1. Computational Modeling

1.1 Design of water-soluble proteins

Systematic sampling of parametric helical backbones

Helical backbones used for design calculations were generated by sampling the generalized Crick coiled-coil parameters[15,16] as described previously[17,18]. Only supercoil and alpha-helical parameters of the inner and outer helices of the monomers were sampled. These parameters include supercoil twist (ω_0), supercoil radius (R_0), alpha-helical phase (φ_1), and the offset along the z-axis (Z_{offset}). The supercoil phases (φ_0) were fixed at 0° for the inner helices and -30° and -22.5° for the outer helices of hexamers and octamers, respectively. The supercoil pitch of the outer helices was constrained to match that of the inner helices to maintain contacts to the inner helices throughout the length of the helical bundles[18]. The Alpha-helical twists (ω_1) were fixed at ideal values of 102.85°/aa for left-handed supercoils (e.g. WSHC6) and 100°/aa for straight helical bundles (e.g. WSHC8). Cyclic symmetries were applied during design calculations. The parameters for the inner helices of WSHC6 were preset according to the fitting result of a previously reported six-helix bundle[13] using the Coiled-coil Crick Parameterization web server (https://grigoryanlab.org/cccp/index.fit.php). Supercoil and alpha-helical parameters used to generate the backbones of WSHC6 and WSHC8 (2nd & 3rd columns) and the fitted parameters for the final design model and the crystal structure of WSHC8 (4th & 5th columns) are:

	WSHC6 bb generation	WSHC8 bb generation	WSHC8 design model	WSHC8 crystal structure
number of residues of the inner helix	34	49	49	49
number of residues of the outer helix	34	49	49	49
supercoil twist ω_0 of the inner helix	-3.18 °/aa	0 °/aa	-0.39 °/aa	-0.96 °/aa
supercoil twist ω_0 of the outer helix	-2.75 °/aa	0 °/aa	-0.42 °/aa	-0.88 °/aa
alpha-helical twist ω_1 of the inner helix	102.85 °/aa	100.00 °/aa	99.70 °/aa	99.64 °/aa
alpha-helical twist ω_1 of the outer helix	102.85 °/aa	100.00 °/aa	100.16 °/aa	100.18 °/aa
supercoil radius R_0 of the inner helix	9.5 Å	14.2 Å	14.4 Å	14.6 Å
supercoil radius R_0 of the outer helix	18.5 Å	23.2 Å	22.8 Å	23.3 Å
supercoil phase φ_0 of the inner helix	0 °	0 °	0 °	0 °
supercoil phase φ_0 of the outer helix	-30 °	-22.5 °	-22.5 °#	-22.5 °#
alpha-helical phase φ_1 of the inner helix	45.8 °	-98 °	-93 °	-94 °
alpha-helical phase φ_1 of the outer helix	152.8 °	74 °	71 °	65 °
z-axis offset Z_{offset} of the inner helix	0 Å	0 Å	0 Å	0 Å
z-axis offset Z_{offset} of the outer helix	2.2 Å	-3.2 Å	-2.0 Å	-1.9 Å

# Approximate calculation using Pymol.

We learned from the failed first round of design of helical bundles with large pores that enforcing supercoil twist and knobs-into-holes packing of conventional left-handed coiled coils will cause a big curvature of the generated helices, as shown in Extended Data Figure 5a-b. In this case, precisely designed inter-subunit interactions may be required to compensate for the strain associated with curving the helices, which, therefore, makes designing large pore-containing helical proteins more challenging. In the second round of design, we fixed the supercoil twist at 0 °/aa and saw a better success rate. Although the helices in the crystal structure of WSHC8 are more tilted compared to the design model, the supercoil twist is still much smaller than those of the conventional coiled coils and does not cause helix curving.

Connecting parametric helical backbones:

The parametrically generated inner and outer helical backbones were connected before design calculations with short 2-5-residue loops using methods described previously[18,35].

Rosetta design calculations:

Cyclic symmetries were applied at the beginning of the calculations. The backbone geometry was then regularized by minimization in Cartesian space. HBNet[18] was used to search for hydrogen-bond networks that span the helical interfaces. With atom pair constraints applied to the newly identified hydrogen-bond networks, Monte Carlo rotamer packing and minimization were performed to design the remaining positions aiming to lower the total energy and optimize the hydrophobic packing around the hydrogen-bond networks. A final minimization and side-chain repacking step was carried out without atom pair restraints on hydrogen-bonding residues to evaluate how well the networks remained intact in the absence of the constraints. Low energy designs with intact hydrogen-bond networks were tested using Rosetta "fold-and-dock" structure prediction calculations[19]. Sequences for which the predicted lowest-energy structures converge on the design models without additional energy minima were selected for experimental characterizations.

1.2 Design of transmembrane proteins

We designed transmembrane versions of WSHC6 and WSHC8 pores by resurfacing the outside of the crystal structures with patterned hydrophobic residues, and adding RK- and YW-rings at the intracellular and extracellular boundary region, respectively[8]. Briefly, the N- and C-termini of the transmembrane designs are designed to localize in the cytoplasmic side, by adding an RK ring with Arg and Lys residues close to the termini. The YW ring, designed at the lipid-water boundary on the periplasmic side, set up the register in the membrane. We designed the hydrophobic transmembrane span in between the YW and RK rings with a length of 26 and 31 Å for TMHC6 and TMHC8, respectively. Hydrophobic residues are designed based on amino acid propensity in the membrane, replacing all polar residues exposed to the membrane. The crystal structure of WSHC8 was used as the starting model.

The residues surrounding the central pores were also redesigned to modulate compatibility within the membrane and channel conductance: if these residues are too hydrophobic, some lipids may block the pores; if too polar, the designed TM segment may not be able to insert into the membrane. Polar residues were designed in the TM pore region, while membrane compatibility was predicted by TMHMM[36] software. For TMHC6, we designed a ring of glutamate residues (E-ring) and two rings of lysine residues (K-rings) at the opening of the central channel on the extracellular and intracellular side, respectively, to modulate channel conductance. For TMHC8, in each monomer, we redesigned 4 charged residues surrounding the pore to apolar residues for insertion into the membrane, leaving 4 polar residues in the TM region to make a relative hydrophilic central channel. We ordered two designs for TMHC8, both of which expressed well, enriched in the membrane fraction. TMHMM[36] software predicts that both TMHC6 and TMHC8 have two transmembrane segments. For the design of TMH4C4, the two monomers of TMHC8 were connected by adding loops using look-ups to a structural database and Rosetta design. The resulting TMH4C4 has four transmembrane segments.

1.3 Figure preparation for structures and maps and HOLE calculations

All the figures for structures and maps are prepared in Chimera[34] or Pymol. The constriction sites in the channels and pores are calculated using HOLE[20]. For the WSHC6 channel structure, the narrowest constriction is at Leu51 with a diameter of approximately 4 Å as calculated. For the TMHC6 design model, the narrowest constriction is at Glu44 with a diameter of approximately 3.3 Å. For the WSHC8 channel structure, the narrowest constriction of this channel has an inner diameter of approximately 10 Å at Lys80. For the TMH4C4 pore, the narrowest constriction is at Lys80 and Lys184 with a diameter of approximately 10 Å, calculated from EM structure.

1.4 Sequences of designed proteins

WSHC6	TEDEIRKLRKLLEEAEKKLYKLEDKTRRSEEISKTDDDPKAQSLQLIAESL MLIAESLLIIAISLLLSSRNG
WSHC8	SAEELLRRSREYLKKVKEEQERKAKEFQELLKELSERSEELIRELEEKGA ASEAELARMKQQHMTAYLEAQLTAWEIESKSKIALLELQQNQLNLELRHI
TMHC6	TENEIRKLRKLLRIAMLLLVFLLIATVVSLWTSKTDDDPSAQSEQLVAMSL MLIAASLLIIAISKLLKSRNG
TMHC8	SAEELLRRSREYLKKVALIQLVIAFVFLILLILLSWRSEELIRELEEKGAASE AELARMKQQHMTAYLQAALTAWEIISKSVIALLLLQQNQLNLELRHI
TMH4C4	SAEELLRRSREYLKKVALIQLVIAFVFLILLILLSWRSEELIRELEEKGAASE AELARMKQQHMTAYLQAALTAWEIISKSVIALLLLQQNQLNLELNTDTDK NVAEELLRRSREYLKKVALIQLVIAFVFLILLILLSWRSEELIRELEEKGAAS EAELARMKQQHMTAYLQAALTAWEIISKSVIALLLLQQNQLNLELRH

2. Experimental Materials and Methods

2.1 Reagents

Chemicals used were of the highest grade commercially available and were purchased from Sigma-Aldrich (St. Louis, MO, USA), Invitrogen (Carlsbad, CA, USA), or Qiagen (Hilden, Germany). Detergents were from Anatrace (Maumee, OH, USA).

2.2 Cloning and expression of water-soluble proteins

Synthetic genes were obtained from Genscript Inc. (Piscataway, N.J., USA) or IDT (Coralville, Iowa, USA) and cloned into the pET28b expression vector via NdeI/XhoI restriction sites. The plasmids were transformed into chemically competent *E. coli* Lemo21(DE3) cells (NEB, Ipswich, MA). Gene expression was facilitated by growing pre-cultures in Terrific Broth (TB) medium with a final concentration of 50 µg/ml kanamycin overnight at 37°C. 10 ml pre-cultures were used to inoculate 1L of autoinduction medium[37], again containing 50 µg/ml kanamycin for plasmid selection. The cultures were grown at 37°C for 3 hours and then 18°C overnight for protein expression. Cells were harvested by centrifugation.

2.3 Cell lysis and purification of water-soluble proteins

Cells were resuspended and homogenized in lysis buffer containing 50 mM Tris-HCl pH 8.0 and 300 mM NaCl 20 mM imidazole, and lysed by sonication in the presence of DNAse and

EDTA-free protease inhibitor tablet (ThermoFisher Scientific). Lysates were cleared by centrifugation at 4°C 18,000 g for at least 30 minutes and applied to Ni-NTA (Qiagen) columns pre-equilibrated in the lysis buffer. The column was washed with 5-10 column volumes (CV) of wash buffer (50 mM Tris, 300 mM NaCl, 30 mM Imidazole, pH 8.0), followed by 5 CV of high-salt wash buffer (50 mM Tris, 1 M NaCl, 30 mM Imidazole, pH 8.0), and then 5 CV of high-imidazole wash buffer (50 mM Tris, 300 mM NaCl, 60 mM Imidazole, pH 8.0). Protein was eluted with 50 mM Tris, 300 mM NaCl, 500 mM Imidazole, pH 8.0. After concentration, proteins were further purified by SEC (Superdex-200 increase 10/300 GL; GE Healthcare) using 1x phosphate buffer saline (PBS), pH 7.4. The N-terminal hexa-histidine tag was removed with restriction grade thrombin (EMD Millipore) overnight at 30°C. After full cleavage, the reaction is stopped by addition of phenylmethanesulfonyl fluoride (PMSF), followed by another round of SEC purification.

2.4 Cloning and expression of transmembrane proteins

Synthetic genes were obtained from Genscript Inc. (Piscataway, N.J., USA) or IDT (Coralville, Iowa, USA) and cloned into the pET29b expression vector via NdeI/XhoI restriction sites. The plasmids were transformed into chemically competent *E. coli* Lemo21(DE3) cells (NEB, Ipswich, MA). Gene expression was facilitated by growing pre-cultures in Luria-Bertani broth (LB) medium with a final concentration of 50 µg/ml kanamycin overnight at 37°C. 10 ml pre-cultures were used to inoculate 1L of LB medium, again containing 50 µg/ml kanamycin for plasmid selection. The cultures were grown at 37°C for 3-4 hours until an OD_{600} of 0.8-1.0 was reached and then 18°C overnight for protein expression induced by IPTG. Cells were harvested by centrifugation.

2.5 Cell lysis and purification of transmembrane proteins

Cells were resuspended and homogenized in lysis buffer containing 25 mM Tris-HCl pH 8.0 and 150 mM NaCl, and lysed by sonication in the presence of serine protease inhibitor phenylmethanesulfonyl fluoride (PMSF). Cell debris was removed by low-speed centrifugation for 10 minutes. The supernatant was collected and ultracentrifuged for 1 h at 150,000 g. The membrane fraction was collected and homogenized in buffer containing 25 mM Tris-HCl pH 8.0 and 150 mM NaCl. n-Decyl-β-D-Maltopyranoside (DM; Anatrace) was added to the membrane suspension to a final concentration of 1.5% (w/v) and then incubated for 2 h at 4 °C. After another ultracentrifugation step at 150,000g for 30 min, the supernatant was collected and loaded on Ni^{2+}-nitrilotriacetate affinity resin (Ni-NTA; Qiagen), followed by a wash with 25 mM Tris-HCl pH 8.0, 150 mM NaCl, 30 mM imidazole and 0.2% DM. Proteins were eluted with buffer containing 25 mM Tris-HCl pH 8.0, 150 mM NaCl, 30 mM imidazole and 0.2% DM. After concentration to 2-3 mg ml^{-1}, proteins were further purified by SEC (Superdex-200 10/30; GE Healthcare). The buffer for SEC contained 25 mM Tris-HCl pH 8.0, 150 mM NaCl and 0.2% DM. To prepare samples for Cryo-EM, TMH4C4 protein was purified in the buffer containing 0.02% (weight/volume) n-dodecyl-β-D-maltopyranoside (DDM, Anatrace). The major peak fractions were taken for further Cryo-EM investigation. For AUC experiments, the proteins, originally in 0.2% DM buffer, were buffer exchanged into the buffer for AUC by SEC, which contained 20 mM sodium phosphate, pH 7.0 and 200 mM NaCl supplemented with 0.5% Pentaethylene Glycol Monooctyl Ether (C8E5).

2.6 Circular dichroism (CD) measurements

CD wavelength-scan measurements were made on an AVIV CD spectrometer model 420. Protein concentrations ranged from 0.1-0.2 mg/ml in PBS (pH 7.4) for soluble proteins and PBS (pH 7.4) plus 0.2% DM for transmembrane proteins. Wavelength-scan spectra from 260 to 190 nm were recorded in triplets and averaged. The scanning increment for full wavelength scans was 1 nm. Temperature melts were conducted in 2 °C steps (heating rate of 2 °C/min) and recorded by following the absorption signal at a wavelength of 220 nm. Three sets of wavelength scan spectra were recorded at 25 °C, 95 °C and after cooling down to 25 °C.

2.7 Analytical ultracentrifugation (AUC) molecular weight determination

AUC experiments were carried out using a Beckman XL-I analytical ultracentrifuge (Beckman Coulter) equipped with an eight-cell An-50 Ti rotor. The protein samples were centrifuged at the speed of 40k rpm for sedimentation velocity (SV) experiments. Sedimentation-equilibrium (SE) experiments were then performed on samples that showed monodisperse distribution (>90%) of sedimentation coefficients by SV to determine the MW of the discrete species. The water-soluble proteins were centrifuged in PBS (pH7.4). The transmembrane proteins were centrifuged in a buffer containing 20 mM sodium phosphate, pH 7.0, 200 mM NaCl and 0.5% C8E5. Density matching was unnecessary as the solvent density was equal to that of C8E5. For sedimentation equilibrium, data were collected by UV absorbance at 230 nm at 20 °C for three protein concentrations in the range of 0.2-0.8 OD_{230}. Samples were centrifuged at three different speeds corresponding to σ, the reduced molecular weight[38], of 2, 3 and 4.5 cm^{-2}, respectively.

Data were globally fit to a single ideal species model using the program SEDPHAT (http://www.analyticalultracentrifugation.com/sedphat/sedphat.htm). The partial specific volume, buffer density, and viscosity were calculated using SEDNTERP (http://www.jphilo.mailway.com/).

2.8 Small-angle X-ray Scattering

Samples were purified by SEC in PBS (pH7.4) with 1% glycerol; fractions preceding the void volume of the column were used as blanks for buffer subtraction. Scattering measurements were performed at the SIBYLS 12.3.1 beamline at the Advanced Light Source[39]. For each sample, data were collected for two different concentrations to test for concentration-dependent effects; “high” concentration samples ranged from 4-7 mg/ml and “low” concentration samples ranged from 1-2 mg/ml. Data were analyzed using ATSAS software[40] and the experimental scattering profiles were fit to design models using CRYSOL[41].

2.9 X-ray crystallography

WSHC6 samples for crystallization were purified with an additional chromatography step using a Hiload 16/600 Superdex 75 pg SEC column (GE Healthcare) with a running buffer of 20 mM HEPES pH7.5, 150 mN NaCl, 2 mM TCEP, and 1 mM EDTA. Pure fractions were pooled and concentrated to 10-15 mg/mL. Crystallizations were prepared by vapor diffusion in sitting drops at 18°C. After optimising the salt, pH and precipitant concentration combining streak seeding, the crystals for data collection were grown in a condition of 0.1 M MES, pH 6.6, 17.5% PEG 550 MME with a protein concentration of 1.7 OD at 280 nm. WSHC6 crystals were frozen in liquid nitrogen in the crystal growing buffer with additional 30% glycerol as cryo-protectant. Typical crystals diffracted to 2.45 Å. X-ray diffraction data from single crystals were collected at I02 beamline, Diamond Light Source, UK and Proxima-2A beamline in France National synchrotron facility, Soleil, France. Crystals are grown in the space group of $P2_122_1$. The structure of WSHC6 was determined using molecular replacement with the Rosetta design model. Molecular replacement using PHASER in PHENIX package[42–44] yielded translation function Z-score of 6.9. The biological unit of the WSHC6 molecule is made by 2-fold crystallographic symmetry. The model building was done with COOT[45]. Structure refinement performed with PHENIX[42] with ML (Maximum likelihood) as the target function in rigid-body refinement plus TLS for motion correction in the flat bulk solvent model and cartesian NCS average.

WSHC8 samples for crystallization were purified with an additional SEC step using a Superdex 200 Increase 10/300 GL column (GE Healthcare) with 20 mM Tris-HCl buffer pH8.0, 100 mN NaCl. Pure fractions were pooled and concentrated to 34 mg/mL. Crystallization screens were prepared using a 5-position deck Mosquito Crystal (TT Labtech) with an active humidity chamber. Crystals were grown by vapor diffusion in hanging drops at 17°C. Single crystals for data collection were grown in the No. 85 crystallization solution of JCSG+ suite (Qiagen) containing 0.3 M magnesium formate and 0.1 M Bis-Tris pH5.5. WSHC8 crystals were frozen in liquid nitrogen in the crystal growing buffer with additional 20% glycerol as cryo-protectant. The X-ray datasets were collected at the ALS-ENABLE beamlines. Crystals are grown in the space group of $P2_12_12$. Model building and structure refinement were performed as described above for WSHC6 (Extended Data Table 1).

2.10 Whole-cell patch-clamp experiments

Recombinant baculovirus was generated by using the Bac-to-Bac system (Invitrogen). *Trichoplusia ni* insect cells (Hi5, Thermo Fisher) were grown on 35-mm Petri dishes in Grace's insect medium (Gibco) supplemented with FBS (10%) and antibiotics (100 µg/ml streptomycin and 100 U/ml penicillin). Cells were infected by replacing the incubation medium with a medium containing the baculovirus encoding the designed channels protein constructs. After 1 h, 2 ml incubation medium was added to the virus-containing medium. Cells were maintained at 25–27°C for at least 24 h before the study.

Whole-cell patch-clamp currents were recorded using an amplifier (Axopatch 200; Molecular Devices) with glass micropipettes resistance (1.5–3 MΩ). Capacitance was subtracted and series resistance was compensated using internal amplifier circuitry. To test the ion selectivity of the designed channels, Hi5 cells expressing the design of interest were bathed in a solution containing 100 mM of the chloride salt of the monovalent cations K^+, Na^+, Cs^+, and $CH_3NH_3^+$. The patch pipettes contained the equivalent concentration of the fluoride salt of the same cation, except for $CH_3NH_3^+$, for which the pipette contained 100 mM CsF. The standard voltage-clamp protocol for measuring ionic currents consisted of 20-ms test pulses from a holding potential of 0 mV to voltages ranging from −100 mV to +100 mV in 10-mV steps. No leak subtraction protocols were used during the measuring of currents. The data were filtered at 2 KHz. For analysis of measured ionic current, the value of the current at the end of each pulse was measured. All current measurements were normalized to the cell capacitances. Each measurement is carried out on a different cell and the reported values are the means for the number of cells studied. The error bars in figures indicate the standard error of measurement (s.e.m.). Voltage-clamp pulses were generated, and currents were recorded using Pulse software controlling an Instrutech ITC18 interface (HEKA). Data were analyzed using Igor Pro 6.37 software (WaveMetrics).

2.11 MTS blocking experiments

The methanethiosulfonate (MTS) reagents [2-(trimethy-lammonium)ethyl] methanethiosulfonate (MTSET), N,N-Dibutyl-N-[2-(2-methyl-2,2-dioxidodithio)ethyl]-1-butanaminium Bromide; (MTS-TBAE) and sodium(2-sulfonatoethyl) methanethiosulfonate (MTSES) were purchased from Toronto Research Chemicals (North York, Canada). A 10 mM stock solution of MTS reagents dissolved in water was prepared, and aliquots of the stock solution were kept on ice until used. Before perfusing MTS reagents, the previously mentioned pulse protocol was applied to measure the peak current at +100 mV. Then, cells were perfused with 2.5 mM MTS reagent for 5-10 min while they are being held at 0 mV. This time was needed to allow the reagent to react with the SH groups of cysteine residues. 3 cells were measured for the control and each reagent.

2.12 Yeast growth assay

TMHC6 gene optimized for yeast codon usage was cloned into the HindIII-XhoI sites of pYES2_MET25 vector under the transcription control of MET-25 promoter[46]. The empty vector with multiple cloning sites (MCS) were used as a control plasmid in the yeast growth assay. The yeast strain SGY1528[47](Mat **a** *ade* 2–1 *can* 1–100 *his* 3–11,15 *leu* 2–3,112 *trp* 1–1 ura 3–1 *trk* 1::HIS3 *trk* 2::TRP1) was transformed with TMHC6 and MCS plasmids by lithium acetate and plated on Uracil-depleted agar plate supplemented with 100mM KCl. Transformed yeast colonies were inoculated in Uracil-depleted medium supplemented with 100mM KCl (C-Ura, 100K) for overnight growth. Estimated 1X10^7 cells were spun down at 6000rpm for 3 min and resuspended in 1mL Uracil- and Methionine-depleted medium with 0 mM KCl (C-Ura-Met, 0K) for 10 minutes at room temperature. Cells were collected by spinning down and resuspended in 10 mL C-Ura-Met 0K medium without or with additional NaCl to reach 1X10^6 cell per mL. 200 uL resuspended cells were plated into each well in a clear-bottom 96-well plate (Greiner, Ref No. 655892). K^+ concentration in each well was adjusted by adding 1-10uL KCl from 2M KCl stock concentration. The 96-well culture plate was incubated at 30 °C and cell densities were monitored in a microplate reader (Tecan infinite M1000) with orbital shaking and 600nm absorption measurement every 30 minutes for continuous 40 hours.

Yeast nitrogen base without amino acids and phosphate (Formedium LTD; Ref No. CYN0801; Batch No. FM0318/8269) supplemented with 7 mM NaH2PO4 was used to eliminate the basal potassium concentration; yeast drop-out mix C-Ura (Formedium LTD, Ref No.DSCK102; Batch No. FM0618/8547) or C-Ura-Met (Formedium LTC, Ref No. DSCK192; Batch No. FM/0718/8601) were added according to the manufacturer's protocol. Ten-fold concentrated medium without KCl or NaCl was made from the powder and sterilized by filtering, from which the liquid medium with varied concentration of KCl and NaCl was made. The growth medium also contains 2% glucose, 100mM Tris (pH 7.5) and 1% penicillin-streptomycin (Gibco, Ref No. 15240-122).

2.13 *In vitro* protein synthesis and liposome permeability assays

DNA templates used for *in vitro* protein synthesis inside liposomes were amplified by PCR and purified using Qiaquick (Qiagen). Liposomes containing a reconstituted *in vitro* protein synthesis system (PURE*frex*1.0, GeneFrontier) was prepared using the water-in-oil (W/O) emulsion/transfer method[48] as previously described[49]. Briefly, 20 μL of an *in vitro* protein synthesis system was supplemented with template 0.29 nM and 1 nM DNA for TMHC6 and TMH4C4, respectively, 400 mM sucrose, 250 mM potassium glutamate, 0.8 U/μL RNase inhibitor, 5 μM streptavidin and 1.5 μM ovalbumin Alexa Fluor 647 conjugate (OA647, Thermo Fisher Scientific). (Note: Based on results shown on Extended Data Figure 6b-c, we controlled for differences in protein expression and membrane incorporation of the two proteins by adjusting the template DNA concentration such that the amount of protein present in the membrane was the same.) To this solution, 200 μL of liquid paraffin (Wako, Osaka, Japan) containing 9 mg of egg PC (L-α-phosphatidylcholine from egg yolk, Sigma, P3556) was added. The mixtures were vortexed for 30 s to form W/O emulsions that were then equilibrated on ice for 10 min. The solution was gently placed on top of 200 μL of the outer solution (see below for the composition) and centrifuged at 18,000 *g* for 30 min at 4°C. The pelleted liposomes were collected through an opening at the bottom of the tube. Proteins were synthesized by incubating the liposomes at 37°C. The outer solution contained the low-molecular-weight components of an *in vitro* protein synthesis system (0.5 mM concentrations of each amino acid, 3 mM ATP, 3 mM GTP, 1 mM CTP and UTP, 2 mM spermidine, 20 mM creatine phosphate, 2 mM dithiothreitol

(DTT), 0.01 μg/μL 10-formyl-5,6,7,8-tetrahydrofolic acid, 250 mM potassium glutamate, 18 mM $Mg(OAc)_2$, 400 mM glucose and 20 mM HEPES (pH 7.6)). For the influx assay, 300 nM AlexaFluor488-biocytin (Thermo Fisher Scientific) or 300 nM 5'-AlexaFluor488-poly-A (11 deoxyadenosine)-biotin-3' (GeneDesign,Inc, Japan) was added and incubated at 25ºC for the indicated time. The liposomes were analyzed using a FACSVerse (BD, Franklin Lakes, NJ). Before FCM analysis, the liposome suspension was diluted forty-fold in dilution buffer (20 mM HEPES-KOH (pH 7.6), 250 mM potassium glutamate, 18 mM $Mg(OAc)_2$, and 400 mM glucose). 30,000 liposomes were measured. Among 30,000, analysis was conducted with the population defined as the giant unilamellar vesicles in previous reports[49,50] (45 to 70% of the total population). The two-dimensional plots in Extended Data Figure 6e show the relationship between OA647 fluorescence intensity, representing the aqueous volume of the liposome, and Alexa Fluor 488 fluorescence intensity. Liposomes with OA647 fluorescence intensity larger than 10^2 (a.u.) were used for analysis. This assay was repeated 3 times with a total of 7 independent measurements. When analyzing the pore formation with α-hemolysin, the experimental condition was identical to that with TMHC6 and TMH4C4, except for the lipid composition (egg phosphatidylcholine: cholesterol = 5:5). This lipid composition was found to be suited for the analysis of the pore formation with α-hemolysin previously[28–30]. 3 independent measurements was performed with α-hemolysin.

In vitro synthesized proteins were analyzed by the SEC with *E. coli* purified proteins as the control. In all of the chromatograms, the fluorescence of RED-tris-NTA (NanoTemper Technologies), a reagent to label the his-tag with red fluorescence, was measured. Purified and *In vitro* synthesized proteins were labeled by adding 90 nM RED-tris-NTA and incubating at 25 ºC for 30 min. SEC was performed using Superdex 200 Increase 5/150 GL (GE Healthcare) on an HPLC instrument (Prominence, Shimadzu) at 30ºC and 0.12 ml/ min. The mobile phase buffer was 0.2% DM, 25 mM Tris-HCl (pH 8.0), 150 mM NaCl. Results were obtained with excitation and emission wavelength of 650 and 670 nm, respectively. To investigate the amount of protein synthesized *in vitro*, the reaction was done with 10 nM DNA and incubated for 3 h at 37ºC in the presence of [^{35}S]-methionine. The synthesized products were analyzed by SDS-PAGE and autoradiography as described previously[51].

2.14 Large unilamellar vesicle preparation and protein-membrane binding assay

Three hundred microliters of 50 mg/mL egg phosphatidylcholine (COATSOME NC-50 (EPC)) (Yuka-Sangyo Co., Ltd., Japan) dissolved in chloroform was rotated under vacuum in a round-bottom flask for 1 h. The lipid film was hydrated with buffer A (10 mM HEPES, pH 7.6, 50 mM potassium glutamate) to obtain 300 µL of 50 mg/mL lipid solution. The lipid solution was sonicated for 10 min and vortexed for 10 s. The lipid solution was further subjected to five rounds of freeze-thaw cycles. The liposome suspension was then extruded with a mini-extruder (Avanti Polar Lipids) using a 0.2 µm VCTP isopore membrane filter at room temperature, resulting in a large unilamellar vesicle (LUV) with 60.2±4.59 nm (s.e.m., *n*=3) in diameter measured by dynamic light scattering (Malvern Panalytical). The LUV solution (final concentration 15 mg/mL) was added to an *in vitro* protein synthesis system to quantify the amount of proteins bound to the lipid membrane. Plasmids encoding EmrE, *E. coli* multidrug transporter, and beta-glucoronidase (GusA) were prepared previously[52,53]. Each protein was synthesized in the presence of [^{35}S]-methionine with or without large unilamellar vesicles (LUV, 60 nm in diameter). 60 nm LUV remains in solution even after centrifugation at 20,000 *g* for 20 min at 4ºC. The amount of protein in the supernatant was analyzed by SDS-PAGE and autoradiography.

2.15 Cryo-EM sample preparation, data collection, and image processing

Cryo-EM grids were prepared using a Vitrobot Mark IV (FEI) with an environmental chamber set at 4 °C. The protein of the peak fractions from SEC was concentrated to ~6 mg/ml and aliquots (3.3 µl) were placed on glow-discharged holey carbon grids (Quantifoil Au R1.2/1.3). Grids were blotted with filter paper (595 Filter paper from Schleicher & Schuell, diameter 55/20 mm) for 3 – 3.5 s and flash-frozen in a mixture of liquid ethane and propane cooled by liquid nitrogen. The cryo grids were then transferred to a Titan Krios operating at 300 kV equipped with Gatan K3 Summit detector and Gatan Imaging Filter (GIF) Quantum energy filter, with a slit width of 20 eV to remove inelastically scattered electrons. Movie stacks were automatically collected using AutoEMation[54], in super-resolution mode with a defocus range of -1.2 µm to -2.2 µm at a nominal magnification of 81,000x. The total exposure rate was approximately 50 e^{-}/Å^{2} for each stack, which was exposed for 2.56 s with an exposure time of 0.08 s per frame, resulting in a total of 32 frames per stack. The stacks were motion corrected with MotionCor2[55] and binned at 2-fold, resulting in a pixel size of 1.087 Å/pixel. Meanwhile, dose weighting[56] was performed and the defocus values were estimated with Gctf[57].

2.16 Cryo-EM data processing

Particles were automatically picked using Relion 3.0.6[58]. After 2D classification with Relion, particles with clear protein features were selected, then subjected to 3D classification with C1 symmetry with Relion. To avoid model bias, a design model of TMHC8 monomer docked in C7 symmetry, which is low-pass filtered to 60 angstroms, was used as the initial reference model for 3D classification. The particles in a dominant specimen containing ~40% of all the 3D classified particles emerged during the first round of 3D classification, with the most continuous and intact map among all classes and were selected and subject to local defocus correction[57], classification, 3D auto-refinement and post-processing. To improve the map quality, an adapted mask was applied to the TM region. Relion post-processing was also used for local resolution estimation (Extended Data Table 2). The resolution was estimated with the gold-standard Fourier shell correlation 0.143 criterion[59]. Refer to Extended Data Figure 7 and Table 2 for details of data collection and processing.

2.17 Cryo-EM structure refinement

The tetrameric design model was docked into the density using Chimera's Fit in Map tool[60]. Because the directionality of the helices and the connections between helices was not clear in the density map, the design model was docked four different ways, encompassing all possible positions when taking into account the symmetry of the design. Each docked model was refined[61] asymmetrically into the density map and 50 models were produced for each dock. The Rosetta force field was augmented with a "fit to density"[62] score term and the lowest energy model from all docks was chosen.

3. References

35. Chen, Z. *et al.* Programmable design of orthogonal protein heterodimers. *Nature* **565**, 106–111 (2019).

36. Krogh, A., Larsson, B., von Heijne, G. & Sonnhammer, E. L. Predicting transmembrane

protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol.* **305**, 567–580 (2001).

37. Studier, F. W. Protein production by auto-induction in high density shaking cultures. *Protein Expr. Purif.* **41**, 207–234 (2005).
38. Cole, J. L., Lary, J. W., P Moody, T. & Laue, T. M. Analytical ultracentrifugation: sedimentation velocity and sedimentation equilibrium. *Methods Cell Biol.* **84**, 143–179 (2008).
39. Dyer, K. N. *et al.* High-throughput SAXS for the characterization of biomolecules in solution: a practical approach. *Methods Mol. Biol.* **1091**, 245–258 (2014).
40. Franke, D. *et al.* ATSAS 2.8: a comprehensive data analysis suite for small-angle scattering from macromolecular solutions. *J. Appl. Crystallogr.* **50**, 1212–1225 (2017).
41. Svergun, D., Barberato, C. & Koch, M. H. J. CRYSOL--a program to evaluate X-ray solution scattering of biological macromolecules from atomic coordinates. *J. Appl. Crystallogr.* **28**, 768–773 (1995).
42. Adams, P. D. *et al.* PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 213–221 (2010).
43. McCoy, A. J. *et al.* Phaser crystallographic software. *J. Appl. Crystallogr.* **40**, 658–674 (2007).
44. Terwilliger, T. C. *et al.* phenix.mr_rosetta: molecular replacement and model rebuilding with Phenix and Rosetta. *J. Struct. Funct. Genomics* **13**, 81–90 (2012).
45. Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. Features and development of Coot. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 486–501 (2010).
46. Minor, D. L., Masseling, S. J., Jan, Y. N. & Jan, L. Y. Transmembrane Structure of an Inwardly Rectifying Potassium Channel. *Cell* vol. 96 879–891 (1999).

47. Tang, W. *et al.* Functional expression of a vertebrate inwardly rectifying K+ channel in yeast. *Mol. Biol. Cell* **6**, 1231–1240 (1995).

48. Pautot, S., Frisken, B. J. & Weitz, D. A. Production of Unilamellar Vesicles Using an Inverted Emulsion. *Langmuir* **19**, 2870–2879 (2003).

49. Nishimura, K. *et al.* Cell-free protein synthesis inside giant unilamellar vesicles analyzed by flow cytometry. *Langmuir* **28**, 8426–8432 (2012).

50. Nishimura, K. *et al.* Population analysis of structural properties of giant liposomes by flow cytometry. *Langmuir* **25**, 10439–10443 (2009).

51. Kazuta, Y., Matsuura, T., Ichihashi, N. & Yomo, T. Synthesis of milligram quantities of proteins using a reconstituted in vitro protein synthesis system. *J. Biosci. Bioeng.* **118**, 554–557 (2014).

52. Soga, H. *et al.* In vitro membrane protein synthesis inside cell-sized vesicles reveals the dependence of membrane protein integration on vesicle volume. *ACS Synth. Biol.* **3**, 372–379 (2014).

53. Matsuura, T., Hosoda, K., Ichihashi, N., Kazuta, Y. & Yomo, T. Kinetic analysis of β-galactosidase and β-glucuronidase tetramerization coupled with protein translation. *J. Biol. Chem.* **286**, 22028–22034 (2011).

54. Lei, J. & Frank, J. Automated acquisition of cryo-electron micrographs for single particle reconstruction on an FEI Tecnai electron microscope. *J. Struct. Biol.* **150**, 69–80 (2005).

55. Zheng, S. Q. *et al.* MotionCor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy. *Nat. Methods* **14**, 331–332 (2017).

56. Grant, T. & Grigorieff, N. Measuring the optimal exposure for single particle cryo-EM using a 2.6 Å reconstruction of rotavirus VP6. *Elife* **4**, e06980 (2015).

57. Zhang, K. Gctf: Real-time CTF determination and correction. *J. Struct. Biol.* **193**, 1–12

(2016).

58. Zivanov, J. *et al.* New tools for automated high-resolution cryo-EM structure determination in RELION-3. *Elife* **7**, (2018).

59. Rosenthal, P. B. & Henderson, R. Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. *J. Mol. Biol.* **333**, 721–745 (2003).

60. Goddard, T. D., Huang, C. C. & Ferrin, T. E. Visualizing density maps with UCSF Chimera. *J. Struct. Biol.* **157**, 281–287 (2007).

61. Conway, P., Tyka, M. D., DiMaio, F., Konerding, D. E. & Baker, D. Relaxation of backbone bond geometry improves protein energy landscape modeling. *Protein Sci.* **23**, 47–55 (2014).

62. DiMaio, F., Tyka, M. D., Baker, M. L., Chiu, W. & Baker, D. Refinement of protein structures into low-resolution density maps using rosetta. *J. Mol. Biol.* **392**, 181–190 (2009).

4. Appendices

Appendix A. Example RosettaScripts XML protocol for design of water-soluble pores. This script is provided solely as a guideline since it has not been optimised for compatibility with the most recent versions of RosettaScripts.

```
<ROSETTASCRIPTS>

  <SCOREFXNS>
    <ScoreFunction name="soft_symm" weights="beta_soft_rep.wts" symmetric="1">
      <Reweight scoretype="aa_composition" weight="1.0" />
    </ScoreFunction>
    <ScoreFunction name="hard_symm" weights="beta_cst" symmetric="1">
      <Reweight scoretype="coordinate_constraint" weight="0.5" />
      <Reweight scoretype="aa_composition" weight="1.0" />
    </ScoreFunction>
    <ScoreFunction name="hard_cart" weights="beta_cart" symmetric="1" >
      <Reweight scoretype="cart_bonded" weight="0.5" />
      <Reweight scoretype="coordinate_constraint" weight="1" />
    </ScoreFunction>
    <ScoreFunction name="hard_ele" weights="beta_cart" symmetric="1">
      <Reweight scoretype="fa_elec" weight="1.4" />
      <Reweight scoretype="hbond_sc" weight="2.0" />
    </ScoreFunction>
    <ScoreFunction name="output" weights="beta" symmetric="1"/>
  </SCOREFXNS>

  <RESIDUE_SELECTORS>
    <Chain name="select_chainA" chains="A" />
    <Not name="not_chainA" selector="select_chainA" />
    <ResiduePDBInfoHasLabel name="hbnet_residues" property="HBNet" />
    <Neighborhood name="around_hbnet" selector="hbnet_residues" distance="5.0" />
    <Layer name="hbnet_core" select_core="true" core_cutoff="3.6" />
    <And name="core_around_hbnet" selectors="hbnet_core,around_hbnet"/>
  </RESIDUE_SELECTORS>
  <TASKOPERATIONS>
    <ReadResfile name="resfile" filename="res.resfile"/>
    <InitializeFromCommandline name="init"/>
    <IncludeCurrent name="current"/>
    <LimitAromaChi2 name="arochi" />
    <ExtraRotamersGeneric name="ex1_ex2" ex1="1" ex2="1"/>
    <ExtraRotamersGeneric name="ex1" ex1="1"/>
    <LayerDesign name="init_layers" layer="core_boundary_surface_Nterm_Cterm" ignore_pikaa_natro="true" make_pymol_script="0" use_sidechain_neighbors="True"
core="3.6" >
      <core>
      <Helix append="MSDNTQHYW"/>
      <Helix exclude="C"/>
      </core>
      <boundary>
      <Helix exclude="C"/>
      </boundary>
      <surface>
      <Helix exclude="C"/>
      </surface>
    </LayerDesign>
    <SelectBySASA name="select_core" state="bound" mode="mc" core="1" probe_radius="2.0" core_asa="35" surface_asa="45" verbose="1"/>
    <SelectBySASA name="select_boundary" state="bound" mode="mc" boundary="1" probe_radius="2.0" core_asa="35" surface_asa="45" verbose="1"/>
    <SelectBySASA name="select_surface" state="bound" mode="mc" surface="1" probe_radius="2.0" core_asa="35" surface_asa="45" verbose="1"/>
    <RestrictAbsentCanonicalAAS name="ala_only" resnum="0" keep_aas="A" />
    <RestrictToRepacking name="repack_only" />
  </TASKOPERATIONS>

  <FILTERS>
  </FILTERS>

  <MOVERS>
    <SetupForSymmetry name="setup_symm" definition="C8_Z.sym" />
    <ConstraintSetMover name="add_cst_file" add_constraints="1" />
    <SymPackRotamersMover name="transform_sc" scorefxn="hard_symm" task_operations="ala_only" />
    <AddConstraintsToCurrentConformationMover name="add_cst" use_distance_cst="0" coord_dev="1" />
    <ClearConstraintsMover name="clearconstraints"/>
    <PDBReload name="reload" />
    <SymMinMover name="hardmin_bb" scorefxn="hard_cart" type="lbfgs_armijo_nonmonotone" tolerance="0.0001"  chi="0" bb="1" bondangle="1" bondlength="1" jump="all"
cartesian="1"/>
    <SymMinMover name="hardmin_sconly" scorefxn="hard_symm" chi="1" bb="0" bondangle="0" bondlength="0" />

    <DumpPdb name="dumppdb" fname="dump.pdb" />
    <SymMinMover name="hardmin_cart" scorefxn="hard_cart" type="lbfgs_armijo_nonmonotone" tolerance="0.0001" max_iter="2000" chi="1" bb="1" bondangle="1"
bondlength="1" jump="ALL" cartesian="1"/>

    <HBNetStapleInterface monte_carlo_branch="true" total_num_mc_runs="10000" find_only_native_networks="0" scorefxn="hard_symm" name="hbnet_interf"
start_selector="hbnet_core" hb_threshold="-0.5" design_residues="SDNTQHYW" min_helices_contacted_by_network="2" min_intermolecular_hbonds="1"
```

```
core_selector="hbnet_core" show_task="0" store_subnetworks="0" verbose="0" combos="2" no_heavy_unsats_allowed="1" write_network_pdbs="0" min_network_size="3"
max_network_size="8" max_unsat_Hpol="0" min_core_res="3" write_cst_files="0" min_networks_per_pose="3" max_networks_per_pose="4"
use_aa_dependent_weights="true" max_replicates_before_branch="0" allow_onebody_networks="true" onebody_hb_threshold="-0.2" min_percent_hbond_capacity="0.5"
task_operations="init,arochi,init_layers"/>
  <MultiplePoseMover name="MPM_design" max_input_poses="10">
    <ROSETTASCRIPTS>

      <SCOREFXNS>
        <ScoreFunction name="soft_symm" weights="beta_soft_rep.wts" symmetric="1">
        <Reweight scoretype="aa_composition" weight="1.0" />
        </ScoreFunction>
        <ScoreFunction name="hard_symm" weights="beta_cst" symmetric="1">
        <Reweight scoretype="coordinate_constraint" weight="0.5" />
        <Reweight scoretype="aa_composition" weight="1.0" />
        </ScoreFunction>
        <ScoreFunction name="hard_cart" weights="beta_cart" symmetric="1" >
        <Reweight scoretype="cart_bonded" weight="0.5" />
        <Reweight scoretype="coordinate_constraint" weight="1" />
        </ScoreFunction>
        <ScoreFunction name="hard_ele" weights="beta_cst" symmetric="1">
        <Reweight scoretype="fa_elec" weight="1.4" />
        <Reweight scoretype="hbond_sc" weight="2.0" />
        </ScoreFunction>
        <ScoreFunction name="output" weights="beta" symmetric="1"/>
      </SCOREFXNS>

      <RESIDUE_SELECTORS>
        <Chain name="select_chainA" chains="A" />
        <Not name="not_chainA" selector="select_chainA" />
        <ResiduePDBInfoHasLabel name="hbnet_residues" property="HBNet" />
        <Neighborhood name="around_hbnet" selector="hbnet_residues" distance="5.0" />
        <Not name="not_around_hbnet" selector="around_hbnet" />
        <Layer name="hbnet_core" select_core="true" core_cutoff="3.6" />
        <And name="core_around_hbnet" selectors="hbnet_core,around_hbnet"/>
        <Layer name="pick_core_and_boundary" select_core="true" select_boundary="true" core_cutoff="3.6"/>
        <Layer name="pick_core_and_boundary_SASA" use_sidechain_neighbors="false" select_core="true" select_boundary="true" core_cutoff="35" surface_cutoff="45" />
        <Layer name="pick_core_and_surface" select_core="true" select_surface="true" core_cutoff="3.6"/>
        <Layer name="pick_core_and_surface_SASA" use_sidechain_neighbors="false" select_core="true" select_surface="true" core_cutoff="35" surface_cutoff="45"/>
        <Layer name="pick_surface_and_boundary" select_surface="true" select_boundary="true" core_cutoff="3.6"/>
        <Layer name="pick_surface_and_boundary_SASA" use_sidechain_neighbors="false" select_surface="true" select_boundary="true" core_cutoff="35" surface_cutoff="45"
/>
      </RESIDUE_SELECTORS>

      <TASKOPERATIONS>
        <InitializeFromCommandline name="init"/>
        <IncludeCurrent name="current"/>
        <LimitAromaChi2 name="arochi" />
        <ExtraRotamersGeneric name="ex1_ex2" ex1="1" ex2="1" ex1_sample_level="2" ex2_sample_level="2" />
        <ExtraRotamersGeneric name="ex1" ex1="1" ex1_sample_level="2"/>
        <ReadResfile name="resfile" filename="res.resfile" />
        <LayerDesign name="init_layers" layer="core_boundary_surface_Nterm_Cterm" ignore_pikaa_natro="true" make_pymol_script="0" use_sidechain_neighbors="True"
core="3.6" >
          <core>
          <Helix append="M"/>
          <Helix exclude="WY"/>
          </core>
          <boundary>
          <Helix exclude="WCDRHQ"/>
          </boundary>
          <surface>
          <Helix exclude="WCDHNQSTR"/>
          </surface>
        </LayerDesign>
        <LayerDesign name="sasa_layers" layer="core_boundary_surface_Nterm_Cterm" ignore_pikaa_natro="true" make_pymol_script="0" use_sidechain_neighbors="0"
core="35" surface="45" >
          <core>
          <Helix exclude="WY"/>
          <Helix append="M"/>
          </core>
          <boundary>
          <Helix exclude="WCDRH"/>
          </boundary>
          <surface>
          <Helix exclude="WCDST"/>
          </surface>
        </LayerDesign>
        <SelectBySASA name="select_core" state="bound" mode="mc" core="1" probe_radius="2.0" core_asa="35" surface_asa="45" verbose="1"/>
        <SelectBySASA name="select_boundary" state="bound" mode="mc" boundary="1" probe_radius="2.0" core_asa="35" surface_asa="45" verbose="1"/>
        <SelectBySASA name="select_surface" state="bound" mode="mc" surface="1" probe_radius="2.0" core_asa="35" surface_asa="45" verbose="1"/>

        <RestrictToRepacking name="repack_only" />
        <ConsensusLoopDesign name="disallow_non_abego_aas"/>

        <OperateOnResidueSubset name="hbnet_task" selector="hbnet_residues">
        <RestrictToRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="hbnet_task_init" selector="hbnet_residues">
        <PreventRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="repack_only_if_not_around_hbnet" selector="not_around_hbnet">
          <RestrictToRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="design_core" selector="pick_surface_and_boundary">
          <PreventRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="design_boundary" selector="pick_core_and_surface">
          <PreventRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="design_surface" selector="pick_core_and_boundary">
          <PreventRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="design_core_SASA" selector="pick_surface_and_boundary_SASA">
          <PreventRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="design_boundary_SASA" selector="pick_core_and_surface_SASA">
          <PreventRepackingRLT/>
        </OperateOnResidueSubset>
        <OperateOnResidueSubset name="design_surface_SASA" selector="pick_core_and_boundary_SASA">
          <PreventRepackingRLT/>
        </OperateOnResidueSubset>
      </TASKOPERATIONS>

      <FILTERS>
        <ShapeComplementarity name="sc" min_sc="0.0" verbose="1" write_int_area="1" residue_selector1="select_chainA" residue_selector2="not_chainA" />
        <Ddg name="ddg" scorefxn="output" threshold="-10" jump="1" repeats="1" repack="true" repack_bound="true" relax_bound="false"/>
        <Sasa name="sasa" threshold="800" upper_threshold="1000000000" hydrophobic="0" polar="0" jump="1"/>
        <ScoreType name="cst_score" scorefxn="hard_symm" score_type="atom_pair_constraint" threshold="50"/>
        <Holes name="network_holes" threshold="-0.5" residue_selector="core_around_hbnet" normalize_per_atom="true" exclude_bb_atoms="true" confidence="1"/>
        <BuriedUnsatHbonds name="new_buns_sc_heavy" scorefxn="hard_symm" cutoff="1" report_sc_heavy_atom_unsats="true" print_out_info_to_pdb="true"
ignore_surface_res="true" residue_surface_cutoff="20.0" ignore_bb_heavy_unsats="false" confidence="0"/>
        <BuriedUnsatHbonds name="new_buns_HBNet" scorefxn="hard_symm" use_hbnet_behavior="true" cutoff="20" print_out_info_to_pdb="true" ignore_surface_res="true"
residue_surface_cutoff="20.0" ignore_bb_heavy_unsats="true" residue_selector="hbnet_residues" confidence="1"/>
        <SSShapeComplementarity name="ss_sc" verbose="1" loops="0" helices="1" residue_selector="select_chainA" confidence="1" min_sc="0.65"/> #use cutoff 0.7
        <SSPrediction name="ss_pred" use_svm="0" use_probability="1" cmd="~/bin/psipred/runpsipred_single" />
        <SSPrediction name="ss_pred_percentage" threshold="0.90" use_probability="0" use_svm="0" cmd="~/bin/psipred/runpsipred_single" />
        <ScoreType name="total_score" scorefxn="hard_symm" score_type="total_score" threshold="0" confidence="0" />
        <ResidueCount name="ala_count" max_residue_count="30" residue_types="ALA" confidence="0"/>
        <PreProline name="prepro" use_statistical_potential="0" />
      </FILTERS>

      <MOVERS>
        <AddCompositionConstraintMover name="addcomp" filename="aa.comp" />
        <SymPackRotamersMover name="softpack_all" scorefxn="soft_symm" task_operations="init,init_layers,current,arochi,hbnet_task_init,disallow_non_abego_aas"/>
        <SymPackRotamersMover name="softpack_core" scorefxn="soft_symm"
task_operations="init,resfile,init_layers,current,design_core,arochi,hbnet_task_init,disallow_non_abego_aas" />
        <SymPackRotamersMover name="softpack_boundary" scorefxn="soft_symm"
task_operations="init,resfile,init_layers,current,design_boundary,arochi,hbnet_task_init,disallow_non_abego_aas" />
        <SymPackRotamersMover name="softpack_surface" scorefxn="soft_symm"
task_operations="init,resfile,init_layers,current,design_surface,arochi,hbnet_task_init,disallow_non_abego_aas" />
        <SymPackRotamersMover name="hardpack_core" scorefxn="hard_symm"
task_operations="init,resfile,sasa_layers,current,design_core,arochi,ex1_ex2,hbnet_task,disallow_non_abego_aas" />
        <SymPackRotamersMover name="hardpack_boundary"  scorefxn="hard_symm"
task_operations="init,resfile,sasa_layers,current,design_boundary,arochi,ex1,hbnet_task,disallow_non_abego_aas" />
        <SymPackRotamersMover name="hardpack_surface" scorefxn="hard_ele"
task_operations="init,resfile,sasa_layers,current,design_surface,arochi,hbnet_task,disallow_non_abego_aas" />
        <SymPackRotamersMover name="hardpack_all"  scorefxn="hard_symm" task_operations="init,resfile,sasa_layers,current,arochi,hbnet_task,disallow_non_abego_aas"
/>
        <SymMinMover name="hardmin_sconly" scorefxn="hard_symm" type="lbfgs_armijo_nonmonotone" chi="1" bb="0" bondangle="0" bondlength="0" />
        <SymMinMover name="hardmin_cart" scorefxn="hard_cart" type="lbfgs_armijo_nonmonotone" tolerance="0.0001" max_iter="2000" chi="1" bb="1" bondangle="1"
bondlength="1" jump="ALL" cartesian="1"/>
        <SymPackRotamersMover name="repack" scorefxn="hard_symm" task_operations="init,repack_only,ex1_ex2" />
        <InterfaceAnalyzerMover name="interface_analyzer" scorefxn="hard_symm" packstat="1" pack_input="0" pack_separated="1" interface_sc="1"/>
        <ParsedProtocol name="pack_min">
          <Add mover_name="hardmin_cart"/>
          <Add mover_name="hardpack_core"/>
          <Add mover_name="hardpack_boundary"/>
          <Add mover_name="hardpack_surface"/>
        </ParsedProtocol>

        <GenericSimulatedAnnealer name="GSA_pack_min"
        mover_name="pack_min" trials="4"
        periodic_mover="hardpack_all" eval_period="10" history="10"
        bolz_rank="1" recover_low="1" preapply="0" drift="1"
        filter_name="ss_sc" temperature="10" sample_type="high" >
          <Filters>
            <AND filter_name="ss_pred" temperature="10" sample_type="low"/>
            <AND filter_name="network_holes" temperature="1.5" sample_type="low"/>
          </Filters>
        </GenericSimulatedAnnealer>


      </MOVERS>
```

```
        <PROTOCOLS>
          <Add mover="addcomp"/>
          <Add mover="softpack_all"/>
          <Add mover="softpack_core"/>
          <Add mover="softpack_boundary"/>
          <Add mover="softpack_surface"/>
          <Add mover="hardmin_sconly"/>
          <Add mover="hardpack_core"/>
          <Add mover="hardpack_boundary"/>
          <Add mover="hardpack_surface"/>
          <Add mover="GSA_pack_min"/>
          <Add mover="hardmin_cart" />
          <Add mover="repack" />
          <Add filter="cst_score"/>
          <Add filter="new_buns_HBNet"/>
          <Add filter="new_buns_sc_heavy"/>
          <Add filter="ss_sc"/>
          <Add filter="ss_pred_percentage"/>
          <Add filter="network_holes"/>
          <Add filter="sc"/>
          <Add filter="ddg"/>
          <Add filter="sasa"/>
          <Add filter="prepro"/>
          <Add filter="ala_count"/>
        </PROTOCOLS>
      </ROSETTASCRIPTS>
    </MultiplePoseMover>
  </MOVERS>

  <PROTOCOLS>
    <Add mover="setup_symm"/>
    <Add mover="transform_sc"/>
    <Add mover="add_cst"/>
    <Add mover="hardmin_bb"/>
    <Add mover="clearconstraints"/>
    <Add mover="hbnet_interf" />
    <Add mover_name="MPM_design"/>
  </PROTOCOLS>

  <OUTPUT scorefxn="output" />

</ROSETTASCRIPTS>
```

Appendix B. Example python protocol for design of transmembrane pores. The output files of this script can be used by RosettaScripts XML protocol described in Appendix C.

```
# coding: utf-8


# In[2]:



# This python script take a helix bundle pose, and out put a pdb with defined TM region with PDB label info and a .span file
# This python script is written by Hua Bai and  Peilong Lu.
# This python is python2
# huabai@uw.edu    lpl15@uw.edu

import math
import pyrosetta
import argparse
# from datetime import datetime
from glob import glob

# In[3]:

## Step1: directly check per residue sasa score
def based_on_sasa(input_pose, ball_size=1, sasa_threshold = 10):
    ## define two vector placeholders for the scores. One for atoms, one for residues
    atom_sasa = pyrosetta.rosetta.core.id.AtomID_Map_double_t()
    rsd_sasa = pyrosetta.rosetta.utility.vector1_double()

    pyrosetta.rosetta.core.scoring.calc_per_atom_sasa(input_pose,atom_sasa,rsd_sasa,ball_size)
    selected_by_sasa = []
    for idx, resi_sasa in enumerate(rsd_sasa):
        if resi_sasa > sasa_threshold:
            selected_by_sasa.append(idx+1)
    return selected_by_sasa


# In[4]:

## Step2: select polar residues
def based_on_polarity(input_pose):
    selected_by_polar = []
    for i in range(input_pose.size()):
        idx = i+1
        if input_pose.residue_type(idx).is_polar():
            selected_by_polar.append(idx)
    return selected_by_polar


# In[5]:

## Step2.1: select chain A
def chain_A(input_pose):
    selected_by_chain = []
    chaina = input_pose.split_by_chain()[1]
    for i in range(chaina.size()):
        idx = i+1
        selected_by_chain.append(idx)
    return selected_by_chain


# In[6]:

## Step3-Method1: select outward pointing residues by angles. NOT USE.
## output unit: degree
## the angle between CA-CB ray and origin-CA ray
## for example, if the angle larger than 90 degree, smaller than 270 degree, the CA-CB may point to inward
def calc_angle(pose, resi_idx):
    ## special case: GLY
    if pose.residue(resi_idx).name() == "GLY":
        return 999

    ## for CA
    ca_x = pose.residue(resi_idx).atom("CA").xyz()[0]
    ca_y = pose.residue(resi_idx).atom("CA").xyz()[1]
    ca_angle = math.atan2(ca_y,ca_x)

    ## for CB
    cb_x = pose.residue(resi_idx).atom("CB").xyz()[0]
    cb_y = pose.residue(resi_idx).atom("CB").xyz()[1]
    cb_a_x = cb_x - ca_x
    cb_a_y = cb_y - ca_y
    cb_a_angle = math.atan2(cb_a_y,cb_a_x)

    ## return the difference
    return ((cb_a_angle - ca_angle) / math.pi * 180) % 360


def based_on_CA_CB_angle(input_pose, angle_threshold = 120):
    selected_by_angle = []
    for resi_idx in range(input_pose.size()):
        check_angle = calc_angle(input_pose, resi_idx+1)
        if check_angle < angle_threshold or check_angle > 360-angle_threshold and check_angle != 999:
            selected_by_angle.append(resi_idx+1)
    return selected_by_angle




# In[7]:

## Step3-Method2: select outward pointing residues by distance differencs
## output unit: distance, A
## the compare the distances between origin-CA and both origin-N ray and origin-C ray
## sum the distance differences
## the more negative the difference, the higher chance it is pointing inward
def compare_distance_to_origin(pose, resi_idx):
    n_x = pose.residue(resi_idx).atom("N").xyz()[0]
    n_y = pose.residue(resi_idx).atom("N").xyz()[1]
    # n_z = pose.residue(resi_idx).atom("N").xyz()[2]
    n_distance = math.sqrt(n_x**2 + n_y**2)

    c_x = pose.residue(resi_idx).atom("C").xyz()[0]
    c_y = pose.residue(resi_idx).atom("C").xyz()[1]
    # c_z = pose.residue(resi_idx).atom("C").xyz()[2]
    c_distance = math.sqrt(c_x**2 + c_y**2)

    ca_x = pose.residue(resi_idx).atom("CA").xyz()[0]
    ca_y = pose.residue(resi_idx).atom("CA").xyz()[1]
    # ca_z = pose.residue(resi_idx).atom("CA").xyz()[2]
    ca_distance = math.sqrt(ca_x**2 + ca_y**2)

    ## CA-N distance, corrected by helix pitch angle, not use
    ca_n_distance_diff = ca_distance - n_distance
    ## ca_n_z_diff = ca_z - n_z
    ## angle = math.atan2(ca_n_z_diff, math.fabs(ca_n_distance_diff))
    ## ca_n_distance_diff_corrected = ca_n_distance_diff / math.cos(angle)

    ## CA-C distance, corrected by helix pitch angle, not use
    ca_c_distance_diff = ca_distance - c_distance
    ## ca_c_z_diff = ca_z - c_z
    ## angle = math.atan2(ca_c_z_diff, math.fabs(ca_c_distance_diff))
    ## ca_c_distance_diff_corrected = ca_c_distance_diff / math.cos(angle)

    return ca_n_distance_diff + ca_c_distance_diff
def based_on_distance(input_pose, dist_diff_threshold = -0.6):
    selected_by_distance = []
    for resi_idx in range(input_pose.size()):
        check_distance = compare_distance_to_origin(input_pose, resi_idx+1)
        if check_distance > dist_diff_threshold:
            selected_by_distance.append(resi_idx+1)
    return selected_by_distance



# In[8]:

## Step4: choose residues inside the membrane
import numpy as np
def survey_z(input_pose, percentile , midpoint):
    all_z = []
    for resi_idx in range(input_pose.size()):
        ca_z = input_pose.residue(resi_idx+1).atom("CA").xyz()[2]
        all_z.append(ca_z)
    N_z=all_z[1]
    C_z=all_z[-1]
    z_array = np.array(all_z)
    selected_by_z = []
    for idx, value in enumerate(z_array):
        if float(value) > float(midpoint-percentile) and  float(value) < float(midpoint+percentile):
                selected_by_z.append(idx+1)
    return selected_by_z


def WY_z(input_pose, percentile , midpoint):
    all_z = []
    for resi_idx in range(input_pose.size()):
        ca_z = input_pose.residue(resi_idx+1).atom("CA").xyz()[2]
        all_z.append(ca_z)
    N_z=all_z[1]
    C_z=all_z[-1]
    z_array = np.array(all_z)
    selected_by_z = []
    for idx, value in enumerate(z_array):
            if float(value) > float(midpoint+percentile-2.5) and  float(value) < float(midpoint+percentile+2.5):
                selected_by_z.append(idx+1)

    return selected_by_z


def RK_z(input_pose, percentile , midpoint):
    all_z = []
    for resi_idx in range(input_pose.size()):
        ca_z = input_pose.residue(resi_idx+1).atom("CA").xyz()[2]
        all_z.append(ca_z)
    N_z=all_z[1]
    C_z=all_z[-1]
    z_array = np.array(all_z)
    selected_by_z = []
    for idx, value in enumerate(z_array):
            if float(value) > float(midpoint-percentile-2.5) and  float(value) < float(midpoint-percentile+2.5):
                selected_by_z.append(idx+1)
    return selected_by_z


def F_z(input_pose, percentile , midpoint):
    all_z = []
    for resi_idx in range(input_pose.size()):
        ca_z = input_pose.residue(resi_idx+1).atom("CA").xyz()[2]
        all_z.append(ca_z)
    N_z=all_z[1]
    C_z=all_z[-1]
    z_array = np.array(all_z)
    selected_by_z = []
    for idx, value in enumerate(z_array):
            if float(value) > float(midpoint-1.2) and  float(value) < float(midpoint+1.2):
                selected_by_z.append(idx+1)
    return selected_by_z

"""import numpy as np
def survey_z(input_pose, percentile = 90):
    all_z = []
    for resi_idx in range(input_pose.size()):
        ca_z = input_pose.residue(resi_idx+1).atom("CA").xyz()[2]
        all_z.append(ca_z)
    z_array = np.array(all_z)

selected_by_z = []
    for idx, value in enumerate(z_array):
        if value > np.percentile(z_array, (100-percentile)/2) and            value < np.percentile(z_array, 100- (100-percentile)/2):
            selected_by_z.append(idx+1)
    return selected_by_z"""



# In[9]:

## Step5: find the interdections of the selected residues
## https://stackoverflow.com/questions/3852780/python-intersection-of-multiple-lists
def intersect(*d):
    sets = iter(map(set, d))
    result = sets.next()
    for s in sets:
        result = result.intersection(s)
    return result


# In[10]:

## Step6: Combine the previous steps, and add remarks to pdb file
from shutil import copyfile
import os
## pdb_input: the location/path of the input pdb
## final_selection: the list of index of the selected residues, 1-based
def add_PDBinfo_remark (pdb_input, final_selection,WY_selection,RK_selection,F_selection):
    path_list = list(os.path.split(pdb_input))
    design_id = path_list[-1].split(".pdb")[0]
    modified_file_name = "{}_with_remarks.pdb".format(design_id)
    path_list[-1] = modified_file_name
    output_path = os.path.join(*path_list)

    try:
        os.remove(output_path)
    except OSError:
        pass

    os.system("cat "+pdb_input+"|~/scripts/filt_chain.pl A > "+output_path)
    with open(output_path,"a") as output_pdb:
        for idx in final_selection:
            pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} OUTWARD_POLAR\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
        for idx in WY_selection:
            pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} WY_RING\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
        for idx in RK_selection:
            pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} RK_RING\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
        for idx in F_selection:
            pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} F_RING\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
"""
## Step6.1: add to run.sh
from shutil import copyfile
import os
## pdb_input: the location/path of the input pdb
## final_selection: the list of index of the selected residues, 1-based
def add_PDBinfo_remark (pdb_input, final_selection,WY_selection,RK_selection):
    path_list = list(path.split(pdb_input))
    design_id = path_list[-1]
    modified_file_name = "{}.sh".format(design_id)
    path_list[-1] = modified_file_name
    output_path = os.path.join(*path_list)

    try:
        os.remove(output_path)
    except OSError:
        pass

    os.system("cat "+pdb_input+"|~/scripts/filt_chain.pl A > "+output_path)


with open(output_path,"a") as output_pdb:
    for idx in final_selection:
        pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} OUTWARD_POLAR\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
        for idx in WY_selection:
            pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} WY_RING\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
    for idx in RK_selection:
        pdbinfo_label_remark = "REMARK PDBinfo-LABEL: {:>4} RK_RING\n".format(idx)
            output_pdb.write(pdbinfo_label_remark)
"""



# In[11]:

## create span file for illustration in pymol
def add_span_file(selection_keyword, selection_list, length):
    with open("{}.span".format(selection_keyword), "w") as script:
        script.write("TM region prediction for BRD4 predicted using OCTOPUS\n")
        TM_split=[]
        for idx in range(len(selection_list)-1):
            if selection_list[idx] != selection_list[idx+1]-1:
                TM_split.append(idx)
        TM_N = len(TM_split)+1
        script.write("{} {}\n".format(TM_N, length))
        script.write("antiparallel\n")
        script.write("n2c\n")
        for i in range(TM_N):
            if i ==0:
                script.write("{} {} {} {}\n".format(selection_list[0],selection_list[TM_split[i]],selection_list[0],selection_list[TM_split[i]] ))
            elif i == TM_N-1:
                script.write("{} {} {} {}\n".format(selection_list[TM_split[-1]+1],selection_list[-1],selection_list[TM_split[-1]+1],selection_list[-1] ))
            else:
                script.write("{} {} {} {}\n".format(selection_list[TM_split[i-1]+1],selection_list[TM_split[i]],selection_list[TM_split[i-1]+1],selection_list[TM_split[i]] ))


# In[12]:

## create pml file for illustration in pymol
def create_pml_file(selection_keyword, selection_list):
    with open("select_surface_{}.pml".format(selection_keyword), "w") as script:
        script.write("select {}, resi ".format(selection_keyword))
        for idx, resi_idx in enumerate(selection_list):
            if idx == 0:
                script.write("{}".format(resi_idx))
            else:
                script.write("+{}".format(resi_idx))
        script.write("\n")
        script.write("util.cbag all\n")
        script.write("util.cbay ({})\n".format(selection_keyword))


# In[13]:

## argparse argument
def get_argparse():
    parser = argparse.ArgumentParser(description='Select the surface polar residues for memebrane channel designs')
    parser.add_argument("-b", '--ball_size', metavar = "\b", type=float, dest='ball_size',
                default=1,
                help='The ball size for calculating sasa, the smaller the ball, the deeper the surface can reach. The default value is 1')
    parser.add_argument("-s", '--sasa_threshold', metavar = "\b", type=float, dest='sasa_threshold',
                default=10,
                help='The sasa threshold for determining whether the residue is surface or core. The higher the sasa, the residue is more like to be on surface. The default value is
10')
    parser.add_argument("-d", '--distance_threshold', metavar = "\b", type=float, dest='distance_threshold',
                default=-0.6,
                help='the sum of the distance differences of (CA-axis - N-axis) and (CA-axis - C-axis), the more negative the sum is, the more likely the residue is pointing inward.
The default value is -0.6')
    parser.add_argument("-z", '--z_percentile', metavar = "\b", type=float, dest='z_percentile',
                default=13.5,
                help='the percentage of the residues are considered to be inside membrane. The default value is 3')
    parser.add_argument("-i", '--input_dir', metavar = "\b", type=str, dest='input_dir',
                default=".",
                help='This is where the input pdbs are located. The default is current dir ".")
    parser.add_argument("-m", '--imidpoint', metavar = "\b", type=float, dest='imidpoint',
                default=0,
                help='the midpoint in the membrane. The default value is 0')


    ## parser.add_argument('--output_dir', type=str, dest='output_dir',
    ##             default=datetime.now().strftime("pdbs_with_remarks_%y%m%d_%p_%I%M%S"),
    ##             help='This is where the pdbs with remarks will be stored')
    return parser


# In[21]:

if __name__ == "__main__":
    ## load arguments


    parser = get_argparse()
    args=parser.parse_args("-m -10 -z 15 -s 8".split())        #This is important for running in jupyter
#    args=parser.parse_args()
    ## init pyrosetta
    pyrosetta.init(options="")

    target_pdbs = glob("{}/*4.pdb".format(args.input_dir))

    for target in target_pdbs:
        path_list = os.path.split(target)
        design_id = path_list[-1].split(".pdb")[0]

        input_pose = pyrosetta.pose_from_file(target)

        length = len(chain_A(input_pose))

        final_selection = list(intersect(based_on_sasa(input_pose, args.ball_size, args.sasa_threshold),                                    based_on_polarity(input_pose),
based_on_distance(input_pose, args.distance_threshold),                                    chain_A(input_pose),                                    survey_z(input_pose, args.z_percentile,
args.imidpoint) ))

        WY_selection = list(intersect(based_on_sasa(input_pose, args.ball_size, args.sasa_threshold),                                    based_on_polarity(input_pose),
based_on_distance(input_pose, args.distance_threshold),                                    chain_A(input_pose),                                    WY_z(input_pose, args.z_percentile,
args.imidpoint) ))
        RK_selection = list(intersect(based_on_sasa(input_pose, args.ball_size, args.sasa_threshold),                                    based_on_polarity(input_pose),
based_on_distance(input_pose, args.distance_threshold),                                    chain_A(input_pose),                                    RK_z(input_pose, args.z_percentile,
args.imidpoint) ))
        F_selection = list(intersect(based_on_sasa(input_pose, args.ball_size, args.sasa_threshold),                                    based_on_polarity(input_pose),
based_on_distance(input_pose, args.distance_threshold),                                    chain_A(input_pose),                                    F_z(input_pose, args.z_percentile,
args.imidpoint) ))
        TM_selection = list(intersect(chain_A(input_pose), survey_z(input_pose, args.z_percentile, args.imidpoint)))

        add_span_file("{}".format(design_id), TM_selection, length)

        add_PDBinfo_remark(target,final_selection,WY_selection,RK_selection,F_selection)


        create_pml_file("{}_final".format(design_id), final_selection)
        create_pml_file("{}_RK".format(design_id), WY_selection)
        create_pml_file("{}_WY".format(design_id), RK_selection)
        create_pml_file("{}_F".format(design_id), F_selection)
```

Appendix C. Example RosettaScripts XML protocol for design of transmembrane pores. This script is provided solely as a guideline since it has not been optimised for compatibility with the most recent versions of RosettaScripts.

```
<ROSETTASCRIPTS>
<SCOREFXNS>
   <ScoreFunction name="sfxn_cstWt1_symm" weights="mbeta" symmetric="1">
      <Reweight scoretype="res_type_constraint" weight="2"/>
      <Reweight scoretype="aa_composition" weight="1.0" />
      <Reweight scoretype="coordinate_constraint" weight="1.0"/>
   </ScoreFunction>
</SCOREFXNS>
################################################
# RESIDUE_SELECTORS
################################################
<RESIDUE_SELECTORS>
   <ResidueName name="TS_selector" residue_name3="THR,SER" />
   <ResiduePDBInfoHasLabel name="RK_selector" property="RK_RING"/>
   <ResiduePDBInfoHasLabel name="TM_selector" property="OUTWARD_POLAR"/>
   <ResiduePDBInfoHasLabel name="WY_selector" property="WY_RING"/>
   <ResiduePDBInfoHasLabel name="F_selector" property="F_RING"/>
   <Not name="not_designable_res_selector_F" selector="F_selector"/>
   <Not name="not_designable_res_selector_TM" selector="TM_selector"/>
   <Not name="not_designable_res_selector_RK" selector="RK_selector"/>
   <Not name="not_designable_res_selector_WY" selector="WY_selector"/>
   <Layer name="core" select_core="true" core_cutoff="3.6" />
</RESIDUE_SELECTORS>
################################################
# Task Operations
################################################
<TASKOPERATIONS>
   <InitializeFromCommandline name="IFC"/>
   <IncludeCurrent name="IC"/>
   <LimitAromaChi2 name="aroChi"/>
   <ReadResfile name="readTmpResFile" filename="restrictRandom.resfile"/>
   <LayerDesign name="layerCBS" layer="core_boundary_surface_Nterm_Cterm">
      <boundary>
         <all exclude="W" />
         <Loop append="GM" />
         <Helix append="GM" />
      </boundary>
      <surface>
         <all exclude="W" />
      </surface>
      <core>
         <all exclude="W" />
         <Loop append="GM" />
         <Helix append="GM" />
      </core>
   </LayerDesign>
   <OperateOnResidueSubset name="restrict_to_pack_TM" selector="not_designable_res_selector_TM">
     PreventRepackingRLT/>
     <RestrictToRepackingRLT/>
   </OperateOnResidueSubset>
   <OperateOnResidueSubset name="restrict_to_pack_RK" selector="not_designable_res_selector_RK">
      PreventRepackingRLT/>
      <RestrictToRepackingRLT/>
   </OperateOnResidueSubset>
   <OperateOnResidueSubset name="restrict_to_pack_WY" selector="not_designable_res_selector_WY">
      PreventRepackingRLT/>
      <RestrictToRepackingRLT/>
   </OperateOnResidueSubset>
   <OperateOnResidueSubset name="restrict_to_pack_F" selector="not_designable_res_selector_F">
      PreventRepackingRLT/>
      <RestrictToRepackingRLT/>
   </OperateOnResidueSubset>
   <OperateOnResidueSubset name="restrict_to_pack_TS" selector="TS_selector">
      PreventRepackingRLT/>
      <RestrictToRepackingRLT/>
   </OperateOnResidueSubset>
  </TASKOPERATIONS>
#################################################
# Filters
#################################################
<FILTERS>

   <SSPrediction name="psipred_conf" confidence="0" cmd="/home/brunette/src/psipred3.21/runpsipred_single" use_probability="1" use_svm="false"/>
   <ResidueCount name="ALAResCount" residue_types="ALA"/>
   <ResidueCount name="ALLResCount" />
   <ResidueCount name="nres" confidence="0" />
   <ScoreType name="total_score" scorefxn="sfxn_cstWt1_symm" score_type="total_score" threshold="0.0" confidence="0"/>
   <Sasa name="Sasa" threshold="1500" upper_threshold="1000000" jump="1" confidence="1"/>
</FILTERS>
```

```
 ###############################################
  # Underlying relax/repack/min/pack movers
  ###############################################
  <MOVERS>
    AddResidueLabel name="addTS" residue_selector="TS_selector" label="TS" />
    <AddCompositionConstraintMover name="addcomp_TM" filename="TM.comp" selector="TM_selector" />
    <ClearCompositionConstraintsMover name="clear_comp_TM" />
    <AddCompositionConstraintMover name="addcomp_RK" filename="RK.comp" selector="RK_selector" />
    <ClearCompositionConstraintsMover name="clear_comp_RK" />
    <AddCompositionConstraintMover name="addcomp_WY" filename="WY.comp" selector="WY_selector" />
    <ClearCompositionConstraintsMover name="clear_comp_WY" />
    <AddCompositionConstraintMover name="addcomp_F" filename="F.comp" selector="F_selector" />
    <ClearCompositionConstraintsMover name="clear_comp_F" />
    <AddConstraintsToCurrentConformationMover name="constrainCA" cst_weight="1.0" use_distance_cst="False" coord_dev="1.0" bound_width="0.5" CA_only="True"
bb_only="False"/> #
    <SymPackRotamersMover name="pack_TM" scorefxn="sfxn_cstWt1_symm" task_operations="restrict_to_pack_TM,restrict_to_pack_TS"/>
    <SymPackRotamersMover name="pack_RK" scorefxn="sfxn_cstWt1_symm" task_operations="restrict_to_pack_RK,restrict_to_pack_TS"/>
    <SymPackRotamersMover name="pack_WY" scorefxn="sfxn_cstWt1_symm" task_operations="restrict_to_pack_WY,restrict_to_pack_TS"/>
    <SymPackRotamersMover name="pack_F" scorefxn="sfxn_cstWt1_symm" task_operations="restrict_to_pack_F,restrict_to_pack_TS"/>
    <FastRelax name="frelax" scorefxn="sfxn_cstWt1_symm" repeats="3" />
    <DetectSymmetry name="detect" keep_pdb_info_labels="true"/>
    <SymMinMover name="min" scorefxn="sfxn_cstWt1_symm" bb="1" chi="1" jump="ALL" />
    <ExtractAsymmetricUnit name="asymm_pose" keep_virtual="False" keep_unknown_aas="False" />
    <SetupForSymmetry name="symm_pose" definition="C8_Z.sym" preserve_datacache="False" keep_pdb_info_labels="true"/>
    <ParsedProtocol name="combo_min">
        <Add mover_name="pack_TM"/>
        <Add mover_name="min"/>
    </ParsedProtocol>

      </MOVERS>
<APPLY_TO_POSE>
</APPLY_TO_POSE>
###############################################
#  main
###############################################
<PROTOCOLS>
    <Add mover_name="symm_pose"/>
    <Add mover_name="addcomp_TM"/>
    <Add mover_name="pack_TM"/>
    <Add mover_name="clear_comp_TM"/>
    <Add mover_name="addcomp_RK"/>
    <Add mover_name="pack_RK"/>
    <Add mover_name="clear_comp_RK"/>
    <Add mover_name="addcomp_WY"/>
    <Add mover_name="pack_WY"/>
    <Add mover_name="clear_comp_WY"/>
    <Add mover_name="addcomp_F"/>
    <Add mover_name="pack_F"/>
    <Add mover_name="clear_comp_F"/>
    <Add mover_name="frelax"/>
    <Add filter_name="Sasa"/>
</PROTOCOLS>
<OUTPUT scorefxn="sfxn_cstWt1_symm" />
</ROSETTASCRIPTS>
```