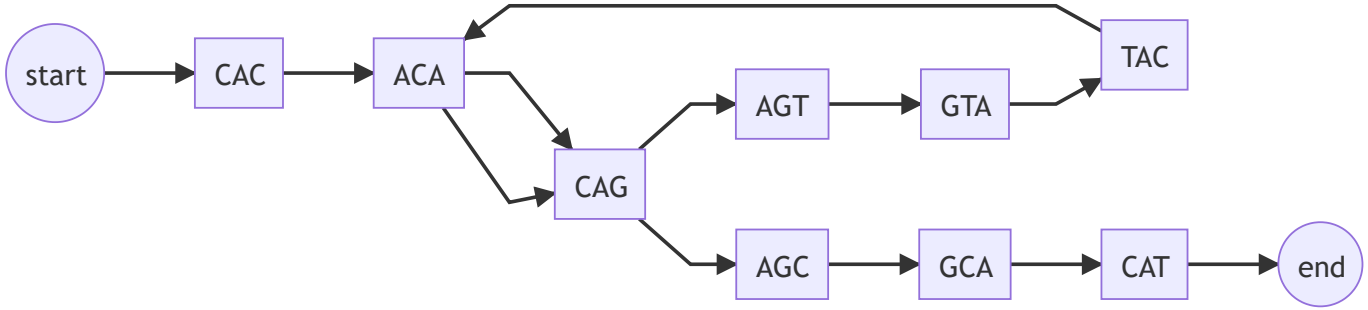
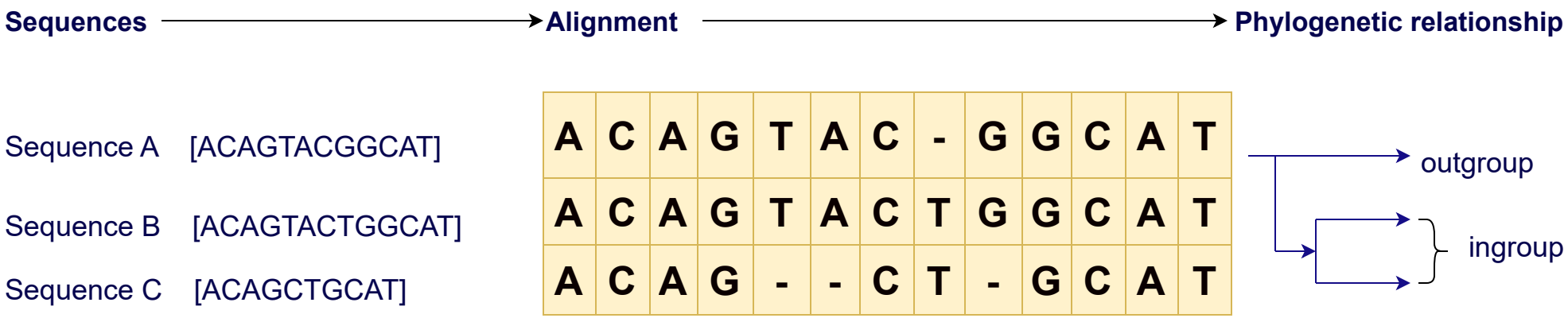
BIOL8706: Dividing and conquering sequence alignment using De Bruijn Graphs



- Student: Richard Morris
- Huttley lab, Australian National University
- Supervisors: Gavin Huttley, Vijini Mallawaarachchi

Australian National University

Sequence alignment



What?: *arranges sequences of DNA, RNA or Protein to identify regions of similarity*

Why?: *Uncover evolutionary relationships between sequences*

How?: *comparing each letter in each sequence with every other letter*

3 big questions

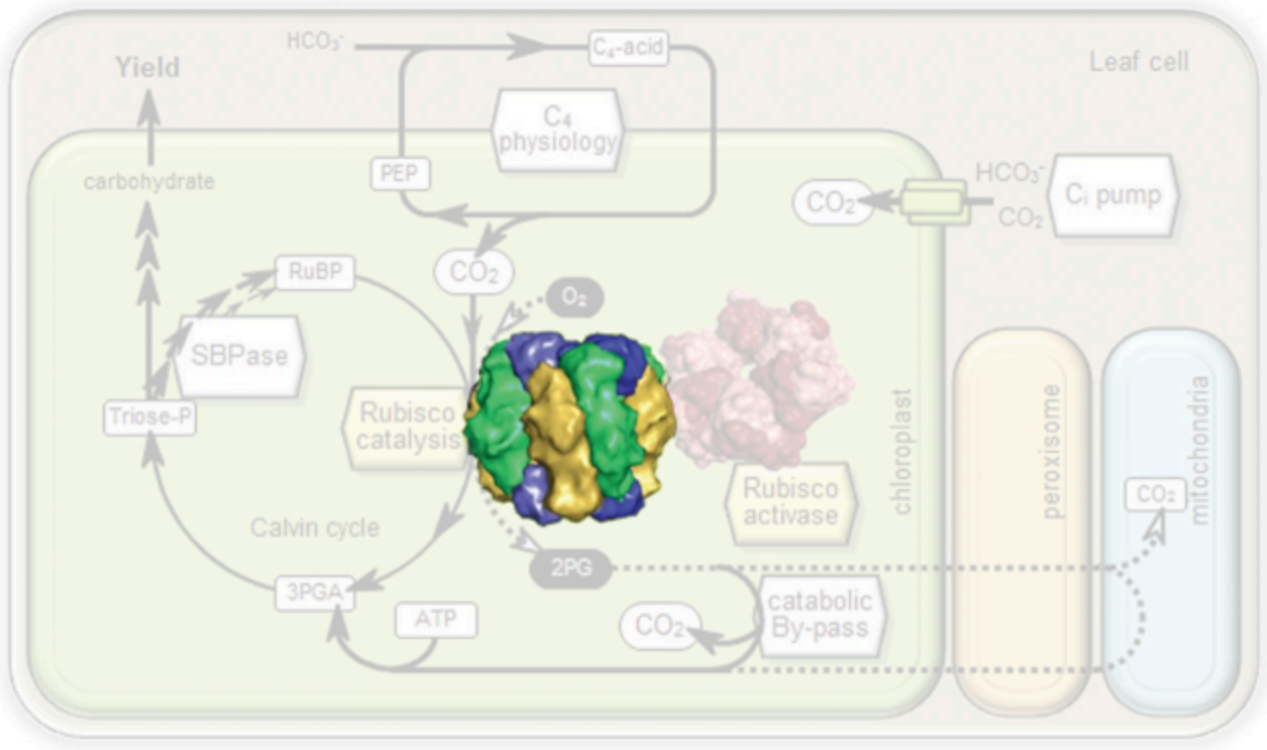
Imagine being able to unravel the path of evolution of any protein

- To design crops for better yield
- To predict the future behaviour of a virus
- Or to understand our own evolution.

These are 3 big questions that require sequence alignment

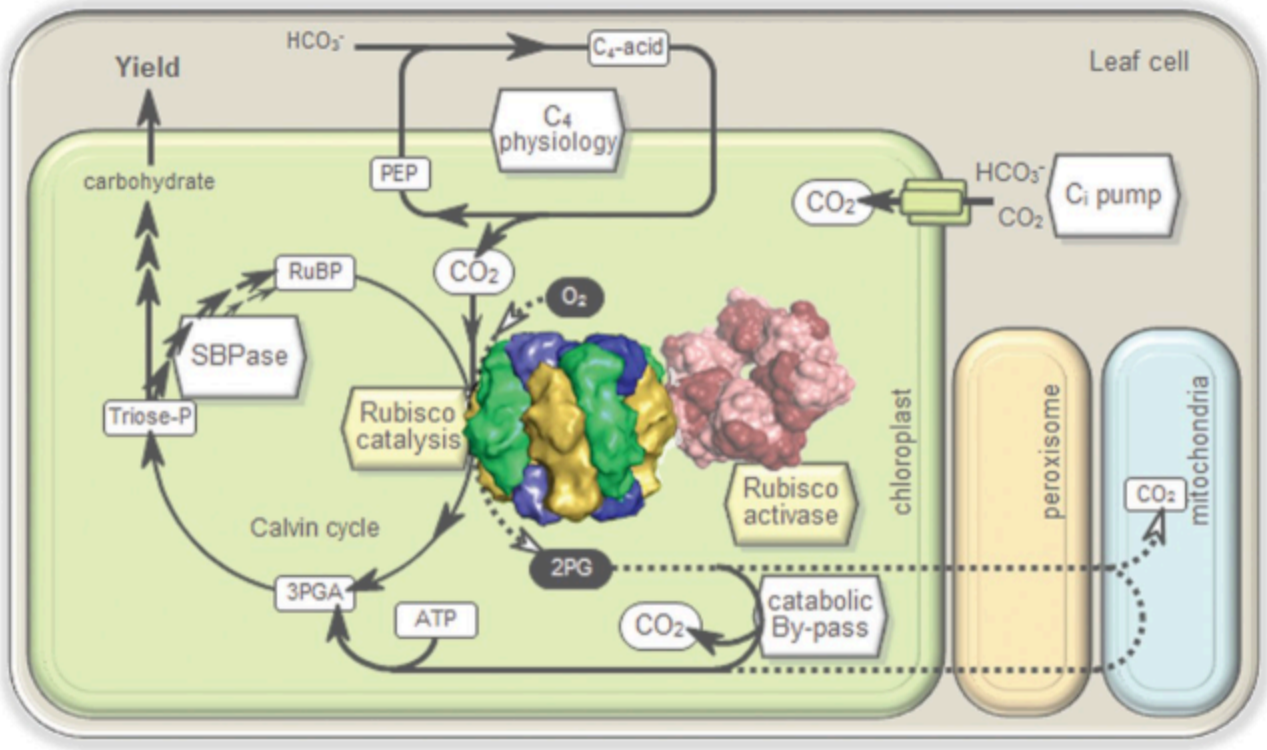
Consider Rubisco

- one of the most abundant proteins on Earth



Consider Rubisco

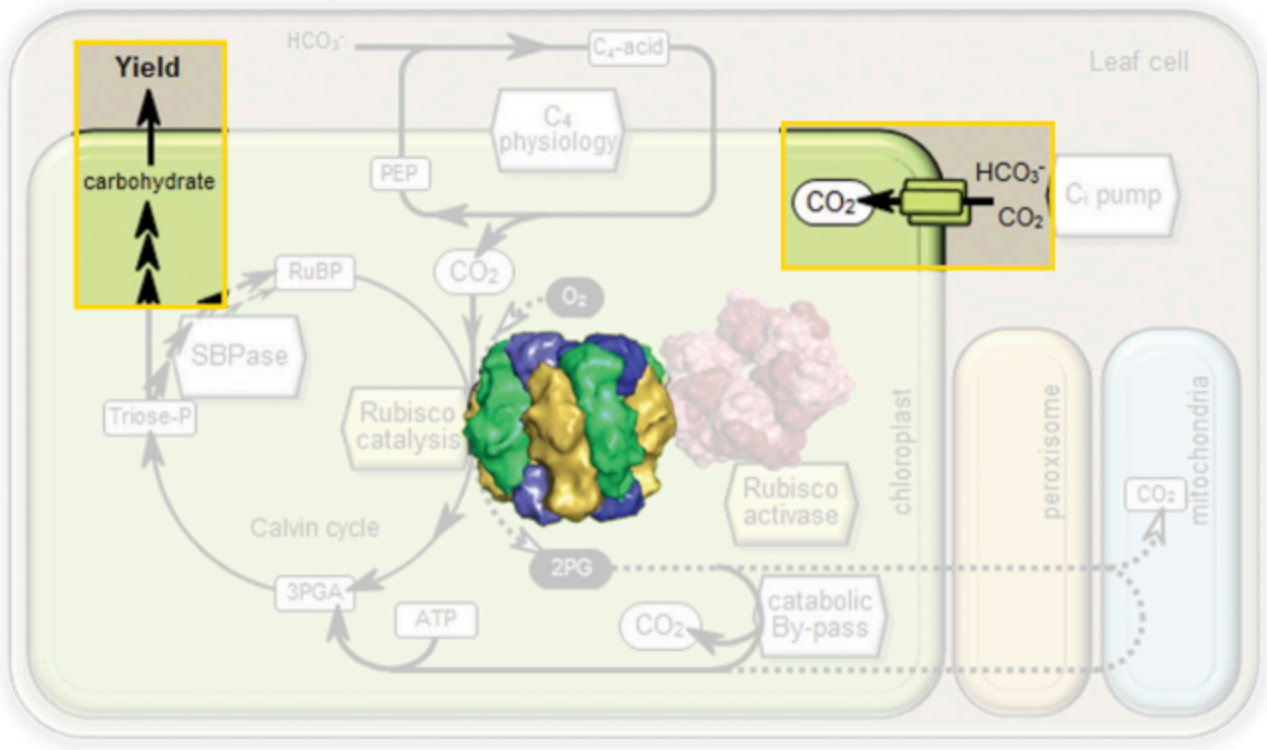
- one of the most abundant proteins on Earth
- **essential component of photosynthesis**



Parry et al 2012 doi.org/10.1093/jxb/ers336

Consider Rubisco

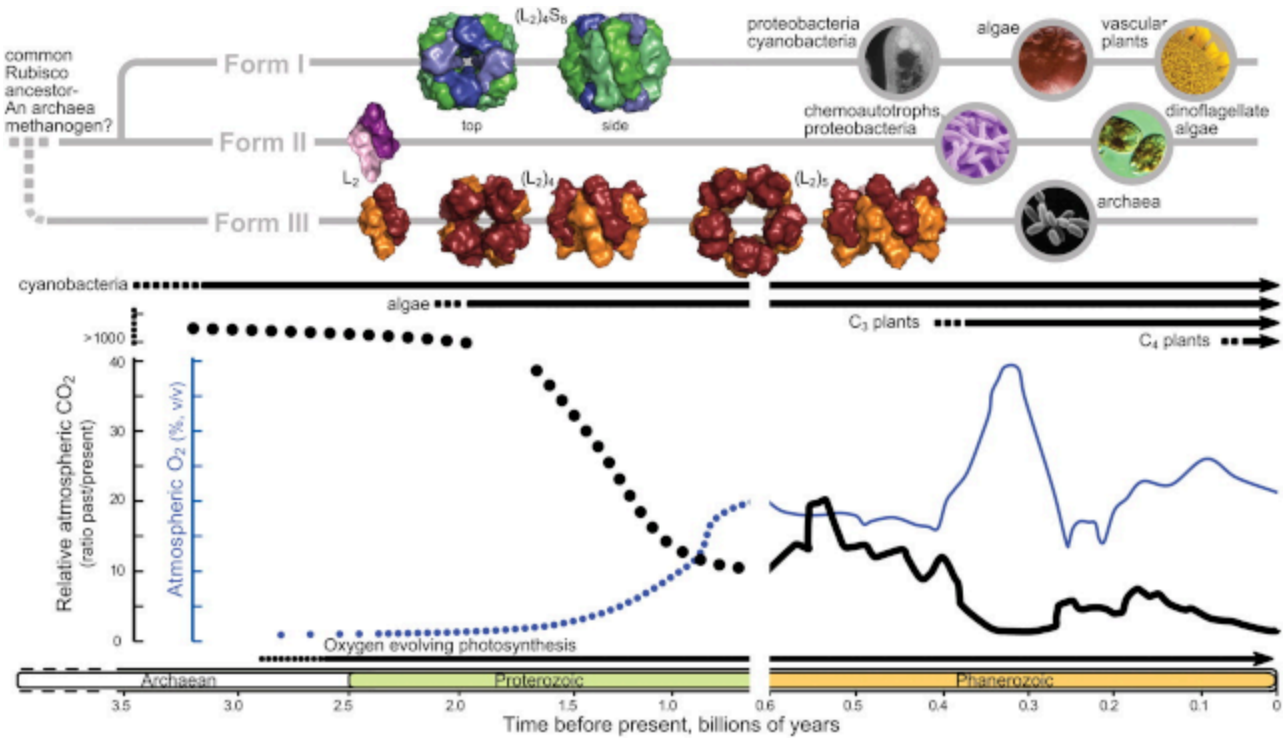
- one of the most abundant proteins on Earth
- essential component of photosynthesis
- **primary role is to convert CO_2 to organic carbon**



Parry et al 2012 doi.org/10.1093/jxb/ers336

Evolution of Rubisco

- Genomic sequencing identified different clades of Rubisco in 3 kingdoms of life
- Phylogenetic analysis suggests when **innovations** in Rubisco's lineage appeared
- We can compare that to the Earths atmosphere at that time



Whitney et al 2012 doi.org/10.1104/pp.110.164814

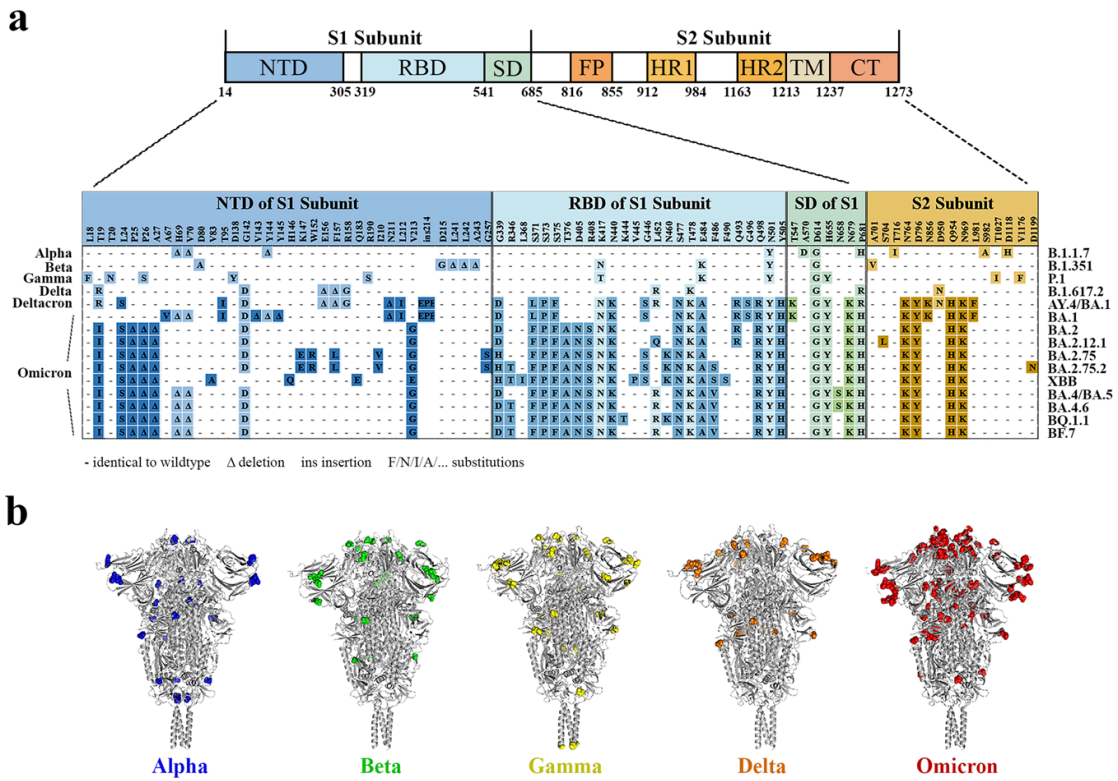
What is the value of understanding Rubisco innovations?

- Rubisco is very slow so plants make a lot of it
- Q: Can we design more efficient Rubisco?
- ↑ Rubisco efficiency would lead to
  - Food crop yield ↑
  - Carbon sequestration ↑
  - Biological hydrocarbon (eg: CH_4) production ↑

Consider the spike protein of SARS-CoV-2

Alignment of eg: a viral sequences allows us to:

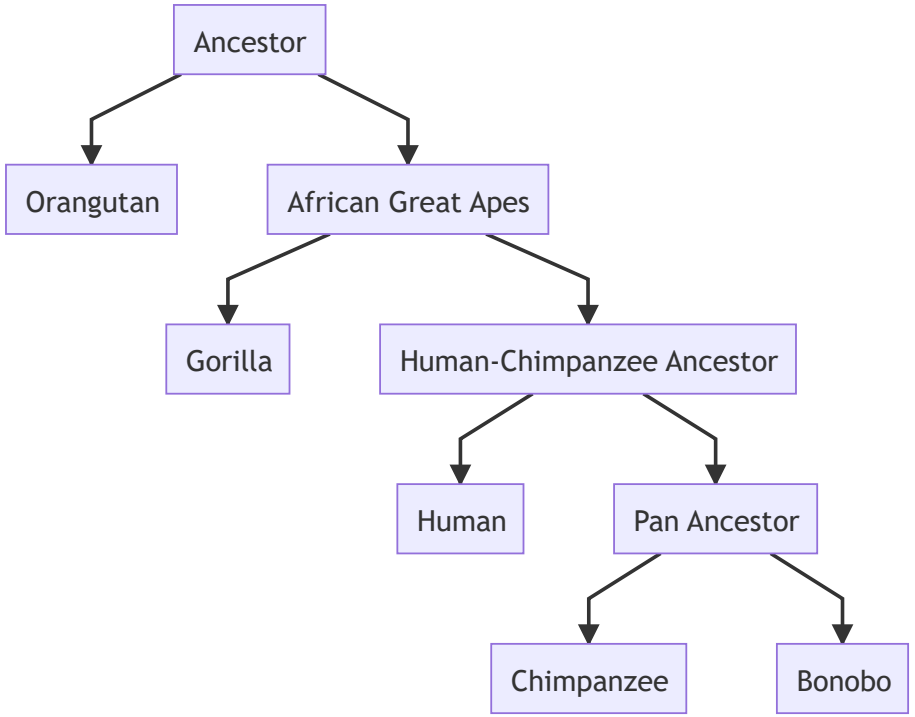
- Identify conserved regions for vaccine/drug development
- Identify changes in function to make predictions about the virus' behaviour
- Identify and prepare for emerging variants



Alignment of S mutation points of SARS-CoV-2 variants

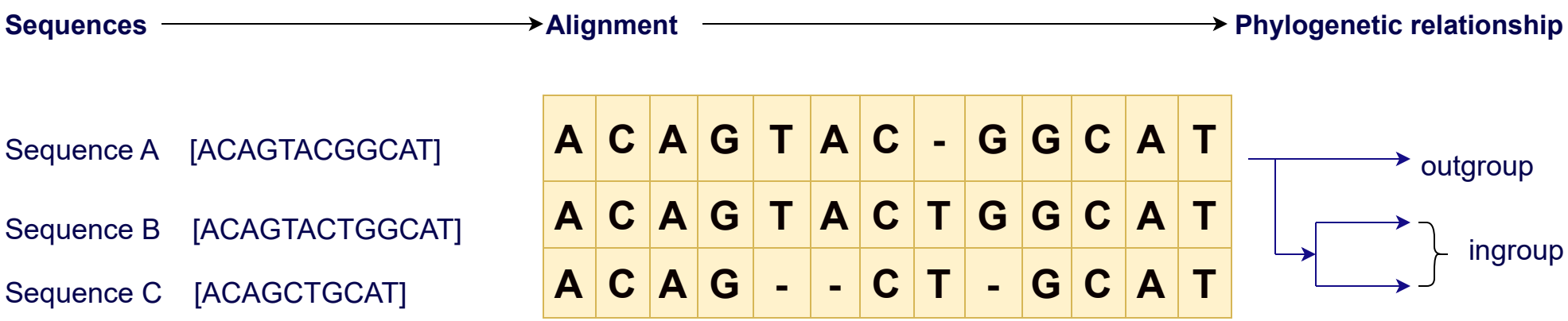
Consider our immediate family

- How do we differ from our closest relatives?
- What was happening when our species diverged one from another?
- What can we learn about our own evolution from our closest relatives?
- Can that knowledge inform biomedical science



The family tree of great apes

How we build a phylogeny from extant sequences



- Sequence alignment is the first step in building a phylogeny
- Exhaustive alignment compares every character in each sequence with every character in every other sequence

Exhaustive alignment takes time

A computational scientist might say that the asymptotic complexity of an exhaustive alignment is given by

$$O(L^n)$$

Where:

- L is the average length of the sequence
- n is the number of sequences

> “ Big-O tells you how code slows as data grows ” *Ned Batchelder (Python guru)*

That's way too much math

Let's rephrase this big-O notation into a more biologically relevant concept of "**Work**"

So we can say that "**Work**" slows as data grows

Sequence length	number of sequences	"Work" required (comparisons)
1,000	2	1 Million
1,000	3	1 Billion
1,000	4	1 Trillion
1,000	5	1 Quadrillion

Let's rephrase this big-O notation into a more biologically relevant concept of "**Work**"

So we can say that "**Work**" slows as data grows

Sequence length	**number of sequences**	**"Work" required (comparisons)**
1,000	3	1 Billion
2,000	3	8 Billion
3,000	3	27 Billion
4,000	3	64 Billion

The scale of our big questions

Genomes	Length (bp)	Number	"Work" required
Rubisco producers	1.5-500 mbp	>350,000[1]	$\text{millions}^{\text{hundreds of thosands}}$
SARS-CoV-2	~29 kbp	>5 million[2]	$29\ \text{thousand}^{5\ \text{million}}$
Great apes	~30mbp	5	$30\ \text{million}^{5}$

[1] ~ 300k species of plants + 10's of thousands of species of algae + thousands of species of cyanobacteria

[2] 5.1 million as of Oct 2021 - www.nature.com/articles/s41588-022-01033-y

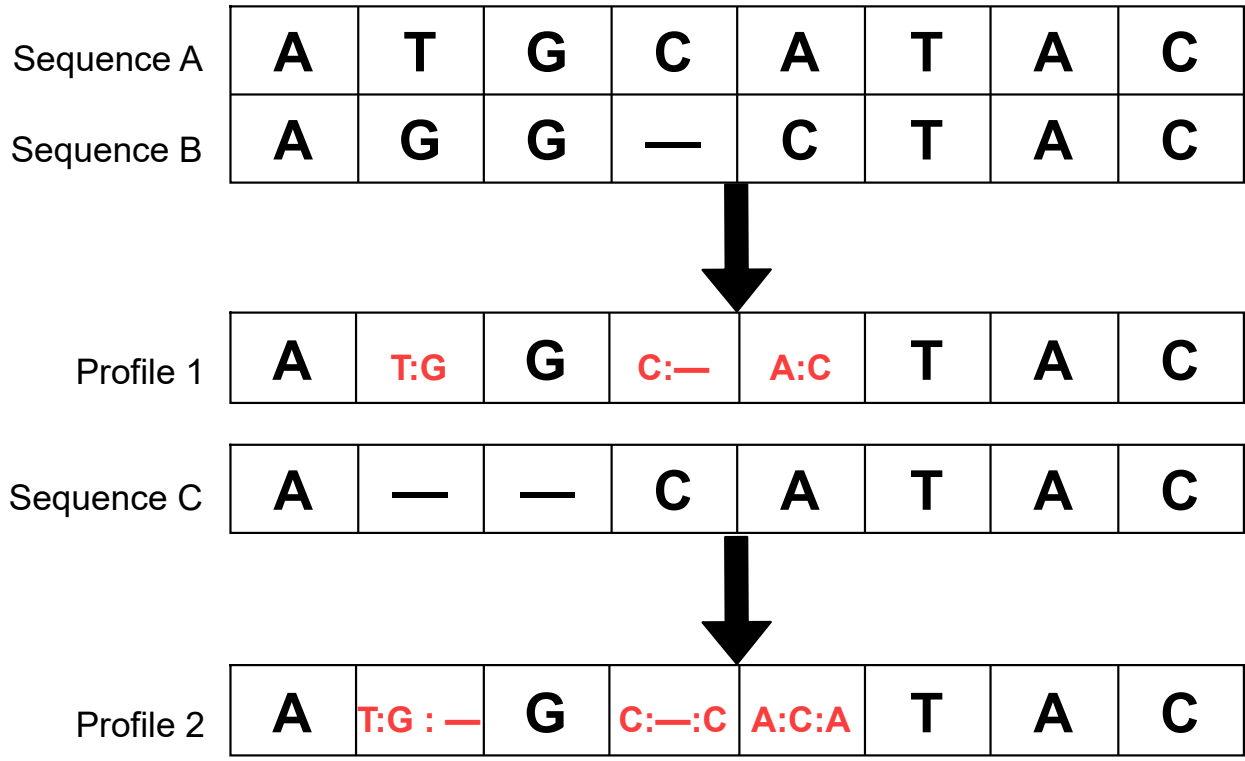
Created with the Imgflip Meme Generator

Progressive alignment

We can reduce the work required as follows

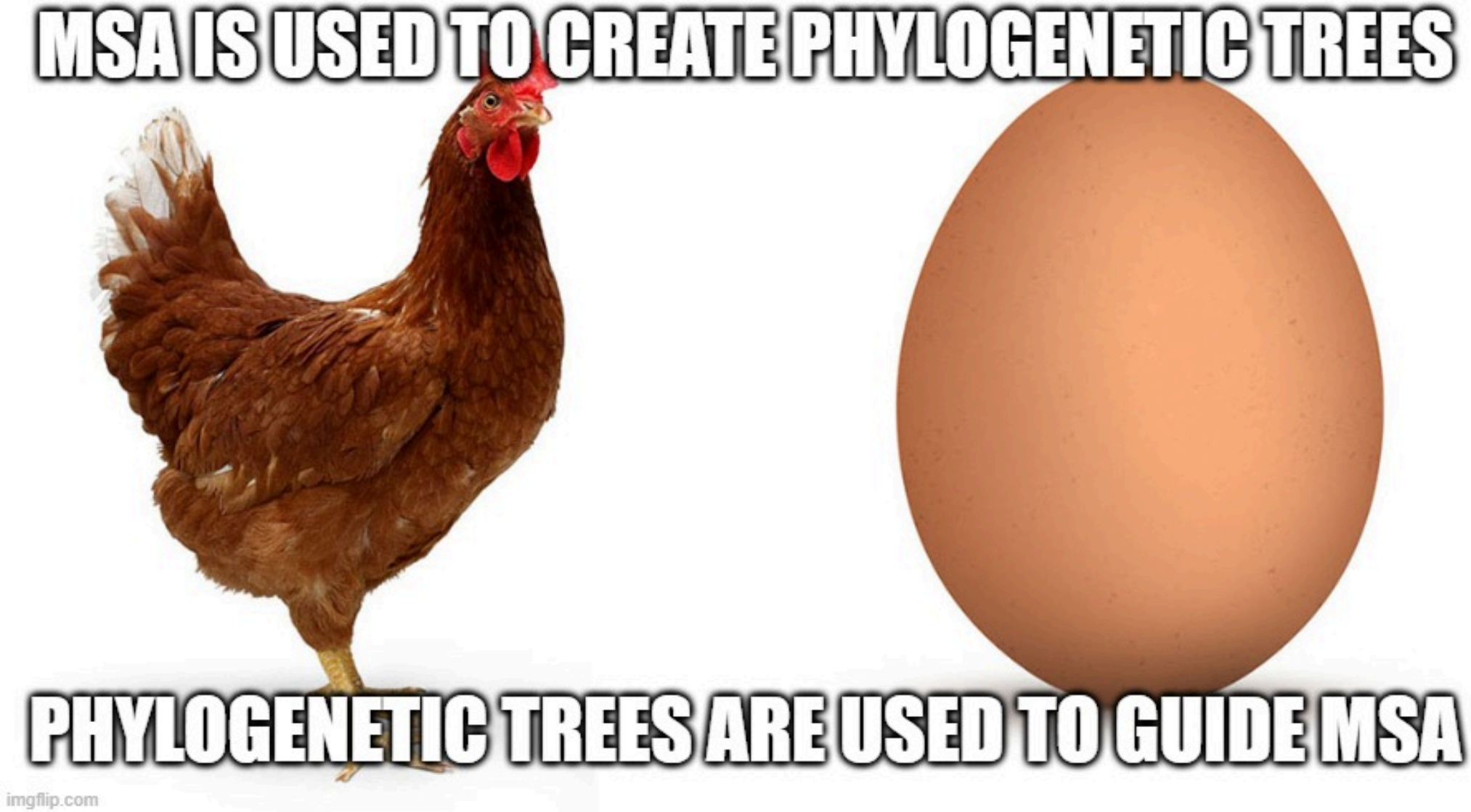
- align the 2 most closely related sequences $O(L^2)$ into a statistical model called a profile
- align that profile with the next most closely related sequence $\binom{n}{2}$ times

This reduces the work required from $O(L^n) \rightarrow O(n^2.L^2)$



Do you see the problem?

- To align multiple sequences first reconstruct a phylogeny so that you can find the 2 most closely related
- To reconstruct a phylogeny first align all sequences


MSA IS USED TO CREATE PHYLOGENETIC TREES
PHYLOGENETIC TREES ARE USED TO GUIDE MSA
imgflip.com

The problem space

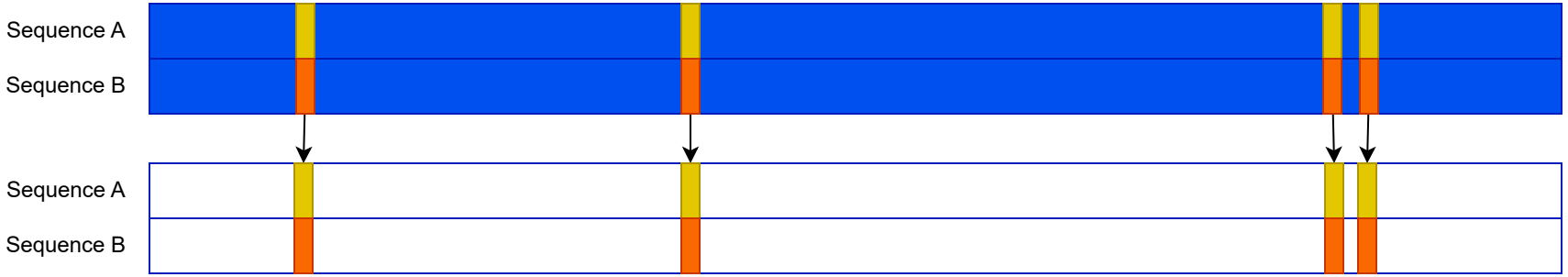
Recall: Sequence alignment is sensitive to

- The **length** of sequences to be aligned
- The **number** of sequences to be aligned
- the “ Chicken and Egg ” problem

An ideal strategy would reduce

- The **length** of sequences to be aligned
- The **number** of sequences to be aligned
- Reliance on knowing the phylogeny in advance

What if we could quickly remove similar regions?



We'd could focus our computational resources on just the regions that differ

Sequence alignment using De Bruijn Graphs

This work builds on the work by Xingjian Leng in 2022, under the supervision of Dr. Yu Lin and Prof. Gavin Huttley.

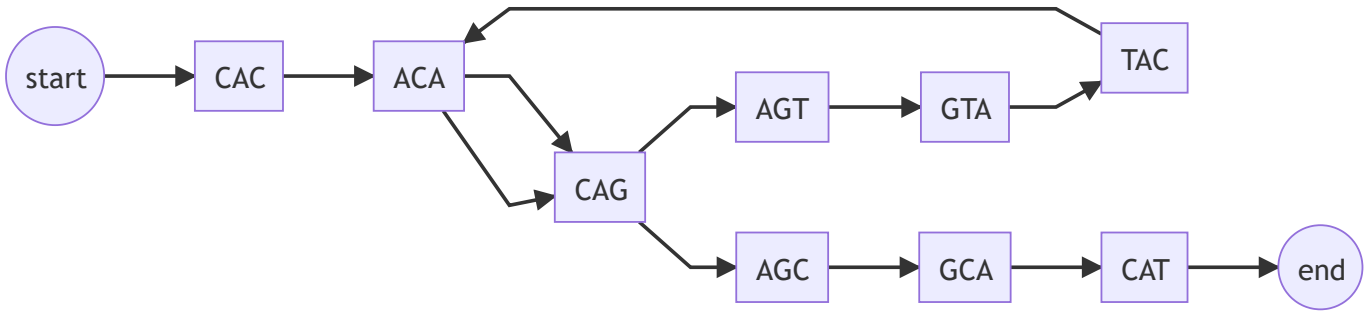
Xingjian tackled the length problem using de Bruijn graphs

De Bruijn graphs

A De Bruijn graph is a directed graph that represents unique overlapping subsequences (or k-mers) at the nodes.

Building a De Bruijn graph has an order complexity of $O(nL)$ in other words "Work" scales linearly not exponentially.

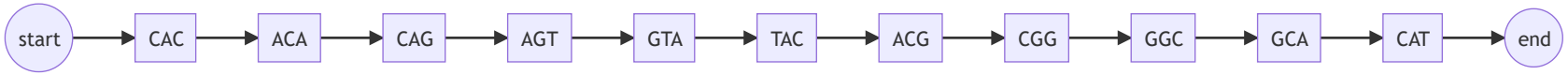
The sequence CACAGTACAGCAT as a de Bruijn graph looks like this;



Reducing the length of sequence to be aligned

Consider the DNA sequence $\boxed{CACAGTACGGCAT}$

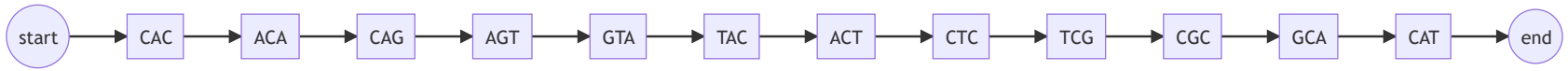
When we represent that as a de Bruijn graph it looks like this:



Reducing the length of sequence to be aligned

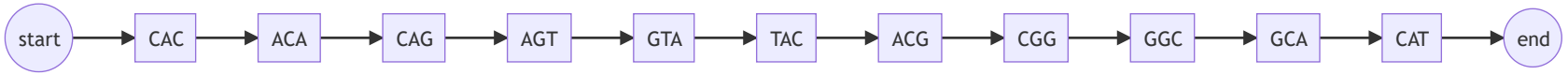
Consider we want to align that sequence $CACAGTAC\boxed{G}GCAT$ to the very similar sequence $CACAGTAC\boxed{T}CGCAT$

Which as a De Bruijn graph looks like this:

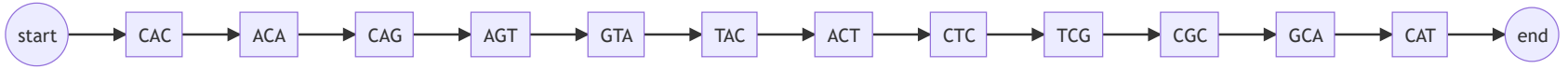


Reducing the length of sequence to be aligned

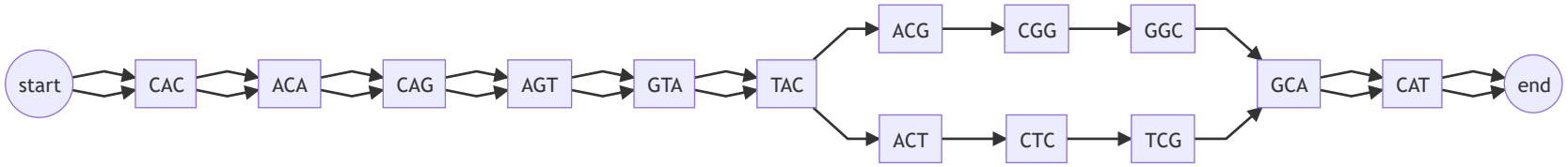
Sequence A:



Sequence B:

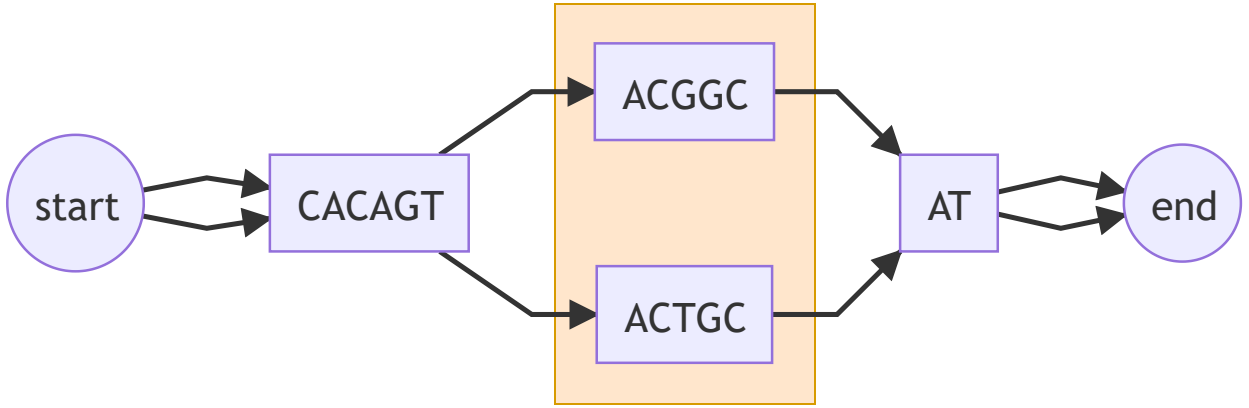


If we combine both sequences into a single de Bruijn graph, it will develop “bubbles” where regions are different.



Reducing the length of sequence to be aligned

We can collect nodes with 2 edge, or 1 edge into single nodes, and we can see the regions that are similar which we don't need to align, and the regions that are different (in the gold box) which we do.

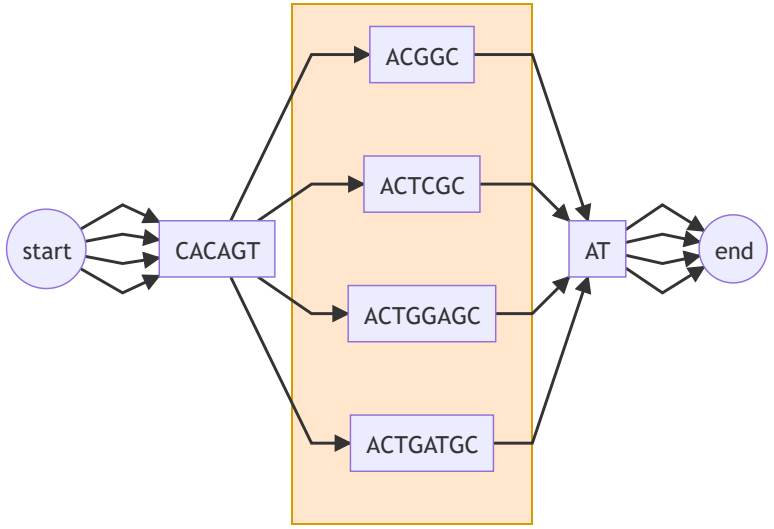


Now we can use a traditional algorithm to align the regions $AC\boxed{G}GC$ and $AC\boxed{T}GC$, and we've reduced our "Work" function from $O(14^2)$ down to $O(5^2)$ = **7.8x** less work.

De Bruijn multiple sequence alignment

And we can extend this to multiple sequences. Consider aligning the following sequences

`CACAGTACGGCAT` `CACAGTACTGCAT` `CACAGTACTGGAGCAT` & `CACAGTACTGATGCAT`



Now we've reduced O(13x13x16x16) down to O(6x6x8x8) = **18.8x** less work

Taking the de Bruijn graph to the next level

- recall an exact alignment has an order complexity of $O(L^n)$
- if we reduce the length of the sequences we need to align we reduce L

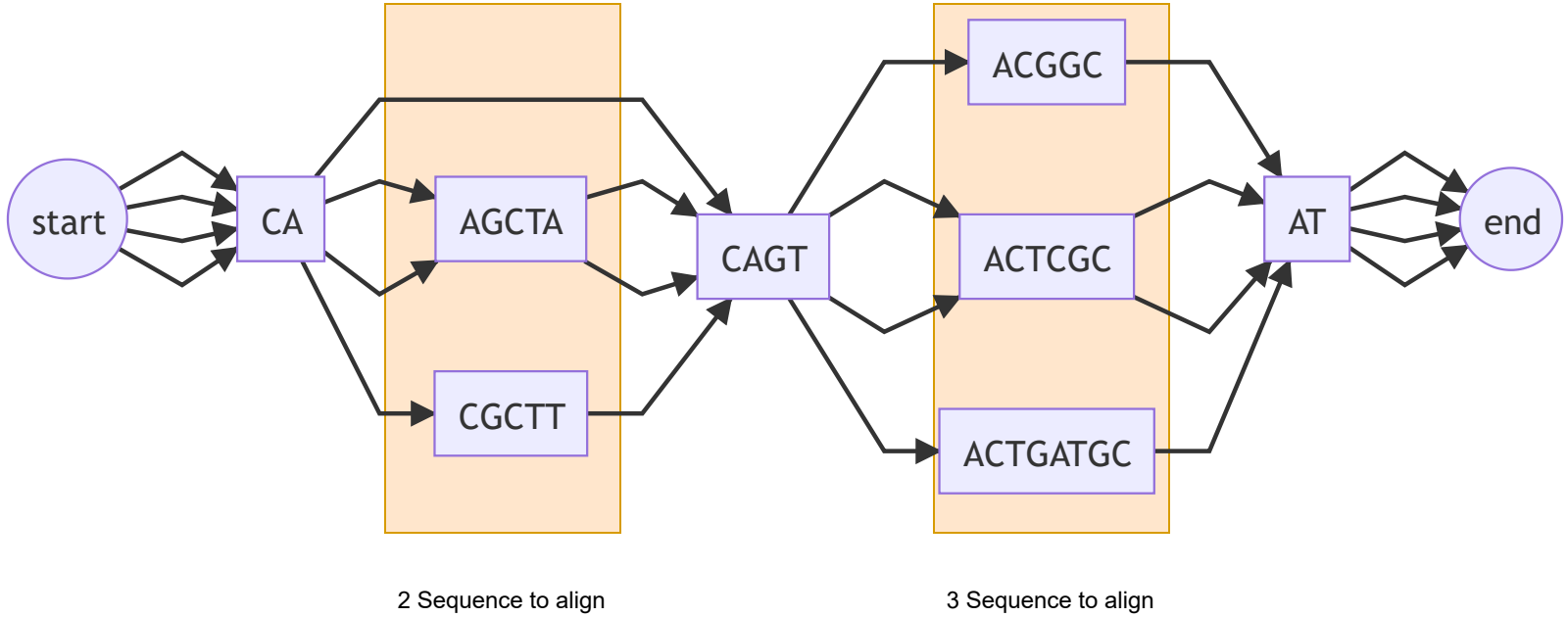
Taking the de Bruijn graph to the next level

- recall an exact alignment has an order complexity of $O(L^n)$
- if we reduce the length of the sequences we need to align we reduce L

How about n?

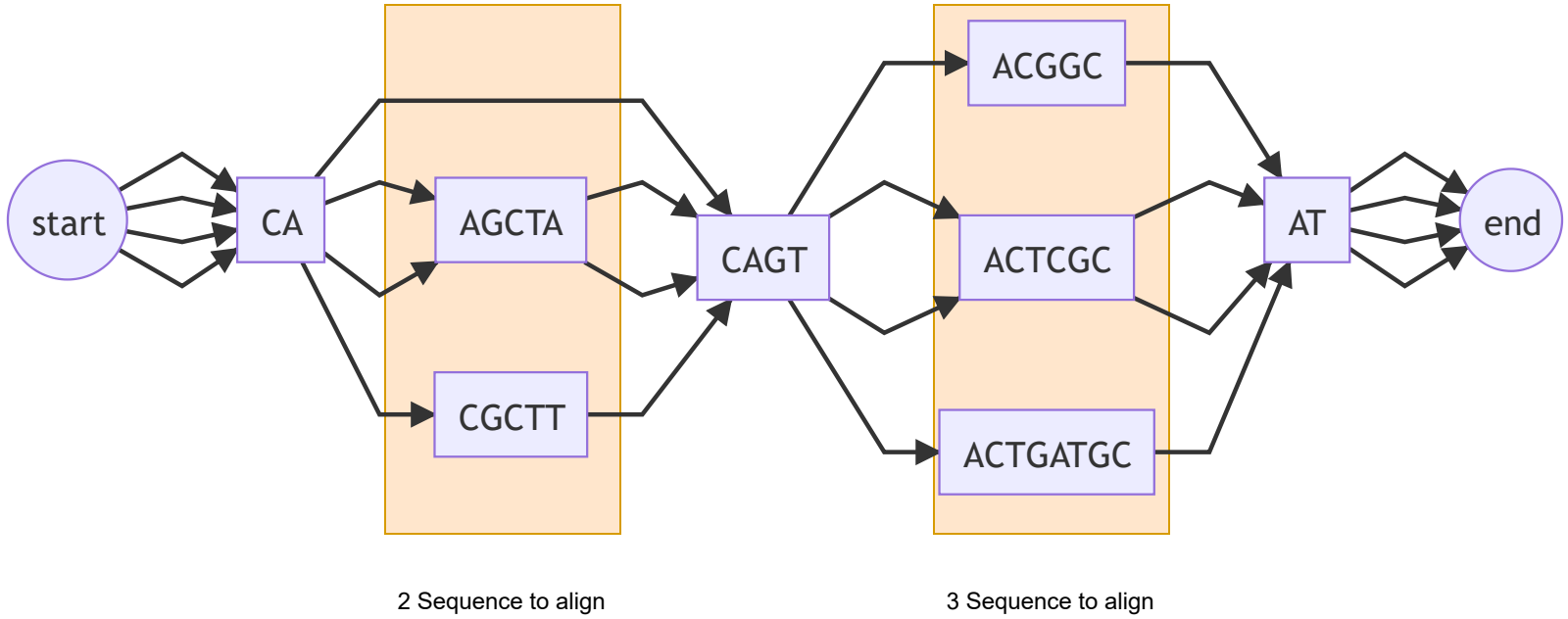
Reducing the number of sequences to be aligned

Consider this de Bruijn graph containing 4 sequences



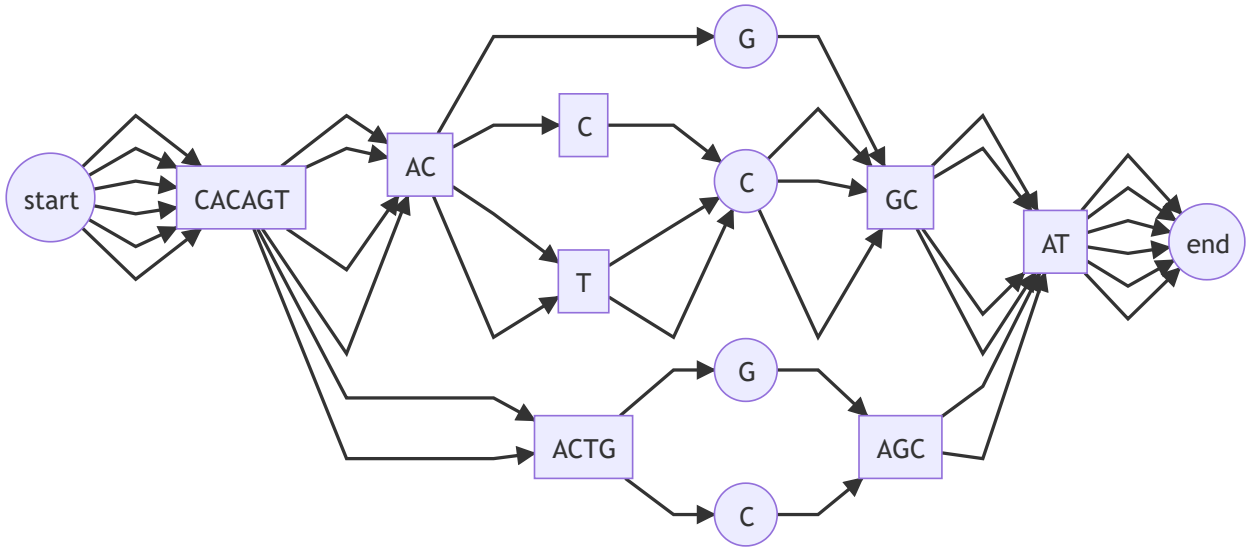
Reducing the number of sequences to be aligned

Consider this de Bruijn graph containing 4 sequences



We don’t have to align 4 sub-sequences at each alignment if sub-sequences that are the same have braided together.

Reducing the reliance on the phylogeny



"Bubbles" that have shorter edges will be more closely related than "bubbles" with longer edges.

By ordering progressive alignment by ascending "bubbles" size, we can progressively align without needing to know in advance the phylogenetic relation between sequences.

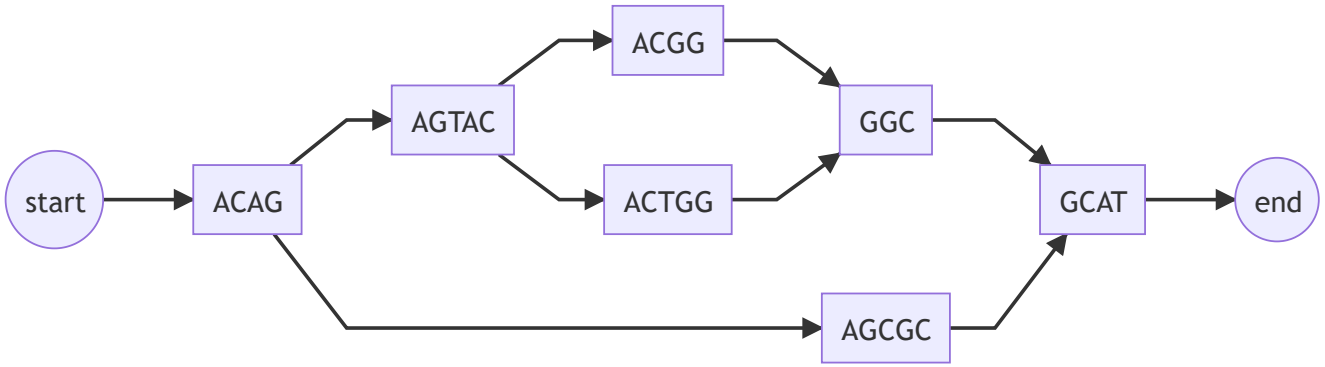
Project aims

- Investigate De Bruijn graphs for multi-sequence alignment (MSA)
- Build a python library
  - Resolve the De Bruijn graph to a partial order graph
  - identify “bubbles”
  - Develop unit tests to verify correctness of the algorithm
- Develop statistics for de Bruijn graphs to predict efficiency

Results: Fastwork statistic

Consider this de Bruijn graph containing 3 sequences [ACAGTACGGCAT, ACAGTACTGGCAT, ACAGCGCAT] of length 12, 13 and 9

When Transformed into a partial order graph



Contains the following nodes (left to right) with overlap removed AC+T+G+C+AT

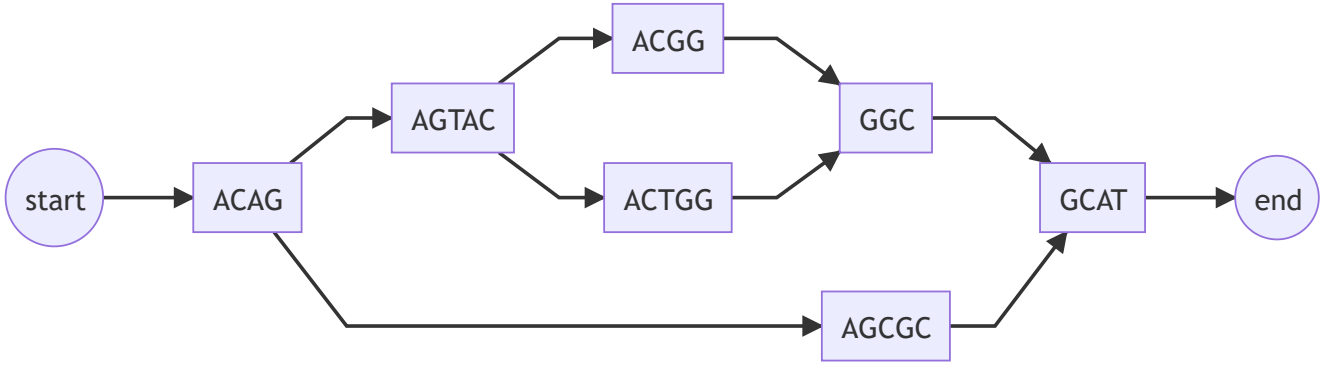
Fastwork = $\sum \text{node length - overlap} = 7$

Fastwork is an estimate of alignment required in the de Bruijn graph

Fastwork has a " Work " function of $O(node\_count)$

Results: Work statistic

Consider the same de Bruijn graph



- Work calculates the order complexity of alignment using 4 strategies
  - Exact = $13 \times 12 \times 9 = 1404$
  - Progressive = $13 \times 12 + 13 \times 9 = 285$
  - DBG_L = $7 \times 8 + 8 \times 1 = 64$ (simplification of sequence length)
  - DBG_LN = $0 \times 1 + 5 \times 1 = 5$ (simplification of sequence length and count)

Results: Calculated from alignable sequences

- BRCA1 genes in 56 species (citation needed)
- BRCA1 genes in primates (citation needed)
- SARS-CoV-2 genomes (citation needed)
- IBD phage components (https://doi.org/10.1016/j.cell.2015.01.002)
- Tara oceans phage components (https://doi.org/10.1126/science.1261605)

Results: Calculated order complexity from alignable sequences

kmer	Genomes	Exact	Progressive	dBG_L	dBG_LN
3	BRCA1 56 species				
3	BRCA1 primates				
3	SARS-CoV-2				
3	IBD phage				
3	Tara oceans phage				

Results: Calculated order complexity from alignable sequences

kmer	Genomes	Exact	Progressive	dBG_L	dBG_LN
6	BRCA1 56 species				
6	BRCA1 primates				
6	SARS-CoV-2				
6	IBD phage				
6	Tara oceans phage				

Results: Calculated order complexity from alignable sequences

kmer	Genomes	Exact	Progressive	dBG_L	dBG_LN
9	BRCA1 56 species				
9	BRCA1 primates				
9	SARS-CoV-2				
9	IBD phage				
9	Tara oceans phage				

Results: Calculated fastwork from alignable sequences

Genomes	dBG(3)	dBG(4)	dBG(5)	dBG(6)	dBG(7)	dBG(8)	dBG(9)
BRCA1 56 species							
BRCA1 primates							
SARS-CoV-2							
IBD phage							
Tara oceans phage							

Sample unit tests: cyclic sequences

```
def test_pog_cycle(output_dir: Path):
    dbg = dbg_align.DeBrujinGraph(3,cogent3.DNA)
    dbg.add_sequence({
        "seq1": "ACAGTACGGCAT",
        "seq2": "ACAGTACTGGCAT",
        "seq3":"ACAGCGCGCAT" # contains cycle
        })
    with open(output_dir / "cycle.md", "w") as f:
        f.write("```mermaid\n")
        f.write(dbg.to_mermaid())
        f.write("```")
    assert dbg.has_cycles()
    assert len(dbg) == 3
    assert dbg.names() == ["seq1", "seq2", "seq3"]
    assert dbg["seq1"] == "ACAGTACGGCAT"
    assert dbg["seq2"] == "ACAGTACTGGCAT"
    assert dbg["seq3"] == "ACAGCGCGCAT" # contains cycle

    dbg.to_pog()
    # write mermaid out to testout folder
    with open(output_dir / "cycle_compressed.md", "w") as f:
        f.write("```mermaid\n")
        f.write(dbg.to_mermaid())
        f.write("```")
```

Discussion

de Bruijn graphs offer an interesting method to

- Break through the tautology at the heart of both Sequence alignment, and Phylogenetic reconstruction
- Reduce the impact of sequence length and sequence number on traditional alignment approaches

This method may make some very big questions tractable

Future directions

Investigate the potential of using de Bruijn Graphs to;

- Identify reverse complimented regions from a dBG
- Identify genetic distance and infer phylogeny from a dBG
- Process sequences in databases storing dBG structures back to the database, reducing active memory limits for large numbers of large sequences
- Investigate advantage wrt species subject to lateral gene flow
  - eg: Bacteria, Archaea
  - identifying multi-rooted phylogenies
- Investigate using dBG's for targeted sequence extraction using pattern recognition templates (start and stop fragments similar to PCR primers)

Thanks

- Gavin Huttley
- Yu Lin
- Vijini Mallawaarachchi
- Xinjian Leng

… and the Huttleylab

Questions

Errata

> Abandon all hope ye who pass this point
>
> *Tolkein … probably*

Sequence alignment order complexity

Pairwise sequence alignment

- Compare every letter in one sequence to every letter in the other
- order complexity of $O(mn)$
  - where **m** and **n** are lengths of the sequences

Multiple sequence alignment (MSA)

- Perform a pairwise alignment of every sequence to every other sequence
- order complexity of $O(L^n)$
  - where **L** is the length of the sequences
  - **n** is the number of sequences

    - scoring system that penalises gaps and mismatches
- Smith-Waterman algorithm: better for local alignment to find conserved domains
    - allows for alignment to reset when the score falls to 0

-1 mismatch, -2 gap (δ)

Where $F(i,j) = \max$ of the following

$$\begin{aligned}
&\nwarrow F(i-1,j-1) + s(A_i, B_j), \quad \text{(match/mismatch)} \\
&\Uparrow F(i-1,j) + \delta, \quad \text{(deletion)} \\
&\Leftarrow F(i,j-1) + \delta, \quad \text{(insertion)}
\end{aligned}$$

	gap	A	G	C	A	
gap	***0***	⇐-2	⇐-4	⇐-6	⇐-8	⇐
A	⇑-2	⇖ ***1***	⇐ -1	⇐-3	⇐-5	⇐
C	⇑-4	⇑-1	⇖0	⇖ ***0***	⇐-2	⇐
G	⇑-6	⇑-3	⇖0	⇖-1	⇖ ***1***	⇐
A	⇑-8	⇖⇑-5	⇑-2	⇖-1	⇑-1	⇖
A	⇑-10	⇖⇑-7	⇑-4	⇖⇑-3	⇖-2	⇖

backtrace from bottom right selecting the

Multiple sequence alignment (MAS) strategies

Dividing and conquering sequence alignment using De Bruijn Graphs

- Pairwise alignment of each possible pair
  - $\binom{n}{2} \times O(L^2) = \frac{n(n-1)}{2} \times O(L^2) = O(n^2.L^2)$
- Progressive alignment eg: ClustalW
  - create a guide tree
  - Progressively align pairs most closely related to profiles, and then align profiles
- Iterative methods eg: MUSCLE, T-Coffee, MAAFT
  - create an preliminary fast less accurate alignment
  - iteratively improve alignment using some scoring function
  - Complete when some convergence criterion is met
- Hidden markov models $O(nL) + O(LM)$ (M is the number of states in the model)
  - eg: HMMER
  - create a statical model of the transition between states

Unit tests

library against edge case sequence alignments

* long sequences
* numerous sequences
* cyclic sequences
* bubbles within bubbles
* sequential bubbles