A Generic Framework for Efficient and Effective Subsequence Retrieval

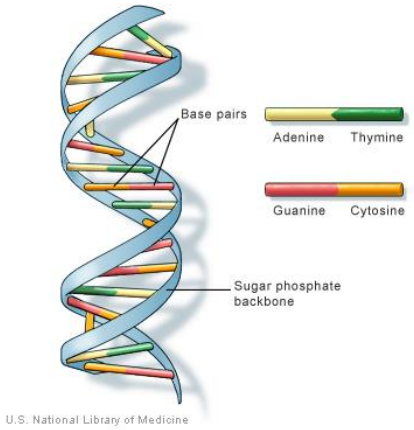
Haohan Zhu[1], George Kollios[1], Vassilis Athitsos[2]

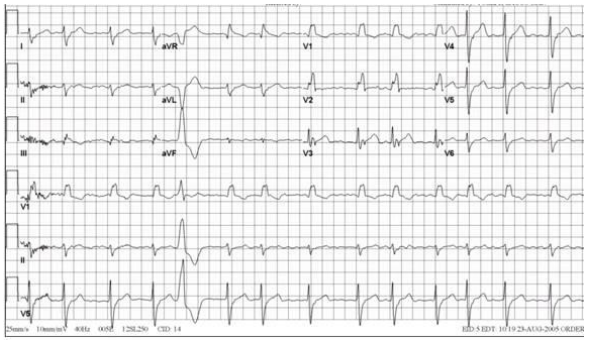
1. Boston University
2. University of Texas at Arlington

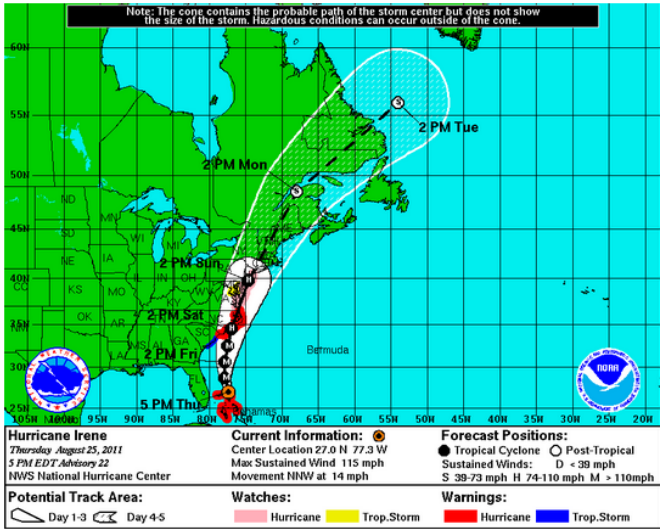
BOSTON UNIVERSITY

Generic Framework for Sequences

- Time-series and String Databases
  - Songs, Trajectories, Video, etc.
  - DNA, Proteins, Text, etc.



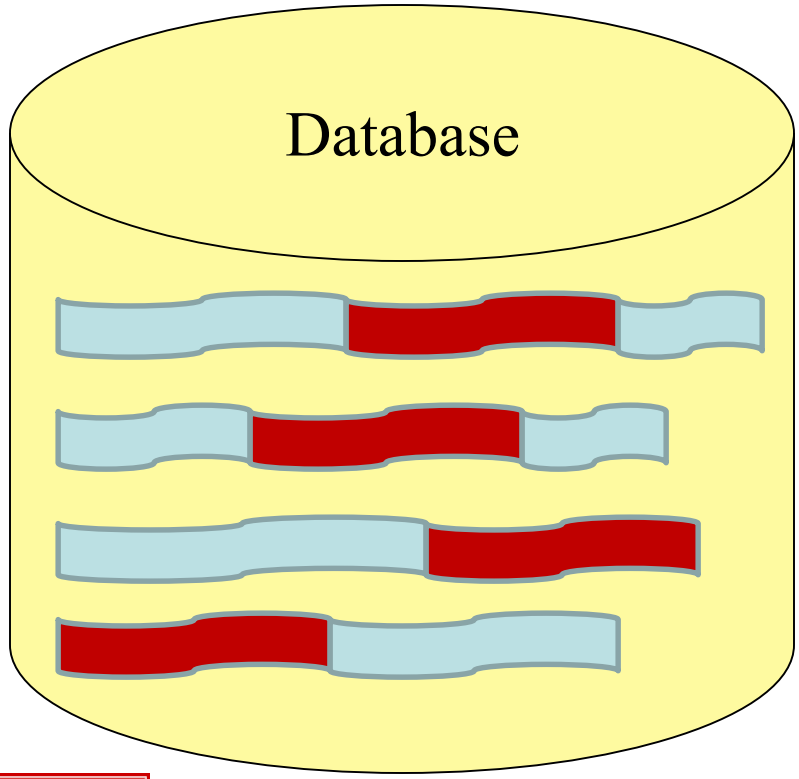




BOSTON UNIVERSITY

Subsequence Retrieval

- Subsequence for both Query and Database



Similar Subsequences

- Sequences
  - $Q := (q_1, q_2, q_3, \ldots, q_n)$ $SQ := (q_i, q_{i+1}, q_{i+2}, \ldots, q_j)$
- Similar Subsequences (ε, λ)
  - δ(SX, SQ): Distance between subsequences SX and SQ
  - |SX|: Length of subsequence SX
  - **$\delta(SX, SQ) \leqq \varepsilon$**
  - **$|SX| \geqq \lambda, |SQ| \geqq \lambda$**
  - $| |SQ| - |SX| | \leqq \lambda_0$
  - λ_0 is to avoid distortion

Query Types

- I: Range Query (Too many results)
  - $|SX| \geqq \lambda$, $|SQ| \geqq \lambda$, $\delta(SX, SQ) \leqq \varepsilon$ and $| |SQ| - |SX| | \leqq \lambda_0$
- II: Longest Similar Subsequences Query
  - Maximize: $|SQ|$
  - Subjects to: $|SX| \geqq \lambda$, $\delta(SX, SQ) \leqq \varepsilon$ and $| |SQ| - |SX| | \leqq \lambda_0$
- III: Nearest Neighbor Query
  - Minimize: $\varepsilon = \delta(SX, SQ)$
  - Subjects to: $|SX| \geqq \lambda$, $|SQ| \geqq \lambda$ and $| |SQ| - |SX| | \leqq \lambda_0$

Computing Similar Subsequences

- However in many cases computing distance of sequences is expensive
  - Dynamic programming
  - Edit Distance, DTW, Discrete Frechet distance, ERP, etc.

- $O(m^2n^2)$ pairs of subsequences
  - m^2 subsequences SX from Sequence Database X
  - n^2 subsequences SQ from Query Sequence Q

- Reduce number of pairs need to be checked to O(mn).

Framework

- 1. Segment Database and Query Sequences
- 2. Range Query on Segments
  - lengths of segments are fixed or in a certain range
  - Only O(mn) pairs of segments
- 3. Generate Candidate Pairs of Similar Subsequences from Similar Segments
- 4. Validate Candidates (According to Query Type)

Consistency Property

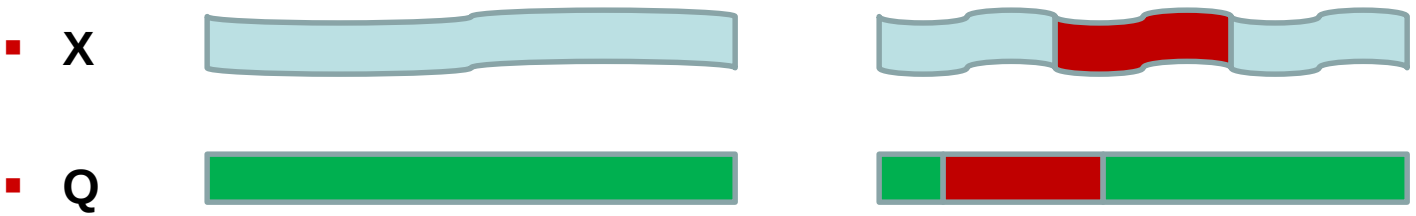
- Guarantee every pair of similar subsequences must have a pair of similar segments.

- Definition: Let X and Q be two sequences, then for **every** subsequence SX of X, there **exists** a subsequence SQ of Q, such that $\delta(SQ, SX) \leqq \delta(Q,X)$. Then δ is a consistent distance.

- Consistency property is for sequence measurements.

Consistency Property

- Definition: For **every** subsequence SX of X, there **exists** a subsequence SQ of Q, such that δ(SQ, SX) ≦ δ(Q,X).

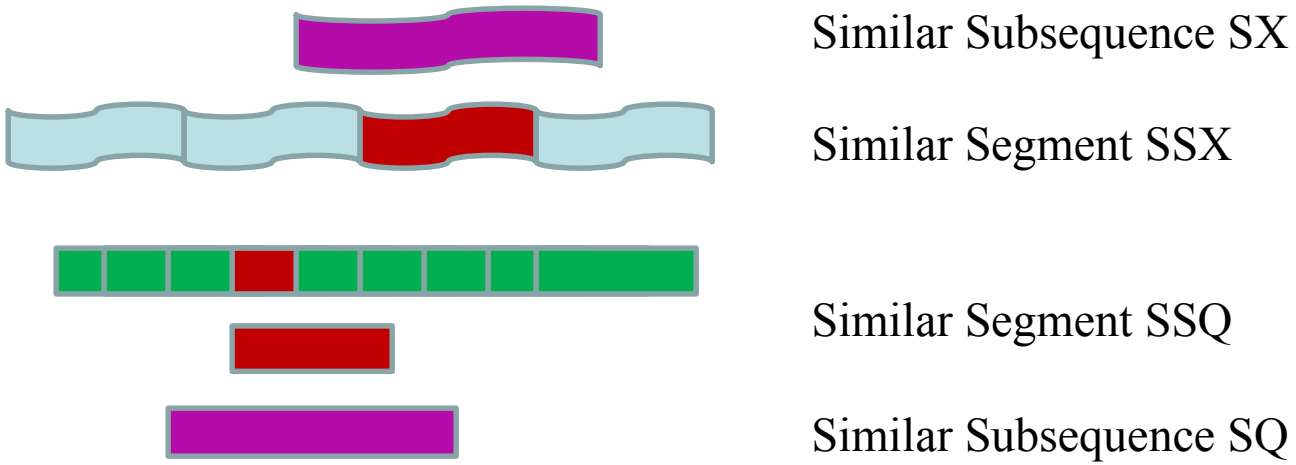


- DTW, Discrete Frechet Distance, ERP, Levenshtein distance, Euclidean Distance, Hamming Distance are all “consistent”.

Segmentation

- Guarantee every similar subsequence must include a whole segment.

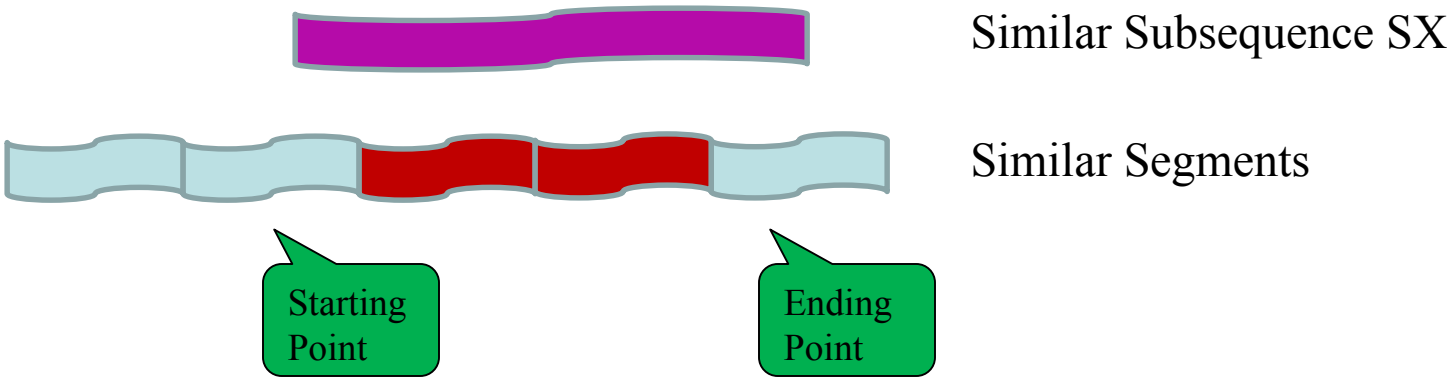
- Segmentation: Let sequence X be partitioned into windows of fixed length $l \leqq \lambda/2$. Let sequence Q be partitioned into windows of length in $[l-\lambda_0, l+\lambda_0]$



Generate Candidates

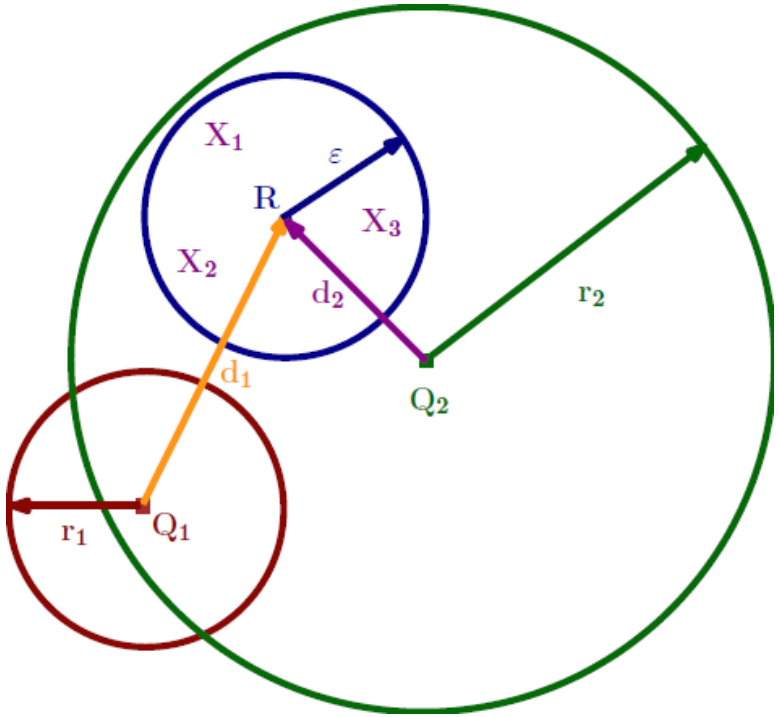
- Only if there exists similar segments, there may exist similar subsequences.

- If a similar subsequence has length larger than k*(λ/2), it has at least k-1 consecutive similar segments



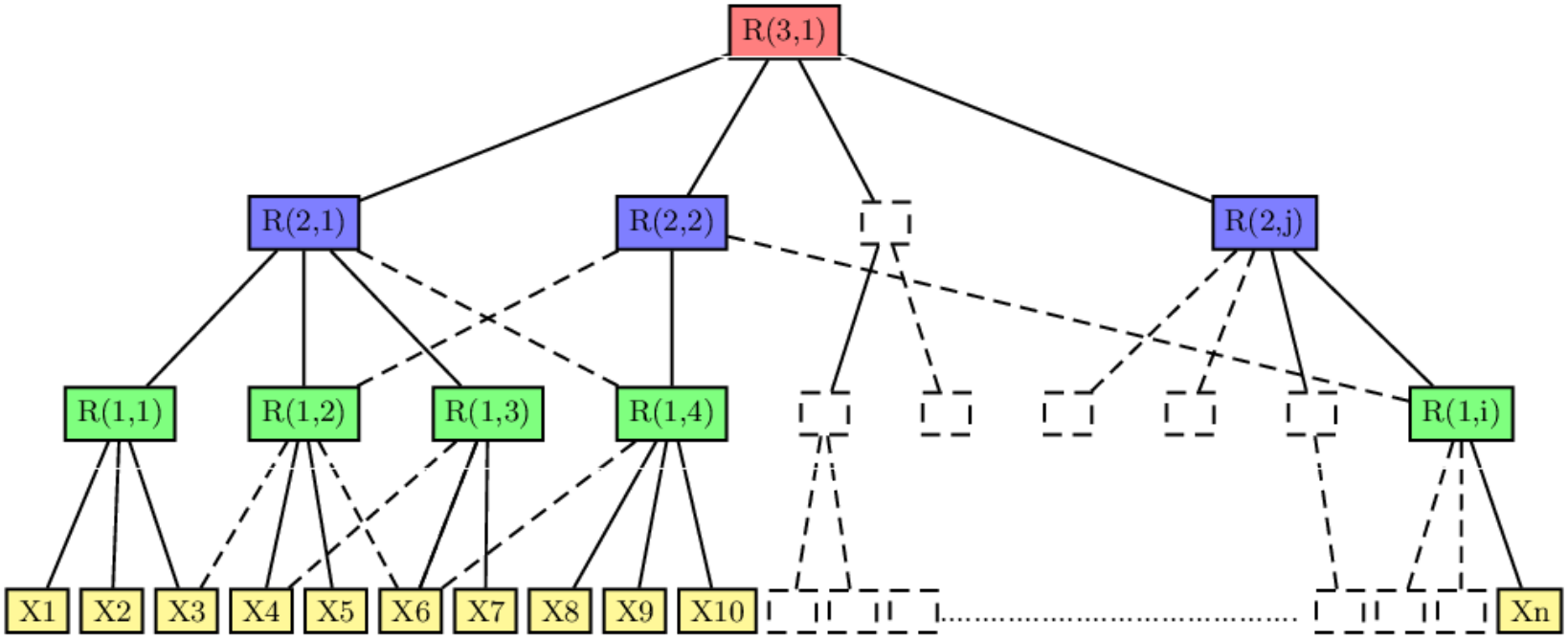
Range Query on Segments

- Referenced-based Index
- Metric Distance
  - Since $\delta(Q_2, R) + \varepsilon < r_2$, we can claim that all distances between Q_2 and every X_i are smaller than r_2.
  - Also because $\delta(Q_1, R) - \varepsilon > r_1$, we can claim that all distances between Q_1 and every X_i are larger than r_1.



Reference Net

- Each node R(i, j) is a reference. The range of reference R(i, j) is 2^i



Query Method

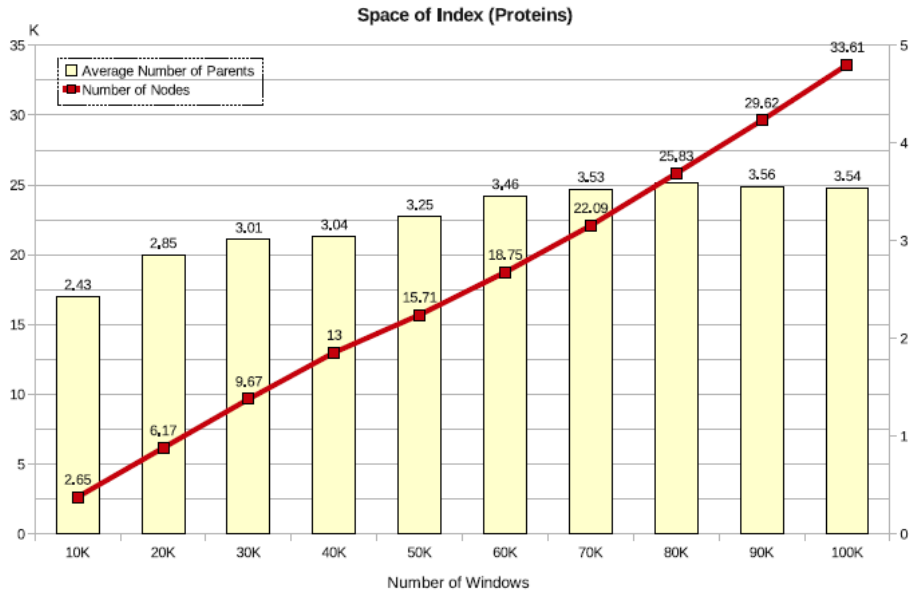
- I: Range Query
  - Check all candidates from pairs of similar segments
- II: Longest Similar Subsequences Query
  - Find the longest consecutive sequence of similar segments
  - Check from the longest candidates
- III: Nearest Neighbor Query
  - Binary search minimal ε when there exists some similar segments
  - Check all candidates

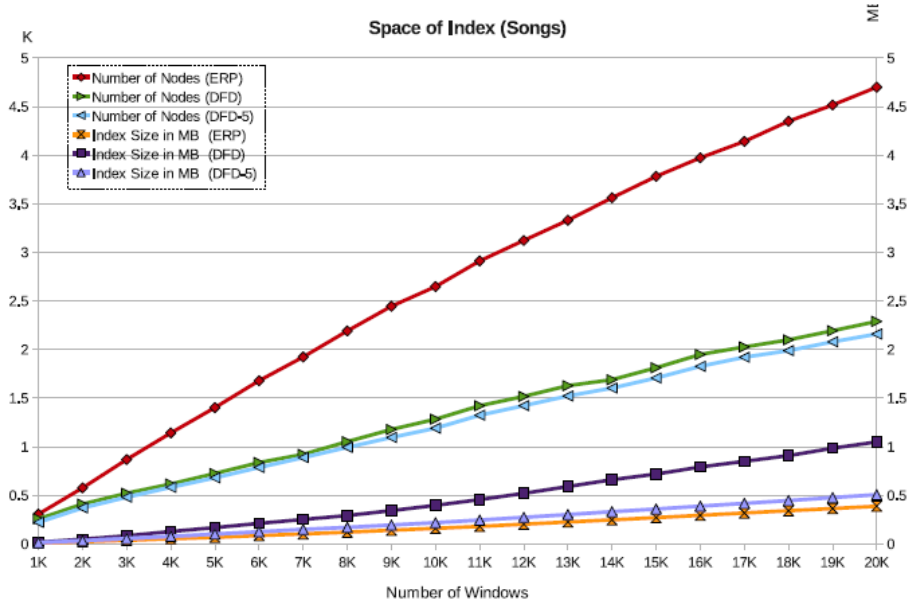
Experiments

- Datasets:
  - Protein
    - UniProt (100K total segments)
  - Songs
    - Songs Database (20K total segments)
  - Trajectories
    - Camera from Parking Lot (100K total segments)
- Distance Functions
  - Edit Distance (for Proteins)
  - Discrete Frechet Distance (for Songs and Trajectories)
  - ERP Distance (for Songs and Trajectories)

Experiments

- Space of Reference Net: O(n)

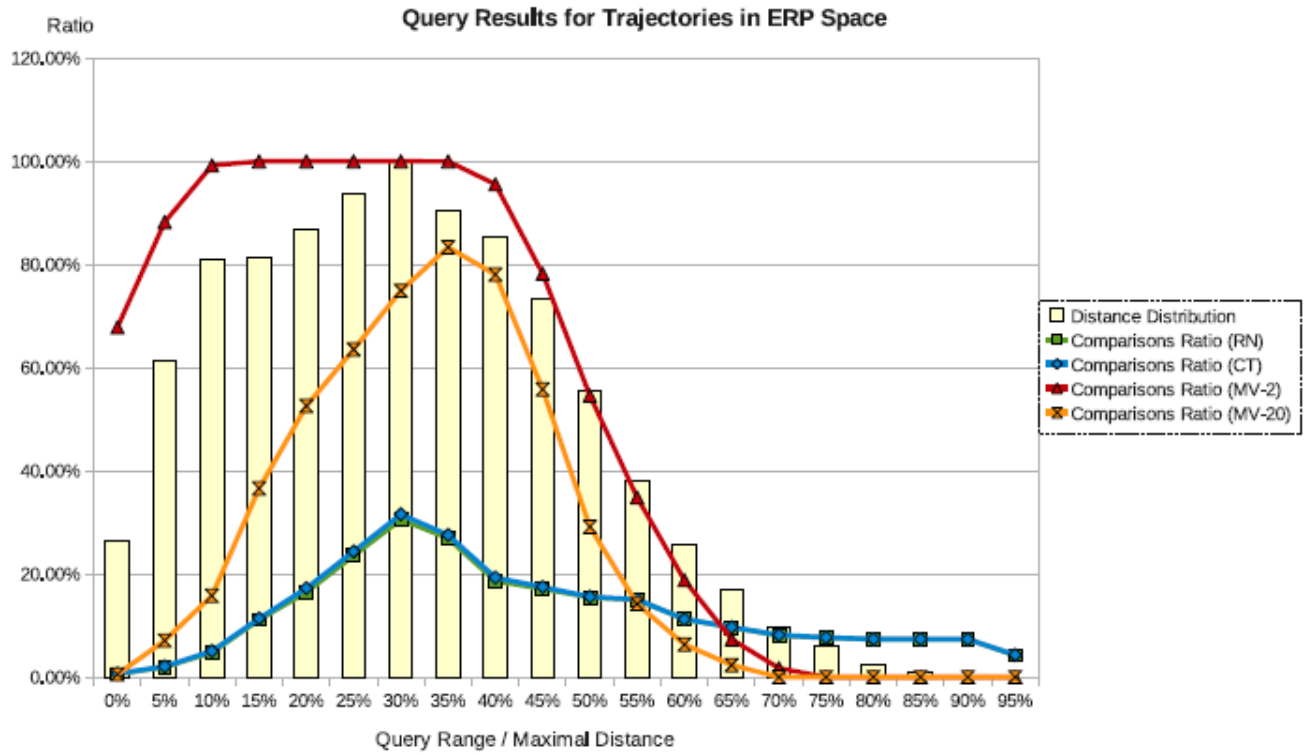




BOSTON UNIVERSITY

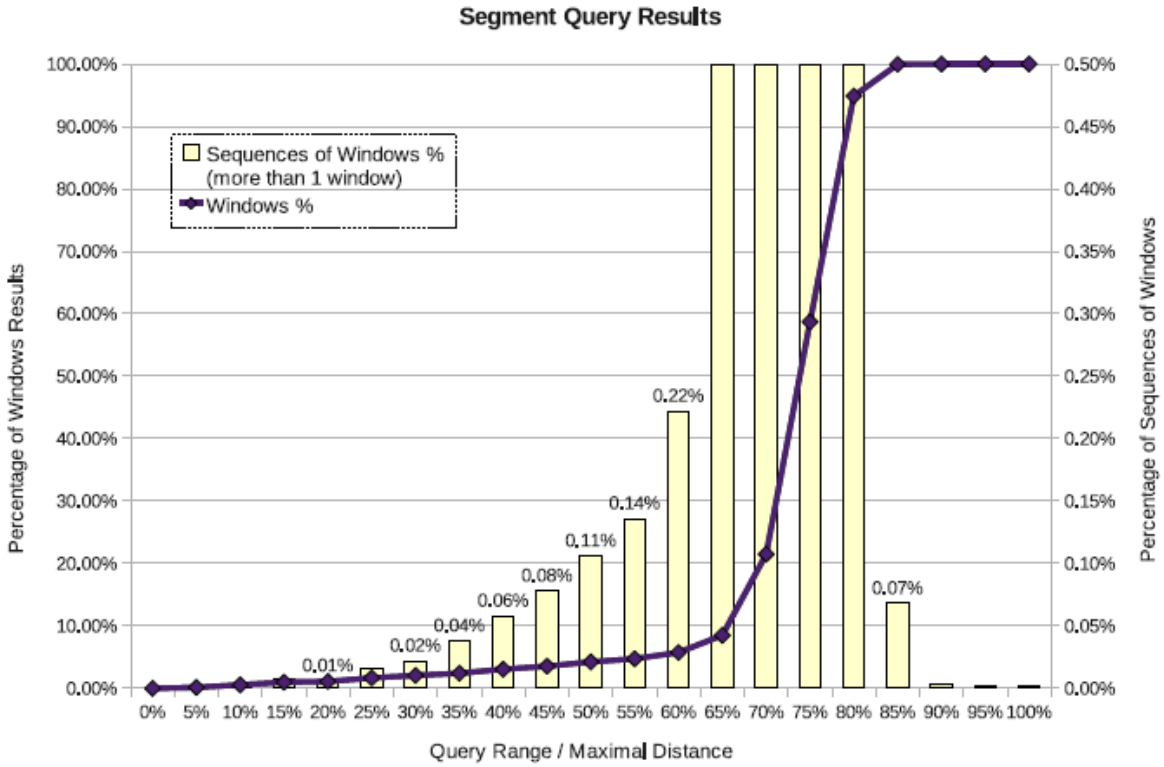
Experiments

- Query Computation Ratio (Distance Distribution)



Experiments

- Overall Subsequence Query Result



BOSTON UNIVERSITY

Conclusions & Future work

- A generic framework for subsequence matching.
- Defined a property for distance functions that allows efficient query processing and guarantees exact answers.
- Provided an index scheme that is generic for metric distances and fits well our framework.

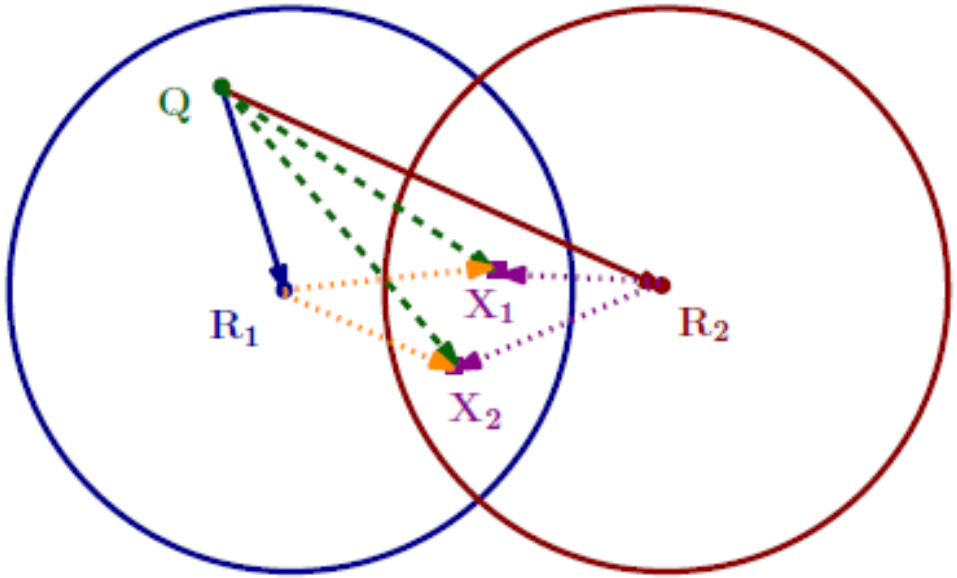
- Future work:
  - Improve the performance of the query part using GPUs and/or cluster-based system.

Thanks

BOSTON UNIVERSITY

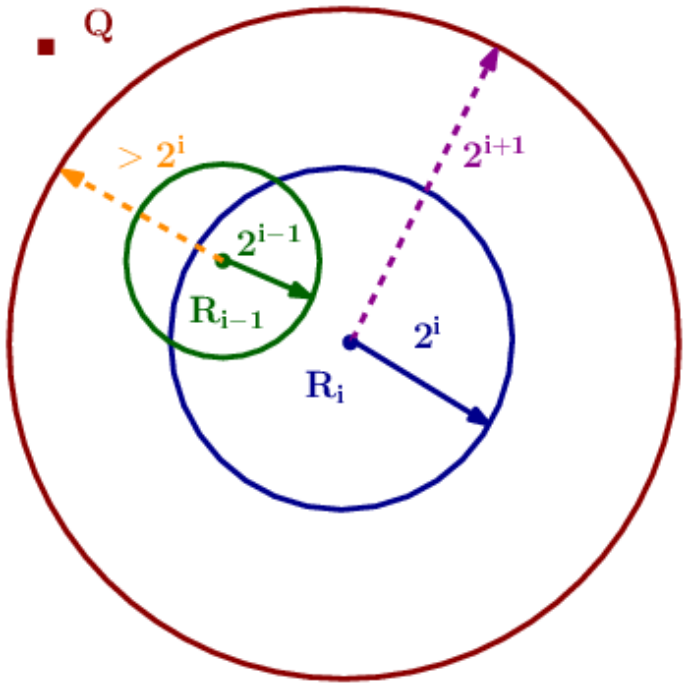
Reference Net

- Net, not Tree
  - we show why it is important to have a multiparent hierarchy and not a tree. Assume $\delta(R_1,X_i) \leq \varepsilon$ and $\delta(R_2,X_i) \leq \varepsilon$, but X_i are only in the list of R_2. If $\delta(Q,R_2) + \varepsilon > r$ we do not know whether $\delta(Q,X_i) \leq r$ or not. However, if we maintain X_i also in R_1, and $\delta(Q,R1) + \varepsilon \leq r$, we know $\delta(Q,X_i) \leq r$



Reference Net

- 4 Pruning Rules:
  - Single list removal
  - Single list commit
  - Generated list removal
  - Generated list commit



Experiment

- Query Computation Ratio (Tree VS Net)