structuprint

Documentation

Version 1.001
17th September 2015

https://dgkontopoulos.github.io/Structuprint/

Contents

1 Introduction 3

2 Installing Structuprint 5
2.1 From prebuilt packages 5
2.2 From the source code (un*x systems only) 5

3 `structuprint`'s manpage 6

4 `structuprint_frame`'s manpage 9

5 Structuprint's GUI 11

6 Tutorial: visualizing frames from an MD simulation 12

7 Common warnings/errors 14

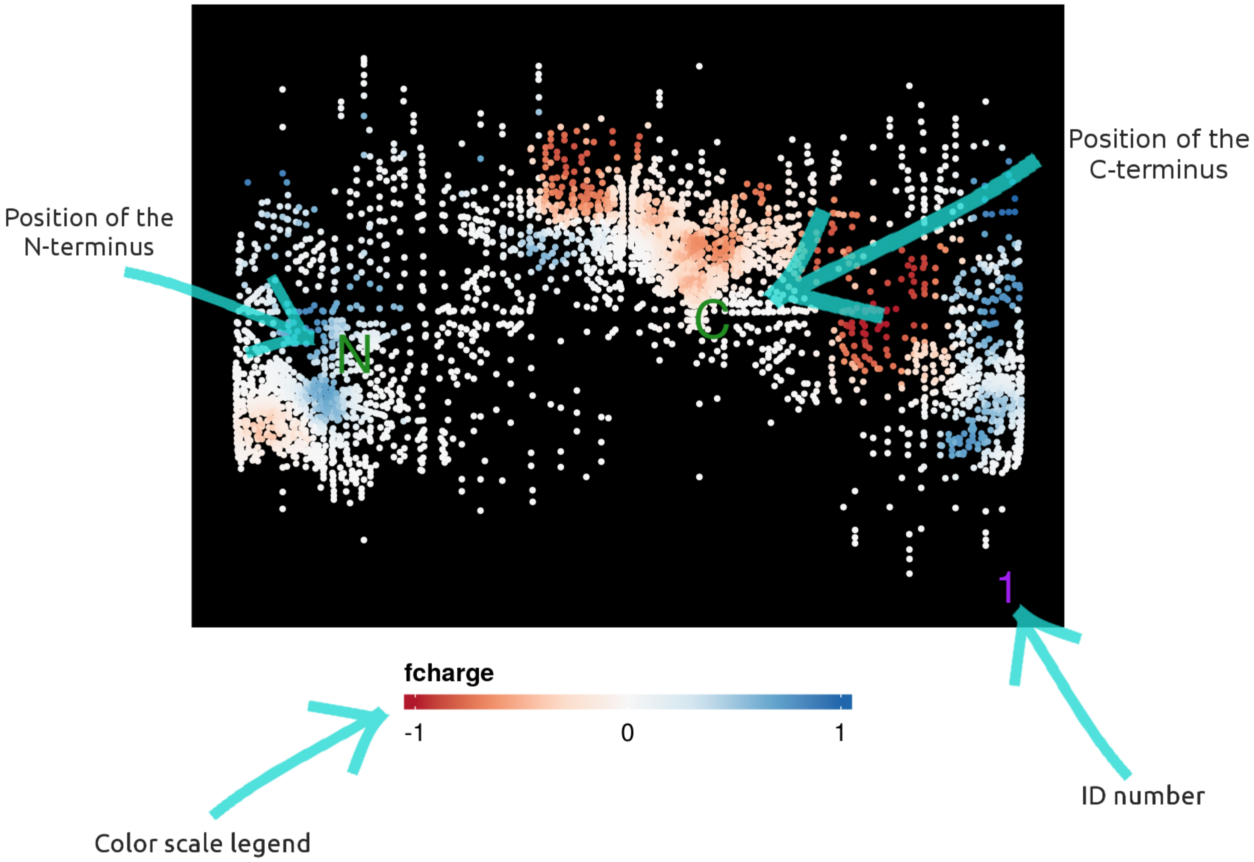
8 Advanced usage 15
8.1 Extending the default database 15
8.2 Custom database 15

9 Known issues 16

1 Introduction

Structuprint is a software tool for two-dimensional visualisation of surfaces of PDB structures. Details about the algorithm executed by Structuprint will be soon made available, in the form of a scientific publication. The name stems from the fingerprint-like figures that it produces (see below); one could think of them as the fingerprints of protein structures. As this name can be used both for the tool itself and its resulting figures, from now on Structuprint (with an uppercase "S") will refer to the software, whereas structuprint (with a lowercase "s") to the figure generated with it.

This is how a structuprint looks like:



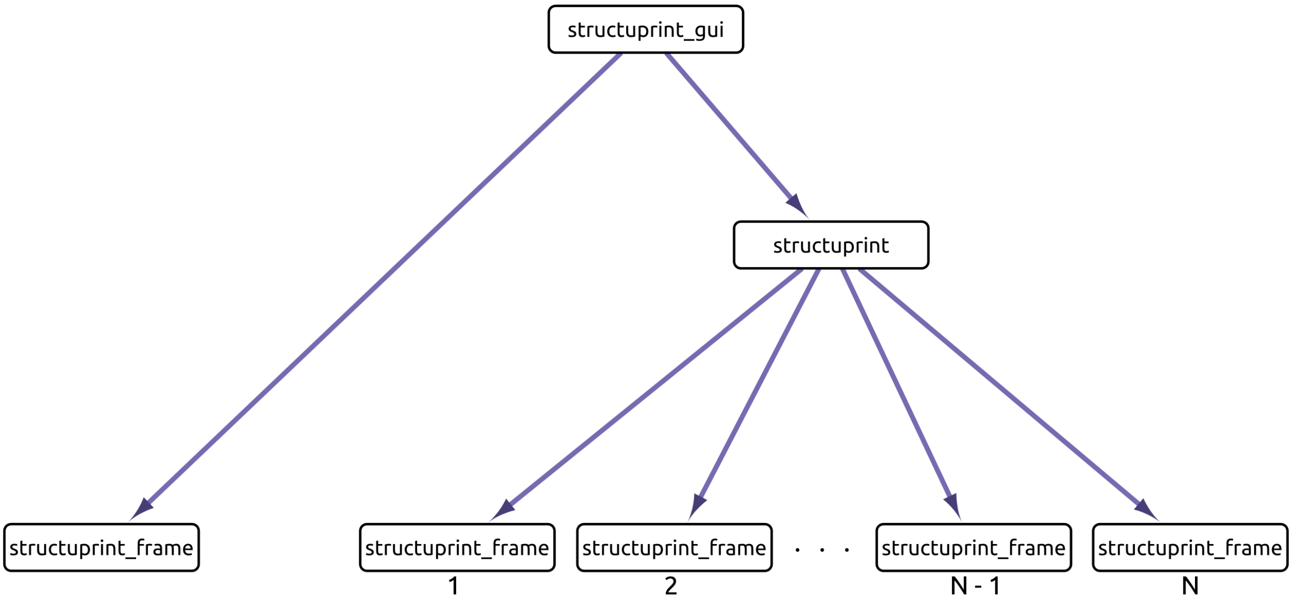
The data points in the figure are colored according to the value of a selected property, in this case `fcharge`, i.e. charge. In total, the values of 328 physico-chemical properties have been pre-calculated for the 20 common amino acids using MOE 2012.10. See the amino acid properties codebook for a list of the available properties and their explanations. At the bottom of the figure lies the color scale legend, along with its title which can be set by the user.

The green N and C letters show the approximate positions of the N- and C-termini. Structuprint treats each model in the PDB file as a different polypeptide chain, so the N- and C-termini correspond to the first and last residue in the model.

The purple number at the bottom right corner indicates the ID number of the PDB file. All of these features are optional and can be disabled.

Apart from a single figure like the one shown above, Structuprint can also produce animations from multiple PDB files. Therefore, to allocate the execution of different tasks, the tool consists of three scripts: i) `structuprint`, ii) `structuprint_frame` and iii) `structuprint_gui`.

The core algorithm that transforms a 3D PDB file into a 2D figure is executed by `structuprint_frame`. For animations, `structuprint` coordinates the generation of multiple frames and joins them into a GIF file at the end. Finally, `structuprint_gui` provides a Graphical User Interface to the above tasks. The figure below shows the relationships among the 3 scripts.



The `structuprint` script expects PDB filenames to contain an underscore and an ID number before the '.pdb' suffix (e.g. mds_1.pdb). The input directory can contain multiple numbered PDB files. Instead, `structuprint_frame` requires that the input directory contains only a single PDB file that may or may not be numbered.

2 Installing Structuprint

2.1 From prebuilt packages

Prebuilt packages and installers are available from Structuprint's website. They handle the installation of the program along with any needed dependencies. An exception is the package for CentOS Linux, which requires that the `epel-release` package is already installed, before the Structuprint package can be used.

2.2 From the source code (un*x systems only)

The source code is available from Structuprint's GitHub repository at https://github.com/dgkontopoulos/Structuprint. Download the latest release as a compressed file and uncompress it:

```
tar xzvf structuprint_src_1_00.tar.gz && cd structuprint_1_00/
```

A `structuprint_1_00` directory will be created. Inside it, you will find a `Makefile`. While inside that directory, type the following commands:

```
1 make test
2 make
3 make install
```

Line 1 will run some tests to make sure that all dependencies are available. When installing Structuprint from the source code, you have to install all required dependencies manually. These comprise the `perl` and `R` executables, but also Perl modules (`Astro::MapProjection`, `Bio::PDB::Structure::Atom`, `DBI` ...) and R packages (`ggplot2`, `grid`, `scales`, `labeling`). Some of them have their own dependencies, but you may be able to install them all at once via your package manager (e.g., `apt-get` on Debian or `pacman` on Arch Linux).

Useful installation tutorials

-Perl modules:	http://perlmaven.com/how-to-install-a-perl-module-from-cpan
-R packages:	http://www.r-bloggers.com/installing-r-packages/

Line 2 will make the scripts executable and line 3 will install all the necessary files at `/opt/structuprint/`. To change the installation directory, type:

```
make install -PREFIX=/PATH/TO/CUSTOM/DIR
```

To uninstall Structuprint, type the following command, followed by a `PREFIX` argument, if needed:

```
make uninstall
```

3 STRUCTUPRINT'S MANPAGE

NAME

structuprint - utility for 2D animations of protein surfaces

SYNOPSIS

```
structuprint -prop PROPERTY
             -dir INPUT_DIRECTORY
             [-outdir OUTPUT_DIRECTORY]
             [-custom_db PATH_TO_DATABASE]
             [-height HEIGHT] [-width WIDTH]
             [-res PPI_NUMBER]
             [-point_size SIZE]
             [-bgcol HEX_COLOR|COLOR_NAME] [-bgalpha ALPHA_VALUE]
             [-legend_title TITLE]
             [--no_ID] [--no_legend] [--no_NC]
             [--del_temp_dirs]
             [-delay DELAY]
             [-nloops NUMBER_OF_LOOPS]
             [-nthreads NUMBER_OF_THREADS]
             [--help]
             [--properties]
```

DESCRIPTION

structuprint is a tool for generating two-dimensional animations of protein surfaces. Given an input directory with properly named PDB files, structuprint will render each frame separately and join them into an animation at the end.

OPTIONS

-bgalpha ALPHA_VALUE

> Set the transparency level of the background color from 0 (fully transparent) to 1 (no transparency). Default value: 1

-bgcol HEX_COLOR|COLOR_NAME

> Set the background color either as an HTML hex color or as a color name that R understands. Default: #000000

-custom_db PATH_TO_DATABASE

> Provide the path to a custom SQLite database of amino acid properties. For information about the schema, refer to the documentation.

--del_temp_dirs

> By default, structuprint creates one directory per frame and one for the final animation. With this flag, only the final animation directory will remain at the end.

Do NOT use this flag if you are trying to report a bug, as all the log files from the individual frames will be deleted.

-delay **DELAY**

Set the delay between individual frames in milliseconds. Default value: 100

-dir **INPUT_DIRECTORY**

Specify the directory location of the input PDB files. The PDB filenames must contain an underscore and an ID number before the pdb suffix, e.g. mds_1.pdb, mds_2.pdb ...

-height **HEIGHT**

Specify the height of the animation in mm. If only **-width** is set, then `structuprint` will automatically adjust the height to the appropriate value. Default values: 76.56 by default or 66 when the **--no_legend** flag is active.

--help

Show the available options and exit.

-legend_title **TITLE**

Specify the title of the legend. If this is not set, then `structuprint` will use the name of the selected property as the legend title.

-nloops **NUMBER_OF_LOOPS**

Set the number of loops for the animation. Default value: 0 (infinite loops)

--no_ID

Remove the ID numbers from the frames of the animation.

--no_legend

Remove the legend from the frames of the animation.

--no_NC

Do not show the N/C-termini positions in the frames of the animation.

-nthreads **NUMBER_OF_THREADS**

Set the number of parallel threads to be launched by `structuprint` in order to speed up the execution. Do NOT ask for more threads than the number of cores in your system or you will suffer a decrease in performance! Default value: 1

-outdir **OUTPUT_DIRECTORY**

Specify the location for `structuprint`'s output directories. The directory must be empty. If this is not set, `structuprint` will write its output in the input directory.

-point_size **SIZE**

Specify the size of the data points in the frames of the animation. Default value: 1

-prop **PROPERTY**

Specify the amino acid property based on which the animation will be colored.

--properties

List all the amino acid properties available in the default database along with their explanation, and quit.

-res PPI_NUMBER

Specify the resolution of the animation in pixels per inch. Default value: 100

-width WIDTH

Specify the width of the animation in mm. If only **-height** is set, then Structuprint will automatically adjust the width to the appropriate value. Default value: 90

EXAMPLE

```
structuprint -dir '/Data/' -prop FCharge -legend_title 'Charge' -width 250 -res
300 -outdir '/Results/' -nthreads 4
```

SEE ALSO

Basic R colors - http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

4 STRUCTUPRINT_FRAME'S MANPAGE

NAME

structuprint_frame - utility for generation of 2D maps of protein surfaces (stuctuprints)

SYNOPSIS

```
structuprint_frame -prop PROPERTY
                   -dir INPUT_DIRECTORY
                   [-outdir OUTPUT_DIRECTORY]
                   [-custom_db PATH_TO_DATABASE]
                   [-height HEIGHT] [-width WIDTH]
                   [-res PPI_NUMBER]
                   [-point_size SIZE]
                   [-bgcol HEX_COLOR|COLOR_NAME] [-bgalpha ALPHA_VALUE]
                   [-legend_title TITLE]
                   [--no_ID] [--no_legend] [--no_NC]
                   [--help]
                   [--properties]
```

DESCRIPTION

structuprint_frame is a tool for generating two-dimensional maps of protein surfaces. Given an input directory with a PDB file, structuprint_frame will execute an algorithm that involves i) creating a mould of the structure, ii) transforming the mould into a sphere and iii) projecting it on two dimensions using the Miller cylindrical projection.

OPTIONS

-bgalpha ALPHA_VALUE

Set the transparency level of the background color from 0 (fully transparent) to 1 (no transparency). Default value: 1

-bgcol HEX_COLOR|COLOR_NAME

Set the background color either as an HTML hex color or as a color name that R understands. Default: #000000

-custom_db PATH_TO_DATABASE

Provide the path to a custom SQLite database of amino acid properties. For information about the schema, refer to the documentation.

-dir INPUT_DIRECTORY

Specify the directory location of the input PDB file. The directory must have no more than a single PDB file inside it.

-height HEIGHT

Specify the height of the structuprint in mm. If only **-width** is set, then structuprint_frame will automatically adjust the height to the appropriate value. Default values:

76.56 by default or 66 when the **--no_legend** flag is active.

--help

Show the available options and exit.

-legend_title TITLE

Specify the title of the legend. If this is not set, then `structuprint_frame` will use the name of the selected property as the legend title.

--no_ID

Remove the ID number from the structuprint.

--no_legend

Remove the legend from the structuprint.

--no_NC

Do not show the N/C-termini positions in the structuprint.

-outdir OUTPUT_DIRECTORY

Specify the location for `structuprint_frame`'s output files. If this is not set, `structuprint_frame` will write its output in the input directory.

-point_size SIZE

Specify the size of the data points in the structuprint. Default value: 1

-prop PROPERTY

Specify the amino acid property based on which the structuprint will be colored.

--properties

List all the amino acid properties available in the default database along with their explanation, and quit.

-res PPI_NUMBER

Specify the resolution of the structuprint in pixels per inch. Default value: 100

-width WIDTH

Specify the width of the structuprint in mm. If only **-height** is set, then `structuprint_frame` will automatically adjust the width to the appropriate value. Default value: 90

EXAMPLE

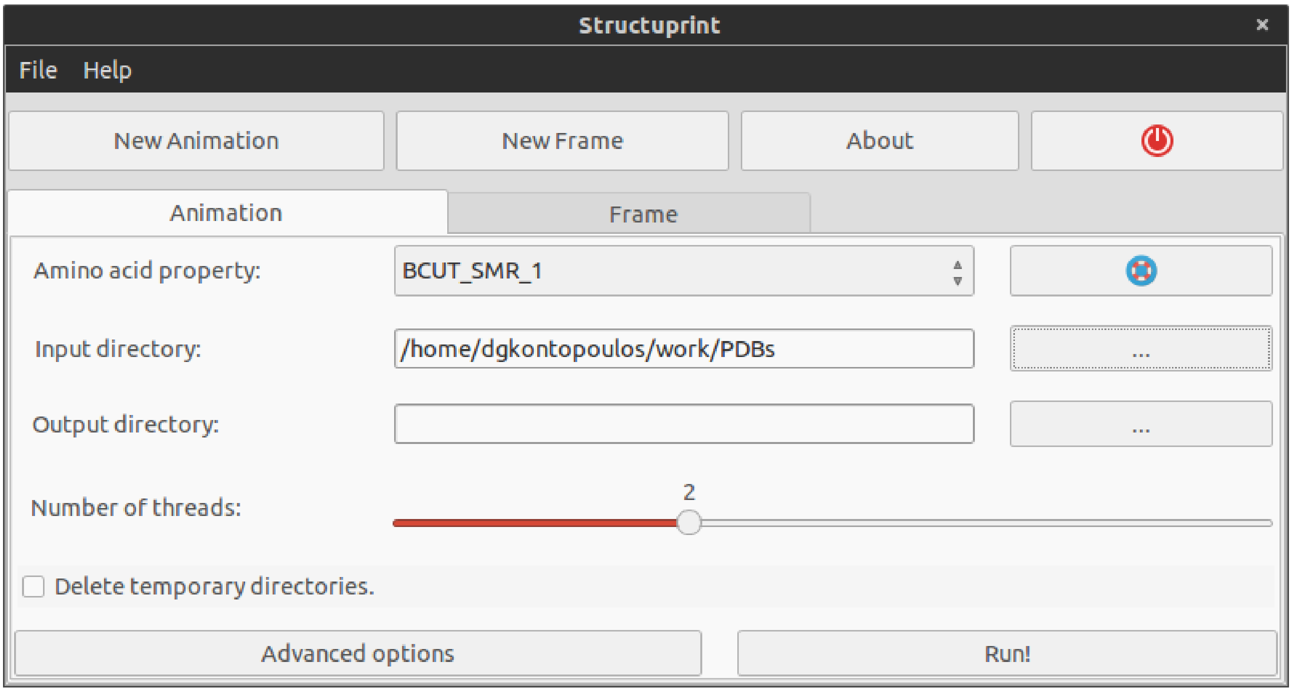
```
structuprint_frame -dir './Data/' -prop FCharge -legend_title 'Charge' -width
250 -res 300 -outdir './Results/'
```

SEE ALSO

Basic R colors - http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

5 Structuprint's GUI

Structuprint also comes with a user-friendly Graphical User Interface for GNU/Linux distributions.



Amino acid property: BCUT_SMR_1

Input directory: /home/dgkontopoulos/work/PDBs

Output directory:

Number of threads: 2

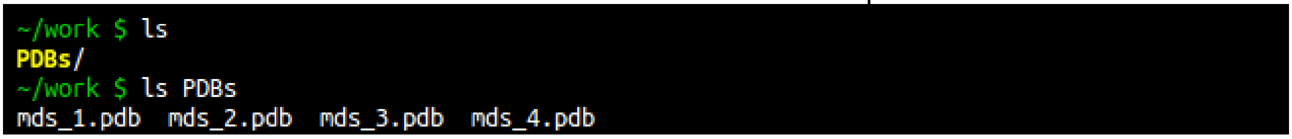
□ Delete temporary directories.

The user can switch between two tabs in order to generate an animation or a single still image. Some basic options are immediately available, whereas all remaining ones can be modified via the 'Advanced options' button.

Clicking 'Run!' launches a temporary terminal in which Structuprint's progress is logged. At the end of the run, the user is notified about the status of the job (success or failure) and is also prompted to view the resulting figure, if successfully generated.

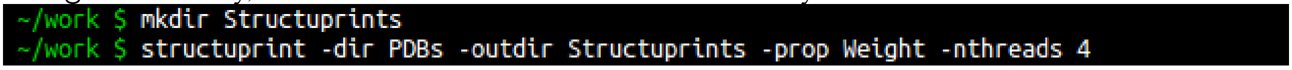
6 Tutorial: visualizing frames from an MD simulation

Here we will visualize 4 PDB frames from a molecular dynamics simulation. We assume that the frames have been superimposed to each other prior to this tutorial. The filenames need to have an underscore and an ID number before the '.pdb' suffix.



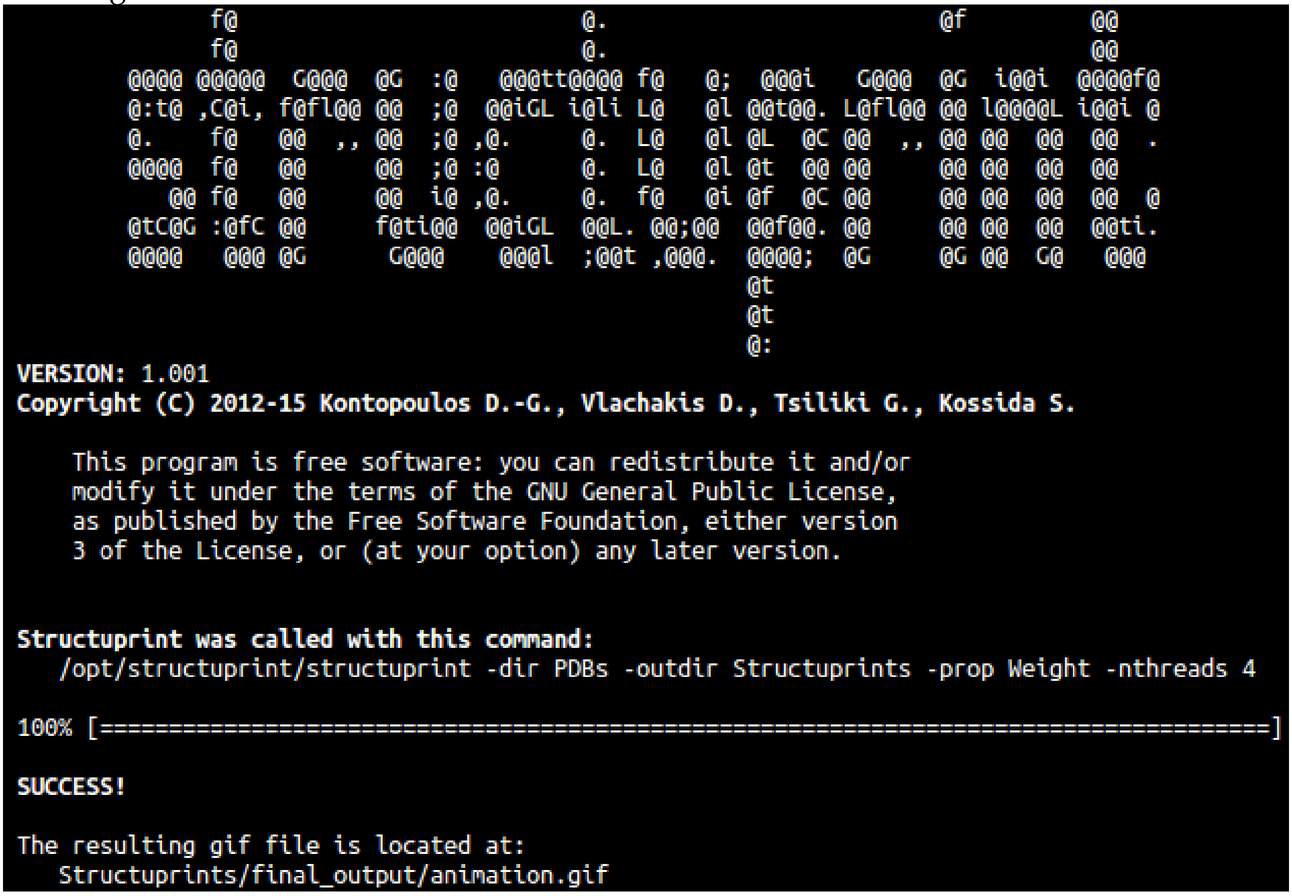
```
~/work $ ls
PDBs/
~/work $ ls PDBs
mds_1.pdb  mds_2.pdb  mds_3.pdb  mds_4.pdb
```

We will then create an empty directory to store Structuprint's output. Structuprint will read (`-dir`) files from the `PDBs` directory and save its results (`-outdir`) to the `Structuprints` directory. For coloring, the property it will use (`-prop`) is the molecular weight one (`Weight`). Finally, it will use 4 CPU cores simultaneously (`-nthreads4`).



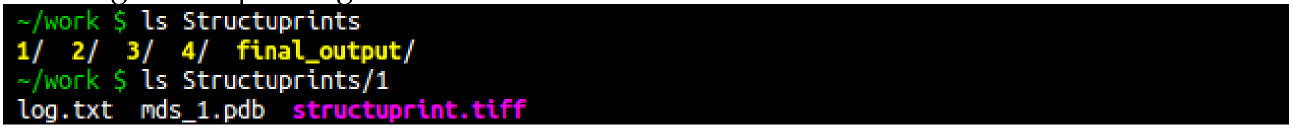
```
~/work $ mkdir Structuprints
~/work $ structuprint -dir PDBs -outdir Structuprints -prop Weight -nthreads 4
```

Running...



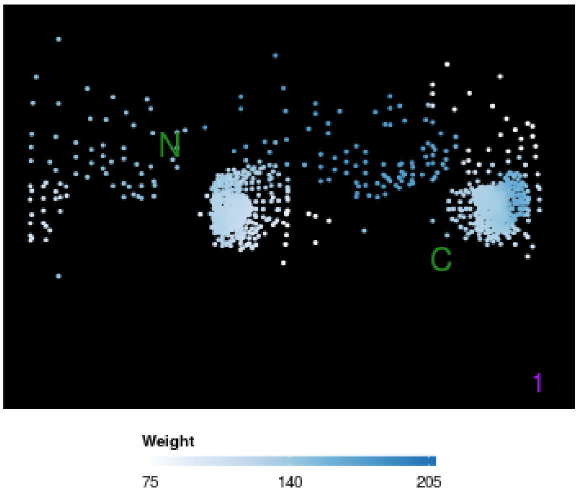
```
              f@                         @.                        @f         @@
              f@                         @.                                   @@
        @@@@ @@@@@  G@@@  @G  :@   @@@tt@@@@ f@   @;  @@@i   G@@@  @G  i@@i  @@@@f@
        @:t@ ,C@i, f@fl@@ @@  ;@  @@iGL i@li L@   @l @@t@@. L@fl@@ @@ l@@@@L i@@i @
        @.    f@   @@  ,, @@  ;@ ,@.     @.  L@   @l @L  @C @@  ,, @@ @@  @@  @@  .
        @@@@  f@   @@     @@  ;@ :@      @.  L@   @l @t  @@ @@     @@ @@  @@  @@
           @@ f@   @@     @@  i@ ,@.     @.  f@   @i @f  @C @@     @@ @@  @@  @@  @
        @tC@G :@fC @@     f@ti@@  @@iGL  @@L. @@;@@  @@f@@. @@     @@ @@  @@  @@ti.
        @@@@   @@@ @G      G@@@    @@@l  ;@@t ,@@@.  @@@@;  @G     @G @@  G@   @@@
                                                     @t
                                                     @t
                                                     @:
VERSION: 1.001
Copyright (C) 2012-15 Kontopoulos D.-G., Vlachakis D., Tsiliki G., Kossida S.

    This program is free software: you can redistribute it and/or
    modify it under the terms of the GNU General Public License,
    as published by the Free Software Foundation, either version
    3 of the License, or (at your option) any later version.


Structuprint was called with this command:
   /opt/structuprint/structuprint -dir PDBs -outdir Structuprints -prop Weight -nthreads 4

100% [===============================================================================================]

SUCCESS!

The resulting gif file is located at:
   Structuprints/final_output/animation.gif
```

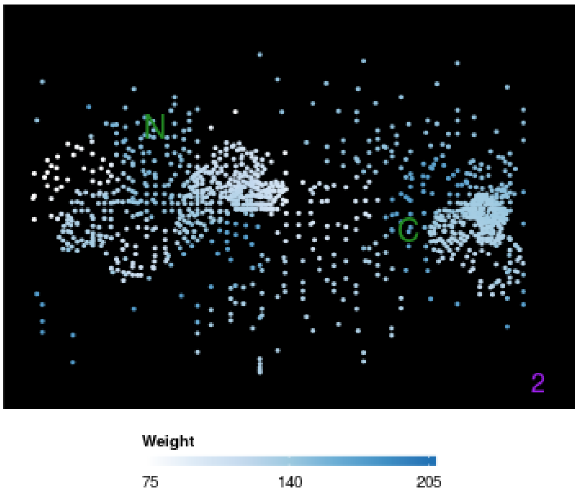
Structuprint created one directory per frame and a `final_output` directory to store the animation. Each numbered directory contains its starting PDB file, a log file and the resulting structuprint figure.

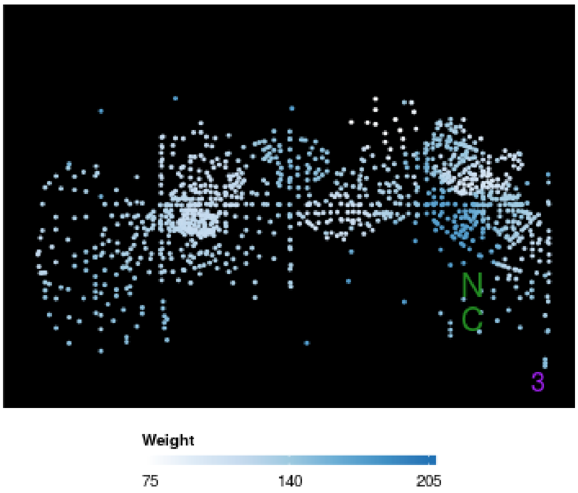


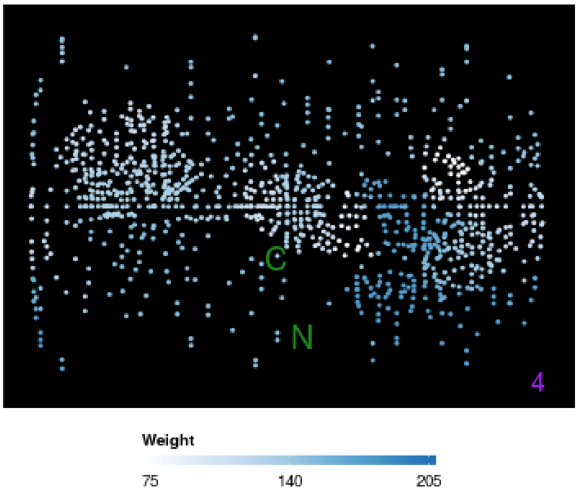
```
~/work $ ls Structuprints
1/  2/  3/  4/  final_output/
~/work $ ls Structuprints/1
log.txt  mds_1.pdb  structuprint.tiff
```

The two-dimensional representations of the PDB structures, as produced by Structuprint, are shown below. A simple comparison of the panels indicates that a stable conformation has yet to be found.



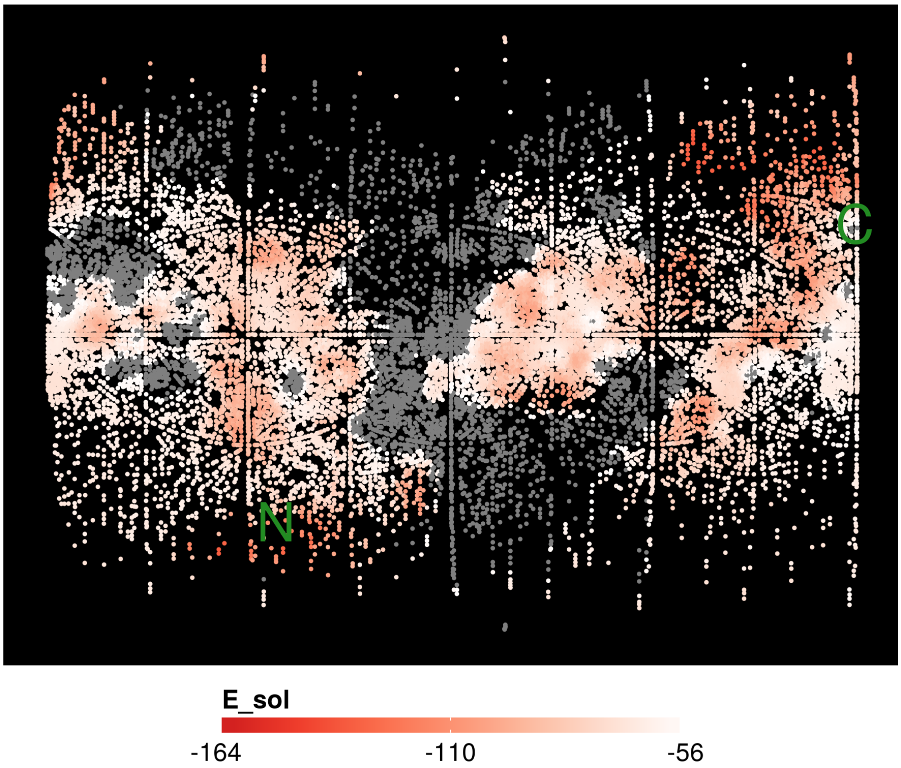






7 Common warnings/errors

- **ERROR! No models could be found in that PDB file! Please try and fix any format errors. For example, is there an "ENDMDL" keyword after each model?** The number of models is obtained by counting the ENDMDL/END occurrences. You can avoid this error by inserting an "END" keyword at the end of the input file.
- **Warning! 'XXXX' is not a color that R can understand. Setting the background color to '#000000'.**
  The `-bgcol` parameter allows for changing the background of structuprint figures from black to another color. Only HTML hex colors or default R colors are supported.
- **Warning! A residue (?) for which there is no recorded value was found: HOH Its value was set to 0; that may lead to wrong results!**
  Structuprint's default database contains values only for the 20 common amino acids. If Structuprint detects the presence of a modified amino acid or any other chemical component, it will set its value to 0, no matter what property is selected. In some rare cases that may not bias the results. For example, if the `FCharge` property is selected, a ligand with a charge of 0 will not affect the final figure, as its value would be set to 0 anyway. In any other situation though, the results may be significantly different. The figure below shows a structuprint of a PDB file with ligands, with a color scale that does not include 0. Grey patches correspond to atoms with a value outside the color scale. To fix this issue, see section 8.1 or 8.2.

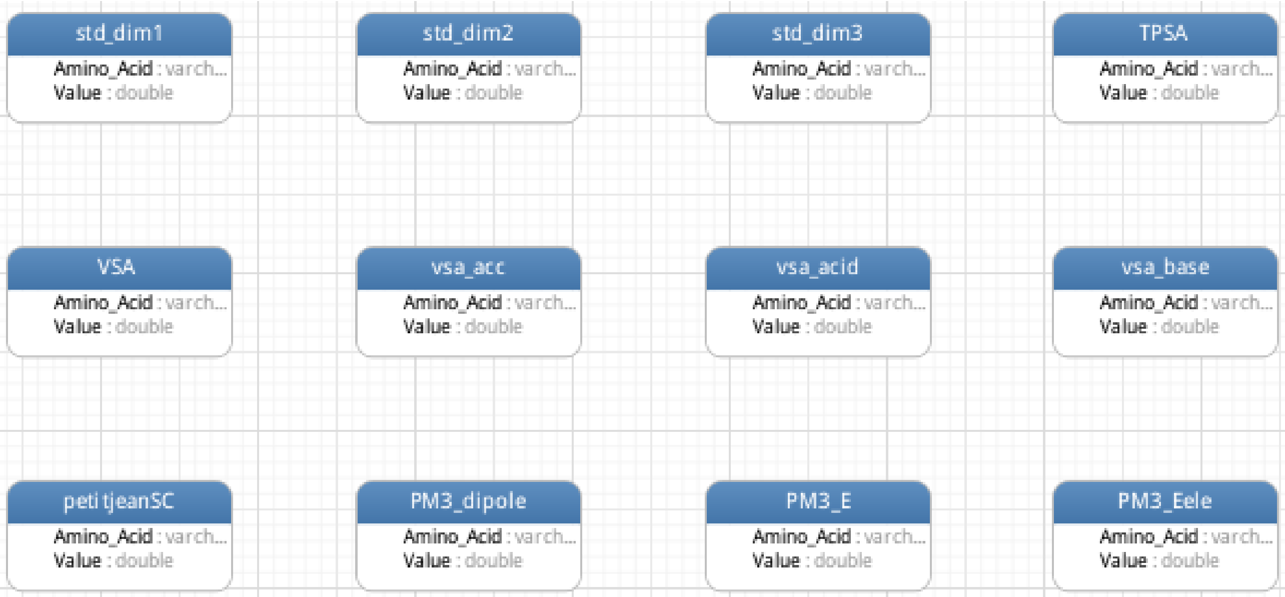


- **Warning! The width to height ratio (excluding the legend) is not close to 1.3639. The map will be distorted!**
  This message appears when both height and width are specified by the user and their ratio is not appropriate. The solution would be to enter just the desired height or width. Structuprint would then calculate the appropriate value for the other dimension.

8 Advanced usage

8.1 Extending the default database

When faced with a PDB file that contains chemical components for which Structuprint does not have measured values, one option is to extend the default SQLite database. On GNU/Linux systems, it is usually located at `/opt/structuprint/amino_acid_properties.db`. Each table in the database corresponds to a distinct property and contains two columns: i) `Amino_Acid` (the 3-letter identifier) and ii) `Value`.



To add the molecular weight of a novel chemical component, for example, you only have to create a new row in the `weight` table and insert its 3-letter code and its weight value.

8.2 Custom database

A more portable solution would be to create a custom database. By doing that, the default database remains in pristine state and new properties or measurements are kept in a separate SQLite database. The schema should be similar to the one of the default database (see above), i.e. one table per property, with columns for the 3-letter code and value. For `structuprint` or `structuprint_frame` to use the custom database, you need the `-custom_db` option. Note, however, that **Structuprint will only read values from one database at a time** and will not combine measurements from both databases!

Resources for properties of chemical components

-PDBeChem:	https://www.ebi.ac.uk/pdbe-srv/pdbechem/
-PubChem:	http://pubchem.ncbi.nlm.nih.gov/
-DrugBank:	http://www.drugbank.ca/
-ChemSpider:	http://www.chemspider.com/

9 Known issues

- **When the dimensions of the structuprint are small, data points may overlap.** This usually leads to some degree of color variation even betwen different runs against the same PDB file, as the data points are drawn in random order. To address that issue, you could reduce the size of the data points by setting `-point_size` to a value that is smaller than 1. Alternatively, increasing the size of the figure will provide enough space for overlapping not to occur.

- **When the ID number is plotted on the structuprint, there may be some small differences in the color scale between frames (see below).** The colors preserve their positions across the scale, but the borders between them may slightly fluctuate. This problem does not occur when the ID number is omitted from the figure (`--no_ID`) and is possibly a shortcoming of the underlying `ggplot2` R code that Structuprint uses.