Relational Text-type for Biological Sequences

Cristian Tristão, Antonio Basílio de Miranda,
Edward Hermann Haeusler and Sérgio Lifschitz

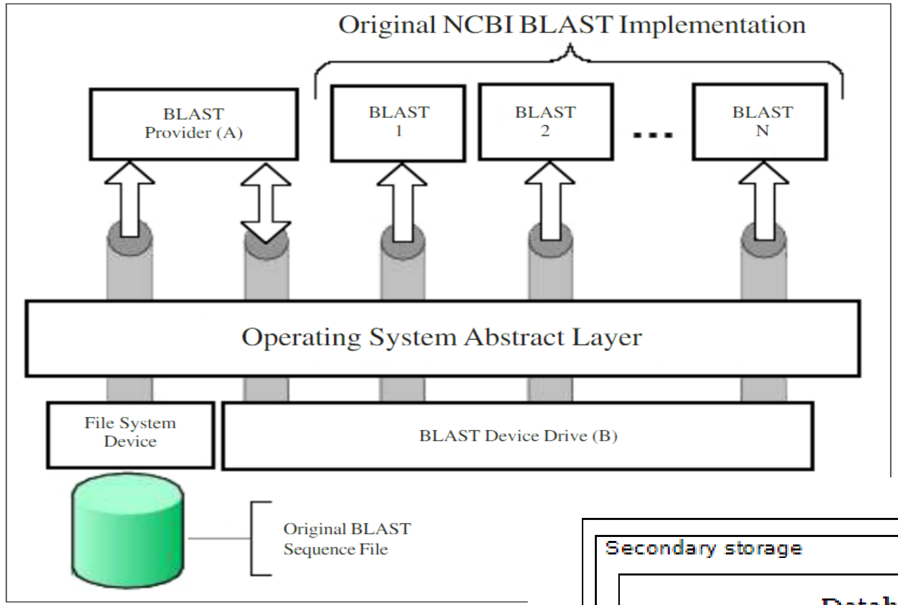
bioBD BIOINFORMÁTICA E BANCO DE DADOS
PUC RIO
DEPARTAMENTO DE INFORMÁTICA PUC-RIO
Ministério da Saúde
FIOCRUZ
Fundação Oswaldo Cruz

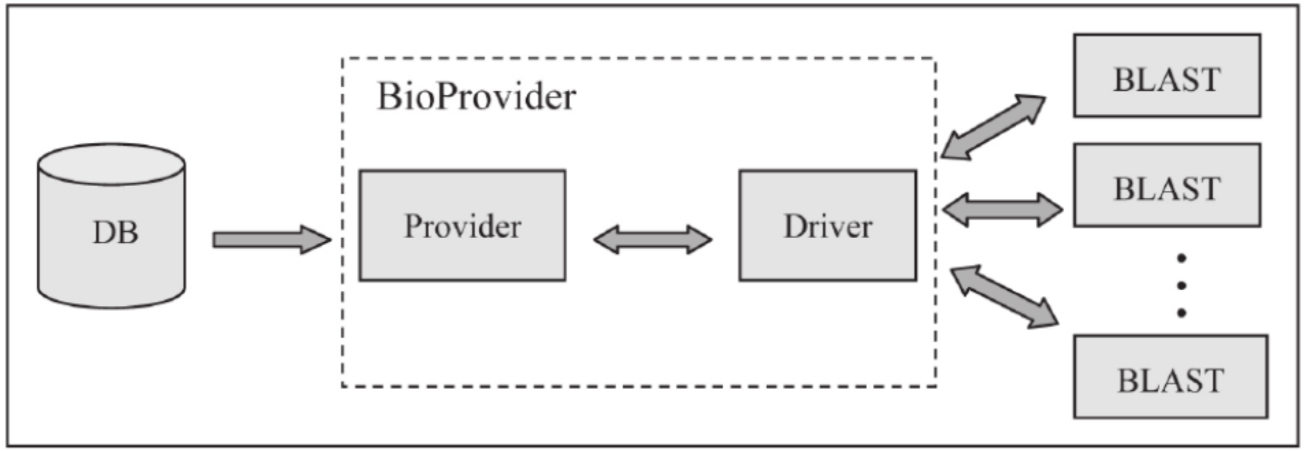
CMLS 2020

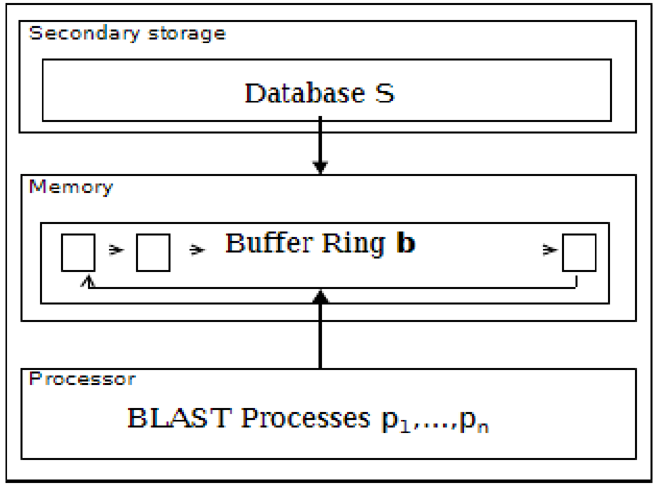
ER 2020

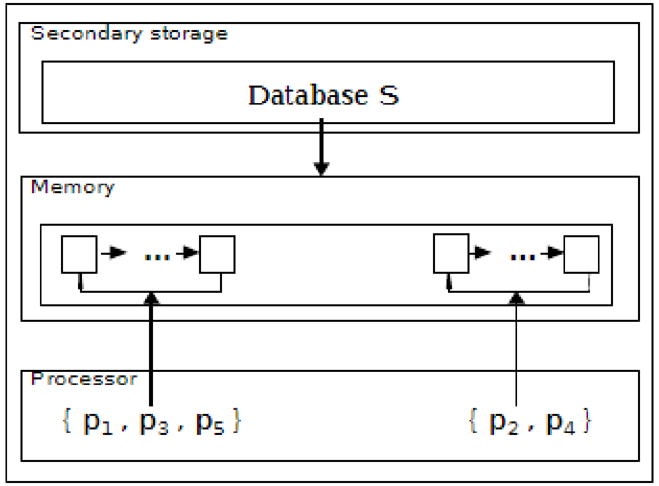
November 3-6, 2020 in Vienna, Austria

BioBD: DB-oriented approaches

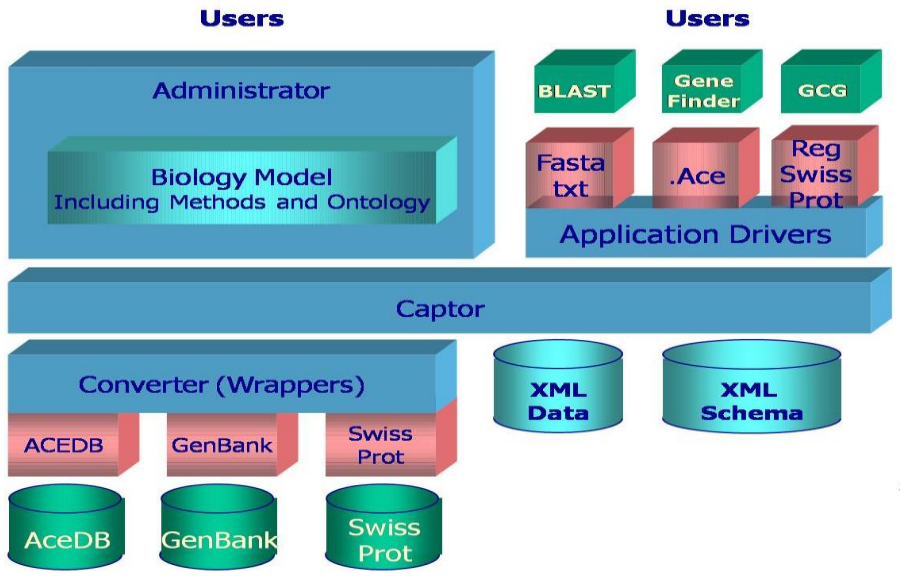

Original NCBI BLAST Implementation
BLAST Provider (A)
BLAST 1
BLAST 2
...
BLAST N
Operating System Abstract Layer
File System Device
BLAST Device Drive (B)
Original BLAST Sequence File

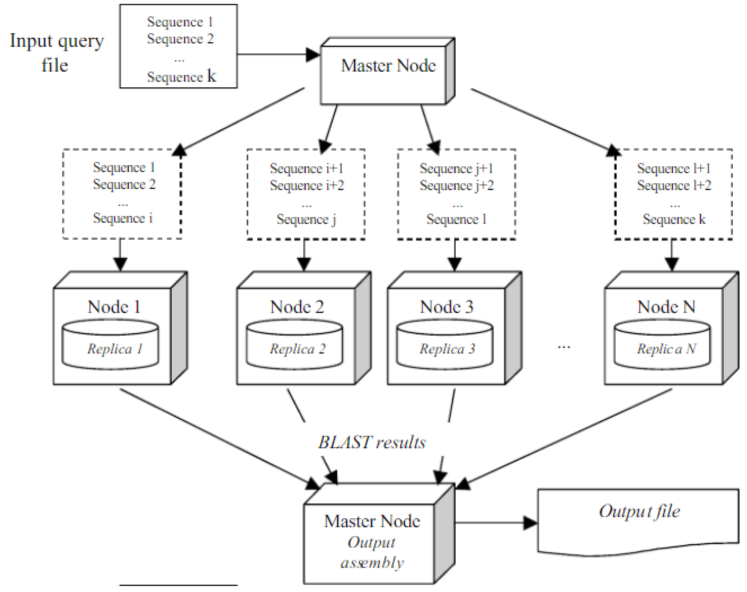




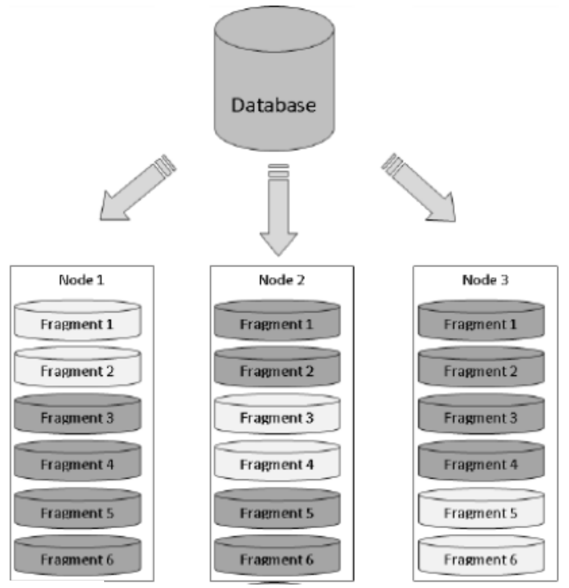


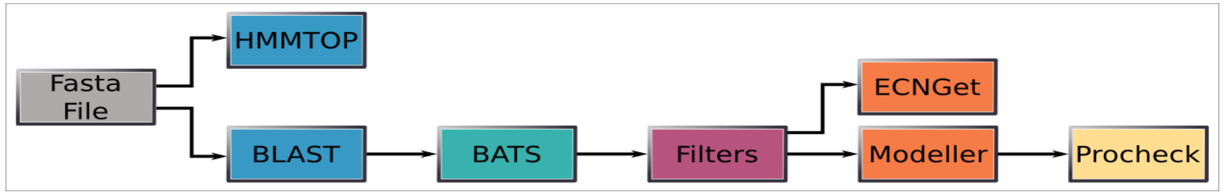
BioBD: DB-oriented approaches





ıre 4. Replicated schema.

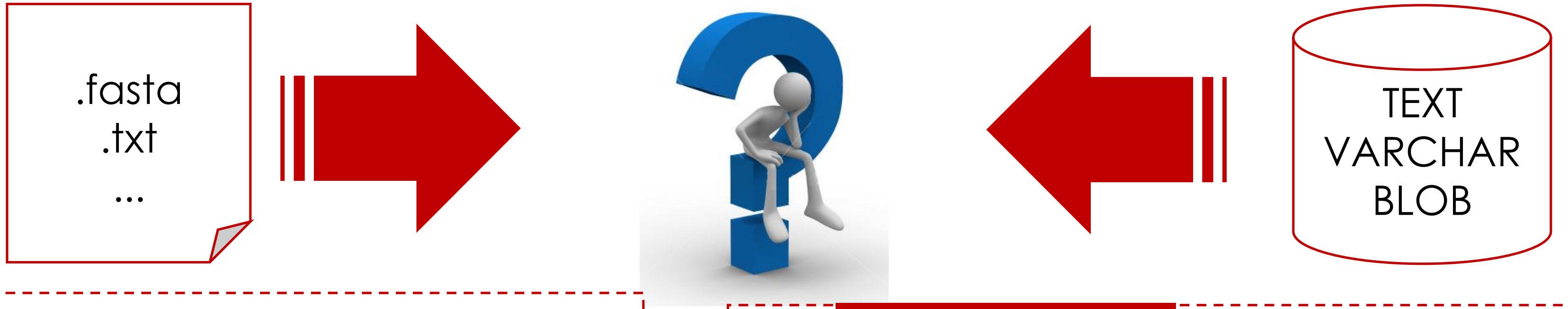




Research question: VL... Sequences!

Nucleotide Sequences

atgaaggcaatactagtagtctgctatatacatttgcaaccgcaaatgcagacacattatgtataggttatcatgcgaacaattcaacagacact
gtagacacagtactagaaaagaatgtaacagtaacacactctgttaaccttctagaagacaagcataacgggaaactatgcaaactaagag
gggtagccccattgcatttgggtaaatgtaacattgctggctggatcctgggaaatccagagtgtgaatcactctccacagcaagctcatggtcct
acattgtggaaacacctatttaccaggtgttcagacaatggaacgtgttacccaggagatttcatcgattatgaggagctaagagag ...



scripts

functions

- replace(string text, from text, to text)
- strpos(string, substring)
- substr(string, from [, count])
- translate(string text, from text, to text)

Bio-strings as ADTs

CLAIM: strings and BLOBs → no semantics!

Also: no standard for sequence persistency

Then: Bio-strings as "new data types"

Alternatives:

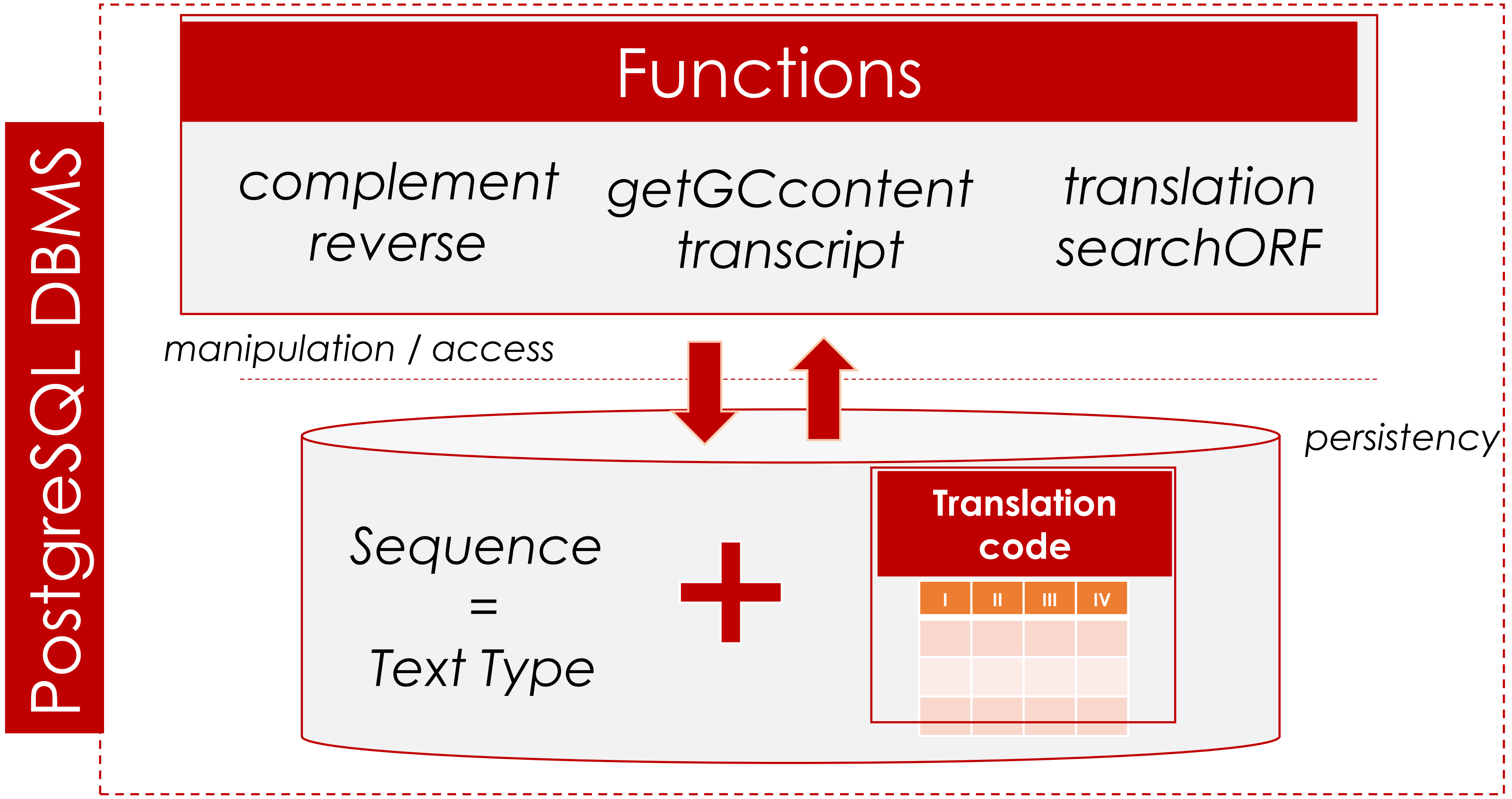
1. From scratch: modeling + implementation!
2. Extensions: already exists but more abstractions

Inspiring Idea: temporal and geo DBs

E.g date/time type

- Representation + persistent view
  - 03/11/2020 (dd/mm/aaaa)
  - 11-03-2020 (mm-dd-aaaa)
  - 20201103 (aaaammdd)
- Functions and access methods
  - SELECT EXTRACT(DAY FROM TIMESTAMP '2020-11-03' = 03
  - date '2020-11-03' + interval '1 hour' = timestamp '2020-11-03 01:00:00'

Towards relational ADTs



Useful functions (1/2)

complement(“sequence”)

complement (‘ACGGCTATTTAGAC’) = TGCCGATAAATCTG

reverse(“sequence”)

reverse (‘ACGGCTATTTAGAC’) = CAGATTTATCGGCA

getGCcontent (“sequence”)

getGCcontent(‘ACGGCTATTTAGACT’) = 6

Useful functions (2/2)

transcript("sequence")

transcript('ACGGCTATTTAGACT') = ACGGCUAUUUAGACU

translation ("position", "sequence")

translation(2,'ACGGCTATTTAGACT') = RLFR

searchORF ("position", "sequence", "size")

searchORF(1,'ACGAUGCUAUUUAGAUAGCUG', 10) = AUGCUAUUUAGAUAG

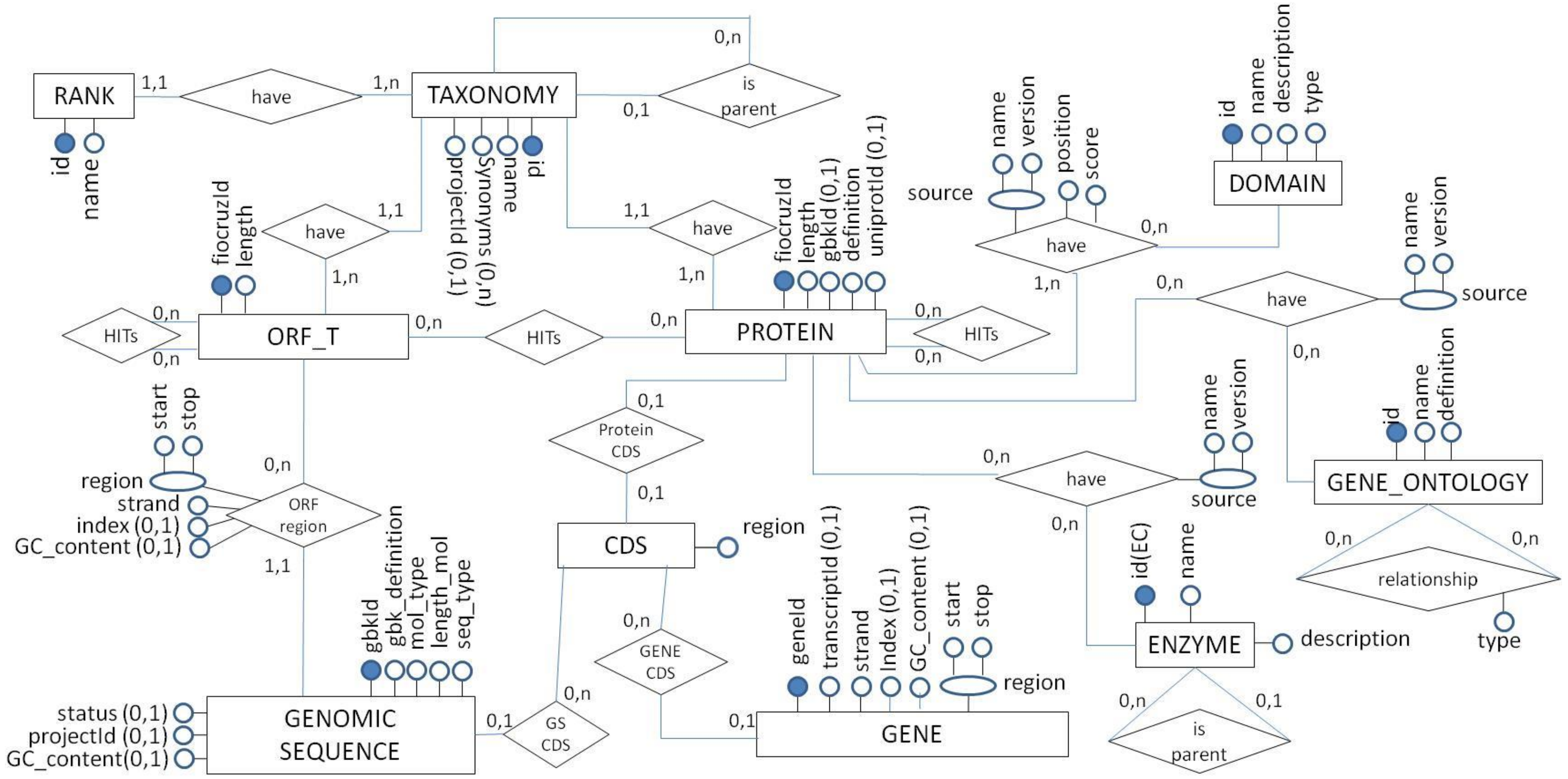
Case study: similarity management

```
query gi, subject gi, SW score, bit score, e-value, % identity, alignment length, query start, query end, subject start, subject end, query gaps, subject gaps

67523787,67540134,2166,488.8,2.6e-138,0.336,1320,35,1275,67,1367,79,19
```

- Proteins from original genomic sequences?
- Given a taxonomic group, how many genomes or proteins?
- Amount of hits for a specific protein?
- Are the unique genes? Paralogous or Orthologous?

Pure conceptual schema



CMLS 2020

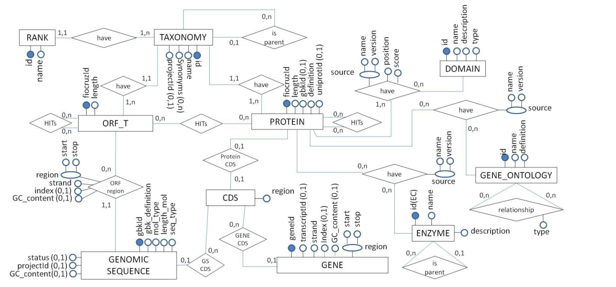
ER 2020

November 3-6, 2020 in Vienna, Austria

Proposal Overview

Analysis

Conceptual Schema



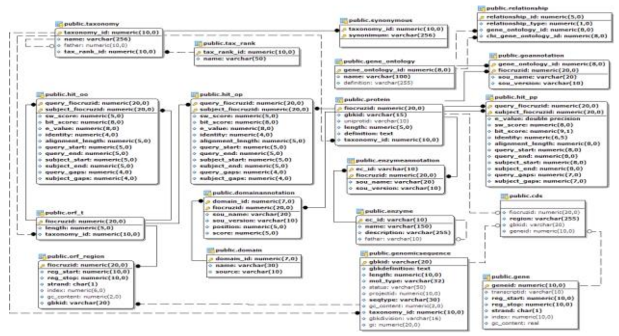
+

Domain Queries

- Comparing proteins
- Proteins from Genomic sequences
- Taxonomies and amount of proteins
- Hits for given proteins
- Unique genes
- Paralogues and orthologues

Validation

Logical Schema



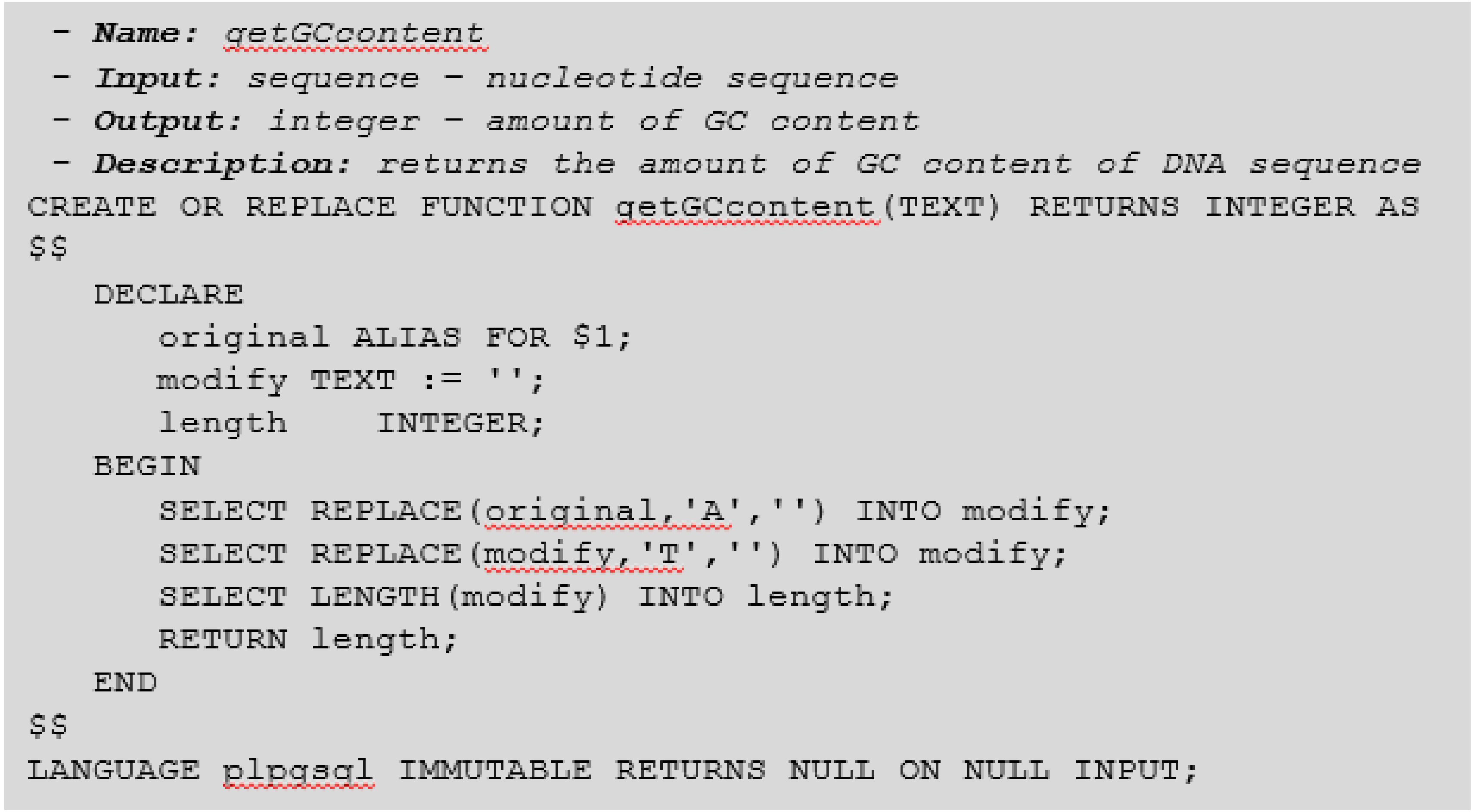
+

SQL/PGSQL

- getTaxonomyIdChildren
- getTaxonomyIdChildrenSet
- getCountGenomeTaxonomy
- getCountProteinTaxonomy
- getCountHitsProtein

- getProteinTaxonomy
- getSimilarProtein
- getSingleGene
- getOrthologousGene
- getParalogousGene

PostgreSQL implementation



```
  - Name: getGCcontent
  - Input: sequence - nucleotide sequence
  - Output: integer - amount of GC content
  - Description: returns the amount of GC content of DNA sequence
CREATE OR REPLACE FUNCTION getGCcontent(TEXT) RETURNS INTEGER AS
$$
   DECLARE
      original ALIAS FOR $1;
      modify TEXT := '';
      length    INTEGER;
   BEGIN
      SELECT REPLACE(original,'A','') INTO modify;
      SELECT REPLACE(modify,'T','') INTO modify;
      SELECT LENGTH(modify) INTO length;
      RETURN length;
   END
$$
LANGUAGE plpgsql IMMUTABLE RETURNS NULL ON NULL INPUT;
```

Final Comments

Relational model still alive: Bio-strings are OK!

Problem: lack of semantics in existing data structures and types

Github: https://github.com/sergiolif/BioBD_SGBDBio

Obrigado! Thank you! Danke! Grazie!

CMLS 2020 @ ER 2020

November 3-6, 2020 in Vienna, Austria