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

Ranjan V. Mannige*

* **ranjanmannige@gmail.com**

ABSTRACT

Protein backbones display complicated structures that often undergo numerous types of structural transformations. Due to the large number of structural degrees of freedom available to a backbone, it is often difficult to assess exactly where and how regions of a protein structure undergo structural transformation. This large structural phase makes it hard to survey new structural data, such as molecular dynamics trajectories or NMR-derived structural ensembles. This report discusses the Ramachandran number $\mathcal{R}$ as a residue-level structural metric that could simply the life of anyone contending with large numbers of structural data associated with protein backbones. In particular, this report 1) presents a new tool – BACKMAP– that can be universally installaed using '> pip install backmap', 2) discusses a much simpler closed form of $\mathcal{R}$ that makes it more easy to calculate, 4) introduces a signed Ramachandran number ($\mathcal{R}_S$) for achiral peptide backbones, and 3) shows how $\mathcal{R}$ dramatically reduces the dimensionality of the protein backbone, thereby making it ideal for simultaneously interrogating large number of protein structures. In short, $\mathcal{R}$ is a simple and succinct descriptor of protein backbones and their dynamics.

INTRODUCTION

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

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

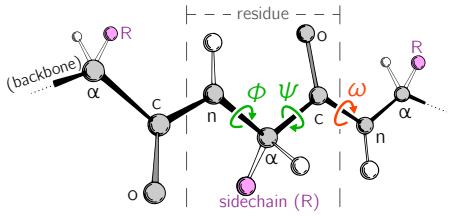


Figure 1. Backbone conformational degrees of freedom dominantly depend on the dihedral angles ϕ and ψ (green), and to a smaller degree depend on the third dihedral angle (ω; red) as well as bond lengths and angles (unmarked).

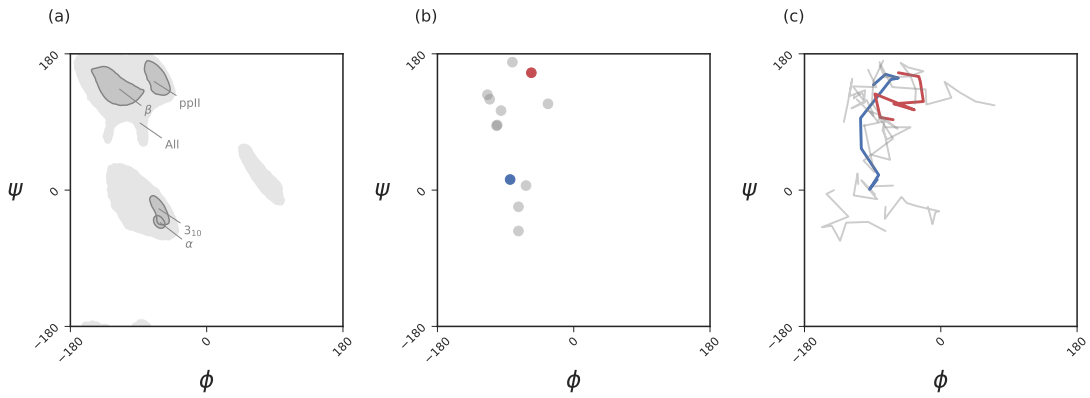


Figure 2. While the Ramachandran plot is useful for getting a *qualitative* sense of peptide backbone structure (a, c), it is not a convenient representation for exploring peptide backbone dynamics (c).
Secondary structure keys used here and throughout the document:
'α' – α-helix, '3_{10}' – 3_{10}-helix, 'β' – β-sheet/extension, 'ppII' – polyproline II helix.

by darkly shaded bonds. Ramachandran *et al.* (1963) had recognized that the backbone conformational degrees of freedom available to an amino acid (residue) i is almost completely described by only two dihedral angles: ϕ_i and ψ_i (Fig. 1, green arrows). Today, protein structures described in context of the two-dimensional (ϕ, ψ)-space are called Ramachandran plots.

The Ramachandran plot is recognized as a powerful tool for two reasons: 1) it serves as a map for structural 'correctness' (Laskowski *et al.*, 1993; Hooft *et al.*, 1997; Laskowski, 2003), since many regions within the Ramachandran plot space are energetically not permitted (Momen *et al.*, 2017); and 2) it provides a qualitative snapshot of the structure of a protein (Berg *et al.*, 2010; Alberts *et al.*, 2002; Subramanian, 2001). For example, particular regions within the Ramachandran plot indicate the presence of particular secondary locally-ordered structures such as the α-helix and β-sheet (see Fig. 2a).

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

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

It has recently been shown that the two Ramachandran backbone parameters (ϕ,ψ) may be conveniently combined into a single number – the Ramachandran *number* $[\mathcal{R}(\phi,\psi)$ or simply $\mathcal{R}]$ – with little loss of information (Mannige *et al.*, 2016). In a previous report, detailed discussions were provided regarding the reasons behind and derivation of $\mathcal{R}$ (Mannige *et al.*, 2016). This report provides a simpler version of the equation previously published (Mannige *et al.*, 2016), and further discusses how $\mathcal{R}$ may be used to provide information about protein ensembles and trajectories. Finally, we introduce a software

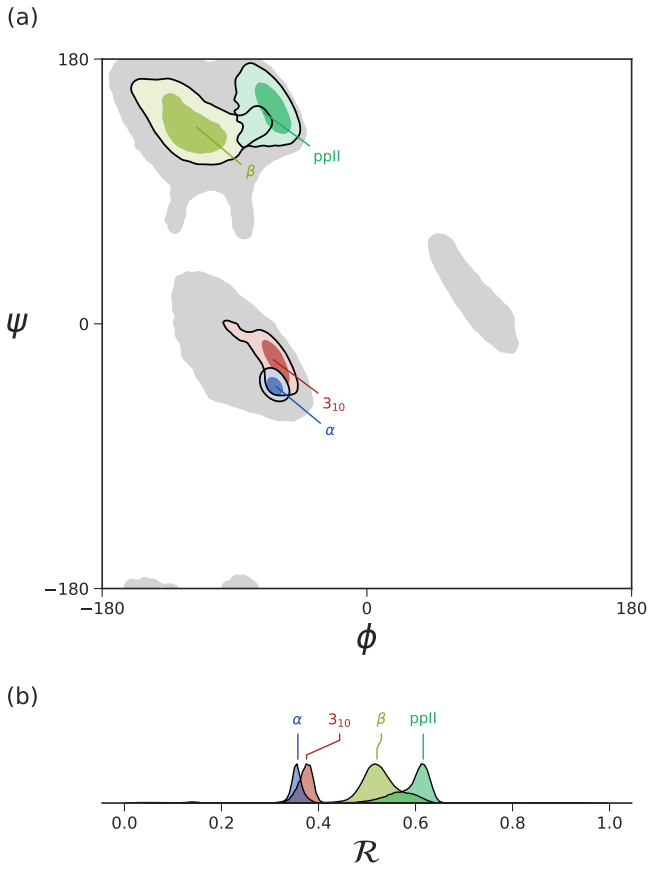


Figure 3. The distribution of dominant regular secondary sctuctures are shown in $[\phi, \psi]$-space (a) and in $\mathcal{R}$-space (b). Both Ramachandran plots (a) and Ramachandran 'lines' (b) show equivalent resolution of secondary structure , allowing for a more compact representation of Ramachandran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

package – BACKMAP– that can be used by to produce MAPs that describe the behavior of a protein backbone within user-inputted conformations, structural ensembles and trajectories. This package is presently available on GitHub (`https://github.com/ranjanmannige/BackMAP`).

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

The Ramachandran number is both an idea and an equation. Conceptually, the Ramachandran number ($\mathcal{R}$) is any closed form that collapses the dihedral angles ϕ and ψ into one structurally meaningful number (Mannige *et al.*, 2016). Mannige *et al.* (2016) presented a version of the Ramachandran number that was complicated in closed form, threby reducing its utility. Here, a much more simplified version of the Ramachandran number is introduced. Section 1.1 shows how this simplified form was derived from the original closed form (Eqns. 4 and 5).

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°, the most general and accurate equation for the Ramachandran number is

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

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

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

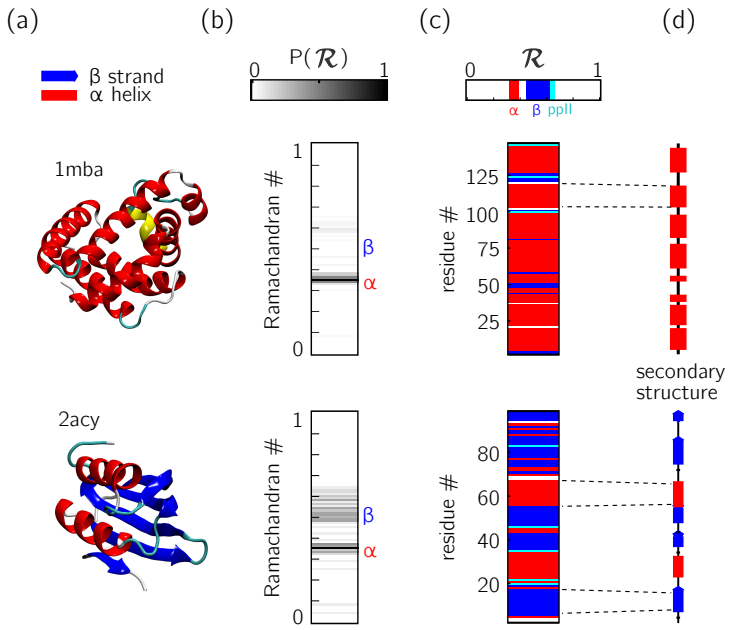


Figure 4. Two types of $\mathcal{R}$-codes. Digesting protein structures (a) using $\mathcal{R}$ numbers either as histograms (b) or per-residue codes (c) allow for compact representations of salient structural features. For example, a single glance at the histograms indicate that protein 1mba is likely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

As evident in Fig. 3, the distributions within the Ramachandran plot are faithfully reflected in corresponding distributions within Ramachandran number space. This paper shows how the Ramachandran 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 (ϕ, ψ). Say we have an N-residue peptide. Then, switching from (ϕ, ψ) to $\mathcal{R}$ *appears* to only reduce the number of variables from $2N$ to N, and hense by half. However, (ϕ, ψ) values are *coupled*, i.e., for any N-length peptide, any ordering of $[\phi_1, \phi_2, \ldots, \phi_N, \psi_1, \psi_2, \ldots, \psi_N]$ can not describe the structure, it is only *pairs* – $[(\phi_1, \psi_1), (\phi_2, \psi_2), \ldots, (\phi_N, \psi_N)]$ – that can. Therefore, we must think of switching from (ϕ, ψ)-space to $\mathcal{R}$-space as a switch in structure space per residue from N two-tuples (ϕ_i, ψ_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/columnarly, consititue 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).

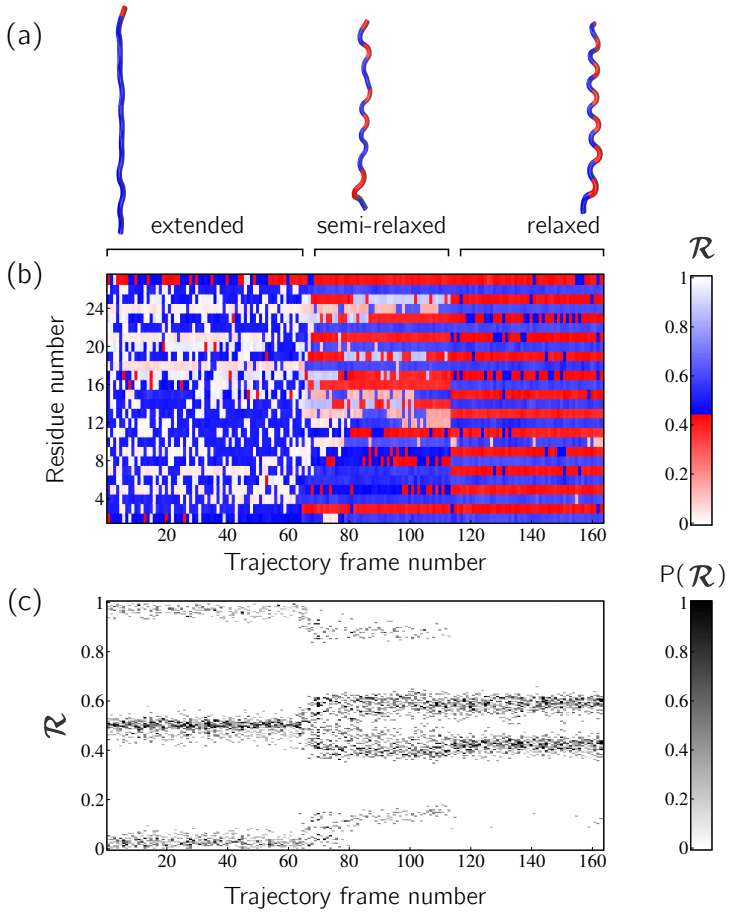


Figure 5. Stacked $\mathcal{R}$-codes provide useful information at a glance.

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 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 reseased (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 ϕ and ψ 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 α-helical regions. Due to the two-dimensionality

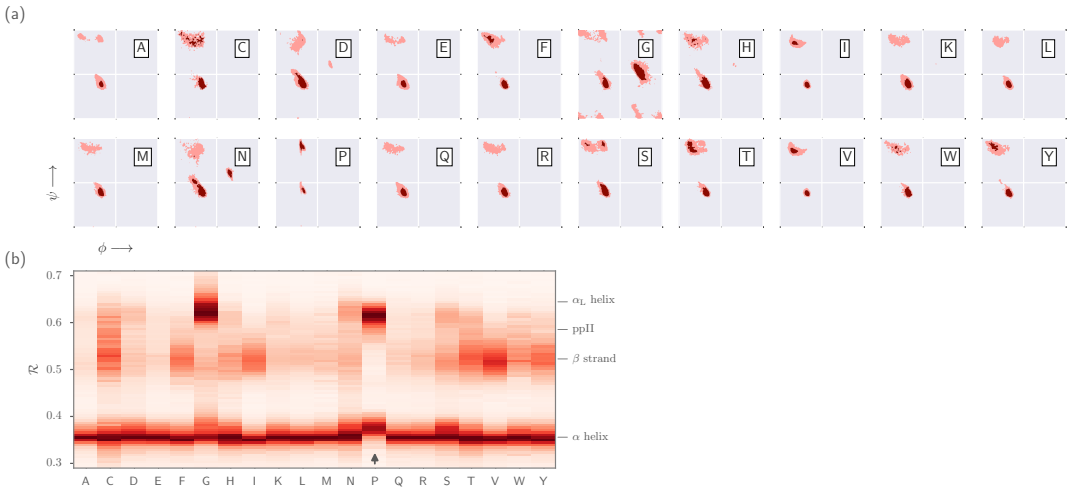


Figure 6. Ramachandran lines are stackable – Part I. Panel (a) shows the per-amino acid backbone behavior of an average protein found in the protein databank (PDB). While these plots are useful, it is difficult to compare such plots. For example, it is hard to pick out the change in the α-helial region of the proline plot (P). However, when we convert Ramachandran plots to Ramachanran *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 α-helix region, which is not evident to an untrained eye in (a).

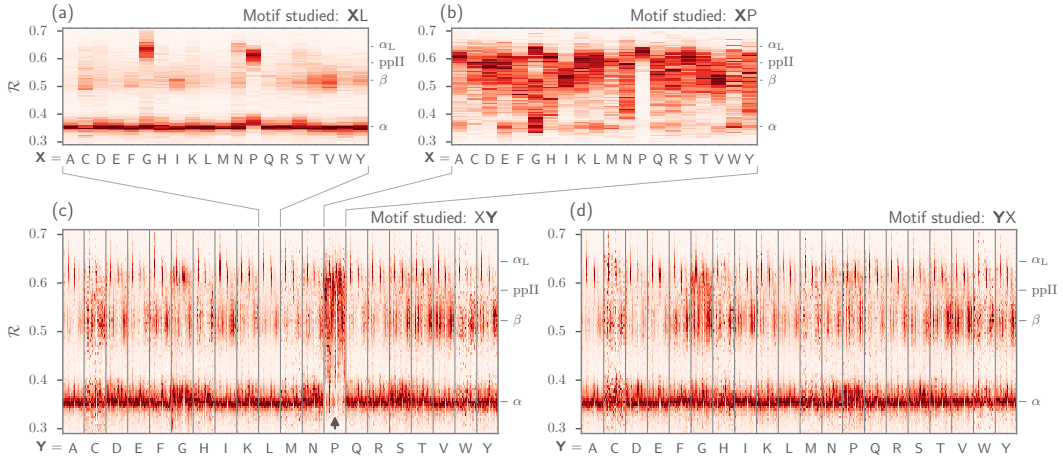


Figure 7. Ramachandran lines are stackable – Part II. Similar to Fig. 6b, Panel (a) represents the behavior of an amino acid 'Y' situated *before* a leucine (assuming that we are reading a sequence from the N terminal to the C terminal). Panel (b) similarly represents the behavior of specific amino acids situated before a proline. While residues preceeding a leucine behave similarly to their average behavior (Fig. 6a), most residues preceeding prolines appear to be enriched in structures that change 'direction' or backbone chirality ($\mathcal{R} > 0.5$). 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 preceeding 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 (20×20) distinct Ramachandran plots to compare.

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 arrow).

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

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

USING THE BACKMAP PYTHON MODULE

Installation

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

Usage

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

In-script usage: first simple test

The simplest test would be to generate Ramachandran numbers from (ϕ, ψ) pairs:

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

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

As seen above, the generation of Ramachandran numbers from (ϕ, ψ) pairs is simple. However, greating MAPs – Multi-Angle Pictures of protein backbones – requires a few more steps (present as a test in the downloadable module):

1. **Select and read a protein PDB structure**
   Each trajectory frame must be a set of legitimate protein databank "ATOM" records separated by "MODEL" keywords.

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

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

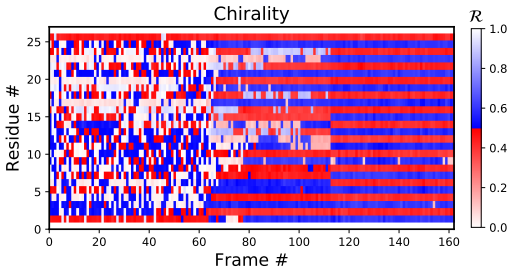
2. **Select color scheme (color map)**
   In addition to custom colormaps listed in the next section, one can also use standardly available at matplotlib.org (e.g., '`Reds`' or '`Reds_r`').

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

3. **Draw per-chain MAPs**

```
# Grouping by chain                                                          6
grouped_data = backmap.group_by(data,group_by='chain',                       7
                                columns_to_return=['model','resid','R'])     8
for chain in grouped_data.keys(): # Going through each chain                 9
  # Getting the X,Y,Z values for each entry                                  10
  models, residues, Rs = grouped_data[chain]                                 11
  # Finally, creating (but not showing) the graph                            12
  backmap.draw_xyz(X = models   ,      Y = residues   ,     Z = Rs           13
             ,xlabel ='Frame #', ylabel ="Residue #",zlabel ='$\mathcal{R}$' 14
               ,cmap = cmap    ,  title = "Chain: '"+chain+"'"               15
               ,vmin=0,vmax=1)                                               16
  # Now, we display the graph:                                               17
  plt.show() # ... one can also use plt.savefig() to save to file            18
```

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:

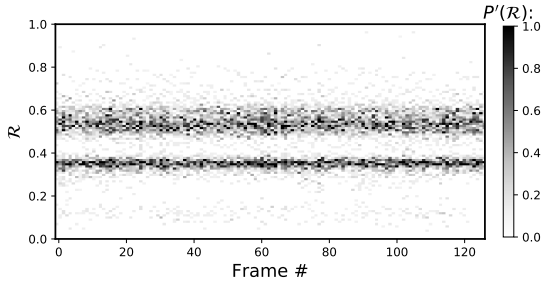


In-script usage: Creating custom graphs

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

```
for chain in grouped_data.keys():                                            10
  models, residues, Rs = grouped_data[chain]                                 11
                                                                             12
  'Begin custom code'                                                        13
  X = []; Y=[]; Z=[]; # Will set X=model, Y=R, Z=P(R)                        14
  # Bundling the three lists into one 2d array                               15
  new_data =  np.array(zip(models,residues,Rs))                              16
  # Getting all R values, model by model                                     17
  for m in sorted(set(new_data[:,0])): # column 0 is the model column        18
    # Getting all Rs for that model #                                        19
    current_rs = new_data[np.where(new_data[:,0]==m)][:,2] # column 2 contains R 20
    # Getting the histogram                                                  21
    a,b = np.histogram(current_rs,bins=np.arange(0,1.01,0.01))               22
    max_count = float(np.max(a))                                             23
    for i in range(len(a)):                                                  24
      X.append(m); Y.append((b[i]+b[i+1])/2.0); Z.append(a[i]/float(np.sum(a)));  25
  'End custom code'                                                          26
                                                                             27
  # Finally, creating (but not showing) the graph                            28
  draw_xyz(X = X       ,      Y = Y   ,                Z = Z                 29
     ,xlabel ='Frame #', ylabel ="$\mathcal{R}$",zlabel ="$P'(\mathcal{R})$" 30
     ,cmap = 'Greys', ylim=[0,1])                                            31
  plt.yticks(np.arange(0,1.00001,0.2))                                       32
  # Now, we display the graph:                                               33
  plt.show() # ... one can also use plt.savefig() to save to file            34
```

The code above results in the following graph:



In-script usage: Available color schemes (CMAPs)

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

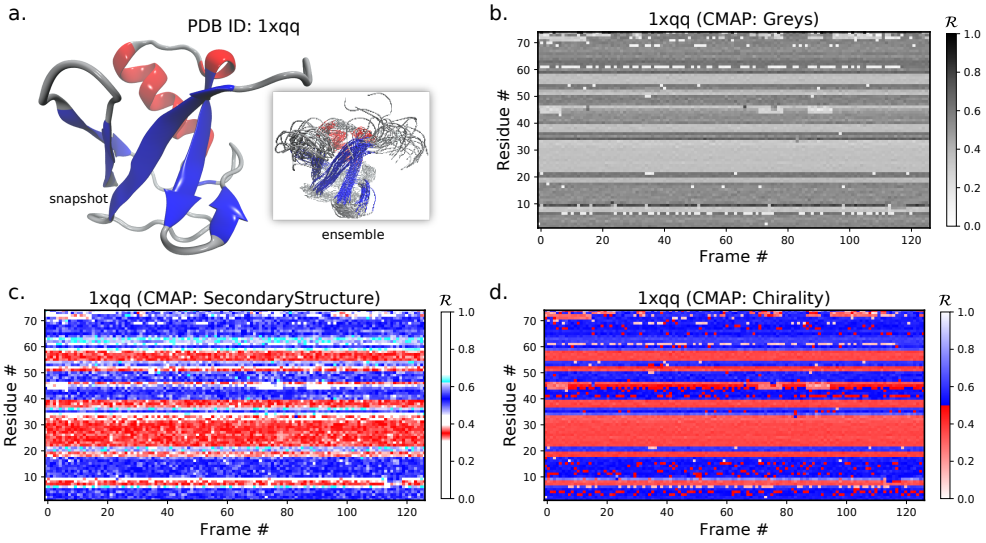


Figure 8. A protein ensemble (a) along with some MAPs colored with different themes (b-d). Panels (c) and (d) are provided by the BACKMAP module. In Panel (a), β-sheets are shown in blue and all helices are shown in red.

Stand Alone Usage

BACKMAP can be used as a stand along package by running '> python -m backmap -pdb <pdb_dir_or_file>'. The sectons below describes the expected outputs and how they may be interpreted.

Stand Alone Example I: A Stable Protein

Panels (b) through (f) of Fig. 9 below were created by running '> python -m backmap ./tests/pdbs/1xqq.pdb' (Panel (b) was created using VMD). These graphs indicates that protein 1xqq describes a conformationally stable protein.

In particular, each column in Panel (b) describes the histogram in Ramachandran number (R) space for a single model/timeframe. These histograms show the presence of both α-helices (at $\mathcal{R} \approx 0.34$) and β-sheets (at $\mathcal{R} \approx 0.52$). Additionally, Panels (c) and (d) describe per-residue conformational plots

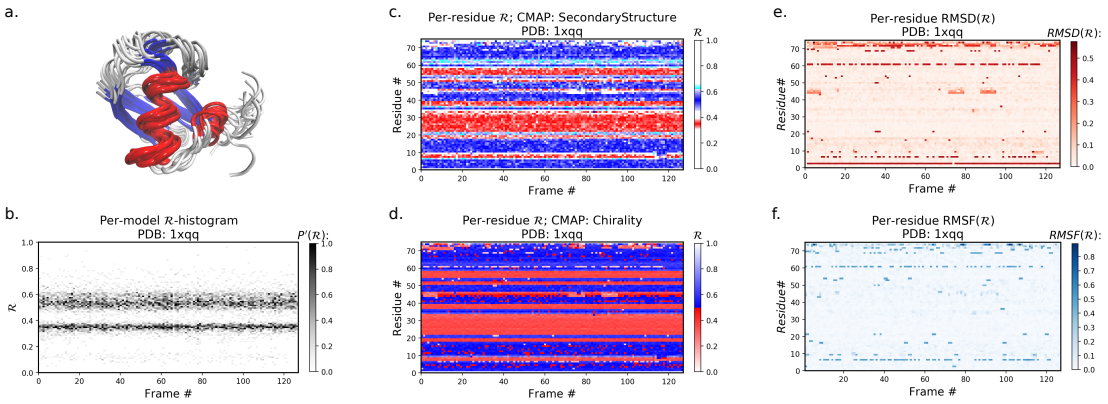


Figure 9. Protein 1xqq describes a stable protein.

(colored by two different metrics or CMAPs), which show that most of the protein backbone remains relatively stable over 'time' (e.g., few fluctuations in state or 'color' are evident over frame #). Finally, Panel (e) describes the extent towards which a single residue's state has deviated from the first frame, and Panel (f) describes the extent towards which a single residue's state has deviated from its state in the previous frame. Both these graphs, show that this protein is relatively conformationally stable.

Stand Alone Example II: An Intrinsically Disrodered Protein

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

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

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

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

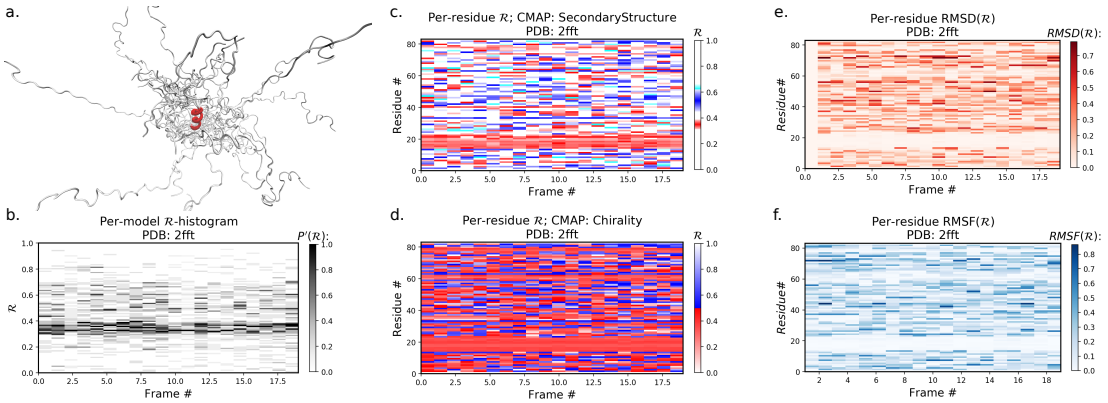


Figure 10. Protein 2fft describes an intrinsically disordered protein.

A signed Ramachandran number for 'misbehaving' backbones

The Ramachandran number increases in value from the bottom left of the Ramachandran plot to the top right in sweeps that are parallel to the negative sloping diagonal. As discussed in (Mannige *et al.*, 2016),

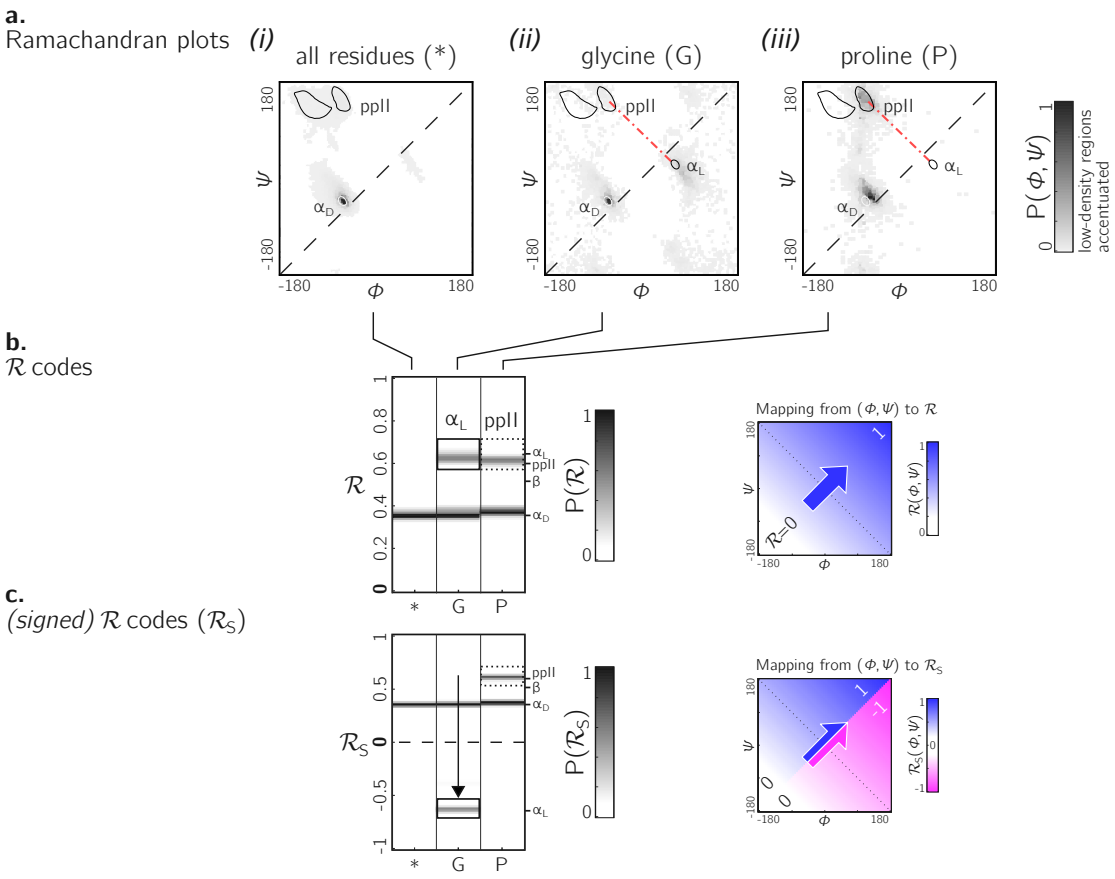


Figure 11. Signed $\mathcal{R}$s are required for non-chiral backbones. While the backbones of most amino acids occupy the topc of the positively sloped diagonal (dashed in b), non chiral amino acids such as Glycines (or their N-substituted variants – peptoids) display no such preference, which causes distinct secondary structures that lie on the same 'sweep' to be localized at similar regions in $\mathcal{R}$ (e.g., in b, polyproline-II and α_D helices both localize at $\mathcal{R} \approx 0.6$). However, a signed Ramachandran number ($\mathcal{R}_S$) solves this overlap by multiplying those $\mathcal{R}$'s derived from backbones with $\phi > \psi$ by -1. This extra resolution is evident available by the separation of polyproline-II and α_D helices (c). The mapping of (ϕ, ψ) to $\mathcal{R}$ and $\mathcal{R}_S$ are shown to the right of each respective $\mathcal{R}$-plot (b,c).

this method of mapping a two-dimensional space into one number is still structurally meaningful and descriptive because 1) most structural features of the protein backbone – e.g. radius of gyration (Mannige *et al.*, 2016), end-to-end distance (Mannige *et al.*, 2016), and chirality (Mannige, 2017) – vary little along lines parallel to the negatively-sloping diagonal (called $-$ve lines), and 2) most protein backbones display chiral centers and therefore predominantly appear on the top left region of the Ramachandran plot (above the dashed diagonal in Fig. 11a-*(i)*).

However, not all backbones localize in only one half of the Ramachandran plot. Particularly, among biologically relevant amino acids, glycine occupies both regions of the Ramachandran plot (Fig. 11a-*(ii)*; of note, the α_D helix region becomes relatively prominent). On the other hand, prolines are known to form polyproline-II helices (ppII in Fig. 11a-*(iii)*), which falls on almost the same 'sweep' as glycine rich peptides (red dot-dashed line α_D). In situations where both prolines and glycines are abundant, the Ramachandran number ($\mathcal{R}$) would fail to distinguish α_D from ppII (Fig. 11b).

To accomodate the situation where achiral backbones are expected (eg., if peptoids or polygycines are being studied), an additional Ramachandran number – the *signed* Ramachandran number $\mathcal{R}_S$ – is introduced here. $\mathcal{R}_S$ 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.,

$$\mathcal{R}_S = \begin{cases} \mathcal{R} & \text{, if } \psi \geq \phi \\ \mathcal{R} \times -1 & \text{, if } \psi < \phi \end{cases} \tag{3}$$

An example of the utility of $\mathcal{R}_S$, Fig. 11b shows that $\mathcal{R}_S$ easily distinguishes α_D from ppII.

Note that, while useful, would be important in very limited scenarios, as amino acids in the PDB occupy the lower-right side of the Ramachandran plot fewer than 3.5% of the time (mostly gue to glycines and IDPs).

CONCLUSION

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

ACKNOWLEDGMENTS

During the development of this paper, RVM was partially supported by the Defense Threat Reduction Agency under contract no. IACRO-B0845281. RVM thanks Alana Canfield Mannige for her critique. The notion of the signed Ramachandran number emerged from discussions with Joyjit Kundu and Stephen Whitelam while at the Molecular Foundry at Lawrence Berkeley National Laboratory (LBNL). This work was partially done at the Molecular Foundry at LBNL, supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

1 APPENDIX

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

This section will derive the simplified Ramachandran number presented in this paper from the more complicated looking Ramachandran number introduced previously (Mannige *et al.*, 2016).

Assuming the bounds $\phi, \psi \in [-180°, 180°)$, and the range λ equals 360°, the previously described Ramachandran number takes the form

$$\mathcal{R}(\phi,\psi) \equiv \frac{R_{\mathbb{Z}}(\phi,\psi) - R_{\mathbb{Z}}(\phi_{\min},\phi_{\min})}{R_{\mathbb{Z}}(\phi_{\max},\phi_{\max}) - R_{\mathbb{Z}}(\phi_{\min},\phi_{\min})}, \tag{4}$$

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

$$R_{\mathbb{Z}}(\phi,\psi) = \left\lfloor (\phi - \psi + \lambda)\sigma/\sqrt{2} \right\rceil + \left\lfloor \sqrt{2}\lambda\sigma \right\rfloor \left\lfloor (\phi + \psi + \lambda)\sigma/\sqrt{2} \right\rceil. \tag{5}$$

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

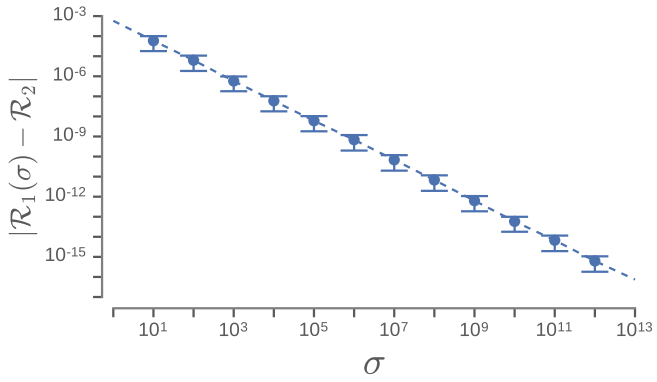


Figure 12. The increase in the acccuracy measure (σ) for the original Ramachandran number (Eqn. 5) results in values that tend towards the new Ramachandran number proposed in this paper (Eqn. 2).

However useful Eqn. 4 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. 4 as σ tends towards ∞. In particular, when $\sigma \to \infty$, Eqn. 4 becomes

$$\mathcal{R}(\phi,\psi) = \lim_{\sigma\to\infty} \mathcal{R}(\phi,\psi) = \frac{\phi+\psi+\lambda}{2\lambda} = \frac{\phi+\psi+2\pi}{4\pi}. \tag{6}$$

Conformation of this limit is shown numerically in Fig. 12. Since larger σs indicate higher accuracy, $\lim_{\sigma\to\infty} \mathcal{R}(\phi,\psi)$ represents an exact representation of the Ramachandran number. Using this closed form, this report shows how both static structural features and complex structural transitions may be identified with the help of Ramachandran number-derived plots.

1.2 Other frames of reference

The Ramachandran number shown in Eqn. 6 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

$$\mathcal{R}(\phi,\psi) \equiv \frac{\phi+\psi-(\psi_{\min}+\psi_{\min})}{(\psi_{\max}+\psi_{\max})-(\psi_{\min}+\psi_{\min})}. \tag{7}$$

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

$$\mathcal{R}(\phi,\psi)_{\phi,\psi\in[0,2\pi)} = \frac{\phi+\psi}{4\pi}. \tag{8}$$

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

REFERENCES

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* .

Baruah A, Rani P, Biswas P. 2015. Conformational entropy of intrinsically disordered proteins from amino acid triads. *Scientific reports* **5**.

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.

Berg JM, Tymoczko JL, Stryer L. 2010. *Biochemistry, International Edition*. WH Freeman & Co., New York, 7 edition.

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.

Espinoza-Fonseca LM. **2009**. Reconciling binding mechanisms of intrinsically disordered proteins. *Biochemical and biophysical research communications* **382(3)**:479–482.

Fink AL. **2005**. Natively unfolded proteins. *Curr Opin Struct Biol* **15(1)**:35–41.

Geist L, Henen MA, Haiderer S, Schwarz TC, Kurzbach D, Zawadzka-Kazimierczuk A, Saxena S, Żerko S, Koźmiński W, Hinderberger D, *et al.* **2013**. Protonation-dependent conformational variability of intrinsically disordered proteins. *Protein Science* **22(9)**:1196–1205.

Gunasekaran K, Nagarajaram H, Ramakrishnan C, Balaram P. **1998**. Stereochemical punctuation marks in protein structures: glycine and proline containing helix stop signals. *Journal of molecular biology* **275(5)**:917–932.

Ho BK, Brasseur R. **2005**. The ramachandran plots of glycine and pre-proline. *BMC structural biology* **5(1)**:1.

Hooft RW, Sander C, Vriend G. **1997**. Objectively judging the quality of a protein structure from a ramachandran plot. *Computer applications in the biosciences: CABIOS* **13(4)**:425–430.

Kosol S, Contreras-Martos S, Cedeño C, Tompa P. **2013**. Structural characterization of intrinsically disordered proteins by nmr spectroscopy. *Molecules* **18(9)**:10802–10828.

Laskowski RA. **2003**. Structural quality assurance. *Structural Bioinformatics, Volume 44* pages 273–303.

Laskowski RA, MacArthur MW, Moss DS, Thornton JM. **1993**. Procheck: a program to check the stereochemical quality of protein structures. *Journal of applied crystallography* **26(2)**:283–291.

Mannige RV. **2014**. Dynamic new world: Refining our view of protein structure, function and evolution. *Proteomes* **2(1)**:128–153.

Mannige RV. **2017**. An exhaustive survey of regular peptide conformations using a new metric for backbone handedness (h). *PeerJ* **5**:e3327. ISSN 2167-8359. doi:10.7717/peerj.3327.

Mannige RV, Haxton TK, Proulx C, Robertson EJ, Battigelli A, Butterfoss GL, Zuckermann RN, Whitelam S. **2015**. Peptoid nanosheets exhibit a new secondary structure motif. *Nature* **526**:415–420.

Mannige RV, Kundu J, Whitelam S. **2016**. The Ramachandran number: an order parameter for protein geometry. *PLoS One* **11(8)**:e0160023.

Midic U, Oldfield CJ, Dunker AK, Obradovic Z, Uversky VN. **2009**. Protein disorder in the human diseasome: unfoldomics of human genetic diseases. *BMC Genomics* **10 Suppl 1**:S12. doi:10.1186/1471-2164-10-S1-S12.

Momen R, Azizi A, Wang L, Yang P, Xu T, Kirk SR, Li W, Manzhos S, Jenkins S. **2017**. The role of weak interactions in characterizing peptide folding preferences using a qtaim interpretation of the ramachandran plot (ϕ-ψ). *International Journal of Quantum Chemistry* .

Ramachandran G, Ramakrishnan C, Sasisekharan V. **1963**. Stereochemistry of polypeptide chain configurations. *Journal of molecular biology* **7(1)**:95–99.

Sibille N, Bernado P. **2012**. Structural characterization of intrinsically disordered proteins by the combined use of nmr and saxs. *Biochemical society transactions* **40(5)**:955–962.

Subramanian E. **2001**. Gn ramachandran. *Nature Structural & Molecular Biology* **8(6)**:489–491.

Tompa P. **2011**. Unstructural biology coming of age. *Curr Opin Struct Biol* **21(3)**:419–425. doi:10.1016/j.sbi.2011.03.012.

Uversky VN. **2003**. Protein folding revisited. a polypeptide chain at the folding-misfolding-nonfolding cross-roads: which way to go? *Cell Mol Life Sci* **60(9)**:1852–1871.

Uversky VN, Dunker AK. **2010**. Understanding protein non-folding. *Biochim Biophys Acta* **1804(6)**:1231–1264.