FLEXAMINO:
B-FACTOR PREDICTOR APPROACH FOR THE ASSESSMENT OF PROTEIN FLEXIBILITY

MSc Bioinformatics (UPF)

Structural Bioinformatics (SBI) & Python (PYT)

Eric Toro Delgado

Marina Vallejo Vallés

Sara Vega Abellaneda

CONTENTS

1. INTRODUCTION

1.1. FLEXIBILITY

Proteins are not static biopolymers, they possess conformational flexibility. It is an essential feature, without flexibility many proteins wouldn't be able to perform their biological functions (Halle, 2002).

Flexibility can be also defined as structural mobility, it is the opposite of rigidity. Both concepts are derived from mechanical engineering. Some authors state that they are not properly describing the reality, as they describe static properties and don't take into account the relative motion that is possible in certain regions of a protein (Karshikoff *et al.*, 2015).

In order to define the flexibility of a system, we must take into account several things such as the number, geometry and stability of conformers, the energy barrier among them, some kinetic parameters and the thermal motion of the atoms (Huber, 1979).

Structural flexibility is an essential attribute, without which few proteins could carry out their biological functions. For example, regulatory proteins take advantage of this flexibility and adopt certain conformations such as bound and unbound to bind their partners (Pabon and Camacho, 2017).

1.2. APPLICATIONS

The inherent flexibility of proteins allows them to function through molecular interactions with their environments, be it inside cells, among cells or even between organisms. Flexibility is highly relevant for protein function, since many of these functions depend on the ability of proteins to change their conformation, and it has been observed that highly dynamic sites are often involved in the interaction (Teilum *et al.*, 2009).

An example of this is proteolysis, where it has been observed that cleavage does not necessarily take place with the presence of sites for a specific protease, but instead adaptation of the protein to be cleaved to the active site of the protease is necessary for the cleavage reaction to take place (Bothner *et al.*, 1998; Fontana *et al.*, 2002; Falconi *et al.*, 2002).

Both low-amplitude and large-amplitude motions are relevant for protein function. For example, side-chain conformations that are predicted to be unfavorable when considering a receptor in its unbound state may actually be observed experimentally, being favored by the interaction with the ligand, and MD simulations can show fast conformational sampling of side chain conformations (Xu *et al.*, 2008). Proteins that are released in an inactive state can also experiment with large-scale movements of more than 25Å when changing to the active state.

1.3. APPROACHES

There are different approaches to assess protein flexibility. The main ones are:

- **Crystallographic B-factors**: In X-ray crystallography experiments, a quantity called temperature factor or beta-factor can be computed for each atom in the structure, which is related to the displacement of the atoms from their mean positions (Sherwood and Cooper, 2011). B-factors are computed with the following formula:

  $$B = 8\,\pi^2 < u^2 >$$

  $$Units\text{: } \text{Å}^2$$

  Where $<u^2>$ is the mean squared displacement, that is, the mean of the square of distances between the positions of the atoms at a given time and their mean position.

  In general, we can assume that the higher the B-factor value, the higher the flexibility of a residue. B-factors are a measure of the uncertainty in position of the atoms in a given structure, and they reflect the mobility of the backbone.

  However, it must be noted that B-factors are affected by other features of the crystalline state, such as packing, resolution of the structure, the refinement procedure, contacts in the crystal or the structural environment. It also reflects thermal vibrations (Ringe and Petsko, 1986), with temperature estimated to be the main contributor to B-factors (Hartmann *et al.*, 1982; Bryn Fenwick *et al.*, 2014).

  As a consequence, B-factors are not directly comparable across different structures, and some normalization approach is required to compute a standardized flexibility score from them.

- **NMR-based**: Nuclear Magnetic Resonance (NMR) can be used in order to calculate the coordinate variance between different models of a given protein. Experimental NMR data should be an ensemble of fluctuating conformation, and not only a rigid conformation (Sunada *et al.*, 1998). This approach requires the superimposition of ensembles. The superimposition can be biased by the inclusion of poorly converged coordinates, and therefore reducing their applicability (Reinknecht *et al.*, 2021). A limitation of this approach is that sometimes the structural coordinates are not explicitly included, limiting the understanding of the structural dynamics. In order to address this problem, NMR is usually combined with molecular dynamics (Salmon *et al.*, 2011)

- **Normal Mode Analysis (NMA)**: In silico technique that is used to describe the possible flexible states that are accessible for a certain protein in an equilibrium position (Bauer *et al.*, 2019). It is based on small oscillation physics, the normal modes of vibration are oscillations that characterize the system dynamics around the energy minimum. With a linear combination of normal modes the behavior of the system can be studied.

From the previously explained methods, **FlexAmino** will be based on **crystallographic B-factors**. We have chosen this approach because the program is based on the use of available information from homologous proteins, and this parameter is found on many of the existing PDB files. We are aware that it has important limitations (exposed later on), but the idea of the project is to build a program to make a preliminary assessment of the flexibility, which can be compared with other methods such as PredyFlexy.

If we wanted to make an advanced application for assessment of protein flexibility, **crystallographic B-factors** wouldn't be suitable as they don't explain all the possible flexibility of a structure.

2. FLEXAMINO WORKFLOW

FlexAmino is a tool to predict flexibility based on the search of homologs for a given protein sequence. Crystallographic B-factors are extracted from the homologs that have an available known structure and with them, a beta-factor profile is generated and assigned to the query sequence using a structural alignment.

FlexAmino's main execution program is in the python script **flexamino.py**. Within this script we import the modules containing the different functions needed for the program to run. The following modules are the ones created during this project:

- work_with_input_sequence.py
- pdb_functions.py
- profiles_bfactor.py
- results_to_file.py

Further information about the content of these modules can be found on their respective documentation.

Note that the code in the main program is written under the line *"if __name__ == "__main__"*, as we want it to be executed when invoked directly.

The first step of the main program is to obtain the command line arguments given by the user. Among them, the user must provide the path to an existent FASTA file. This file must contain the query sequence, which is the sequence of the protein which is wanted to assess its flexibility, in FASTA format with a UNIPROT header. The UNIPROT header is crucial, as it allows the program to obtain the UNIPROT identifier from the query sequence provided, which will be later used during the obtention of the alpha fold model.

Then it runs BLAST remotely against the PDB database to find homologous sequences for which empirically solved structures exist. Homologous sequences are very likely to share structure with the query protein. We assume that similar structures will share similar flexibility behavior, so we will use them as templates to infer the flexibility scores of our protein. Later on we will extract the crystallographic B-factors from these structures and generate a profile.

The BLAST hits are stored in an object and parsed to obtain their PDB codes and chain IDs. Then this information is used to download the PDB files from the database. As B-factors are only obtained in X-ray crystallography, only structures solved with this method are useful for the computation. Thus, first the script checks that the downloaded PDB is an X-ray structure. If the condition is met, it saves a pdb file with only the relevant chain and stores its path. The command-line argument “pdb_limit” determines the maximum number of X-ray structures that will be used for the remaining steps.

Once the templates are obtained, they need to be superimposed to the query (with structural superimposition) to establish the equivalences among their residues. But prior to this, some additional steps are required.

First, the structure of the query protein is needed. As mentioned before, it is downloaded from the AlphaFold database using the UNIPROT code of the protein. This is why our program only accepts input sequences with UNIPROT header.

Second, we need to provide a Multiple Sequence Alignment. For this purpose, the program first creates a multifasta file with the templates and the query sequences and then it runs ClustalW on it. ClustalW is an external application that needs to be installed by the user in order to use Flexamino, as explained later in the installation part of the tutorial.

Now we have all the necessary elements to perform the structural alignments. For each template, the program performs a pairwise structural alignment of the query structure against the template and assigns the template B-factors of the C-alpha of each residue to the corresponding residue of the query. As mentioned before, B-factors are highly affected by experimental conditions, so the templates’ B-factors are previously normalized to make them comparable across structures. This is done by subtracting the mean and dividing by the standard deviation of all the C-alpha B-factors of that structure.

Finally, for each residue in our query, a flexibility score is computed as the mean of all the templates’ B-factors that have been assigned to that residue. We also compute the standard deviation as a measure of the uncertainty in the assigned score.

If a window size has been provided with the “WINSIZE” flag, the program also computes a set of “smoothed” scores. For a given residue R and window size W, the set of W residues centered on R is selected, the average of their scores is computed, and this value is assigned as the “smoothed” score of R. For example, if window size is 11, residue 6 smoothed score corresponds to the mean of scores from residues 1 to 11. Note that, for a given window size W, **residues at the beginning and end of the sequence** that do not have W/2 residues before or after them **will not be assigned a smoothed score**.

After all calculations have finished, the program provides two outputs: 1) a text file in tabular format and 2) a plot of the flexibility profile. The names of the output files are generated with a prefix users must provide with the “OUTFILE” argument.

The first line of the text file is a header consisting of a ‘>’ followed by the UniProt identifier of the query sequence. The following lines consist in a set of columns, containing: 1) the residue number, 2) the residue type in the one letter aminoacid alphabet, 3) the flexibility score (unsmoothed), 4) the standard deviation and 5) the smoothed flexibility score (only if a window size different than 1 is applied). The plot contains the residue positions on the X axis, and the corresponding flexibility scores (smoothed if requested, unsmoothed otherwise) on the Y axis. **See the Tutorial section for examples of both files**.

During its execution, FlexAmino also creates a temporary directory where it will store files generated or downloaded during the program execution. Inside this directory the user will find:

- The variable putative_homologs which contains the results from BLAST and can be later used if the user executes the recovery option.
- All the PDB files downloaded and used by the program.
- The multifasta file containing all the sequences that were used in the alignment.
- The Multiple Sequence Alignment generated by ClustalW.
- The Alpha Fold model of the query sequence.

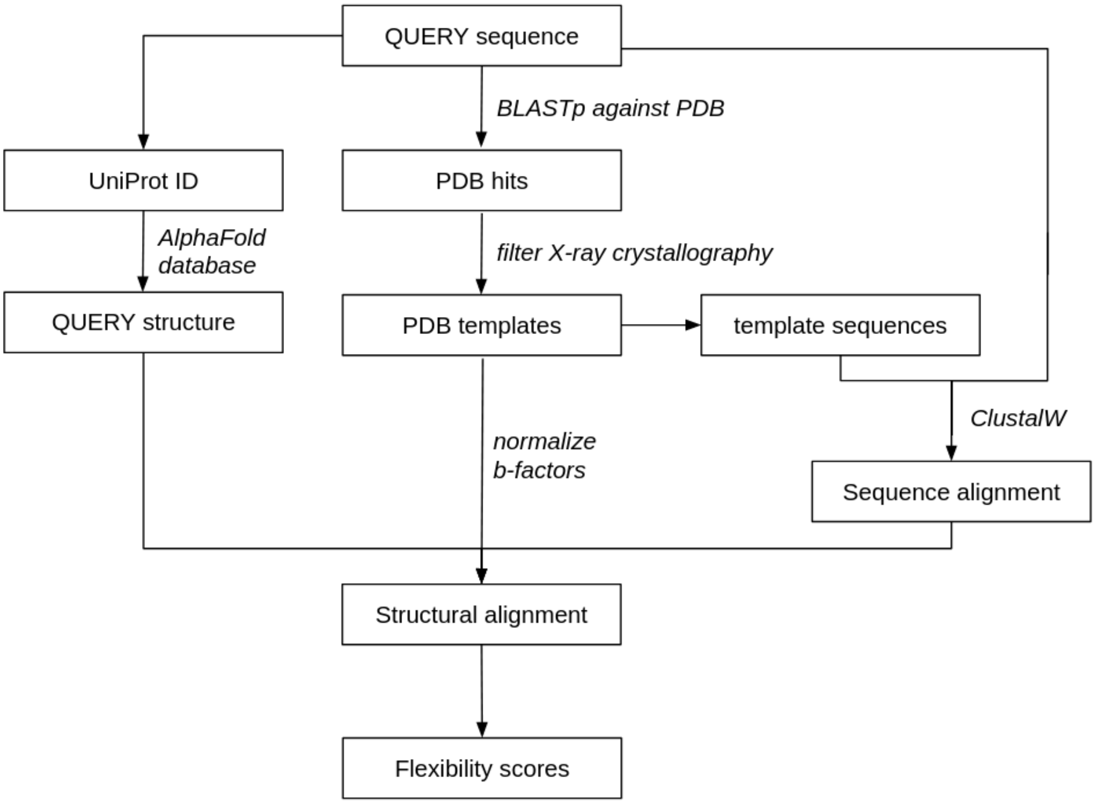

Figure 1. FlexAmino workflow.

3. TUTORIAL

3.1. INSTALLATION

FlexAmino is available in a Github public repository. In order to install the program, the user must clone the repository to the local machine with the command:

```
git clone https://github.com/upcmarina/flexamino.git
```

Once the repository is cloned, it is necessary to install some dependencies. There are two ways of doing it, install the dependencies one by one or install all of them together in an automated way. For the second option, type the following commands (having root permissions):

```
apt-get install python3-setuptools

python3 setup.py install
```

If the installation is successful the following message will appear in the terminal: *Finished processing dependencies for FlexAmino* == *1.0.*

FlexAmino also requires ClustalW to be installed. This can be done using the built-in APT package manager. Then the user only needs to type the following command (having root privileges):

```
apt-get install -y clustalw
```

Now it is possible to run *flexamino.py* from any directory in the local machine.

3.2. RUNNING THE PROGRAM

There are different running options for FlexAmino, depending on the needs of the user.

For first time users, we recommend the **Basic Execution**. With this mode the program is executed from the very beginning to the end, providing the results in a parseable text output and a figure.

But the user may encounter different scenarios where the Basic Execution is not suitable, this is why we provide other options such as recovering BLAST results, among others.

Lastly, please remember that the input files must be in **FASTA format with UniProt headers** for the program to work properly.

- Basic execution

To run the program, simply execute the flexamino.py file using the Python3 interpreter and indicate the necessary options. Only the **-i** flag, to indicate the input file, and the **-o** flag, to indicate the prefix of the output files, are mandatory.

For example, we can run the program for the protein sequence Q8IU85. In this case we will also add the **-t** flag, which allows us to keep the temporary folder generated during the calculation, and the **-v** flag, for verbose output:

```
flexamino.py -i Q8IU85.fasta -o Q8IU85 -t -v
```

As previously mentioned, the input is in **FASTA format with UniProt header**. The output name will be used to create the output file names of the text and image files, so **no extension should be provided**.

This will generate two output files:

1 One text file, named Q8IU85.txt, which is a tab-separated file with one line per aminoacid residue:

```
>Q8IU85
1    M    nan    nan
2    A    nan    nan
3    R    nan    nan
4    E    nan    nan
5    N    nan    nan
6    G    nan    nan
```

```
7    E    nan    nan
8    S    nan    nan
9    S    nan    nan
10   S    nan    nan
11   S    1.597467314706615    0.414467024249193
12   W    1.6284330752258882    0.8044023628742437
13   K    1.56350988287936    1.1103471131062173
14   K    1.397036151514169    1.01827724194396
15   Q    1.1797852689170298    0.9775657612979843
...
```

The first line of the file is a header with the UniProt identifier of the sequence; the remaining are the data columns. The content of the columns is as follows:

- Column 1: residue number

- Column 2: residue name

- Column 3: flexibility score (mean normalized beta-factor)

- Column 4: standard deviation of the flexibility score

We can see that some of the columns have ‘nan’ values on them. This corresponds to those residues that were not aligned with any template residue in the structural alignment, for which no flexibility score could be assigned.

2 An image file, named Q8IU85.png, containing the prediction scores on the Y axis and the residue numbers on the X axis (Figure 2):

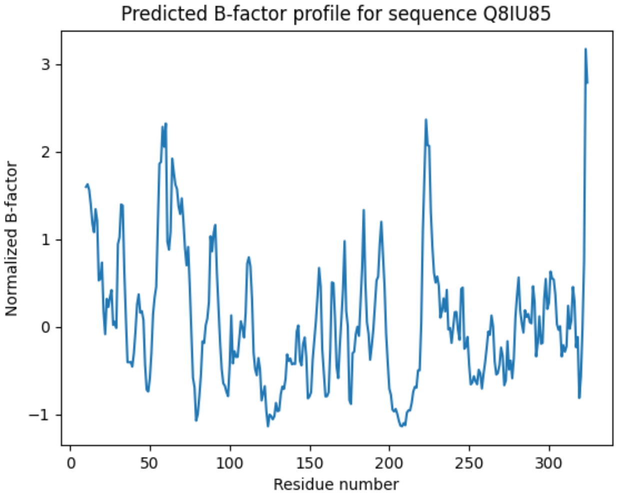

Figure 2. Example of flexibility profile for protein Q8IU85.

Since we used the **-t** option, we can also inspect the contents of the *tmp* folder. Inside it we can find:

- A file named putative_homologs.dat, which contains a python object with the results of the BLAST search in binary format as a byte stream.
- A series of PDB files, named by their PDB identifiers, corresponding to the crystallography structures of the top n BLAST matches.
- A series of PDB files, named by their identifier and a suffix corresponding to the chain ID, which contain the specific chains that are the actual homologs to the query sequence (by default, a maximum of 10).
- A PDB file named queryName_alphaFold.pdb, with the AlphaFold2 model of the query sequence
- A FASTA file, seqs.fasta, with the sequences of the matches and the query.
- The files seqs.dnd and seqs.aln, which are the output of ClustalW. seqs.aln contains the alignment of the query and homologous sequences in clustal format.

- Recovering a calculation

The BLAST search performed by the program can sometimes take a relatively long time depending on the traffic of the BLAST server. In addition, users may wish to repeat the calculation and plotting for different window sizes or using a different number of reference structures (see following sections for these options).

For these reasons, FlexAmino provides an option to recover a calculation that did not finish without the need to repeat the BLAST search again. Keep in mind that **this will only work if the BLAST search was completed** and the program was stopped afterwards; also, the **“tmp” folder should not be removed**.

To resume a calculation after the BLAST search, users can use the -r flag. To keep the BLAST results and intermediate PDB files generated during the computations, the -t flag should be used. For example, we can compute the scores again for Q8IU85 recovering the results of the previous BLAST:

```
flexamino.py -i Q8IU85.fasta -o Q8IU85 -r -t -v
```

We can see that the program now runs much faster, since the BLAST search is skipped.

- Applying a sliding window

By default, the direct score of each residue is used, but a sliding window of a specific size can be specified to smooth the scores. This is done with the **-w** flag.

For example, we can repeat the same calculation using a sliding window of 20 residues:

```
flexamino.py -i Q8IU85.fasta -o Q8IU85_smoothed -w 20 -r -v -t
```

We obtain again two output files:

1 The text output Q8IU85_smoothed.txt:

```
>Q8IU85
1     M     nan     nan     nan
2     A     nan     nan     nan
3     R     nan     nan     nan
4     E     nan     nan     nan
5     N     nan     nan     nan
6     G     nan     nan     nan
7     E     nan     nan     nan
8     S     nan     nan     nan
9     S     nan     nan     nan
10    S     nan     nan     nan
11    S   1.597467314706615    0.414467024249193     1.1655241061012234
12    W   1.6284330752258882   0.8044023628742437    1.0835488673500737
13    K   1.56350988287936     1.1103471131062173    0.9938107781262697
14    K   1.397036151514169     1.01827724194396    0.9457394504197015
15    Q   1.1797852689170298    0.9775657612979843   0.897789071922778
```

In this case we can see there is an additional column, which corresponds to the smoothed flexibility scores. The previous columns are the same as before, that is:

- Column 1: residue number
- Column 2: residue name
- Column 3: flexibility score (mean normalized beta-factor)
- Column 4: standard deviation of the flexibility score
- Column 5: smoothed flexibility score

2 The plot file Q8IU85_smoothed.png (Figure 3):

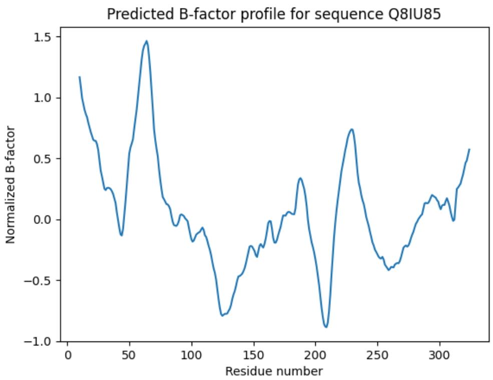

Figure 3. Example of flexibility profile for protein Q8IU85, smoothed with a window size of 20 residues.

As can be noticed, the plot now looks smoother, as the plotted values are those of the smoothed scores (i.e. correspond to the fifth column of the output file).

- Changing the number of templates

By default, FlexAmino uses the top 10 X-ray crystallography structures obtained from the BLAST search to assign the predicted flexibility scores. This default value offers a good trade-off between the computing time and the accuracy of the prediction (see section 4.3 for an assessment of the accuracy). However, users may choose to increase or decrease this limit to speed up the program or to improve the accuracy.

This can be done with the **-p** flag. For example, we can calculate again the scores for protein Q8IU85, but using 20 homologs instead of 10:

```
flexamino.py -i Q8IU85.fasta -o Q8IU85_large_smooth -r -v -p 20 -t -w 20
```

- List of available options

Below is a summary of all the program flags options:

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

-i, --input         Mandatory argument. Specify the input FASTA file with
                    the protein sequence to analyze.

-o, --output        Mandatory argument. Prefix to generate the different
                    output files.

-v, --verbose       Print the progression of the program execution
                    to the terminal (Standard Error).

-t, --tmp           Keep the temporary files directory when the
                          program finishes.

-r, --rescue        Recover a computation from a BLAST result to
                          avoid  running BLAST again.

-p, --pdb_cutoff          Set a maximum number of pdb structures to use
                          for the computation. Default value of 10.

-w, --winsize             Set a sliding window for smoothing the
                          results.Default value of 1 (i.e. no smoothing)
```

4. RESULTS AND DISCUSSION

For the current section, we are going to present the results obtained for two different proteins selected from a pool of proteins proposed by the professors (P38401 and P65206). And also we present the results of two proteins of our choice (Q8IU85 and P0DP23).

4.1. RESULTS WITH THE PROTEINS SELECTED FROM THE SUGGESTED

From the set of proteins given by the professors, we selected the proteins **P38401** and **P65206** to analyze with FlexAmino. These proteins were chosen at random.
Unless specified otherwise, in all cases we used the top 10 BLAST matches to assess the flexibility.

4.1.1. P38401

Protein P38401 is the alpha-1 subunit of the guanine nucleotide-binding protein (G protein) from Guinea pig, which functions as a transducer in multiple signaling pathways (UniProt). This subunit is the one containing the guanine binding site which, inferred by similarity, seems to involve residues 43-48, 150-151, 175-178, 200-204 and 269-272.

Although with no window size applied the flexibility profile is too noisy to detect specific regions of higher or lower flexibility, the predicted flexibility scores correlate with the solvent-accessible surface area (SAS) computed from the AlphaFold model using Chimera (Pearson r=0.48, p=$7.9 \cdot 10^{-22}$; Figure 4), which is in agreement with previous results indicating a relationship between B-factors and SAS (Craveur *et al.*, 2015). Residues at the core of the protein tend to have the lowest displacement, followed by those at interfaces and lastly those fully exposed to the surface. Although the AlphaFold model may not be completely accurate, it is similar to other resolved structures of G-protein alpha-1 subunits (e.g. 5KDO), so this correlation is likely to hold for the real structure.

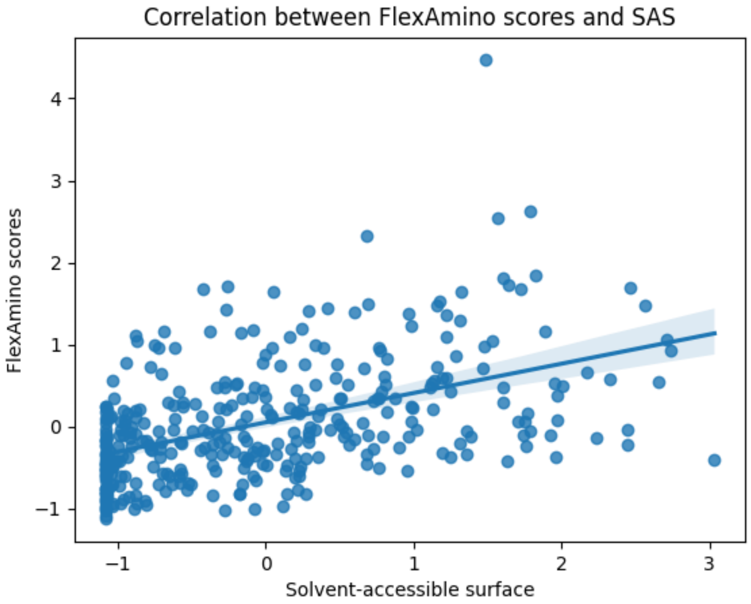

Figure 4. Correlation between FlexAmino flexibility scores and solvent-accessible surface area computed from the AlphaFold model using Chimera.

Based on studies with the bovine homolog, this protein should consist of two domains, GαsRas and GαsAH. While the former contains most of the catalytic residues, the latter (approximately residues 60-180) undergoes substantial conformational changes upon binding and release of GTP (Chung *et al.*, 2011; Rasmussen *et al.*, 2011). In this case, the flexibility scores do tend to be higher around the center of this region, most evidently when applying a window size of 50 residues (Figure 5).

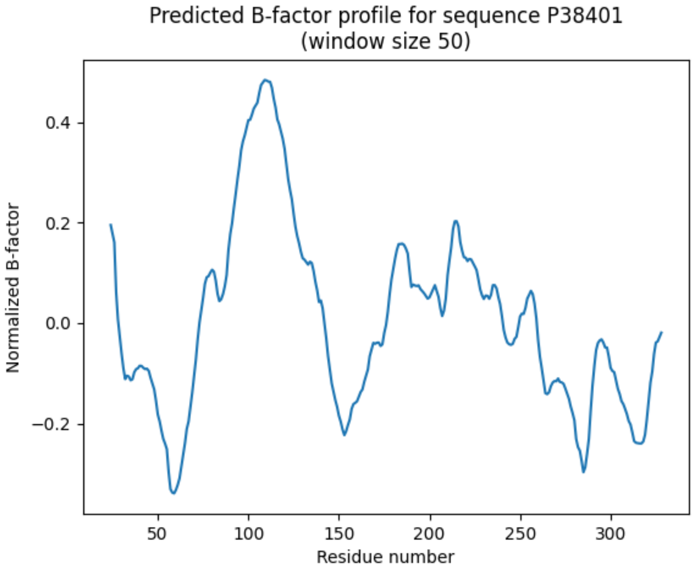

Figure 5. Flexibility profile for protein P38401 smoothed with a window size of 50 residues. The peak centered around residue 100 corresponds approximately to the domain that undergoes most conformation changes.

4.1.2. P65206

Protein P65206 is the mscB arginine kinase from *Staphylococcus aureus*. It catalyzes the specific phosphorylation of arginine residues of proteins (UniProt). Based on studies of the *Bacillus subtilis* mscB homolog, this protein inhibits the binding to DNA of CtsR, a transcriptional repressor of genes related to the heat shock response (Fuhrmann *et al.*, 2009).

The mscB of *Geobacillus stearothermophilus* has been shown to have a loop acting as a "lid" over the active site, which changes conformation upon binding of AMP (Suskiewicz *et al.*, 2019). This structure is also found in remote homologs of arthropods and other invertebrates, and B-factor profiles for the horseshoe crab *Limulus polyphemus* show a peak in the region corresponding to this "lid" loop in the unbound state, in which it is disordered (Yousef *et al.*, 2003).

Using the default value of 10 putative homolog PDBs, the flexibility score profile does not match with what is known about the movement of this "lid", as the flexibility scores are not particularly high in this region (around residue 200), and instead there is a peak around residue 284 (Figure 7).

However, closer inspection reveals that, while all top ten BLAST hits have low e-values, the top 2 hits are clearly much more significant (e-values of 10^{-90} to 10^{-86}) than the others (order of 10^{-24} to 10^{-25}). These two hits correspond to bacteria (as the query sequence), whereas the others are invertebrates; in addition, the bacterial PDBs match the AlphaFold-predicted structure much more closely than the invertebrate ones (Figure 6). When running the program again taking only the top 2 matches, the flexibility score profile does show a peak in the region of the "lid", in agreement with what has been found in the homologs (Figure 7).

As in the previous case, we also observe a significant correlation with SAS ($r=0.44$, $p=1.12\cdot10^{-17}$).

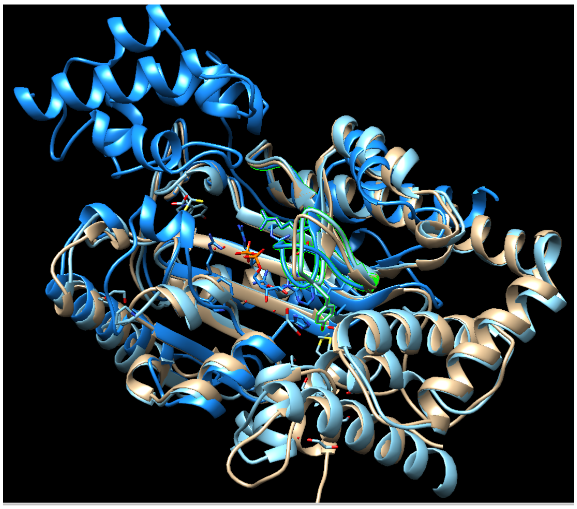

Figure 6. Superimposition of the AlphaFold model for P65206 (golden), the bacterial (*G. stearothermophilus*) homolog 6FH1 (cyan) and the invertebrate (*L. polyphemus*) homolog 1RL9 (blue). The region highlighted in green corresponds to the region of the “lid” loop.

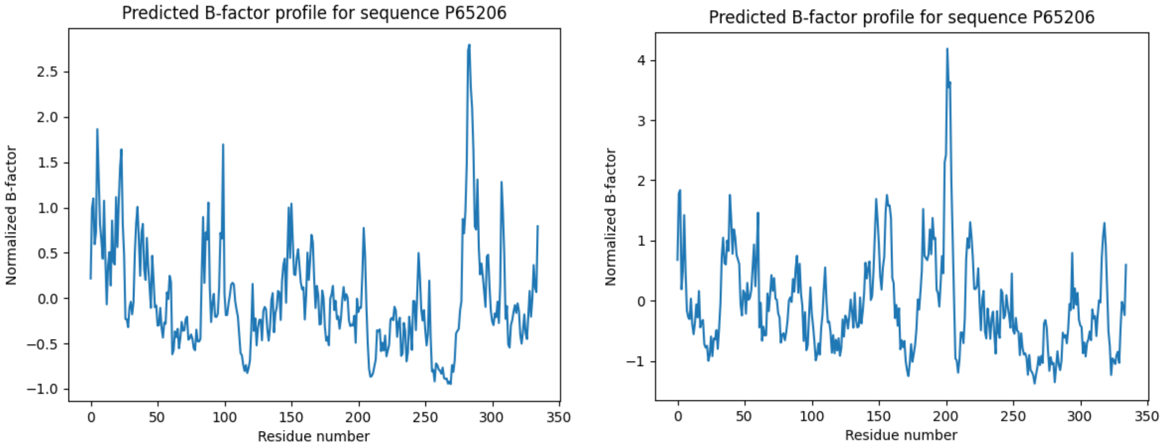

Figure 7. Comparison of the flexibility score profiles using the default of ten templates (left) and only two templates (right). Due to the relatively low similarity with the more distantly related homologs, the results with less templates are better in this case.

4.2. RESULTS WITH THE PROTEINS OF OUR CHOICE

As additional proteins, we selected the human calcium/calmodulin-dependent protein kinase type 1D (**Q8IU85**) and the human calmodulin-1 (**P0DP23**). We wanted to explore the flexibility of a calmodulin protein, as we worked with them during the subject and were interested in their flexibility. For the kinase protein, we chose it as it appears to be calmodulin dependent and we were also interested in it.

4.2.1. Q8IU85

Protein Q8IU85 is a protein kinase known to be involved in different signaling cascades related to the respiratory burst (an increase in production of reactive oxygen species) in granulocytes and neutrophils, among other processes (UniProt).

The C-terminal region of calmodulin-dependent protein kinases is involved in autoinhibition of the protein as well as in calmodulin binding. This region covers the active site in the unbound state, but calmodulin binding displaces this region, allowing ATP to enter the active site (Goldberg *et al.*, 1996; Soderling and Stull, 2001; Hoffman *et al.*, 2011).

Therefore, we would expect this region (approximately from residue 280 until the end) to have high flexibility scores. However, although it is in the upper half of the scores in the profile, it does not stand out as particularly more flexible than other regions (Figure 8).

Again, there is correlation between the flexibility scores and the SAS (r=0.63, p=$7.13 \cdot 10^{-32}$).

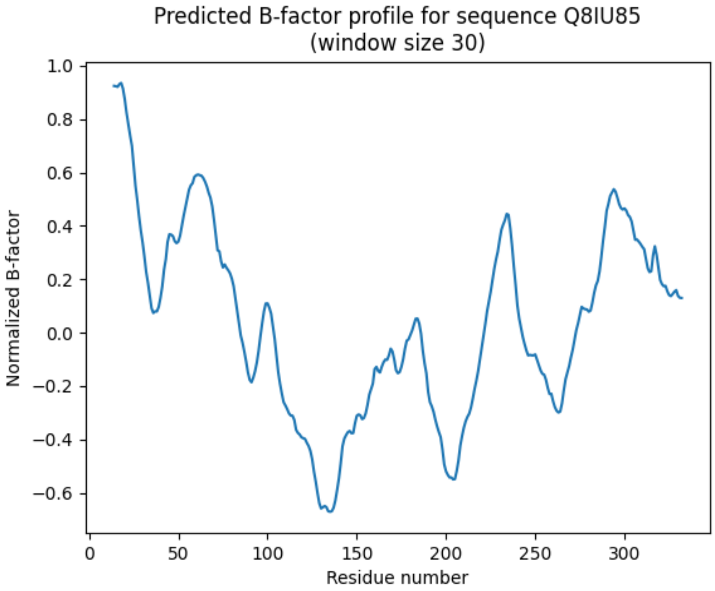

Figure 8. Flexibility profile for protein Q8IU85 smoothed with a window size of 30 residues. Residues from 280 until the end do not stand out as particularly flexible.

4.2.2. P0DP23

Protein P0DP23, the human calmodulin-1, is a protein that mediates the control of a large number of other soluble and membrane proteins via calcium binding. Thus, it is involved in many pathways regulated by Ca^{2+} concentrations.

Calmodulin-1 is known to be formed by two domains united by a region which is a straight alpha-helix in crystal structures, but is known to be flexible in solution, and also bends when interacting with other proteins (Komeiji *et al.*, 2002).

This protein has been extensively studied and many crystallographic structures are available for this specific sequence; therefore we discarded these structures when running the program to avoid overestimating the performance of the method relative to proteins without an empirically resolved structure.

When applying a window size of 20 residues, the region of the center helix that acts as the calmodulin's "hinge" is in a peak of flexibility (around residue 80), but it does not appear as particularly higher than other peaks present in the profile (Figure 9).

One more time, we observe a correlation between the flexibility scores and SAS (Pearon's r = 0.56, p = $1.12 \cdot 10^{-13}$), consistent with observations from empirical B-factors.

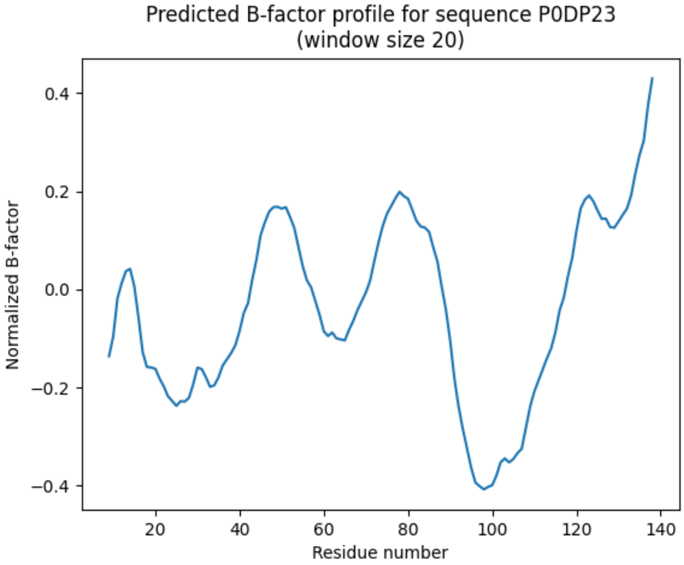

Figure 9. Flexibility profile for protein P0DP23 smoothed with a window size of 20 residues. There is a peak around residue 80, but is it not particularly higher than that around residue 50.

4.3. ASSESSMENT

In order to evaluate the performance of FlexAmino, we explored two different approaches.

First, we checked the overall performance of the program. We did it in two separated parts:

1 Using an external tool, PredyFlexy (De Brevern *et al.*, 2012), a web server that predicts flexibility and local structure from a sequence given by the user. PredyFlexy is based on a method that relies on the usage of Library of Structural Prototypes (LSPs) to predict flexibility along the sequence. LSPs describe the structure of a protein with a set of local structures that are limited and recurrent.

2 Comparing flexibility scores to empirical B-factors from X-ray crystallography.

Second, we assessed the performance using different numbers of homologs, to see how this parameter affects the correlation between the scores and the actual B-factors.

4.3.1. OVERALL PERFORMANCE ASSESSMENT

To assess the performance of FlexAmino, we 1) compared its results to those of another available program of flexibility prediction, called PredyFlexy, and 2) compared the flexibility scores to the empirical B-factors of proteins with resolved crystallographic structures.

In both cases, we computed the Pearson correlation between the score obtained with FlexAmino and those of the other program/empirical B-factors. We performed this comparison for protein O08989 from Uniprot (mouse M-Ras protein) and we used the 1X1R PDB structure for the empirical assessment. For this test we used a modified version of the program to avoid including in the homologs the PDB of the query sequence itself.

When comparing with PredyFlexy scores, we obtained a significant, intermediate Pearson correlation of 0.40 (p-value = $4.0 \cdot 10^{-8}$). When comparing with the empirical p-values of the resolved structure, the results were better, as we obtained a correlation coefficient of 0.71 (p-value = $6.6 \cdot 10^{-27}$; Figure 10). Therefore we can conclude that our flexibility scores correlate well with empirical B-factors, and so they are a good indicator of “static” flexibility (i.e. of the range of movement of a residue around its mean position).

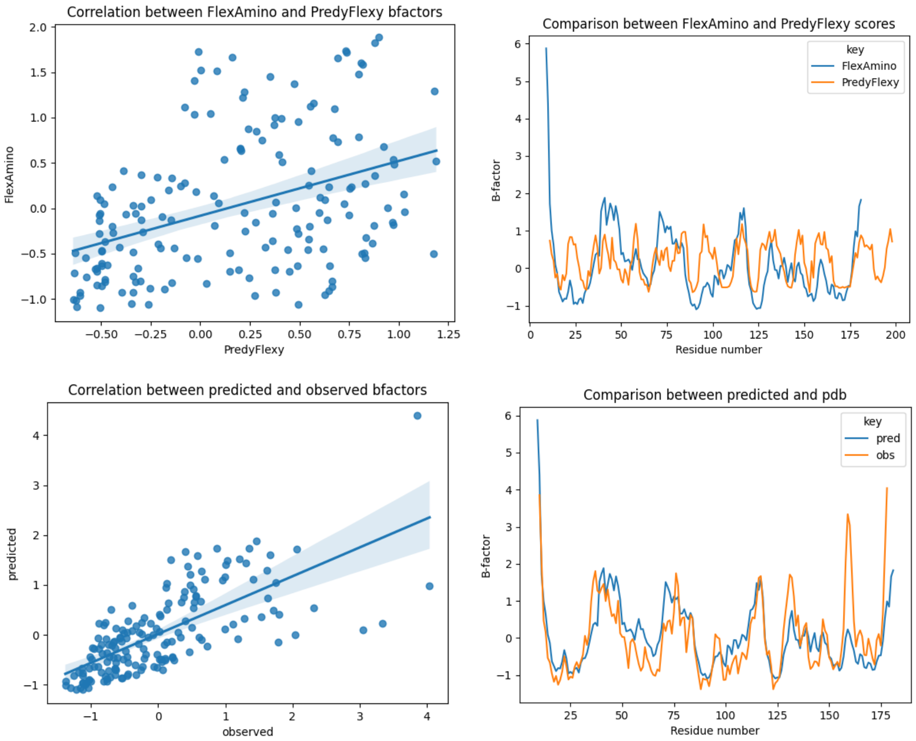

Figure 10. Comparison of the FlexAmino flexibility scores with B-factors predicted by PredyFlexy (top row) and with empirical B-factors from the X-ray crystallography structure (bottom row).

- Performance using different numbers of homologs

Using proteins O08989 and P0DP23, we computed the flexibility scores using different numbers of templates and we compared each set of scores to the actual, normalized crystallographic B-factors (again discarding the PDB corresponding to the target). Figure 11 shows the Pearson correlation coefficient obtained with different numbers of templates for both cases. In general, the correlation tends to increase with the number of templates, but the speed of the increase and the optimal number of templates is variable depending on the situation and the quality of the subsequent templates.

By default, FlexAmino uses the top ten BLAST results to compute the prediction, as this number provides an overall good performance in both of the studied cases.

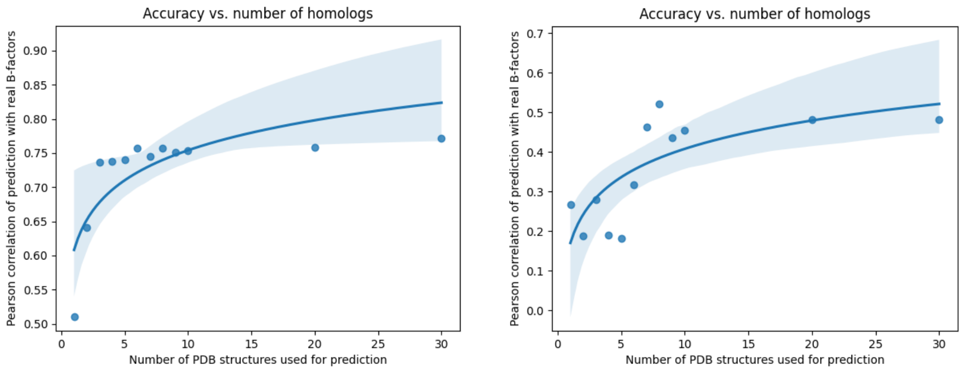

Figure 11. Pearson correlation coefficient between the flexibility scores and the real B-factors obtained using different numbers of templates, for proteins O08989 (left) and P0DP23 (right). As the number of templates increases, the correlation tends to stabilize.

5. LIMITATIONS

From the obtained results, we can conclude that although FlexAmino scores do contain some information about the flexibility of the different regions of a protein, they are generally not accurate enough to extract conclusions about the conformation changes of the protein without searching for more detailed information about better studied homologs. In addition, several attempts with different window sizes may be required for the profile to be interpretable.

This is most likely due to the program being based on B-factors, as these have been shown to depend on many variables (such as SAS) and so their information regarding the movements of the protein can vary. In this regard, an approach based on nuclear magnetic resonance (NMR) may provide better results, as this experimental technique measures proteins in solution and directly considers their movements. However, it must be kept in mind that NMR has size limitations to the structures it can resolve, so it may not be applicable to all proteins.

Another limitation is the reliance of the program on homologous proteins of known structure, which limits its applicability to those cases where there are at least some close homologs with resolved crystallographic structures. A possibility to be able to model proteins without directly relying on homologs would be to use machine learning approaches trained in a general way.

On the same line, we can also note that the current version of the program does not directly allow assessment of the reliability of the results regarding the “closeness” of the homologies, as evidenced by the results with protein P65206. Therefore, a potential improvement to the program would be to consider the reliability of the BLAST results and of the structural alignments in order to determine the templates to use, and to add some output regarding the quality and reliability of this part of the method.

The need to use UniProt identifiers and AlphaFold models is another limitation of the program, as it means it can only be used with protein sequences available in these databases, so it cannot be applied to newly discovered sequences without uploading them to UniProt and waiting for the AlphaFold model to be available. This could be solved in future versions by implementing the AlphaFold software or by calling the API.

6. BIBLIOGRAPHY

Bauer,J.A. *et al.* (2019) Normal mode analysis as a routine part of a structural investigation. *Molecules*, **24**, 3293.

Bothner,B. *et al.* (1998) Evidence of Viral Capsid Dynamics Using Limited Proteolysis and Mass Spectrometry *. *J. Biol. Chem.*, **273**, 673–676.

De Brevern,A.G. *et al.* (2012) PredyFlexy: flexibility and local structure prediction from sequence. *Nucleic Acids Res.*, **40**, W317.

Bryn Fenwick,R. *et al.* (2014) Integrated description of protein dynamics from room-temperature X-ray crystallography and NMR. *Proc. Natl. Acad. Sci. U. S. A.*, **111**, E445–E454.

CAMK1D - Calcium/calmodulin-dependent protein kinase type 1D - Homo sapiens (Human) - CAMK1D gene & protein.

Chung,K.Y. *et al.* (2011) Conformational changes in the G protein Gs induced by the β2 adrenergic receptor. *Nature*, **477**, 611–617.

Craveur,P. *et al.* (2015) Protein flexibility in the light of structural alphabets. *Front. Mol. Biosci.*, **2**, 1–20.

Falconi,M. *et al.* (2002) Flexibility in monomeric Cu,Zn superoxide dismutase detected by limited proteolysis and molecular dynamics simulation. *Proteins Struct. Funct. Bioinforma.*, **47**, 513–520.

Fontana,A. *et al.* (2002) Correlation between sites of limited proteolysis and segmental mobility in thermolysin. *Biochemistry*, **25**, 1847–1851.

Fuhrmann,J. *et al.* (2009) McsB Is a protein arginine kinase that phosphorylates and inhibits the heat-shock regulator ctsr. *Science*, **324**, 1323–1327.

GNAI1 - Guanine nucleotide-binding protein G(i) subunit alpha-1 - *Cavia porcellus* (Guinea pig) - GNAI1 gene & protein.

Goldberg,J. *et al.* (1996) Structural Basis for the Autoinhibition of Calcium/Calmodulin-Dependent Protein Kinase I. *Cell*, **84**, 875–887.

Halle,B. (2002) Flexibility and packing in proteins. *Proc. Natl. Acad. Sci. U. S. A.*, **99**, 1274–1279.

Hartmann,H. *et al.* (1982) Conformational substates in a protein: structure and dynamics of metmyoglobin at 80 K. *Proc. Natl. Acad. Sci. U. S. A.*, **79**, 4967–4971.

Hoffman,L. *et al.* (2011) Conformational changes underlying calcium/calmodulin-dependent protein kinase II activation. *EMBO J.*, **30**, 1251–1262.

Huber,R. (1979) Conformational flexibility in protein molecules. *Nature*, **280**, 538–538.

Karshikoff,A. *et al.* (2015) Rigidity versus flexibility: the dilemma of understanding protein thermal stability. *FEBS J.*, **282**, 3899–3917.

Komeiji,Y. *et al.* (2002) Molecular dynamics simulations revealed Ca2+-dependent conformational change of Calmodulin. *FEBS Lett.*, **521**, 133–139.

mcsB - Protein-arginine kinase - *Staphylococcus aureus* (strain N315) - mcsB gene & protein.

Pabon,N.A. and Camacho,C.J. (2017) Probing protein flexibility reveals a mechanism for selective promiscuity. *Elife*, **6**, e22889.

Rasmussen,S.G.F. *et al.* (2011) Crystal structure of the β2 adrenergic receptor–Gs protein complex. *Nature*, **477**, 549–557.

Reinknecht,C. *et al.* (2021) Patterns in protein flexibility: A comparison of NMR “ensembles”, MD Trajectories, and crystallographic B-factors. *Molecules*, **26**, 1484.

Ringe,D. and Petsko,G.A. (1986) Study of protein dynamics by X-ray diffraction. *Methods Enzymol.*, **131**, 389–433.

Salmon,L. *et al.* (2011) Nuclear magnetic resonance provides a quantitative description of protein conformational flexibility on physiologically important time scales. *Biochemistry*, **50**, 2735–2747.

Sherwood,D. and Cooper,J. (2011) Crystals, X-rays and Proteins: Comprehensive Protein Crystallography Oxford University Press.

Soderling,T.R. and Stull,J.T. (2001) Structure and Regulation of Calcium/Calmodulin-Dependent Protein Kinases. *Chem. Rev.*, **101**, 2341–2351.

Sunada,S. *et al.* (1998) Calculation of nuclear magnetic resonance order parameters in proteins by normal mode analysis. *J. Chem. Phys.*, **104**, 4768.

Suskiewicz,M.J. *et al.* (2019) Structure of McsB, a protein kinase for regulated arginine phosphorylation. *Nat. Chem. Biol. 2019 155*, **15**, 510–518.

Teilum,K. *et al.* (2009) Functional aspects of protein flexibility. *Cell. Mol. Life Sci.*, **66**, 2231–2247.

Xu,Y. *et al.* (2008) Flexibility of Aromatic Residues in the Active-Site Gorge of Acetylcholinesterase: X-ray versus Molecular Dynamics. *Biophys. J.*, **95**, 2500–2511.

Yousef,M.S. et al. (2003) Induced fit in guanidino kinases--comparison of substrate-free and transition state analog structures of arginine kinase. Protein Sci., 12, 103–111.