

INPUTS  $\rightarrow$  rname\_pdb (Input file), nres (number of residues)  
(sequence)

$\rightarrow$  \*tmp [Template file]

buildres

For  $i = 1 : nres$

call rname\_pdb(i)  $\rightarrow$  Identify  $i^{\text{th}}$  residue.

call filename = \*tmp  $\rightarrow$  ~~call~~  $i^{\text{th}}$  residue's template is called.

call backbone\_dihedral  $\rightarrow$  For each residue, their backbone Z-coordinates are initiated. ~~Zmat~~  
(backbone\_zmat)

call backbone\_zmat  $\rightarrow$  For each residue  
Zmatrix for the residue (Zmat\_res)  
and parent\_res are initialized.

call readres  $\rightarrow$  This will update Zmat\_res  
& parent\_res.

call build\_atom  $\rightarrow$  This creates Parent\_atom  
Zmat based. From Parent  
-atom,  
it creates Parent.

end.

For a given Parent & Zmat from buildres

xyzint  $\rightarrow$  calls cart\_pdb (Cartesian coordinates)  
& updates the Zmatrix.

Coord. pdb.

rname. pdb

Structure. h

build\_res

①

ALA ASP LEU PRO ALA PRO

↑  
ires

$n = nres$

→ template

→ It has  $\phi, \chi, \text{bond length, bond angle}$   
for a residue.

ires

Final backbone ( $\phi, \chi, \text{theta}$ )

Cart. pdb.

Atom	x	y	z	
i				→ Cart i
j				→ Cart j
k				→ Cart k
l				→ Cart l

Conversion

→

Parent\_res → Residue alone  
↓  
Parent\_atom → both residue & backbone

Parent

(This serves as a reference to update molecule)



buildres file is the main file. It

Input: rname.pdb  $\rightarrow$  This has the sequence information. (The order of residues)

A For loop in this file reads the  $i$ th residue. For each residue, backbone\_dihedral function update initialize phi, psi & omega array based on the index ires. This updation is only for the back bone. The size of phi, psi & omega array is equal to num of residues. nres is equal to number of residues. Before backbone\_dihedral is being called ires based on the amino acid sequence letter, the corresponding template file name is updated which is then loaded when readres file is called.

After backbone\_dihedral is called, backbone\_zmat is called. Here, based on the ires (residue number), zmat\_res matrix is initialized and also parent\_res array is updated. parent\_res is updated based on the number of atoms in previous side chain (sidecnt). For 1st residue, sidecnt is assigned to be zero.

template  
has bond  
length, bond  
angle, etc.  
positions of  
atoms in a residue

After

readres file opens the template & updates the `zmat_res` array (Z-matrix for the residues) and also it updates `parent_res`.

For `parent` the size of the `parent_res` array is  $H + \text{number of atoms in the residues}$ . This is because  $H$  atoms are corresponds to ~~zmat~~ back bone. It also calculates the number of atoms in a residue.

Finally `build_atom` function is called. This has the input for this functions are number of atoms in residues. <sup>nbuild (res)</sup> `nbuild` will take zero for the first residue, the value for the `nbuild` is updated in `built_atom`. After per residue, the value of the `nbuild` is equal to number of atom in the first residue + 4. Essentially this is equal to the number of atom in side chain & backbone. For each residue it keeps on increasing.

Parent atom is created in `built_atom`. This array updates all the atoms.

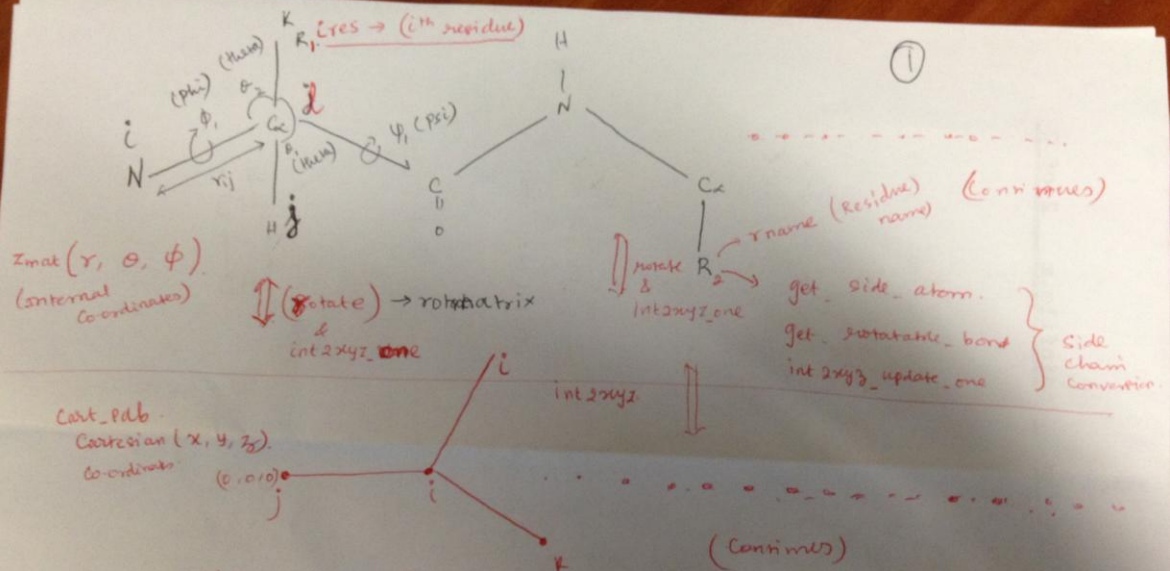
At any point on the protein sequence, Parent\_atom will give the current position of the atom.

Also here Zmat is created & updated from Zmat\_res which was updated based on the template file.

From Parent\_atom array, Parent array is created & updated for each residue.

Once Parent is updated, then xyz2int\_atom gets the zmat being xyz2int, bond angle, tors angle functions.



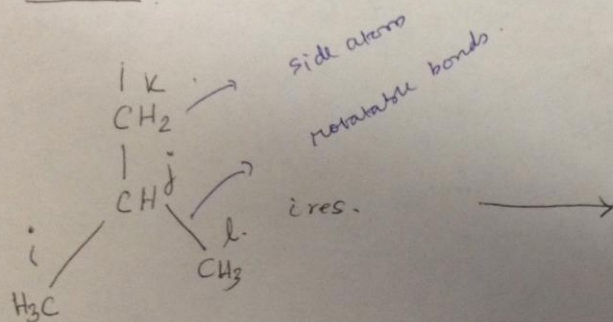


Input: totalrot → Product of all the rotamatrix.

- libname → load\_lib(libname) → loading library name.

Res	Rotamer	n (81)	n(1234)	P(81234)	P(8234/81)	chi1	chi2	chi3	chi4
ARG	1 1 1	568	2	0.04		55.4	79.7	62.4	82.3

## Residue



get\_side\_atoms.

get\_rotatable\_bonds



(This identifies which bond can be rotated in order to make the conformational change).

intxyz\_update\_simple, add\_side\_~~atom~~<sup>root</sup>.