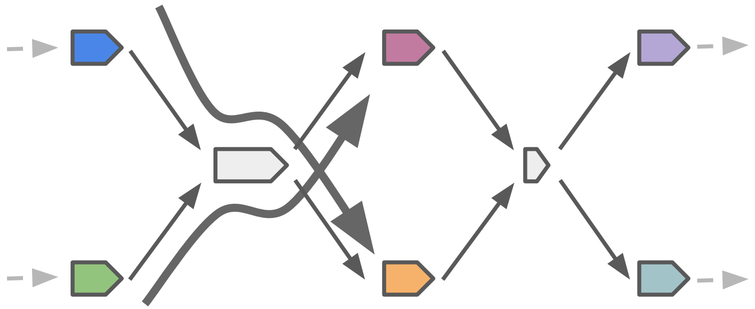
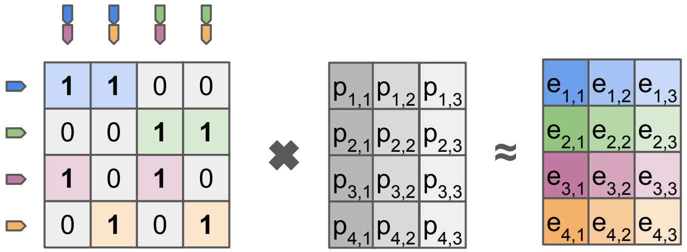
# Unzipping the metagenome:

## strain-level discovery in the gut microbiome



Byron J. Smith

Bhatt Lab Computational Subgroup
2024-09-10



$$\begin{bmatrix}1&1&0&0\\0&0&1&1\\1&0&1&0\\0&1&0&1\end{bmatrix} \times \begin{bmatrix}p_{1,1}&p_{1,2}&p_{1,3}\\p_{2,1}&p_{2,2}&p_{2,3}\\p_{3,1}&p_{3,2}&p_{3,3}\\p_{4,1}&p_{4,2}&p_{4,3}\end{bmatrix} \approx \begin{bmatrix}e_{1,1}&e_{1,2}&e_{1,3}\\e_{2,1}&e_{2,2}&e_{2,3}\\e_{3,1}&e_{3,2}&e_{3,3}\\e_{4,1}&e_{4,2}&e_{4,3}\end{bmatrix}$$

# First Thing: Thank You!

## Pollard Lab

Katie Pollard
Veronika Dubinkina
and *everyone*

## Collaborators

Archit Verma
Dylan Cable

## Funders

Gladstone Institutes
NIH
CZ Biohub
UC Noyce Initiative
Helmsley Charitable Trust

CHAN ZUCKERBERG BIOHUB

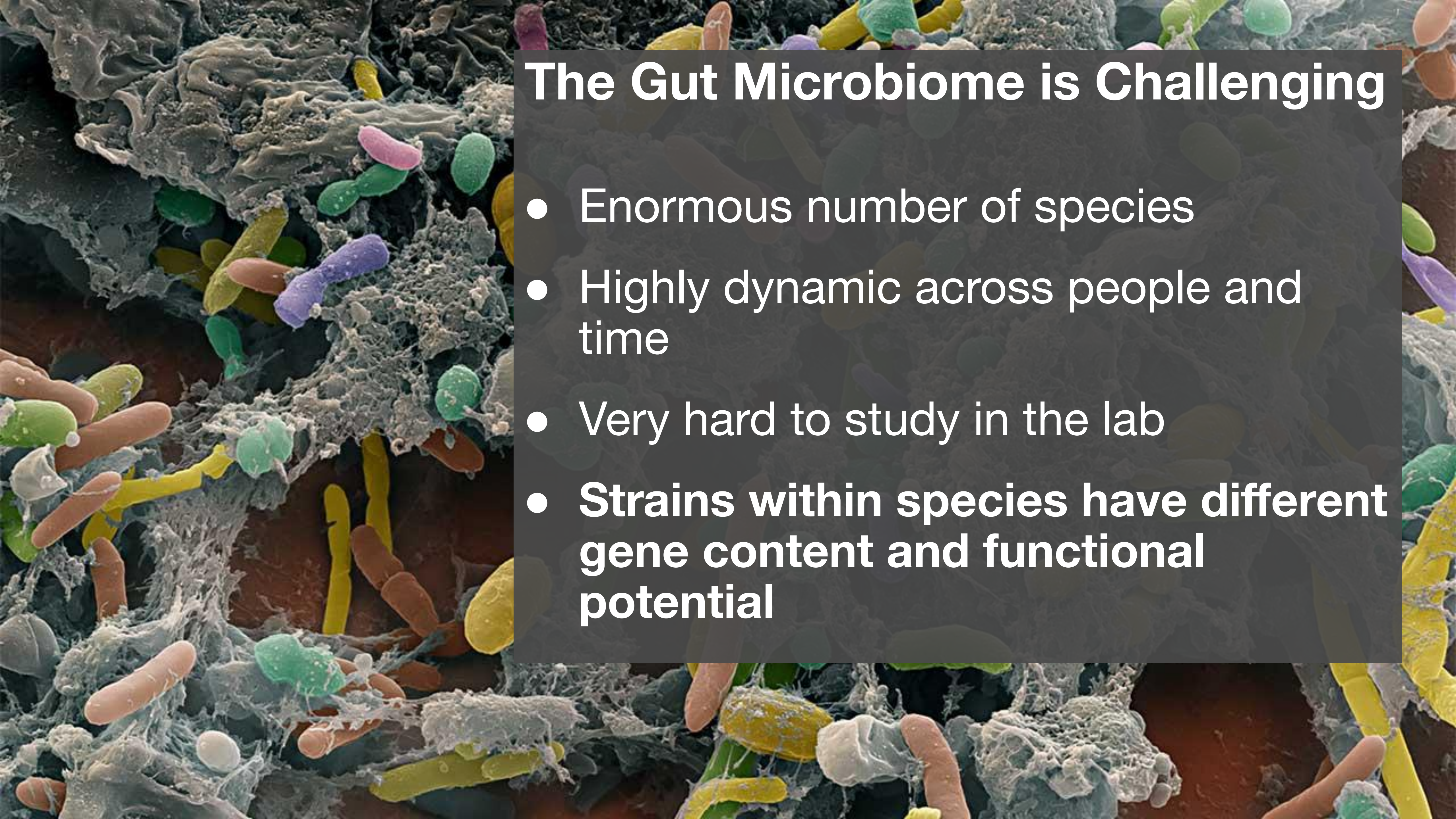
NIH National Institutes of Health

GLADSTONE INSTITUTES

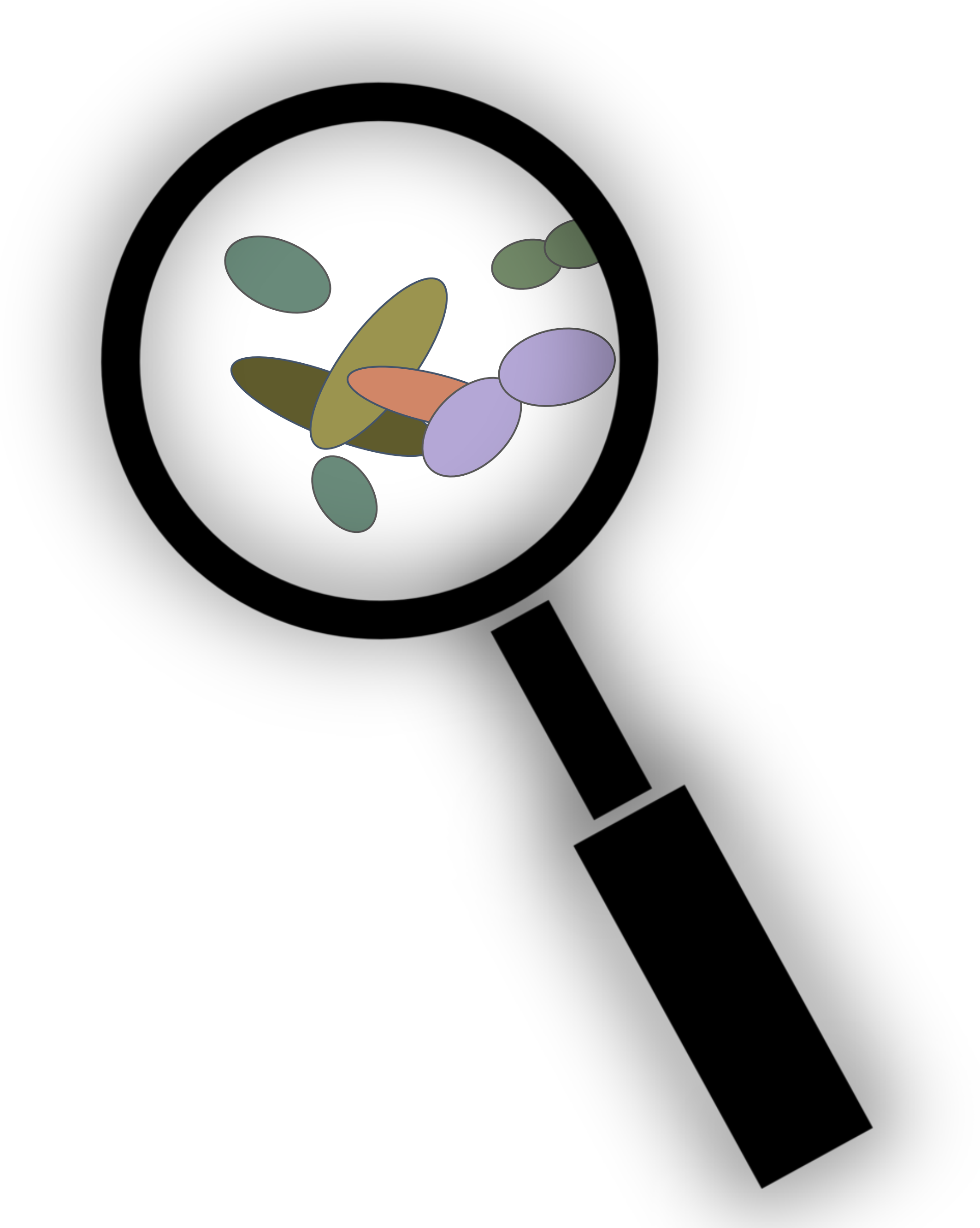
## Introduction:

The gut microbiome and shotgun metagenomics

# The Gut Microbiome is Challenging

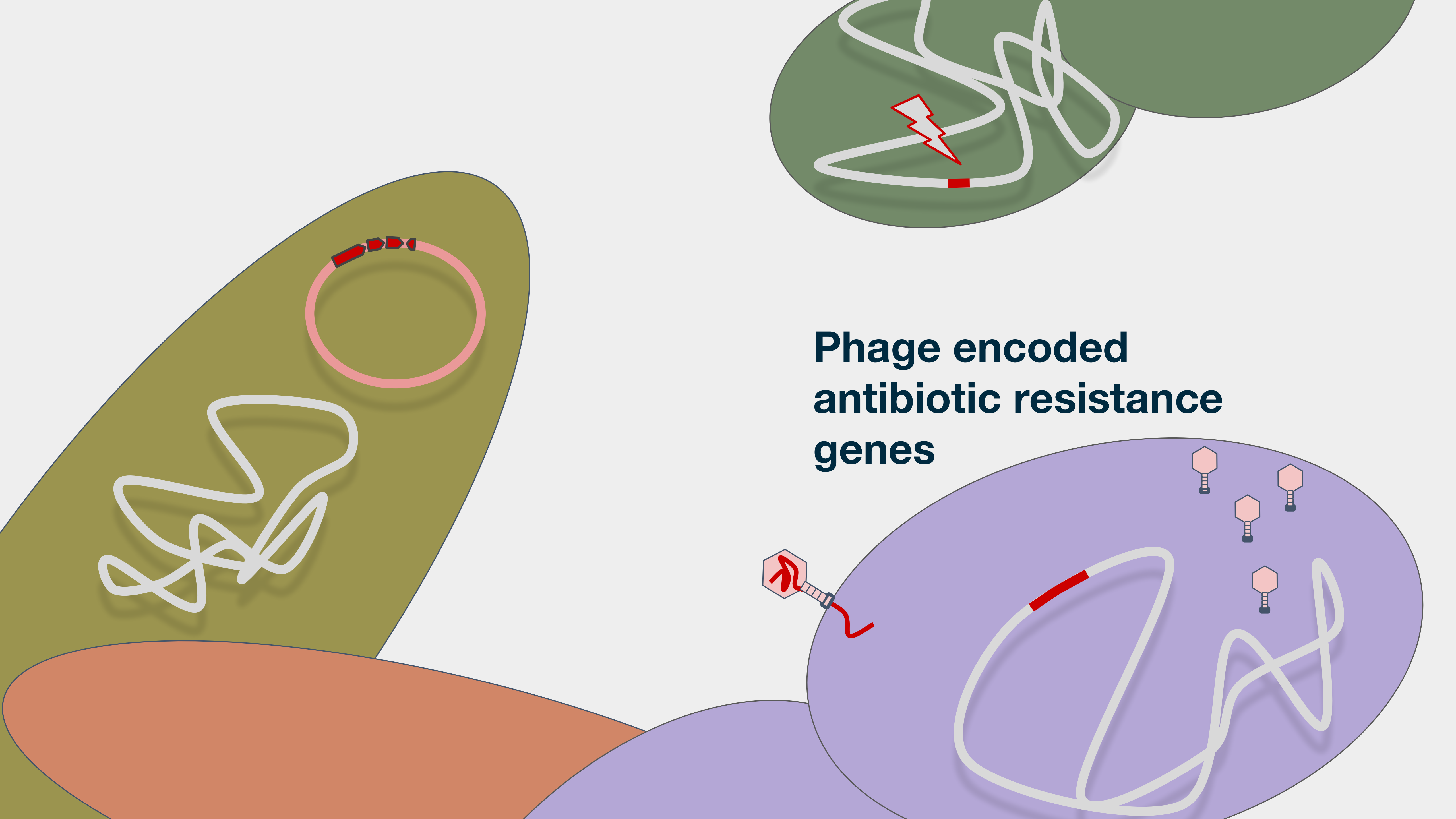


- Enormous number of species
- Highly dynamic across people and time
- Very hard to study in the lab
- **Strains within species have different gene content and functional potential**

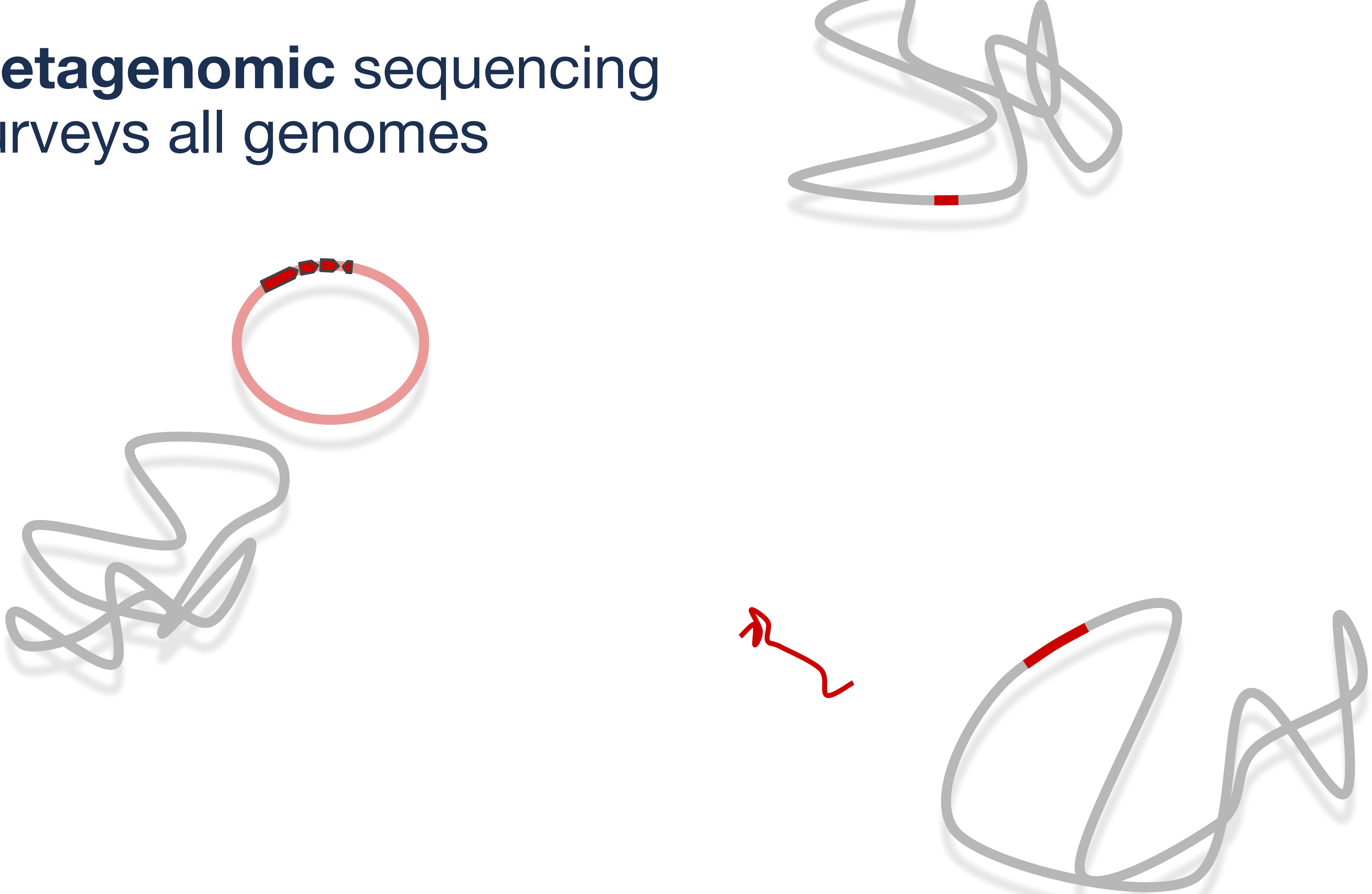


Bacterial genomes are key to understanding strain diversity

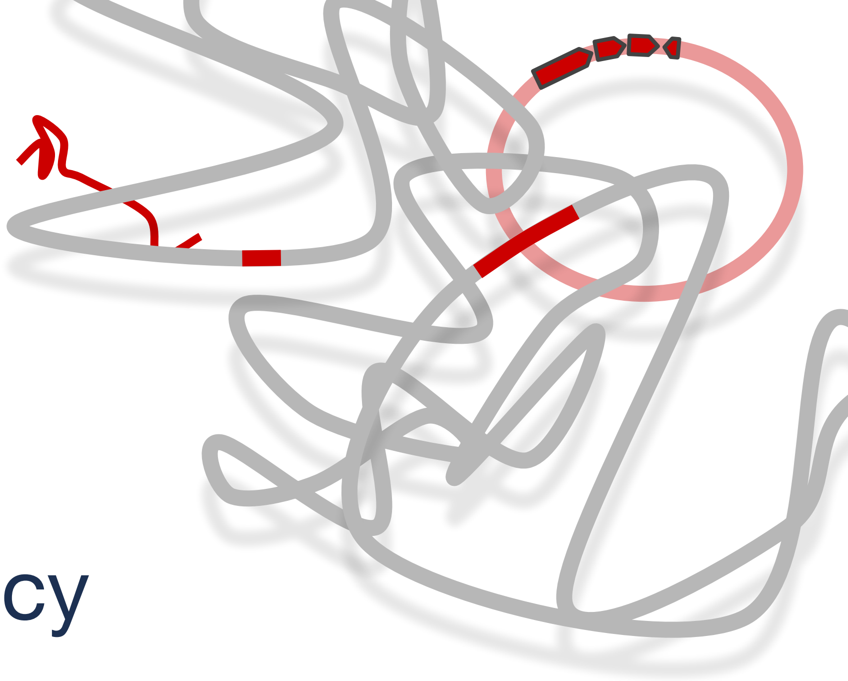
# Phage encoded antibiotic resistance genes



# **Metagenomic** sequencing surveys all genomes



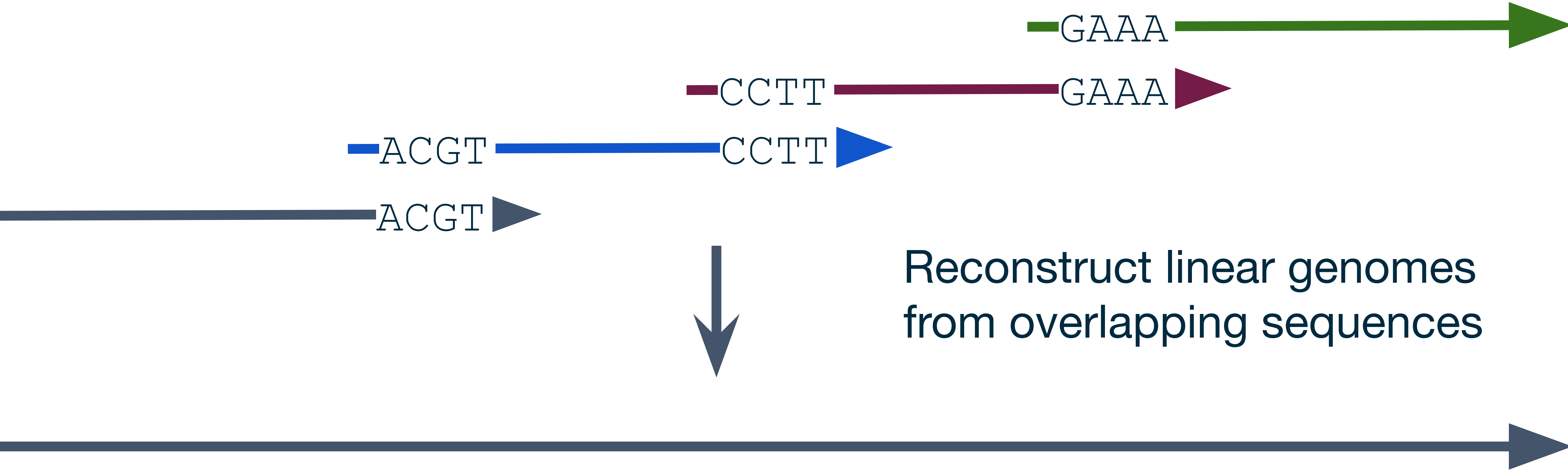
# Short-read, shotgun metagenomes enable modern microbiome science



Requirements:

- strain-resolved genome sequences ➢ high accuracy
- capture low-abundance organisms ➢ very deep sequencing
- longitudinal designs and lots of samples ➢ cheap
- long sequences ➢ …

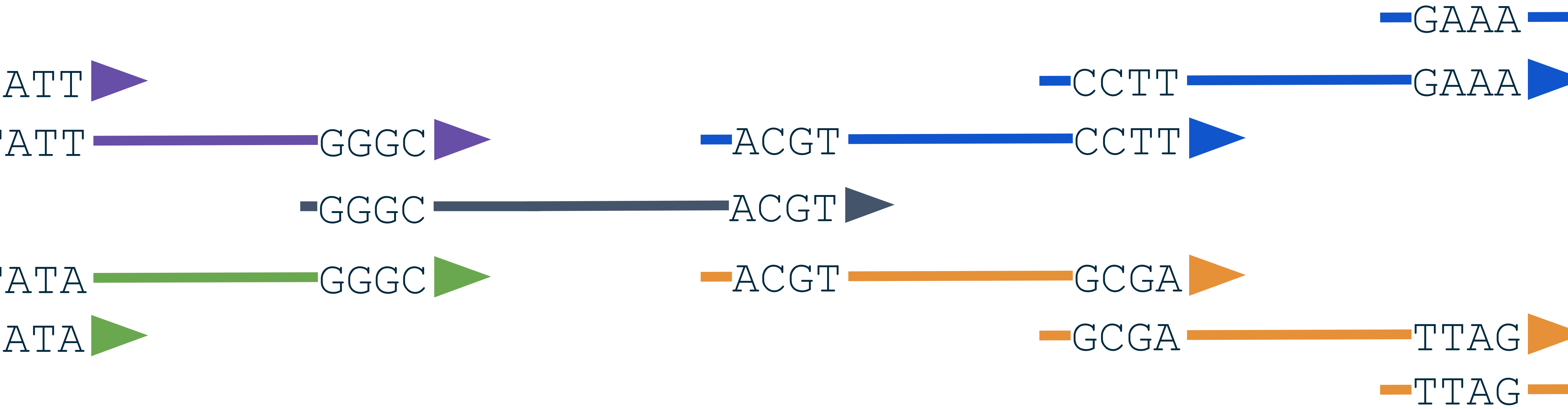
# Turning short reads into long sequences



Reconstruct linear genomes from overlapping sequences

...GGTAGAGCGTGGACGTAGGGTTAACCTTAGAAAGCTAGAAAACCGCGCGCCCT...

# **Problem:** Closely related strains make read-chaining ambiguous



GAAA

ATT

CCTT GAAA

ATT GGGC

ACGT CCTT

GGGC ACGT

ATA GGGC

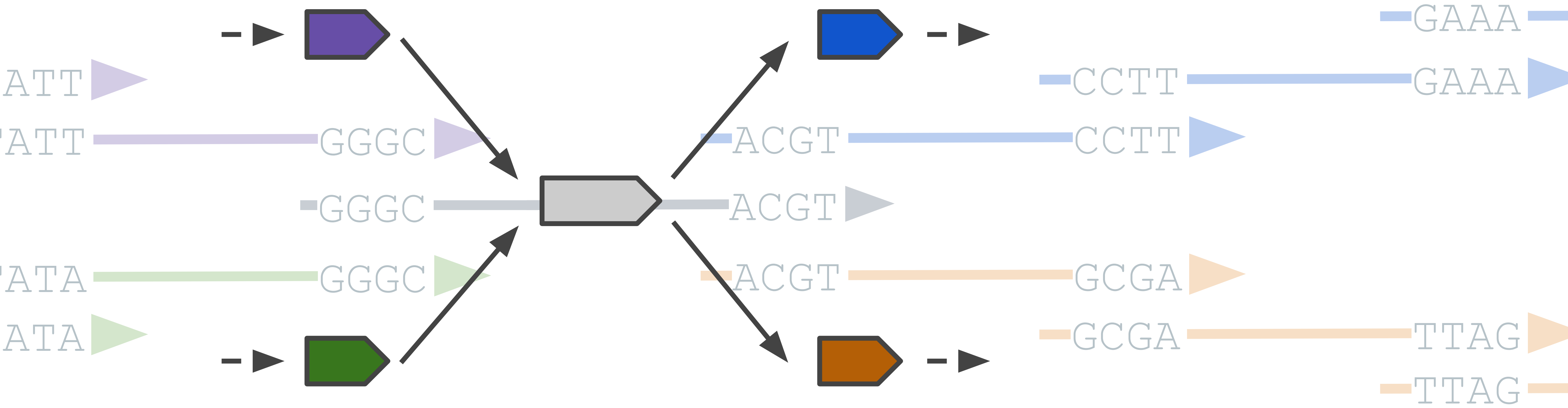
ACGT GCGA

ATA

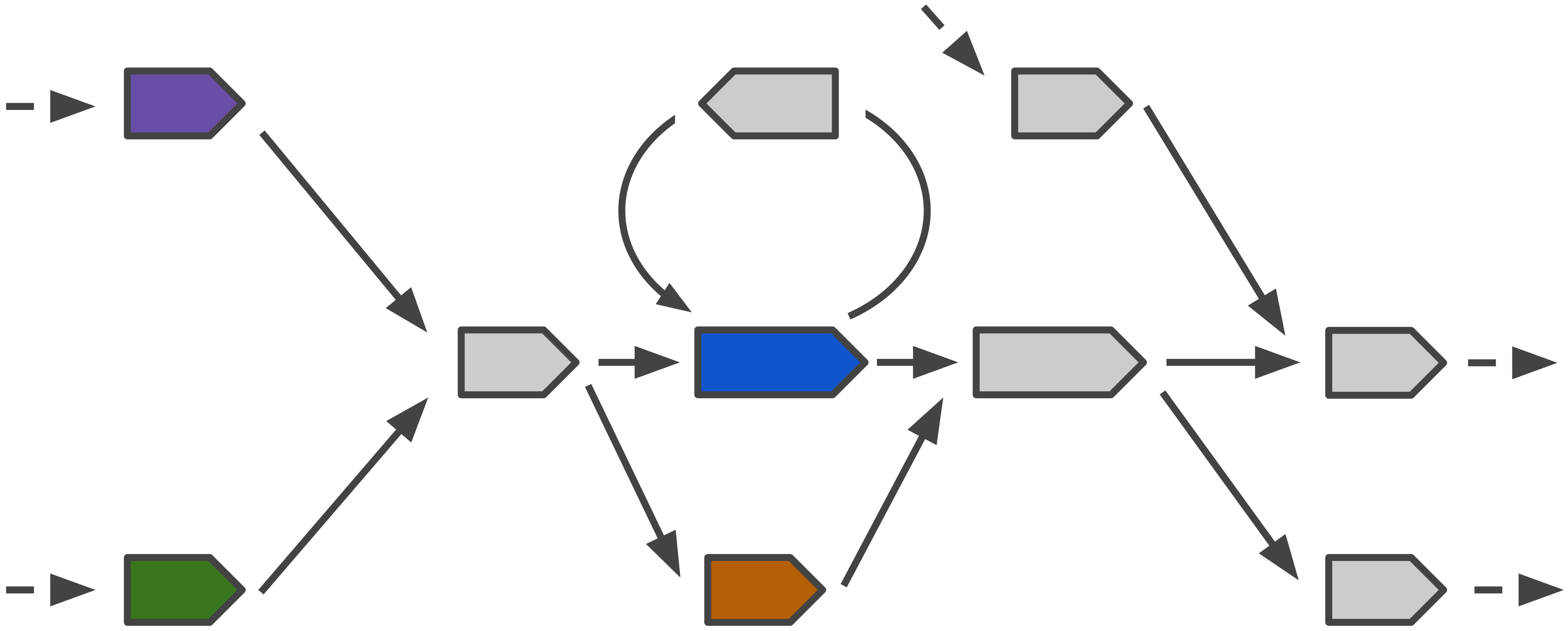
GCGA TTAG

TTAG

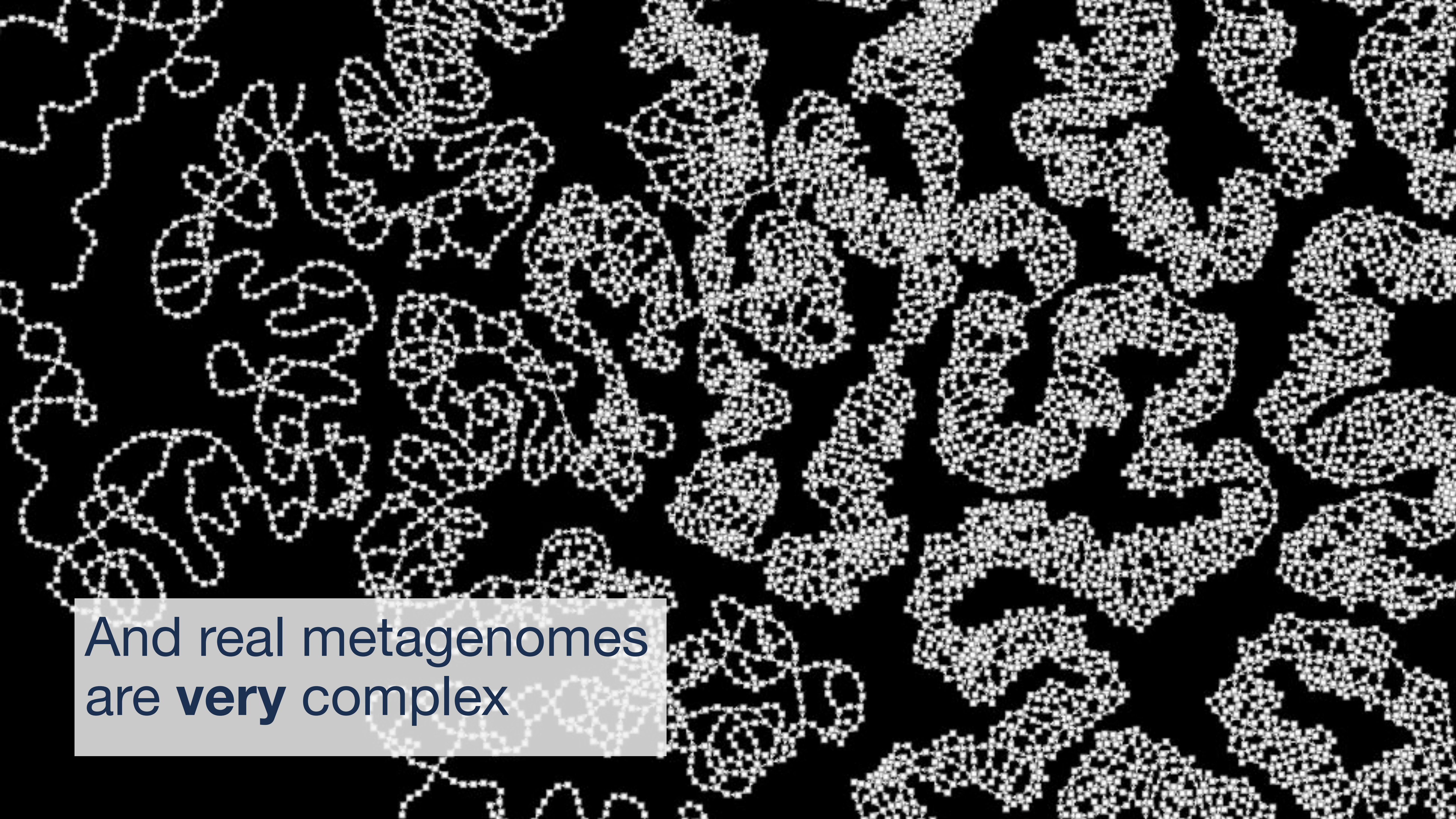
# Can be represented as a graph of sequences linked by their overlaps



# Can be represented as a graph of sequences linked by their overlaps

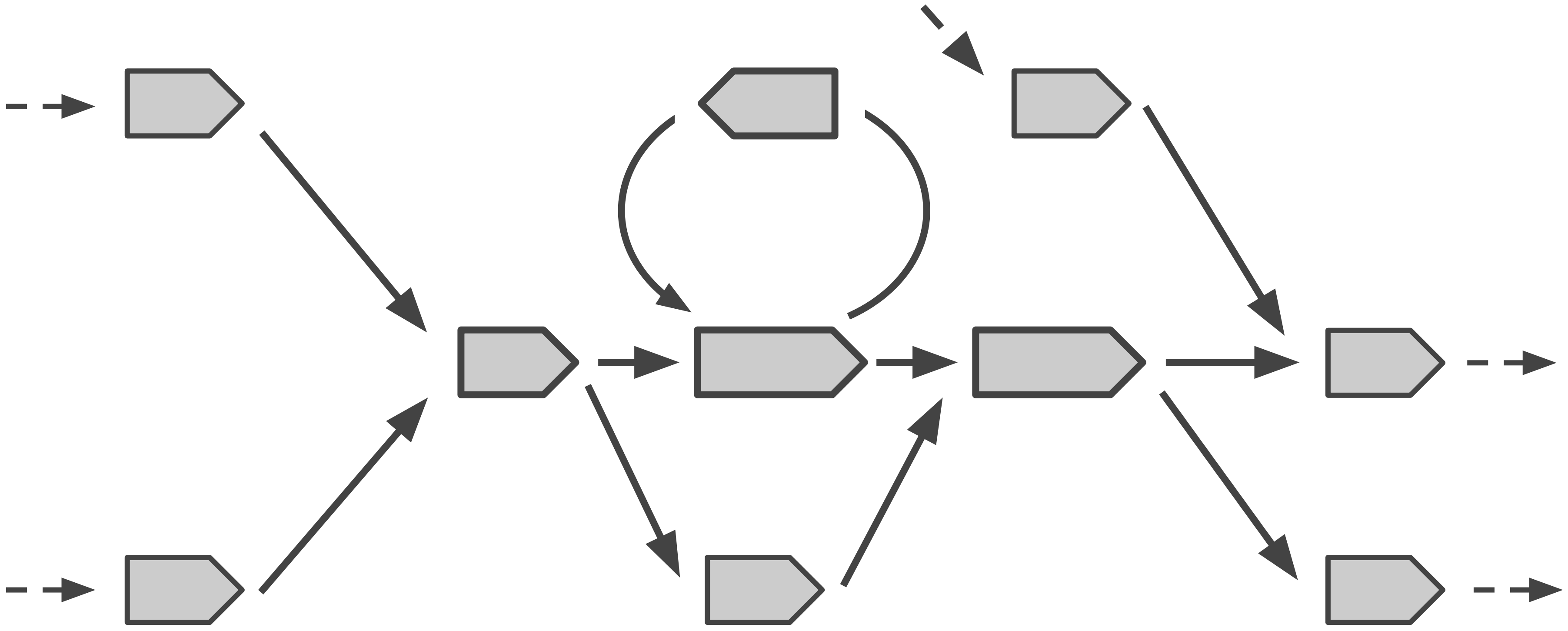


(This problem also comes up for mRNA alternative splicing)

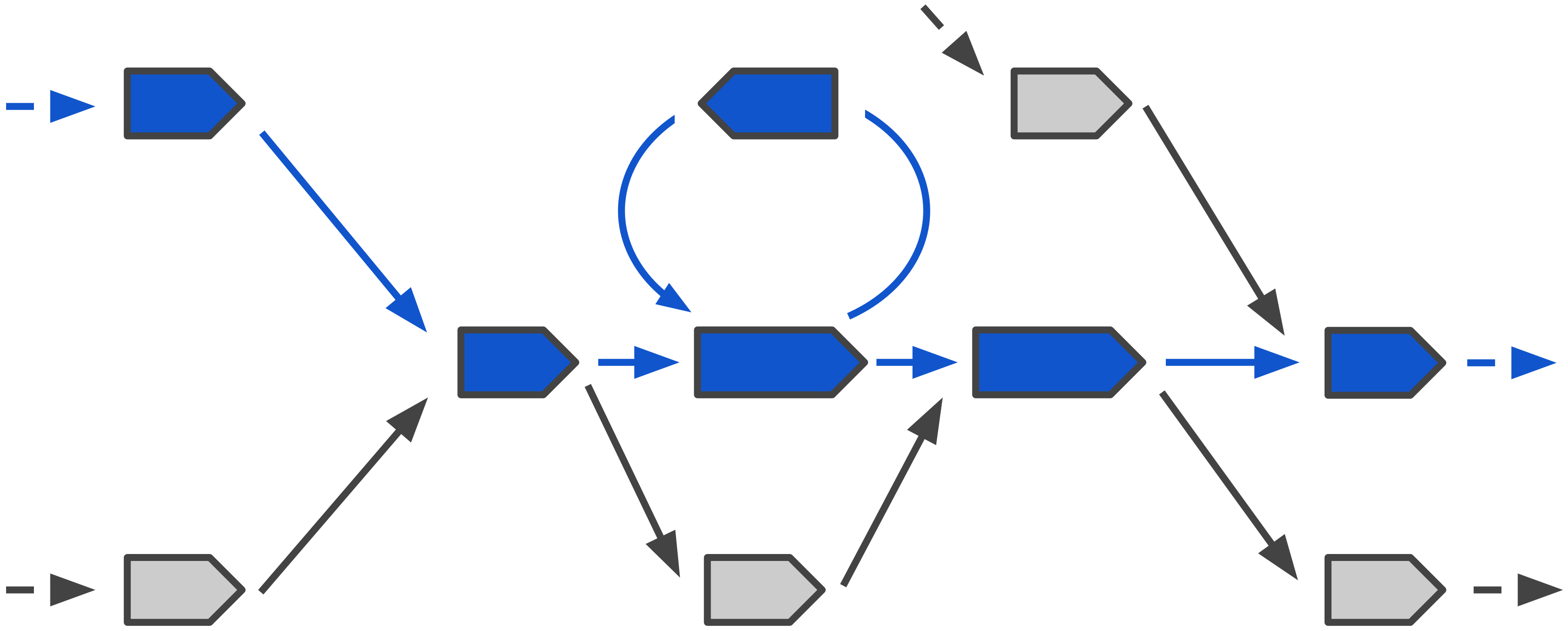


And real metagenomes are **very** complex

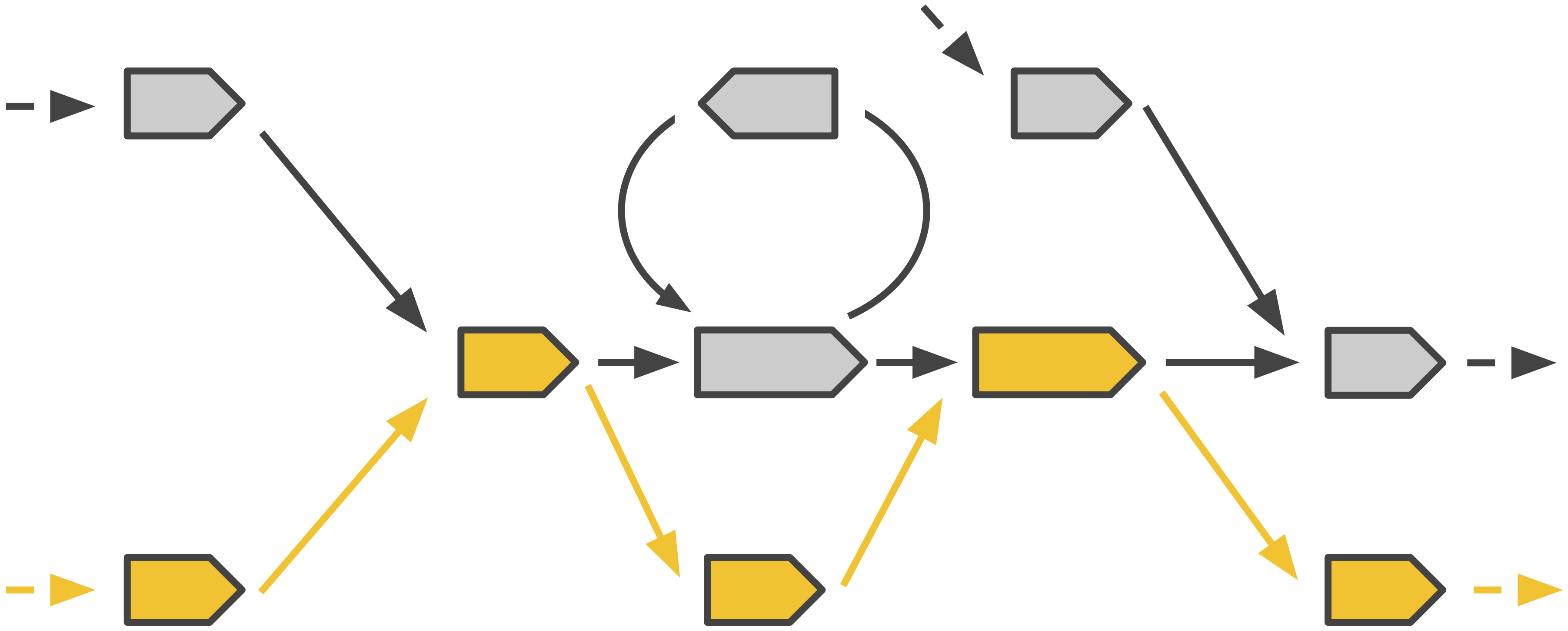
Real genomic sequences are paths on the graph



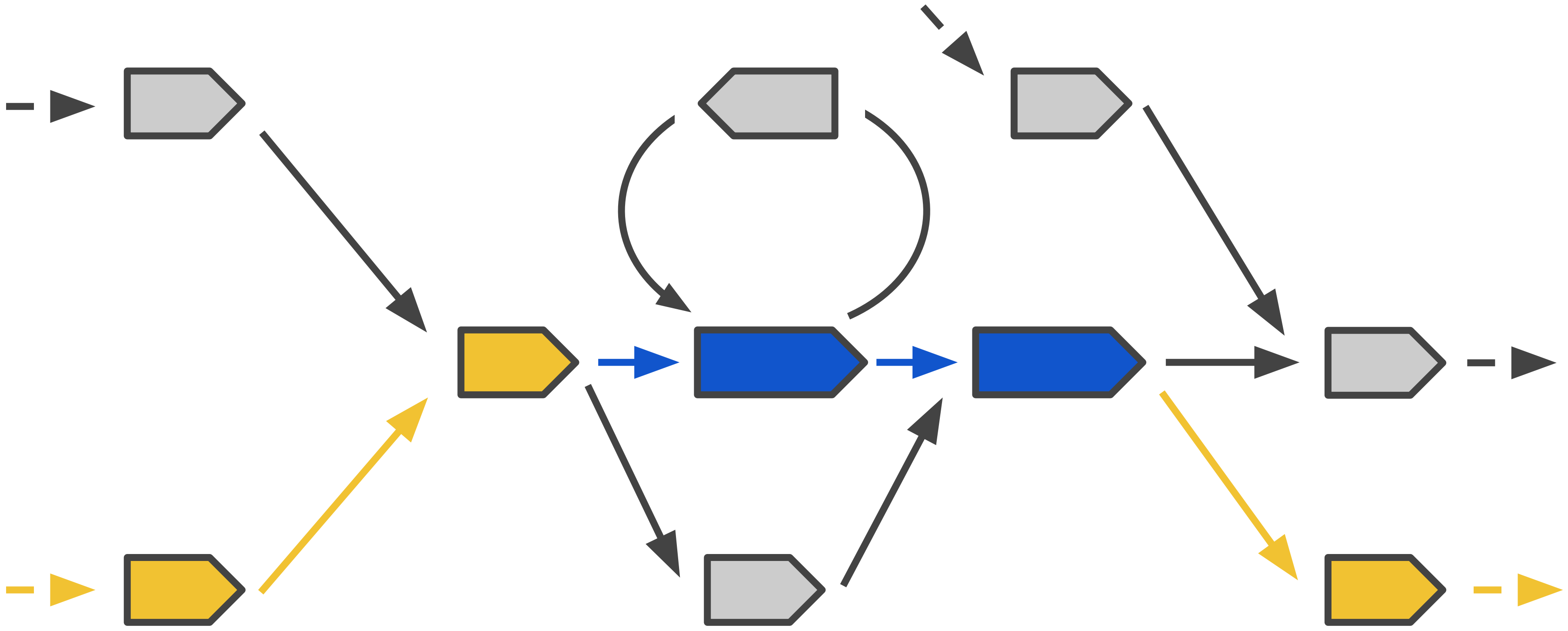
Real genomic sequences are paths on the graph



# Real genomic sequences are paths on the graph



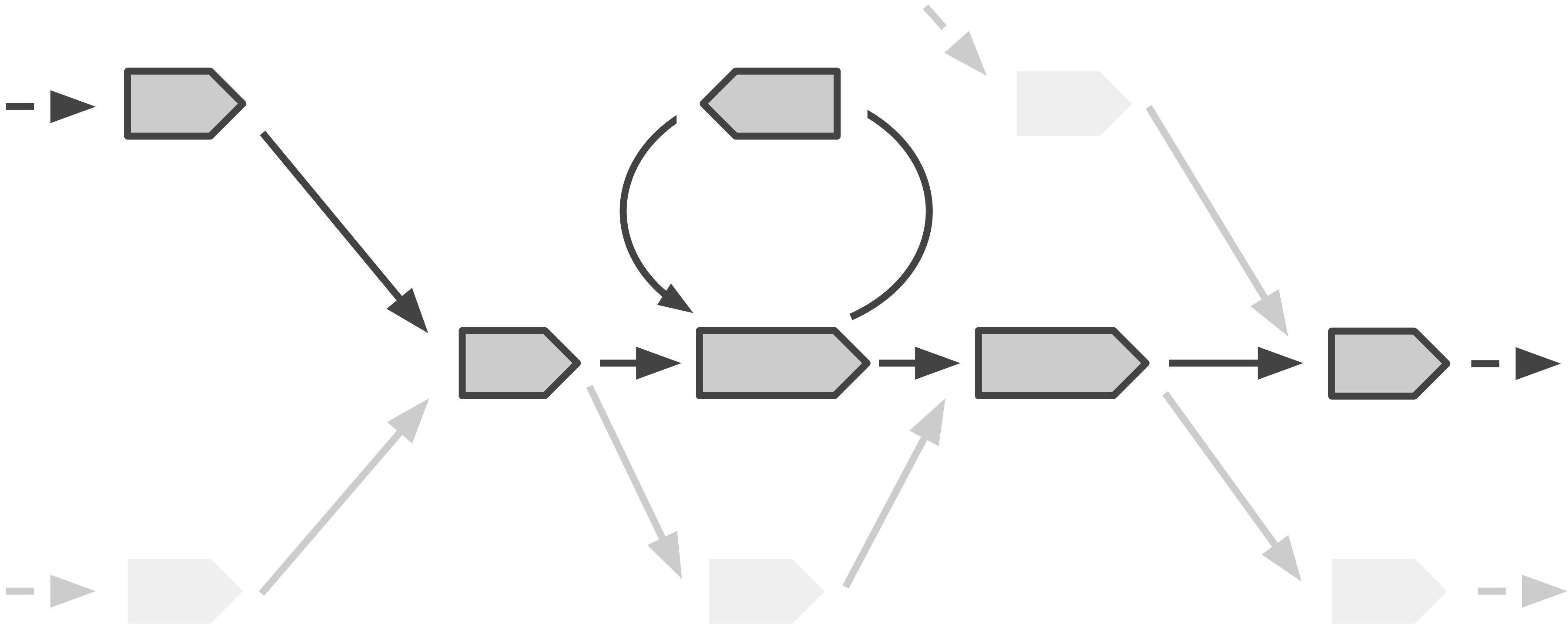
# Lots of incorrect paths also exist…
*How do we avoid these?*



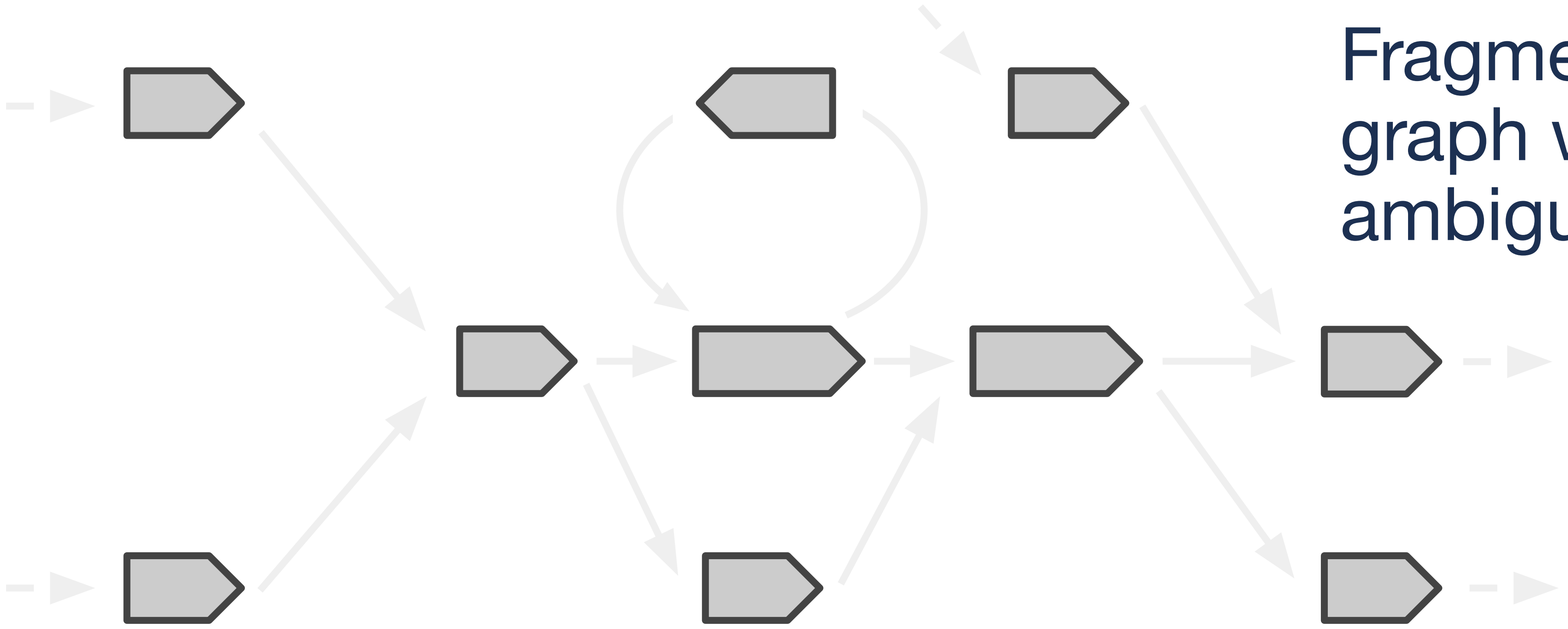
# Lots of incorrect paths also exist…
*How do we avoid these?*

**Standard Tools:**

Filter out low-abundance sequences



# Lots of incorrect paths also exist…
*How do we avoid these?*

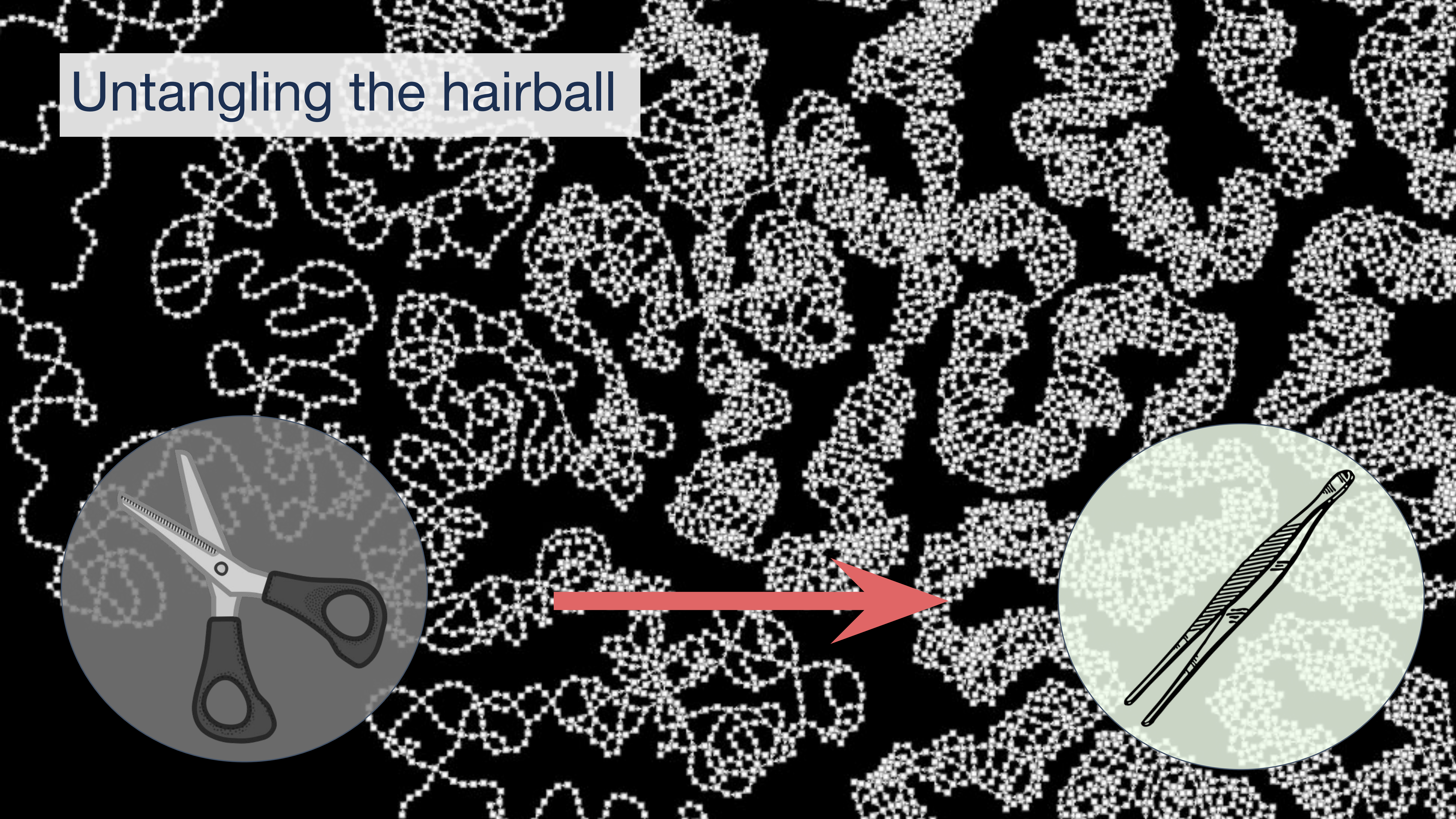


**Standard Tools:**

Filter out low-abundance sequences

Fragment the graph when it's ambiguous

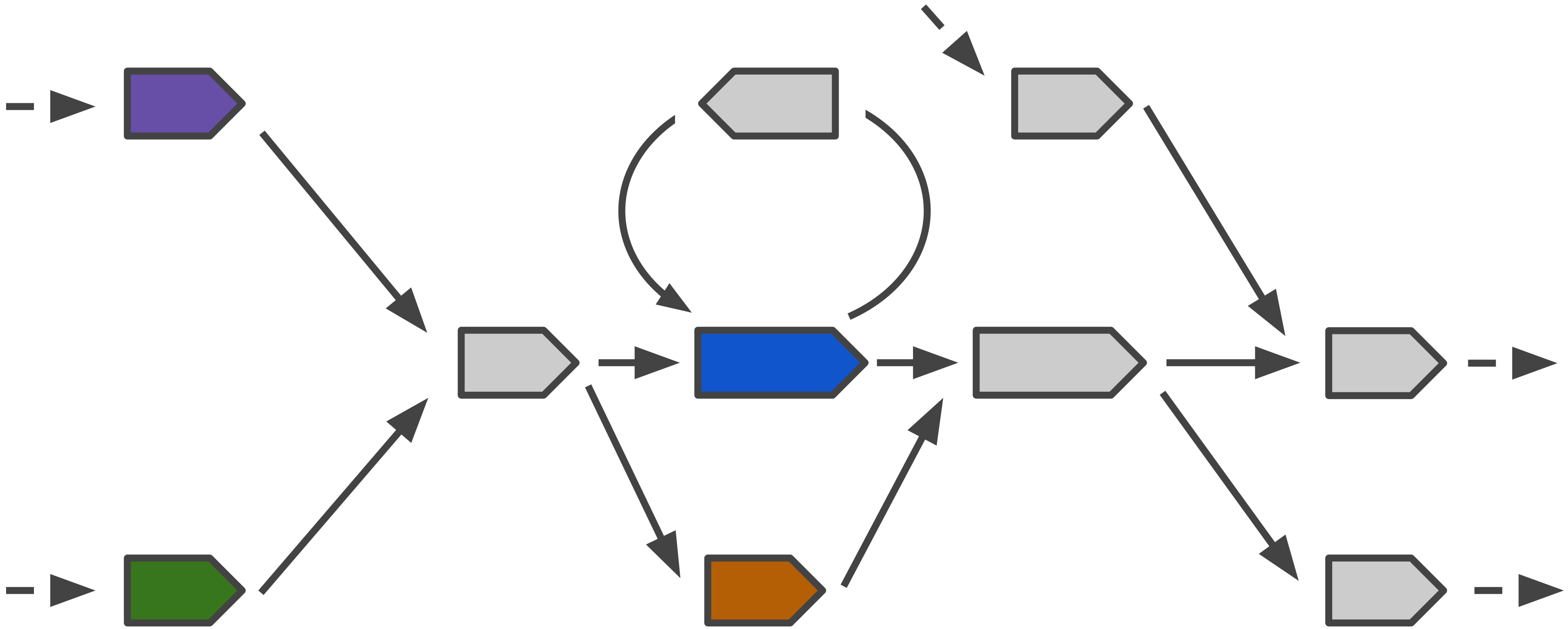
# Untangling the hairball



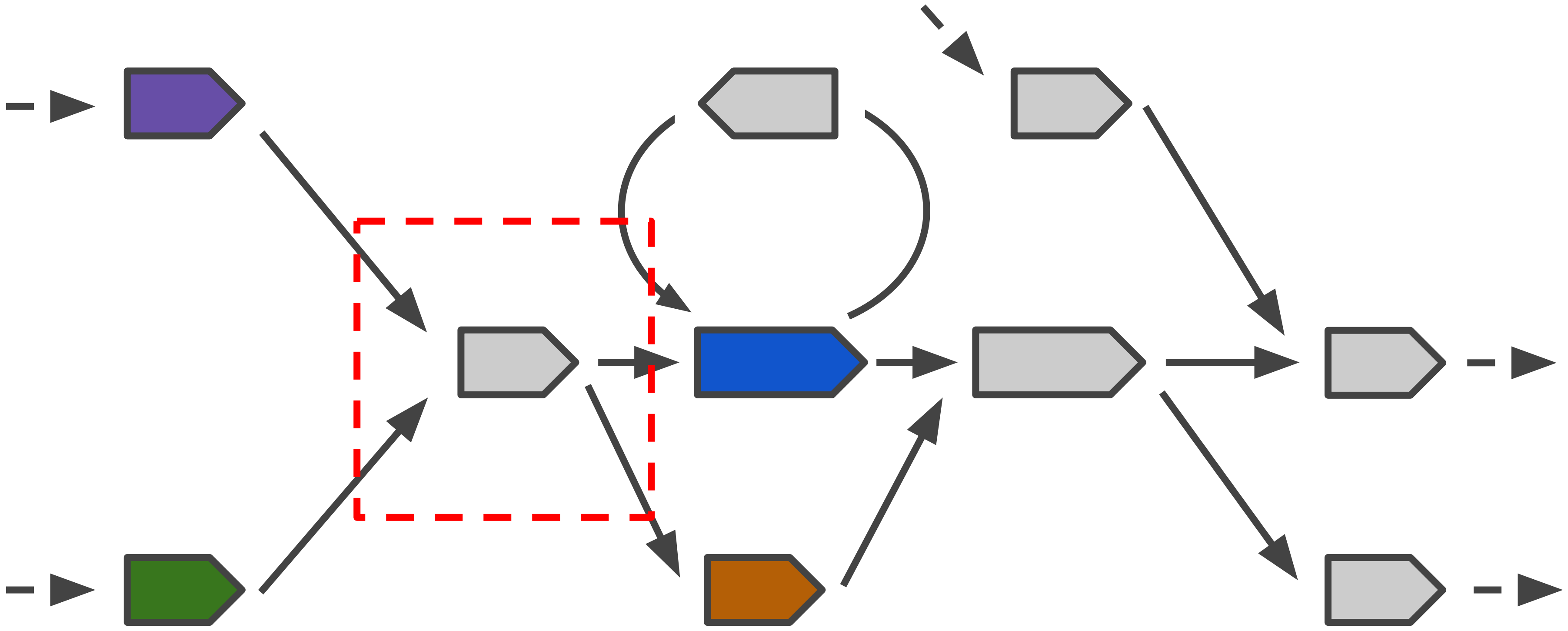
# StrainZip:

Untangling the metagenome graph

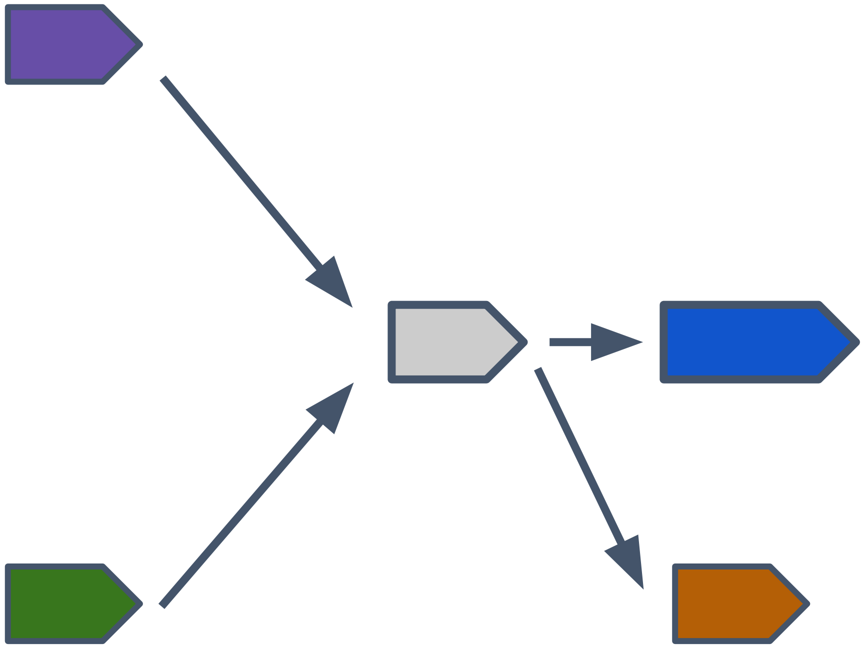
# How can we recover long, accurate genome sequences from short reads?



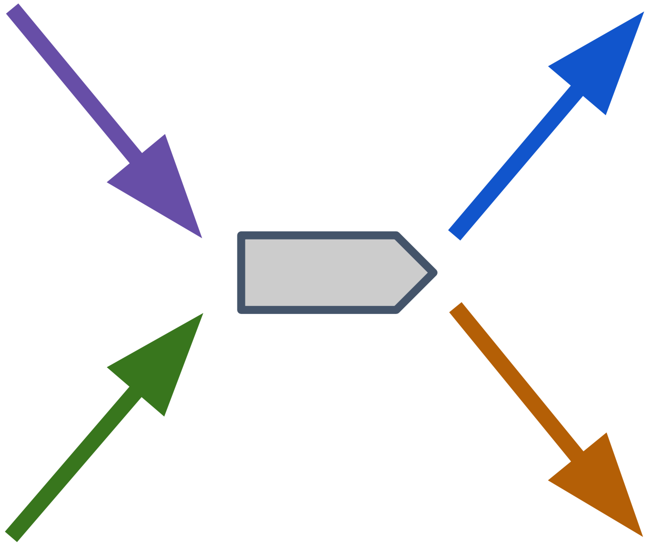
# How can we recover long, accurate genome sequences from short reads?



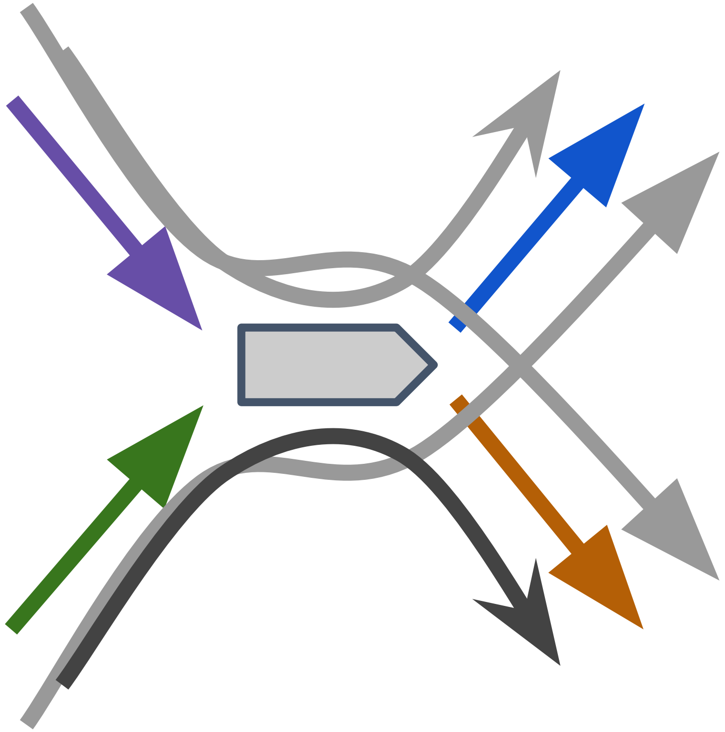
# Focus on just one junction at a time

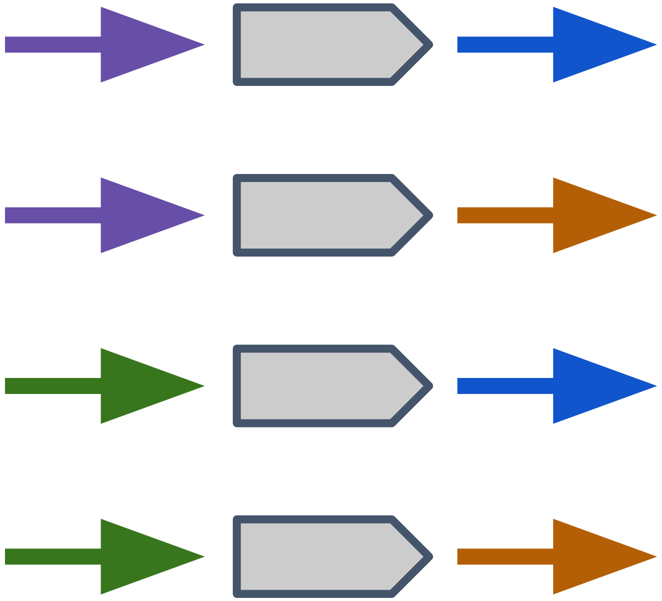


# Focus on just one junction at a time

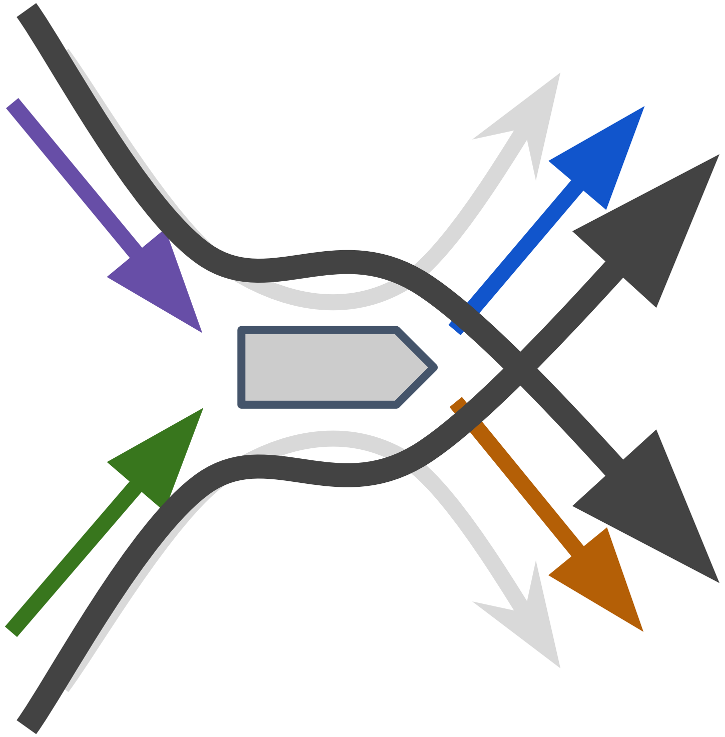


# Focus on just one junction at a time





# Focus on just one junction at a time
# Select local paths



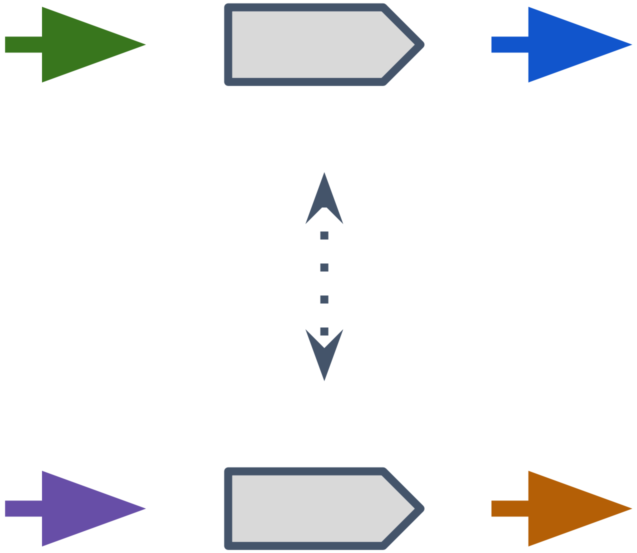
$$\underset{X}{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}} \times \underset{\beta}{\begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \\ p_{4,1} & p_{4,2} & p_{4,3} \end{bmatrix}} \approx \underset{Y}{\begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,1} & e_{3,2} & e_{3,3} \\ e_{4,1} & e_{4,2} & e_{4,3} \end{bmatrix}}$$

Sparse linear regression across multiple samples

Focus on just one junction at a time

Select local paths

Unzip

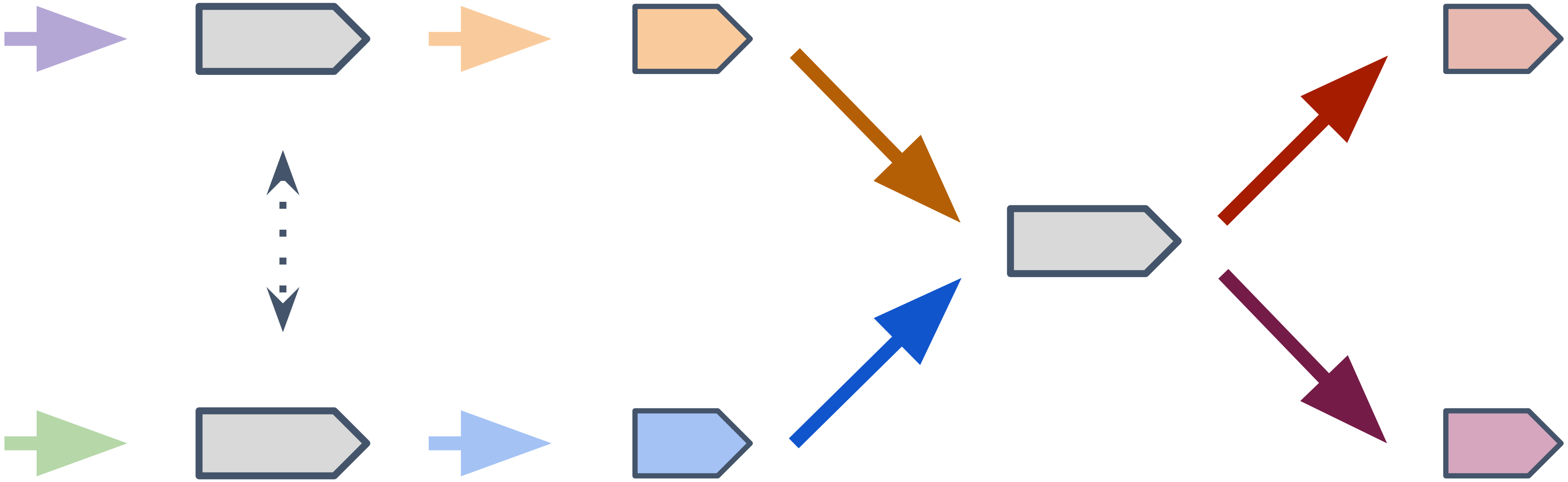


Focus on just one junction at a time

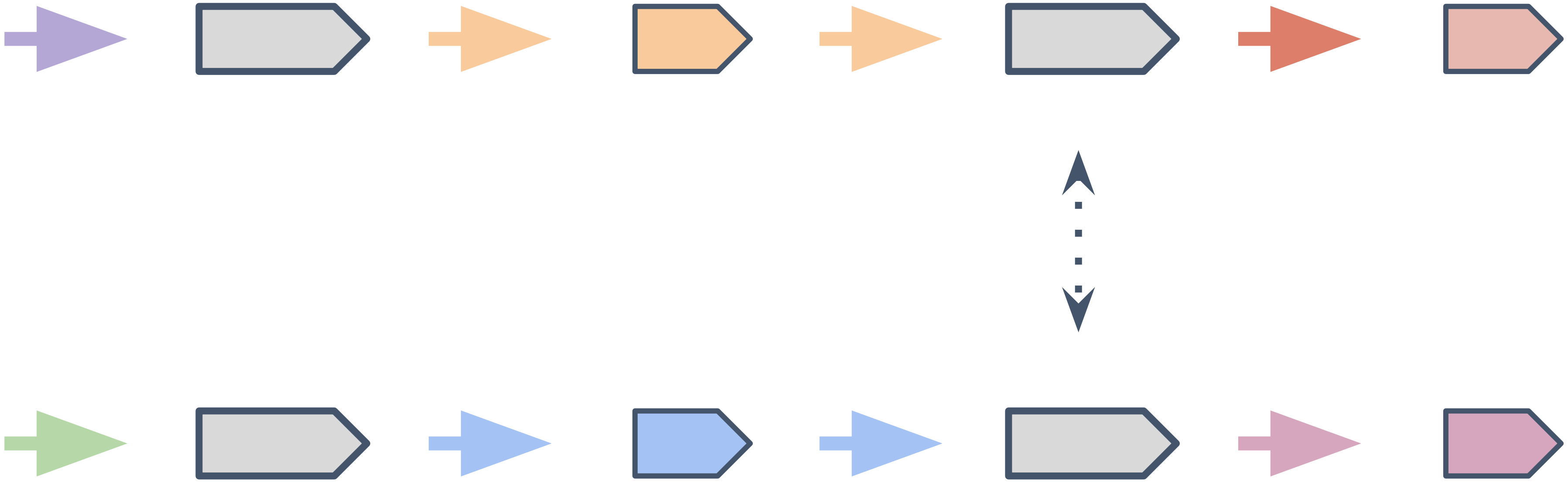
Select local paths

Unzip

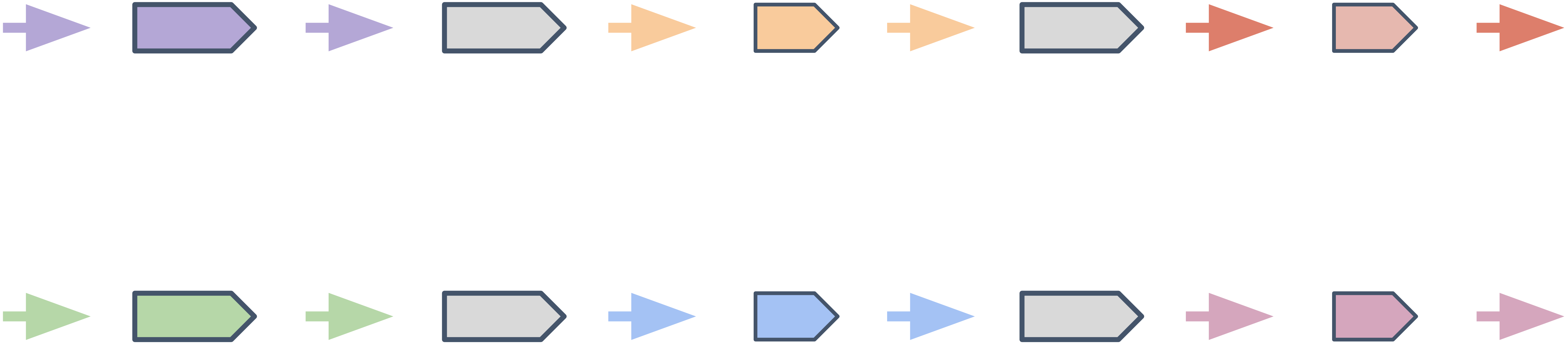
Repeat

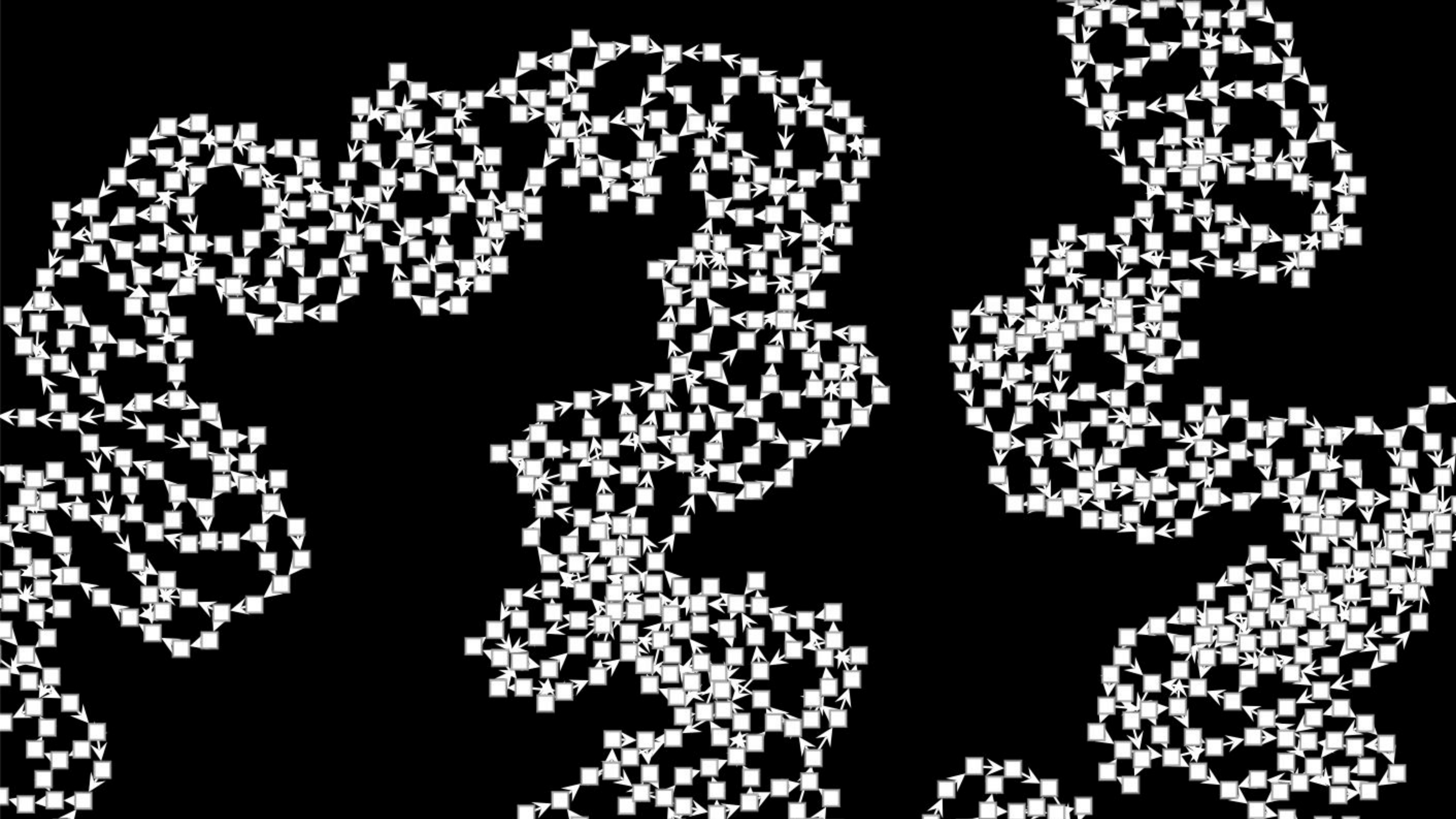


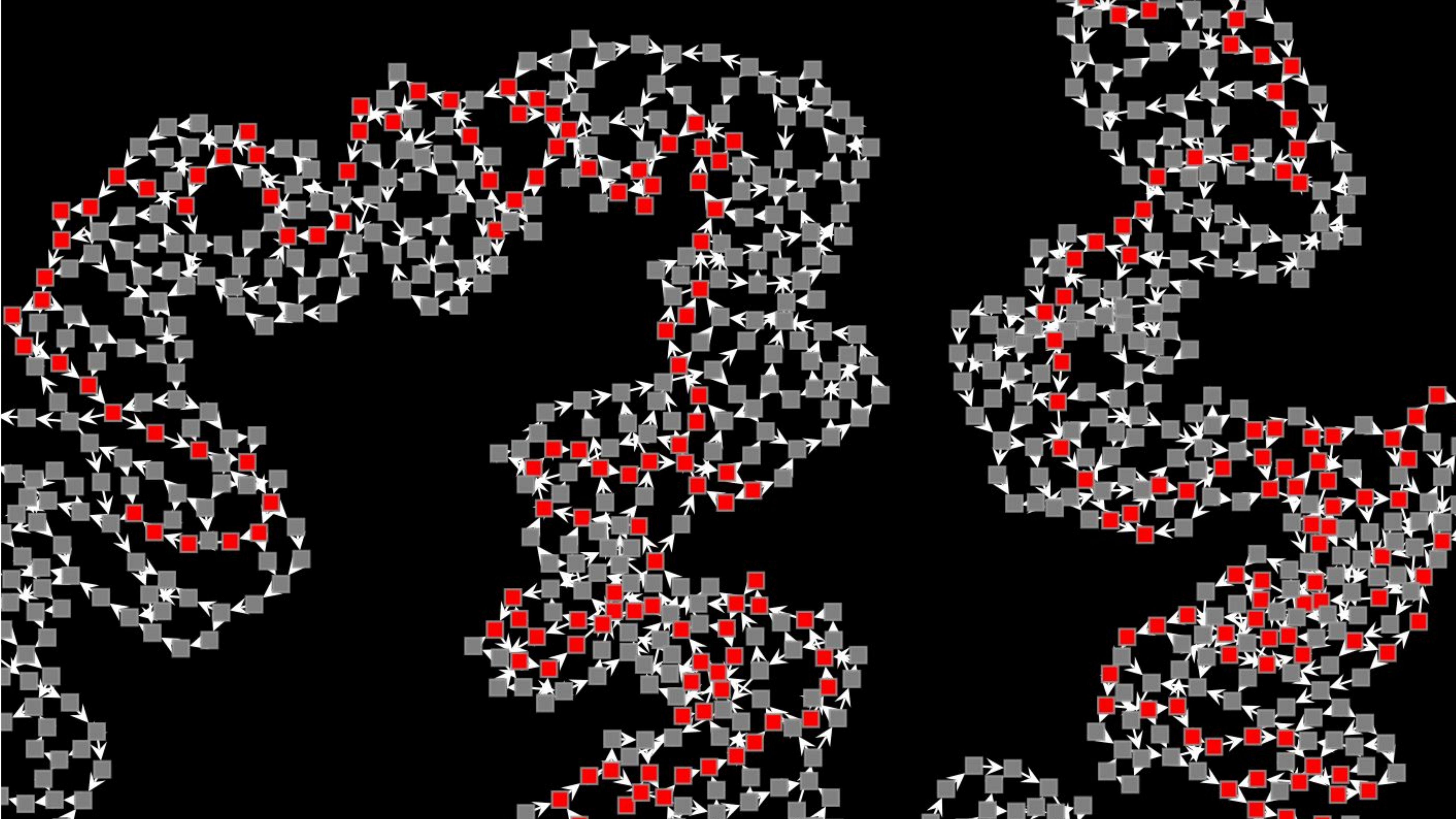
Focus on just one junction at a time
Select local paths
Unzip
Repeat



Focus on just one junction at a time
Select local paths
Unzip
Repeat

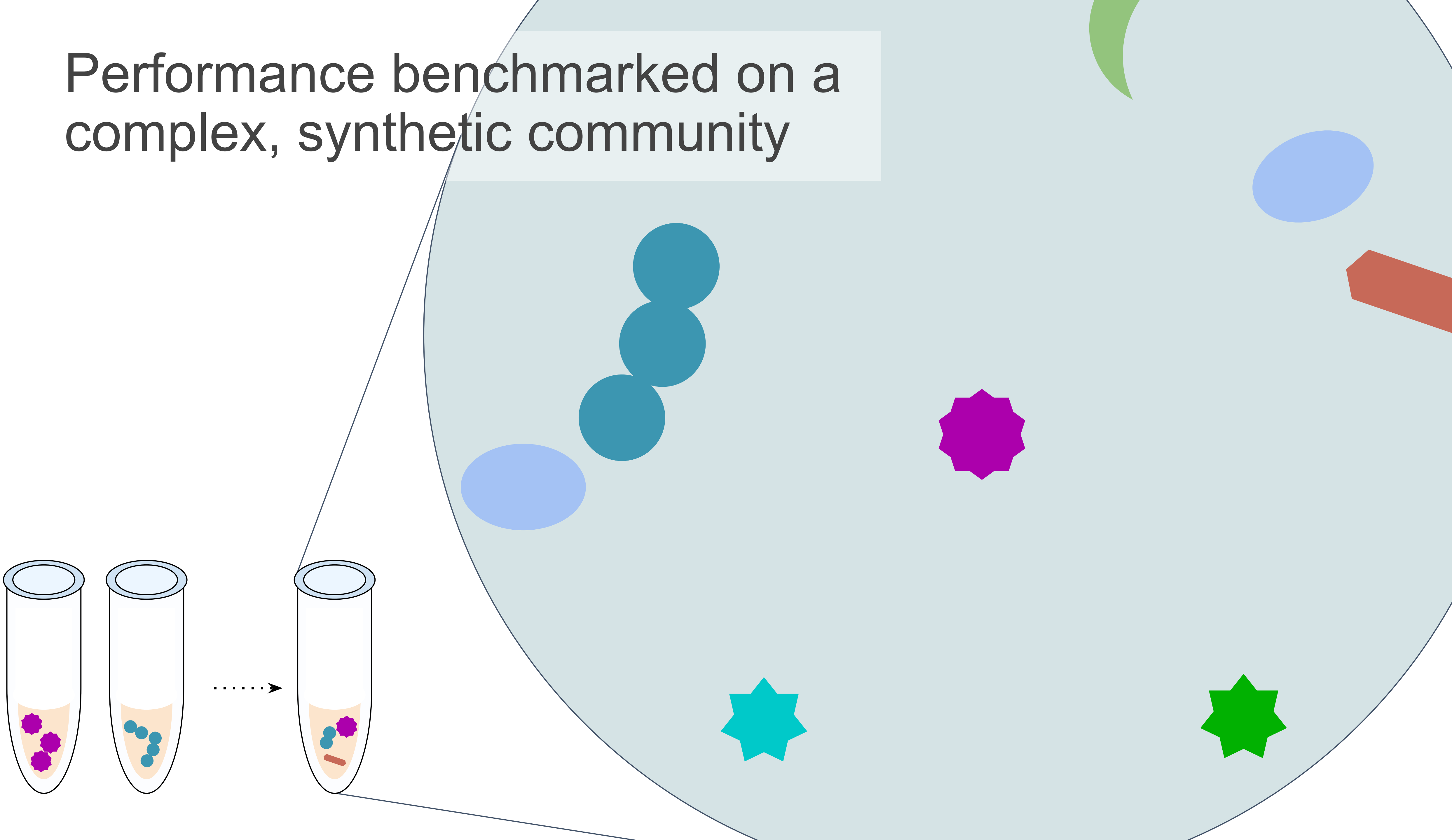






# Strain-resolved discovery

# Performance benchmarked on a complex, synthetic community



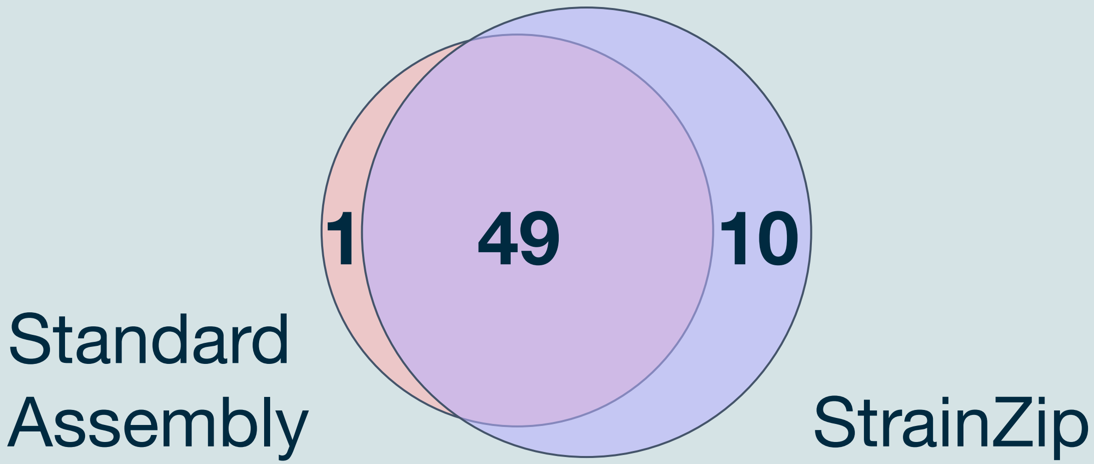
# Antibiotic resistance genes are widespread in the gut microbiome

- Detection can inform treatment

# Antibiotic resistance genes are widespread in the gut microbiome

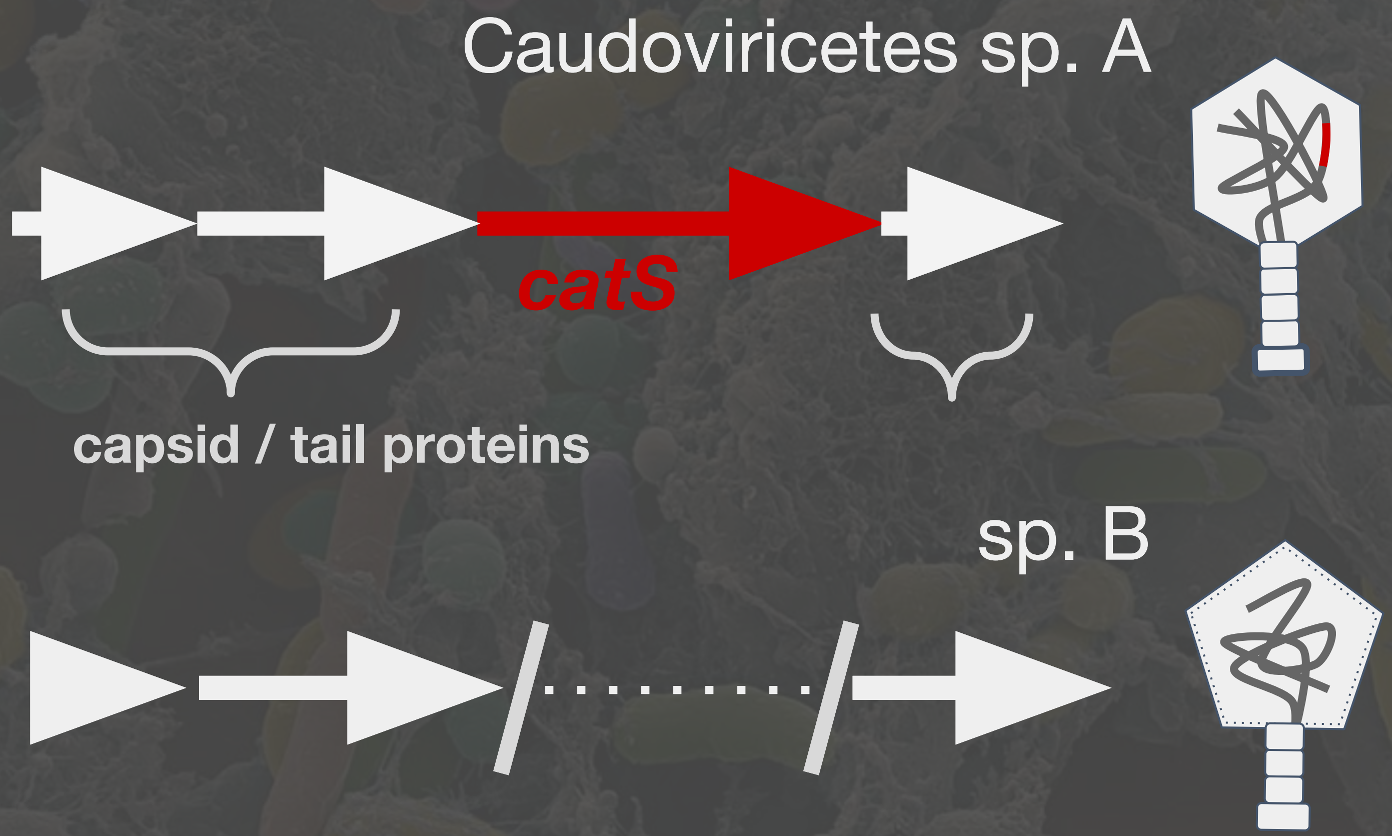
- Detection can inform treatment

No. of Unique Resistance Genes Found



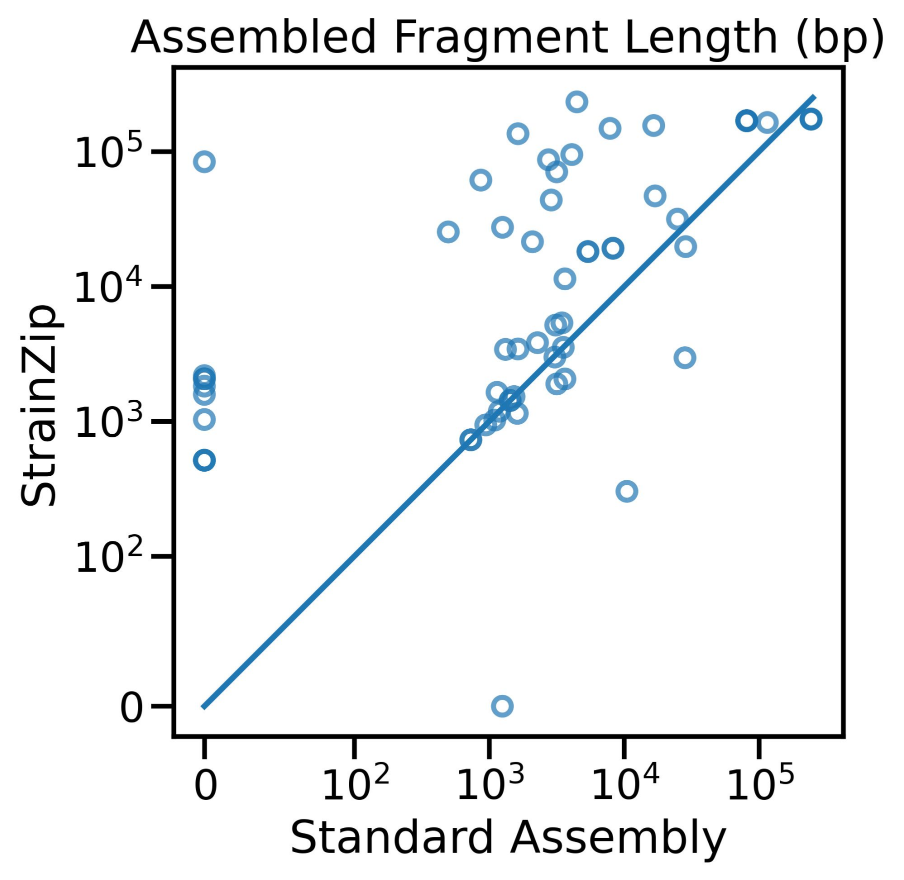
# Antibiotic resistance genes are widespread in the gut microbiome

- Detection can inform treatment
- Can be carried in phage genomes
- Long sequence fragments provide useful information

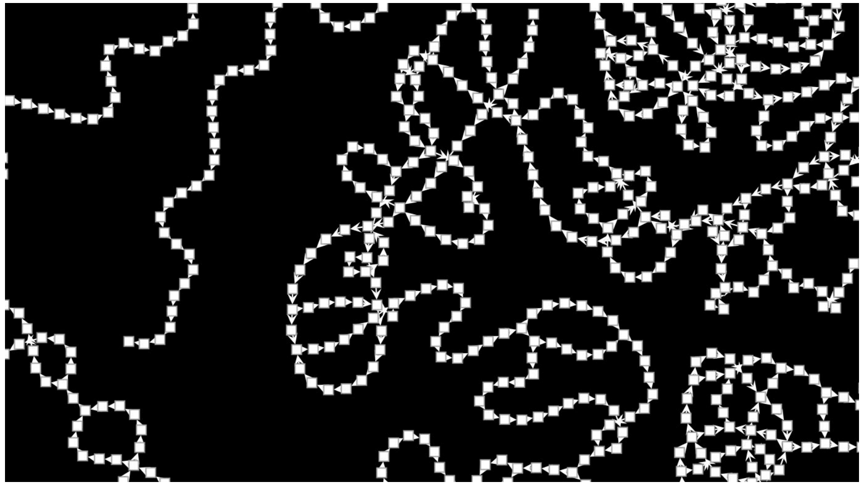


# Antibiotic resistance genes are widespread in the gut microbiome

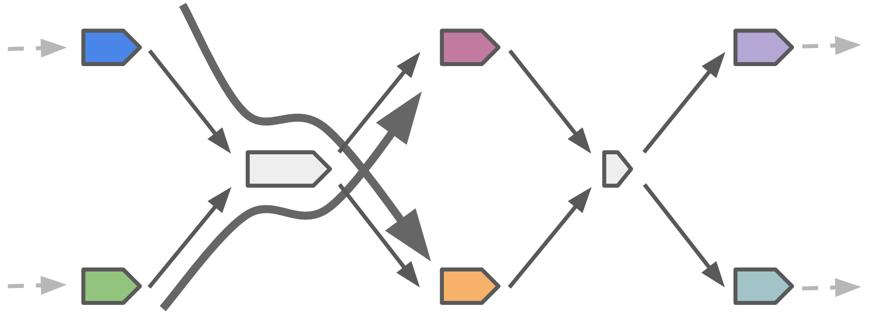
- Detection can inform treatment
- Can be carried in phage genomes
- Long sequence fragments provide useful information



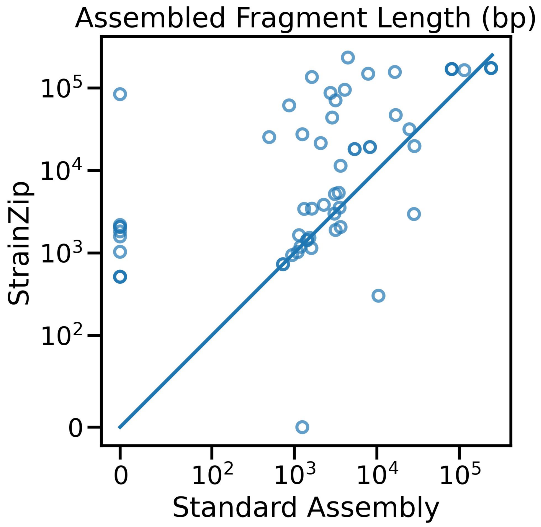
## Complex Metagenome Graphs



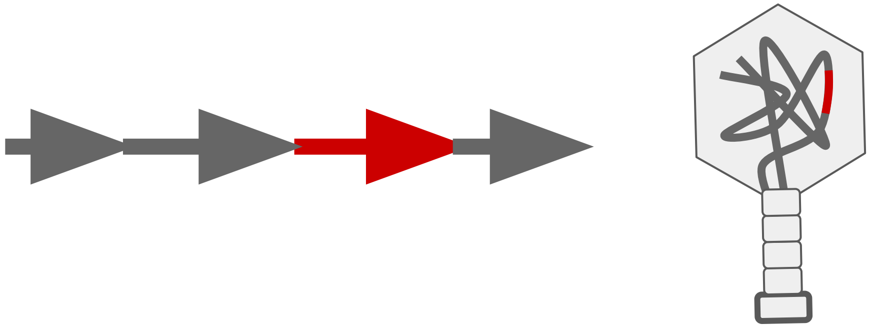
## StrainZip Iteratively Unzips Junctions



## Strain-Resolved Metagenomics

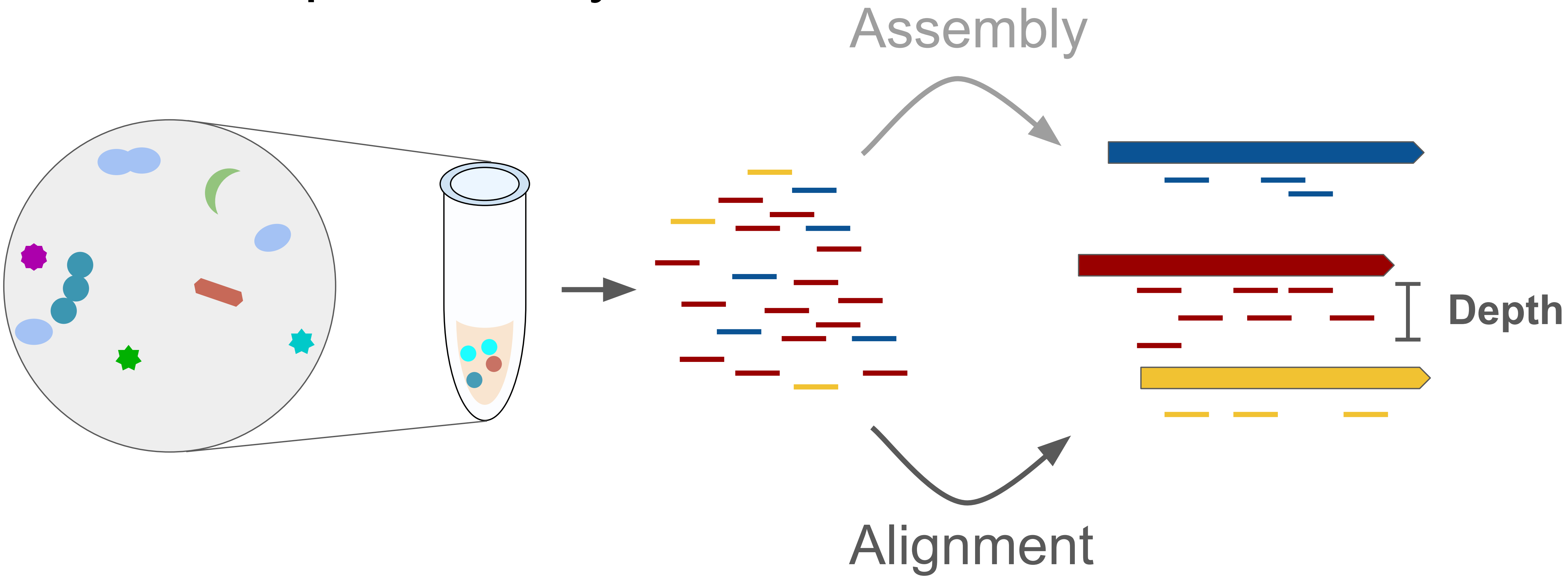


## Antibiotic Resistance Potential of Phage

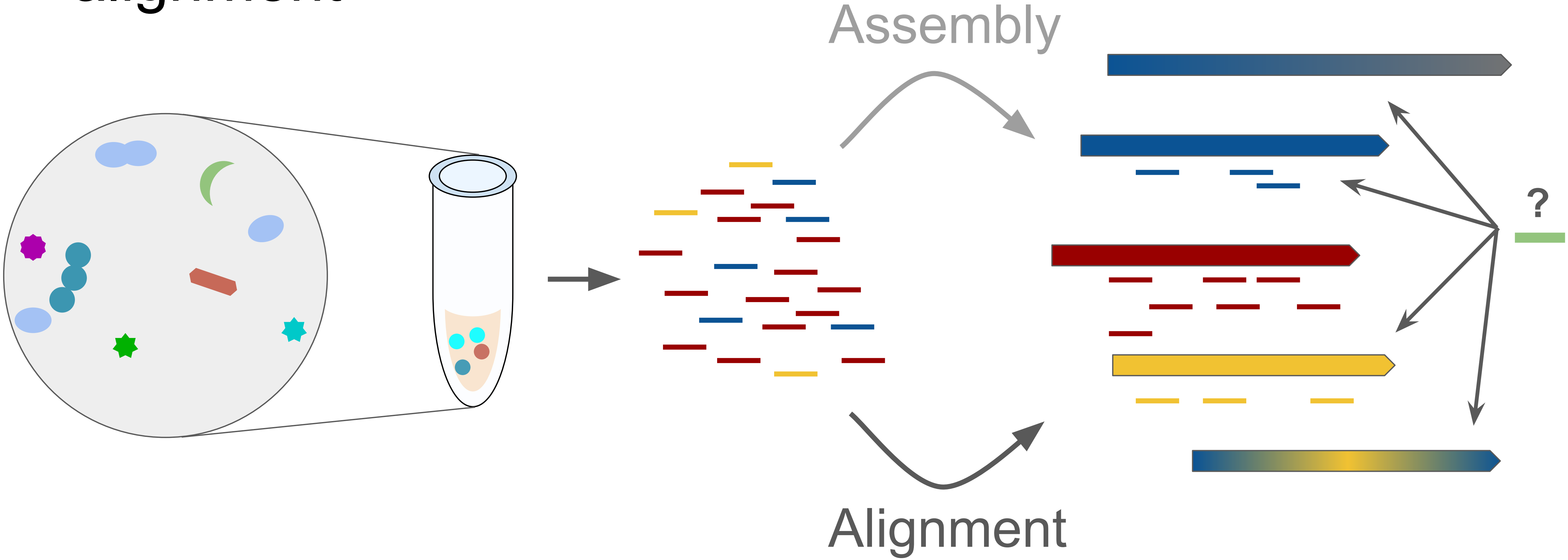


**Rewind:** I also care about depth quantification

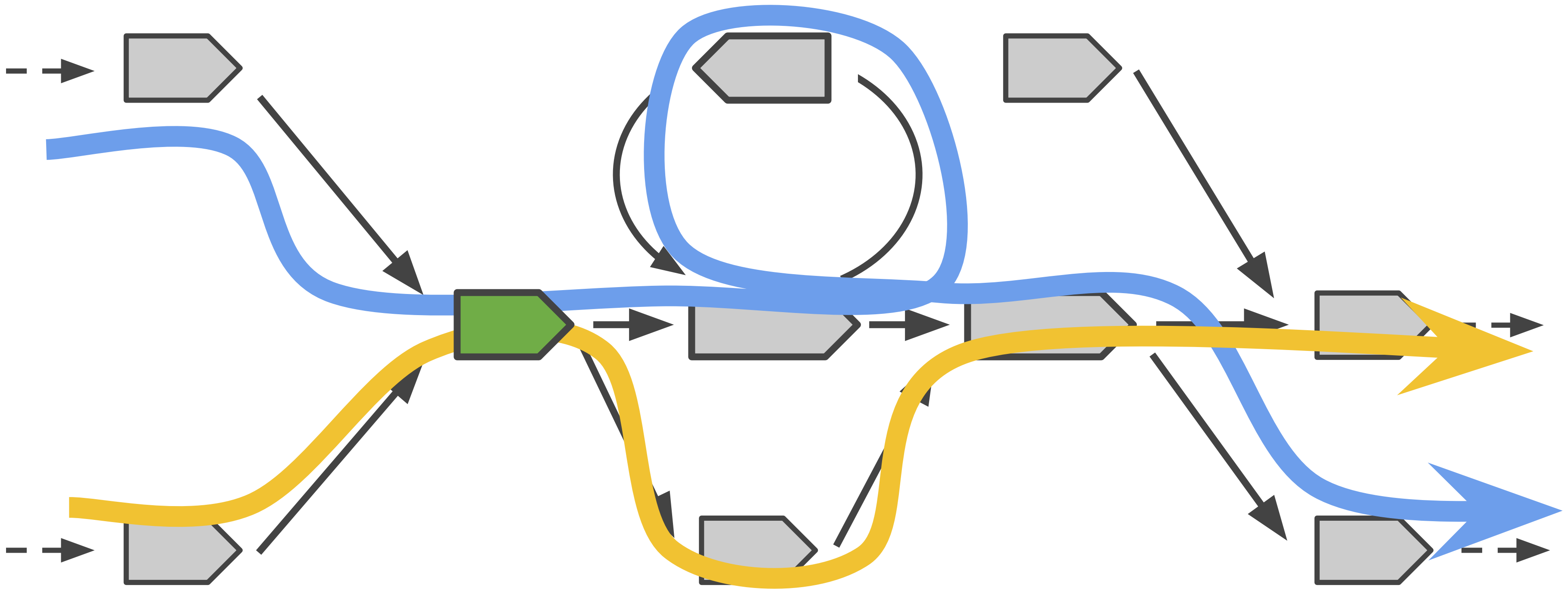
# Assembly and depth quantification are complementary



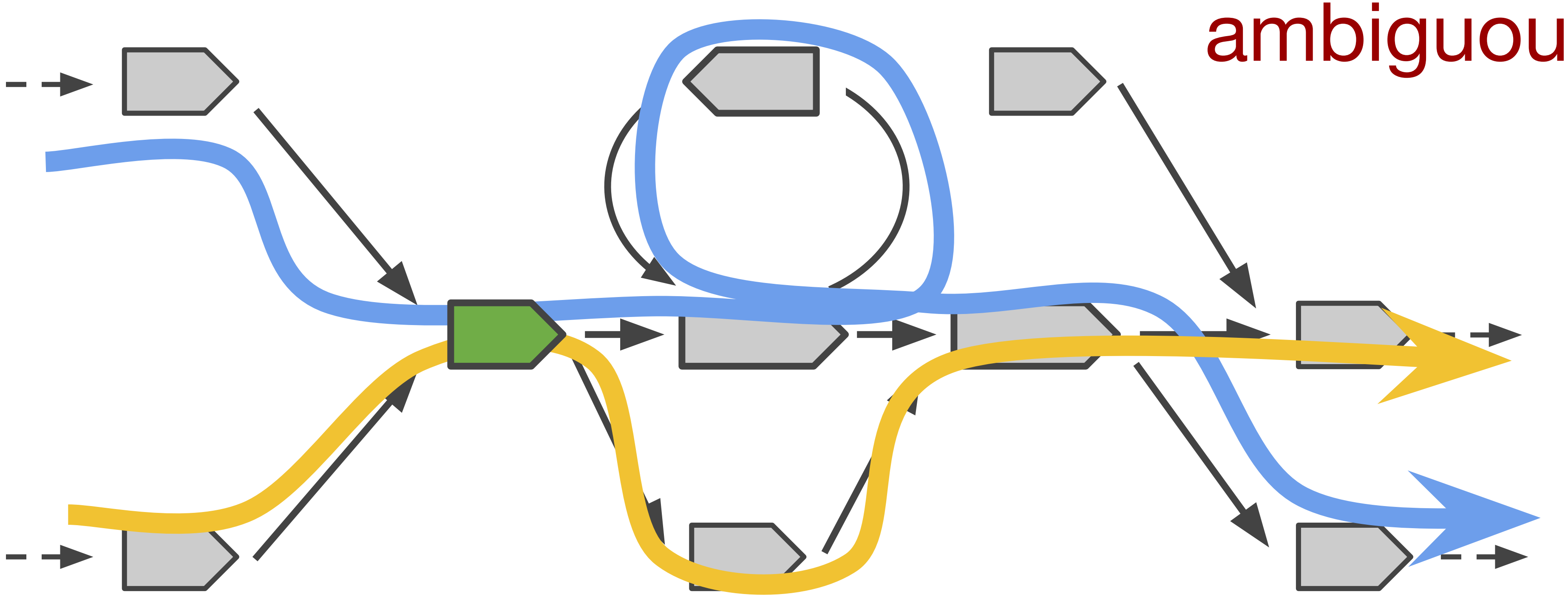
# Closely related sequences are a major challenge for alignment



# Shared sequences mean reads map ambiguously



Shared sequences mean ~~reads map ambiguously~~ kmers are ambiguous



# Quick intro to de Bruijn graphs

Read #1

…**CGTA**CCTGGATTAC…

Read #2

CCTGGATTAC**TTAA**…

Assembly

…**CGTA**CCTGGATTAC**TTAA**…

# De Bruijn graphs

Motivation: **Assembly** - stitching together longer sequences using overlapping portions

# Fragment reads into k-mers

Read #1

```
…CGTACCTGGATTAC
 CGTA
  GTAC
   TACC
    ACCT
     CCTG
      CTGG
       TGGA
        GGAT
         GATT
          ATTA
           TTAC
```

Read #2

```
CCTGGATTACTTAA…
CCTG
 CTGG
  TGGA
   GGAT
    GATT
     ATTA
      TTAC
       TACT
        ACTT
         CTTA
          TTAA
```

All k-mers

```
…
 CGTA
  GTAC
   TACC
    ACCT
     CCTG (x2)
      CTGG (x2)
       TGGA (x2)
        GGAT (x2)
         GATT (x2)
          ATTA (x2)
           TTAC (x2)
            TACT
             ACTT
              CTTA
               TTAA
                  …
```

# Collect unique k-mers

CGTA GTAC TACC ACCT CCTG CTGG TGGA GGAT GATT ATTA TTAC TACT ACTT CTTA TTAA

# Identify k-mer pairs where (k-1) suffix on one is same as other's prefix

CGTA GTAC TACC ACCT CCTG CTGG TGGA GGAT GATT ATTA TTAC TACT ACTT CTTA TTAA

# Draw edge

CGTA → GTAC TACC ACCT CCTG CTGG TGGA GGAT GATT ATTA TTAC TACT ACTT CTTA TTAA

# Linear paths (unitigs) are assembled sequence

CGTA → GTAC → TACC → ACCT → CCTG → CTGG → TGGA → GGAT → GATT → ATTA → TTAC → TACT → ACTT → CTTA → TTAA

Unitig:

…CGTACCTGGATTACTTAA…

# Mutations / errors introduce new k-mers

**Read #1**

```
…CGTACCTGGATTAC
 CGTA
  GTAC
   TACC
    ACCT
     CCTG
      CTGG
       TGGA
        GGAT
         GATT
          ATTA
           TTAC
```

**Read #2**

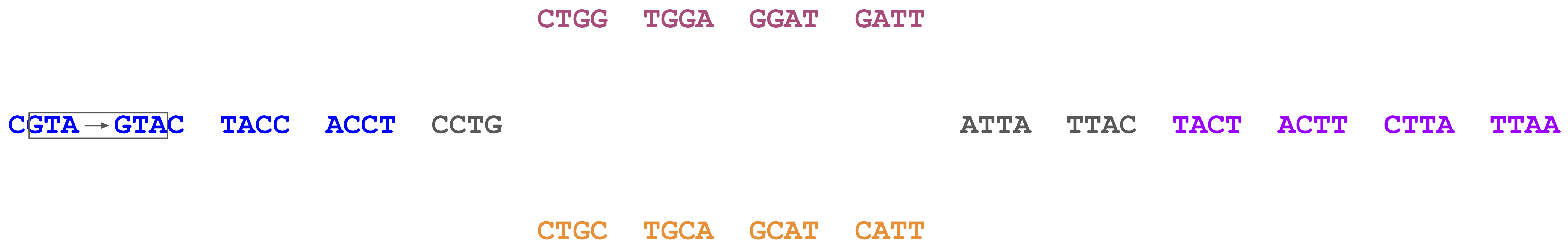
```
CCTGCATTACTTAA…
CCTG
 CTGC
  TGCA
   GCAT
    CATT
     ATTA
      TTAC
       TACT
        ACTT
         CTTA
          TTAA
```

**Diversity / Errors**

…CGTACCTGGATTACTTAA…

…CGTACCTGCATTACTTAA…

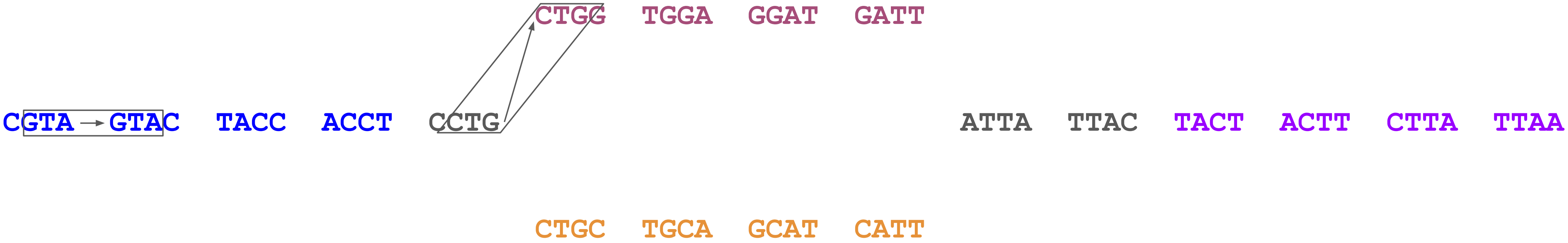
# Same edge-drawing process


CTGG TGGA GGAT GATT

CGTA → GTAC TACC ACCT CCTG ATTA TTAC TACT ACTT CTTA TTAA

CTGC TGCA GCAT CATT

# Same edge-drawing process

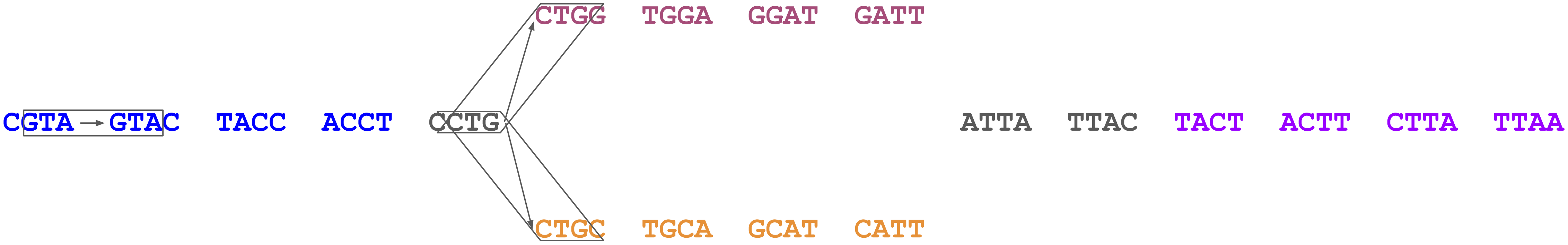


CTGG TGGA GGAT GATT

CGTA GTAC TACC ACCT CCTG ATTA TTAC TACT ACTT CTTA TTAA

CTGC TGCA GCAT CATT

# But now some k-mers have multiple edges

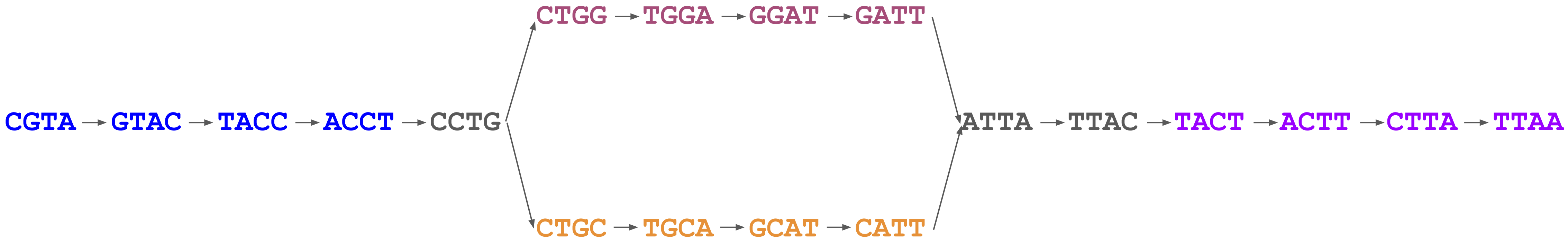


CTGG TGGA GGAT GATT

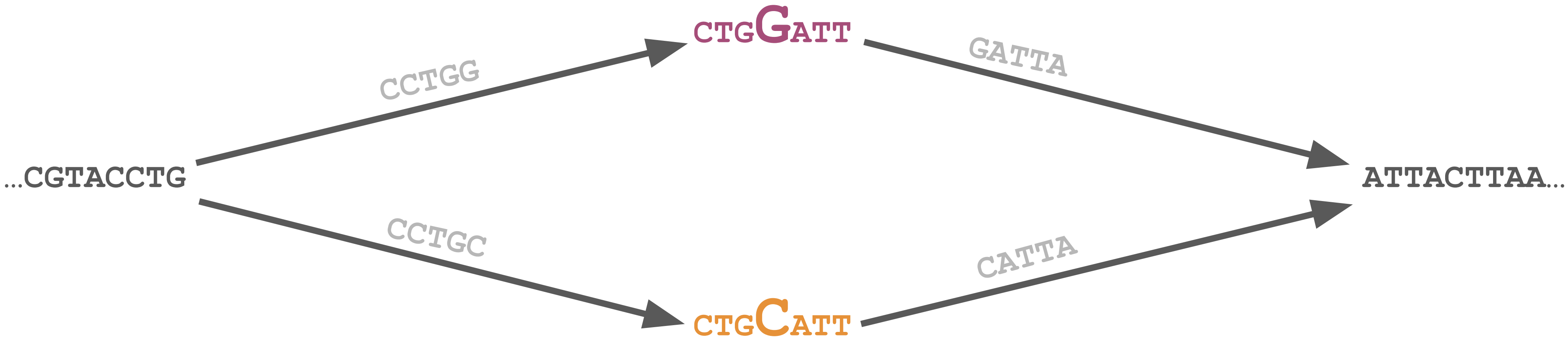
CGTA GTAC TACC ACCT CCTG ATTA TTAC TACT ACTT CTTA TTAA

CTGC TGCA GCAT CATT

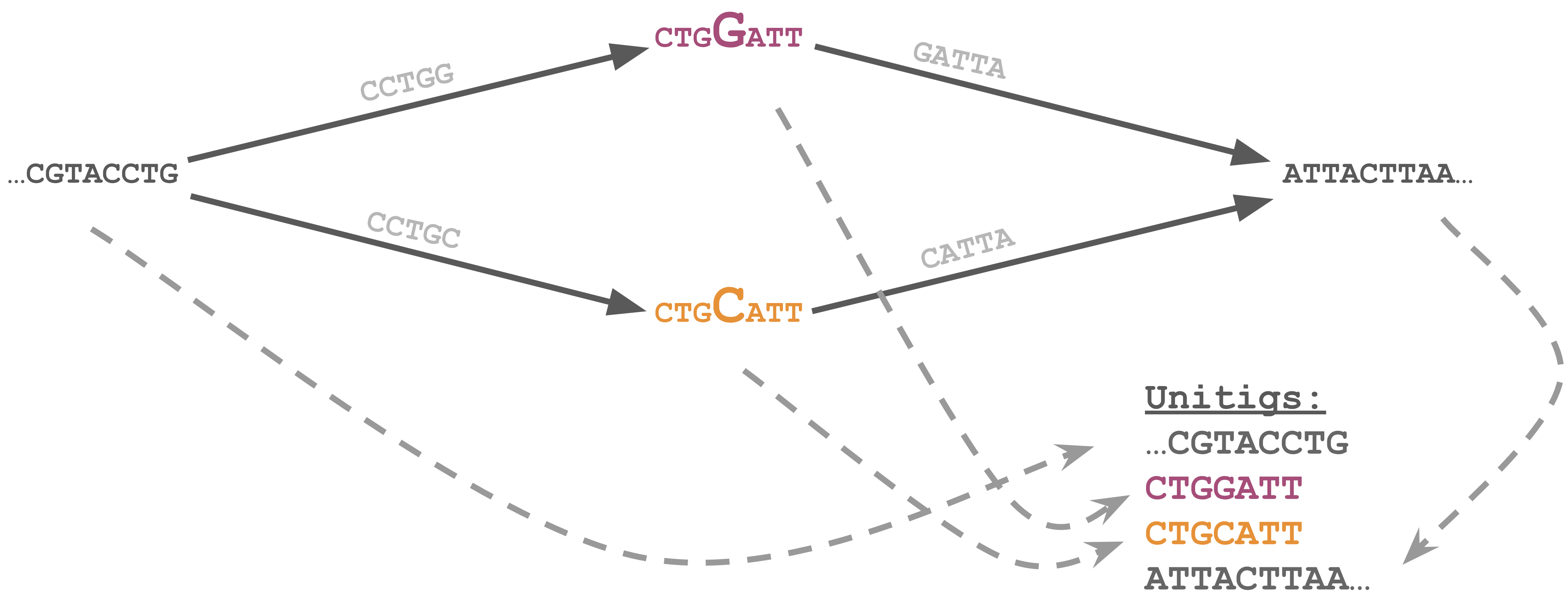
# This introduces a "bubble"



# The two sides of the bubble reflect the observed diversity



# Again we extract unitigs, but now they're shorter, fragmented



**Unitigs:**

...CGTACCTG

CTGGATT

CTGCATT

ATTACTTAA...

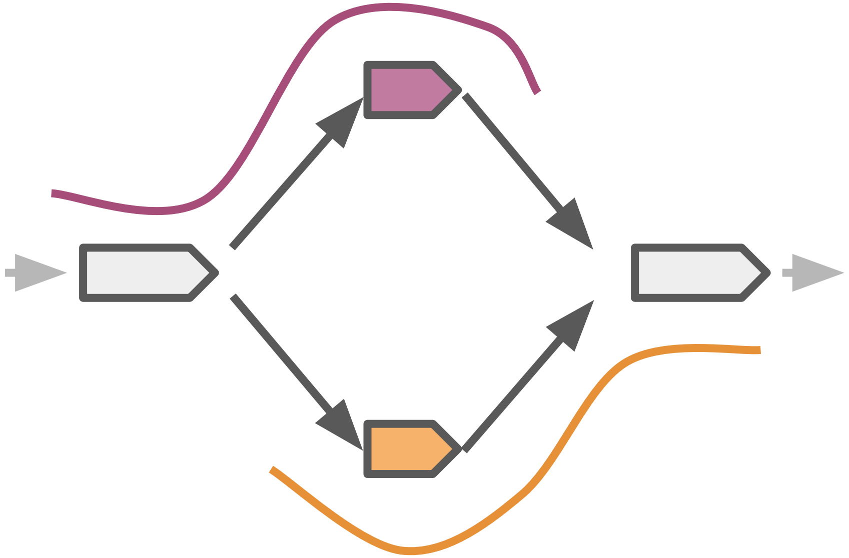
# Sequences are walks along the graph; can align reads without worrying about fragmentation

**Read #1**

...CGTACTGGATTAC

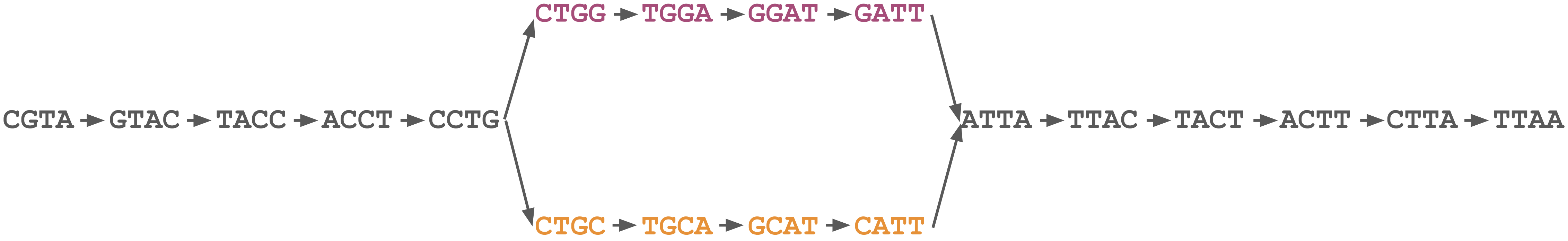
**Read #2**

CCTGCATTACTTAA...



# Alternatively: Exact k-mer counting

| Unitig #1 | Unitig #2 | Unitig #3 | Unitig #4 |
|---|---|---|---|
| CGTA | CTGG | CTGC | ATTA (x2) |
| GTAC | TGGA | TGCA | TTAC (x2) |
| TACC | GGAT | GCAT | TACT |
| ACCT | GATT | CATT | ACTT |
| CCTG (x2) | | | CTTA |
| | | | TTAA |



CTGG → TGGA → GGAT → GATT

CGTA → GTAC → TACC → ACCT → CCTG

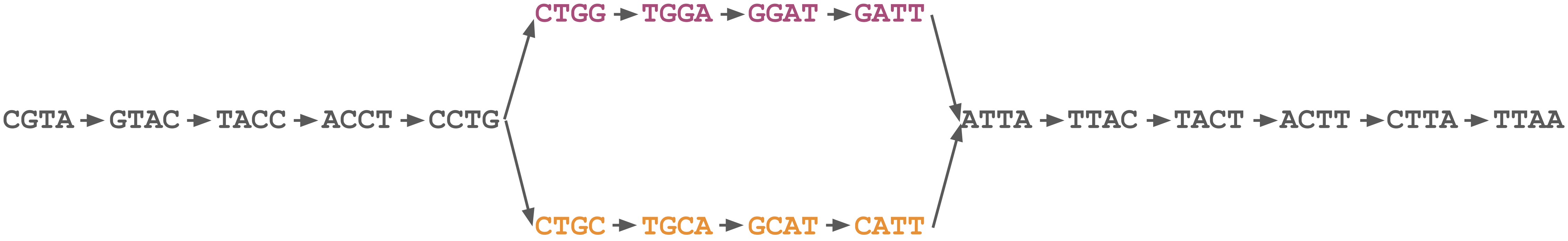
ATTA → TTAC → TACT → ACTT → CTTA → TTAA

CTGC → TGCA → GCAT → CATT

# Alternatively: Exact k-mer counting

Much faster than read alignment

Every k-mer in the sample is in the dBG, by construction

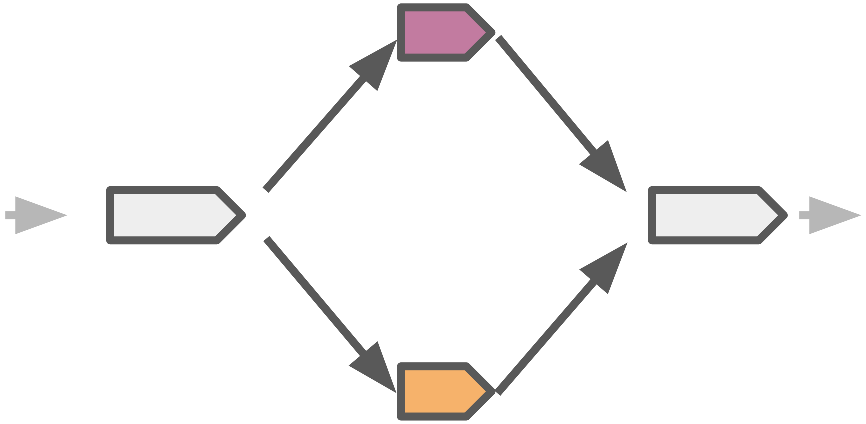


# Alternatively: Exact k-mer counting

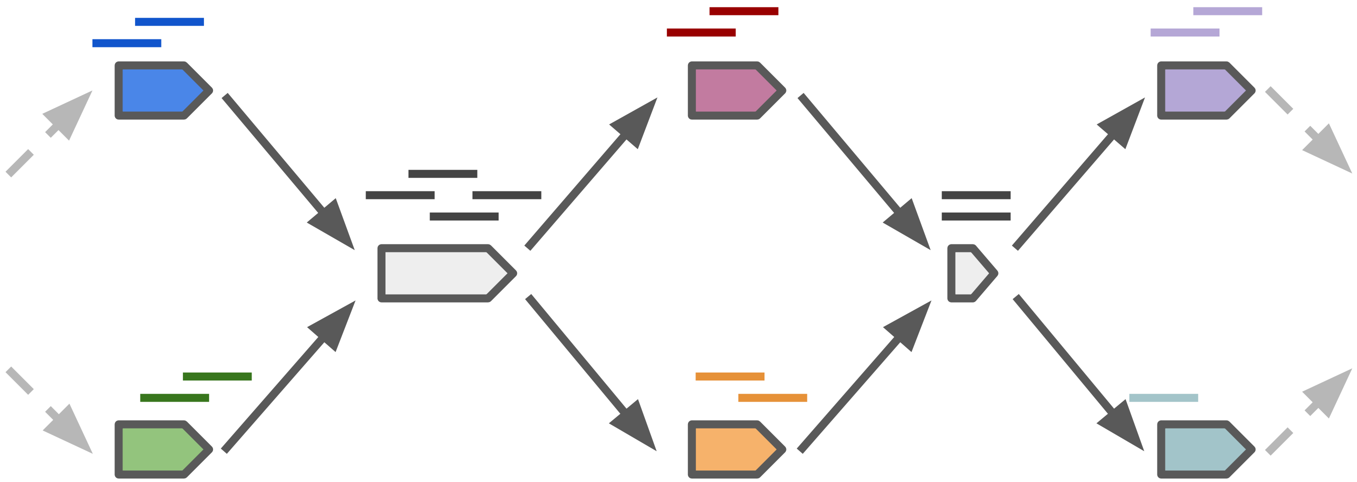
Much faster than read alignment

Every k-mer in the sample is in the dBG, by construction

No ambiguity about what is being quantified: it's unitigs



# KEY IDEA: The expected depth of a k-mer is the sum of the paths that include that k-mer



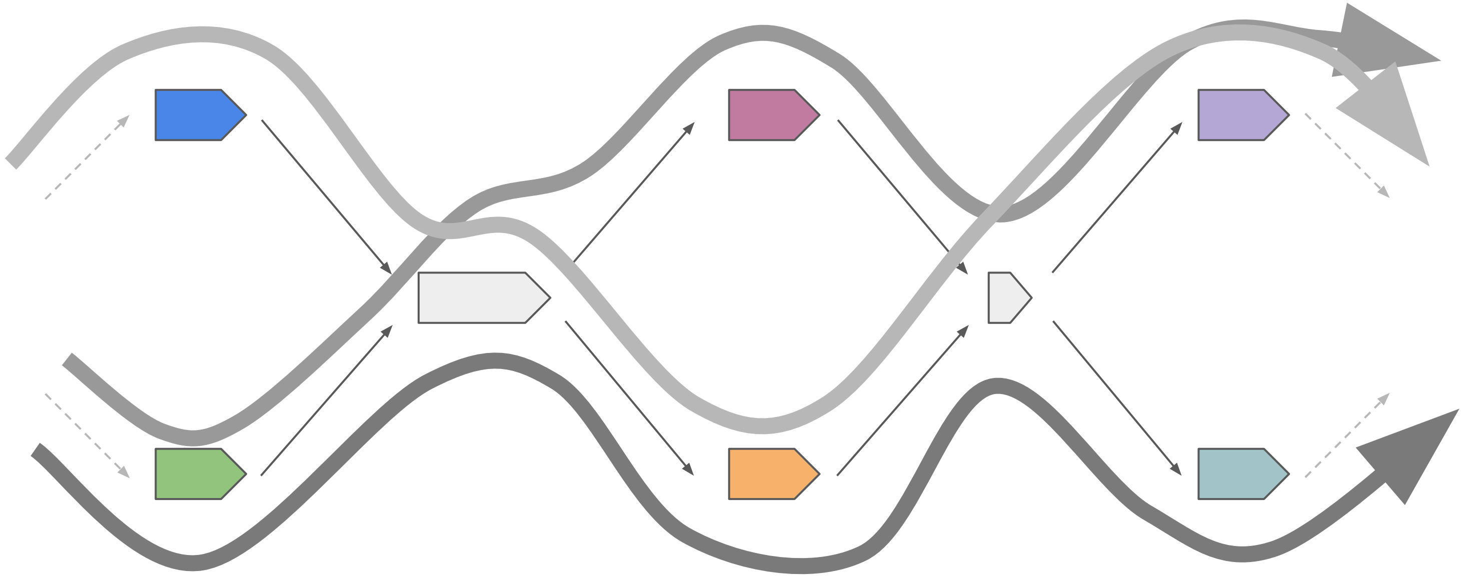
Path depths (unknown)

Indicator: k-mer in path

Observed depths

$$\sum_p x_{pk}\beta_p \approx Y_k$$

# KEY IDEA: The expected depth of a k-mer is the sum of the paths that include that k-mer



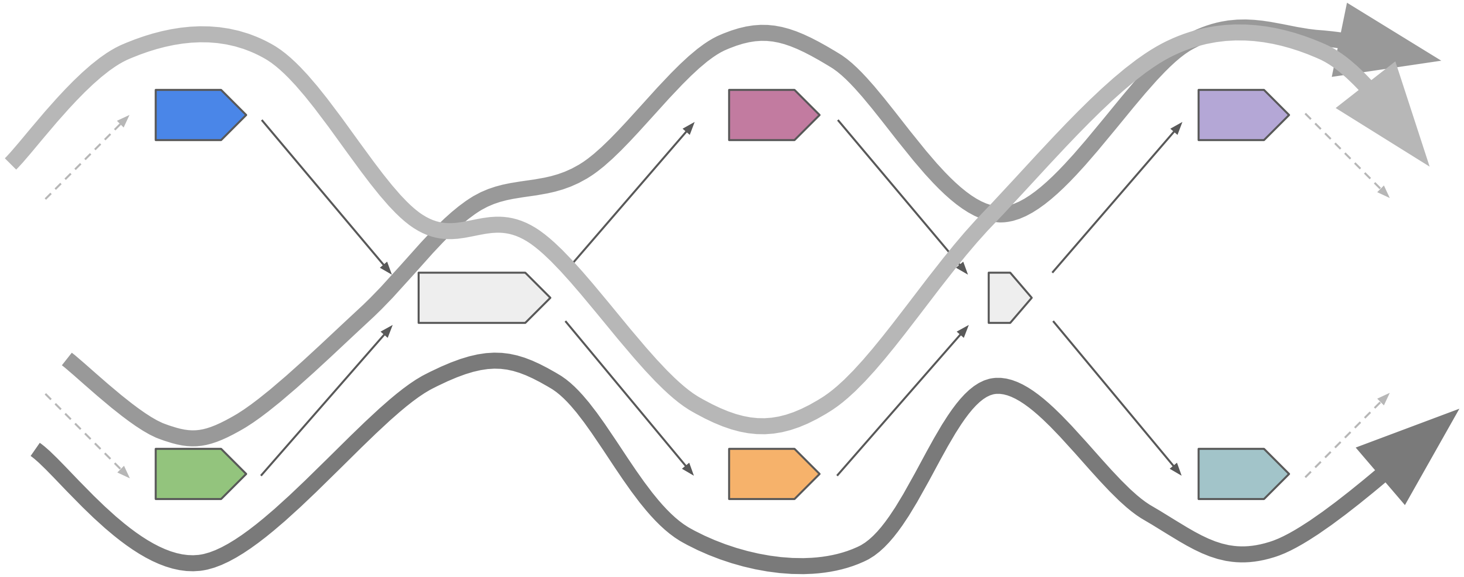
Path depths (unknown)

Path encoding

Observed depths

$$X\beta \approx Y$$

KEY IDEA: The expected depth of a k-mer is the sum of the paths that include that k-mer



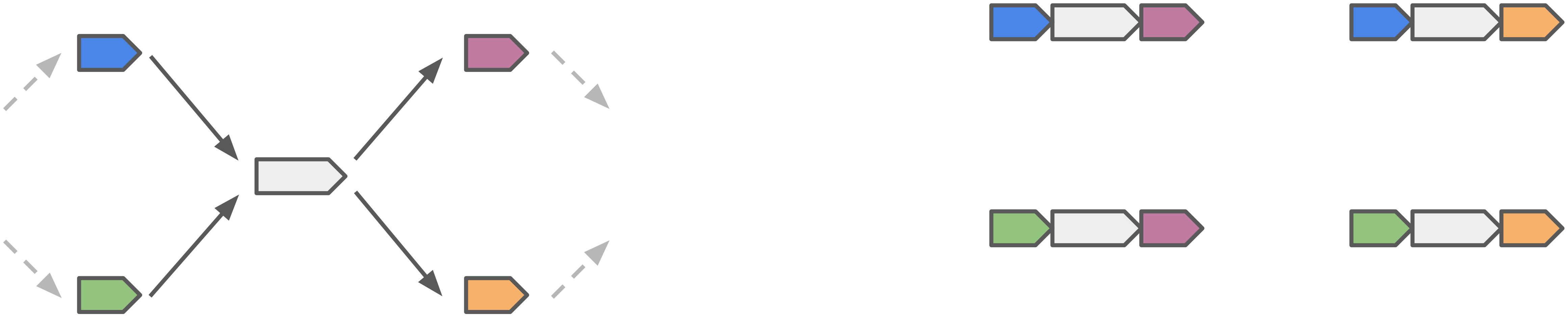
**Estimate this**

From these

$$X\beta \approx Y$$

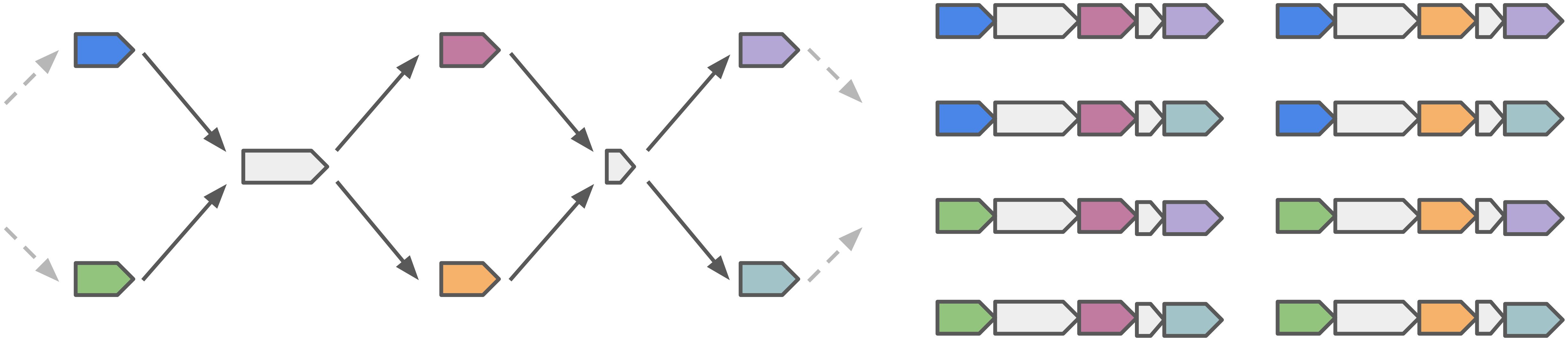
**Deconvolution**: Inferring the depth of these latent paths based on observed k-mer depths

# We can enumerate all possible paths on our assembly graph



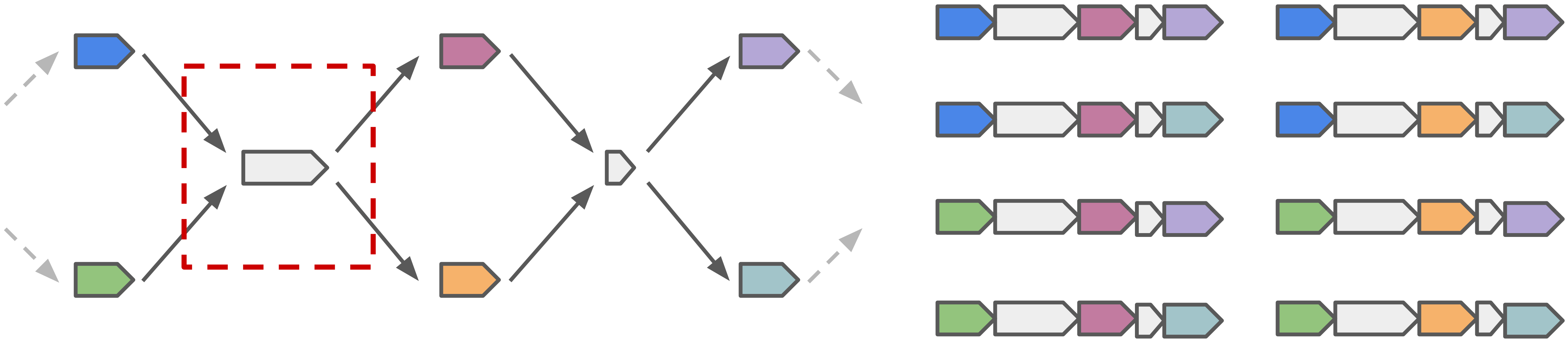
$$\boxed{X}\beta \approx Y$$

# We can enumerate all possible paths on our assembly graph

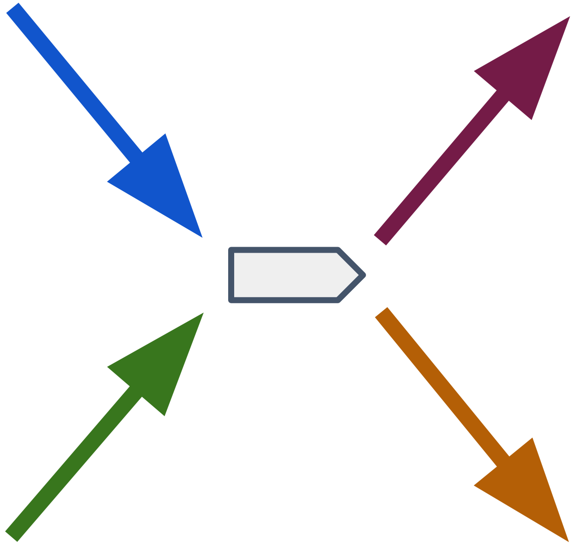


…but this grows exponentially with graph complexity

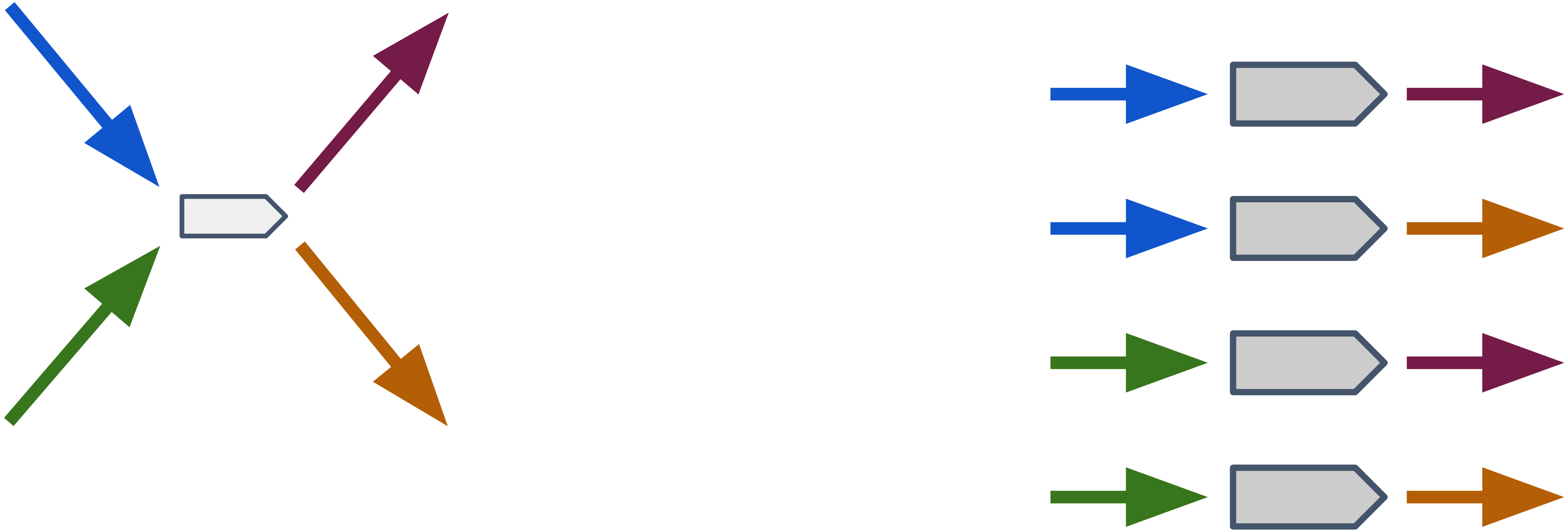
# KEY IDEA: A single "junction" is the minimum unit of deconvolution



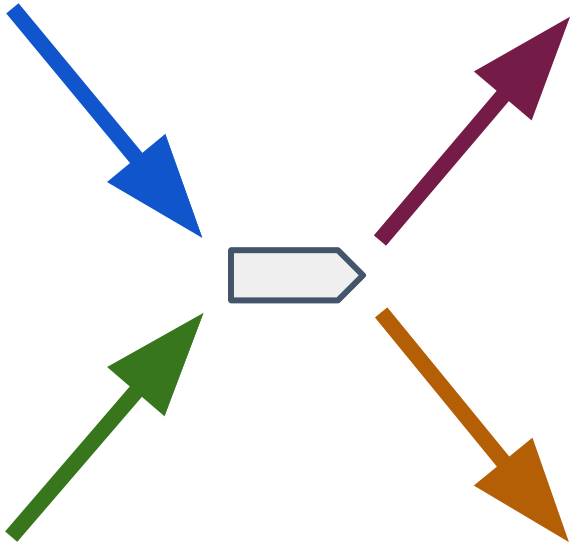
# KEY IDEA: A single "junction" is the minimum unit of deconvolution

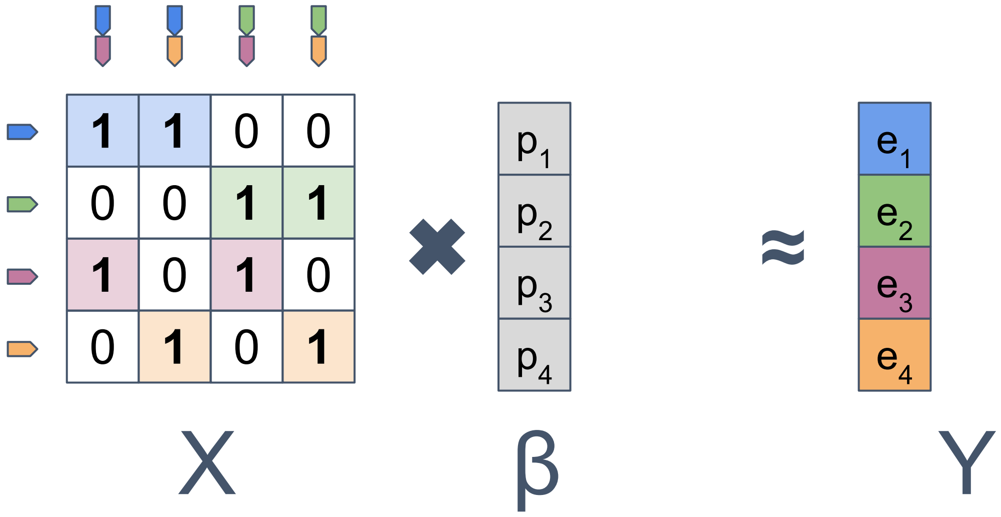


# KEY IDEA: A single "junction" is the minimum unit of deconvolution



# Focus on just one junction at a time
# Quantify local paths

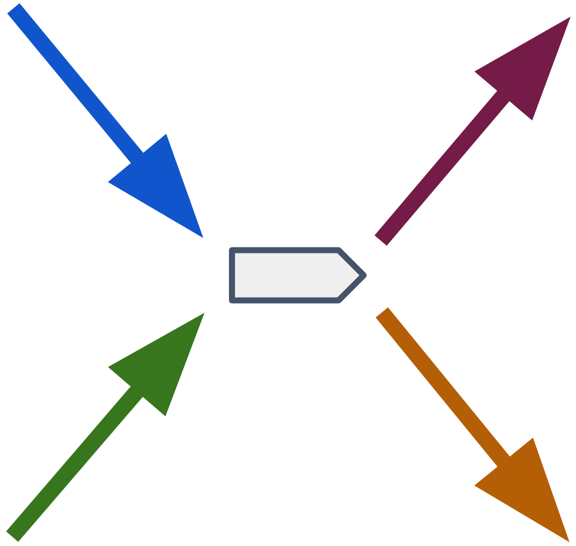




$$\underbrace{\begin{bmatrix}1&1&0&0\\0&0&1&1\\1&0&1&0\\0&1&0&1\end{bmatrix}}_{X} \times \underbrace{\begin{bmatrix}p_1\\p_2\\p_3\\p_4\end{bmatrix}}_{\beta} \approx \underbrace{\begin{bmatrix}e_1\\e_2\\e_3\\e_4\end{bmatrix}}_{Y}$$

Linear regression

# Focus on just one junction at a time
# Select (and quantify) local paths

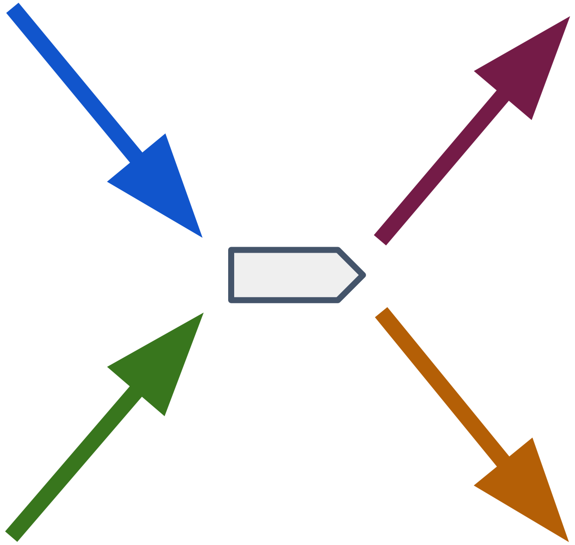


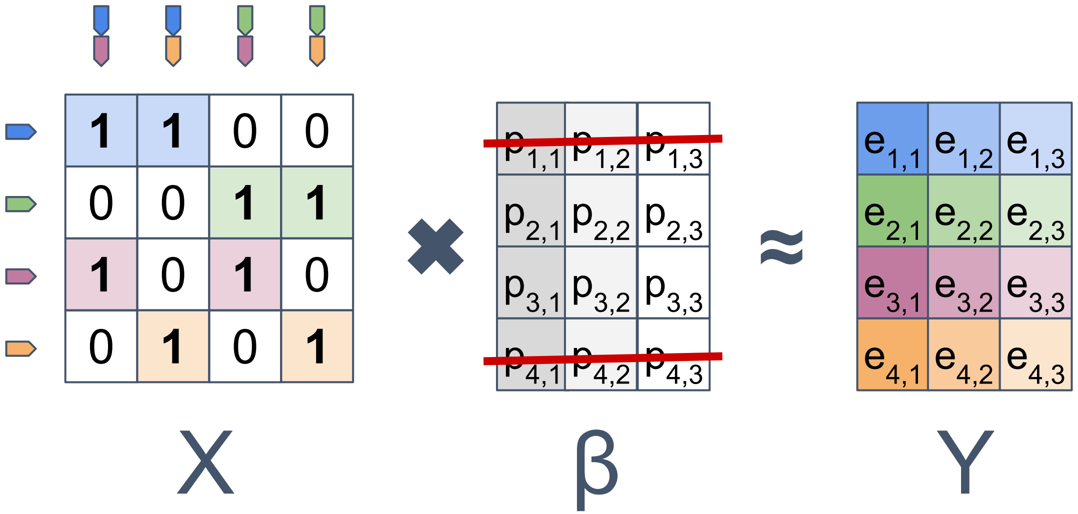
$$\underbrace{\begin{pmatrix}1&1&0&0\\0&0&1&1\\1&0&1&0\\0&1&0&1\end{pmatrix}}_{X} \times \underbrace{\begin{pmatrix}\cancel{p_1}\\p_2\\p_3\\\cancel{p_4}\end{pmatrix}}_{\beta} \approx \underbrace{\begin{pmatrix}e_1\\e_2\\e_3\\e_4\end{pmatrix}}_{Y}$$

Linear regression

Model selection

# Focus on just one junction at a time
# Select (and quantify) local paths





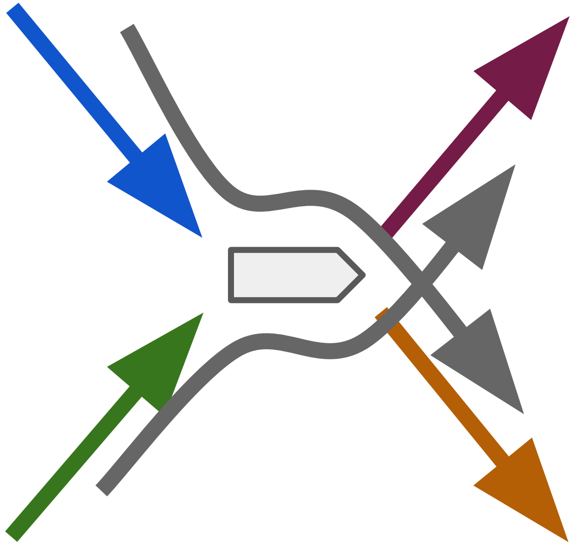
$$\underset{X}{\begin{bmatrix}1&1&0&0\\0&0&1&1\\1&0&1&0\\0&1&0&1\end{bmatrix}} \times \underset{\beta}{\begin{bmatrix}\cancel{p_{1,1}}&\cancel{p_{1,2}}&\cancel{p_{1,3}}\\p_{2,1}&p_{2,2}&p_{2,3}\\p_{3,1}&p_{3,2}&p_{3,3}\\\cancel{p_{4,1}}&\cancel{p_{4,2}}&\cancel{p_{4,3}}\end{bmatrix}} \approx \underset{Y}{\begin{bmatrix}e_{1,1}&e_{1,2}&e_{1,3}\\e_{2,1}&e_{2,2}&e_{2,3}\\e_{3,1}&e_{3,2}&e_{3,3}\\e_{4,1}&e_{4,2}&e_{4,3}\end{bmatrix}}$$

Linear regression

Model selection

Across multiple samples

# Drop paths with no depth in any sample

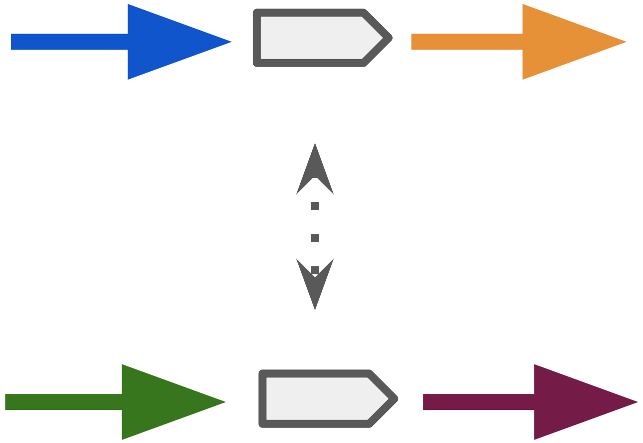


| | | | |
|---|---|---|---|
| 🟦🟪 | ~~0~~ | ~~0~~ | ~~0~~ |
| 🟦🟧 | 0.1 | 0.9 | 0 |
| 🟩🟪 | 0.5 | 0 | 0.5 |
| 🟩🟧 | ~~0~~ | ~~0~~ | ~~0~~ |

$$\hat{\beta}$$

Used statistical linkage to resolve ambiguity about which of possible paths are "real"

# Resolve ambiguity, longer linear sequences

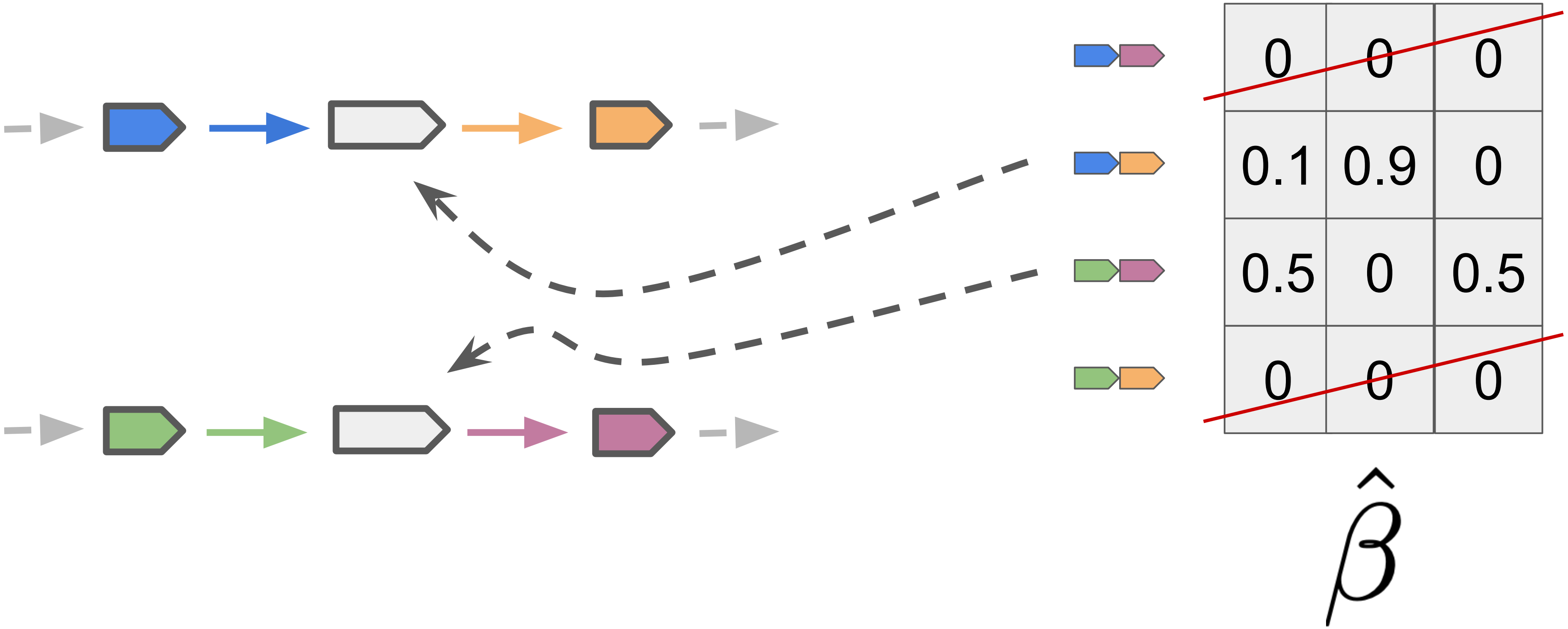


|  |  |  |  |
|---|---|---|---|
| 🟦🟪 | ~~0~~ | ~~0~~ | ~~0~~ |
| 🟦🟧 | 0.1 | 0.9 | 0 |
| 🟩🟪 | 0.5 | 0 | 0.5 |
| 🟩🟧 | ~~0~~ | ~~0~~ | ~~0~~ |

$\hat{\beta}$

Can "unzip" this unitig into two paths

# Resolve ambiguity, longer linear sequences

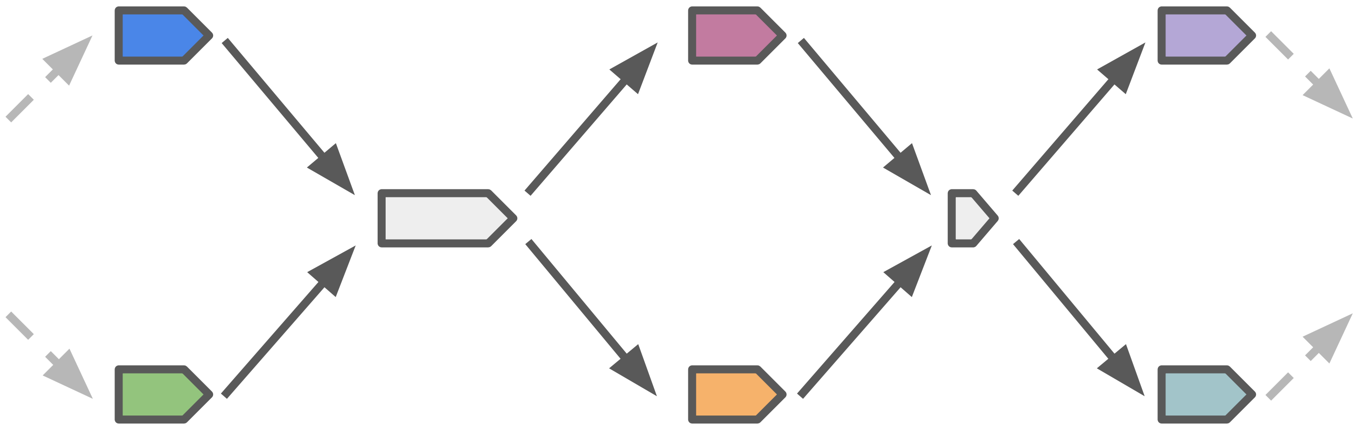


| | | | |
|---|---|---|---|
| 🟦🟪 | ~~0~~ | ~~0~~ | ~~0~~ |
| 🟦🟧 | 0.1 | 0.9 | 0 |
| 🟩🟪 | 0.5 | 0 | 0.5 |
| 🟩🟧 | ~~0~~ | ~~0~~ | ~~0~~ |

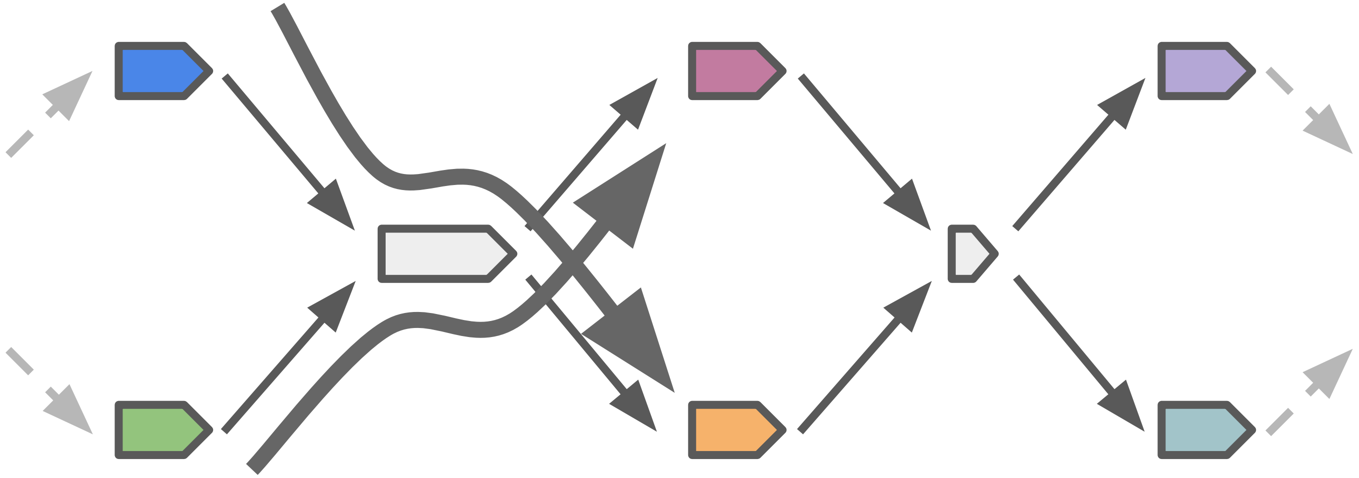
$\hat{\beta}$

Newly split unitigs already have depths estimated across samples

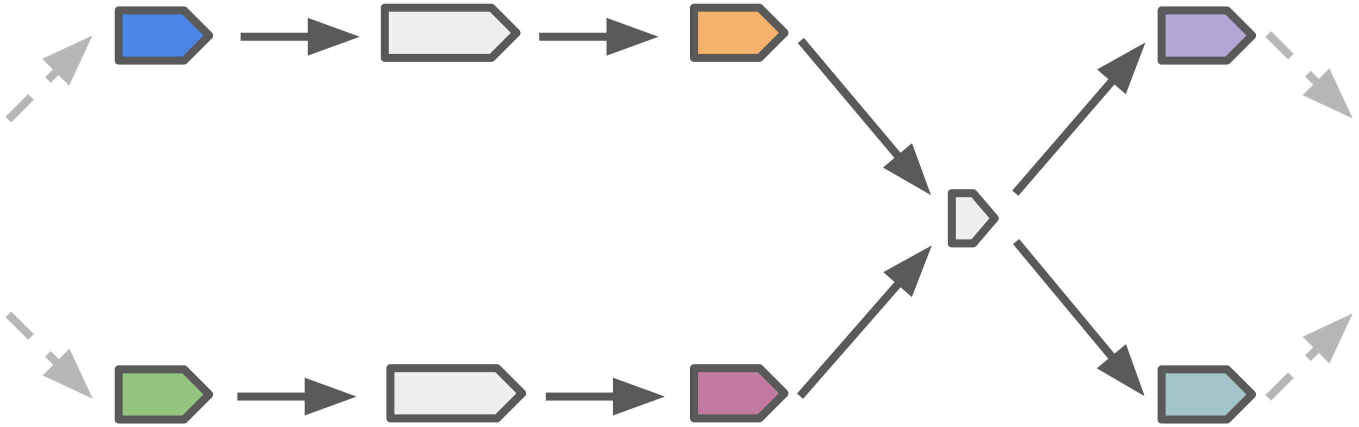
# Iteratively unzipping local junctions



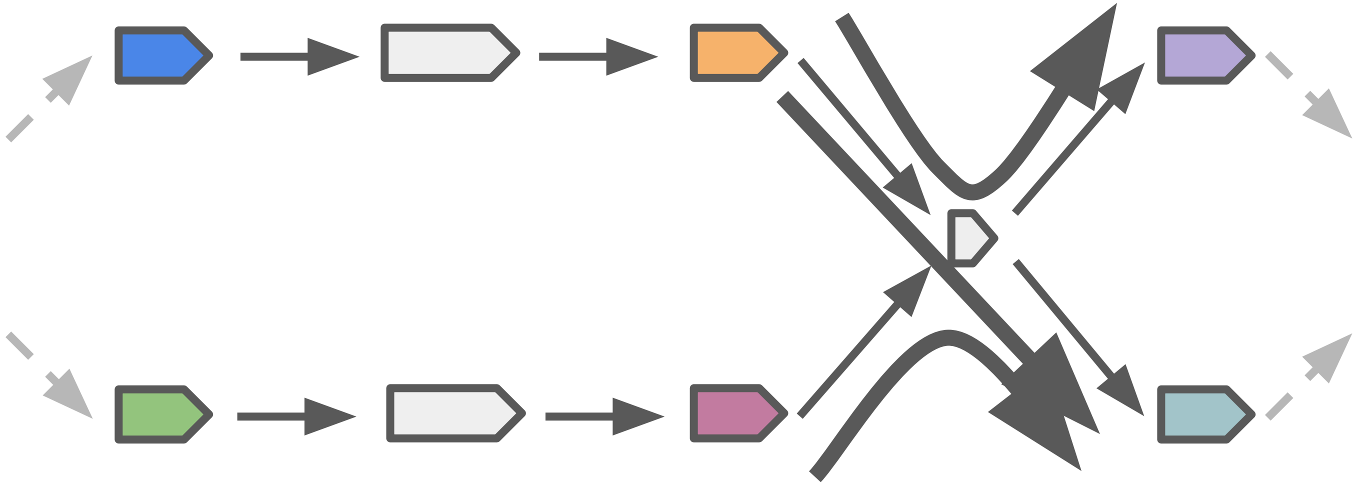
# Iteratively unzipping local junctions



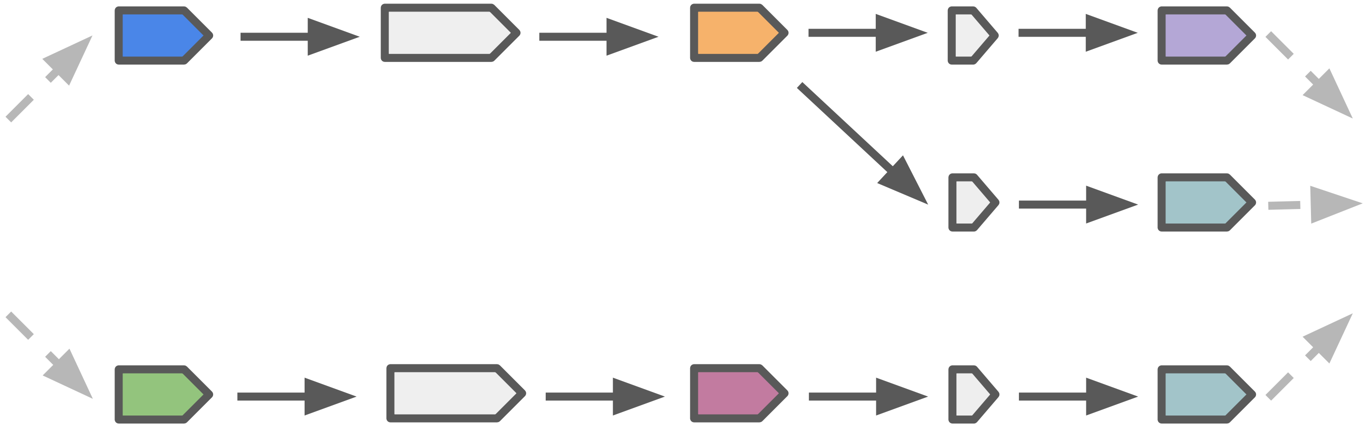
# Iteratively unzipping local junctions



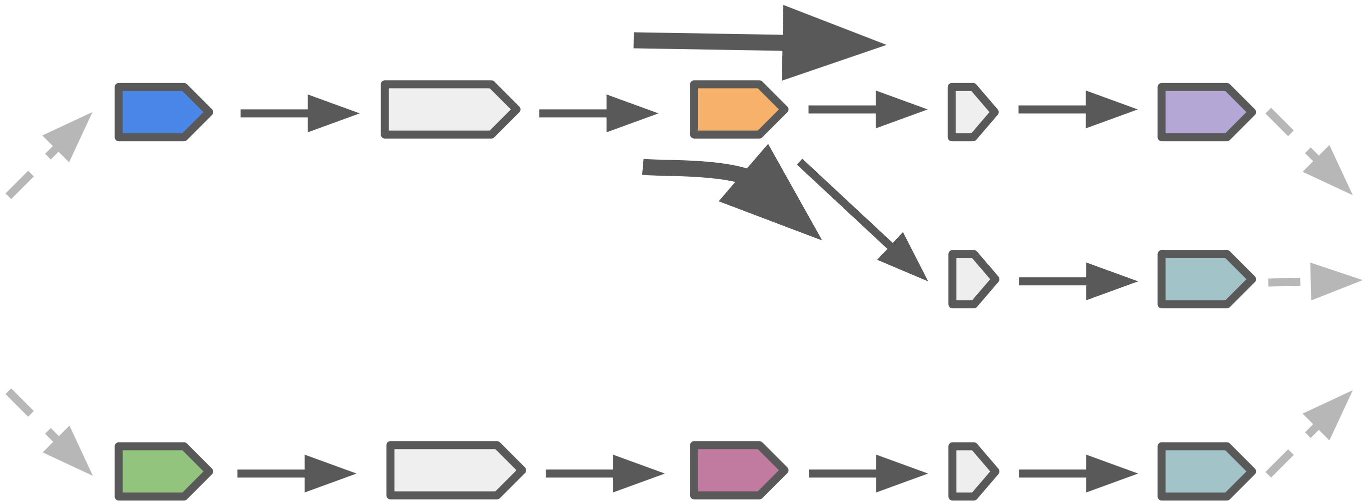
# Iteratively unzipping local junctions



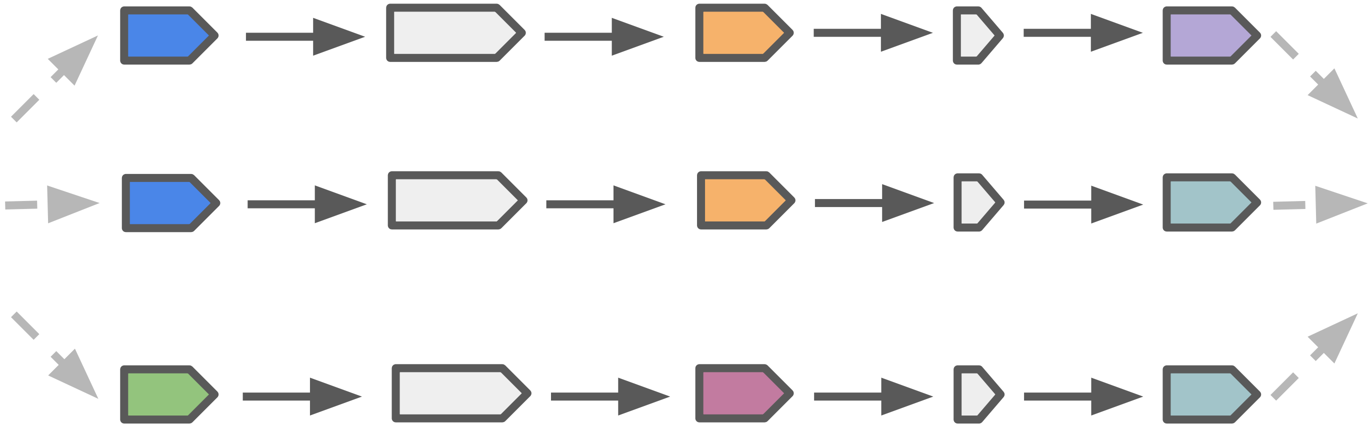
# Iteratively unzipping local junctions



# Iteratively unzipping local junctions



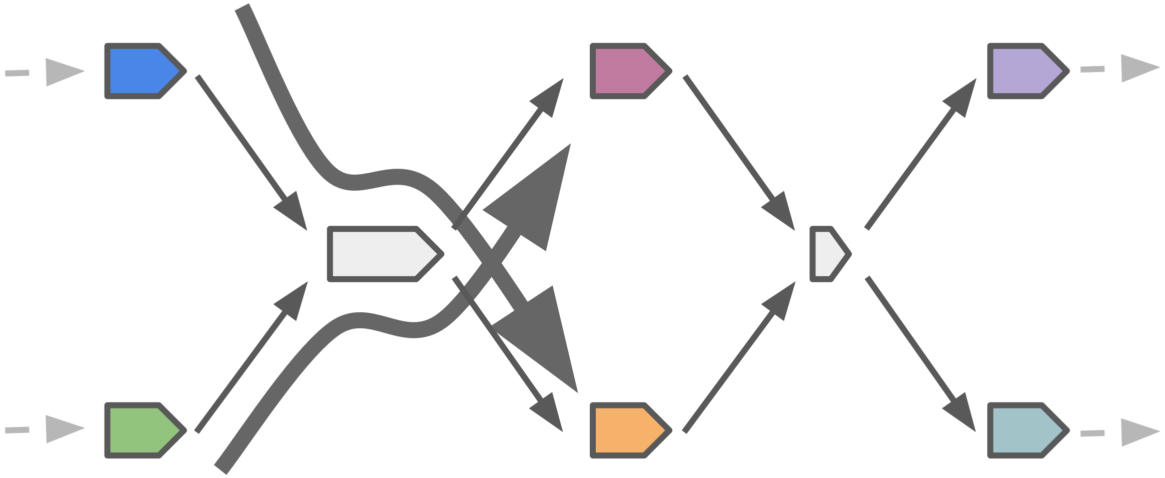
# Iteratively unzipping local junctions

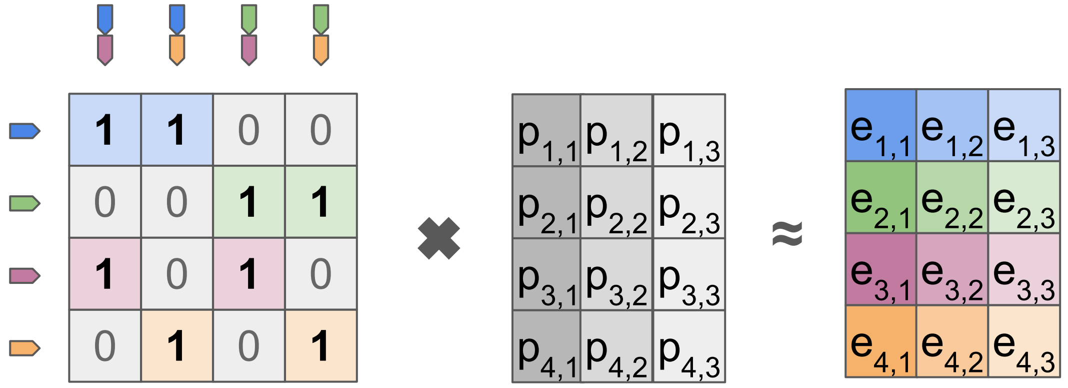


# StrainZip

Assembly Graph Deconvolution for Quantification of Strain-Specific Sequences across Metagenomes

https://github.com/bsmith89/StrainZip

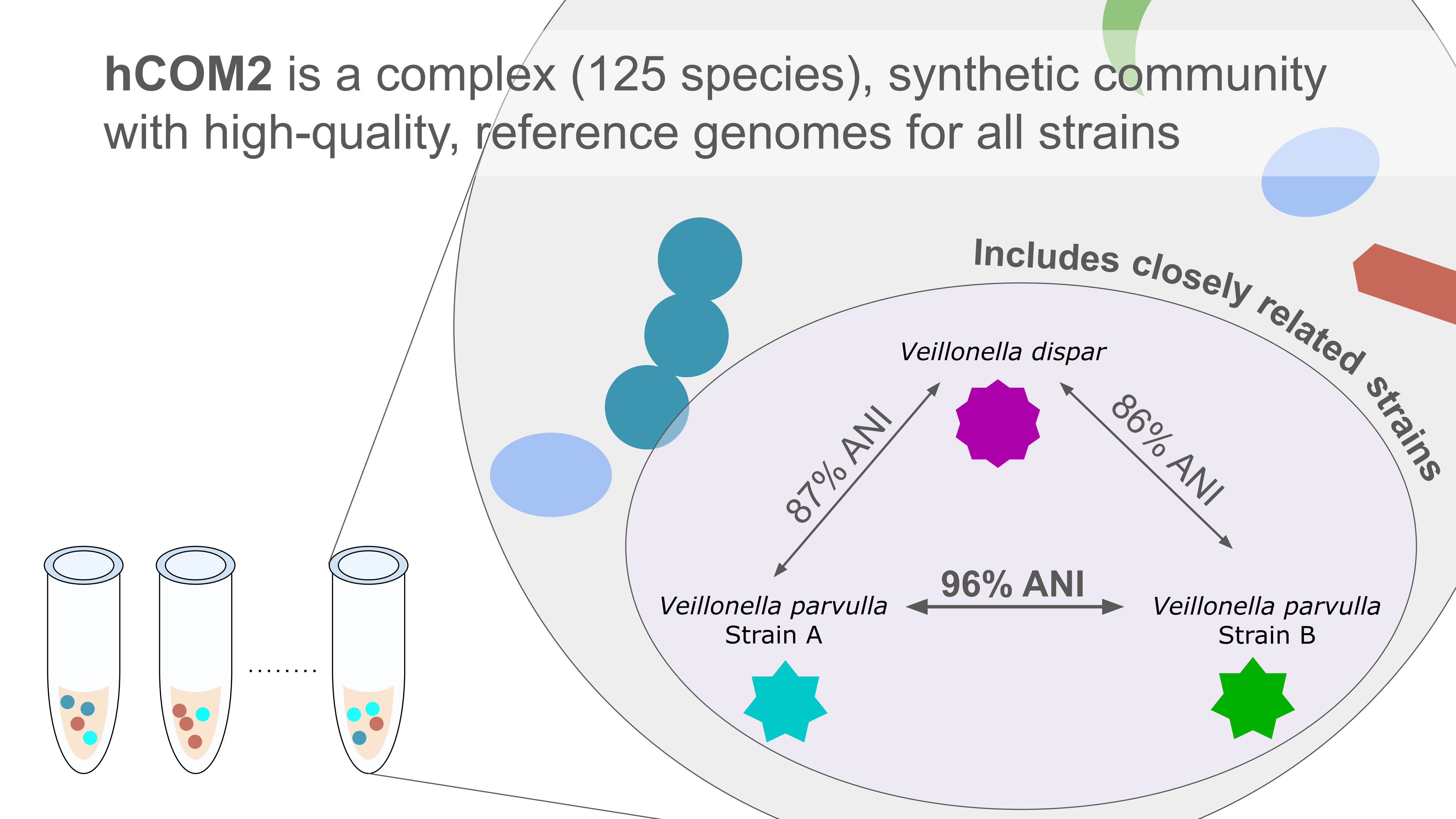




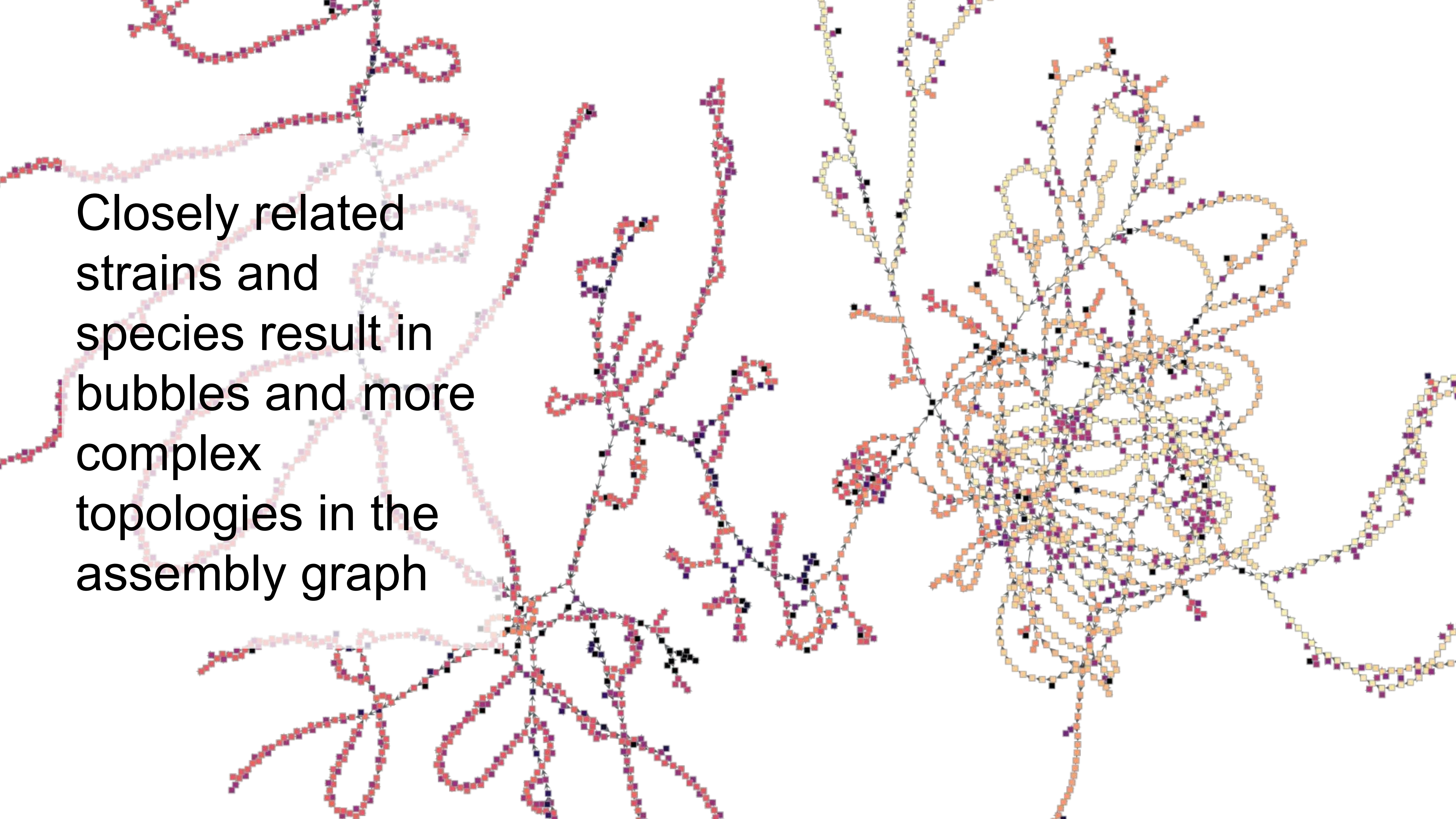
$$\begin{bmatrix}1&1&0&0\\0&0&1&1\\1&0&1&0\\0&1&0&1\end{bmatrix} \times \begin{bmatrix}p_{1,1}&p_{1,2}&p_{1,3}\\p_{2,1}&p_{2,2}&p_{2,3}\\p_{3,1}&p_{3,2}&p_{3,3}\\p_{4,1}&p_{4,2}&p_{4,3}\end{bmatrix} \approx \begin{bmatrix}e_{1,1}&e_{1,2}&e_{1,3}\\e_{2,1}&e_{2,2}&e_{2,3}\\e_{3,1}&e_{3,2}&e_{3,3}\\e_{4,1}&e_{4,2}&e_{4,3}\end{bmatrix}$$

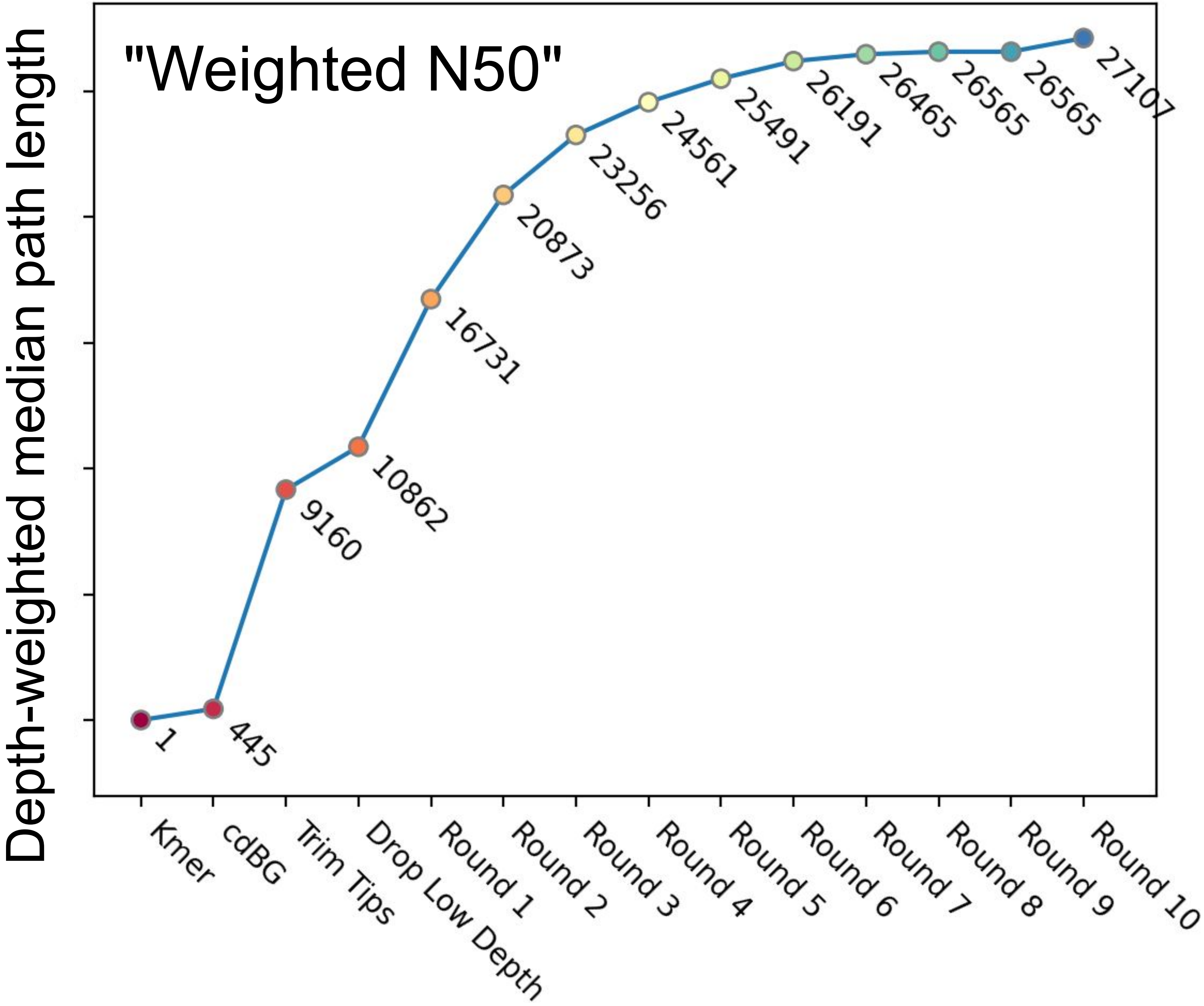
# Benchmarking

**hCOM2** is a complex (125 species), synthetic community with high-quality, reference genomes for all strains



Closely related strains and species result in bubbles and more complex topologies in the assembly graph





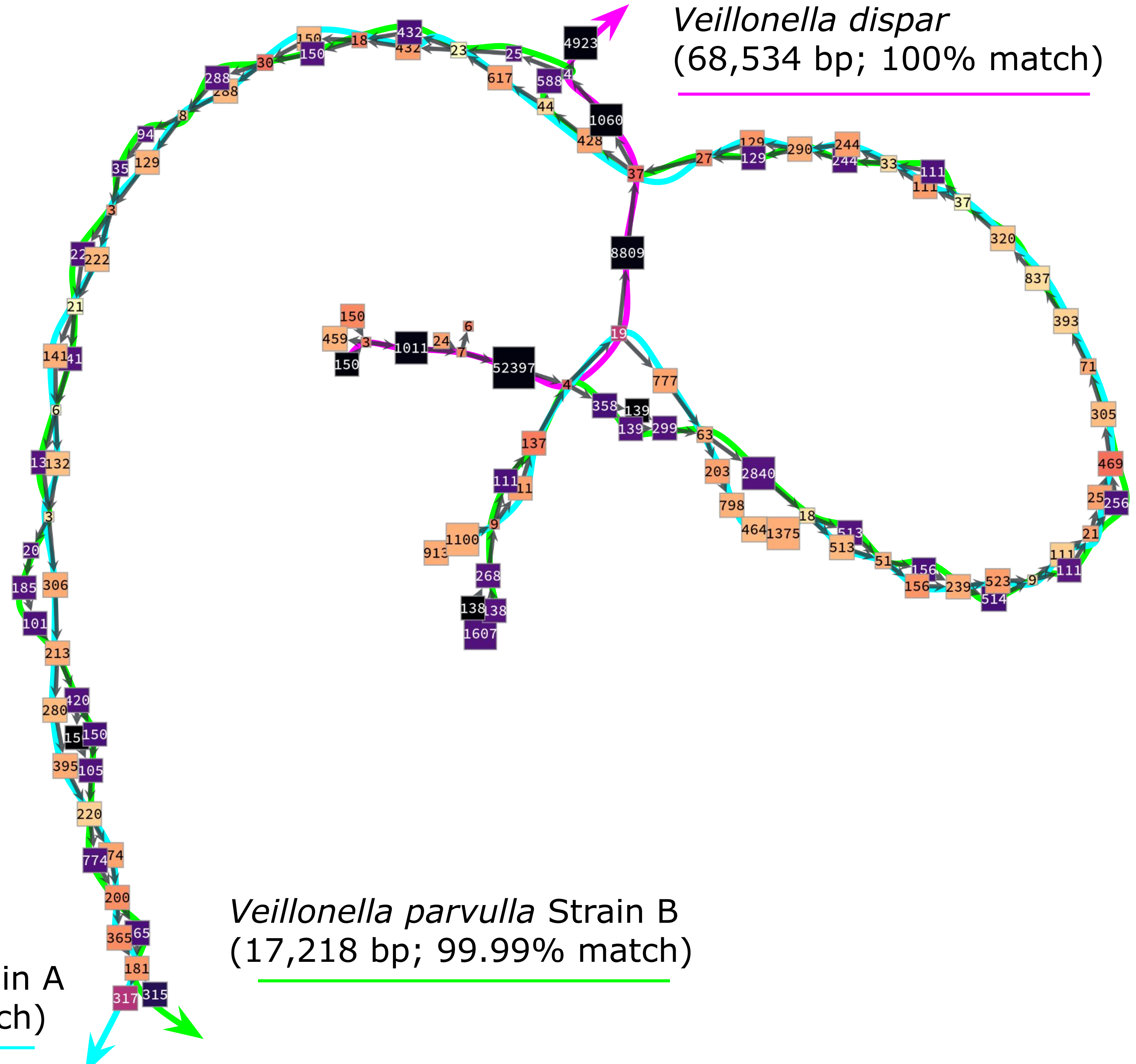
Path lengths increase over successive rounds of deconvolution

Deconvolution recovers longer, strain-specific sequences

…including lower-abundance strains

…and species

**…accurately**

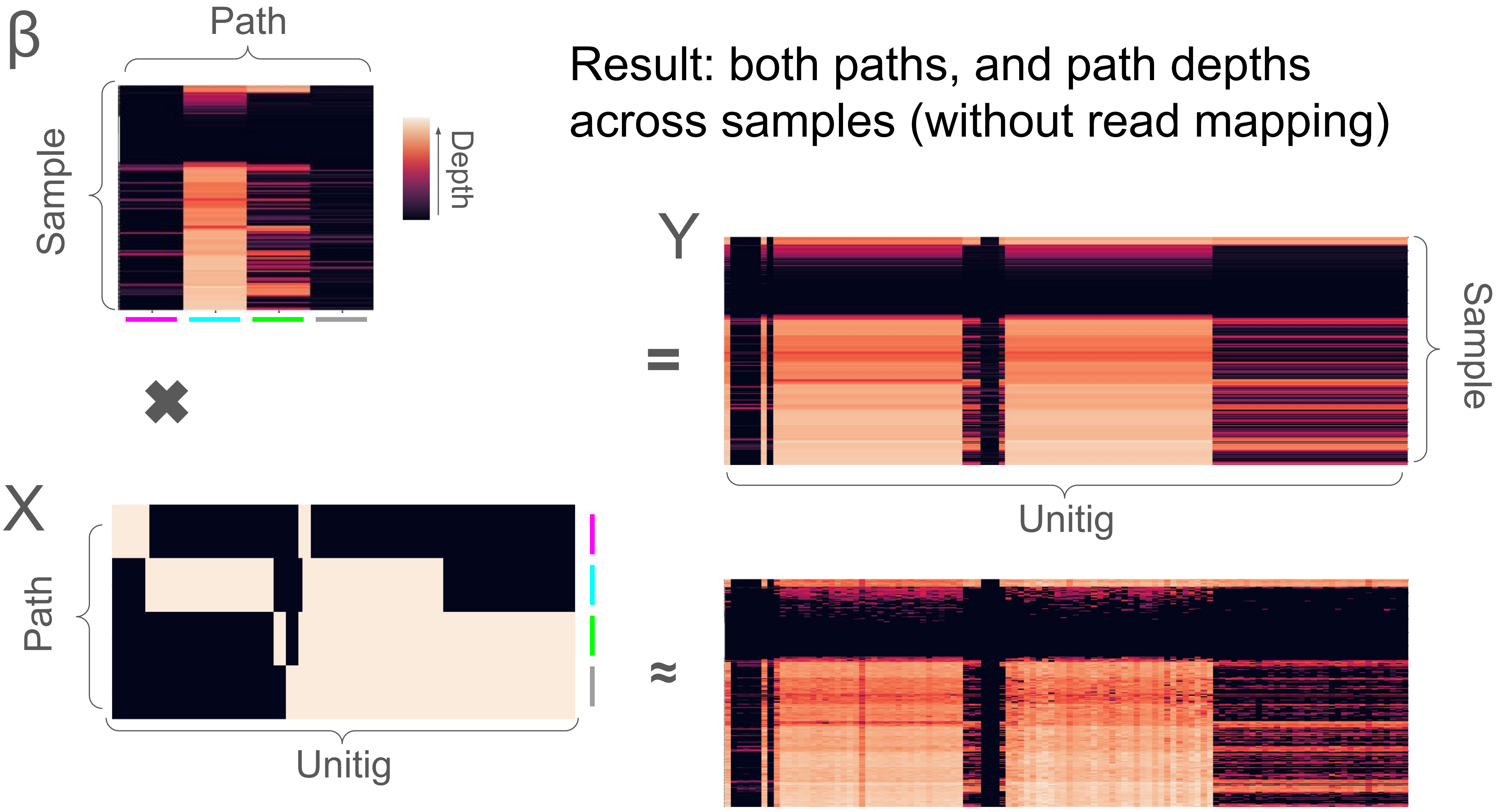


*Veillonella dispar*
(68,534 bp; 100% match)

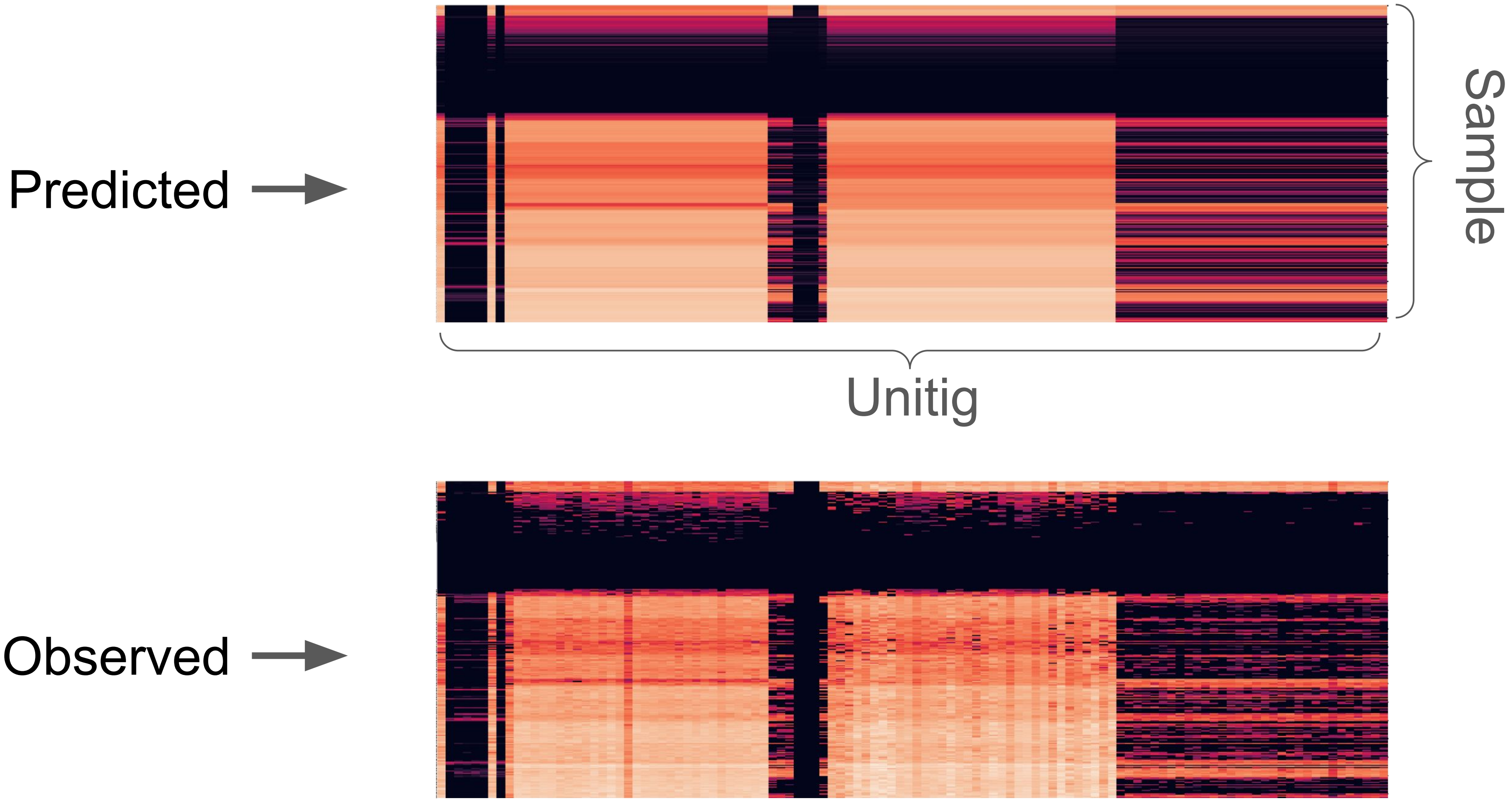
*Veillonella parvulla* Strain B
(17,218 bp; 99.99% match)

*Veillonella parvulla* Strain A
(17,229 bp; 100% match)

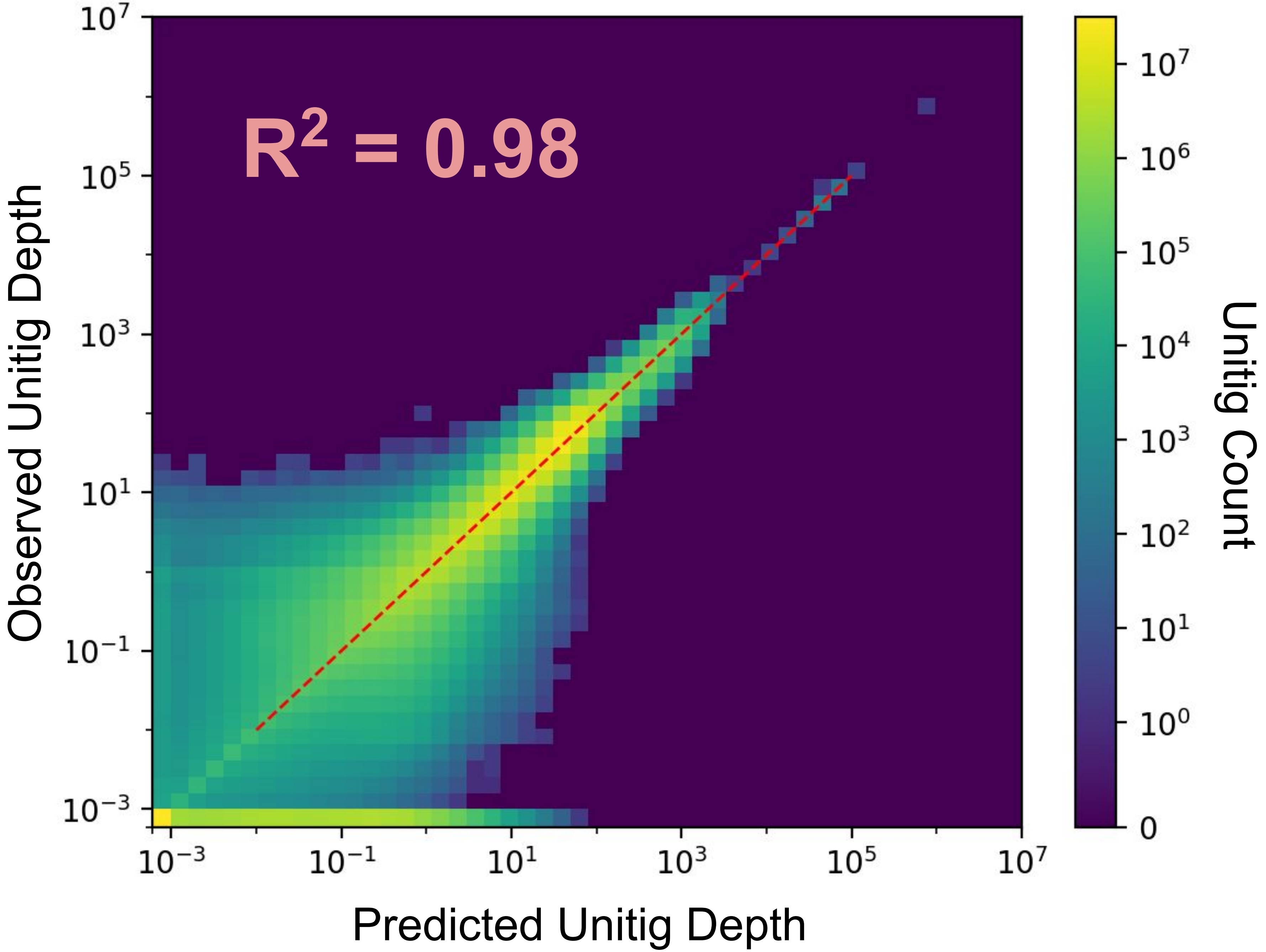
# Result: both paths, and path depths across samples (without read mapping)



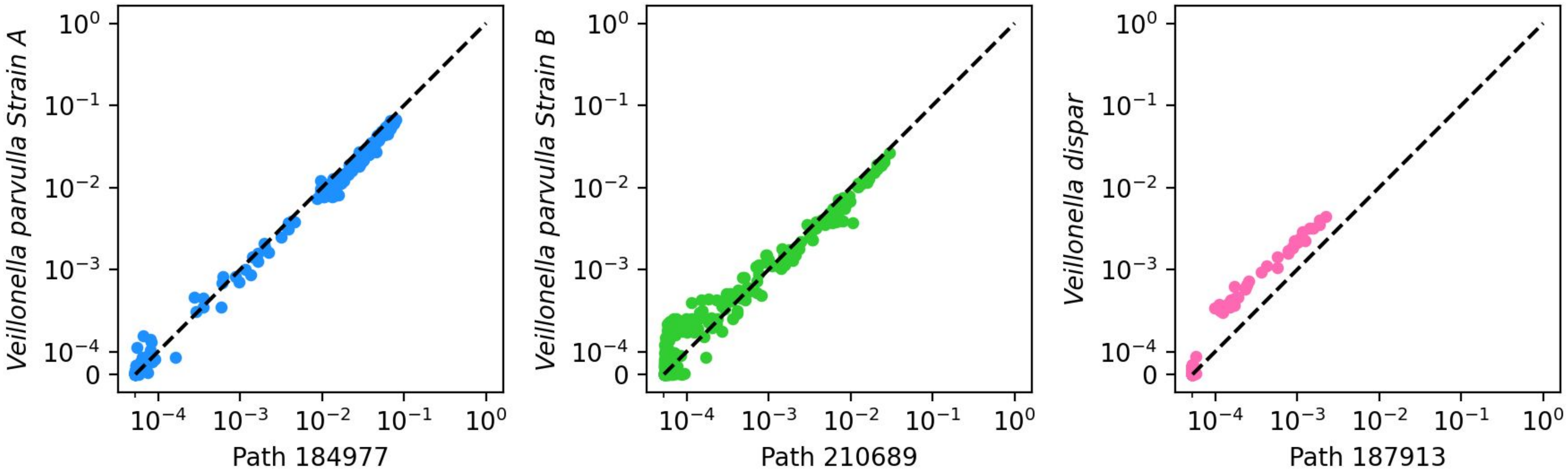
# Estimated unitig depths closely match observed depths



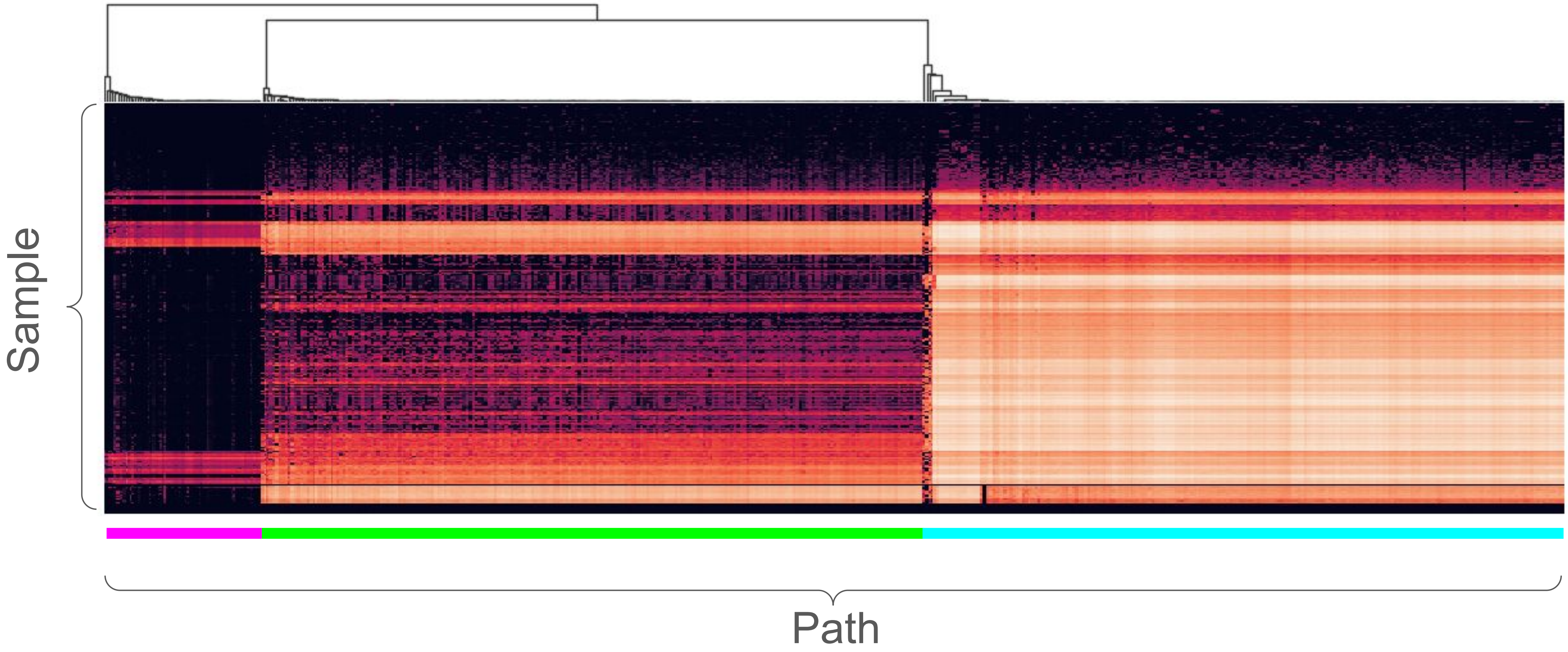
Estimated unitig depths closely match observed depths



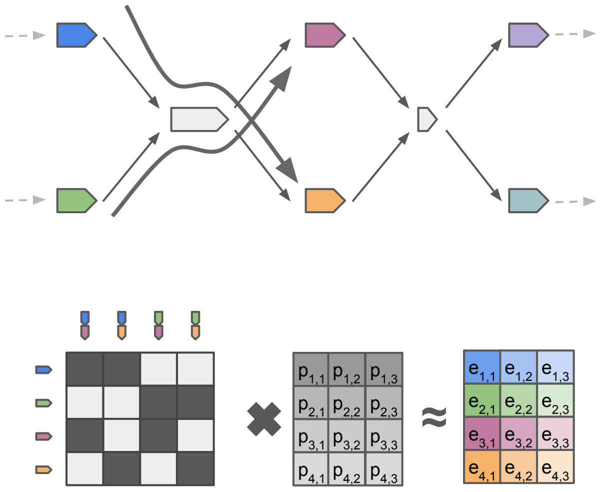
# Path depths match reference-based strain depth estimates



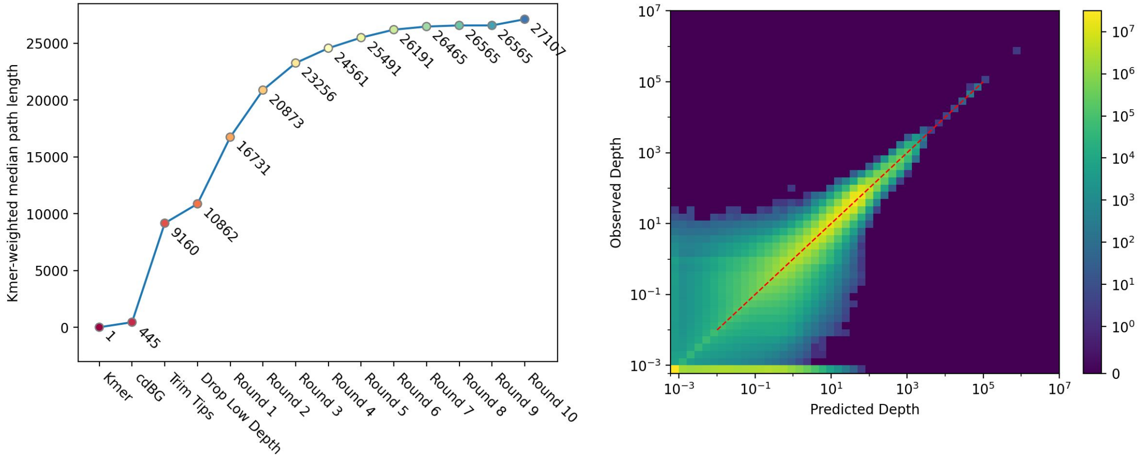
# Clustering paths by depth combines multiple sequences from the same strain



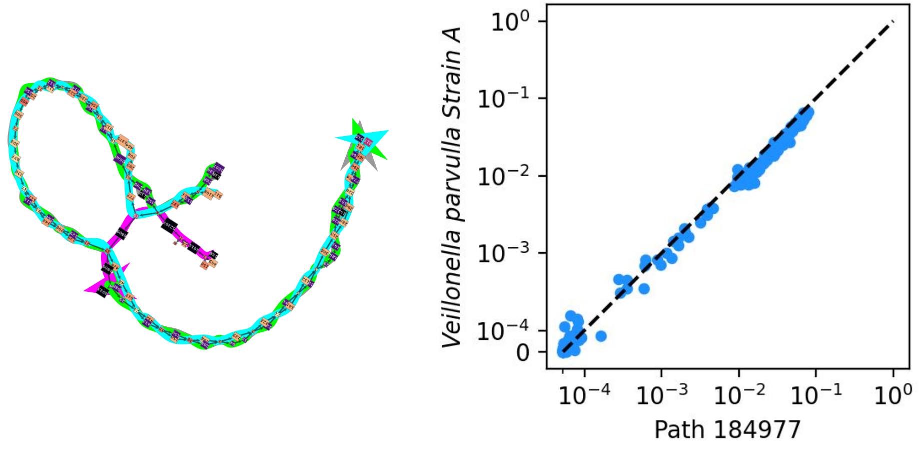
# Iterative Junction Deconvolution



# Combines Assembly, Depth Estimation



# Recovers Closely Related Genomes



# Enables Strain-Resolved Metagenomics