

PPCIC -Programa de Pós-graduação em Ciência da Computação

Anotação funcional das proteínas da bactéria *Pseudomonas aeruginosa* CCBH4851

Aluno: Ribamar Matias

E-mail: ribamar.matias@eic.cefet-rj.br

Orientadora: Kele Belloze



Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
CEFET/RJ

Objetivo

- Criação de uma base de dados que amplie o conhecimento sobre a bactéria *Pseudomonas aeruginosa* CCBH4851
 - Anotação funcional de proteínas baseada em ontologia
 - Análise comparativa com modelos de referência
 - Levantamento de essencialidade de proteínas

Importância do estudo da espécie *P. aeruginosa*

- São praticamente onipresentes
- Possuem grande versatilidade metabólica e fisiológica
- Consideradas como uma das três principais causas de infecções humanas oportunistas
- Citada no relatório mais recente da Organização Mundial de Saúde
- Alvo prioritário para pesquisas orientadas ao desenvolvimento de novos antibióticos



WHO PRIORITY PATHOGENS LIST FOR R&D OF NEW ANTIBIOTICS

Priority 1: CRITICAL[#]

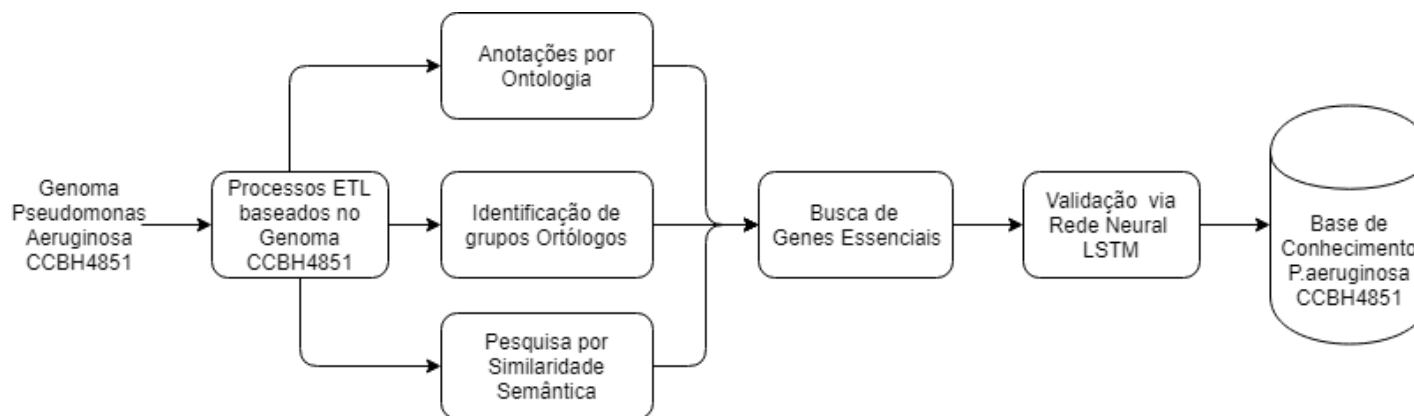
Acinetobacter baumannii, carbapenem-resistant

Pseudomonas aeruginosa, carbapenem-resistant

*Enterobacteriaceae**, carbapenem-resistant, 3rd generation cephalosporin-resistant

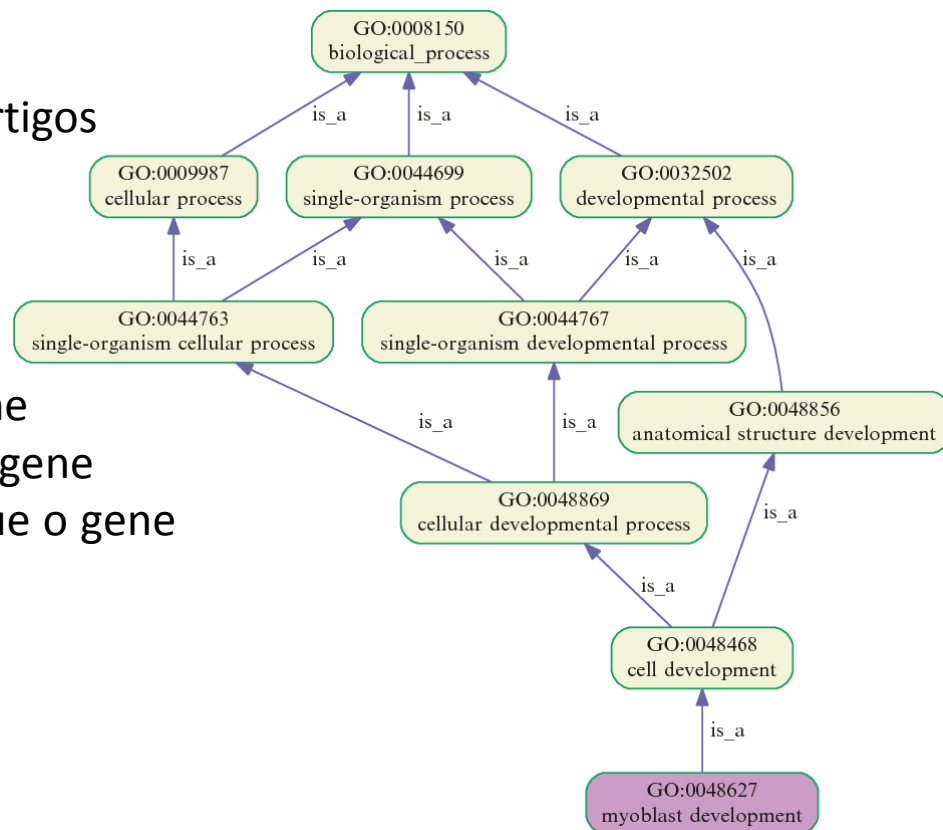
Metodologia

- Tarefas restritas ao escopo de processos computacionais
- Pré-processamento para extração de proteínas do genoma publicado pela Fiocruz
- Anotação baseada em ontologia
- Similaridade semântica entre modelos
- Identificação de grupos ortólogos
- Predição de essencialidade por métodos de Aprendizado de Máquina



Anotação baseada em ontologia

- Gene Ontology
- Apoiada nos resultados de mais 140.000 artigos
- 600.000 anotações experimentais
- Estrutura definida como grafo
- Cada nó é um termo
- Divide-se em três aspectos
 - Função Molecular: atividades do gene
 - Componente Celular: localização do gene
 - Processo Biológico: processos em que o gene participa



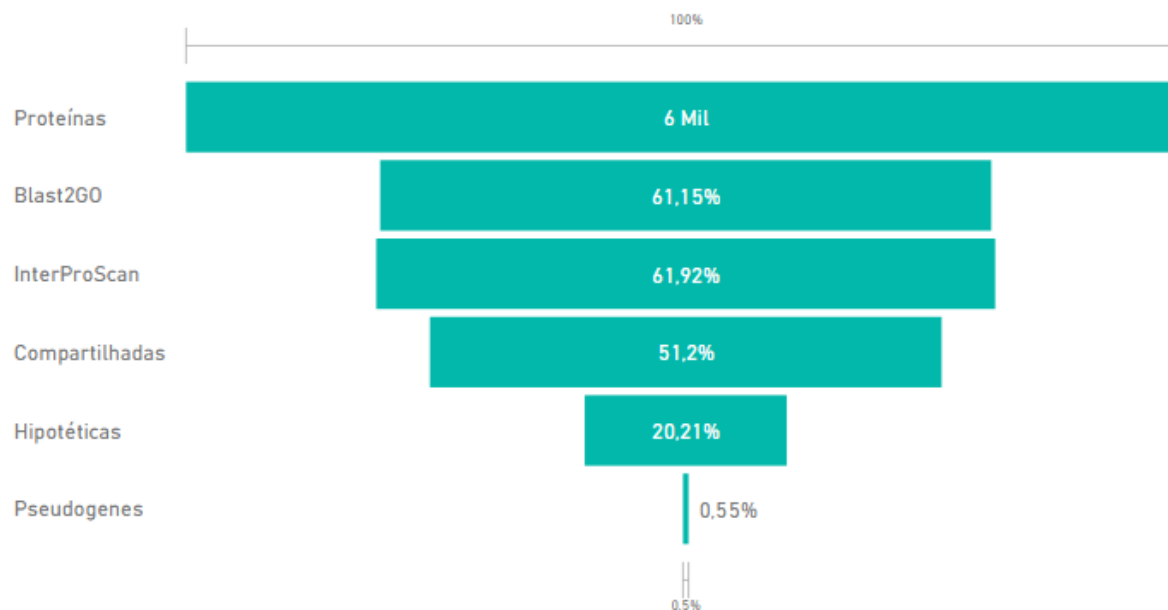
Anotação baseada em ontologia

- Ferramentas Blast2GO e InterProScan
- Blast2GO é comercial, mas oferece versão básica sem custo
- InterProScan é de uso livre
- Blast2GO é citada em 6245 resultados no PUBMED NCBI
- InterProScan é citada em 5504 resultados no PUBMED NCBI
- Ambas usam o proteoma como entrada de dados
- Resultado do processamento são termos anotados via Gene Ontology



Anotação *P.aeruginosa* CCBH4851 - resultados globais

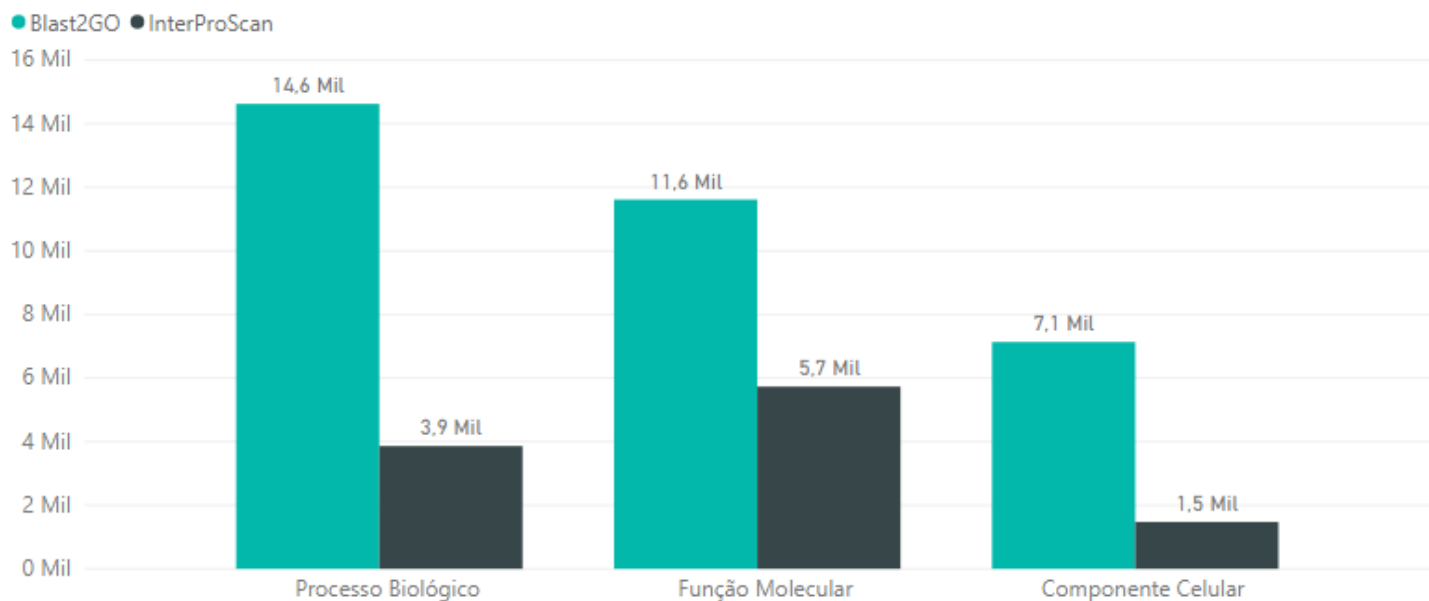
- Dados disponíveis - Github <https://github.com/rjmatias/cefet-rj>
- 6211 proteínas
- Mais de 60% anotadas
- Cerca de 21% das proteínas descritas como hipotéticas (não anotadas)
- Ferramentas compartilharam anotações



Anotação *P.aeruginosa* CCBH4851 - aspectos Gene Ontology

- Blast2GO detalhou maior número de aspectos
- Processos Biológicos (PB) e Funções Moleculares (FM) com maior representatividade
- PB são os maiores processos, executados por múltiplas atividades moleculares
- FM são atividades que ocorrem no interior da célula, como catálise ou transporte

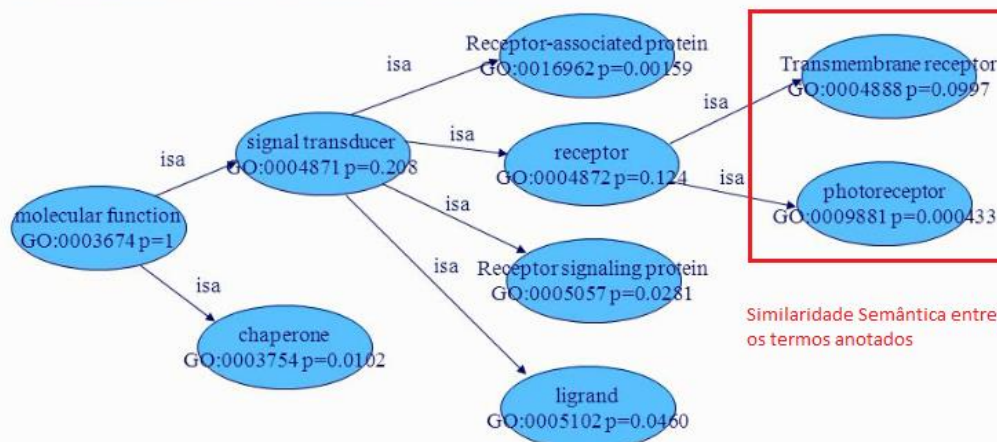
Termos Identificados via Gene Ontology



Similaridade semântica entre termos (SS)

- Estrutura em Grafo GO permite inferência por SS
- SS avalia semelhança no significado de dois conceitos
- Quanto mais próximos dois termos estão, mais semelhante é seu significado
- Permite medir similaridade funcional entre duas proteínas
- SS pode ser definida como uma função
- Dadas duas ontologias, termos ou conjuntos, a função retorna um %
- Este % reflete a proximidade no significado entre eles

Gene Ontology

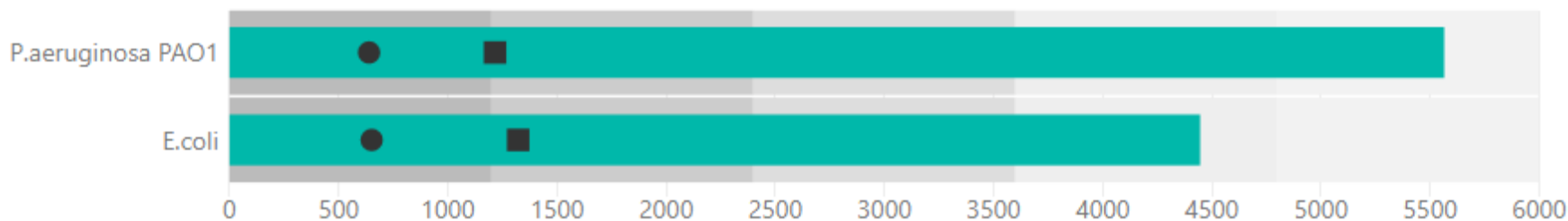


Similaridade semântica entre Modelos e CCBH4851

- Organismos *E.coli* e *P.aeruginosa* PAO1
- Grafo da GO permite a comparação de termos por Similaridade Semântica.
 - Processamento BLAST entre as proteínas da *P. aeruginosa* CCBH4851 e organismos modelo (proteomas obtidos do UniProtKB)
 - Seleção das proteínas com maior similaridade no processamento BLAST
 - Execução da ferramenta GOGO para obter os indicadores de similaridade semântica
- Similaridade Semântica Funcional de aproximadamente 1300 proteínas por modelo

Indicadores de Similaridade Semântica por Total de Proteínas Anotadas via Gene Ontology

● Proteínas ● sim.sem.>0.75% ■ 0.5% < sim.sem. <= 0.75%

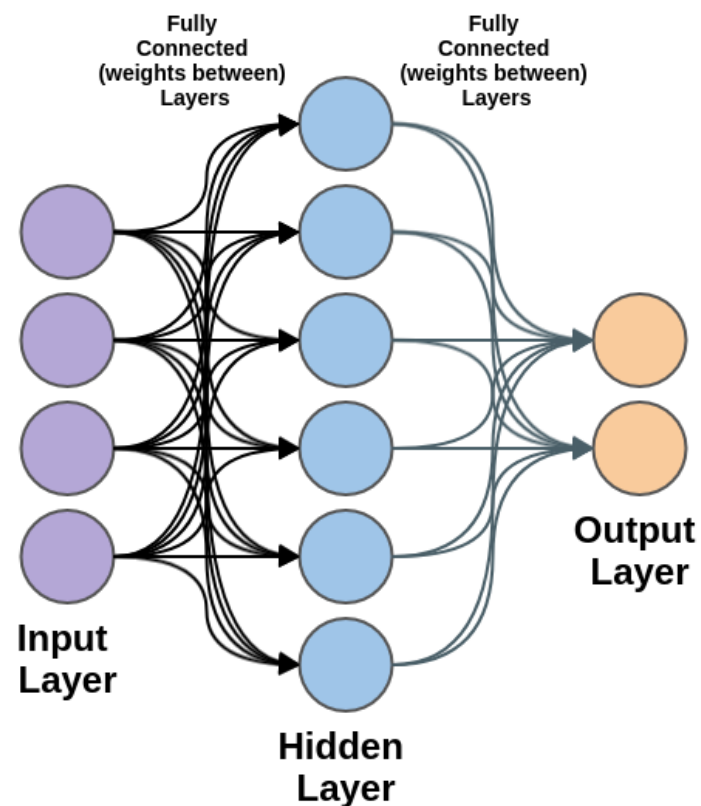


Em desenvolvimento

- Correlações entre os termos anotados entre CCBH4851 e modelos
- Identificação de grupos ortólogos
 - OrthoFinder
- Predição de proteínas essenciais por meio de Aprendizado de Máquina

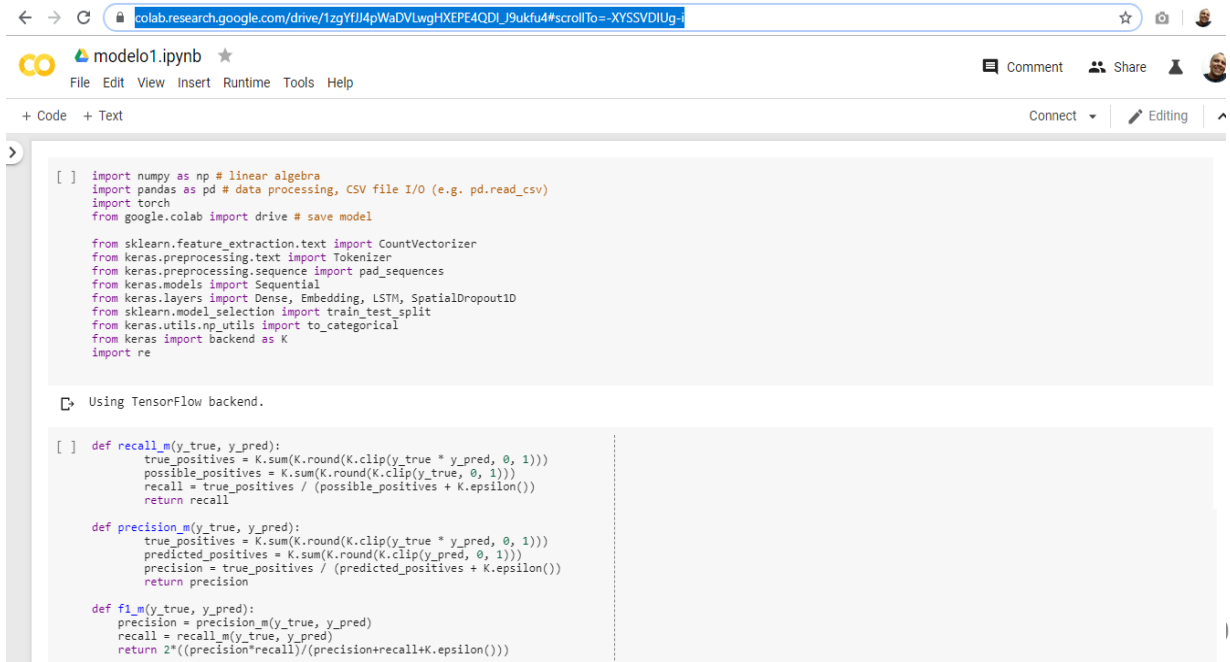
Redes Neurais

- Redes Neurais existem desde a década de 50
- Usadas em diversos cenários
- Detecção de fraudes, reconhecimento de imagens, marketing direcionado, previsões financeiras, análise textual, e muitos outros
- Podem ser aplicadas em Bioinformática
- Tecnologia em constante aprimoramento



Rede Neural LSTM (*Long Short-Term Memory*)

- Modelo escrito em Python + Keras + Tensorflow
- Treinada a partir dos proteomas do organismos modelos *E.coli* e *P. aeruginosa* PAO1
- Disponível via plataforma colaborativa Google Collab
- Proposta é treinar um modelo capaz de inferir essencialidade
- Foco em classificação binária



```

[ ] import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import torch
from google.colab import drive # save model

from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras import backend as K
import re

[ ] Using TensorFlow backend.

[ ] def recall_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

def precision_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision

def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))

```

Obrigado!

Ribamar Matias

ribamar.matias@eic-cefet.br

Orientadora: Kele Belloze

Kele.belloze@cefet-rj.br

