# Simple Atom Depth Index Calculator

*Release 1.0*

Daniele Varrazzo

March 12, 2005

Dept. of Molecular Biology
University of Siena



**Abstract**

Atom depth has been considered as a structural descriptor to correlate protein structure with folding and functional properties. The distance between an atom and the nearest water molecule or the closest surface dot has been proposed as a measure of the atom depth, but, in both cases, the three-dimensional character of depth is largely lost: an atom located at the bottom of a pocket may be at the same distance from the protein surface of an atom located in a protruding loop, but the former will be less accessible than the latter.

SADIC implements a new algorithm to calculate atom depth: it calculates the intersection between the molecular volume and spheres centered on the atoms whose depth has to be quantified. SADIC takes PDB files as input and can emit different measurements of exposed surfaces and volumes as well as a novel depth index, designed to promote comparison between atoms in a molecule and in different molecules.

## Contents

# 1 Atoms exposition

The depth of atom from protein surface has been proposed as a criterion to define protein structures. Current methods consist in evaluating the smaller distance between an atom and a dot of the solvent accessible surface [1] or the distance between an atom and its closest solvent accessible neighbor [2]. These methods have the drawback to lose the contribution from the three-dimensional molecular shape.

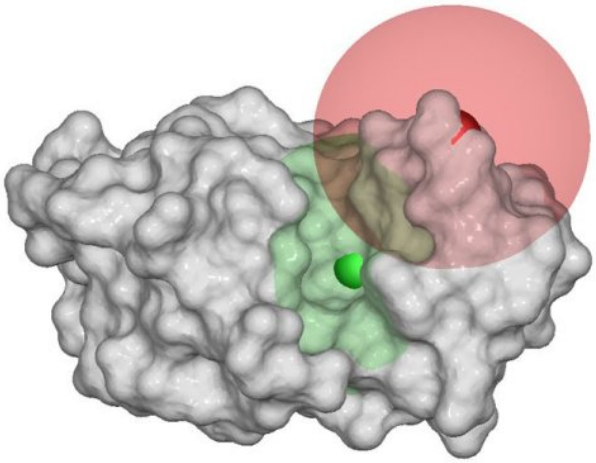

Figure 1: Hen egg white lysozyme: $\alpha$-carbon 47 (red) and 58 (green) are both on the molecule surface, but accessibility of the former is quite greater

The purpose of SADIC is to calculate atoms exposition keeping into account the three-dimensional shape of the molecule. The program simulates the molecule probing by a probe of given radius and calculates the exposed volume and surface as seen by the probe, as well as the *depth index*.

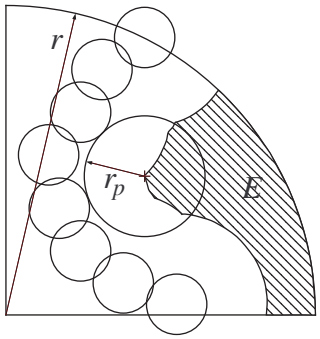


Figure 2: Area $E$ represents the space external to the molecule as seen by the probe

### 1.1 Depth index

The depth index is defined as

$$D_{i,r} = 2V_{i,r}/V_{0,r},$$

where $V_{i,r}$ is the exposed volume of a sphere of radius *r* centered on atom *i* and $V_{0,r}$ is the exposed volume of the same sphere when centered on an isolated atom. Given this definition, the depth index results in a number ranging from 0 to 2: its value is 0 if the atom is completely buried inside the molecule and no external point is closer than *r*; a theoretical value of 2 would be obtained if the atom were completely isolated. If an atom center lied on a theoretical perfect plane — splitting the space into an "internal" and an "external" half-space — its $D_{i,r}$ would be exactly 1.

## 2 Program installation

### 2.1 Prerequisites

SADIC is implemented as a Python script. Before installing SADIC you should have Python installed on your system. At least Python version 2.3 is required. No Python experience is required to run SADIC. You can find the most recent Python distribution at http://www.python.org/download.

The **numarray** package is also required to run the program. You can download it at http://www.stsci.edu/resources/software_hardware/numarray.

### 2.2 Installation

SADIC is distributed in two versions:

- a multi-platform source installation;
- a Windows self-installing executable.

Windows users are encouraged to use the self-installing version. Just run the install program to get SADIC installed. The install program will also create a file called 'RunSadic.bat' in the Windows folder: this way the program can be run by simply typing **RunSadic** on a command prompt.

To install SADIC on a POSIX system, you should have `root` privileges. Unpack the source package, move into the unpacked directory and then type

```
python setup.py install
```

The script **RunSadic** will be installed in the /usr/bin directory.

## 3 Program usage

SADIC is implemented as a command line tool. The basic usage is:

**RunSadic** [*options* [...]] [**--**] [*entity*]

The full list of options is described in the section 4. You can use the **--** characters if the last option gets confused by the entity.

*entity* is the input data. Further details are provided in the section 3.1.

## 3.1 Program input

The program input is always a PDB entity file. This file can be entered:

- by file name for a file in the local file system; es.

```
RunSadic c:\Dati\Pdb\1AON.pdb
```

- by URL if the file is in a network location; the protocol can be `http`, `ftp`, `file`; es.

```
RunSadic ftp://someserver/somepath/1AON.pdb
```

- by `pdbId` code: the file will be looked for in a list of databases given in the pdblist.conf file (see further); es.

```
RunSadic 1AON
```

- through the stdin; es.

```
gzip -dc 1AON.pdb.gz | RunSadic
```

The type of entity is automatically detected from the command line argument.

A PDB entity file can contain a single model or many models in different `MODEL...ENDMDL` blocks. SADIC treats each model as a different problem, emitting a different output for each one of them; a set of output files containing statistical data are also emitted. The models to perform calculations on are selected through the **--models** option; by default calculations are performed on all the models.

PDB entities can be read from files compressed in the **compress** format: in this case they must present a '.z' or '.Z' extension. For patent issues, the program is not provided with a compress tool: it uses an external program such as **gzip** to perform decompression. The program is quite standard in Linux installations; for Windows platform it can be downloaded it from http://www.gzip.org. The tool must reside in a PATH directory.

### Databases list

The pdblist.conf file tells the program where to look for PDB entities if the user searches for them by `pdbId` code. pdblist.conf is a text file containing a sequence of URLs with a variable part in the form `%(`*string*`)s`. When a search is to be performed, the variable part is replaced with a real value. Currently *string* can be:

- `pdbid`: replaced by the lowercase `pdbId`
- `PDBID`: replaced by the uppercase `pdbId`

The file can also contain any whitespace and comment (marked with a '`#`' character).

The resource is looked for in the provided URLs in the order they appear in the pdblist.conf file: the first URL found is used for calculation. If you want to look for PDB files in a local directory before trying some public database over the Internet, you can specify a `file` URL in the first position of the pdblist.conf file.

**Note:** On Windows platforms, the `file` URL syntax is slightly different from the standard way to express a directory path. For example, you can look for files named 'pdb*abcd*.ent.Z' in the directory 'C:\path\to\files' (where *abcd* is the `pdbId`) using the syntax

```
file:///C|path/to/files/pdb%(pdbid)s.ent.Z
```

The pdblist.conf file can reside in any of the following positions:

- the user's 'home' (on Windows is the location in the HOMEPATH environment variable, usually 'C:\Documents and Settings*username*');
- the '/etc/' directory (for POSIX systems);
- the program package directory. The package is installed in the standard 'site-package' directory of your Python installation. To know exactly where it is, you can use the command:

```
python -c "import os, sadic; print os.path.dirname(sadic.__file__)"
```

The program looks for the file in the given order and uses the first found. A sample (and already useful) pdblist.conf file is already installed in the package directory: you can copy it in your 'home' directory and update the copied version to override defaults without loosing the original values.

## 3.2 Sampling points

By default calculation is performed on every $\alpha$-carbon atoms in the molecule. The option **--atom-name** tells the program to sample all the molecule atoms with a given `name`. If you want to limit the selection to a list of residues and/or chains you can use the **--residues** and **--chains** options. A range of residues can be expressed in *N1–N2* form (both included).

Alternatively the atoms to be sampled can be selected by serial through the **--serials** option (which can also be in *N1–N2* form). Options **--residues** and **--chains** are not used in this case.

**Note:** Each `serial` in an entity file is unique across models: through the **--serials** option, atoms from different models can be selected. If the queried models have a different number of selected atoms, totals cannot be computed.

Finally, sampling can be performed around a generic point in the space. The point can be expressed through the **--point** option using coordinates in the entity space. The option can be used more than once to specify many points.

## 3.3 Sampling pattern

Calculation of the depth index is performed through sampling. The molecule is modelled as an assembly of spheres whose centers are the atoms center and whose radius are the atoms Van der Waals (VdW) radius. Samples are gathered into a spherical volume around the sampling point. Samples are placed over concentric spherical surfaces with regularly growing radii and are about equally spaced on each sphere. The parameters controlling the sampling pattern are:

**Sampling radius:** radius in Å of the sampling area. It is the fundamental parameter *r* in the depth index definition. It can be changed with the **--radius** option. If not specified, sampling will be performed along growing radii until no sampled atom is completely inside the molecule. When an entity file presents many models, radius is selected on the first model and applied unchanged on further ones.

**Radial step:** length in Å of the sampling steps in radial direction. This parameter influences the calculation precision. In the case of tabular output, a value for each sampling radius is returned. It can be changed through the **--step** option.

**Cross step:** maximum distance in Å between adjacent samples on each concentric sphere. Their number is kept as low as possible without exceed this value. By default this value is homogeneous to the step in the radial direction , but it can be varied independently through the **--cstep** option.

**Note:** If the **--cstep** parameter is not used, halving the radial step through the **--step** parameter results in about eight time the samples obtained, and consequently calculation time. If such a dramatic increment is not needed, the parameters **--step** and **--cstep** can be varied independently:

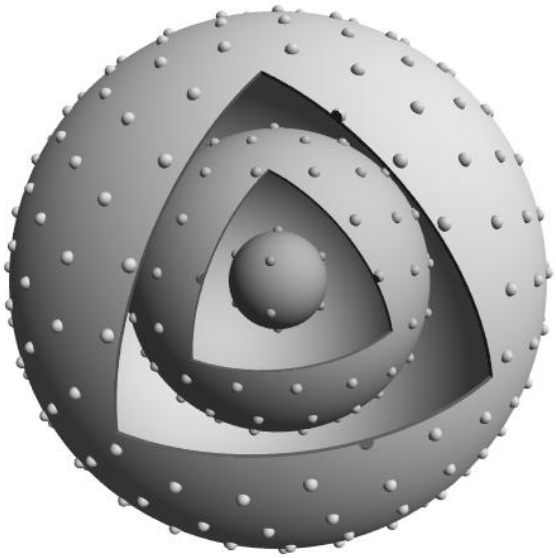

Figure 3: Sampling scheme. Samples are about regularly spaced over spheres.

- halving the **--step** value while keeping **--cstep** fixed results in about twice the samples. Furthermore, using a tabular output with **--format** `table`, the table will have twice the columns: this may be useful if you are interested, for example, in plotting the depth index per radius;
- halving the **--cstep** value while keeping **--step** fixed results in about four time the samples. Precision increases, but the number of columns in a **--format** `table` output doesn't change.

## 3.4 Model control

The model representation is built with a conventional VdW radius for each element. Conventional radii are obtained from [3] and can be found in section 4.5. Only radii for the most fundamental elements (C, H, N, O, S, P) are set; if you need to change any of the default value or add some missing element in the list, use the **--atom-radii** option. If the radius for any atom name in the entity file is missing, computation can't be performed.

By default, sampling is performed only on residues atoms (i.e. the records `ATOM` in the PDB file). If you want to include `HETATM` records too, you need to use the **--hetatm** option and select the atom radii for all the elements through the **--hetatm-radii** option. An `HETATM` radius which is not found in the **--hetatm-radii** list is searched in the **--hetatm** list (including the default radii) before giving up.

The radius selection, both for `ATOM`s and `HETATM`s, is peformed on the base of the `name` field. The name doesn't have to be perfectly equal to what selected through **--atom-radii** and **--hetatm-radii**: if a name is not found, the last character is stripped away and what remains is tested again, until a match is found (or nothing remains). Furthermore a leading digit is stripped from the `name`. For example atoms `CA` match the radius set for the element `C` and, for an atom whose name appears as `1HH1` in the PDB file, the radius set for `H` is used.

Solvent molecules are excluded from the computation. The default list of `resName`s values for `HETATM` to be ignored can be altered through the **--solvent** option.

**Note:** Many NMR generated models include dummy atoms whose name begins with `Q`. If you want to avoid them in calculation, you can set their radius to zero through the **--atom-radii** option. Es.

```
RunSadic --atom-radii Q 0 -- 1pit
```

## 3.5 Program output

SADIC can emit many informations for each model sampled. Each data stream is stored in a different file, and if the entity contains many models, each model generates a different set of data streams.

The data streams to save are chosen through the **--data** option with the sequence of codes the user is interested in. The meaning of each output symbol is:

**di:** the depth index;

**hv:** the hidden volume;

**ev:** the exposed volume;

**hs:** the hidden surface;

**es:** the exposed surface.

By default only the depth index is calculated.

The output files names are obtained by mangling the input entity name (if not applicable, 'out' is used) with the output symbol and the model serial. If the input file name was '8cho.pdb', the output file for the depth index will be '8cho_di.*ext*'. If the input file was the entity '1PIT', containing 20 models, the exposed volume files will be named from '1PIT_m01_ev.*ext*' to '1PIT_m20_ev.*ext*'. The value for *ext* depends on the output format.

If the entity contains more than one model, two cumulative files for each output streams are generated too. In the latter example, their names are '1PIT_avg_ev.*ext*' and '1PIT_std_ev.*ext*'; respectively containing the average and the standard deviation over the models of the data stored in the corresponding files.

A different base name, as well as a different path for storing output files, can be chosen through the **--output** option.

### Tabular output

By default, the data streams are stored in a tabular output file. Each atom sampled is stored in a different row; the table has a column for each sampling radius. The file also presents an header row with the sampling radius in Angstroms and an header column with the atom serial (or the point coordinates for **--point** sampling points. Many details in the output format can be chosen: refer to the section 4.6. The file extension for tabular output is '.txt'.

### PDB output

Using the option **--format** `pdb`, the output files are stored as PDB entity files, useful for visualization into a molecular display program such as **MolMol** (http://hugin.ethz.ch/wuthrich/software/molmol/). In this case only data relative to the biggest radius are stored. The file extension for PDB output is '.pdb'.

Data are stored in the `tempFactor` field of the ATOM records. Because the field can only range from 0 to 99.99, volume and surface data are normalized in the 0–1 range. Depth index values range in the 0–2 interval, so further rescale is not needed. Atoms not sampled are reported with a 99.99 value.

SADIC provides a **MolMol** macro displaying atoms with `tempFactor` values ranging in 0–2 in a colour blend and out of range atoms in gray. The macro is called 'di.mac' and is located in the package directory.

# 4 Command line arguments

## 4.1 Synopsis

Usage: **RunSadic** [*options*] [**--**] [*entity*]

*entity* can be either a file name, a fully qualified URL or a pdbId code. The URL protocol can be `file`, `http` or `ftp`. If a pdbId is specified, the entity is searched by mangling the id with the URL schema specified in the file

pdblist.conf. If entity is missing, data are read from stdin. The entity can be compressed in the **compress** format: in this case it must present a .Z or .z extension.

## 4.2 Options

**--version**
show program's version number and exit

**-h, --help**
show this help message and exit

**-d *S1* [*S2* [...]], --data *S1* [*S2* [...]]**
Output the data streams *SN*. Each *SN* can be one of the strings:

- `di` (depth index),
- `hv` (hidden volume),
- `ev` (exposed volume),
- `hs` (hidden surface),
- `es` (exposed surface).

Default: `di`.

**--models *M1* [*M2* [...]]**
Select the models to query. Useful if the entity contains more than one model (`MODEL`/`ENDMDL` blocks). Each *MN* is an integer or an integers range in *N1−N2* format. If not specified, query all the models.

**-o [*PATH*]*FILENAME.EXT*, --output [*PATH*]*FILENAME.EXT***
Base name for output files. Many files may be written: one for each model in the entity and for each data stream (option **--data**). Generate files FILENAME_*SN*.EXT (if the entity contains one model) or FILENAME_m*MN*_*SN*.EXT (if the entity contains many models), where *SN* is replaced by the symbol of the data stream, and *MN* by the model serial. By default *FILENAME* is the entity name, *EXT* depends from the output format. If *PATH* is not specified, store files in the current directory.

**-f *FORMAT*, --format *FORMAT***
Output format. Currently `table` and `pdb` are implemented. If *FORMAT* is `table`, store a table with the output values, with a sampling point in each row and a column for each sampling radius. If *FORMAT* is `pdb`, store a PDB entity with the required data in the `tempFactor` field. Default: `table`.

**--quiet**
Don't emit info messages. If not specified, print info messages on stdout.

## 4.3 Query options

These options control the points where to sample. At most one of the options **--atom-name** (default), **--serials** and **--point** can be specified.

**--atom-name *ATOM***
Process every atom with `name` field *ATOM*. Default: `CA`.

**--residues *R1* [*R2* [...]]**
Only useful with **--atom-name**: process only atoms named *ATOM* with `resSeq` field *RN*. Each *RN* is an integer or an integers range in *N1−N2* format. If not specified, sample all the residues.

**--chains *C1* [*C2* [...]]**
Only useful with **--atom-name**: process only atoms named *ATOM* with `chainID` field *CN*. Each *CN* is a letter. If not specified, sample all the chains.

**--serials *S1* [*S2* [...]]**
Process only atoms with `serial` field *SN*. If the entity contains many models, only one model must be selected (through the **--models** option) and the atoms must be in the selected model.

**--point *X Y Z***
Process the point *X Y Z* in the entity space. Many points can be sampled by selecting this option more than once.

## 4.4 Sampling options

These options control how the sampling is performed on the selected points.

**-r *RADIUS*, --radius *RADIUS***
Sampling radius. If not specified, automatically select a radius such that no atom has depth index = 0.

**--step *STEP***
The sampling step size, in Angstroms. If the **--cstep** is specified, the sampling step in radial direction. If **--radius** is used too, the program may reduce the step to exactly fit the chosen radius. Default: `1.0`.

**--cstep *STEP***
Sampling step along directions orthogonal to the radius, in Angstroms. If not specified, use the value selected with **--step**.

## 4.5 Entity options

These options control the entity details.

**--probe-radius *RADIUS***
Approximate the probe with a sphere of radius *RADIUS*. The approximation is performed by adding *RADIUS* to the radius of every atom in the model. Default: `1.4`.

**--atom-radii *A1 R1* [*A2 R2* [...]]**
Radii for `ATOM` records. *AN* is an atom name and *RN* its radius. User defined values replace or are added to the default mapping. Every atom name in the entity must be specified. Default:

- H: 1.000
- O: 1.480
- N: 1.625
- C: 1.700
- S: 1.782
- P: 1.871

**--hetatm**
Include `HETATM` atoms in sampling. A radius for each atom name in the entity `HET` group must be provided through the **--hetatm-radii** option.

**--hetatm-radii *A1 R1* [*A2 R2* [...]]**
Radii for `HETATM` records. *AN* is an atom name and *RN* its radius. If a name is not found in this list, its radius is looked for in the **--atom-radii** list. Every atom name in the entity must be specified. Default:

- F: 1.560
- CL: 1.735
- BR: 1.978
- I: 2.094

**--solvent *S1* [*S2* [...]]**
`HETATM` records with `resName` *SN* is a solvent and must not be sampled. Default: `H2O WAT SOL HOH DIS`.

## 4.6 Output options

These options control the output details.

**--no-totals**
: If the entity contains more models, don't compute totals. If not specified, store the average and standard deviation of the **--data** stream *SN* in files named respectively *FILENAME*_avg_*SN*.*EXT* and *FILENAME*_std_*SN*.*EXT*.

**--table-bare**
: Don't save column and row headers in the output files for **--format** `table`. If not specified, the first column will contain the atom serial (or point coordinates) and the first row the sampling radius.

**--table-fmt *FMT***
: The format of numbers in output for **--format** `table`. Must be a proper floating point placeholder. Default: `%.3f`.

**--table-sep *SEP***
: The separator of numbers in output for **--format** `table`. Enter `/t` for a tab character, `//` for a single /. Default: `/t`.

# References


[1] S. Chakravarty and R. Varadarajan. Residue depth: a novel parameter for the analysis of protein structure and stability. *Structure Fold Des*, 7:723–732, 1999.

[2] A. Pintar, O. Carugo, and S. Pongor. Atom depth as a descriptor of the protein interior. *Biophysical Journal*, 84:2553–2561, April 2003.

[3] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. R. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comp. Phys. Commun.*, 91:1–41, 1995.