CONTENTS

1 Contents: 3
2 License 81
3 Citing Pfam 83
4 Get in touch 85
Pfam is a large collection of protein families, each represented by multiple sequence alignments and profile hidden Markov models (HMMs).
1.1 Summary

Proteins are generally comprised of one or more functional regions, commonly termed domains. The presence of different domains in varying combinations in different proteins gives rise to the diverse repertoire of proteins found in nature. Identifying the domains present in a protein can provide insights into the function of that protein.

The Pfam database is a large collection of protein domain families. Each family is represented by multiple sequence alignments and a profile hidden Markov model (HMM).

Each Pfam family, sometimes referred to as a Pfam-A entry, consists of a curated seed alignment containing a small set of representative members of the family, profile HMMs built from the seed alignment, and an automatically generated full alignment, which contains all detectable protein sequences belonging to the family, as defined by profile HMM searches of primary sequence databases.

Pfam entries are classified in one of six types:

- **Family**  A collection of related protein regions
- **Domain**  A structural unit
- **Repeat**  A short unit which is unstable in isolation but forms a stable structure when multiple copies are present
- **Motifs**  A short unit found outside globular domains
- **Coiled-Coil**  Regions that predominantly contain coiled-coil motifs, regions that typically contain alpha-helices that are coiled together in bundles of 2-7.
- **Disordered**  Regions that are conserved, yet are either shown or predicted to contain bias sequence composition and/or are intrinsically disordered (non-globular).

Related Pfam entries are grouped together into clans; the relationship may be defined by similarity of sequence, structure or profile HMM.

1.2 Getting Started

1.2.1 Site organisation

The **family** page is the major page for accessing information contained within Pfam as it describes the Pfam family entries. Most referring sites link to this page. Alternatively, users can navigate to family pages by entering the Pfam identifier or accession number, either via the home page, the “Jump-to” boxes or the keyword search box, or by clicking on a domain name or graphic from anywhere on the website. As with all Pfam pages, there is the context-sensitive icon bar in the top right hand corner that provides a quick overview about the contents of the tabs. The tabs on the family
The order in which the search tries the various types of ID and accession is given below:

- Pfam A accession, e.g. PF02171
- Pfam A identifier, e.g. piwi
- UniProt sequence ID, e.g. CANX_CHICK
- NCBI “GI” number, e.g. 113594566
- NCBI secondary accession, e.g. BAF18440.1
- Pfam clan accession, e.g. CL0005
- metaseq ID, e.g. JCVI_ORF_1096665732460
- metaseq accession, e.g. JCVI_PEP_1096665732461
- Pfam clan accession, e.g. CL0005
- Pfam clan ID, e.g. Kazal
- PDB entry, e.g. 2abl
- Proteome species name, e.g. Homo sapiens
1.2.3 Keyword search

Every page in the Pfam site includes a search box in the page header. You can use this to find Pfam-A families which match a particular keyword. The search includes several different areas of the Pfam database:

- text fields in Pfam entries, e.g. family descriptions
- UniProt sequence entry description and species fields
- HEADER and TITLE fields from PDB entries
- Gene Ontology IDs and terms
- InterPro entry abstracts

Each Pfam-A entry is listed only once in the results table, although it might have been found in more than one area of the database.

1.2.4 Searching a protein sequence against Pfam

Searching a protein sequence against the Pfam library of HMMs will enable you to find out the domain architecture of the protein. If your protein is present in the version of UniProt, NCBI Genpept or the metagenomic sequence set that we used to make the current release of Pfam, we have already calculated its domain architecture. You can access this by entering the sequence accession or ID in the ‘view a sequence’ box on the Pfam homepage.

If your sequence is not in the Pfam database, you could perform a single-sequence or a batch search by clicking on the ‘Search’ link at the top of the Pfam page.

Single protein search

If your protein is not recognised by Pfam, you will need to paste the protein sequence into the search page. We will search your sequence against our HMMs and instantly display the matches for you.

Batch search

If you have a large number of sequences to search (up to several thousand), you can use our batch upload facility. This allows you to upload a file of your sequences in FASTA format, and we will run them against our HMMs and email the results back to you, usually within 48 hours. We request that you put a maximum of 5000 sequences in each file.

Local protein searches

If you have a very large number of protein searches to perform, or you do not wish to post your sequence across the web, it may be more convenient to run the Pfam searches locally using the ‘pfam_scan.pl’ script. To do this you will need the HMMER3 software, the Pfam HMM libraries and a couple of additional data files from the Pfam website. You will also need to download a few modules from CPAN, most notably Moose.

Full details on how to get ‘pfam_scan.pl’ up and running can be found on our FTP site.
Proteome analysis

Pfam pre-calculates the domain compositions and architectures UniprotKB reference proteomes. To see the list of proteomes, click on the ‘browse’ link at the top of the Pfam website, and click on a letter of the alphabet in the ‘proteomes’ section. By clicking on a particular organism, you will be able to view the proteome page for that organism. From here you can view the domain organisation and the domain composition for that proteome.

The taxonomy query allows quick identification of families/domains which are present in one species but are absent from another. It can also be used to find families/domains that are unique to a particular species (note this can be very slow).

1.2.5 Finding proteins with a specific set of domain combinations (‘architectures’)

For a detailed study of domain architectures you can use PfamAlyzer. PfamAlyzer allows you to find proteins which contain a specific combination of domains and to specify particular species and the evolutionary distances allowed between domains.

1.2.6 Wikipedia annotation

The Pfam consortium is now coordinating the annotation of Pfam families via Wikipedia. On the summary tab of some family pages, you’ll find the text from a Wikipedia article that we feel provides a good description of the Pfam family. If a family has a Wikipedia article assigned to it, we now show the text of that article on the summary tab, in preference to the traditional Pfam annotation text.

If a family does not yet have a Wikipedia article assigned to it, there are several ways for you to help us add one. You can find much more information about the process in the Pfam Annotation in Wikipedia section.

1.3 Training materials

1.3.1 Pfam Quick tour

- Quick tour provides a brief introduction to the Pfam website and database.

1.3.2 Creating Families

- Creating families provides a tutorial on how to create a Pfam entry.

1.3.3 Repeats in Pfam

- Repeats describes how repeats are represented in Pfam.
1.4 Frequently Asked Questions (FAQs)

- What is Pfam?
- What is on a family page?
- What is a clan?
- What criteria do you use for putting families into clans?
- Is possible to build Wise2 with HMMER3 support?
- What happened to the Pfam_ls and Pfam_fs files?
- How can I search Pfam locally?
- Why doesn’t Pfam include my sequence?
- Why is there apparent redundancy of UniProtKB IDs in the full-length FASTA sequence file?
- How many accurate alignments do you have?
- How can I submit a new domain?
- Can I search my protein against Pfam?
- Why do I get slightly different results when I search my sequence against Pfam versus when I look up a sequence on the Pfam website?
- What is the difference between the ‘-’ and ‘.’ characters in your full alignments?
- What do the SS lines in the alignment mean?
- Why don’t you have domain YYYY in Pfam!
- Are there other databases which do this?
- So which database is better?

1.4.1 What is Pfam?

Pfam is a collection of multiple sequence alignments and profile hidden Markov models (HMMs). Each Pfam profile HMM represents a protein family or domain. By searching a protein sequence against the Pfam library of profile HMMs, you can determine which domains it carries i.e. its domain architecture. Pfam can also be used to analyse proteomes and questions of more complex domain architectures.

For each Pfam accession we have a family page, which can be accessed in several ways: from the ‘View a Pfam Family’ search box on the HOME page, by clicking on any graphical image of a domain, by searching for a particular family using the ‘Keyword Search’ box on the top right hand corner of most website pages, or by pasting the family identifier or accession into the ‘JUMP TO’ box that is present on most pages in the site.
1.4.2 What is on a family page?

From the family page you can view the Pfam annotation for a family. We also provide access to many other sources of information, including annotation from the InterPro database, where available, cross-links to other databases and other tools for protein analysis. Since release 25.0 we have also started displaying relevant articles from Wikipedia where available.

Via the tabs on the left-hand side of the page, you can view:

- the domain architectures in which this family is found
- the alignments for the family in various formats, including alignments of matches to UniProKB and representative proteomes, as well as in ‘heat-map’ format. All alignments can be downloaded
- the phylogenetic and species distribution trees, either as a traditional, interactive tree or as a “sunburst” plot
- the profile HMM logo
- the structural information for each family where available

1.4.3 What is a clan?

Some of the Pfam families are grouped into clans. Pfam defines a clan as a collection of families that have arisen from a single evolutionary origin. Evidence of their evolutionary relationship can be in the form of similarity in tertiary structures, or, when structures are not available, from common sequence motifs. The seed alignments for all families within a clan are aligned and the resulting alignment (called the clan alignment) can be accessed from a link on the clan page. Each clan page includes a clan alignment, a description of the clan and database links, where appropriate. The clan pages can be accessed by following a link from the family page, or alternatively they can be accessed by clicking on ‘clans’ under the ‘browse’ by menu on the top of any Pfam page.

1.4.4 What criteria do you use for putting families into clans?

We use a variety of measures. Where possible we do use structures to guide us and that is always the gold standard. In the absence of a structure we use:

- profile comparisons such as HHsearch
- the fact that a sequence significantly matches two profile HMMs in the same region of the sequence
- a method called SCOOP, that looks for common matches in search results that may indicate a relationship

All of this information is used by one of our curators to make a decision about where families are related and we strive to find information in literature that support the relationship, e.g. common function.

1.4.5 Is possible to build Wise2 with HMMER3 support?

The way we get round the problem with the difference in HMMER versions, is to convert the profile HMMs that are in HMMER3 format to HMMER2 format using the HMMER3 program “hmconvert” (with -2) flag. To make the searches feasible, we screen the DNA for potential domains using ncbi-blast and the Pfam-A.fasta as a target library. GeneWise is then used to calculate a subset of profile HMMs against the DNA. There is some down-weighting of the bits-per-position between H2 and H3 HMMs that the conversion does not account for, leading inevitably to some false negatives for some families/sequences. However, until GeneWise is patched to deal with HMMER3 models, this is the best course of action.
1.4.6 What happened to the Pfam_ls and Pfam_fs files?

In the past, each Pfam family was represented by two profile-hidden Markov models (HMMs). One of these could match partially to a family and was called local or fs mode, the other required a sequence to match to the whole length of the profile HMM, and was called glocal or ls mode. With HMMER2, we found that the combination of the two models gave us the most sensitive searches. However, HMMER3 models are only available for searching in local (fs) mode. Because of the improvements in HMMER3, this single model is as sensitive as the two combined HMMER2 models. This means that we no longer provide two profile HMM libraries called ‘HMM_ls’ and ‘HMM_fs’. Instead, a single library is available called ‘Pfam-A hmm’.

1.4.7 How can I search Pfam locally?

If you have a large number of sequences or you don’t want to post your sequence across the web, you can search your sequence locally using the ‘pfam_scan.pl’ script.

In terms of HMMs and formats, Pfam is based around the HMMER package. This will need to be installed on your local machine. You will need also to download the Pfam profile HMM libraries from the FTP site, as well as a few modules from CPAN, most notably Moose.

Full details on how to get ‘pfam_scan.pl’ up and running can be found on our FTP site.

1.4.8 Why doesn’t Pfam include my sequence?

Pfam is built from a fixed release of UniProtKB. At each Pfam release we incorporate sequences from the latest release of UniProtKB. This means that, at any time, the sequences used by Pfam might be several months behind those in the most up-to-date versions of the sequence databases. If your sequence isn’t in Pfam, you can still find out what domains it contains by pasting it into the sequence search box on the search page.

1.4.9 Why is there apparent redundancy of UniProtKB IDs in the full-length FASTA sequence file?

A given Pfam family may match a single protein sequence multiple times, if the domain/family is a repeating unit, for example, or when the profile HMM matches only to short stretches of the sequence but matches several times. In such cases the FASTA file with the full length sequences will contain multiple copies of the same sequence.

1.4.10 How many accurate alignments do you have?

Release 33.1 has 18259 families. Over 75.1% of the proteins in SWISSPROT 2019_08 and TrEMBL 2019_08 have at least one match to a Pfam-A family.

1.4.11 How can I submit a new domain?

If you know of a domain that is not present in Pfam, you can submit it to the Pfam helpdesk and we will endeavour to build a Pfam entry for it. We ask that you supply us with a multiple sequence alignment of the domain (please send the alignment file as a text file (e.g. .txt) and not in the format of a specific application such as Microsoft Word (e.g. a .doc file), and associated literature evidence if available.

1.4. Frequently Asked Questions (FAQs)
1.4.12 Can I search my protein against Pfam?

Of course! Please use this search form.

1.4.13 Why do I get slightly different results when I search my sequence against Pfam versus when I look up a sequence on the Pfam website?

When a sequence region has overlapping matches to more than one family within the same clan, we only show one of those matches. If the sequence region is also in the seed alignment for a family, only the match to that family is shown. Otherwise we show the family that corresponds to the match with the lowest E-value.

There are cases where a sequence region is in the seed alignment of a Pfam family (family A), but it does not have a significant match to that family’s profile HMM. Occasionally, the same sequence region has a significant match to another family (family B) in the same clan. In this situation, the Pfam website will not show the match to family B as it is present in the seed alignment of family A. The sequence search will however show the match to family B as the seed alignment information is unknown. This scenario, where the sequence search shows a match that the Pfam website does not, is very rare (affecting less than 0.01% of all matches in the Pfam database).

1.4.14 What is the difference between the ‘-‘ and ‘.’ characters in your full alignments?

The ‘-‘ and ‘.’ characters both represent gap characters. However they do tell you some extra information about how the profile HMM has generated the alignment. The ‘-‘ symbols are where the alignment of the sequence has used a delete state in the profile HMM to jump past a match state. This means that the sequence is missing a column that the profile HMM was expecting to be there. The ‘.’ character is used to pad gaps where one sequence in the alignment has sequence from the profile HMMs insert state. See the alignment below where both characters are used. The profile HMM states emitting each column are shown. Note that residues emitted from the Insert (I) state are in lower case.

```
FBRL_XENLA/86–131    RKVIVEPHR. HEGIFICR GK. EDALVTKNLVPGE SVYEGK RISVEDGE
Q9ZSE3_EUGGR/37–85   --AVVVVEPHKvHAGIFVSRGksEDSLATLNVLPGVS VYEGKRVTETTD
FBRL_MOUSE/90–135    KNMVVEPHR. HEGVFICRGK. EDALVTKNLVPGE SVYEGKRVSIGED
FBRL_TETTH/64–108    KTIIVK–HR. LEGVFICQ. QEALVTKNFPGESVYNEKRM SVEENG
HMM STATES          MMMMMMMMMMMMIMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
```

1.4.15 What do the SS lines in the alignment mean?

These lines are structural information. The SS stands for secondary structure, and this is taken from DSSP. The following list gives the definitions for each code letter:

- C: random Coil
- H: alpha-helix
- G: 3(10) helix
- I: pi-helix
- E: hydrogen bonded beta-strand (extended strand)
- B: residue in isolated beta-bridge
- T: h-bonded turn (3-turn, 4-turn, or 5-turn)
- S: bend (five-residue bend centered at residue i)
1.4.16 Why don’t you have domain YYYY in Pfam!

We are very keen to be alerted to new domains. If you can provide us with a multiple sequence alignment then we will try hard to incorporate it into the database. If you know of a domain, but don’t have a multiple sequence alignment, we still want to know, for simple families just one sequence is enough. Again contact the Pfam helpdesk.

1.4.17 Are there other databases which do this?

To a certain extent yes, there are a number of “second generation” databases which are trying to organise protein space into evolutionarily conserved regions. Examples include:

- **PROSITE** This originally was based around regular expression patterns but now also includes profiles.
- **PRINTS** This is based around protein “finger-prints” of a series of small conserved motifs making up a domain.
- **SMART** This is a database concentrating on extracellular modules and signaling domains.
- **InterPro** Combines information from Pfam, Prints, SMART, Prosite and PRODOM.
- **CDD** The Conserved Domain Database is derived from Pfam and SMART databases.

1.4.18 So which database is better?

As with everything, it depends on your problem: we would certainly suggest using more than one method. Pfam is likely to provide more interpretable results, with crisp definitions of domains in a protein.

1.5 Glossary

- Alignment coordinates
- Architecture
- Clan
- Domain
- Domain score
- DUF
- Envelope coordinates
- Family
- Full alignment
- Gathering threshold (GA)
- HMMER
- Hidden Markov model (HMM)
- Motif
- Noise cutoff (NC)
- Pfam-A
1.5.1 Alignment coordinates

HMMER3 reports two sets of domain coordinates for each profile HMM match. The envelope coordinates delineate the region on the sequence where the match has been probabilistically determined to lie, whereas the alignment coordinates delineate the region over which HMMER is confident that the alignment of the sequence to the profile HMM is correct. Our full alignments contain the envelope coordinates from HMMER3.

1.5.2 Architecture

The collection of domains that are present on a protein.

1.5.3 Clan

A collection of related Pfam entries. The relationship may be defined by similarity of sequence, structure or profile-HMM.

1.5.4 Domain

A structural unit.

1.5.5 Domain score

The score of a single domain aligned to a profile HMM. Note that, for HMMER2, if there was more than one domain, the sequence score was the sum of all the domain scores for that Pfam entry. This is not quite true for HMMER3.

1.5.6 DUF

Domain of unknown function.
1.5.7 Envelope coordinates
See Alignment coordinates.

1.5.8 Family
A collection of related protein regions.

1.5.9 Full alignment
An alignment of the set of related sequences which score higher than the manually set threshold values for the profile HMMs of a particular Pfam entry.

1.5.10 Gathering threshold (GA)
Also called the gathering cut-off, this value is the search threshold used to build the full alignment. The gathering threshold is assigned by a curator when the family is built. The GA is the minimum score a sequence must attain in order to belong to the full alignment of a Pfam entry. For each Pfam profile HMM we have two GA cutoff values, a sequence cutoff and a domain cutoff.

1.5.11 HMMER
The suite of programs that Pfam uses to build and search profile HMMs. Since Pfam release 24.0 we have used HMMER version 3 to make Pfam. See the HMMER site for more information.

1.5.12 Hidden Markov model (HMM)
A profile HMM is a probablistic model. In Pfam we use profile HMMs to transform the information contained within a multiple sequence alignment into a position-specific scoring system. We search our profile HMMs against the UniProt protein database to find homologous sequences.

1.5.13 Motif
A short unit found outside globular domains.

1.5.14 Noise cutoff (NC)
The bit scores of the highest scoring match not in the full alignment.
1.5.15 Pfam-A

A profile HMM based hand curated Pfam entry which is built using a small number of representative sequences. We manually set a threshold value for each profile-HMM and search our models against the UniProtKB database. All of the sequences which score above the threshold for a Pfam entry are included in the entry’s full alignment.

1.5.16 Pfam-B

A set of unannotated, computationally generated multiple sequence alignments. They are one of the sources we use for creating Pfam-A entries.

1.5.17 Posterior probability

HMMER reports a posterior probability for each residue that matches a ‘match’ or ‘insert’ state in the profile HMM. A high posterior probability shows that the alignment of the amino acid to the match/insert state is likely to be correct, whereas a low posterior probability indicates that there is alignment uncertainty. This is indicated on a scale with ‘*' being 10, the highest certainty, down to 1 being complete uncertainty. Within Pfam we display this information as a heat map view, where green residues indicate high posterior probability, and red ones indicate a lower posterior probability.

1.5.18 Repeat

A short unit which is unstable in isolation but forms a stable structure when multiple copies are present.

1.5.19 Seed alignment

An alignment of a set of representative sequences for a Pfam entry. We use this alignment to construct the profile HMMs for the Pfam entry.

1.5.20 Sequence score

The total score of a sequence aligned to a profile HMM. If there is more than one domain, the sequence score is the sum of all the domain scores for that Pfam entry. If there is only a single domain, the sequence and the domains score for the protein will be identical. We use the sequence score to determine whether a sequence belongs to the full alignment of a particular Pfam entry.

1.5.21 Trusted cutoff (TC)

The bit scores of the lowest scoring match in the full alignment.
1.6 Pfam scores

1.6.1 E-values and Bit-scores

Pfam-A is based around hidden Markov model (HMM) searches, as provided by the HMMER3 package. In HMMER3, like BLAST, E-values (expectation values) are calculated. The E-value is the number of hits that would be expected to have a score equal to or better than this value by chance alone. A good E-value is much less than 1. A value of 1 is what would be expected just by chance. In principle, all you need to decide on the significance of a match is the E-value.

E-values are dependent on the size of the database searched, so we use a second system in-house for maintaining Pfam models, based on a bit score (see below), which is independent of the size of the database searched. For each Pfam family, we set a bit score gathering (GA) threshold by hand, such that all sequences scoring at or above this threshold appear in the full alignment. It works out that a bit score of 20 equates to an E-value of approximately 0.1, and a score 25 of to approximately 0.01. From the gathering threshold both a “trusted cutoff” (TC) and a “noise cutoff” (NC) are recorded automatically. The TC is the score for the next highest scoring match above the GA, and the NC is the score for the sequence next below the GA, i.e. the highest scoring sequence not included in the full alignment.

1.6.2 Sequence versus domain scores

There’s an additional wrinkle in the scoring system. HMMER3 calculates two kinds of scores, the first for the sequence as a whole and the second for the domain(s) on that sequence. The “sequence score” is the total score of a sequence aligned to the model (the HMM); the “domain score” is the score for a single domain — these two scores are virtually identical where only one domain is present on a sequence. Where there are multiple occurrences of the domain on a sequence any individual match may be quite weak, but the sequence score is the sum of all the individual domain scores, since finding multiple instances of a domain increases our confidence that that sequence belongs to that protein family, i.e. truly matches the model.

1.6.3 Meaning of bit-score for non-mathematicians

A bit score of 0 means that the likelihood of the match having been emitted by the model is equal to that of it having been emitted by the Null model (by chance). A bit score of 1 means that the match is twice as likely to have been emitted by the model than by the Null. A bit score of 2 means that the match is 4 times as likely to have been emitted by the model than by the Null. So, a bit score of 20 means that the match is 2 to the power 20 times as likely to have been emitted by the model than by the Null.

1.7 Citing Pfam

1.7.1 Pfam References


1.7.2 Book Chapters on Pfam


1.8 Pfam Annotation in Wikipedia

The Pfam consortium coordinates the annotation of Pfam families via Wikipedia. This is some background on the process.

1.8.1 A new approach to annotation

Pfam families have traditionally been annotated by our curators as they were added into the Pfam database, but the annotation step has become by far the most time-consuming part of building a Pfam family. As we adapt the Pfam model to cope with the dramatic increases in sequence data that are on the horizon, we have had to consider ways to make this step quicker and more efficient. We are also striving constantly to improve the quality and depth of our annotations, and to this end we have now adopted the Wikipedia model of annotation that was pioneered by the Pfam resource.

In this approach we will gradually reduce the prominence of our traditional, curator-produced family annotations, replacing them with Wikipedia articles. Starting from Pfam release 25.0, you will see that some family pages in the Pfam website show Wikipedia content rather than our own annotations. Ultimately we hope to be able to assign a detailed, high-quality Wikipedia article in Pfam, but we need the help of the Pfam and Wikipedia communities to make this happen.

1.8.2 Wikipedia content in the Pfam website

When we build a new Pfam family, we try to find a Wikipedia article that describes the family and provides what we feel to be a valuable annotation for it. If we can’t find a suitable article we will, in many cases, generate a new Wikipedia article ourselves. Hence, most new families will be assigned a Wikipedia article as soon as they are created.

Where a Wikipedia article has been assigned to a family, the main summary tab of the family page will show the content of the article, rather than the Pfam annotation. You can still see the old Pfam annotation, along with the Interpro annotation text, in adjacent tabs. Note that we will no longer be updating Pfam annotation text for any family that has a Wikipedia article. Instead we will try to make improvements or corrections to the article itself and will encourage our users to make improvements and corrections themselves.
Unfortunately, while new Pfam families will have Wikipedia articles assigned when they are created, we simply do not have the resources to be able to revisit older, pre-existing Pfam families. The family pages for these families still continue to show the Pfam annotation, but we hope to replace this with Wikipedia content wherever possible.

### 1.8.3 Contributing annotations

You can now contribute to the improvement of Pfam annotations in several ways. First and foremost, if you come across a family that does not yet have a Wikipedia article assigned to it, we would really like to add one. If you know of an article that would provide a useful description of a family, please let us know via our annotation submission form (click the “Add annotation” button on the family page) or by email. You can find our email address at the bottom of every page.

One of the advantages of using Wikipedia to provide our annotations is that any user can now contribute to that annotation text. In many cases, families that do not yet have a Wikipedia article can be assigned an article that already exists. In some cases, however, no suitable article exists, and in that case we would encourage you to consider adding one to Wikipedia yourself.

### 1.8.4 Editing Wikipedia articles

You can see these notes on every family page by clicking “More” on the Wikipedia content tab.

**Before you edit for the first time**

Wikipedia is a free, online encyclopedia. Although anyone can edit or contribute to an article, Wikipedia has some strong editing guidelines and policies, which promote the Wikipedia standard of style and etiquette. Your edits and contributions are more likely to be accepted (and remain) if they are in accordance with this policy.

You should take a few minutes to view the following pages:

- Five pillars
- Policies and guidelines
- Wikipedia help contents
- Wikipedia Tips
- Editing help

**How your contribution will be recorded**

Anyone can edit a Wikipedia entry. You can do this either as a new user or you can register with Wikipedia and log on. When you click on the “Edit Wikipedia article” button, your browser will direct you to the edit page for this entry in Wikipedia. If you are a registered user and currently logged in, your changes will be recorded under your Wikipedia user name. However, if you are not a registered user or are not logged on, your changes will be logged under your computer’s IP address. This has two main implications. Firstly, as a registered Wikipedia user your edits are more likely seen as valuable contribution (although all edits are open to community scrutiny regardless). Secondly, if you edit under an IP address you may be sharing this IP address with other users. If your IP address has previously been blocked (due to being flagged as a source of ‘vandalism’) your edits will also be blocked. You can find more information on this and creating a user account at Wikipedia.

If you have problems editing a particular page, contact us at pfam-help@ebi.ac.uk and we will try to help.
Does Pfam agree with the content of the Wikipedia entry?

Pfam has chosen to link families to Wikipedia articles. In some cases we have created or edited these articles but in many other cases we have not made any direct contribution to the content of the article. The Wikipedia community does monitor edits to try to ensure that (a) the quality of article annotation increases, and (b) vandalism is very quickly dealt with. However, we would like to emphasise that Pfam does not curate the Wikipedia entries and we cannot guarantee the accuracy of the information on the Wikipedia page.

Contact us

If you have problems editing or experience problems with these pages please contact us at pfam-help@ebi.ac.uk

1.9 Linking to the Pfam website

1.9.1 Linking to family pages

You can refer to Pfam families either by accession or ID. You can also refer to a family by “entry”, although this is a convenience that should be used only if you’re not sure if what you have is an accession or an ID.

Pfam accession numbers are more stable between releases than IDs and we strongly recommend that you link by accession number.

Here are some examples of linking to Pfam families:

**Directly, using either accession or ID:** /family/PF00002 or /family/7tm_2

**Using a parameter, by accession:** /family?acc=PF00002

**Using a parameter, by ID:** /family?id=7tm_2

**Using the “entry” parameter with either accession or ID:** /family?entry=PF00002 or /family?entry=7tm_2

1.9.2 Linking to protein sequence pages

As for Pfam family pages, you can refer to protein sequence pages by accession, ID or entry. Protein IDs are unstable and do change between releases, so, again, we strongly recommend that you use protein accessions where possible.

Here are some examples of linking to protein sequence pages at EBI:

**Directly:** /protein/P15498 or /protein/VAV_HUMAN

**By accession:** /protein?acc=P15498

**By ID:** /protein?id=VAV_HUMAN

**Using “entry”:** /protein?entry=P15498 or /protein?entry=VAV_HUMAN
1.9.3 Linking to the “jump to” search

The Pfam website features a search tool that tries to guess the type of any accession or ID that it is given. For example, if given “VAV_HUMAN”, the search returns the URL for the protein sequence page for the VAV_HUMAN entry. If given “1w9h”, the search returns the URL for the PDB entry (structure) 1w9h.

You can use the “jump to” search if you need to link to Pfam but can’t be sure what type of accession or ID you will be using in your link. By default, the search returns the URL that it has found, as a simple, plain text HTTP response. Adding the parameter redirect=1 will make the “jump to” tool redirect to the URL that it finds or, if it couldn’t find an appropriate URL, to the Pfam homepage.

Return URL: /search/jump?entry=P15498
Redirect: /search/jump?entry=P15498&redirect=1

Note that, although it may be convenient to link to Pfam using this search tool, there is no error reporting for your users if the search fails to find an appropriate URL in the Pfam site. It is much safer to link directly to the correct section of the site. Please contact us if you need help with building specific links.

1.10 A Guide to the Pfam Domain Graphics

One of the visualisations provided by the Pfam website is a graphical representation of the features found within a sequence, termed domain graphics. There are a variety of different shapes and styles and each one has a particular meaning. This page gives an in-depth description of the elements of Pfam domain graphics.

The library that generates the images in this page and throughout the Pfam site uses a JSON string to describe the domain graphic. Each of the example graphics in this page is followed by a link that can be used to show the JSON snippet that produced it.

1.10.1 Generating graphics

You can try generating your own graphics using the domain graphics generator. The JSON descriptions in this page can be pasted directly into the generator to produce the graphics that you see here.

You can also generate the domain graphics for specific sequences, using the UniProt graphics generator.

1.10.2 Using the domain graphics code

Finally, if you would like to use the javascript library in your own site, we have put together an example page, showing how to set up the library and its dependencies. Look at the source code of the page for an explanation.

1.10.3 The sequence

The base sequence, undecorated by any domains or features, is represented by a plain grey bar:

```json
{
    "length" : "400"
}
```
The length of the domain graphic that is drawn is proportional to the length of the sequence itself. The graphics in this page are drawn with a X-scale of 0.5 pixels per amino-acid, so that a 400 residue sequence will result in a 200 pixel-wide image. Any domains or features which are drawn on the sequence are also scaled by the same factor.

1.10.4 Pfam-A

The high quality, curated Pfam-A domains are classified into one of six different types: family, domain, coiled-coil, disordered, repeat and motif (for more details see Summary). These different classification types are rendered slightly differently.

Family/domain

It is possible for a sequence to match either the full length of a Pfam HMM (a full length match), or to match a portion of an HMM (a fragment match). The two types of match are rendered differently.

Both family and domain entries are rendered as rectangles with curved ends when the sequence is a full length match. Different types of domain are displayed with different colours. When the domain image is long enough, the domain name is shown within the domain itself. In most cases, you can click on the domains to visit the “family page” for that domain. Moving the mouse over the domain image should also display a tooltip showing the domain name, as well as the start and end positions of the domain.

```
{
  "length" : "400",
  "regions" : [
    {
      "type" : "pfama",
      "text" : "Domain",
      "colour" : "#9999ff",
      "display" : "true",
      "startStyle" : "curved",
      "endStyle" : "curved",
      "start" : "40",
      "end" : "200",
      "aliStart" : "50",
      "aliEnd" : "175"
    },
    {
      "type" : "pfama",
      "text" : "LongFamilyNamesNotShown",
      "colour" : "#399",
      "display" : true,
      "startStyle" : "straight",
      "endStyle" : "straight",
      "start" : "210",
      "end" : "250",
      "aliStart" : "215",
      "aliEnd" : "245"
    }
  ]
}
```

From Pfam 24.0 onwards, Pfam has been generated using HMMER3, which introduces the concept of “envelope coordinates” for a match. Envelope regions are represented in domain graphics as lighter coloured regions. The
graphic above shows short envelope regions at the ends of both domains.

When the sequence does not match the full length of the HMM that models a Pfam entry, matching domain fragments are shown. When a sequence match does not pass through the first position in the HMM, the N-terminal side of the domain graphic is drawn with a jagged edge instead of a curved edge. Similarly, when a sequence match does not pass through the last position of the HMM, the C-terminal side of the domain graphic is drawn with a jagged edge. In some rarer cases, the sequence match may not pass through either of the first or last positions of the HMM, in which case both sides are drawn with jagged edges. Examples of all three cases are shown here:
Repeat/motif

Repeats and motifs are types of Pfam domain which do not form independently folded units. In order to distinguish them from domains of type family and domain, repeats and motifs are represented by rectangles with straight edges. As for families and domains, partial matches are represented with jagged edges.

```json
{
  "length": "200",
  "regions": [
    {
      "type": "pfama",
      "text": "HEAT",
      "colour": "#1fc01f",
      "display": "true",
      "startStyle": "straight",
      "endStyle": "straight",
      "start": "2",
      "end": "34"
    },
    {
      "type": "pfama",
      "text": "HEAT",
      "colour": "#1fc01f",
      "display": "true",
      "startStyle": "straight",
      "endStyle": "straight",
      "start": "82",
      "end": "118"
    },
    {
      "type": "pfama",
      "text": "HEAT",
      "colour": "#1fc01f",
      "display": "true",
      "startStyle": "straight",
      "endStyle": "straight",
      "start": "120",
      "end": "155"
    },
    {
      "type": "pfama",
      "text": "HEAT",
      "colour": "#1fc01f",
      "display": "true",
      "startStyle": "straight",
      "endStyle": "straight",
      "start": "159",
      "end": "195"
    }
  ]
}
```
Discontinuous nested domains

Some domains in Pfam are disrupted by the insertion of another domain (or domains) within them. A number of names have been given to this arrangement: *discontinuous* (referring to the outer domain), *inserted* or *nested* (both referring to the inner domain). For example, in many sequences containing an IMPDH domain, the IMPDH domain is continuous along the primary sequence. However, in some cases the linear sequence of the IMPDH domain is broken by the insertion of a CBS domain, as shown below.

Where three-dimensional structures are available for representatives of a Pfam domain, it is generally clear that the three-dimensional arrangement of the domain containing the nested domain is maintained. Typically the nested domain is found inserted within a surface exposed loop, having little or no effect on the structure of the other domain. Such an arrangement explains why and how these nested domains can be functionally tolerated.

To represent this arrangement of domain graphically, the discontinuous domain is represented in two parts (as shown below). These two parts are joined by a line bridging them.

```json
{
    "length": "200",
    "regions": [
        {
            "type": "pfama",
            "text": "IMPDH",
            "colour": "#1fc01f",
            "display": "true",
            "startStyle": "curved",
            "endStyle": "jagged",
            "start": "5",
            "end": "80"
        },
        {
            "type": "pfama",
            "text": "CBS",
            "colour": "#c00f0f",
            "display": "true",
            "startStyle": "curved",
            "endStyle": "curved",
            "start": "81",
            "end": "135"
        },
        {
            "type": "pfama",
            "text": "IMPDH",
            "colour": "#1fc01f",
            "display": "true",
            "startStyle": "jagged",
            "endStyle": "curved",
            "start": "136",
            "end": "197"
        }
    ],
    "markups": [
        {
            "type": "Nested",
            "colour": "#000000",
            "display": "true",
            "start": "5",
            "end": "197"
        }
    ]
}
```
1.10.5 Other sequence motifs

In addition to domains, smaller sequences motifs are represented by the domain graphics. Currently the following motifs are represented: signal peptides, low complexity regions, coiled-coils and transmembrane regions. These usually take lower priority than other regions that are drawn and they are therefore often obscured by, for example, a Pfam-A graphic being drawn over the top of them. An example of each motif is shown here.

```
{
    "length": "200",
    "motifs": [
        {
            "type": "sig_p",
            "colour": "#ff9c00",
            "display": true,
            "start": 1,
            "end": 27
        },
        {
            "type": "low_complexity",
            "colour": "#0FF",
            "display": true,
            "start": 39,
            "end": 47
        },
        {
            "type": "low_complexity",
            "colour": "#0FF",
            "display": true,
            "start": 67,
            "end": 76
        },
        {
            "type": "coiled_coil",
            "colour": "#9cff00",
            "display": true,
            "start": 103,
            "end": 123
        },
        {
            "type": "transmembrane",
            "colour": "#F00",
            "display": true,
            "start": 155,
            "end": 175
        }
    ]
}
```
Signal peptides

Signal peptides are short regions (<60 residues long) found at the N-terminus of proteins, which direct the post-translational transport of a protein and are subsequently removed by peptidases. More specifically, a signal peptide is characterised by a short hydrophobic helix (approximately 7-15 residues). This helix is preceded by a slight positively charged region of highly variable length (approximately 1-12 residues). Between the hydrophobic helix and the cleavage site is a somewhat polar and uncharged region, of between 3 and 8 amino-acids. In Pfam, we use Phobius for the prediction of signal peptides and represent them graphically by a small orange box.


Low complexity regions

Low complexity regions are regions of biased sequence composition, usually comprised of different types of repeats. These regions have been shown to be functionally important in some proteins, but they are generally not well understood and are masked out to focus on globular domains within the protein.

Within Pfam, we use SEG to calculate low complexity regions in Pfam. The presence of a low complexity region is indicated by a cyan rectangle.


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.


IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. Z. Dosztanyi, V. Csizmok, P. Tompa, I. Simon Bioinformatics (2005) 21:3433-3434
Coiled-coils

Coiled coils are motifs found in proteins that structurally form alpha-helices that wrap or wind around each other. Normally, two to three helices are involved, but cases of up to seven alpha-helices have been reported. Coiled-coils are found in a wide variety of proteins, many functionally very important. In Pfam we use ncoils, to identify these motifs. Coiled-coils are represented by a small lime-green rectangle.

Transmembrane regions

Integral membrane proteins contain one or more transmembrane regions that are comprised of an alpha-helix that passes through or “spans” a membrane. Transmembrane helices are quite variable in length, with the average being about 20 amino-acids in length. Again, Phobius is used for the prediction of transmembrane regions, which are represented by a red rectangle.

1.10.6 Other Sequence features

Below is a demonstration of how disulphide bridges and active site residues* are represented in Pfam. Each of these features can appear above or below the sequence, but in this case the disulphide bridges are shown above the sequence and the active site residues below the line.

```json
{
"length": "400",
"regions": [
{
"colour": "#1fc01f",
"endStyle": "curved",
"startStyle": "curved",
"display": true,
"end": "104",
"href": "/family/Inhibitor_I29",
"text": "Inhibitor_I29",
"metadata": {
"scoreName": "e-value",
"score": "1.3e-38",
"description": "Inhibitor_I29",
"accession": "PF08246",
"end": "104",
"database": "pfam",
"identifier": "Inhibitor_I29",
"type": "Domain",
"start": "48"
},
"type": "pfama",
"start": "48"
},
{
"colour": "#c00f0f",
"endStyle": "curved",
"startStyle": "curved",
"display": true,
"end": "104",
"href": "/family/Inhibitor_I29",
"text": "Inhibitor_I29",
"metadata": {
"scoreName": "e-value",
"score": "1.3e-38",
"description": "Inhibitor_I29",
"accession": "PF08246",
"end": "104",
"database": "pfam",
"identifier": "Inhibitor_I29",
"type": "Domain",
"start": "48"
},
"type": "pfama",
"start": "48"
}
]
}
```
"end": "343",
"href": "/family/Peptidase_C1",
"text": "Peptidase_C1",
"modellLength": "307",
"metadata": {
  "scoreName": "e-value",
  "score": "1.3e-38",
  "description": "Peptidase_C1",
  "accession": "PF00112",
  "end": "343",
  "database": "pfam",
  "identifier": "Peptidase_C1",
  "type": "Domain",
  "start": "134"
},
"type": "pfama",
"start": "134"
},
"markups": [
{
  "lineColour": "#CCC",
  "colour": "#CCC",
  "display": "true",
  "end": "196",
  "v_align": "top",
  "metadata": {
    "database": "pfam",
    "type": "Disulphide, 155-196",
    "end": "196",
    "start": "155"
  },
  "type": "Disulphide",
  "start": "155"
},
{
  "lineColour": "#CCC",
  "colour": "#CCC",
  "display": "true",
  "end": "228",
  "v_align": "top",
  "metadata": {
    "database": "pfam",
    "type": "Disulphide, 189-228",
    "end": "228",
    "start": "189"
  },
  "type": "Disulphide",
  "start": "189"
},
{
  "lineColour": "#CCC",
  "colour": "#CCC",
  "display": "true",
  "end": "333",
  "v_align": "top",
  "metadata": {
    "database": "pfam",
    "type": "Disulphide, 333-333",
    "end": "333",
    "start": "333"
  },
  "type": "Disulphide",
  "start": "333"
}
]
1.10. A Guide to the Pfam Domain Graphics
Disulphide bridges

Disulphide bridges play a fundamental role in the folding and stability of some proteins. They are formed by covalent bonding between the thiol groups from two cysteine residues. The disulphide bridge annotations used in Pfam come from UniProt and are represented by a solid bridge-shaped line. When multiple disulphide bonds occur, the heights of the bridges are adjusted to avoid overlaps between them. Inter-protein disulphides are represented by single vertical lines. As always, moving the mouse over the “bridge graphic” shows the details of the bond in a tooltip.

Active site residues

Within an enzyme, a small number of residues are directly involved in catalysis of a reaction. These are termed active site residues. Within Pfam there are three categories of active site: those that are experimentally determined, those that are predicted by UniProt and those predicted by Pfam. All three types are represented by a “lollipop” with a diamond head. The head is coloured red, pink and purple for each of the three types respectively.

Pfam-predicted active sites are determined by using the experimental data and transferring these annotations through a Pfam alignment.

“Lollipops”

A wide range of different lollipop styles can be create by combining different line and head colours with different drawing styles. The lollipop head can be drawn as a square, circle or diamond, as a simple coloured bar, or as an arrow (pointing away from the sequence) or a “pointer” (an arrow pointing towards the sequence).
1.10. A Guide to the Pfam Domain Graphics
1.10.7 Tooltips

If appropriate metadata are present in the sequence description, the domain graphics library can also add tooltips to the image. The example below shows a domain graphic and its description includes the necessary metadata for generating tooltips.

```json
{
    "length": "950",
    "regions": [
        {
            "modelStart": "5",
            "modelEnd": "292",
            "colour": "#2dcf00",
            "endStyle": "jagged",
            "startStyle": "jagged",
            "display": true,
            "end": "361",
            "aliEnd": "361",
            "href": "/family/PF00082",
            "text": "Peptidase_S8",
            "modelLength": "307",
            "metadata": {
                "scoreName": "e-value",
                "score": "1.3e-38",
                "description": "Subtilase family",
                "accession": "PF00082",
                "end": "587",
                "database": "pfam",
                "aliEnd": "573",
                "identifier": "Peptidase_S8",
                "type": "Domain",
                "aliStart": "163",
                "start": "159"
            }
        },
        {
            "modelStart": "5",
            "modelEnd": "292",
            "colour": "#2dcf00",
            "endStyle": "jagged",
            "startStyle": "jagged",
            "display": true,
            "end": "587",
            "aliEnd": "573",
        }
    ]
}
```

(continues on next page)
1.10. A Guide to the Pfam Domain Graphics
"aliEnd": "726",
"href": "/family/PF06280",
"text": "DUF1034",
"modellLength": "112",
"metadata": {
  "scoreName": "e-value",
  "score": "1.1e-13",
  "description": "Fn3-like domain (DUF1034)",
  "accession": "PF06280",
  "end": "726",
  "database": "pfam",
  "aliEnd": "726",
  "identifier": "DUF1034",
  "type": "Domain",
  "aliStart": "613",
  "start": "613"
},
"type": "pfama",
"aliStart": "613",
"start": "613"}
1.10. A Guide to the Pfam Domain Graphics
1.11 RESTful interface

This is an introduction to the “RESTful” interface to the Pfam website. REST (or Representation State Transfer) refers to a style of building websites which makes it easy to interact programmatically with the services provided by the site. A programmatic interface, commonly called an Application Programming Interface (API) allows users to write scripts or programs to access data, rather than having to rely on a browser to view a site.

1.11.1 Basic concepts

URLs

A RESTful service typically sends and receives data over HTTP, the same protocol that’s used by websites and browsers. As such, the services provided through a RESTful interface are identified using URLs.

In the Pfam website we use the same basic URL to provide both the standard HTML representation of Pfam data and the alternative XML representation. To see the data for a particular Pfam-A family, you would visit the following URL in your browser:

/family/Piwi

To retrieve the data in XML format, just add an extra parameter, output=xml, to the URL:

/family/Piwi?output=xml

The response from the server will now be an XML document, rather than an HTML page.

Sending requests

Using curl

Although you can use a browser to retrieve family data in XML format, it’s most useful to send requests and retrieve XML programmatically. The simplest way to do this is using a Unix command line tool such as curl:

```
curl -LH 'Expect:' -F output=xml '/family/Piwi'
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- information on Pfam-A family PF02171 (Piwi), generated: 16:35:52 26-Oct-2009 -->
<pfam xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="https://pfam.xfam.org/"
  xsi:schemaLocation="https://pfam.xfam.org/
  https://pfam.xfam.org/static/documents/schemas/pfam_family.xsd"
  release="24.0"
  release_date="2009-10-07">
  <entry entry_type="Pfam-A" accession="PF02171" id="Piwi">
    ...
  </entry>
</pfam>
```

Note: we have recently changed the web server that we use for serving the Pfam site. Due to a bug in the server itself, requests that come from curl are normally rejected. The current work-around is to add an extra parameter to the curl command line: -H ‘Expect:’. This should avoid problems with requests being rejected.
Using a script

Most programming languages have the ability to send HTTP requests and receive HTTP responses. A Perl script to retrieve data about a Pfam family might be as trivial as this:

```perl
#!/usr/bin/perl
use strict;
use warnings;
use LWP::UserAgent;
my $ua = LWP::UserAgent->new;
$ua->env_proxy;
my $res = $ua->get( '/family/Piwi?output=xml' );
if ( $res->is_success ) {
    print $res->content;
} else {
    print STDERR $res->status_line, "\n";
}
```

1.11.2 Retrieving data

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:

```perl
#!/usr/bin/perl
use strict;
use warnings;
use LWP::UserAgent;
use XML::LibXML;
my $ua = LWP::UserAgent->new;
$ua->env_proxy;
my $res = $ua->get( '/family/Piwi?output=xml' );
die "error: failed to retrieve XML: " . $res->status_line . "\n" unless $res->is_success;
my $xml = $res->content;
my $xml_parser = XML::LibXML->new();
my $dom = $xml_parser->parse_string( $xml );
my $root = $dom->documentElement();
my ( $entry ) = $root->getChildrenByTagName( 'entry' );
print 'accession: ' . $entry->getAttribute( 'accession' ) . "\n";
```

This script now prints out the accession for the family “Piwi” (PF02171).
1.11.3 Available services

The following is a list of the sections of the website which are currently available as RESTful services.

**Pfam ID/accession conversion**

This is a simple service to return the accession and ID for a Pfam family, given either the ID or accession as input. Any of the following URLs will return the same simple XML document:

- /family/acc?id=Piwi&output=xml
- /family/Piwi/acc?output=xml
- /family/id?output=xml&acc=PF02171
- /family/Piwi/id?output=xml
- /family?entry=Piwi&output=xml

```bash
curl -LH 'Expect:' -F output=xml '/family/Piwi/acc'
```

<?xml version="1.0" encoding="UTF-8"?>
<!-- information on Pfam-A family PF02171 (Piwi), generated: 16:37:09 26-Oct-2009 -->
<pfam xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="https://pfam.xfam.org/
      xsi:schemaLocation="https://pfam.xfam.org/
      https://pfam.xfam.org/static/documents/schemas/pfam_family.xsd"
      release="24.0"
      release_date="2009-10-07">
  <entry entry_type="Pfam-A" accession="PF02171" id="Piwi"/>
</pfam>

You can see the XML schema for this XML document [here](#).

Note that, as a convenience, you can also omit the output=xml parameter and the response will contain only the ID or accession, as a plain text string:

```bash
curl -LH 'Expect:' '/family/Piwi/acc'
```

PF02171

```bash
curl -LH 'Expect:' '/family/PF02171/id'
```

Piwi

**Pfam-A annotations**

You can retrieve a sub-set of the data in a Pfam-A family page as an XML document using any of the following styles of URL:

- /family?id=Piwi&output=xml
- /family?output=xml&acc=PF02171
- /family?entry=Piwi&output=xml
- /family/Piwi?output=xml

The last two styles, using the entry parameter or an extended URL, accept either accessions or identifiers. The accession/ID is case-insensitive in all cases.
This domain is found in the protein Piwi and its relatives. The function of this domain is the dsRNA guided hydrolysis of ssRNA. Determination of the crystal structure of Argonaute reveals that PIWI is an RNase H domain, and identifies Argonaute as Slicer, the enzyme that cleaves mRNA in the RNAi RISC complex [2]. In addition, Mg+2 dependence and production of 3'-OH and 5' phosphate products are shared characteristics of RNaseH and RISC. The PIWI domain core has a tertiary structure belonging to the RNase H family of enzymes. RNase H fold proteins all have a five-stranded mixed beta-sheet surrounded by helices. By analogy to RNase H enzymes which cleave single-stranded RNA guided by the DNA strand in an RNA/DNA hybrid, the PIWI domain can be inferred to cleave single-stranded RNA, for example mRNA, guided by double stranded siRNA.
You can see the XML schema for this XML document here.

Some Pfam families are removed or merged into others, in which case they become “dead” families. If you try to retrieve annotation information about a dead family, you’ll get a simple XML document that only includes information on the replacement (if any) for the family:

```
curl -LH 'Expect:' -F output=xml '/family/PF06700'
```

```xml
<?xml version="1.0" encoding="UTF-8"?><dead_pfam xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="https://pfam.xfam.org/"
         xsi:schemaLocation="https://pfam.xfam.org/static/documents/schemas/pfam-
         family.xsd" release="24.0" release_date="2009-10-07">
  <entry accession="PF06700"
           id="2oxo_fer_oxidoB">
    <forward_to>PF02775</forward_to>
    <comment>Merged into TPP binding domain</comment>
  </entry>
</dead_pfam>
```

You can see the XML schema for this XML document here.

**Pfam-A family list**

You can retrieve a list of all Pfam-A families in the latest Pfam release, either as an XML document or as a tab-delimited text file. Both formats contain the Pfam-A accession, Pfam-A identifier and description:

```
families?output=xml /families?output=text
```

You can also view the list in a web browser by removing the output=xml parameter from the URL.

```
curl -LH 'Expect:' -F output=xml '/families'
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- all Pfam-A families, generated: 16:12:54 26-Oct-2009 -->
<pfam xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="https://pfam.xfam.org/"
      xsi:schemaLocation="https://pfam.xfam.org/
      https://pfam.xfam.org/static/documents/schemas/pfam_
      families.xsd"
      release="24.0" release_date="2009-10-07">
  <entry accession="PF06700"
           id="2oxo_fer_oxidoB">
    <forward_to>PF02775</forward_to>
    <comment>Merged into TPP binding domain</comment>
  </entry>
</pfam>
```
release="24.0"
release_date="2009-10-07">
<entry entry_type="Pfam-A" accession="PF00001" id="7tm_1">
<description>
<![CDATA[
7 transmembrane receptor (rhodopsin family)
]]>
</description>
</entry>
...

You can see the XML schema for this XML document [here](#).

**Protein sequence data**

You can retrieve a sub-set of the data in a protein page as an XML document using any of the following styles of URL:

- `/protein?id=CANX_CHICK&output=xml`
- `/protein?output=xml&acc=P00789`
- `/protein?entry=P00789&output=xml`
- `/protein/P00789?output=xml`

As for Pfam-A families, arguments are all case-insensitive and the entry parameter accepts either ID or accession.

```bash
curl -LH 'Expect:' -F output=xml '/protein/P00789'
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- information on UniProt entry P00789 (CANX_CHICK), generated: 16:28:26 26-Oct-2009 -->
<pfam xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://pfam.xfam.org/"
xsi:schemaLocation="https://pfam.xfam.org/static/documents/schemas/protein.xsd"
release="24.0"
release_date="2009-10-07">
<entry entry_type="sequence" db="uniprot" db_release="57.6" accession="P00789" id="CANX_CHICK">
<description>
<![CDATA[
Calpain-1 catalytic subunit EC=3.4.22.52
]]>
</description>
<taxonomy tax_id="9031" species_name="Gallus gallus (Chicken)">
Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Archosauria; Dinosauria; Theropoda; Coelurosauria; Aves; Neognathae; Galliformes; Phasianidae; Phasianinae; Gallus.
</taxonomy>
<sequence length="705" md5="934014b14eb71623fa5898c7f81862a" crc64="ABCDDCS56298E48AA" version="2">
MMPFGGGARLRQDRVLRAEGVGEHNNAVKYNQDYALQECIESGTLFRDPQFPAAPGTLGF KelgySPK TRGVGWKPSELVDIPQFYVGGATRDTICQGALGCWLLAAIGSLTLNEELLHRVPHGQSPEQDYAGIFHFQIWQ FGEWDVVPDDLPTKGELLFVFHSAECTEFWSALLERLAKNLGCMESLSSGSTEGFEDFTGGVAEMYDLKR APRMNMGHIIRKALERGSLLGCSDIDTSAFDMEDAVTFKKLKVGHAYSVTAFKDYNRQQEQLIRINRPWQLVFW TGAWSDGSEWNDPSDREELQLKMEGREFWMSFRDFMREFSRLEICNLTPDALTKELESKRWHTQVFEVTWRR
```
Sequence searches

The Pfam website includes a form that allows users to upload a protein sequence and see a list of the Pfam domains that are found on their search sequence. We’ve now implemented a RESTful interface to this search tool, making it possible to run single-sequence Pfam searches programmatically.

Running a search is a two step process:

1. submit the search sequence and specify search parameters
2. retrieve search results in XML format

The reason for separating the operation into two steps rather than performing a search in a single operation is that the time taken to perform a sequence search will vary according to the length of the sequence searched. Most web clients, browsers or scripts, will simply time-out if a response is not received within a short time period, usually less than a minute. By submitting a search, waiting and then retrieving results as a separate operation, we avoid the risk of a client reaching a time-out before the results are returned.

The following example uses simple command-line tools to submit the search and retrieve results, but the whole process is easily transferred to a single script or program.

Save your sequence to file

It is usually most convenient to save your sequence into a plain text file, something like this:
The sequence should contain only valid sequence characters, i.e. letters, excluding “J” and “O”. You can break the sequence across multiple lines to make it easier to handle.

Submit the search

```
curl -LH 'Expect:' -F seq='<test.seq' -F output=xml 'https://pfam.xfam.org/search/sequence'
```

You can see the XML schema for this XML document here.

When using curl the value of the parameter “seq” needs to be quoted so that its value is taken correctly from the file “test.seq”. The second parameter can also be added directly to the URL, as a regular CGI-style parameter, if you prefer.

The search service accepts the following parameters (you can see a more complete description of these settings here):

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted values</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>evalue</td>
<td>use this E-value cut-off</td>
<td>valid float &lt; 10.0</td>
<td>none</td>
<td>the default is to have no E-value and to use the gathering threshold. See note below. If an E-value is given, it will be used, regardless of the value of ga</td>
</tr>
<tr>
<td>ga</td>
<td>use gathering threshold</td>
<td>0</td>
<td>1</td>
<td>the default is to have no E-value and to use the gathering threshold. See note below. If an E-value is given, it will be used, regardless of the value of ga</td>
</tr>
<tr>
<td>seq</td>
<td>protein sequence</td>
<td>valid sequence characters</td>
<td>none</td>
<td>required</td>
</tr>
</tbody>
</table>

Note: this documentation previously suggested that searches submitted through the RESTful interface used an E-value cut-off of 1.0 by default. This is incorrect. RESTful searches use the gathering threshold and not an E-value of 1.0. This is the opposite of the behaviour of the searches run through the web interface. We apologise for the inconsistency.
Wait for the search to complete

Although you can check for results immediately, if you poll before your job has completed, you won’t receive an XML document. Instead, the HTTP response to your request will have its status set appropriately and the body of the response will contain only string giving the status. You should ideally check the HTTP status of the response, rather than relying on the body of the response.

These are the possible status codes for the response:

<table>
<thead>
<tr>
<th>HTTP status code</th>
<th>Status description</th>
<th>Response body</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>202</td>
<td>Accepted</td>
<td>PEND / RUN</td>
<td>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.</td>
</tr>
<tr>
<td>502</td>
<td>Bad gateway</td>
<td>FAIL</td>
<td>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.</td>
</tr>
<tr>
<td>503</td>
<td>Service unavailable</td>
<td>HOLD</td>
<td>Your job was accepted but is on hold. This status will not be assigned by the search system, but by an administrator. There is probably a problem with the job and you should contact the help desk for assistance with it.</td>
</tr>
<tr>
<td>410</td>
<td>Gone</td>
<td>DEL</td>
<td>Your job was deleted from the search system. This status will not be assigned by the search system, but by an administrator. There was probably a problem with the job and you should contact the help desk for assistance with it.</td>
</tr>
<tr>
<td>500</td>
<td>Internal server error</td>
<td>Error message</td>
<td>There was some problem with running your job, but it does not fall into any of the other categories. The body of the response will contain an error message from the server. Contact the help desk for assistance with the problem.</td>
</tr>
</tbody>
</table>

When writing a script to submit searches and retrieve results, please add a short delay between the submission and the first attempt to retrieve results. Most search jobs are returned within four to five seconds of submission, depending greatly on the length of the sequence to be searched.

Retrieve results

The XML that was returned from the first query includes one or more URLs from which you can now retrieve results, given in the <result_url>. You can now poll these URLs to retrieve XML documents with the search hits.

curl -LH 'Expect:' 'https://pfam.xfam.org/search/sequence/resultset/adabec68-703f-48c4-bec7-07f1ab965fbb?output=xml'

<?xml version="1.0" encoding="UTF-8"?>
<pfam xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="https://pfam.xfam.org/">
  <xsi:schemaLocation="https://pfam.xfam.org/
                       https://pfam.xfam.org/static/documents/schemas/results.xsd"
      release="24.0"
</pfam>
release_date="2009-10-07">
<results job_id="adabec68-703f-48c4-bec7-07f1ab965fbb">
<matches>
<protein length="669">
<database id="pfam" release="24.0" release_date="2009-10-07">
<match accession="PF08310.4" id="LGFP" type="Pfam-A" class="Repeat">
<location start="422" end="475" ali_start="422" ali_end="473"
hmm_start="1" hmm_end="52" evalue="5.9e-10" bitscore="38.3" evidence="hmmer v3.0b2" significant="1" />
<location start="476" end="529" ali_start="476" ali_end="528"
hmm_start="1" hmm_end="53" evalue="4.8e-23" bitscore="80.2" evidence="hmmer v3.0b2" significant="1" />
<location start="530" end="580" ali_start="530" ali_end="578"
hmm_start="1" hmm_end="51" evalue="3e-06" bitscore="26.4" evidence="hmmer v3.0b2" significant="1" />
<location start="581" end="633" ali_start="581" ali_end="633"
hmm_start="1" hmm_end="54" evalue="1e-19" bitscore="69.5" evidence="hmmer v3.0b2" significant="1" />
</match>
<match accession="PF00756.13" id="Esterase" type="Pfam-A" class="Family">
<location start="122" end="392" ali_start="122" ali_end="390"
hmm_start="1" hmm_end="250" evalue="3.1e-62" bitscore="209.9" evidence="hmmer v3.0b2" significant="1" />
</match>
</database>
</protein>
</matches>
</results>
</pfam>

You can see the XML schema for this XML document here.

Since the search is performed by the same server as searches in the Pfam website, you can view your results in a web page by modifying the URL slightly:

/search/sequence/resultset/adabec68-703f-48c4-bec7-07f1ab965fbb

Note that old search results are generally cleared out after some time, so if you wait too long before trying to view your hits in the website, you may find that they are already gone.

Retrieve domain graphics description

When you run a sequence search via the browser, the results page includes a Pfam domain graphic, showing the locations of any matching Pfam families on your search sequence. When running a search via the RESTful interface, you can’t retrieve the domain graphic directly, since it’s generated using a javascript class in the browser. However, you can retrieve the JSON string that describes the graphic:

/search/sequence/graphic/adabec68-703f-48c4-bec7-07f1ab965fbb

Check the A Guide to the Pfam Domain Graphics for details on how you can use the JSON string locally.
1.12 Pfam MySQL database

The Pfam MySQL database contains all of the data accessible via the website. The database currently consists of 65 tables. Below is some basic documentation on the schema layout and how smaller numbers of tables can be put together to enable access to a subset of the data. At the time of writing, between releases 28.0 and 29.0, the fields within the tables and the results of queries are correct. The data within the tables will change with each release. Although we do not anticipate any major changes to the database, we reserve the right to make changes with or without warning; we will endeavour to update this document if such changes are made.

A red diamond in the images below indicates a foreign key. In some images there are tables which appear not to be linked to any other table in the image. This is due to a foreign key being populated late in production of the database. The ‘floating’ table can still be joined and example queries of how to do so are given under each image.

1.12.1 Version table

The version table contains information that relates to a particular Pfam release. It contains the version number of the Pfam database, the version numbers of the Swiss-Prot and TrEMBL databases that were used to build Pfam, and some statistics about the number of families and coverage. This table is stand-alone and does not link to any of the other tables.

Example query: Give me all of the version information for the Pfam database

```
SELECT * FROM version
```

Example output:

```
pfam_release: 28.0
pfam_release_date: 2015-05-15
swiss_prot_version: 2014_07
trembl_version: 2014_07
hmmer_version: 3.1b1
pfamA_coverage: 81.5
pfamA_residue_coverage: 61.4
```
1.12.2 Domain Information

Two of the central tables in the Pfam database are `pfamseq`, which contains UniProtKB reference proteomes and `pfamA`, which contains information about the Pfam-A families. Most of the other tables in the database link to one or both of these tables, either directly or indirectly. Note that prior to Pfam 29.0, the `pfamseq` table contained the whole of UniProtKB. From Pfam 29.0, this table contains only the reference proteome portion of UniProtKB. The full alignments in Pfam are based on the sequences in the `pfamseq` table.

The table `pfamA_reg_seed` contains the Pfam regions that are present in a seed alignment. All sequences in `pfamA_reg_seed` are in the `pfamseq` table or the `uniprot` table (the `uniprot` table contains all the sequences in UniProtKB). The `pfamA_reg_full_significant` table contains all of the sequence regions from the `pfamseq` table that match the HMM and score above the curated threshold, i.e. are significant matches, for each family. There is also a table named `pfamA_reg_full_insignificant` which contains, as the name suggests, all the insignificant matches for each family. Insignificant matches are those which match the HMM with an E-value less than 1000, but score below the curated bit score threshold for each family.

In addition to providing matches to the sequences in the `pfamseq` table, we also provide the significant matches for the sequences in the `uniprot` table. These can be found in the table `uniprot_reg_full`.

The tables `pfamA_reg_full_significant` and `uniprot_reg_full` contain a column called ‘in_full’. The matches that are present in the full alignment for a Pfam family have this column set to 1, while those that are not present in the full alignment have the ‘in_full’ column set to 0. A significant match will only be excluded from the full alignment (in_full = 0) if it matches a family that belongs to a clan, and the match overlaps with another more significant (lower E-value) match to a family within the clan.

For each sequence match we store two sets of coordinates, the envelope coordinates and the alignment coordinates. The envelope co-ordinates delineate the region on the sequence where the match has been probabilistically determined to lie, whereas the alignment coordinates delineate where HMMER is confident that the alignment of the sequence to the profile HMM is correct. Our full alignments contain the envelope coordinates. In the database, envelope start and end positions are stored in the seq_start and seq_end fields columns, and the alignment coordinates are stored in the ali_start and ali_end fields.

The Pfam database has historically been built on the UniProtKB database. However, as of release 22.0 we also provide Pfam domain data for the NCBI sequence database (GenPept) and a set of metagenomics sequences. As of release 28.0, we no longer store Pfam information at the sequence level for the NCBI and metagenomics data sets in the MySQL database, but we still provide the family alignments for them in the alignment_and_tree table. The Pfam website can still be queried using NCBI and metagenomics accessions.

Example query: Give me all of the domains for sequence ‘VAV_HUMAN’

```
SELECT pfamA.pfamA_acc, pfamA_id, seq_start, seq_end
FROM pfamseq, pfamA, pfamA_reg_full_significant
WHERE pfamseq_id = 'VAV_HUMAN'
AND in_full = 1
AND pfamseq.pfamseq_acc = pfamA_reg_full_significant.pfamseq_acc
AND pfamA_reg_full_significant.pfamA_acc = pfamA.pfamA_acc
ORDER BY seq_start
```

Example output:

```
<table>
<thead>
<tr>
<th>pfamA.acc</th>
<th>pfamA_id</th>
<th>seq_start</th>
<th>seq_end</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF00307</td>
<td>CH</td>
<td>1</td>
<td>115</td>
</tr>
<tr>
<td>PF00621</td>
<td>RhoGEF</td>
<td>198</td>
<td>372</td>
</tr>
<tr>
<td>PF00169</td>
<td>PH</td>
<td>403</td>
<td>504</td>
</tr>
<tr>
<td>PF00130</td>
<td>C1_1</td>
<td>516</td>
<td>568</td>
</tr>
<tr>
<td>PF00018</td>
<td>SH3_1</td>
<td>615</td>
<td>652</td>
</tr>
<tr>
<td>PF00017</td>
<td>SH2</td>
<td>671</td>
<td>745</td>
</tr>
<tr>
<td>PF00018</td>
<td>SH3_1</td>
<td>788</td>
<td>834</td>
</tr>
</tbody>
</table>
```
To report all of the overlapping domains within any clans, leave out the `in_full = 1` clause. More information on clans can be found later in this document.

Example query: Give me all the sequences in the **full** alignment for the family ‘B12D’

```sql
SELECT pfamseq_id, pfamseq.pfamseq_acc, seq_start, seq_end, pfamA_id
FROM pfamA, pfamseq, pfamA_reg_full_significant
WHERE pfamA_id = 'B12D'
AND in_full = 1
AND pfamA.pfamA_acc = pfamA_reg_full_significant.pfamA_acc
AND pfamA_reg_full_significant.pfamseq_acc = pfamseq.pfamseq_acc
```

Example output:

<table>
<thead>
<tr>
<th>pfamseq_id</th>
<th>pfamseq_acc</th>
<th>seq_start</th>
<th>seq_end</th>
<th>pfamA_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>U3I8N7_ANAPL</td>
<td>U3I8N7</td>
<td>16</td>
<td>81</td>
<td>B12D</td>
</tr>
<tr>
<td>M0XBC6_HORVD</td>
<td>M0XBC6</td>
<td>1</td>
<td>49</td>
<td>B12D</td>
</tr>
<tr>
<td>D92FG6_HORSE</td>
<td>D92FG6</td>
<td>9</td>
<td>50</td>
<td>B12D</td>
</tr>
<tr>
<td>G3VJG8_SARHA</td>
<td>G3VJG8</td>
<td>16</td>
<td>84</td>
<td>B12D</td>
</tr>
<tr>
<td>B6DDV9_ANODA</td>
<td>B6DDV9</td>
<td>11</td>
<td>78</td>
<td>B12D</td>
</tr>
<tr>
<td>V9LJR2_CALMI</td>
<td>V9LJR2</td>
<td>11</td>
<td>79</td>
<td>B12D</td>
</tr>
<tr>
<td>G7EA12_MIXOS</td>
<td>G7EA12</td>
<td>7</td>
<td>73</td>
<td>B12D</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example query: Give me all the sequences in the **seed** alignment for the family ‘B12D’

```sql
SELECT pfamseq_id, pfamseq.pfamseq_acc, seq_start, seq_end, pfamA_id
FROM pfamA, pfamseq, pfamA_reg_seed
WHERE pfamA_id = 'B12D'
AND pfamA.pfamA_acc = pfamA_reg_seed.pfamA_acc
AND pfamA_reg_seed.pfamseq_acc = pfamseq.pfamseq_acc
```

Example output:

<table>
<thead>
<tr>
<th>pfamseq_id</th>
<th>pfamseq_acc</th>
<th>seq_start</th>
<th>seq_end</th>
<th>pfamA_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>V8NRR6_OPHHA</td>
<td>V8NRR6</td>
<td>44</td>
<td>110</td>
<td>B12D</td>
</tr>
<tr>
<td>M1C9M8_SOLUTU</td>
<td>M1C9M8</td>
<td>7</td>
<td>75</td>
<td>B12D</td>
</tr>
<tr>
<td>E2B831_HARSA</td>
<td>E2B831</td>
<td>11</td>
<td>78</td>
<td>B12D</td>
</tr>
<tr>
<td>G3VW47_SARHA</td>
<td>G3VW47</td>
<td>9</td>
<td>75</td>
<td>B12D</td>
</tr>
<tr>
<td>M0U3S7_MUSAM</td>
<td>M0U3S7</td>
<td>5</td>
<td>73</td>
<td>B12D</td>
</tr>
<tr>
<td>I1H296_BRADI</td>
<td>I1H296</td>
<td>6</td>
<td>74</td>
<td>B12D</td>
</tr>
<tr>
<td>K7DIW7_MONDO</td>
<td>K7DIW7</td>
<td>9</td>
<td>75</td>
<td>B12D</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.12.3 Pfamseq - other tables

This section contains a few tables that link to the **pfamseq** table, but don’t fit nicely into any of the sections described above.

The **pfam_annseq** table contains binary Perl data structures which are used internally to generate the Pfam domain graphics. This table is not intended for use by Pfam users, as it is very dependent on Perl module versions.

The **evidence** table contains the UniProtKB evidence code key that is used in the evidence field in the **pfamseq** and uniprot tables.

UniProtKB sequences have secondary accessions if they have been merged or split. Secondary accession numbers are stored in the table called **secondary_pfamseq_acc**.
Example query: Give me the secondary accession(s) for the sequence ‘P15455’

```
SELECT secondary_acc
FROM pfamseq, secondary_pfamseq_acc
WHERE pfamseq.pfamseq_acc = secondary_pfamseq_acc.pfamseq_acc
AND pfamseq.pfamseq_acc= 'P15455'
```

Example output:

```
secondary_acc
Q3E711
Q56Z11
Q9FFH7
```

### 1.12.4 Other regions

These tables contain sequence specific information about the reference proteome sequences. The `other_regions` table contains coiled coil, low complexity, signal peptide, transmembrane and disordered regions data. The `pfamseq_markup` table contains active site information which is taken from the UniProtKB feature table. Additional active site residues are predicted by Pfam based on conserved residues in a Pfam alignment. The `pfamseq_disulphide` tables contains disulphide bond information from the UniProtKB feature table.
Example query: Give me all of the transmembrane, signal-peptide, coiled-coils, low-complexity and disorder information for the sequence ‘VAV_HUMAN’

```sql
SELECT type_id, source_id, seq_start, seq_end
FROM other_reg, pfamseq
WHERE pfamseq.pfamseq_id = 'VAV_HUMAN'
AND other_reg.pfamseq_acc = pfamseq.pfamseq_acc
```

Example output:

<table>
<thead>
<tr>
<th>type_id</th>
<th>source_id</th>
<th>seq_start</th>
<th>seq_end</th>
</tr>
</thead>
<tbody>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>128</td>
<td>129</td>
</tr>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>140</td>
<td>141</td>
</tr>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>160</td>
<td>161</td>
</tr>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>173</td>
<td>177</td>
</tr>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>179</td>
<td>180</td>
</tr>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>568</td>
<td>588</td>
</tr>
<tr>
<td>disorder</td>
<td>IUPred</td>
<td>635</td>
<td>636</td>
</tr>
<tr>
<td>low_complexity</td>
<td>segmasker</td>
<td>41</td>
<td>50</td>
</tr>
<tr>
<td>low_complexity</td>
<td>segmasker</td>
<td>355</td>
<td>366</td>
</tr>
</tbody>
</table>

Example query: Give me all of the active site information for sequence ‘F7PG13’

```sql
SELECT pfamseq.pfamseq_acc, pfamseq_id, residue, label
FROM pfamseq, pfamseq_markup, markup_key
WHERE pfamseq.pfamseq_acc = pfamseq_markup.pfamseq_acc
AND pfamseq_markup.auto_markup = markup_key.auto_markup
AND pfamseq.pfamseq_acc = 'F7PG13'
```

Example output:

<table>
<thead>
<tr>
<th>pfamseq_acc</th>
<th>pfamseq_id</th>
<th>residue</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7PG13</td>
<td>F7PG13_9EURY</td>
<td>92</td>
<td>Active site</td>
</tr>
<tr>
<td>F7PG13</td>
<td>F7PG13_9EURY</td>
<td>248</td>
<td>Pfam predicted active site</td>
</tr>
<tr>
<td>F7PG13</td>
<td>F7PG13_9EURY</td>
<td>283</td>
<td>Metal ion binding</td>
</tr>
<tr>
<td>F7PG13</td>
<td>F7PG13_9EURY</td>
<td>280</td>
<td>Metal ion binding</td>
</tr>
<tr>
<td>F7PG13</td>
<td>F7PG13_9EURY</td>
<td>278</td>
<td>Metal ion binding</td>
</tr>
</tbody>
</table>

Example query: Give me all the residues involved in disulphide bonds in the sequence ‘Q43495’

```sql
SELECT pfamseq.pfamseq_acc, pfamseq_id, bond_start, bond_end
FROM pfamseq, pfamseq_disulphide
WHERE pfamseq_disulphide.pfamseq_acc = pfamseq.pfamseq_acc
AND pfamseq.pfamseq_acc = 'Q43495'
```

Example output:

<table>
<thead>
<tr>
<th>pfamseq_acc</th>
<th>pfamseq_id</th>
<th>bond_start</th>
<th>bond_end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q43495</td>
<td>108_SOLLC</td>
<td>67</td>
<td>92</td>
</tr>
<tr>
<td>Q43495</td>
<td>108_SOLLC</td>
<td>51</td>
<td>66</td>
</tr>
<tr>
<td>Q43495</td>
<td>108_SOLLC</td>
<td>41</td>
<td>77</td>
</tr>
<tr>
<td>Q43495</td>
<td>108_SOLLC</td>
<td>79</td>
<td>99</td>
</tr>
</tbody>
</table>
1.12.5 Architecture information for a family

In Pfam, an architecture is the combination of domains that are present on a protein. The **architecture** table can be used to find out which combination of domains are found on particular sets of proteins, or to find out which proteins share the same domains architecture.

Example query: Give me all of the architectures and sequences for the family ‘Dehyd-heme_bind’

```
SELECT architecture_acc, pfamseq_id, pfamseq_acc
FROM architecture, pfamseq
WHERE architecture like '%Dehyd-heme_bind'
AND pfamseq.auto_architecture = architecture.auto_architecture
```

Example output:

```
1.12. Pfam MySQL database
```
Example query: Give me all the sequences which have the architecture ‘PF09098 PF09099 PF09100’

```
SELECT pfamseq_acc, pfamseq_id
FROM architecture, pfamseq
WHERE architecture.auto_architecture = pfamseq.auto_architecture
AND architecture_acc = "PF09098 PF14930 PF09099 PF09100"
```

Example output:

<table>
<thead>
<tr>
<th>pfamseq_acc</th>
<th>pfamseq_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0A013VTV5</td>
<td>A0A013VTV5_9SPHN</td>
</tr>
<tr>
<td>A0A023DP27</td>
<td>A0A023DP27_GEOTM</td>
</tr>
<tr>
<td>A0A024E848</td>
<td>A0A024E848_9PSED</td>
</tr>
<tr>
<td>A0A024HHX9</td>
<td>A0A024HHX9_PSESB</td>
</tr>
<tr>
<td>A0A059KVBO</td>
<td>A0A059KVBO_9PSED</td>
</tr>
<tr>
<td>A0NPA8</td>
<td>A0NPA8_9RHOB</td>
</tr>
<tr>
<td>A1B2Q6</td>
<td>A1B2Q6_PARDP</td>
</tr>
<tr>
<td>A1N18Q</td>
<td>A1N18Q_TEYCO</td>
</tr>
</tbody>
</table>

1.12.6 Annotation information for a family

In addition to the Pfam annotation, we also store InterPro annotation and their associated GO terms for each family. Links to other databases, e.g. SCOP) are also stored where appropriate. The pfamA table contains the GA, TC and NC cut-offs for each family, and additional information surrounding the Pfam-A family, including the number of sequences in the seed and full alignment. The pfamA_interactions table contains, where data are available, pairs of interacting Pfam domains. The data in this table are taken from the iPfam resource (no longer maintained), which describes physical interactions between Pfam domains that have a representative structure in the PDB.

Example query: Give me the Pfam annotation for the family ‘CBS’

```
SELECT comment
FROM pfamA WHERE pfamA_id = 'CBS'
```

Example output:

comment: CBS domains are small intracellular modules that pair together to form a stable globular domain [2]. This family represents a single CBS domain. Pairs of these domains have been termed a Bateman domain [6]. CBS domains have been shown to bind ligands with an adenosyl group such as AMP, ATP and S-AdoMet [5]. CBS domains are found attached to a wide range of other protein domains suggesting that CBS domains may play a regulatory role making proteins sensitive to adenosyl carrying ligands. The region containing the CBS domains in Cystathionine-beta synthase is involved in regulation by S-AdoMet [4]. CBS domain pairs from AMPK bind AMP or ATP [5]. The CBS domains from IMPDH and the chloride channel CLC2 bind ATP [5].
1.12. Pfam MySQL database
Example query: Give me the interpro annotation for the family ‘CBS’

```sql
SELECT interpro_id, abstract
FROM interpro, pfamA
WHERE pfamA.pfamA_acc = interpro.pfamA_acc
AND pfamA_id = 'CBS'
```

Example output:

```
<table>
<thead>
<tr>
<th>go_id</th>
<th>term</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0020037</td>
<td>heme binding</td>
</tr>
<tr>
<td>GO:0016705</td>
<td>oxidoreductase activity, acting on paired donors, with incorporation or reduction of molecular oxygen function</td>
</tr>
<tr>
<td>GO:0005506</td>
<td>iron ion binding</td>
</tr>
<tr>
<td>GO:0055114</td>
<td>oxidation-reduction process</td>
</tr>
</tbody>
</table>
```

Example query: Give me all of the literature references for the family ‘CBS’

```sql
SELECT pfamA_literature_reference.comment, order_added, pmid, title, literature_reference.author, journal
FROM pfamA, pfamA_literature_reference, literature_reference
WHERE pfamA_id = 'CBS'
AND pfamA.pfamA_acc = pfamA_literature_reference.pfamA_acc
AND pfamA_literature_reference.auto_lit = literature_reference.auto_lit
```

Example output:

```
comment: Discovery and naming of the CBS domain.
order_added: 1
pmid: 9020585
title: The structure of a domain common to archaebacteria and the homocystinuria disease protein.
author: Bateman A;
```

Example query: Give me all of the database references for the family ‘A2M’

```sql
SELECT db_id, pfamA_database_links.comment, db_link, other_params
FROM pfamA, pfamA_database_links
WHERE pfamA_id = 'A2M'
AND pfamA.pfamA_acc = pfamA_database_links.pfamA_acc
```

Example output:

```
db_id  comment  db_link  other_params
PROSITE PDOC00440
SCOP    1c3d     fa
HOMSTRAD A2M_A
HOMSTRAD A2M_B
```

Note: The other_params column contains ‘fa;’ where the Pfam family corresponds to a SCOP family, and ‘sf;’ where the Pfam family corresponds to a SCOP superfamily.

Example query: Give me the interacting domains for the domain ‘EGF’
**1.12. Pfam MySQL database**

A Pfam clan is a set of related Pfam-A families. The information we use to determine which families belong to the same clan includes related structure, related function, matching of the same sequence to HMMs from different families, and profile-profile comparisons. **Note** that not all Pfam-A families belong to a clan and that a Pfam-A family cannot belong to more than one clan.

Example query: Give me the id and accession of the clan to which Pfam family ‘EGF’ belongs

```sql
SELECT clan_id, clan.clan_acc
FROM clan, clan_membership, pfamA
WHERE clan.clan_acc = clan_membership.clan_acc
AND clan_membership.pfamA_acc = pfamA.pfamA_acc
AND pfamA.pfamA_id = 'EGF'
```

Example output:

<table>
<thead>
<tr>
<th>clan_id</th>
<th>clan_acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGF</td>
<td>CL0001</td>
</tr>
</tbody>
</table>

Example query: Give me all of the Pfam-A families that belong to clan ‘CL0001’

```sql
SELECT pfamA.pfamA_acc, pfamA_id
FROM clan, clan_membership, pfamA
WHERE clan.clan_acc = clan_membership.clan_acc
AND clan_membership.pfamA_acc = pfamA.pfamA_acc
AND clan.clan_acc = 'CL0001'
```

Example output:

<table>
<thead>
<tr>
<th>pfamA_acc</th>
<th>pfamA_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF01414</td>
<td>DSL</td>
</tr>
<tr>
<td>PF04863</td>
<td>EGF_alliinase</td>
</tr>
<tr>
<td>PF00053</td>
<td>Laminin_EGF</td>
</tr>
<tr>
<td>PF07645</td>
<td>EGF_CA</td>
</tr>
<tr>
<td>PF00008</td>
<td>EGF</td>
</tr>
<tr>
<td>PF07974</td>
<td>EGF_2</td>
</tr>
<tr>
<td>PF09064</td>
<td>Tme5_EGF_like</td>
</tr>
<tr>
<td>PF09289</td>
<td>FOLN</td>
</tr>
<tr>
<td>PF12661</td>
<td>hEGF</td>
</tr>
</tbody>
</table>

(continues on next page)
Example query: Give me the clan description and comment for clan ‘CL0001’

```
SELECT clan.clan_acc, clan_id, clan_description, clan_comment
FROM clan
WHERE clan_acc = 'CL0001'
```

Example output:

```
clan_acc: CL0001
clan_id: EGF
clan_description: EGF superfamily
clan_comment: Members of this clan all belong to the EGF superfamily ...
```

Example query: Give me the literature references for clan ‘CL0001’

```
SELECT comment, order_added, pmid, title, author, journal
FROM clan, literature_reference, clan_lit_ref
WHERE clan.clan_acc = clan_lit_ref.clan_acc
AND clan_lit_ref.auto_lit = literature_reference.auto_lit
AND clan.clan_acc = 'CL0001'
```

Example output:

```
comment: NULL
order_added: 2
pmid: 11852228
  title: Domain structure and organisation in extracellular matrix proteins.
  author: Hohenester E, Engel J;

comment: NULL
order_added: 1
pmid: 3282918
  title: Structure and function of epidermal growth factor-like regions in...
  author: Appella E, Weber IT, Blasi F;
```

Example query: Give me the first 5 architectures for the clan ‘CL0001’

```
SELECT architecture, architecture_acc, type_example, no_seqs
FROM architecture, clan_architecture, clan
WHERE architecture.auto_architecture = clan_architecture.auto_architecture
AND clan_architecture.clan_acc = clan.clan_acc
AND clan.clan_acc = 'CL0001'
LIMIT 5
```

Example output:
architecture: DSL-hEGF-EGF
architecture_acc: PF01414 PF12661 PF00008
  type_example: E5STS8
  no_seqs: 2

architecture: DSL-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF-EGF_CA-EGF-EGF-
  → EGF-EGF
architecture_acc: PF01414 PF00008 PF00008 PF00008 PF00008 PF00008 PF00008 PF00008 PF00008 PF00008 PF00008
  → PF00008 PF00008 PF00008 PF00008 PF00008 PF00008 PF07645 PF00008 PF00008 PF00008
  → PF00008
  type_example: V4A2Z5
  no_seqs: 1

architecture: MNNL-DSL-EGF-hEGF-EGF-hEGF-EGF-EGF-EGF-EGF-EGF
architecture_acc: PF07657 PF01414 PF00008 PF12661 PF00008 PF12661 PF00008 PF00008
  → PF00008 PF00008 PF00008
  type_example: G6DTI0
  no_seqs: 1

architecture: MNNL-DSL-EGF-EGF-EGF
architecture_acc: PF07657 PF01414 PF00008 PF00008 PF00008
  type_example: Q8I497
  no_seqs: 2

architecture: MNNL-DSL-hEGF-EGF-EGF
architecture_acc: PF07657 PF01414 PF12661 PF00008 PF00008 PF00008
  type_example: Q95YG0
  no_seqs: 3

Example query: Give me the database links for clan ‘CL0001’

```sql
SELECT db_id, comment, db_link, other_params
FROM clan_database_links, clan
WHERE clan_database_links.clan_acc = clan.clan_acc
AND clan.clan_acc = 'CL0001'
```

Example output:

<table>
<thead>
<tr>
<th>db_id</th>
<th>comment</th>
<th>db_link</th>
<th>other_params</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCOP</td>
<td>NULL</td>
<td>57196</td>
<td></td>
</tr>
<tr>
<td>CATH</td>
<td>NULL</td>
<td>2.10.25.10</td>
<td></td>
</tr>
</tbody>
</table>

### 1.12.8 Dead families and clans

Sometimes we find that two or more Pfam-A families can be merged into a single family, which leads to the deletion of Pfam-A families. Likewise we might merge two clans together, which results in the deletion of a clan. The dead_family and dead_clan tables contain information about Pfam-A families and clans that have been deleted. These tables may be of use if you need to track what happened to the members of a particular family/clan that is no longer in Pfam.

Example query: Give me all of the information about ‘dead’ Pfam-A family ‘PF06700’

```sql
SELECT * FROM dead_family WHERE pfamA_acc = 'PF06700'
```
1.12.9 Nested domains

Some Pfam-A domains are disrupted by the insertion of another domain (or domains) within them. The domain that is inserted into another is known as a nested domain. The nested_locations table stores all the nested Pfam-A domains. It also stores the coordinates of the nested domain with respect to a sequence that is present in the seed alignment of the domain in which it nests.

Example query: Give me all of the nested domains and the domains in which they are nested

```
SELECT A.pfamA_id, B.pfamA_id AS nested_domain
FROM pfamA AS A, pfamA AS B, nested_domains
```
WHERE A.pfamA_acc = nested_domains.pfamA_acc
AND B.pfamA_acc = nested_domains.nests_pfamA_acc

Example output:

<table>
<thead>
<tr>
<th>pfamA_id</th>
<th>nested_domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>YL1</td>
<td>S1</td>
</tr>
<tr>
<td>7tm_1</td>
<td>HTH_Tnp_Tc3_2</td>
</tr>
<tr>
<td>7tm_1</td>
<td>DDE_Tnp_4</td>
</tr>
<tr>
<td>DUF2330</td>
<td>TonB_C</td>
</tr>
<tr>
<td>DUF2577</td>
<td>NLPC_P60</td>
</tr>
<tr>
<td>ATP-synt_ab</td>
<td>Hom_end_hint</td>
</tr>
<tr>
<td>ATP-synt_ab</td>
<td>Hom_end</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Example query: Give me the nested data for the family IMPDH

```
SELECT pfamA_id, nested_pfamA_acc, pfamseq_acc, seq_version, seq_start, seq_end
FROM pfamA, nested_locations
WHERE pfamA.pfamA_acc = nested_locations.pfamA_acc
AND pfamA_id = "IMPDH"
```

Example output:

```
<table>
<thead>
<tr>
<th>pfamA_id</th>
<th>nested_pfamA_acc</th>
<th>pfamseq_acc</th>
<th>seq_version</th>
<th>seq_start</th>
<th>seq_end</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPDH</td>
<td>PF00571</td>
<td>Q21KQ8</td>
<td>1</td>
<td>155</td>
<td>271</td>
</tr>
</tbody>
</table>
```

### 1.12.10 Structural data

In order for the Protein DataBank (PDB) information to be useful to Pfam, we need to map between PDB residues and UniProtKB sequence residues, which is not a trivial task. We store the residue-by-residue mapping that is provided by the PDBe group in the `pdb_residue_data` table. Note that the `pdb_pfamA_reg` table is based on sequences in the `uniprot` table, and not the `pfamseq` table. This is to maximise the number of structures that can be mapped to each Pfam entry.

Example query: Give me the first 10 residue mappings for the structure ‘2abl’

```
SELECT pdb.pdb_id, pdb_res, pdb_seq_number, pfamseq_acc, pfamseq_res, pfamseq_seq_number
FROM pdb_residue_data, pdb
WHERE pdb.pdb_id = pdb_residue_data.pdb_id
AND pdb.pdb_id = '2abl'
LIMIT 10
```

Example output:

```
<table>
<thead>
<tr>
<th>pdb_id</th>
<th>pdb_res</th>
<th>pdb_seq_number</th>
<th>pfamseq_acc</th>
<th>pfamseq_res</th>
<th>pfamseq_seq_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2abl</td>
<td>GLY</td>
<td>76  P00519</td>
<td>G</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>PRO</td>
<td>77  P00519</td>
<td>P</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>SER</td>
<td>78  P00519</td>
<td>S</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>GLU</td>
<td>79  P00519</td>
<td>E</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>ASN</td>
<td>80  P00519</td>
<td>N</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>ASP</td>
<td>81  P00519</td>
<td>D</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>PRO</td>
<td>82  P00519</td>
<td>P</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>2abl</td>
<td>ASN</td>
<td>83  P00519</td>
<td>N</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>
```

(continues on next page)
66 Chapter 1. Contents:
Example query: Give me all the structures that map to the family ‘Globin’

```
SELECT pdb_pfamA_reg.pdb_id, chain, pdb_res_start, pdb_res_end, seq_start, seq_end
FROM pdb, pdb_pfamA_reg, pfamA
WHERE pfamA_id = 'Globin'
AND pfamA.pfamA_acc = pdb_pfamA_reg.pfamA_acc
AND pdb_pfamA_reg.pdb_id = pdb.pdb_id
```

Example output:

```
pdb_id  chain  pdb_res_start  pdb_res_end  seq_start  seq_end
101M     A        6             112            7            113
102M     A        6             112            7            113
103M     A        6             112            7            113
104M     A        6             112            7            113
105M     A        6             112            7            113
106M     A        6             112            7            113
107M     A        6             112            7            113
...```

1.12.11 Proteomes

As of Pfam 29.0, all sequences in the `pfamseq` table belong to a reference proteome, and therefore a complete proteome. Prior to Pfam 29.0 this was not the case. The `complete_proteomes` table contains statistics about the number of families and coverage. The tables in this section allow you to retrieve domain information about a particular species, or to retrieve all of the species which contain a particular Pfam domain.

Example query: Give me the Pfam summary for the human (Homo sapiens, ncbi taxid 9606) proteome

```
SELECT * FROM complete_proteomes WHERE ncbi_taxid=9606
```

Example output:

```
ncbi_taxid: 9606
species: Homo sapiens (Human)
grouping: Eukaryota
num_distinct_regions: 0
num_total_regions: 93089
num_proteins: 69682
sequence_coverage: 70
residue_coverage: 44
total_genome_proteins: 69682
total_aa_length: 23542137
total_aa_covered: 10345679
total_seqs_covered: 49101
```

Example query: Give me all the Pfam-A domains for the species ‘Arabidopsis thaliana’

```
SELECT r.pfamA_acc, pfamA_id, description, sum(number_domains)
FROM pfamA p, proteome_regions r
```

(continues on next page)
WHERE p.pfamA_acc=r.pfamA_acc
AND ncbi_taxid=3702
GROUP BY r.pfamA_acc

Example output:

<table>
<thead>
<tr>
<th>pfamA_acc</th>
<th>pfamA_id</th>
<th>description</th>
<th>sum(number_domains)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF00004</td>
<td>AAA</td>
<td>ATPase family associated with various cellular activities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(AAA)</td>
<td></td>
<td>143</td>
</tr>
<tr>
<td>PF00005</td>
<td>ABC_tran</td>
<td>ABC transporter</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>197</td>
</tr>
<tr>
<td>PF00006</td>
<td>ATP-synt_ab</td>
<td>ATP synthase alpha/beta family, nucleotide-binding domain</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>197</td>
</tr>
<tr>
<td>PF00009</td>
<td>GTP_EFTU</td>
<td>Elongation factor Tu GTP binding domain</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>PF00010</td>
<td>HLH</td>
<td>Helix-loop-helix DNA-binding domain</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>148</td>
</tr>
<tr>
<td>PF00011</td>
<td>HSP20</td>
<td>Hsp20/alpha crystallin family</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>PF00012</td>
<td>HSP70</td>
<td>Hsp70 protein</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The ncbi_code for the species ‘Arabidopsis thaliana’ is 3702. This information can be found in the ncbi_taxonomy table.

Example query: Give me all of the protein sequences for the species ‘Arabidopsis thaliana’

```sql
SELECT pfamseq_acc
FROM pfamseq
WHERE ncbi_taxid = '3702'
```

Example output:

<table>
<thead>
<tr>
<th>pfamseq_acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0A023T4L8</td>
</tr>
<tr>
<td>A0A023T774</td>
</tr>
<tr>
<td>A0A076</td>
</tr>
<tr>
<td>A0A077</td>
</tr>
<tr>
<td>A0A078</td>
</tr>
<tr>
<td>A0EJE2</td>
</tr>
<tr>
<td>A0EJE3</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Example query: Give me all of the protein sequences from the species ‘Arabidopsis thaliana’ that belong to Pfam-A domain ‘PF00106’

```sql
SELECT pfamseq.pfamseq_acc
FROM pfamseq, pfamA_reg_full_significant
WHERE ncbi_taxid = '3702'
AND pfamseq.pfamseq_acc = pfamA_reg_full_significant.pfamseq_acc
AND pfamA_acc = 'PF00106'
```

Example output:
1.12.12 Related families

SCOOP and HHsearch are two pieces of software that we use to help to determine which Pfam-A families are related. The scores from these programs have been a very useful aid in deciding which Pfam-A families should belong to the same clan. As a rough guide, a SCOOP score greater than 50 or a HHsearch E-value score of less than 0.01 is an indication that two families are closely related.
Example query: Give me all of the Pfam-A families that have a SCOOP score greater than 50 when compared with the family 'ABC1'

```
SELECT a.pfamA_id, b.pfamA_id, score
FROM pfamA AS a, pfamA AS b, pfamA2pfamA_scoop AS p
WHERE a.pfamA_acc = p.pfamA_acc_1
AND b.pfamA_acc = p.pfamA_acc_2
AND score > 50
AND a.pfamA_id = "ABC1"
```

Example output:

```
<table>
<thead>
<tr>
<th>pfamA_id</th>
<th>pfamA_id</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC1</td>
<td>Pkinase_Tyr</td>
<td>154.936</td>
</tr>
</tbody>
</table>
```

Example query: Give me all of the Pfam-A families that have a HHsearch E-value score of less than 0.01 when compared with the family ‘AAA’

```
SELECT a.pfamA_id, b.pfamA_id, evalue
FROM pfamA AS a, pfamA AS b, pfamA2pfamA_hhsearch AS p
WHERE a.pfamA_acc = p.pfamA_acc_1
AND b.pfamA_acc = p.pfamA_acc_2
AND evalue < 0.01
AND a.pfamA_id = "AAA"
```

Example output:

```
<table>
<thead>
<tr>
<th>pfamA_id</th>
<th>pfamA_id</th>
<th>evalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>RuvB_N</td>
<td>1.3E-25</td>
</tr>
<tr>
<td>AAA</td>
<td>AAA_2</td>
<td>1.1E-17</td>
</tr>
<tr>
<td>AAA</td>
<td>AAA_5</td>
<td>2.5E-17</td>
</tr>
<tr>
<td>AAA</td>
<td>TIP49</td>
<td>5.6E-13</td>
</tr>
<tr>
<td>AAA</td>
<td>Sigma54_activat</td>
<td>1.7E-12</td>
</tr>
<tr>
<td>AAA</td>
<td>Mg_chelatase</td>
<td>1.1E-11</td>
</tr>
<tr>
<td>AAA</td>
<td>AAA_3</td>
<td>1.9E-11</td>
</tr>
</tbody>
</table>
```

**1.12.13 Data Files - Alignments, trees and HMMs**

The seed, full, UniProtKB, NCBI, representative proteome and metaseq alignments are all stored as gzipped files in the database, as is the HMM for each family. Note that the NCBI and metaseq alignments may contain overlapping matches to Pfam-A families that belong to the same clan, however, the UniprotKB alignments (seed, full, uniprot and representative proteome sets) will not. This is because we have performed a clan filtering step on the UniProtKB data such that where there are overlapping Pfam-A matches within a clan, only the lowest E-value scoring match is included in the full alignment.
1.13 FTP Site

The Pfam FTP site is organised into the following structure:

```
|   +-- Tools/
    |   +-- papers/
    |   +-- current_release/
    |       +-- database_files/
    +-- releases/
        +-- Pfam23.0/
        |   +-- database_files/
        +-- Pfam22.0/
        |   +-- database_files/
        +-- ...
        |   +-- Pfam1.0/
```

The most important directory is probably the current_release directory. This contains the flat-files for the current release. Some of these files may be very large (of the order of several hundred megabytes). Please check the sizes on the FTP site before trying to download them over a slow connection. The files, most of which are compressed using gzip, are:

- **Pfam-A.dead.gz**  Listing of families that have been deleted from the database
- **Pfam-A.fasta.gz**  A 90% non-redundant set of fasta formatted sequence for each Pfam-A family. The sequences are only the regions hit by the model and not full length protein sequences.
- **Pfam-A.full.gz**  The full alignments of the curated families, searched against pfamseq/UniProtKB reference proteomes (prior to Pfam 29.0, this file contained matches against the whole of UniProtKB).
- **Pfam-A.full.uniprot.gz**  The full alignments of the curated families, searched against UniProtKB.
- **Pfam-A.full.metagenomics.gz**  The full alignments of the curated families, searched against Metagenomic proteins.
- **Pfam-A.full.ncbi.gz**  The full alignments of the curated families, searched against NCBI GenPept proteins.
- **Pfam-A.hmm.dat.gz**  A data file that contains information about each Pfam-A family
- **Pfam-A.hmm.gz**  The Pfam HMM library for Pfam-A families
- **Pfam-A.seed.gz**  The seed alignments of the curated families
- **Pfam-C.gz**  A file that contains the information about clans and the Pfam-A membership
- **active_site.dat.gz**  Tar-ball of data required for the predictions of active sites by Pfam scan.
- **database.tar**  A tar-ball of the database_files directory.
- **database_files**  Directory contains two files per table from the MySQL database. The .sql.gz file contains the table structure, the .txt.gz files contains the content of the table as a tab delimited file with field enclosed by a single quote (').
- **diff.gz**  Stores the change status of entries between this release and last.
The papers directory contains each NAR database issue article describing Pfam. For a detailed description of the latest changes to Pfam, please consult (and cite) these papers.

The releases directory contains all the flat files and database dumps (where appropriate) for all version of Pfam to-date. The files in more recent releases are the same as described for the current release, but in older releases the contents do change.

The Tools directory contains code for running pfam_scan.pl. The README file in this directory contains detailed information on how to install and run the script. Note that we have gone for a modular design for the script, enabling the functionally on the script to be easily incorporated into other Perl scripts. The ChangeLog file lists the versions and changes to the current version of pfam_scan.pl (and modules). There is also an archived version of pfam_scan.pl that works with HMMER2. This is no longer supported. There is also Perl code for predicting active sites found in the ActSitePred directory, the functionality of which has been rolled into the latest version of pfam_scan.pl

The top level directory contains the sitesearch folder, that contain a subset of information from Pfam in an XML file. This XML file, primarily for use by the EMBL-EBI Web Team, is indexed using lucene and used in the EMBL-EBI site search. This is updated at each release.

### 1.14 Privacy

This section outlines the ways in which the Pfam website handles information about users. This should not be read as a legal document, but as a description of how we handle information that could be considered sensitive. If you have any concerns about the way that information is used in the website, please contact us at pfam-help@ebi.ac.uk and we will be more than happy to discuss your concerns.

Although we make every possible effort to keep this site and the data that it manipulates safe and secure, we make no claim to be able to protect sensitive or privileged information. If you are at all concerned about sensitive information being released, please do not use the site and consider installing the Pfam database and/or this website locally.
1.14.1 Google analytics

We use Google analytics (GA), to track the usage of this website. GA uses a single-pixel “web bug” image, which is served from every page, a javascript script that collects information about each request, and cookies that maintain information about your usage of the site between visits. You can read more about how GA works on the GA website, which includes a detailed description of how traffic is tracked and analysed.

We use the information generated by GA purely for audit and accounting purposes, and to help us assess the usefulness and popularity of different features of the site. It does not provide the ability to track individual users’ usage of the site. However, GA does provide a high-level overview of the traffic that passes through the site, including such information as the approximate geographical location of users, how often and for how long they visited the site, etc.

We understand that this level of tracking may be worrying to some of our users. If you have any concerns about our use of Google analytics, please feel free to contact us.

1.14.2 Browsing

All web servers maintain fairly detailed logs of their activity. This includes keeping a record of every request that they serve, usually along with the IP address of the client that made the request. This is true of the web servers that host the Pfam websites.

Although our servers do collect information about your IP address during the normal process of serving the Pfam website, we do not use this information explicitly. The Pfam group uses server logs only to help with development and debugging of the site.

1.14.3 Searches

The sequence search feature of the site allows you to upload a protein or DNA sequence to be searched against our library of HMMs. The sequence that you upload is stored in a database and is retrieved by a set of scripts that actually perform the search. Although we do not have any information that could be used to link that sequence to you personally, you should be aware that the sequence itself is accessible to systems administrators and other users who maintain the Pfam site.

The batch search function allows you to submit larger searches, the results of which are emailed to you. Obviously, this requires you to provide identifiable information, namely an email address. However, beyond the routine backups of our databases, we do not store any information about email addresses and sequences in the longer term and we make no attempt to keep track of the searches that a particular user may be performing.

Information from other types of search, such as a keyword search, is held only in the web server logs but, as described above, no attempt is made to interpret these logs except as part of development or debugging of the site.

1.14.4 Cookies

We use the following cookies to maintain some information about you between your visits to the site. The information that is stored cannot be used to identify you personally and cannot be used to track your usage of the site.

<table>
<thead>
<tr>
<th>Cookie name</th>
<th>Purpose</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts</td>
<td>Timestamp when annotation submission form was loaded in browser</td>
<td>Required</td>
</tr>
<tr>
<td>hide_posts</td>
<td>Keep track of whether blog posts have been hidden in home page</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Table 3: Cookies used in the Pfam site
In addition to these Pfam-specific cookies, GA uses a series of cookies. You can read more about these in the GA documentation, or in EMBL-EBI’s cookie policy.

If you are at all concerned about the use of cookies in the Pfam site, you are free to block all cookies from this site and you should not experience any problems. You may see some unintended behaviour, such as being notified of all new features every time you visit the index page, but the core functionality of the site should be unaffected.

### 1.14.5 Third-party javascript libraries

This site makes heavy use of javascript and relies on javascript libraries that are developed by various groups and companies. In order to improve the performance of the Pfam website, we no longer serve these files ourselves, but rely on files that are hosted on third-party web-servers. In particular, we use various files that are provided by the AJAX libraries APIs, hosted by Google code, and components of the Yahoo! User Interface Library (YUI), hosted by Yahoo!

As these services are provided by commercial sites, it’s likely that their usage will be carefully monitored by the companies that provide them. Although the Pfam site does not pass any information about you to these third-party sites, the sites themselves may use cookies to track your usage of the files that they serve. If you are concerned about the privacy implications of this monitoring, you may want to block cookies from the third-party hosting sites.

### 1.15 Authorship

We greatly appreciate the contribution made to Pfam from our user community. To acknowledge these contributions, and allow them to be an integral part of researchers’ profiles, we have incorporated ORCID identifiers, displaying these in the ‘curation and model’ tab of each Pfam entry.

To claim Pfam entries against your ORCID, first go to the EMBL-EBI website and search by putting your ORCID into the search box and selecting ‘Protein Families’ from the drop down.

From the results page, navigate to Pfam and you should then see a link at the top of the results inviting you to ‘Claim to ORCID’. Click this and you will then be taken to the ORCID site to authenticate.
1.16 Team Members

1.16.1 The Pfam Consortium

Pfam is the product from an international consortium of researchers that has been borne out of its original development by Erik Sonnhammer, Sean Eddy and Richard Durbin. The current list of consortium members, their institutes and primary roles are listed below.

European Bioinformatics Institute (EMBL-EBI), UK

- Alex Bateman - Pfam team leader and head of Protein Sequence resources at EMBL-EBI
- Lowri Williams - Curator
- Sarah Chuguransky - Curator
- Jaina Mistry - Developer
- Matloob Qureshi - Web developer, Lead
- Gustavo Salazar - Web developer

Harvard University, USA

- Sean Eddy - Founding developer and author of HMMER software

Stockholm Bioinformatics Center, Sweden

- Erik Sonnhammer - Coordinator of Pfam-Sweden and founding developer

1.16.2 External contributors

Pfam includes families that have been built by external contributors:

NCBI, USA

- Lakshminarayan Iyer
- L. Aravind
- Zhang Dapeng
- Vivek Anantharaman
Sanford-Burnham Medical Research Institute, USA

- Adam Godzik

### 1.16.3 Previous contributors

- Gabriel Aldam
- Shimelis Assefa
- Matthew Bashton
- Ewan Birney
- Lorenzo Cerrutti
- Yuanyuan Chang
- Jody Clements
- Penny Coggill
- Lachlan Coin
- Robson De Souza
- Richard Durbin
- Ruth Eberhardt
- Sara El-Gebali
- Kyle Ellrott
- Matthew Fenech
- Kristoffer Forslund
- O. Luke Gavin
- Prasad Gunasekaran
- Sam Griffiths-Jones
- Kevin Howe
- Lukasz Jaroszewski
- Nicola Kerrison
- Marta Llagostera
- Aurélien Luciani
- Mhairi Marshall
- Nina Mian
- William Mifsud
- Simon Moxon
- Simon Potter
- Joanne Pollington
- Marco Punta
Pfam is a collaborative venture and we hope to be able to interact with as many people as possible, in order to provide a quality database. Please get in touch with any one of us for more information about Pfam. You can email Pfam at pfam-help@ebi.ac.uk.

1.17 Contact us

1.17.1 Helpdesk

We run a Pfam helpdesk, which handles annotation comments, data enquiries and general problems with the Pfam database and websites. We use a request tracking system to monitor emails to the helpdesk, so you should receive an automated response to your email, letting you know that the system has logged your mail and notified us of its arrival.

1.17.2 Mailing list

The Pfam mailing list is a low traffic list that has important announcements, such as releases or major changes. To join the mailing list send a mail to mailto:pfamlist-subscribe@ebi.ac.uk.

If you should want to unsubscribe from the list send a mail to mailto:pfamlist-unsubscribe@ebi.ac.uk.

1.17.3 Xfam blog

The Pfam group contributes to the Xfam blog. The blog is used to announce releases, new features and important changes to Pfam, as well as for posts discussing general issues surrounding the Pfam resource. You can see blog posts that are specific to Pfam here

1.17.4 RSS feed

You can keep in touch with the latest goings on by subscribing to the RSS feed from the Xfam blog.
1.17.5 Twitter

You can follow the @PfamDB team at EMBL-EBI.
Pfam is freely available under the Creative Commons Zero ("CC0") licence.
CITING PFAM

If you use Pfam in your work, please consider citing the *Pfam References.*
GET IN TOUCH

If you have any questions or feedback, email us at pfam-help@ebi.ac.uk.