



Laboratórios de Bioinformática

Docentes: Pedro Ferreira e Miriam Santos
1 Ano, 2 Semestre

Grupo C:
Adriana Oliveira, 202307136
Lara Gonçalves, 202307857
Leonor Ribeiro, 202305139



Motivação

- Adquirir novos conhecimentos que podem levar a avanços nas áreas da genética.
- Compreender mecanismos fundamentais biológicos



Objetivo

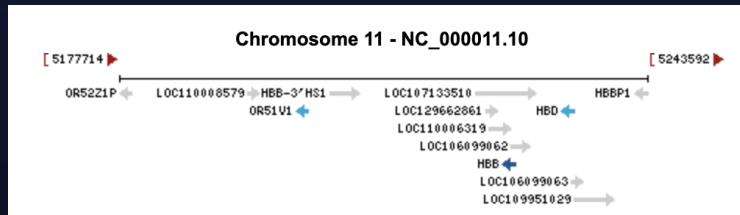
- Caracterizar o gene e a proteína
- Compreender conceitos de homologia de proteínas, a semelhança de sequências e árvores filogenéticas
- Ganhar experiência na utilização de certas ferramentas de bioinformática
- Aprofundar conhecimentos lecionados nas aulas



01 Caracterização do Gene HBB

Gene HBB

- Este gene encontra-se no cromossoma 11, na posição 11p15.4., do organismo Homo Sapiens. Contém 3 exões. O ID deste gene é: 3043
- O gene HBB (beta-globina) é regulado por uma série de elementos regulatórios que garantem a expressão precisa e controlada da beta-hemoglobina.



Fonte da Imagem: NCBI



Elementos Regulatórios



Fonte da Imagem: Genome Browser

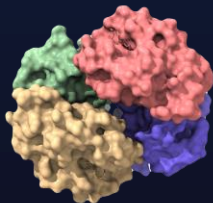




01 Caracterização da Proteína

Proteína

- (NP_000509), que contém 147 aminoácidos.
- É altamente expressa na medula óssea
- Envolvidas no transporte de oxigénio e dióxido de carbono pelo organismo
- Regulação e manutenção dos vasos sanguíneos e da circulação do sangue.



Estrutura 3D da proteína
(HBB)

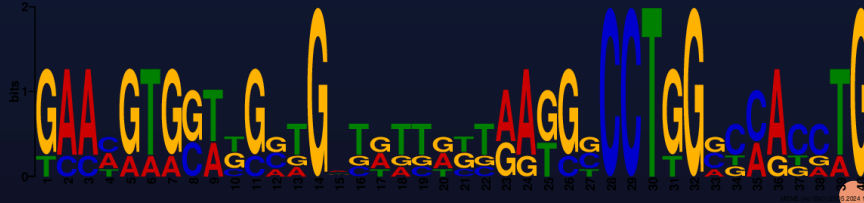


Doenças Associadas

- Beta Talassemia (B-THAL)
- Doença Falciforme
- Anemia de Heinz

Motif

- Sequência curta partilhada por diferentes sequências, de outros organismos.



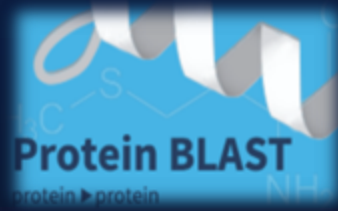
Fonte da Figura: The MEME suite



02 Sequências Homólogas

Com a ajuda do software BLASTP, encontramos 10 sequências homólogas da proteína HBB ao organismo Homo Sapiens:

- Gorilla gorilla gorilla
- Pongo Abeli
- Trachypithecus francoisi
- Nomascus leucogenys
- Chlorocebus Sabaeus
- Rhinopithecus roxellana
- Erythrocebus patas
- Miopithecus talapoin
- Cercocebus atys
- Sapajus Apella



- A homologia refere-se à similaridade de sequência que sugere uma origem evolutiva comum.

- Pequenas variações indicam adaptações específicas ou diversidade genética.



1

Alinhamento



Sequence Alignment

2

Árvore Filogenética



CLUSTAL O(1.2.4) multiple sequence alignment

```

XP_032096415.1:1-146      MVHLTAEKSAVTTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSTPDVAMGNPK 60
NP_000509.1               MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSTPDVAMGNPK 60
XP_018891709.1:1-146      MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSTPDVAMGNPK 60
XP_002822173.1           MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSTPDVAMGNPK 60
XP_004090697.3           MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSTPDVAMGNPK 60
NP_001292888.1:1-146      MVHLTPEEKNAVTTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSSPDVAMGNPK 60
AY089367.1:1-146         MVHLTPEEKNAVTTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSSPDVAMGNPK 60
NP_001316847.1:1-146      MVHLTPEEKTAVTTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSSPDVAMGNPK 60
AY089363.1:1-146         MVHLTPEEKTAVTTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSSPDVAMGNPK 60
XP_033062959.1:1-146      MVHLTPEEKAATVTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSSPDVAMGNPK 60
XP_010361646.1:1-146      MVHLTPDEKAATVTLWGKVNVDVGGGALGRLLVVYPWTQRFESFGDLSSPDVAMGNAK 60

***** :** ***:*****:*****:*****:*****:*****:*****:

```



```

XP_032096415.1:1-146      VKAHGKKVLGAFSDGLTHLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
NP_000509.1               VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
XP_018891709.1:1-146      VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
XP_002822173.1           VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
XP_004090697.3           VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
NP_001292888.1:1-146      VKAHGKKVLGAFSDGLNHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
AY089367.1:1-146         VKAHGKKVLGAFSDGLTHLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
NP_001316847.1:1-146      VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
AY089363.1:1-146         VKAHGKKVLGAFSDGLTHLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
XP_033062959.1:1-146      VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120
XP_010361646.1:1-146      VKAHGKKVLGAFSDGLAHLNDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFG 120

*****:*****:*****:*****:*****:*****:*****:*****:

```



```

XP_032096415.1:1-146      KEFTPQVQAAYQKVAVGANALAHKY- 146
NP_000509.1               KEFTPQVQAAYQKVAVGANALAHKY- 146
XP_018891709.1:1-146      KEFTPQVQAAYQKVAVGANALAHKY- 146
XP_002822173.1           KEFTPQVQAAYQKVAVGANALAHKYH 147
XP_004090697.3           KEFTPQVQAAYQKVAVGANALAHKYH 147
NP_001292888.1:1-146      KEFTPQVQAAYQKVAVGANALAHKY- 146
AY089367.1:1-146         KEFTPQVQAAYQKVAVGANALAHKY- 146
NP_001316847.1:1-146      KEFTPQVQAAYQKVAVGANALAHKY- 146
AY089363.1:1-146         KEFTPQVQAAYQKVAVGANALAHKY- 146
XP_033062959.1:1-146      KEFTPQVQAAYQKVAVGANALAHKY- 146
XP_010361646.1:1-146      KEFTPQVQAAYQKVAVGANALAHKY- 146

*****:*****:*****:*****:*****:*****:*****:*****:

```

COLOR SCHEME

clustal2

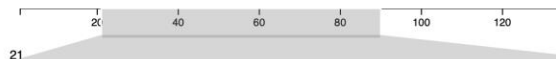
LEGEND

ARNDCEGHILKMFPSTWYVBXZ



10 sequences

XP_032096415.1:1-146
 XP_018891709.1:1-146
 XP_004090697.3
 NP_002822173.1
 NP_001292888.1:1-146
 AY089367.1:1-146
 NP_001316847.1:1-146
 AY089363.1:1-146
 XP_033062959.1:1-146
 XP_010361646.1:1-146



```

VDEVGGEALGRLLVVYPWTQRFESFGDLSTPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSTPDVAMGNPKVAHKKVVGAFSDGLAHLNDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSTPDVAMGNPKVAHKKVVGAFSDGLAHLNDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSTPDVAMGNPKVAHKKVVGAFSDGLAHLNDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSSPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSSPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSSPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSSPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSSPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS
VDEVGGEALGRLLVVYPWTQRFESFGDLSSPDVAMGNPKVAHKKVVGAFSDGLTHLDNLKGTFAQLS

```

- (*) - resíduos idênticos;
- (:) - resíduos similares, mesmo aminoácido, mas com propriedades químicas diferentes;
- () - resíduos todos diferentes;
- (.) - resíduos parcialmente semelhantes, havendo semelhanças biológicas e funcionais.

Fontes das Figuras: Clustal Omega

Análise da árvore filogenética

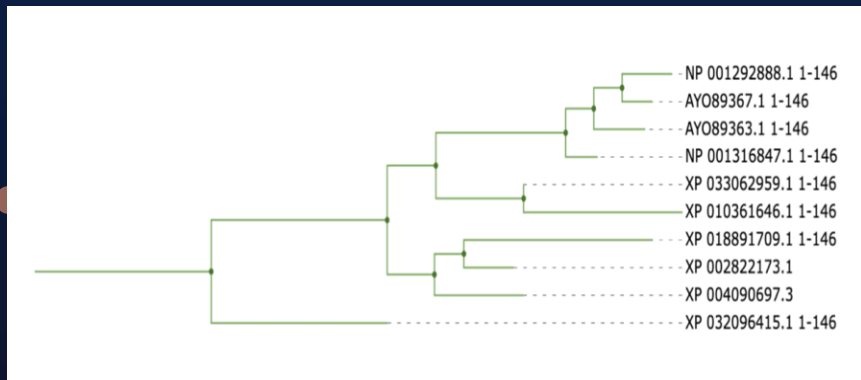


Imagem obtida através do iTOL

**Erythrocebus patas
e Miopithecus
talopoin**

0.68%

AYO89367.1 e AYO89363.1
mostra menor divergência entre eles

Sapajus Apella

~3%

XP_032096415.1
mostra maior
divergência



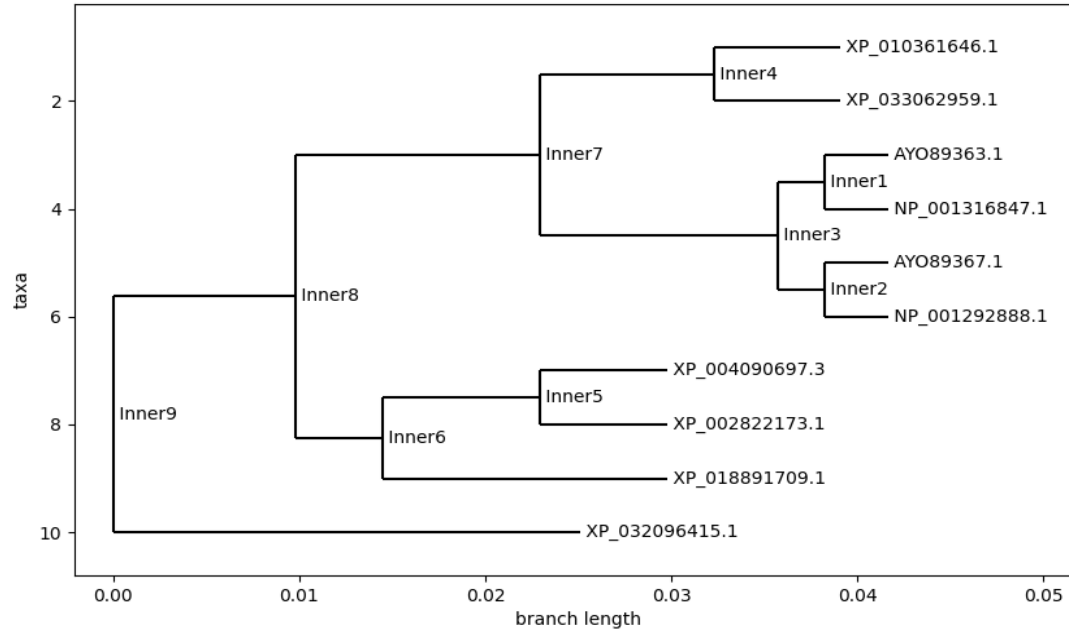
Website

Para a realização deste site foi utilizado o Notepad++.

Além da linguagem html (como pedido), utilizou-se também a linguagem css, que incidiu na parte estrutural e de design do site.



Automatização da Análise



from Bio.Align.py

Users > lmr > Dropbox > bioinf > trabalho grupo > from Bio.Align.py > ...

```
1 import random
2 import matplotlib.pyplot as plt
3 from Bio import AlignIO, SeqIO
4 from Bio.SeqRecord import SeqRecord
5 from Bio.Seq import Seq
6 import numpy as np
7 import subprocess
8
9 # Sequências fornecidas
10 sequences = [
11     SeqRecord(Seq("MVHLTAEKSAVTTLWGKVNVDVGGGALGRLLVVPWTRQFFDSFGDLSTPDAMNPNPKVKAHGKKVLGAFSDGLTHLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
12     SeqRecord(Seq("MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSTPDAMGNPNPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDKLHVDPENFKLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
13     SeqRecord(Seq("MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSTPDAMGNPNPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
14     SeqRecord(Seq("MVHLTPEEKSAVTALWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSTPDAMGNPNPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
15     SeqRecord(Seq("MVHLTPEEKNAVTTLWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSSPDAMGNPNPKVKAHGKKVLGAFSDGLNLDNLKGTFAQLSELHCDKLHVDPENFKLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
16     SeqRecord(Seq("MVHLTPEEKNAVTTLWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSSPDAMGNPNPKVKAHGKKVLGAFSDGLTHLDNLKGTFAQLSELHCDKLHVDPENFKLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
17     SeqRecord(Seq("MVHLTPEEKTAVTTLWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSSPDAMGNPNPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDKLHVDPENFKLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
18     SeqRecord(Seq("MVHLTPEEKTAVTTLWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSSPDAMGNPNPKVKAHGKKVLGAFSDGLTHLDNLKGTFAQLSELHCDKLHVDPENFKLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
19     SeqRecord(Seq("MVHLTPEEKAAVTALWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSSPDAMGNPNPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
20     SeqRecord(Seq("MVHLTPDEKAAVTALWGKVNVDVGGGALGRLLVVPWTRQFFESFGDLSSPDAMGNAKPKVKAHGKKVLGAFSDGLAHLNLDNLKGTFAQLSELHCDKLHVDPENFRLLGNVLCVLAHHFGKEFTPPVQAAAYQKVVAGVANALAHKY
21 ]
22
23 # Escrevendo as sequências em um arquivo fasta temporário
24 with open("temp_sequences.fasta", "w") as f:
25     SeqIO.write(sequences, f, "fasta")
26
27 # Executando o Clustal Omega para alinhamento via subprocess
28 subprocess.run(["/Users/lmr/Downloads/clustalo/clustal-omega-1.2.3-macosx", "-i", "temp_sequences.fasta", "-o", "aligned.fasta", "--auto", "-v"])
29
30 # Lendo o arquivo de alinhamento resultante
31 alignment = AlignIO.read("aligned.fasta", "fasta")
32
33 Codumate: Options | Test this function
34 def generate_random_color(colors_in_use):
35     color = "#" + "".join([random.choice('0123456789ABCDEF') for _ in range(6)])
36     while color in colors_in_use:
37         color = "#" + "".join([random.choice('0123456789ABCDEF') for _ in range(6)])
38     return color
```

```

39 def plot_alignment(alignment):
40     # Atributos de estilo
41     unique_bases = set(''.join([str(record.seq) for record in alignment]))
42     colors_in_use = {}
43     base_colors = {base: generate_random_color(colors_in_use) for base in unique_bases}
44
45     fig, ax = plt.subplots(figsize=(10, 6))
46
47     # Plotagem do alinhamento
48     for i, record in enumerate(alignment):
49         sequence = str(record.seq)
50         y = len(alignment) - i - 1
51         for j, residue in enumerate(sequence):
52             x = j
53             color = base_colors[residue]
54             if i > 0: # Ignora a primeira linha (linha de referência)
55                 ref_residue = str(alignment[0].seq)[j]
56                 if residue != ref_residue:
57                     color = 'red'
58             ax.text(x, y, residue, ha='center', va='center', color=color, fontsize=8)
59
60     # Ajustes estéticos
61     ax.set_xlim(-1, len(alignment[0]))
62     ax.set_ylim(-1, len(alignment))
63     ax.set_xticks(np.arange(len(alignment[0].seq)))
64     ax.set_yticks(np.arange(len(alignment)))
65     ax.set_xticklabels(range(1, len(alignment[0].seq) + 1), fontsize=5, rotation=90)
66     ax.set_yticklabels([record.id for record in alignment], fontsize=8)
67     ax.grid(color='gray', linestyle='-', linewidth=0.5)
68     ax.set_title('Multiple Sequence Alignment', fontsize=14)
69     ax.set_xlabel('Position', fontsize=12)
70     ax.set_ylabel('Sequence', fontsize=12)
71
72     # Criando a legenda
73     legend_elements = [plt.Line2D([0], [0], marker='w', markerfacecolor=color, markersize=10, label=residue) for residue, color in base_colors.items()]
74     legend_elements.append(plt.Line2D([0], [0], marker='w', markerfacecolor='red', markersize=10, label='Different'))
75     ax.legend(handles=legend_elements, loc='upper right', title='Amino Acids', fontsize=8)
76

```

from Bio.Align.py

Users > lmr > Dropbox > bioinf > trabalho grupo > from Bio.Align.py > ...

```
39 def plot_alignment(alignment):
68     ax.set_title('Multiple Sequence Alignment', fontsize=14)
69     ax.set_xlabel('Position', fontsize=12)
70     ax.set_ylabel('Sequence', fontsize=12)
71
72     # Criando a legenda
73     legend_elements = [plt.Line2D([0], [0], marker='o', color='w', markerfacecolor=color, markersize=10, label=residue) for residue, color in base_colors.items()]
74     legend_elements.append(plt.Line2D([0], [0], color='w', markerfacecolor='red', markersize=10, label='Different'))
75     ax.legend(handles=legend_elements, loc='upper right', title='Amino Acids', fontsize=8)
76
77     plt.tight_layout()
78     plt.show()
79
80 plot_alignment(alignment)
81
82 from Bio import AlignIO
83 from Bio.Phylo.TreeConstruction import DistanceCalculator, DistanceTreeConstructor
84 from Bio.Phylo import draw
85 import matplotlib.pyplot as plt
86
87 # Lendo o arquivo de alinhamento
88 alignment = AlignIO.read("aligned.fasta", "fasta")
89
90 # Calculando as distâncias entre as sequências no alinhamento
91 calculator = DistanceCalculator('identity')
92 dm = calculator.get_distance(alignment)
93
94 # Construindo a árvore filogenética usando UPGMA
95 constructor = DistanceTreeConstructor()
96 tree = constructor.upgma(dm)
97
98 # Desenhando a árvore
99 plt.figure(figsize=(10, 6))
100 draw(tree)
101 plt.title("Phylogenetic Tree")
102 plt.show()
```

Interações do Gene

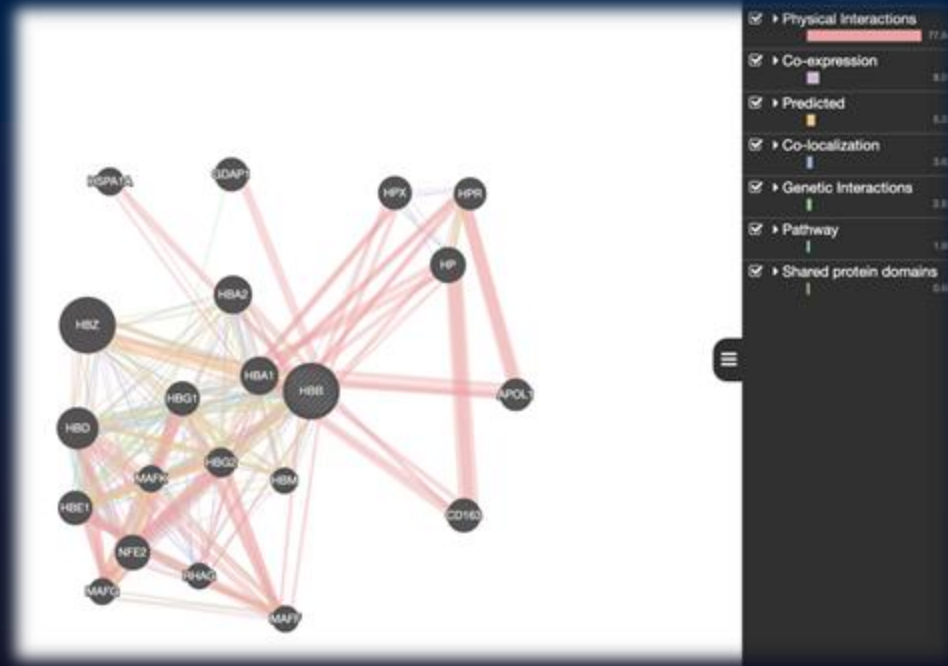


Imagem obtida através do Genemania

