HMMER web server Documentation

Release 1.0

Rob Finn & Simon Potter
EMBL-EBI

Jan 11, 2019

Contents

Contents:

CHAPTER 1

Target databases

The HMMER web service supports querying against a range of regularly updated sequence and HMM target databases.

Sequence databases

- Large, comprehensive sequence collection
  - UniProtKB - Comprehensive resource for protein sequence and annotation data produced by the Universal Protein Resource consortium.
- Annotated sequences and determined 3D structures
  - Swiss-Prot - Manually reviewed, high quality protein sequence and functional annotation - produced by UniProt.
  - PDB - Sequences with an experimentally determined structure.
- Representative Sets
  - Representative Proteomes - Representative Proteomes (RPs) are determined by selecting one proteome from a representative proteome group containing similar proteomes calculated based on sequence co-membership in UniRef50 clusters. A Representative Proteome is the proteome that can best represent all the proteomes in its group in terms of the majority of the sequence space and information. RPs at 75%, 55%, 35% and 15% co-membership threshold are available as target databases. More information on Representative Proteomes is available. The data set also includes model organisms and viral reference proteomes as defined by UniProt. The complete proteomes database comes from PIR.
  - Reference Proteomes - A set of proteomes from UniProt that gives broad coverage of the tree of life, and constitutes a representative cross-section of the taxonomic diversity to be found within UniProtKB. Produced by UniProt, in collaboration with Ensembl and the NCBI Reference Sequence collection.
- Other
  - Ensembl Genomes - Ensembl Genomes is a resource for genomic data for several thousands of invertebrate species. All translations resulting from known and novel gene predictions in Ensembl Genomes, including hypothetical proteins, are included. For lists of all the species in each sub division within Ensembl Genomes please see Bacteria, Fungi, Metazoa, Plants and Protists.
  - Ensembl - Searches may be performed across the entire set or one of Human, Mouse, or Zebrafish
  - Quest for Orthologs
  - MEROPS - a set of domain sequences from the MEROPS database of proteolytic enzymes. For each peptidase in the collection, the sequence of the known or predicted domain that carries the active site residues is included. Homologues that are not proteolytically active because one or more active site residues are

missing or replaced are also included. For each inhibitor, the sequence is that of each inhibitory domain. Domains homologous to an inhibitory domain are also included, even if no inhibitory activity is known.

  - ChEMBL - A manually curated database of bioactive molecules with drug-like properties. It brings together chemical, bioactivity and genomic data to aid the translation of genomic information into effective new drugs.

The default database is UniProt reference proteomes.

Profile HMM databases

- Pfam - A large comprehensive collection of protein families.
- TIGRFAMs - Models that are designed for automated sequence annotation and that are aimed at matching the full length (or near) of the sequence.
- Gene3D - A collection of models that are based on CATH structural protein domains.
- SUPERFAMILY - A collection of models, which represent structural protein domains at the SCOP superfamily level.
- PIRSF - Models that are designed to provide a comprehensive and non-overlapping clustering of UniProtKB.
- TreeFam - A database of phylogenetic trees of animal gene families.

The default database is Pfam.

Search provenance

Clicking 'Search Details' at the end of the result page reveals a box that provides details of the search, including the query sequence (if applicable) and information regarding the date/release of the target databases, which should be recorded for future reference when trying to recreate the results, discussing with colleagues or reporting bugs.

CHAPTER 2

Searches

Four search types are supported: **phmmer**, **hmmsearch**, **hmmscan** and **jackhmmer**. See *HMMER algorithms* for more information.

There are many different ways that a search on the website can be modified. Below is a list of the different accepted inputs and the parameters that can be modified. Also included are the parameter names that are required when using the API. This section is meant to be a guide to using the website, but further information can be found in the extensive HMMER guide. The parameter names used on the site are typically the same as the command line parameters, with the exception of the input data parameters. Each section is followed by a summary table that can be used as a quick reference.

Search query

phmmer, **hmmscan** and jackhmmer searches take a **single protein amino acid sequence** as the input, controlled by the **seq** parameter. The website accepts either FASTA format or an amino acid sequence. Alternatively, a sequence can be specified by **accession** or **identifier**. When using the website, suggestions will be offered as the name is typed.

Parameter name	seq	acc
Description	Sets the query sequence	
Algorithm(s)	phmmer, hmmscan, jackhmmer	
Accepted values	Protein sequence (FASTA)	Accession or identifier from one of the supported databases
Default	None	None
Required	Yes (seq or acc)	

hmmsearch and **jackhmmer** searches can take either a multiple protein sequence alignment as an input or a profile HMM. The alignment formats currently accepted are:

- Aligned FASTA
- Clustal (and Clustal-like)
- PSI-BLAST
- PHYLIP
- Selex
- GCG/MSF
- STOCKHOLM format
- UC Santa Cruz A2M (alignment to model)

The algorithms **hmmsearch** and **jackhmmer** also permit searches to be initiated with a profile HMM. This can be entered as text via the website, or via the seq or file parameters when using the API. Alternatively, it is also possible to retrieve HMMs from one of the supported HMM databases using the accession/identifier look up (in a similar manner to the sequence look up described earlier). To restrict the look up to one particular HMM database, append "@" followed by the database name (all lower case) e.g. CBS@pfam.

Query examples

For each of the search algorithms, examples sequences/alignments are provided (click on the 'example' button). These examples have been chosen to show a result set that demonstrates the various features available on the results pages.

Default search parameters

The searches on the website, when used in the simple mode, hide most of the search parameters and default values are used. Below is a list of the parameters and values used in the default search for each algorithm:

phmmer

Sequence database	UniProt reference proteomes
Significance threshold (E-value)	0.01 for sequence matches; 0.03 for hit matches
Reporting threshold (E-value)	1 for both sequences and hits
Gap penalties	open: 0.02; extend: 0.4; scoring matrix: BLOSUM62
Filter	Bias composition filtering on
Pfam search	Enabled, with gathering thresholds applied

hmmscan

HMM database	Pfam
Significance threshold	The Pfam gathering thresholds are used to determine hit significance
Filter	Bias composition filtering on

hmmsearch

Sequence database	UniProt reference proteomes
Significance threshold (E-value)	0.01 for sequence matches, 0.03 for hit matches
Reporting threshold (E-value)	1 for both sequences and hits
Filter	Bias composition filtering on

jackhmmer

Sequence database	UniProt reference proteomes
Significance threshold (E-value)	0.01 for sequence matches; 0.03 for hit matches
Reporting threshold (E-value)	1 for both sequences and hits
Gap penalties	open: 0.02; extend: 0.4; scoring matrix: BLOSUM62
Filter	Bias composition filtering on

Databases

Sequence databases

The sequence database field changes which target sequence database is searched. The default is UniProt references proteomes. This is one of the few parameters that is required by phmmer, hmmsearch or jackhmmer.

Parameter name	seqdb
Description	Sets the target sequence database
Algorithm	phmmer, hmmsearch, jackhmmer
Accepted values	uniprotrefprot, uniprotkb, swissprot, pdb, rp15, rp35, rp55, rp75, ensemblgenomes, ensembl, qfo
Default	uniprotrefprot (see below)
Required	Yes

HMM databases

This field indicates which profile HMM database the query should be searched against.

Parameter name	hmmdb
Description	Sets the target HMM database
Algorithm	hmmscan
Accepted values	gene3d, pfam, tigrfam, superfamily, pirsf, treefam
Default	pfam
Required	Yes

Thresholds

All four algorithms have the ability to set two different categories of cut-offs: **significance** and **reporting** thresholds. These cut-offs can be defined either as E-values (the default option) or bit scores. When setting either category of threshold, there are two values for each of the threshold categories: **sequence** and **hit**. A query can match a target in multiple places, defined as a hit (or domain) score. The sum of all hits on the sequence is the sequence score.

For example, trying to match repeating motifs can often be difficult, due to sequence variation in the repeating sequence motif. However, it can be possible to capture all examples of the motif, by relaxing the hit parameter while maintaining a stringent sequence parameter. This means that multiple matches, even if they are not strong matches, can be detected, but the sum of these matches must be sufficient to achieve the sequence score, there by limiting the rate of false positives.

Significance thresholds

Significance (or inclusion) thresholds are stricter than reporting thresholds and take precedence over them. These determine whether a sequence/hit is significant or not.

Significance E-values

Sequence and hit significance E-value thresholds will set matches with E-values less than or equal to the cut-off E-value as being significant (defaults below). If using the API, the incE and incdomE parameters are used to set the sequence and hit E-value thresholds respectively. In the absence of any threshold parameters the server will default to using E-value thresholds with the defaults.

Alternatively, the sequence and hit significance thresholds can be specified as bit scores. Any sequence or hit scoring greater than or equal to that given threshold will be considered a significant hit. By default, the form on the website is filled with typical values (defaults below). If using the API, the incT and incdomT parameters are used to set the sequence and hit bit thresholds respectively. This threshold is not used by default. If only one of these two parameters is set, then the unassigned parameter is set to the other assigned parameter value.

Parameter name	incE	incdomE
Description	Sequence E-value threshold	Hit E-value threshold
Algorithm	phmmer, hmmscan, hmmsearch, jackhmmer	
Accepted values	0 < x 10	10 < x 10
Default	0.01 or set to sequence threshold, if present	0.03 or set to hit threshold, if present
Required	No	No

Significance bit scores

Alternatively, the sequence and hit significance thresholds can be specified as bit scores. Any sequence or hit scoring greater than or equal to that given threshold will be considered a significant hit. By default, the form on the website is filled with typical values (defaults below). If using the API, the incT and incdomT parameters are used to set the sequence and hit bit thresholds respectively. This threshold is not used by default. If only one of these two parameters is set, then the unassigned parameter is set to the other assigned parameter value.

Parameter name	incT	incdomT
Description	Sequence bit score threshold	Hit bit score threshold
Algorithm	phmmer, hmmscan, hmmsearch, jackhmmer	
Accepted values	x > 0	x > 0
Default	25.0	22.0
Required	No	No

Reporting thresholds

The reporting thresholds controls how many matches that fall below the significance threshold are still shown in the results (i.e. reported). As every entity in the target database is compared to the query, if all matches were reported, then potentially vast outputs would be generated. However, it can often be useful to view border-line matches as they may reveal more distant **potential** informative similarities to the model. As with the significance thresholds, there is a value for both the sequence and the hit, which again can be defined as either an E-value or a bit score. Such reported matches are indicated by a yellow background in the results table produced in the website.

Reporting E-values

Parameter name	E	domE
Description	Sequence E-value threshold (reporting)	Hit E-value threshold (reporting)
Algorithm	phmmer, hmmscan, hmmsearch, jackhmmer	
Accepted values	0 < x 10	10 < x 10
Default	1 or set to sequence threshold, if present	1 or set to hit threshold, if present
Required	No	No

Reporting bit scores

The sequence and hit reporting thresholds can also be specified as bit scores. Any sequence or hit scoring greater than or equal to that given threshold will be reported (defaults below). If using the API, the T and domT parameters

are used to set the sequence and hit bit thresholds respectively. If significance thresholds are set, yet either or both reporting thresholds are undefined, these default form values will be set server side.

Parameter name	T	domT
Description	Sequence E-value threshold (reporting)	Hit E-value threshold (reporting)
Algorithm	phmmer, hmmscan, hmmsearch, jackhmmer	
Accepted values	x > 0	x > 0
Default	7.0	5.0
Required	No	No

Gathering thresholds

Specific to hmmscan, the gathering threshold indicates to HMMER to use the sequence and hit thresholds defined in the HMM file to be searched. In the cases of Pfam and TIGRFAMs these are set conservatively to ensure that there are no known false positives. Thus, if a query sequence scores with a bit score greater than or equal to the gathering thresholds, then that match can be treated with high confidence. This threshold is the default setting for hmmscan. If you are using the API, you can use the cut_ga parameter to signify that the gathering threshold should be used.

Gene3D and Superfamily thresholds

Both of these HMM databases apply sophisticated post-processing steps on the HMMER results to make the domain assignments and disentangle overlapping matches. Each database uses an internal E-value cut-off of 0.0001 for a domain match and does not employ the use of HMM specific bit score thresholds. Thus, cut-off manipulation has been disabled for these databases, thereby faithfully replicating the results of these HMM databases.

CHAPTER 3

Advanced search options

Taxonomy Restrictions

Search

You can add for taxa from all taxonomic levels (e.g. Homo sapiens or Metazoa) to be included in your search. You can add several taxa.

To *only* remove taxa, but keep all the other taxa, you can select the "Include all taxa" button. Now, the search box will only be removing taxa, instead of adding them

Pre-defined Taxonomic Tree

You can select different levels of a given taxonomic tree. All species within the selected levels will be included in your search.

Customisation of results

The result table may be customised to display different columns and/or to restrict the number of rows in the table to a manageable number. This can be performed before or after the search, with the customisation stored in a cookie so that you will not have to keep re-configuring the table after each search.

Pfam search

By default when performing a phmmer search via the website (and when JavaScript is enabled), a default hmmscan search against the Pfam HMM library is also performed. This feature is not available via the API, but can be mimicked by making separate requests to phmmer and hmmscan.

Filters

Bias composition

Turning off the bias composition filter can increases sensitivity, but at a high cost in speed, especially if the query has biased residue composition (such as a repetitive sequence region, or a membrane protein with large regions of

hydrophobicity). Without the bias filter, too many sequences may pass the filter with biased queries, leading to slower than expected performance, hence it is switched on by default. This feature can be disabled using the nobias parameter.

Parameter name	nobias
Description	Turns off the bias composition filtering
Algorithms	phmmer, hmmscan, hmmsearch, jackhmmer
Accepted Values	1
Required	No

Gap penalties

These are specific to phmmer and jackhmmer (initiated with a single sequence).

Open

The open parameter (called popen in HMMER) sets the probability for opening a gap in an alignment between target sequence against the model (or query sequence). The default value is 0.02, but can be set any value from 0 (no gaps) to less than 0.5 (more likely to extend the gap).

Extend

The extend parameter (called pextend in HMMER) sets the probability for extending the gap for a target sequence against the model or query sequence. The default value is 0.4, but can be set anywhere from 0 (less likely to extend) to less than 1 (more likely to extend the gap).

Scoring Matrix

When using phmmer, the query is a single sequence so the residue alignment probabilities are calculated from a substitution matrix. Substitution matrices provide scores that indicate the likelihood of two aligned amino acids appearing due to conservation rather than by chance. There are five different matrices available for selection: BLOSUM45, BLOSUM62 (default), BLOSUM90, PAM30 and PAM70. These BLOSUM matrices are based on observed alignments between amino acids in the BLOCKS database, where as the PAM matrices have been extrapolated from comparisons of closely related proteins. The different matrices alter the stringency of the alignment e.g. PAM90 can be used to find more distantly related sequences than PAM70, as PAM70 is more stringent; BLOSUM62 can be used to find more closely related sequence than using BLOSUM45, as BLOSUM45 is less stringent.

This is required for phmmer and jackhmmer and default values will be used if no value is set.

Parameter name	popen	pextend	mx
Description	Gap open penalty	Gap extend penalty	Substitution matrix
Algorithm(s)	phmmer, jackhmmer		
Accepted values	0 x < 0.5	0 x < 1	BLOSUM45, BLOSUM62, BLOSUM90, PAM30, PAM70
Default	0.02	0.4	BLOSUM62
Required	No		

Batch searches

It is also possible to search multiple protein sequences in 'offline' batch mode. With both **phmmer** and **hmmscan**, files containing sequences in FASTA format can be uploaded via the "Upload a file" link. These sequences will then be searched, in turn, against the specified databases. There is a limit of 500 sequences per batch request. This is only to prevent overload of the servers: multiple batch requests are permitted. Once the job is submitted, a different results page will be returned, showing a table with each row in that table representing a sequence in your file. This table periodically updates, indicating the progress of your batch job. As results appear in the table, you can view the details. If you have many sequences, you can also request that an e-mail be sent when the batch job has completed. It is also possible to use **hmmsearch** in batch mode, again with a single multiple alignment or profile HMM.

The **jackhmmer** batch system operates in a slightly different manner. Under the advance settings you can select the number of iterations to be performed and the batch mode will automaticaly run through each iteration (or until convergence), taking the results and using all the sequences scoring above the significance threholds to generate the input multiple sequnece alignment for the next round. Only one sequence, multiple sequence aligment or profile HMM can be submitted at a time.

The batch system also works via the API, except the seq parameter is substituted for the file parameter; the other parameters remain the same. Requesting an e-mail notification can be set using the email parameter.

Glossary

Bit score A bit score in HMMER is the log of the ratio of the sequence's probability according to the profile (the homology hypothesis) to the null model probability (the non-homology hypothesis).

E-value An E-value (expectation value) is the number of hits that would be expected to have a score equal to or better than this by chance alone. A good E-value is much less than 1, for example, an E-value of 0.01 would mean that on average about 1 false positive would be expected in every 100 searches with different query sequences. An E-value around 1 is what we expect just by chance. E-values are widely used as all you need to decide on the significance of a match is the E-value, but note that they vary according to the size of the target database.

Gathering threshold Also called the gathering cut-off, the gathering threshold is actually comprised of two bit scores, a sequence cut-off and a domain cut-off, used to define the significance of a sequence and a hit respectively. These are defined in the profile HMM and set both significance and reporting thresholds so that no insignificant hits are reported.

Null model The "null model" calculates the probability that the target sequence is not homologous to the query profile and is a one-state HMM configured to generate "random" sequences of the same mean length L as the target sequence, with each residue drawn from a background frequency distribution (a standard i.i.d. model: residues are treated as independent and identically distributed). This background frequency is based on the mean residue frequencies in Swiss-Prot 50.8 (October 2006).

Profile HMM Profile hidden Markov Models (HMMs) are a way of turning a multiple sequence alignment into a position-specific scoring system, which is suitable for searching databases for remotely homologous sequences.

STOCKHOLM format STOCKHOLM format is a multiple sequence alignment format supported by HMMER.

CHAPTER 4

Results

There are three ways of viewing results. The traditional score view is the default, but all three may be selected via the navigation buttons at the top of the page.

Score view The sequences matched are listed in order of decreasing score

Taxonomy view The matched sequences are arranged according to the taxonomic lineage of the source organism(s)

Domain view Significant matches are grouped by Pfam domains and presented in order of decreasing architecture frequency

Score view

Sequence Matches

Searches can result in many thousands of matches. Returning large numbers of results across the web and rendering them as a table is very time and memory consuming. As such, the first 100 matches are returned by default, allowing immediate analysis of the top matches. The remaining results can be viewed by clicking on the pagination links found above and below the table. You can see the range of matches currently selected in the bottom right corner of the table. Rows in the sequence match table that have a yellow background indicate sequences that score above the reporting thresholds, yet below the inclusion or significance thresholds. Therefore all hits, even if they score above the hit significance threshold will be deemed insignificant. Rows that have a red background indicate sequences that score above the significance/inclusion threshold, but where no single match exceeds the domain significance/inclusion thresholds.

Significant Query Matches (11) Customize

	Target	Description	Species	E-value
>	A8DYL7_DROME	Polar granule component, isoform A	Drosophila melanogaster	6.0e-44
>	B4I832_DROSE	GM15900	Drosophila sechellia	1.4e-43
>	B3NNN0_DROER	GG22179	Drosophila erecta (Fruit fly)	3.4e-43
>	B4QH43_DROSI	GD11659	Drosophila simulans	4.3e-43
>	B4P7W4_DROYA	GE14173	Drosophila yakuba (Fruit fly)	6.6e-42
>	B3MEV0_DROAN	GF11882	Drosophila ananassae	2.0e-30
>	B4GHJ2_DROPE	GL17535	Drosophila persimilis	1.3e-26
>	B5E0H9_DROPS	GA24195	Drosophila pseudoobscura pseudoobscura	1.3e-26
>	B4MJH1_DROWI	GK20810	Drosophila willistoni	2.2e-20
>	A8DYL6_DROME	CG34207	Drosophila melanogaster	7.7e-17
>	B3NNM8_DROER	GG20701	Drosophila erecta (Fruit fly)	1.2e-15
>	B4LMX3_DROVI	GJ22404	Drosophila virilis	1.3e-14
>	B4P7W2_DROYA	GE11685	Drosophila yakuba (Fruit fly)	2.0e-14
>	B4KT04_DROMO	GI20550	Drosophila mojavensis	1.9e-11
>	B3MEV2_DROAN	GF13026	Drosophila ananassae	2.3e-07

(show all) alignments Your search took:1.56 secs
showing rows 1 - 15 of 15

The dark red line in the table provides a visual clue as to where the threshold lies in the results.

Clicking on the right facing arrows (>) in the very first column of the table will reveal the alignment. The **show all** link in the table footer allows the display of all hit alignments for the sequences shown in the display (this is limited to tables of 100 rows or fewer).

Alignments

At the end of each row in the sequence hit table there is a "show" link. Clicking on this link displays the maximum expected accuracy (MEA) alignment between the query and the target. For each hit between the query and targets there are five rows in the alignment:

Position line (*) occur every 10th column of the alignment.

Query line the most probable sequence from the HMM that is coloured according to the match. In the case of a single sequence search, it is the query sequence.

Match line indicates identical residues (letters) or similar residues (+)

Target line the sequence aligned to the MODEL which is coloured according to the posterior probability.

PP line the per position posterior probability

Above the alignment the match details are presented:

Query start/end The start/end of the MEA alignment of this domain/hit with respect to the profile HMM, which directly relates to the query sequence for phmmer. For hmmsearch, the number corresponds to the match states that HMMER determined from the initial input alignment.

Target Envelope the domain envelope on the sequences defines a subsequence for which their is substantial probability mass supporting a homologous domain/hit, whether or not a single discrete alignment can be identified. The envelope may extend beyond the positions of the MEA alignment.

Target Alignment The start/end of the maximum expected accuracy (MEA) alignment of this domain with respect to the target sequence.

Bias The bias composition correction is the bit score difference contributed by the null2 model. High bias scores may be a red flag for a false positive. It is difficult to correct for all possible ways in which nonrandom but nonhomologous biological sequences can appear to be similar, such as short-period tandem repeats, so there are cases where the bias correction is not strong enough (creating false positives).

Accuracy is the mean posterior probability of aligned residues in the maximum expected accuracy alignment, essentially a measure of the reliability of the overall alignment. The accuracy ranges from 0 to 1, with 1.00 indicating a completely reliable alignment according to the model.

Bit score The bit score for this domain.

% Identity (count) The percentage of identical residues between the query and the target. The shortest length of the query or target is taken as the denominator. The number of identical residues is shown in brackets.

% Similarity (count) Similar to percent identity, except the sum of identical and similar residues (denoted by the + in the match state line) is used in the calculation.

There are also two E-values for the domain:

Conditional E-value This is the E-value that the inclusion and reporting significant thresholds that are measured against (if defined as E-values). The conditional E-value is an attempt to measure the statistical significance of each domain, given that it has already been decided that the target sequence is a true homolog. It is the expected number of additional domains or hits that would be found with a domain/hit score this big in the set of sequences reported in the top hits list, if those sequences consisted only of random nonhomologous sequence outside the region that sufficed to define them as homologs.

Independent E-value This is the significance of the sequence in the whole database search, if this were the only domain/hit that had been identified. If this E-value is not good, but the full sequence E-value is good, this is a potential red flag. Weak hits, none of which are good enough on their own, are summing up to lift the sequence up to a high score.

There can be multiple hits per sequence because HMMER performs local-local searches (meaning any subsequence of the query model can align to any subsequence of the target sequence). These are shown sequentially, according to the position on the sequence. An alignment with a yellow background indicates a reported domain/hit that falls below the domain/hit significance threshold.

Note: In the case of **hmmscan** the query and target lines correspond to different data. The second line (previously query) is the "Model" and the fourth line (previously target) is the "query".

Jackhmmer iterations

Iteration summary

After each iteration for jackhmmer, rather than proceeding to the results page, you are taken to a summary page, which gives an overview of the number of gained, lost or dropped sequences. Sequences gained are those that are new sequences compared to the previous iterations, scoring above the significance threshold. Lost are previously significant sequences, that are no longer reported in the results. Dropped sequences are sequences that were previously significant, but have fallen below the threshold but are still reported.

From this table it is possible to view the results of all previous iterations. Thus, if you decide that you want to re-run the latest iteration you can simply go back one and add/remove sequences. Alternatively, if you are happy with the way searches are proceeding, trigger of the next search, with will take all significant hits for the next iteration. If you job converges before 5 iterations (which is the current maximum), the table will be updated to indicate convergence and the run next iteration button will be remove.

Jackhmmer results

The results for jackhmmer are much the same as described above for phmmer. However, there are a few additions. The first is the inclusions of some navigation at the top of the page. The (lost matches) will show a table of the sequences

that have been completely lost compared to the previous iteration. There are links to the first new match and to the page of results where the threshold appears. There are also grey buttons in this block that allow you to move between iterations.

Another difference is that each row in the results has a check box, which allows sequences to be either removed or added to the results (a checked box denotes that they will be used in the next iteration). This allows you to modify which sequences are included in successive rounds of jackhmmer. By default, all sequences above the significance threshold are included. As a convenience, an option to override this and deselect all sequences is provided. This might useful if you wish to manually add only a small number of sequences. A button at the top and bottom of each page will allow you to start the next iteration.

New sequences in the results are denoted with a green background behind the target accession/identifier. Sequences that have dropped below threshold compared to the previous iteration are shown with a red background behind the target accession/identifier.

HMM logos

Below the results table for hmmsearch and jackhmmer (after first iteration if started with a single sequence), you will find an HMM logo. This produces a graphical representation of the profile HMM, with large letters representing more probably/conserved amino acids.

Customisation of Results

The default sequence match table contains four information columns: **Target** (accessions and/or identifiers), **Description** (functional annotations), **Species** and **E-value**. Additional columns can be added by clicking on the "Customise" link at the top right of table. This will reveal a form (shown below) that facilitates a range of custom display options.

The columns that can be selected are:

Row Count Number the columns

Secondary Accessions & Ids Additional identifiers that the sequence may also be known as in the literature and other databases

Description The sequence description

Species Shows the species to which this sequence belongs and provides a link to the NCBI taxonomy Browser

Cross-refs Displays cross-references to other resources available at the EBI through EBI Search.

Kingdom Shows the kingdom to which this sequence belongs

Known Structure (PDB) Shows whether a structure has been deposited in the PDB for some or all of the sequence, based on SIFTS

Identical Sequences As most of the target sequence databases contain some redundancy, we collapse identical sequences into a single row of the table. The redundant sequence information (accessions, description and species) is accessible by clicking the number found in the [Identical Seqs] column. This produces a pop-up table like the one shown below

Number of Hits The number of regions that score above the reporting threshold

Number of Significant Hits The number of regions that score above the inclusion threshold

Bit Score A bit score in HMMER is the log of the ratio of the sequence's probability according to the profile (the homology hypothesis) to the null model probability (the non-homology hypothesis).

Hit Positions A graphical representation showing the location of the matches of the query sequence to the target. Below is an example of a query sequence (top) that has 2 regions matching 4 regions in the target sequence (bottom). Note that there are 3 hits coloured red. These hits are all the same colour as they are found in an overlapping region of the query sequence. The fourth hit is labeled differently because it does not overlap any of the other sequences. The query and target images are scaled according to each other, so the query may scale differently from row to row in the table.

Rows Per Page In addition to column selection you can also choose the number of rows to be displayed per page. The default value is currently set to 100 rows per page, which shows you a reasonable amount of information, without over loading your browser. While an "All" option is provided, it is recommend that an initial limit be set as some searches can produce a large number of results, which may crash your browser during the rendering of the page.

The ability to **show all** hit alignments is disabled when more that 100 results are shown in the page.

Identical Sequences

As most of the target sequence databases contain some redundancy, identical sequences are collapsed into a single row of the table. The redundant sequence information (accessions, description and species) is accessible by clicking the number found in the [Identical Seqs] column. This will reveal a table like the one below, which shows information about the other identical sequences.

When more than 20 identical sequences are present, the "Next" link allows navigation through the list of redundant sequences.

Profile HMM Matches

This table differs slightly from the Query Match table above. As one sequence is being compared to a profile HMM database, we just report the **domain** hits.

This table is shown automatically for hmmscan searches and can be revealed on phmmer searches by clicking on the "Show hit details" link under the domain graphic. This gives the basic list of matches to Pfam domains, including the Pfam identifier, accession, clan accession and short description. The start/end positions in the basic view relate to the domain envelope. Finally, the domain conditional and independent E-values (described above). As before, rows in the match table that have a yellow background indicate matches that score above the reporting thresholds, yet below the inclusion or significance thresholds.

Pfam and TIGRFAMs both curate significance thresholds for their families. If a search is performed that uses either bit score or E-value thresholds, it is possible to match entries that are not deemed to be significant by those databases. To indicate when this is the case, we have included a symbol to signify that these matches fall below the database curated thresholds.

The alignment start/end positions (that indicate the position of maximum alignment accuracy), HMM model length and match start/end positions, as well as the bit score can be obtained by clicking on the **advanced** option in the top right of the table heading row.

Similar to the sequence hits, the show link reveals the alignment. This produces a similar formated pairwise alignment. Notice, that the query is now in the bottom row as the sequence is compared to a profile, not converted into a profile as with phmmer.

For searches against TreeFam the best hit only is shown, calculated as the hit with the lowest E-value below the threshold of 1e-29. If more than one hit have this E-value, that with the highest score is used.

Database specific result fields

Gene3D

The table of Gene3D results shows a final architecture of domain hits. These are selected from the full list by cath-resolve-hits, which resolves candidate hits to a final architecture using a dynamic-programming algorithm. The table includes a column for the hits' regions, which include any gaps of 30 residues or more.

Pfam

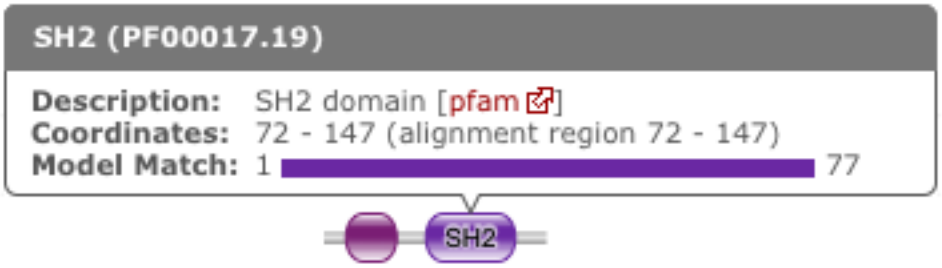
In Pfam related entries are grouped into Clans, and as such can often match the same, or similar, regions on the query sequence. An additional column in the results table contains the clan accession for the family, if it belongs to a clan. Pfam employs a specific post processing on families from the same clan where the best match (determined by lowest E-value), is taken and the rest are out-competed. In the results, the entry that has won the competition is indicated by a next to the clan accession and will be rendered in the domain graphic.

Superfamily

Similar to Gene3D, after hmmscan with Superfamily models as the target database, the matches are post processed to assign refined domain boundaries and E-value for the superfamily match. Thus, the results table for Superfamily is substantially different to those for the other HMM databases. Based on the superfamily match, the post processing then assigns a 'Family' based on sequence belonging to that Superfamily in the SCOP classification. If the family E-value is greater than 0.0001, the family match details have a yellow background. This E-value does not come from HMMER, but rather from the Superfamily post processing. The superfamily E-values are adjusted from HMMER to compensate for the fact that the Superfamily database can have multiple models representing each superfamily, and are thus not independent as assumed in the E-value calculation. To access the actual model/sequence data as calculated by HMMER, click advanced in the top right corner. The domain boundaries that should be cited for Superfamily are those in the 'Regions' column.

Domain Graphic

By default, a search using hmmscan is run when running a phmmer search. This will indicate the presence of any known Pfam domains on your query sequence. As with Pfam, we present the hits graphically as shown below:



In this example, there are two domains on the sequence. The second domain is label SH2, the first domain is an SH3 domain. You can reveal which domain the first representation is by mousing over the graphic or by viewing the table of domain hits. Note that the number of domains in the table and in the graphic may differ due to Pfam Clans, where multiple HMMs are used to represent large, divergent families. We apply the same post processing to remove overlaps as Pfam to produce the graphic, but unlike Pfam, we show all matches in the table.

Model Match

The model match section in the domain graphic pop up provides a graphical representation of the location the alignment to the model occurred. A full length match is indicated by the coloured bar spanning the entire length of the graphic. A shorter match will show the coloured bar overlaid onto a thinner grey bar.

Other Sequence Features

When a sequence is searched using hmmscan, phmmer or jackhmmer, the query sequence is also searched with three additional methods to identify sequence features, namely regions of disorder, signal peptides, transmembranes and coiled-coils.

If a search returns no results, then the graphic is not displayed. To make it clear when a search has been run, we have added small indicators at the bottom of the sequence features section. When a search has successfully completed it will be shown with a small green tick () next to it.

Disordered regions

We use the IUPred method for the prediction of disordered regions in the query sequence. The IUPred server provides more detailed disorder prediction results than currently offered here.

Bioinformatical approaches to characterize intrinsically disordered/unstructured proteins.

Dosztányi Z., Mészáros B., Simon I.

Briefings in Bioinformatics (2010) 11:225-43.

IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content.

Dosztányi Z., Csizmok V., Tompa P., Simon I.

Bioinformatics (2005) 21:3433-3434.

Signal peptides and Transmembrane regions

The Phoibus program is used to identify both signal peptides and transmembrane regions in the query sequence. The Phobius server provides more detailed prediction results than currently offered here.

A combined transmembrane topology and signal peptide prediction method.

Käll L., Krogh A., Sonnhammer E.L.

Journal of Molecular Biology (2004) 338:1027-36.

Coiled-coil regions

A derivative of Rob Russels ncoils program that was based on the Lupas et al. program for predicting coiled-coils in the query sequence.

Predicting coiled coils from protein sequences.

Lupas A., Van Dyke M., Stock J.

Science (1991) 252:1162-1164.

Note: When the above algorithms do not return any significant regions, the results are not drawn as part of the domain graphic.

Hit Coverage & Similarity

The coverage graph provides an overview of how the ensemble of target sequences matches the query sequence. As a match between a query and target sequence can be to a sub-region on either sequence, the presence of a ubiquitous domain in the query sequence can skew the set of matches to that region. The red line denotes the positional match information, which we term coverage, and is calculated on a per column bases, so gaps on the target sequence are taken into account. The coverage data can provide an indication of conserved regions or domains. We also summarise sequence conservation information that would normally be gleaned from inspecting the multiple sequence alignment, in the same graph. For each position in the query, we determine the relative percentage identity (grey area) and similarity (blue line) of the sequences covering that position. This allows the rapid identification of more conserved positions in query sequence.

As the variation of sequence similarity and identity can vary substantially from position to position, it can lead to very noisy looking graphs. To reduce this noise, we average the score over a window of 3 positions (one position either side of the current position). Although this may produce a visually more attractive graph, it can mask some information, in particular invariant positions. Thus, we also provide access to the unsmoothed or raw graph, using the button to the right of the graph.

Hit Graph

When the target is a sequence database (phmmer or hmmsearch), we produce a graph to show the distribution of matches. This can be found just above the 'Query Matches' table. The x-axis is hits that have been binned or grouped by E-value, the y-axis is the number of hits in the bin: An example is shown below:

The columns of the graph link to the table containing the sequence hits. Thus, to view hits with a higher e-value, click on one of the bins closer to the right side of the graph and the table will be scrolled to that position. Furthermore, each

bar in the graph is broken down according to the taxonomic kingdom to which the source organism belongs. It is then simple to assess the taxonomic range of sequence matches to the query sequence.

Under each table, there is a row of two links.

Downloading

The downloads section is accessed by clicking on the download link below the results table. There are a total of 8 different download formats for the different search algorithms:

Format | Description		Algorithm				Gzipped
		phmmer	hmmsearch	hmmscan	jackhmmer	
FASTA	Single file containing all the regions matched in your hits in FASTA format					
Full Length FASTA	As for FASTA, but the full length sequences for significant search hits					
Aligned FASTA	Significant search hits returned in the aligned FASTA format					
STOCKHOLM	Significant search hits returned in STOCKHOLM format. Useful if you wish to use your results with the command line version of HMMER					
ClustalW	Significant search hits returned in ClustalW format					
PSI-BLAST	Significant search hits returned in PSI-BLAST format					
PHLIP	Significant search hits returned in PHLIP format					
Plain text	Designed to be human readable with less information compared to the other formats					
XML	Machine readable with all the output data from HMMER					
JSON	As XML, but in JSON format					
HMM	A profile HMM generated from the uploaded multiple sequence alignment. LogoMat-M can be used to generate a graphical representation of the HMM					

Search details

The search details provides you with the exact time that the search was performed on our servers, the complete command used to perform the search and the database searched against. If the database has a version associated with it this will be documented, as well as the date that we downloaded the database. An example of the provenance data is shown here:

- Date Started: 2010-12-31 09:58:14
- Cmd: phmmer -E 10 –domE 10 –incE 0.01 –incdomE 0.03 –mx BLOSUM62 –pextend 0.4 –popen 0.02 –seqdb 6
- Database: uniprotrefprot, downloaded on 2010-12-11
- Search Sequence:

```
>2abl_A mol:protein length:163  ABL TYROSINE KINASE
MGPSENDPNLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGEWCEAQ
TKNGQGWVPSNYITPVNSLEKHSWYHGPVSRNAAEYLLSSGINGSFLV
```

```
RESESSPGQRSISLRYEGRVYHYRINTASDGKLYVSSESRFNTLAELV
HHHSTVADGLITTLHYPAP
```

We also include your query sequence in FASTA format, where applicable. Should you have bookmarked or performed multiple searches and have lost track of which job id corresponds to which job, then this provides a way of tracking the search. You should also double check that this sequence is the same as the one you submitted.

Taxonomy view

Tree Graphic

The first item on the Taxonomy view page is the taxonomic tree graphic. This shows all the sequence hits distributed across a tree derived from the NCBI taxonomy database. The tree starts on the left side with "All" sequences and each step to the right divides the data further until the species level is reached. Each node in the tree contains the classification name and the count of all hits from that point down. There is also a small hit distribution graphic located below each node, which indicates the proportion of significant hits found within that taxonomic group. Directly above the tree there is a directory like listing, which indicates all the parent nodes of the currently selected node. Clicking on one of the parents allows you to traverse back up to that level of the tree.

Species Distribution

The "Species Distribution" table is linked to the Tree graphic and displays all the species in which a hit occurred. As you descend down the tree, the number of species listed in the table will be reduced to show only those species that are found within the current top-level node. Along with each name we also show the number of hits that were found against sequences from the species. The last column is a link back to the score page that will provide more details on the hits associated with that species.

Downloading

This section is exactly the same as the Downloading section for the Score view

Search details

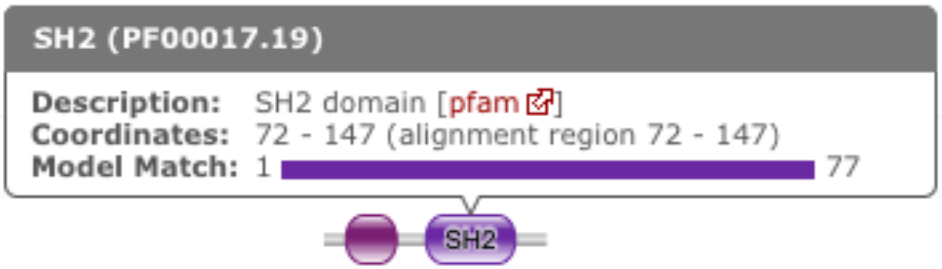
This is exactly the same as the Search details section for the Score view

Domain Architecture view

The "Domain Architecture" view is designed to group all significant sequence matches based on their constituent Pfam domains. The Pfam domains are defined using the Pfam curated gathering thresholds and can not be altered by search parameters. The results of a search are then displayed with the most frequently occurring architectures first.

Domain Graphic (Query)

This section is only available when running phmmer. An hmmscan is run against the pfam database for the query sequence. Domains found on that sequence are represented graphically as shown by the example below. This graphic is exactly the same as the one that can be found on the score view page, if the hmmscan was run as part of the original query. If not, a hmmscan is run using the default Pfam gathering thresholds. This allows the query sequence domain architecture to be compared to those found on the matched target sequences. Below this graphic, there is a link that will will take the users to the same architecutre as the query sequence architecture, if found in the set of target sequences.



Domain Architecture list

The domain architecture list is a breakdown of all the sequences found by your search according to the Pfam domains found within each sequence. Sequences with identical domain architectures are grouped together and ordered by the most frequently occurring. Note, sequences with no domains on them is also considered as an architecture. Each architecture group is represented on the page by a row in the tbale and each row can be divided into four subsections. An example is shown below:

Row Subsections

Sequence Count This is the number of sequences that share the domain architecture. Clicking on this count will reveal the domain architecture graphics for all of the sequences in this group. If there are more than 40 sequences with the same architecture, the results are paginated in sets of 40. The "Show More" will reveal the next set of matching sequences.

Example Here you are shown the name and order of each domain found in the architecture.

Graphic A graphical representation of the example sequence. This shows all the domains that were found for that architecture and can be used like the domain graphics for the query. The black line(s) along the bottom of the image indicate where your query aligned to the target sequence. Hovering over the black line will reveal a pop-up with the alignment coordinates of the hit.

View Scores Clicking this link will take you back to the score view and restrict the results shown to only those that have the selected architecture.

Downloading

This section is exactly the same as the Downloading section for the Score view

Search details

This is exactly the same as the Search details section for the Score view

Refining Searches

Searches can be refined by either selecting hits matching a specific domain architecture, a taxonomic level, or both.

Refine by domain architecture

Click on the "Domain" tab to see all hits clustered by the domain architecuture they match. To drill down into a specific architecture click on "view scores". The resulting page shows all sequence hits matching the domain architecture and there is a box telling you that your results have been filtered.

Refine by taxonomic level

Click on the "Taxonomy" tab to see all hits organised according to a species. To show sequences from a given taxonomic level only, click on an internal or leaf node of the species tree which updates the species in the lower part of the page. Click on "Show" to show all sequences for the corresponding species. If you have clicked on an internal node, then you will find an additional button "Show scores for all" at the bottom of the page. The resulting page shows all sequence hits matching the taxonomic level and there is a box telling you that your results have been filtered.

CHAPTER 5

API

Introduction

Using curl

The following section demonstrates a simple way of sending and retrieving XML using the simple Unix command line tool curl. The following example POSTs the request to the server (our server configuration requires you to also unset the default value in the header for Expect, -H 'Expect:'):

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F seqdb=pdb -F algo=phmmer -F seq='<test.
↪seq' https://www.ebi.ac.uk/Tools/hmmer/search/phmmer
```

```
<?xml version="1.0" encoding="UTF-8"?>
<opt>
  <data name='results' resultSize='224339'>
    <_internal highbit='370.5' lowbit='19.0' numberSig='242' offset='42280'>
      <timings search='0.283351' unpack='0.176821' />
    </_internal>
    <hits
        name='2abl_A'
        acc='2abl_A'
        bias='0.1'
        desc='mol:protein length:163  ABL TYROSINE KINASE'
        evalue='1.1e-110'
        ndom='1'
        nincluded='1'
        nregions='1'
        reported='1'
        score='370.5'
        species='Homo sapiens'
        taxid='9606' >
            <domains
                aliL='163'
                aliM='163'
                aliN='163'
                aliaseq=
↪'MGPSENDPNLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGEWCEAQTKNGQGWVPSNYITPVNSLEKHSWYHGPVSRNAAEYLLSSGINGSFLVR
↪'
                alihmmfrom='1'
                alihmmname='2abl_A'
                alihmmto='163'
```

```
                alimline=
↪'+gpsendpnlfvalydfvasgdntlsitkgeklrvlgynhngewceaqtkngqgwvpsnyitpvnslekhswyhgpvsrnaaeyllssgingsflvr
↪'
                alimodel=
↪'lgpsendpnlfvalydfvasgdntlsitkgeklrvlgynhngewceaqtkngqgwvpsnyitpvnslekhswyhgpvsrnaaeyllssgingsflvr
↪'
                alippline=
↪'8*********************************************************************************************
↪'
                alisqacc='2abl_A'
                alisqdesc='mol:protein length:163  ABL TYROSINE KINASE'
                alisqfrom='1'
                alisqname='2abl_A'
                alisqto='163'
                bias='0.05'
                bitscore='370.357543945312'
                envsc='250.653518676758'
                cevalue='4.21e-121'
                ievalue='4.21e-121'
                                iali='1'
                ienv='1'
                is_included='1'
                is_reported='1'
                jali='163'
                jenv='163'
            />
    </hits>
    .
    .
    .
  </data>
</opt>
```

In this example, the sequence to be searched is in the file test.seq. The value of the parameter "seq" needs to be quoted so that its value is taken correctly from the file. The other parameters can also be added directly to the URL, as a regular CGI-style parameter, if you prefer.

Using a script

Most programming languages have the ability to send HTTP requests and receive HTTP responses. A Perl script to submit a search and receive the responses as XML might be as trivial as this:

```
#!/usr/bin/perl

use strict;
use warnings;
use LWP::UserAgent;
use XML::Simple;

#Get a new Web user agent.
my $ua = LWP::UserAgent->new;
$ua->timeout(20);
$ua->env_proxy;

my $host = "https://www.ebi.ac.uk/Tools/hmmer";
my $search = "/search/phmmer";
```

```

#Parameters
my  $seq = qq(>2abl_A mol:protein length:163  ABL TYROSINE KINASE
MGPSENDPNLFVALYDFVASGDNT
LSITKGEKLRVLGYNHNGEWCEAQ
TKNGQGWVPSNYITPVNSLEKHSW
YHGPVSRNAAEYLLSSGINGSFLV
RESESSPGQRSISLRYEGRVYHYR
INTASDGKLYVSSESRFNTLAELV
HHHSTVADGLITTLHYPAP);

my $seqdb = 'pdb';

#Make a hash to encode for the content.
my %content = ( 'seqdb' => $seqdb,
                'content'   => "<![CDATA[$seq]]>" );

#Convert the parameters to XML
my $xml = XMLout(\%content, NoEscape => 1);

#Now post it off
my $response = $ua->post( $host.$search, 'content-type' => 'text/xml', Content =>
 ↪$xml );

#By default, we should get redirected!
if($response->is_redirect){

  #Now make a second requests, a get this time, to get the results.
  $response =
  $ua->get($response->header("location"), 'Accept' => 'text/xml' );

  if($response->is_success){
    print $response->content;
  }else{
    print "Error with redirect GET:".$response->content;
    die $response->status_line;
  }
}else{
  die $response->status_line;
}
```

Retrieving results

Although XML is just plain text and therefore human-readable, it's intended to be parsed into a data structure. Extending the Perl script above, we can add the ability to parse the XML using an external Perl module, XML::LibXML:

```
#!/usr/bin/perl

use strict;
use warnings;
use LWP::UserAgent;
use XML::Simple;
use XML::LibXML;

#Get a new Web user agent.
my $ua = LWP::UserAgent->new;
```

```
$ua->timeout(20);
$ua->env_proxy;

my $host = "https://www.ebi.ac.uk/Tools/hmmer";
my $search = "/search/phmmer";

#Parameters
my  $seq = qq(>2abl_A mol:protein length:163  ABL TYROSINE KINASE
MGPSENDPNLFVALYDFVASGDNTLSITKGE
KLRVLGYNHNGEWCEAQTKNGQGWVPSNYIT
PVNSLEKHSWYHGPVSRNAAEYLLSSGINGS
FLVRESESSPGQRSISLRYEGRVYHYRINTA
SDGKLYVSSESRFNTLAELVHHHSTVADGLI
TTLHYPAP);

my $seqdb = 'pdb';

#Make a hash to encode for the content.
my %content = ( 'seqdb' => $seqdb,
                'content'   => "<![CDATA[$seq]]>" );

#Convert the parameters to XML
my $xml = XMLout(\%content, NoEscape => 1);

#Now post it off
my $response = $ua->post( $host.$search, 'content-type' => 'text/xml', Content =>
↪$xml );

die "error: failed to successfully POST request: " . $response->status_line . "\n"
  unless ($response->is_redirect);

#By default, we should get redirected!
$response =
  $ua->get($response->header("location"), 'Accept' => 'text/xml' );

die "error: failed to retrieve XML: " . $response->status_line . "\n"
  unless $response->is_success;


my $xmlRes = '';

$xmlRes .= $response->content;
my $xml_parser = XML::LibXML->new();
my $dom = $xml_parser->parse_string( $xmlRes );

my $root = $dom->documentElement();

my ( $entry ) = $root->getChildrenByTagName( 'data' );
my @hits  = $entry->getChildrenByTagName( 'hits' );

foreach my $hit (@hits){
  next if($hit->getAttribute( 'nincluded' ) == 0 );
  print $hit->getAttribute( 'name' )."\t".$hit->getAttribute( 'desc' )."\t".$hit->
↪getAttribute( 'evalue' )."\n";
}
```

This script now prints out the name, description and E-value of all significant sequence hits for the given query sequence in tab delimited format:

```
2abl_A        mol:protein length:163  ABL TYROSINE KINASE     1.1e-110
2fo0_A        mol:protein length:495  Proto-oncogene tyrosine-protein kinase ABL1 (
↪8.4e-109
1opk_A        mol:protein length:495  Proto-oncogene tyrosine-protein kinase ABL1
↪8.4e-109
1opl_A        mol:protein length:537  proto-oncogene tyrosine-protein kinase  9.7e-109
1ab2_A        mol:protein length:109  C-ABL TYROSINE KINASE SH2 DOMAIN        3.3e-62
3k2m_A        mol:protein length:112  Proto-oncogene tyrosine-protein kinase ABL1
↪3.1e-61
2ecd_A        mol:protein length:119  Tyrosine-protein kinase ABL2    6.5e-58
1abo_A        mol:protein length:62  ABL TYROSINE KINASE     1.1e-38
3eg1_A        mol:protein length:63  Proto-oncogene tyrosine-protein kinase ABL1
↪1.6e-38
3eg0_A        mol:protein length:63  Proto-oncogene tyrosine-protein kinase ABL1
↪1.7e-38
3eg3_A        mol:protein length:63  Proto-oncogene tyrosine-protein kinase ABL1
↪3.3e-38
1ju5_C        mol:protein length:61  Abl      8.4e-38
1bbz_A        mol:protein length:58  ABL TYROSINE KINASE     7.0e-36
2o88_A        mol:protein length:58  Proto-oncogene tyrosine-protein kinase ABL1
↪9.1e-35
1awo_A        mol:protein length:62  ABL TYROSINE KINASE     1.7e-34
```

Available services

phmmer searches

The main two input parameters to a phmmer search are a protein sequence and the target database, defined using the seq and seqdb parameters respectively. Other parameters for controlling the search are defined in the search section. If any of these parameters are omitted, then the default values for that parameter will be set.

Searches should be POST-ed to the following url:

```
https://www.ebi.ac.uk/Tools/hmmer/search/phmmer
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F seqdb=pdb -F seq='<test.seq' https://www.
↪ebi.ac.uk/Tools/hmmer/search/phmmer
```

When using the website, we also perform a Pfam search by default. However, when using the API you will only be returned the phmmer results. To get Pfam search results, use the hmmscan interface.

hmmscan searches

Hmmscan also has two main parameters - a sequence and a profile HMM database - defined using the seq and hmmdb parameters respectively. We currently offer six profile HMM databases: Pfam, TIGRFAMs, Gene3D, Superfamily, PIRSF and TreeFam. When searching against the first two, the cut-offs can be defined by the user (other parameters for controlling the search are defined in the search section). With the remaining databases all cut-off parameters will be ignored and the default HMM database parameters will be used. This is because these databases use their own post-processing mechanisms to define their domains, in addition to the hmmscan results.

Searches should be POST-ed to the following url:

```
https://www.ebi.ac.uk/Tools/hmmer/search/hmmscan
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F hmmdb=pfam -F seq='<test.seq' https://
↪www.ebi.ac.uk/Tools/hmmer/search/hmmscan
```

hmmsearch searches

The input to hmmsearch on the web is either a multiple sequence alignment or a hidden Markov model in HMMER3 format. We do not support HMMER2 format as these HMMs are not forward compatible with HMMER3. When uploading a multiple sequence alignment, an HMM is built on the server using hmmbuild with the default parameters.

Searches should be POST-ed to the following url:

```
https://www.ebi.ac.uk/Tools/hmmer/search/hmmsearch
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F seqdb=pdb -F seq='<test.ali' https://www.
↪ebi.ac.uk/Tools/hmmer/search/hmmsearch
```

jackhmmer searches

Jackhmmer is an iterative search algorithm that can be initiated with a sequence, multiple sequence alignment or profile HMM. The number of iterations to run can be supplied as an additional parameter and will perform a succession of searches until the job has completed. Fetching the results is a little more complicated, as the search may finish before the number of iterations if it converges.

Searches should be POST-ed to the following url:

```
https://www.ebi.ac.uk/Tools/hmmer/search/jackhmmer
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F seqdb=pdb -F iterations=5 -F seq='<test1.
↪fa' https://www.ebi.ac.uk/Tools/hmmer/search/jackhmmer
```

Taxonomic restrictions

For searches against a sequence database (i.e. all types excluding hmmscan) you may restrict your search by taxonomy. To do this, set the parameter taxFilterType=search, alongwith either or both of tax_included and tax_excluded, each of which takes a comma delimited list of taxonomy IDs.

Example:

```
curl -L -H 'Expect:' -H 'Accept:application/json' -F taxFilterType=search -F tax_
↪included=40674 -F tax_excluded=9606,10090 -F seqdb=pdb -F seq='<seq.fa' https://www.
↪ebi.ac.uk/Tools/hmmer/search/phmmer
```

Annotation searches

In addition to the standard HMMER searches an uploaded sequence can be annotated to show signal peptide & transmembrane regions, disordered regions and coiled-coil regions.

Annotation requests should be POST-ed to the following urls.

Disorder:

```
https://www.ebi.ac.uk/Tools/hmmer/annotation/disorder
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F  seq='<test.fa' https://www.ebi.ac.uk/
↪Tools/hmmer/annotation/disorder
```

Coiled-coil:

```
https://www.ebi.ac.uk/Tools/hmmer/annotation/coils
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F  seq='<test.fa' https://www.ebi.ac.uk/
↪Tools/hmmer/annotation/coils
```

Transmembrane & Signal Peptides:

```
https://www.ebi.ac.uk/Tools/hmmer/annotation/phobius
```

Example:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F  seq='<test.fa' https://www.ebi.ac.uk/
↪Tools/hmmer/annotation/phobius
```

Annotation results can be fetched with a GET request using the UUID supplied in the POST response:

```
https://www.ebi.ac.uk/Tools/hmmer/annotation/<annotation-type>/UUID
```

Example:

```
curl -H 'Expect:' -H 'Accept:text/xml' https://www.ebi.ac.uk/Tools/hmmer/annotation/
↪phobius/4162F712-1DD2-11B2-B17E-C09EFE1DC403
```

Results

Search results can be retrieved using the job identifier that is returned in your initial search response. The job identifier is a UUID (format such as 4162F712-1DD2-11B2-B17E-C09EFE1DC403). Thus, to retrieve your job, you can use the following URL in a GET request:

```
https://www.ebi.ac.uk/Tools/hmmer/results/$your_uuid?output=html
```

Example:

```
https://www.ebi.ac.uk/Tools/hmmer/results/4162F712-1DD2-11B2-B17E-C09EFE1DC403?
↪output=html
```

This is one of the few services where the returned format can be modified using a parameter.

Parameter	range	ali	output
Description	The range of the results to retrieve	Return alignments	Modify the format that the results are returned in
Accepted values	Integer,Integer	true | 1	xml | json | text | yaml
Example	range=1,100	ali=1	html
Default/Without Parameter	All results	No alignments will be returned	output=text
Notes	The results are ordered by E-value and as there can be thousands of matches to your query, it can be useful to retrieve a subset of results. The range is two, unsigned, comma separated integers. The first integer is expected to be less than the second integer. To retrieve one row, just fetch using a range where the two integers are the same value. If your first integer is in range, and your second is out of range, the second integer will be modified to include all results. i.e. If your results set is only 300 in size, and a range of 1,1000 is requested, then you will get 300 results. If your starting integer is "out" of range, then no results will be returned.	Sometimes you are not so interested in the alignment of the match to the query sequence. By default no alignments are returned, to keep results compact.	The format of the results can be modified with by setting "output=$format". The same can be achieved by setting the "Accept" field in the HTTP header. If both the HTTP header and the parameter are set, we currently assume that the parameter is the desired format.

Deleting results

The results will normally only remain on the server for a maximum of one week; however they may be deleted by sending a DELETE request:

```
curl -X DELETE -H 'Accept:application/json' https://www.ebi.ac.uk/Tools/hmmer/results/
↪F36F96A4-0806-11E8-A990-C006DCC3747A/score
```

Taxonomy and domain views

The API may also be used to retrive the data behind the taxonomy and domain architecture tabs on the results page. For taxonomy the URL has the form:

```
https://www.ebi.ac.uk/Tools/hmmer/results/$your_uuid/taxonomy
```

Example:

```
curl -s -H "Content-type: application/json" 'https://www.ebi.ac.uk/Tools/hmmer/
↪results/8D5B74A0-6158-11E7-B311-1331132D729D/taxonomy'
```

The fields returned are described in *Appendix F - JSON format*

For domain architecture, two endpoints are provided. The first returns an overview of all architectures:

```
https://www.ebi.ac.uk/Tools/hmmer/results/$your_uuid/domain
```

Example:

```
curl -s -H "Content-type: application/json" 'https://www.ebi.ac.uk/Tools/hmmer/
↪results/8D5B74A0-6158-11E7-B311-1331132D729D/domain'
```

The second queries an individual architecture identifier:

```
https://www.ebi.ac.uk/Tools/hmmer/results/$your_uuid/domain/$arch_id
```

Example:

```
curl -s -H "Content-type: application/json" 'https://www.ebi.ac.uk/Tools/hmmer/
↪results/D33FBDA4-6230-11E7-BC34-E492DBC3747A/domain/36055491190690'
```

Examples

phmmer

The following piece of python is a little more complex than those discussed previously. In this case, we submit a search to the server, but stop the HTTP handler from automatically following the redirection to the results page. Instead, a custom handler is define that grabs the redirection URL and modifies it by the addition of parameters such that it fetches just the first 10 matches in JSON format, rather than grabbing the whole response. This can be useful when the results are large and you want to paginate the response, or if you are only interested in the most significant sequence matches.

```
1  import urllib, urllib2
2
3  # install a custom handler to prevent following of redirects automatically.
4  class SmartRedirectHandler(urllib2.HTTPRedirectHandler):
5      def http_error_302(self, req, fp, code, msg, headers):
6          return headers
7  opener = urllib2.build_opener(SmartRedirectHandler())
8  urllib2.install_opener(opener);
9
10 parameters = {
11     'seqdb':'pdb',
12     'seq':'>Seq\nKLRVLGYHNGEWCEAQTKNGQGWVPSNYITPVNSLENSIDKHSWYHGPVSRNAAEY'
```

```
}
enc_params = urllib.urlencode(parameters);

#post the seqrch request to the server
request = urllib2.Request('https://www.ebi.ac.uk/Tools/hmmer/search/phmmer',enc_
↪params)

#get the url where the results can be fetched from
results_url = urllib2.urlopen(request).getheader('location')

# modify the range, format and presence of alignments in your results here
res_params = {
    'output': 'json',
    'range': '1,10'
}

# add the parameters to your request for the results
enc_res_params = urllib.urlencode(res_params)
modified_res_url = results_url + '?' + enc_res_params

# send a GET request to the server
results_request = urllib2.Request(modified_res_url)
data = urllib2.urlopen(results_request)

# print out the results
print data.read()
```

hmmscan

The following is a very basic Java source file that, once compiled and and executed performs an hmmscan search. The response is returned in JSON format. With this two stage POST and GET, you can POST the request in one format and get a response back in another by setting the Accept type. To get this example to work, you should save the code in a file called RESTClient.java. Then run the command "javac RESTClient.java". Assuming that this is successful and a file called RESTClient.class is produced, you can execute the class by running the command "java RESTClient"

```
import java.net.*;
import java.io.*;

public class RESTClient{
  public static void main(String[] args) {
    try {
        URL url = new URL("https://www.ebi.ac.uk/Tools/hmmer/search/hmmscan");
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setDoOutput(true);
        connection.setDoInput(true);
        connection.setInstanceFollowRedirects(false);
        connection.setRequestMethod("POST");
        connection.setRequestProperty("Content-Type", "application/x-www-form-
↪urlencoded");
        connection.setRequestProperty("Accept", "application/json");

        //Add the database and the sequence. Add more options as you wish!
        String urlParameters = "hmmdb=" + URLEncoder.encode("pfam", "UTF-8") +
        "&seq=" + ">
↪seq\nEMGPSENDPNLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGEWCEAQTKNGQGWVPSNYITPV" +
        "NSLEKHSWYHGPVSRNAAEYLLSSGINGSFLVRESESSPGQRSISLRYEG" +
```

```
        "RVYHYRINTASDGKLYVSSESRFNTLAELVHHHSTVADGLITTLHYPAP";

         connection.setRequestProperty("Content-Length", "" +
               Integer.toString(urlParameters.getBytes().length));


        //Send request
        DataOutputStream wr = new DataOutputStream (
                  connection.getOutputStream ());
        wr.writeBytes (urlParameters);
        wr.flush ();
        wr.close ();



        //Now get the redirect URL
        URL respUrl = new URL( connection.getHeaderField( "Location" ));
        HttpURLConnection connection2 = (HttpURLConnection) respUrl.openConnection();
        connection2.setRequestMethod("GET");
        connection2.setRequestProperty("Accept", "application/json");


        //Get the response and print it to the screen
        BufferedReader in = new BufferedReader(
                                new InputStreamReader(
                                connection2.getInputStream()));

        String inputLine;

        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);
        in.close();


    } catch(Exception e) {
        throw new RuntimeException(e);
    }
  }
}
```

jackhmmer

A jackhmmer is a multipart search. The following Perl code performs a series of requests to the server. The first POST request generates the jobs, the while loop then performs GET requests to get the job status, until the status of the job is done. The last request GETs the results of the last iteration, which are returned in JSON format.

```
#!/usr/bin/env perl
use strict;
use warnings;
use LWP::UserAgent;
use JSON;

#Get a new Web user agent.
my $ua = LWP::UserAgent->new;
$ua->timeout(60);
$ua->env_proxy;
```

```perl
#Set a new JSON end encoder/decoder
my $json = JSON->new->allow_nonref;

#-------------------------------------------------------------------------------
#Set up the job

#URL to query
my $rootUrl = "https://www.ebi.ac.uk/Tools/hmmer";
my $url = $rootUrl."/search/jackhmmer";

my $seq = ">2abl_A mol:protein length:163  ABL TYROSINE KINASE
MGPSENDPNLFVALYDFVASGDNTLSITKGEKLRVLGYNHNGEWCEAQTKNGQGWVPSNYITPVNSLEKHS
WYHGPVSRNAAEYLLSSGINGSFLVRESESSPGQRSISLRYEGRVYHYRINTASDGKLYVSSESRFNTLAE
LVHHHSTVADGLITTLHYPAP";

my %content = (
  'algo'     => 'jackhmmer',
  'seq'      => $seq,
  'seqdb'    => 'pdb',
  iterations => 5,
);

#-------------------------------------------------------------------------------
#Now POST the request and generate the search job.
my $response = $ua->post(
  $url,
  'content-type' => 'application/json',
  Content        => $json->encode( \%content )
);

if($response->status_line ne "201 Created"){
  die "Failed to create job, got:".$response->status_line;
}

my $job = $json->decode( $response->content );
print "Generated job UUID:".$job->{job_id}."\n";

#Follow the redicrection to the resouce create for the job.
my $job_location = $response->header("location");
#Now poll the server until the job has finished
$response = $ua->get( $job_location, 'Accept' => 'application/json' );

my $max_retry = 50;
my $count     = 1;

while ( $response->status_line eq '200 OK' ) {
  my $status = $json->decode( $response->content );

  print "Checking status ($count).....";
  if ( $status->{status} eq 'DONE' ) {
    print "Job done.\n";
    last;
  }
  elsif ( $status->{status} eq 'ERROR' ) {
    print "Job failed, exiting!\n";
    exit(1);
  }
  elsif ( $status->{status} eq 'RUN' or $status->{status} eq 'PEND' ) {
```

```
    my ($lastIteration) = $status->{result}->[-1]->{uuid} =~ /\.(\d+)/;
    print "Currently on iteration $lastIteration [$status->{status}].\n";
  }

  if ( $count > $max_retry ) {
    print "Jobs should have finished.....exiting\n";;
    exit(1);
  }
  #Job still running, so give it a chance to complete.
  sleep(5);
  #Check again on the job status...
  $response = $ua->get( $job_location, 'Accept' => 'application/json' );
  $count++;
}

#Job should have finished, but we may have converged, so get the last job.
my $results = $json->decode( $response->content );
my $lastIteration = pop( @{ $results->{result} } );
#Now fetch the results of the last iteration
my $searchResult = $ua->get( $rootUrl."/results/" . $lastIteration->{uuid} . "/score
↪", 'Accept' => 'application/json' );
unless( $searchResult->status_line eq "200 OK"){
  die "Failed to get search results\n";
}

#Decode the content of the full set of results
$results = $json->decode( $searchResult->content );
print "Matched ".$results->{'results'}->{'stats'}->{'nincluded'}." sequences (
↪$lastIteration->{uuid})!\n";
#Now do something more interesting with the results......
```

Batch searches

So far, the submission of batch searches via REST has not really been mentioned. This is because we do not anticipate this being so useful as you can programmatically send sequence after sequence. However, a batch upload of sequences is possible for phmmer and hmmscan. The main difference is that instead of using the seq parameter, we use the file parameter. There is also a subtle difference in the way that the curl command is formulated. Rather than using a redirect (<), a the symbol is used to force the content part of the request to be what is contained within the file, rather than being attached to the parameter:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' -F seqdb=pdb -F file='@batch.fasta' https://
↪www.ebi.ac.uk/Tools/hmmer/search/phmmer
```

It is also possible to include an email address for notification of when the batch search has been processed. Again, not particularly useful for an API, but it may be useful for keeping track of a pipeline. To specify an email via the command line, simply use the parameter email and set this to a valid email address. All of the other phmmer or hmmscan search parameters apply to the batch search.

Fetching results

Using curl to fetch results is very easy:

```
curl -L -H 'Expect:' -H 'Accept:text/xml' https://www.ebi.ac.uk/Tools/hmmer/results/
↪CF5BCDA4-0C7E-11E0-AF4F-B1E277D6C7BA?output=text&ali=1&range=1,2
```

In this case we want to fetch the first two hits, with their alignments as a textual output format.

Downloading files from batch searches

For batch searches, it is unfortunately not possible to download files for all sequences in a single request. A single combined output might make sense for some formats (e.g. tsv), but not others (such as an alignment). The results of each search need to be downloaded individually.

The summary page may be requested as json or xml to make it easier to parse the list of individual:

```
curl -H 'Accept: application/json' 'https://www.ebi.ac.uk/Tools/hmmer/results/
↪A67B56FE-CA07-11E7-A02C-F964E976C163/score'
```

Once you have the list of IDs you can download any of the files programmatically, for example:

```
curl 'https://www.ebi.ac.uk/Tools/hmmer/download/A67B56FE-CA07-11E7-A02C-F964E976C163.
↪5/score?format=csv'
```

CHAPTER 6

About HMMER

HMMER project

The HMMER project is a collaborative project between the HMMER algorithm developers, led by Sean Eddy at HHMI/Harvard University and the HMMER web service team, lead by Rob Finn at EMBL-EBI. The software is available at hmmer.org.

While the HMMER algorithm developers focus on improving the speed and sensitivity of searches, the HMMER web service team takes these algorithms and deploys them in a production environment to enable optimal performance, given a finite set of resources. The service team also works on ensuring that the underlying HMM and sequences databases are regularly updated and that search results are presented in intuitive visualisations. The web interface is freely accessible, allowing users to perform rapid sequence analyses using the HMMER software suite. The servers allow HMMER to be used to address a wide variety of questions involving sequence function, conservation and evolution.

A paper describing the web server has been published in Nucleic Acids Research. In addition to the human interactive website, we have developed an API that allows simple machine access to the same infrastructure. This should allow relatively large scale analysis to be performed in a timely fashion.

Sponsors

HMMER is supported by

EMBL	EMBL is EMBL-EBI's parent organisation; it provides core funding for HMMER.
hhmi	Howard Hughes Medical Institute supports the Eddy group
Supported by wellcome trust	WT maintains the site at which EMBL-EBI is situtated and provides funding for HMMER.

EMBL is EMBL-EBI's parent organisation. It provides core funding for HMMER

The Howard Hughes Medical Institute supports the Eddy group

As well as providing and maintaining the campus on which the EMBL-EBI is located, the Wellcome Trust also now provides funding for HMMER

How to cite

If you have used the HMMER website, please consider citing the following publication that describes this work:

HMMER web server: 2015 update R.D. Finn, J. Clements, W. Arndt, B.L. Miller, T.J. Wheeler, F. Schreiber, A. Bateman and S.R. Eddy **Nucleic Acids Research** (2015) Web Server Issue 43:W30-W38. 10.1093/nar/gkv397 (PDF)

CHAPTER 7

The HMMER software

HMMER algorithms

The following HMMER algorithms/programs are supported by this server:

phmmer used to search one or more query protein sequences against a protein sequence database

hmmscan search protein sequences against collections of profiles, such as Pfam. In HMMER2 this was called hmmpfam

hmmsearch used to search one or more profiles against a protein sequence database

jackhmmer iteratively search a query protein sequence, multiple sequence alignment or profile HMM against the target protein sequence database

This software has been released as part of the HMMER software package (version 3.1)

Other programs

The following is a brief description of the other programs in the HMMER suite. These are only available from downloaded distributions. However they are used indirectly when performing the searches on the server

hmmalign performs a multiple sequence alignment of all the sequences (usually identified by running an hmmsearch) in the input, by aligning them individually to the profile HMM

hmmbuild builds a profile HMM for each multiple sequence alignment in the input multiple sequence alignment file, and saves it to a new file

hmmconvert utility converts an input profile file to different HMMER formats

hmmfetch retrieves one or more profile HMMs from a profile database

hmmpress takes a profile database in standard HMMER3 format and constructs binary compressed data files for hmmscan

hmmstat utility prints out a tabular file of summary statistics for each profile

CHAPTER 8

Help

These help pages are primarily designed to give users a very brief introduction to HMMER, sufficient such that the user will have a better understanding of the website search methods and results. They do not describe the details of profile hidden Markov models (HMMs) in the use of sequence analysis.

Helpdesk

Your questions are important to us. Please contact us through our contact form at https://www.ebi.ac.uk/support/hmmer. We will respond as quickly as possible, but please bear in mind that we do not have a dedicated staff member to run the helpesk.

Software bug reports will typically be dealt with by **Sean Eddy**, but may get deferred to others within the HMMER project team. (See http://hmmer.org/documentation.html)

To expedite our response to your questions, please provide us with as much information as possible so that we can recreate the problem. Useful things to include are:

- Input data (or examples, but just sufficient to recreate the problem).
- The URL of the page where you are having the problem.
- The steps to follow to reproduce the problem.
- Information about the browser that you are using and it version, and also the OS you are running.

Staying informed

The target databases are updated on a monthly basis. Additionally there will be bug fixes and new website features. To keep up to date with these, see one of the following:

- *Changelog* - lists new features, bug fixes and improvements made to the site.
- Twitter - follow hmm3r for micro-blogs about the HMMER software updates, target database updates and new Web features.
- Cryptogenomicon blog - more detailed discussions of HMMER related topics, including the website.

CHAPTER 9

Appendices

Appendix A - Result object format

The results are returned from the search servers as a binary data object. This can be a little complex when first looked at. However, the data structure is fairly simple and is represented pictorially below:

In the following sections the contents of each part of the results data structure will be described. Parts of the data structure will be referred to as hashes (key, value pairs) or arrays, but depending on the type of response requested will translate into different entities, for example elements and attributes for an XML response.

"Results" hash

stats	The stats hash
hits	Array of sequence hashes
uuid	The unique job identifier
algo	The HMMER search algorithm
searchDB	The target search database
_internal	Hash containing some internal accounting

"Stats" hash

nhits	The number of hits found above reporting thresholds
Z	The number of sequences or models in the target database
domZ	The number of hits in the target database
nmodels	The number of models in this search
nincluded	The number of sequences or models scoring above the significance threshold
nreported	The number of sequences or models scoring above the reporting threshold

"Sequence" hash

The hits array contains one or more sequences. Only parts of the response actually deemed useful will be described. With the non-redundant databases, the redundant sequence information will also be included, but as the sequences are identical, the information about the hit is identical.

name	Name of the target (sequence for phmmer/hmmsearch, HMM for hmmscan)
acc	Accession of the target
acc2	Secondary accession of the target
id	Identifier of the target
desc	Description of the target
score	Bit score of the sequence (all domains, without correction)
pvalue	P-value of the score
evalue	E-value of the score
nregions	Number of regions evaluated
nen-velopes	Number of envelopes handed over for domain definition, null2, alignment, and scoring.
ndom	Total number of domains identified in this sequence
nreported	Number of domains satisfying reporting thresholding
nincluded	Number of domains satisfying inclusion thresholding
taxid	The NCBI taxonomy identifier of the target (if applicable)
species	The species name of the target (if applicable)
kg	The kingdom of life that the target belongs to - based on placing in the NCBI taxonomy tree (if applicable)
seqs	An array containing information about the 100% redundant sequences
pdbs	Array of pdb identifiers (which chains information)

"Domain" Hash

The domain or hit hash contains the details of the match, in particular the alignment between the query and the target.

ienv	Envelope start position
jenv	Envelope end position
iali	Alignment start position
jali	Alignment end position
bias	null2 score contribution
oasc	Optimal alignment accuracy score
bitscore	Overall score in bits, null corrected, if this were the only domain in seq
cevalue	Conditional E-value based on the domain correction
ievalue	Independent E-value based on the domain correction
is_reported	1 if domain meets reporting thresholds
is_included	1 if domain meets inclusion thresholds
alimodel	Aligned query consensus sequence phmmer and hmmsearch, target hmm for hmmscan
alimline	Match line indicating identities, conservation +'s, gaps
aliaseq	Aligned target sequence for phmmer and hmmsearch, query for hmmscan
alippline	Posterior probability annotation
alihmm-name	Name of HMM (query sequence for phmmer, alignment for hmmsearch and target hmm for hmmscan)
alihmmacc	Accession of HMM
alihmmdesc	Description of HMM
alihmmfrom	Start position on HMM
alihmmto	End position on HMM
aliM	Length of model
alisqname	Name of target sequence (phmmer, hmmscan) or query sequence(hmmscan)
alisqacc	Accession of sequence
alisqdesc	Description of sequence
alisqfrom	Start position on sequence
alisqto	End position on sequence
aliL	Length of sequence

Appendix B - response codes

One of the philosophies of a RESTful API is to also pass the appropriate HTTP status code in response to the query URL. Most of the time a 200 (success) status code will be received. However, there may be times when that is not the case. There is a complete list of HTTP codes elsewhere, but we have listed most of the status codes that may be returned and how they relate to what is actually going on at the server.

200 (OK) The job has either been run or queued up successfully. In the former case, the body should contain the results, whereas the latter will contain your job identifier that can be used to query/fetch the results in the future.

201 (Create) The job has been created successfully. Response will contain either the content describing the job and/or a redirection to the created resource in the HTTP header.

202 (Accepted) The job has been accepted by the search system and is either pending (waiting to be started) or running. After a short delay, your script should check for results again.

302 (Found/Redirection) The request was found, but the client must take additional action to complete the request. Usually there is a redirection URL found in the response header.

400 (Bad Request) Your job contained either invalid parameters or parameter values. The body of your response should contain information about which parameter or value failed and possibly the reason why it failed. If you continue to receive this in response to a request and can not understand why it is failing, you should contact the help desk for assistance.

410 (Gone) Your job was deleted from the search system. This may be because the time that we have been able to store the results has expired or that you have explicitly asked for the results to be deleted.

500 (Internal server error) There was a problem with running your job, typically due to a problem with the back-end compute servers, rather than the job itself. The body of the response may contain an error message from the server. Contact the help desk for assistance with the problem.

502 (Bad gateway) There was a problem scheduling or running the job. The job has failed and will not produce results. There is no need to check the status again.

503 (Service unavailable) The body of the response may contain a message as to why the job has been put on hold. This may be due to site maintenance, database updates, queue overload or if there is a problem. This status is set typically by an administrator and should this status code be present for longer that a few hours, you should contact the help desk.

Appendix C - data formats

The RESTful interface supports three different, commonly used, machine readable formats: XML, JSON and YAML. In addition to these, we also provide HTML and text. Which format used is really down to personal choice. XML is widely used with libraries in many different languages. JSON is readily applicable to use with websites, in which a server may make a call to a HMMER web service and pass the resulting JSON string back to the client/browser, where the HMMER result may be post-processed by JavaScript running on the client. YAML is a more recent markup language which, despite being readily parsed by software, is more human-readable than XML or JSON. The HTML responses are not really meant for anything other than a browser or command line tools such as curl or wget. The text output is the best output if you want to cut and paste results into a lab book.

Appendix D - unsupported features

We have tried to provide as many services as possible via REST. However, there are still a few things that we do not provide. For example, there is no way of generating a domain graphic or getting a graph of the distribution of hits. We can not provide this via REST as the both of these are generated client side using JavaScript libraries and the HTML5 canvas element. The RESTful services are also, naturally, restricted to just the set of HMMER programs that are available via the website. But, if there is something that you think would be useful, then please get in touch and we will consider it for inclusion.

Appendix E - Job ID

The job ID, also refered to as UUID (Universally Unique IDentifier), is a 36 character sequence that looks like *10F15DB0-2E1C-11E0-B944-D59DDB6B6FDE* and that uniquely identifies a job submitted on the website.

Appendix F - JSON format

The results visualised in the score, taxonomy and architecture views are all available using the API in JSON format. For the score endpoint, the object returned includes the Stats, Sequence and Domain hashes referred to above (Appendix A).

The taxonomy endpoint provides a recursive hash of the tree with the keys

id	NCBI taxonomy identifier
parentid	taxonomy identifier of parent
name	taxonomy name
hitcount	number of hits to this node
hitdist	binned log e-values of hits to this node
children	children of this node (recursive)

CHAPTER 10

Changelog

Version 2.31, December 2018

- Changes
  - UniProtKB release 2018_11

Version 2.30, November 2018

- Changes
  - UniProtKB release 2018_10
  - Ensembl Genomes release 41
  - Ensembl release 94
- New Features
  - Added phylum as an optional column in the results table

Version 2.29, October 2018

- Changes
  - UniProtKB release 2018_09

Version 2.28, September 2018

- Changes
  - Pfam release 32.0
- Bug fixes
  - Fix to display of position in queue for batch searches

Version 2.27, September 2018

- Changes
  - UniProtKB release 2018_08

Version 2.26, August 2018

- Changes
  - Ensembl Genomes release 40
  - Ensembl release 93
  - ChEMBL added as a supported sequence database (version 24)

- Bug fixes
  - Fix to searches against TreeFam using the API (prior to this, searches used the gathering threshold, which does not apply to this database)

Version 2.25, July 2018

- Changes
  - UniProt release 2018_07

Version 2.24, June 2018

- Changes
  - UniProt release 2018_06
- New Features
  - Added new "fisheye" mode on the taxonomy viewer

Version 2.23, May 2018

- Changes
  - UniProt release 2018_05
  - Ensembl Genomes release 39
  - Ensembl release 92
- Bug fixes
  - Improvements to the taxonomy page performance

Version 2.22, April 2018

- Changes
  - UniProt release 2018_04
  - Introduced TreeFam, version 9, with post-processing using the default TreeFam e-value threshold and hit selection
  - Website annotated using Schemas.org and current working version of BioSchemas

Version 2.21, January 2018

- Changes
  - UniProt release 2018_01
  - Ensembl Genomes release 38

Version 2.20, December 2017

- Changes
  - UniProt release 2017_12
  - Ensembl release 91

Version 2.19, November 2017

- Changes
  - UniProt release 2017_11
  - MEROPS 12

Version 2.18, October 2017

- Changes
  - UniProt release 2017_10
  - Gene3D version 16.0.0
- Bug fixes
  - Fix XML output for some API endpoints

Version 2.17, September 2017

- Changes
  - UniProt release 2017_09
  - Ensembl Genomes release 37
  - Ensembl release 90
- New features
  - Better taxonomy viewer using taxonomy-visualisation library

Version 2.16, August 2017

- Changes
  - UniProt release 2017_08
  - Change some email templates to have tab-delimited headers and rows
- New features
  - Added information warning about next release across the website

Version 2.15, July 2017

- Changes
  - UniProt release 2017_07
  - Ensembl Genomes release 36
  - Ensembl release 89
  - New option "unselect all" for jackhmmer iterations
  - New endpoints available as JSON (taxonomy and domain architecture)
- Bug fixes
  - Download of ClustalW, PSI-BLAST and PHYLIP file formats fixed

Version 2.14, June 2017

- Changes
  - UniProt release 2017_06

Version 2.13, May 2017

- Changes
  - UniProt release 2017_05
  - Ensembl Genomes release 35
  - Ensembl release 88
  - Gene3D post-processing now uses cath-resolve-hits

Version 2.12

- Changes
  - Website now follows EBI guidelines
  - EBI Search cross-references added for all supported databases

Version 2.11, March 2017

- Changes
  - UniProt release 2017_03
  - Pfam release 31.0
  - MEROPS 11 added as a supported sequence database
  - PIRSF: new post-processing enables the unification of two or more matches that are separated due to the HMMER3 local-local matching model
  - (beta version) Added EBI Search cross-references in sequence database results

Version 2.10, February 2017

- Changes
  - UniProt release 2017_02
- Bug fixes
  - Improved handling of HMM logos (some HMMs are unable to be rendered owing to the way they are constructed)

Version 2.9, January 2017

- Changes
  - UniProt release 2017_01

Version 2.8, December 2016

- Changes
  - Pfam active sites
  - Ensembl

Version 2.7, September 2016

- Changes
  - UniProt release 2016_08
  - Gene3D version 14

Version 2.6, August 2016

- Changes
  - Ensembl Genomes 32
- Bug fixes
  - Fixes in search and download pages

Version 2.5, July 2016

- Changes
  - small UI improvements

Version 2.4, June 2016

- New features
  - Integration of complete Ensembl Plants, and of Ensembl Protists as supported databases for searches.
  - Update to Pfam 30.0
- Changes
  - More UI changes to the search page

Version 2.3, May 2016

- New features
  - Integration of Ensembl Bacteria, Ensembl Fungi, Ensembl Metazoa, and Ensembl Plants as supported databases for searches.
- Changes
  - Small changes in the UI (especially in the search page)
  - Improved performance and better caching

Version 2.2, March 2016

- New features
  - Integration of Ensembl Genomes as a supported database for searches.
- Bug fixes
  - Fixed error on selection between iterations of Jackhmmer searches

Version 2.1, January 2016

- New features
  - RP levels that were previously removed have been reinstated by popular demand.
  - Revisions to the help documentation.
  - PDB search results now link to both PDBe and RCSB.

Version 2.0, August 2015

- New features
  - Move from Janelia to EBI.
  - Now supporting Ensembl Genomes Plants as a new target database.
  - RP levels removed.

Version 1.4, May 2013

- New features
  - We have enabled the searching of **multiple** hmm databases via hmmscan. This allows the results of Gene3D, Superfamily, Pfam and TIGRGAMs to be compared in a single page.
  - The **HMM length** and the coverage of the HMM is now indicated in the tool tip associated with the domain graphic, located in the 'sequence features' section. The HMM length has also been added to the hmmscan results table.
  - The website is now using HMMER **version 3.1**, with the software due to be released shortly. We have added the option of downloading HMMs in both 3.0 and 3.1 formats.

  - **Alignment downloads** have been improved, particularly for large alignments, which were often so big that the server would timeout.
  - We have also work on several speed optimisations in the website to improve interactivity.

- Bug Fixes
  - Based on user feedback, we have updated the validation of E-value cut-offs to allow **scientific notation** with the exponent as E or e.
  - Fixed issue with **long taxon names** which are now being truncated to ensure that tree, in taxonomy results visualisation, remains aligned.