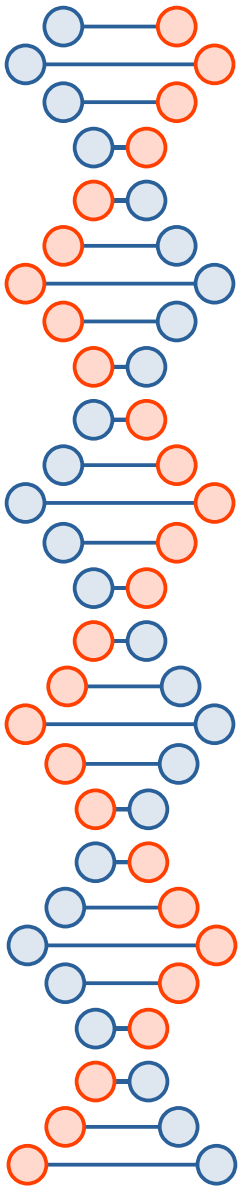
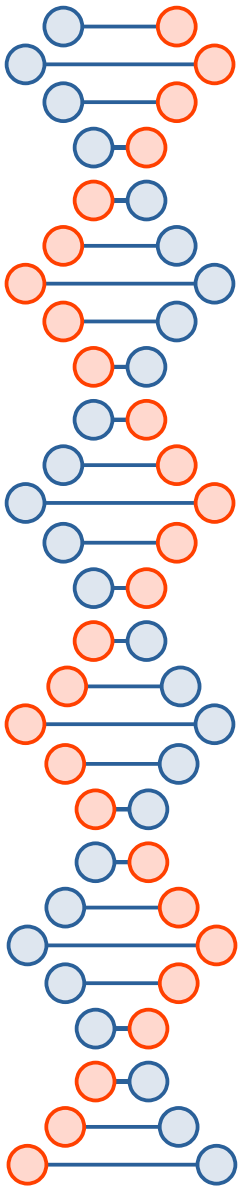


StrainR2
(Shotgun Metagenomic Strain Abundances)

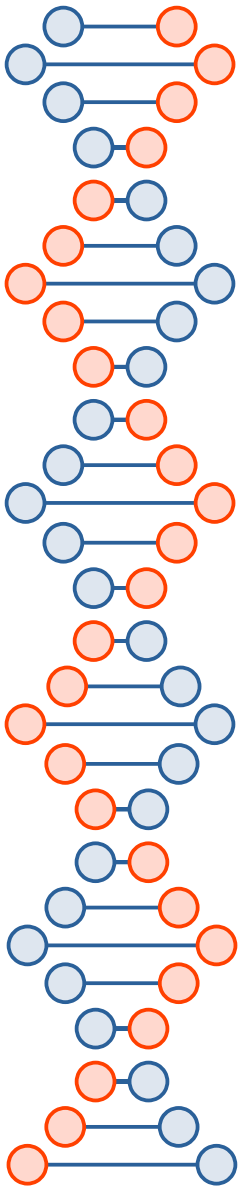


Setup

conda create -n strainr2 -c bioconda -c conda-forge strainr2

conda activate strainr2

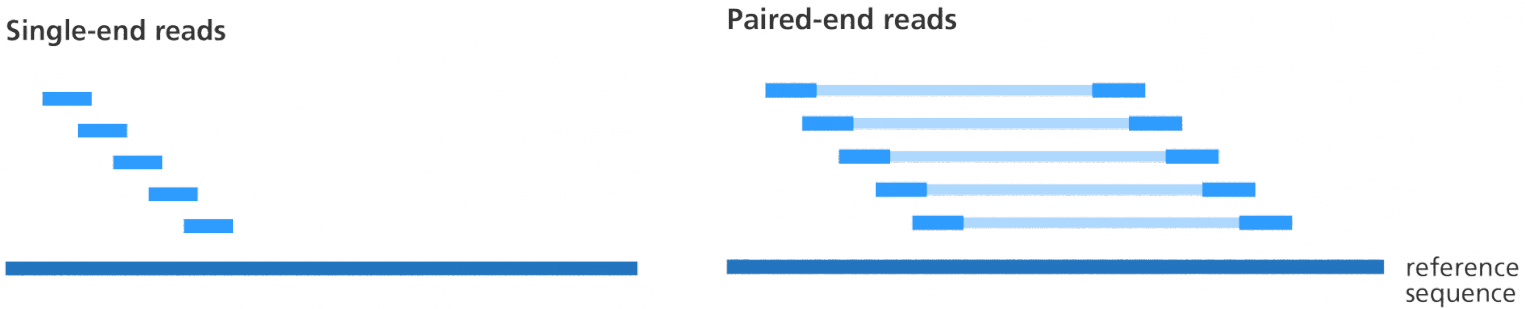
- Conda is a package manager that makes sure tools run consistently and makes life easier
- We will be running StrainR2 on a small mock data set
  - Download data from github



What is Shotgun Metagenomic Sequencing?

- Sequencing thousands of organisms at the same time
  - “Shotgun” = untargeted; “Meta” = multiple organisms
  - Allows for analysis of bacterial diversity and abundance
- In contrast to 16S rRNA amplicon sequencing or other PCR-based approaches
  - More expensive
  - Must sequence 1 organism at a time
  - More accurate abundances

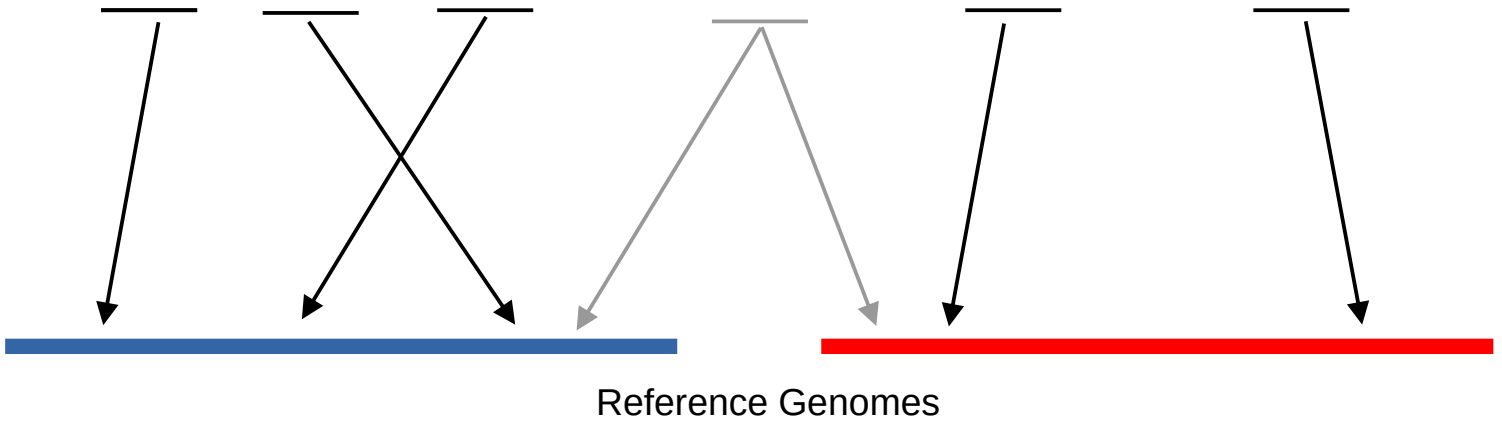
Shotgun Sequencing Output



- Reads are at most a couple hundred bases
- Note: A pair of paired end reads is one “Fragment”
- Billions of reads can be generated each run
- They carry no information about which organism they came from

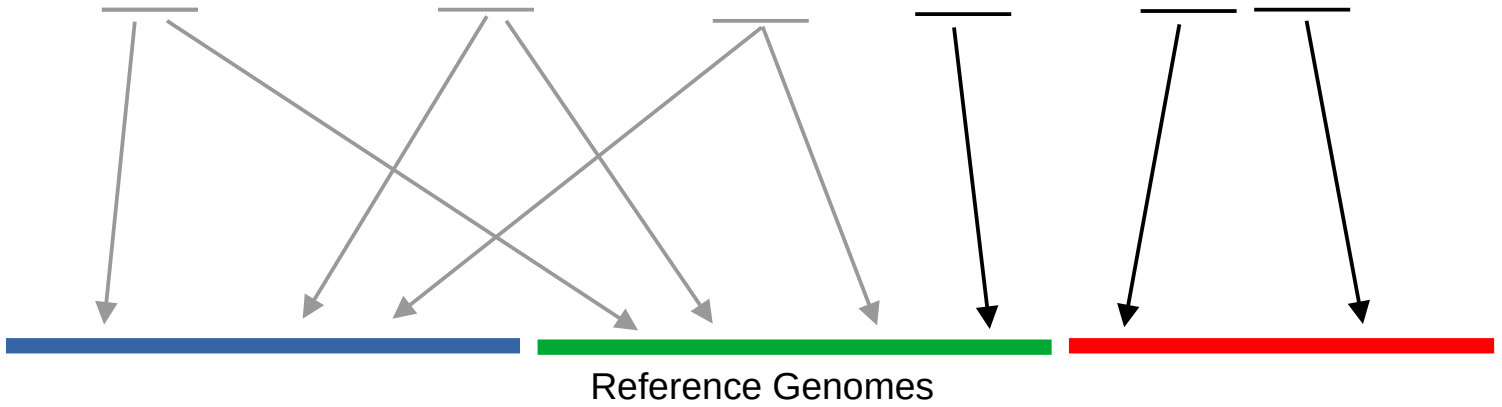
How do you measure abundance?

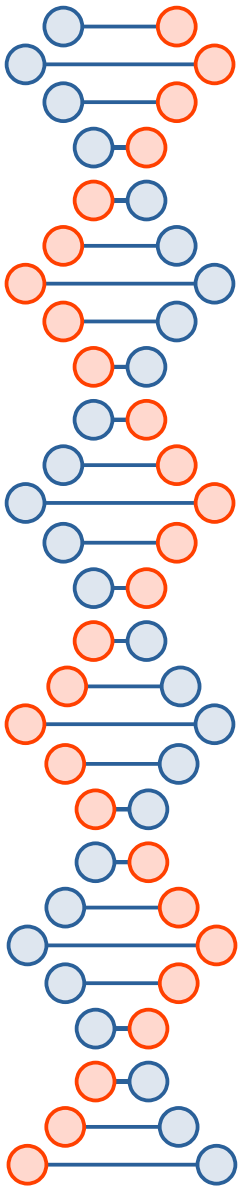
- Idea 1: Just map reads to the genome that they match and count them
  - Problem: Some reads map to multiple genomes



Ambiguous Reads

- Some reads are ambiguous: they map to multiple genomes. This is a big problem if you're dealing with multiple strains of a species.
- Idea 2: Map and count reads but ignore ambiguous reads
  - Problem: what if some genomes are very similar?
  - Why wouldn't partially assigning reads work?





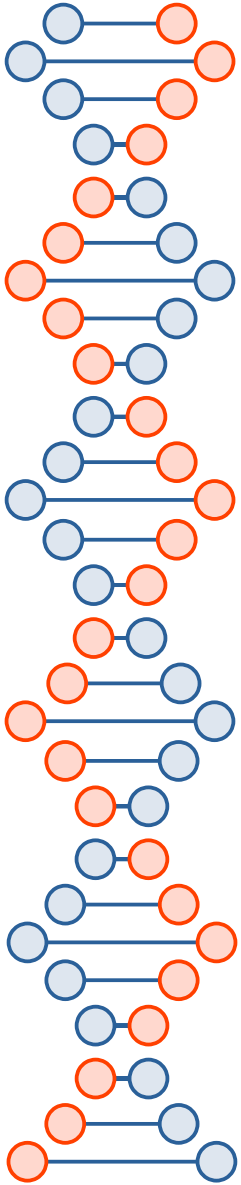
Read Normalization

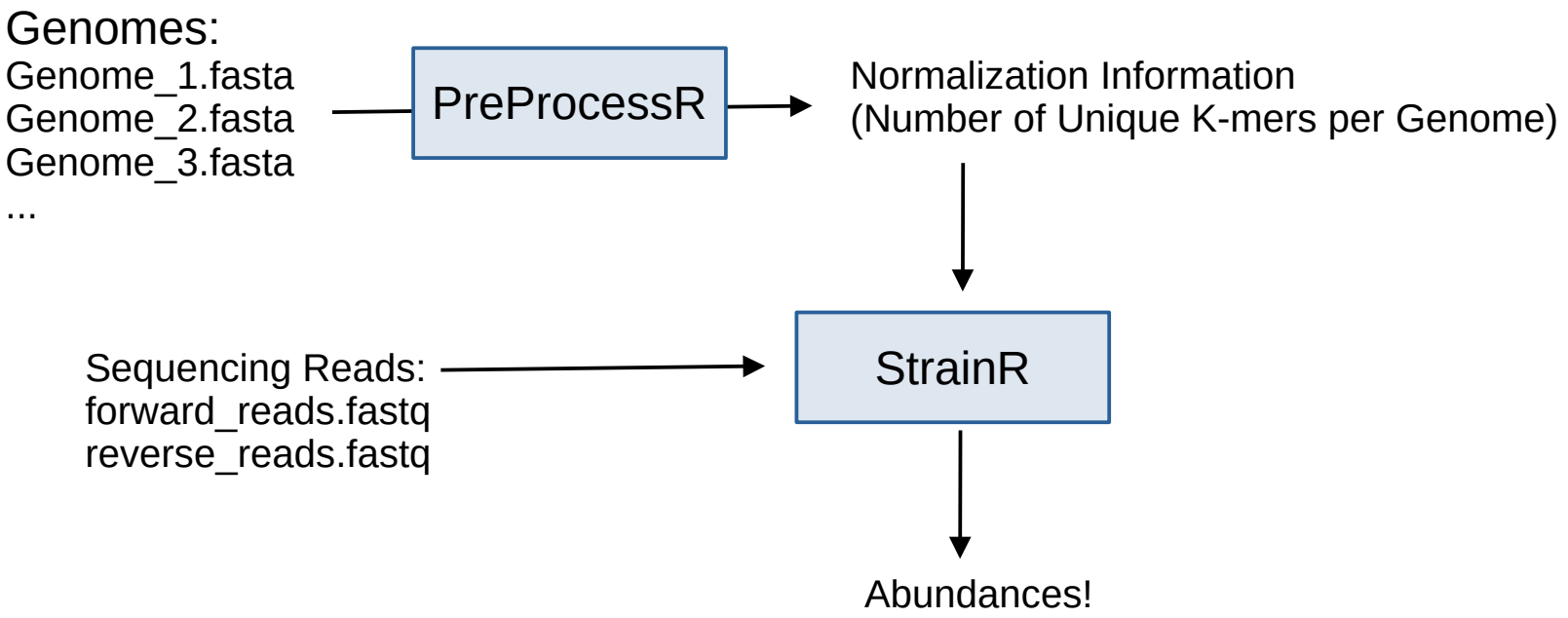
- Uniquely mapped reads are biased towards unique genomes
- Idea 3: Normalize unique reads by “uniqueness” of genomes
  - i.e. how many k-mers of the genomes are unique?
- This is what StrainR2 does. The hard part is computing it fast enough that it’s viable

$$FPKM = \frac{Mapped\ Fragments}{(Kilobases\ in\ Genome) \cdot (Millions\ of\ Total\ Fragments)}$$

$$FUKM = \frac{Mapped\ Fragments}{(Thousands\ of\ Unique\ K-mers) \cdot (Millions\ of\ Total\ Fragments)}$$

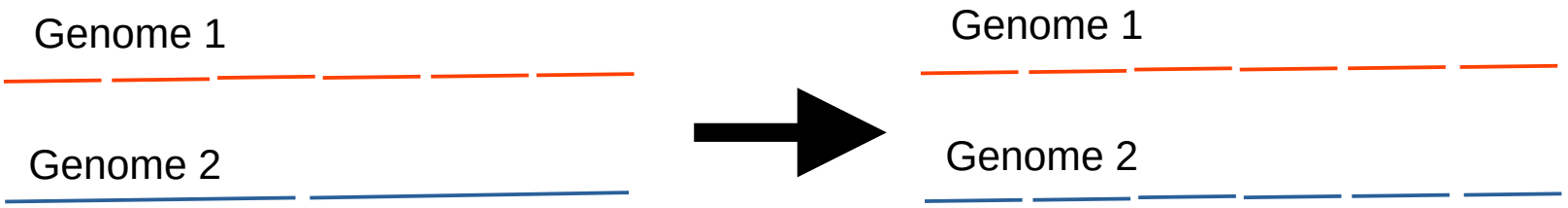
StrainR2 Inputs and Outputs





StrainR2 Caveats

- Genome qualities may differ and need to be equalized
- This greatly changes presence/absence detection, but not so much relative abundances
- StrainR2 uses the median FUKM of all contigs in a genome

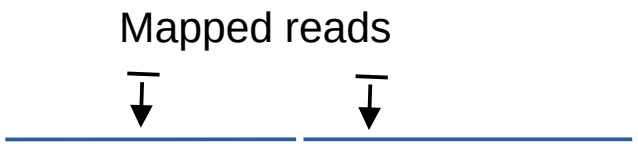


Genomes are made of contigs of varying sizes

StrainR breaks contigs into evenly sized “subcontigs”, ensuring there are no biases from genome quality

StrainR2 Caveats

- With low abundance organisms it can be hard to differentiate between read errors and actual presence
- Presence or Absence of low abundance organism can change depending on your chosen subcontig size
  - Remember median FUKM is used

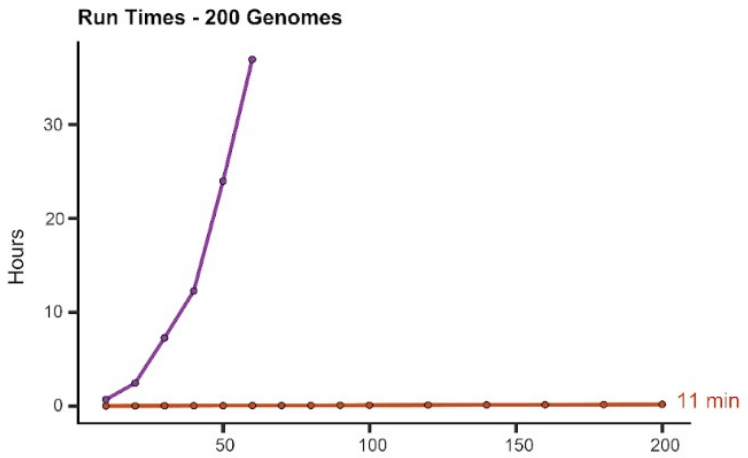


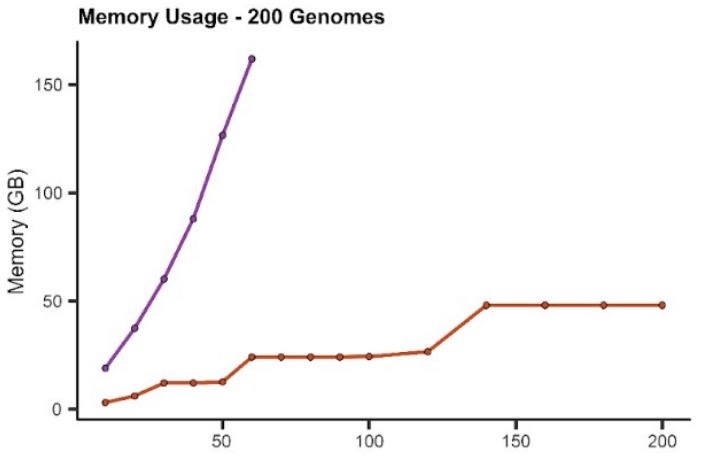
2/2 subcontigs have mapped reads (present)

2/6 subcontigs have mapped reads (absent)

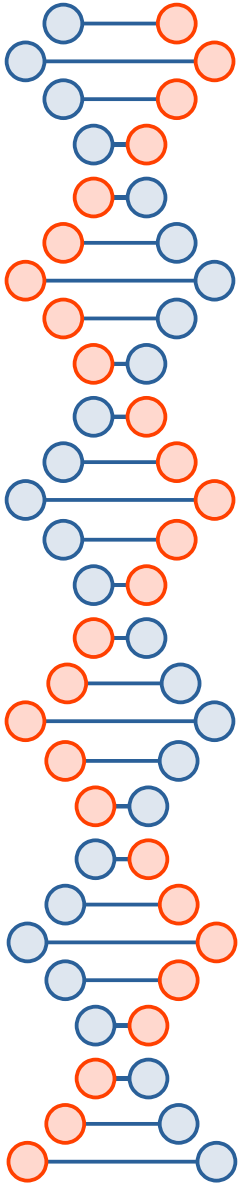
StrainR2 Caveats

- StrainR2 may get too memory hungry if you input hundreds of genomes (for now)

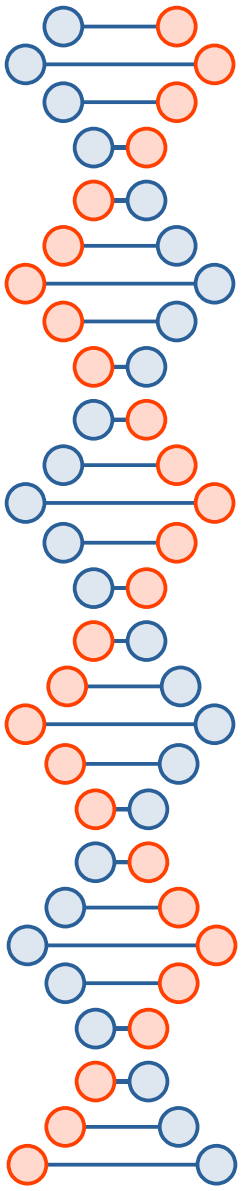




Red = StrainR2, Purple = StrainR1

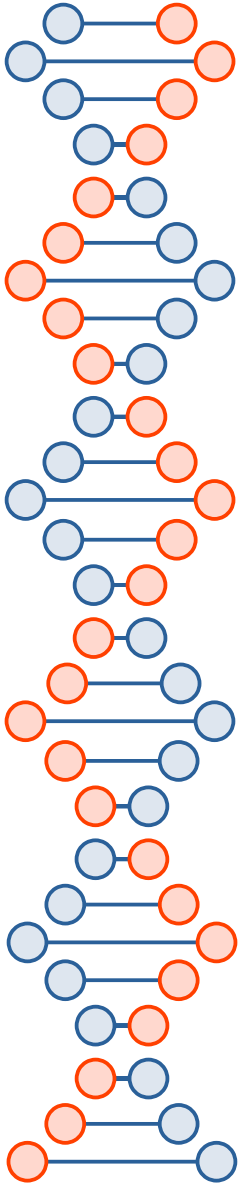


Questions before the demo?



Demo

- PreProcessR
  - Calculates Normalization values for a set of genomes
  - Run this once per community
- StrainR
  - Calculates FUKM
  - Run this once per sample

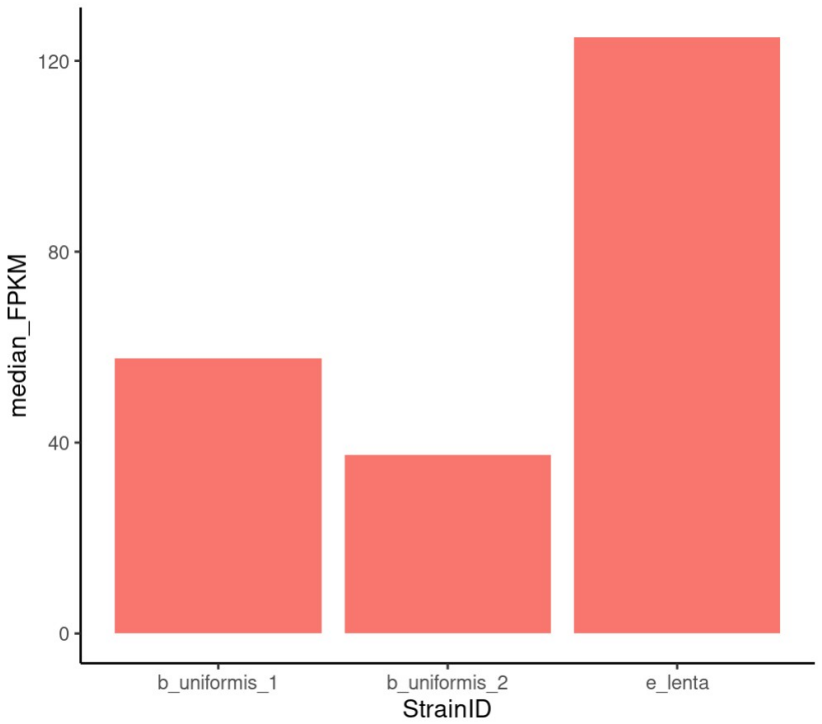


StrainR2 Interpretation

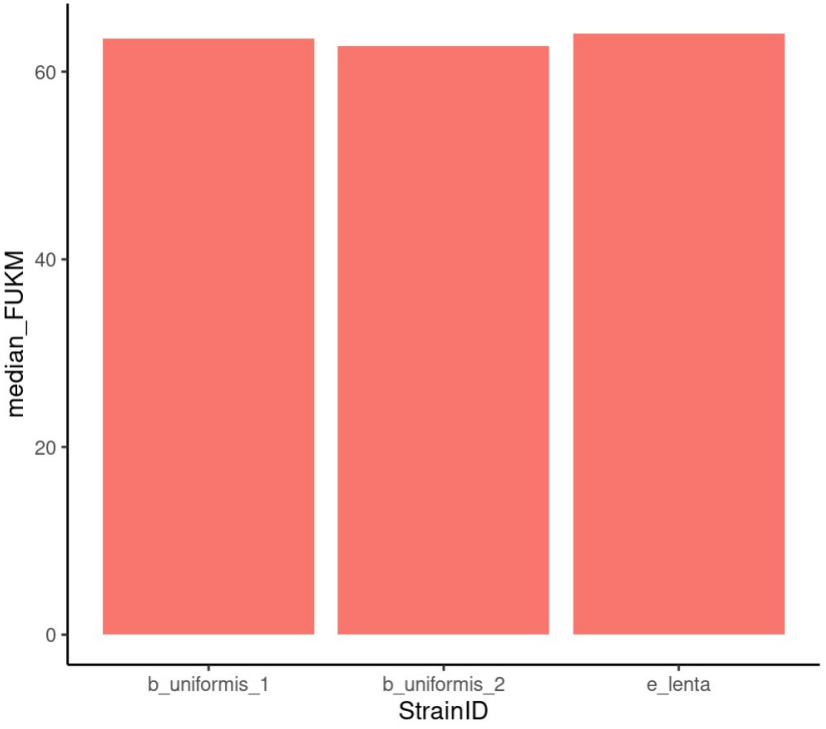
- How many subcontigs are marked present?
- Strain absence vs presence?
- It changes depending on your subcontig sizes: make conclusions carefully

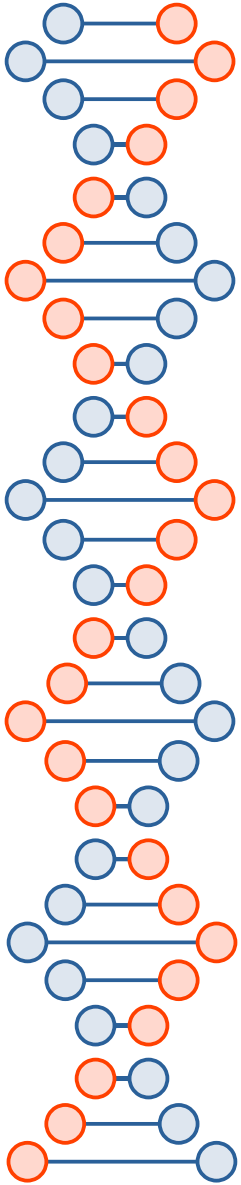
StrainR2 Results

unnormalized



normalized





Questions?