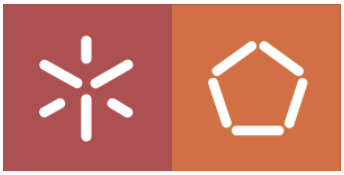

Universidade do Minho
Escola de Engenharia

Trabalho Prático

Introdução aos Algoritmos, à Programação e às Bases de Dados

Mestrado em Bioinformática
2022/2023

Ana Lisboa PG49831
Bruna Almeida PG51187
Luciana Martins PG49839

Índice

1. Introdução

1.1 Contextualização, motivação e objetivos

Os bancos de dados biológicos acarretam um papel central na bioinformática, uma vez que proporcionam aos cientistas o acesso a uma ampla e vasta variedade de dados biologicamente relevantes, incluindo sequências genómicas de uma gama cada vez mais abrangente de organismos e as suas informações biológicas associadas. Este armazenamento de dados torna-se necessário, uma vez que, é notável um aumento de volume gerado nos últimos anos e, sobretudo, da necessidade de tratamento de informações através de pesquisas, processamentos e análises de resultados. Assim, os bancos de dados biológicos utilizam aplicações e diversos sistemas de armazenamento de forma a ser mais acessível e funcional as informações relativas às sequências e respectivas anotações.

O principal objetivo deste projeto foi selecionar organismos da nossa preferência, sem qualquer critério definido e retirar os respectivos ficheiros do GenBank para realizar a nossa base de dados.

Recolhemos ficheiros Genbank de cinco organismos, sendo estes o peixe balão (*Tetraodontidae*), um ácaro (*Tetranychus urticae*), uma bactéria de urso (*Mycoplasma sp. Bear*), o chimpanzé (*Pan troglodytes*) e a planta floco-de-neve (*Trevesia palmata*), para possibilitar posteriormente a busca por artigos relacionados e o armazenamento da informação nestes contida. Optamos por escolher organismos com características distintas, desde bactérias, plantas, mamíferos e peixes, aumentando a diversidade da nossa amostra.

2. Implementação

Para permitir a implementação da base de dados desejada, é necessário primeiro definir as entidades e seus atributos, bem como os relacionamentos entre as diferentes entidades. É importante destacar que cada entidade está representada por uma tabela com atributos em colunas e as linhas contém dados para cada item que corresponde ao referente atributo.

2.1 Identificação das identidades, atributos e relacionamentos

Como entidades temos "**Genbank informations**", "**PubMed Informations**" e "**Protein Informations**". Como o próprio nome indica, "**Genbank informations**" representa a informação contida nos ficheiros Genbank como a sequência de DNA "*dna sequence*", o tamanho da sequência "*length*", o organismo "*dnasource*", a definição "*definition*" e o código do PubMed correspondente "*pubmedcod*" . Neste caso, como chave primária não nula temos o *acession number* do locus "*locusid*", possibilitando o estabelecimento de relações com outras entidades. A identidade "**PubMed Informations**" tem como chave primária, não nula o código do PubMed "*pubmedcod*", tendo outros atributos como data de publicação "*publication_date*",

nome dos autores "*authors*", título "*title*", afiliação "*affiliation*", abstract "*abstract*", o link da publicação "*publication_link*" e *acession number* "*locusid*", sendo este último chave estrangeira não nula e única. Por fim a entidade "**Protein Informations**" tem como chave primária, não nula e única o *acession number* da proteina *"locus_protein"*, como chave estranegira, não nula e única o acession number do locus "*locus_dna*", o organismo "*source*", definição *"definition"*, a sequência da proteína "*protein_sequence*" e o seu respectivo tamanho "*length*". Criamos dois relacionamentos "*included"* e "*related to*" que associam as nossas entidades numa proporção de 1 para N.

2.2 Modelo conceptual

Através do software TerraER foi possível criar o modelo conceptual da nossa base de dados (Figura 1). No esquema estão representadas as entidades, os atributos e as suas respectivas relações, que ligam as nossas entidades numa proporção de 1 para N. Este esquema irá servir de base para os passos seguintes.

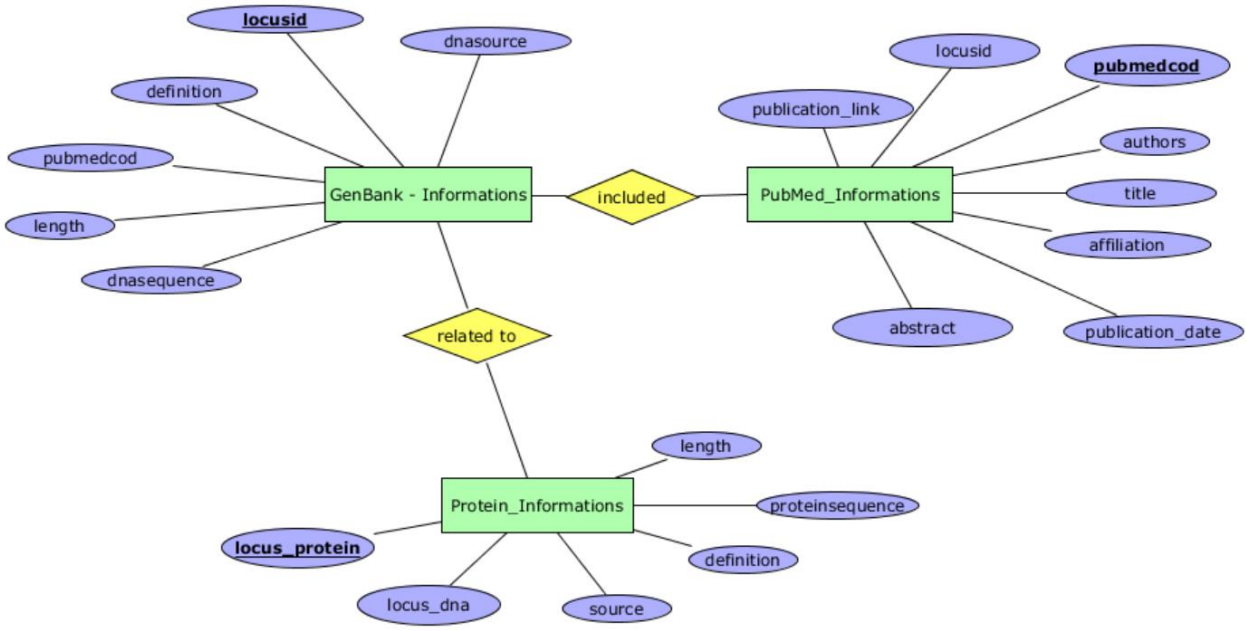

Figura 1- Esquema conceptual de uma base de dados aplicada às sequência dos 5 organismos e respetivas informações, realizado através do software TerraER.

2.3 Modelo lógico

Após a execução do modelo conceptual, e tendo este por base, utilizamos o MySQL Workbench para o desenvolvimento do modelo lógico (Figura 2).

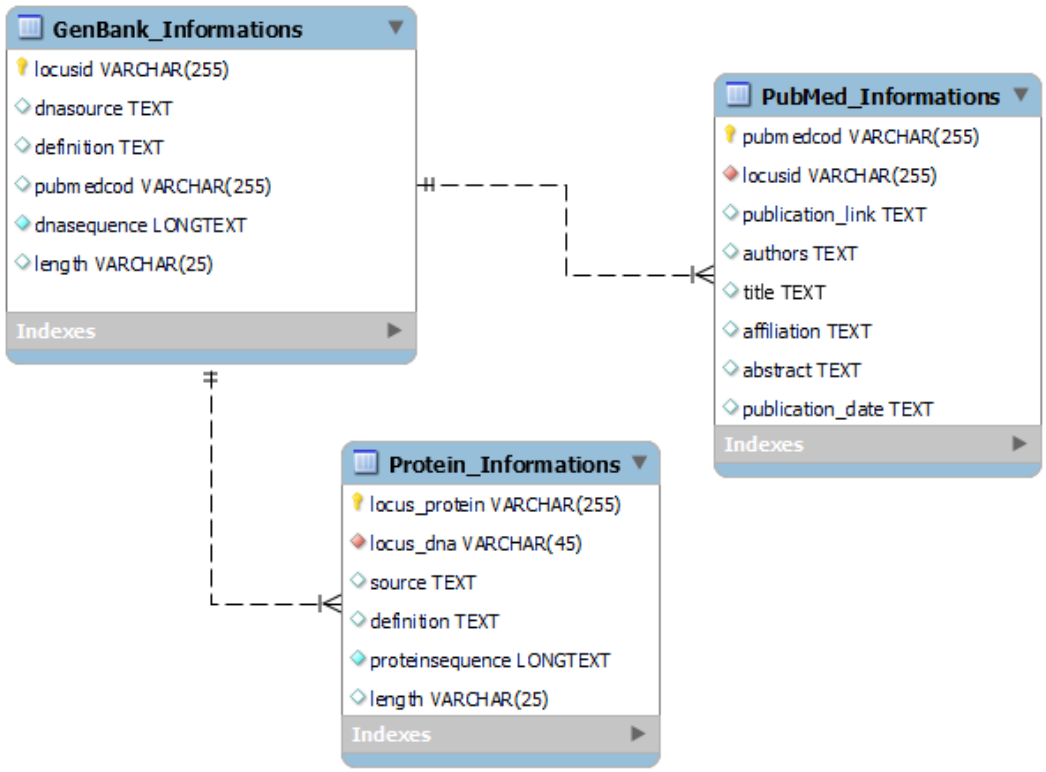

Figura 2- Esquema lógica da base de dados aplicada às sequências dos 5 organismos e respetivas informações, realizado através do MySQL Workbench.

2.4 Modelo físico

2.4.1 Povoação do modelo físico

Primeiramente, fizemos download dos ficheiros GenBank dos nossos organismos escolhidos. Utilizando o MySQL Workbench criamos uma conexão denominada "TrabalhoIAP" e de seguida criamos um novo esquema denominado da mesma forma, para conseguirmos através do módulo *mysql.connector* do Python, ligarmo-nos à base de dados. Com esta etapa concluída, conseguimos criar as nossas entidades enquanto tabelas através do Python ("GenBank_Informations", "PubMed_Informations" e "Protein_Informations"). Antes de iniciarmos a povoação das nossas tabelas, recorremos também ao *Entrez* do BioPython para conseguirmos retirar informações igualmente relevantes como o abstrato, a afiliação e a data de publicação apenas através do *acession number* de cada ficheiro GenBank. Posto isto, utilizando o módulo de expressões regulares, foi-nos possível aceder aos restantes dados

necessários, tais como, o *locus* e *locus da proteína*, *source*, *definition*, *pubmedcod*, *authors*, *title*, *dnasequence* e *protein_sequence*, *length* (de ambas). Para finalizar, povoamos as tabelas usando as funções *cursor*, *execute* e *commit*. Todo o código referente a esta explicação, encontra-se no GitHub partilhado.

locusid	dnasource	definition	pubmedcod	dnasequence	length
AB725596	Mycoplasma sp. Bear	Mycoplasma sp. Bear genes for 16S rRNA, 16S-...	23313325	gaccttggcttcggccttggttagtggcaaacgggtgagtaat...	2057 bp
AL954205	Pan troglodytes (chimpanzee)	Pan troglodytes chromosome 22 clone RP43-04...	15164055	gaattctcaaacctttcaagaagcacactttcttttttcattattt...	174216 bp
KF591508	chloroplast Trevesia palmata	Trevesia palmata voucher US:Jun Wen 5669 NA...	24184542	gttcctatgttaataggagtgggacttcttctttttccgacggca...	1891 bp
KY675904	mitochondrion Tetraodontidae sp. 1 JP-2017	Tetraodontidae sp. 1 JP-2017 isolate RSFL561 c...	28771590	ccttctcattcgggctgaactcagccaaccaggcgccctcctag...	561 bp
NC_010526	mitochondrion Tetranychus urticae (two-spotte...	Tetranychus urticae mitochondrion, complete g...	18408150	ataaaatgaattatatcaacaaatcataaaaatattggaacta...	13103 bp
NULL	NULL	NULL	NULL	NULL	NULL

Tabela 1 - Tabela obtida no MySQL através das funções do Python referente às informações do GenBank.

pubmedcod	locusid	publication_link	authors	title	affiliation
15164055	AL954205	https://pubmed.ncbi.nlm.nih.gov/15164055/	Watanabe,H., Fujiyama,A., Hattori,M., Taylor,...	DNA sequence and comparative analysis of chi...	RIKEN, Genomic Sciences Center, Yokohama 23...
18408150	NC_010526	https://pubmed.ncbi.nlm.nih.gov/18408150/	Van Leeuwen,T., Vanholme,B., Van Pottelberge...	Mitochondrial heteroplasmy and the evolution o...	Faculty of Bioscience Engineering, Ghent Univer...
23313325	AB725596	https://pubmed.ncbi.nlm.nih.gov/23313325/	Iso,T., Suzuki,J., Sasaoka,F., Sashida,H., Wat...	Hemotropic mycoplasma infection in wild black b...	Department of Veterinary Microbiology, School ...
24184542	KF591508	https://pubmed.ncbi.nlm.nih.gov/24184542/	Valcarcel,V., Fiz-Palacios,O. and Wen,J.	The origin of the early differentiation of Ivies (H...	Universidad Autonoma de Madrid, Campus Cant...
28771590	KY675904	https://pubmed.ncbi.nlm.nih.gov/28771590/	Isari,S., Pearman,J.K., Casas,L., Michell,C.T., ...	Exploring the larval fish community of the centr...	Red Sea Research Center, Biological and Enviro...
NULL	NULL	NULL	NULL	NULL	NULL

abstract	publication_date
Human-chimpanzee comparative genome resear...	2004 May 27
Genes encoded by mitochondrial DNA (mtDNA) ...	2008 Apr 22
This is the first report on Mycoplasma infection i...	2013 Apr 12
The Asian Palmate group is one of the four maj...	2014 Jan
An important aspect of population dynamics for ...	2017
NULL	NULL

Tabela 2 - Tabela obtida no MySQL através das funções do Python referente às informações do PubMed.

locus_dna	locus_protein	source	definition	proteinsequence	length
KF591508	AHB62690	chloroplast Trevesia palmata	NADH dehydrogenase subunit F, partial (chloro...	vpmligvgllfptatknirrmwafqsilllsivmifsinlsiqqinsssiy...	630 aa
AL954205	CAH18579	Pan troglodytes (chimpanzee)	human mRNA for KIAA0539 protein, partial [Pa...	fryqdhtfkktltavqllyspessvrtkiqlpvvyvmlmqhslflptl...	832 aa
AB725596	CDN41090	Paenibacillus sp. P22	hypothetical protein BN871_AB_00880 [Paeniba...	mrayspggmlivltsaprvskpltpsihrlrrglpgylilfaphafap...	217 aa
KY675904	KAB0391140	Balaenoptera physalus (Fin whale)	hypothetical protein E2I00_017264, partial [Bal...	vgysnvwyggghpfnhskfvvvsstvntspldvntsqimktis...	249 aa
NC_010526	YP_001795371	mitochondrion Tetranychus urticae (two-spotte...	cytochrome c oxidase subunit I (mitochondrion) ...	mkwimstnhknigtmyflfslfsglmgtsmsiiirlelmtpgsliqnd...	512 aa
NULL	NULL	NULL	NULL	NULL	NULL

Tabela 3- Tabela obtida no MySQL através das funções do Python referente às informações das proteínas associadas.

Posteriormente, dispondo da ferramenta *forward engineering*, foi possível exportar o modelo lógico para uma script. A script cria a base de dados, as tabela com as respectivas colunas e relações.

```
CREATE SCHEMA IF NOT EXISTS `TrabalhoIAP` DEFAULT CHARACTER SET utf8
USE `TrabalhoIAP` ;

-- -----------------------------------------------------
-- Table `TrabalhoIAP`.`GenBank_Informations`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TrabalhoIAP`.`GenBank_Informations` (
  `locusid` VARCHAR(255) NOT NULL,
  `dnasource` TEXT NULL,
  `definition` TEXT NULL,
  `pubmedcod` VARCHAR(255) NULL,
  `dnasequence` LONGTEXT NOT NULL,
  `length` VARCHAR(25) NULL,
  PRIMARY KEY (`locusid`),
  UNIQUE INDEX `locusid_UNIQUE` (`locusid` ASC) VISIBLE)
ENGINE = InnoDB;


-- Table `TrabalhoIAP`.`PubMed_Informations`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TrabalhoIAP`.`PubMed_Informations` (
  `pubmedcod` VARCHAR(255) NOT NULL,
  `locusid` VARCHAR(255) NOT NULL,
  `publication_link` TEXT NULL,
  `authors` TEXT NULL,
  `title` TEXT NULL,
  `affiliation` TEXT NULL,
  `abstract` TEXT NULL,
  `publication_date` TEXT NULL,
  PRIMARY KEY (`pubmedcod`),
  INDEX `FK1_idx` (`locusid` ASC) VISIBLE,
  UNIQUE INDEX `locusid_UNIQUE` (`locusid` ASC) VISIBLE,
  UNIQUE INDEX `pubmedcod_UNIQUE` (`pubmedcod` ASC) VISIBLE,
  CONSTRAINT `FK1`
    FOREIGN KEY (`locusid`)
    REFERENCES `TrabalhoIAP`.`GenBank_Informations` (`locusid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- Table `TrabalhoIAP`.`Protein_Informations`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `TrabalhoIAP`.`Protein_Informations` (
  `locus_protein` VARCHAR(255) NOT NULL,
  `locus_dna` VARCHAR(45) NOT NULL,
  `source` TEXT NULL,
  `definition` TEXT NULL,
  `proteinsequence` LONGTEXT NOT NULL,
  `length` VARCHAR(25) NULL,
  UNIQUE INDEX `locus_dna_UNIQUE` (`locus_protein` ASC) VISIBLE,
  PRIMARY KEY (`locus_protein`),
  UNIQUE INDEX `locus_dna_UNIQUE` (`locus_dna` ASC) VISIBLE,
  CONSTRAINT `FK2`
    FOREIGN KEY (`locus_dna`)
    REFERENCES `TrabalhoIAP`.`GenBank_Informations` (`locusid`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

3. Análise crítica e perspectivas futuras

Ao longo do desenvolvimento da base de dados tivemos mais dificuldades em estabelecer a ligação da base de dados com o código Python, uma vez que era uma situação completamente nova para nós, foi um desafio, no entanto, foi realizado com sucesso.

Uma ação que futuramente poderia ser realizada seria a adição de mais atributos que fossem considerados igualmente relevantes. Ao introduzir mais informação, a base de dados teria melhor qualidade e funções variadas, desta forma com uma melhor funcionalidade. De notar que, a nossa base de dados e respetivo código podia estar mais otimizado, mais complexo e podíamos ter explorado a criação de funções para trabalhar com o SQL.