Chapter 1

Tutorial

I hate reading documentation. I just want examples of how stuff works, just enough to get me started and doing something productive. So, here's a tutorial walk-through of some small projects with HMMER. If you want the introduction, that's the second chapter. The tutorial should be sufficient to get you started on work of your own. You can read the other chapters later if you want.

1.1 The programs in HMMER

There are currently nine programs supported in the HMMER 2 package:

hmmalign	Align sequences to an existing model.
hmmbuild	Build a model from a multiple sequence alignment.
hmmcalibrate	Takes an HMM and empirically determines parameters that are used to make searches more sensitive, by calculating more accurate expectation value scores (E-values).
hmmconvert	Convert a model file into different formats, including a compact HMMER 2 binary format, and "best effort" emulation of GCG profiles.
hmmemit	Emit sequences probabilistically from a profile HMM.
hmmfetch	Get a single model from an HMM database.
hmmindex	Index an HMM database.
hmmpfam	Search an HMM database for matches to a query sequence.
hmmsearch	Search a sequence database for matches to an HMM.

HMMER also provides a number of utility programs which are not HMM programs, but may be useful. These programs are from the SQUID sequence utility library that HMMER uses:

afetch	Retrieve an alignment from an alignment database
alistat	Show some simple statistics about a sequence alignment file.
seqstat	Show some simple statistics about a sequence file.

sfetch Retrieve a (sub-)sequence from a sequence file.

shuffle Randomize sequences in a sequence file.

sreformat Reformat a sequence file into a different format.

1.2 Files used in the tutorial

The subdirectory `/tutorial` in the HMMER distribution contains the files used in the tutorial, as well as a number of examples of various file formats that HMMER reads. The important files for the tutorial are:

globins50.msf An MSF format alignment file of 50 aligned globin sequences.

globins630.fa A FASTA format file of 630 unaligned globin sequences.

fn3.slx A SELEX format alignment file of fibronectin type III domains.

rrm.slx A SELEX format alignment file of RNA recognition motif domains.

rrm.hmm An example HMM, built from rrm.slx.

pkinase.slx A SELEX format alignment file of protein kinase catalytic domains.

Artemia.fa A FASTA file of brine shrimp globin, which contains nine tandemly repeated globin domains.

7LES_DROME A SWISSPROT file of the *Drosophila* Sevenless sequence, a receptor tyrosine kinase with multiple domains.

Create a new directory that you can work in, and copy all the files in `tutorial` there. I'll assume for the following examples that you've installed the HMMER programs in your path; if not, you'll need to give a complete path name to the HMMER programs (e.g. something like `/usr/people/eddy/hmmer-2.2/binaries/hmmbuild` instead of just `hmmbuild`).

1.3 Searching a sequence database with a single profile HMM

One common use of HMMER is to search a sequence database for homologues of a protein family of interest. You need a multiple sequence alignment of the sequence family you're interested in. (Profile HMMs can be trained from unaligned sequences; however, this functionality is temporarily withdrawn from HMMER. I recommend CLUSTALW as an excellent, freely available multiple sequence alignment program.)

HMM construction with `hmmbuild`

Let's assume you have a multiple sequence alignment of a protein domain or protein sequence family. To use HMMER to search for additional remote homologues of the family, you want to first build a profile HMM from the alignment. The following command builds a profile HMM from the alignment of 50 globin sequences in `globins50.msf`:

```
> hmmbuild globin.hmm globins50.msf
```

```
hmmbuild - build a hidden Markov model from an alignment
HMMER 2.2 (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Alignment file:                   globins50.msf
File format:                      MSF
Search algorithm configuration:   Multiple domain (hmmls)
Model construction strategy:      MAP (gapmax hint: 0.50)
Null model used:                  (default)
Prior used:                       (default)
Sequence weighting method:        G/S/C tree weights
New HMM file:                     globin.hmm
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Alignment:           #1
Number of sequences: 50
Number of columns:   308

Determining effective sequence number    ... done. [2]
Weighting sequences heuristically        ... done.
Constructing model architecture          ... done.
Converting counts to probabilities       ... done.
Setting model name, etc.                 ... done. [globins50]

Constructed a profile HMM (length 148)
Average score:      194.97 bits
Minimum score:      -17.88 bits
Maximum score:      242.22 bits
Std. deviation:      55.12 bits

Finalizing model configuration           ... done.
Saving model to file                     ... done.
//
```

The process takes a second or two. `hmmbuild` create a new HMM file called `globin.hmm`. This is a human and computer readable ASCII text file, but for now you don't care. You also don't care for now what all the stuff in the output means; I'll describe it in detail later. The profile HMM can be treated as a compiled model of your alignment.

HMM calibration with `hmmcalibrate`

This step is optional, but doing it will increase the sensitivity of your database search.

When you search a sequence database, it is useful to get "E-values" (expectation values) in addition to raw scores. When you see a database hit that scores x, an E-value tells you the number of hits you would've expected to score x or more just by chance in a sequence database of this size.

HMMER will always estimate an E-value for your hits. However, unless you "calibrate" your model before a database search, HMMER uses an analytic upper bound calculation that is extremely conservative. An empirical HMM calibration costs time (about 10% the time of a SWISSPROT search) but it only has to be done once per model, and can greatly increase the sensitivity of a database search. To empirically calibrate the E-value calculations for the globin model, type:

```
> hmmcalibrate globin.hmm
hmmcalibrate -- calibrate HMM search statistics
HMMER 2.2 (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HMM file:                 globin.hmm
Length distribution mean: 325
Length distribution s.d.: 200
Number of samples:        5000
random seed:              997047045
histogram(s) saved to:    [not saved]
POSIX threads:            1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

HMM    : globins50
mu     :   -38.963402
lambda :     0.256441
max    :   -11.779000
//
```

This takes several minutes. Go have a cup of coffee. When it is complete, the relevant parameters are added to the HMM file.

Calibrated HMMER E-values tend to be relatively accurate. E-values of 0.1 or less are, in general, very significant hits. Uncalibrated HMMER E-values are also reliable, erring on the cautious side; uncalibrated models may miss remote homologues.

Sequence database search with `hmmsearch`

As an example of searching for new homologues using a profile HMM, we'll use the globin model to search for globin domains in the example *Artemia* globin sequence in `Artemia.fa`:

```
> hmmsearch globin.hmm Artemia.fa
```

The output comes in several sections, and unlike building and calibrating the HMM (where we treated the HMM as a black box), now you *do* care about what it's saying.

The first section is the *header* that tells you waht program you ran, on what, and with what options:

```
hmmsearch - search a sequence database with a profile HMM
HMMER 2.2 (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HMM file:                   globin.hmm [globins50]
Sequence database:          Artemia.fa
per-sequence score cutoff:  [none]
per-domain score cutoff:    [none]
per-sequence Eval cutoff:   <= 10
per-domain Eval cutoff:     [none]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
Query HMM:   globins50
Accession:   [none]
Description: [none]
  [HMM has been calibrated; E-values are empirical estimates]
```

The second section is the *sequence top hits* list. It is a list of ranked top hits (sorted by E-value, most significant hit first), formatted in a BLAST-like style:

```
Scores for complete sequences (score includes all domains):
Sequence Description                                    Score    E-value  N
-------- -----------                                    -----    ------- ---
S13421   S13421 GLOBIN - BRINE SHRIMP                   496.2   4.3e-150   9
```

The first field is the name of the target sequence, then followed by the description line for the sequence. The last three fields are the raw score (in units of "bits"), the estimated E-value, and the total number of domains detected in the sequence. By default, every sequence with an E-value less than 10.0 is listed in this output.

The second section is the *domain top hits* list. By default, for every sequence with an E-value less than 10, every domain with a raw score greater than 0 is listed. (Read that carefully. In a later chapter we'll discuss some caveats about how `hmmsearch` identifies domains, and how to control its output in different ways.) Each domain detected in the search is output in a list ranked by E-value:

```
Parsed for domains:
Sequence Domain  seq-f seq-t    hmm-f hmm-t      score  E-value
-------- ------- ----- -----    ----- -----      -----  -------
S13421     7/9     927  1075 ..     1   148 []    82.2  1.8e-25
S13421     2/9     148   293 ..     1   148 []    66.2  1.2e-20
S13421     3/9     302   450 ..     1   148 []    63.7  6.8e-20
S13421     8/9    1084  1234 ..     1   148 []    60.7  5.2e-19
S13421     9/9    1243  1390 ..     1   148 []    55.9  1.5e-17
S13421     4/9     459   607 ..     1   148 []    52.6  1.5e-16
S13421     6/9     770   918 ..     1   148 []    46.6  9.6e-15
S13421     1/9       1   143 [.     1   148 []    44.7  3.5e-14
S13421     5/9     618   762 ..     1   148 []    23.6  7.8e-08
```

The first field is the name of the target sequence. The second field is the number of this domain: e.g. "6/9" means the sixth domain of nine total domains detected.

The fields marked "seq-f" and "seq-t" mean "sequence from" and "sequence to": the start and end points of the alignment on the target sequence. After these two fields is a shorthand annotation for whether the alignment is "global" with respect to the sequence or not. A dot (.) means the alignment does not go all the way to the end; a bracket ([or]) means it does. Thus, .. means that the alignment is local within the sequence; [. means that the alignment starts at the beginning of the sequence, but doesn't go all the way to its end; .] means the alignment starts somewhere internally and goes all the way to the end; and [] means the alignment includes the entire sequence.

Analogously, the fields marked "hmm-f" and "hmm-t" indicate the start and end points with respect to the consensus coordinates of the model, and the following field is a shorthand for whether the alignment is global with respect to the *model*. Here, for instance, all the globin domains in the *Artemia* sequence are complete matches to the entire globin model – *because, by default,* `hmmbuild` *built the HMM to only look for those kinds of alignments*. We'll discuss later how to modify the profile HMM for other search styles.

The final two fields are the raw score in bits and the estimated E-value, *for the isolated domain*. The scores for the domains sum up to the raw score of the complete sequence.

The next section is the *alignment output*. By default, every domain that appeared in the domain top hits list now appears as a BLAST-like alignment. For example:

```
Alignments of top-scoring domains:
S13421: domain 7 of 9, from 927 to 1075: score 82.2, E = 1.8e-25
                    *->vhlsaeekalvksvwgkveknveevGaeaLerllvvyPetkryFpkF
                        +lsa+e a vk+ w+ v+ ++  vG  +++ l++ +P+ +++FpkF
      S13421   927    TGLSAREVAVVKQTWNLVKPDLMGVGMRIFKSLFEAFPAYQAVFPKF 973

                    kdLssadavkgsakvkahgkkVltalgdavkkldd...lkgalakLselH
                     d+   d++++++ v +h   V t+l++ ++ ld++ +l+   ++L+e H
      S13421   974 SDVPL-DKLEDTPAVGKHSISVTTKLDELIQTLDEpanLALLARQLGEDH 1022

                    aqklrvdpenfkllsevllvvlaeklgkeftpevqaalekllaavataLa
                    +  lrv+   fk +++vl+  l++ lg+ f+  ++ +++k+++++++ ++
      S13421  1023 IV-LRVNKPMFKSFGKVLVRLLENDLGQRFSSFASRSWHKAYDVIVEYIE 1071

                    akYk<-*
                    +  +
      S13421  1072 EGLQ    1075
```

The top line is the HMM consensus. The amino acid shown for the consensus is the highest probability amino acid at that position according to the HMM (not necessarily the highest *scoring* amino acid, though). Capital letters mean "highly conserved" residues: those with a probability of > 0.5 for protein models, or > 0.9 for DNA models.

The center line shows letters for "exact" matches to the highest probability residue in the HMM, or a "+" when the match has a positive score and is therefore considered to be "conservative" according to the HMM's view of *this particular position in the model* – not the usual definition of conservative changes in general.

The third line shows the sequence itself, of course.

The next section of the output is the *score histogram*. It shows a histogram with raw score increasing along the Y axis, and the number of sequence hits represented as a bar along the X axis. In our example here, since there's only a single sequence, the histogram is very boring:

```
Histogram of all scores:
score    obs    exp  (one = represents 1 sequences)
-----    ---    ---
  489      1      0|=
```

Notice though that it's a histogram of the whole sequence hits, not the domain hits.

You can ignore the rest of the `hmmsearch` output:

```
% Statistical details of theoretical EVD fit:
              mu =   -38.9634
          lambda =     0.2564
chi-sq statistic =     0.0000
  P(chi-square)  =          0

Total sequences searched: 1

Whole sequence top hits:
tophits_s report:
```

```
     Total hits:           1
     Satisfying E cutoff:  1
     Total memory:         16K

Domain top hits:
tophits_s report:
     Total hits:           9
     Satisfying E cutoff:  9
     Total memory:         21K
```

This is just some trailing internal info about the search that's useful to me sometimes, but probably not to you.

Searching major databases like NR or SWISSPROT

HMMER reads all major database formats and does not need any special database indexing. You can search any large sequence database you have installed locally just by giving the full path to the database file, e.g. something like:

```
> hmmsearch globin.hmm /nfs/databases/swiss35/sprot35.dat
```

If you have BLAST installed locally, it's likely that you have a directory (or directories) in which the BLAST databases are kept. These directories are specified in an environment variable called `BLASTDB`. HMMER will read the same environment variable. For example, if you have BLAST databases in directories called `/nfs/databases/blast-db/` and `/nfs/databases/golden-path/blast/`, and you want to search `/nfs/databases/blast-db/swissprot`, the following commands will work:

```
> setenv BLASTDB /nfs/databases/blast-db/:/nfs/databases/golden-path/blast/
> hmmsearch globin.hmm swissprot
```

Obviously, you'd tend to have the `setenv` command as part of the local configuration of your machine, rather than typing it at the command line.

Local alignment searches with `hmmsearch`

This is extremely important. HMMER does not do local (Smith/Waterman) and global (Needleman/Wunsch) style alignments in the same way that most computational biology analysis programs do it. To HMMER, whether local or global alignments are allowed is part of the *model*, rather than being accomplished by running a different *algorithm*. (This will be discussed in greater detail later; it is part of the "Plan7" architecture of the new HMMER2 models.)

Therefore, you need to choose what kind of alignments you want to allow *when you build the model* with `hmmbuild`. By default, `hmmbuild` builds models which allow alignments that are global with respect to the HMM, local with respect to the sequence, and allows multiple domains to hit per sequence. Such models will only find complete domains.

`hmmbuild` provides some standard options for common alignment styles. The following table shows the four alignment styles supported by `hmmbuild`, and also shows the equivalent old HMMER 1.x search program style (to orient experienced HMMER users).

Command	w.r.t. sequence	w.r.t. HMM	multidomain	HMMER 1 equivalent
`hmmbuild`	local	global	yes	hmmls
`hmmbuild -f`	local	local	yes	hmmfs
`hmmbuild -g`	local	global	no	hmms
`hmmbuild -s`	local	local	no	hmmsw

In brief, if you want maximum sensitivity at the expense of only finding complete domains, use the `hmmbuild` default. If you need to find fragments (local alignments) too, and are willing to give up some sensitivity to see them, use `hmmbuild -f`. If you want the best of both worlds, build two models and search with both of them.

1.4 Searching a query sequence against a profile HMM database

A second use of HMMER is to look for known domains in a query sequence, by searching a single sequence against a library of HMMs. (Contrast the previous section, in which we searched a single HMM against a sequence database.) To do this, you need a library of profile HMMs. One such library is our PFAM database (Sonnhammer et al., 1997; Sonnhammer et al., 1998), and you can also create your own.

Creating your own profile HMM database

HMM databases are simply concatenated single HMM files. You can build them either by invoking the `-A` "append" option of `hmmbuild`, or by concatenating HMM files you've already built. For example, here's two ways to build an HMM database called `myhmms` that contains models of the rrm RNA recognition motif domain, the fn3 fibronectin type III domain, and the pkinase protein kinase catalytic domain:

```
> hmmbuild rrm.hmm rrm.slx
> hmmbuild fn3.hmm fn3.slx
> hmmbuild pkinase.hmm pkinase.slx
> cat rrm.hmm fn3.hmm pkinase.hmm > myhmms
> hmmcalibrate myhmms
```

or:

```
> hmmbuild -A myhmms rrm.slx
> hmmbuild -A myhmms fn3.slx
> hmmbuild -A myhmms pkinase.slx
> hmmcalibrate myhmms
```

Notice that `hmmcalibrate` can be run on HMM databases as well as single HMMs.

Parsing the domain structure of a sequence with `hmmpfam`

Now that you have a small HMM database called `myhmms`, let's use it to analyze the *Drosophila* Sevenless sequence, `7LES_DROME`:

```
> hmmpfam myhmms 7LES_DROME
```

Like `hmmsearch`, the `hmmpfam` output comes in several sections. The first section is the *header*:

```
hmmpfam - search one or more sequences against HMM database
HMMER 2.2 (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HMM file:                 myhmms
Sequence file:            7LES_DROME
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Query sequence: 7LES_DROME
Accession:      P13368
Description:    SEVENLESS PROTEIN (EC 2.7.1.112).
```

The next section is the *sequence family classification* top hits list, ranked by E-value. The scores and E-values here reflect the confidence that this query sequence contains one *or more* domains belonging to a domain family. The fields have the same meaning as in `hmmsearch` output, except that the name and description are for the HMM that's been hit.

```
Model    Description                                        Score    E-value  N
-------- -----------                                        -----    ------- ---
pkinase                                                     304.1    1.1e-91   1
fn3                                                         176.3    3.5e-53   6
rrm      RNA recognition motif. (aka RRM, RBD, or RNP       -44.5       0.72   1
```

The next section is the *domain parse* list, ordered by position on the sequence (not by score). Again the fields have the same meaning as in `hmmsearch` output:

```
Parsed for domains:
Model    Domain  seq-f seq-t    hmm-f hmm-t      score  E-value
-------- ------- ----- -----    ----- -----      -----  -------
fn3        1/6     437   522 ..     1    84 []    49.0  7.1e-15
fn3        2/6     825   914 ..     1    84 []    13.6  4.3e-06
fn3        3/6    1292  1389 ..     1    84 []    16.2  2.4e-06
rrm        1/1    1300  1364 ..     1    72 []   -44.5     0.72
fn3        4/6    1799  1891 ..     1    84 []    63.5  3.2e-19
fn3        5/6    1899  1978 ..     1    84 []    14.6  3.4e-06
fn3        6/6    1993  2107 ..     1    84 []    19.4  1.2e-06
pkinase    1/1    2209  2483 ..     1   278 []   304.1  1.1e-91
```

Note how it's showing us an "rrm" hit - `7LES_DROME` doesn't have any RRM domains. You have to notice for yourself that the hit is insignificant (with a negative score, and an E-value of nearly 1). By default, like BLAST, the search programs report well down into the noise. If you want the output to be cleaner, set an E-value threshold; for example `hmmpfam -E 0.1`.

The final output section is the *alignment output*, just like `hmmsearch`:

```
Alignments of top-scoring domains:
fn3: domain 1 of 6, from 437 to 522: score 49.0, E = 7.1e-15
                   *->P.saPtnltvtdvtstsltlsWsppt.gngpitgYevtyRqpkngge
                      P saP   + +++ ++ l ++W p +  ngpi+gY+++  ++++g+
  7LES_DROME   437    PiSAPVIEHLMGLDDSHLAVHWHPGRfTNGPIEGYRLRL-SSSEGNA 482
```

```
                    wneltvpgtttsytltgLkPgteYtvrVqAvnggG.GpeS<-*
                    + e+ vp+   sy+++ L++gt+Yt+ +  +n +G+Gp
  7LES_DROME   483 TSEQLVPAGRGSYIFSQLQAGTNYTLALSMINKQGeGPVA    522
...
```

Downloading the PFAM database

The PFAM database is available from either `http://pfam.wustl.edu/` or `http://www.sanger.ac.uk/Pfam/`. Download instructions are on the Web page. The PFAM HMM library is a single large file, containing several hundred models of known protein domains. Install it in a convenient directory and name it something simple like `pfam`.

HMMER will look for PFAM and other files in a directory (or directories) specified by the `HMMERDB` environment variable. For instance, if you install the PFAM HMM library as `/nfs/databases/hmmer/pfam`, the following commands will search for domains in `7LES_DROME`:

```
> setenv HMMERDB /nfs/databases/hmmer/
> hmmpfam pfam 7LES_DROME
```

1.5 Maintaining multiple alignments with `hmmalign`

Another use of profile HMMs is to create multiple sequence alignments of large numbers of sequences. A profile HMM can be build of a "seed" alignment of a small number of representative sequences, and this profile HMM can be used to efficiently align any number of additional sequences.

This is in fact how the PFAM database is updated as the main SPTREMBL database increases in size. The PFAM seed alignments are (relatively) stable from release to release; PFAM full alignments are created automatically by searching SPTREMBL with the seed model and aligning all the significant hits into a multiple alignment using `hmmalign`.

For example, to align the 630 globin sequences in `globins630.fa` to our globin model `globin.hmm`, and create a new alignment file called `globins630.ali`, we'd do:

```
> hmmalign -o globins630.ali globin.hmm globins630.fa
hmmalign - align sequences to an HMM profile
HMMER 2.2 (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HMM file:             globin.hmm
Sequence file:        globins630.fa
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Alignment saved in file globins630.ali
```

Using the `-o` option to specify a save file for the final alignment is a good idea; else, the alignment will be displayed on the screen as output (and an alignment of several hundred sequences will give a fairly voluminous output).