SCOP [Home]

Release notes (*Last modified jmc:2005-07-29*).

Note for SCOP 1.69: A small number of PDB entries with 'official' release dates towards the end of September are not included because they were only released in the first weekly PDB update of October 2004. These entries are 1up9, 1upd, 1urm, 1w0c, 1w0h, 1w1w, 1w28, 1w2a, 1w2n, 1w2o, 1w2p, 1w2v, 1w30, 1w32, 1w3h, 1w7o, 1w7p, 1w8l, and 1w8m.

These notes mainly cover new features introduced in SCOP 1.55 and subsequent releases. They are organized by topics for easy reference, and include detailed explanations and examples. Although release 1.55 and following ones are superficially similar to previous releases, there are several innovations that may affect the way you use SCOP [4]. The SCOP history, originally designed to document fully all modifications of the classification, has been extended to provide an easy way to browse the subset of the classification affected by revisions occurred since 1.55 [4], including major rearrangements of the classification that have been carried out in release 1.63 and 1.65. Please cite [6] for the SCOP reclassification itself.

Please read carefully and don't hesitate to get in touch with us, should you have any further question. Your comments and suggestions are more than welcome.

- Overview
- SCOP identifiers
- Reclassified entries history
- Searching SCOP
- Linking to SCOP
- Linking from SCOP
- SCOP parseable files
- SCOP sequences and coordinate files
- References

- Overview

The SCOP (Structural Classification of Protein) database is a comprehensive ordering of all proteins of known structure according to their evolutionary, functional and structural relationships. The basic classification unit is the *protein domain*. Domains are hierarchically classified into *species*, *proteins*, *families*, *superfamilies*, *folds*, and *classes* whose meaning is described in the original articles [1,2,3].

As an example, see the classification of the bacteriophage 434 protein, PDB entry 2cro. The set of buttons at the top of each web page provides help and an easy way of navigating through the hierarchy. Protein domains can be visualized using either RASMOL or CHIME. Click on the L icon for a set of links to other sites containing useful information related to the protein domain. Use the entry at the bottom of each page to search the SCOP database.

Note that only the first seven classes are true classes. The remaining ones are place holders for PDB entries that is useful to keep together, but are not part of the SCOP classification. Also, one of the classes includes multi-domain proteins only, that is, proteins not yet split into domains and such that folds of the individual domains belong to more than one class. Multi-domain proteins with individual domain folds belonging to the same class can be found under the corresponding fold and class, mixed with single domain proteins.

In computational terms, although SCOP is essentially a hierarchy, a superposed mechanism for cross-linking between nodes of the tree makes it a more general form of graph. This allows the representation of biological relationships more complex than the parent-child relationships in a tree.

At the moment, this cross-linking ability is only used to group together domains belonging to the same PDB file but classified separately (see the ⤨ icon), or to indicate connections between SCOP entries, as in the case of the circular permutation that is likely to relate the threonyl-tRNA synthetase superfamily and LuxS family. However, the same mechanism can be easily extended to add new, possibly sparse, intermediate levels in the hierarchy.

Moreover, since the original design and implementation of SCOP [3] is based on a *description* of the underlying data structure rather than the data structure itself, it would be easy to introduce new classification levels everywhere, such a *suprafamily* between *family* and *superfamily*, for example. Once the description is modified accordingly, the rest of it will fall into place automatically. Extensions of SCOP were designed with this possibility in mind.

Keeping in mind the underlying, deep structure of SCOP while reading these notes may help you understand its surface representation both in terms of web pages and parseable files, as well as possible future extensions.

Access to the knowledge encoded in SCOP has always been a major concern. That's why all the information is also available in compact text files that are easy to parse, making large scale analysis of the SCOP content straightforward. A set of unique identifiers associated to each SCOP entry is used both for web pages and as keys in these parseable files. Unique identifiers remain stable across releases and provide an unambigous way of searching SCOP, linking to and from SCOP, tracking the history of any node in the hierarchy, and referring to SCOP entries in related research and in the literature.

Stable identifiers, parseable files, extended linking and searching options, and the concept of a history tracking the evolution of the SCOP classification itself were introduced in release 1.55 and have been developed since then to accommodate structural genomics requirements as well as other usages of SCOP for both research and educational purposes [4]. SCOP genetic domain sequences and coordinates, mapping between sequences from SEQRES and ATOM records in PDB files and other SCOP related data can be downloaded from the associated ASTRAL web site and were first described in [5].

SCOP is undergoing continual evolution. Older SCOP releases remain accessible, allowing redirection for entries that become obsolete over time. In this way, no information is ever lost and it is always possible to go back to the original data used in SCOP related research even after domain definitions or the classification itself has changed.

Major rearrangements of the classification have actually occurred in release 1.63 and 1.65 and more are expected in the near future. So far, the rearrangement involved about 11% of the domain entries, as described in [6]. The subset of the classification affected by these changes can be browsed in a SCOP-like form (see reclassified entries) and the history helps users follow all modifications in several useful ways [4].

- SCOP identifiers (see also SCOP parseable files)
  - Introduced: 1.55
  - Apply to: 1.55 --> current release
  - Reference: [4]

Starting with release 1.55, there are two series of new identifiers, SCOP unique identifiers (*sunid*) and SCOP concise classification strings (*sccs*). They are *stable across releases*, with the caveat that a domain definition or the classification itself can change. Nodes in the hierarchy can merge or split as more evidence about evolutionary relationships becomes available from experimental data. Corresponding identifiers become obsolete. No identifier is ever reused, and all modifications of the classification are documented and easily tracked (see SCOP history). Therefore, *stable* means that the same identifier will always refer to the same object in SCOP, but it does not mean that the SCOP entries themselves will not change.

A *sunid* is simply a number which uniquely identifies each entry in the SCOP hierarchy, including leaves (i.e., the SCOP domains) and entries corresponding to the protein level (for which there was no explicit reference before).

A *sccs* is a compact representation of a SCOP domain classification. A *sccs* identifier includes only the *class* (alphabetical), *fold*, *superfamily*, and *family* (all numerical) to which each domain belongs to.

Together, *sunid* and *sccs* replace the old *classification page numbers* (like 1.002.044.001.002.021). All these page names have been renamed in SCOP 1.55. If you still link to SCOP using a page name, despite this having been deprecated since 1994, all your links are broken. This eliminates the even worse situation of misleading links where a reference using a *classification page number* points to a completely different entry as soon as SCOP is updated because most of these page names change with every new release. (For those interested, page names are generated *via* enumeration of the SCOP tree and therefore change as soon as new nodes are inserted, and old nodes are deleted, merged, split or moved to a different location in the tree.)

sunid and, when applicable, *sccs*, can be visualized by placing the mouse on the appropriate link (i.e., either in the lineage or in the PDB entry section in a SCOP web page). Look at the bar at the bottom of your browser. Starting with release 1.65, *sunid* also appear explictly in the web pages.

- Introduced: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

- SCOP history

  - Introduced: 1.55
  - Apply to: 1.55 --> current release
  - References: [4], 1.63-1.65 reclassification [6]

Not only proteins evolve. Their classification evolves as well, as more evidence about evolutionary relationships become available from experimental data, or simply as the result of refinements, rethinking and bug fixing. This process is a natural one and it would be a mistake to freeze the SCOP content, as it is a mistake to freeze any form of knowledge and pretend it to be absolute and unquestionable.

On the other hand, data accuracy and data reliability are a must, especially for well established collections like SCOP, which are widely used to build upon and derive further research results.

Dynamic databases, and the best way of maintaining a history of their content are open research issues in the database field. In science, it is a basic tenet that results should be reproducible, which implies that the original data should remain accessible. These issues are addressed in SCOP at several levels and along different dimensions in order to satisfy different needs.

First of all, old SCOP releases remain available online in their entirety, in their original form and content, possibly with some additions that are backpropagated once introduced in more recent releases, like in the case of the reclassified entries history, but without deletions or changes. This is possible without exponentially growing resources thanks to a minimalist design that results in maximum information content with minimum waste of space. Moreover, SCOP parseable files, also available online for current and previous releases, provide a very compact summary of all the information encoded in SCOP.

Because old releases are available on the web, they can be searched in the same way as if they were the current release. Also, any reference in any release to an identifier that becomes obsolete for various reasons at a certain point in time remains valid: the search engine returns a pointer to the page(s) in the old release where it last appeared, together with links to what it became. See, for example, what happens when searching for 46781, the TnsA endonuclease, C-terminal domain superfamily in SCOP 1.55. This historical information is not only accessible interactively and for single entries, but it is also summarized in a list that can be downloaded from the SCOP web site.

In order to talk about what an entry was and what it becomes, there must be a unique, unambiguous way of referring to it. This function is fulfilled by *sunid*, the SCOP unique identifiers introduced in release 1.55 for this and other purposes. Nothing else would possibly work, as names in SCOP are not unique and may change even if the corresponding node in the hierarchy remains the same. As emphasized more than once in these notes, *sunid* are kept stable across releases. This means that if a node does not change, its *sunid* remains the same but, if the node does change, then its *sunid* changes, and the way in which it changes is explicitly documented.

Note that additions of new entries and deletion of obsolete ones do not constitute a problem. The former will get a new *sunid*, the *sunid* associated with the latter will simply become obsolete. Problems arise when an old entry in SCOP is modified without becoming obsolete, and new entries appear not because of the classification of new structures, but as a consequence of splitting or merging nodes that are already part of the hierarchy.

A typical example is when a domain in a multidomain protein already classified in SCOP is observed for the first time by itself, or in a different context, and therefore qualifies as a separate domain. The newly split domains can be classified in different parts of the hierarchy and, in principle, all their attributes, from species to class, may change. A more apparent case is when a full class, in SCOP for convenience, but marked as *Not a true class* is rearranged and becomes part of the classification, like the Membrane and cell surface proteins and peptides class, in which a single fold was replaced by 20 or more new folds in release 1.63 and several other folds were amply restructured.

Minor rearrangements of the classification are common. They occur at any new release and are documented through lists and redirections, as explained before. However, in SCOP 1.63 and 1.65, about 11% of the total number of already classified domains changed in terms of their definition, their classification, or both. These changes are described in [6]. More are expected in future releases.

Therefore, we introduced a SCOP-like way of browsing the subset of the classification that is modified in between releases, to help users analyze the effects of this reclassification process in a familiar way [4]. This is now available online for all releases, starting with 1.55. When fully expanded, it gives a detailed overview of all entries in the classification affected by the revision. See the changes that occurred between release 1.55 and 1.57, for example.

In the reclassified entries history, changes appear as comments associated to domain entries, with links to *sunid* in the corresponding revised classification, so that both the previous classification context for an entry and the new one can be easily browsed at the same time. Those SCOP features that are meaningful for the reclassified entries history are available. For example, you can expand and compress the history using the navigation buttons at the top of the page, look at protein domain structures using RASMOL or CHIME, and perform searches, though the search itself is limited to the history content and keywords include SCOP *sunid*, PDB identifiers, and words that appear in web history pages only.

- Introduced: 1.55
- Apply to: 1.55 --> current release
- References: [4], 1.63-1.65 reclassification [6]

- Searching SCOP

- Expanded options set starting with: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

There are two ways of searching SCOP: a simple keyword search using the search entry at the bottom of any of the web pages, and a more sophisticated way using the search engine from the location bar. The latter is at the same time a way of linking to SCOP.

A separate search is also possible for reclassified entries that you can browse in a SCOP-like form. Most of the options described below are available in that context as well, but search results are restricted to the subset of entries for which the domain definition changed or that have been reclassified. See SCOP history for more details.

Starting with release 1.65, there is a new option (see the search engine) to use MSDlite to search text fields in PDB files and return links to the corresponding SCOP entries (*2004-04-07*).

- Keyword search

Any of the SCOP identifiers (*sunid*, *sid*, *sccs*) can be used in the keyword search.

sunid returns the corresponding page in the SCOP hierarchy for any node in the tree. Being the primary key for both web pages and SCOP parseable files, *sunid* are by far the fastest way of retrieving a SCOP entry or linking to it. They are also stable across releases. Starting with release 1.65, they appear explicitly in the web pages (in square brackets), next to the entry that they encode. Please use a *sunid* to search or link to SCOP any time you can. They are the most reliable and general way of referring to any entry in SCOP.

sid are maintained for backward compatibility. Searching using a *sid* as key returns the page corresponding to that domain.

sccs can be right truncated at any level: searching with `a.1.1.1` returns the page corresponding to the truncated hemoglobin family, searching with `a.1.1` returns the page corresponding to the globin-like superfamily, searching with `a.1` returns the page corresponding to the globin-like fold, and searching with `a` as key returns the page corresponding to the all alpha proteins class.

PDB identifiers can also be used as keywords to search SCOP. If a PDB file content is split into several SCOP domains, a list of links to all domains is returned, as in the case of 1dan. If a PDB file becomes obsolete and is superseded by another one, the search engine provides a pointer to the latter. See the results of searching for 1hs0.

You can also use E.C. numbers as keywords. For example, 3.1.21.4 returns the list of restriction endonucleases enzymes classified in SCOP for which a mapping to E.C. numbers can be derived from PDB files. A more sofisticated mapping, on a *per*-residue basis, is also available for individual domains (see linking from SCOP).

Finally, you can search using any (at least two character long) word that appears in any of the SCOP pages. `fivefold`, for example, returns the page corresponding to the five-fold pentein fold. Note that `fivefold` is the spelling of the compound adjective in the SCOP comment, but words don't necessarily need to be complete. They can be right truncated by appending a `+` at the end, like in `hypoth+`, which returns a list of links to pages containing any completion of "hypoth".

Two or more words can be combined using `+`(*and*) and `-`(*and not*) word-prefix operators. Try `structural +genomic+`. This returns the list of pages in which both `structural` and (any completion of) `genomic` appear in comments. Conversely, `structural -genomic+` returns the list of pages containing `structural`, but not (any completion of) the word `genomic`.

Right truncation allows you to enter fewer characters *per* search. On the other hand, operators can be used to focus your search: `yeast +saccharom+` returns the list of yeast proteins for *Saccharomyces cerevisiae* only, while `yeast +saccharom+ +elongation` returns a list further restricted to elongation factor domains from *saccharomyces cerevisiae*. Compare with the results of searching for `yeast` only.

Useful keywords are `newfa`, `newsf`, `newcf`, `newcl`, first introduced in release 1.59. They return, respectively, the list of new families, superfamilies, folds, or classes for the current SCOP release and are part of the history facility. *New* here refers either to a new node in SCOP that did not exist in previous releases, or a node whose content has been modified not simply by addition of new entries or removal of obsolete ones, and therefore has acquired a new meaning (see SCOP history).

Note that keyword search is case-insensitive. This implies that PDB chain identifiers are treated uniformly, without distinction between upper- and lower-case. Very few PDB files that include both lower- and uppercase chain identifiers cannot be accommodated in the SCOP classification.

Names and comments in SCOP are a sort of primitive form of annotation and help produce biologically meaningful search results. However, please keep in mind that names are not unique, even within the same release. It is not uncommon for the same name to be used for, say, both the family and superfamily of a set of domains. Also, names are reused within the same level of the hierarchy, expecially for proteins. Different proteins, classified in different ways, may nevertheless be called in the same way.

All keyword searches are local to the current release. Previous SCOP releases can be searched using the search engine for the corresponding old release that we keep online as part of the SCOP history facility. In some cases, you can also use the search engine for the current release to search an old one (see Advanced search options). Moreover, a search for an identifier that is now obsolete automatically refers you both to the release in which it was last used, and to the new identifer(s) that replaced it. Follow the fate of a.4.4, for example.

Search options are also available for the reclassified entries history, though the search itself is limited to the history content and keywords include SCOP *sunid*, PDB identifiers, and words that appear in web history pages only.

Starting with release 1.65, if the search for a PDB identifier fails because it is not yet classified in SCOP, there is an option to use SSM to match the PDB file against the set of SCOP domains. See what happens if you search for 1f2y.

- Adavanced search options

Information in SCOP is interactively accessible through a search engine available at `http://scop.mrc-lmb.cam.ac.uk/scop` or any of the several mirrors scattered around the world. In the following explanation and examples, simply replace `scop.mrc-lmb.cam.ac.uk/scop` with your favourite SCOP server. Please also read the previous keyword search section, as most of what is described there applies here as well and will not be repeated.

By explicitly using the search engine from the location bar, you can perform most of the keyword search operations and obtain the same results. Moreover, you can specify options that modify these results. Most of the examples in this section are also a valid way of linking to SCOP.

The general way to search or link to any of the SCOP entries is by using the corresponding *sunid*:

`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=`*`sunid`*

returns the corresponding page, or a page indicating that *sunid* is obsolete, with links to the replacing one(s). Try:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=47419`
and:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=48328`.

You can also search or link to SCOP using a (possibly right-truncated) *sccs*, as in: `http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sccs=a.1.1.1`
or:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sccs=a.1.1`

Searching and linking to scop using a *sid* is still accepted for backward compatibility: `http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sid=d2cro__`

Searching using a PDB identifier returns the corresponding page: `http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?pdb=2cro`
or a list of pages, if the content of the PDB file is split into more than one domain and different domains are classified separately:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?pdb=1dan`

You can also search SCOP using E.C. numbers: `http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?ec=3.1.21.4`
and any word that appears in a web page: `http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?key=immunoglobulin`
including special words like `newfa`, `newsf`, `newcf`, `newcl`.

It is possible to pass parameters to the search engine. For example, you can specify for which release you would like the list of, say, new families introduced in that release. Default is the current release. Compare the results of:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?key=newfa`
and:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?key=newfa;ver=1.61`.

Also useful is the `lev` option. This allows you to specify the level in the SCOP hierarchy in which you are interested, independently of the level of your initial query. For example, you can search or link to the superfamily page for a given domain by specifying `lev=sf` in your query: `http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sid=d2cro__;lev=sf`.
Of course, you can obtain the same result by specifying the *sunid* for that superfamily:
`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=47413`
but the `lev` option is maintained for backward compatibility and because it can be useful if you don't have the *sunid* at hand.

By adding a *lev* option to a search using E.C. numbers, like in:
http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?ec=4.2.1.30;lev=sf.
you will get the list of superfamilies for which there is at least one SCOP domain with an E.C. number equal to 4.2.1.30. Compare with the results of:
http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?ec=4.2.1.30.

Valid levels are `cl` (class), `cf` (fold), `sf` (superfamily), `fa` (family), `dm` (protein), `sp` (species), and `px` (domain).

Finally, you can use the search engine to retrieve the SCOP parseable files (for current release only):

```
http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?dir=yyy
```

where `yyy` is either `cla`, `des`, `hie`, `com` (or, for old releases, `dom`).

All parseable files, for current and previous releases, are available online and can be retrieved using:

```
http://scop.mrc-lmb.cam.ac.uk/parse/filename_x.xx
```

where `filename` stands for `dir.cla.scop.txt`, or `dir.des.scop.txt`, or `dir.hie.scop.txt`, or `dir.com.scop.txt`, and `x.xx` is the release number (e.g., 1.65).

- Expanded options set starting with: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

- Linking to SCOP

- Expanded options starting with: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

The general way to link to any of the SCOP entries is by using the corresponding *sunid*:

```
http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=sunid
```

returns the corresponding page, or a page indicating that *sunid* is obsolete, with links to the replacing one(s). Try:
http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=47419
and:
http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?sunid=48328.

Please use the *sunid* any time you can and this is appropriate for your purposes. *sunid* are primary keys in the SCOP database and are kept stable across releases. Therefore, they are the fastest and most reliable way of referring to a SCOP entry at any level of the hierarchy, from leaves (e.g., domains) to classes.

All ways described in the advanced search options section are also valid ways of linking to different kinds of information available in SCOP, and allowing retrieval of this information. Please keep in mind that in some cases the link result may not be a specific web page, but a page of links to several web pages, or a warning page telling you that an entry is obsolete and redirecting you to its replacement(s).

- Expanded options starting with: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

- ## Linking from SCOP

- Introduced: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

The L icon associated with each domain integrates the SCOP classification for that domain with several other sources of related information. It can be easily expanded, provided there is a map between a key in SCOP (preferably the *sunid*) and the corresponding key(s) on the other side. It is straightforward to use the same linking mechanism for other intermediate levels in the SCOP hierarchy, if necessary.

See d1ckqa_ [sunid=33229], for example. Links are logically arranged, starting from sequence and coordinate data for the domain and expanding to genomic assignments and search for similar structures both in SCOP and the up-to-date PDB, which includes files not yet classified. Links also point to structural characterization of the domain, as well as further information about sequence and function, when available.

Some of the related information is actually at the level of PDB file rather than the SCOP domain. It is grouped separately, and provides useful summaries from well established web sites.

Most of these sites are synchronously updated with SCOP, shortly after any new release. Some other sites are updated more frequently than SCOP. Sites updated asynchrously are grouped together at the end of the list. Links to these sites appear only if there is a corresponding entry at the other end, to avoid useless and therefore disappointing trips over the net.

All these links can be dynamically updated between SCOP releases, as the information becomes available. Since a link has two ends by definition, none of these links would be possible without the much appreciated collaboration of the people who are taking care of these sites and provide curated mapping between their data and SCOP. For each of the links, credit is given by providing a pointer to the original sources.

- Introduced: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

- ## SCOP parseable files (see also SCOP identifiers)

- Introduced: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

The SCOP web site provides an easy way of interactively exploring the classification. However, this is not the most useful way of processing the SCOP content for large scale analysis projects, or when you want to use a computer program instead of your eyes and brains. Access to the knowledge encoded in SCOP has always been a major concern. That's why this knowledge is also available in compact text files that are easy to parse.

Starting with release 1.55, there are three new parseable files. Together, they replace and extend the now obsolete `dir.dom.scop.txt` and `dir.lin.scop.txt`. Each of these files has a header, starting with the '#' character, which includes release, version, and copyright information. These files have been designed in such a way that the likely inclusion of new levels in the current SCOP hierarchy will not break your code, provided you parse them correctly, without making any assumption about the order in which levels appear in the description. In the example below, don't expect `fa` to come next to `sf`, because we may insert a new "suprafamily" level in SCOP in the near future, or some other levels that may help to model the evolution of proteins better.

- `dir.cla.scop.txt`

```
d1dan.1 1dan    T:,U:91-106     b.1.2.1 21953   cl=48724,cf=48725,sf=49265,fa=49266,dm=49267,sp=49268,px=21953
d1danu1 1dan    U:107-210       b.1.2.1 21954   cl=48724,cf=48725,sf=49265,fa=49266,dm=49267,sp=49268,px=21954
d1danh_ 1dan    H:      b.47.1.2        26292   cl=48724,cf=50493,sf=50494,fa=50514,dm=50550,sp=50551,px=26292
d1danl1 1dan    L:49-86 g.3.11.1        44209   cl=56992,cf=57015,sf=57196,fa=57197,dm=57201,sp=57202,px=44209
d1danl2 1dan    L:87-142        g.3.11.1        44210   cl=56992,cf=57015,sf=57196,fa=57197,dm=57201,sp=57202,px=44210
d1danl3 1dan    L:1-48  g.32.1.1        44965   cl=56992,cf=57629,sf=57630,fa=57631,dm=57632,sp=57633,px=44965
d1mbd__ 1mbd    -       a.1.1.2 15033   cl=46456,cf=46457,sf=46458,fa=46463,dm=46469,sp=46470,px=15033
d1lvk_2 1lvk    2-33,80-759     c.37.1.9        32173   cl=51349,cf=52539,sf=52540,fa=52641,dm=52642,sp=52645,px=32173
```

`dir.cla.scop.txt` replaces and extends `dir.dom.scop.txt`. There is an entry for each SCOP domain. The first column is the old scop identifier (*sid*), the second column is the PDB identifier, and the third column is the domain definition, specified in terms of PDB chain and begin-end residue identifiers (including insertion code) from the ATOM field in the corresponding PDB file.

Note that begin-end can be missing, like in d1danh_, meaning that the domain corresponds to the whole chain. Also, some domains are composed of fragments from the same or from different chains, like d1lvk_2 and d1dan.1. In this case, fragments are separated by commas, and appear in the domain definition in the same order as in the original gene sequence, which is not necessarily the same as the order of chains in the corresponding PDB file. d1mbd__ and d1lvk_2 illustrate the case of a 'blank' PDB chain identifiers, the former spanning the whole chain, the latter being composed of two non-contiguous fragments from the same chain. Circular permutations are not uncommon and generally indicated as such in comments.

For details about chain and residue identifiers, please refers to the documentation for the PDB format available online at:

`http://www.rcsb.org/pdb/docs/format/pdbguide2.2/guide2.2_frame.html`

The fourth column in `dir.cla.scop.txt` is the *sccs*, the fifth column the *sunid* (`px`) for that domain, listed separately to ease parsing; the sixth column is the list of *sunid* for the domain: class (`cl`), fold (`cf`), superfamily (`sf`), family (`fa`), protein domain (`dm`), species (`sp`), and domain entry (`px`). They appear as pairs (key=*sunid*) so that order does not need to be taken into account in parsing the file and therefore intermediate levels can be added in the future without breaking code written now.

In all parseable files columns are `tab`-delimited.

- `dir.des.scop.txt`

```
48724   cl      b       -       All beta proteins
48725   cf      b.1     -       Immunoglobulin-like beta-sandwich
49265   sf      b.1.2   -       Fibronectin type III
49266   fa      b.1.2.1 -       Fibronectin type III
49267   dm      b.1.2.1 -       Extracellular region of human tissue factor
49268   sp      b.1.2.1 -       Human (Homo sapiens)
21953   px      b.1.2.1 d1dan.1 1dan T:,U:91-106
```

Each entry corresponds to a node in the SCOP hierarchy. The first field is the *sunid*. The second field is the entry type. The third is the *sccs* (for domain entries, species and proteins, it is the *sccs* for the corresponding family level). The fourth field is a "short name" for that entry (the scop identifier, or *sid*, in the case of domains, currently an empty place-holder for other types). The last field is the English description for that entry, the same one which appeared in the now obsolete `dir.lin.scop.txt` file, with the exception of domain entries, in which case it is the domain definition.

Note that names are not unique. *sunid* are unique. They are the only way of referring to an entry in SCOP without ambiguity.

Entries in `dir.des.scop.txt` and other parseable files are listed in the same order in which they appear in the SCOP classification. This is equivalent to explore the classification tree in depth-first order, always following the leftmost node first. The order of the entries is important, as often closely related entries appear next to each other in SCOP. This useful information is preserved in all parseable files. At the domain level only, entries are sorted according to the quality of the structure, as indicated by the corresponding AEROSPACI score. Therefore, the first entry in a list of domains for any given species in SCOP is the best molecule for that species in terms of structure quality.

By itself, `dir.des.scop.txt` contains all the information about a given SCOP release. It is the most compact way of representing SCOP and its tree-like structure allows for very efficient usage and search.

- `dir.hie.scop.txt`

```
0       -       46456,48724,51349,53931,56572,56835,56992,57942,58117,58231,58788
21953   49268   -
49267   49266   49268,49269
```

This file has no precursor in SCOP before release 1.55. It represents the SCOP hierarchy in terms of *sunid*. The first field is the *sunid* for a node, the second the *sunid* of its parent, and the third the list of *sunid* of its children. "0" is the root, which has no parent (-). "21953" is a scop domain (a leaf) and therefore has no children. "49267" is a protein-type *sunid*, it has a parent, and two children. As for all parseable files, if you properly parse `dir.hie.scop.txt` without making assumptions about the parent of a `fa`-type *sunid* be a `sf`-type *sunid*, your code will not break, should we decide to introduce new intermediate levels in SCOP.

- `dir.com.scop.txt`

```
80760 ! CASP5 ! structural genomics protein; complexed with fe, mse
```

Starting with release 1.63, we also distribute comments as they appear in the SCOP pages. Some of them are automatically generated, some are manually edited. Comments associated with a given sunid appear in the same line, as in the example above. Each comment starts with an esclamation mark ('!'). The primary key is the *sunid* associated with the entry in SCOP to which the comment refers. Thanks to Dave Howorth for writing the code to generate this file.

All parseable files for current and previous releases are available online at:

`http://scop.mrc-lmb.cam.ac.uk/parse`

and can be retrieved at:

`http://scop.mrc-lmb.cam.ac.uk/parse/filename_x.xx`

or using the SCOP search engine (for current release only):

`http://scop.mrc-lmb.cam.ac.uk/scop/search.cgi?dir=yyy`

`filename` is the name of the parseable file, `x.xx` the release number (e.g., 1.65) and `yyy` is either `cla`, `des`, `hie`, `com` (or, for old releases, `dom`).

-

Relation between old and new identifiers

The old *classification page numbers* (like 1.002.044.001.002.021) appear nowhere in recent SCOP releases. *Within* each SCOP release, there is a one-to-one correspondence between *sid* (like `d1dan.1`) and px-type *sunid* (like `21953`). This one-to-one correspondence is not necessarily maintained *across* releases.

While we can keep the *sunid* (and *sccs*) stable in the sense defined above, we don't have a direct control over a *sid*, which is formed using a PDB filename and a PDB chain identifier. *sid* are more like short informative names than unique object identifiers. Please use the new set of identifiers, especially for computer-based analyses, linking to SCOP, and referring to SCOP domains and classification in the literature. If you do so, it works across releases automatically. If you don't, any update or rearrangment of the classification will result in wrong references, wrong links, and so on.

- Introduced: 1.55
- Apply to: 1.55 --> current release
- Reference: [4]

- SCOP sequences and coordinate files (from the ASTRAL web site)
  - Introduced: 1.55
  - Apply to: 1.55 --> current release
  - Reference: [5]

-

Rapid access format file (RAF)

A manually curated mapping between SEQRES and ATOM fields for all PDB chains in SCOP is available at the associated ASTRAL web site. This provides a SCOP user with the official definition of a domain in terms of: a) SEQRES sequence; b) ATOM sequence; c) mapping between the two sequences; and d) the original PDB residue identifiers from the ATOM field (these latter uniquely define the domain in terms of PDB coordinates).

Given the nature of old and new PDB files, it is basically impossible to parse them or use any other derived data in a way that uniquely identifies the set of residues belonging to a chain. The RAF file is meant to overcome this difficulty by providing a compact reference to the original PDB file for SCOP domains.

Although in some cases this precision may seem unnecessary, in other cases it is indispensable because individual residues do matter, as in multiple sequence and structural alignments or detailed analyses of protein structural features and functionally important sites.

- *Genetic domain* sequences

PDB chains are not necessarily biologically meaningful units. A SCOP domain corresponds to an evolutionary unit and may include non-contiguous fragments from the same or from different PDB chains. See 21953 from PDB entry 1dan for example. In these cases, fragments are listed in the order in which they appear in the original single chain precursor, independently of their order in the PDB chain or the order of the chains in the PDB file. The order of the fragments is manually checked against sequence databases, and the few cases in which the order is presumed from a different source are reported in comments. Also reported in comments are the cases of circular permutations, in which the C and N terminal domains joined at a certain point, and then the sequence was broken again, but in a different point.

The set of SCOP genetic domain sequences is available at the ASTRAL web site, together with several other sets of sequences filtered according different criteria. Old ASTRAL SCOP sequences, in which fragments belonging to different chains are in separate fasta entries, are also available at the same site.

-

Genetic domain coordinate files

The set of SCOP domain coordinate files in PDB-style format is available at the ASTRAL web site. As for sequences, the order in which fragments appear in a domain coordinate set is not necessarily the same as the order in which they appear in a PDB chain, but rather reflects the order in the original single chain precursor, i.e., in the gene product.

We would like to encourage you to use these reference data sets in your research work. This would make comparison, linking, and integration of SCOP-based results a trivial task. The purpose is to develop a common language that we can use without ambiguities when talking about a SCOP domain and its classification. The original PDB file is the only shared basis among research groups worldwide, and by explicitly referring to the PDB file via the RAF file and the datasets derived from it we hope to reduce the confusion.

- Introduced: 1.55
- Apply to: 1.55 --> current release
- Reference: [5]

- References

Original work:

1) Murzin,A., Brenner,S.E., Hubbard,T.J.P. and Chothia,C. (1995) SCOP: a Structural Classification of Proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247, 536-540. [PDF]

2) Brenner,S.E., Chothia,C., Hubbard,T.J.P. and Murzin,A. (1996) Understanding protein structure: using SCOP for fold interpretation. *Methods Enzymol.*, 266, 635–643. [Medline]

3) Brenner,S.E. (1996) *Molecular Propinquity*, PhD dissertation, Cambridge University, Cambridge, UK.

Major additions to SCOP (stable identifiers, parseable files, extended searching and linking to and from SCOP options, reclassified entries history):

4) Lo Conte L., Brenner S. E., Hubbard T.J.P., Chothia C., Murzin A. (2002). SCOP database in 2002: refinements accommodate structural genomics. *Nucl. Acid Res.* 30(1), 264-267. [PDF]

Major additions to SCOP (sequences and coordinate files):

5) Chandonia,J.M., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M. and Brenner,S.E. (2002) ASTRAL compendium enhancements. *Nucleic Acids Res.*, 30, 260–263. [PDF]

Reclassification of SCOP entries carried out in release 1.63 and 1.65:

6) Andreeva A., Howorth D., Brenner S.E., Hubbard T.J.P., Chothia C., Murzin A.G. (2004). SCOP database in 2004: refinements integrate structure and sequence family data. *Nucl. Acid Res.* 32:D226-D229. [PDF]

Loredana Lo Conte (Last update 2004-04-07,2004-02-23)