Supplemental Information

Computational modeling and prediction of deletion mutants

Hope Woods[1,2], Dominic L. Schiano[1,3], Jonathan I. Aguirre[3], Kaitlyn V. Ledwitch[1,3], Eli F. McDonald[1,3], Markus Voehler[1,3], Jens Meiler[1,3,4*,#] and Clara T. Schoeder[1,3,4*,#,†]

[1] Center of Structural Biology, Vanderbilt University, Nashville, Tennessee, TN 37235, United States
[2] Chemical and Physical Biology Program, Vanderbilt University, Nashville, Tennessee, TN 37235, United States
[3] Department of Chemistry, Vanderbilt University, Nashville, Tennessee, TN 37235, United States
[4] Institute for Drug Discovery, Leipzig University Medical School, Leipzig, 04103, Germany

* To whom correspondence should be addressed.

[#] these authors contributed equally

† Lead Contact

Corresponding authors:

Dr. Clara T. Schoeder
Institute for Drug Discovery, Leipzig University Medical Faculty, Liebigstr. 19, 04103 Leipzig, Germany Phone: +49 341 97-25756, Email: clara.schoeder@medizin.uni-leipzig.de

Jens Meiler, PhD
Department of Chemistry, Vanderbilt University, 21st Ave S, Nashville, TN 37235
and
Institute for Drug Discovery, Institute for Pharmacy, Leipzig University, Liebigstr. 19, 04103 Leipzig, Germany
Phone: +1 615 936 5662, Fax: +1 615 936 2211, Email: jens.meiler@vanderbilt.edu.

Content

Table S1: Pathogenic Deletion Mutations used for testing AlphaFold2-RosettaRelax protocol, related to Figure 6.

Protein	UniProt	PDB	Deletion	Rosetta Pose #
Phenylalanine-4-hydroxylase	P00439	1J8U	L194	L77
			L197	L80
			Y198	Y81
			L364	L247
Ornithine transcarbamylase	P00480	1EP9	L82	L48
			E309	E275
			E272	E238
Phosphoglycerate kinase 1	P00558	5NP8	K191	K187
Adenosine deaminase	P00813	3IAR	E337	E333
Thyroid hormone receptor beta	P10828	6KKB	T337	T120
Glycogen phosphorylase	P11217	1Z8D	F709	F693
Arylsulfatase A	P15289	1AUK	P137	P119
			F398	F380
Arylsulfatase B	P15848	1FSU	Y86	Y45
Neutrophil cytosol factor 2	P19878	1HH8	K58	K57
			E96	E95
Alanine--glyoxylate aminotransferase	P21549	6RV0	V139	V134
			A296	A291
Iduronate 2-sulfatase	P22304	5FQL	R95	R62
			S117	S84
			T118	T85
Ferrochelatase	P22830	2QD4	F417	F353
CD40 ligand	P29965	1ALY	G227	G112
acyl-CoA dehydrogenase	P49748	2UXW	E130	E62
			K278	K210
			E381	E313
Alpha-N-acetylglucosaminidase	P54802	4XWH	F142	F119
60S ribosomal protein L11	P62913	4XXB	E161	E133
Inositol polyphosphate 5-phosphatase OCRL	Q01968	3QIS	V742	V147
Unconventional myosin-VIIa	Q13402	5MV9	F1962	F247
Ribonucleoside-diphosphate reductase subunit M2 B	Q7LG56	4DJN	E85	E58
4-hydroxy-2-oxoglutarate aldolase	Q86XE5	3S5O	E315	E283
Atypical kinase COQ8A	Q8NI60	4PED	T584	T327
Glutathione S-transferase omega-1	P78417	6PNM	E155	E153

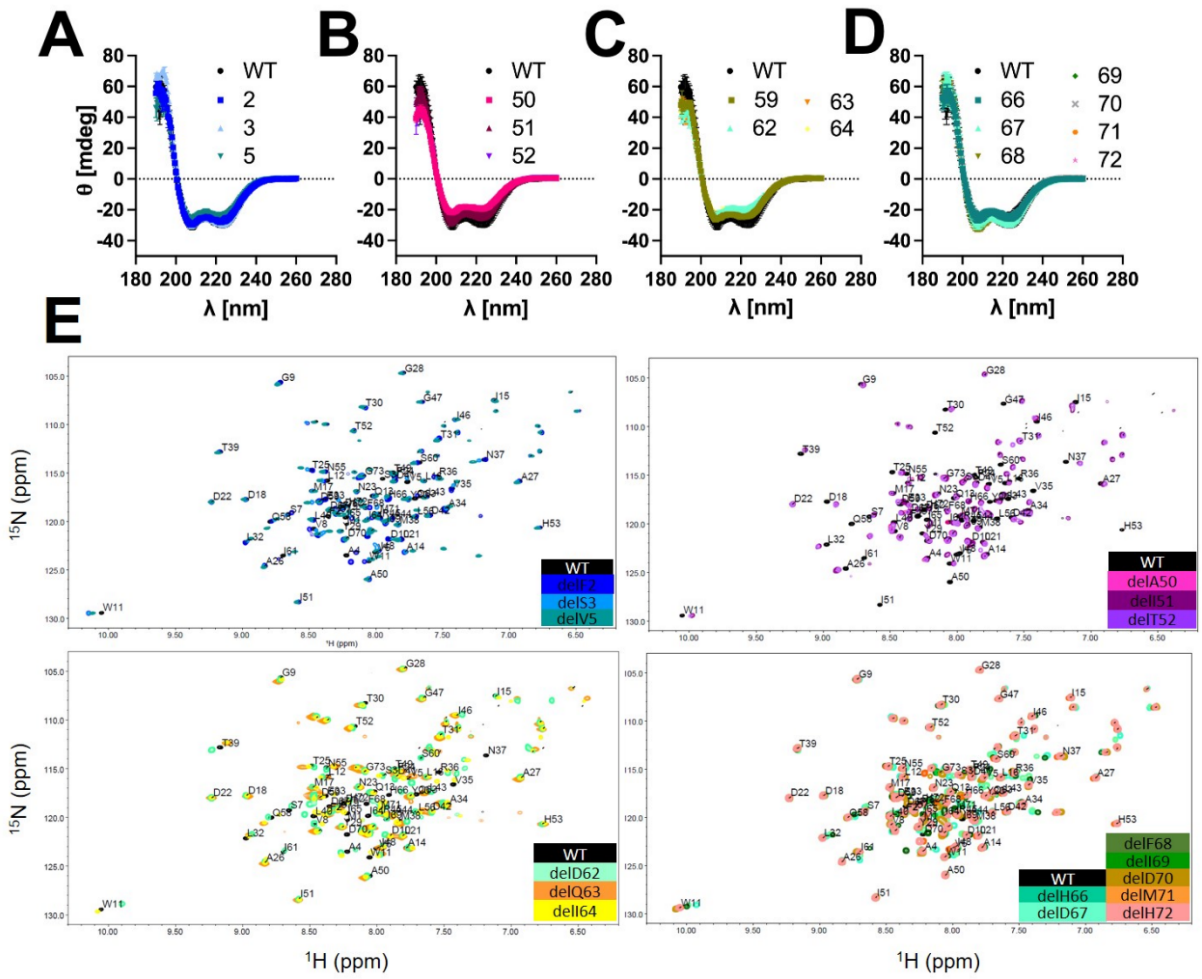

Figure S1. Sam domain deletion mutant structures, related to Figure 1 and 2. A-D Circular dichroism wavelength scans showing secondary structure of Sam domain mutants compared to WT shown in black. Scans for each mutant were averaged with standard deviation shown as error bars. **A.** N-terminal deletion mutants del 2, 3, 5, **B.** Loop IV deletion mutants del50-52, **C.** Helix V deletion mutants del59, 62-64 **D.** C-terminal deletion mutants del66-72 **E.** Overlay of the wildtype SAM domain and deletion mutants from 2D NMR spectroscopy. Amino acid are labeled for wildtype SAM domain peaks shown in black.

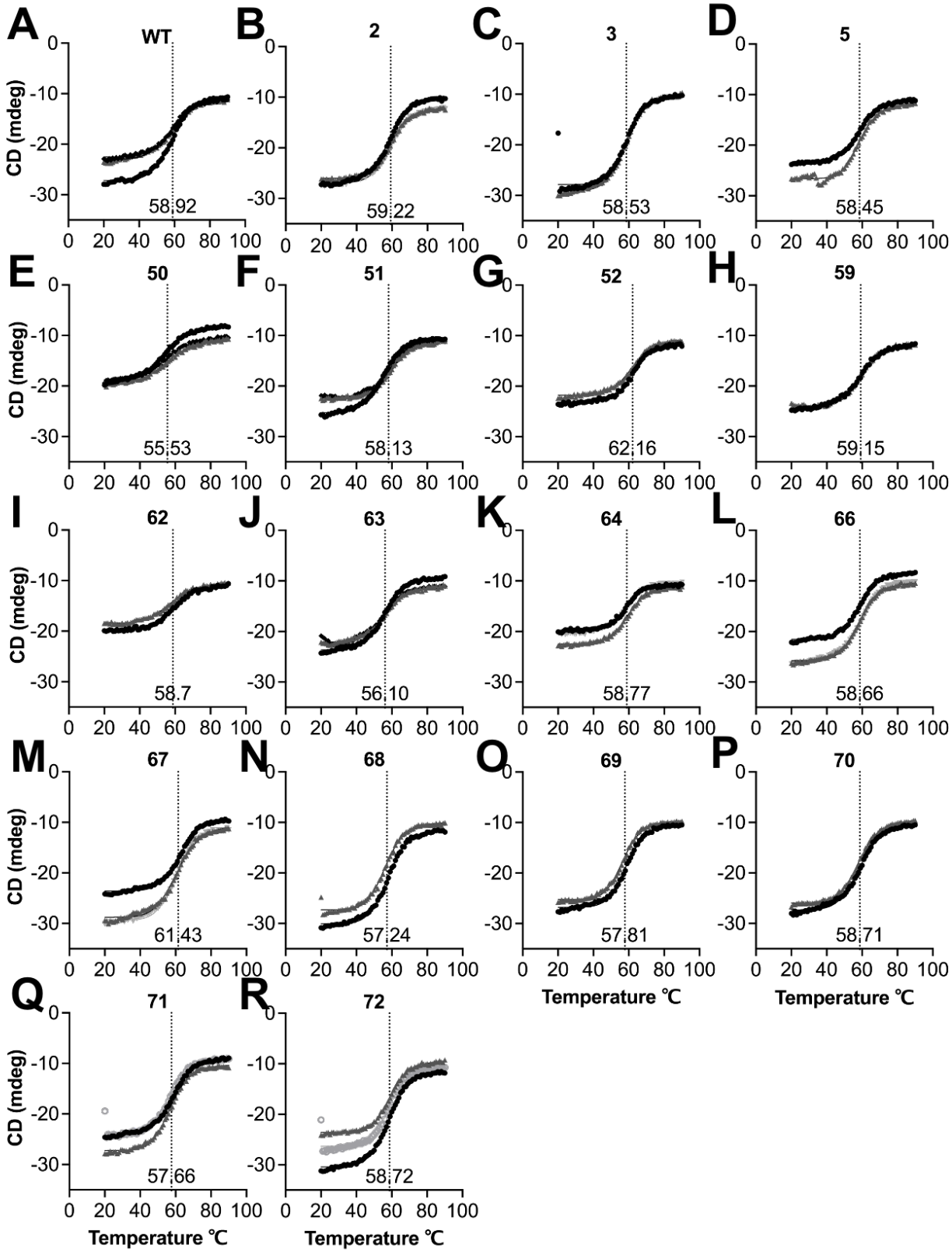
A
WT
58.92
B
2
59.22
C
3
58.53
D
5
58.45
E
50
55.53
F
51
58.13
G
52
62.16
H
59
59.15
I
62
58.7
J
63
56.10
K
64
58.77
L
66
58.66
M
67
61.43
N
68
57.24
O
69
57.81
P
70
58.71
Q
71
57.66
R
72
58.72
CD (mdeg)
Temperature ℃

Figure S2. Circular dichroism melting curves, related to Figure 1. Melting curves for each SAM domain mutant measured at 222 nm wavelength which changes as α-helical secondary structure is lost. The average melting temperature for each mutant is plotted with a dotted line. **A.** WT **B.** del2 **C.** del3 **D .** del5 **E.** del50 **F.** del51 **G.** del52 **H.** del59 **I.** del62 **J.** del63 **K.** del64 **L.** del66 **M.** del67 **N.** del68 **O.** del69 **P.** del70 **Q.** del71 **R.** del72

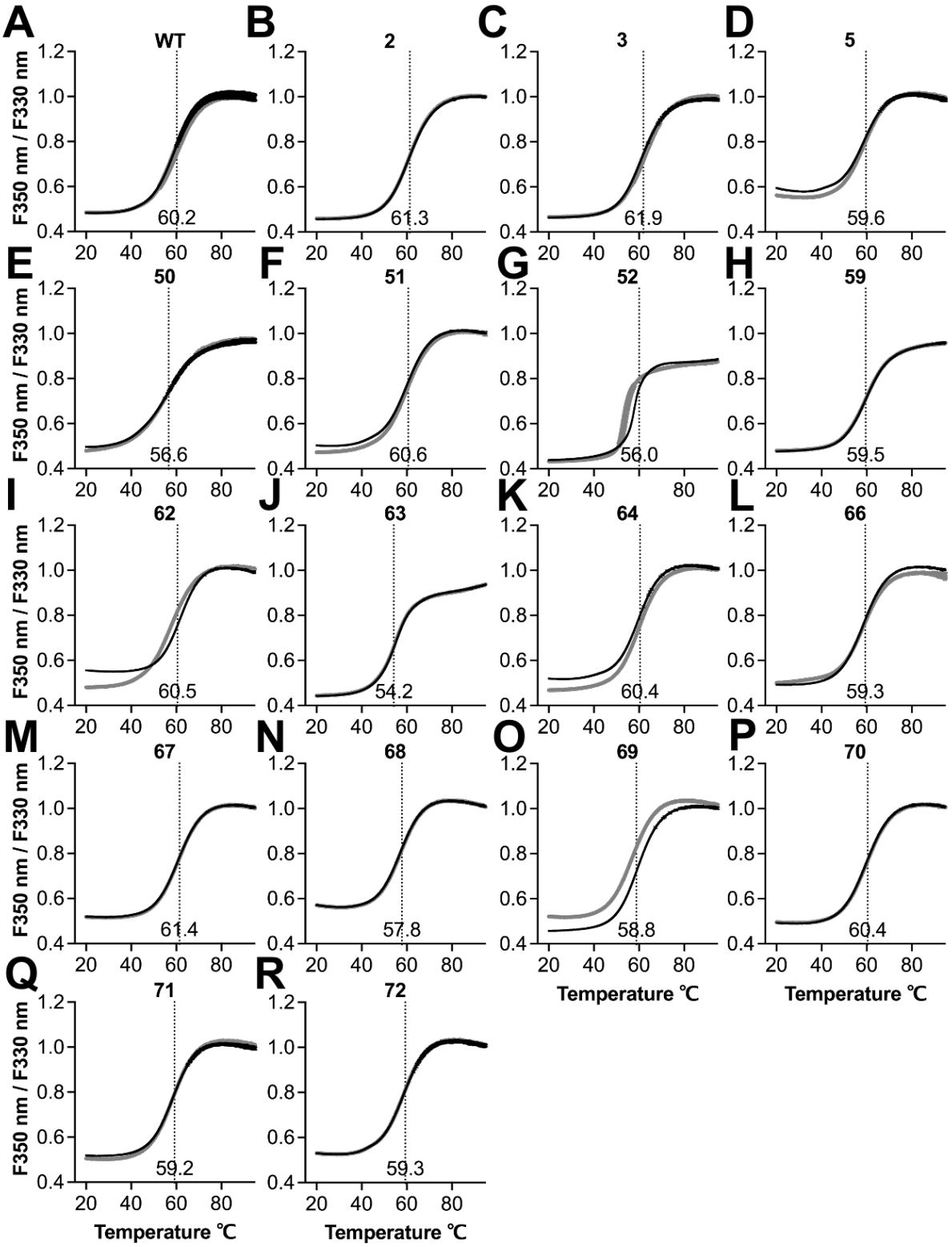

Figure S3. Nanoscale differential scanning fluorimetry, related to Figure 1 and 3. Melting curves for each SAM domain mutant done by measuring changes in the protein's intrinsic fluorescence. The average melting temperature for each mutant is plotted with a dotted line. **A.**

WT **B.** del2 **C.** del3 **D .** del5 **E.** del50 **F.** del51 **G.** del52 **H.** del59 **I.** del62 **J.** del63 **K.** del64 **L.** del66 **M.** del67 **N.** del68 **O.** del69 **P.** del70 **Q.** del71 **R.** del72

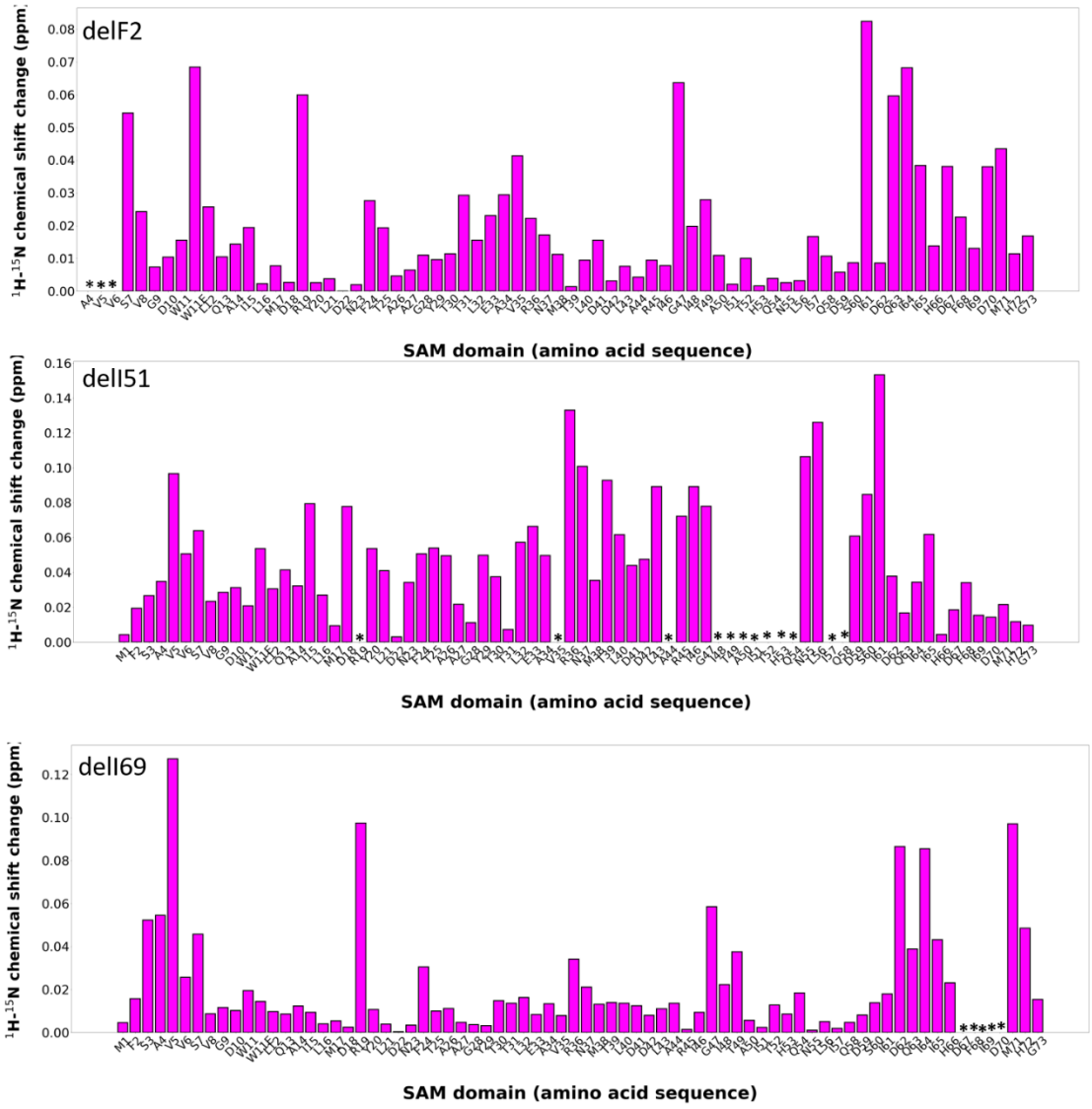

Figure S4. Exemplary CSP plots for del2,51,69, related to Figure 1 * indicates a residue that could not reliably mapped. All spectra were aligned to D22 in NMRViewJ.

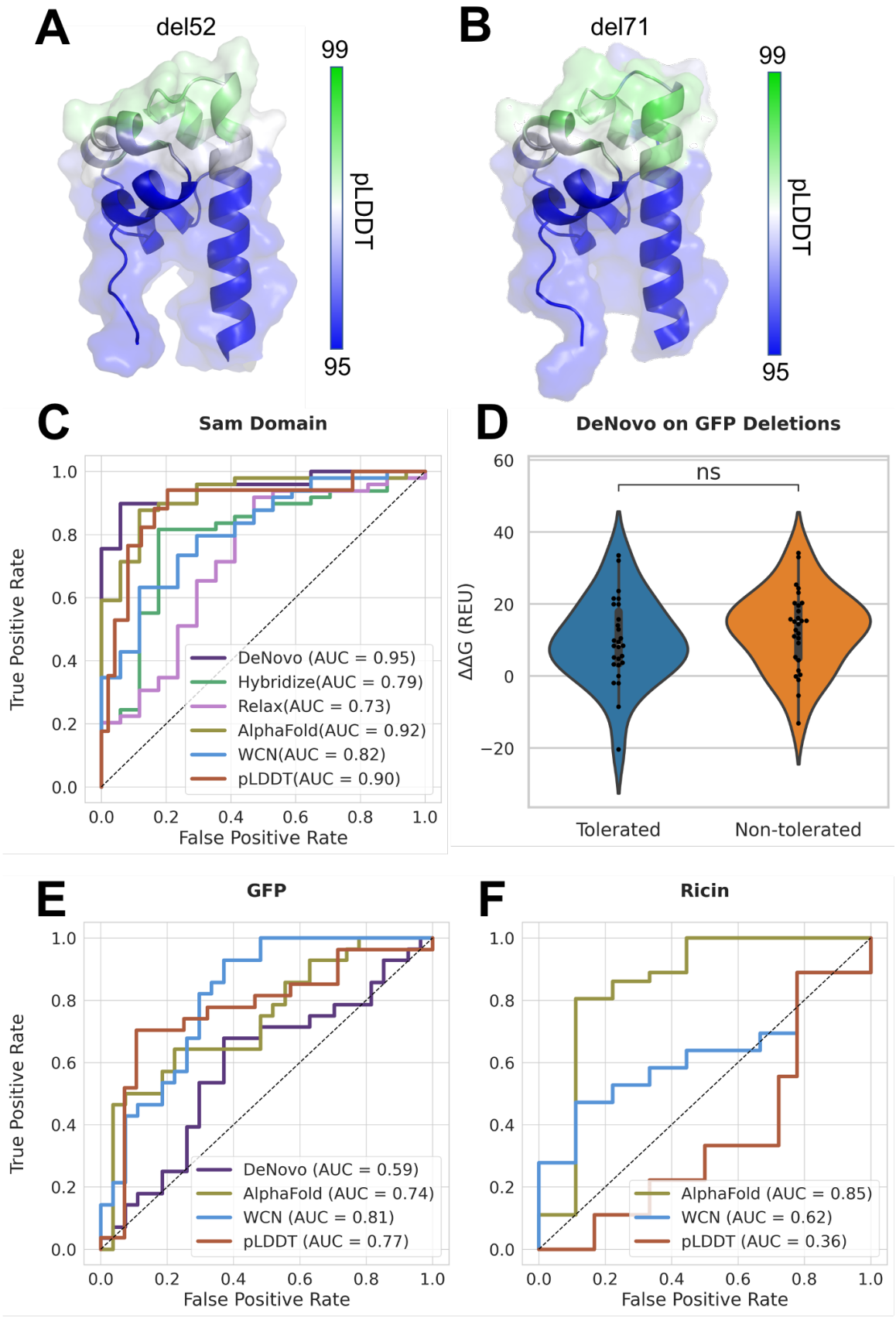
A
del52
99
pLDDT
95
B
del71
99
pLDDT
95
C
Sam Domain
True Positive Rate
False Positive Rate
DeNovo (AUC = 0.95)
Hybridize(AUC = 0.79)
Relax(AUC = 0.73)
AlphaFold(AUC = 0.92)
WCN(AUC = 0.82)
pLDDT(AUC = 0.90)
D
DeNovo on GFP Deletions
ΔΔG (REU)
ns
Tolerated
Non-tolerated
E
GFP
True Positive Rate
False Positive Rate
DeNovo (AUC = 0.59)
AlphaFold (AUC = 0.74)
WCN (AUC = 0.81)
pLDDT (AUC = 0.77)
F
Ricin
False Positive Rate
AlphaFold (AUC = 0.85)
WCN (AUC = 0.62)
pLDDT (AUC = 0.36)

Figure S5. Predictions for Sam Domain, GFP and Ricin deletion mutants, related to Figure 3 and 5. A. Per-residue pLDDT from AlphaFold for deletion 52 mapped onto the lowest scoring structures from the AlphaFold protocol. **B.** for deletion 71. **C.** Receiver operating characteristic (ROC) curves for classifying SAM domain mutants as soluble or insoluble using ΔΔG values from computational protocols DeNovo, Hybridize, Relax, and AlphaFold, predicted LDDT (pLDDT) values from AlphaFold output, and weighted contact number (WCN). Area under the curve (AUC) for each is shown in the legend. An AUC of 0.5 would be expected if classifications were done randomly. **D.** Distributions of scores from *de novo* protocol on GFP deletion tolerated and non-tolerated mutations were not found to be different based on Mann-Whitney test, p-value 0.24. **E.** ROC curves for classifying GFP mutants using ΔΔG values from computational protocols DeNovo and AlphaFold, pLDDT from AlphaFold, and WCN. **F.** ROC curves for classifying Ricin mutants using ΔΔG values from computational protocols DeNovo and AlphaFold, pLDDT from AlphaFold, and WCN.

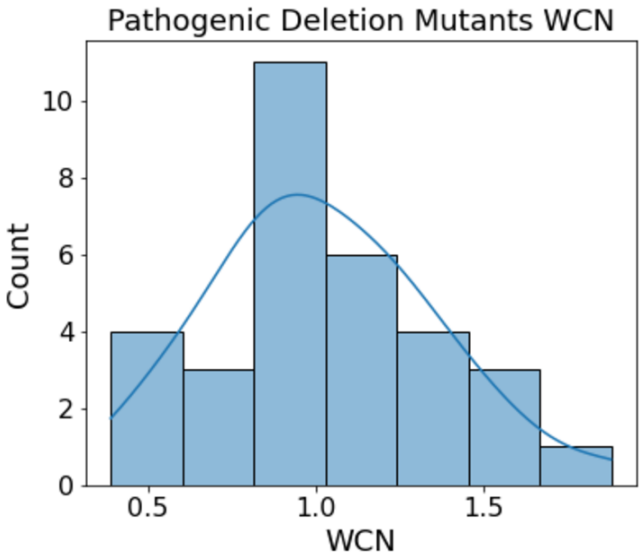

Figure S6. Pathogenic deletion side-chain weighted contact number, related to Figure 6. Distribution of side-chain weighted contact number of residues that result in pathogenic phenotypes when deleted from protein sequence.

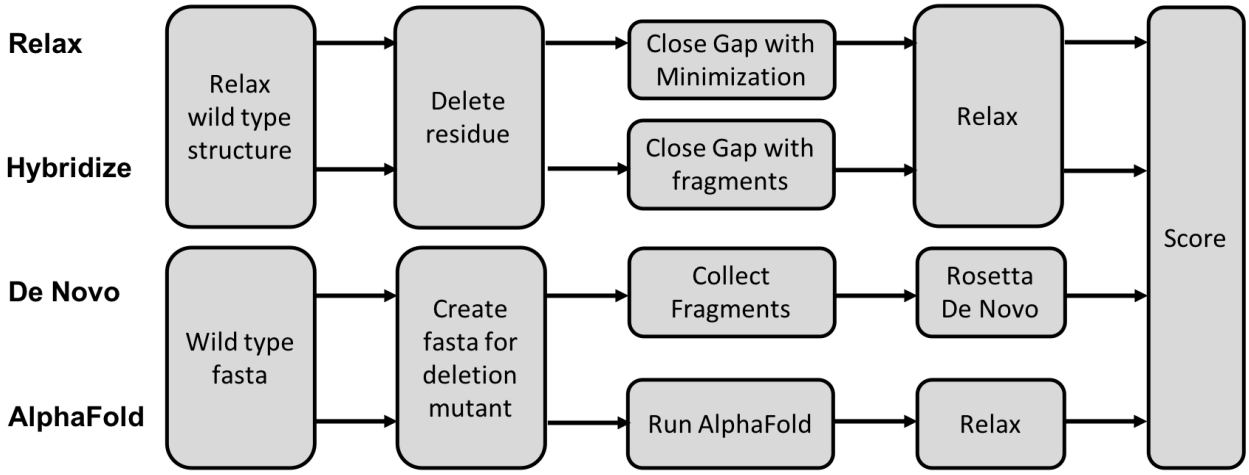

Figure S7. Computational Protocols, related to STAR Methods: Steps for each computational protocol tested on SAM domain deletion mutants.

Methods S1. Computational Protocol captures, related to STAR Methods

The following steps were followed for each protocol listed in figure S2. Example command lines and scripts are shown for deletion 52 in SAM domain. Both the deletion and wildtype structure were modeled for each protocol to produce scores that are easily comparable. Rosetta release version 3.13 was used for all Rosetta based modeling and scoring. All AlphaFold models were created with version 2.2.0 [S1]. If command lines for steps were shown previously, the step is listed without a command line.

Modeling of deletion mutants using AlphaFold

Wildtype:
Obtain fasta from PDB file with clean_pdb

```
/Path/To/Rosetta/main/tools/protein_tools/scripts/clean_pdb.py
1b0x2.pdb C
```

This will produce two files, a fasta file and a pdb file with all HETATM lines removed. For this example: 1b0x2_C.fasta and 1b0x2_C.pdb.
Run AlphaFold

```
python /sb/apps/alphafold220/alphafold/run_alphafold.py \
--fasta_paths=1b0x2_C.fasta \
--data_dir=$AF2_DATADIR \
--output_dir=$CALCDIR \
--max_template_date=9999-12-31 \
--use_gpu_relax \
--uniref90_database_path=$AF2_DATADIR/uniref90/uniref90.fasta \
--mgnify_database_path=$AF2_DATADIR/mgnify/mgy_clusters.fa \
--
uniclust30_database_path=$AF2_DATADIR/uniclust30/uniclust30_2018
_08/uniclust30_2018_08 \
--
bfd_database_path=$AF2_DATADIR/bfd/bfd_metaclust_clu_complete_id
30_c90_final_seq.sorted_opt \
--pdb70_database_path=$AF2_DATADIR/pdb70/pdb70 \
--template_mmcif_dir=$AF2_DATADIR/pdb_mmcif/mmcif_files \
--obsolete_pdbs_path=$AF2_DATADIR/pdb_mmcif/obsolete.dat \
--model_preset=monomer
```

Run Rosetta Relax

```
ls ranked_?.pdb > wt_models.list

/Path/To/Rosetta/main/source/bin/relax.default.linuxgccrelease -
in:file:l wt_models.list -relax:constrain_relax_to_start_coords
1 -nstruct 10 -out:file:silent af_wt.out -out:file:scorefile
af_wt.sc > af_wt.log
```

Mutant:
Create fasta file from PDB

```
cat 1b0x2_C.fasta | sed -n '2 p' | sed "s/.//52" >>
1b0x_del52.fasta
```

Run AlphaFold

```
python /sb/apps/alphafold220/alphafold/run_alphafold.py \
--fasta_paths=1b0x_del52.fasta \
--data_dir=$AF2_DATADIR \
--output_dir=YOUR/WORKING/DIRECTORY \
--max_template_date=9999-12-31 \
--use_gpu_relax \
--uniref90_database_path=$AF2_DATADIR/uniref90/uniref90.fasta \
--mgnify_database_path=$AF2_DATADIR/mgnify/mgy_clusters.fa \
--
uniclust30_database_path=$AF2_DATADIR/uniclust30/uniclust30_2018
_08/uniclust30_2018_08 \
--
bfd_database_path=$AF2_DATADIR/bfd/bfd_metaclust_clu_complete_id
30_c90_final_seq.sorted_opt \
--pdb70_database_path=$AF2_DATADIR/pdb70/pdb70 \
--template_mmcif_dir=$AF2_DATADIR/pdb_mmcif/mmcif_files \
--obsolete_pdbs_path=$AF2_DATADIR/pdb_mmcif/obsolete.dat \
--model_preset=monomer
```

Run Rosetta Relax

```
ls ranked_?.pdb > del52_models.list

/Path/To/Rosetta/main/source/bin/relax.default.linuxgccrelease -
in:file:l del52_models.list -
relax:constrain_relax_to_start_coords 1 -nstruct 10 -
out:file:silent af_del52.out -out:file:scorefile af_del52.sc >
af_del52.log
```

pLDDT values were taken from ranking_debug.json. The pLDDT values reported here is the pLDDT averaged across all residues. Scores from the lowest three scoring models produced from the Rosetta Relax were averaged together to obtain a final score for the model.

Modeling of deletion mutants using *de novo* folding

Wildtype:

Obtain fasta file

Create sequence profile and secondary structure prediction files for picking fragments. Details can be found at https://www.rosettacommons.org/docs/latest/application_documentation/utilities/app-fragment-picker or /Path/To/Rosetta/main/tools/fragment_tools/fragments.README.

```
fragmentpicker_runss 1b0x2_C.fasta
```

fragment_picker_quota.options

```
#input databases
-in::file::vall /dors/meilerlab/apps/rosetta/rosetta-
3.13/main/tools/fragment_tools/vall.jul19.2011.gz #constant

#query-related input files
-in::file::checkpoint 1b0x2_C.checkpoint #need to update
-in::file::fasta 1b0x2_C.fasta #need to update
-frags::ss_pred 1b0x2_C.psipred_ss2 psipred 1b0x2_C.jufo9d_ss
jufo #need to update

#the name root for the output fragment files
-out::file::frag_prefix wt-frags #need to update

#show score components for each selected fragment
-frags::describe_fragments wt-frags.fsc #need to update

#weights
-frags::scoring::config ../fragment_picker_quota.wghts #constant

#we need 9-mers and 3-mers
-frags::frag_sizes 9 3 #constant

#select 200 fragments from 1000 candidates. we need more
candidates than fragments to fill quota pools
-frags::n_candidates 1000 #constant
-frags::n_frags 200 #constant

#quota.def file defines the shares between different quota
pools. the total should be 1.0
-frags::picking::quota_config_file ../fragment_picker_quota.cfg
#constant
```

fragment_picker_quota.weights

```
#score  name     priority       weight  min_allowed     extras
SecondarySimilarity     350     0.5     -       psipred
SecondarySimilarity     250     0.5     -       jufo
RamaScore               150     1.0     -       psipred
RamaScore               150     1.0     -       jufo
ProfileScoreL1          200     1.0     -
```

fragment_picker_quota.cfg

```
#pool_id        pool_namefraction
1       psipred 0.6
2       jufo    0.4
```

Pick fragments

```
/Path/To/Rosetta/main/source/bin/fragment_picker.linuxgccrelease
@ fragment_picker_quota.options
```

This will produce fragment files wt-frags.200.3mers, wt-frags.200.9mers, wt-frags.200.3mers, wt-frags.fsc.200.3mers, wt-frags.fsc.200.9mers

del_broker.options

```
# Make sure all variable names have been replaced with absolute
path and that no line begins with a $ or ~s
-in
        -file
                -fasta 1b0x2_C.fasta    # protein sequence in
fasta format
                -frag3 wt-frags.200.3mers
                -frag9 wt-frags.200.9mers
-abinitio
        -increase_cycles 10     # Increase the number of cycles
at each stage in AbinitioRelax by this factor
        -rg_reweight 0.5        # Reweight contribution of
radius of gyration to total score by this scale factor
        -rsd_wt_helix 0.5       # Reweight env, pair, and cb
scores for helix residues by this factor
        -rsd_wt_loop 0.5        # Reweight env, pair, and cb
scores for loop residues by this factor
        -relax
-relax
        -fast   # At the end of the de novo protein_folding, do
a relax step of type "FastRelax".  This has been shown to be the
best deal for speed and robustness.
-broker
        -setup del_topology_broker.tpb
-run
        -protocol broker
        -reinitialize_mover_for_each_job # jd generate fresh
copy of its mover before each apply (once per job)
-score
        -find_neighbors_3dgrid  # Use a 3D lookup table for
doing neighbor calculations. For spherical, well-distributed
conformations
-out
        -nstruct 1000      # how many structures do you want to
generate?  Usually want to fold at least 1,000.
        -file
                -silent 1b0x2_C_broker.out      # full path to
silent file output
                -silent_struct_type binary      # we want binary
silent files
```

```
                -scorefile 1b0x2_C_broker.sc
-overwrite      # overwrite any existing output with the same
name you may have generated
```

Run Rosetta DeNovo

```
/Path/To/Rosetta/main/source/bin/AbinitioRelax.default.linuxgccr
elease @ del_broker.options -out:file:silent
1b0x2_C_wt_broker.out -out:file:scorefile 1b0x2_C_wt_broker.sc >
1b0x2_C_wt.log
```

Mutant:

Obtain fasta file

Setup for fragment picker

```
/dors/meilerlab/apps/scripts/rosetta_tools/fragmentpicker_runss
1b0x2_C_del52.fasta
```

fragment_picker_quota.options

```
#input databases
-in::file::vall /dors/meilerlab/apps/rosetta/rosetta-
3.13/main/tools/fragment_tools/vall.jul19.2011.gz #constant

#query-related input files
-in::file::checkpoint 1b0x2_C_del52.checkpoint #need to update
-in::file::fasta 1b0x2_C_del52.fasta #need to update
-frags::ss_pred 1b0x2_C_del52.psipred_ss2 psipred
1b0x2_C_del52.jufo9d_ss jufo #need to update

#the name root for the output fragment files
-out::file::frag_prefix del52-frags #need to update

#show score components for each selected fragment
-frags::describe_fragments del52-frags.fsc #need to update

#weights
-frags::scoring::config ../fragment_picker_quota.wghts #constant

#we need 9-mers and 3-mers
-frags::frag_sizes 9 3 #constant

#select 200 fragments from 1000 candidates. we need more
candidates than fragments to fill quota pools
-frags::n_candidates 1000 #constant
-frags::n_frags 200 #constant

#quota.def file defines the shares between different quota
pools. the total should be 1.0
```

```
-frags::picking::quota_config_file ../fragment_picker_quota.cfg
#constant
```

fragment_picker_quota.weights

```
#score  name     priority       weight  min_allowed     extras
SecondarySimilarity     350     0.5     -       psipred
SecondarySimilarity     250     0.5     -       jufo
RamaScore               150     1.0     -       psipred
RamaScore               150     1.0     -       jufo
ProfileScoreL1          200     1.0     -
```

fragment_picker_quota.cfg

```
#pool_id        pool_namefraction
1       psipred 0.6
2       jufo    0.4
```

Pick fragments

```
/Path/To/Rosetta/main/source/bin/fragment_picker.linuxgccrelease
@ fragment_picker_quota.options
```

This will produce fragment files del52-frags.200.3mers, del52-frags.200.9mers, del52-frags.200.3mers, del52-frags.fsc.200.3mers, del52-frags.fsc.200.9mers

del_broker.options

```
# Make sure all variable names have been replaced with absolute
path and that no line begins with a $ or ~s
-in
        -file
                -fasta 1b0x2_C_del52.fasta    # protein sequence
in fasta format
                -frag3 del52-frags.200.3mers
                -frag9 del52-frags.200.9mers
-abinitio
        -increase_cycles 10     # Increase the number of cycles
at each stage in AbinitioRelax by this factor
        -rg_reweight 0.5        # Reweight contribution of
radius of gyration to total score by this scale factor
        -rsd_wt_helix 0.5       # Reweight env, pair, and cb
scores for helix residues by this factor
        -rsd_wt_loop 0.5        # Reweight env, pair, and cb
scores for loop residues by this factor
        -relax
-relax
        -fast   # At the end of the de novo protein_folding, do
a relax step of type "FastRelax".  This has been shown to be the
best deal for speed and robustness.
```

```
-broker
        -setup del_topology_broker.tpb
-run
        -protocol broker
        -reinitialize_mover_for_each_job # jd generate fresh
copy of its mover before each apply (once per job)
-score
        -find_neighbors_3dgrid  # Use a 3D lookup table for
doing neighbor calculations. For spherical, well-distributed
conformations
-out
        -nstruct 1000      # how many structures do you want to
generate?  Usually want to fold at least 1,000.
        -file
                -silent 1b0x2_C_del52_broker.out      # full
path to silent file output
                -silent_struct_type binary      # we want binary
silent files
                -scorefile 1b0x2_C_del52_broker.sc
-overwrite      # overwrite any existing output with the same
name you may have generated
```

Run Rosetta DeNovo

```
/Path/To/Rosetta/main/source/bin/AbinitioRelax.default.linuxgccr
elease @ del_broker.options -out:file:silent
1b0x2_C_del52_broker.out -out:file:scorefile
1b0x2_C_del52_broker.sc > 1b0x2_C_del52.log
```

Scores from the lowest three scoring models were averaged to obtain the final score for each deletion and wildtype.

Modeling of deletion mutants using RosettaRelax

Clean PDB

```
/Path/To/Rosetta/main/tools/protein_tools/scripts/clean_pdb.py
1b0x2.pdb C
```

Run prep relax

```
/Path/To/Rosetta/main/source/bin/relax.default.linuxgccrelease -
in:file:s 1b0x2_C.pdb -nstruct 100 -relax:fast -
relax:constrain_relax_to_start_coords -out:file:scorefile
1b0x2_C.sc
```

Save lowest scoring model from prep relax to be used as starting structure for next step, in this case saved as 1b0x2_C_relaxed.

Wildtype:

delete_relax_wt.xml

```
<ROSETTASCRIPTS>
```

```
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <SCOREFXNS>
     <ScoreFunction name="ref2015_cart" weights="ref2015_cart"/>
     <ScoreFunction name="ref2015" weights="ref2015"/>
    </SCOREFXNS>
    <FILTERS>
    </FILTERS>
    <RESIDUE_SELECTORS>
     Index name="deletion" resnums="%%del_pos%%"/>
    </RESIDUE_SELECTORS>
    <TASKOPERATIONS>
     <InitializeFromCommandline name="commandline_init"/>
     <RestrictToRepacking name="restrict_to_repacking"/>
    </TASKOPERATIONS>
    <MOVERS>
        DeleteRegionMover name="delete"
residue_selector="deletion" rechain="false"/>
     <FastRelax name="relax" scorefxn="ref2015_cart" repeats="1"
dualspace="1" bondangle="1"/>
     <MinMover name="minimize" type="lbfgs_armijo_nonmonotone"
cartesian="1" bondlength="1" bondangle="1" chi="1" bb="1"
scorefxn="ref2015_cart"/>
    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>
    <PROTOCOLS>
        Add mover="delete"/>
     <Add mover="minimize"/>
     <Add mover="relax"/>
    </PROTOCOLS>
    <OUTPUT scorefxn="ref2015"/>
</ROSETTASCRIPTS>
```

Run delete_relax_wt rosetta script

```
/Path/To/Rosetta/main/source/bin/rosetta_scripts.default.linuxgc
crelease -parser:protocol ./delete_and_relax_wt.xml -in:file:s
1b0x2_C_relaxed.pdb -out:file:silent 1b0x2_relax.out -
out:file:scorefile 1b0x2_relax.sc -nstruct 100 > 1b0x2_relax.log
```

Mutant:

delete_relax.xml

```
<ROSETTASCRIPTS>
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <SCOREFXNS>
     <ScoreFunction name="ref2015_cart" weights="ref2015_cart"/>
     <ScoreFunction name="ref2015" weights="ref2015"/>
    </SCOREFXNS>
```

```
    <FILTERS>
    </FILTERS>
    <RESIDUE_SELECTORS>
     <Index name="deletion" resnums="%%del_pos%%"/>
    </RESIDUE_SELECTORS>
    <TASKOPERATIONS>
     <InitializeFromCommandline name="commandline_init"/>
     <RestrictToRepacking name="restrict_to_repacking"/>
    </TASKOPERATIONS>
    <MOVERS>
        <DeleteRegionMover name="delete"
residue_selector="deletion" rechain="false"/>
     <FastRelax name="relax" scorefxn="ref2015_cart" repeats="1"
dualspace="1" bondangle="1"/>
     <MinMover name="minimize" type="lbfgs_armijo_nonmonotone"
cartesian="1" bondlength="1" bondangle="1" chi="1" bb="1"
scorefxn="ref2015_cart"/>
    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>
    <PROTOCOLS>
        <Add mover="delete"/>
     <Add mover="minimize"/>
     <Add mover="relax"/>
    </PROTOCOLS>
    <OUTPUT scorefxn="ref2015"/>
</ROSETTASCRIPTS>
```

Run delete_relax rosetta script

```
/Path/To/Rosetta/main/source/bin/rosetta_scripts.default.linuxgc
crelease -parser:protocol -parser:protocol
./delete_and_relax.xml -parser:script_vars del_pos=52 -in:file:s
../../1b0x2_C_relaxed.pdb -out:file:silent 1b0x2_del52_relax.out
-out:file:scorefile 1b0x2_del52_relax.sc -nstruct 100
```

Scores from the lowest three scoring models were averaged to obtain the final score for each deletion and wildtype.

Modeling of deletion mutants using Rosetta Hybridize

Clean pdb
Run prep relax
Save lowest scoring model from prep relax to be used as starting structure for next step, in this case saved as 1b0x2_C_relaxed
Wildtype:
Note that the hybridize protocol does fragment insertion sampling based locally around where the deletion is, therefore, to have comparable scores each deletion has a corresponding wild type run where the sampling is done on the same set of residues for both mutant and wildtype. The below command lines will be specific for deletion 52.

delete_segment_hybridize_wt.xml

```
<ROSETTASCRIPTS>
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <SCOREFXNS>
     <ScoreFunction name="ref2015_cart" weights="ref2015_cart"/>
     <ScoreFunction name="ref2015" weights="ref2015"/>
    </SCOREFXNS>
    <FILTERS>
    </FILTERS>
    <RESIDUE_SELECTORS>
     Index name="deletion" resnums="%%del_pos%%"/>
    </RESIDUE_SELECTORS>
    <TASKOPERATIONS>
     <InitializeFromCommandline name="commandline_init"/>
     <RestrictToRepacking name="restrict_to_repacking"/>
    </TASKOPERATIONS>
    <MOVERS>
        DeleteRegionMover name="delete"
residue_selector="deletion" rechain="false"/>
     <SegmentHybridizer name="segment_hybridize" frag_big="5"
nfrags="200" rms_frags="0.5" use_frags="1" use_seq="1">
          <Span begin="%%start%%" end="%%stop%%"
extend_outside="3" extend_inside="1"/>
     </SegmentHybridizer>
     PrepareForFullatom name="fullatom"/>
     <SwitchResidueTypeSetMover name="fullatom"
set="fa_standard"/>
     <FastRelax name="relax" scorefxn="ref2015_cart" repeats="1"
dualspace="1" bondangle="1"/>
     <MinMover name="minimize" type="lbfgs_armijo_nonmonotone"
cartesian="1" bondlength="1" bondangle="1" chi="1" bb="1"
scorefxn="ref2015_cart"/>
     <ClearConstraintsMover name="clearconstraints"/>
    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>
    <PROTOCOLS>
        Add mover="delete"/>
     <Add mover="segment_hybridize"/>
     <Add mover="fullatom"/>
     Add mover="minimize"/>
     <Add mover="relax"/>
    </PROTOCOLS>
    <OUTPUT scorefxn="ref2015"/>
```

Run delete_segment_hybridize_wt rosetta script

```
/Path/To/Rosetta/main/source/bin/rosetta_scripts.default.linuxgc
crelease -parser:protocol ./delete_segment_hybridize_wt.xml -
parser:script_vars del_pos=52 start=42 stop=62 -in:file:s
1b0x2_C_relaxed.pdb -out:file:silent 1b0x2_wt52_seghyb.out -
out:file:scorefile 1b0x2_wt52_seghyb.sc -nstruct 100
```

Mutant:

delete_segment_hybridize.xml

```
<ROSETTASCRIPTS>
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <SCOREFXNS>
     <ScoreFunction name="ref2015_cart" weights="ref2015_cart"/>
     <ScoreFunction name="ref2015" weights="ref2015"/>
    </SCOREFXNS>
    <FILTERS>
    </FILTERS>
    <RESIDUE_SELECTORS>
     <Index name="deletion" resnums="%%del_pos%%"/>
    </RESIDUE_SELECTORS>
    <TASKOPERATIONS>
     <InitializeFromCommandline name="commandline_init"/>
     <RestrictToRepacking name="restrict_to_repacking"/>
    </TASKOPERATIONS>
    <MOVERS>
        <DeleteRegionMover name="delete"
residue_selector="deletion" rechain="false"/>
     <SegmentHybridizer name="segment_hybridize" frag_big="5"
nfrags="200" rms_frags="0.5" use_frags="1" use_seq="1">
          <Span begin="%%start%%" end="%%stop%%"
extend_outside="3" extend_inside="1"/>
     </SegmentHybridizer>
     PrepareForFullatom name="fullatom"/>
     <SwitchResidueTypeSetMover name="fullatom"
set="fa_standard"/>
     <FastRelax name="relax" scorefxn="ref2015_cart" repeats="1"
dualspace="1" bondangle="1"/>
     <MinMover name="minimize" type="lbfgs_armijo_nonmonotone"
cartesian="1" bondlength="1" bondangle="1" chi="1" bb="1"
scorefxn="ref2015_cart"/>
     <ClearConstraintsMover name="clearconstraints"/>
    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>
    <PROTOCOLS>
        <Add mover="delete"/>
     <Add mover="segment_hybridize"/>
     <Add mover="fullatom"/>
```

```
     Add mover="minimize"/>
     <Add mover="relax"/>
    </PROTOCOLS>
    <OUTPUT scorefxn="ref2015"/>
</ROSETTASCRIPTS>
```

Run delete_segment_hybridize rosetta script

```
/Path/To/Rosetta/main/source/bin/rosetta_scripts.default.linuxgc
crelease -parser:protocol ./delete_segment_hybridize.xml -
parser:script_vars del_pos=52 start=42 stop=62 -in:file:s
1b0x2_C_relaxed.pdb -out:file:silent 1b0x2_del52_seghyb.out -
out:file:scorefile 1b0x2_del52_seghyb.sc -nstruct 100
```

Scores from the lowest three scoring models were averaged to obtain the final score for each deletion and wildtype.

Weighted Contact Number:

The below command line and script were used for calculating side-chain weighted contact number (WCN), as described by Marcos and Echave [S2,S3]. The script requires the installation of PyRosetta; directions can be found here: https://www.pyrosetta.org/ [S4]. The calc_wcn.py script was ran with PyRosetta version: PyRosetta4.Release.python37.linux.

Command line:

```
python calc_wcn.py 1b0x2_C.pdb 1b0x_wcn.csv
```

calc_wcn.py

```
import os,sys
import numpy as np
from math import log,sqrt
from pyrosetta import *

init()

#script for calculating side-chain weighted contact number (WCN)

#first argument should be the pdb file
#second argument should be the name of the output file you want
to write the WCN in
#output file is a csv file, the first column being the residue
number and the second the WCN for that residue

pdb_file = sys.argv[1]
output_file = sys.argv[2]


pose = pose_from_pdb(pdb_file)

nres = pose.total_residue()
```

```
#create array of coordinates
sc_centers = []


#loop through residues
for resn in range(1,nres+1):
     residue = pose.residue(resn)
     center = []

     #if glycine, keep Calpha coordinate
     if ( residue.name() == "GLY" or residue.name() ==
"GLY:NtermProteinFull" or residue.name() ==
"GLY:CtermProteinFull"):
          center = np.array([residue.xyz(2).x, residue.xyz(2).y,
residue.xyz(2).z]) #Calpha
     else:
          n_heavy_atoms = residue.nheavyatoms()
          atom_coordinates = []
          #only loop through atoms starting at 5 because first 4
are backbone heavy atoms
          for atom in range(5,n_heavy_atoms+1):
               #collect heavy atom side chain coordinates
               atom_coordinates.append(residue.xyz(atom))

          #calculate geometric center
          center = np.sum(atom_coordinates,
axis=0)/(n_heavy_atoms - 4)

     sc_centers.append(center)

#initiate wcn array, length nres
wcn = []

with open( output_file, 'w') as f:

     f.write("position,wcn\n")

     #loop through array
     for i in range(0,nres):
          wcn_i = 0
          center_i = sc_centers[i]
          #loop through array again
          for j in range(0,nres):
               if( i != j ):
                    center_j = sc_centers[j]
```

```
                    #calculate distance between each residue sc
geometric center
                    sum_sq = np.sum(np.square(center_i -
center_j))
                    r = np.sqrt(sum_sq)
                    #sum r^(-2)
                    wcn_i += r**(-2)
          wcn.append(wcn_i)
          #with open( output_file, 'w') as f:
          f.write("%i,%f\n" %(i+1, wcn_i))
```

References

S1. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zidek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature *596*, 583-589. 10.1038/s41586-021-03819-2.

S2. Marcos, M.L., and Echave, J. (2015). Too packed to change: side-chain packing and site-specific substitution rates in protein evolution. PeerJ *3*, e911. 10.7717/peerj.911.

S3. Jackson, E.L., Spielman, S.J., and Wilke, C.O. (2017). Computational prediction of the tolerance to amino-acid deletion in green-fluorescent protein. Plos One *12*, e0164905. 10.1371/journal.pone.0164905.

S4. Chaudhury, S., Lyskov, S., and Gray, J.J. (2010). PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. Bioinformatics *26*, 689-691. 10.1093/bioinformatics/btq007.