BRAZILIAN
SYMPOSIUM ON
BIOINFORMATICS
BSB 2011

Efficiently Querying Protein Sequences with the Proteinus Index

Felipe Alves da Louza

Ricardo Rodrigues Ciferri

Cristina Dutra de Aguiar Ciferri

GBD
Grupo de Banco de Dados
UFSCar

ufscar

Gbd
Grupo de Bases de Dados e Imagens

ICMC USP
SÃO CARLOS

FAPESP

Contents

- Motivation
- Background
  - Reduced Alphabet
  - The nsP-index
- The Proteinus
- Experiments
- Conclusions

Motivation

- Finding similarities in protein sequences stored in biological databases has become a core problem in bioinformatics

- Applications
  - Protein evolution studies
  - Predicting biological structure
  - Functional characterization of novel protein sequences
  - …
  - And much more…

Protein sequence databases

- Protein sequence databases (PSD) store data related to the primary structure of proteins and their amino acids chains
  - These chains are stored in the databases as strings
    - Representing the 20 molecules present in proteins

$$\boxed{\Sigma_{20}^{p} = \{\mathrm{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}}$$

Similarities in PSD

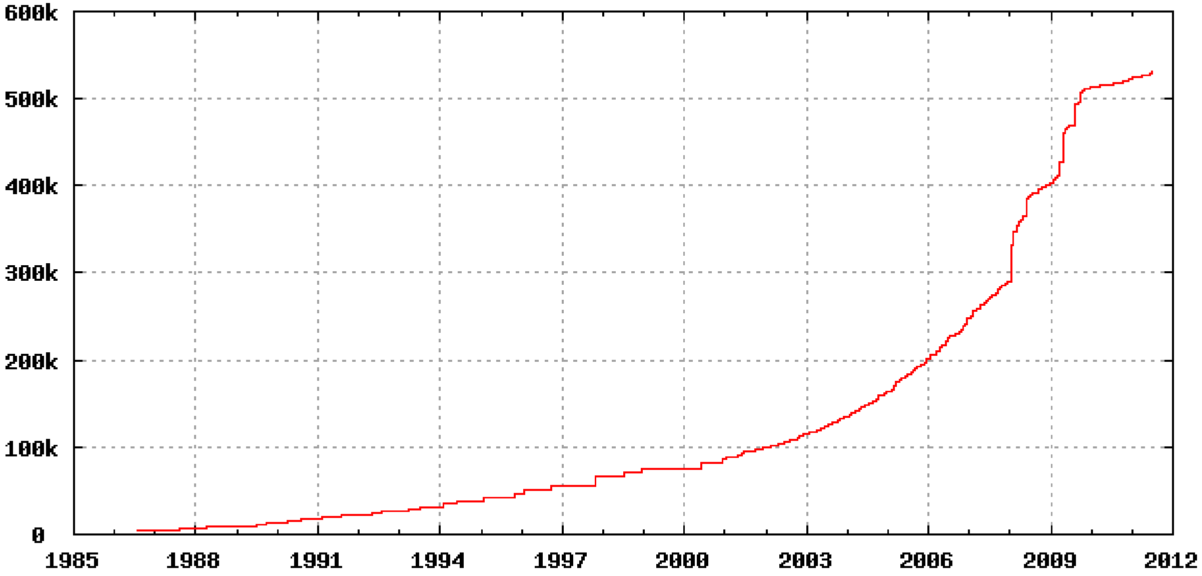
- An important type of query used to search for similarities in PSD is the range query

- The Edit Distance (ED) is the most used distance function to compare sequences
  - ED is defined as the minimum number of edit operations (*insertion*, *deletion* and *update*)

- Due to protein sequences characteristics, the ED function must consider weights related to amino acid exchanges
  - these weights are obtained by *replacing matrices*, such as the PAM and the BLOSUM

Production of biological data

With the technological advances in molecular biology, the amount of protein sequences has increased exponentially



Number of entries in UniProtKB/Swiss-Prot

Similarities in PSD

- In the literature, tools widely used to search for similarities in PSD are:
  - BLASTP, PSI-BLAST and PHI-BLAST

- Another research area uses indices to improves the performance of similarity search
  - MRS, ComRI, nsP-Index

However, these approaches are aimed at indexing nucleotide sequences databases, and are not directly applicable on PSD

Similarities in PSD

- PSD are different from nucleotide sequences databases
  - PSDs store a greater number of protein sequences
  - Protein sequences (typically of hundreds to thousands residues) are much smaller than nucleotide sequences (typically of millions of bases)
  - The amino acid alphabet is bigger than the nucleotide alphabet (A, C, G, T)

Recall that computing similarity among protein sequences is necessary to handle *replacing matrices* (e.g., PAM, BLOSUM, …)

Proposal

- We propose Proteinus index (acronym for Protein sEquence sImilarity Search)
  - a new protein sequence persistent index
- It uses a reduced amino acid alphabet
  - proposed by Li *et al.* [1]
- It allows the persistent storage of the index on disk
  - Proteinus extends the nsP-index

[1] Li, T., Fan, K., Wang, J., Wang, W.: Reduction of protein sequence complexity by residue grouping. Protein Eng. 16(5), 323-330 (2003)

Contents

- Motivation
- Background
  - Reduced Alphabet
  - The nsP-index
- The Proteinus
- Experiments
- Conclusions

Reduced Amino Acid Alphabet

- Some amino acids (e.g., ILE and LEU) are chemically similar, and can be replaced by one another without changing the global structure (the fold) or the function of a protein

- The amino acid alphabet can be reduced to a new alphabet wherein similar amino acids are clustered together

Reduced Amino Acid Alphabet

- The first reduced amino acid alphabet was introduced by Dill [2]:
  - Hydrophobic-polar (HP) model for study of the folding of globular proteins

- Since then, more than 50 reduced alphabets of different sizes have been proposed

[2] Dill K.: "Theory for the Folding and Stability of Globular Proteins" Biochemistry, 24:1501-1509, 1985

Reduced Amino Acid Alphabet

- Li et al [3] proposed and compared several reductions on the complete alphabet
  - by grouping amino acids based on their physical an chemical characteristics
  - by seeking to maximize a similarity score derived from the BLOSUM62 matrix

- The authors concluded that a reduced alphabet, which is composed by 10 groups, preserves the maximum information on the original protein sequence

Using a reduced alphabet in protein sequence comparisons avoids using a replacing matrix

Reduced Amino Acid Alphabet

$$\Sigma_{20}^{p} = \{\mathrm{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}$$

$$\downarrow$$

$$\boxed{\Sigma_{10}^{p} = \{C', F', M', I', G', P', A', N', Q', R'\}, such\ that}$$

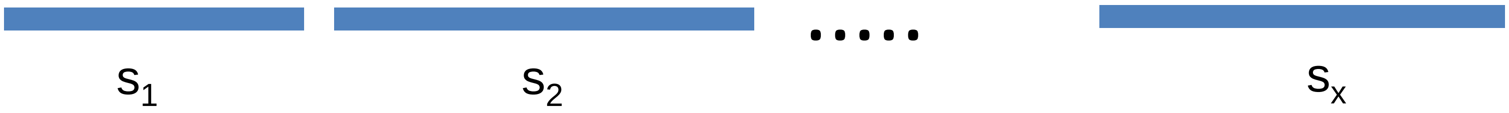
$$C' = \{C\},\ F' = \{F, Y, W\},\ M' = \{M, L\},\ I' = \{I, V\},\ G' = \{G\},\ P' = \{P\},$$
$$A' = \{A, T, S\},\ N' = \{N, H\},\ Q' = \{Q, E, D\},\ R' = \{R, K\}$$

Contents

- Motivation
- Background
  - Reduced Alphabet
  - The nsP-index
- The Proteinus
- Experiments
- Conclusions

The nsP-index

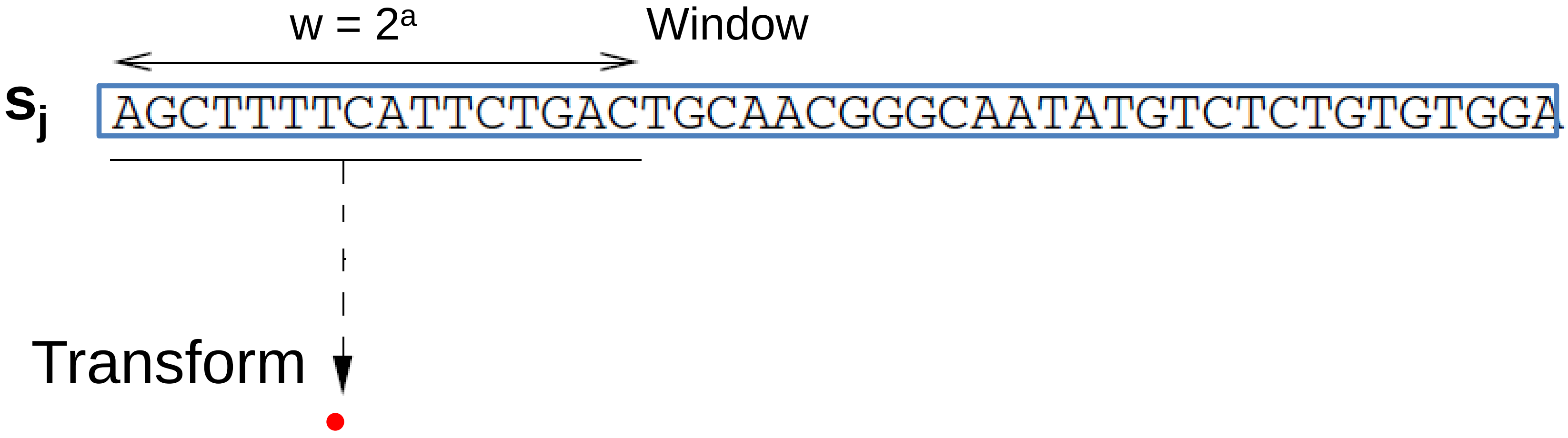
- The nsP-Index extends the MRS-index into a persistent index aimed at nucleotide sequences
  - Let S_n = {s1, s2, …, s_x} be a database composed of x nucleotide sequences
  - and w = 2^a be the length of the shortest query sequence



The nsP-index

- For all sequence s_j in the database
  - A window of length $w = 2^i$ ($a \leq i \leq b$) slides over the sequence s_j

- For each placement of the window, a subsequence of s_j is mapped into a point using a wavelet-based method



$w = 2^a$

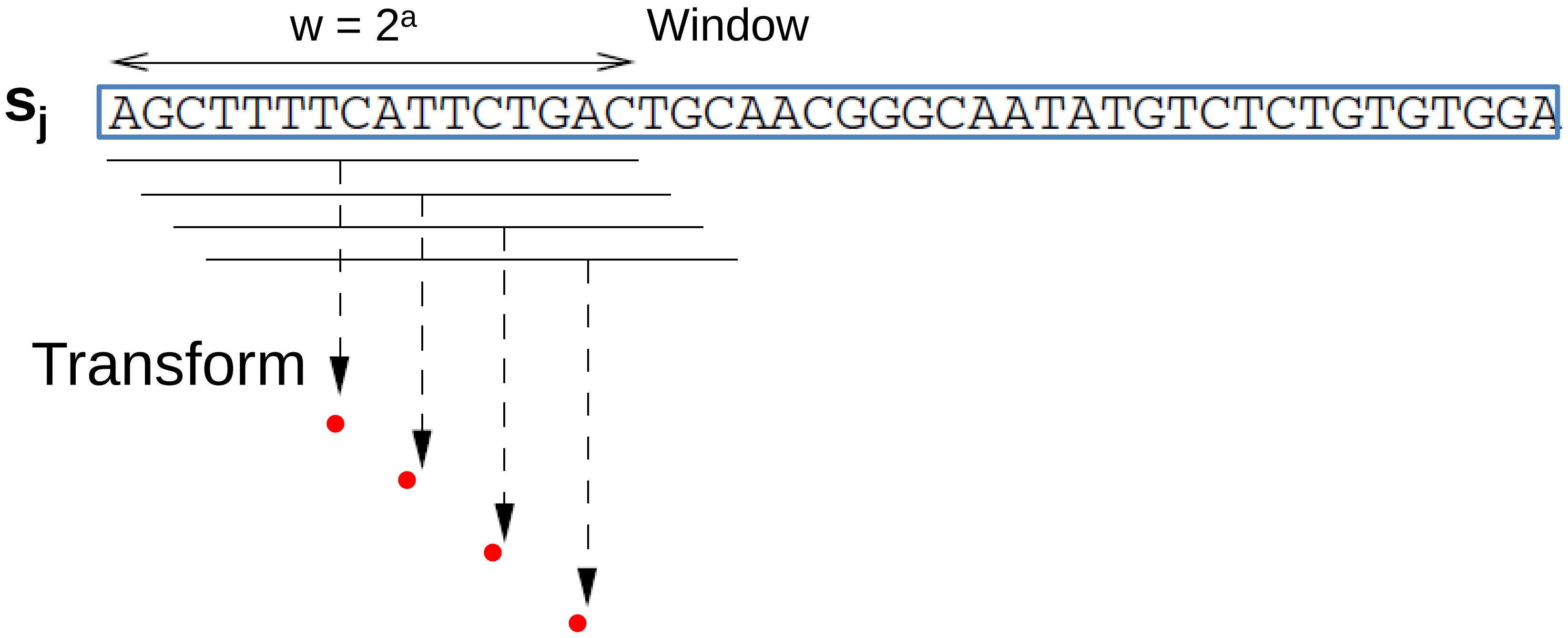
Window

s_j AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGA

Transform

The nsP-index

- The sliding window starts from the leftmost point to the rightmost point of the sequence



w = 2^a

Window

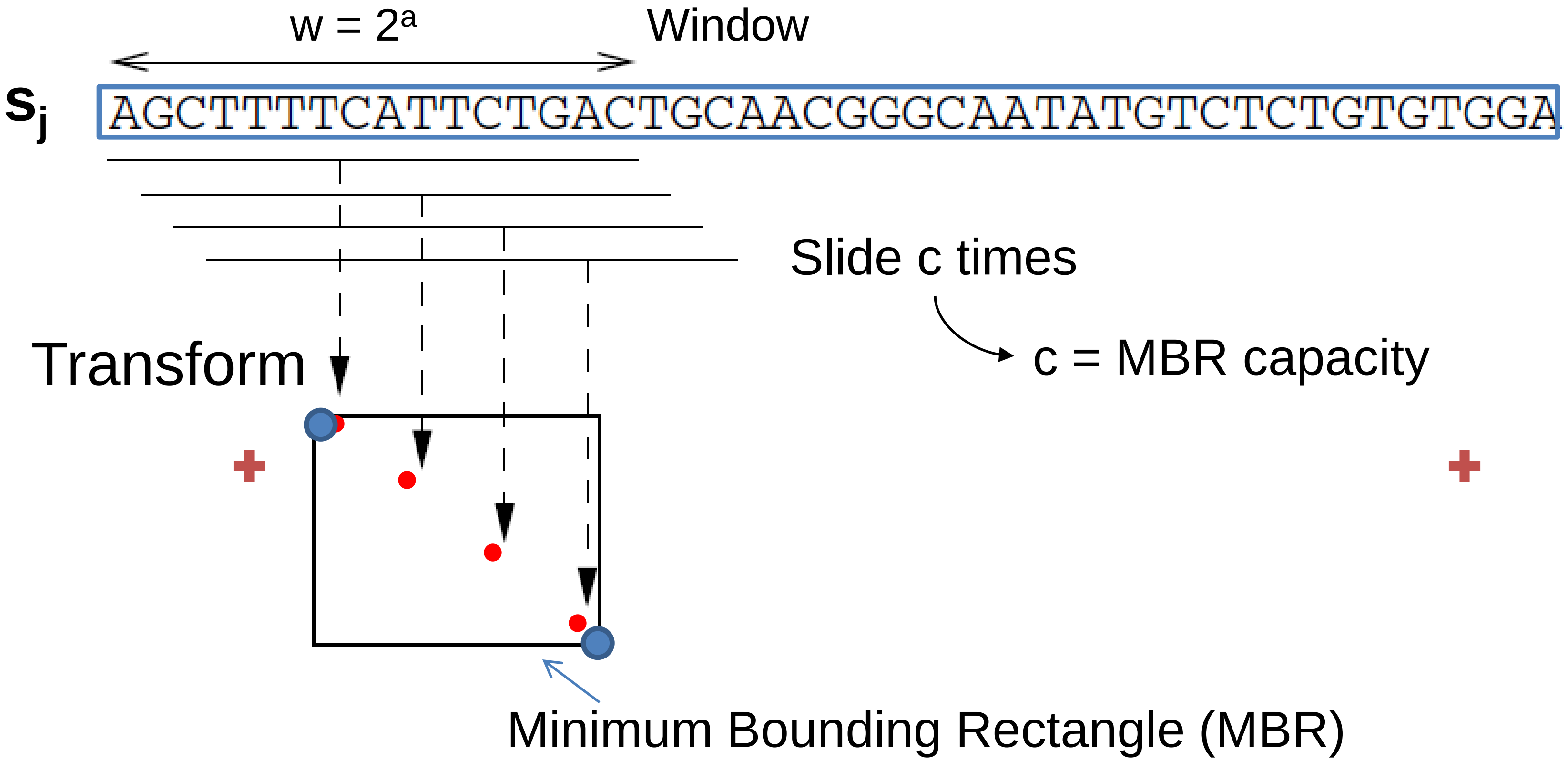
s_j

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGA

Transform

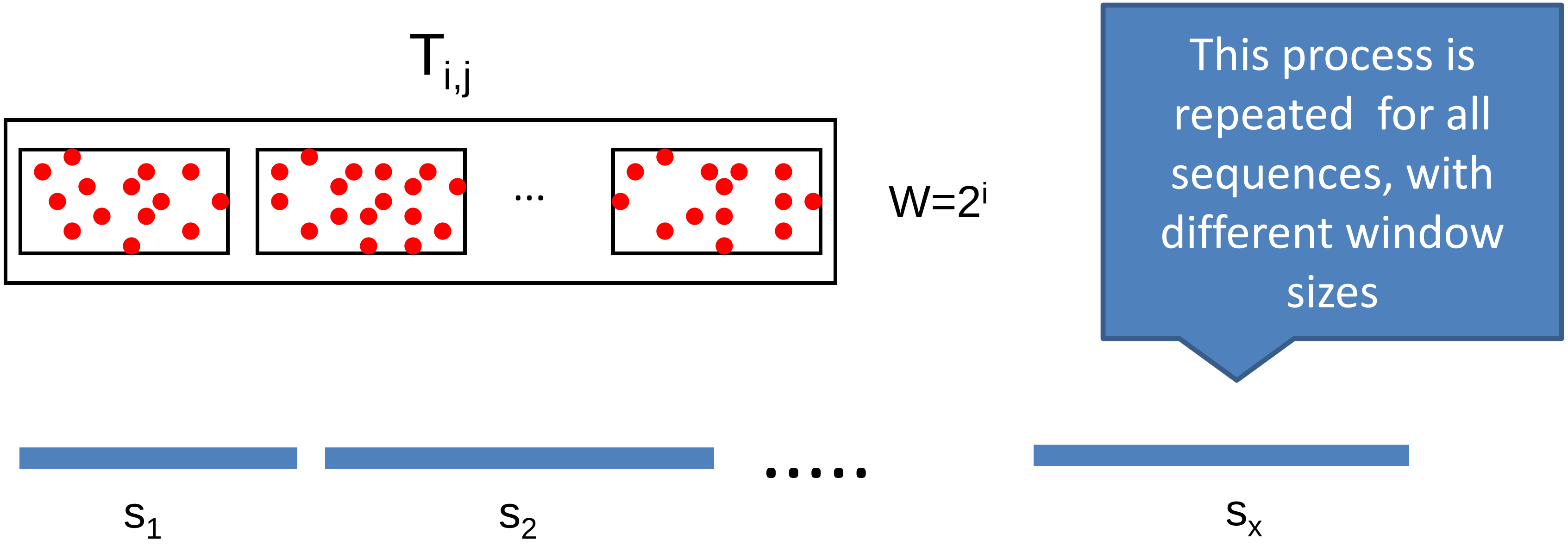
The nsP-index

- These points are comprised within a minimum bounding rectangle (MBR)


w = 2a
Window
sj
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGA
Slide c times
c = MBR capacity
Transform
Minimum Bounding Rectangle (MBR)

The nsP-index

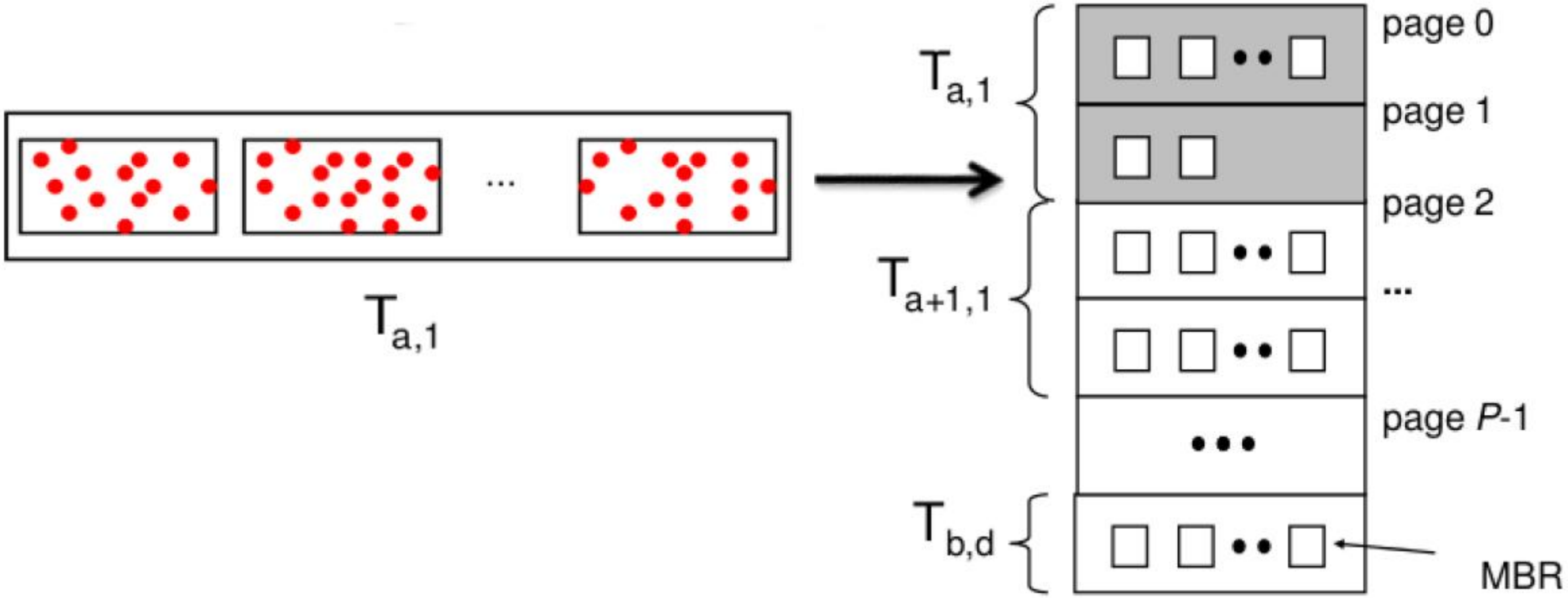
- A list of MBRs represents an element $T_{i,j}$
- Each element $T_{i,j}$ indexes the jth sequence with window sizes 2^i



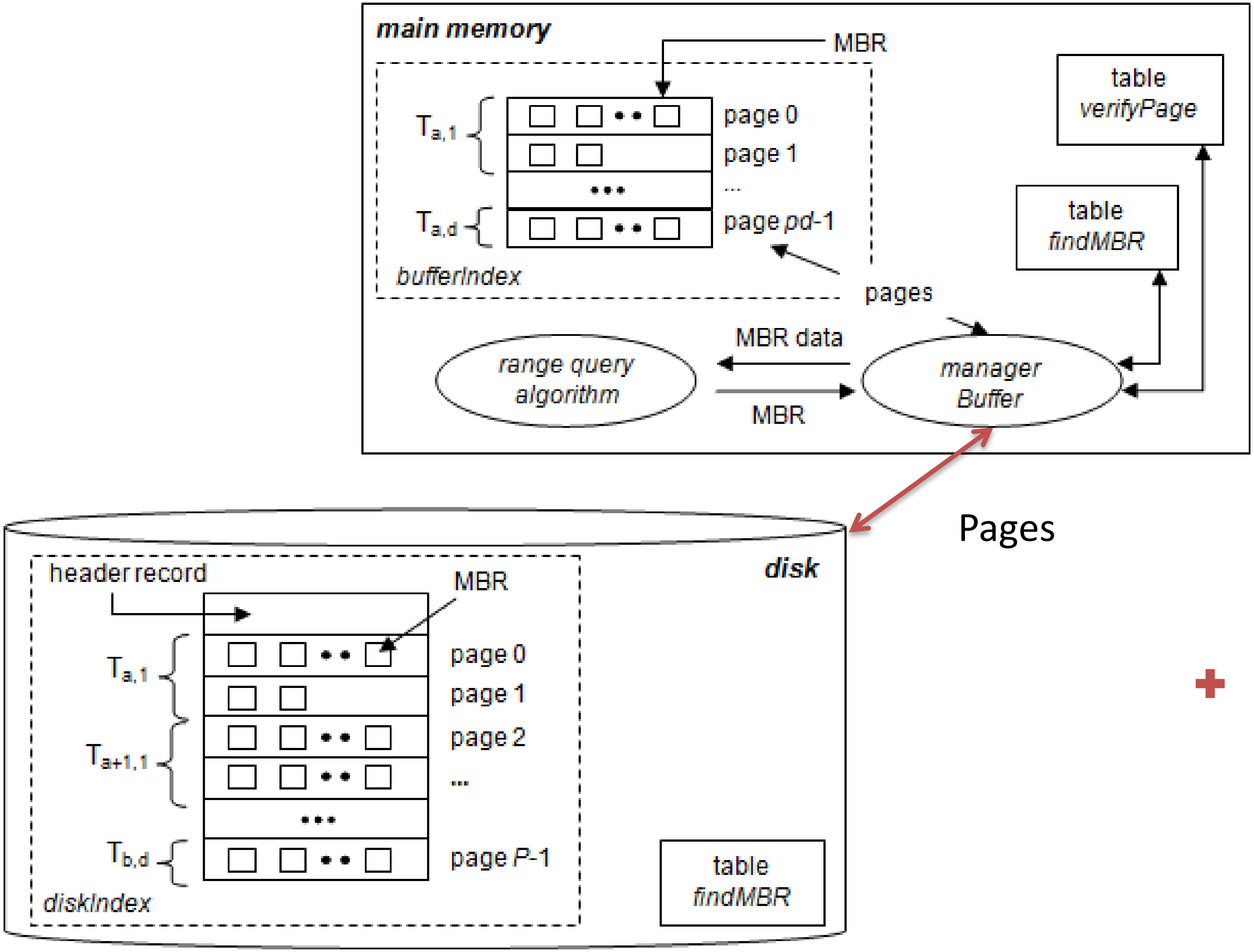
This process is repeated for all sequences, with different window sizes

The nsP-index

- These $T_{i,j}$ elements are stored on disk, in different pages

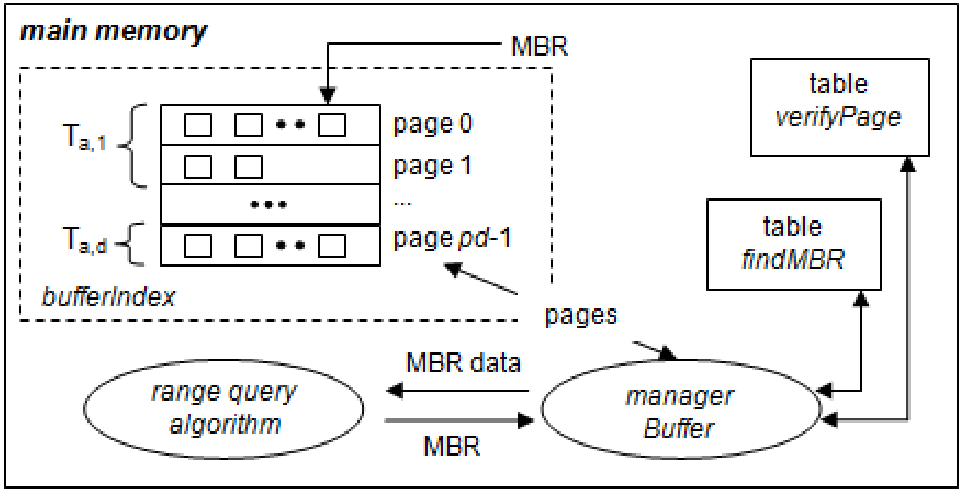


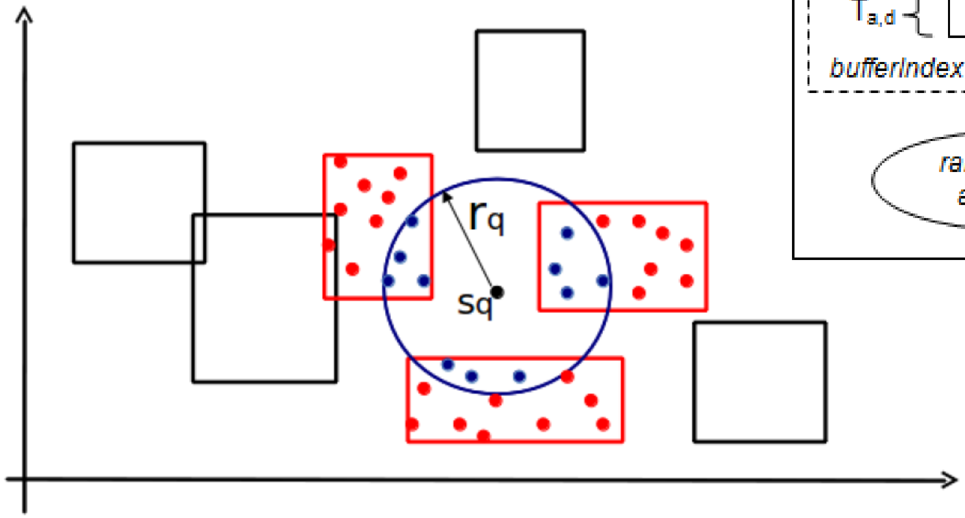
The nsP-index (physical view)



The nsP-index (range query)

- Range query processing calculates the distance between the (transformed point) query sequence and the MBRs
  - using the buffer-pool



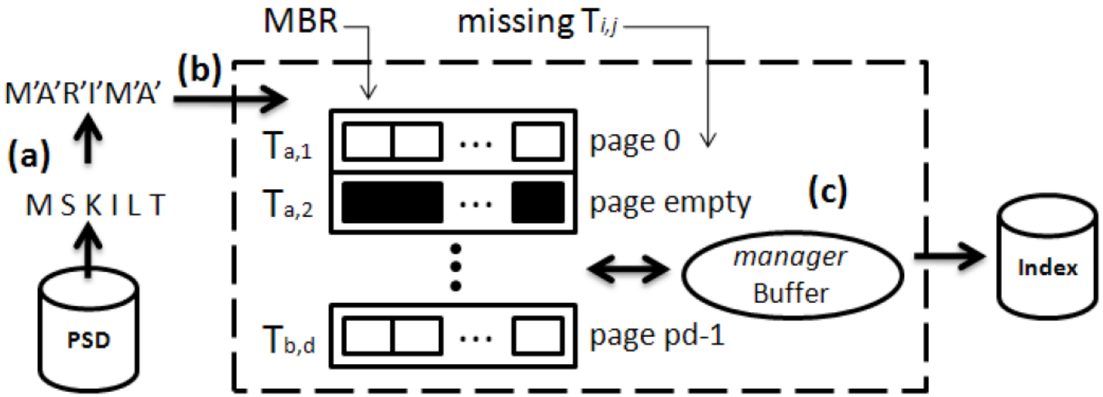


Contents

- Motivation
- Background
  - Reduced Alphabet
  - The nsP-index
- The Proteinus
- Experiments
- Conclusions

Proteinus

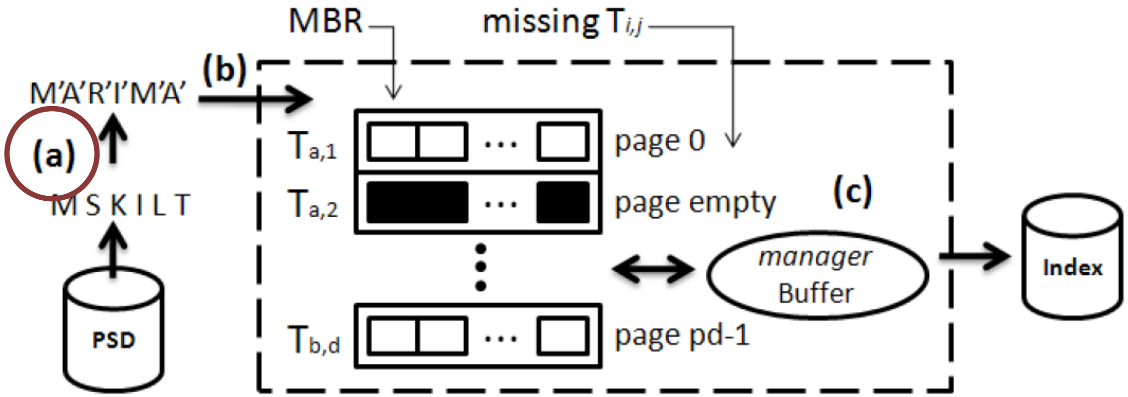
- The **Proteinus Index** extends the nsP-Index to process protein sequences
  - Inherits characteristics of the nsP-Index
  - Uses a reduced amino acid alphabet



Overview of the Proteinus approach

Proteinus

- The **Proteinus Index** extends the nsP-Index to carry on protein sequences
  - Proteinus inherits characteristics of the nsP-Index
  - It uses a reduced amino acid alphabet

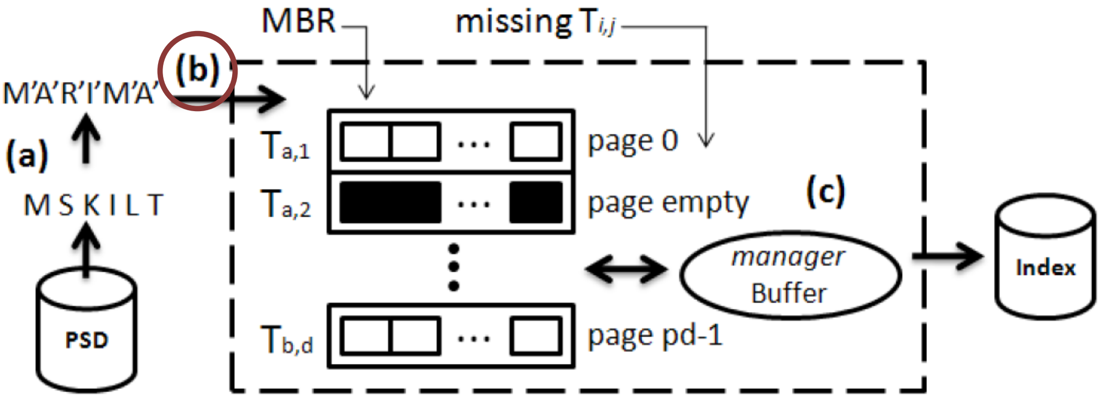


Overview of the Proteinus approach

A protein sequence is extracted from the PSD

Proteinus

- The **Proteinus Index** extends the nsP-Index to carry on protein sequences
  - Proteinus inherits characteristics of the nsP-Index
  - It uses a reduced amino acid alphabet

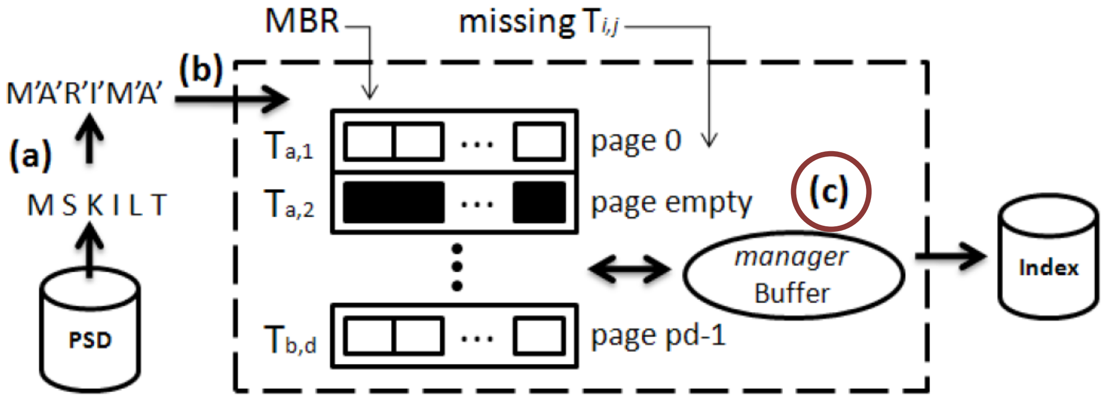


Overview of the Proteinus approach

The extracted protein sequence is converted following a reduced amino acid alphabet

Proteinus

- The **Proteinus Index** extends the nsP-Index to carry on protein sequences
  - Proteinus inherits characteristics of the nsP-Index
  - It uses a reduced amino acid alphabet



Overview of the Proteinus approach

The converted protein sequence is then indexed by Proteinus

Proteinus

- **Proteinus** focuses on five tasks**:**

1. It extends the index data structure to handle the protein sequence alphabet
2. It defines a mapping function to transform a subsequence s_j into a multidimensional point
3. It defines an approximated distance function among the transformed points
4. It adapts the index structure to handle missing $T_{i,j}$
5. It provides a range query routine to support the new characteristics of the index data structure

Proteinus - tasks

- To accomplish Task 1, we designed a dedicated Proteinus data structure to support the reduced alphabet:

$$\Sigma_{10}^{p} = \{C', F', M', I', G', P', A', N', Q', R'\}$$

- Maintaining the same memory requirements of the nsP-Index:
  - Through the use of a shorter data type to represent a protein sequence

1. Extends the index data structure to handle the protein sequence alphabet

Proteinus - tasks

- Proteinus's data structure increases twice the storage requirements
  - as the reduced amino acid alphabet has 10 symbols instead of 5 symbols to handle nucleotide sequences.

- Due to the small sizes of the protein sequences
  - Proteinus-index decreases by half the storage requirements through the use of a shorter data type.

Therefore, we used a 16 bit variable in the Proteinus Index to represent each multidimensional point, while the nsP-index uses a 32 bit variable

Proteinus - tasks

- Regarding tasks 2 and 3,
  - The mapping and approximated distance functions should be easily computable

  - To do so, we used:
    - The wavelet-method inherited from the nsP-Index
    - Together with the reduced alphabet

2. Defines a mapping function to transform a subsequence sj into a multidimensional point
3. Defines an approximated distance function among the transformed points

Proteinus - tasks

- Task 4 is required since the protein sequence sizes vary greatly
  - Thus, not all sequences can be indexed in all resolutions
  - To this end, Proteinus data structure allow missing $T_{i,j}$
  - When a sequence s_i is shorter than a window resolution, an empty $T_{i,j}$ is generated
    - Empty Ti,j do not persist on disk

4. Adapts the index structure to handle missing $T_{i,j}$

Proteinus - tasks

- To accomplish Task 5, we extended the nsP-index range query algorithm to support missing $T_{i,j}$
  - When a query sequence s_q is larger than a sequence s_j in PSD
    - Our search algorithm bypasses sj in the similarity evaluation task

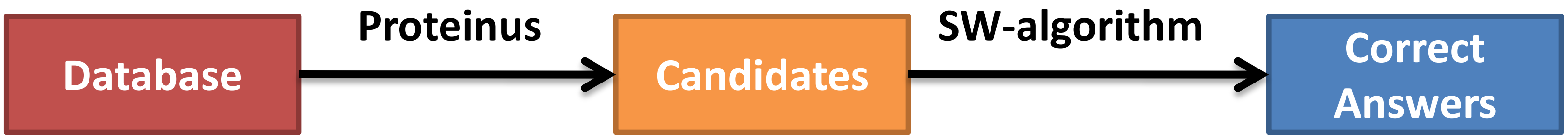
5. It provides a range query routine to support the new characteristics of the index data structure

Proteinus

- Finally, the use of approximations, such as MBRs, introduces loss of accuracy in the exact representation
  - Proteinus returns candidates (in MBRs)
  - Thus, Proteinus should provide both a filter phase and a refinement phase

++

  - To this end, we also developed a simple post-processing routine to eliminate false answers
    - **Smith-Waterman algorithm**



Contents

- Motivation
- Background
  - Reduced Alphabet
  - The nsP-index
- The Proteinus
- Experiments
- Conclusions

Setup of experiments

- The experiments were conducted on a computer with:
  - Intel Core i7 2.67 GHz processor,
  - 12 GB of main memory,
  - 2 SATA 1 TB hard disks and
  - Linux Ubuntu 9.04.
  - We employed the BLASTP version 1:2.2.21.20090809-1.

- Proteinus was implemented in C++

- We used four resolutions with **window sizes** of 32, 64, 128 and 256 characters, and set the **MBR capacity** to 80 strings

- The **bufferIndex** size was set to 200 MB and the disk page size **c** was set to 672 bytes.
  - These values were identified experimentally

Setup of experiments

- In our tests, we used real-world protein sequences obtained from the Uniprot database
  - We created several PSDs containing increasing volumes of protein sequences: 4 GB, 6 GB, 8 GB (original dataset) and 10 GB (synthetic dataset)
- We perform range queries by
  - submitting four different sizes of queries to each PSD
    - 128, 256, 512 and 1,024 amino acids
  - With an error rate E = 0.005 to calculate query radius

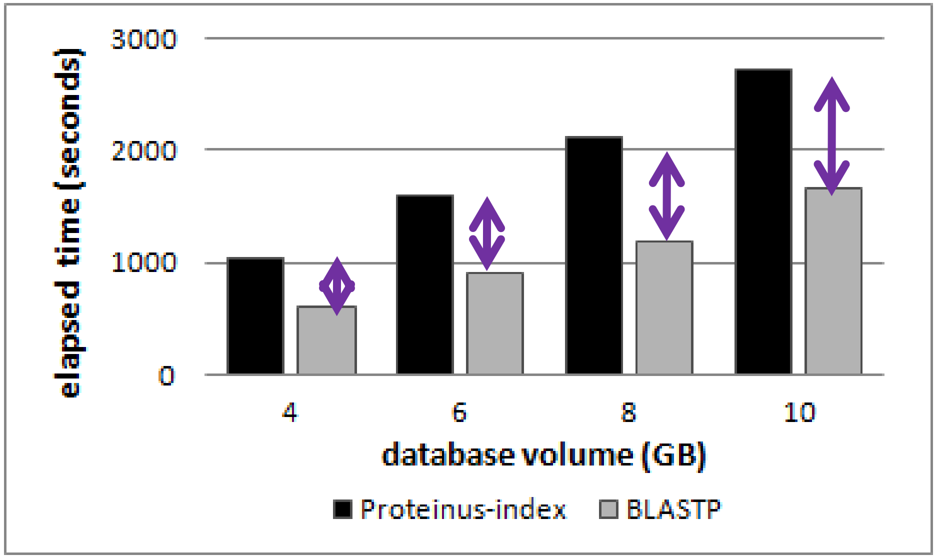
Setup of experiments

- We compared the performance of Proteinus with the BLASTP tool
  - Taking into account the elapsed time (in seconds)

- Although Proteinus is not a complete alignment strategy as the BLASTP…..
  - to have a measure of comparison

Experiment 1 – Index Construction

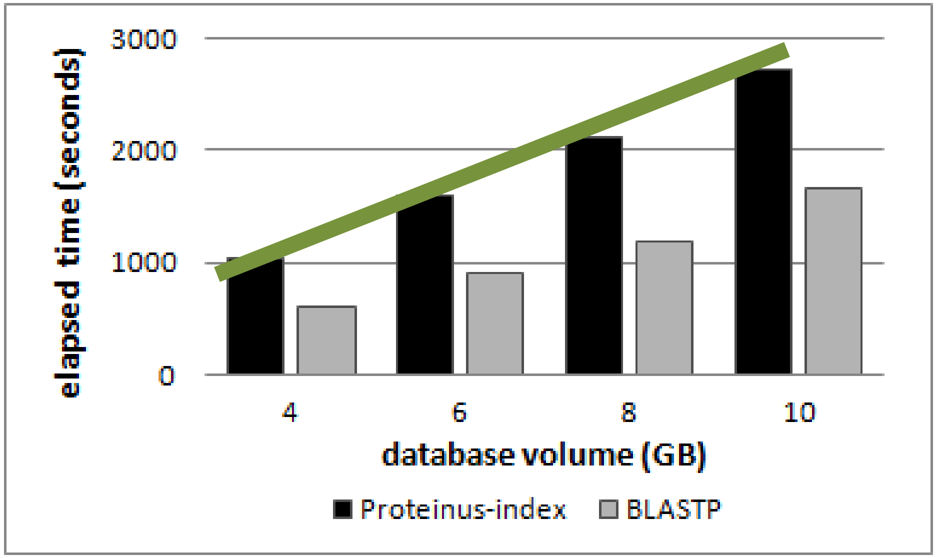
- Time spent by Proteinus to build the indices
  - Compared to the time spent by BLASTP to create its data structure (command *formatdb*)



The time spent by Proteinus was (on average 43%) higher than that spent by BLASTP

Experiment 1 – Index Construction

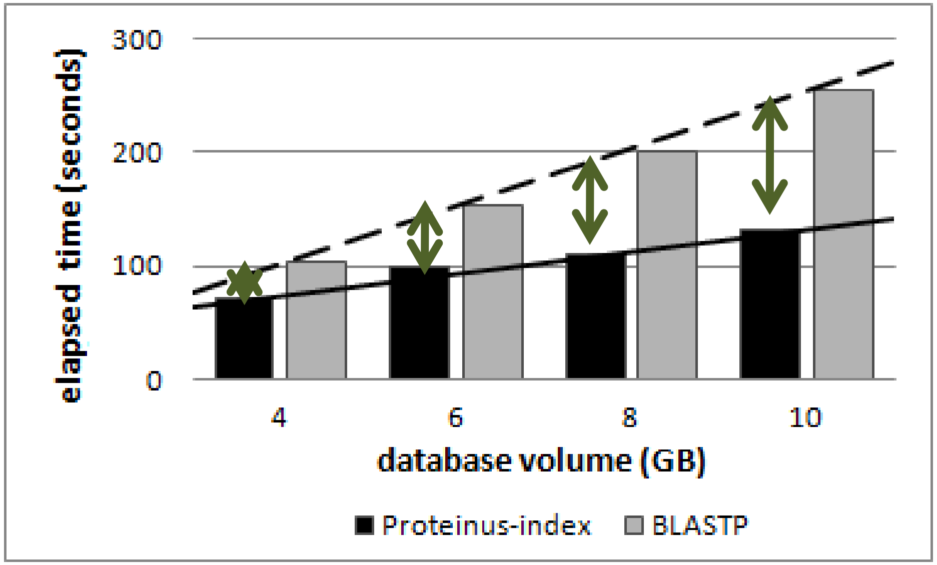
- Time spent by Proteinus to build the indices
  - Compared to the time spent by BLASTP to create its data structure (command *formatdb*)



However, Proteinus proved to be scalable

Experiment 2 – Range Query

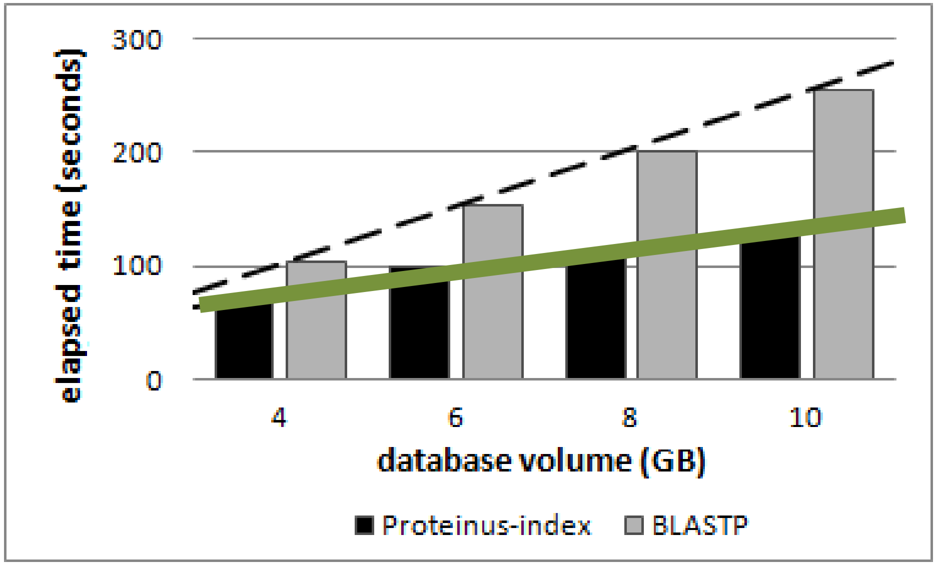
- Proteinus provided an impressive performance gain that ranged from 45% up to 93% over the BLASTP



The larger the dataset, the better the performance gain

Experiment 2 – Range Query

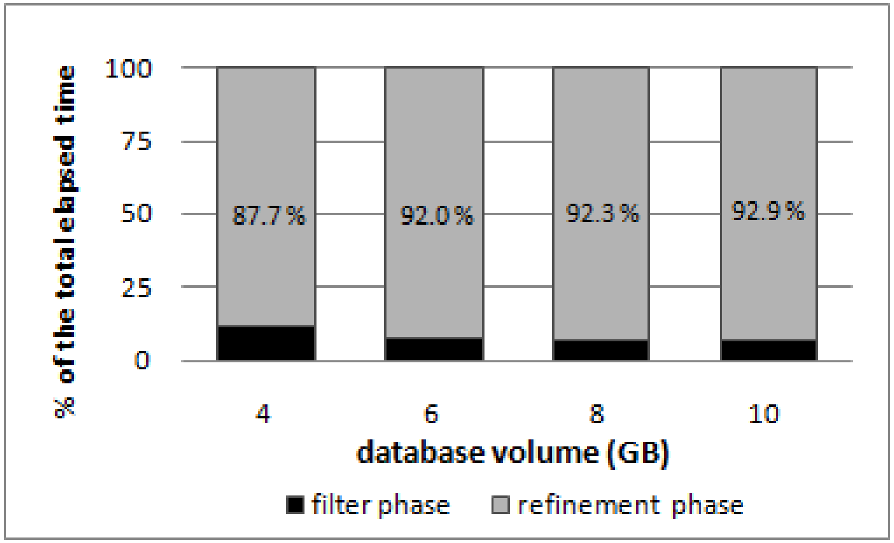
- Proteinus provided an impressive performance gain that ranged from 45% up to 93% over the BLASTP

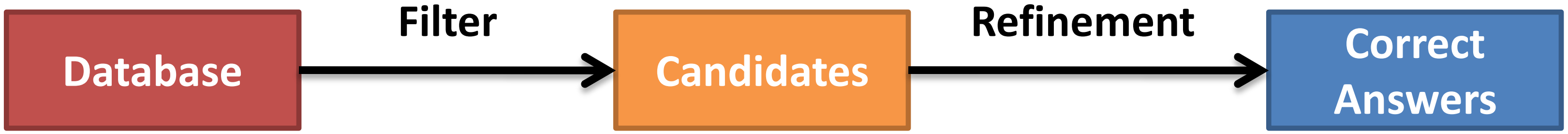


Also, Proteinus proved to be scalable in query processing

Experiment 3 – Range Query

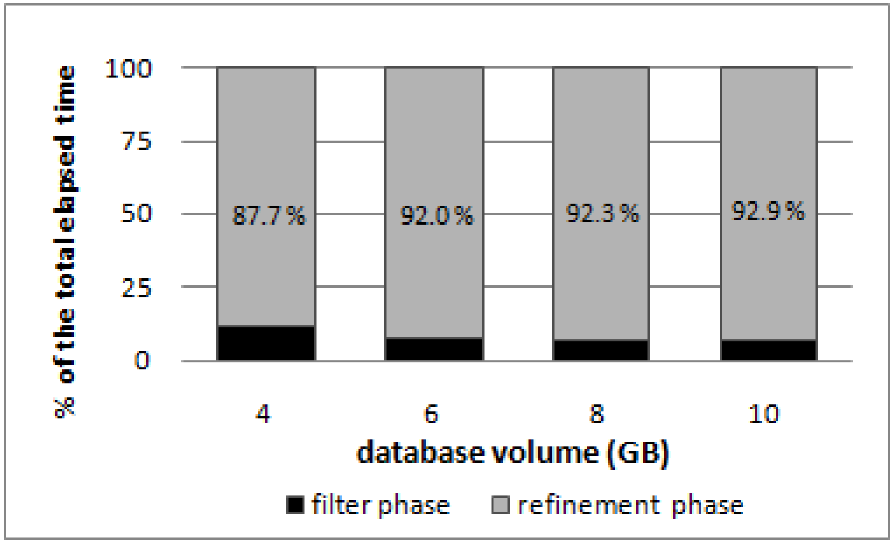
- Percentage of the total elapsed time spent by Proteinus to process the filter and the refinement phases separately





Experiment 3 – Range Query

- Percentage of the total elapsed time spent by Proteinus to process the filter and the refinement phases separately



The larger the dataset, the cost of filter phase is maintained

Contents

- Motivation
- Background
  - Reduced Alphabet
  - The nsP-index
- The Proteinus
- Experiments
- Conclusions

Conclusions

- We presented Proteinus, a new index based on approximations to process range queries over PSD

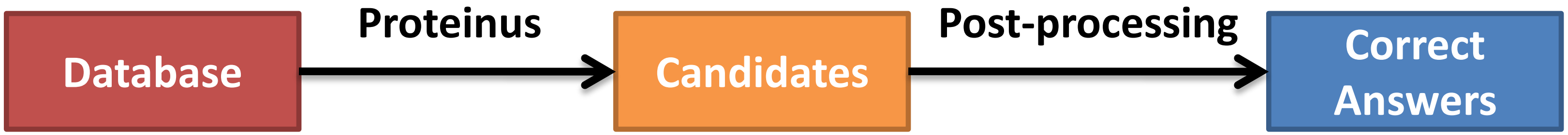
- The results showed that Proteinus was efficient in query processing, providing an performance gain that range from 45% up to 93% over BLASTP tool
  - Despite the time need for index creation

- Proteinus proved to be scalable both
  - Index creation
  - Query processing

Conclusions

- Proteinus can use any reduced amino acid alphabet

- Proteinus can be used as an efficient filter phase to select a reduced set of protein sequences
  - which can be further analyzed by any post-processing tool



Questions?
Suggestions?

Felipe Alves da Louza

louza@grad.icmc.usp.br - University of São Paulo

São Carlos – SP, Brazil

GBD
Grupo de Banco de Dados
UFSCar

ufscar

Gbdi
Grupo de Bases de Dados e Imagens

ICMC USP
SÃO CARLOS

FAPESP

Similarities in PSD

- An important type of query used to search for similarities in PSD is the range query
  - Consider that S = {s1, s2, …, sd} is a PSD
  - Sq represents a protein sequence query
  - And d(si, sq) represents the distance between a protein sequence si stored in S and sq
  - Given a query radios rq, the range query retrieve every protein sequence that satisfies the condition d(si,sq) <= rq.