annotation of ribosomal RNA genes

Karin Lagesen,[1,2,*] Peter Hallin,[3] Einar Andreas Rødland,[1,2,4,5] Hans-Henrik Stærfeldt,[3] Torbjørn Rognes,[1,2,4] and David W. Ussery[1,2,3]

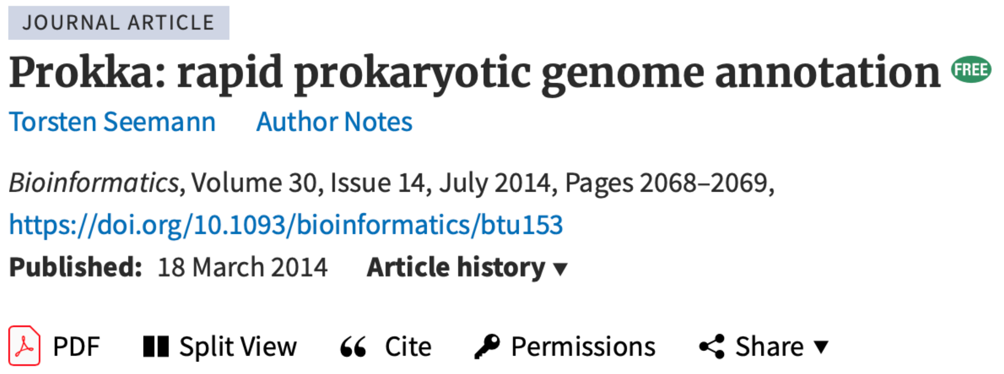
Author information Article notes Copyright and License information Disclaimer

- This can be done using hidden Markov models
- Hidden Markov models are trained with a database and can then with very high sensitivity detect similar sequence
- rRNA sequence is very different from protein coding sequences makes it quite a straightforward problem

Earlham Institute

Pipelines for genome annotation

- Command line pipelines e.g. Prokka
- Web based servers e.g. RAST https://rast.nmpdr.org or IMG https://img.jgi.doe.gov



JOURNAL ARTICLE

Prokka: rapid prokaryotic genome annotation FREE

Torsten Seemann   Author Notes

Bioinformatics, Volume 30, Issue 14, July 2014, Pages 2068–2069, https://doi.org/10.1093/bioinformatics/btu153

Published: 18 March 2014   **Article history** ▾

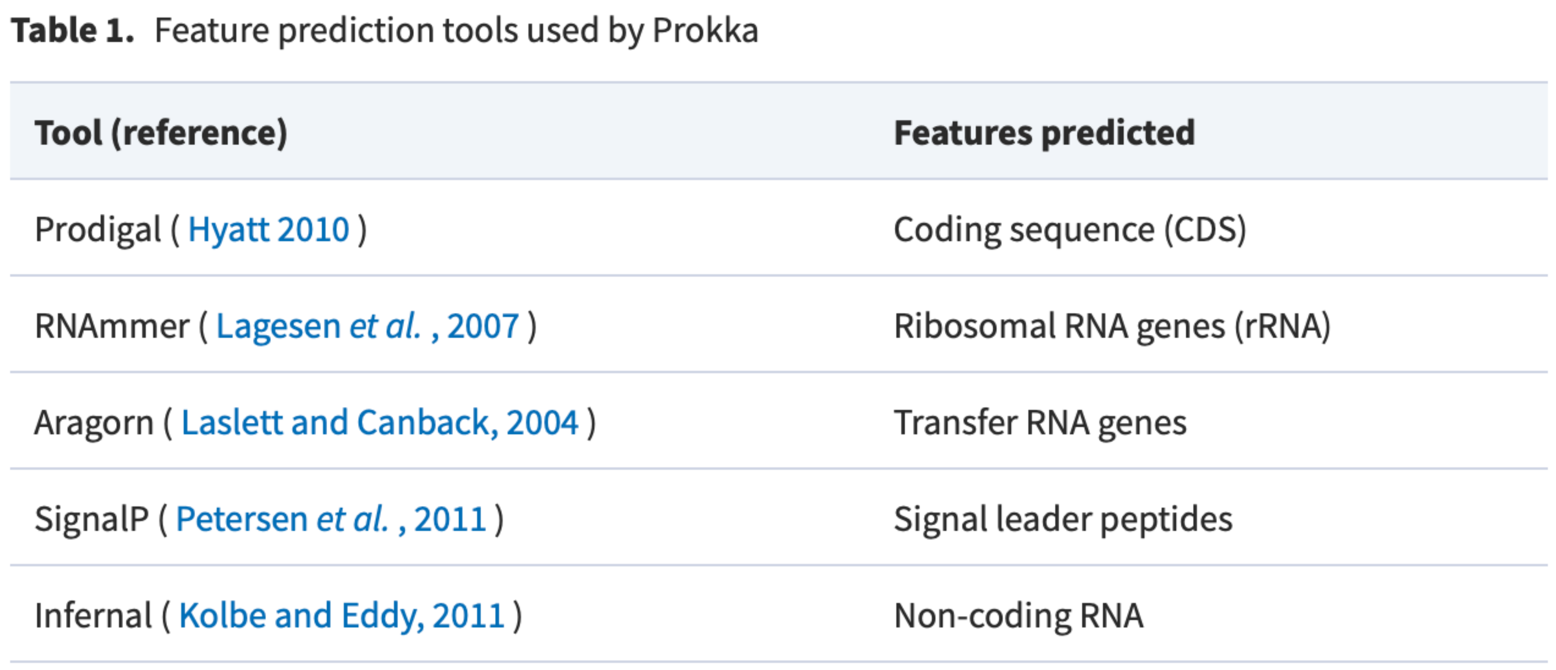


Table 1. Feature prediction tools used by Prokka

Tool (reference)	**Features predicted**
Prodigal (Hyatt 2010)	Coding sequence (CDS)
RNAmmer (Lagesen *et al.* , 2007)	Ribosomal RNA genes (rRNA)
Aragorn (Laslett and Canback, 2004)	Transfer RNA genes
SignalP (Petersen *et al.* , 2011)	Signal leader peptides
Infernal (Kolbe and Eddy, 2011)	Non-coding RNA

Earlham Institute

Other things you may want to do with a genome

- Find prophages
- Virulence factors
- Annotate plasmids
- Variant detection
- Pangenomes
- Phylogenetic trees: FastTree, RaxML, IQTree

Future directions

- Functional annotation through structure (AlphaFold)