Introduction to HMMER: Profile Hidden Markov Models for Homology Search

Authored by: Hazel Li, Sean Liu, Pranav Siva

Table of Contents

Overview

With the explosive growth of biological sequence databases, homology search has become popular for inferring protein function and evolutionary relationships. Although BLAST has long been the dominant solution, its local alignment framework and manually tuned gap penalties limit both sensitivity and compatibility with more advanced probabilistic scoring models such as profile HMMs [1].

Profile Hidden Markov Models (profile HMMs) provide a statistical and position-specific framework for modeling amino acid conservation, substitutions, and insertions/deletions, capturing evolutionary patterns present in multiple sequence alignments [2]. By computing log-likelihood ratios relative to a background model, profile HMMs allow more sensitive detection of remote homology compared to BLAST's heuristic local scoring approach.

HMMER is a widely used software suite that implements profile HMMs for homology search. It enables researchers to build profile HMMs from multiple sequence alignments and use them to search sequence databases for homologous sequences or to annotate functional domains within query sequences. HMMER's algorithms are optimized for speed and accuracy, making it a powerful tool for bioinformatics applications. In this section, we will introduce the fundamental concept of profile HMMs and provide an overview of the HMMER workflow.

Profile Hidden Markov Model (profile HMM)

Before diving into the HMMER suite, it's necessary to understand the underlying data structure it uses: the **profile Hidden Markov Model** (profile HMM). It is a generative model built from the Multiple Sequence Alignment (MSA) of the family. To measure how well a query sequence fits the family, we can use the probability of the profile HMM generating the sequence.

Structure of Profile HMM

Let's look at the structure through an example profile HMM. The main body is the automaton, built from the multiple sequence alignment above it. (note: the family contains more sequences than the four shown). It consists of nodes called states and arrows denoting unidirectional transitions between states.

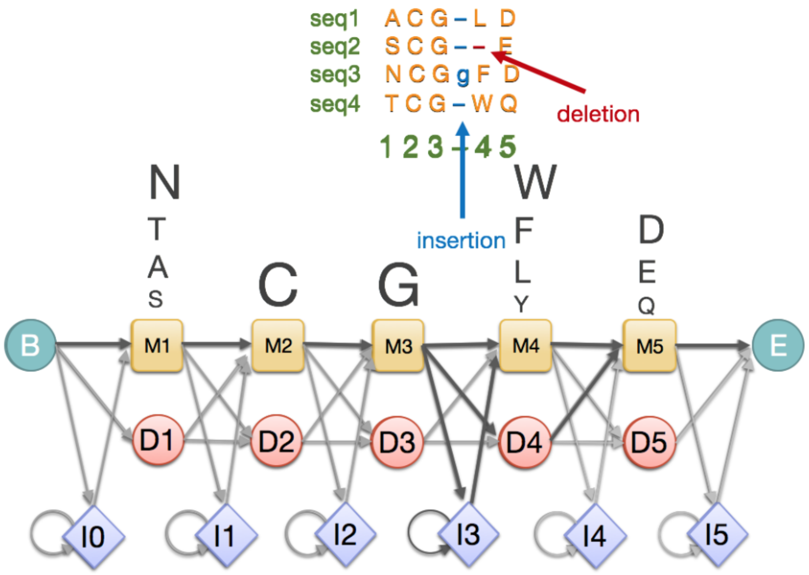


Figure 1. Example Profile HMM structure. Image source: European Bioinformatics Institute (EMBL-EBI), Pfam Training Course [3].

States

The profile HMM is centered around the linear set of **match (M) states** [4], in between the beginning (B) state and the end (E) state. Each match state corresponds to a ***conserved*** MSA column, where most sequences contain a valid residue at that position, in contrast to a ***gappy*** column. Upon visiting the state, it will emit one residue from the set of characters (20 amino acids for proteins and nucleotides for DNA/RNA sequences). The **emission probability** is calculated from the residue frequency distribution within the MSA column it represents.

Additional states are used to account for gaps in the MSA. Each **deletion (D) state** is connected across a match state, and allows skipping the bounded match state and emits a "-" instead of a valid residue. It accounts for gaps ***within*** conserved MSA columns, as illustrated by the red arrow in the diagram. **Insertion (I) state** accounts for gaps of variable length, and represents the extra residues ***between*** conserved MSA columns (indicated by the blue arrow).

Transition and Other Parameters

To move between states, we also need **transition probabilities**, the probabilities of moving from current states to the neighboring states downstream. For example, if we are at `M3`, we will have probabilities

> P1 = P(`M3` → `M4`)
> P2 = P(`M3` → `D4`)
> P3 = P(`M3` → `I3`)

The full set of parameters used in profile HMM is listed below:

- Σ: the set of symbols
- Q: the set of states
- e_j(S): the probability of emitting residue S while in state j
- T: the matrix of transition probabilities
  - T[j,k]: probability of moving from state j to state k
- π: the probability distribution on the initial state

Path and Probability

Consider an example string `S` = `"TCLD"`. Our profile HMM `M` can generate this sequence by following multiple paths. One possible path is:

> `PATH` = `B` → `M1` → `M2` → `D3` → `M4` → `M5` → `E`
>
> `T` → `C` → `-` → `L` → `D`

The probability of generating this sequence by following the path is the product of emission probabilities of each match state emitting the corresponding residue:

> **P**(`S` | `PATH`, `M`) = e_{M1}[T] * e_{M2}[C] * e_{M4}[L] * e_{M5}[D]

The probability of traversing through this path in our profile HMM is the product of transition probabilities:

> **P**(`PATH` | `M`) = π[`M1`] * T[`M1`, `M2`] * T[`M2`, `D3`] * T[`D3`, `M4`] * T[`M4`, `M5`]

And the overall probability of the model generating the sequence `S` through this path is:

> **P**(`S` | `PATH`, `M`) = **P**(`S` | `PATH`, `M`) * **P**(`PATH` | `M`)

Among all possible paths that can generate the sequence `S`, we can either sum over the probabilities of all possible paths, or find the max probability (Viterbi path) among them. Both algorithms are supported in HMMER, and this final probability is used to evaluate how well the sequence fits the profile HMM.

HMMER Workflow

The effective use of HMMER relies on understanding its workflow. Unlike simple pairwise alignment tools that function as a single step (sequence A vs. sequence B), HMMER operates as a multi-stage pipeline. This pipeline transforms raw biological data into a statistical model, which is then utilized to either discover new homologs in sequence databases or to annotate functional domains by querying profile databases.

This section lists out HMMER workflow that follows a linear progression consisting of **four distinct stages**:

1. Input

To get started, we need to input a high-quality **Multiple Sequence Alignment (MSA)**. HMMER cannot build a profile from a single sequence; it requires the evolutionary context provided by a family of aligned sequences to determine which residues are conserved and which are variable.

2. Profile Construction

This stage uses `hmmbuild` command to convert the biological alignment into a mathematical Profile HMM. It implements the theoretical concepts of **Match, Insert, and Delete** states to model the evolutionary constraints of the family.

3. Application: The Search Strategies

Once the profile is built, the workflow can go into two directions based on the research objective:

- **Homology Search (`hmmsearch`)**: Used for discovery. It takes the custom profile HMM and scans a target sequence database to find new homologs.
- **Domain Annotation (`hmmscan`):** Used for identification. It takes a single query sequence and scans it against a library of profile HMMs to identify functional domains within the sequence.

4. Output: Ranked Results

Both strategies generate a list of matches. HMMER filters biological signals from random noise, presenting hits ranked by bit score and e-value.

HMMER Tools

While the workflow provides the roadmap, understanding the specific mechanics of each tool is essential for effective analysis. This section details the primary tools used in HMMER analysis: `hmmbuild`, `hmmsearch` and `hmmscan` [5].

`hmmbuild`: Profile Construction

`hmmbuild` creates a profile from an alignment. It is responsible for transforming the evolutionary information contained in a multiple sequence alignment into a binary-compatible statistical model.

Input The input for this command is a Multiple Sequence Alignment (MSA). The software is designed to accept several standard bioinformatics alignment formats, including Stockholm (`.sto`), Clustal (`.aln`), or aligned FASTA.

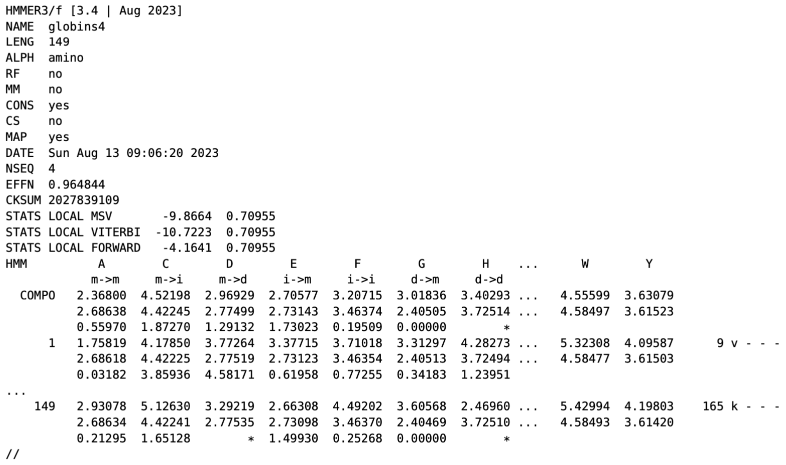
Process The tool reads the alignment column-by-column and applies the Profile HMM statistical architecture (calculating Match, Insert, and Delete state probabilities) as explained in the previous section. It converts the observed biological data into a probabilistic model without requiring manual parameterization.

Output The command generates a Profile HMM file (ending with `.hmm` extension). This serves as a binary-compatible file that contains all information needed for later search strategies.

Usage The command requires the user to specify an output filename followed by the input alignment. For example, if you want to create a new profile HMM named `hemoglobin.hmm` from the source alignment `hemoglobin.sto`, you can use the command

```
# Syntax: hmmbuild [output_hmm_file] [input_msa_file]
hmmbuild hemoglobin.hmm hemoglobin.sto
```

Below is an example of output HMM file. It begins with a header section containing metadata such as the family name, length, and type (amino acid or nucleotide). This is followed by the main model section, which contains a matrix of position-specific log-odds scores for match, insert, and delete states.



```
HMMER3/f [3.4 | Aug 2023]
NAME  globins4
LENG  149
ALPH  amino
RF    no
MM    no
CONS  yes
CS    no
MAP   yes
DATE  Sun Aug 13 09:06:20 2023
NSEQ  4
EFFN  0.964844
CKSUM 2027839109
STATS LOCAL MSV       -9.8664  0.70955
STATS LOCAL VITERBI  -10.7223  0.70955
STATS LOCAL FORWARD   -4.1641  0.70955
HMM          A        C        D        E        F        G        H    ...      W        Y
            m->m     m->i     m->d     i->m     i->i     d->m     d->d
  COMPO   2.36800  4.52198  2.96929  2.70577  3.20715  3.01836  3.40293 ...   4.55599  3.63079
          2.68638  4.42245  2.77499  2.73143  3.46374  2.40505  3.72514 ...   4.58497  3.61523
          0.55970  1.87270  1.29132  1.73023  0.19509  0.00000        *
      1   1.75819  4.17850  3.77264  3.37715  3.71018  3.31297  4.28273 ...   5.32308  4.09587      9 v - - -
          2.68618  4.42225  2.77519  2.73123  3.46354  2.40513  3.72494 ...   4.58477  3.61503
          0.03182  3.85936  4.58171  0.61958  0.77255  0.34183  1.23951
...
    149   2.93078  5.12630  3.29219  2.66308  4.49202  3.60568  2.46960 ...   5.42994  4.19803    165 k - - -
          2.68634  4.42241  2.77535  2.73098  3.46370  2.40469  3.72510 ...   4.58493  3.61420
          0.21295  1.65128        *  1.49930  0.25268  0.00000        *
//
```

Figure 2. Example output of `hmmbuild`. Image source: HMMER User's Guide [5].

`hmmsearch`: Homology Search

Once a profile has been constructed using `hmmbuild`, the most common tool to use is **`hmmsearch`**. It is designed to find homologs in a large sequence database using your profile HMM, providing a highly sensitive method for detecting homologs that traditional tools might miss.

Input `hmmsearch` requires two inputs.

1. Query: the **Profile HMM** file generated by `hmmbuild`.
2. Target: a **sequence database**. It can be a standard FASTA file containing protein sequences, such as a whole genome or a comprehensive repository like UniProt.

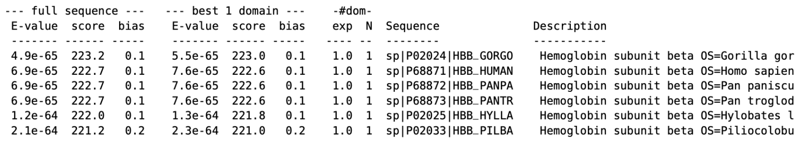
Process The software performs a "Profile v.s. Sequence" search. It compares the statistical pattern of the profile against every sequence in the target database to detect **remote homologs**, which are sequences that have diverged significantly in primary structure but retain the essential structural or functional constraints of the family.

Output The software produces a ranked list of hits as output. These hits are sorted based on e-value and bit score, which measures the statistical significance and raw quality of the match. The output also includes the specific alignments showing where the model matched the target sequences.

Usage To execute a search, the user runs the command followed by the profile file and the target sequence database. For example, if you want to search the `uniprot.fasta` database using the `hemoglobin.hmm` profile and print the results to the standard output, you can use the command

```
# Syntax: hmmsearch [query_hmm_file] [target_msa_file] > [output_file]
hmmsearch hemoglobin.hmm uniprot.fasta > hemoglobin.out
```

Below is an example of the output list. This list is ranked by the "E-value" column, with the most statistically significant matches appearing at the top. Beside this, the "score" column provides the bit score, which is a standardized metric derived from the alignment's raw score. The "sequence" column indicates the matched sequence identifier, and the "Description" column shows a brief description of the sequence. This structure allows users to quickly filter low-quality hits and focus on the most biologically relevant alignments.



```
  --- full sequence ---    --- best 1 domain ---    -#dom-
   E-value  score  bias    E-value  score  bias    exp  N  Sequence             Description
   ------- ------ -----    ------- ------ -----   ---- --  --------             -----------
   4.9e-65  223.2   0.1    5.5e-65  223.0   0.1    1.0  1  sp|P02024|HBB_GORGO   Hemoglobin subunit beta OS=Gorilla gor
   6.9e-65  222.7   0.1    7.6e-65  222.6   0.1    1.0  1  sp|P68871|HBB_HUMAN   Hemoglobin subunit beta OS=Homo sapien
   6.9e-65  222.7   0.1    7.6e-65  222.6   0.1    1.0  1  sp|P68872|HBB_PANPA   Hemoglobin subunit beta OS=Pan paniscu
   6.9e-65  222.7   0.1    7.6e-65  222.6   0.1    1.0  1  sp|P68873|HBB_PANTR   Hemoglobin subunit beta OS=Pan troglod
   1.2e-64  222.0   0.1    1.3e-64  221.8   0.1    1.0  1  sp|P02025|HBB_HYLLA   Hemoglobin subunit beta OS=Hylobates l
   2.1e-64  221.2   0.2    2.3e-64  221.0   0.2    1.0  1  sp|P02033|HBB_PILBA   Hemoglobin subunit beta OS=Piliocolobu
```

Figure 3. Example output of `hmmsearch`. Image source: HMMER User's Guide [5].

`hmmscan`: Domain Annotation

`hmmscan` is a tool used to search protein sequences against a database of profile HMMs. It can identify known domains present within the query sequences, which makes it an essential software for domain annotation.

Input `hmmscan` requires two inputs.

1. Query: a file containing one or more **protein sequences**, typically in FASTA format.
2. Target: a **Profile HMM Database** (e.g., Pfam). This database must be prepared using `hmmpress`, which compresses and indexes your profile database for faster lookups.

Process `hmmscan` performs a "Sequence v.s. Profile" search. It operates by reading the query sequence and scanning it against the entire indexed profile database. It annotates the query sequence by identifying which known domains or families within the database are present in the query.

Output The output is a ranked list of profile HMMs that match the query sequence with statistical significance, organized by E-value and bit score.

Usage First, compress the profile database:

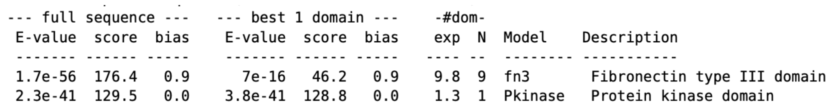
```
# Combine multiple HMM files into a single database file
cat fn3.hmm Pkinase.hmm > hmmdb

# Compress and index the database
hmmpress minifam
```

When the profile database is prepared, we can run `hmmscan` followed by the profile database and the query sequence file.

```
# Syntax: hmmscan [target_hmm_database] [query_seq_file] > [output_file]
hmmscan hmmdb seq.fasta > results.out
```

Below is an example of the `hmmscan` output ranked list. Similar to `hmmsearch`, the results are sorted by E-value. The difference is that now each row represents a valid hit of matched profile model. The **Model** column identifies the profile name (e.g., "fn3", "Pkinase"), and the **Description** column provides a brief summary of that domain's function. This layout allows researchers to rapidly identify which functional domains are present within their query protein.



```
--- full sequence ---   --- best 1 domain ---    -#dom-
 E-value  score  bias    E-value  score  bias    exp  N  Model    Description
 ------- ------ -----    ------- ------ -----   ---- --  -------- -----------
 1.7e-56  176.4   0.9      7e-16   46.2   0.9    9.8  9  fn3       Fibronectin type III domain
 2.3e-41  129.5   0.0    3.8e-41  128.8   0.0    1.3  1  Pkinase   Protein kinase domain
```

Figure 4. Example output of `hmmscan`. Image source: HMMER User's Guide [5].

Difference between `hmmsearch` and `hmmscan`

It's easy to confuse `hmmsearch` and `hmmscan` because they both use HMMs and sequences to find similarities, but they work in opposite directions depending on what you have as a query and what you are searching against.

	hmmsearch	**hmmscan**
Input Query	**Profile HMM** (e.g., a model of the hemoglobin family)	**Sequence** (e.g., a specific protein FASTA file)
Target Database	**Sequence Database** (e.g., UniProt)	**Profile HMM Database** (e.g., Pfam)
Primary Goal	To find new sequences that belong to the query profile family.	To annotate the query sequence with known domains.

Table 1. Comparison table between `hmmsearch` and `hmmscan`.

Discussions

A central challenge in bioinformatic algorithms is balancing sensitivity and computational efficiency. BLAST accelerates database scanning by using a seed-and-extend strategy [6], which improves speed but can miss remote homologs when the initial seeding stage fails under weak similarity. In contrast, HMMER's profile HMM framework provides more sensitive detection of evolutionary relationships, but at the cost of high computational expense. HMMER3 addressed this performance bottleneck by introducing CPU-level parallelization and multistage filtering, enabling it to approach BLAST-level runtimes while maintaining its characteristic sensitivity [1].

GPU-accelerated homology search has enabled further advances beyond CPU-optimized tools like HMMER3. MMseqs2 on GPUs provides significantly faster sequence searching than HMMER-based workflows, making MSA generation no longer the major bottleneck in structure prediction pipelines. This speed enabled MMseqs2 to replace HMMER in ColabFold [7], which is ~30× faster overall compared to the original AlphaFold2 pipeline [8].

References

[1] Eddy SR. Accelerated Profile HMM Searches. PLoS Computational Biology. 2011; 7(10):e1002195.
[2] Krogh A, Brown M, Mian IS, Sjolander K, Haussler D. Hidden Markov models in computational biology: Applications to protein modeling. Journal of Molecular Biology. 1994; 235:1501-1531.
[3] European Bioinformatics Institute. What are profile hidden Markov models? PFAM: Creating protein families. EMBL-EBI; n.d. [link]

[4] Wikipedia contributors. HMMER. Wikipedia, The Free Encyclopedia; n.d. https://en.wikipedia.org/wiki/HMMER
[5] Eddy SR; HMMER development team. HMMER User's Guide. http://eddylab.org/software/hmmer/Userguide.pdf/.
[6] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology. 1990; 215(3):403-410.
[7] Mirdita M, Schutze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. ColabFold: making protein folding accessible to all. Nat Methods. 2022;19:679-682.
[8] Kallenborn F, Chacon A, Hundt C, Sirelkhatim H, Didi K, Cha S, et al. GPU-accelerated homology search with MMseqs2. Nat Methods. 2025;22:2024-2027.