Supplementary Information

Computationally Guided Redesign of a Heme-Free Cytochrome with Native-like Structure and Stability

Alexander M. Hoffnagle, Vanessa H. Eng, Ulrich Markel, and F.Akif Tezcan*

Department of Chemistry and Biochemistry, University of California, San Diego, La Jolla, California 92093, United States

*Corresponding Author: tezcan@ucsd.edu

Supplementary Information Contents:

Contents..S2

Supplementary Tables 1-6..S3-S8

Supplementary Figures 1-19..S9-S27

Supplementary Data...S28-S32

Supplementary References...S33

LOWEST Rosetta SCORES			Mutations				
Design	**Rosetta Score (REU)**	**RMSD (Helix 3)**	**3**	**7**	**98**	**101**	**102**
89	-364.724	0.56036	W	Y	V	I	W
33	-361.909	0.67509	W	F	F	I	A
118	-360.869	0.56484	W	M	L	I	W
34	-360.531	0.48516	F	F	H	I	W
142	-360.496	0.54472	L	L	L	L	Q
…							
ApoCyt (135)	-347.225	0.25520	W	M	V	H	W
…							
186	-320.032	0.77800	W	Q	H	R	H
Cyt cb_{562}	n.d.	n/a	L	M	C	C	H

Table S1. Rosetta energies of computationally screened variants. Listed in the table are scores in Rosetta energy units (REU), helix 3 RMSD, and mutations of the 5 highest scoring designs, ApoCyt, and the lowest scoring design. ApoCyt (Design 135) is highlighted with the red box.

LOWEST HELIX 3 RMSD			Mutations				
Design	**Rosetta Score (REU)**	**RMSD (Helix 3)**	**3**	**7**	**98**	**101**	**102**
ApoCyt (135)	-347.225	0.25520	W	M	V	H	W
148	-336.591	0.29536	W	M	H	I	W
92	-350.342	0.32102	L	Y	H	I	W
108	-352.178	0.33398	L	M	V	H	W
157	-344.025	0.35460	Y	L	H	R	W
…							
60	-345.108	1.04548	W	M	H	I	Y
Cyt cb_{562}	n.d.	n/a	L	M	C	C	H

Table S2. Rosetta energies of computationally screened variants. Listed in the table scores in Rosetta energy units (REU), helix 3 RMSD, and mutations of the 5 designs with the lowest helix 3 RMSD values, as well as the design with the largest RMSD value. ApoCyt (Design 135) is highlighted with the red box.

Cyt cb_{562}
*MLRTVIVAGALVLTASAVMA*AD**L**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKAT
PPKLEDKSPDSPEMWDFRHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**C**NA**CH**QKYR*

TriCyt3
*MLRTVIVAGALVLTASAVMA*AD**L**EDN**M**ETLNDNLKVIEKADNAAQVKDAL**K**KMR**K**AALDA**K**KA
TPPKLEDKSP**A**SPEMIDFR**V**GFD**E**L**AWE**I**H**DAA**H**LA**K**EGKVKEAQAAAEQLKTT**C**NA**CH**QKYR
*

Cyt-RIL
MAD**L**EDN**W**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMKDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**R**NA**YI**QKYL*

ApoCyt
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMWDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-A3
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMWDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-A7
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMWDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-A101
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMWDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-A102
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMWDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-A106
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMWDF
RHGFDILVGQIDDALKLANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-TriCyt3
MAAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDAL**K**KMR**K**AALDA**K**KATPPKLEDKSP**A**SPEMIDF
R**V**GFD**E**L**AWE**I**H**DAA**H**LA**K**EGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-TriCyt3 + 6xHis
MHHHHHHSAAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDAL**K**KMR**K**AALDA**K**KATPPKLEDKSP
ASPEMIDFR**V**GFD**E**L**AWE**I**H**DAA**H**LA**K**EGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

ApoCyt-RIDC1
MAD**W**EDN**M**ETLNDNLKVIEKADNAAQVKDALTKM**A**AAA**A**DA**WS**ATPPKLEDKSPDSPEMHDF
R**H**GF**W**IL**I**GQI**HD**AL**H**LANEGKVKEAQAAAEQLKTT**V**NA**HW**QKYR*

Table S3. Amino acid sequences of proteins used in this study. The cyt cb_{562} leader peptide is italicized, positions that were designed in the ApoCyt design protocol are in bold and underlined, mutations responsible for the assembly of oligomeric variants are in red, and residues mutated in the ApoCyt-Ala variants are highlighted in yellow. The cyt cb_{562} CXXCH motif (residues 98-102) has been replaced in ApoCyt by the C98V, C101H, and H102W mutations.

Protein	Beamline	[Protein]	Precipitant
ApoCyt	SSRL 9-2	4.62 mM	22% PEG 2000, 200 mM $MgCl_2$, 100 mM Tris (pH 8.5)
ApoCyt-TriCyt3	APEX II	2.0 mM	25% PEG 1500, 200 mM $(NH_4)_2SO_4$, 100 mM Bis-Tris (pH 5.5)

Table S4. Crystallization conditions for ApoCyt variants. [Protein] refers to the concentration of the monomeric protein. Diffraction data for ApoCyt-TriCyt3 was collected on a Bruker APEX II CCD detector (UCSD-Chemistry) using Cu Kα radiation.

	ApoCyt	**ApoCyt-TriCyt3**
PDB ID	8DEN	8DEL
Data Collection		
Space Group	P 1 21 1	C 1 2 1
Cell Dimensions (Å)	44.59 85.8799 47.57	43.778, 77.076, 92.653
Cell Angles (°)	90 92.49 90	90, 98.69, 90
Resolution (Å)	47.53 - 1.69	45.80 - 2.56
No. Unique Reflections	37769 (1569)	9904 (981)
R_{merge}	0.087 (0.627)	0.064 (0.533)
Multiplicity	6.2 (5.7)	8.9 (5.3)
CC 1/2	0.998 (0.731)	0.999 (0.926)
$< I / \sigma(I) >$	10.97 (2.3)	17.2 (2.2)
Completeness (%)	94.80 (76.7)	99.9 (99.4)
Refinement		
R_{work} / R_{free}	0.1843/0.2166	0.2025/0.2785
No. Atoms	3800	2636
Protein	3380	2528
Ligand/Ion	0	3
Solvent	420	105
R.m.s. Deviations		
Bond Lengths (Å)	0.007	0.009
Bond Angles (°)	0.87	1.104
Clashscore	5.65	7.71
Ramachandran Plot (%)		
Favored	99.52	99.04
Outliers	0.00	0.00
Rotamer Outliers (%)	2.28	3.50

Table S5. X-ray data collection and refinement statistics. Numbers in parentheses correspond to the highest resolution shell.

Variant	Alpha Helix	Beta Sheet	Random Coil
Cyt *cb*$_{562}$	1.00	0.00	0.00
ApoCyt	0.87	0.00	0.13
Cyt-RIL	0.60	0.07	0.33
ApoCyt-A3	0.26	0.43	0.31
ApoCyt-A7	0.26	0.43	0.31
ApoCyt-A101	0.26	0.43	0.32
ApoCyt-A102	0.24	0.38	0.38
ApoCyt-A106	0.26	0.43	0.31

Table S6. Secondary structure contents of cytochrome variants obtained by fitting CD spectra with the K2D algorithm[1] on the Dichroweb server.[2] Estimated secondary structure contents for all ApoCyt Ala variants are very similar, which may reflect the challenge of accurately calculating the secondary structure of proteins with complex structures and/or large regions of disorder.[3]

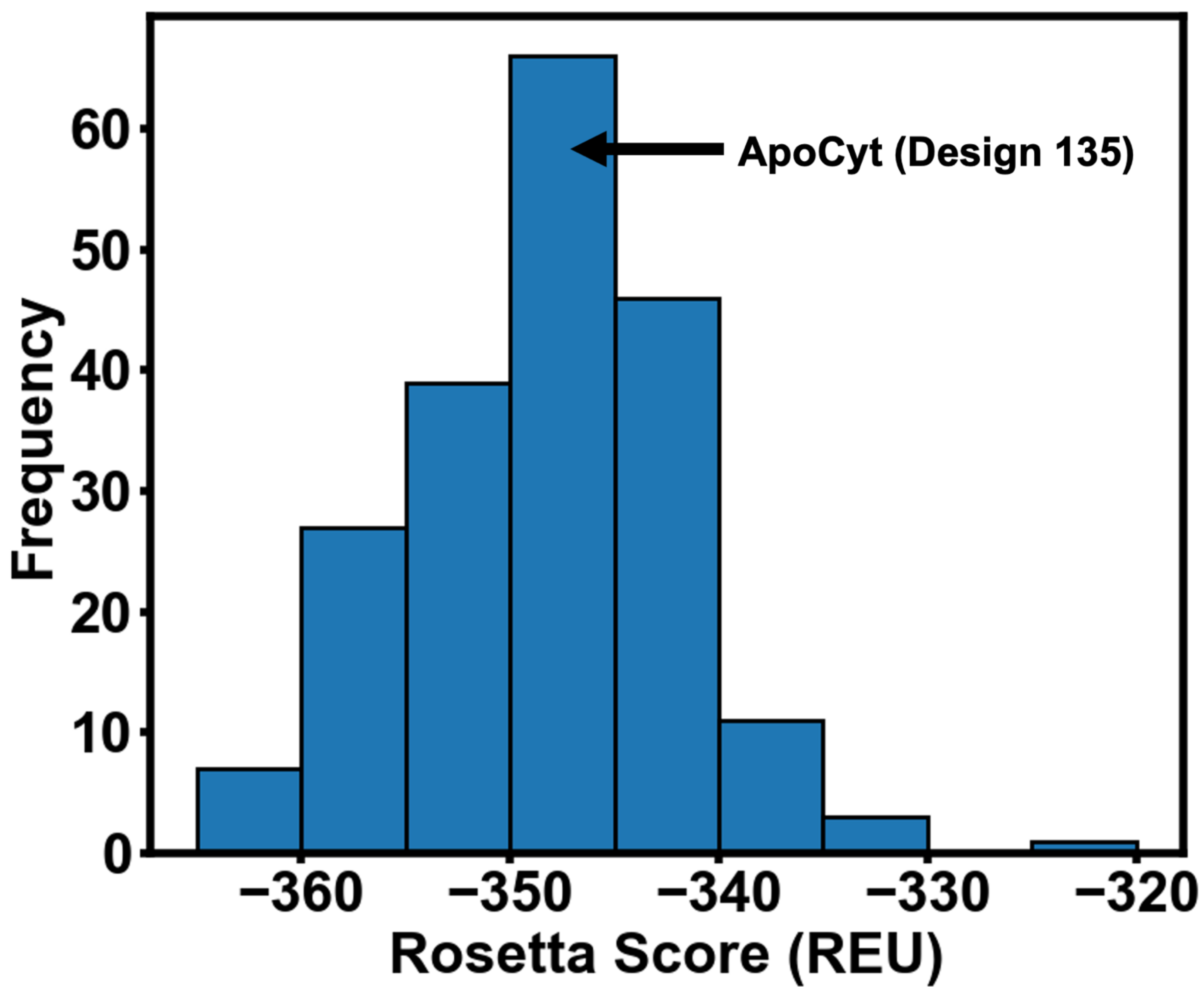

Figure S1. Histogram of the Rosetta Scores of the 200 computational designs that were obtained in the design process. The bin in which ApoCyt resides is indicated with an arrow.

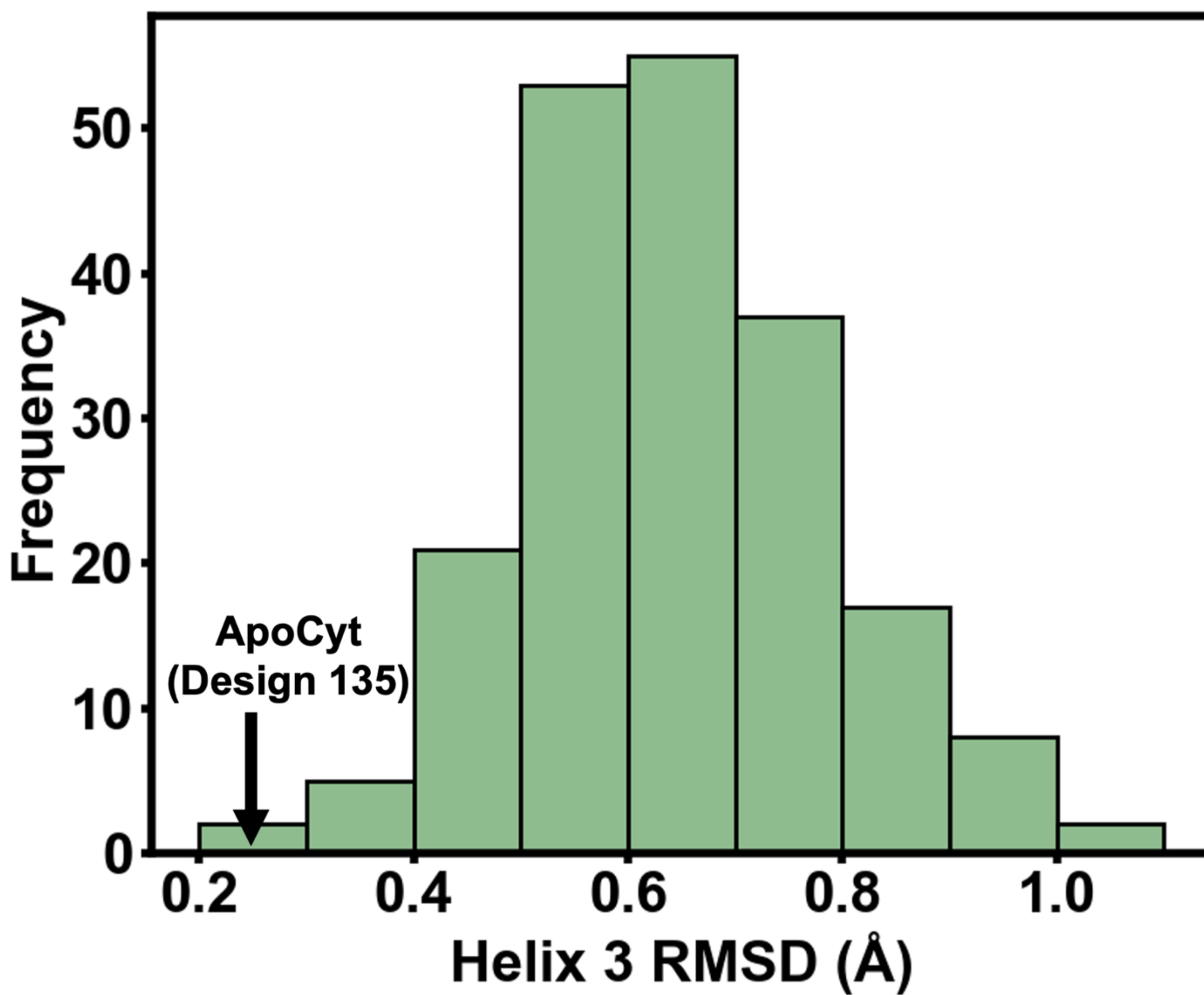

Figure S2. Histogram of the Helix 3 RMSD values of the 200 computational designs that were obtained in the design process. The bin in which ApoCyt resides is indicated with an arrow.

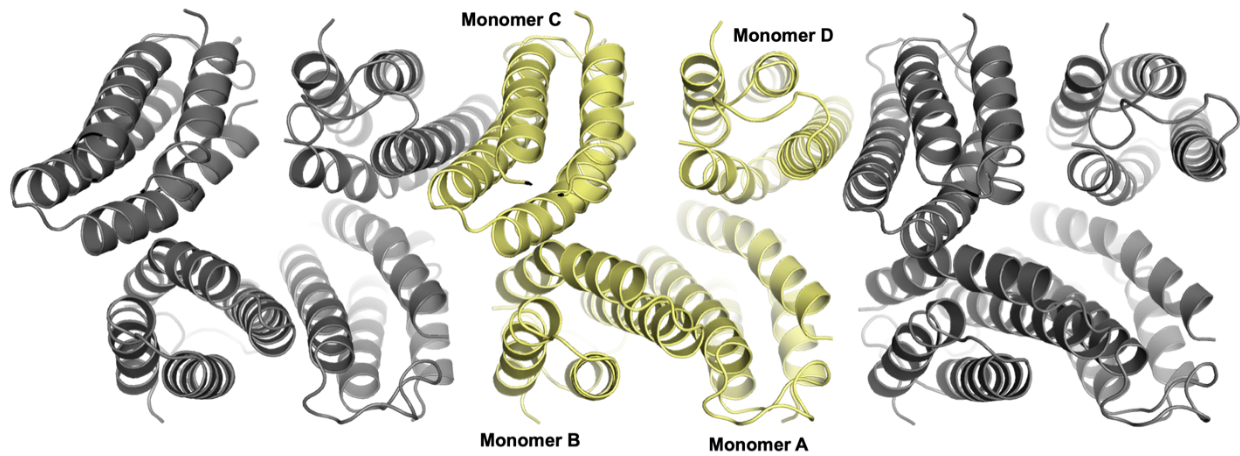

Figure S3. Overview of the ApoCyt crystal lattice. The asymmetric unit is highlighted in yellow and adjacent asymmetric units are shown in gray.

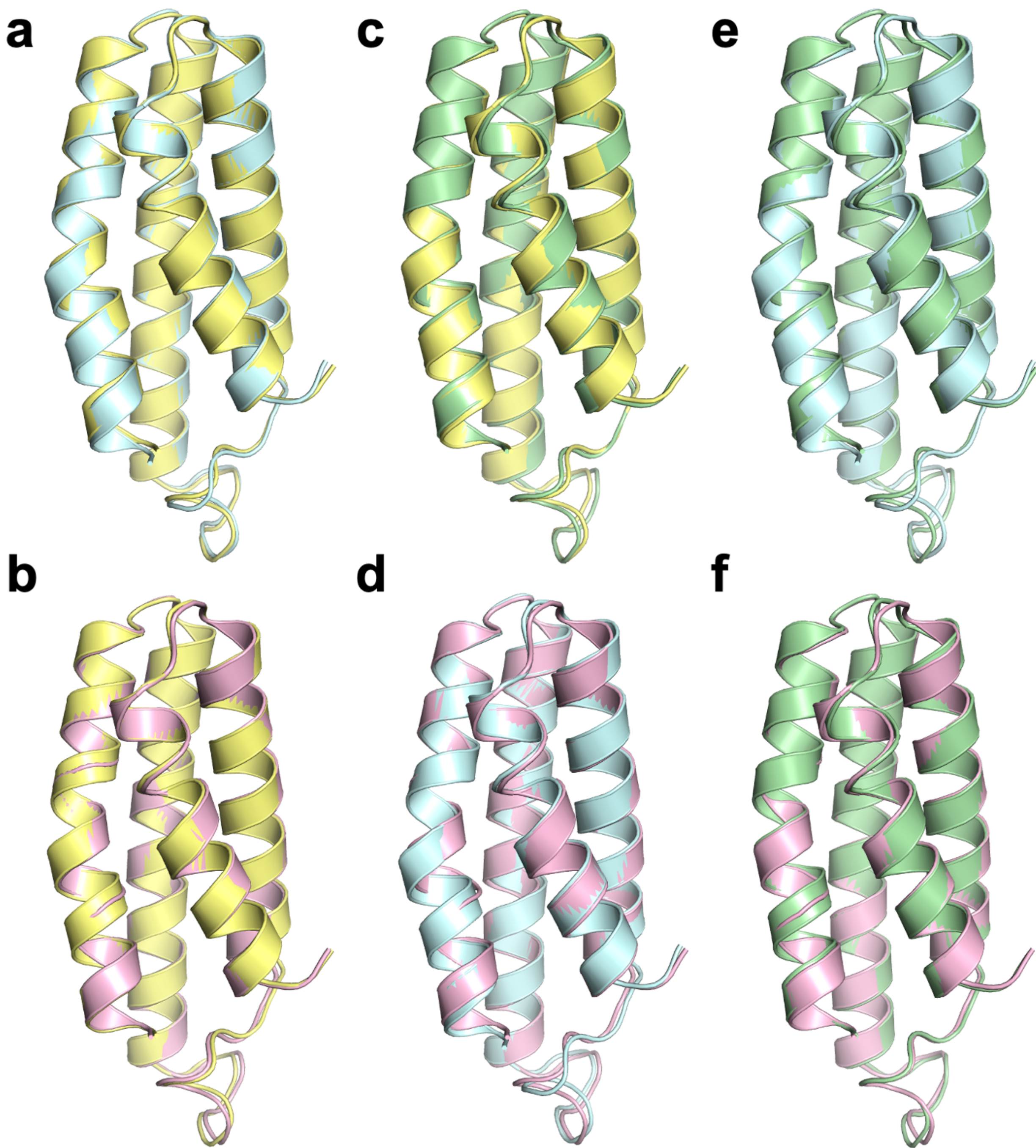

Figure S4. Pairwise structural alignments of the four ApoCyt chains in the asymmetric unit of the crystal structure. Chain A – yellow, Chain B – cyan, Chain C – magenta, Chain D – green. (a) Chain A vs. Chain B, (b) Chain A vs. Chain C, (c) Chain A vs. Chain D, (d) Chain B vs. Chain C, (e) Chain B vs. Chain D, (f) Chain C vs. Chain D.

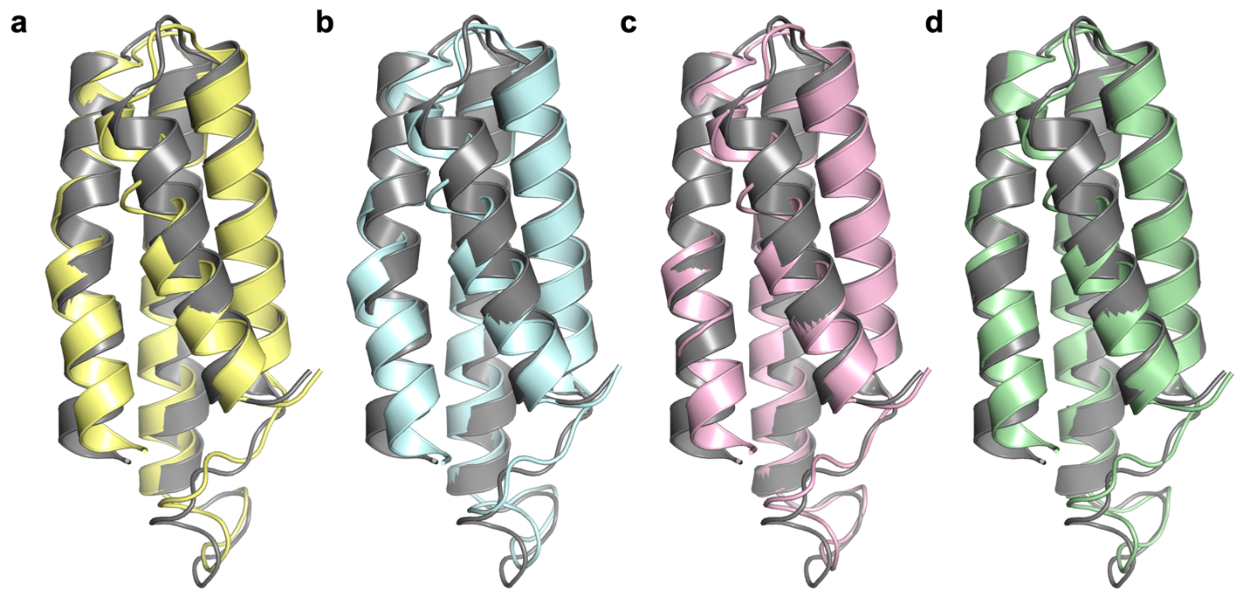

Figure S5. Structural alignments of (a) ApoCyt Chain A, (b) Chain B, (c) Chain C, and (d) Chain D to the ApoCyt design model. Chain A – yellow, Chain B – cyan, Chain C – magenta, Chain D – green, Design Model – gray.

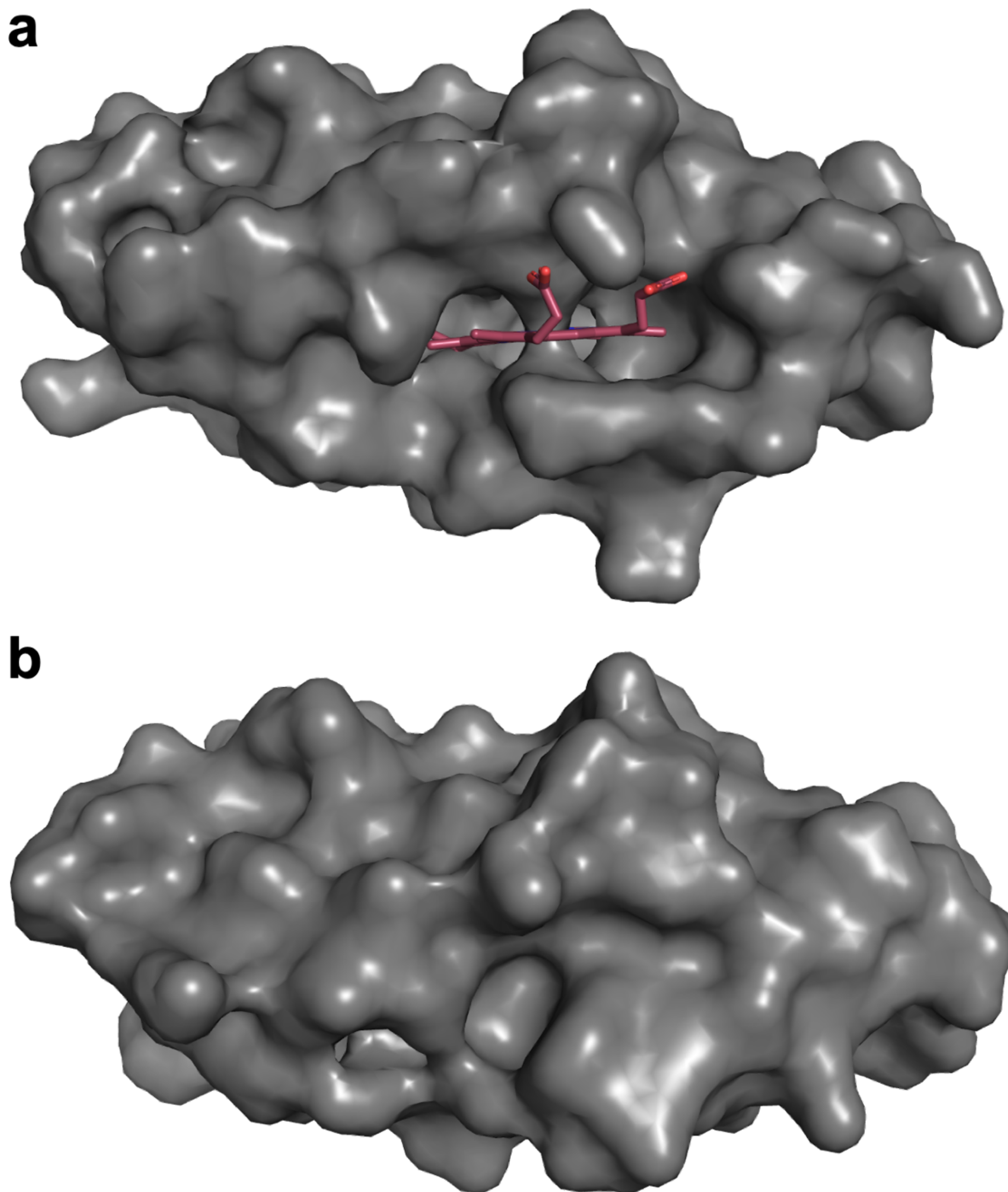

Figure S6. Surface representations of (a) cyt *cb*$_{562}$ and (b) ApoCyt. The heme pocket of cyt *cb*$_{562}$ has been filled due to the computationally designed mutations.

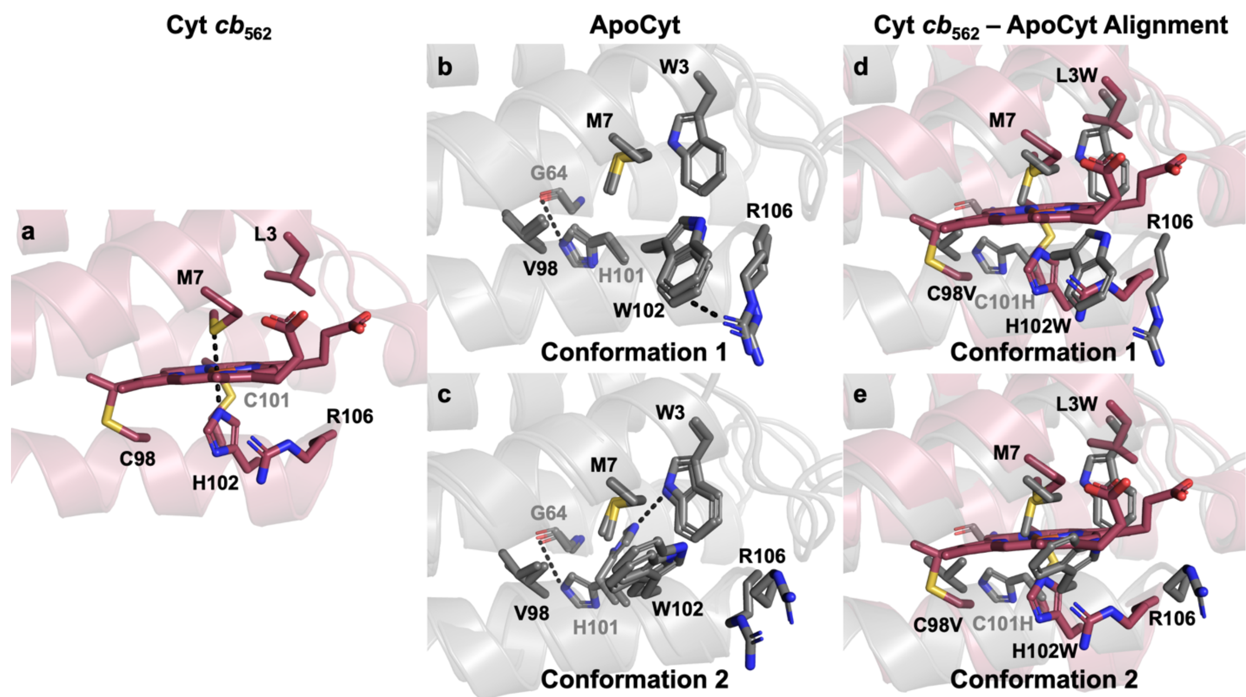

Figure S7. (a) Heme binding pocket of cyt cb_{562}. (b) Redesigned heme binding pocket of ApoCyt Monomers A and C (Conformation 1). (c) Redesigned heme binding pocket of ApoCyt Monomers B and D (Conformation 2). (d) Structural alignment of cyt cb_{562} and Conformation 1 of ApoCyt. (e) Structural alignment of cyt cb_{562} and Conformation 2 of ApoCyt.

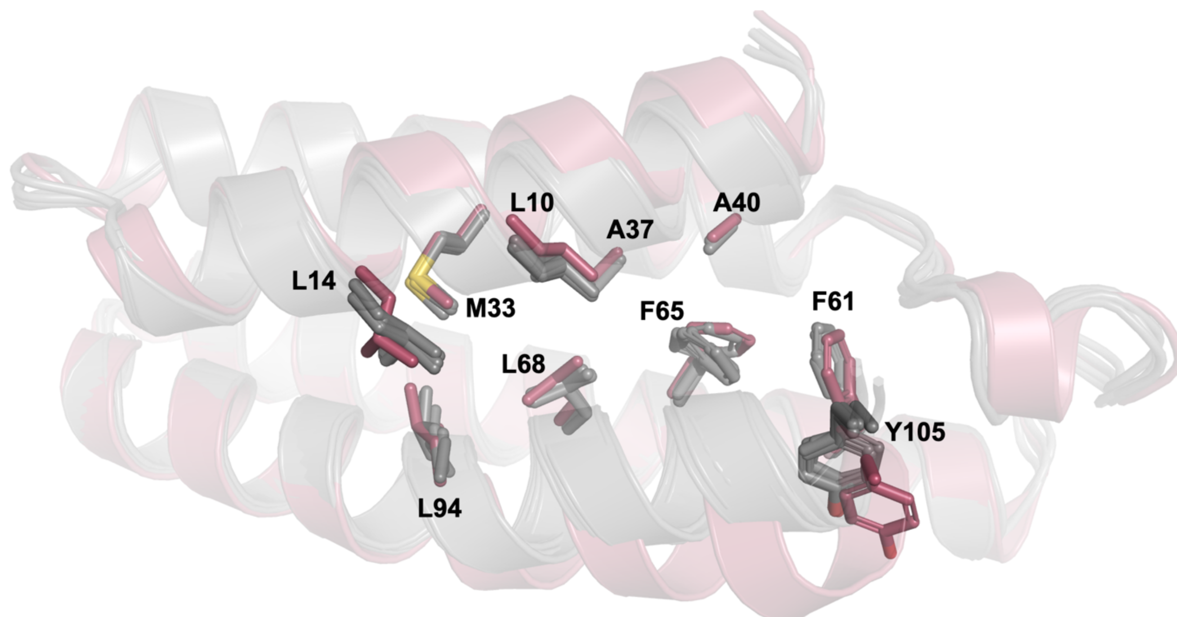

Figure S8. Structural alignment of cyt cb_{562} and all four ApoCyt monomers in the asymmetric unit. Residues within 4 Å of the heme cofactor in cyt cb_{562} that reside in the hydrophobic core of the protein are shown in sticks. These residues align well with the corresponding ApoCyt residues, with all-atom rmsd values between 1.36 and 1.43 Å for the four ApoCyt monomers.

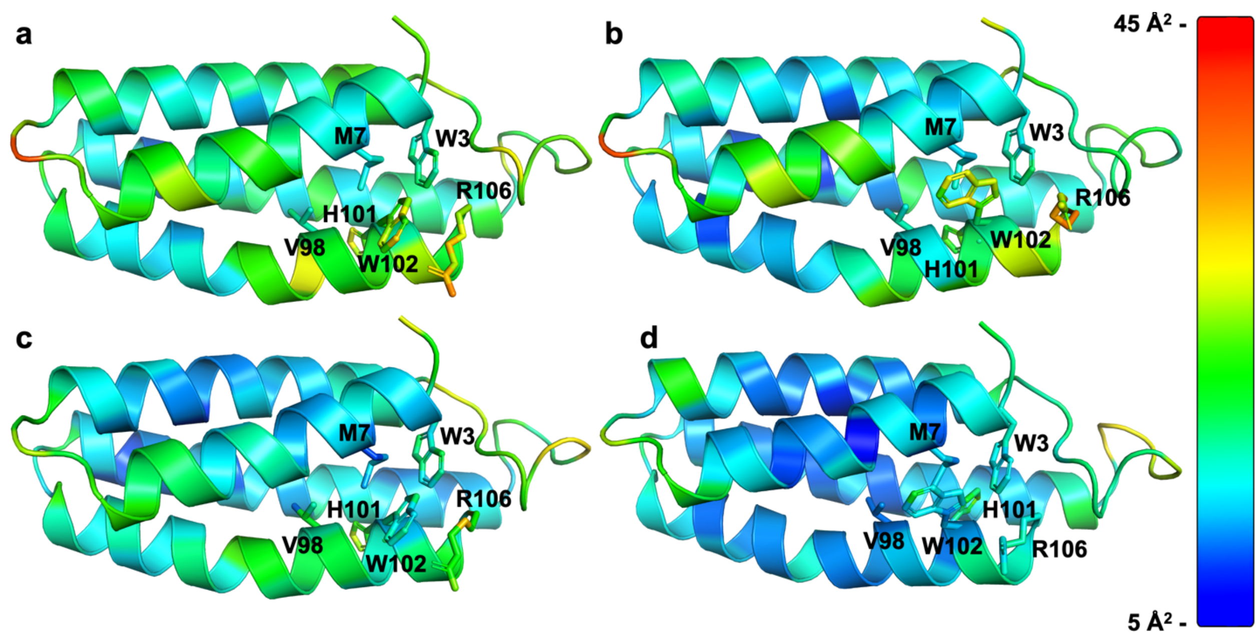

Figure S9. The four monomers of the ApoCyt asymmetric unit colored by B-factor. (a) Monomer A, (b) Monomer B, (c) Monomer C, (d) Monomer D.

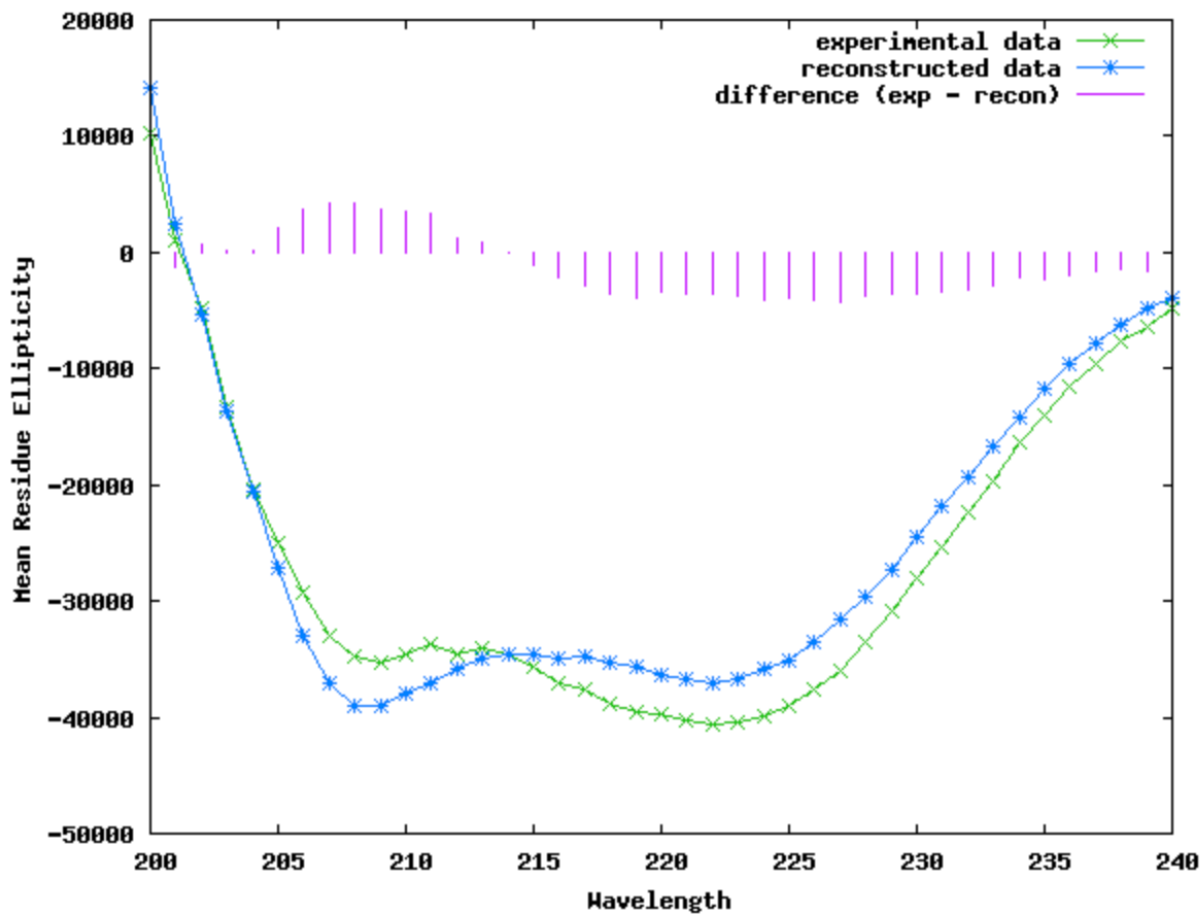

Figure S10. Experimental CD spectrum of cyt cb_{562} and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

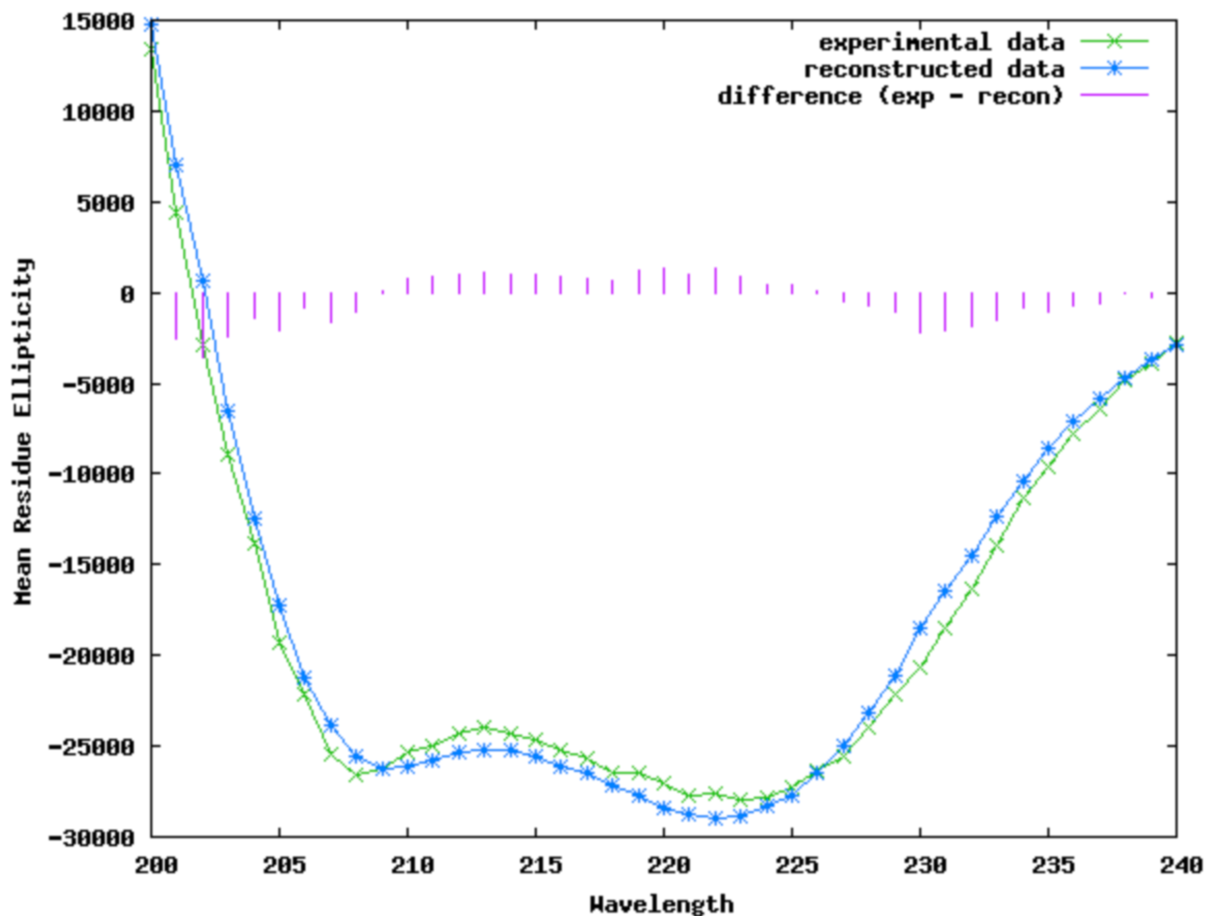

Figure S11. Experimental CD spectrum of ApoCyt and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

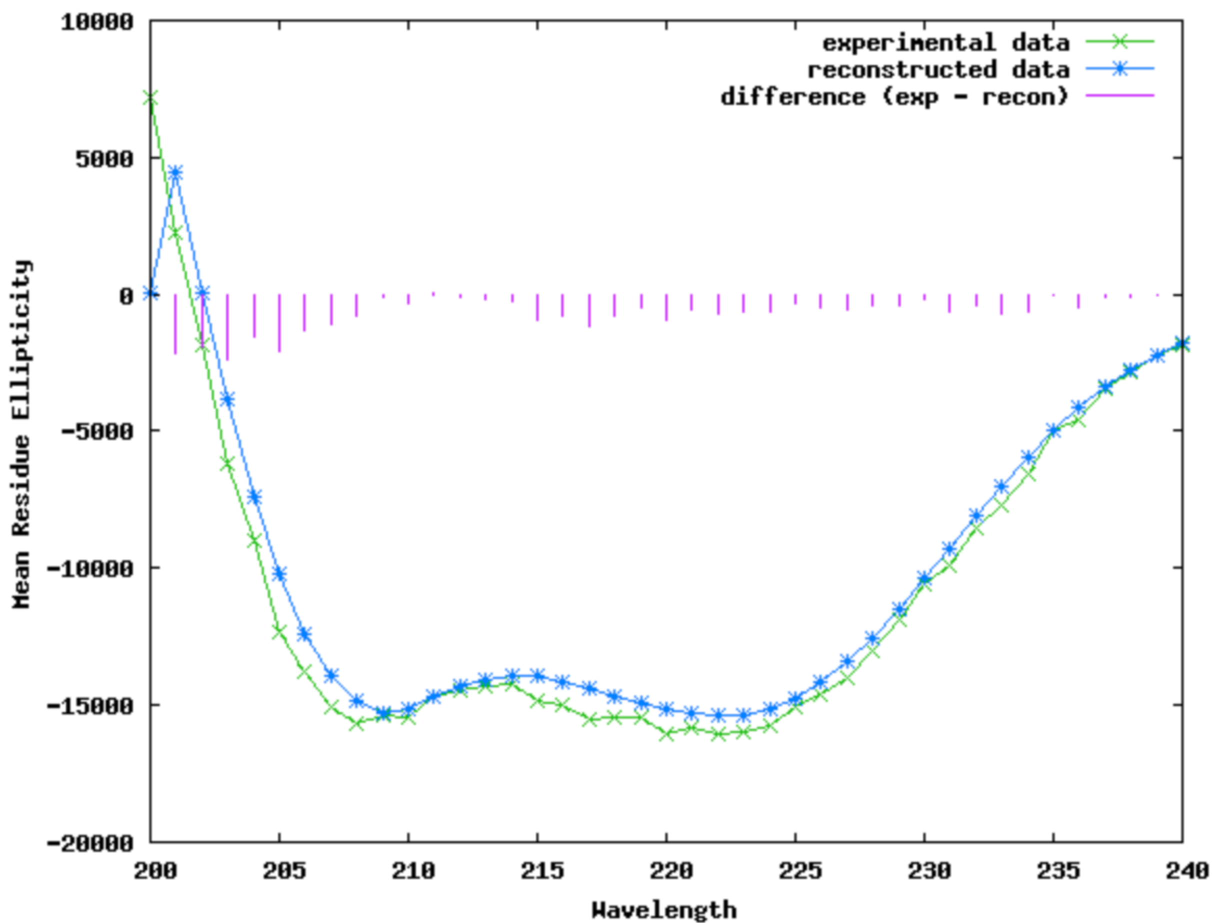

Figure S12. Experimental CD spectrum of Cyt-RIL and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

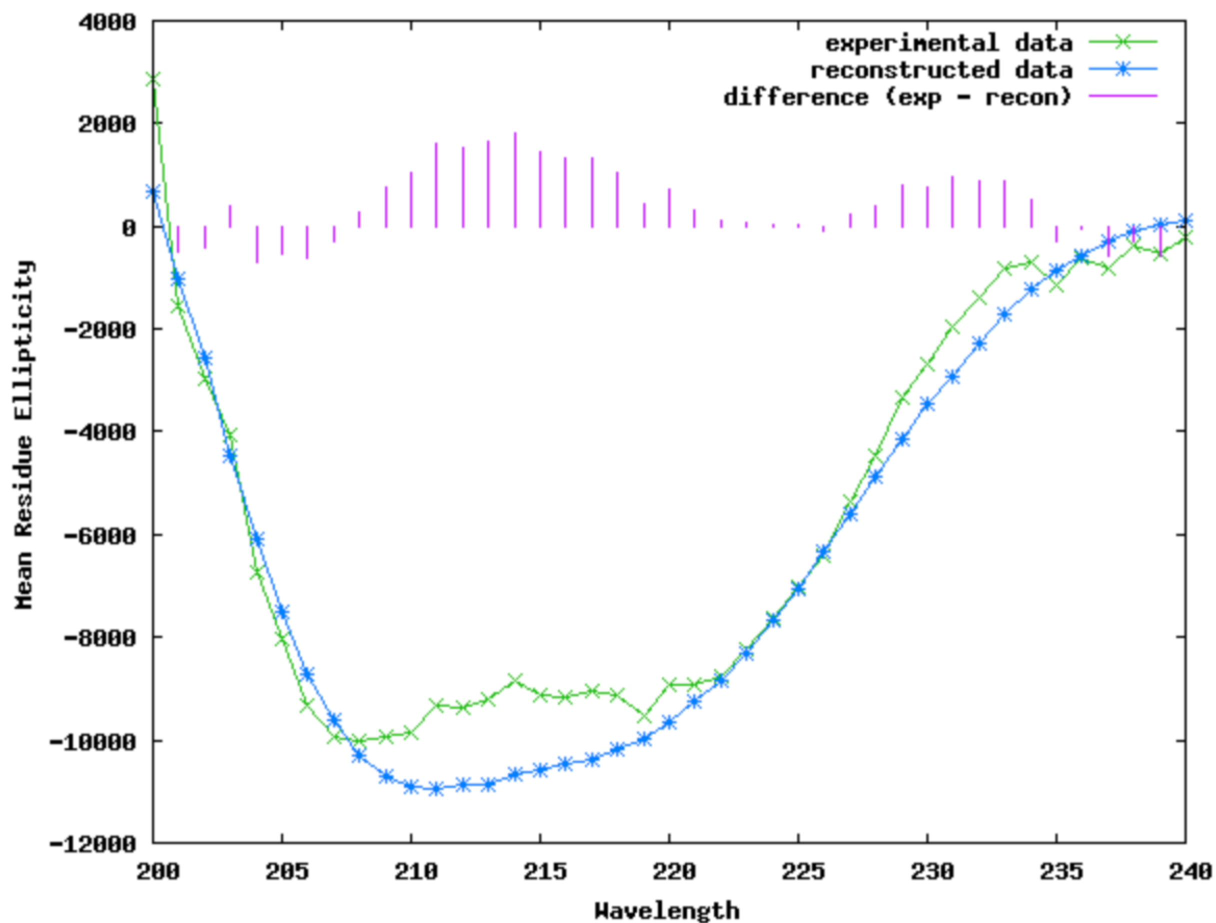

Figure S13. Experimental CD spectrum of ApoCyt-A3 and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

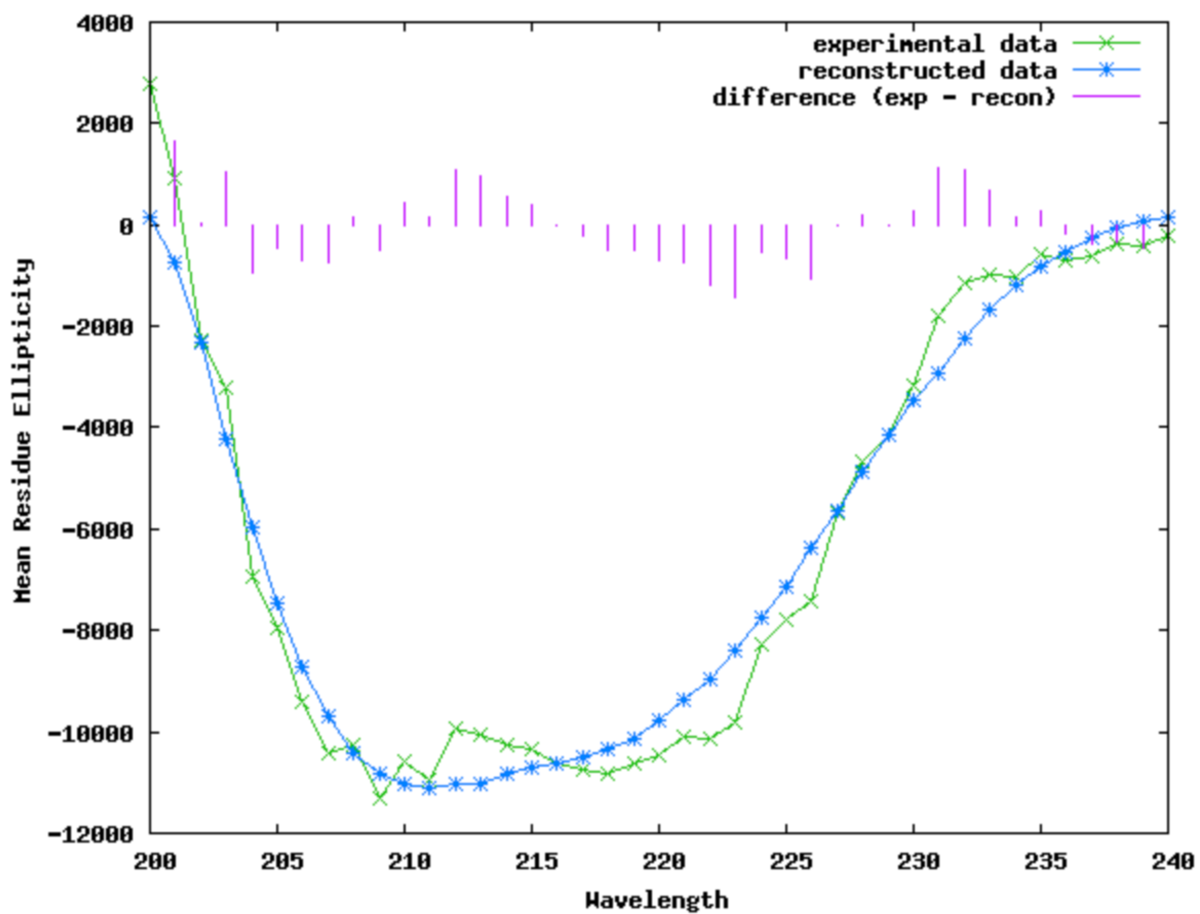

Figure S14. Experimental CD spectrum of ApoCyt-A7 and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

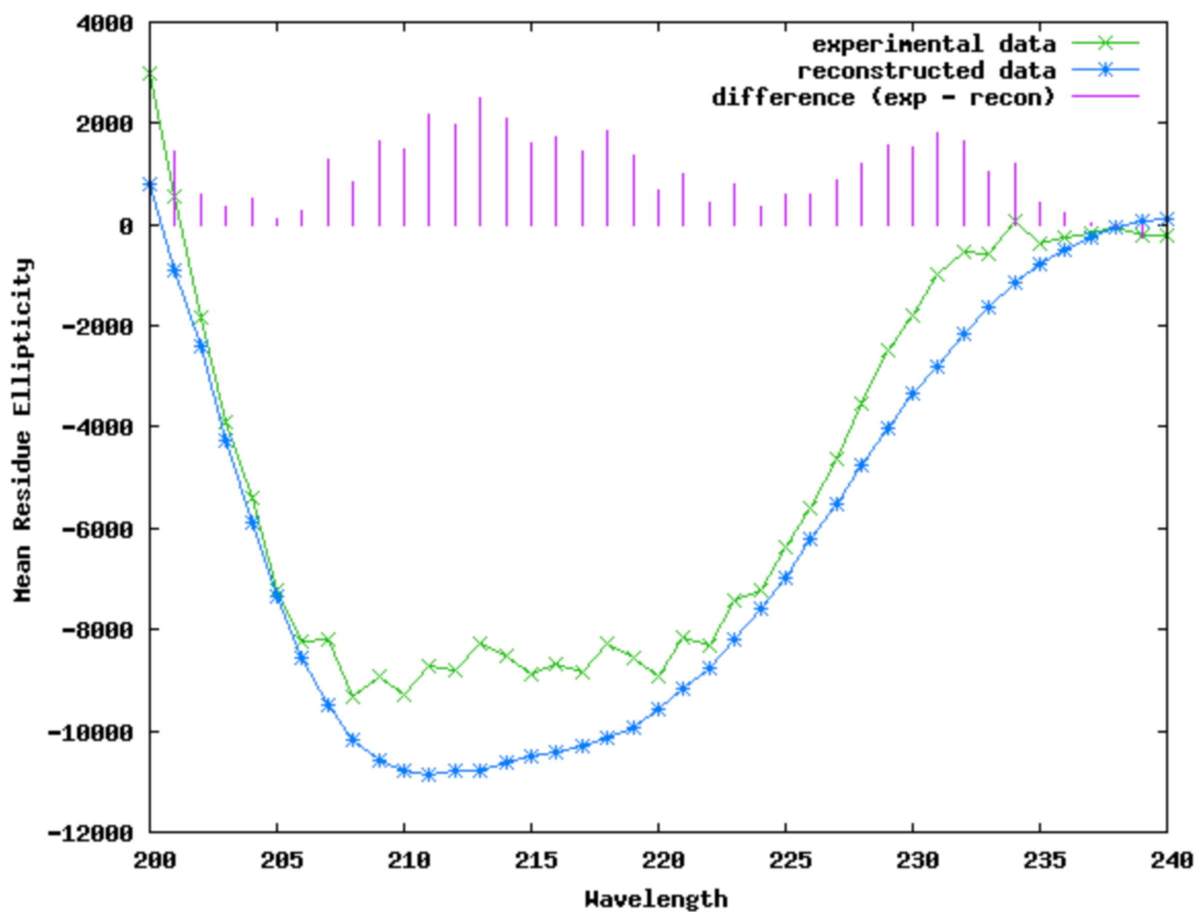

Figure S15. Experimental CD spectrum of ApoCyt-A101 and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

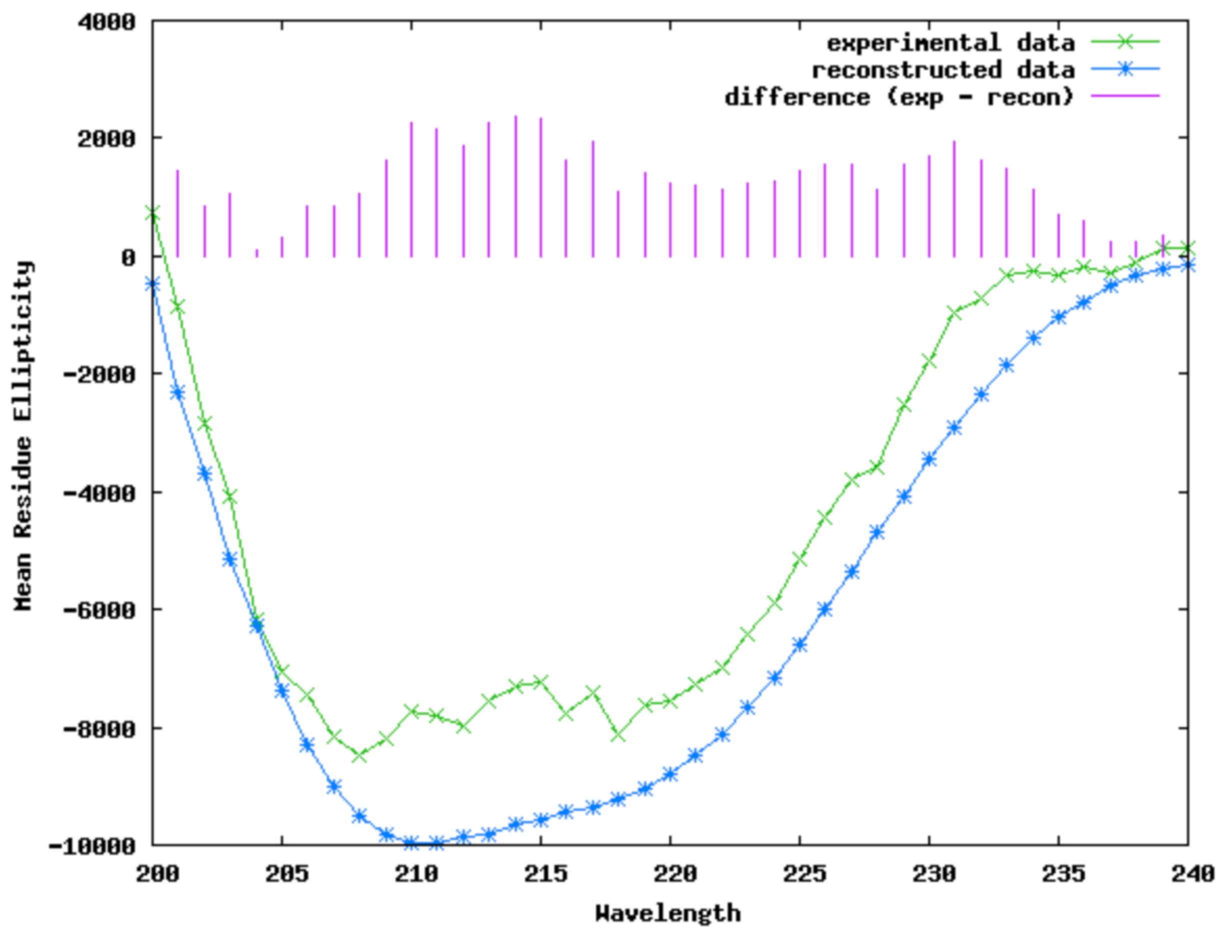

Figure S16. Experimental CD spectrum of ApoCyt-A102 and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

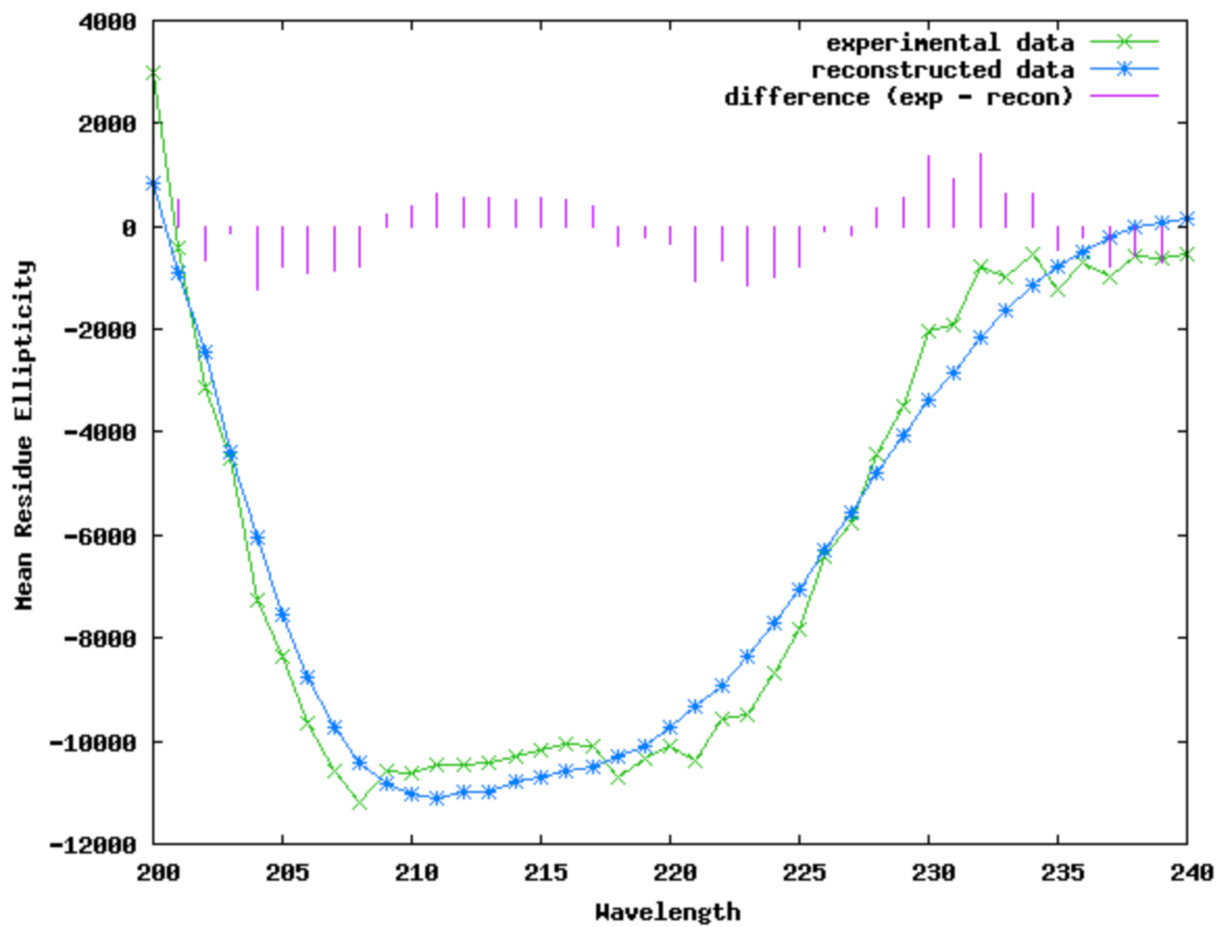

Figure S17. Experimental CD spectrum of ApoCyt-A106 and simulated CD spectrum by K2D that yielded the secondary structure estimates in Table S6.

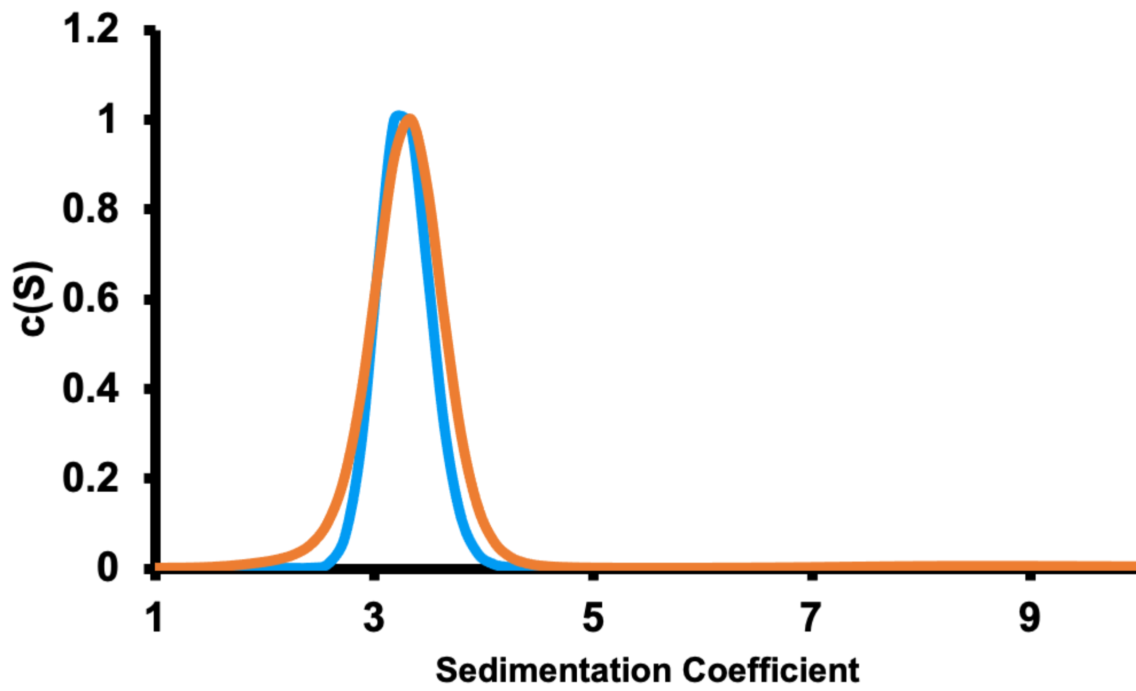

Figure S18. SV-AUC profiles of ApoCyt-TriCyt3 with (orange) and without (cyan) a 6xHis tag.

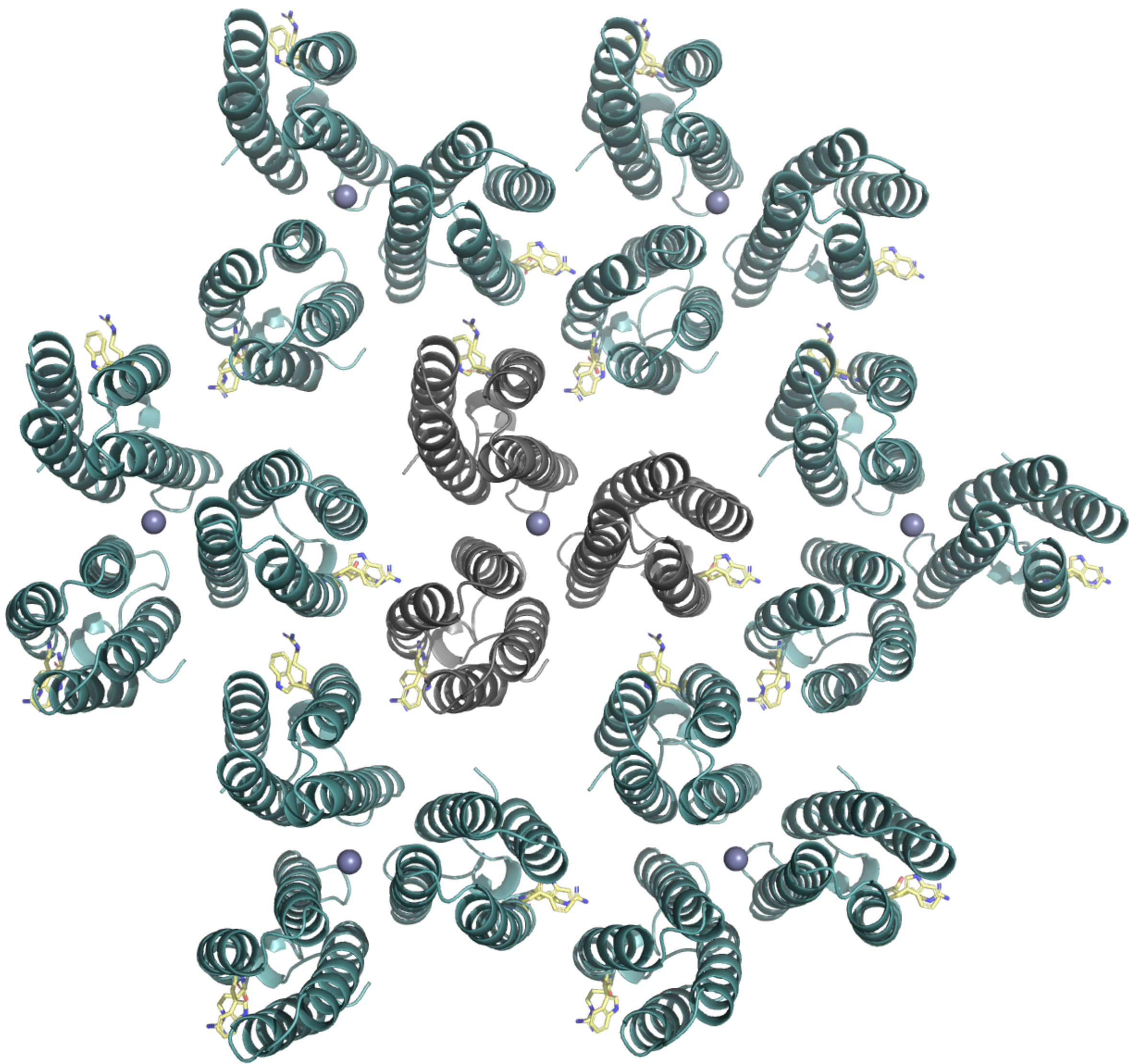

Figure S19. Overview of the ApoCyt-TriCyt3 crystal lattice. The asymmetric unit is shown in gray, and residues W102 and R106, which form significant crystal contacts with each other in ApoCyt but not in ApoCyt-TriCyt3, are highlighted in yellow.

Supplementary Data

PyRosetta Script for ApoCyt Design.

```
from pyrosetta import *
from pyrosetta.rosetta import * #Used from rosetta import * before, but that has been
deprecated.
from rosetta.core.scoring import *
from rosetta.core.scoring.constraints import *
from rosetta.core.scoring.func import *
from rosetta.core.pack.task import *
from rosetta.protocols.simple_moves import *
from pyrosetta.rosetta.protocols.minimization_packing import *
from rosetta.protocols.relax import *
from rosetta.protocols.docking import *
from rosetta.protocols.rigid import *
from pyrosetta.mpi import *
#Allow for mutiple process source
from mpi4py import MPI
import os
import csv
from rosetta.protocols.analysis import *
from rosetta.core.scoring.sc import *
import math
from rosetta.protocols.grafting.simple_movers import *
from rosetta.core.pose import append_pose_to_pose
from rosetta.basic import options
from rosetta.core.kinematics import Jump
from rosetta.core.kinematics import Stub
from rosetta.protocols.geometry import centroids_by_jump
from rosetta.numeric import xyzVector_double_t
from rosetta import numeric
import numpy
from rosetta.core.scoring.sasa import SasaCalc
from rosetta.protocols.toolbox.task_operations import LinkResidues

mpi_init()
list_scoreterms=[fa_atr, fa_rep, fa_sol, fa_intra_rep, fa_elec, pro_close, hbond_sr_bb,
hbond_lr_bb, hbond_bb_sc, hbond_sc, dslf_fa13, atom_pair_constraint, angle_constraint, rama,
omega, fa_dun, p_aa_pp, yhh_planarity, ref, total_score]

# Just need ta void score to try to fill the hole let by the heme.
void_score=ScoreFunction()
void_score=get_fa_scorefxn()
void_score.set_weight(voids_penalty,0.10)

#Load any protein as a pose

def make_resfile_job(AAstart,AAend,mutationlistA,rotamerdesignlistA,jobnumber):
```

```
    # e.g. use with
make_resfile(1,107,[34,38,41,52,54,59,62,63,66,69,76,80],[31,42,51,53,55,64,67,70,73,74,78,8
1],['A','B','C'])
    try:
        os.mkdir('resfiles')
    except:
        pass
    fi=open("resfiles/autogen_packer_"+str(jobnumber)+".resfile",'w')
    startAA=AAstart
    endAA=AAend
    mutationsA=mutationlistA
    rotamerdesignA=rotamerdesignlistA
    default="NATRO"
    fi.write(default+'\n')
    fi.write('EX 1 LEVEL 3'+'\n')
    fi.write('EX 2 LEVEL 2'+'\n')
    fi.write('EX 3 LEVEL 2'+'\n')
    fi.write('EX 4 LEVEL 2'+'\n')
    fi.write('start'+'\n')
    fi.write('\n')
    for aa in range(startAA,endAA):
        if aa in mutationsA:
            fi.write(str(aa)+' A ALLAAxc'+'\n') #Could use POLAR instead. Otherwise google rosetta
resfile syntax for more info. ALLAAxc would be All amino acids except cystein.
        if aa in rotamerdesignA:
            fi.write(str(aa)+' A NATAA'+'\n')
    fi.close()
    return


def create_packer_object_job(inputpose,jobnumber,scorefunction=void_score):
    firstpacker=TaskFactory.create_packer_task(inputpose)
    # next line genereates a resfile with
make_resfile_job(AAstart,AAend,mutationlistA,rotamerdesignlistA,jobnumber)
    make_resfile_job(1,106,[3,7,98,101,102,106],[],jobnumber)
    #make_resfile_job(1,212,[],[],[],[],jobnumber) # debugging only
    #EXAMPLE:
make_resfile_job(1,107,[31,34,38,41,52,54,59,62,63,66,69,76,78,80,81],[],['A','B','C'],jobnumber
)
    parse_resfile(inputpose, firstpacker, "resfiles/autogen_packer_"+str(jobnumber)+".resfile")
    ##Since we want a homodimer, we need to restrict the mutations in a way, that every position
in Chain A stays the same AA as that position in Chain B. That is done by 'linking' these amino
acids.
    # print 'Now working on linking the residues.'
    # linker=LinkResidues()
    # #This should go through all amino acids of chain a stopping just before the heme and make
a link-group with their cousin from chain B.
    # for j in range(1,106):
    #     linker.add_group(unicode(str(j)+','+str(j+106)))
    # linker.apply(inputpose, firstpacker) #This should apply the linker object to the firstpacker
Task, which later is bound to the firstpackermover, which then is applied to the pose.
```

```
    # print 'Done linking and applying the linkage to the packermover.'
    firstpackermover=PackRotamersMover(scorefunction,firstpacker)
    return firstpackermover


#Here starts the actual program:
start=Pose()
pose_from_file(start,'clean_monomer_rosettafit.pdb')
# for debugging: print start.pdb_info()

try:
    os.mkdir('01packer')
except:
    pass

try:
    os.mkdir('02relax')
except:
    pass

try:
    os.mkdir('03packer')
except:
    pass

try:
    os.mkdir('04relax_final-mc')
except:
    pass




start.dump_pdb('start.pdb')

def interfacedesignfunction(jobnumber):
    #Make the packingpose and assign from the startpose with the constraints and all
    looppose=Pose()
    looppose.assign(start)
    firstpackermover=create_packer_object_job(looppose,jobnumber,void_score)
    print 'Packer object created. Going to run the packer now.'
    firstpackermover.apply(looppose)
    looppose.dump_pdb("01packer/01packer1_"+str(jobnumber)+".pdb")
    firstrelax=FastRelax()
    firstrelax.set_scorefxn(void_score)
    firstrelax.max_iter(200) #I seemed to have problems before with convergence in the
minimizer. Default value is 2000 cycles before giving up. Apparently 200 are good enough and
no need to worry about the result. I tested it briefly. Looked fine; same score as when running
the looooong default 2000 cycles.
    firstrelax.apply(looppose)
    looppose.dump_pdb("02relax/02relax1_"+str(jobnumber)+".pdb")
```

```
    secondpackermover=create_packer_object_job(looppose,jobnumber,void_score)
    secondpackermover.apply(looppose)
    looppose.dump_pdb("03packer/03packer2_"+str(jobnumber)+".pdb")
    ### for the second and final relax we want the best of 3 (JE recommendation) to get a useful
answer
    secondrelax=FastRelax()
    secondrelax.set_scorefxn(void_score)
    mc=MonteCarlo(looppose,void_score,1)
    relaxpose=Pose()
    counter = 0
    while counter < 2:
        relaxpose.assign(looppose)
        secondrelax.max_iter(200)
        secondrelax.apply(relaxpose)
        mc.boltzmann(relaxpose)
        counter += 1
    looppose=mc.lowest_score_pose()
    looppose.dump_pdb("04relax_final-mc/04relax2final_"+str(jobnumber)+".pdb") #note, do not
include an _ in the file name, JE's script for extracting scores doesn't like that do to the split at _

JobDist=MPIJobDistributor(200,interfacedesignfunction)
```

PyRosetta Script for Structure Alignments.

```
from pyrosetta import *
from pyrosetta.rosetta import * #Used from rosetta import * before, but that has been
deprecated.
from rosetta.core.scoring import *
from rosetta.core.scoring.constraints import *
from rosetta.core.scoring.func import *
from rosetta.core.pack.task import *
from rosetta.protocols.simple_moves import *
from pyrosetta.rosetta.protocols.minimization_packing import *
from rosetta.protocols.relax import *
from rosetta.protocols.docking import *
from rosetta.protocols.rigid import *
from pyrosetta.mpi import *
#Allow for mutiple process source
from mpi4py import MPI
import os
import csv
from rosetta.protocols.analysis import *
from rosetta.core.scoring.sc import *
import math
from rosetta.protocols.grafting.simple_movers import *
from rosetta.core.pose import append_pose_to_pose
from rosetta.basic import options
from rosetta.core.kinematics import Jump
from rosetta.core.kinematics import Stub
from rosetta.protocols.geometry import centroids_by_jump
from rosetta.numeric import xyzVector_double_t
```

```
from rosetta import numeric
import numpy
import glob
import pandas as pd

mpi_init()

ReferencePose=Pose()
pose_from_file(ReferencePose, '/path/clean_monomer_rosettafit.pdb')
DesignedPose=Pose()

filelist=glob.glob('/path/20210128_NoChromeRedesign_LimitedMutations/04relax_final-
mc/*.pdb')
correct_order_filelist=[]
RMSDlist=[]
for file in filelist:
   pose_from_file(DesignedPose, file)
   RMSD=CA_rmsd(ReferencePose,DesignedPose,56,81)
   RMSDlist.append(RMSD)
   correct_order_filelist.append(file)

RMSD_dataframe=pd.DataFrame({'file':correct_order_filelist,'RMSD':RMSDlist})

RMSD_dataframe.to_csv('/path/RMSD_Output_Helix3.csv',index=False)
```

References

1. Andrade, M. A.; Chacon, P.; Merelo, J. J.; Moran, F., Evaluation of Secondary Structure of Proteins from UV Circular-Dichroism Spectra Using an Unsupervised Learning Neural-Network. *Protein Eng.* **1993,** *6* (4), 383-390.
2. Miles, A. J.; Ramalli, S. G.; Wallace, B. A., DichroWeb, a website for calculating protein secondary structure from circular dichroism spectroscopic data. *Protein Sci.* **2022,** *31* (1), 37-46.
3. Khrapunov, S., Circular dichroism spectroscopy has intrinsic limitations for protein secondary structure analysis. *Anal. Biochem.* **2009,** *389* (2), 174-176.