Supporting Information for

LassoHTP: a High-throughput Computational Tool for Lasso Peptide Structure Construction and Modeling

Reecan J. Juarez,[5] Yaoyukun Jiang,[1] Matthew Tremblay,[1] Qianzhen Shao,[1] A. James Link,[7] and Zhongyue J. Yang[1-4,6]*

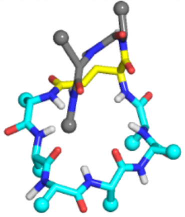
[1]*Department of Chemistry, Vanderbilt University, Nashville, Tennessee 37235, United States* [2]*Center for Structural Biology, Vanderbilt University, Nashville, Tennessee 37235, United States* [3]*Vanderbilt Institute of Chemical Biology, Vanderbilt University, Nashville, Tennessee 37235, United States* [4]*Data Science Institute, Vanderbilt University, Nashville, Tennessee 37235, United States* [5]*Chemical and Physical Biology Program, Vanderbilt University, Nashville, Tennessee 37235, United States* [6]*Department of Chemical and Biomolecular Engineering, Vanderbilt University, Nashville, Tennessee 37235, United States* [7]*Department of Chemical and Biological Engineering, Chemistry and Molecular Biology, Princeton University, 207 Hoyt Laboratory, Princeton, New Jersey 08544, United States*

Contents

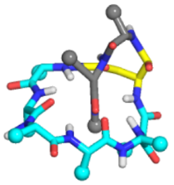
```
from Class_PDB import *

from Class_Conf import *
from seq_parser import *

def main():
    #Module 1: Scaffold Constructor
    #specify the annotated sequence and working directory
        seq, ring_length, upper_plug, wk_dir =
'GGAGHVPEYFVRGDTPISFYG', 8, 11, ''
        proto_lasso=construct_scaffold(seq, ring_length,
upper_plug, wk_dir)
        print(proto_lasso)
        PDB1=PDB(proto_lasso,wk_dir)
        #Module 2: Mutation
        sequence=seq_flags(seq, 8) #lasso seq
        print(PDB1.Add_MutaFlag(sequence))
        PDB1.PDB2PDBwLeap()
    #Module 3: Molecular Dynamics
    #use minimization to relax each mutated lasso
        PDB1.PDB2FF()
        PDB1.PDBMin()
    #run MD
        PDB1.rm_wat()
        PDB1.PDB2FF()
        PDB1.conf_prod['nstlim'] = 50500000 # Edit MD
configuration (see default in Class_Conf.py - Config.Amber)
        PDB1.PDBMD(tag='_mccJ25_RGD')

if __name__ == "__main__":
    main()
```

Figure S1. input_main.py: sample python code to run LassoHTP

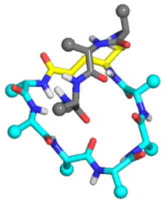
Glutamate Linker **Aspartate Linker** **Glutamate Linker** **Aspartate Linker**



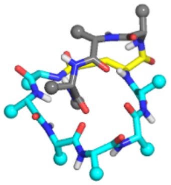
Ring size: 7
Loop size: 2



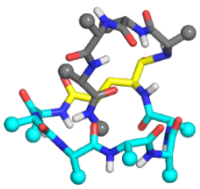
Ring size: 7
Loop size: 2



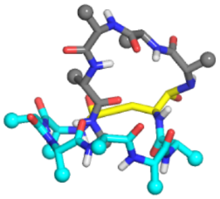
Ring size: 7
Loop size: 3



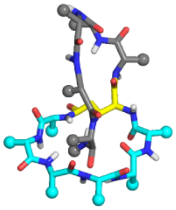
Ring size: 7
Loop size: 3



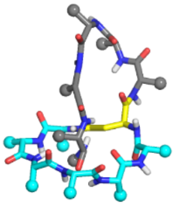
Ring size: 7
Loop size: 4



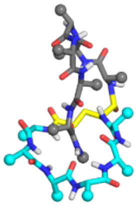
Ring size: 7
Loop size: 4



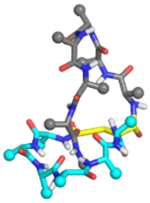
Ring size: 7
Loop size: 5



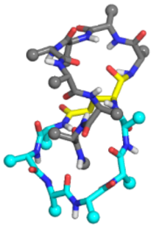
Ring size: 7
Loop size: 5



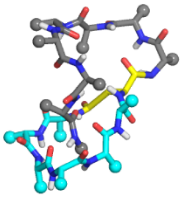
Ring size: 7
Loop size: 6



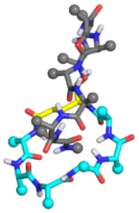
Ring size: 7
Loop size: 6



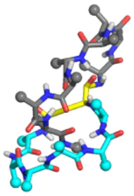
Ring size: 7
Loop size: 7



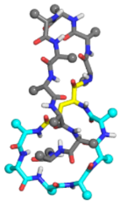
Ring size: 7
Loop size: 7



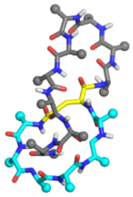
Ring size: 7
Loop size: 8



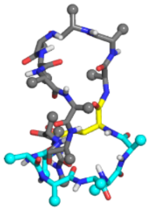
Ring size: 7
Loop size: 8



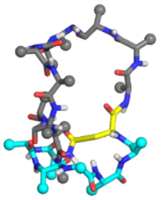
Ring size: 7
Loop size: 9



Ring size: 7
Loop size: 9

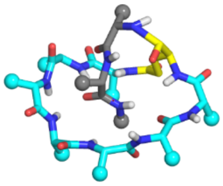


Ring size: 7
Loop size: 10

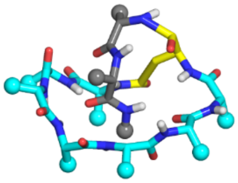


Ring size: 7
Loop size: 10

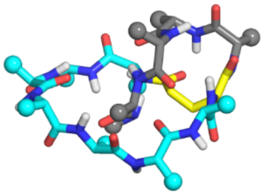
Glutamate Linker **Aspartate Linker**



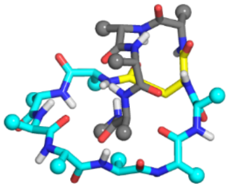
Ring size: 8
Loop size: 2



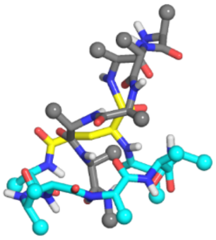
Ring size: 8
Loop size: 2



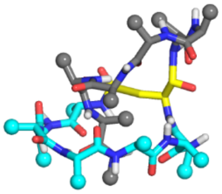
Ring size: 8
Loop size: 4



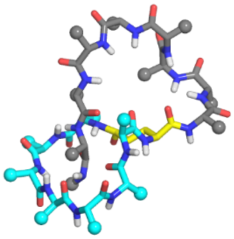
Ring size: 8
Loop size: 4



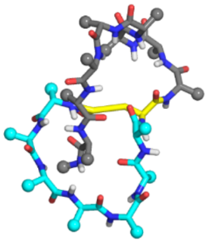
Ring size: 8
Loop size: 6



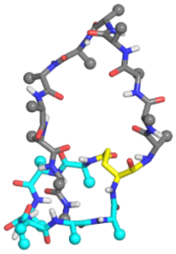
Ring size: 8
Loop size: 6



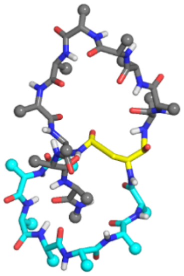
Ring size: 8
Loop size: 8



Ring size: 8
Loop size: 8

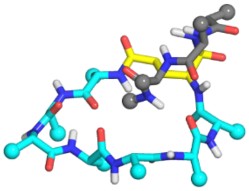


Ring size: 8
Loop size: 10

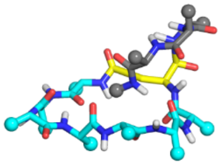


Ring size: 8
Loop size: 10

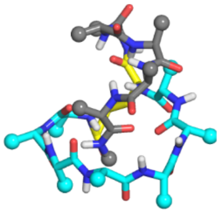
Glutamate Linker **Aspartate Linker**



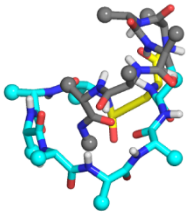
Ring size: 8
Loop size: 3



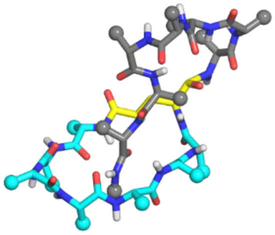
Ring size: 8
Loop size: 3



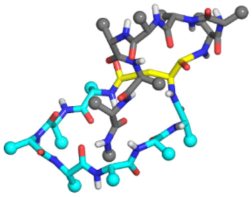
Ring size: 8
Loop size: 5



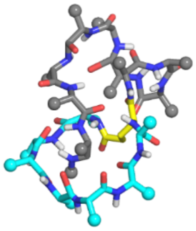
Ring size: 8
Loop size: 5



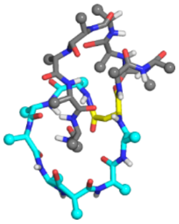
Ring size: 8
Loop size: 7



Ring size: 8
Loop size: 7

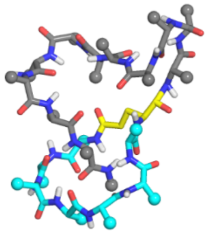


Ring size: 8
Loop size: 9
PDB: 6mw6

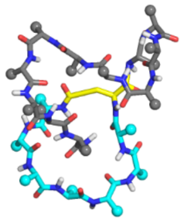


Ring size: 8
Loop size: 9
PDB: 6mw6

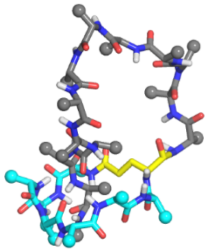
Glutamate Linker **Aspartate Linker** **Glutamate Linker** **Aspartate Linker**



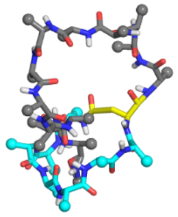
Ring size: 8
Loop size: 11
PDB: 2mmw



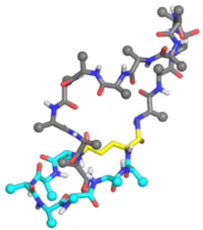
Ring size: 8
Loop size: 11
PDB: 2mmw



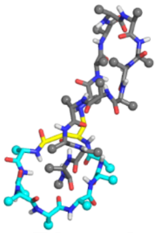
Ring size: 8
Loop size: 12



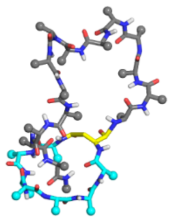
Ring size: 8
Loop size: 12



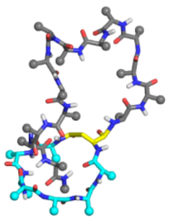
Ring size: 8
Loop size: 13



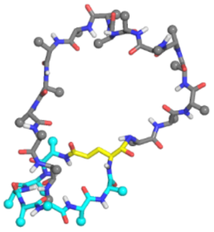
Ring size: 8
Loop size: 13



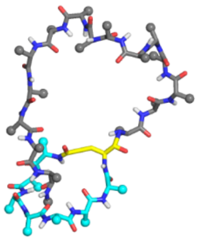
Ring size: 8
Loop size: 14



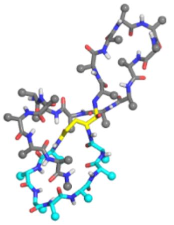
Ring size: 8
Loop size: 14



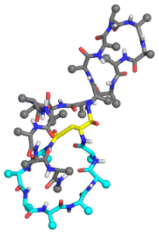
Ring size: 8
Loop size: 15



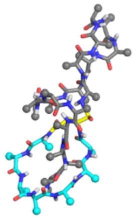
Ring size: 8
Loop size: 15



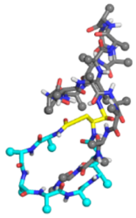
Ring size: 8
Loop size: 16



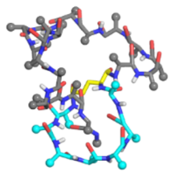
Ring size: 8
Loop size: 16



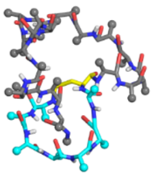
Ring size: 8
Loop size: 17



Ring size: 8
Loop size: 17

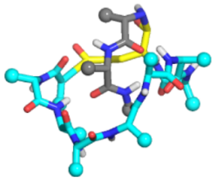


Ring size: 8
Loop size: 18
PDB: 6por

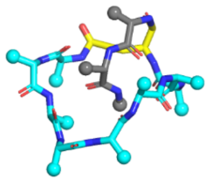


Ring size: 8
Loop size: 18
PDB: 6por

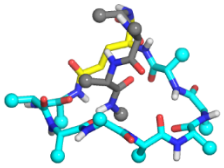
Glutamate Linker **Aspartate Linker** **Glutamate Linker** **Aspartate Linker**



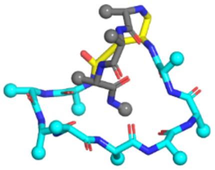
Ring size: 9
Loop size: 2



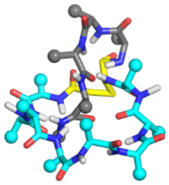
Ring size: 9
Loop size: 2



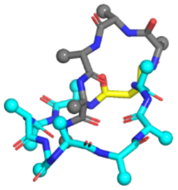
Ring size: 9
Loop size: 3



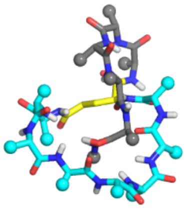
Ring size: 9
Loop size: 3



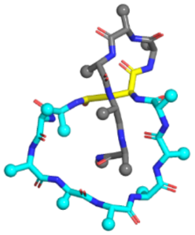
Ring size: 9
Loop size: 4
PDB: 5zcn



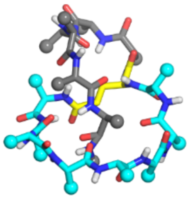
Ring size: 9
Loop size: 4
PDB: 5zcn



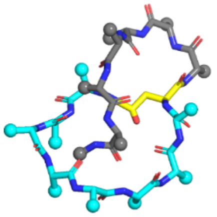
Ring size: 9
Loop size: 5
PDB: 5gvo



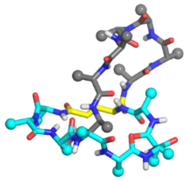
Ring size: 9
Loop size: 5
PDB: 5gvo



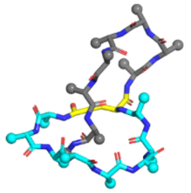
Ring size: 9
Loop size: 6
PDB: 5d9e



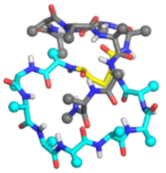
Ring size: 9
Loop size: 6
PDB: 5d9e



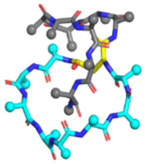
Ring size: 9
Loop size: 7
PDB: 2mlj



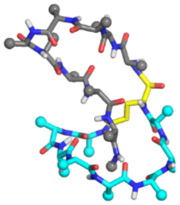
Ring size: 9
Loop size: 7
PDB: 2mlj



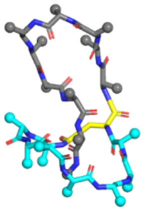
Ring size: 9
Loop size: 8



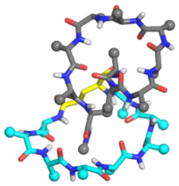
Ring size: 9
Loop size: 8



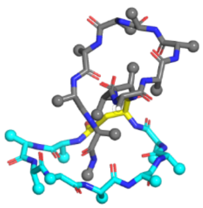
Ring size: 9
Loop size: 9



Ring size: 9
Loop size: 9



Ring size: 9
Loop size: 10



Ring size: 9
Loop size: 10

Figure S2. Scaffold library of scaffold constructor module. The library consists of 70 scaffold structures with 18 seven-membered ring structures, 34 eight-membered ring structures, and 18 nine-membered ring structures. For scaffolds that were constructed from experimentally-determined structures, the PDB ID is provided.

Text S1. seq_parser.py: parsing module for user input sequence
The main module, seq_parse, receives a lasso peptide sequence as input and indicates the portions of the sequence that belong to the ring, loop, and tail. It also indicates what type of isopeptide linker utilized. construct_scaffold uses seq_parse's output to select a lasso peptide scaffold with the appropriate ring and loop sizes. The tail extender submodule constructs a tail of an appropriate size. The resulting scaffold is shuttled by outfile_mover to the specified working directory.

```
import sys
sys.path.insert(0, '/usr/LassoHTP/lasso_extension')
import subprocess
import os
from lasso_peptide_gen import *

def get_cwd(directory):
    """get directory
    """

    if(not directory):
        cwd = os.getcwd()
    else:
        cwd = os.path.abspath(directory)

    return cwd

def seq_parse(seq:str, ring_len: int, upper_plug: int):

    sequence = seq
    seq_len = len(sequence)
    ring = ring_len
    loop = upper_plug
    isopeptide = ''
    tail_length = seq_len - ( ring + loop )

    for idx , char in enumerate(seq):
        idx += 1
        if idx == ring:
            isopeptide = 'e' if char in 'E' else 'd'

    return sequence, ring, loop, tail_length, isopeptide

def seq_flags(seq: str, ring_num: int):
    """generates flags from sequence
    """
    lis = []
    for idx, char in enumerate(seq):
        idx +=1
        if idx == ring_num: #acidic residue
            continue
        lis.append('A' + str(idx) + char)
    return lis

def construct_scaffold(seq:str, ring_len: int, upper_plug:int,
wk_dir):
    """constructs scaffold from given sequence
    """
    cwd = get_cwd(wk_dir)

    seq, ring, loop, tail_length, isopeptide = seq_parse(seq,
ring_len, upper_plug)
    outfile = f"{ring}{isopeptide}_{loop}_{tail_length}.pdb"
    lasso_peptide_gen(ring, loop, tail_length, isopeptide,
outfile)
    out_path = "lasso_extension/output_structures/" + outfile
    outfile_mover(out_path, cwd)

    return out_path

def outfile_mover(lasso,wk_dir):
    """moves outfile to working directory.
    """
    command = ['cp', lasso, wk_dir]
    subprocess.run(command)
```

Table S1. Force field parameters for isopeptide linkers. GLX represents glutamate and ASX represents aspartate. Highlighted entries with red font indicate modified forcefield parameters to accommodate the isopeptide bond for each respective modified residue.

GLX force field (ff14SB) parameters

Atom type	Mass	Atomic polarizability
N	14.01	0.53
H	1.008	0.161
CT	12.01	0.878
C	12.01	0.616
OH	16	0.465
HC	1.008	0.135
HO	1.008	0.135
H1	1.008	0.135

O	16	0.434

Bond	Harmonic force constant	Equilibrium bond length
C -NT	490	1.335
H -N	434	1.01
CT-N	337	1.449
C -CT	317	1.522
CT-H1	340	1.09
CT-CT	310	1.526
C -OH	450	1.364
HO-OH	553	0.96
CT-HC	340	1.09
C -O	570	1.229

Angle	Harmonic force constant	Equilibrium bond angle
O - C-N	80	122.9
C -NT-H	50	120
C -NT-CT	50	121.9
CX-C -NT	70	116.6
C -C -N	70	116
CT-C -C	63	117
N - C-OH	50	107.1
C - N- C	50	133.9
N -CT-HC	50	111
C -CT-N	63	110.1
H1-CT-N	50	109.5
CT-CT-N	80	109.7
CT-N -H	50	118.04
CT-C -OH	80	110
CT-CT-HC	50	109.5
CT-CT-CT	40	109.5
C -CT-H1	50	109.5
C -CT-CT	63	111.1
C -OH-HO	50	113
CT-CT-H1	50	109.5
HC-CT-HC	35	109.5
CT-C -O	80	120.4

C -CT-HC	50	109.5		
Non-bonded	**Coefficient of 12th power term**	**Coefficient of 6th power term**		
N	1.824	0.17		
H	0.6	0.0157		
CT	1.908	0.1094		
C	1.908	0.086		
OH	1.721	0.2104		
HC	1.487	0.0157		
HO	0	0		
H1	1.387	0.0157		
O	1.6612	0.21		
Improper	**Barrier height**	**Phase shift angle in torsional function**	**Periodicity of torsional barrier**	
CT-OH-C -OH	1.1	180	2	
C -CT-CT-HC	1.1	180	2	
Dihedral	**Torsional barrier division factor**	**Barrier height, divided by 2**	**Phase shift angle in torsional function**	**Periodicity of torsional barrier**
OH-C -CT-N	6	0	0	2
HC-CT-CT-N	9	1.4	0	3
CT-CT-CT-N	9	1.4	0	3
C -CT-N -H	6	0	0	2
H1-CT-N -H	6	0	0	2
CT-CT-N -H	6	0	0	2
CT-C -OH-HC	2	4.6	180	2
CT-C -OH-HO	2	4.6	180	2
CT-CT-CT-HC	1	0.16	0	3
C -CT-CT-CT	9	1.4	0	3
C -CT-CT-HC	9	1.4	0	3
OH-C -OH-HO	2	4.6	180	2
OH-C -OH-HC	2	4.6	180	2
OH-C -CT-H1	6	0	0	2
H1-CT-CT-HC	9	1.4	0	3

CT-CT-CT-H1	9	1.4	0	3
OH-C -CT-CT	6	0	0	2
O -C -CT-CT	6	0	0	2
HC-CT-CT-HC	1	0.15	0	3
O -C -CT-HC	1	0.8	0	-1
O -C -CT-HC	1	0	0	-2
O -C -CT-HC	1	0.08	180	3

ASX force field (ff14SB) parameters

Mass	Mass	Atomic polarizability
N	14.01	0.53
H	1.008	0.161
CT	12.01	0.878
C	12.01	0.616
O	16	0.434
OH	16	0.465
HO	1.008	0.135
H1	1.008	0.135
HC	1.008	0.135

Bond	Harmonic force constant	Equilibrium bond length
H -N	403.2	1.013
CT-N	328.7	1.462
C -CT	313	1.524
CT-H1	330.6	1.097
CT-CT	300.9	1.538
C -O	637.7	1.218
C -OH	400.1	1.351
HO-OH	371.4	0.973
CT-HC	330.6	1.097

Angle	Harmonic force constant	Equilibrium bond angle
C -CT-N	67	109.06
H1-CT-N	49.84	108.88
CT-CT-N	65.91	111.61
CT-N -H	45.8	117.68
CT-C -O	67.4	123.2

CT-C -OH	68.4	112.73
CT-CT-HC	46.34	109.8
C -CT-CT	63.27	111.04
C -CT-H1	47.04	108.22
C -OH-HO	49.88	106.55
O -C -OH	75.92	122.1
CT-CT-H1	46.39	109.56
HC-CT-HC	39.4	107.58
C -CT-HC	46.93	108.77
N - C- N	70	120
C - N- C	50	121.9

Non-bonded	**Coefficient of 12th power term**	**Coefficient of 6th power term**
N	1.824	0.17
H	0.6	0.0157
CT	1.908	0.1094
C	1.908	0.086
O	1.6612	0.21
OH	1.721	0.2104
HO	0	0
H1	1.387	0.0157
HC	1.487	0.0157
	1.824	0.17

Improper		**Barrier height**	**Phase shift angle in torsional function**	**Periodicity of torsional barrier**
CT-O -C -OH		1.1	180	2

Dihedral	**Torsional barrier division factor**	**Barrier height, divided by 2**	**Phase shift angle in torsional function**	**Periodicity of torsional barrier**
O -C -CT-N	6	0	180	2
OH-C -CT-N	6	0	180	2
HC-CT-CT-N	9	1.4	0	3
C -CT-CT-N	9	1.4	0	3
C -CT-N -H	6	0	0	2

H1-CT-N -H	6	0	0	2
CT-CT-N -H	6	0	0	2
CT-C -OH-HO	2	4.6	180	2
O -C -CT-CT	6	0	180	2
C -CT-CT-HC	9	1.4	0	3
C -CT-CT-C	9	1.4	0	3
O -C -CT-H1	1	0.8	0	-1
O -C -CT-H1	1	0	0	-2
O -C -CT-H1	1	0.08	180	3
O -C -OH-HO	1	2.3	180	-2
O -C -OH-HO	1	1.9	0	1
OH-C -CT-H1	6	0	180	2
OH-C -CT-CT	6	0	180	2
H1-CT-CT-HC	9	1.4	0	3
C -CT-CT-H1	9	1.4	0	3
O -C -CT-HC	1	0.8	0	-1
O -C -CT-HC	1	0	0	-2
O -C -CT-HC	1	0.08	180	3

```
def topology_sort(self):
        '''
        sorts the sequence of the lasso peptide into sections
        representative of its topology
        '''
        self.upperLoop=[]
        self.tail=[]
        self.plugs=[]
        self.ring=[]
        self.upperPlug=[]
        self.lowerPlug=[]
        self.essentialRes=[] #this residue (ASX) connects the
ring and tail

        #these ranges can be changed to accomodate for
        #different lasso peptide structures.
        upperRange=range(6)
        plugRange=range(7,9)
        tailRange=range(10,12)
        ringRange=range(13,21) # 21 = 7mr, 22 = 8mr, 23 = 9mr

        #pluglocation
        uPlugRes = 7
        lPlugRes = 8

        #sort the PDB
        with open(self.path) as f:
            lines = [line.split() for line in f]
            i = 4 #by residue number
            for l in lines:
                if i < len(l): #avoid 'END'
                    if int(l[i]) in upperRange:
                        self.upperLoop.append(str(l[i]))
                    if int(l[i]) in plugRange:
                        self.plugs.append(str(l[i]))
                        if int(l[i]) == uPlugRes:
                            self.upperPlug.append(str(l[i]))
                        if int(l[i]) == lPlugRes:
                            self.lowerPlug.append(str(l[i]))
                    if int(l[i]) in tailRange:
                        self.tail.append(str(l[i]))
                    if int(l[i]) in ringRange:
                        self.ring.append(str(l[i]))
                    if (l[i-1]) == 'ASX':
                        self.essentialRes.append(str(l[i]))
                else:
                    None

        self.upperLoop = list(set(self.upperLoop))
        self.tail = list(set(self.tail))
        self.plugs = list(set(self.plugs))
        self.ring = list(set(self.ring))
        self.upperPlug = list(set(self.upperPlug))
        self.lowerPlug = list(set(self.lowerPlug))
        self.essentialRes = list(set(self.essentialRes))

        return self.upperLoop, self.tail, self.plugs, self.ring,
self.upperPlug, self.lowerPlug, self.essentialRes

def partition(self, sect=''):
        '''
        partitions the lasso structure into parts.
        '''

        a = list(self.topology_sort())
        upperLoop = a[0] #loop
        tail = a[1] #tail
        plugs = a[2] #plugs
        ring = a[3] #ring
        upperPlug = a[4] #upper plug
        lowerPlug = a[5] #lower plug
        essentialRes = a[6]

        sect = '' #blank string: mutate any topological section.

        self.get_stru()

        chain = choice(self.stru.chains)
        resi = choice(chain.residues)

        if sect == 'loop':
            if str(resi.id) not in upperLoop:
                while str(resi.id) not in upperLoop:
                    resi = choice(chain.residues)
        elif sect == 'tail':
            while str(resi.id) not in tail:
                resi = choice(chain.residues)
        elif sect == 'plugs':
            while str(resi.id) not in plugs:
                resi = choice(chain.residues)
        elif sect == 'ring':
            while str(resi.id) not in ring:
                resi = choice(chain.residues)
        elif sect == 'upper plug':
            while str(resi.id) not in upperPlug:
                resi = choice(chain.residues)
        elif sect == 'lower plug':
            while str(resi.id) not in lowerPlug:
                resi = choice(chain.residues)
        elif sect == '':
            resi = choice(chain.residues)
            if str(resi.id) in essentialRes:
                while str(resi.id) in essentialRes:
                    resi = choice(chain.residues)

        return chain, resi
```

Figure S3. Parsing submodule for random mutation.

Text S2. Steered molecular dynamics protocol

To construct each scaffold, we applied steered molecular dynamics (sMD) to a peptide ring-thread system. The ring-thread system's amino acid sequence featured almost exclusively alanine residues except for a glutamate or aspartate isopeptide linker. The center-of-mass (COM) of the N-methyl capped C-terminus of the thread was docked into the COM of the ring. By using AmberMD's tLEaP submodule, we parameterized the system with the ff14SB force field and solvated the system with a truncated octahedron TIP3P water box with a 40 Angstrom cutoff. We used the Antechamber submodule to assign charges and atom types to the aspartate or glutamate linker and used the prepgen program and tLEaP to parameterize the acidic linker. For sMD, we selected the N atom of the of the peptide thread's second alanine and the C-terminus carbon of the acidic linker. With a harmonic restraint of 2000kJ/mol, the 30 ps sMD simulation brought the selected atoms to a 1.5 Angstrom distance. We used tLEaP to bond the atoms, remove extraneous oxygen and hydrogen atoms, and create a PDB file of the partial lasso peptide structure

Table S2. Comparison between RESP charge model and AM1-BCC charge model. Average RMSD for RESP charge models were taken from a 10 ns production MD and compared to the first 10 ns of the original LHTP production MD, which used the AM1-BCC charge model.

Construct	**Upper Plug**	**Structure**	**Modality**	**Average RMSD RESP (Å)**	**Average RMSD AM1-BCC (Å)**
benenodin-1 state 1	E14	full	LHTP	3.17	2.95
benenodin-1 state 2	A16	full	LHTP	2.15	2.55
	Q15	full	LHTP	2.28	2.69
caulosegnin-II (A Proline)	H15	full	LHTP	1.10	1.03
caulosegnin-II (B Proline)	H15	full	LHTP	1.10	1.03
citrocin	R17	full	LHTP	2.34	2.24
microcin J25 (RGD Mutant)	F19	full	LHTP	1.74	1.64
streptomonomicin	A15	full	LHTP	3.83	3.51
	P14	full	LHTP	3.81	4.83
	Y13	full	LHTP	3.86	3.99
ubonodin	Y26	full	LHTP	2.93	3.07
xanthomonin-II	G10	full	LHTP	3.05	2.43
	G11	full	LHTP	3.25	3.20
	M9	full	LHTP	3.88	3.22

```
Minimize
  &cntrl
    imin  = 1,  ntx   = 1,  irest = 0,
    ntc   = 2,    ntf = 2,
    cut   = 10.0,
    maxcyc= 20000, ncyc  = {0.5maxcyc},
    ntpr  = {0.01maxcyc},   ntwx  = 0,
    ntr   = 1,  restraint_wt = 2.0, restraintmask =
'@C,CA,N',
    /

Heat
 &cntrl
  imin  = 0,  ntx = 1, irest = 0,
  ntc   = 2, ntf = 2,
  cut   = 10.0,
  nstlim= 20000, dt= 0.002,
  tempi = 0.0,  temp0=300.0,
  ntpr  = {0.01nstlim},  ntwx={nstlim},
  ntt   = 3, gamma_ln = 5.0,
  ntb   = 1,  ntp = 0,
  iwrap = 1,
  nmropt= 1,
  ig    = -1,
  ntr   = 1,    restraint_wt = 2.0, restraintmask =
'@C,CA,N',
 /
 &wt
  type  = 'TEMP0',
  istep1= 0, istep2={0.9nstlim},
  value1= 0.0, value2=300.0,
 /
 &wt
  type  = 'TEMP0',
  istep1= {A_istep2+1}, istep2={nstlim},
  value1= 300.0, value2=300.0,
 /
 &wt
  type  = 'END',
 /

Equilibration: constant pressure
 &cntrl
  imin  = 0,  ntx = 5,  irest = 1,
  ntf   = 2,  ntc = 2,
  nstlim= 500000, dt= 0.002,
  cut   = 10.0,
  temp0 = 300.0,
  ntpr  = {0.002nstlim}, ntwx = 5000,
  ntt   = 3, gamma_ln = 5.0,
  ntb   = 2,  ntp = 1,
  iwrap = 1,
  ig    = -1,
  ntr   = 1,    restraint_wt = 2.0,
  restraintmask = '@C,CA,N',
 /

Production: constant pressure
 &cntrl
  imin  = 0, ntx = 1, irest = 0,
  ntf   = 2,  ntc = 2,
  nstlim= 50000000, dt= 0.002,
  cut   = 10.0,
  temp0 = 300.0,
  ntpr  = 50000, ntwx = 5000,
  ntt   = 3, gamma_ln = 5.0,
  ntb   = 2,  ntp = 1,
  iwrap = 1,
  ig    = -1,
 /
```

Figure S4. Default input files for molecular dynamics simulation.

Text S3. Description of caulosegnin-II crystal structure.
The crystal structure of caulosegnin-II has two defined forms: A and B. These forms reference the stereochemical positions of P8 and P18 of caulosegnin. The A form features both prolines' gamma carbons oriented towards the plane of the peptide bond. In contrast, the B form features the same prolines' gamma carbons facing away from the peptide bond plane.

Table S3. RMSD values for full and ring, loop, and tail substructures of all eight benchmarked lasso peptides. RMSD values account for backbone heavy atoms.

Construct	Upper Plug	Structure	Modality	Average RMSD (Å)
benenodin-1 state 1	E14	full	LHTP	2.96
			NMR	3.04
		loop	LHTP	1.55
			NMR	1.04
		ring	LHTP	1.39
			NMR	1.09
		tail	LHTP	2.34
			NMR	2.34
benenodin-1 state 2	A16	full	LHTP	2.24
			NMR	1.21
		loop	LHTP	1.64
			NMR	0.94
		ring	LHTP	1.50
			NMR	0.58
		tail	LHTP	1.30
			NMR	0.74

	Q15	full	LHTP	2.66
			NMR	1.21
		loop	LHTP	1.28
			NMR	0.95
		ring	LHTP	1.38
			NMR	0.58
		tail	LHTP	0.85
			NMR	0.77
caulosegnin-II (A Proline)	H15	full	LHTP	1.48
			NMR	1.48
		loop	LHTP	0.70
			NMR	0.68
		ring	LHTP	0.70
			NMR	0.71
		tail	LHTP	0.84
			NMR	0.86
caulosegnin-II (B Proline)	H15	full	LHTP	1.48
			NMR	1.55
		loop	LHTP	0.70
			NMR	0.67
		ring	LHTP	0.70
			NMR	0.66
		tail	LHTP	0.84
			NMR	0.85
citrocin	R17	full	LHTP	2.24
			NMR	2.26
		loop	LHTP	1.95
			NMR	2.00
		ring	LHTP	0.87
			NMR	1.08
		tail	LHTP	1.22
			NMR	1.21
microcin J25 (RGD Mutant)	F19	full	LHTP	1.93
			NMR	1.65
		loop	LHTP	1.67
			NMR	1.15
		ring	LHTP	0.88
			NMR	0.85
		tail	LHTP	0.61
			NMR	0.81

streptomonomicin	A15	full	LHTP	3.38
			NMR	2.49
		loop	LHTP	1.38
			NMR	0.51
		ring	LHTP	1.85
			NMR	0.88
		tail	LHTP	2.96
			NMR	2.43
	P14	full	LHTP	3.77
			NMR	2.49
		loop	LHTP	0.73
			NMR	0.42
		ring	LHTP	1.64
			NMR	0.88
		tail	LHTP	2.75
			NMR	2.57
	Y13	full	LHTP	3.77
			NMR	2.49
		loop	LHTP	1.18
			NMR	0.36
		ring	LHTP	1.60
			NMR	0.88
		tail	LHTP	3.08
			NMR	2.67
ubonodin	Y26	full	LHTP	3.33
			NMR	3.04
		loop	LHTP	2.96
			NMR	2.63
		ring	LHTP	1.34
			NMR	0.76
		tail	LHTP	0.54
			NMR	0.87
xanthomonin-II	G10	full	LHTP	2.53
			NMR	2.46
		loop	LHTP	0.81
			NMR	0.78
		ring	LHTP	0.80
			NMR	0.79
		tail	LHTP	1.88
			NMR	1.77
	G11	full	LHTP	3.31
			NMR	2.46
		loop	LHTP	1.44

		NMR	1.04
	ring	LHTP	1.57
		NMR	0.79
	tail	LHTP	1.13
		NMR	1.01
M9	full	LHTP	2.75
		NMR	2.46
	loop	LHTP	0.40
		NMR	0.28
	ring	LHTP	1.08
		NMR	0.79
	tail	LHTP	2.00
		NMR	2.05

Table S4. PDB IDs for the seven lasso peptides used in the benchmark.

Lasso peptides	**PDB ID**
benenodin-1 conformer 1	5TJ1
benenodin-1 conformer 2	6B5W
citrocin	6MW6
RGD variant of microcin J25	2MMW
streptomonomicin	2MW3
ubonodin	6POR
xanthomonin-II	2MFV

Text S4. Building multiple constructs for a lasso peptide
Benenodin-1 conformer 2, streptomonomicin, and xanthomonin-II required multiple LassoHTP constructs for benchmarking studies. We used two separate constructs of loop size 7 and 8 for benenodin-1 conformer 2, three separate constructs of loop size 4, 5, and 6 for streptomonomicin, and three separate constructs of loop size 2, 3, and 4 for xanthomonin-II. For each construct, the tail extender submodule constructed the appropriate tail length in accordance with the difference in tail residues.

Table S5. NOE violation represented by the root mean square error (RMSE) of H-H distances. For each lasso peptide, the NOE violation is evaluated by the deviation of MD snapshots from the NMR restraints. From PDB files of NMR structures, we extracted the upper bounds of the H–H distance restraints. In each MD snapshot, an H–H pair was considered to have an NOE violation if their distance value is greater than the corresponding NMR restraint upper bound. To highlight the conformational fluctuation, the reference list only incorporates the H–H distances with an upper bound value greater than 5.0 Å. To assess the fluctuation of MD conformational snapshots from the NMR restraints, we computed the root mean square error (i.e., RMSE) for the excess distance values of the H–H pairs where NOE violation was observed. Caulosegnin-II is not included because the structure is derived from crystal structure, not the NMR.

Construct	Upper Plug	Structure	Modality	Root mean square error (RMSE) (Å)	Average RMSE for all sub-constructs (Å)
benenodin-1 state 1	E14	full	LHTP	3.66	3.66
			NMR	2.97	2.97
benenodin-1 state 2	A16	full	LHTP	1.72	1.58
			NMR	0.69	0.69
	Q15	full	LHTP	1.44	
			NMR	0.69	
citrocin	R17	full	LHTP	1.47	1.47
			NMR	1.39	1.39
microcin J25 (RGD Mutant)	F19	full	LHTP	3.26	3.26
			NMR	2.39	2.39
streptomonomicin	A15	full	LHTP	2.47	2.61
			NMR	1.69	1.69
	P14	full	LHTP	2.43	
			NMR	1.69	
	Y13	full	LHTP	2.92	
			NMR	1.69	
ubonodin	Y26	full	LHTP	1.55	1.55
			NMR	0.74	0.74
xanthomonin-II	G10	full	LHTP	0.33	0.56
			NMR	0.04	0.04
	G11	full	LHTP	0.85	
			NMR	0.04	
	M9	full	LHTP	0.51	
			NMR	0.04	

Table S6. Comparison of average RMSD for lasso peptide constructs before and after MD.

Construct	Upper Plug	Structure	Modality	Average RMSD before MD (Å)	Average RMSD after MD (Å)
benenodin-1 state 1	E14	full	LHTP	4.18	2.96
benenodin-1 state 2	A16	full	LHTP	2.72	2.24
	Q15	full	LHTP	3.95	2.66

caulosegnin-II (A Proline)	H15	full	LHTP	1.79	1.48
caulosegnin-II (B Proline)	H15	full	LHTP	1.79	1.48
citrocin	R17	full	LHTP	1.77	2.24
microcin J25 (RGD Mutant)	F19	full	LHTP	0.75	1.93
streptomonomicin	A15	full	LHTP	3.55	3.38
	P14	full	LHTP	5.05	3.77
	Y13	full	LHTP	4.92	3.77
ubonodin	Y26	full	LHTP	1.39	3.33
xanthomonin-II	G10	full	LHTP	3.87	2.53
	G11	full	LHTP	2.86	3.31
	M9	full	LHTP	4.60	2.75

Text S5. Procedure for dihedral PCA (dPCA) calculations. For each lasso peptide, dPCA was performed for both LassoHTP construct and NMR-initiated conformational ensembles. The backbone phi and psi dihedral angles were calculated for every residue for each snapshot in the conformational ensembles. Because of circular statistics involved with dihedral angles, we normalized each snapshot's dihedral angles. Specifically, we utilized min-max normalization. The two equations below show the process for rescaling data to fit a range of [0,1]. X_{scaled} is then utilized for PCA analysis.

$$x_{std} = (x - \min(x))/\ (\max(\mathrm{x}) - \min\ (\mathrm{x}))$$

$$x_{scaled} = x_{std} * (max - \min)\ +\ \min$$

For each lasso peptide, the principal components are calculated from the diagonalization of covariance matrix of the dihedral angle arrays that contain both LHTP and NMR-initiated conformational ensembles. PC1 and PC2, the top two principal components with greatest variance values, are used to project the dihedral angle array of each conformer on a two-dimensional plane.

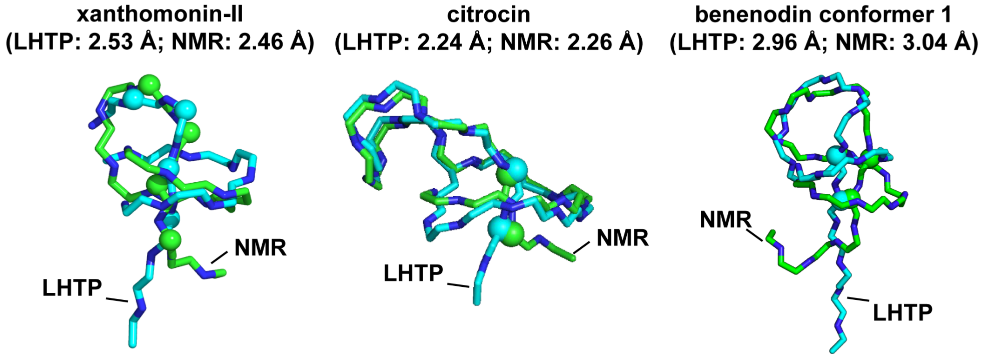

xanthomonin-II
(LHTP: 2.53 Å; NMR: 2.46 Å)
citrocin
(LHTP: 2.24 Å; NMR: 2.26 Å)
benenodin conformer 1
(LHTP: 2.96 Å; NMR: 3.04 Å)
NMR
LHTP
LHTP
NMR
NMR
LHTP

Figure S5. Structural superposition of initial NMR (green) and LHTP (cyan) scaffolds for citrocin and benenodin-1 conformer 1.