From zero to hero

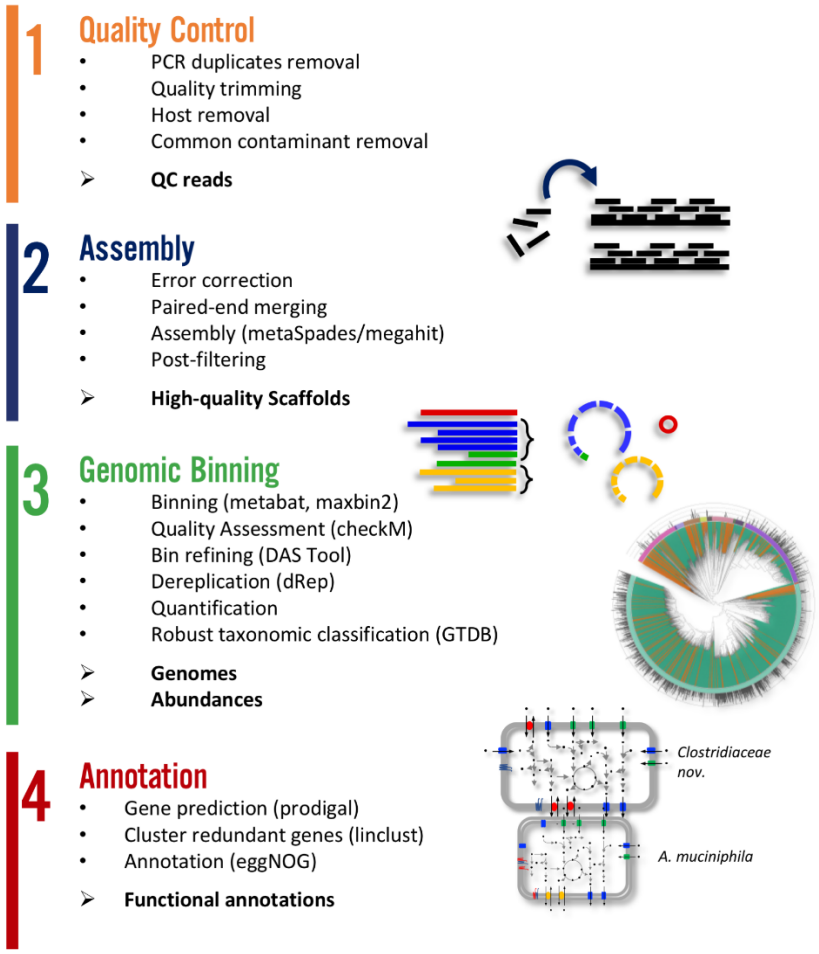
Metagenomics analysis from raw data to annotated assembled genomes

Vittorio Tracanna – Wageningen 10.10.22

Institut für Pflanzenwissenschaften | Mathematisch-Naturwissenschaftliche Fakultät |
Vittorio Tracanna

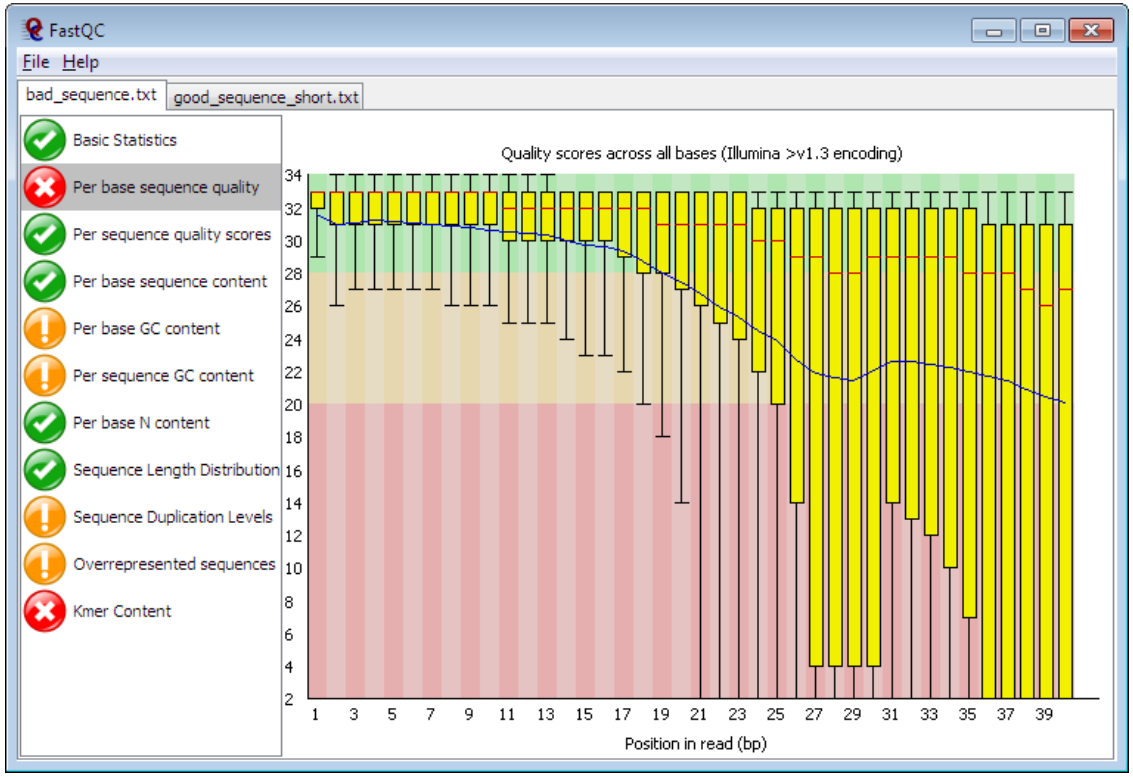
Topics

- **Introduction – who are you? Why are you here?**
- Data preprocessing
- Assembly
- Binning
- Annotation



Universität zu Köln

Data preprocessing – Quality check



Universität zu Köln

Data preprocessing - Adapters

Adapters are nucleotide sequences placed at either one or both ends of the DNA fragments that are being sequenced

They are composed of 3 sections:

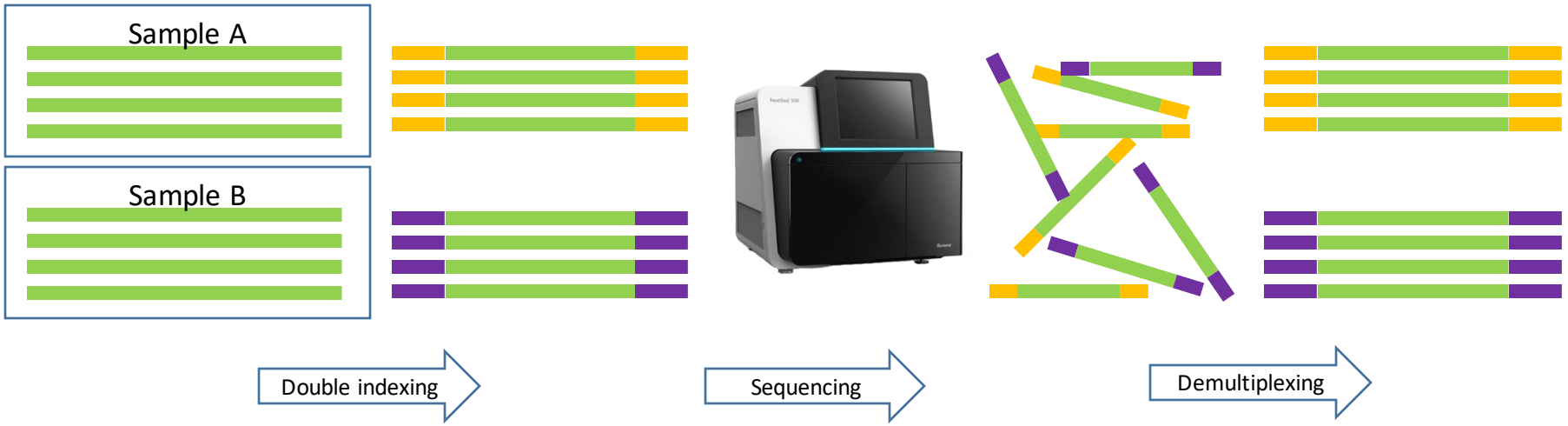
- Sequencer binding site (illumina)
- Multiplexing index (P5-P7)
- Sequencing primer binding site (illumina)

They are necessary* for sequencing but should be removed early on in data pre-processing steps

Data preprocessing - Demultiplexing

Sample multiplexing is a library preparation technique where multiple samples are sequenced simultaneously in the same flow cell lane through the addition of a multiplexing barcode (index adapter) either at the end or both ends of the read.

Data preprocessing - Demultiplexing



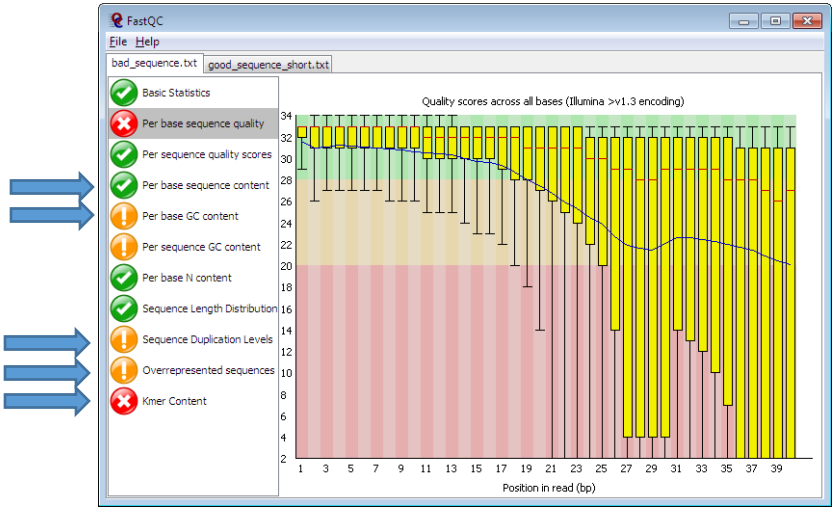
Index sample A
Index sample B
DNA sequence

Demultiplexing tools: Sabre, iDemux etc..

Generally performed by sequencing companies before sending the data. Good to know what it is to be able to spot it in QC.

How do you spot adapters in QC?

Data preprocessing - adapter trimming



From QC data you may notice that adapters are still present in your sequence. You should remove them either by providing the adapter sequence or using a de-novo search.

Recommended tools: Trimmomatic, Cutadapt, bbduk

After adapter removal, rerun QC on the fastq files

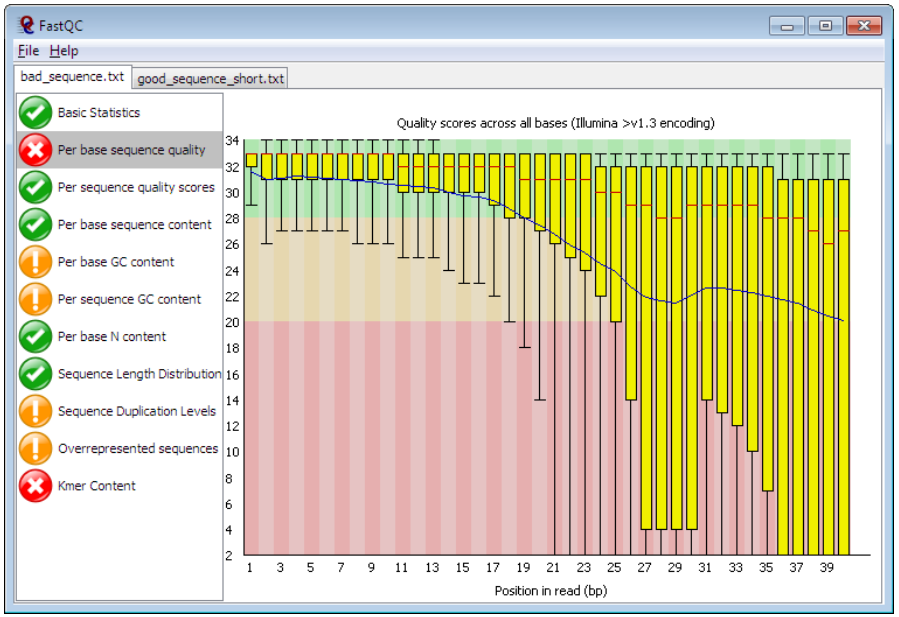
Keep an eye out for polyA and polyG sequences

Universität zu Köln

Data preprocessing - Quality filter

Phred quality score – Logarithmic score representing the quality of a nucleotide

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%



Universität zu Köln

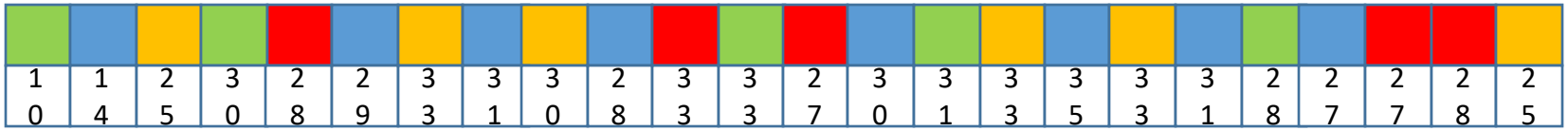
Data preprocessing – quality filter

Recommended tools: Trimmomatic, Sickle, bbduk

Sliding window quality threshold: 28-30

Sliding window size: 4-5

Min sequence length: 35-250



19.75

24.25

28.00

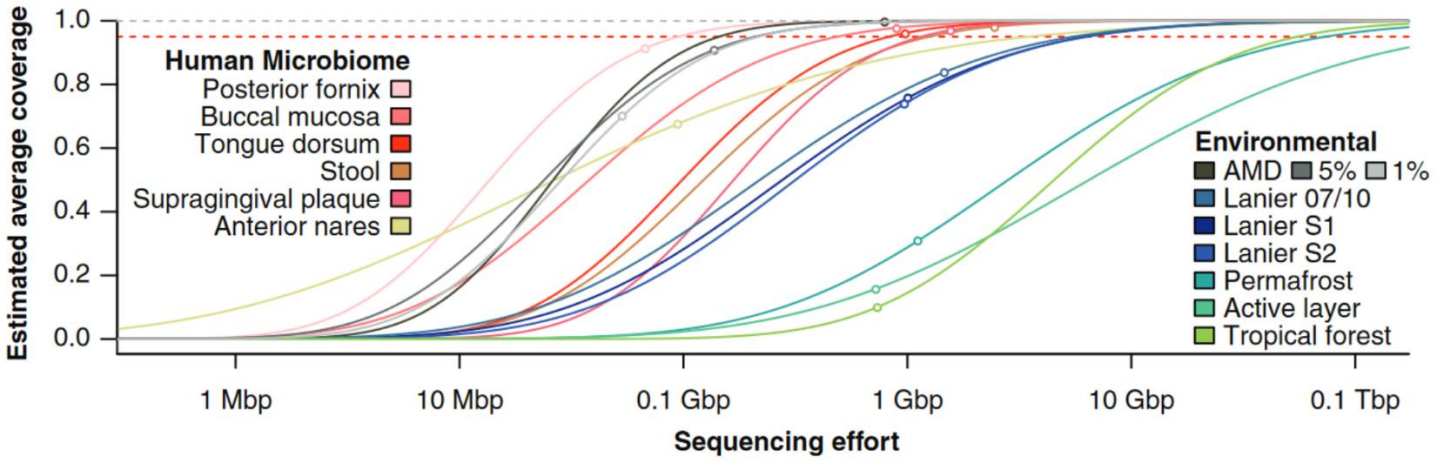
our choice? 30.00

What factors should influence

Data preprocessing - Rarefaction analysis

'*Everything is everywhere*, but, *the environment selects*' Becking and Beijerinck

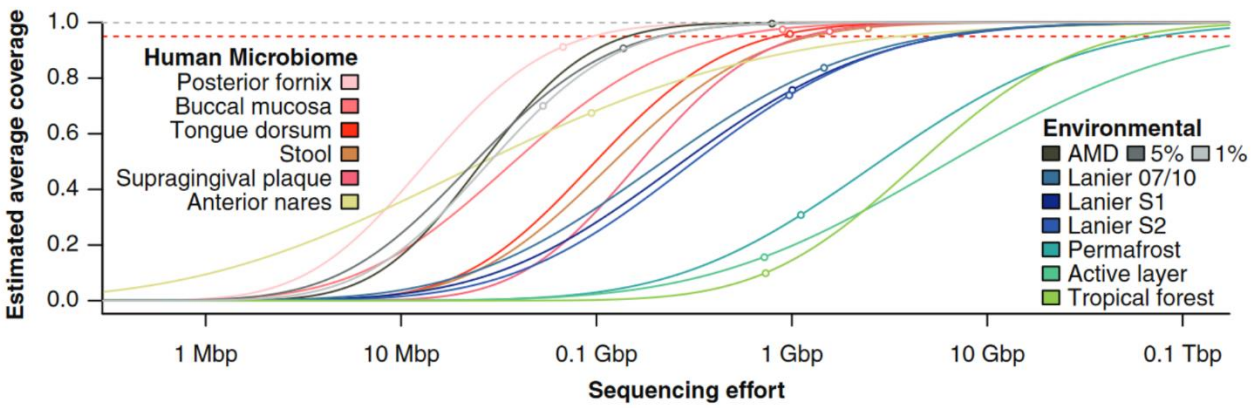
However, funding is limited. How much should I sequence to cover my metagenome?



Universität zu Köln

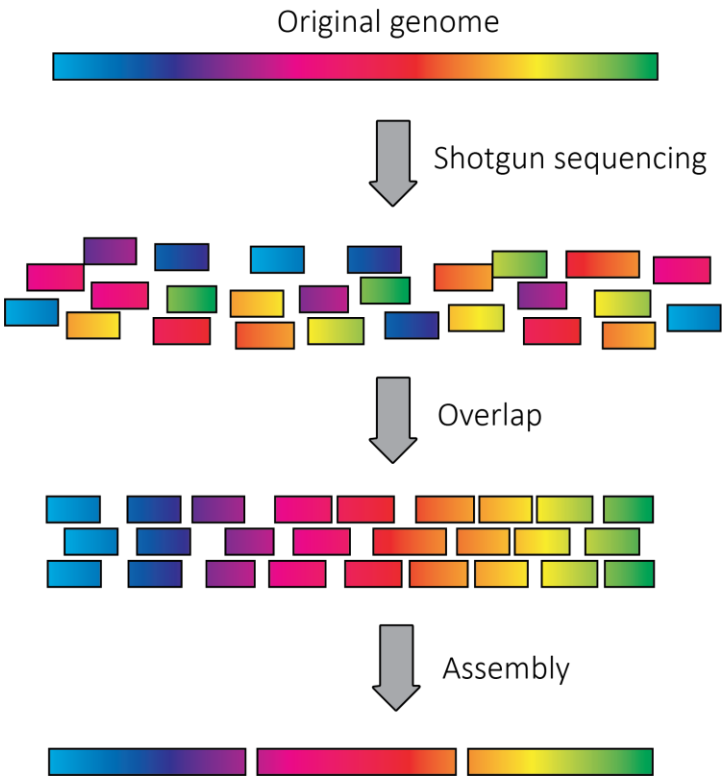
Data preprocessing – Nonpareil estimate

- Redundancy estimation
  - Pairwise read alignment
  - Kmer-based
- Estimation of the abundance-weighted average coverage at different sequencing efforts (nonpareil curve)



Universität zu Köln

Assembly – Overlap layout consensus [OLC]



- Find all **approximate overlaps** between reads at a level of stringency consistent with the estimated error rate of the underlying technology
- Use the overlaps to decide on a layout or tiling of the reads
- Produce a consensus sequence of the reads covering a given region

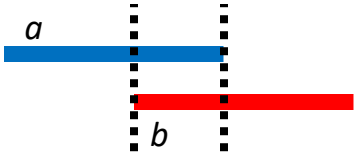
A history of DNA sequence assembly
Eugene W. Myers Jr

Universität zu Köln

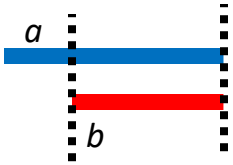
Assembly – Overlap layout consensus [OLC]

Approximate overlaps at rate ϵ between reads a and b

Proper overlap - aligns a suffix of a with a prefix of b



Containment overlap – aligns a segment of a with all of b

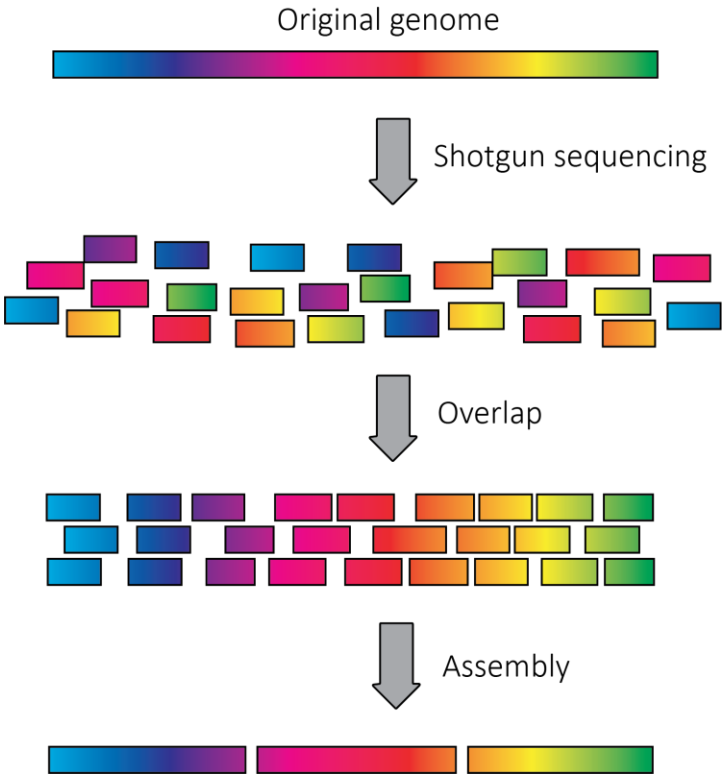


Error rate ϵ is based on the technology

Alignments are computed with classic string alignment tools (Needleman–Wunsch algorithm)

A history of DNA sequence assembly
Eugene W. Myers Jr

Assembly – Overlap layout consensus [OLC]



Find all **approximate overlaps** between reads at a level of stringency consistent with the estimated error rate of the underlying technology

Computing pairwise alignments is a computationally complex task $O(\epsilon N^2/2)$

How would you speed it up?

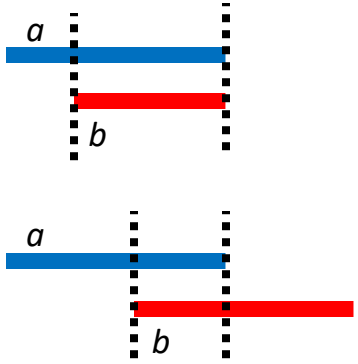
Universität zu Köln

Assembly – Overlap layout consensus [OLC]

- Containment overlap reads do not contribute to the assembly and once found do not need to be included in the matrix

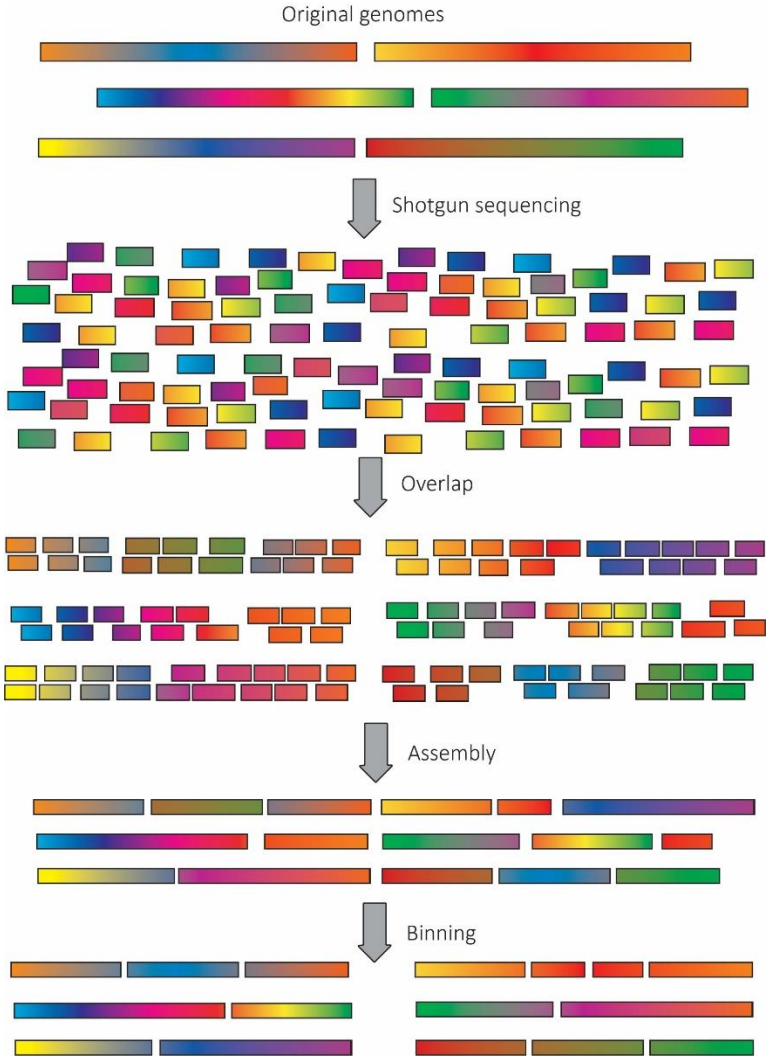
- Reducing the allowed error rate ϵ

What changes for metagenomes?



$O(\epsilon N^2/2)$

Universität zu Köln



Assembly - metagenomics

- We are attempting to solve multiple assembly problems at the same time
- Thankfully, most genomes do not have significant overlaps
- Coverage is uneven for different organisms
- Technology error rate versus strain divergence

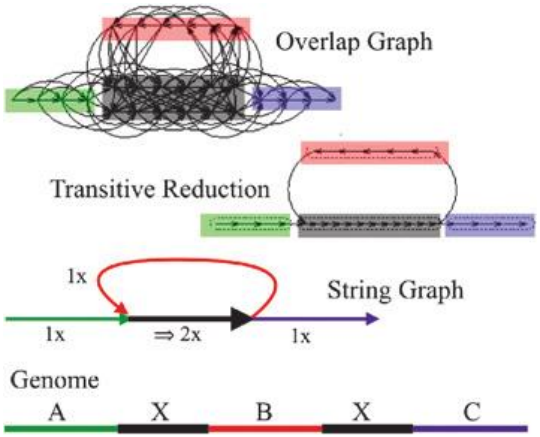
Universität zu Köln

Assembly – String graph

String graph was the first attempt at addressing the assembly as a global problem rather than a local problem

Repeats in an OLC approach appear as high-coverage strings and result in fragmentation of the contig

- String graph performs a **transitive reduction** of an overlap graph
- Compressing chains of vertices into a single compound edge
- Unique, non-repetitive parts of the genome collapse into single compound edges

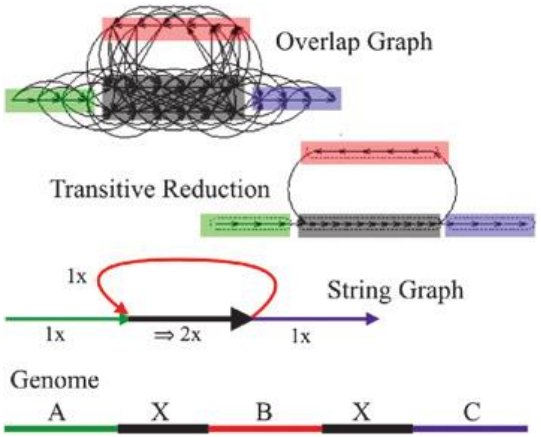


A history of DNA sequence assembly
Eugene W. Myers Jr

Universität zu Köln

Assembly – String graph

- The OLC problem was “unconsciously” being solved by finding a tour through the string graph
- The string graph explicitly formulated the problem with the advantage of being significantly simpler than the overlap graph without eliminating any potential solutions
- The correct assembly is a generalized Eulerian tour that respects the copy number of every compound edge



A history of DNA sequence assembly
Eugene W. Myers Jr

Universität zu Köln

Assembly – De Bruijn graph

Consider all the k-mers of size k of every read, find all k-mers overlapping for k-1 bases. This generates a k-mer “overlap graph” of order k

The graph is transitively reduced and vertices are further collapsed as for string graph construction.

Rather than overlapping “reads” you are overlapping k-mers

It’s based on the assumption that most positions do not have errors.

Why is this approach employed by all modern assemblers?

A history of DNA sequence assembly
Eugene W. Myers Jr

Assembly – De Bruijn graph - 1

ATACGCA =>
ACGCACA =>
CACATAC =>
CACAGCA =>

Separate reads into k-mers. In this example we will use k=4

Assembly – De Bruijn graph

```
ATACGCA => (ATAC),(TACG),(ACGC),(CGCA)
ACGCACA => (ACGC),(CGCA),(GCAC),(CACA)
CACATAC => (CACA),(ACAT),(CATA),(ATAC)
CACAGCA => (CACA),(ACAG),(CAGC),(AGCA)
```

Find k-mers overlapping for k-1 positions

Assembly – De Bruijn graph

```
ATACGCA => (ATAC),(TACG),(ACGC),(CGCA)
ACGCACA => (ACGC),(CGCA),(GCAC),(CACA)
CACATAC => (CACA),(ACAT),(CATA),(ATAC)
CACAGCA => (CACA),(ACAG),(CAGC),(AGCA)
```

(ATAC)

(TACG)

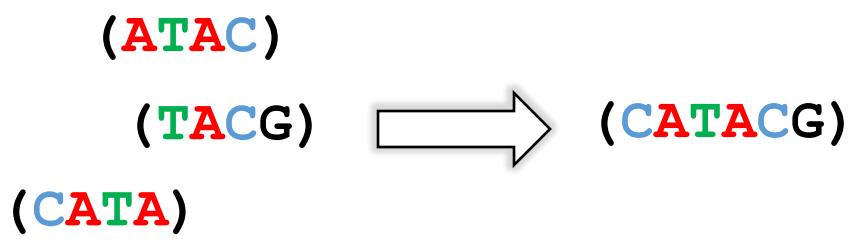
(CATA)

Create an overlap graph, consolidating when possible

Universität zu Köln

Assembly – De Bruijn graph

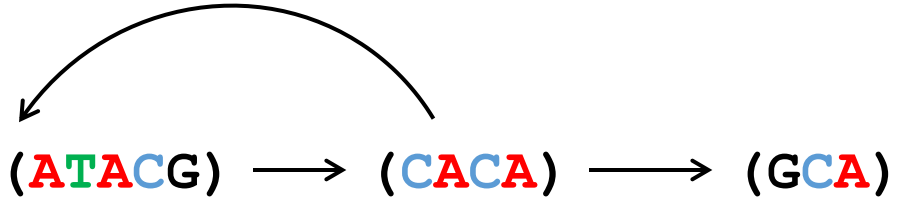
```
ATACGCA => (ATAC),(TACG),(ACGC),(CGCA)
ACGCACA => (ACGC),(CGCA),(GCAC),(CACA)
CACATAC => (CACA),(ACAT),(CATA),(ATAC)
CACAGCA => (CACA),(ACAG),(CAGC),(AGCA)
```



Continue extending the graph for all k-mers

Assembly – De Bruijn graph

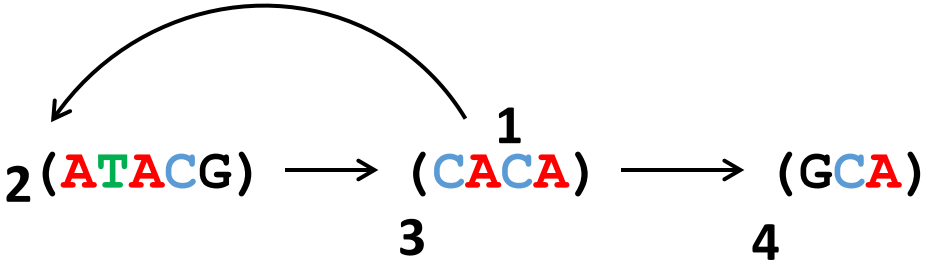
```
ATACGCA => (ATAC),(TACG),(ACGC),(CGCA)
ACGCACA => (ACGC),(CGCA),(GCAC),(CACA)
CACATAC => (CACA),(ACAT),(CATA),(ATAC)
CACAGCA => (CACA),(ACAG),(CAGC),(AGCA)
```



Now traverse the graph using every arrow once (eulerian path)

Assembly – De Bruijn graph

ATACGCA => (ATAC),(TACG),(ACGC),(CGCA)
ACGCACA => (ACGC),(CGCA),(GCAC),(CACA)
CACATAC => (CACA),(ACAT),(CATA),(ATAC)
CACAGCA => (CACA),(ACAG),(CAGC),(AGCA)



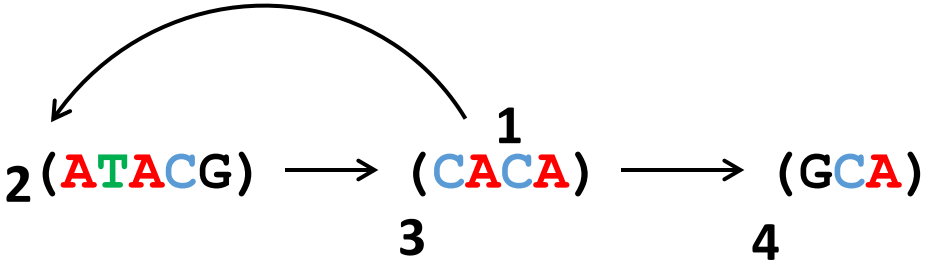
CACATACGCACAGCA

Assembly – De Bruijn graph

What happens for repeats longer than k?

What happens in case of incorrectly called bases?

(Remember that you still have the original reads)



Assembly – Which assembler to choose?

- Metagenomics analysis is still heavily reliant on paired end short reads sequencing.
- Assemblers dedicated to metagenomics need to pay close attention on coverage
  - **what does different coverage mean in standard genome assembly?**
  - **What does it mean in a metagenome?**

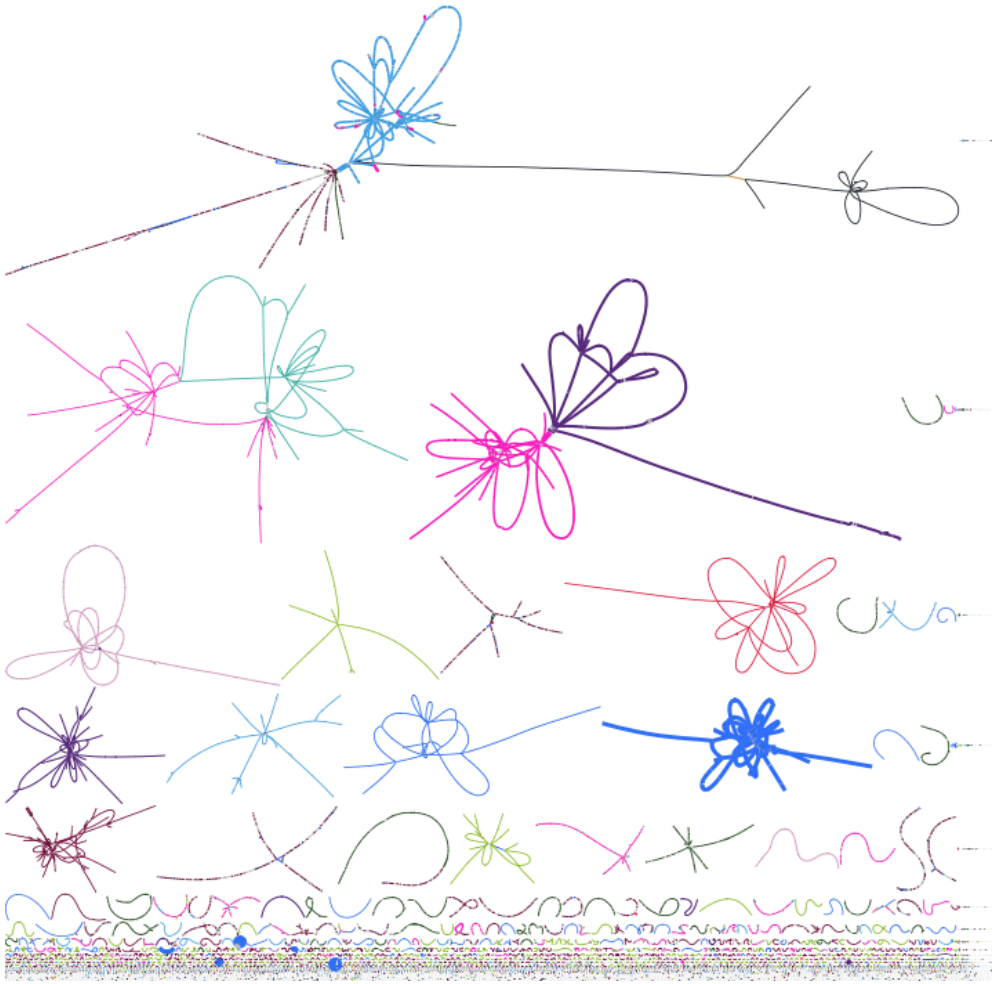
Currently, most used assemblers are megahit (light, fast) and metaSPAdes (slower, more accurate*) both are de Bruijn-based assemblers

Universität zu Köln

Assembly - Bandage

- Bandage is a tool for assembly graph visualization
- It can be used to inspect metagenomic assembly graphs for manual curation

Tutorial at: https://tylerbarnum.com/2018/02/26/how-to-use-assembly-graphs-with-metagenomic-datasets/



Universität zu Köln

Assembly – Quality check

Many tools can produce a QC that can be used to assess the state of your assembly (for example, Quast)

Which one is better?

Report

	final.contigs
# contigs (>= 0 bp)	1299
# contigs (>= 1000 bp)	1101
# contigs (>= 5000 bp)	830
# contigs (>= 10000 bp)	600
# contigs (>= 25000 bp)	297
# contigs (>= 50000 bp)	81
Total length (>= 0 bp)	21549110
Total length (>= 1000 bp)	21436012
Total length (>= 5000 bp)	20718103
Total length (>= 10000 bp)	18993070
Total length (>= 25000 bp)	14095881
Total length (>= 50000 bp)	6605648
# contigs	1214
Largest contig	266122
Total length	21516804
GC (%)	61.48
N50	34088
N90	9074
auN	50048.1
L50	183
L90	640
# N's per 100 kbp	0.00

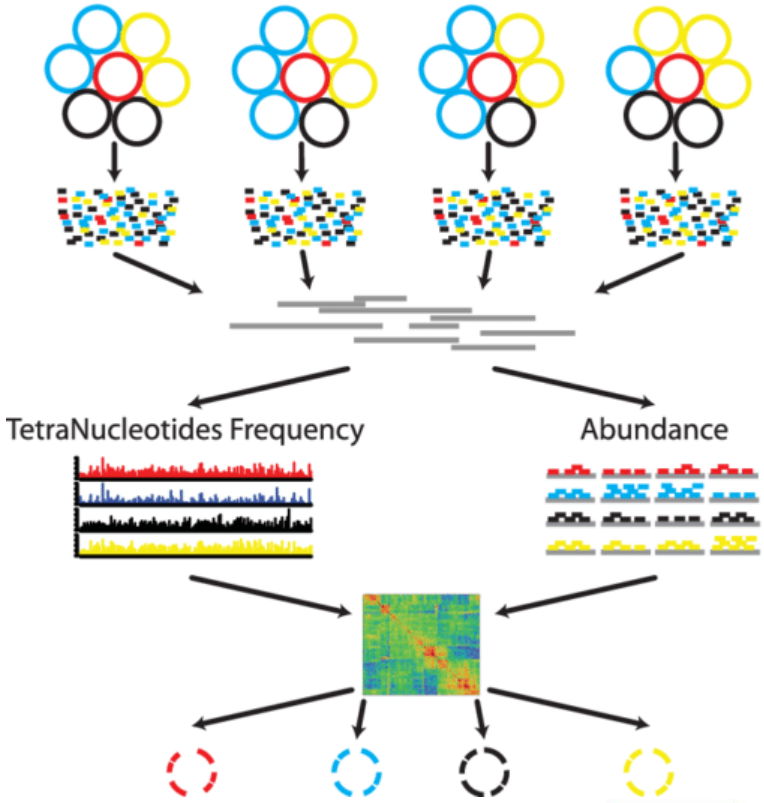
Report

	final.contigs
# contigs (>= 0 bp)	1486
# contigs (>= 1000 bp)	1255
# contigs (>= 5000 bp)	901
# contigs (>= 10000 bp)	624
# contigs (>= 25000 bp)	278
# contigs (>= 50000 bp)	68
Total length (>= 0 bp)	21546615
Total length (>= 1000 bp)	21412775
Total length (>= 5000 bp)	20478745
Total length (>= 10000 bp)	18414478
Total length (>= 25000 bp)	12831591
Total length (>= 50000 bp)	5644773
# contigs	1390
Largest contig	265653
Total length	21509414
GC (%)	61.49
N50	30125
N90	8044
auN	46168.7
L50	203
L90	730
# N's per 100 kbp	0.00

Universität zu Köln

Binning – what is metagenome binning?

- Binning is the process of separating the metagenome assembly contig into “bins” representing the original organism
- Different tools are available, they use a mixture of differential coverage across samples and contig k-mer profiling
- Recommended tools: VAMB, DASTool*, MetaBat2m MetaWrap



Universität zu Köln

Binning – How to validate your bins?

- Assuming that your bins are of bacterial origin, CheckM is an excellent tool to validate their completeness and contamination
- Align predicted genes to collection of bacteria specific* single copy genes (104 genes)
- Count completeness (number of single copy genes with at least one match in the bin) and contamination (number of double, triple, etc. matches)
- What is “high quality bin”?
  - 90% > complete, <10% contamination, complete 16S rRNA region

What does it mean when you have a 100% complete 100% contaminated bin? What happens if you have a fungal origin bin?

Universität zu Köln

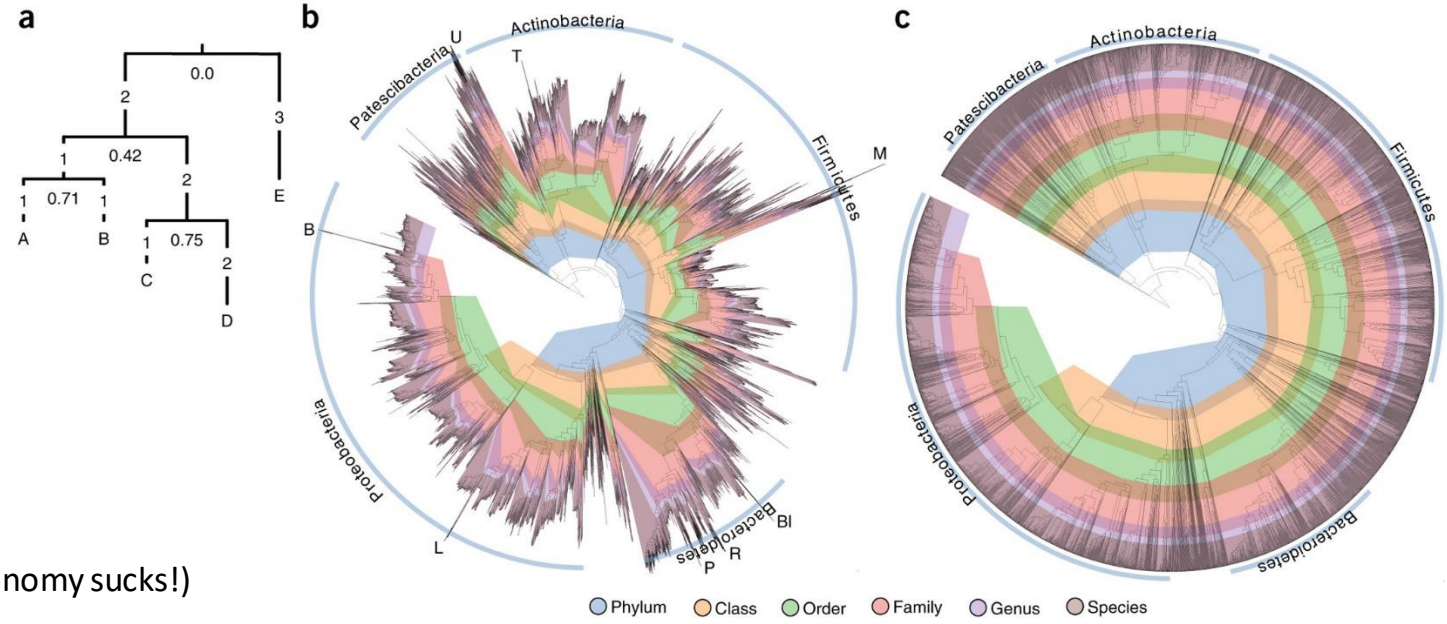
Annotation – taxonomical annotation

- Taxonomical annotation can be performed at 2 stages:
  - Assembled contigs
  - Raw reads
- Assembled contigs or bin taxonomical annotation often relies on blast-like tools (diamond) to find matches to ncbi database
  - In case of multiple equally likely hits uses last common ancestor solutions to assign taxonomy
- Raw reads can be taxonomically annotated using exact-matches (k-mer) based approaches (kraken, centrifuge)

What do you think are the caveats?

Annotation – taxonomical annotation

Binned genomes can also be taxonomically annotated, using any of the approaches mentioned before or GTDB-tk



(ncbi taxonomy sucks!)

Universität zu Köln

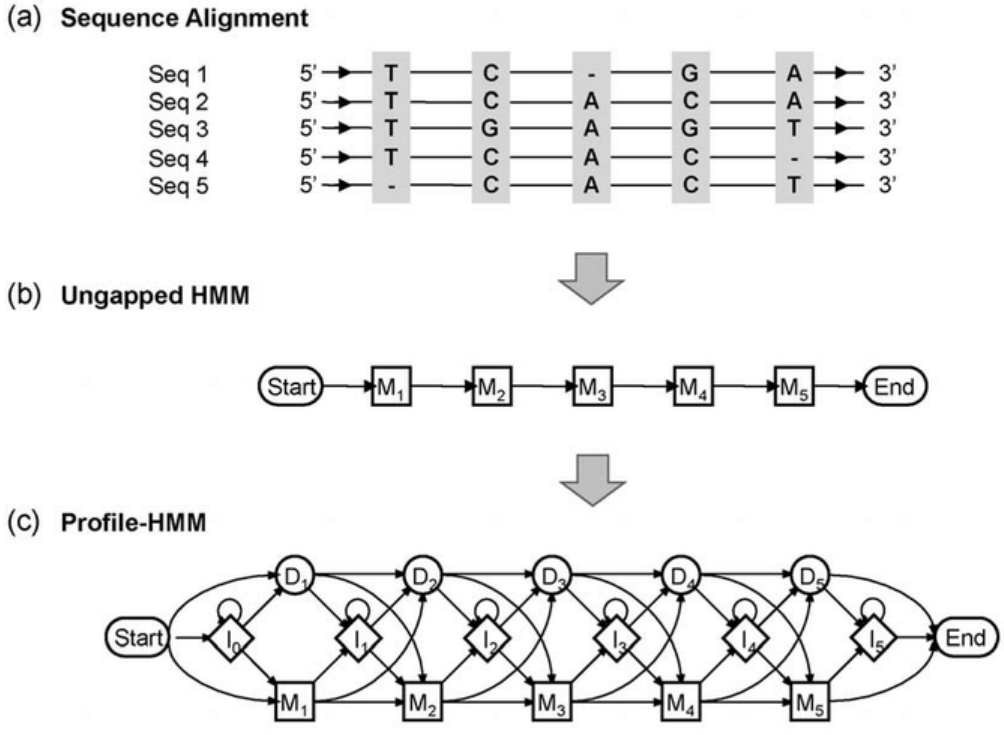
Annotation – functional annotation

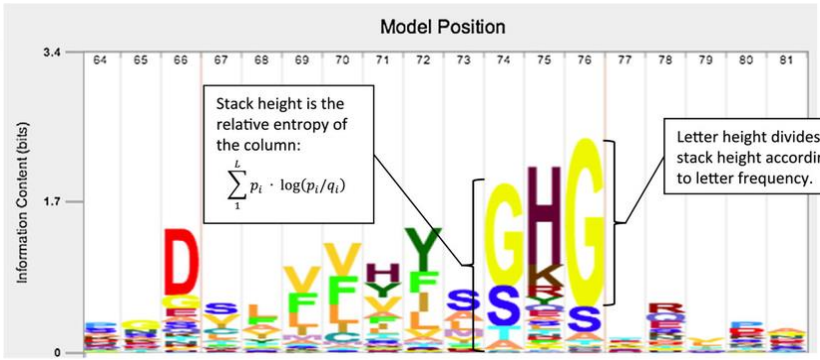
- Functional (gene) annotation starts from gene prediction
- Genes can then be annotated using a multitude of databases, **can you name some?**
- These tools generally use hmm-profiles or blast-like searches against the database to infer function

- **What is an hmm profile?**

Hidden Markov Model – what is it?

HMM are a probabilistic based model to represent a group of (aligned) sequences





M_k ***Match*** *states*

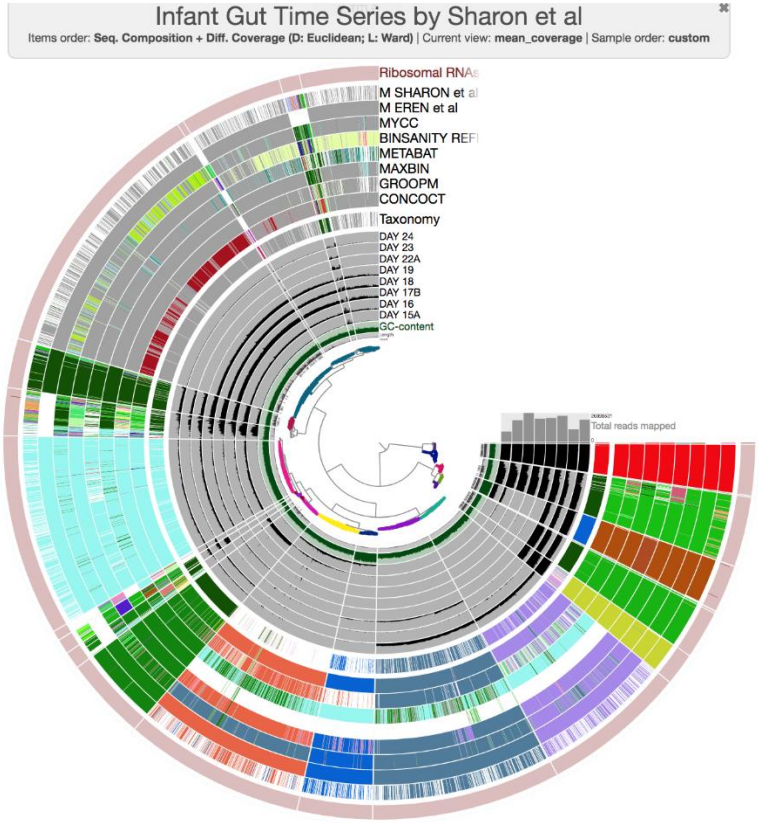
I_k ***Insert*** *states*

D_k ***Delete*** *states*

Wheeler et al. 2014
Yoon et al., 2009

Universität zu Köln

Complete pipelines

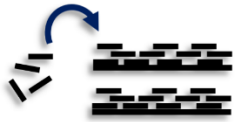


https://anvio.org/

1 Quality Control

- PCR duplicates removal
- Quality trimming
- Host removal
- Common contaminant removal

➢ **QC reads**



2 Assembly

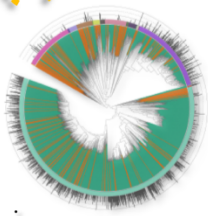
- Error correction
- Paired-end merging
- Assembly (metaSpades/megahit)
- Post-filtering

➢ **High-quality Scaffolds**

3 Genomic Binning

- Binning (metabat, maxbin2)
- Quality Assessment (checkM)
- Bin refining (DAS Tool)
- Dereplication (dRep)
- Quantification
- Robust taxonomic classification (GTDB)

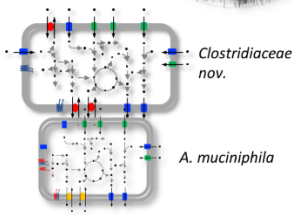
➢ **Genomes**
➢ **Abundances**



4 Annotation

- Gene prediction (prodigal)
- Cluster redundant genes (linclust)
- Annotation (eggNOG)

➢ **Functional annotations**



https://github.com/metagenome-atlas/atlas

Universität zu Köln

Summary – take home messages

- Metagenomic assembly is a branch of the genome assembly problem with unique challenges in up and downstream analysis
- Requires large volumes of sequencing data
- Some steps of the analysis require manual curation
- It's not an "exact science"

Do you have any questions for us?

Universität zu Köln

Introduction to experiment

- In the practical you will go from raw reads to taxonomical characterization of metagenome assembled genomes
- In this mock experiment we have a small diverse synthetic rhizosphere community
- We expose the plant to fungal chitin and we expect the plant to respond to this treatment by modulating the microbial community it associates with in the rhizosphere
- Can you find out how the microbial community adapts?