

```
#!/usr/bin/perl
# map_to_B_v1.pl
# MODIFIED FROM: map_conservation_to_B.pl ; Cameron Mura (07/2001).
# AUTHOR: Ulrich Krauss (07/2011)
# LAST MODIFIED: 22/07/2011
#
# USAGE:
# map_to_B_v1.pl PDB-File.pdb SITES-File.txt
#
# DESCRIPTION:
# This Perl script takes 2 input files: argv[0] ("PDB_File.pdb") is the PDB file
# and argv[1] ("SITES-File.txt") is a file tabulating the values which should
# be mapped to the B-factor field (i.e. conservation scores, NMR chemical shift
# changes, hydrophobicity values etc. ...)
#
# FORMAT of SITES-File:
#
# "res#,ss,value"
#
# e.g.
# 1,ss,0.501
# 2,ss,1.011
# ...
# 140,22,20.00
#
# where "res#" is the residue #, ss is used as a necessary identifier and value can be any
# number. The script will read in the values from the SITES-File and replace the
# B-factor of the corresponding residue with the listed value. For missing
# residues the B-factor field will be filled with zeros.
#
# The output is a PDB file (named mapped.pdb) with all of the B-factors
# flattened to 0.00, except for those residues for which a value has been
# provided in the SITES-file.
#
# NOTE1: make sure all occupancies are 1.00 for any atoms with B-factors you
# want changed.
#
# NOTE2: please be aware that all heteroatoms will be removed from the output
# pdb-file.
#
# NOTE3: output tested with pymol v0.99
#
# NOTE4: Input SITES-File must be located in the same directory as the script

$pdb_in = $ARGV[0];
$site_file = $ARGV[1];
$value="23.00";
$bfac = "";
@cons = "";
$site_line = "";

open (MYFILE, '>mapped.pdb');
open (PDB, $pdb_in) || die "Cannot open file \"$pdb_in\"\n";

while (<PDB>
```

```
{
$line = $_;
chomp ($line);
$resnum = ""; $bfac = "";
if ($line =~ /^(ATOM|HET)\s+\d+\s+[\w\*]+\s+\w\w\w
[A-Z]\s+(\d+)(\s+)(1\.00\s+)(\d+\.\d+)(\s+)(.*)/
or $line =~ /^(ATOM|HET)\s+\d+\s+[\w\*]+\s+\w\w\w
[A-Z]\s+(\d+)(\s+)(1\.00)(\d+\.\d+)(\s+)(.*)/)
{
$resnum = $3; $bfac = "0.00";
$counter = 0;
open (SITES, $site_file) || die "Cannot open file \"$site_file\"\n";
while (<SITES>)
{if ($counter == 0)
{
$site_line = $_; chomp ($site_line); @cons = ""; $abs = ""; chomp ($abs);
@cons = split(/,/, $site_line);
if ($cons[0] == $resnum and $cons[1] eq "ss") {$bfac = "@cons[2]" ; $resnum
++; last;}
next;
}
}
close (SITES);
if ($bfac eq "") { $bfac = "0.00";}
$shit = $5 . " " . $bfac;
print "$1$3$4$shit$7\n";
print MYFILE "$1$3$4$shit$7\n";
}
}
close (MYFILE);
exit;
```