BIOINFORMÁTICA ESTRUTURAL

Modelagem computacional de proteínas

By **Laboratório de Bioinformática e Sistemas**

11 de julho de 2021

Modelagem computacional de proteínas

Letícia Xavier Silva, Luana Luiza Bastos, Lucianna Helene Santos

Revisão: Diego Mariano

BIOINFO – Revista Brasileira de Bioinformática. Edição #01. Julho, 2021.
DOI: 10.51780/978-6-599-275326-08

As proteínas são as macromoléculas mais abundantes e cada célula de um ser vivo pode conter milhares de proteínas, cada uma com uma função única. A função de uma proteína é definida pelo arranjo dos átomos, presentes na sequência de aminoácidos, em sua estrutura tridimensional [1]. A relação arranjo tridimensional e função pode, por exemplo, depender da posição dos resíduos catalíticos no sítio ativo da proteína, ou uma possível resposta conformacional ao interagir com outras moléculas, entre outros fatores. Com isso, a determinação da estrutura proteica fornece uma melhor compreensão do funcionamento da proteína, permitindo criar proposições sobre como afetá-la, controlá-la ou modificá-la. Por exemplo, com a estrutura podemos projetar mutações pontuais em uma região da proteína com a intenção de alterar a função ou tentar prever moléculas que possivelmente se ligam a ela.

Todas as estruturas tridimensionais de macromoleculares são modelos, com níveis variáveis entre dados experimentais e predição computacional [2]. Geralmente, para se obter as coordenadas atômicas de átomos pesados com uma certa precisão são necessárias técnicas experimentais, como a cristalografia de Raios-X, Ressonância Magnética Nuclear (RMN) e Crio Microscopia Eletrônica (cryo-EM) [3,4]. Os dados oriundos dessas técnicas dependem em sua maioria de ferramentas computacionais para a interpretação espacial dos dados, construção e refinamento dos modelos [2]. Apesar da confiabilidade dos modelos estruturais gerados por técnicas experimentais, resolver estruturas usando essas técnicas requer treinamento extremamente especializado, um alto grau de habilidade, um bom orçamento, e o alvo molecular expresso e purificado em grande quantidade.

Considerando a taxa em que novas sequências de proteínas são descobertas, a dificuldade de resolver uma estrutura experimental, com as tecnologias disponíveis atualmente, é evidente. Embora o número de estruturas tridimensionais esteja crescendo continuamente, o banco de dados de proteínas, *Protein Data Bank* (PDB) [5], possui cerca de 175.000 estruturas resolvidas atualmente (março/21), uma grande lacuna entre estruturas e sequências disponíveis (Figura 1) ainda persiste. Isso se observa

no número de sequências disponíveis no UniProt [6], que é 1200 vezes maior que o número de estruturas tridimensionais disponíveis. Portanto, comparando os dois conjuntos estamos provavelmente perdendo importantes informações biológicas e biofísicas, já que nem todas as novas proteínas sendo identificadas e sequenciadas tem sua estrutura tridimensional elucidada [2]. Nesse sentido, a predição computacional (*in silico*) da estrutura tridimensional de proteínas se torna uma alternativa à medida que essa lacuna cresce [7].

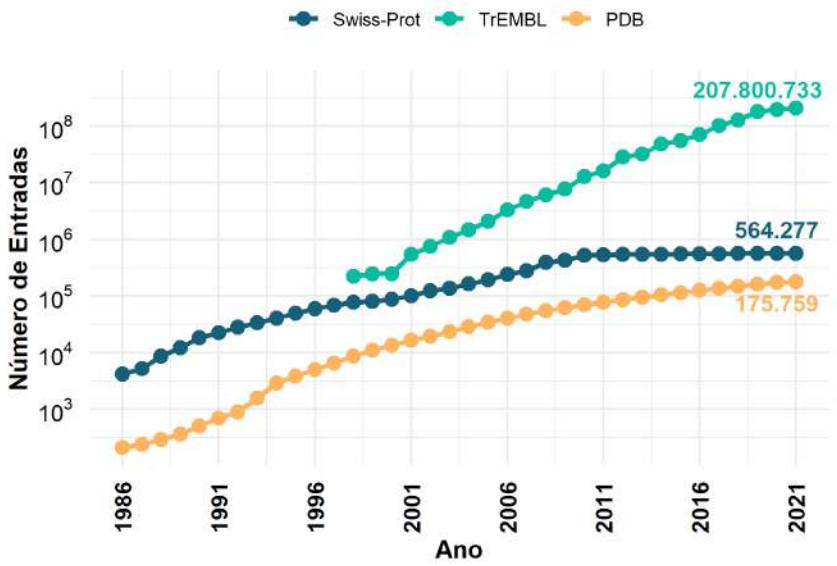


Figura 1. Crescimento do número de sequências de proteínas e de estruturas tridimensionais ao longo do tempo em bases de dados específicas. Swiss-Prot e TrEMBL são bases de dados de sequências e fazem parte do UniProt [6]. Porém, Swiss-Prot contém apenas sequências manualmente anotadas, enquanto o TrEMBL compreende as sequências automaticamente anotadas. Como a diferença no número de entradas entre TrEMBL, Swiss-Prot e PDB [5] é muito significativa, a escala logarítmica foi usada para aproximar a visualização no gráfico. Os dados foram obtidos em março de 2021.

A partir dos métodos de predição computacional é possível obter informações estruturais utilizando a sequência de aminoácidos de uma proteína cuja estrutura não foi determinada experimentalmente. No passado esse tipo de predição era visto como um desafio, porém, com o progresso dos algoritmos computacionais ao longo dos anos e uma disponibilidade maior de enovelamentos proteicos conhecidos, se tornou funcional com previsões plausíveis e razoavelmente precisas em muitos casos [8]. As técnicas de predição de estrutura computacionais são classificadas em dois grupos: técnicas baseadas em estruturas tridimensionais conhecidas e técnicas independentes de estruturas conhecidas. Com uma estrutura conhecida, o espaço de busca por uma nova proteína é diminuído, pois a exploração se dá por modificação da estrutura (chamada de molde ou *template*) tridimensional resolvida por métodos experimentais [9]. Dentro desse grupo se encontram as abordagens por modelagem comparativa e por *threading*. Para as técnicas independentes de um molde, informações estruturais são obtidas através de vários fragmentos ou da predição de

estrutura secundária de proteínas não relacionadas a proteína que se quer modelar. Nesse grupo se encontra as abordagens *ab initio* e *de novo* [10].

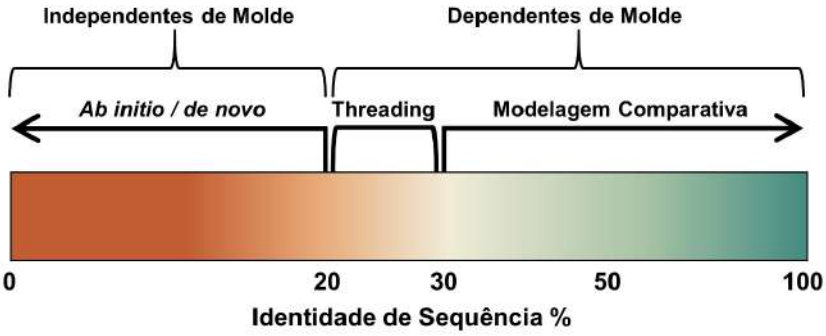


Figura 2. Escala entre métodos de predição de estrutura tridimensional de proteínas e identidade de sequência com as estruturas existentes. Para cada técnica um certo grau de similaridade é necessário, medido pela taxa de identidade entre a sequência alvo e sequências de estruturas conhecidas (a serem usadas como moldes).

Consequentemente, a escolha da metodologia de predição computacional a ser utilizada está condicionada a disponibilidade de estruturas tridimensionais, e a taxa de semelhança entre a sequência e uma estrutura do PDB (Figura 2). A semelhança entre molde e estrutura a ser modelada pode ser determinada pelo alinhamento de sequências, onde se obtém os valores de similaridade, identidade e cobertura entre elas. Por exemplo, abordagens de modelagem comparativa funcionam bem para proteínas com pelo menos 70% de identidade entre as sequências. Aproximando-se de 50%, a seleção de modelos torna-se mais difícil. Próximo dos 30%, ou a "*twilight-zone*", torna-se extremamente difícil, porque quaisquer dois pares aleatórios de proteínas podem ter esse nível de identidade de sequência.

Os métodos de predição estrutural computacional também possuem limitações que devem ser atenciosamente avaliadas para entender o grau de confiança depositada nos modelos [9]. Para modelos baseados em moldes, podemos dizer que as estruturas resultantes terão qualidade comparável com as estruturas experimentais utilizadas ou pior. Dependendo das métricas de confiança, avaliadas por ferramentas de validação, os modelos podem ser utilizados em conjunto com outros métodos, tais como dinâmica molecular e atracamento molecular. Porém, existe um interesse contínuo dos pesquisadores em melhorar a predição de estruturas tridimensionais. Esse interesse pode ser visto na competição bienal chamada de CASP (*Critical Assessment of protein Structure Prediction*; predictioncenter.org). Desde 1994, o CASP oferece melhorias significativas na acurácia da predição os modelos, no alinhamento de sequências, na modelagem de estruturas secundárias, na montagem de proteínas e no refinamento final dos modelos [9]. E, como resultado dessa competição, diferentes técnicas são implementadas e aprimoradas, podendo ser usadas com maior confiança pela comunidade científica.

Métodos dependentes de molde

Como mencionado anteriormente, os métodos baseados em molde partem do princípio de que a estrutura tridimensional de uma proteína se mantém mais conservada ao longo da evolução. Consequentemente, alterações na sequência dos aminoácidos podem acarretar apenas pequenas modificações em sua estrutura tridimensional [11]. Ou seja, os métodos dessa categoria consideram que proteínas que possuem sequências semelhantes se enovelam em estruturas praticamente idênticas. Até mesmo sequências que possuem identidade baixa entre si (até 20% de identidade) podem assumir estruturas tridimensionais semelhantes. Portanto, existindo uma estrutura experimentalmente resolvida é possível construir um modelo tridimensional para uma proteína com estrutura desconhecida.

A origem das abordagens baseadas em molde pode ser datada no ano de 1969 quando tentativas de construção da estrutura de alfa-lactalbumina usando a estrutura da lisozima da clara de ovo de galinha como modelo foram publicadas por Browne e colaboradores [12]. A partir dessa década vários trabalhos surgiram melhorando e dando maior confiabilidade as técnicas de predição de estrutura, desempenhando um papel econômico em aplicações baseadas em estrutura e na caracterização de propriedades e funções de proteínas [13]. Nas próximas subseções discutiremos as duas metodologias dependentes de moldes mais populares, modelagem comparativa e *threading*.

Modelagem comparativa

Entre as técnicas baseadas em molde, a modelagem comparativa, também chamada anteriormente de modelagem por homologia, é a metodologia mais utilizada para a predição da estrutura da proteína quando apenas os dados da sequência estão disponíveis. Para que se possa adotar essa abordagem, é necessária uma proteína-molde (ou *template*) com estrutura tridimensional resolvida disponível. Esta deve apresentar uma estrutura primária com identidade mínima, entre 25% e 30%, com a sequência da proteína que se deseja modelar (proteína-alvo). É a partir da base estrutural do molde que será possível propor um modelo tridimensional para a sequência de aminoácidos da proteína-alvo [14,15].

A obtenção de um modelo tridimensional através da modelagem comparativa segue quatro etapas principais (Figura 3) [16]. São elas:

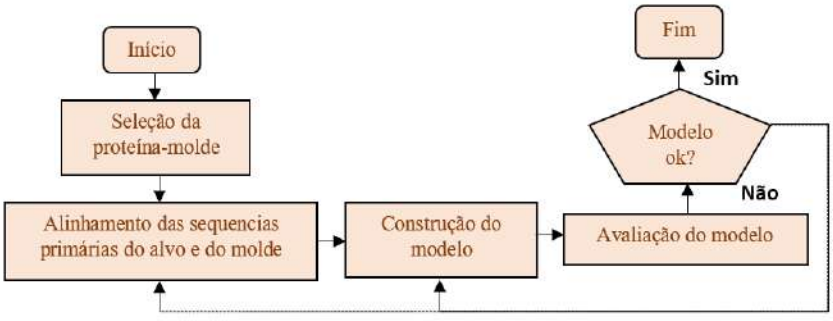
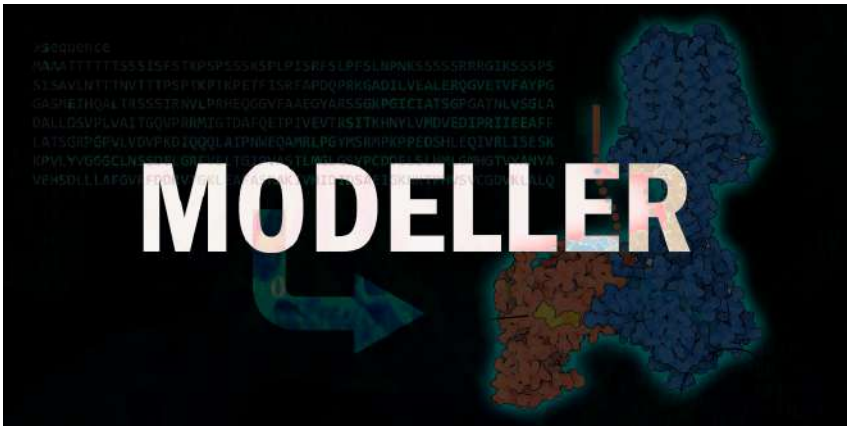


Figura 3. Fluxograma etapas da modelagem comparativa.

1. **seleção da proteína-molde** – identificação de uma ou múltiplas estruturas primárias de proteínas resolvidas experimentalmente com similaridade com a sequência da proteína-alvo pela ferramenta *Basic Local Alignment Search Tool* (BLAST) [17]. Fatores como similaridade, identidade, número de *gaps* e cobertura são avaliados contra as sequências na base de dados de estruturas conhecidas, PDB [5], para determinar os melhores moldes. Encontrando resultados, outros fatores como função biológica, qualidade da estrutura experimental, presença de ligantes, substratos e cofatores são empregados para a escolha do molde;
2. **alinhamento da estrutura primária do molde e do alvo** – escolhido o(s) molde(s), é feito o alinhamento entre sequência alvo e molde(s). Os alinhamentos da etapa anterior são feitos para buscar as sequências apenas. Porém, nessa segunda etapa, um alinhamento mais rebuscado é necessário para gerar a cadeia principal da estrutura [9]. Regiões que não possuem correspondência nas sequências precisam ser desconsideradas ou preenchidas com *gaps*. Ligantes, substratos, e outros cofatores precisam ter sua importância estudada nas estruturas de referência para serem incluídos ou não nos modelos criados;
3. **construção do modelo** – feita a partir das informações estruturais do(s) molde(s) escolhido(s). Os dois métodos mais aplicados para a construção são os métodos de satisfação de restrições espaciais [14] e união de corpos rígidos [18]. O método de satisfação de restrições espaciais assume que vários parâmetros geométricos, como distâncias e ângulos são conservados entre proteínas homólogas, ao comparar as posições equivalentes oriundas do alinhamento de sequências. Já nos métodos baseados em união de corpos rígidos, o modelo é montado a partir de um pequeno número de corpos rígidos obtidos das cadeias principais das regiões alinhadas [19,20]. Nesse método a modelagem envolve encaixar as regiões rígidas comuns na estrutura modelada e reconstruir as regiões não conservadas, ou seja, cadeias laterais e alças (*loops*) [21]; e
4. **avaliação do modelo** – gerados os modelos, estes são avaliados para determinar a qualidade e adequação da estrutura tridimensional criada. Geralmente, os programas geram muitos modelos e os classificam de acordo com um ou mais método de pontuações. Uma vez que cada método avalia o modelo criado de uma perspectiva diferente, a combinação de vários métodos de avaliação pode permitir a obtenção de um modelo mais confiável [9]. Uma das avaliações empregada é o gráfico de Ramachandran, que mostra se os resíduos do modelo tridimensionais estão em regiões previamente estabelecidas como permitidas de acordo com os ângulos de torção φ e ψ dos resíduos. A avaliação pode não ser a etapa final na modelagem comparativa, uma vez que alguns erros no alinhamento ou na construção podem acontecer e exigir a repetição das etapas anteriores do processo (Figura 3).

Os softwares para modelagem comparativa, MODELLER [14] e SWISS-MODEL [18] serão discutidos em detalhes e com exemplos práticos mais adiante.



Tutorial: modelagem de proteínas usando MODELLER

Nesta seção será abordado a ferramenta MODELLER. O MODELLER é um *software* com vários pacotes criado por Andrej Sali e Tom L. Blundell em 1989 [14]. O MODELLER é uma ferramenta gratuita com uso restrito a linha de comando e não possui interface gráfica de usuário. Atualmente, o MODELLER utiliza a linguagem *Python* como linguagem de controle, o que também é um requisito para o funcionamento do programa. Com isso, todos os *scripts* para realizar a modelagem são desenvolvidos em *Python*. O programa pode ser rodado nos sistemas operacionais baseados em UNIX, Windows e Mac.

Para a construção do modelo tridimensional, o MODELLER utiliza o método de satisfação de restrições espaciais. Através do alinhamento das sequências, características espaciais como as distâncias entre carbonos-alfa ($C\alpha - C\alpha$) e ângulos diedrais da cadeia principal e lateral dos resíduos são transferidos da estrutura molde para a estrutura alvo. Essas restrições espaciais são obtidas de forma empírica, a partir de uma base de dados contendo informações sobre o alinhamento de proteínas com estruturas conhecidas presentes em famílias proteicas.

As restrições estereoquímicas, como comprimentos e ângulos de ligação, e contatos atômicos não ligados, são obtidos dos campos de força da mecânica molecular. As restrições espaciais e os termos obtidos pelo campo de força são combinados em uma função objetivo. A função objetivo é otimizada no espaço Cartesiano visando minimizar as violações de todas as restrições utilizando os métodos de gradiente conjugado e dinâmica molecular por *simulated annealing*. Portanto, vários modelos com pequenas variações são calculados amostrando a estrutura inicial, e a variabilidade entre os modelos contribuem para melhor estimar o enovelamento da proteína-alvo.

Usando o MODELLER para modelagem comparativa

Como falado anteriormente, o MODELLER não possui uma interface gráfica, sendo restrito o uso da linha de comando. Para instalação é necessária uma licença de utilização para usuários que pode ser solicitada no site do software.

Link para download e guias de instalação | Link para licença

A seguir vamos detalhar o passo-a-passo para a construção de um modelo utilizando o MODELLER, para uma sequência de interesse, seguindo as etapas mencionadas anteriormente.

1. Seleção da proteína-molde

Para seleção do melhor molde, considera-se inicialmente alguns fatores importantes entre a estrutura primária da proteína-alvo e da proteína-molde. Como, identidade entre as sequências acima de 25%, e se a semelhança entre sequências é significativa com toda a extensão da nossa sequência alvo (parâmetro de cobertura). A medida de significância estatística, *E-value*, também deve ser avaliada. O valor de *E-value* compara o número de alinhamentos que seriam esperados apresentando valores iguais ou melhores que o encontrado por acaso, dado o tamanho do banco de dados. A qualidade experimental da estrutura tridimensional da proteína-molde também é outro fator importante. A preferência é para estruturas resolvidas de alta qualidade, com resolução menor ou igual a 2 Å, fator R menor de 20%, e em caso de enzimas, estruturas complexadas com o substrato. Todos esses fatores podem garantir uma melhor confiabilidade no modelo que será construído.

i) Busque a sequência de sua proteína-alvo:

A sequência pode ser encontrada em bancos de dados, como o UniProt e Genbank. Usaremos o UniProt nessa etapa. Entre no site e digite o nome da sua proteína no local de busca, em seguida escolha o "*Entry*" (conhecido como ID, ou identificador) que melhor represente sua proteína, vá em "*Sequence*" e baixe o formato fasta. Essa etapa só é necessária quando não se tem a sequência da proteína-alvo. Em casos em que se tem o sequenciamento da proteína inicia-se pela etapa de busca do molde.

Acesso ao UniProt: www.uniprot.org/uniprot/

Como exemplo será apresentado a modelagem da enzima Acetolactato sintase (ALS), importante para síntese de aminoácidos de cadeia ramificada em organismos vegetais. A sequência de ALS de *Arabidopsis thaliana* está disponível no Uniprot (ID P17597) e foi usada em todos os passos seguintes. Portanto, a sequência de interesse, de acordo com o arquivo FASTA do Uniprot, é:

```
>sp|P17597|ILVB_ARATH Acetolactate synthase
MAAATTTTTTSSSISFSTKPSPSSSKSPLPISRFSLPFSLNPNKSSSSSRRRGIKSSSPS
SISAVLNTTTNVTTTPSPTKPTKPETFISRFAPDQPRKGADILVEALERQGVETVFAYPG
GASMEIHQALTRSSSIRNVLPRHEQGGVFAAEGYARSSGKPGICIATSGPGATNLVSGLA
DALLDSVPLVAITGQVPRRMIGTDAFQETPIVEVTRSITKHNYLVMDVEDIPRIIEEAFF
LATSGRPGPVLVDVPKDIQQQLAIPNWEQAMRLPGYMSRMPKPPEDSHLEQIVRLISESK
KPVLYVGGGCLNSSDELGRFVELTGIPVASTLMGLGSYPCDDELSLHMLGMHGTVYANYA
VEHSDLLLAFGVRFDDRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPHVSVCGDVKLALQ
GMNKVLENRAEELKLDFGVWRNELNVQKQKFPLSFKTFGEAIPPQYAIKVLDELTDGKAI
ISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASVANPDAIVVDIDGDGS
FIMNVQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPN
MLLFAAACGIPAARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDV
```

ii) Busque o molde:

O molde pode ser encontrado no banco de dados PDB (*Protein Data Bank*). Para isso utilizaremos o servidor Web BLAST [17], escolhendo a opção Protein BLAST, e buscaremos pela estrutura onde sua sequência tem identidade >25%, melhor resolução cristalográfica (quanto menor melhor), melhor cobertura e o *E-value* baixo (quanto mais próximo de 0, mais chances de ser significativa a correspondência, ou seja, não aconteceu por acaso) [5,22].

Quando não se encontra um modelo que satisfaça essas exigências, é necessário aplicar outras abordagens como *threading* ou modelagem *ab initio*.

Acesso ao BLAST: blast.ncbi.nlm.nih.gov/Blast.cgi

Faça a *upload* do arquivo fasta com a sequência de interesse, a qual deseja modelar, ou copie e cole em "*Enter accession number(s), gi(s), or FASTA sequence(s)*". Na opção "*Database*" escolha "*PROTEIN DATA BANK proteins* (pdb)" e depois clique em *BLAST* (Figura 4).

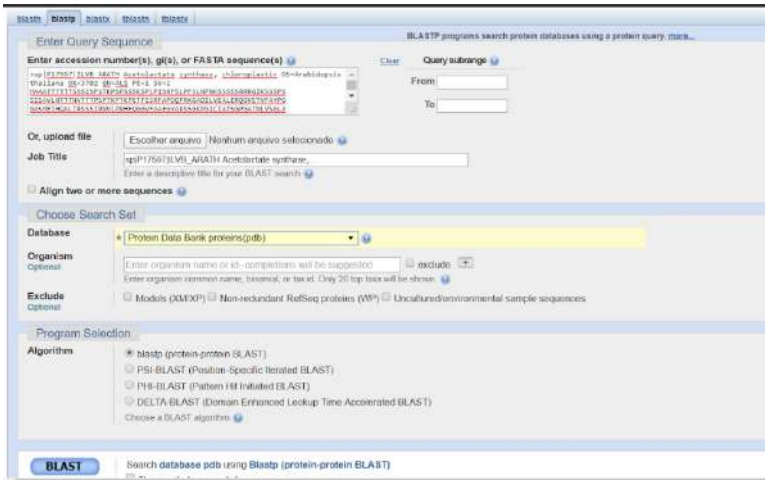


Figura 4. Página do BLASTp para busca de um molde.

Analise os valores de identidade (*Per Ident*), cobertura (*Query Cover*) e *E-value* (Figura 5) para escolher o melhor molde e buscar na base de dados do PDB. O código PDB se encontra na coluna *Accession*. Nesse exemplo, os três primeiros resultados são muito parecidos, diferindo muito pouco na identidade. Quando vamos na base de dados do PDB e verificamos os dois primeiros da lista, que possuem alta identidade, alta cobertura e baixo *E-value*, percebemos que o primeiro tem uma resolução melhor, mas o interesse está em enzimas que foram resolvidas com um ligante em específico. E este é o caso do segundo da lista. Possui uma resolução razoável, alta identidade, alta cobertura, baixo *E-value* e contém o ligante desejado. A existência do ligante pode auxiliar no uso de outras técnicas computacionais, como o atracamento molecular, já que essa estrutura pode estar na conformação necessária para ocorrer a ligação entre proteína e ligante.

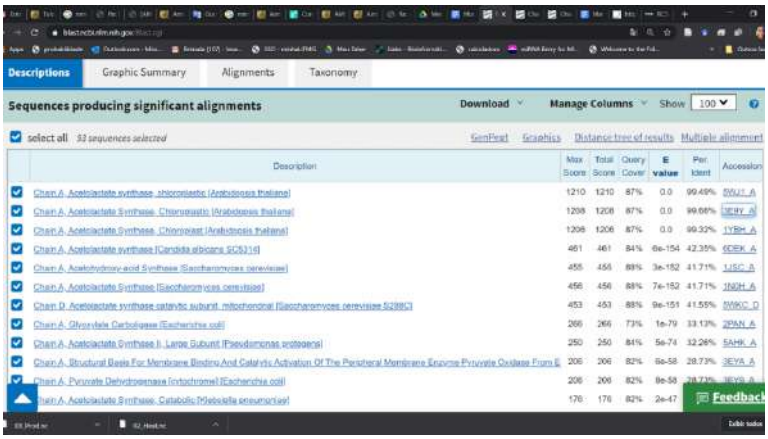


Figura 5. Resultado do BLAST para a seleção do molde.

Portanto, escolhemos como molde a estrutura de código PDB 3E9Y [23], também uma ALS de *A. thaliana*. Com o código escolhido deve-se ir ao banco de dados *Protein Data Bank* e digitar o código PDB do molde selecionado (Figura 6).

Figura 6. Página da base de dados PDB para busca da estrutura

3D do molde. Acesso PDB: https://www.rcsb.org/

E, a seguir, baixar o arquivo PDB (Figura 7).

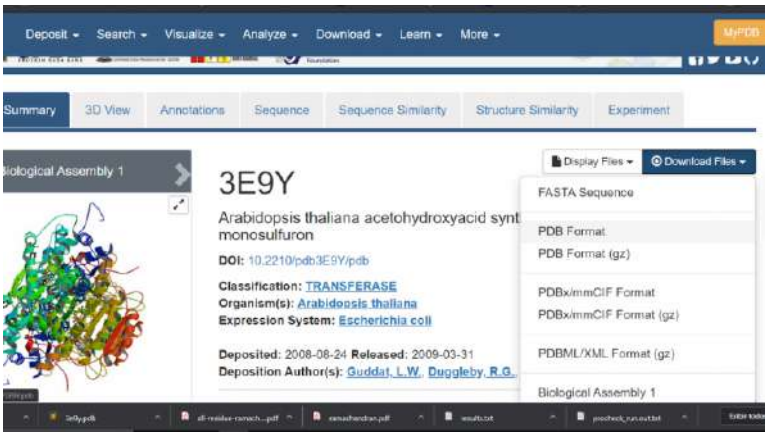


Figura 7. Página do código PDB associado ao molde. O arquivo pode ser baixado na guia Download Files no formato PDB ou mmCIF.

2. Alinhamento das sequências do molde e do alvo

O alinhamento permite encontrar correspondência entre resíduos estruturalmente equivalentes levando em conta suas posições nas sequenciais. Com o alinhamento é possível distinguir entre regiões estruturalmente conservadas e variáveis [24]. O MODELLER aceita como entrada alinhamentos de outros programas como por exemplo BLAST [17] e CLUSTAL [25], contanto que esteja no formato correto. Porém, o MODELLER possui sua própria rotina de alinhamento, produzindo os arquivos necessários de maneira eficiente.

Para a etapa de alinhamento, três arquivos são necessários (todos os arquivos devem estar no mesmo diretório):

i) Arquivo da sequência da proteína a ser modelada em formato PIR.

O MODELLER utiliza o formato PIR, que é parecido com o FASTA baixado na busca da sequência do alvo, mais com um cabeçalho característico do formato. Vá até o arquivo FASTA baixado e insira o novo cabeçalho:

```
> P1; nome da proteína-alvo
sequence: nome da proteína-alvo::::::::
```

Após a sequência, no final do arquivo deve-se inserir um '*' e salvar como ".txt" (Figura 8). O nome dado para a proteína-alvo não deve ser alterado nos próximos passos.

```
>P1;p17597
sequence:p17597:::::::
MAAATTTTTTSSSISFSTKPSPSSSKSPLPISRFSLPFSLNPNKSSSSSRRRGIKSSSPS
SISAVLNTTTNVTTTPSPTKPTKPETFISRFAPDQPRKGADILVEALERQGVETVFAYPG
GASMEIHQALTRSSSIRNVLPRHEQGGVFAAEGYARSSGKPGICIATSGPGATNLVSGLA
DALLDSVPLVAITGQVPRRMIGTDAFQETPIVEVTRSITKHNYLVMDVEDIPRIIEEAFF
LATSGRPGPVLVDVPKDIQQQLAIPNWEQAMRLPGYMSRMPKPPEDSHLEQIVRLISESK
KPVLYVGGGCLNSSDELGRFVELTGIPVASTLMGLGSYPCDDELSLHMLGMHGTVYANYA
VEHSDLLLAFGVRFDDRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPHVSVCGDVKLALQ
GMNKVLENRAEELKLDFGVWRNELNVQKQKFPLSFKTFGEAIPPQYAIKVLDELTDGKAI
ISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASVANPDAIVVDIDGDGS
FIMNVQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPN
MLLFAAACGIPAARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDV
ITEGDGRIKY*
```

Figura 8. Arquivo com a sequência do alvo em formato PIR.

ii) Arquivo PDB do molde.

O MODELLER aceita o formato mmCIF, basta fazer o download do mesmo no passo de busca do molde (Figura 7).

iii) Script de alinhamento em *python*.

Digite o script abaixo em um editor de texto, substitua o nome dos arquivos (em negrito) pelo nome dos seus arquivos e salve como “ `alinhar.py` ”.

```python
# Importando o modeller
--------------------------
from modeller import *

# Importando a classe automodel
---------------------------------------
from modeller.automodel import *

# Novo ambiente para o modeller
-----------------------------------
env = environ()

# Novo ambiente para o alinhamento
--------------------------------------
aln = alignment(env)

# Modelo alvo. File= ID do PDB molde.
-----------------------------------------
# Model_segment= Cadeia usada do molde
-------------------------------------------
md1 = model(env, file='3e9y', model_segment=('FIRST:A','LAST:A'))

# Alinhamento.
-----------------
# Align_codes= PDB do molde e cadeia.
-----------------------------------------
# Atom_files= Nome do arquivo PDB do molde
-----------------------------------------------
```

```
aln.append_model(md1, align_codes='3e9yA', atom_files='3e9y.pdb')

# Fazer o alinhamento.
--------------------------
# File= arquivo com sequência do alvo.
-------------------------------------------
# Aling_codes= ID do alvo.
-----------------------------
aln.append(file='ahas.txt', align_codes='p17597')

# Alinhamento de sequencias
-----------------------------
aln.align2d()

# Arquivos de alinhamento formato PIR
-----------------------------------------
aln.write(file='ahas_3e9y.ali', alignment_format='PIR')

# Arquivos de alinhamento formato PAP
-----------------------------------------
aln.write(file='ahas_3e9y.pap', alignment_format='PAP')
```

Vá ao terminal de linhas de comando, navegue até o diretório onde os arquivos se encontram e digite (o $ representa o prompt do terminal e não deve ser digitado):

```
$ python alinhar.py
```

No caso do Windows, salve os arquivos dentro de um diretório na pasta do MODELLER (geralmente estará nos arquivos de programa do disco c:), busque por **modeller** no local de busca do Windows e abra o arquivo MODELLER. Um terminal será aberto. Vá até a pasta onde estão os arquivos com o comando " `cd` " e digite " `python alinhar.py` ".

Os arquivos gerados serão " `.ali` " (Figura 9) e " `.pap` " (Figura 10). O primeiro será usado para a etapa de modelagem e o segundo contém os resíduos conservados.

```
>P1;3e9yA
structureX:3e9y.pdb:   87 :A:+581 :A:MOL_ID  1; MOLECULE  ACETOLACTATE SYNTHASE, CHLOROPLASTIC;
------------------------------------------------------------------------------
-----------FISRFAPDQPRKGADILVEALERQGVETVFAYPGGASMEIHQALTRSSSIRNVLPRHEQGGVFA
AEGYARSSGKPGICIATSGPGATNLVSGLADALLDSVPLVAITGQVPRRMIGTDAFQETPIVEVTRSITKHNYLV
MDVEDIPRIIEEAFFLATSGRPGPVLVDVPKDIQQQLAIPNWEQAMRLPGYMSRMPKPPEDSHLEQIVRLISESK
KPVLYVGGGCLNSSDELGRFVELTGIPVATTLMGLGSYP-DDELSLHMLGMHGTVYANYAVEHSDLLLAFGVRFD
DRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPHVSVCGDVKLALQGMNKVLENRAEELKLDFGVWRNELNVQKQK
FPLSFKTFGEAIPPQYAIKVLDELTDGKAIISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASV
ANPDAIVVDIDGDGSFIMNVQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPN
MLLFAAACGIPAARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDVITEGDGRI--*

>P1;p17597
sequence:p17597:     : :     : :::-1.00:-1.00
MAAATTTTTTSSSISFSTKPSPSSSKSPLPISRFSLPFSLNPNKSSSSSRRRGIKSSSPSSISAVLNTTTNVTTT
PSPTKPTKPETFISRFAPDQPRKGADILVEALERQGVETVFAYPGGASMEIHQALTRSSSIRNVLPRHEQGGVFA
AEGYARSSGKPGICIATSGPGATNLVSGLADALLDSVPLVAITGQVPRRMIGTDAFQETPIVEVTRSITKHNYLV
MDVEDIPRIIEEAFFLATSGRPGPVLVDVPKDIQQQLAIPNWEQAMRLPGYMSRMPKPPEDSHLEQIVRLISESK
KPVLYVGGGCLNSSDELGRFVELTGIPVASTLMGLGSYPCDDELSLHMLGMHGTVYANVAVEHSDLLLAFGVRFD
DRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPHVSVCGDVKLALQGMNKVLENRAEELKLDFGVWRNELNVQKQK
FPLSFKTFGEAIPPQVAIKVLDELTDGKAIISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASV
ANPDAIVVDIDGDGSFIMNVQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPN
MLLFAAACGIPAARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDVITEGDGRIKY*
```

Figura 9. Arquivo com o alinhamento gerado em formato .ali. A

falta de resíduos (missing residues) nas posições correspondentes entre as sequências é assinada com o caractere (-), chamado de gap.

```
 _aln.pos         10        20        30        40        50        60
3e9yA     ------------------------------------------------------------
p17597    MAAATTTTTTSSSISFSTKPSPSSSKSPLPISRFSLPFSLNPNKSSSSSRRRGIKSSSPSSISAVLNT
 _consrvd

 _aln.p   70        80        90       100       110       120       130
3e9yA     ------------------FISRFAPDQPRKGADILVEALERQGVETVFAYPGGASMEIHQALTRSSSI
p17597    TTNVTTTPSPTKPTKPETFISRFAPDQPRKGADILVEALERQGVETVFAYPGGASMEIHQALTRSSSI
 _consrvd                   **************************************************

 _aln.pos  140       150       160       170       180       190       200
3e9yA     RNVLPRHEQGGVFAAEGYARSSGKPGICIATSGPGATNLVSGLADALLDSVPLVAITGQVPRRMIGTD
p17597    RNVLPRHEQGGVFAAEGYARSSGKPGICIATSGPGATNLVSGLADALLDSVPLVAITGQVPRRMIGTD
 _consrvd ************************************************************************

 _aln.pos    210       220       230       240       250       260       270
3e9yA     AFQETPIVEVTRSITKHNYLVMDVEDIPRIIEEAFFLATSGRPGPVLVDVPKDIQQQLAIPNWEQAMR
p17597    AFQETPIVEVTRSITKHNYLVMDVEDIPRIIEEAFFLATSGRPGPVLVDVPKDIQQQLAIPNWEQAMR
 _consrvd ************************************************************************

 _aln.pos      280       290       300       310       320       330       340
3e9yA     LPGYMSRMPKPPEDSHLEQIVRLISESKKPVLYVGGGCLNSSDELGRFVELTGIPVATTLMGLGSYP-
p17597    LPGYMSRMPKPPEDSHLEQIVRLISESKKPVLYVGGGCLNSSDELGRFVELTGIPVASTLMGLGSYPC
 _consrvd ************************************************************ ********

 _aln.pos        350       360       370       380       390       400
3e9yA     DDELSLHMLGMHGTVYANYAVEHSDLLLAFGVRFDDRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPH
p17597    DDELSLHMLGMHGTVYANYAVEHSDLLLAFGVRFDDRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPH
 _consrvd ************************************************************************

 _aln.p  410       420       430       440       450       460       470
3e9yA     VSVCGDVKLALQGMNKVLENRAEELKLDFGVWRNELNVQKQKFPLSFKTFGEAIPPQYAIKVLDELTD
p17597    VSVCGDVKLALQGMNKVLENRAEELKLDFGVWRNELNVQKQKFPLSFKTFGEAIPPQYAIKVLDELTD
 _consrvd ************************************************************************

 _aln.pos  480       490       500       510       520       530       540
3e9yA     GKAIISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASVANPDAIVVDIDGDGSFIMN
p17597    GKAIISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASVANPDAIVVDIDGDGSFIMN
 _consrvd ************************************************************************

 _aln.pos    550       560       570       580       590       600       610
3e9yA     VQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPNMLLFAAACGIPA
p17597    VQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPNMLLFAAACGIPA
 _consrvd ************************************************************************

 _aln.pos      620       630       640       650       660       670
3e9yA     ARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDVITEGDGRI--
p17597    ARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDVITEGDGRIKY
 _consrvd ************************************************************
```

*Figura 10. Arquivo com o alinhamento gerado em formato .pap, mostrando os resíduos conservados entre as sequências com *.*

3. Construção do modelo

Para gerar cada modelo, o MODELLER utiliza a cadeia principal da estrutura molde e a otimiza em relação da sequência alvo, aplicando um grau de aleatoriedade nas coordenadas. Essas coordenadas são otimizadas através da busca pelo mínimo de energia das funções objetivo do MODELLER. Como encontrar o mínimo global de energia através de uma função objetivo não é garantido, recomenda-se repetir o procedimento de construção do modelo várias vezes. Com a aleatoriedade embutida no procedimento modelos diferentes são gerados a cada rodada, aumentando a amostragem de conformações de modelos gerados. Considere a construção de algumas dezenas a centenas de modelos, para então selecionar o mais adequado. Porém, a tendência na geração de muitos modelos é que esses se aproximem em conformação e energia.

Para essa etapa três arquivos são necessários:

1. Arquivo de alinhamento gerado na etapa anterior (formato ".ali").
2. Arquivo PDB do molde.
3. *Script* do MODELLER de construção de modelos em *Python*, que será executado como o anterior. Substitua o nome dos arquivos (negrito) pelo nome dos seus arquivos e salve como "gerar_modelo.py". O número de modelos será indicado nas 3 últimas linhas do código.

```
# Importando o modeller
#-------------------------
from modeller import *

# Importando a classe automodel
#---------------------------------
from modeller.automodel import *


# Novo ambiente para o modeller
#---------------------------------
env = environ()


a = automodel(
    env,
    alnfile='ahas_3e9y.ali',
    knowns= '3e9yA',
    sequence='p17597',
    assess_methods=(
        assess.DOPE,
        assess.GA341
    )
)

# Começar no modelo 1
#----------------------
a.starting_model = 1

# Terminar no modelo 5
#-----------------------------------------
a.ending_model = 5


# Construir os modelos
#-----------------------------------------
a.make()
```

Vá ao terminal de linhas de comando, navegue até o diretório onde os arquivos se encontram e digite:

```
$ python gerar_modelo.py >&1 | tee genmodelo.log
```

A inserção de "`>&1 | tee genmodelo.log`" no comando é apenas para garantir a criação de um arquivo de registro com todas as informações da geração dos modelos. Ao finalizar, o script gera cinco modelos tridimensionais para ser avaliados (Figura 11). Os modelos podem ser

visualizados por qualquer programa que leia o formato PDB, como o PyMOL. O arquivo de log também mostra as pontuações de cada modelo.

Figura 11. Final do arquivo .log mostrando informações sobre os modelos construídos. Os nomes dos modelos se encontram na primeira linha, seguida das pontuções.

4. Avaliação do modelo

Após a construção de modelos para a proteína-alvo é necessário verificar se existem possíveis erros, como por exemplo erros no alinhamento ou escolha errada do molde tridimensional usado. Vale ressaltar que modelos construídos por métodos computacionais sempre serão passíveis de erros. A etapa de avaliação deve conduzir a bons modelos com base na magnitude dos erros [26].

A escolha do "melhor" modelo (ou "melhores" modelos) pode ser feita de várias maneiras. Podemos escolher o melhor modelo a partir do menor valor da função de energia do MODELLER (*molpdf*) ou através do menor valor de DOPE score (*Discrete Optimized Protein Energy*) (Figura 11). Neste exemplo usaremos o DOPE score e o modelo selecionado com menor valor é o de número quatro (`p17597.B99990004.pdb`). Porém, pode-se selecionar mais de um modelo para avaliação de acordo com os mais bem ranqueados pelo MODELLER.

Avaliação do modelo pelo servidor SAVES

Para essa etapa de avaliação do modelo, usa-se o arquivo pdb do modelo escolhido para submissão ao servidor web SAVES. O SAVES concentra vários programas que avaliam pontos específicos da estrutura de forma a dar mais confiabilidade ao modelo.

Acesso ao SAVES: saves.mbi.ucla.edu/

Escolha as opções de verificação a partir do PROCHECK (avalia quanto à qualidade estereoquímica), WHATCHECK (qualidade dos contatos atômicos de todos os átomos de cada resíduo) e VERIFY 3D (compatibilidade do modelo tridimensional com sua estrutura primária).

Os resultados apresentados pelo SAVES aqui são para o modelo de ALS com o menor valor de pontuação DOPE (`p17597.B99990004.pdb`). O gráfico do VERIFY 3D (Figura 12) mostra que menos de 80% dos resíduos estão em ambientes químicos confiáveis. Para essa interpretação é atribuído uma pontuação para cada resíduo referente a base de dados de estrutura do PDB. Como padrão do programa, para que um modelo seja aceito, ou seja, seja confiável, mais de 80% dos resíduos devem ser aceitos.

Figura 12. Gráfico de saída VERIFY 3D.

Os resultados do PROCHECK, ilustrado pelo gráfico de Ramachandran (Figura 13), mostram que o resíduo PHE 87 não se encontra em uma região favorável.

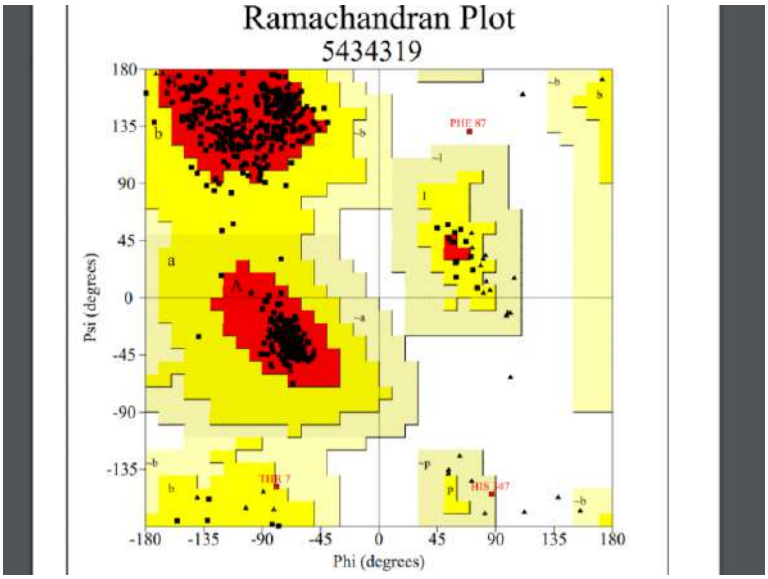


Figura 13. Gráfico de Ramachandran gerado pelo Procheck.

Porém, os resultados estatísticos do Ramachandran demonstram que 92,4% dos resíduos se encontram em regiões favoráveis (Figura 14). Portanto, deve-se observar na estrutura tridimensional em qual região estrutural ele se encontra. Por exemplo, se o resíduo se encontra em uma região flexível como uma região de alça, o MODELLER pode não ter encontrado uma conformação

favorável para esse resíduo. Regiões de *loop* são as mais difíceis de serem modeladas e precisam de maior atenção.

Phi (degrees)

Plot statistics

Residues in most favoured regions [A,B,L]	524	92.4%
Residues in additional allowed regions [a,b,l,p]	39	6.9%
Residues in generously allowed regions [~a,~b,~l,~p]	3	0.5%
Residues in disallowed regions	1	0.2%
Number of non-glycine and non-proline residues	567	100.0%
Number of end-residues (excl. Gly and Pro)	2	
Number of glycine residues (shown as triangles)	53	
Number of proline residues	48	
Total number of residues	670	

Based on an analysis of 118 structures of resolution of at least 2.0 Angstroms and R-factor no greater than 20%, a good quality model would be expected to have over 90% in the most favoured regions.

Figura 14. Resultados estatísticos do Ramachandran gerado pelo PROCHECK.

Para analisar este resíduo utilizou-se um programa de visualização molecular. Utilizando o PyMOL [27] (Figura 15) podemos confirmar que o resíduo se encontra no final de um loop, uma região desordenada que não participa de nenhuma interação com a proteína. Essa região não foi bem alinhada ao molde, por se tratar de resíduos inexistentes no molde e presentes apenas na sequência do alvo. Esses resíduos correspondem a uma região de peptídeo sinal. Naturalmente essa proteína tem sua função após ser transferida ao cloroplasto, perdendo o peptídeo sinal.

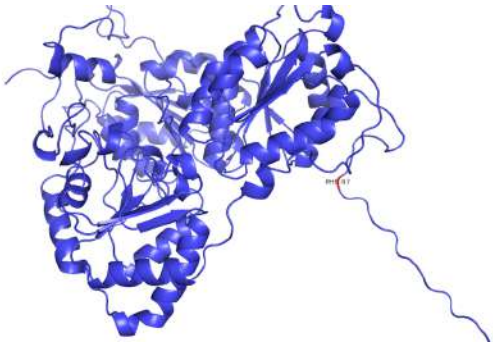


Figura 15. Visualização em cartoon da proteína modelada. Em vermelho o resíduo PHE 87.

Para correção, editou-se manualmente o alinhamento usado para construção do modelo. Retirou-se os 87 primeiros aminoácidos e o script de modelagem foi executado novamente (Figura 16).

```
>P1;3e9yA
structureX:3e9y.pdb:  87 :A:+581 :A:MOL_ID  1; MOLECULE  ACETOLACTATE SYNTHASE, CHLOROPLAST
-------------------------------------------------------------------------------
-----------FISRFAPDQPRKGADILVEALERQGVETVFAYPGGASMEIHQALTRSSSIRNVLPRHEQGGVFA
AEGYARSSGKPGICIATSGPGATNLVSGLADALLDSVPLVAITGQVPRRMIGTDAFQETPIVEVTRSITKHNYLV
MDVEDIPRIIEEAFFLATSGRPGPVLVDVPKDIQQQLAIPNWEQAMRLPGYMSRMPKPPEDSHLEQIVRLISESK
KPVLYVGGGCLNSSDELGRFVELTGIPVATTLMGLGSYP-DDELSLHMLGMHGTVYANYAVEHSDLLLAFGVRFD
DRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPHVSVCGDVKLALQGMNKVLENRAEELKLDFGVWRNELNVQKQK
FPLSFKTFGEAIPPQYAIKVLDELTDGKAIISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASV
ANPDAIVVDIDGDGSFIMNVQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPN
MLLFAAACGIPAARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDVITEGDGRI--*

>P1;p17597
sequence:p17597:     : :     : :::-1.00:-1.00
MAAATTTTTTSSSISFSTKPSPSSSKSPLPISRFSLPFSLNPNKSSSSSRRRGIKSSSPSSISAVLNTTTNVTTT
PSPTKPTKPETFISRFAPDQPRKGADILVEALERQGVETVFAYPGGASMEIHQALTRSSSIRNVLPRHEQGGVFA
AEGYARSSGKPGICIATSGPGATNLVSGLADALLDSVPLVAITGQVPRRMIGTDAFQETPIVEVTRSITKHNYLV
MDVEDIPRIIEEAFFLATSGRPGPVLVDVPKDIQQQLAIPNWEQAMRLPGYMSRMPKPPEDSHLEQIVRLISESK
KPVLYVGGGCLNSSDELGRFVELTGIPVASTLMGLGSYPCDDELSLHMLGMHGTVYANYAVEHSDLLLAFGVRFD
DRVTGKLEAFASRAKIVHIDIDSAEIGKNKTPHVSVCGDVKLALQGMNKVLENRAEELKLDFGVWRNELNVQKQK
FPLSFKTFGEAIPPQYAIKVLDELTDGKAIISTGVGQHQMWAAQFYNYKKPRQWLSSGGLGAMGFGLPAAIGASV
ANPDAIVVDIDGDGSFIMNVQELATIRVENLPVKVLLLNNQHLGMVMQWEDRFYKANRAHTFLGDPAQEDEIFPN
MLLFAAACGIPAARVTKKADLREAIQTMLDTPGPYLLDVICPHQEHVLPMIPSGGTFNDVITEGDGRIKY*
```

Figura 16. Resíduos que foram retirados do alinhamento. Esses resíduos correspondem a grande inserção de gaps (-) no alinhamento. Esses gaps iniciais foram retirados da sequência molde, portanto, ambas sequencias começam com FISR. A retirada dessa região faz com que identidade e cobertura entre as sequências sejam maiores.

Após executar os passos 2 e 3 novamente, o melhor modelo foi selecionado mais uma vez por seu valor de DOPE e submetido ao servidor SAVES. Para essa nova estrutura, os resultados do VERIFY 3D foram mais favoráveis e demonstram que o modelo gerado é confiável com 93,66% dos resíduos em ambientes químicos confiáveis (Figura 17).

Figura 17. Gráfico de saída VERIFY 3D do novo modelo gerado após a edição do alinhamento.

Além disso, o novo gráfico de Ramachandran apresenta 94,3% dos resíduos em regiões favoráveis (Figura 18).

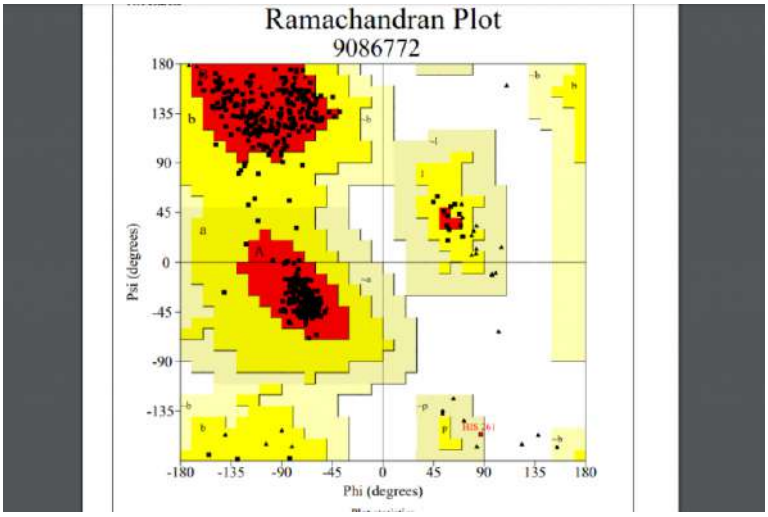


Figura 18. Gráfico de Ramachandran gerado pelo PROCHECK para o novo modelo.

Para finalizar a avaliação, realizou-se alinhamento estrutural entre a proteína-molde e o modelo construído através da ferramenta PyMOL (Figura 19). A inspeção visual dessas estruturas mostra apenas pequenas variações na estrutura secundária, correspondendo bem à conservação dos resíduos, vista no alinhamento. Quando calculado o desvio entre ambas, o valor de *Root Mean Square deviation* (RMSD) foi de 0,133 Å entre os carbonos-alfa das estruturas. Quanto menor esse valor, mais próximas são as estruturas entre si, demonstrando que a variação dos desvios médios dos átomos em relação ao molde foi baixa.

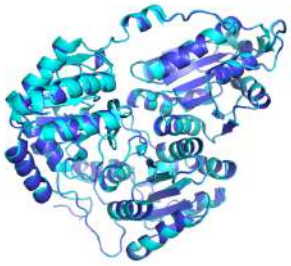


Figura 19. Alinhamento estrutural do molde e modelo construído. RMSD 0.133. Proteína-molde em azul-claro e proteína-modelada em azul-escuro.

O exemplo que apresentamos aqui gerou um possível modelo para uma sequência alvo inicial de forma simplificada pelo MODELLER. Exemplos mais avançados, que utilizam parâmetros mais rebuscados como refinamento de *loops* e estrutural por dinâmica molecular, se encontram no manual do MODELLER e devem ser explorados.

Modelagem de proteínas usando SWISS-MODEL

Nesta seção, será abordado a ferramenta SWISS-MODEL. Diferentemente do MODELLER, SWISS-MODEL é um servidor web que possui interface gráfica. Seu algoritmo também é diferente, pois utiliza regiões estruturais conservadas para construção dos modelos pelo método de união dos corpos rígidos. Porém, o SWISS-MODEL também parte do princípio de que proteínas homólogas compartilham regiões estruturalmente conservadas, como α-hélices e folhas betas, tornando-se um programa de modelagem comparativa.

O modelo é então construído a partir das regiões conservadas do molde que, em seguida, são alinhadas com a predição estrutural do alvo. Para isso, a média das posições assumidas pelos carbonos alfa das regiões conservadas estruturalmente são calculadas e usadas para o encaixe das regiões que faltam. As regiões não conservadas, que possivelmente conectam as regiões conservadas (possíveis *loops*), são como variáveis. Essas regiões são inseridas no modelo através de informações de um banco de dados de estruturas, classificadas de acordo com o tipo de resíduo e tipo de estrutura secundaria que conectam. As cadeias laterais dos aminoácidos são inseridas através da busca de bibliotecas de rotâmeros [22].

O modelo no SWISS-MODEL é gerado de forma automática com mínima interferência do usuário. Com apenas a sequência do alvo em mãos, o modelo é gerado e avaliado pelo próprio programa.

Acesso ao SWISS-MODEL: swissmodel.expasy.org/

Usando o SWISS-MODEL para modelagem comparativa

Etapa 1 – Seleção da proteína-molde

A sequência da proteína-alvo deve ser encontrada como nos passos referentes para o MODELLER, onde a sequência primária de ALS de *Arabidopsis thaliana* (Uniprot ID: P17597) foi encontrada e usada para a modelagem. Na página inicial do SWISS-MODEL, deve-se inserir a sequência do alvo e clicar em "*Search For Template*" (Figura 20). A opção "*Build Model*" também pode ser usada, mas a escolha do molde, em geral, é feita pelo programa de forma automática.

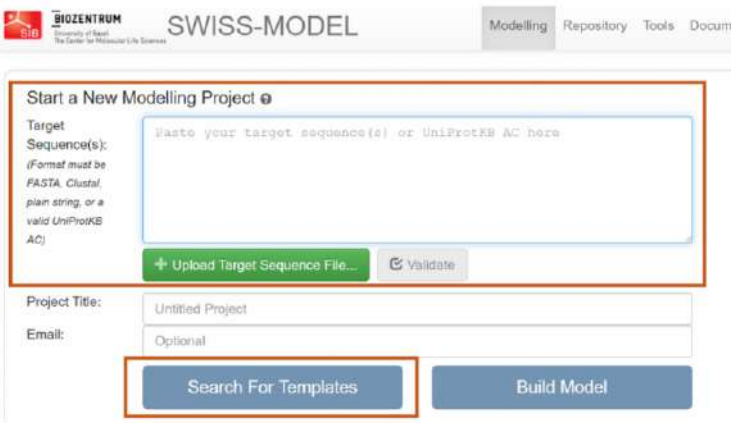


Figura 20. Página do SWISS-MODEL para busca de um molde.

Deve-se selecionar o melhor molde. Critérios de seleção da estrutura usada como referência se mantem iguais aos usados nos passos referentes ao MODELLER. Prestando atenção no valor de identidade (>25%) entre as sequências, melhor cobertura, *E-value* baixo, e melhor resolução cristalográfica (quanto menor melhor). Portanto, escolhemos como molde a estrutura de código PDB 3E9Y [23] (Figura 21).

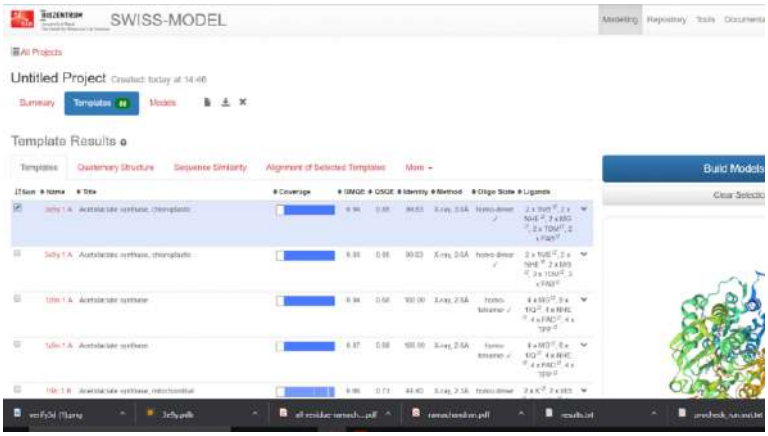


Figura 21. Resultado do Swiss-Model para a busca do molde a partir da sequência alvo.

Etapa 2- Alinhamento e construção do modelo

O SWISS-MODEL já trabalha o alinhamento internamente. Precisa-se apenas selecionar o modelo de acordo com a lista de opções dada (Figura 21) e o

servidor busca o arquivo PDB. O modelo então é construído com base no molde e alinhamento após clicar em "*Build Models*" (Figura 21). Podemos perceber que o SWISS-MODEL elimina automaticamente a região do peptídeo sinal, não permitindo a inserção do mesmo.

Etapa 3 – Avaliação do modelo

O SWISS-MODEL possui as próprias ferramentas de avaliação. Clique em "*Structure Assessment*" para avaliar o modelo (Figura 22).

Figura 22. Avaliação do modelo construído.

Para avaliar o modelo deve-se atentar para os valores de QMEAN (*Qualitative Model Energy ANalysis*) e GMQE (*Global Model Quality Estimation*). O QMEAN é um estimador conhecido como z-score. Quando o valor z está próximo de 0 significa que o modelo é considerado confiável e, portanto, existe uma boa concordância entre o modelo e estruturas experimentais de tamanho semelhantes. As propriedades geométricas oferecem uma estimativa de qualidade absoluta global. Já o GMQE, se encontra em uma faixa de 0 a 1. Quanto mais alto mais preciso é o modelo em relação ao alinhamento alvo-modelo e a cobertura do alvo. O SWISS-MODEL também fornece um gráfico de Ramachandran interativo na página web (Figura 23). A estatística do gráfico de Ramachandran é de 96,72% dos resíduos em regiões favoráveis. Após o *download* do modelo, uma comparação visual entre as estruturas molde e modelo pode ser feita por meio de alinhamento estrutural utilizando a ferramenta PyMOL.

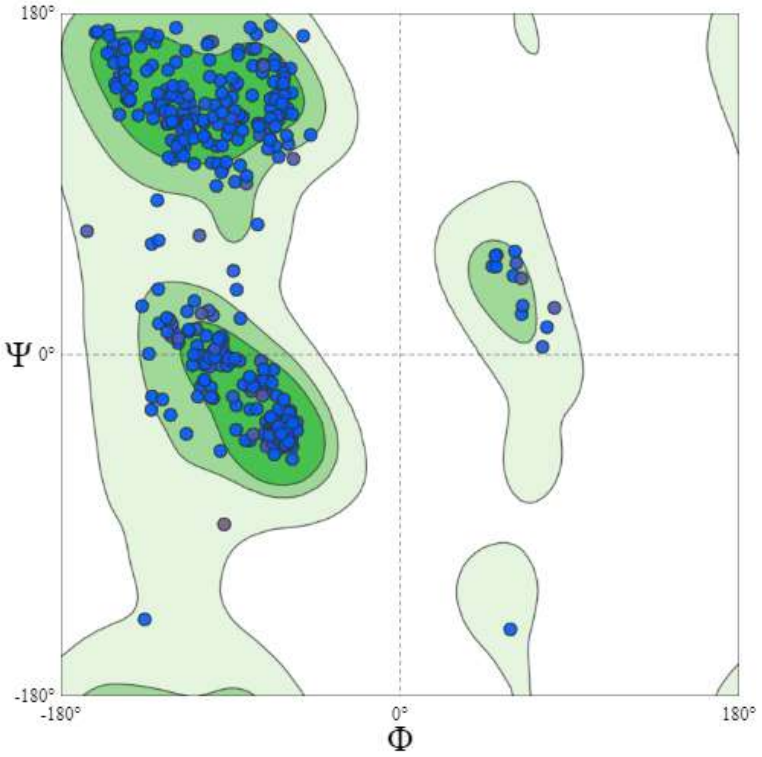


Figura 23. Gráfico de Ramachandran disponibilizado pelo SWISS-MODEL em relação ao modelo gerado.

Threading

O *threading* é um método de modelagem usado para modelar estruturas que possuem enovelamento similar a proteínas de estruturas conhecidas, porém compartilham baixo grau de similaridade. No *threading*, a sequência é fragmentada na busca por homólogos estruturais, explorando muitos alinhamentos, ao invés do alinhamento da sequência inteira de aminoácidos [28]. Portanto, essa metodologia é empregada quando existem modelos de baixa identidade que cobrem regiões diferentes da sequência alvo [29].

A modelagem *threading* baseia-se no reconhecimento das características da sequência utilizada, para isso deve-se realizar um alinhamento local que encontre moldes, estruturas disponíveis no PDB, que cubram determinadas regiões. Em seguida, uma abordagem de modelagem comparativa para cada molde selecionado é realizada, criando assim estruturas secundárias para cada região. É importante ressaltar que fatores como a qualidade das estruturas selecionadas e a identificação de moldes que cubram todos os trechos da sequência têm influência direta na qualidade dos modelos finais gerados [29].

Tutorial I-TASSER

Modelagem de proteínas com I-TASSER

Um dos programas mais populares de *Threading* é o I-TASSER [28,30], que foi premiado diversas vezes na competição CASP (*Critical Assessment of protein Structure Prediction*). O I-TASSER (Figura 24) está disponível como um servidor web para predição automatizada de estrutura de proteínas e suas respectivas funções. A identificação dos moldes a partir da segmentação da sequência de entrada é realizada usando o LOMETS [31]. O LOMETS é um meta-servidor de segmentação local, compilando vários programas de *threading*, para previsões rápidas e automatizadas de estruturas terciárias de proteínas e restrições espaciais. As regiões onde moldes não foram encontrados são modeladas utilizando a metodologia *ab initio*, realizando simulações baseadas no método de Monte Carlo. As estruturas são agrupadas e os modelos são selecionados considerando a menor energia. A última etapa realizada pelo servidor consiste na busca das possíveis funções da sequência alvo na biblioteca BioLip [32].

Figura 24. Página inicial do servidor web I-TASSER.

Acesso ao I-TASSER: zhanglab.ccmb.med.umich.edu/I-TASSER/

Estudo de caso: modelagem da sequência do peptídeo da glândula salivar de *Ixodes scapularis*

Como exemplo foi realizada a modelagem de um peptídeo putativo secretado da glândula salivar de *Ixodes scapularis*, a sequência foi obtida do Genbank, ID AAV80775.1. O arquivo de entrada exigido pelo servidor é a sequência da proteína que pode ser inserida no local indicado, ou pode ser realizado a upload do arquivo fasta (Figura 25). Para submeter o trabalho é necessário realizar a criação de uma conta, criando um usuário e uma senha. O e-mail cadastrado deve ser institucional (ou seja, um e-mail registrado em um domínio pertencente a uma universidade ou instituto de pesquisa).

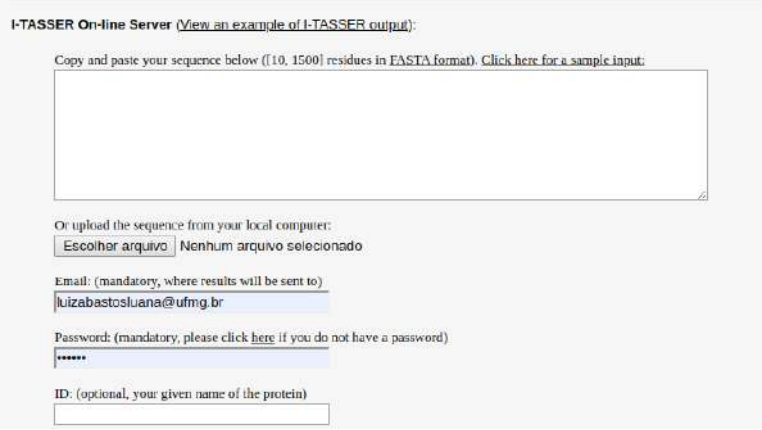


Figura 25. Página de submissão de tarefa ao servidor web I-TASSER.

Apesar de ser um servidor automatizado, o I-TASSER apresenta opções adicionais (Figura 26) que podem ser executadas de forma a personalizar a predição do modelo, são elas:

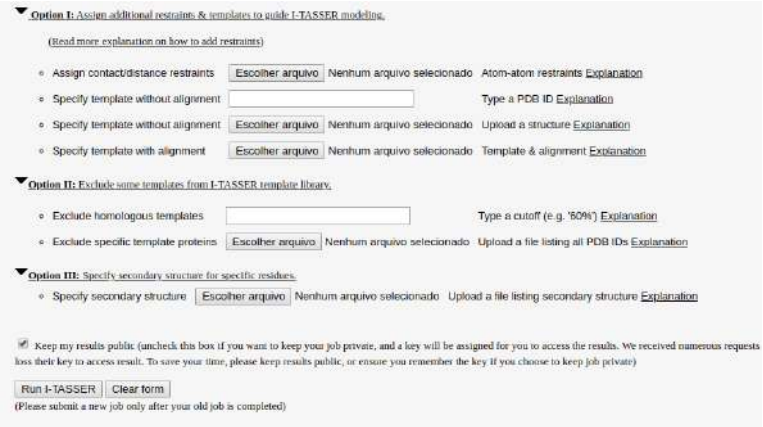


Figura 26. Opções adicionais de personalização da predição estrutural com o servidor web I-TASSER.

Opção I – Se os usuários souberem alguma informação sobre a estrutura da proteína a ser modelada, essa informação pode ser convenientemente inserida nessa opção. A inserção de informações pode melhorar a qualidade da predição estrutural e funcional. O servidor I-TASSER aceita atualmente dois tipos de restrições especificadas pelo usuário: arquivos com restrições de contato e distância; e estruturas molde com e sem alinhamento. O formato do arquivo texto para restrições de distância (Figura 27) consiste em linhas que apresentam a palavra DIST, o número e tipo de átomo do primeiro resíduo, número e tipo do átomo do segundo resíduos, e a distância entre eles em ångström. Para restrições de contato, as linhas contêm a palavra CONTACT e o número dos resíduos em contato (Figura 27). Para especificar estruturas-molde, os usuários podem atribuir o código PDB no formato PDBID:Chain, inserir informações tridimensionais (arquivo similar ao PDB), ou utilizar alinhamento no formato FASTA com informações estruturais anexadas.

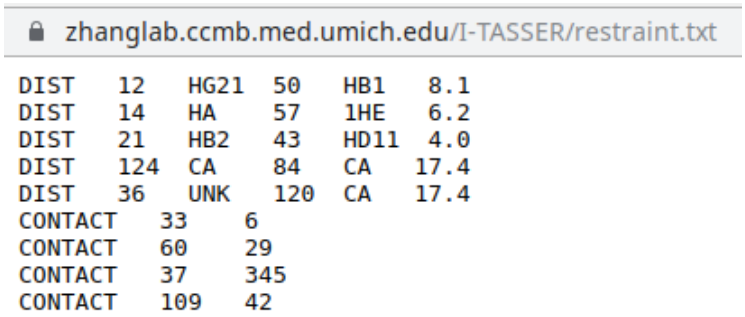
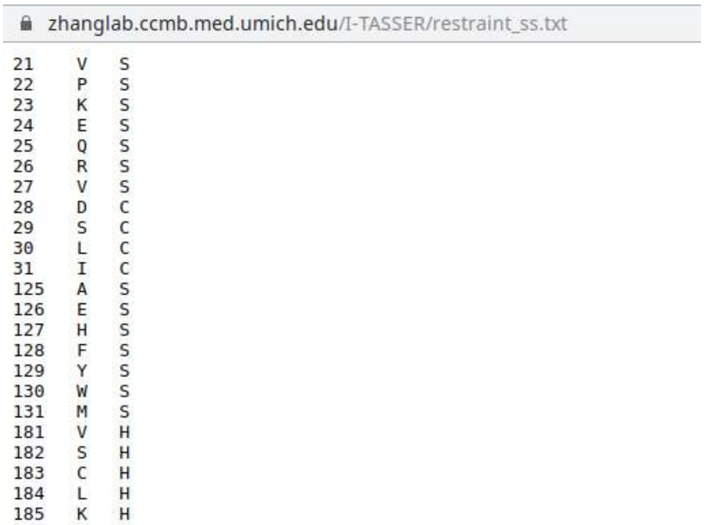


Figura 27. Exemplo de arquivo texto das restrições de contato e distância.

Opção II – É possível excluir moldes parecidos à proteína-alvo presentes no banco de dados do servidor, inserindo um valor de corte. Por exemplo, ao digitar "60%", o I-TASSER excluirá automaticamente todos os modelos que possuem uma identidade de sequência maior que 60%. O corte mínimo é definido em 25%, ou seja, todos os valores abaixo de 25% retornarão como 25%. Porém, a exclusão de moldes com identidade de sequência diminuirá a qualidade da modelagem. Portanto, essa opção foi projetada apenas para alguns fins especiais. Ainda nessa opção, moldes específicos podem ser excluídos através de uma lista de estruturas no formato PDBID:Chain.

Opção III – Caso algum conhecimento sobre a estrutura secundária da proteína a ser modelada exista, como por exemplo, informações extraídas de uma predição de estrutura secundária, um arquivo de texto (Figura 28) com essas informações pode ser inserido. O I-TASSER tentará gerar os modelos seguindo a estrutura secundária especificada no arquivo. O arquivo consiste em uma coluna com o número do resíduo, uma segunda coluna com o símbolo do resíduo, e uma terceira coluna com o tipo de estrutura

secundária, onde S corresponde às folhas-beta, H às hélices-alfa, e C às alças (*loops*).


zhanglab.ccmb.med.umich.edu/I-TASSER/restraint_ss.txt

21	V	S
22	P	S
23	K	S
24	E	S
25	Q	S
26	R	S
27	V	S
28	D	C
29	S	C
30	L	C
31	I	C
125	A	S
126	E	S
127	H	S
128	F	S
129	Y	S
130	W	S
131	M	S
181	V	H
182	S	H
183	C	H
184	L	H
185	K	H

Figura 28. Exemplo de arquivo texto para a especificação de estrutura secundária.

A sequência do peptídeo da glândula salivar de *Ixodes scapularis* foi submetida sem a utilização de nenhuma restrição. Após a submissão, o usuário é notificado por e-mail assim que o trabalho for finalizado. Os resultados ficam disponíveis no servidor durante um período de dois meses. A modelagem leva de um a dois dias para processar a depender da fila de trabalhos submetidos no servidor web. Na figura 29, podemos visualizar os resultados obtidos para a sequência submetida. Nessa página, consta o resultado da predição de estrutura secundária e o *conf.score* que demonstra a qualidade da predição para cada resíduo em relação a sequência alvo. Quanto maior o valor, maior a confiabilidade da predição.

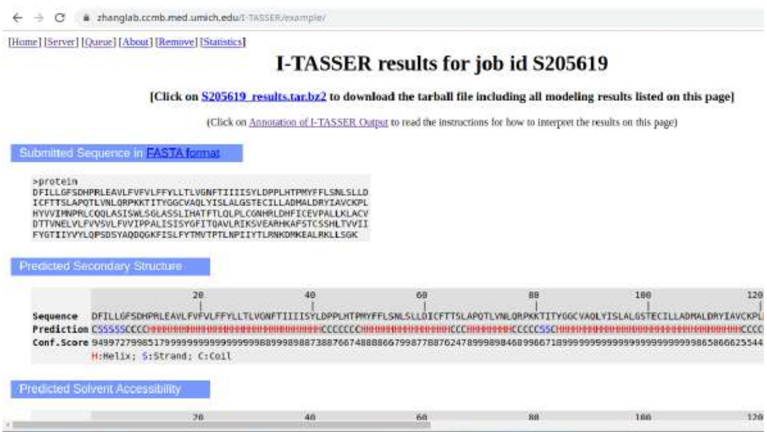


Figura 29. Página de resultado do I-TASSER para o exemplo submetido.

A predição de acessibilidade ao solvente de cada resíduo também é feita. Nesse caso, I-TASSER classifica os resíduos como acessíveis ou inacessíveis

ao solvente (Figura 30). Valores próximos a zero indicam que os resíduos possivelmente estão em posições internas na proteína, enquanto valores próximos a nove indicam resíduos em posições mais expostas ao solvente. Além disso, o fator B, valor que indica a extensão da mobilidade térmica inerente de resíduos ou átomos nas proteínas, também é predito para o modelo (Figura 31). Resíduos com valores negativos no gráfico de fator B mostram ser mais estáveis na estrutura.

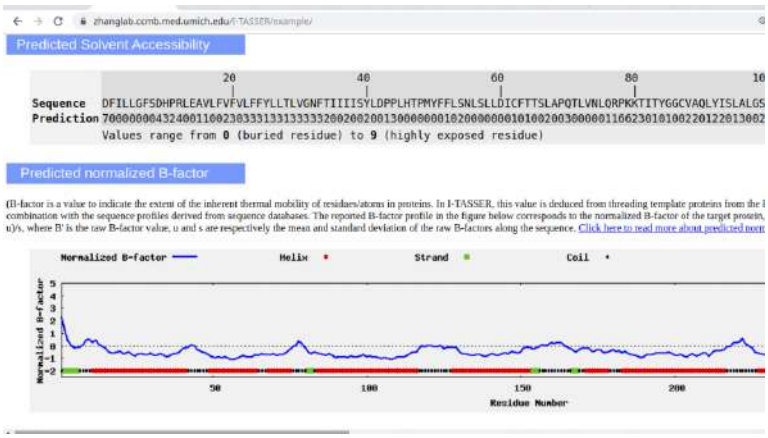


Figura 30. Resultados da predição de acessibilidade ao solvente dos resíduos e o fator B em relação a toda a estrutura do modelo criado.

O alinhamento dos dez primeiros moldes também é disponibilizado na página de resultados (Figura 31). Espera-se encontrar uma maior conservação nos moldes exibidos, o que poderá dar uma maior qualidade ao modelo final. A avaliação do alinhamento entre as sequências molde e alvo pode ser feita pelo parâmetro *Norm. Z-score*. Valores de *Norm. Z-score* acima de um revelam um bom alinhamento entre sequências. Dependendo desses valores, I-TASSER qualifica a proteína-alvo como fácil ou difícil de modelar. Além disso, tanto os alinhamentos quanto as estruturas-molde podem ser baixadas individualmente.

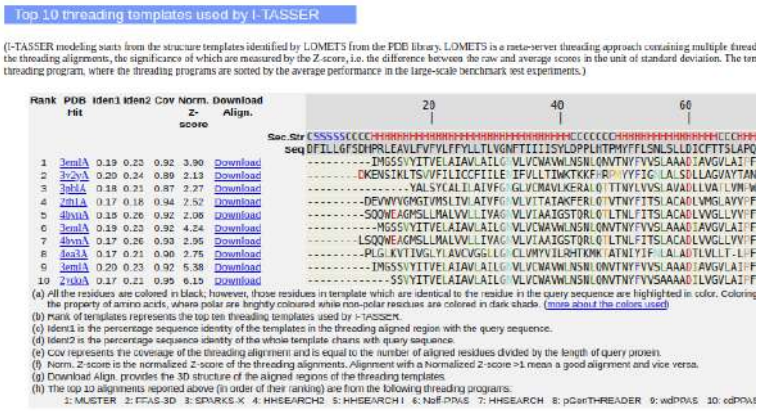


Figura 31. Resultado do alinhamento com os dez primeiros moldes encontrados pelo I-TASSER usando LOMETS e a base de dados de PDB.

Por fim, cinco modelos construídos mais bem ranqueados são apresentados (Figura 32). Além da opção de *download* dos modelos gerados, informações, como precisão global (*C-score*), *TM-score* e RMSD, ficam disponíveis para melhor qualificar os modelos. Porém, o I-TASSER relata apenas a previsão de *TM-score* e RMSD para o primeiro modelo, uma vez que a correlação entre *C-score* e *TM-score* é fraca para modelos de classificação inferior.

O valor de *C-score* é listado para todos os modelos para servir como referência. O *C-score* (precisão global estimada) possui uma faixa de valor entre -5 e 2. Valores maiores que -1,5 indicam modelos que possuem uma boa topologia global predita. Já o *TM-score* [33] é uma escala proposta para medir a semelhança estrutural entre duas estruturas, nesse caso a estrutura do molde e do modelo, que não depende do comprimento da proteína e não é sensível a diferentes estruturações e orientações locais. Valores de *TM-score* acima de 0,50 indicam um modelo na topologia correta, enquanto valores de *TM-score* abaixo de 0,17 significam que a similaridade entre as estruturas é aleatória. O RMSD se refere a sobreposição entre molde e modelo gerado. Um valor alto de RMSD mostra que regiões específicas das proteínas possuem estruturação e orientação desiguais. Se existente, o I-TASSER pode ainda disponibilizar os possíveis ligantes, possíveis sítios ativos e possíveis funções das estruturas modeladas.

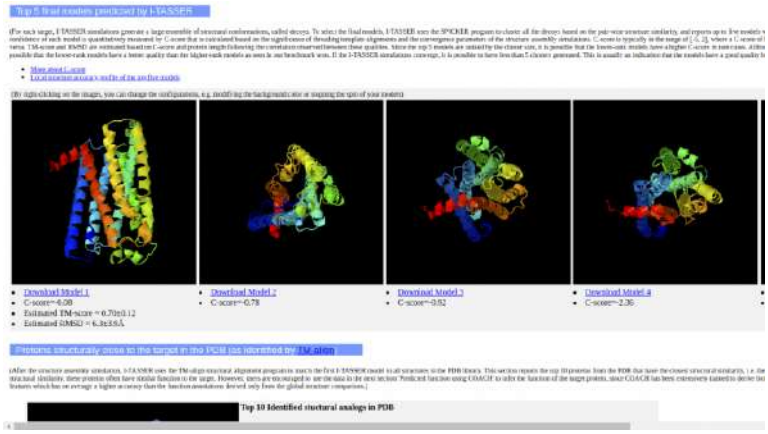


Figura 32. Os cinco melhores modelos ranqueados de acordo com as funções objetivas do I-TASSER.

Os modelos podem ainda ser avaliados usando métricas de avaliação para modelagem comparativa. Porém, como se trata de um modelo construído a partir de sequências de baixa identidade, sua qualidade deve ser considerada baixa, logo tais modelos poderão receber pontuações baixas.

Métodos de modelagem independentes de molde

Devido à grande lacuna entre o número de estruturas primárias e estruturas tridimensionais resolvidas, uma quantidade significativa de dados de sequência não compartilha identidade e similaridade com famílias de proteínas conhecidas. Com isso, surge a necessidade de métodos que predizem a estrutura com nenhuma ou mínima informação estrutural, os chamados de métodos independentes de molde. Esse tipo de modelagem baseia-se na suposição que todas as proteínas se enovelam para um estado nativo ou para um conjunto de estados com o menor nível de energia potencial, mínimo global [29,34]. Existem duas abordagens para essa categoria, modelagem *de novo* e modelagem *ab initio*.

Apesar de serem tratadas como equivalentes na literatura, na prática os algoritmos desses métodos diferem em suas aplicações. Na modelagem *de novo* são usadas informações provenientes de bancos de estruturas determinadas empiricamente, em forma de fragmentos estruturais sem identidade com a sequência alvo, para orientar o estado enovelado do modelo. Enquanto, métodos *ab initio* baseiam-se puramente nas leis da Física, ou seja, primeiros princípios, para determinar as estruturas. Nas abordagens *ab initio* o conhecimento estrutural de proteínas como a previsão de ângulos de torção e inserção dos átomos são feitos através de métodos matemáticos e estatísticos. Porém, ambas metodologias são computacionalmente exigentes, limitando a modelar proteínas pequenas (entre 100 e 200 aminoácidos). Além disso, para a escolha dos melhores modelos, diferente do que é realizado na modelagem comparativa, é necessário realizar a execução dos algoritmos muitas vezes. Dessa forma a definição dos melhores modelos é feita inicialmente pela filtragem de vária conformações a partir de um limiar de energia previamente definido.

Atualmente, as ferramentas de predição estrutural utilizam vários métodos na construção do modelo, tornando-se ferramentas híbridas de modelagem. Por exemplo, o servidor ROBETTA (robetta.bakerlab.org/) usa fragmentos de estruturas PDB existentes, a fim de orientar a pesquisa em conjunto com funções de energia, classificando-se então como um software de modelagem *de novo* [34,35]. Porém, regiões da sequência sem equivalência são construídas a partir de modelagem *ab initio*. Podemos dizer o mesmo do programa QUARK (zhanglab.ccmb.med.umich.edu/QUARK/), pois sua abordagem possui uma etapa de montagem de fragmentos, onde pequenos fragmentos estruturais (1–20 resíduos retirados de estruturas PDB conhecidas) são unidos para construir a estrutura final por Monte Carlo com assistência de um campo de força [36].

Apesar da diferença na forma como os algoritmos de modelagem *de novo* e *ab initio* são implementados, essas terminologias têm sido usadas na literatura como sinônimos. Por isso, neste artigo utilizaremos tanto os termos *de novo* quanto *ab initio* para indicar estratégias de modelagem sem o uso de molde.

Tutorial ROBETTA

Modelagem de proteínas *ab initio* com ROBETTA

A seguir, vamos utilizar o servidor web ROBETTA (http://robetta.bakerlab.org) para a modelagem de estruturas proteicas (Figura 33). O servidor utiliza a implementação automatizada do programa ROSETTA (https://www.rosettacommons.org/) no qual é possível realizar tanto modelagem comparativa quanto *ab initio*. A metodologia ROSETTA baseia-se em dividir a sequência em fragmentos de tamanho entre três e nove aminoácidos. Os segmentos são extraídos da sequência de entrada e comparados com segmentos de uma base de dados de estrutura de proteínas, a partir de suas estruturas secundárias. Em seguida o espaço conformacional é então pesquisado utilizando a metodologia de Monte Carlo, que consiste em realizar um massivo número de simulações com amostragem aleatória. Dessas simulações, um valor de energia é estabelecido através do campo de força do programa [35,37].

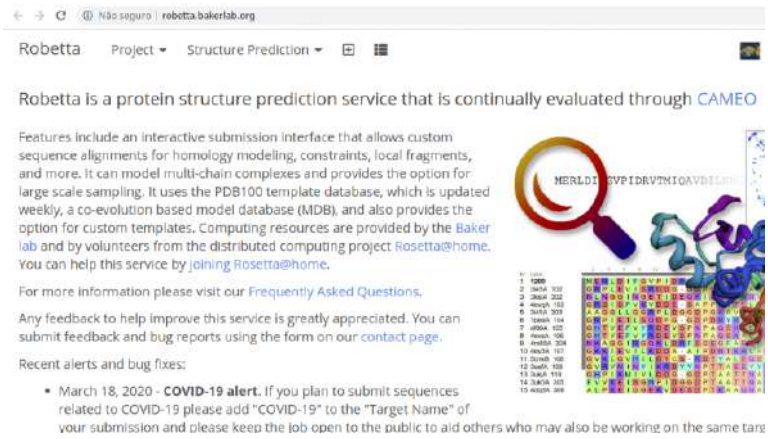


Figura 33. Página inicial do servidor web ROBETTA.

Para cada sequência de destino são geradas 10.000 conformações, futuramente agrupadas com base no RMSD de seus carbonos-alfa. Apenas nove centroides, estruturas representativas de cada agrupamento, são selecionados. Na última etapa, os modelos gerados são buscados no PDB

utilizando o *Mammoth*, um algoritmo estrutural que independe da sequência para encontrar à sobreposição estrutural com maior cobertura. Essa comparação tem o intuito de aumentar a confiabilidade do enovelamento do modelo através de um valor representado pela função de confiança. Quanto maior o valor de confiança, maior é a correspondência estrutural do modelo com estruturas existentes [38].

Link de acesso ao ROBETTA: http://robetta.bakerlab.org

Para utilizar o ROBETTA é necessário criar uma conta gratuita (Figura 34). Nesse processo é criada um usuário para *login* e uma senha, que serão utilizados para acessar os trabalhos que forem submetidos e acompanhar o *status* na fila de execução.

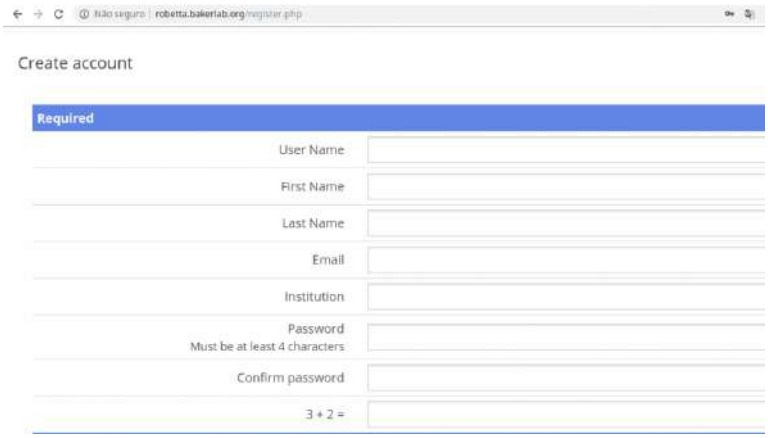


Figura 34. Página de registro e criação de login do ROBETTA.

Depois de criar *login* e senha, o usuário poderá submeter a sequência que tem interesse em modelar. A entrada pode ser submetida inserindo a sequência no local indicado ou fazendo um upload da sequência em formato FASTA (Figura 35). O servidor realiza a modelagem comparativa e *ab initio* de forma automática. Entretanto, é possível selecionar a opção *CM only* para realizar apenas a modelagem comparativa. Ou ainda, selecionar a opção *AB only*, que realiza apenas a modelagem *ab initio.* A opção *predict domains* permite que os domínios da proteína sejam resolvidos separadamente, porém isso implica em um maior tempo de execução. Quando nenhuma das opções é selecionada, o programa tenta realizar a modelagem comparativa primeiro e, caso não seja possível, a modelagem *ab initio* é executada.

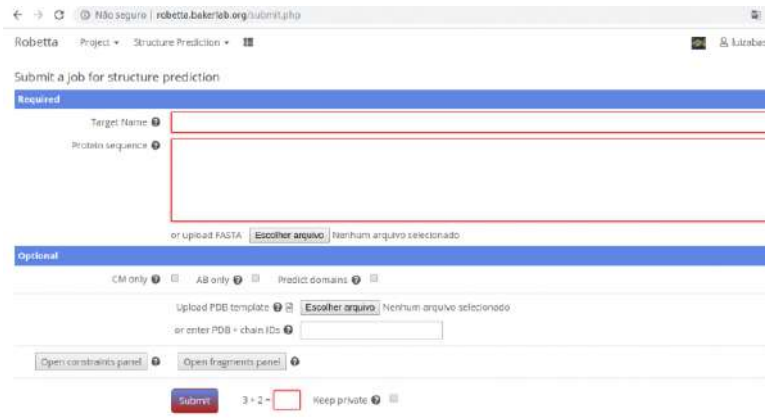


Figura 35. Tela de submissão de sequência para a modelagem do ROBETTA.

Na execução do ROBETTA como programa de modelagem comparativa, o usuário pode inserir o código PDB do molde ou o arquivo do molde que deseja usar. Restrições de ângulos ou distâncias entre dois átomos ou resíduos podem ser inseridos para influenciar a função de energia utilizada. Também é possível inserir um arquivo de fragmentos para serem utilizados na modelagem *ab initio.*

Após realizar a submissão, é possível acompanhar a fila de execução e o status da construção do modelo enviado clicando no menu superior ao lado do seu nome de usuário e, em seguida, na opção "M*y queue*". Quando finalizada a modelagem, o usuário receberá uma notificação via e-mail. O tempo de espera varia, em média, entre dois e três dias, sem a opção *predict domains* (ao selecionar essa opção o tempo de execução é estendido).

Estudo de caso: glicoproteína de superfície do SARS-COV-2

Como exemplo, a sequência da glicoproteína de superfície do SARS-COV-2 (G*enbank* ID: QIU81369.1) foi submetida a modelagem na ferramenta web ROBETTA. Essa proteína apresenta 1261 resíduos de aminoácidos. Na Figura 36, pode-se visualizar informações como a sequência alvo, parâmetro de confiança (*confidence*) e o método utilizado para modelar a glicoproteína de superfície do SARS-COV-2.

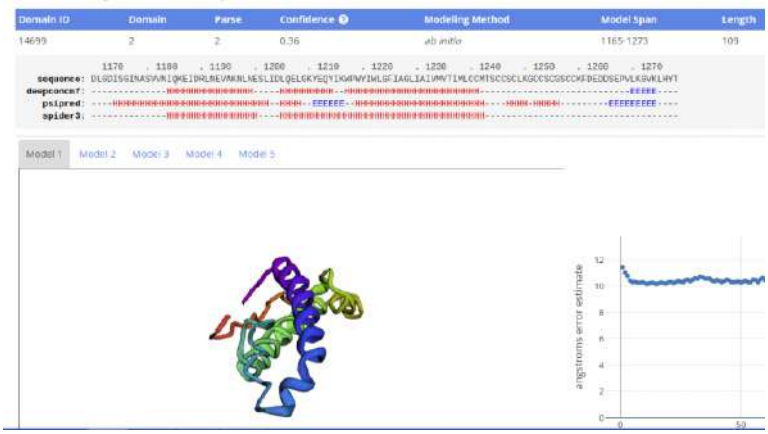


Figura 36. Página de resultado (parte superior) da construção de modelos pelo ROBETTA.

O valor do parâmetro de confiança varia entre zero e um. Quanto mais próximo de um, melhor é a qualidade dos modelos gerados. Valores mais próximo de zero indicam que a qualidade dos modelos é baixa. Para o nosso exemplo, a confiança obtida foi de 0,36, demonstrando uma qualidade inferior à desejada.

Além disso, são apresentadas as previsão de estrutura secundária realizadas por três ferramentas:

- ***deepconcnf*** (raptorx.uchicago.edu/StructurePropertyPred/predict/);
- ***psipred*** (bioinf.cs.ucl.ac.uk/psipred/);
- ***spider3*** (sparks-lab.org/server/spider3/).

onde H representa hélices-alfa, E representa folhas-beta, e o caractere "-" representa regiões de alça.

É possível visualizar os cinco melhores modelos de estrutura (Figura 37), além de um gráfico com a estimativa de erro em ångström para cada resíduo. Com esse gráfico, é possível ver a variação das posições dos resíduos de acordo com cada modelo. Variações muito grandes mostram a dificuldade de modelar certas regiões. Os resultados podem ser baixados (a ferramenta informa a data até quando esses resultados ficarão disponíveis no servidor).

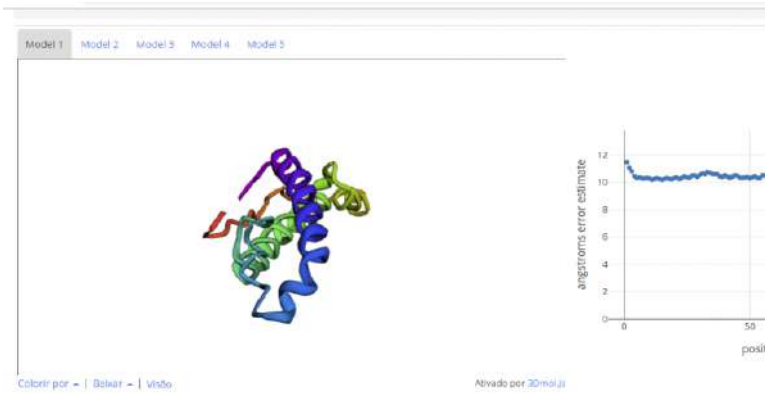


Figura 37. Parte da página de resultados do servidor web ROBETTA.

Referências bibliográficas

1. Wolynes, P.G. Evolution, Energy Landscapes and the Paradoxes of Protein Folding. *Biochimie* **2015**, *119*, 218–230.

2. Schwede, T. Protein Modeling: What Happened to the “Protein Structure Gap”? *Structure* **2013**, *21*, 1531–1540.

3. Carroni, M.; Saibil, H.R. Cryo Electron Microscopy to Determine the Structure of Macromolecular Complexes. *Methods (San Diego, Calif.)* **2016**, *95*, 78–85, doi:10.1016/j.ymeth.2015.11.023.

4. Klebe, G. Experimental Methods of Structure Determination. In *Drug Design: Methodology, Concepts, and Mode-of-Action*; Klebe, G., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013; pp. 265–290 ISBN 978-3-642-17907-5.

5. Berman, H.M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.N.; Weissig, H.; Shindyalov, I.N.; Bourne, P.E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.

6. Consortium, U. UniProt: A Worldwide Hub of Protein Knowledge. *Nucleic acids research* **2019**, *47*, D506–D515.

7. Studer, G.; Tauriello, G.; Bienert, S.; Waterhouse, A.M.; Bertoni, M.; Bordoli, L.; Schwede, T.; Lepore, R. Modeling of protein tertiary and quaternary structures based on evolutionary information. In *Computational Methods in Protein Evolution*; Springer, 2019; pp. 301–316.

8. Liu, H.; Chen, Q. Computational Protein Design for given Backbone: Recent Progresses in General Method-Related Aspects. *Current opinion in*

structural biology **2016**, *39*, 89–95.

9. Haddad, Y.; Adam, V.; Heger, Z. Ten Quick Tips for Homology Modeling of High-Resolution Protein 3D Structures. *PLoS computational biology* **2020**, *16*, e1007449.

10. Kc, D.B. Recent Advances in Sequence-Based Protein Structure Prediction. *Briefings in bioinformatics* **2017**, *18*, 1021–1032.

11. Patel, B.; Singh, V.; Patel, D. Structural Bioinformatics. In *Essentials of Bioinformatics, Volume I*; Springer, 2019; pp. 169–199.

12. Browne, W.J.; North, A.C.T.; Phillips, D.C.; Brew, K.; Vanaman, T.C.; Hill, R.L. A Possible Three-Dimensional Structure of Bovine α-Lactalbumin Based on That of Hen's Egg-White Lysozyme. *Journal of molecular biology* **1969**, *42*, 65–86.

13. Cavasotto, C.N.; Phatak, S.S. Homology Modeling in Drug Discovery: Current Trends and Applications. *Drug discovery today* **2009**, *14*, 676–683.

14. Šali, A.; Blundell, T.L. Comparative Protein Modelling by Satisfaction of Spatial Restraints. *Journal of molecular biology* **1993**, *234*, 779–815.

15. Ginalski, K. Comparative Modeling for Protein Structure Prediction. *Current opinion in structural biology* **2006**, *16*, 172–177.

16. Baker, D.; Sali, A. Protein Structure Prediction and Structural Genomics. *Science* **2001**, *294*, 93–96.

17. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic Local Alignment Search Tool. *Journal of molecular biology* **1990**, *215*, 403–410.

18. Schwede, T.; Kopp, J.; Guex, N.; Peitsch, M.C. SWISS-MODEL: An Automated Protein Homology-Modeling Server. *Nucleic acids research* **2003**, *31*, 3381–3385.

19. Greer, J. Comparative Modeling Methods: Application to the Family of the Mammalian Serine Proteases. *Proteins: Structure, Function, and Bioinformatics* **1990**, *7*, 317–334.

20. Blundell, T.L.; Sibanda, B.L.; Sternberg, M.J.E.; Thornton, J.M. Knowledge-Based Prediction of Protein Structures and the Design of Novel Molecules. *Nature* **1987**, *326*, 347–352.

21. Wallner, B.; Elofsson, A. All Are Not Equal: A Benchmark of Different Homology Modeling Programs. *Protein Science* **2005**, *14*, 1315–1327.

22. Waterhouse, A.; Bertoni, M.; Bienert, S.; Studer, G.; Tauriello, G.; Gumienny, R.; Heer, F.T.; de Beer, T.A.P.; Rempfer, C.; Bordoli, L. SWISS-MODEL: Homology Modelling of Protein Structures and Complexes. *Nucleic acids research* **2018**, *46*, W296–W303.

23. Wang, J.; Lee, P.K.; Dong, Y.; Pang, S.S.; Duggleby, R.G.; Li, Z.; Guddat, L.W. Crystal Structures of Two Novel Sulfonylurea Herbicides in Complex with Arabidopsis Thaliana Acetohydroxyacid Synthase. *The FEBS journal* **2009**, *276*, 1282–1290.

24. Santos Filho, O.A.; Alencastro, R.B. de Modelagem de Proteínas Por Homologia. *Química Nova* **2003**, *26*, 253–259.

25. Higgins, D.G.; Sharp, P.M. Fast and Sensitive Multiple Sequence Alignments on a Microcomputer. *Bioinformatics* **1989**, *5*, 151–153.

26. Xiang, Z. Advances in Homology Protein Structure Modeling. *Current Protein and Peptide Science* **2006**, *7*, 217–227.

27. Schrödinger, L.L.C. The PyMOL Molecular Graphics System, Version 2.0 2020.

28. Zhang, Y. I-TASSER: Fully Automated Protein Structure Prediction in CASP8. *Proteins: Structure, Function, and Bioinformatics* **2009**, *77*, 100–113.

29. Verli, H. Bioinformática: Da Biologia à Flexibilidade Molecular. **2014**.

30. Yang, J.; Zhang, Y. I-TASSER Server: New Development for Protein Structure and Function Predictions. *Nucleic acids research* **2015**, *43*, W174–W181.

31. Wu, S.; Zhang, Y. LOMETS: A Local Meta-Threading-Server for Protein Structure Prediction. *Nucleic acids research* **2007**, *35*, 3375–3382.

32. Yang, J.; Roy, A.; Zhang, Y. BioLiP: A Semi-Manually Curated Database for Biologically Relevant Ligand–Protein Interactions. *Nucleic acids research* **2012**, *41*, D1096–D1103.

33. Zhang, Y.; Skolnick, J. Scoring Function for Automated Assessment of Protein Structure Template Quality. *Proteins: Structure, Function, and Bioinformatics* **2004**, *57*, 702–710.

34. Kim, D.E.; Chivian, D.; Baker, D. Protein Structure Prediction and Analysis Using the Robetta Server. *Nucleic acids research* **2004**, *32*, W526–W531.

35. Song, Y.; DiMaio, F.; Wang, R.Y.-R.; Kim, D.; Miles, C.; Brunette, T.J.; Thompson, J.; Baker, D. High-Resolution Comparative Modeling with RosettaCM. *Structure* **2013**, *21*, 1735–1742.

36. Xu, D.; Zhang, Y. Ab Initio Protein Structure Assembly Using Continuous Structure Fragments and Optimized Knowledge-based Force Field. *Proteins: Structure, Function, and Bioinformatics* **2012**, *80*, 1715–1735.

37. Bradley, P.; Chivian, D.; Meiler, J.; Misura, K.M.S.; Rohl, C.A.; Schief, W.R.; Wedemeyer, W.J.; Schueler-Furman, O.; Murphy, P.; Schonbrun, J. Rosetta Predictions in CASP5: Successes, Failures, and Prospects for Complete Automation. *Proteins: Structure, Function, and Bioinformatics* **2003**, *53*, 457–468.

38. Chivian, D.; Kim, D.E.; Malmström, L.; Schonbrun, J.; Rohl, C.A.; Baker, D. Prediction of CASP6 Structures Using Automated Robetta Protocols. *Proteins: Structure, Function, and Bioinformatics* **2005**, *61*, 157–166.