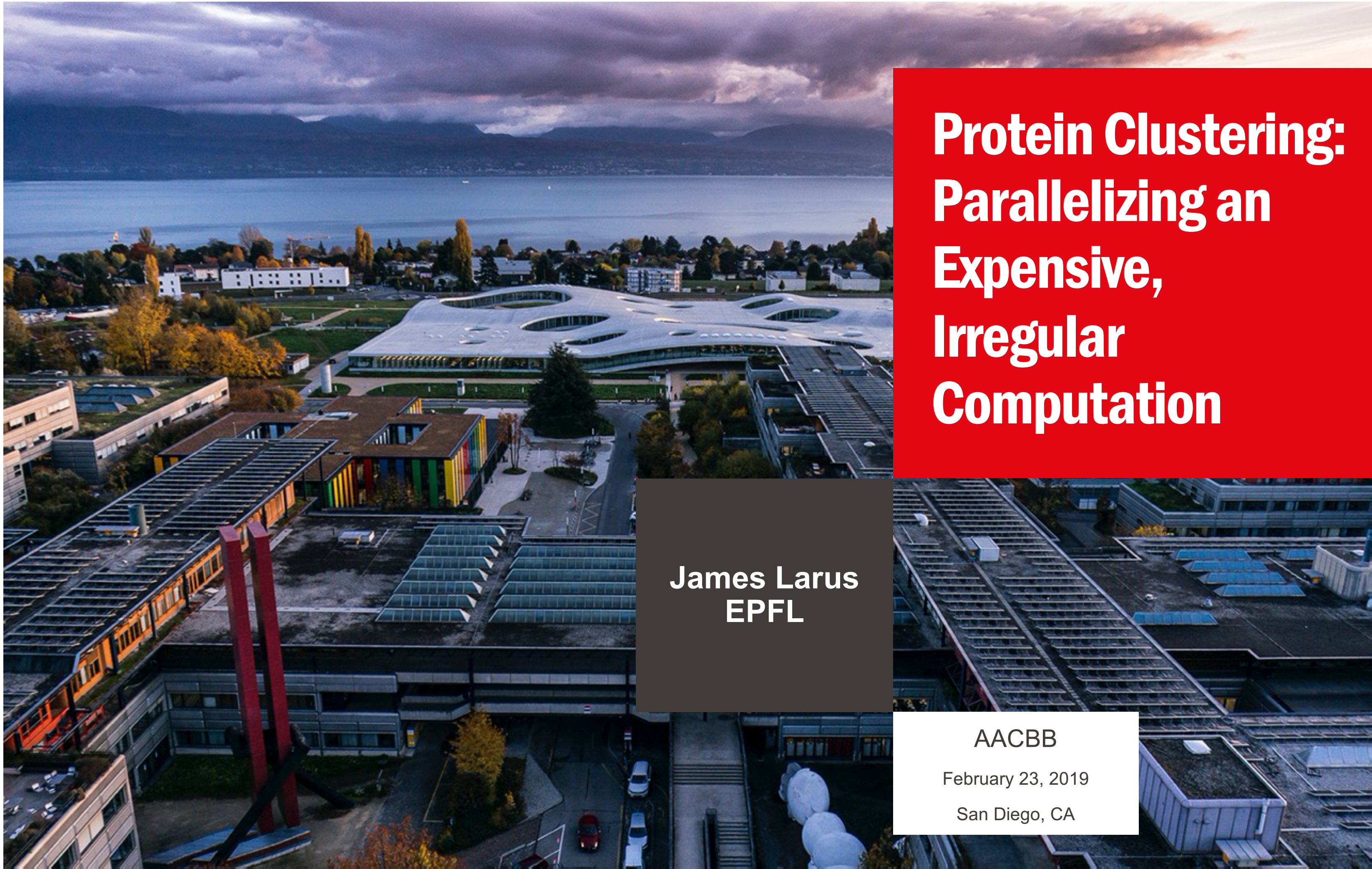
EPFL



Protein Clustering: Parallelizing an Expensive, Irregular Computation

James Larus
EPFL

AACBB

February 23, 2019

San Diego, CA

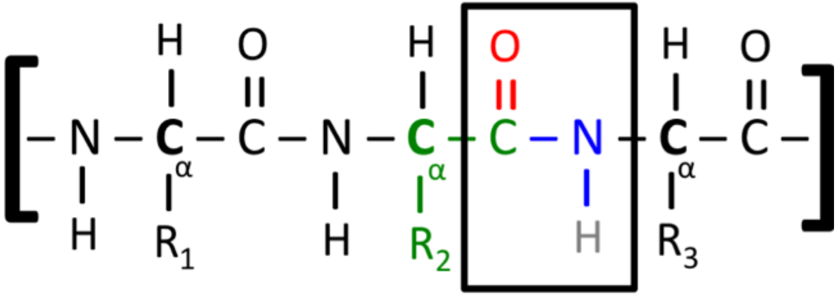
École
polytechnique
fédérale
de Lausanne

PhD research

Stuart Byma

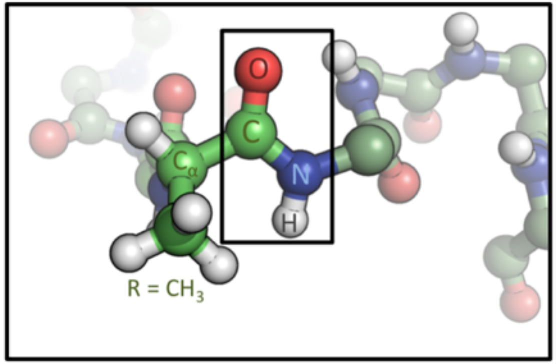
PhD “Parallel and Scalable Bioinformatics”, April 2020

What’s a protein?



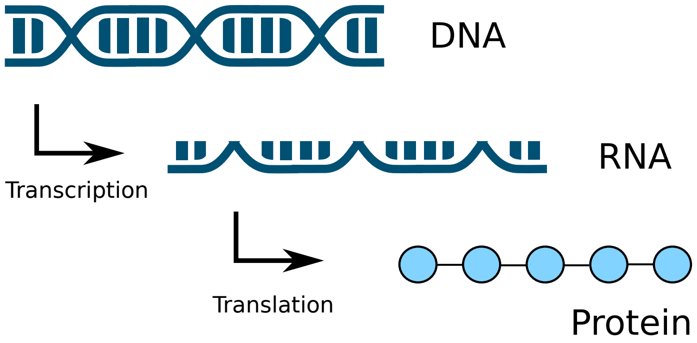
- Linear polymer of amino acids
  - Fold into complex 3D structures

- Perform many biological functions



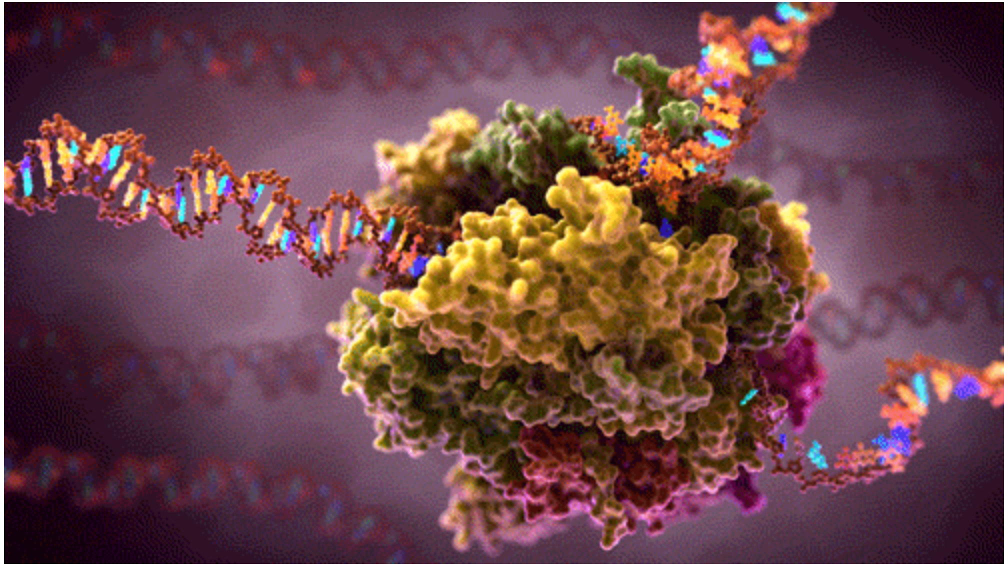
Central dogma of molecular biology

- Gene Expression
  - DNA → Protein
- Encoded by genes in genome
- 19,000 – 20,000 proteins in humans
  - 1.5% of human genome
- Composed of 20 amino acids



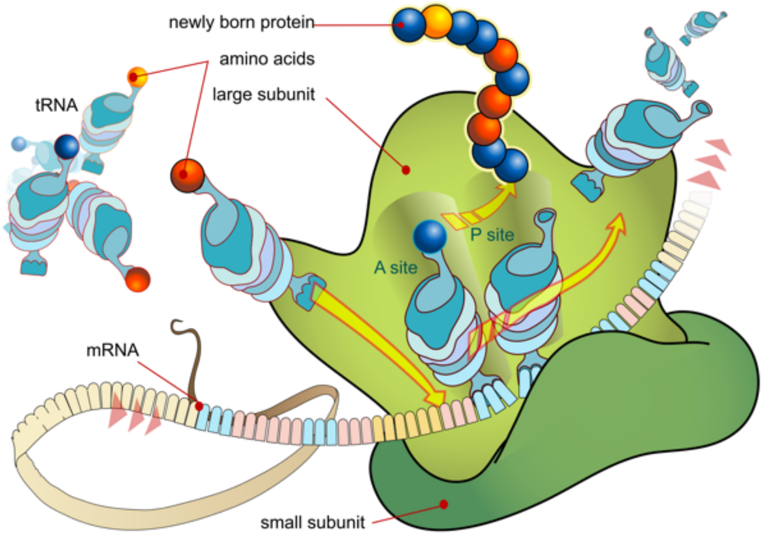
Transcription

- Transcribe DNA to *RNA* inside the nucleus



Translation

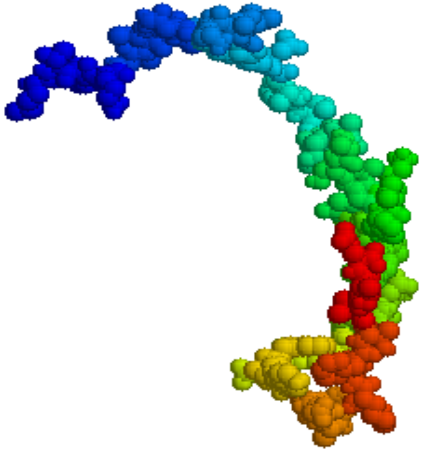
- Once in cytoplasm, mRNA is translated to polypeptide



https://en.wikipedia.org/wiki/Translation_(biology)#/media/File:Ribosome_mRNA_translation_en.svg

Folding

- Polypeptides fold spontaneously, or are assisted by chaperone proteins



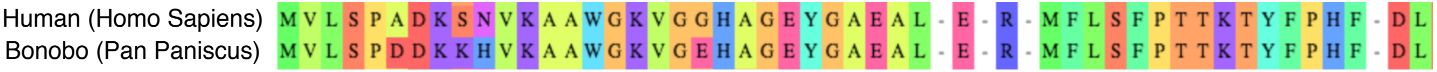
Proteins & evolution

- **Homologous** – similar due to shared ancestry
- **Ortholog** – similar proteins diverged through speciation

- Similarities between proteins are proxies for similarities between genes
  - Infer function of new protein because of its similarity to known protein
    - Extrapolation from small number of model organisms
  - Infer evolutionary relationships between species
    - X evolved from Y
    - X, Y have common ancestor

- Several of 100 most-cited scientific papers are sequence homology

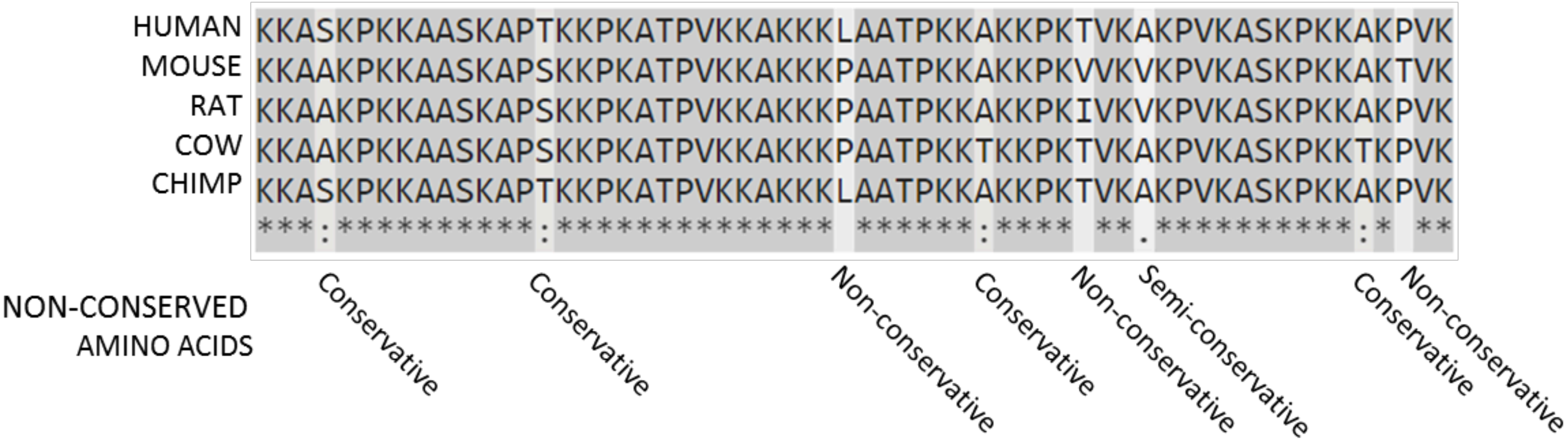
Sequence homology



```
Human (Homo Sapiens)  MVLSPADKSNVKAAWGKVGGHAGEYGAEAL-E-R-MFLSFPTTKTYFPHF-DL
Bonobo (Pan Paniscus) MVLSPDDKKHVKAAWGKVGEHAGEYGAEAL-E-R-MFLSFPTTKTYFPHF-DL
```

Alignment showing protein similarity between hemoglobin α-subunits from human and bonobo proteins

Histone H1 (residues 120-180)



```
HUMAN KKASKPKKAASKAPTKKPKATPVKKAKKKLAATPKKAKKPKTVKAKPVKASKPKKAKPVK
MOUSE KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKAKKPKVVKVKPVKASKPKKAKTVK
  RAT KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKAKKPKIVKVKPVKASKPKKAKPVK
  COW KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKTKKPKTVKAKPVKASKPKKTKPVK
CHIMP KKASKPKKAASKAPTKKPKATPVKKAKKKLAATPKKAKKPKTVKAKPVKASKPKKAKPVK
      ***:**********:************** ******:**** **.**********:* **
```

Identifying similar proteins

- Input → sequenced proteins
- Output → sets of homologous proteins

- All-against-all comparison
  - O(n^2) in number of sequences
  - Sequence comparison also O(n^2) in length of sequences (Smith-Waterman)

- OMA protein database contains proteins from 2000 genomes
  - Required more than 10 million CPU hours

Improvement needed!

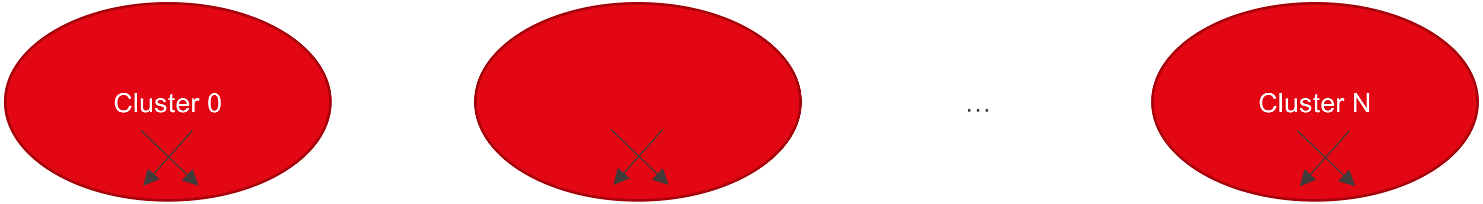
"Computing orthologs between all complete proteomes has recently gone from typically a matter of CPU weeks to hundreds of CPU years, and new, faster algorithms and methods are called for."

– Quest for Orthologs consortium, 2014

Incremental greedy protein clustering

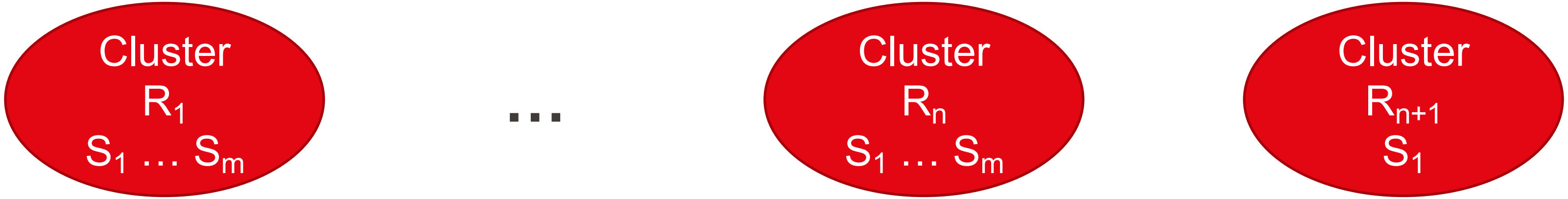
- **Speeding up all-against-all protein comparisons while maintaining sensitivity by considering subsequence-level homology**, *PeerJ*, 2014, Wittwer, Pilizota, Altenhoff, Dessimoz.

- Cluster similar proteins, then perform all-against-all comparison within each cluster
- Reduces computation time by ~75%
- Identify >99.6% of pairs found by all-vs-all



Cluster representative

- Input sequences compared against a cluster **representative**
  - Homologies are transitive
    - A, B homologous; B, C homologous ➔ A, C homologous
- No matches? Create a new cluster!



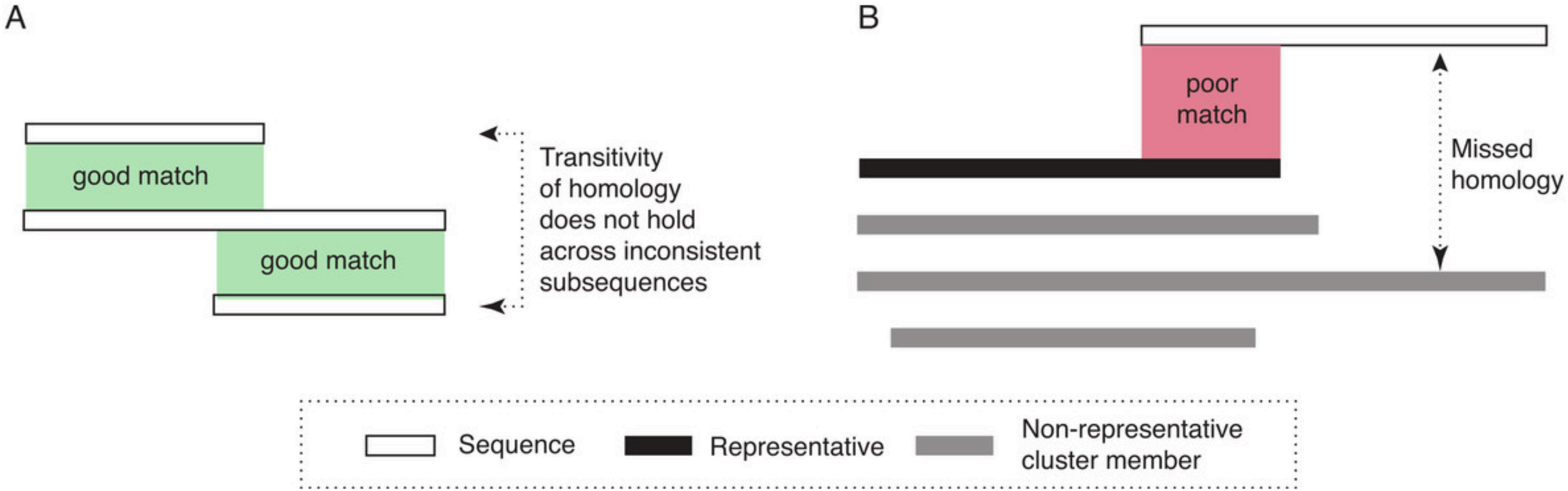
Cluster R_1 $S_1 \ldots S_m$

…

Cluster R_n $S_1 \ldots S_m$

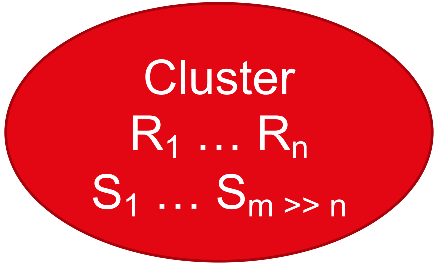
Cluster R_{n+1} S_1

Proteins not transitive



Clustering, v2

- Multiple representatives
- Ensure all sequences in a cluster are covered (± *T* residues)



Cluster

$R_1 \ldots R_n$

$S_1 \ldots S_{m >> n}$

Incremental greedy protein clustering

- Reduction in computation time of ~75%
  - Clusters are small, on average

- Accuracy is excellent
  - Maintain >99.6% of all pairs identified by all-against-all (naive)

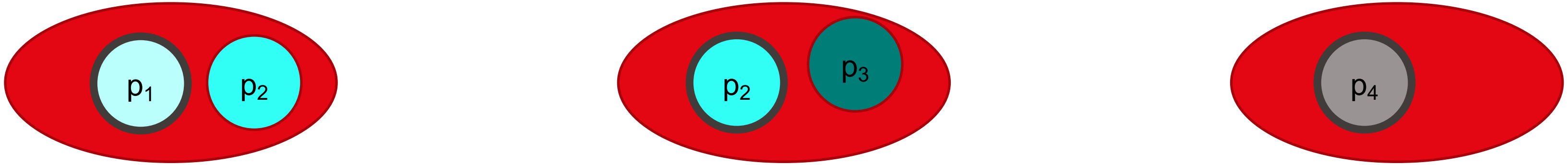
But,

- Algorithm is not easily parallelized

- Order in which clusters and representatives are chosen affects result
- Data (clusters) is shared – difficult to distribute

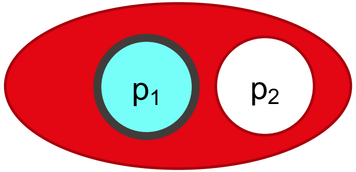
Our approach: precise clustering

- **Precise clustering (PC)**
  - All significant pairs are members of at least one cluster
  - Compare within cluster and find similarity
- A pair of proteins is **significant** if their similarity is above a threshold
  - $f(p_1,\ \ p_2) > T$
- PC is not a partition – a protein can be in more than one cluster
  - Relation f is not transitive, i.e. similarity is not equivalence



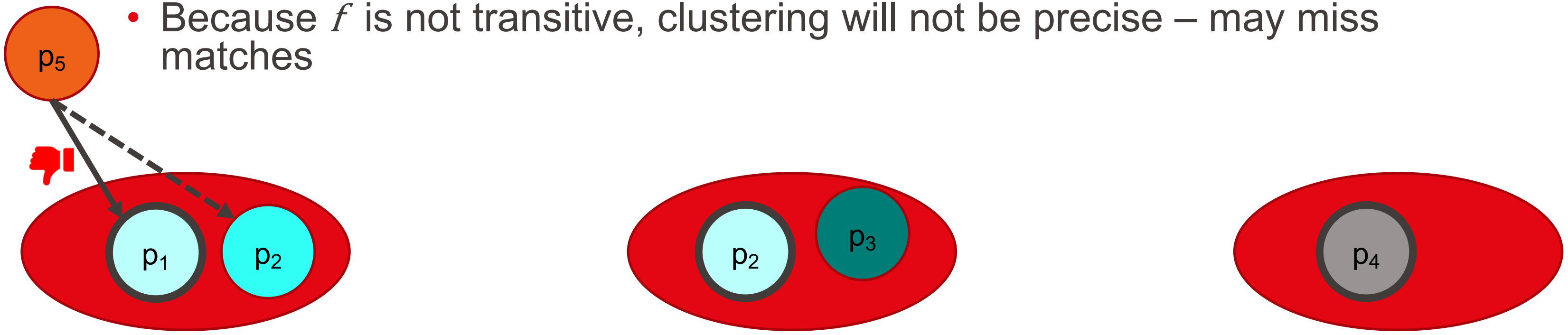
Cluster representative

- Each cluster has a unique **representative *R_C***
  - $\forall e \in C, \quad f(e, R_C) > T$
- Two elements in cluster may not be similar: $e_1, e_2 \in C \;\not\vdash\; f(e_1, e_2) > T$



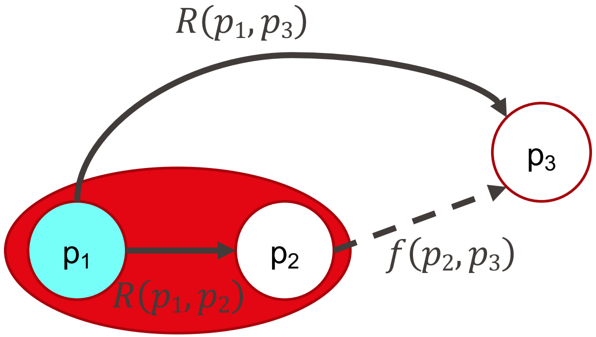
Approach 1

- New element e is compared against cluster representatives
  - If similar, e is added to cluster
- This does not work!
  - e, other than representative, will not be compared against subsequent elements
  - Because f is not transitive, clustering will not be precise – may miss matches



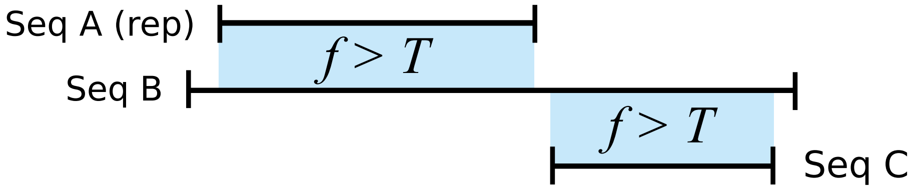
Transitive similarity

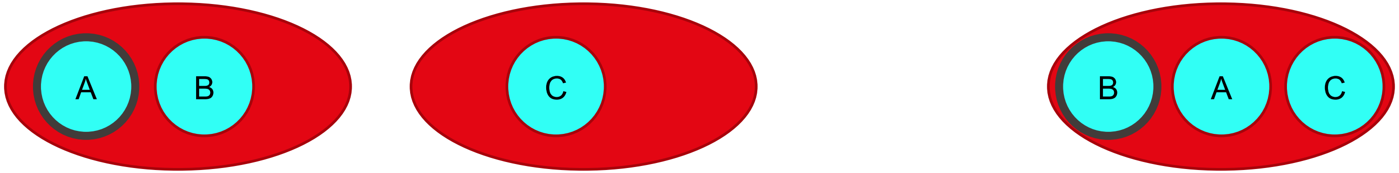
- Transitivity $\mathbf{R(e_1, e_2)}$ implies e_2 will be similar to e_3 if e_1 is similar to e_3
  - $\forall(i,j,k) \in S,\ R(i,j)\ \Rightarrow f(i,k) > T\ \wedge\ f(j,k) > T$



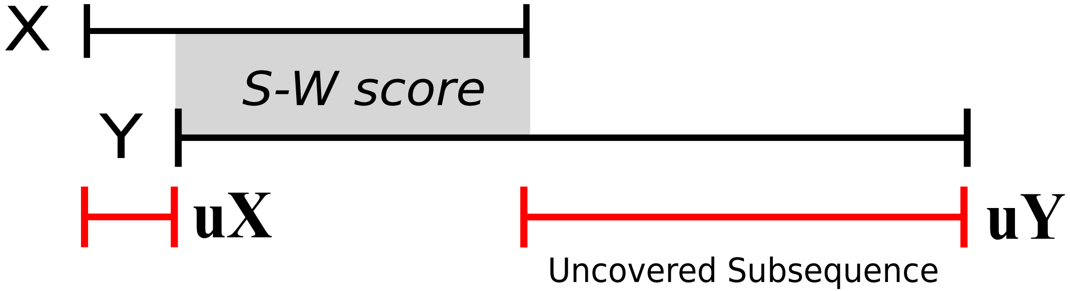
Protein similarity

- Similarity function f
  - Smith Waterman alignment >T (threshold parameter)
- Not transitive
- Comparison order matters





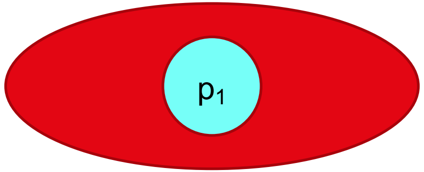
Protein transitivity



$$R(X, Y) \implies score > minT,\ uY < maxU$$
$$R(Y, X) \implies score > minT,\ uX < maxU$$

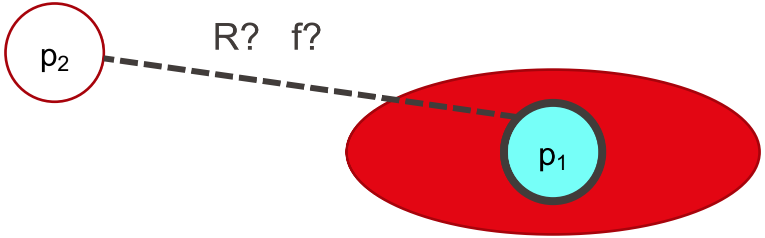
Incremental greedy precise clustering

- Construct clusters one element at a time
- First element becomes cluster representative



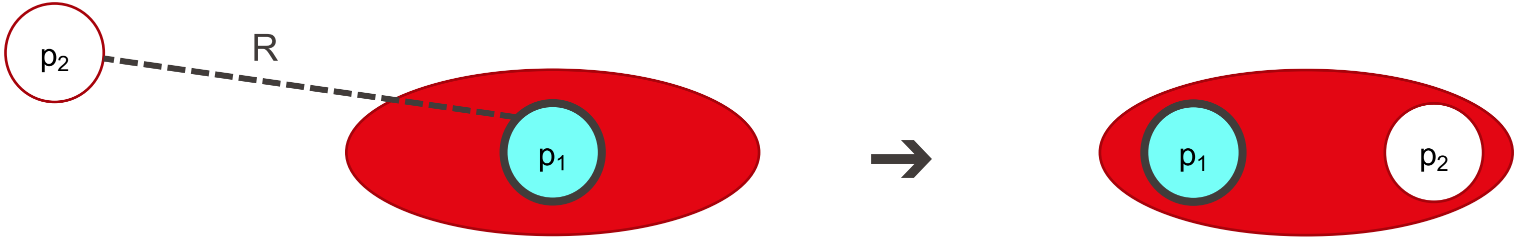
Incremental greedy precise clustering

- Compare subsequent elements against cluster representative



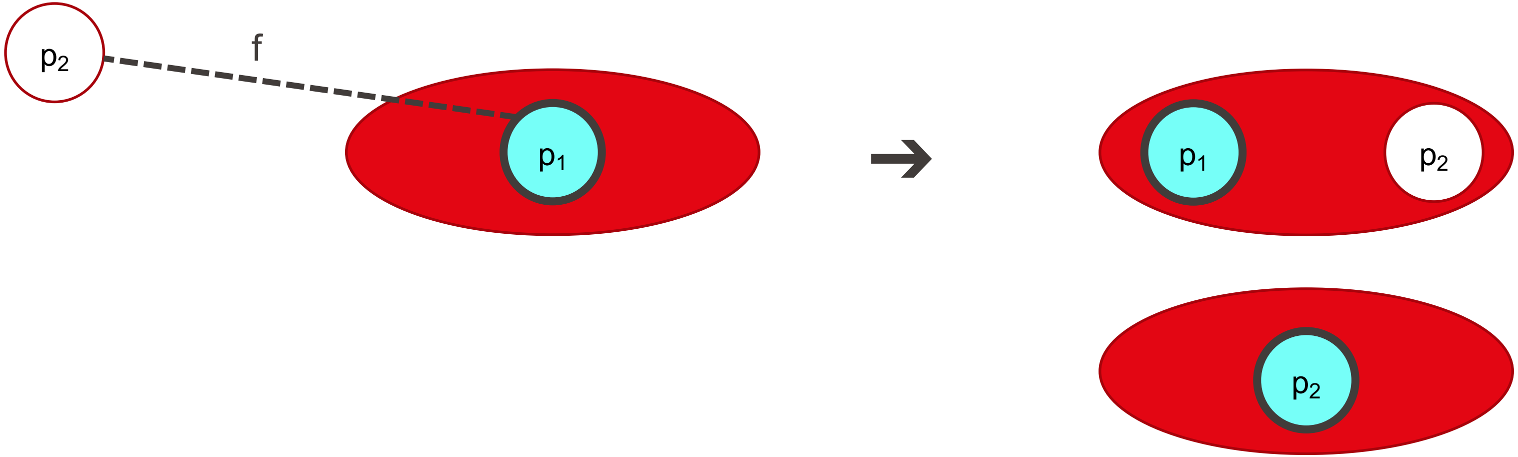
Incremental greedy precise clustering

- If **transitively similar**, add to cluster



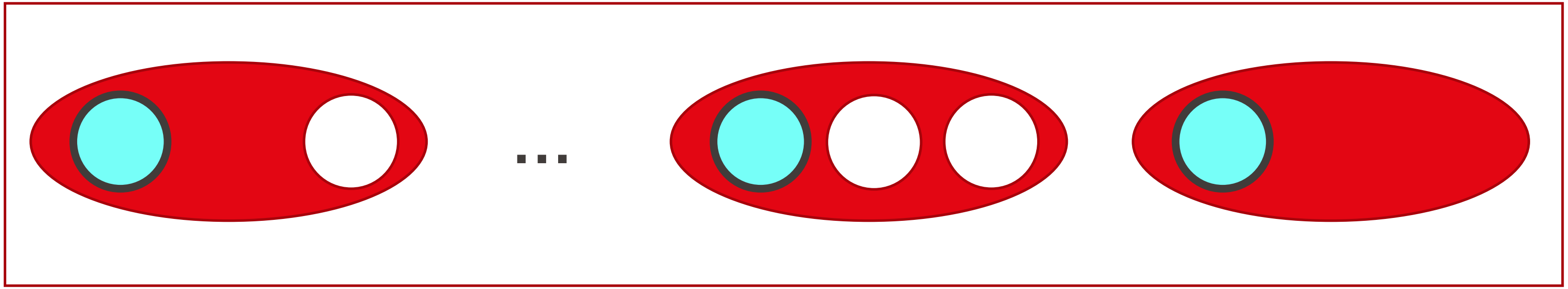
Incremental greedy precise clustering

- If **only similar**, add to cluster and create a new cluster



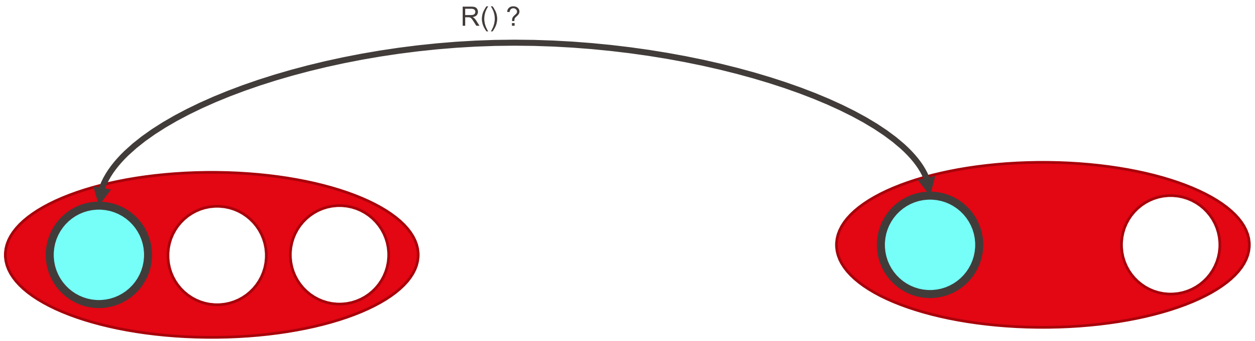
Incremental greedy precise clustering

- Continue until all elements clustered

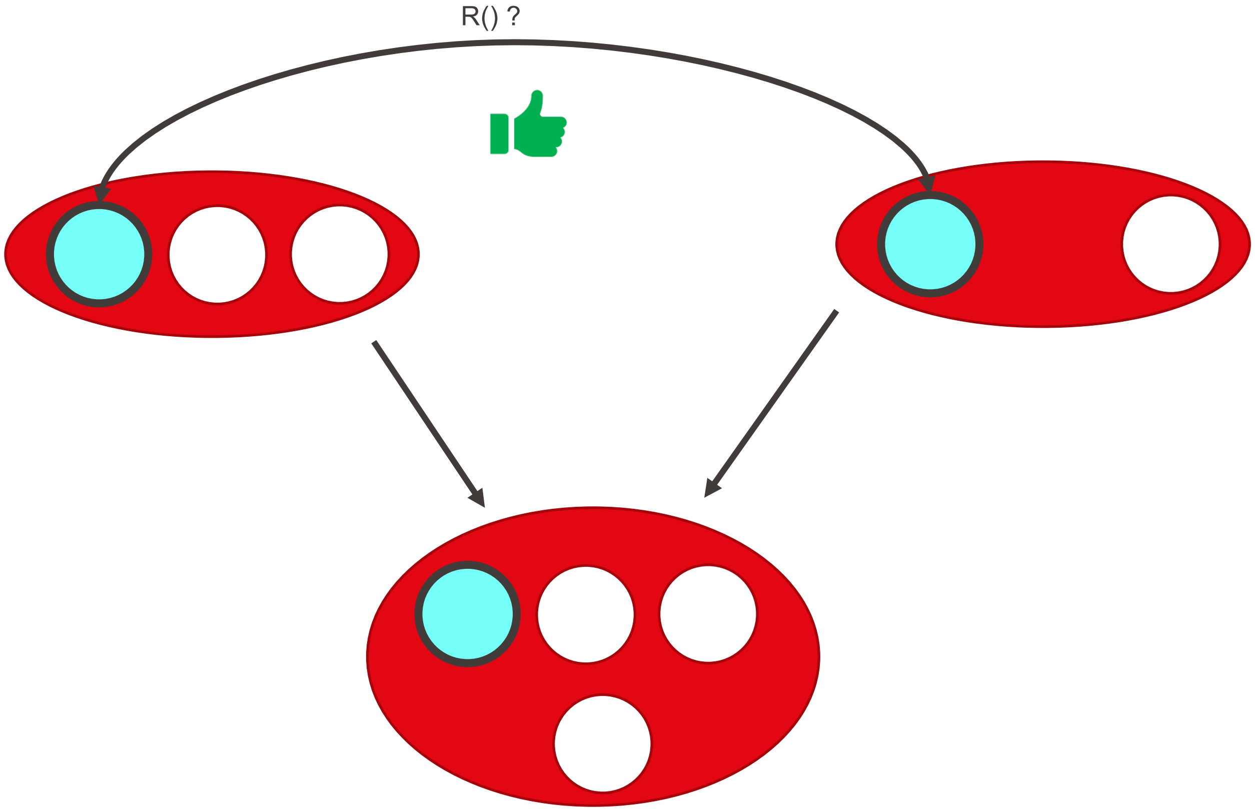


Parallelism

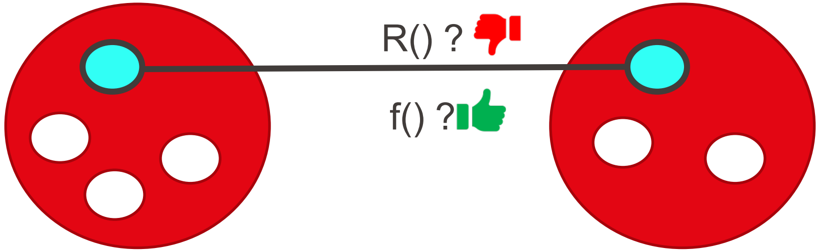
- Unlike original Wittwer algorithm, order does not matter for precise clustering
- Clusters can be constructed independently and **merged**

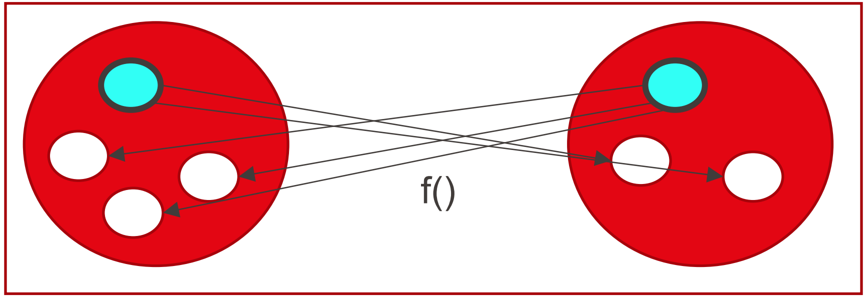


Merging clusters

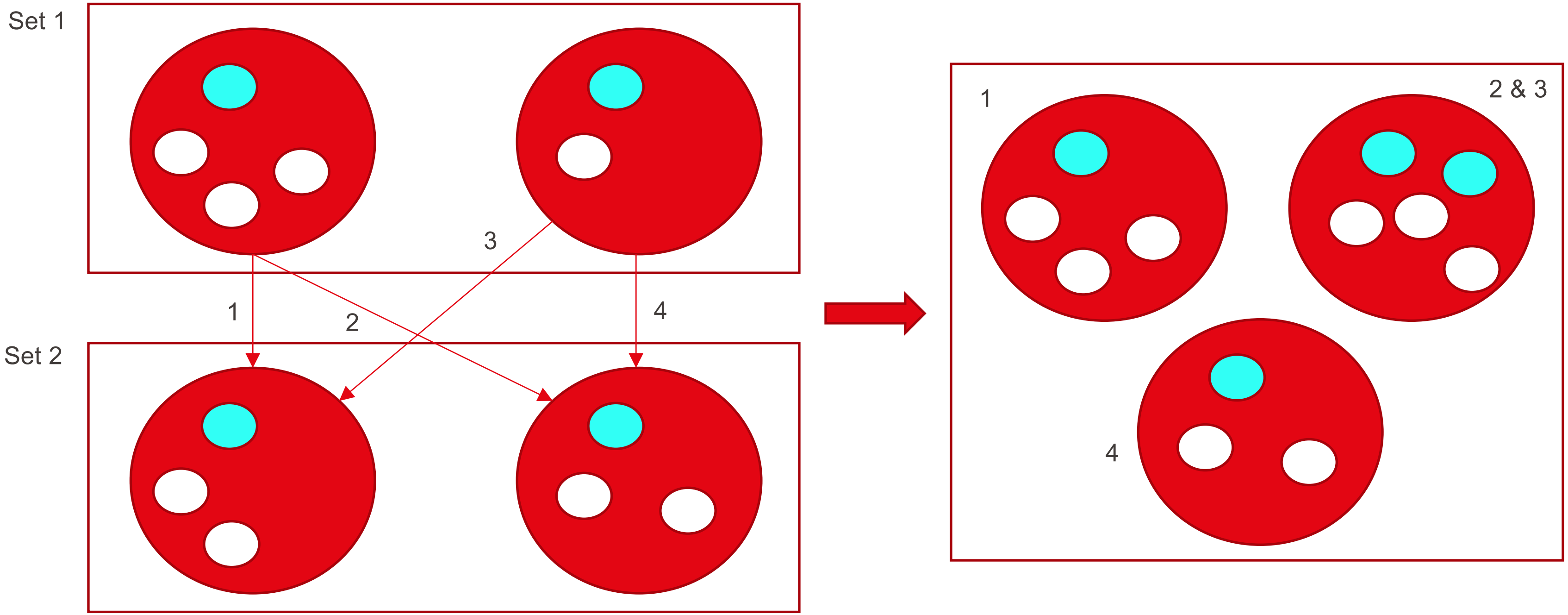


Merging clusters

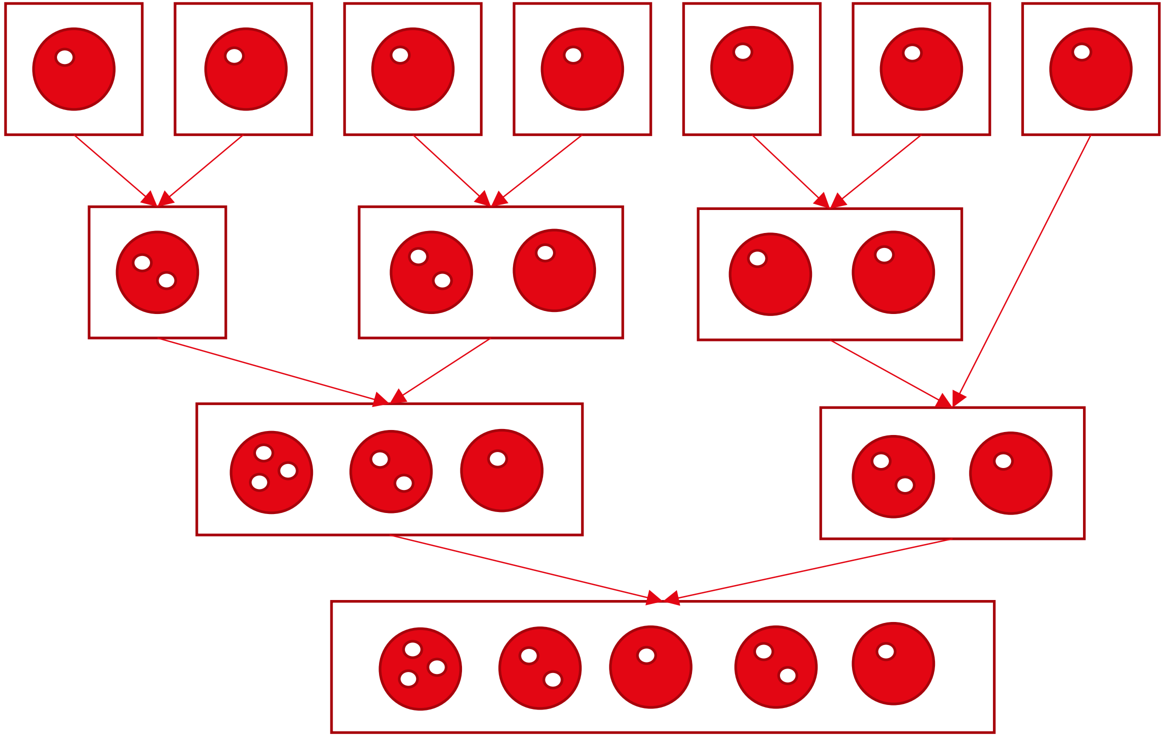




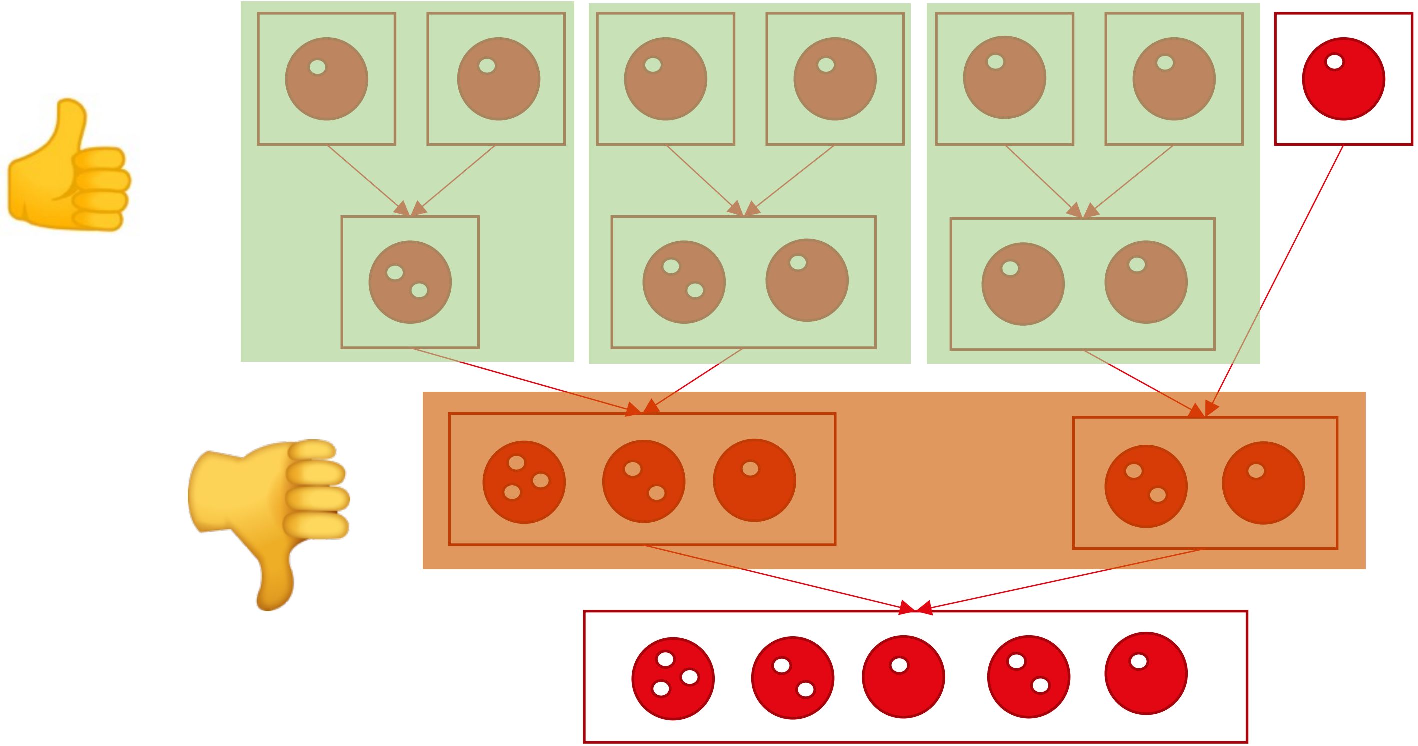
Merging sets of clusters



Cluster merge

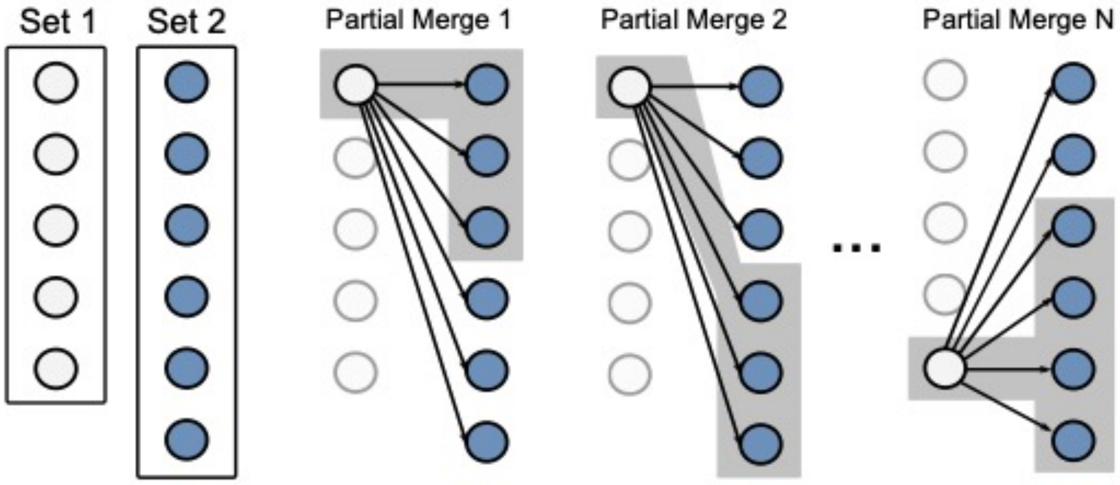


Parallelization 1

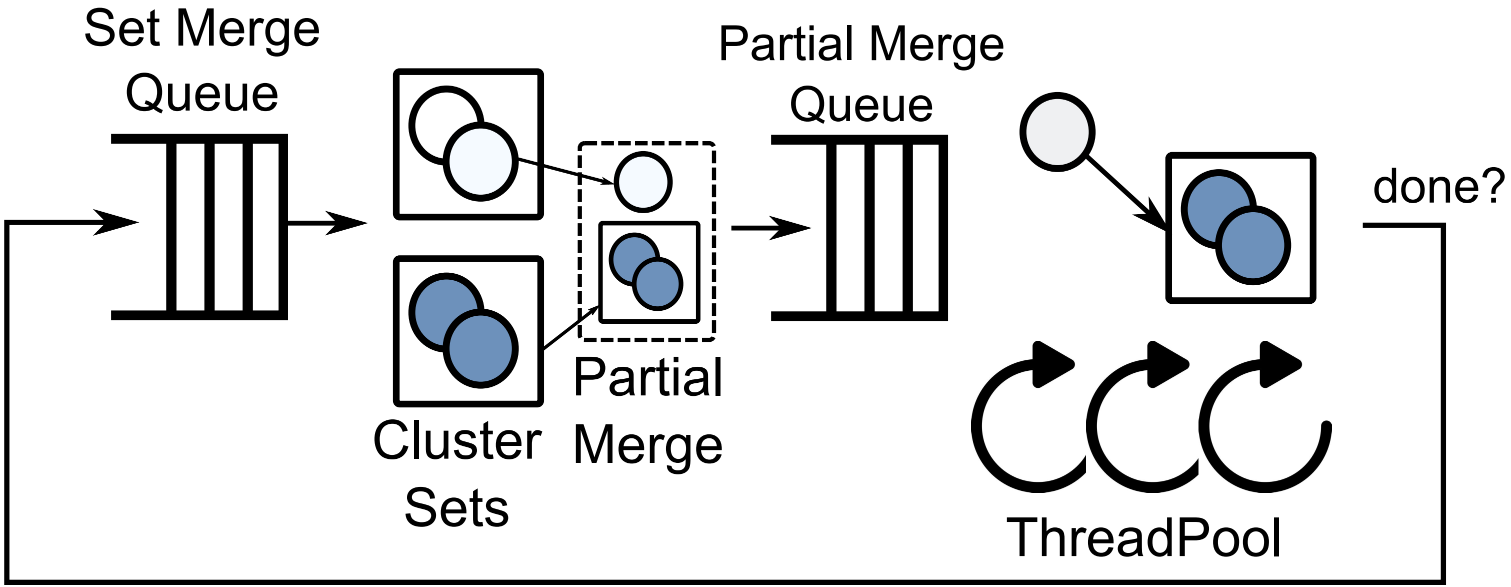


Parallelization 2

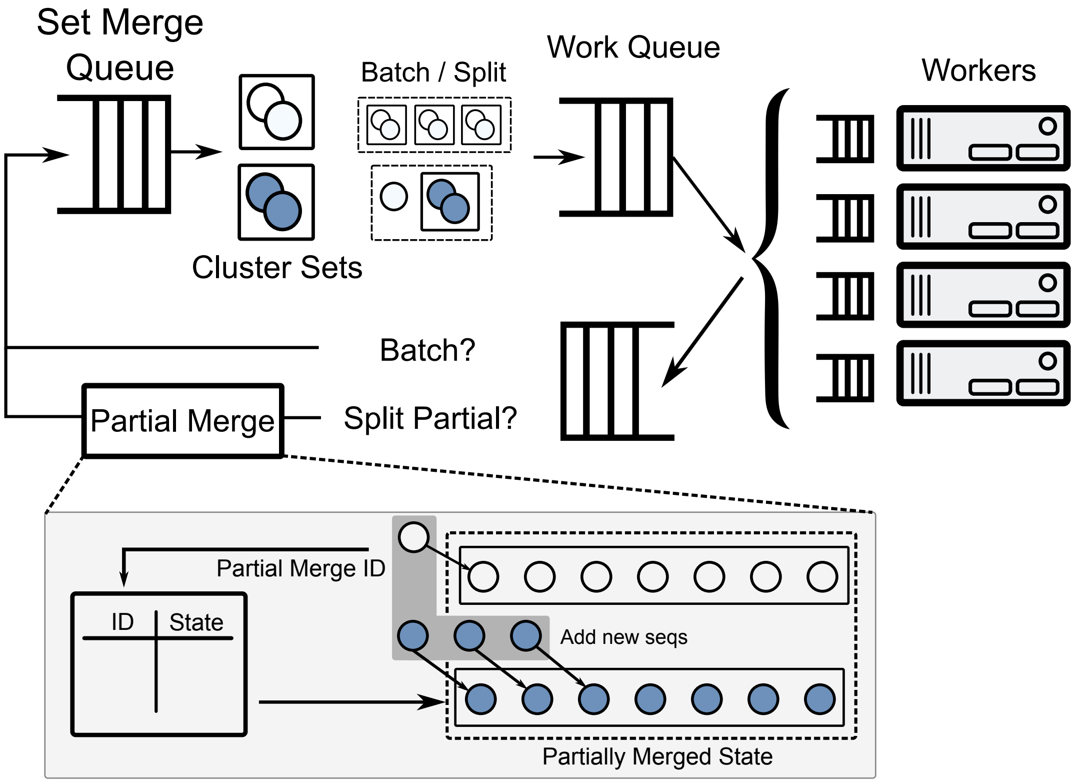
- Parallelize merge of two large sets
- Each computation is a *partial merge*



Shared-memory (Shared-CM)

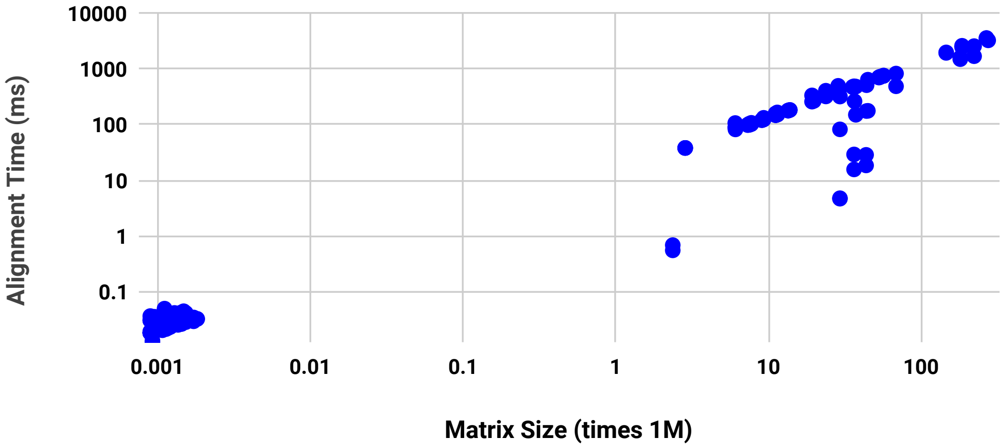


Distributed (Dist-CM)



Dist-CM optimization

- Every remote worker has copy of all sequences
  - Sequences named by index (4-byte)
- Workers cache copies of sets and only transfer diffs
- Careful queue management
- Aggressive load balancing



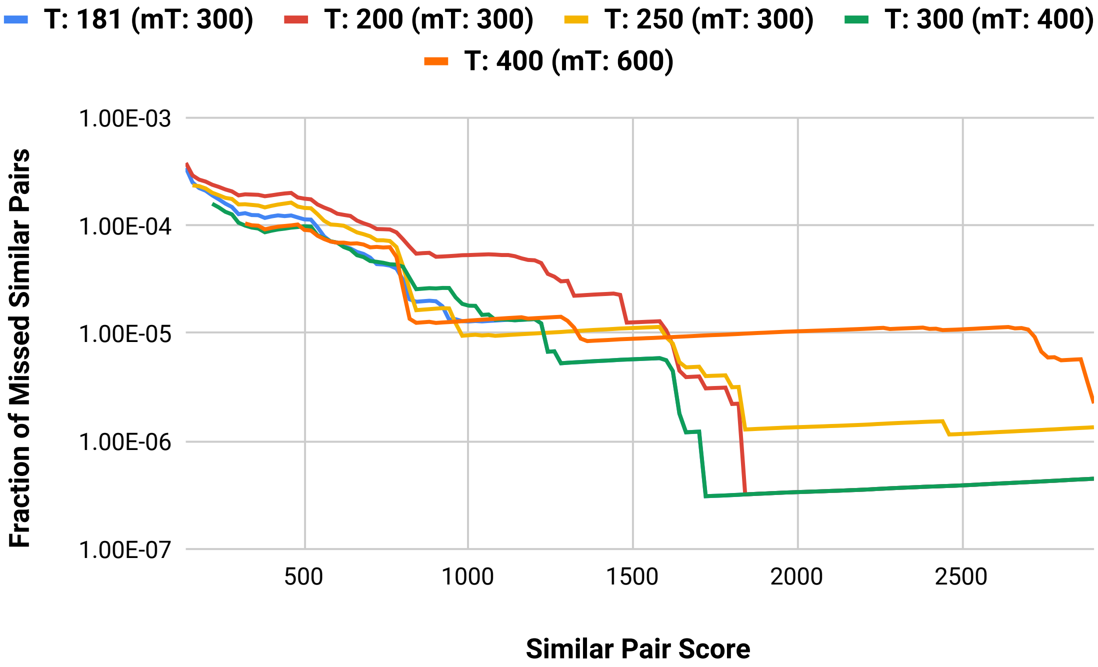
Evaluation

- Recall
  - Number of significant pairs, relative to all-against-all
- Scalability / performance

Recall

- Dataset
  - 13 bacterial genomes, ~59,000 sequences
- Similarity
  - S-W threshold of 181 with PAM250 substitution matrix (Wittwer)
- Transitivity
  - mT = 250, mU = 15
- Increment greedy clustering (1 / 3 representatives)
  - 99.6% / 99.9% recall (compared all-vs-all)
- Precise cluster merge (Shared-CM/Dist-CM)
  - 99.8 ± 0.01% recall
  - Missed 10^{-6} significant pairs, mainly low scoring ones (avg. 191, median 235)

Sensitivity analysis

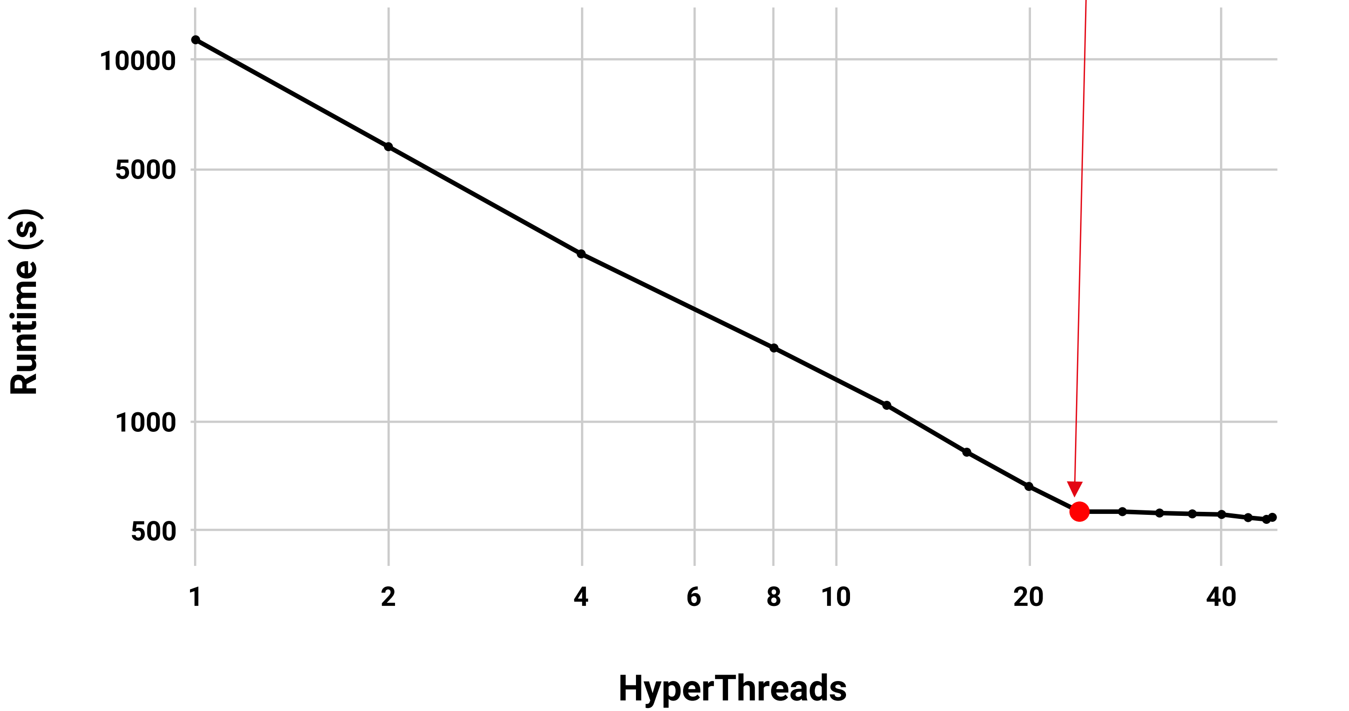


Shared-memory speedup

- Smaller data set (28,600 sequences)
- Incremental greedy clustering [Wittwer] (1 / 3 representatives)
  - 4x / 2x faster than all-vs-all
- Original clustering (1 representative)
  - 89,486 seconds = 24.9 hours
- Shared-CM (48 thread)
  - 1,486 seconds = 0.41 hours

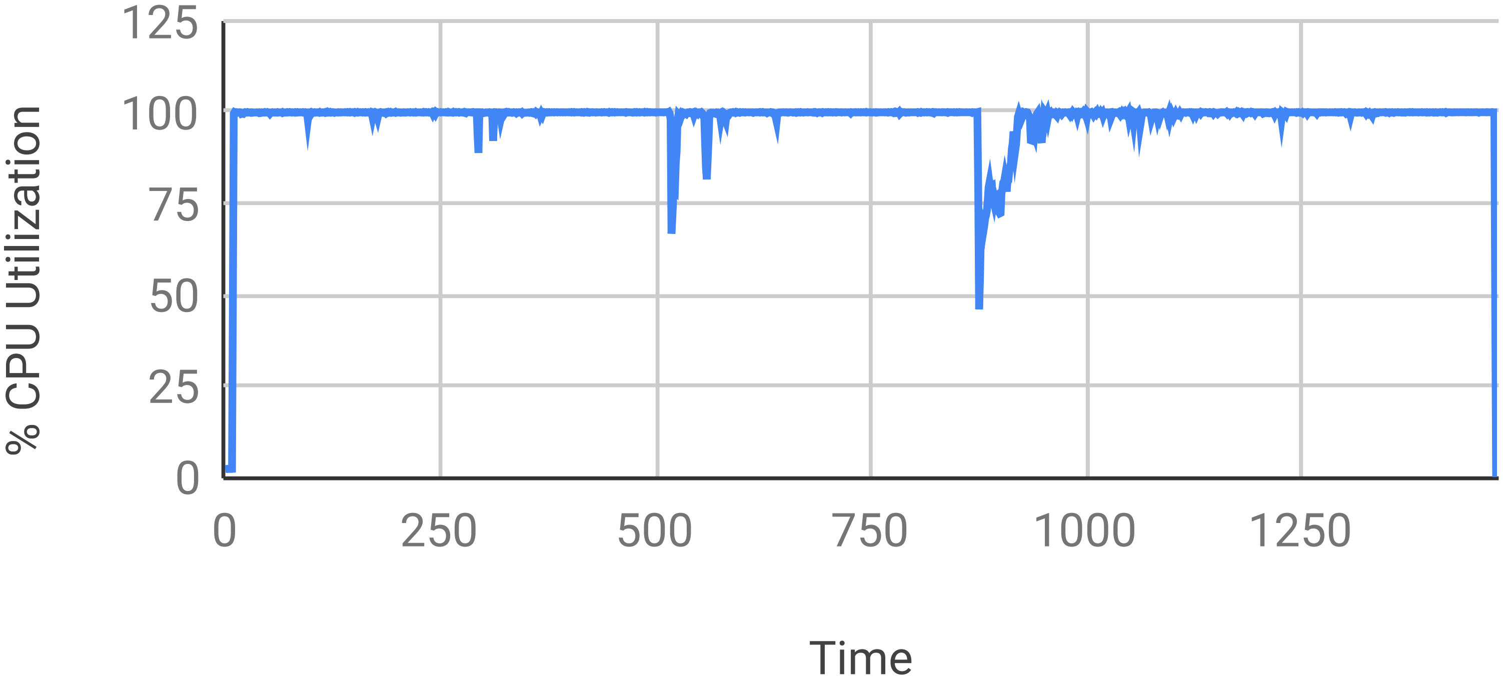
60.2x speedup

Shared-memory scalability



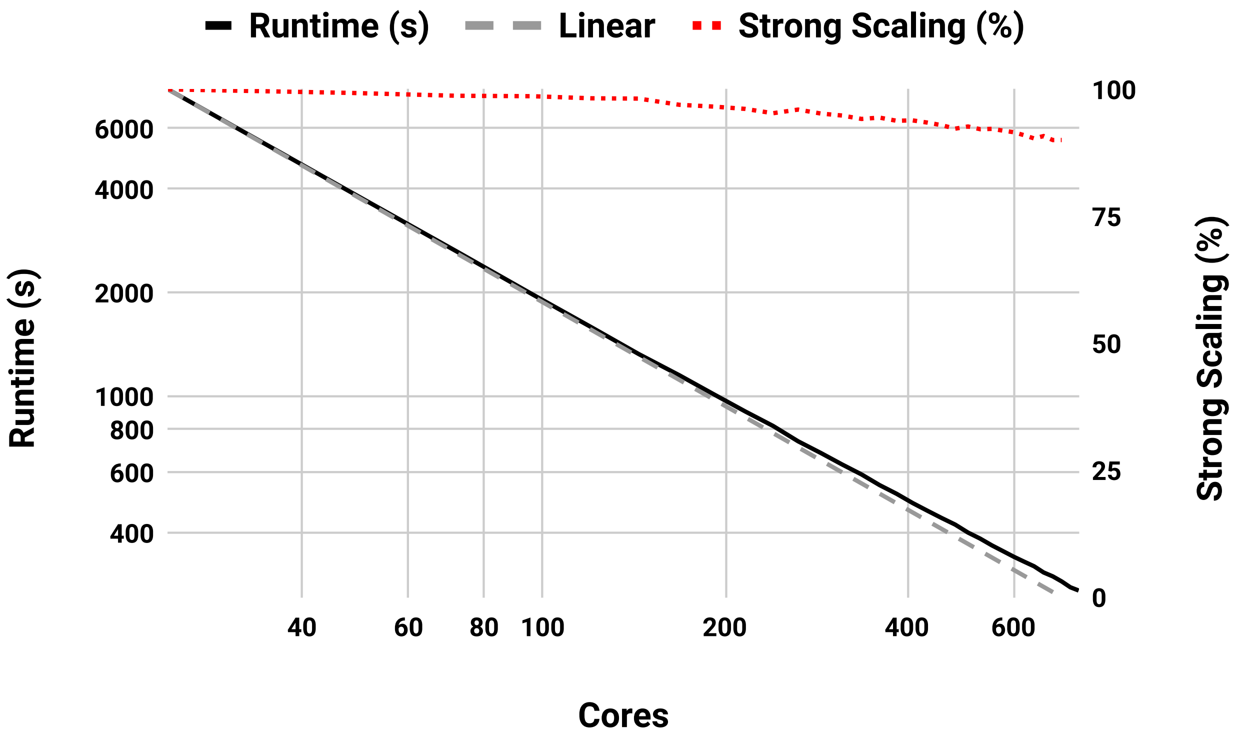
24 core Xeon
48 threads
Hyperthreading of no benefit

Shared-memory scalability



Distributed – strong scaling

Dataset fixed
Vary number of nodes

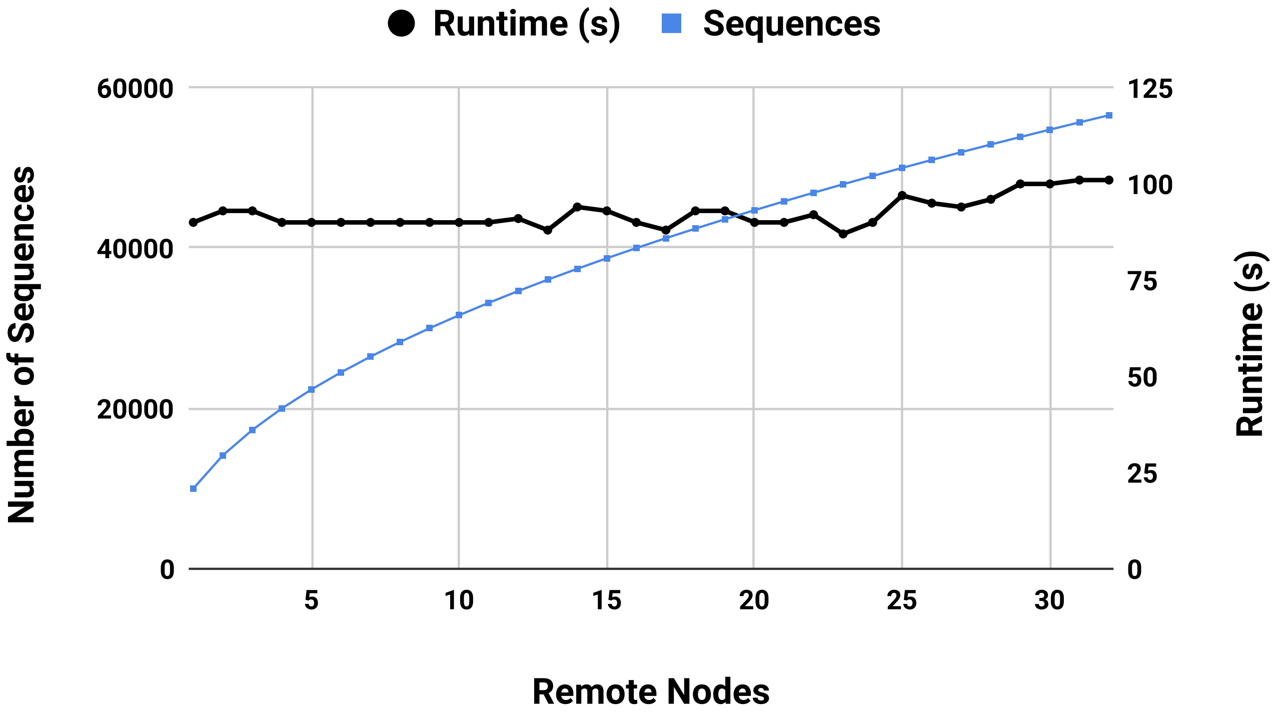


Dist-CM
604x on 32 nodes (768 cores)
79% efficiency

1,400x over Wittwer

Distributed – weak scaling

Dataset grows $\sim\sqrt{n}$



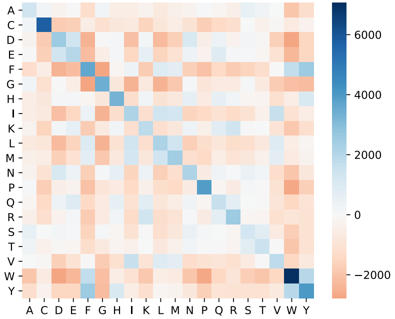
Dataset composition

- Dataset of 13 bacterial genomes
  - 59,013 sequences
- Dataset of 33 closely related Streptococcus bacteria genomes
  - 69,648 sequences
- Closely related ⇒ fewer clusters
  - Closer to $O(n \lg n)$ performance
- Shared-CM (48 threads)
  - Streptococcus 283 sec. and 10,500 clusters
  - (vs 1,486 sec. and 33,562 clusters)

Improvements / Future work

- Larger, more diverse datasets (w/ friends from UNIL)
- Seeding clusters with known significant pairs

- Hardware acceleration of Smith-Waterman comparison
  - Proteins are long (300 - 30,000 amino acids)
  - Alphabet is richer (20 amino acids)
  - More complex scoring function.



Conclusion

- Think beyond DNA!
  - Proteins are richer and more challenging than DNA
- Hardware acceleration is premature if your application does not have near-linear speedup on a cluster
  - Bioinformatics need parallel algorithms and implementations
- Keeping cores busy is key to efficient parallelism
  - Communications efficiency
  - Work distribution and load balancing

EPFL

Merci

James Larus

École
polytechnique
fédérale
de Lausanne