

# 1      The Backmap Python Module: How a 2      Simpler Ramachandran Number Can 3      Simplify the Life of a Protein Simulator

4      Ranjan V. Mannige\*

5      \* ranjanmannige@gmail.com

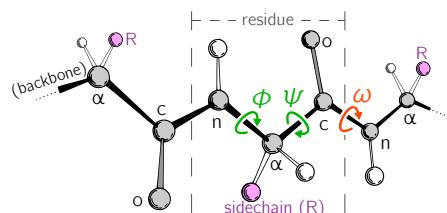
## 6      ABSTRACT

7      Protein backbones display complicated structures that often undergo numerous types of structural  
8      transformations. Due to the large number of structural degrees of freedom available to a backbone,  
9      it is often difficult to assess exactly where and how regions of a protein structure undergo structural  
10     transformation. This large structural phase makes it hard to survey new structural data, such as molecular  
11     dynamics trajectories or NMR-derived structural ensembles. This report discusses the Ramachandran  
12     number  $\mathcal{R}$  as a residue-level structural metric that could simplify the life of anyone contending with large  
13     numbers of structural data associated with protein backbones. In particular, this report 1) presents a new  
14     tool – BACKMAP – that can be universally installed using ‘> pip install backmap’, 2) discusses a much  
15     simpler closed form of  $\mathcal{R}$  that makes it more easy to calculate, and 3) shows how  $\mathcal{R}$  dramatically reduces  
16     the dimensionality of the protein backbone, thereby making it ideal for simultaneously interrogating large  
17     number of protein structures. In short,  $\mathcal{R}$  is a simple and succinct descriptor of protein backbones and  
18     their dynamics.

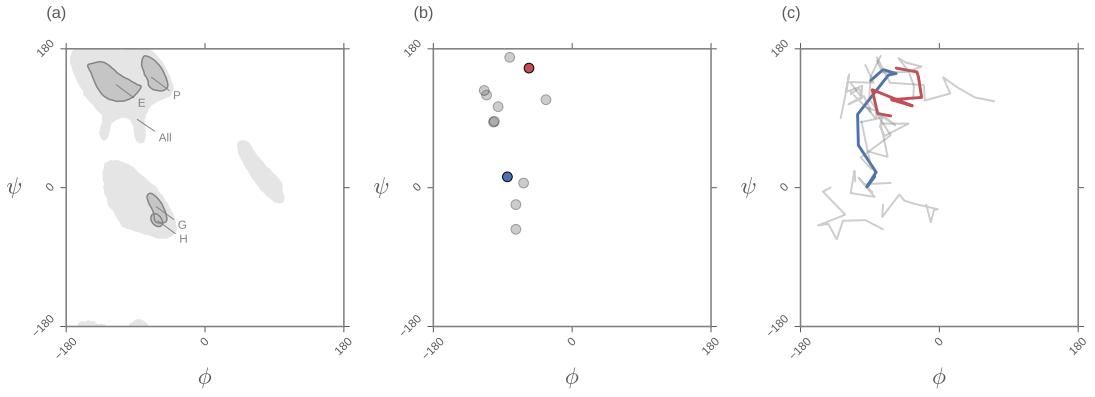
## 19     INTRODUCTION

20     Proteins are a class of biomolecules unparalleled in their functionality (Berg *et al.*, 2010). A natural  
21     protein may be thought of as a linear chain of amino acids, each normally sourced from a repertoire of 20  
22     naturally occurring amino acids. Proteins are important partially because of the structures that they access:  
23     the conformations (conformational ensemble) that a protein assumes determines the functions available  
24     to that protein. However, all proteins are dynamic: even stable proteins undergo long-range motions  
25     in its equilibrium state; i.e., they have substantial diversity in their conformational ensemble (Mannige,  
26     2014). Additionally, a number of proteins undergo conformational transitions, without which they may  
27     not properly function. Finally, some proteins – intrinsically disordered proteins – display massive disorder  
28     whose conformations dramatically change over time (Uversky, 2003; Fink, 2005; Midic *et al.*, 2009;  
29     Espinoza-Fonseca, 2009; Uversky and Dunker, 2010; Tompa, 2011; Sibille and Bernado, 2012; Kosol  
30     *et al.*, 2013; Dunker *et al.*, 2013; Geist *et al.*, 2013; Baruah *et al.*, 2015), and whose characteristic  
31     structures are still not well-understood (Beck *et al.*, 2008).

32     Large-scale changes in a protein occur due to changes in protein backbone conformations. Fig. 1 is a  
33     cartoon representation of a peptide/protein backbone, with the backbone bonds themselves represented



**Figure 1. Backbone conformational degrees of freedom** dominantly depend on the dihedral angles  $\phi$  and  $\psi$  (green), and to a smaller degree depend on the third dihedral angle ( $\omega$ ; red) as well as bond lengths and angles (unmarked).



**Figure 2.** While the Ramachandran plot is useful for getting a *qualitative* sense of peptide backbone structure (a, c), it is not a convenient representation for exploring peptide backbone dynamics (c).

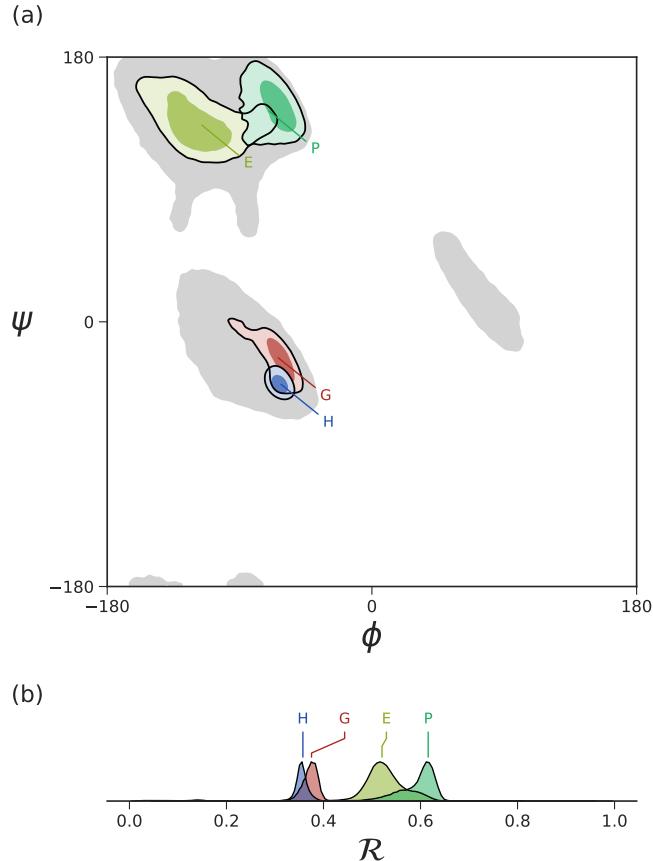
34 by darkly shaded bonds. [Ramachandran et al. \(1963\)](#) had recognized that the backbone conformational  
35 degrees of freedom available to an amino acid (residue)  $i$  is almost completely described by only two  
36 dihedral angles:  $\phi_i$  and  $\psi_i$  (Fig. 1, green arrows). Today, protein structures described in context of the  
37 two-dimensional ( $\phi, \psi$ )-space are called Ramachandran plots.

38 The Ramachandran plot is recognized as a powerful tool for two reasons: 1) it serves as a map  
39 for structural ‘correctness’ ([Laskowski et al., 1993](#); [Hooft et al., 1997](#); [Laskowski, 2003](#)), since many  
40 regions within the Ramachandran plot space are energetically not permitted ([Momen et al., 2017](#)); and  
41 2) it provides a qualitative snapshot of the structure of a protein ([Berg et al., 2010](#); [Alberts et al., 2002](#);  
42 [Subramanian, 2001](#)). For example, particular regions within the Ramachandran plot indicate the presence  
43 of particular secondary locally-ordered structures such as the  $\alpha$ -helix and  $\beta$ -sheet (see Fig. 2a).

44 While the Ramachandran plot has been useful as a measure of protein backbone conformation, it is  
45 not popularly used to assess structural dynamism and transitions (unless specific knowledge exists about  
46 whether a particular residue is believed to undergo a particular structural transition). This is because  
47 of the two-dimensionality of the plot: describing the behavior of every residue involves tracking its  
48 position in two-dimensional ( $\phi, \psi$ ) space. For example, a naive description of positions of a peptide in a  
49 Ramachandran plot (Fig. 2b) needs more annotations for a per-residue analysis of the peptide backbone’s  
50 structure. Given enough residues, it would be impractical to track the position of each residue within a  
51 plot. This is compounded with time, as each point in (b) becomes a curve (c), further confounding the  
52 situation. The possibility of picking out previously unseen conformational transitions and dynamism  
53 becomes a logistical impracticality. As indicated above, this impracticality arises primarily from the fact  
54 that the Ramachandran plot is a two-dimensional map.

55 Consequently, there has been no single compact descriptor of protein structure. This impedes that  
56 naïve or hypothesis-free exploration of new trajectories/ensembles. For example, tracking changes in  
57 protein trajectory is either overly detailed or overly holistic: an example of an overly detailed study is the  
58 tracking on exactly one or a few atoms over time (this already poses a problem, since we would need to  
59 know exactly which atoms are expected to partake in a transition); an example of a holistic metric is the  
60 radius of gyration (this also poses a problem, since we will never know which residues contribute to a  
61 change in radius of gyration without additional interrogation). With protein dynamics undergoing a new  
62 renaissance – especially due to intrinsically disordered proteins and allosteric – having hypothesis-agnostic  
63 yet detailed (residue-level) metrics of protein structure has become even more relevant.

64 It has recently been shown that the two Ramachandran backbone parameters ( $\phi, \psi$ ) may be conve-  
65 niently combined into a single number – the Ramachandran *number* [ $\mathcal{R}(\phi, \psi)$  or simply  $\mathcal{R}$ ] – with little  
66 loss of information ([Mannige et al., 2016](#)). In a previous report, detailed discussions were provided  
67 regarding the reasons behind and derivation of  $\mathcal{R}$  ([Mannige et al., 2016](#)). This report provides a simpler  
68 version of the equation previously published ([Mannige et al., 2016](#)), and further discusses how  $\mathcal{R}$  may be  
69 used to provide information about protein ensembles and trajectories. Finally, we introduce a software  
70 package – BACKMAP – that can be used by to produce MAPs that describe the behavior of a protein  
71 backbone within user-inputted conformations, structural ensembles and trajectories. This package is



**Figure 3.** The distribution of dominant regular secondary structures are shown in  $[\phi, \psi]$ -space (a) and in  $\mathcal{R}$ -space (b). Both Ramachandran plots (a) and Ramachandran ‘lines’ (b) show equivalent resolution of secondary structure , allowing for a more compact representation of Ramachandran plots Mannige *et al.* (2016).

72 presently available on GitHub (<https://github.com/ranganannige/BackMAP>).

### 73 INTRODUCING THE *SIMPLIFIED RAMACHANDRAN NUMBER* ( $\mathcal{R}$ )

74 The Ramachandran number is both an idea and an equation. Conceptually, the Ramachandran number ( $\mathcal{R}$ )  
 75 is any closed form that collapses the dihedral angles  $\phi$  and  $\psi$  into one structurally meaningful number  
 76 (Mannige *et al.*, 2016). Mannige *et al.* (2016) presented a version of the Ramachandran number that  
 77 was complicated in closed form, thereby reducing its utility. Here, a much more simplified version of the  
 78 Ramachandran number is introduced. Section 1.1 shows how this simplified form was derived from the  
 79 original closed form (Eqns. 3 and 4).

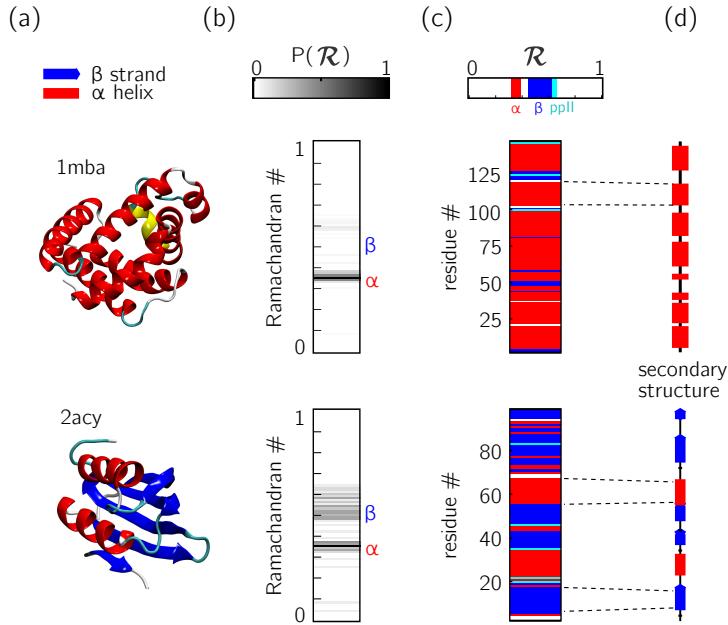
Given arbitrary limits of  $\phi \in [\phi_{\min}, \phi_{\max}]$  and  $\psi \in [\psi_{\min}, \psi_{\max}]$ , where the minimum and maximum values differ by  $360^\circ$ , the most general and accurate equation for the Ramachandran number is

$$\mathcal{R}(\phi, \psi) \equiv \frac{\phi + \psi - (\phi_{\min} + \psi_{\min})}{(\phi_{\max} + \psi_{\max}) - (\phi_{\min} + \psi_{\min})}. \quad (1)$$

For consistency, we maintain throughout this paper that  $\phi_{\min} = \psi_{\min} = -180^\circ$  or  $-\pi$  radians, which makes

$$\mathcal{R}(\phi, \psi) = \frac{\phi + \psi + 2\pi}{4\pi}. \quad (2)$$

80 As evident in Fig. 3, the distributions within the Ramachandran plot are faithfully reflected in corre-  
 81 sponding distributions within Ramachandran number space. This paper shows how the Ramachandran



**Figure 4. Two types of  $\mathcal{R}$ -codes.** Digesting protein structures (a) using  $\mathcal{R}$  numbers either as histograms (b) or per-residue codes (c) allow for compact representations of salient structural features. For example, a single glance at the histograms indicate that protein 1mba is likely all  $\alpha$ -helical, while 2acy is likely a mix of  $\alpha$ -helices and  $\beta$ -sheets. Additionally, residue-specific codes (c) not only indicate secondary structure content, but also exact secondary structure stretches (compare to d), which gives a more complete picture of how the protein is linearly arranged.

number is both compact enough and informative enough to generate immediately useful graphs (map) of a dynamic protein backbone.

## REASON TO USE THE RAMACHANDRAN NUMBER

### Ramachandran numbers are more compact than one might realize

An important aspect of the Ramachandran number ( $\mathcal{R}$ ) lies in its compactness compared to the traditional Ramachandran pair  $(\phi, \psi)$ . Say we have an  $N$ -residue peptide. Then, switching from  $(\phi, \psi)$  to  $\mathcal{R}$  appears to only reduce the number of variables from  $2N$  to  $N$ , and hence by half. However,  $(\phi, \psi)$  values are *coupled*, i.e., for any  $N$ -length peptide, any ordering of  $[\phi_1, \phi_2, \dots, \phi_N, \psi_1, \psi_2, \dots, \psi_N]$  can not describe the structure, it is only *pairs* –  $[(\phi_1, \psi_1), (\phi_2, \psi_2), \dots, (\phi_N, \psi_N)]$  – that can. Therefore, we must think of switching from  $(\phi, \psi)$ -space to  $\mathcal{R}$ -space as a switch in structure space per residue from  $N$  two-tuples  $(\phi_i, \psi_i)$  that reside in  $\phi \times \psi$  space to  $N$  single-dimensional numbers ( $\mathcal{R}_i$ ).

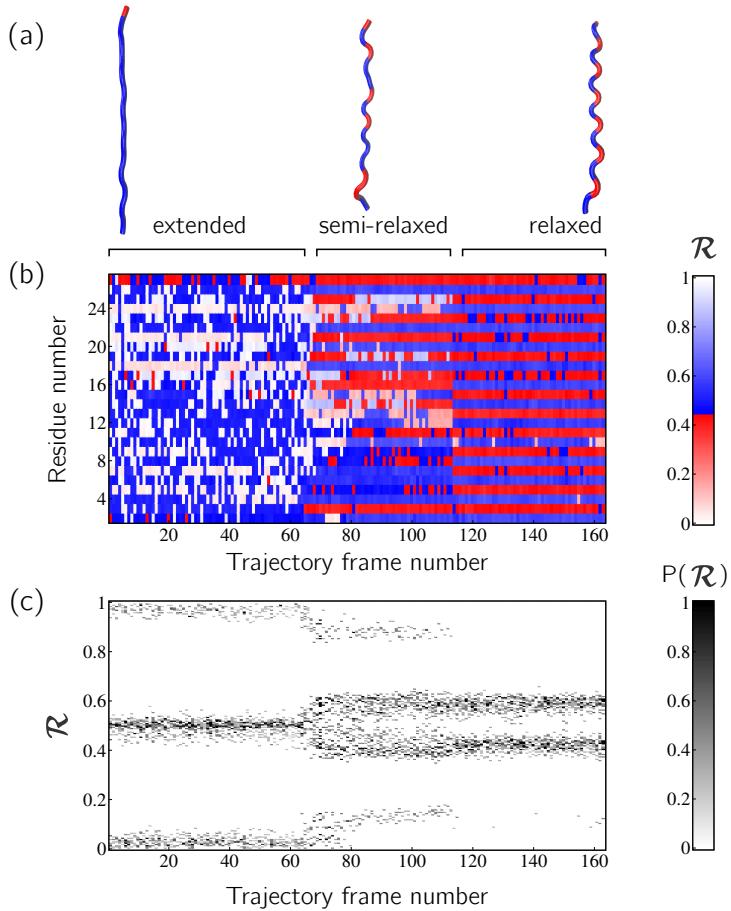
The value of this conversion is that the structure of a protein can be described in various one-dimensional arrays (per-structure “Ramachandran codes” or “ $\mathcal{R}$ -codes”), which, when arranged vertically/columnnarly, constitute easy to digest codes. See, e.g., Fig. 4.

### Ramachandran codes are stackable

In addition to assuming a small form factor,  $\mathcal{R}$ -codes may then be *stacked* side-by-side for visual and computational analysis. There lies its true power.

For example, the one- $\mathcal{R}$ -to-one-residue mapping means that the entire residue-by-residue structure of a protein can be shown using a string of  $\mathcal{R}_i$ s (which would show regions of secondary structure and disorder, for starters). Additionally, an entire protein’s backbone makeup can be shown as a histogram in  $\mathcal{R}$ -space (which may reveal a protein’s topology). The power of this format lies not only in the capacity to distill complex structure into compact spaces, but in its capacity to display *many* complex structures in this format, side-by-side (stacking).

Peptoid nanosheets (Mannige *et al.*, 2015) will be used here as an example of how multiple structures, in the form of  $\mathcal{R}$ -codes, may be stacked to provide immediately useful pictograms. Peptoid nanosheets



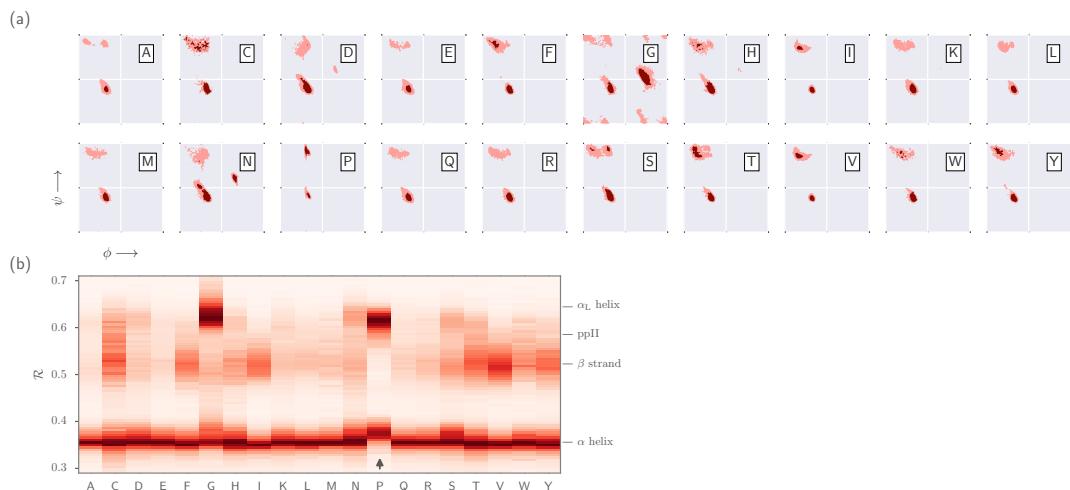
**Figure 5. Stacked  $\mathcal{R}$ -codes provide useful information at a glance.**

are a recently discovered peptide-mimic that were shown to display a novel secondary structure (Mannige *et al.*, 2015). In particular, each peptoid within the nanosheet displays backbone conformations that alternate in chirality, causing the backbone to look like a meandering snake that nonetheless maintains an overall linear direction. This secondary structure was discovered by first setting up a nanosheet where all peptoid backbones are restrained in the extended format (Fig. 5a, left), after which the restraints were energetically softened (a, middle) and completely released (a, right). As evident in Fig. 5b and Fig. 5c, the two types of  $\mathcal{R}$ -code stacks display salient information at first glance: 1) Fig. 5b shows that the extended backbone first undergoes some rearrangement with softer restraints, and then becomes much more binary in arrangement as we look down the backbone (excepting the low-order region in the middle, unshown in Fig. 5a); and 2) Fig. 5c shows that lifting restraints on the backbone causes a dramatic change in backbone topology, namely a birth of a bimodal distribution evident in the two parallel bands.

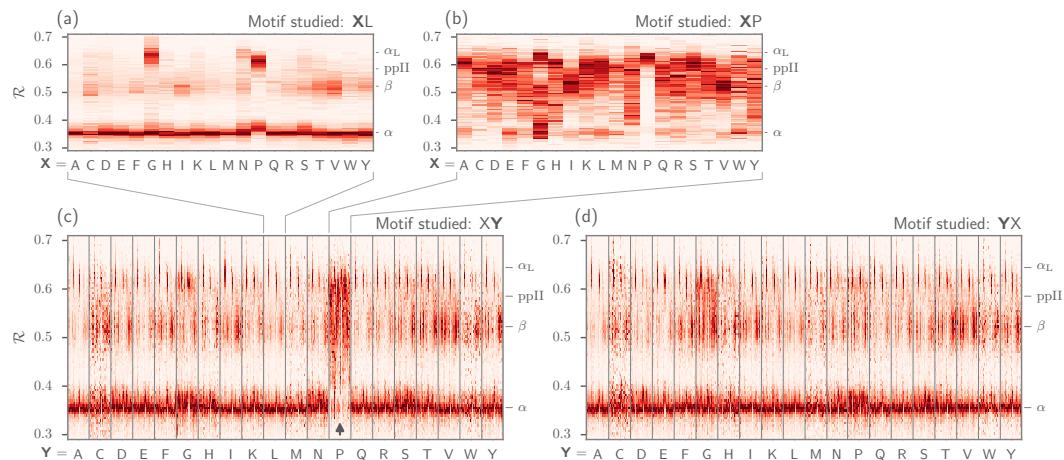
By utilizing  $\mathcal{R}$  maps such as those in Fig. 5 provide information about every  $\phi$  and  $\psi$  within the backbone. As such, these maps are dubbed MAPs, for Multi Angle Pictures. A Python package called BACKMAP created Fig. 5a and b, which is provided as a GitHub repository at <https://github.com/ranjanmannige/BackMAP>. BACKMAP takes in a PDB structure file containing a single structure, or multiple structures separated by the code ‘MODEL’.

### Other uses for $\mathcal{R}$ : picking out subtle differences from high volume of data

This section expands on the notion that  $\mathcal{R}$ -numbers – due to their compactness/stackability – can be used to pick out backbone structural trends that would be hard to decipher using any other metric. For example, it is well known that prolines (P) display unusual backbone behavior: in particular, proline backbones occupy structures that are close to but distinct from  $\alpha$ -helical regions. Due to the two-dimensionality of Ramachandran plots (Fig. 6a), such distinctions are hard to visually pick out from Ramachandran plots. However, stacking per-amino-acid  $\mathcal{R}$ -codes side by side make such differences patent (Fig. 6b; see



**Figure 6. Ramachandran lines are stackable – Part I.** Panel (a) shows the per-amino acid backbone behavior of an average protein found in the protein databank (PDB). While these plots are useful, it is difficult to compare such plots. For example, it is hard to pick out the change in the  $\alpha$ -helical region of the proline plot (P). However, when we convert Ramachandran plots to Ramachandran *lines* [by converting  $(\phi_i, \psi_i) \rightarrow \mathcal{R}_i$ ], we are able to conveniently “stack” Ramachandran lines calculated for each residue. Then, even visually, it is obvious that proline does not occupy the canonical  $\alpha$ -helix region, which is not evident to an untrained eye in (a).



**Figure 7. Ramachandran lines are stackable – Part II.** Similar to Fig. 6b, Panel (a) represents the behavior of an amino acid ‘Y’ situated *before* a leucine (assuming that we are reading a sequence from the N terminal to the C terminal). Panel (b) similarly represents the behavior of specific amino acids situated before a proline. While residues preceding a leucine behave similarly to their average behavior (Fig. 6a), most residues preceding prolines appear to be enriched in structures that change ‘direction’ or backbone chirality ( $R > 0.5$ ). Panel (c) shows the behavior of individual amino acids when situated before each of the 20 amino acids. This graph shows a major benefit of side-by-side Ramachandran line “stacking”: general trends become much more obvious. For example, it is evident that glycines and prolines dramatically modify the structure of an amino acid preceding it (compared to average behavior of amino acids in Fig. 6b). This trend is not as strong when considering amino acids that *follow* glycines or prolines (c). Such trends, while previously discovered [e.g., Gunasekaran *et al.* (1998); Ho and Brasseur (2005)], would not be accessible when naïvely considering Ramachandran plots because one would require 400 ( $20 \times 20$ ) distinct Ramachandran plots to compare.

130 arrow).

131 It is also known that amino acids preceding prolines display unusual shift in chirality. For example,  
132 Fig. 7 shows that amino acids appearing before prolines and glycines behave much more differently than  
133 they would otherwise. While these results have been discussed previously (Gunasekaran *et al.*, 1998; Ho  
134 and Brasseur, 2005), they were reported more than 30 years after the first structures were published; they  
135 would have been relatively easy to find if  $\mathcal{R}$ -codes were to be used regularly.

136 The relationships in Figs. 6 and 7 show how subtle changes in structure can be easily picked out when  
137 structures are stacked side-by-side in the form of  $\mathcal{R}$ -codes. Such subtle changes are often witnessed when  
138 protein backbones transition from one state to another.

## 139 USING THE BACKMAP PYTHON MODULE

### 140 Installation

141 BACKMAP may either be downloaded from the github repository, or installed directly by running the  
142 following line in the command prompt (assuming that pip exists): > pip install backmap

### 143 Usage

144 The module can either be imported and used within existing scripts, or used as a standalone package usign  
145 the command ‘python -m backmap’. First the in-script usage will be discussed.

#### 146 In-script usage: first simple test

147 The simplest test would be to generate Ramachandran numbers from  $(\phi, \psi)$  pairs:

```
148 # Import module
149 import backmap
150 # Convert (phi, psi) to R
151 print backmap.R(phi=0, phi=0) # Expected output: 0.5
152 print backmap.R(-180, -180) # Expected output: 0.0
153 print backmap.R(180, 180) # Expected output: 1.0 (equivalent in meaning to 0)
```

1  
2  
3  
4  
5  
6

#### 156 In-script usage: basic usage for creating Multi-Angle Pictures (MAPs)

157 As seen above, the generation of Ramachandran numbers from  $(\phi, \psi)$  pairs is simple. However, creating  
158 MAPs – Multi-Angle Pictures of protein backbones – requires a few more steps (present as a test in the  
159 downloadable module):

##### 160 1. Select and read a protein PDB structure

161 Each trajectory frame must be a set of legitimate protein databank "ATOM" records separated by  
162 "MODEL" keywords.

```
163 import backmap
164 pdbsfn = './pdbs/nanosheet_birth_U7.pdb' # Set pdb name
165 data = backmap.read_pdb(pdbsfn) # READ PDB in the form of a matrix with columns
```

1  
2  
3

168 Here, ‘data’ is a 2d array with four columns [‘model’, ‘chain’, ‘resid’, ‘R’]. The first row of  
169 ‘data’ is the header (i.e., the name of the column, e.g., ‘model’), with values that follow.

##### 170 2. Select color scheme (color map)

171 In addition to custom colormaps listed in the next section, one can also use standardly available at  
172 [matplotlib.org](http://matplotlib.org) (e.g., ‘Reds’ or ‘Reds\_r’).

```
173 # setting the name of the colormap
174 cmap = "SecondaryStructure"
```

4  
5

##### 177 3. Draw per-chain MAPs

```
178 # Grouping by chain
179 grouped_data = backmap.group_by(data, group_by='chain',
180                                 columns_to_return=['model', 'resid', 'R'])
181 for chain in grouped_data.keys(): # Going through each chain
182     # Getting the X,Y,Z values for each entry
183     models, residues, Rs = grouped_data[chain]
```

6  
7  
8  
9  
10  
11

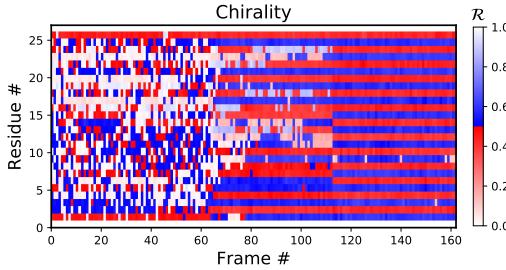
```

185     # Finally, creating (but not showing) the graph
186     backmap.draw_xyz(X = models, Y = residues, Z = Rs
187         , xlabel = 'Frame #', ylabel = "Residue #", zlabel = '$\mathcal{R}$'
188         ,cmap = cmap, title = "Chain: "+chain+""
189         ,vmin=0,vmax=1)
190     # Now, we display the graph:
191     plt.show() # ... one can also use plt.savefig() to save to file

```

---

As one would expect, this is the business end of the code. By changing how one assigns values to ‘X’ and ‘Y’, one can easily construct and draw other types of graphs such as time-resolved histograms, root mean squared fluctuations, root mean squared deviation, etc. Running the module as a standalone script would produce all these graphs automatically. ‘`plt.show()`’ would result in the following image being rendered:



198

### In-script usage: Creating custom graphs

200 Other types of grpahs can be easily created by modifying part three of the code above. For example, the  
201 following code creates histograms of  $R'$ , one for each model (starting from line 10 above).

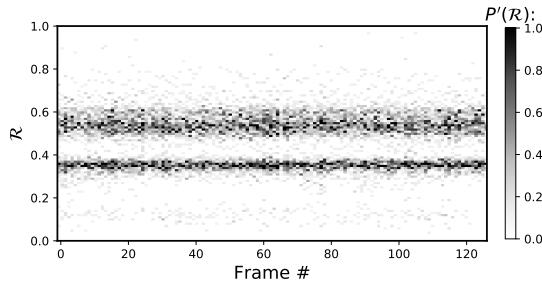
```

202 for chain in grouped_data.keys():
203     models, residues, Rs = grouped_data[chain]
204
205     'Begin custom code'
206     X = []; Y=[]; Z=[]; # Will set X=model, Y=R, Z=P(R)
207     # Bundling the three lists into one 2d array
208     new_data = np.array(zip(models, residues, Rs))
209     # Getting all R values, model by model
210     for m in sorted(set(new_data[:,0])): # column 0 is the model column
211         # Getting all Rs for that model #
212         current_rs = new_data[np.where(new_data[:,0]==m)][:,2] # column 2 contains R
213         # Getting the histogram
214         a,b = np.histogram(current_rs, bins=np.arange(0,1.01,0.01))
215         max_count = float(np.max(a))
216         for i in range(len(a)):
217             X.append(m); Y.append((b[i]+b[i+1])/2.0); Z.append(a[i]/float(np.sum(a)));
218     'End custom code'
219
220     # Finally, creating (but not showing) the graph
221     draw_xyz(X = X, Y = Y, Z = Z
222         , xlabel = 'Frame #', ylabel = "$\mathcal{R}'$"
223         ,zlabel = "$P'(\mathcal{R})$"
224         ,cmap = 'Greys', ylim=[0,1])
225     plt.yticks(np.arange(0,1.00001,0.2))
226     # Now, we display the graph:
227     plt.show() # ... one can also use plt.savefig() to save to file

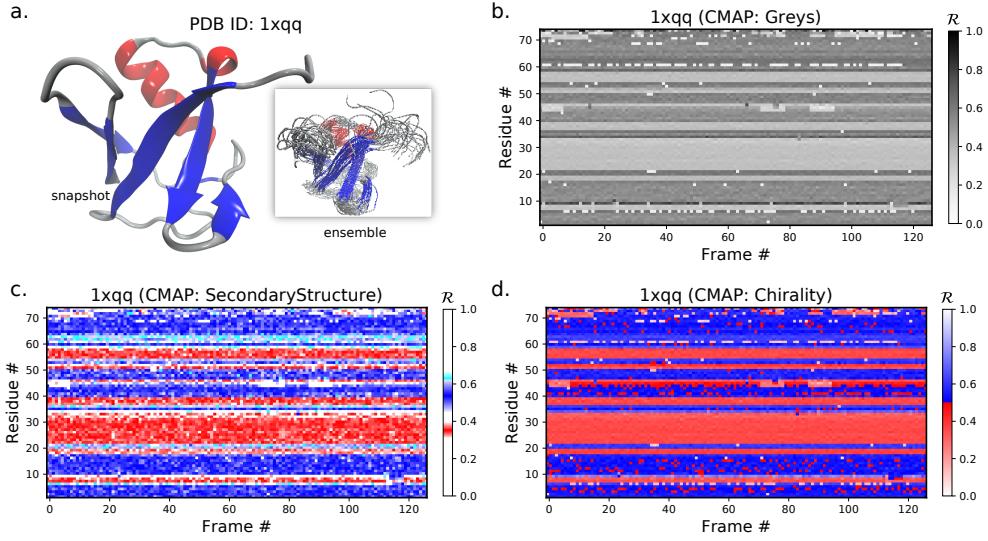
```

---

229 The code above results in the following graph:



230



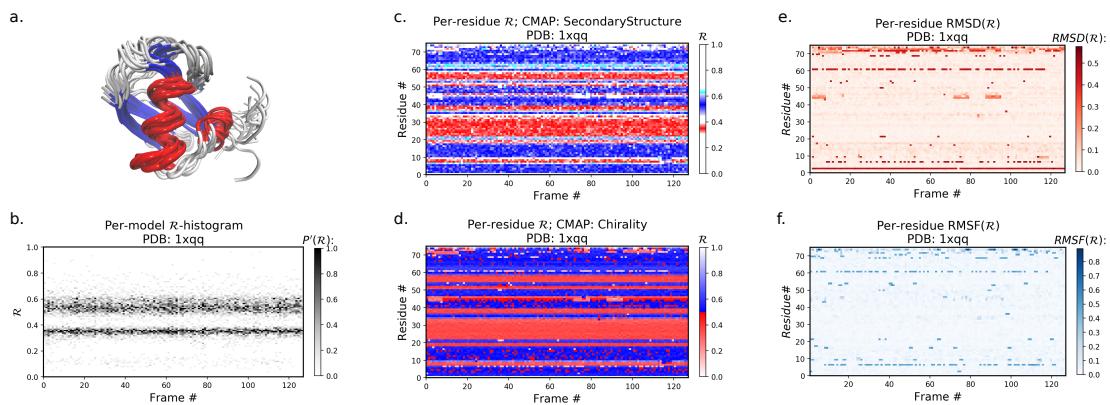
**Figure 8.** A protein ensemble (a) along with some MAPs colored with different themes (b-d). Panels (c) and (d) are provided by the BACKMAP module. In Panel (a),  $\beta$ -sheets are shown in blue and all helices are shown in red.

### 231 In-script usage: Available color schemes (CMAPs)

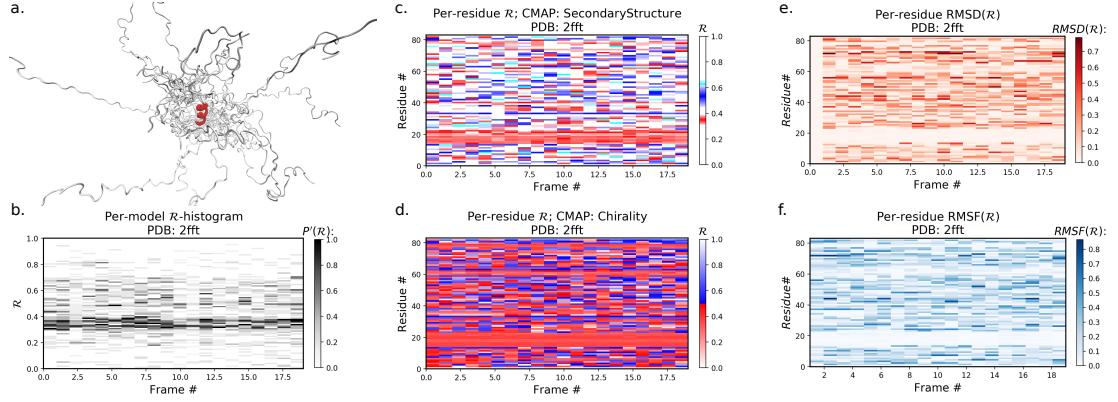
232 Aside from the general color maps (cmaps) that exist in matplotlib (e.g., ‘Greys’, ‘Reds’, or, god forbid,  
 233 ‘jet’), BACKMAP provides two new colormaps: ‘Chirality’ (key: +twists – red; -ve twists: blue),  
 234 ‘SecondaryStructure’ (key: helices – red; sheets – blue; ppII helices – cyan). right twisting  
 235 backbones are shown in red; left twisting backbones are shown in blue). Fig. 8 shows how a single protein  
 236 ensemble may be described using these schematics. As illustrated in Fig. 8b, cmaps available within  
 237 the standard matplotlib package do not distinguish between major secondary structures well, to a great  
 238 extent, while those provided by BACKMAP do. Note that colormap ‘Chirality’ ignores regions that  
 239 are less accessible to regular protein backbones. In case it is known that the protein backbone accesses  
 240 non-protein regions of the Ramachandran plot, a four-color schematic will be needed (see Section 1.1 for  
 241 more discussions).

### 242 Stand Alone Usage

243 BACKMAP can be used as a stand alone package by running ‘> python -m backmap -pdb <pdb\_dir\_or\_file>’.  
 244 The sections below describes the expected outputs and how they may be interpreted.



**Figure 9.** Protein 1xqq describes a stable protein.



**Figure 10.** Protein **2fft** describes an intrinsically disordered protein.

#### 245 Stand Alone Example I: A Stable Protein

246 Panels (b) through (f) of Fig. 9 below were created by running ‘> python -m backmap ./tests/pdfs/1xqq.pdb’  
 247 (Panel (b) was created using [VMD](#)). These graphs indicates that protein **1xqq** describes a conformationally  
 248 stable protein.

249 In particular, each column in Panel (b) describes the histogram in Ramachandran number ( $\mathcal{R}$ ) space  
 250 for a single model/timeframe. These histograms show the presence of both  $\alpha$ -helices (at  $\mathcal{R} \approx 0.34$ )  
 251 and  $\beta$ -sheets (at  $\mathcal{R} \approx 0.52$ ). Additionally, Panels (c) and (d) describe per-residue conformational plots  
 252 (colored by two different metrics or CMAPs), which show that most of the protein backbone remains  
 253 relatively stable over ‘time’ (e.g., few fluctuations in state or ‘color’ are evident over frame #). Finally,  
 254 Panel (e) describes the extent towards which a single residue’s state has deviated from the first frame,  
 255 and Panel (f) describes the extent towards which a single residue’s state has deviated from its state in the  
 256 previous frame. Both these graphs, show that this protein is relatively conformationally stable.

#### 257 Stand Alone Example II: An Intrinsically Disordered Protein

258 Fig. 10 is identical to Fig. 10, except that the panels pertain to an intrinsically disordered protein **2fft**  
 259 whose structural ensemble describes dramatically distinct conformations.

260 As compared to the conformationally stable protein above, protein **2fft** is much more flexible. Panel  
 261 (b) shows that the states accessed per model are diverse and dramatically fluctuate over the entire range of  
 262  $\mathcal{R}$  (this is especially true when compared to a stable protein, see Fig. 9b).

263 The diverse states occupied by each residue (Panels (c) and (d)) confirm the conformational variation  
 264 displayed by most of the backbone (Panels (e) and (f)) similarly show how most of the residues fluctuate  
 265 dramatically.

266 Yet, interestingly, Panels (c) through (f) also show an unusually stable region – residues 15 through  
 267 25 – which consistently display the same conformational ( $\alpha$ -helical) state at  $\mathcal{R} \approx 0.34$  (interpreted as the  
 268 color red in Panel (c)). This trend would be hard to recognize by simply looking at the structure (Panel  
 269 (a)).

## 270 CONCLUSION

271 A simpler Ramachandran number is reported –  $\mathcal{R} = (\phi + \psi + 2\pi)/(4\pi)$  – which, while a single number,  
 272 provides much information. For example, as discussed in [Mannige et al. \(2016\)](#),  $\mathcal{R}$  values above 0.5 are  
 273 left-handed, while those below 0.5 are right handed,  $\mathcal{R}$  values close to 0, 0.5 and 1 are extended,  $\beta$ -sheets  
 274 occupy  $\mathcal{R}$  values at around 0.52, right-handed  $\alpha$ -helices hover around 0.34. Given the Ramachandran  
 275 number’s ‘stackability’, single graphs can hold a detailed information of the progression/evolution of  
 276 molecular trajectories. Indeed, Fig. 7 shows how 400 distinct Ramachandran plots can easily be fit into  
 277 one graph when using  $\mathcal{R}$ . Finally, a python script/module (BACKMAP) has been provided in an online  
 278 [GitHub repository](#).

279 **ACKNOWLEDGMENTS**

280 During the development of this paper, RVM was partially supported by the Defense Threat Reduction  
 281 Agency under contract no. IACRO-B0845281. RVM thanks Alana Canfield Mannige for her critique. This  
 282 work was partially done at the Molecular Foundry at Lawrence Berkeley National Laboratory (LBNL),  
 283 supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy  
 284 under Contract No. DE-AC02-05CH11231.

285 **1 APPENDIX**

286 **1.1 Simplifying the Ramachandran number ( $\mathcal{R}$ )**

287 This section will derive the simplified Ramachandran number presented in this paper from the more  
 288 complicated looking Ramachandran number introduced previously [Mannige et al. \(2016\)](#).

Assuming the bounds  $\phi, \psi \in [-180^\circ, 180^\circ]$ , and the range  $\lambda$  equals  $360^\circ$ , the previously described  
 Ramachandran number takes the form

$$\mathcal{R}(\phi, \psi) \equiv \frac{R_{\mathbb{Z}}(\phi, \psi) - R_{\mathbb{Z}}(\phi_{\min}, \phi_{\min})}{R_{\mathbb{Z}}(\phi_{\max}, \phi_{\max}) - R_{\mathbb{Z}}(\phi_{\min}, \phi_{\min})}, \quad (3)$$

where,  $\mathcal{R}(\phi, \psi)$  is the Ramachandran number with range  $[0, 1]$ , and  $R_{\mathbb{Z}}(\phi, \psi)$  is the *unnormalized* integer-spaced Ramachandran number whose closed form is

$$R_{\mathbb{Z}}(\phi, \psi) = \left\lfloor (\phi - \psi + \lambda) \sigma / \sqrt{2} \right\rfloor + \left\lfloor \sqrt{2} \lambda \sigma \right\rfloor \left\lfloor (\phi + \psi + \lambda) \sigma / \sqrt{2} \right\rfloor. \quad (4)$$

289 Here,  $\lfloor x \rfloor$  rounds  $x$  to the closest integer value,  $\sigma$  is a scaling factor, discussed below, and  $\lambda$  is the  
 290 range of an angle in degrees (i.e.,  $\lambda = \phi_{\max} - \phi_{\min}$ ). Effectively, this equation does the following. 1) It  
 291 divides up the Ramachandran plot into  $(360^\circ \sigma^{1/2})^2$  squares, where  $\sigma$  is a user-selected scaling factor  
 292 that is measured in reciprocal degrees [see Fig. 8b in [Mannige et al. \(2016\)](#)]. 2) It then assigns integer  
 293 values to each square by setting the lowest integer value to the bottom left of the Ramachandran plot  
 294 ( $\phi = -180^\circ, \psi = -180^\circ$ ; green arrow in Fig. 1b) and proceeding from the bottom left to the top right  
 295 by iteratively slicing down  $-1/2$  sloped lines and assigning increasing integer values to each square that  
 296 one encounters. 3) Finally, the equation assigns any  $(\phi, \psi)$  pair within  $\phi, \psi \in [-\phi_{\min}, \phi_{\max}]$  to the integer  
 297 value ( $R_{\mathbb{Z}}$ ) assigned to the divvied-up square that they it exists in.

However useful Eqn. 3 is, the complexity of the equation may be a deterrent towards utilizing it. This  
 paper reports a simpler equation that is derived by taking the limit of Eqn. 3 as  $\sigma$  tends towards  $\infty$ . In  
 particular, when  $\sigma \rightarrow \infty$ , Eqn. 3 becomes

$$\mathcal{R}(\phi, \psi) = \lim_{\sigma \rightarrow \infty} \mathcal{R}(\phi, \psi) = \frac{\phi + \psi + \lambda}{2\lambda} = \frac{\phi + \psi + 2\pi}{4\pi}. \quad (5)$$

298 Conformation of this limit is shown numerically in Fig. 11. Since larger  $\sigma$ s indicate higher accuracy,  
 299  $\lim_{\sigma \rightarrow \infty} \mathcal{R}(\phi, \psi)$  represents an exact representation of the Ramachandran number. Using this closed form,  
 300 this report shows how both static structural features and complex structural transitions may be identified  
 301 with the help of Ramachandran number-derived plots.

302 **1.2 Other frames of reference**

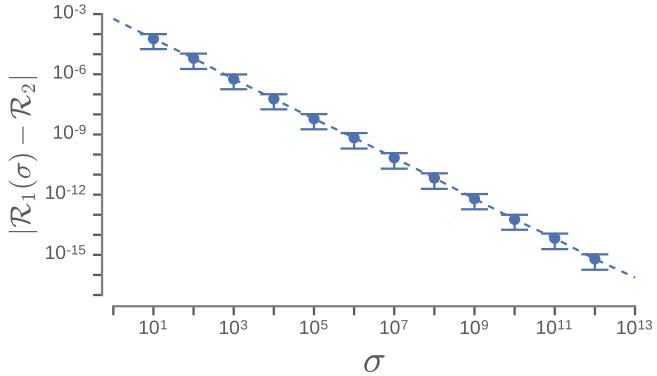
The Ramachandran number shown in Eqn. 5 expects  $\phi, \psi \in [-\lambda/2, \lambda/2]$ . Given arbitrary limits of  
 $\phi \in [\phi_{\max}, \phi_{\min}]$  and  $\psi \in [\psi_{\max}, \psi_{\min}]$ , the most general equation for the Ramachandran number is

$$\mathcal{R}(\phi, \psi) \equiv \frac{\phi + \psi - (\psi_{\min} + \psi_{\max})}{(\psi_{\max} + \psi_{\min}) - (\psi_{\min} + \psi_{\max})}. \quad (6)$$

For example, assuming that  $\phi, \psi \in [0, 2\pi]$ , the Ramachandran number in that frame of reference will be

$$\mathcal{R}(\phi, \psi)_{\phi, \psi \in [0, 2\pi]} = \frac{\phi + \psi}{4\pi}. \quad (7)$$

303 However, in doing so, the meaning of the Ramachandran number will change. The rest of this manuscript  
 304 will always assume that all angles range between  $-\pi$  ( $-180^\circ$ ) and  $\pi$  ( $180^\circ$ )



**Figure 11.** The increase in the accuracy measure ( $\sigma$ ) for the original Ramachandran number (Eqn. 4) results in values that tend towards the new Ramachandran number proposed in this paper (Eqn. 2).

## 2 A SIGNED RAMACHANDRAN NUMBER

An additional Ramachandran number – the *signed* Ramachandran number  $\mathcal{R}_S$  – is introduced here for backbones that are achiral.  $\mathcal{R}_S$  is identical to the original number in magnitude, but which changes sign from + to – as you approach  $\mathcal{R}$  numbers that are to the right (or below) the positively sloped diagonal. I.e.,

$$\mathcal{R}_S = \begin{cases} \mathcal{R} & , \text{if } \psi \geq \phi \\ \mathcal{R} \times -1 & , \text{if } \psi < \phi \end{cases} \quad (8)$$

This metric is important for those glycine-rich peptides (and peptide-mimics such as peptoids) that both left and right regions of the Ramachandran plot; this is because, for such backbones, each  $-1/2$ -sloping slide of the Ramachandran plot may intersect more than one relevant region of the Ramachandran plot, which would put two structurally disparate regions within the Ramachandran plot close in  $\mathcal{R}$ -space. The signed Ramachandran plot  $\mathcal{R}_S$  minimizes the probability of this happening. However, very few residues within structural databases occupy the right side of the Ramachandran plot (3.5%), which means that signed Ramachandran plots would only be useful in special cases (and possibly for IDPs). For this reason, we will proceed below with a focus on the more relevant Ramachandran number  $\mathcal{R}$ .

## REFERENCES

- 315 **Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P.** 2002. Molecular biology of the cell. new york: Garland science; 2002. *Classic textbook now in its 5th Edition* .
- 316 **Baruah A, Rani P, Biswas P.** 2015. Conformational entropy of intrinsically disordered proteins from amino acid triads. *Scientific reports* 5.
- 317 **Beck DA, Alonso DO, Inoyama D, Daggett V.** 2008. The intrinsic conformational propensities of the 20 naturally occurring amino acids and reflection of these propensities in proteins. *Proceedings of the National Academy of Sciences* **105**(34):12259–12264.
- 318 **Berg JM, Tymoczko JL, Stryer L.** 2010. *Biochemistry, International Edition*. WH Freeman & Co., New York, 7 edition.
- 319 **Dunker A, Babu M, Barbar E, Blackledge M, Bondos S, Dosztányi Z, Dyson H, Forman-Kay J, Fuxreiter M, Gsponer J, Han KH, Jones D, Longhi S, Metallo S, Nishikawa K, Nussinov R, Obradovic Z, Pappu R, Rost B, Selenko P, Subramaniam V, Sussman J, Tompa P, Uversky V.** 2013. What's in a name? why these proteins are intrinsically disordered? *Intrinsically Disordered Proteins* **1**:e24157.
- 320 **Espinosa-Fonseca LM.** 2009. Reconciling binding mechanisms of intrinsically disordered proteins. *Biochemical and biophysical research communications* **382**(3):479–482.
- 321 **Fink AL.** 2005. Natively unfolded proteins. *Curr Opin Struct Biol* **15**(1):35–41.
- 322 **Geist L, Henen MA, Haiderer S, Schwarz TC, Kurzbach D, Zawadzka-Kazimierczuk A, Saxena**

- 333 S, Żerko S, Koźmiński W, Hinderberger D, et al. 2013. Protonation-dependent conformational  
334 variability of intrinsically disordered proteins. *Protein Science* **22**(9):1196–1205.
- 335 **Gunasekaran K, Nagarajaram H, Ramakrishnan C, Balaram P. 1998.** Stereochemical punctuation  
336 marks in protein structures: glycine and proline containing helix stop signals. *Journal of molecular  
337 biology* **275**(5):917–932.
- 338 **Ho BK, Brasseur R. 2005.** The ramachandran plots of glycine and pre-proline. *BMC structural biology*  
339 **5**(1):1.
- 340 **Hooft RW, Sander C, Vriend G. 1997.** Objectively judging the quality of a protein structure from a  
341 ramachandran plot. *Computer applications in the biosciences: CABIOS* **13**(4):425–430.
- 342 **Kosol S, Contreras-Martos S, Cedeño C, Tompa P. 2013.** Structural characterization of intrinsically  
343 disordered proteins by nmr spectroscopy. *Molecules* **18**(9):10802–10828.
- 344 **Laskowski RA. 2003.** Structural quality assurance. *Structural Bioinformatics, Volume 44* pages 273–303.
- 345 **Laskowski RA, MacArthur MW, Moss DS, Thornton JM. 1993.** Procheck: a program to check the  
346 stereochemical quality of protein structures. *Journal of applied crystallography* **26**(2):283–291.
- 347 **Mannige RV. 2014.** Dynamic new world: Refining our view of protein structure, function and evolution.  
348 *Proteomes* **2**(1):128–153.
- 349 **Mannige RV, Haxton TK, Proulx C, Robertson EJ, Battigelli A, Butterfoss GL, Zuckermann RN,  
350 Whitelam S. 2015.** Peptoid nanosheets exhibit a new secondary structure motif. *Nature* **526**:415–420.
- 351 **Mannige RV, Kundu J, Whitelam S. 2016.** The Ramachandran number: an order parameter for protein  
352 geometry. *PLoS One* **11**(8):e0160023.
- 353 **Midic U, Oldfield CJ, Dunker AK, Obradovic Z, Uversky VN. 2009.** Protein disorder in the human  
354 diseasesome: unfoldomics of human genetic diseases. *BMC Genomics* **10 Suppl 1**:S12. doi:10.1186/  
355 1471-2164-10-S1-S12.
- 356 **Momen R, Azizi A, Wang L, Yang P, Xu T, Kirk SR, Li W, Manzhos S, Jenkins S. 2017.** The role  
357 of weak interactions in characterizing peptide folding preferences using a qtaim interpretation of the  
358 ramachandran plot ( $\phi$ - $\psi$ ). *International Journal of Quantum Chemistry* .
- 359 **Ramachandran G, Ramakrishnan C, Sasisekharan V. 1963.** Stereochemistry of polypeptide chain  
360 configurations. *Journal of molecular biology* **7**(1):95–99.
- 361 **Sibille N, Bernado P. 2012.** Structural characterization of intrinsically disordered proteins by the  
362 combined use of nmr and sachs. *Biochemical society transactions* **40**(5):955–962.
- 363 **Subramanian E. 2001.** Gn ramachandran. *Nature Structural & Molecular Biology* **8**(6):489–491.
- 364 **Tompa P. 2011.** Unstructural biology coming of age. *Curr Opin Struct Biol* **21**(3):419–425. doi:  
365 10.1016/j.sbi.2011.03.012.
- 366 **Uversky VN. 2003.** Protein folding revisited. a polypeptide chain at the folding-misfolding-nonfolding  
367 cross-roads: which way to go? *Cell Mol Life Sci* **60**(9):1852–1871.
- 368 **Uversky VN, Dunker AK. 2010.** Understanding protein non-folding. *Biochim Biophys Acta*  
369 **1804**(6):1231–1264.