Supplementary Materials for

De novo design of potent and resilient hACE2 decoys to neutralize SARS-CoV-2

Thomas W. Linsky*, Renan Vergara*, Nuria Codina*, Jorgen W. Nelson*, Matthew J. Walker, Wen Su, Christopher O. Barnes, Tien-Ying Hsiang, Katharina Esser-Nobis, Kevin Yu, Z. Beau Reneer, Yixuan J. Hou, Tanu Priya, Masaya Mitsumoto, Avery Pong, Uland Y. Lau, Marsha L. Mason, Jerry Chen, Alex Chen, Tania Berrocal, Hong Peng, Nicole S. Clairmont, Javier Castellanos, Yu-Ru Lin, Anna Josephson-Day, Ralph S. Baric, Deborah H. Fuller, Carl D. Walkey, Ted M. Ross, Ryan Swanson, Pamela J. Bjorkman, Michael Gale Jr., Luis M. Blancas-Mejia, Hui-Ling Yen, Daniel-Adriano Silva†

*These authors contributed equally to this work.
†Corresponding author. Email: dadriano@neoleukin.com

Published 5 November 2020 on *Science* First Release
DOI: 10.1126/science.abe0075

This PDF file includes:

Other Supplementary Material for this manuscript includes the following:
(available at science.sciencemag.org/cgi/content/full/science.abe0075/DC1)

Materials and Methods

Computational de novo protein design of ACE2 decoys

Spike protein RBD-binding motifs interacting with ACE2 were identified and extracted from publicly available structures (PDBs: 6CS2, 6VW1, 6M17); one (H1) spanning residues 19-53; another (H2) spanning residues 55-84; and in some designs, a third beta-hairpin motif (EE3) spanning residues 346-360. The surface-exposed residues from these motifs (ACE2 residue numbers 19, 23, 24, 27, 28, 30, 31, 34, 35, 37, 38, 41, 42 and 45 from H1; 61, 64, 68, 72, 75, 76, 79, 82 and 83 from H2; and 352, 353, 354, 355 and 357 from EE3, if present) were preserved, and the buried residues were allowed to change. The supporting structures were either computationally generated and placed by an available method (e.g. Rosetta combinatorial fragment assembly, parametric generation, etc) or extracted from existing structures. The position and orientation of the new elements was refined using a monte carlo search to optimize core packing. This protocol which idealized the motifs and generated fully-connected backbones using a clustered database of known protein structure fragments. A total of 34,589 designs were generated and filtered based on computational metrics. Of the designs that passed the filters, biased forward folding simulations (see below) were run on 1,899 designs to eliminate designs to study the global energy minima. Of the designs that passed the filters, 196 were ordered from four the sets of top-ranked designs in: 1) Rosetta score per residue; 2) biased folding simulations; 3) ddG as computed using Rosetta. In addition, a set of hand-selected designs were synthesized and screened. First, the core elements (i.e., the starting motifs and supporting secondary structure) were rebuilt by identifying parametric equations of repetitive phi and psi angles (omega fixed to 180°) that result in secondary structures that recapitulated each of the target helices as close as possible, a "pitch" on the phi and psi angles was allowed every 3rd residue in order to allow the helices the possibility to have curvature. By using these parametric equations, the algorithm can vary the length of each of the core-elements up to +/- 12 amino acids (compared to the input structural motifs). All length variations of the core-elements were then reconnected pairwise with loops from a clustered database of highly ideal loops (fragment-size of 7 amino acids). To connect pairs of secondary structure elements, the mimetic building protocol aims to reconnect the idealized elements by pairs in all possible combinations. For each pair of secondary structure elements, the loop database was filtered to remove.

The databases of highly ideal fragments used for the design of the backbones for the de novo mimetics were constructed with the Rosetta application "kcenters_clustering_of_fragments" using an extensive database of non-redundant (publicly available) protein structures from the RCSB protein data bank, which was comprised of 7062 PDBs for the 7-mer database.

Folding simulations were biased towards the designed conformation (*49*) by using a small subset of fragments at each residue position with the lowest RMSD (9- and 3-mers) to the designed structure. The funnel-shaped energy landscape suggests that the designed structure is the global energy minima and has a substantial energy gap with respect to alternative conformations. In the plots shown for the simulations, each dot represents an independent biased

forward folding trajectory which was computed by monte carlo insertion of fragments from solved protein structures. Three sets of trajectories are computed; those using only the 5 fragments with lowest RMSD to the design (brown), those using fragments from the design model plus the 8 fragments with lowest RMSD (orange) and relax trajectories (blue).

Sequence and structural alignment to natural proteins

The CTC-445.2 design model was structurally aligned to the NCBI nr-PDB database of non-redundant PDB chains (*50*); p-value cutoff of 1e-7). Sequential structural alignment was performed using TMalign with the default options. MICAN alignment (*51*) was used for inverse, Ca-only, alternative, and non-sequential structural alignments (flags="-n 1 -r"). TMscores (*52*, *53*) were normalized to CTC-445.2 and sequence identities were computed using only the structurally aligned residues. Alignments with less than 79 amino acids (the total length of the RBD-binding ACE2 motifs used to seed the designs) were discarded.

Yeast display screening

Genes encoding *de novo* ACE2 decoys and hACE2 were ordered as gBlocks (IDT), with 60 bases of overlap on each end to the pETcon3 yeast display plasmid. gBlocks and linearized pETcon3 (digested with NdeI and XhoI restriction enzymes and purified by gel extraction) were transformed together into *S. cerevisiae* strain EBY100 cells. Via homologous recombination, the genes for the designed proteins were placed in frame between the *AGA2* gene and the Myc tag on the plasmid, to allow protein expression on the surface of yeast with a c-terminal Myc tag. Yeast were first grown in C-Trp-Ura selective media to saturation, and later induced for 12-18 h in SGCAA media. Cells were then first washed with chilled PBSF buffer (50 mM $NaPO_4$, 150 mM NaCl, 0.1% w/v BSA, pH 7.4), and allowed to bind to 200 nM SARS-CoV-2 Spike RBD-mFc (Sino Biological; residues Arg319-Phe541), for 30 min on ice. Cells were washed with chilled PBSF buffer again, and tagged with FITC-conjugated chicken anti-cMyc antibody (ICL #CMYC-45F, Immunology Consultants Laboratory, Inc, Lake Oswego, OR, USA), at 0.8 µL per 10^6 cells, and phycoerythrin-conjugated goat anti-mFc antibody (#115-116-071, RRID: AB_2338626, Jackson ImmunoResearch, West Grove, PA, USA), at 1.6 µL per 10^6 cells, for 15 min on ice, and washed again with chilled PBSF buffer. Lastly, cells were analyzed by flow cytometry using the Guava easyCyte flow cytometer.

To characterize the binding affinity and competition to hACE2 of CTC-445, yeast display assays were performed as described above, using varying amounts of SARS-CoV-2 Spike RBD-mFc (0-243 nM) for the titration assay and 9 nM SARS-CoV-2 Spike RBD-mFc with varying concentrations of soluble human ACE2 His-tag (Sino Biological; residues Met1-Ser740) (0-729 nM) for the competition assay.

Directed evolution

Error prone PCRs (epPCR) of the gene coding for CTC-445 were performed using a GeneMorph II Random Mutagenesis Kit (Agilent), according to the standard protocol. Two separate PCR reactions were performed using 10 ng and 1 ng of the CTC-445-coding gene as the initial target DNA, in order to obtain a higher and lower error rate pool. Amplification was

performed with the pETCON_overlap_fwd (AGTGGTGGAGGAGGCTCTGGTGGAGGCGGTAGCGGAGGCGGAGGGTCGGCTAGCCATATG) and pETCON_overlap_rev (AGATCTCTATTACAAGTCCTCTTCAGAAATAAGCTTTTGTTCGGATCCGCCCCCCTCGAG) primers. Thirty cycles of amplification were used for the PCR, with an annealing temperature of 55°C. The products of both PCRs were combined, and purified by ethanol precipitation. 12 µg of CTC-445 epPCR DNA, together with 4 µg of linearized pETcon3 vector, were transformed by electroporation into conditioned *S. cerevisiae* strain EBY100 cells. The transformation efficiency was $2x10^6$ transformants.

Six rounds of cell sorting were performed on the library of CTC-445 epPCR, to identify mutations that improve binding to SARS-CoV-2 and improve its stability to proteases. Yeast were grown in C-Trp-Ura selective media and later induced for 12-18 h in SGCAA media. Cells were pretreated before primary labeling with a mixture of trypsin (12.5 µg/mL) and chymotrypsin (5 µg/mL) in TBS buffer (25 mM Tris-HCl, 150 mM NaCl, pH 8.0) for 5 min at room temperature. This reaction was halted by adding a large excess volume of ice-cold TBSF 1% (25 mM Tris-HCl, 150 mM NaCl, 1% w/v BSA, pH 8.0) and washing the cells four additional times with TBSF 1%. Cells were then labelled with decreasing concentrations of SARS-CoV-2 Spike RBD-mFc at every round of sorting: 200 nM (round 1, 10^8 cells sorted), 10 nM (round 2, 10^7 cells sorted), 1 nM (round 3, 10^6 cells sorted), 100 pM (round 4, 10^6 cells sorted), 10 pM (round 5, 10^6 cells sorted) and 1 pM (round 6, 10^6 cells sorted), for 30 min on ice, and washed with chilled PBSF buffer. Starting at sorting round 4, an additional selection step for improved k_{off} was added, where cells were incubated in 1 ml PBSF buffer at 37 °C for 1 hour, and washed again with chilled PBSF buffer. Lastly, cells were incubated with FITC-conjugated chicken anti-cMyc antibody (ICL #CMYC-45F, Immunology Consultants Laboratory, Inc, Lake Oswego, OR), at 0.3 µL per 10^6 cells for sorts 1 and 2, and 3 µL per 10^6 cells for remaining sorts and phycoerythrin-conjugated goat anti-mFc antibody (#115-116-071, RRID: AB_2338626, Jackson ImmunoResearch, West Grove, PA, USA), at 0.4 µL per 10^6 cells for sorts 1 and 2, and 4 µL per 10^6 cells for remaining sorts, for 15 min on ice, and washed again with chilled PBSF buffer. Fluorescence-activated cell sorting (FACS) was performed with the Sony SH800 instrument, selecting the top 5% of the displaying subpopulation by their PE/FITC fluorescence ratios. Lastly, selected cells were plated on SDCAA agar plates and single colonies were picked, amplified by PCR, and sequenced by Sanger sequencing (Genewiz).

Cloning and molecular biology

The dsDNA CTC-445 gBlock with pETcon3 overlaps (5' overlap: 5'-gctagtggtggaggaggctctggtggaggcggtagcggaggcggagggtcggctagccatatg-3', 3' overlap: 5'-ctcgaggggggcggatccgaacaaaagcttatttctgaagaggacttgtaatagagatct-3') was digested with NdeI and XhoI endonucleases (New England Biolabs) for 1 hour at 37°C followed by ligation into linear pET29b(+) using T4 DNA Ligase (New England Biolabs). The vector was linearized by 100-fold overdigestion by NdeI and XhoI and then purified by gel extraction (Qiagen). For the optimized CTC-445 variants, genes with overlaps for pET29b(+) (5':

AAATAATTTTGTTTAACTTTAAGAAGGAGATATACATATG, 3': CTCGAGCACCACCACCACCACCACTGAGATCCGGCTGCTA) were designed, ordered and cloned by overlap extension or Gibson assembly (*54*). To facilitate protein quantification, and to enable cleavage of the C-terminal His tag, a C-terminal GSGWGSGLVPRGS sequence was added to all CTC-445-derived sequences.

Recombinant protein expression

Protein sequences were synthesized (IDT) and cloned into pET-29b(+) E. coli plasmid expression vectors (both with and without a thrombin-cleavable C-terminal 6-His tag). Plasmids containing the clones of CTC-445, one non-expressing negative control and a positive control protein transformed into chemically competent E. coli Lemo21 cells (NEB). Protein expression was then induced with 1 mM of isopropyl β-d-thiogalactopyranoside (IPTG) at 37 °C. After 4 h expression, samples were collected and run on SDS-PAGE gel along with a PageRuler Plus Prestained Protein Ladder.

Chemical protein denaturation

Samples of CTC-445, CTC-445.2 and CTC-445.2d were incubated overnight at 1 mg/mL with 0-6 M guanidinium chloride in PBS buffer (pH 7.4). Circular dichroism spectra were measured for each sample at 25 °C in a 0.1 mm path-length cuvette using an Chirascan spectropolarimeter V100 (Applied Photophysics). The unfolding curve at 222 nm was fitted to the following equation:

$$fD = \frac{\left((yN+mN[GuHCl])+(yD+mD[GuHCl])*exp^{-\left(\frac{\Delta GND-mND[GuHCl]}{RT}\right)}+(yI+mI[GuHCl])*exp^{-\left(\frac{GID-mID[GuHCl]}{RT}\right)}\right)}{\left(1+exp^{-\left(\frac{\Delta GND-mND[GuHCl]}{RT}\right)}+exp^{-\left(\frac{\Delta GID-mID[GuHCl]}{RT}\right)}\right)}$$

Where fD is the fraction of protein in the unfolded state, yN, yI and yD are the signals of the native, intermediate and unfolding states, mN, mI and mD represent the signal dependence with the denaturant concentration, ΔGND and ΔGID are the free energy changes for the native and intermediate states while mND and mID represent their dependence with de denaturant concentration. Finally, R represents the gas constant in kcal mol^{-1} K^{-1} and T the temperature in K. ΔGNI is calculated from the subtraction of ΔGID to ΔGND.

Circular dichroism

Far-ultraviolet circular dichroism measurements were carried out for de novo decoys and hACE2 (Sino Biological, Cat#40592-V08B-B) with an CHIRASCAN spectrometer V100 (Applied Photophysics) in PBS buffer (pH 7.4) in a 0.1 mm path-length cuvette with protein concentration of 0.2 mg/mL (unless otherwise mentioned in the text). Temperature melts were obtained from 20 to 95 °C and monitored absorption signal at 222 nm (steps of 0.5 °C per min, 30 s of equilibration by step). Wavelength scans (195–250 nm) were collected at 20 °C, 95 °C, and again at 20 °C after refolding (roughly 5 min). Melting temperature (Tm) values were calculated from the fitting of the thermal melts to the following equation:

$$fD = \frac{\left((yN+mNT)+(yD+mDT)*exp\left[\frac{\Delta HvH}{R}\left(\frac{1}{T}-\frac{1}{Tm}\right)\right]\right)}{\left(1+exp\left[\frac{\Delta HvH}{R}\left(\frac{1}{T}-\frac{1}{Tm}\right)\right]\right)}$$

Where fD is the fraction of the protein in the unfolded state, HvH is the enthalpy change associated to the unfolding process, yN and mN represent the signal of the native state and its dependence with the temperature, yD and mD represent the signal of the unfolded state and is dependence with the temperature and R is the gas constant.

Serial thermal ramping protein stability

Serial thermal ramping assays were performed in order to test the unfolding reversibility (thermal recovery) of CTC-445, CTC-445.2, CTC-445.2d and hACE2 (Sino Biological, Cat#40592-V08B-B); to this end, the UNCLE platform (UNcle-0469R, Unchained Labs) was used. 8.8 µL of CTC-445, CTC-445.2 and CTC-445.2d at 0.5 mg/mL, and hACE2 at 0.25 mg/mL, were loaded in capillary cuvettes and fluorescence spectra (~300-400 nm) were measured for each sample while temperature was increased and decreased repeatedly. Temperature ramps were performed in the following order: 20 °C to 35 °C, 20 °C to 50 °C, 20 °C to 65°C, 20 °C to 80 °C, 20 °C to 95 °C, 20 °C to 80 °C, 20 °C to 65 °C, 20 °C to 50 °C and 20 °C to 35 °C. All samples were in PBS buffer pH 7.4. The barycentric mean was calculated for each fluorescence spectrum using the UNCLE analysis program (Unchained Labs).

Size-exclusion chromatography with multi-angle light scattering (SEC-MALS)

SEC-MALS assays were performed using a 1260 infinity II LC HPLC system (Agilent Technologies) with a Superdex 75 10/300 size exclusion column (GE Healthcare), coupled to an inline static light scattering instrument (MiniDawn, Wyatt Technology) and differential refractive index (Optilab rEX, Wyatt Technology) and UV detection systems. Protein samples were prepared in a PBS buffer at concentrations ranging from 2 to 4 mg/mL and filtered with 0.22 µm syringe filters; 100 µL of the samples were injected and ran at a flow rate of 0.5 mL/min. Data were analyzed using the software ASTRA 7 (Wyatt Technologies).

ELISA competition assays

SARS-CoV2 S Protein RBD (Acro, Cat#EP-105) was coated at 0.5 µg/mL (100 uL/well) in flat, clear bottom, high binding polystyrene 96-well plate (Thermo, Cat#15041) overnight at 4°C. On the following day, each assay plate was washed 3X with 0.05% Tween-PBS and blocked with 2% BSA/0.05%Tween-PBS for 1-1.5 hours at 37°C. On the following day, 11-point four-fold serial dilutions of each test article were prepared starting at 2 µM (2X final). In addition, a dose response of biotin-hACE2 was prepared by performing a two-fold serial dilution starting at 0.174 µg/mL (2 nM). From the 0.174 µg/mL biotin-hACE2, a constant concentration of 0.07 µg/mL (0.8 nM, 2X final) was prepared to be used for inhibition by test articles. After blocking, each assay plate was washed (3X) and each test article dilution was added to the plate at 50 µL/well (single replicate well per concentration) followed by 50 µL/well of biotin-hACE2 (~0.035 µg/ml, equivalent to a concentration of 0.4 nM). Incubation of test articles with 0.4 nM biotin-hACE2 was performed for 1 hour at 37°C. Afterwards, each assay plate was washed (3X),

followed by incubation of 0.1 µg/mL (100 µL/well) streptavidin-HRP for 1 hour at 37 °C. Lastly, each plate was washed (3X) before addition of 100 µl/well TMB (abCam, Cat#171524). TMB was incubated for 7-8 minutes and the reaction was stopped with the addition of 50 µL/well TMB Stop Solution (abCam, Cat# ab171529). Mean absorbance in each well was measured from 4 spots at 450 nm for 5 different samples at each concentration.

Biolayer interferometry

Binding function of CTC-445 was analyzed by biolayer interferometry. Binding data were collected in an Octet RED96 (ForteBio) and processed using the instrument's integrated software using a 1:1 binding model for CTC-445, CTC-445.2 and CTC-445.3, and 2:1 for CTC-445.2d, CTC-445.3d, and CTC-445.2t. SARS-CoV-2 Spike Protein (RBD domain only, his Tag, Sino Biological, Cat#40592-V08B-B) were immobilized to Anti-Penta-His sensors (HIS1K, ForteBio) at 2 µg/mL in binding buffer (10 mM HEPES, pH 7.4, 150 mM NaCl, 3 mM EDTA, 0.05% surfactant P20, 0.5% non-fat dry milk). After baseline measurement in the binding buffer alone, the binding kinetics were monitored by dipping the biosensors in wells containing defined concentrations of the designed protein (2.4-200 nM) (association) and then dipping the sensors back into baseline wells (dissociation).

Expression and purification of SARS-CoV-2 S ectodomain

Expression and purification of SARS-CoV-2 ectodomains were conducted as previously described (*55*). Briefly, constructs encoding the SARS-CoV-2 S ectodomain (residues 16-1206) with 6P stabilizing mutations (*56*), a mutated furin cleavage site between S1 and S2, and a C-terminal hexahistidine tag were used to express soluble SARS-CoV-2 S ectodomains. SARS-CoV-2 6P S trimers were purified from the supernatants of transiently-transfected Expi293F cells (Gibco) by nickel affinity and size-exclusion chromatography. Peak fractions were identified by SDS-PAGE, and fractions corresponding to S trimers were stored at 4˚C.

Cryo-EM sample preparation, data collection, processing and refinement

Purified CTC445.2 was incubated on ice for 30 min with SARS-CoV-2 S 6P trimer (1.1:1 molar ratio CTC445.2 per S protomer) to a final complex concentration of 2.5 mg/mL. Cryo-EM sample grids were made as previously described (*55*).

For cryo-EM data collection, 2,250 movies were collected using Serial EM (*57*) with beam-image shift over a 3 by 3 pattern of 1.2 µm holes with a single exposure per hole on a 200 kV Talos Artica (ThermoFisher). Movies were collected in super resolution mode on a K3 with a pixel size of 0.438 Å. Pre-processing was done in cryoSPARC (*58*) v2.15, including patch motion correction with binning by 2 and patch CTF estimation. Micrographs were screened by estimated CTF fit (≤ 6Å) and relative ice thickness (≤ 1.2), resulting in 1,610 images that were used for particle picking via Blob picker. 658,188 particles were extracted and downsampled by 4 (108 box size, 3.344 Å/pixel) and subjected to reference-free 2D classification. 2D class averages which displayed distinct views and secondary structural features were selected, resulting in 420,998 particles used to generate three *ab initio* volumes. The full set of particles (187,629) corresponding to the CTC445.2-S complex were re-extracted unbinned (432 box size,

0.869 Å/pixel), homogeneously refined, and underwent CTF refinement and aberration correction.

The resulting volume was refined to an overall estimated resolution of 3.4Å (FSC=0.143), but flexibility and complex heterogeneity precluded analysis of the CTC445.2-RBD interface. Therefore, particles were further subjected to iterative rounds of 3D classification and homogeneous refinement. This procedure yielded four almost equally distributed classes with different CTC445.2 bound states, with overall estimated resolutions ranging from 3.8-4.2Å (FSC=0.143). To further improve features of the CTC445.2-RBD interface, complex states 3 and 4 were grouped and used to generate a focused mask that removed CTC445.2 bound "up" RBDs. Local refinement that was focused on the CTC445.2 bound "partially-down" RBD resulted in an overall resolution of 4 Å, with improved features at the CTC445.2-RBD interface.

Coordinates for the state 4 complex were generated by rigid-body docking individual chains from reference structures (PDB 6XKL and computationally-derived models of CTC445.2) into the final refined cryo-EM volumes using UCSF Chimera (*59*). Models were then refined using iterative rounds of real space refinement in Phenix (*60*) with rotameric and NCS restraints applied, followed by manual refinement in Coot (*61*) to correct geometry and ramachandran outliers. Validation of model coordinates was performed using MolProbity (*62*) with mean ADP B-factors of 175 and 180 $Å^2$ for protein and ligands, respectively.

Deep mutational scanning

All positions in the CTC445.2 protein, and positions in the SARS-CoV-2 S protein RBD within 10Å of the ACE2 interface were mutated by PCR mutagenesis using custom primers (IDT, Coralville, IA) to introduce an NNK codon at each position of interest, as described previously (*63*). PCR products were purified using a QiaQuick gel extraction kit (Qiagen, Hilden, Germany), and 4 µg of each resulting site-saturation mutagenesis (SSM) library was transformed independently along with 1 µg of linearized pETcon3 vector into EBY100 yeast via electroporation with a final transformation efficiency of $1x10^6$ transformants, as described previously (*64*). The resulting libraries were expressed by surface display. The RBD SSM library was sorted by FACS at varying concentrations of CTC-445.2 or hACE2 (656.1, 218.7, 72.9, 24.3, 8.1, 2.7, 0.3, 0.1 nM, Sino Biological Cat#10108-H08H) and the CTC445.2 SSM library was sorted at varying concentrations of mFc tagged SARS-CoV-2 RBD (100, 50, 25, 12.5, 6.25, 3.125, and 1.625 pM, Sino Biological Cat#40592-V05H), as described for the directed evolution experiment, with all cells showing binding fluorescence above the no-binder baseline collected for downstream analysis. In addition, for each pool, a nonbinding pool was collected representing all cells that showed expression in the absence of either CTC-445.2 or hACE2 or the SARS-CoV2 RBD, respectively. All sorted libraries, as well as an unselected library were processed to recover plasmid DNA, and were then amplified via two rounds of PCR followed by QiaQuick gel extraction using custom primers to add Illumina sequencing primer bind sites, barcodes, and flow cell adapters, as previously described (*65*). Prepared libraries were then sent for Illumina sequencing (GENEWIZ, South Plainfield, NJ), with a yield of between ~802k and ~1440k reads per library. Enrichment was evaluated for mutations at the following positions in

the SARS-CoV-2 S protein RBD: 401, 402, 403, 404, 405, 406, 408, 409, 410, 416, 417, 418, 419, 420, 421, 422, 437, 439, 440, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508. Sequencing data was matched to the DNA sequence for the RBD or CTC-445.2 without alignment (33 bases were trimmed from the 5' end and 30 from the 3' end); any sequences containing insertions or deletions were discarded. Sequencing data was filtered to remove bases with phred quality score Q<15. To ensure sufficient coverage to compute enrichment, only mutations with more than 10 counts in the expression-only dataset were analyzed. To determine enrichment, the number of occurrences in each dataset of each possible position and amino acid were first computed. Next, enrichment factors for all amino acid substitutions were determined. For each dataset, enrichment was calculated as the proportional counts for each dataset divided by the proportional counts for the expression-only dataset, giving 8 enrichment values for each position and amino acid in the RBD library corresponding to the 8 concentrations tested for both CTC-445.2 and ACE2, with 7 values in the CTC445.2 library for its 7 tested concentrations of RBD. The natural logarithm of the maximum enrichment of these values for ACE2, CTC-445.2, or the RBD was used as an estimate of the effect of each amino acid substitution on binding.

VSV-luc pseudovirus neutralization

Neutralization activity was determined using a non-replicative VSV pseudovirus containing a firefly luciferase gene and expressing the spike protein from Wuhan-1 SARS-CoV-2 isolate (GenBank: QHD43416.1).

Neutralization assays were performed with either SARS-CoV-2 / VSV-Luc (expressing the full ectodomain of the SARS-CoV-2 spike protein on the surface of the pseudovirus) or VSVg / VSV-Luc (expressing VSVg). Thirty thousand 293T-ACE2 cells were seeded the day before in white plates in the presence of hygromycin (100 μg/mL). The day of the neutralization assay, 25 μL of media containing pseudovirus was mixed with 25 μL of serial dilutions of the test-item in a different plate, and then incubated for 1 h at 37°C. Pseudovirus and test-item dilutions were performed in DMEM media containing 5% heat-inactivated fetal bovine serum (DMEM5). After the 60 minute incubation, the test-item / pseudovirus mixture was added to the 293T-ACE2 cells. Infection was allowed for twenty-four hours. Firefly luciferase activity was monitored at 24h using the Britelite reporter gene assay (Perkin Elmer).

Test-items were evaluated in duplicate using serial 3-fold dilutions. Controls included cells infected with a VSV pseudovirus lacking spike protein (NoEnv / VSV-Luc), with pseudovirus carrying VSVg (VSVg / VSV-Luc), or uninfected cells ("mock-infected"). Pseudovirus-infected cells were incubated with test-item or vehicle alone (DMEM5). Some wells included cells challenged with pseudoviruses incubated with dilutions of plasma from a convalescent patient recovered from COVID-19 (COV+) or with a control plasma (COV-). Before collecting blood from COV+, this patient had been confirmed infected with SARS-CoV-2 by RT-PCR, and also for the presence of IgGs against SARS-CoV-2 Spike (S) evaluated with a lateral flow antibody

test. COV- plasma was derived from a PCR-negative and IgG-negative for S antibodies.

Average RLU values in mock-infected cells were subtracted from all values ("mock-subtracted values"). The average RLU value of infected samples in the presence of vehicle was obtained. For each data point, once the mock was subtracted, the value was divided by the average value in infected cells in the presence of vehicle alone and then multiplied by 100 to obtain the percentage infectivity value normalized to samples with vehicle alone. The average percentage infectivity for each concentration together with the standard deviation of duplicates was obtained from duplicate values generated as shown in point 4.

The 50% viral inhibitory concentration (EC_{50}) was determined with GraphPad Prism. The EC_{50} was defined as the concentration of test-item at which the relative luminescence units (RLUs) were reduced by 50% as compared with the values in cells infected with pseudovirus in the absence of test-item). All measurements were carried out in 2 biological replicates.

Viral infection assay of the SARS-CoV-2 virus in Vero E6 cells

Vero E6 cells (ATCC #CRL-1586, ATCC, Manassas, VA) were maintained in Dulbecco's Modified Eagle Medium (DMEM) supplemented with 4.5 g/L D-glucose, 100 mg/L sodium pyruvate, 10% FBS, 100,000 U/L penicillin–streptomycin and 25 mM HEPES. SARS-CoV-2 virus (BetaCoV/Hong Kong/ VM20001061/2020), was isolated from the nasopharynx aspirate and throat swab a confirmed COVID-19 patient in Hong Kong at the end of January 2020 (GISAID# EPI_ISL_412028) and was passaged 3 times in Vero E6 cells. All experiments using the SARS-CoV-2 isolate were performed at the core BSL3 facility, LKS Faculty of Medicine, The University of Hong Kong. The inhibitory effects of CTC-445.2 and CTC-445.2d were firstly evaluated with SARS-CoV-2 at the multiplicity of infection (MOI) of 1.0 or 0.01 TCID50/cell in 96-well plates. Three-fold serially diluted CTC445.2 (final concentrations ranged from 30 μM to 0.01 μM) and CTC445.2d (final concentrations range from 10 μM to 0.005 μM) were pre-incubated with diluted SARS-CoV-2 virus for 1 hour at 37°C and the mixture was added to Vero E6 and further incubated 1 hour at 37°C. The inoculum was removed and cells were overlaid with fresh infection media (DMEM supplemented with 4.5 g/L D-glucose, 100 mg/L sodium pyruvate, 2% FBS, 100,000 U/L Penicillin-Streptomycin, and 25 mM HEPES) containing matched concentrations of CTC-445.2 and CTC-445.2d. After 72 hours incubation at 37°C, culture supernatants were collected to determine viral load (viral RNA copies/mL) using TaqMan™ Fast Virus 1-Step Master Mix as described (66). Four-parameter logistic regression (GraphPad Prism) was used to fit the dose-response curves and determined EC_{50} of CTC-445.2 and CTC-445.2d. To confirm the mechanism of inhibition, time-of-addition assay was used to evaluate the inhibitory effect of CTC-445.2 and CTC-445.2d. To test the effect of the, decoy proteins in blocking SARS-CoV-2 virus entry, 3-fold serially diluted CTC-445.2 and CTC-445.2d were pre-incubated with SARS-CoV-2 for 1 hour at 37°C and the mixture was added to Vero E6 cells and further incubated for 1 hour at 37°C. The inoculum was then removed and the cells were washed in PBS three times before fresh infection media (without CTC-445.2 or CTC-445.2d) was added to the cells. To test the effect of the decoy protein in blocking SARS-CoV-2 under multi-cycle replication, SARS-CoV-2 virus was allowed to

incubate with Vero E6 cells for 1 hour at 37°C; the inoculum was removed and the cells were washed in PBS three times before infection media containing 3-fold serially diluted CTC-445.2 and CTC-445.2d were added. Culture supernatants were collected at 72 hours post-infection to determine viral load (viral RNA copies/mL). Three biological replicates were performed. Dose-response data was fit to a three-parameter nonlinear fit using the scipy python module (67), with the upper bound of the curve fixed to the experimentally determined data points at concentration=0 μM.

Viral infection assay of the NanoLuc SARS-CoV-2 virus in Calu-3 cells

Calu-3 (ATCC HTB-55) from American Type Culture Collection (ATCC, Manassas, VA) was maintained in growth medium comprising OptiMEM with 20 mM HEPES, 10 mM GlutaMAX, 4% FBS, 10,000 IU/mL Penicillin, 10,000 μg/mL Streptomycin, and 10ng/mL Epidermal Growth Factor (EGF). Vero cells (USAMRIID) and icSARS-CoV-2-nLuc) (Isolate USA-WA1/2020) were generously provided by Dr. Ralph Baric Lab at University of North Carolina at Chapel Hill (Hou et al., Cell. 182, 1-18. 2020). icSARS-CoV-2-nLuc virus stock was propagated in Vero cells and stored at -80 °C. Calu3 cells are seeded in white 96-well flat-bottomed tissue culture plates. All viral infections were performed in the biosafety level 3 facility at the University of Washington. Three-fold serial dilution of test compound CTC-445.2 (final concentration ranging from 20 μM to 0.01 μM) and CTC-445.2d (final concentration ranging from 10 μM to 0.0046 μM) were added to corresponding wells during virus infection at MOI of 1.0; three independent experiments were performed in duplicate. After virus adsorption at 37℃ for 1 h, inoculum was removed and replaced with fresh growth media containing the same compound concentration to maintain the treatment throughout the entire infection course. To estimate the antiviral effects of test compounds, icSARS-CoV-2-nLuc virus replication was measured by adding Nano-Glo Endurazine Live Cell substrate to each well, and nanoLuc readings of the same well were taken at 24, 48, and 72 hours post-infection. NanoLuc reads were normalized to the untreated wells, and the inhibitory effects of the compounds on virus replication were analyzed by non-linear regression curve fit algorithm by using GraphPad Prism.

Human proteome binding specificity

CTC445.2d at 1 μg/mL (=30 nM) chemically labeled with Alexa-647 was used for the protein-protein interaction assay. A HuProt™ array (CDI Laboratories, Baltimore, MD) with barcodes containing 21,218 proteins and protein isoforms (including S protein RBD) was incubated with the protein sample for 1 hour. After probing,the arrays were washed and images were taken for data collection. CDI software was used to quantify the binding profile of each individual sample to specific proteins on the array based on Z Scores. Z scores were computed using the average Z score of the duplicate spots of a given protein (each protein is printed in duplicate on a HuProt™ array). The Z score of each spot on a given array is calculated according to the formula Z = [F635–F635(avg)] / F635(std), where F635 is the average foreground signal intensity of 2 replicate spots of a given protein in the detection channel (635 nm); B635 is the average background signal intensity of 2 replicate spots of a given protein in the detection

channel; and F635(avg) and F635(std) are the average and standard deviation of the F635 values of all spots on the array, respectively. S score is the difference of the Z scores of a given protein and the one ranked after it. If the S score of the top hit is > 3, the test protein is considered as highly specific against that hit.

Cytotoxicity assay

Cytotoxicity of the test compounds in treated Vero-E6 or Calu-3 cells was tested using the Cell Counting Kit-8 (CCK-8) assay (Abcam). Cells were seeded in 96-well flat-bottom tissue culture treated plates. Four-fold serial dilutions of test compound CTC-445.2 and CTC-445.2d in fresh infection medium (final concentrations ranging from 60 μM to 3.7 nM for CTC-445.2 and 50 μM to 3 nM for CTC-445.2d, respectively) were added onto corresponding Vero E6 cells. Three-fold serial dilutions of the test compounds of CTC-445.2 (final concentration ranging from 20 μM to 0.01 μM) or CTC-445.2d (final concentration ranging from 10 μM to 0.0046 μM) in Calu-3 cell growth media were added onto the cells. Cells were then incubated at 37 ℃, 5% CO_2 for 2 days. 10 μL CCK-8 reagent was added to each well, and the absorbance was measured at 460 nm.

Single-dose pharmacokinetics of the de novo decoys

Eight week old female Balb/c mice (DOB 28-Apr-2020; Charles River Strain Code 028) were anesthetized with isoflurane and a single 100 μg (30 μL of 3.37 mg/mL) dose of low-endotoxin (<5.0 EU/mg) CTC-445.2d was delivered intranasally. Mice were euthanized at indicated time points and whole blood and tissue (lungs, trachea or nasal turbinates) isolated. Five biological replicates were performed for plasma, lungs, and trachea, and three for nasal turbinates. Whole blood was centrifuged, and plasma was separated from cells by pipetting into a separate tube before freezing. Tissue lysates were prepared through mechanical disruption of tissue using Precellys tissue homogenizer (Bertin), followed by lysis with T-PER tissue lysis buffer (ThermoFisher) containing protease/phosphatase inhibitor cocktail (ThermoFisher). Lysates were cleared by centrifugation and were frozen for analysis. A standard 96-well MSD (L15XA) plate was pre-coated with 50 μL/of 0.5 μg/mL SARS-CoV-2 S Protein RBD tagless (Sino, 40592-VNAH) overnight at 4 ℃ and sealed. Dilution of capture reagent was prepared in PBS. On the following day, the assay plate was washed (6X) with 0.05% Tween-PBS and blocked with 150 μL/well MSD Blocker A (5% protein-PBS solution) for a minimum of 2 hours at RT. Plate shaking was performed for all incubation periods. During blocking incubation, a standard curve was prepared by performing a 4-fold serial dilution of CTC-445.2d in either untreated lung lysate or untreated plasma at 10,000 ng/mL. Prior to addition to the assay plate, each sample and standard was diluted 5-fold (MRD = 5) into assay dilution buffer (0.5% MSD Blocker A). In addition, samples were diluted 10-fold in separate wells. After blocking, an assay plate was washed (6X) and 50 μL/well of each sample or standard dilution was added. Assay plate was incubated for additional 1 hour at RT with plate shaking followed by washing (6X). For detection, a 1-step detection method was used. With 1-step detection, rabbit anti-His mAb (RevMab, 31-1048-00) and SULFOTAG goat anti-rabbit IgG (H+L) pAb (MSD, R32AB) were

pre-mixed at 1 µg/mL each for 1 hour prior to addition to assay plate at 50 µL/well. After the last wash, 150 µL/well of MSD Gold Read Buffer was added and luminescence measured on the MSD Quickplex SQ120. Luminescence was converted to concentration based on a standard curve. All procedures were done in accordance with the National Research Council's Guide for the Care and Use of Laboratory Animals and The Animal Welfare Act and were reviewed and approved by The Bloodworks Northwest Research Institute Institutional Animal Care and Use Committee (IACUC, project ID 5360-01).

Multi-dose mouse toxicity of CTC-445.2d

A 100 µg flat dose (30 µL of 3.33 mg/mL stock) of low-endotoxin (<2.0 Eu/mg), His-tagged CTC-445.2d (~5 mg/kg for 20 g mouse) was intranasally administered to 18 female CD-1 mice (DOB 12-May-2020, age ~13 weeks; Charles River Strain Code 022) daily for 14 days. In addition, a control group of 5 untreated mice were dosed similarly with PBS. On days 1, 2, 4, 8, 11 and 14, body weight of surviving mice was recorded and three CTC-445.2d-treated mice were sacrificed for lung examination and CTC-445.2d quantification in lung lysate (data not shown). All procedures were done in accordance with the National Research Council's Guide for the Care and Use of Laboratory Animals and The Animal Welfare Act and were reviewed and approved by The Bloodworks Northwest Research Institute Institutional Animal Care and Use Committee (IACUC, project ID 5360-06).

Syrian hamster *in vivo* model for SARS-CoV-2 infection

Syrian hamsters (Mesocricetus auratus, males, 3 to 4 months of age), were purchased from Charles River (Wilmington, MA, USA). Hamsters were single housed in Tecnipast Greenline IVC unit – GR900 cages purchased from Tecniplast (West Chester, PA) containing Anderson Bed O' Cobs 1/8 inch animal bedding (The Andersons INC., Maumee, OH) as well as Nylabones (Lab Supply, Fort Worth, TX) for enrichment. Hamsters were provided with Lab Diet Rodent Chow (Lab Diet, St. Louis, MO) and fresh water ad libitum. The University of Georgia Institutional Animal Care and Use Committee approved all hamster experiments (A2020 04-024-Y1-A3; 27-May-2020), which were conducted in accordance with the National Research Council's Guide for the Care and Use of Laboratory Animals, The Animal Welfare Act, and the CDC/NIH's Biosafety in Microbiological and Biomedical Laboratories guide.

The hamsters were anesthetized using isoflurane and i.n. treated with 100 µL of CTC-445.2d at a concentration of 5.6 mg/mL, or with vehicle (PBS) for the control group. Hamsters were challenged, 12-hours later, with 100 µL of i.n. administered SARS-CoV-2 (Washington/1/2020) at 5e+5 plaque forming units (PFU) per hamster. Clinical signs for this study included lethargy, rough coat, moderate dyspnea, hunched back, and weight loss of 10-20%. Severe clinical signs requiring immediate euthanasia included sever dyspnea (open mouth gasping), lack of response to external stimuli, and weight loss >20%. All challenged hamsters were weighed daily up to day 7. Any hamster that exceeded 20% weight loss from its original weight or displayed severe clinical symptoms was humanely euthanized. The scoring system for assessing clinical symptoms is as follows, signs of: lethargy (1 point), rough coat (1

point), moderate dyspnea (2 points), hunched back (1 point), and body weight loss 10-20% (1 point). Hamsters that accumulated a clinical score of 3 or higher were euthanized. All procedures were performed in accordance with the Guide for the Care and Use of Laboratory Animals, Animal Welfare Act, and Biosafety in Microbiological Laboratories.

ACE2 solution enzymatic activity assay

Inhibition of ACE2 enzymatic activity was performed using the ACE2 Inhibitor Screening Kit (Promocell Heidelberg, Germany). 50 µL of ACE2 enzyme solution was added to each well of a 96-well white walled plate (Corning, New York). Duplicate test samples were diluted to 20 µM in the provided assay buffer for the highest concentration, with an 8 point 1 to 4 serial dilution. DX600 was used as the ACE2 inhibitor positive control. 10 µL of diluted samples were added to the ACE2 enzyme solution and incubated for 15 min at room temperature. 40 µL of ACE2 substrate mix was then added to each well. After 15 min, fluorescence was measured in an Infinite M1000 plate reader (Tecan Männedorf, Switzerland) with 320 nm excitation and 420 emission filters. Fluorescence values were normalized using negative controls.

ACE2 in vitro cellular enzymatic activity assessment

Vero E6 cells (ATCC #CRL-1586, ATCC, Manassas, VA) were transduced with an ACE2 Lentivirus (BPS Bioscience San Diego, CA) to increase ACE2 surface expression. ACE2-Vero cells were seeded in a 96-well white walled plate at 10,000 cells per well in 100 µL of DMEM with 10% FBS and 1% pen-strep and rested overnight. Test compounds CTC-445.2, CTC-445.2d, and positive control DX600 were serially diluted in a 96-well polypropylene plate. Concentration of test compounds started at 8 µM with an 8 point 1 to 4 serial dilution. Culture media was aspirated and cells were incubated with 50 µL of diluted test compounds for 1 hour at 37 °C. Mca-APK(Dnp) ACE2 substrate was diluted to 50 µM. 50 µL diluted ACE2 substrate was added to each well and incubated at 37°C. Data were collected for 10 wells (DX600), 5 wells (CTC-445.2) or 6 wells (CTC-445.2d). 20 hours after adding ACE2 substrate, fluorescence was read at 320 nm excitation and 400 nm emission on the Infinite M1000 plate reader (Tecan Männedorf, Switzerland).

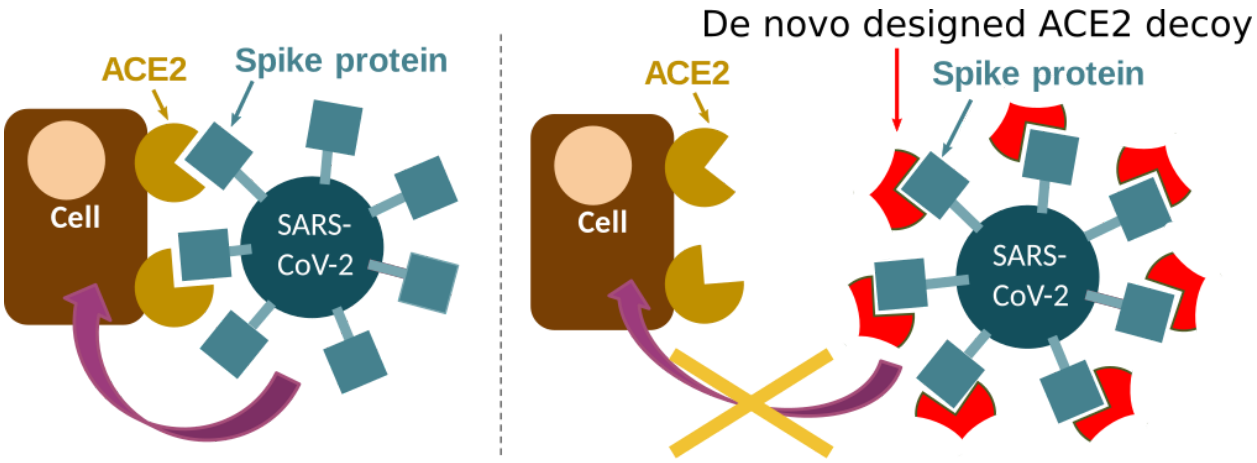


Fig. S1. Target mechanism of inhibition of SARS-CoV-2 by the de novo designed ACE2 decoys. SARS-CoV-2 (and SARS-CoV) gains entry into cells by first binding to the ACE2 receptor on the surface of human cells via the spike protein (left). The spike protein binds to the de novo designed ACE2 decoys using the same binding interface as natural ACE2, effectively making decoys functionally resilient to mutation. The SARS-CoV-2 spike protein binds to the decoys instead of native ACE2, sequestering the virus and preventing viral entry into cells while keeping native ACE2 function intact (right).

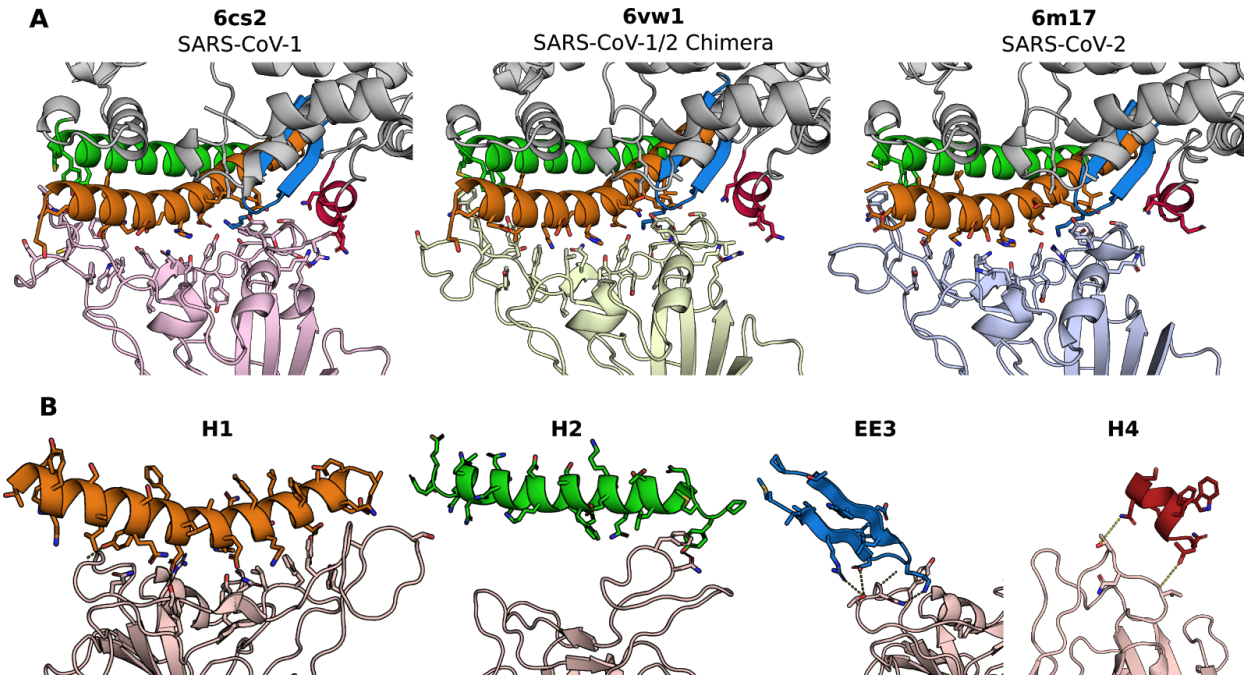


Fig. S2. Three dimensional structures of ACE2 in-complex with SARS-CoV-1 and SARS-CoV-2 RBD. A) Comparison of ACE2 models used to identify motifs for designed proteins. H1 (orange), H2 (green), EE3 (blue), and H4 (red) are the motifs that interact with the RBD of SARS-CoV-1 (pink, left), a SARS-CoV-1+SARS-CoV-2 chimera (light green, middle) and SARS-CoV-2 (light blue, right). Computational designs were created against one of these structures, but evaluated against the S protein RBD from all three RBDs. **B)** Structural motifs involved in binding of spike protein to ACE2 and their interactions with SARS-CoV-2 spike protein. H1 and H2 were present in all designs, and EE3 was present in the majority of designs. To simplify the design process of the de novo decoys, we did not incorporate the fourth binding motif (H4) in our design strategy because we predicted it to have a minor contribution to binding.

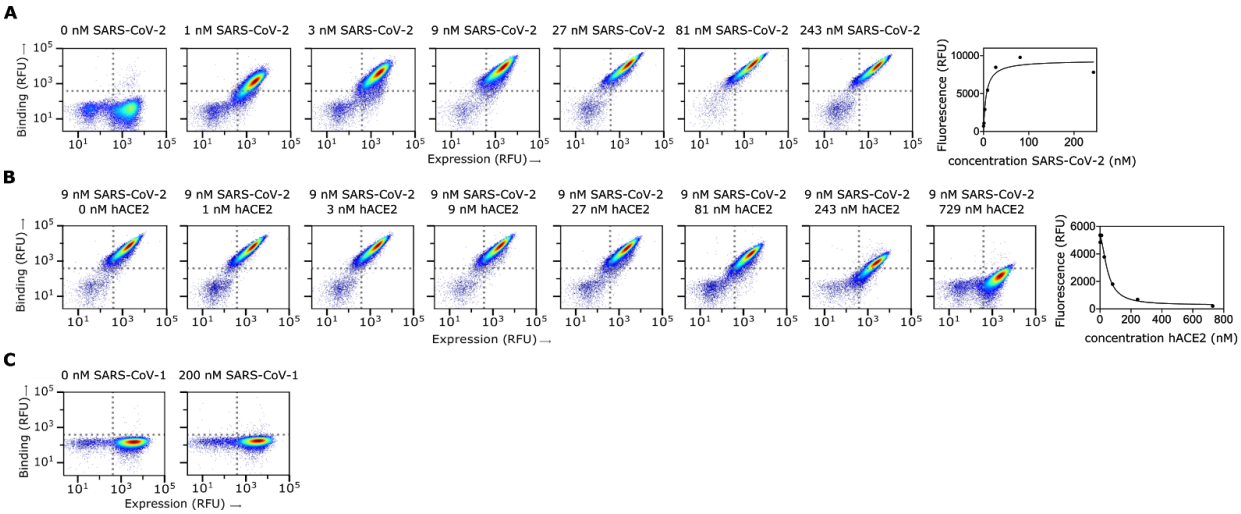


Fig. S3. Yeast display characterization of CTC-445 binding to SARS-CoV-2 RBD and SARS-CoV-1 RBD. A) Yeast-display titration assay to estimate the binding K_D of CTC-445 for SARS-CoV-2 RBD. Yeast displaying CTC-445 were incubated with increasing concentrations of SARS-CoV-2 RBD (0, 1, 3, 9, 27, 81 and 243 nM, see methods). On the right, steady state plot to reveal a K_D of 6 nM. **B)** Yeast-display competition assay to estimate the $IC_{50\ @\ hACE2[0.4nM]}$ of hACE2 to compete binding of CTC-445 to SARS-CoV-2 RBD. Yeast displaying CTC-445 were incubated with SARS-CoV-2 9 nM and increasing concentrations of hACE2 (0, 1, 3, 9, 27, 81, 243 and 729 nM, see methods). On the right, steady state plot to reveal an $IC_{50\ @\ hACE2[0.4nM]}$ = 49 nM. **C)** Screen for binding of CTC-445 to SARS-CoV-1 RBD. Yeast displaying CTC-445 were incubated with SARS-CoV-1 RBD at 200 nM (see methods), with no observable binding at this concentration.

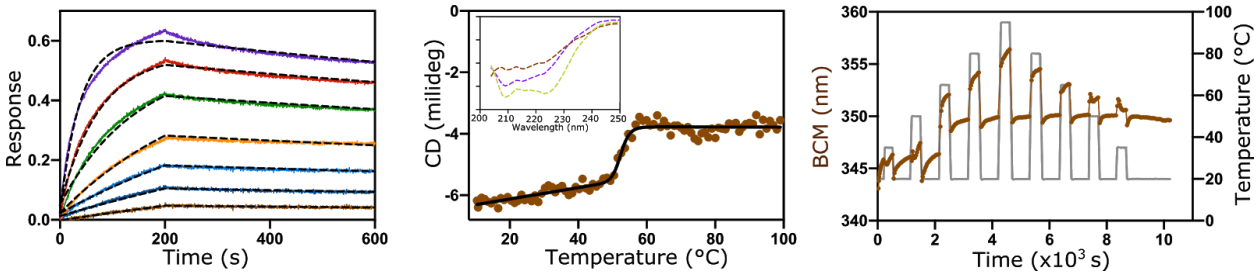


Fig. S4: Binding kinetics and thermal stability of hACE2. Binding kinetics (left) of hACE2 to immobilized SARS-CoV-2 RBD (300 nM to 4.7 nM); data were globally fit using a 1:1 binding model. Thermal-induced unfolding (center) measured by circular dichroism at 208 nm was fit using a two-states equation; a T_m value of 52.5 ± 0.5 °C was calculated. The inset shows far UV spectra at 20 °C, 99 °C and 20 °C after heating and cooling the sample. Thermal-recovery assay (right); BCM signal does not return to the initial value after repeated cycles of heating and cooling which indicates poor reversibility.

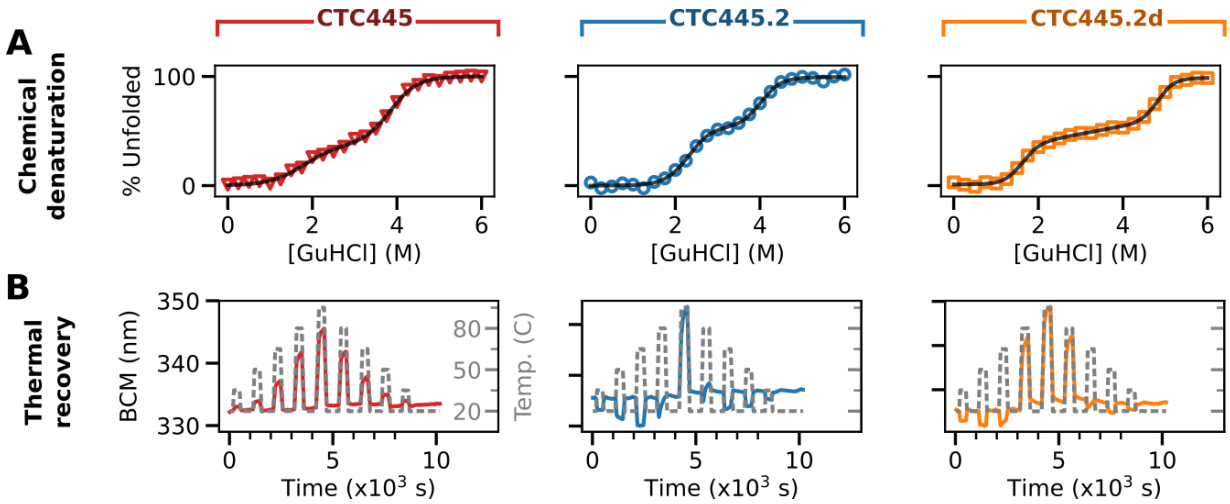


Fig. S5: Chemical and thermal stability of the de novo protein decoys. A) Chemical denaturation of CTC-445, CTC-445.2 and CTC-445.2d by incubation with guanidine hydrochloride (0-6 M). Calculated values of ΔG_{ND}, ΔG_{NI} and ΔG_{ID} were -9.8 ± 0.8, -2.7 ± 1.1 and -7.1 ± 0.7 kcal mol^{-1} for CTC-445, -14.6 ± 2.0, -5.0 ± 2.8 and -9.6 ± 2.0 kcal mol^{-1} for CTC-445.2, and -19.0 ± 1.7, -4.4 ± 2.3 and -14.6 ± 1.6 kcal mol^{-1} for CTC-445.2d; **B)** Thermal recovery of CTC-445, CTC-445.2 and CTC-445.2d. Similar values of BCM were observed before and after repeated cycles of heating and cooling for the three proteins.

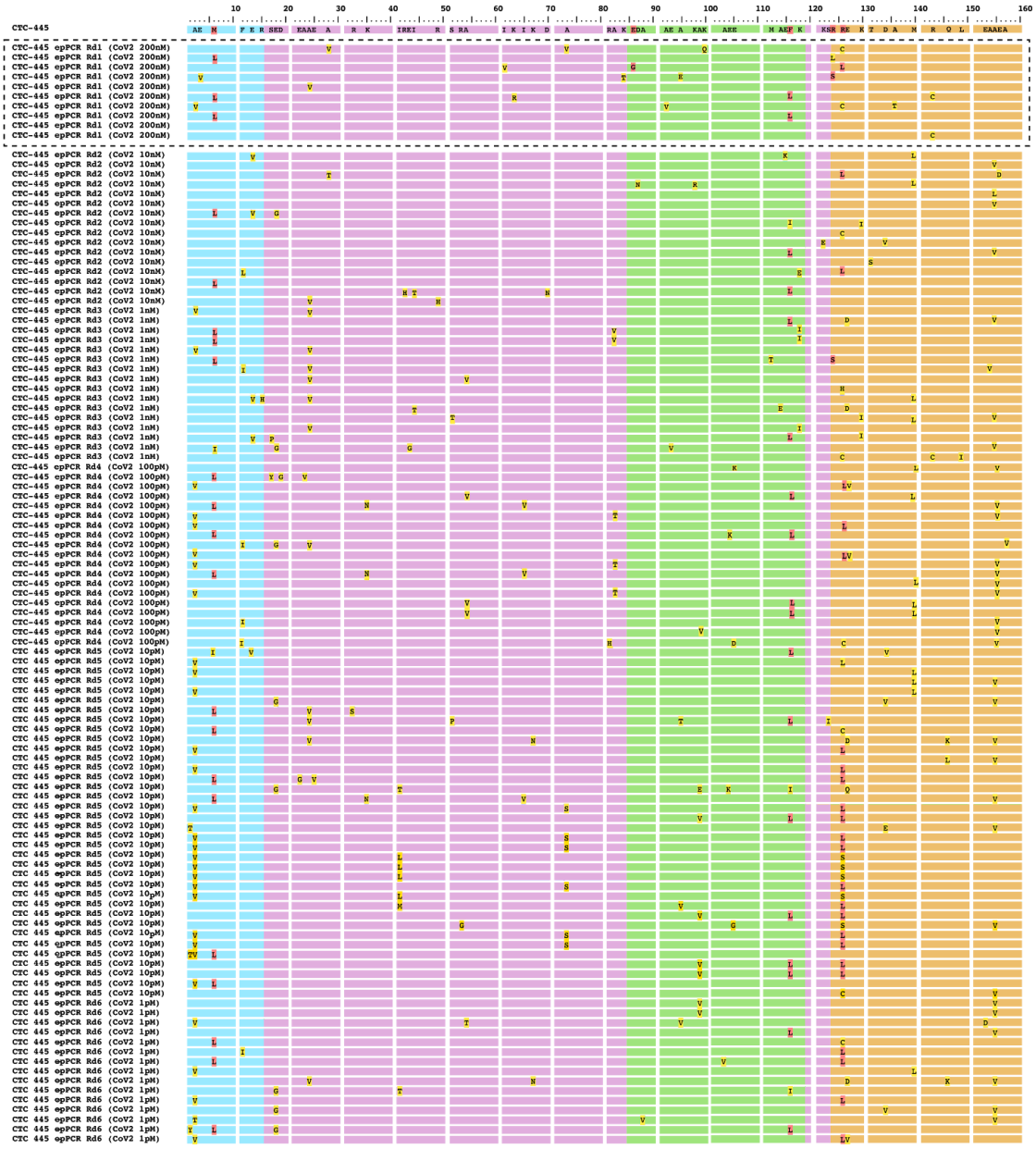


Fig. S6. Sequence alignment for the directed evolution of CTC-445 using error-prone PCR and yeast display FACS. Error Prone PCR was performed on the gene coding for CTC-445 to identify mutations that improve its stability and affinity to SARS-CoV-2. A DNA library of $2x10^6$ CTC-445 variants was generated and transformed into yeast for yeast display. Six rounds of yeast sorting were performed, with increasingly stringent binding conditions. For every round of cell sorting, the CTC-445 library was first treated with a mixture of trypsin and chymotrypsin (see methods), washed, and then allowed to bind to SARS-CoV-2 RBD at concentrations of: 200 nM for round 1, 10 nM for round 2, 1 nM for round 3, 100 pM for round 4, 10 pM for round 5 and 1 pM for round 6 (see methods). The top 5% binders for every round were sorted using fluorescence-activated cell sorting (FACS) and selected cells were plated on selective SDCAA plates for sequencing. Cartoon representations of the sequences are shown here with H1 in orange, H2 in green, EE3 in blue and supporting motifs in magenta. CTC-445 is shown on the top and positions where mutations were found are indicated with the residue letter. Ten sequences were picked in the first round of sorting, highlighted with a dashed box. Mutations found in round 1 in combination with rational design led to the CTC-445 second generation variants in Table S2, which includes CTC-445.2. CTC-445.2 is five mutations away from CTC-445 (M6L, E86G, F116L, R124S and R126L, highlighted in red). The more stringent rounds of sorting further confirmed mutations M6L, F116L and R126L and revealed new mutations (e.g. A2V and A155V) that are located in the core of the design and were shown to further improve stability (see Fig. S21).

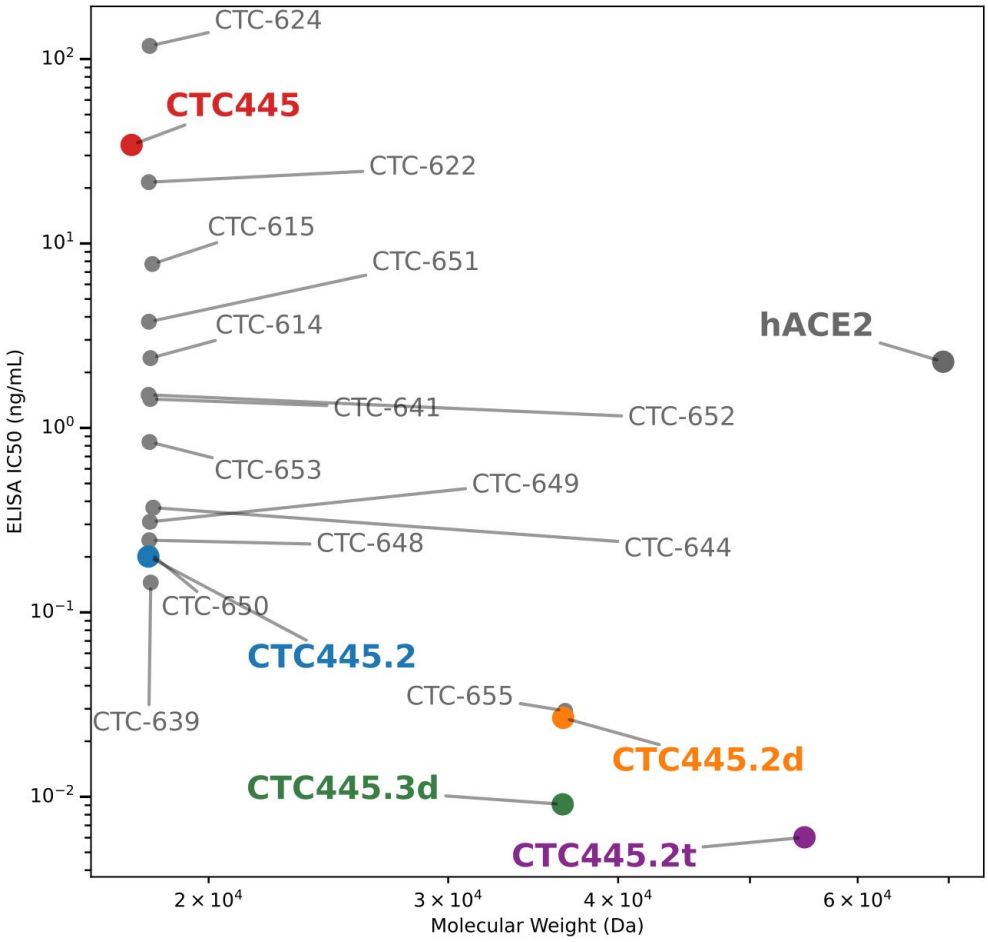


Fig. S7. Potency of CTC-445 variants vs. their molecular weights. $IC_{50\ @\ hACE2[0.4nM]}$ values for CTC-445 variants for binding to SARS-CoV-2 S/RBD were measured by ELISA in the presence of 0.4 nM biotinylated soluble hACE2. hACE2 bound to RBD was quantified by treatment with streptavidin-HRP and measurement of absorbance at 450 nm.

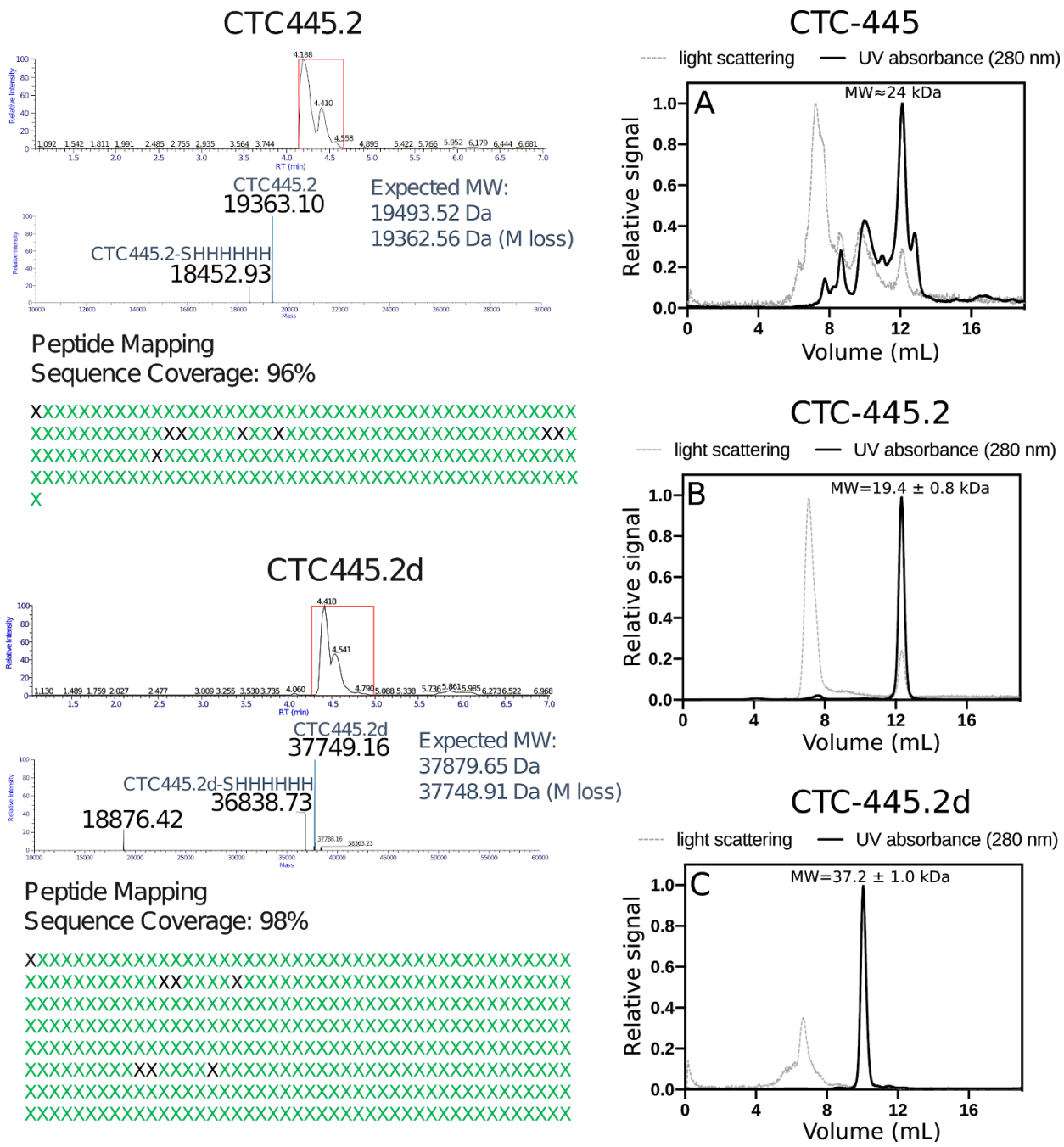


Fig. S8. Mass spectrometry and SEC-MALS analysis of de novo designed decoys CTC-445, CTC-445.2 and CTC-445.2d using a Superdex 75 Increase 10/300 GL size-exclusion column. Mass spectrometry and peptide mapping coverage for CTC-445.2 (upper left) and CTC-445.2d (lower left). Light scattering signal (gray, dashed) and 280nm UV absorption signal (black, continuous). Chromatogram traces are normalized to the peak with the highest intensity for clarity. Flow rate 0.5 mL/min, injection volume: 100 mL. After isolation of the monomeric CTC-445 (upper right) and CTC-445.2 (center right) chromatograms shown a small population of high molecular weight aggregates and oligomeric species with high light scattering; while CTC445.2d (lower right) shows evidence of a minuscule population of dimeric oligomer with small light scattering. In all cases chromatograms exhibit excellent monomer-oligomer separation and the light scattering signal exhibits high signal-to-noise ratio. The calculated molecular weights are shown for the fractions of the highest UV signal. For CTC445.2 and CTC445.2d the monomer MW values exhibit high homogeneity and the Mw/Mn ratio was 1.0 which indicates the proteins are monodisperse.

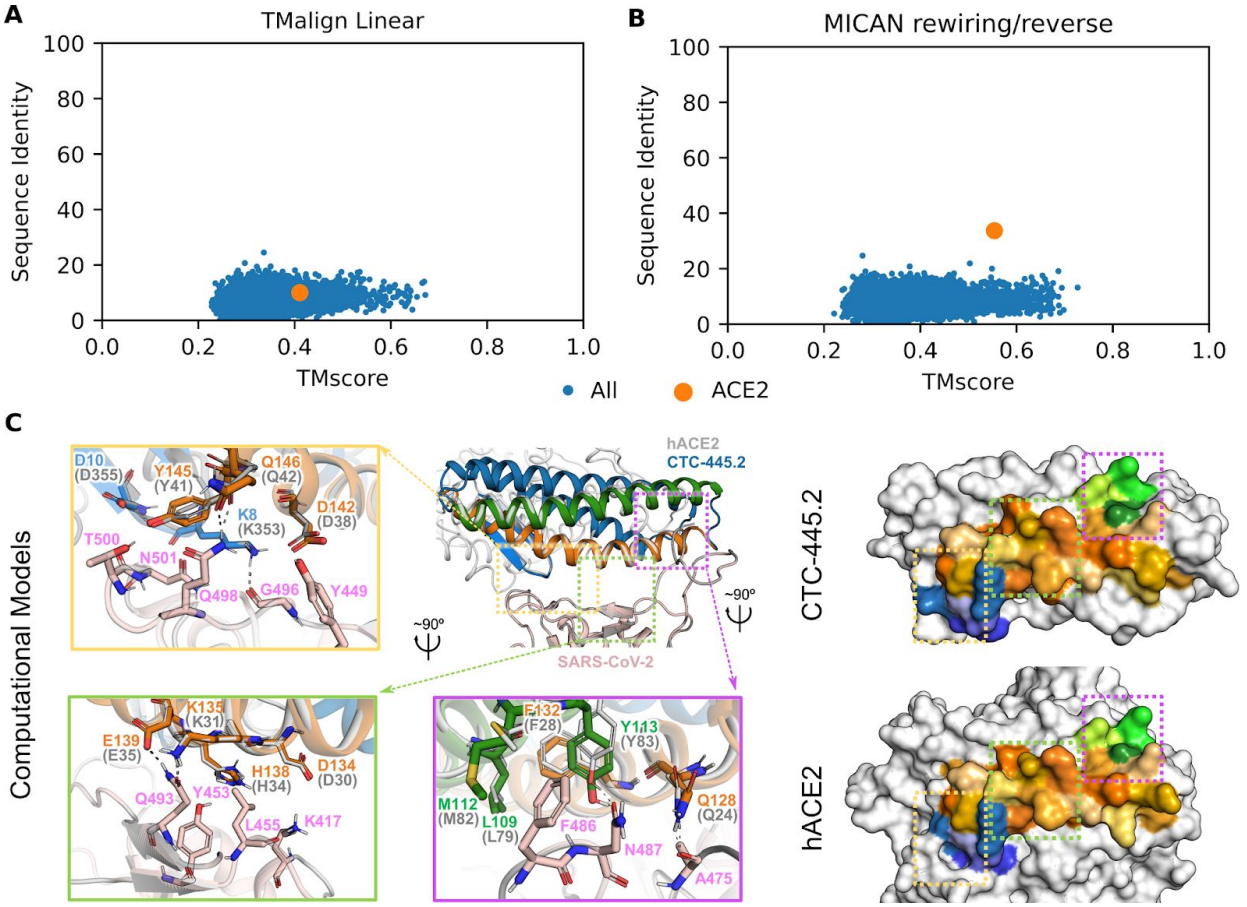


Fig. S9. Sequence and structural similarity of CTC-445.2 to natural proteins. A) Sequential (lineal) structural alignments, using TMalign, take into account the order of the secondary structure elements in the polypeptide chain. Each datapoint represents a structural alignment of CTC-445.2 to a single PDB from a non-redundant structure database of 14,433 PDB chains (see methods). Sequence identity is computed based on the structurally aligned residues. Although CTC-445.2 mimics the ACE2 binding surface, it does not align well in lineal structure or sequence to any protein in the database, including ACE2. **B)** MICAN in rewiring/reverse mode, which allows inverse direction of secondary structures, alternative alignments, and non-sequential alignments. The highest sequence identity observed for CTC-445.2 is with ACE2 (33.7% for aligned residues), which is approximately the fraction of solvent-exposed ACE2 residues from the starting RBD-binding motifs that were preserved (35%). **C)** Comparison of the SARS-CoV-2 RBD (beige) binding interface with the computational model of CTC-445.2 and the hACE2 interface (gray; PDB: 6M17). CTC-445.2 residues from each binding motif are colored: H1 (orange), H2 (green) and EE3 (blue). ACE2 residue numbers are shown in parentheses. Right, comparison of the SARS-CoV-2 binding interface surface of CTC-445.2 (top) and hACE2 (bottom). CTC-445.2 is designed to present the same binding interface as the one that SARS-CoV-2 targets in hACE2, down to the level of individual amino acid interactions.

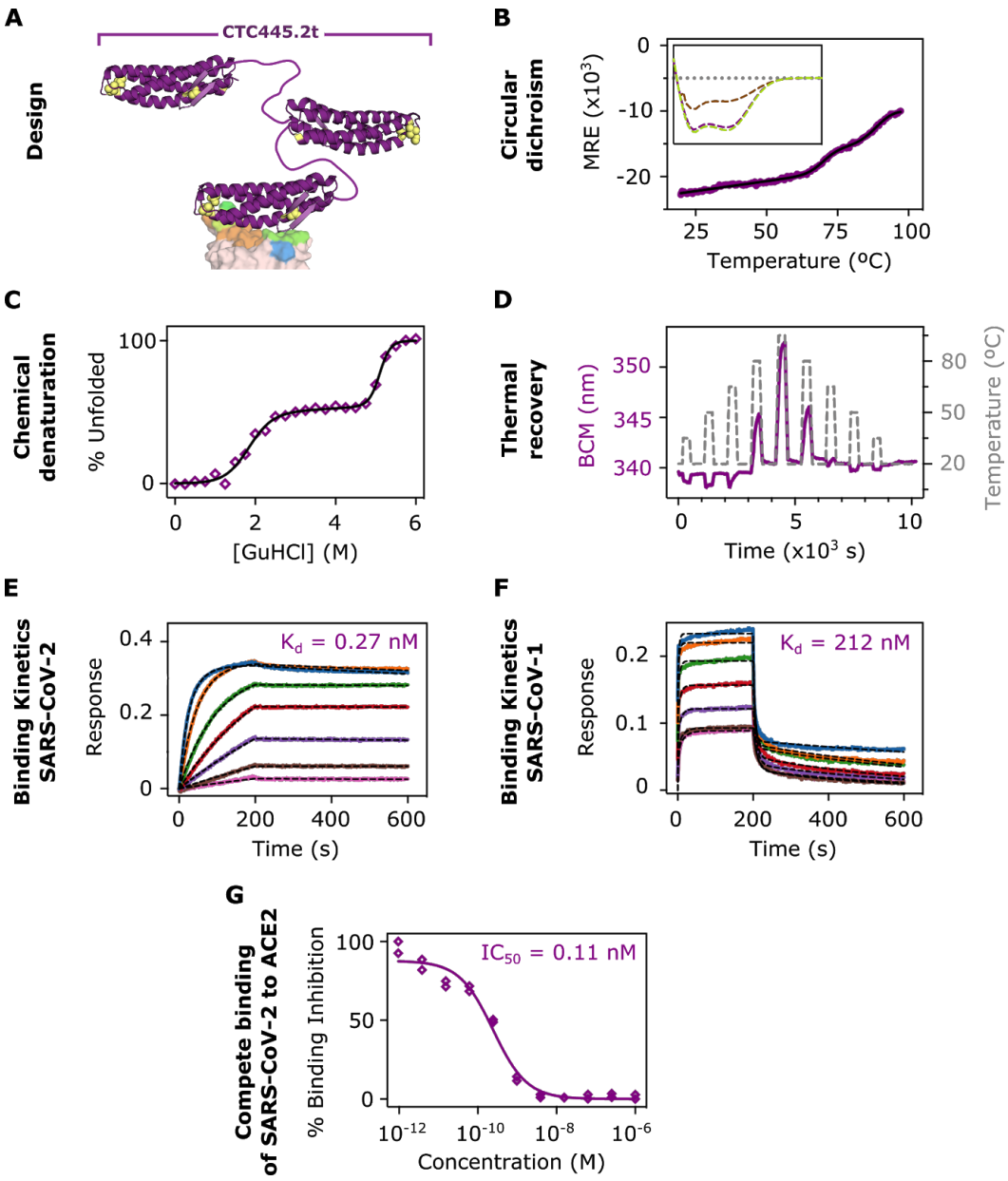
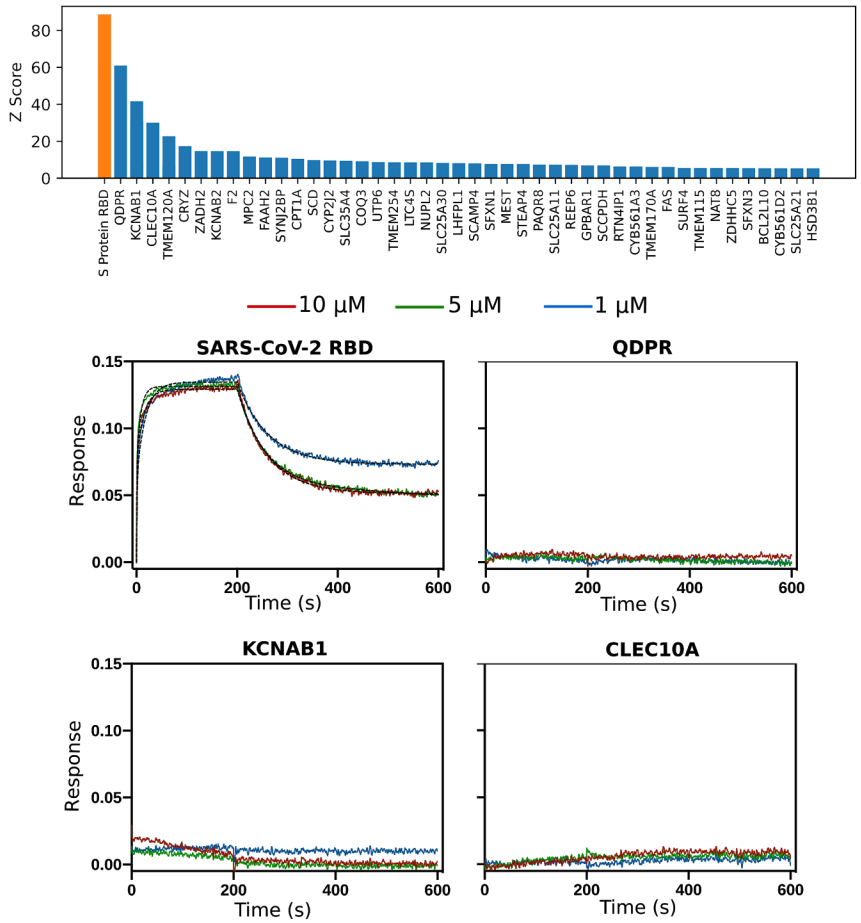


Fig. S10. Characterization of CTC-445.2t. CTC-445.2t is a tri-valent avid version of CTC-445 with further improved binding affinity and high thermodynamic stability. **A)** Design model of CTC-445.2t, a serial repeat of three CTC-445.2 molecules separated by two 16-residue linkers (sequence -GGGSGGSGSGGSGGGS-). **B)** Thermal denaturation of CTC-445.2t, followed by circular dichroism at 208 nm (20 °C to 99 °C at 2 °C/min). A three-state fit revealed two melting temperatures, $T_{m,1}$ = 71 ± 0.5 °C and $T_{m,2}$ = 90 ± 1.6 °C. The inset shows the far UV spectra at 20 °C (purple), after heating to 99 °C (brown) and after cooling to 20 °C again (green), to show complete recovery of the ellipticity-spectra. **C)** Chemical denaturation using guanidine hydrochloride (0-6 M). Calculated values of ΔG_{ND}, ΔG_{NI}, and ΔG_{ID} were -26.5 ± 5, 3.7 ± 6 and -22.8 ± 4 kcal mol^{-1}. **D)** Thermal recovery was observed after cycles of heating and cooling, as CTC-445.2t went back to similar BCM values after each cycle. **E)** Binding kinetics to immobilized SARS-CoV-2 RBD measured using biolayer interferometry (CTC-445.2t concentrations of 0.5, 0.9, 1.9, 3.8, 7.5, 15 and 30 nM). Results were fit using a 2:1 binding model to reveal a K_D = 0.27 ± 0.07 nM. **F)** Binding kinetics to immobilized SARS-CoV-1 RBD measured using biolayer interferometry (CTC-445.2t concentrations of 78.1, 156.2, 312.5, 625, 1250, 2500 and 5000 nM). Results were fit using a 2:1 binding model to reveal a K_D = 212 ± 54 nM. **G)** Potency of CTC-445.2t to outcompete binding of SARS-CoV-2 to ACE2, measured by ELISA with a constant concentration of 0.4 nM ACE2 and a range of CTC-445.2t concentrations from 0.001 to 1000 nM, revealed an $IC_{50\ @\ hACE2[0.4nM]}$ = 0.11 nM.



Rank	Protein	Fluorescence	Z Score	S Score
1	SARS-CoV-2 RBD	65535	88.6	27.6
2	QDPR	45252	61.0	19.3
3	KCNAB1	31053.5	41.7	10.6
4	KCNAB1	23248.5	31.0	1.0
5	CLEC10A	22530	30.1	7.3
6	TMEM120A	17165	22.8	5.4
7	CRYZ	13218.5	17.4	2.7
8	ZADH2	11256.5	14.7	0.0
9	KCNAB2	11221.5	14.7	0.0
10	F2	11214.5	14.6	2.9

Fig. S11. Binding specificity of CTC445.2d assessed using a comprehensive human proteome binding assay. A) The binding profile of CTC445.2d (concentration of 1 µg/mL (30 nM), chemically labeled with Alexa-647, see methods) to a complete array of 21,218 (17,164 unique) human proteins (HuProt human proteome array, CDI Labs, see methods). Binding to each individual protein is quantified by a Z score based on the mean fluorescence of each spot in the array. If the S score (Z score difference) of the top hit is > 3.0 from the next hit, the test protein is considered highly specific against the top hit. The table shows all hits with Z score > 10. CTC445.2d demonstrates specific binding to the SARS-CoV-2 RBD, with a Fluorescence value of 65,535 translating to the highest Z Score of 88.6. The S score to the next highest binding gene, QDPR, is 27.6, demonstrating highly specific binding to the intended target RBD of SARS-CoV-2. **B)** The binding of CTC-445.2d to immobilized SARS-CoV-2 RBD-his, QDPR-his and CLEC10A-his was tested by biolayer interferometry. No binding to QDPR and CLEC10A was detected at any tested concentrations of CTC-445.2d (10 µM, 5 µM and 1 µM).

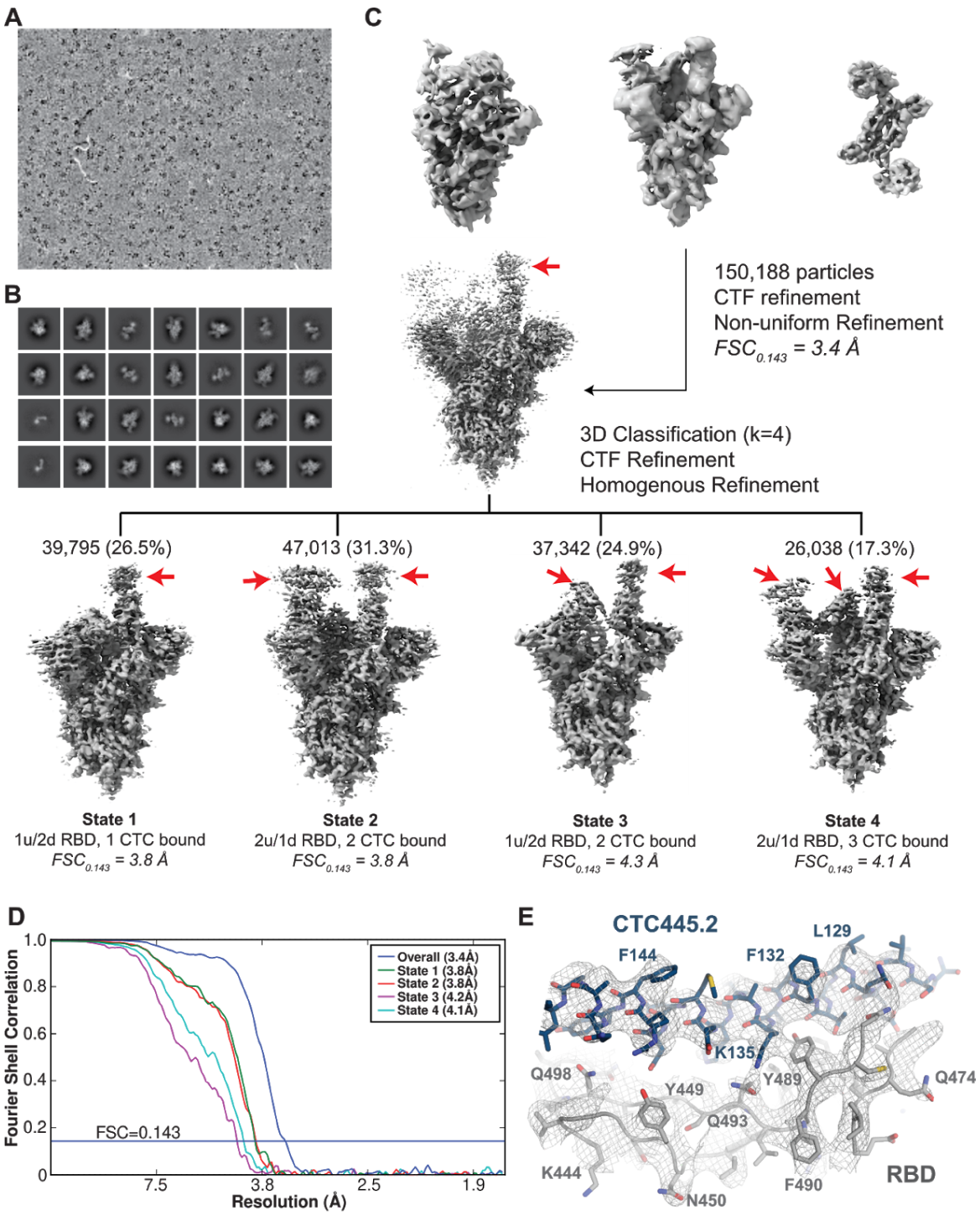


Fig. S12. cryoEM data processing and validation of the CTC-445.2-S complex. A) Representative micrograph and **B)** 2D class averages. **C)** 3D classification and refinement workflow. After homogenous refinement of particles corresponding to the CTC-445.2-S trimer ab initio model, particles were classified into four distinct classes and subjected to iterative rounds of CTF refinement, homogenous refinement, and local refinement. **D)** Final fourier shell correlation (FSC) plots calculated from half-maps of masked data for each state. Blue line represents FSC=0.143. **E)** Representative density from focused refinement of CTC-445.2-RBD interface. While the overall map resolution was 4.1Å (FSC=0.143), discernable side chain features allowed for accurate modeling of CTC-445.2 (blue).

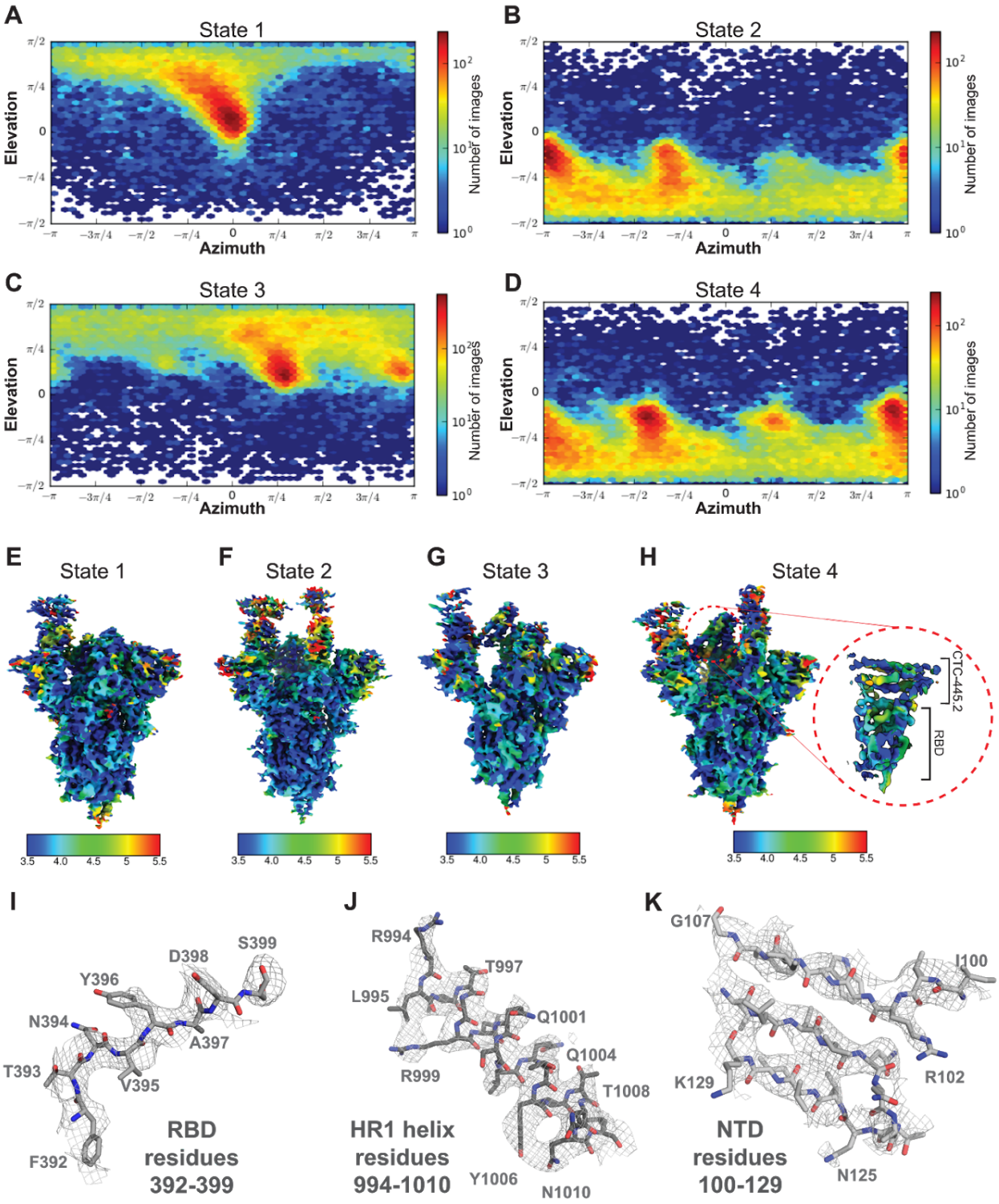


Fig. S13. Assessment of cryoEM map quality. A-D) Euler angle distribution plots, and **E-H)** local resolution estimations for reconstructions corresponding to States 1-4. The local resolution estimate for the focused classification and local refinement of CTC-445.2 bound to a "partially-down" RBD is highlighted. **I-K)** Representative density from SARS-CoV-2 S trimer regions of the State 4 map in which coordinates were modeled (map contour level = 3σ).

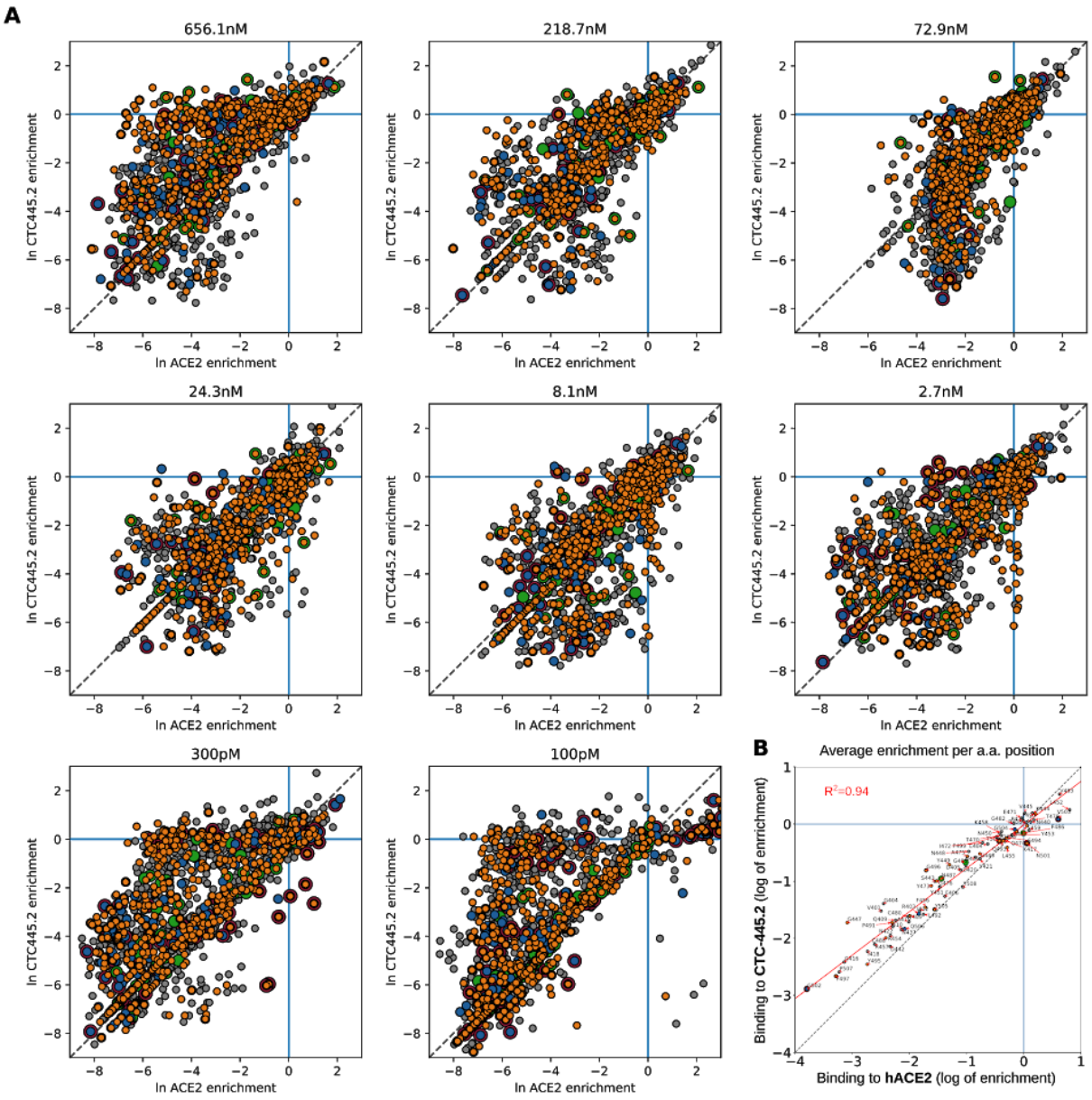


Fig. S14. Deep mutational scanning of the SARS-CoV-2 RBD binding interface, effect on binding (by yeast display) of CTC-445.2 and hACE2 at 8 different concentrations. A) The upper left plot shows the correlation between the observed counts of each mutant in the initially constructed site-saturation mutagenesis library (SSM) and the observed counts in the library after sorting for successful yeast display (see methods). The remaining 8 plots show the enrichments of RBD mutations after selection for binding to CTC-445.2 vs. ACE2. Each data point is a single amino acid mutation at a given position of the RBD. Yeast display FACS for binding were performed at concentrations of: 656.1, 218.7, 72.9, 24.3, 8.1, 2.7, 0.3, 0.1 nM of CTC-445.2 or hACE2 (see methods). The dotted line is the trace of a theoretical perfect correlation. The SSM library was designed to include all the 19 possible natural-amino acid mutations for the 75 residues composing the RBD binding interface with hACE2 and it's second layer of interaction. At the lowest concentration (100 pM), five outlier mutations were found to increase RBD binding affinity to hACE2 but decrease binding affinity to CTC445.2. **B)** Enrichments averaged across all 8 selection conditions as in Fig. 4B and also averaged per residue. For all the scatter plots, the coloration indicates the secondary structure(s) interacting with the amino acid position (same color scheme in A and B).

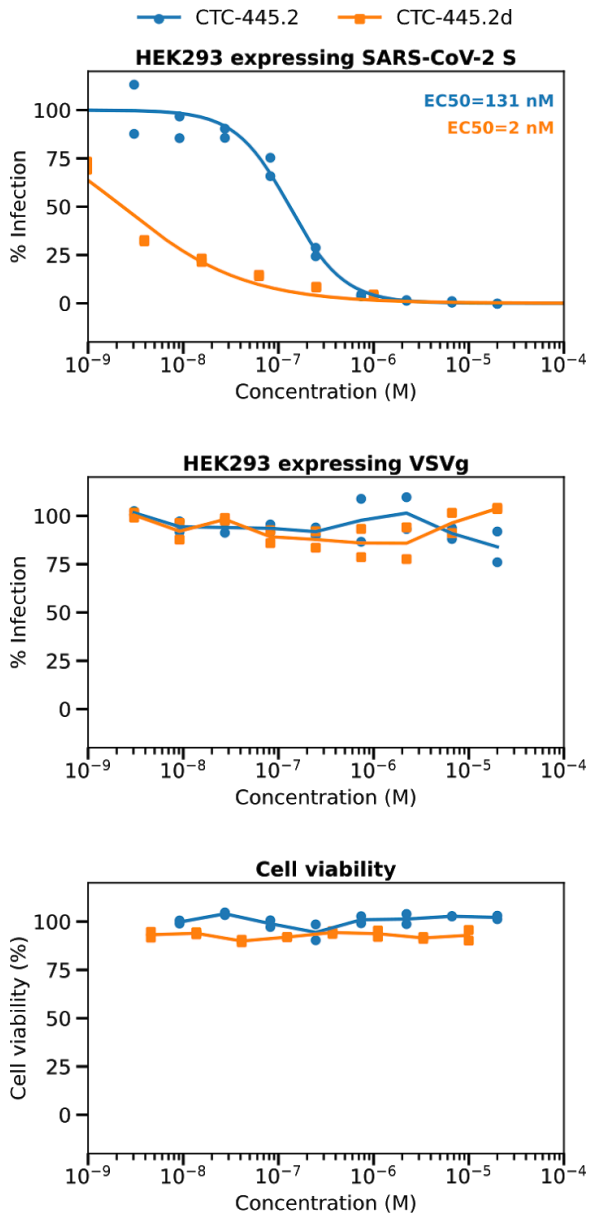


Fig. S15. In vitro neutralization of VSV-Luc pseudovirus in ACE2-transfected HEK293T cells. CTC-445.2 and CTC-445.2d neutralized VSV pseudovirus expressing SARS-CoV-2 (top). We observed no inhibition of pseudovirus infection in HEK293T cells expressing VSVg with CTC-445.2 and CTC-445.2d (center), demonstrating specificity for the RBD of SARS-CoV-2, with no discernable loss of HEK293 cell viability (bottom). All data points are shown (n=2).

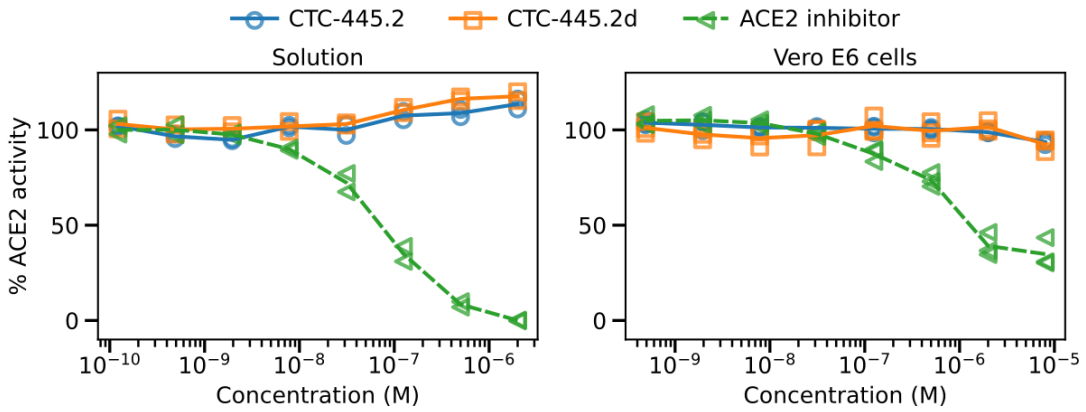


Fig. S16. ACE2 functional activity in the presence of CTC-445.2 and CTC-445.2d. ACE2 functional activity was measured by enzymatic release of a free fluorophore fromMca-APK(DNP) substrate in solution (left) and in Vero E6 cells (right). The de novo decoys did not affect ACE2 function. DX600 was used as positive control (see methods).

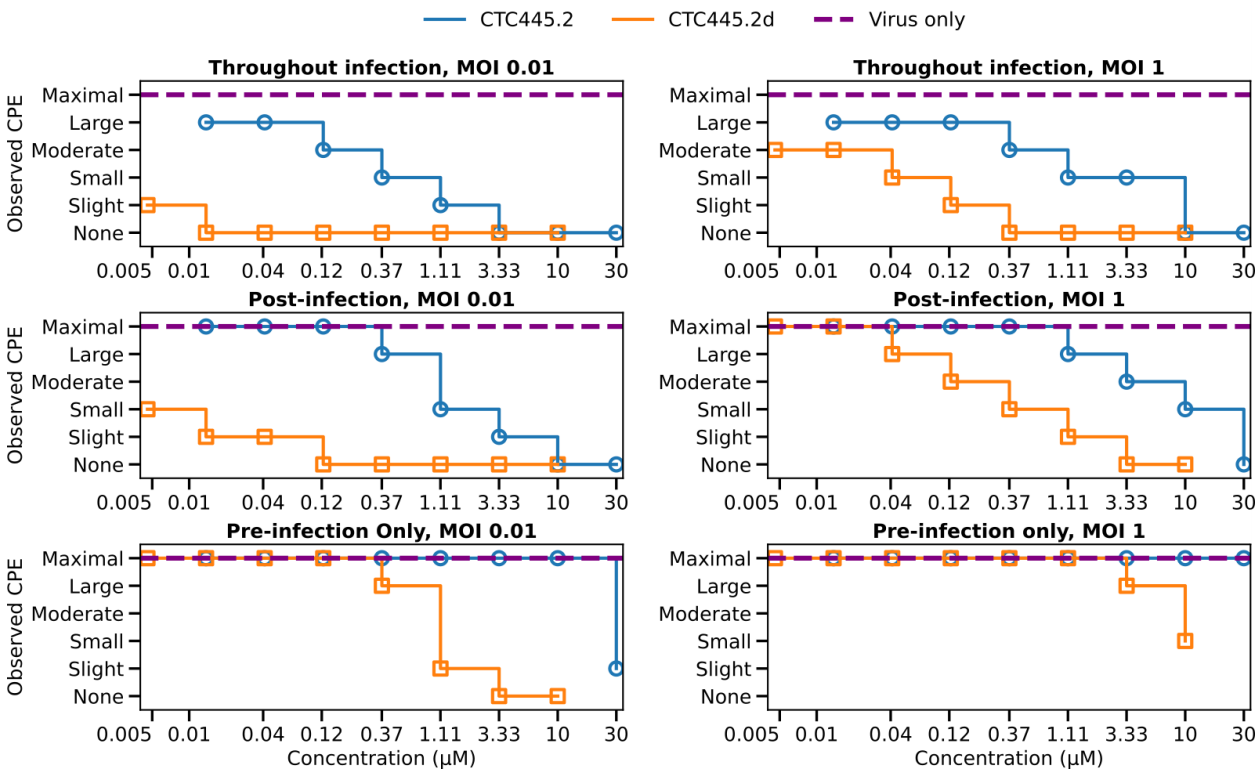


Fig. S17. Inhibition of SARS-CoV-2 (cytopathic effect) in Vero E6 cells. Observed qualitative cytopathic effect of SARS-CoV-2 infection of Vero E6 cells at 72 hours post infection, where CTC-445.2 or CTC-445.2d is present throughout infection, post-infection only and pre-infection only.

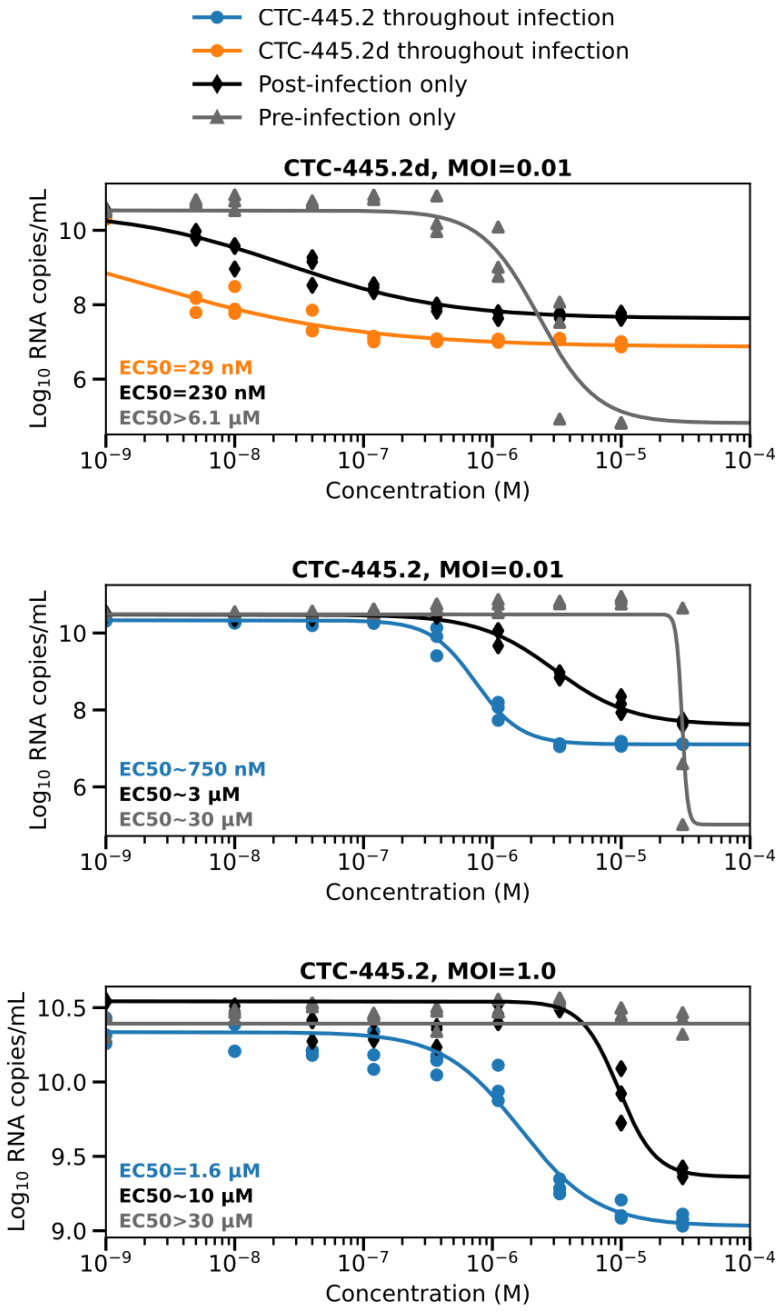


Fig. S18: In vitro neutralization of live BetaCoV/Hong Kong/VM20001061/2020 SARS-CoV-2 virus in Vero E6 cells. The cells were incubated with CTC-445.2d at MOI 0.01 (left) and CTC-445.2 at MOI=0.01 (center) and MOI=1.0 (right) throughout infection (i.e. before infection, during infection and post infection) (orange/blue), post-infection only (black) and pre-infection only (gray). SARS-CoV-2 RNA copy numbers were determined by quantitative real-time RT-PCR. All assays were performed in triplicate unless otherwise noted, and all data points are shown.

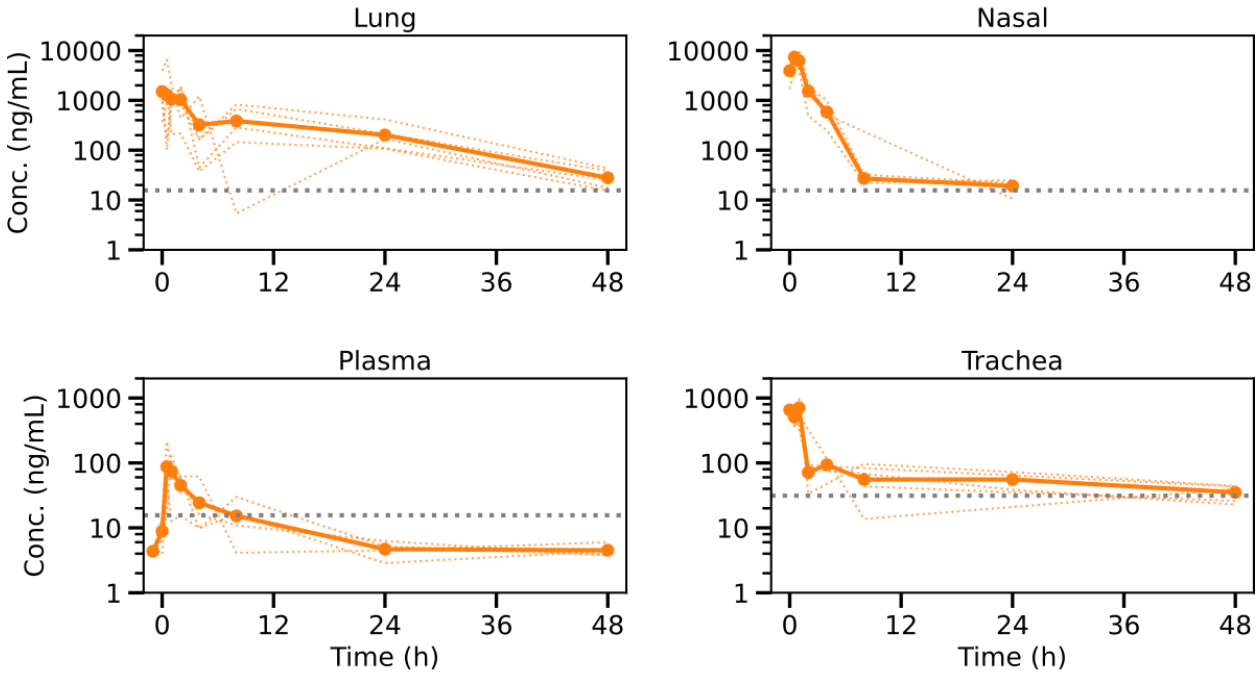


Fig. S19. In vivo mouse pharmacokinetics of intranasally administered CTC-445.2d. Concentration of fully functional CTC-445.2d (i.e. capable of binding to the SARS-CoV-2 RBD, see methods) found in homogenized lungs, nasal turbinates, plasma, and trachea of Balb/c mice after a single 100 µg dose, measured at indicated times after dosing (n=5 mice for plasma, lung, and trachea, n=3 mice for nasal turbinates). The dotted gray line indicates the quantification limit.

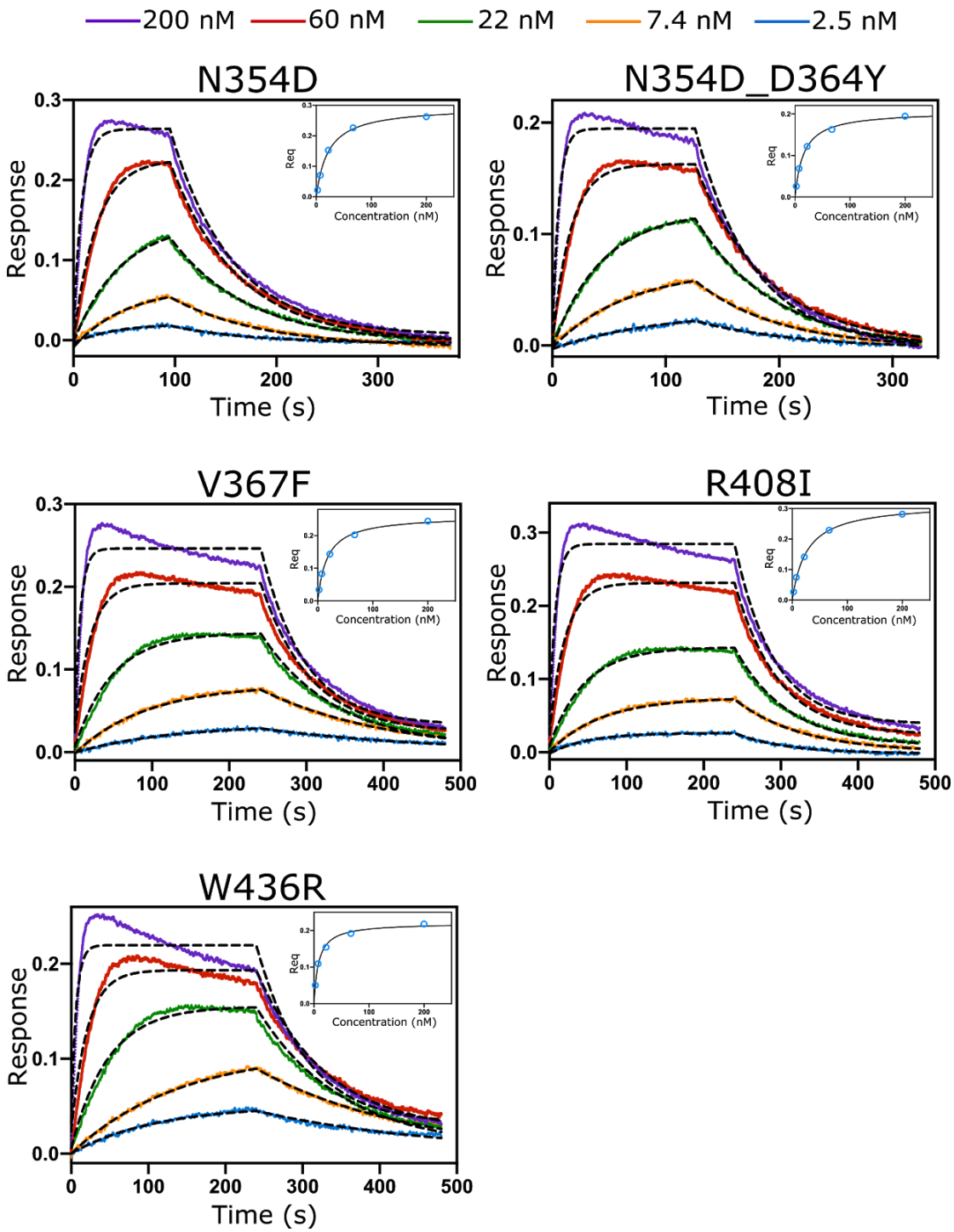


Fig. S20: Kinetics of binding for CTC-445.2 to SARS-CoV-2 RBD mutants. Five RBD mutants were immobilized via Anti-Penta-His sensors, and CTC-445.2 was titrated in solution. Data was globally fit to a 1:1 model for each RBD mutant. The inset shows the steady state plot.

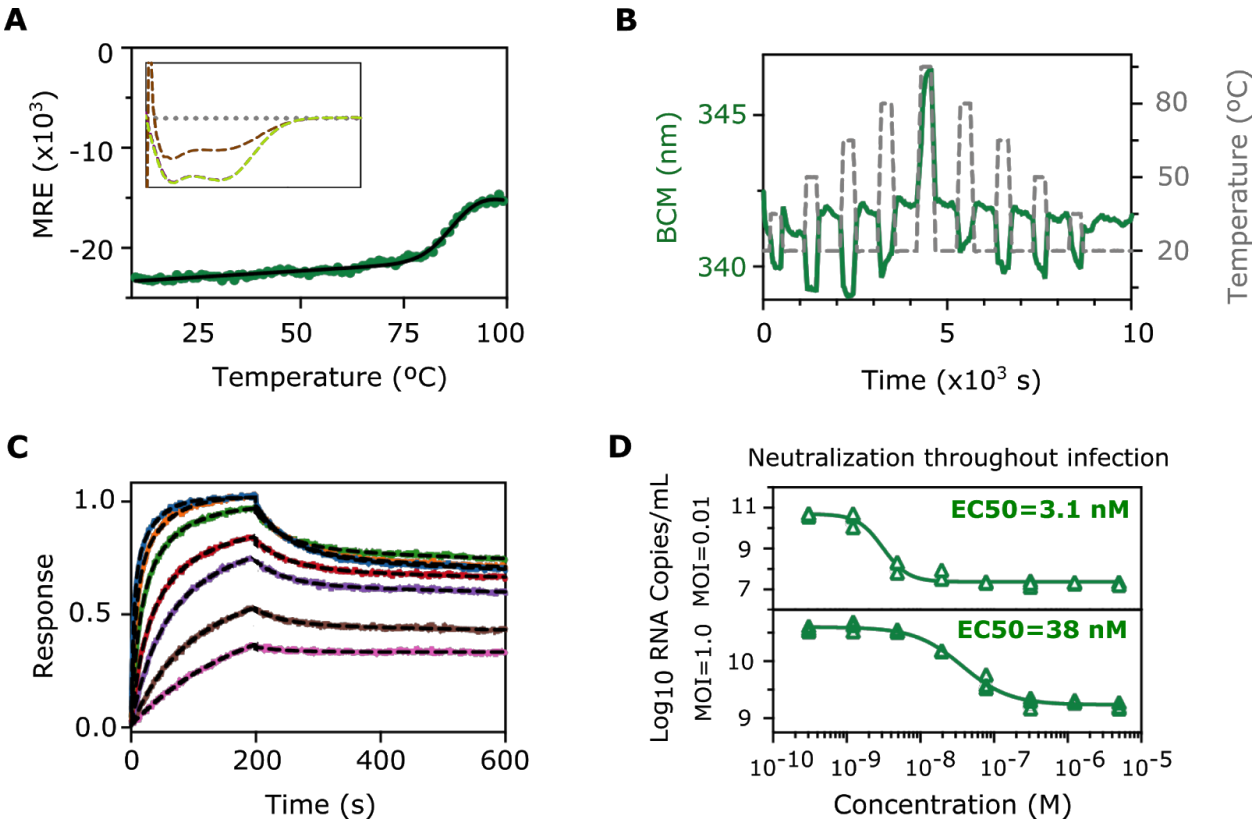


Fig. S21. CTC-445.3d, a further optimized CTC-445.2d variant. Stability, thermal kinetics, binding and in vitro neutralization. CTC-445.3d is a further stability-enhanced variant. Similarly to CTC-445.2d, CTC-445.3d is a bivalent molecule that contains three additional (i.e. compared to CTC-445.2) amino acid substitutions (A2V, I41L and R142L) in each of the (two) functional domains. **A)** Thermal denaturation of CTC445.3d was followed by circular dichroism at 208 nm, from 20 °C to 99 °C at 2°C/min. A two-state fit revealed a Tm of 89 ± 2 °C, which is 17 °C higher than the Tm of CTC-445.2d. The inset shows the far UV spectra at 20 °C (purple), 99 °C (brown) and 20 °C (green), to show recovery after heating and cooling. **B)** Heating cycles (20, 35, 50, 65, 80, 95 °C, and vice versa down) revealed that CTC-445.3d only loses its secondary structure at the highest heating cycle (95 °C) and goes back to similar BCM values after each temperature cycle. **C)** Binding kinetics of CTC-445.3d to immobilized SARS-CoV-2 RBD (300 nM to 4.7 nM) using octet. Results were fit using a 2:1 binding model to reveal a K_D= 1 nM, k_{on}= 4.2 x 10^5 M^{-1} s^{-1}, and k_{off}= 1.4 x 10^{-2} s^{-1}. **D)** In vitro neutralization of the live BetaCoV/HongKong/VM20001061/2020 SARS-CoV-2 virus by CTC-445.3d in Vero E6 cells, with the protein present throughout the infection and using two different multiplicity of infections (MOIs) (0.01 and 1.0), three replicates each. Determination of the SARS-CoV-2 RNA copy numbers revealed an EC50 of 3.1 nM at MOI 0.01 and an EC50 of 38 nM at MOI 1.0.

Table S1. De novo designed decoys, sequences and kinetics of binding to the SARS-CoV-2 S protein RBD. The RBD (either for SARS-CoV-1 or SARS-CoV-2) was immobilized to the OCTET sensor (Streptavidin SA for SARS-CoV-1 and Anti-Penta-HIS HIS1K SARS-CoV2, ForteBio) and incubated with varying concentrations of ACE2 decoys. Binding kinetics were monitored by dipping the biosensors in wells containing defined concentrations of the de novo decoys (5 nM, 19 nM, 38 nM, 75 nM, 150 nM and 300 nM for binding to SARS-CoV-2 and 39 nM, 78 nM, 156 nM, 312 nM, 625 nM, 1250 nM and 2500 nM for binding to SARS-CoV-1) and then dipping the sensors back into baseline wells (dissociation). Both, steady state and fitted kinetic values are shown.

Decoy	RBD	k_{on} (M^{-1} s^{-1})	k_{off} (s^{-1})	K_D kinetics (nM)	K_D steady-state (nM)
hACE2	SARS-CoV-1	6.8×10^{4}	1.0×10^{-3}	14	43
	SARS-CoV-2	8.6×10^{4}	4×10^{-4}	4.7	31
CTC-445	SAEIDMGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAKDEDAA RNAENAARKAKEFAEEQAKLADMYAEFAKNGDKSRVREQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAG				
	SARS-CoV-1	~ 4.5×10^{4}	~2.5×10^{-1}	~ 5.5×10^{4}	ND
	SARS-CoV-2	8.6×10^{4}	5.5×10^{-2}	646	357
CTC-445.2	SAEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAKDGDAAR NAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAGGSGWGSG				
	SARS-CoV-1	1.0×10^{5}	3.9×10^{-1}	3.8×10^{3}	~ 7×10^{3}
	SARS-CoV-2	3.9×10^{5}	1.2×10^{-2}	30	21
	SARS-CoV-2_N354D	5.0×10^{5}	1.5×10^{-2}	30	22
	SARS-CoV-2_N354D_D364Y	7.4×10^{5}	1.5×10^{-2}	20	16
	SARS-CoV-2_V367F	5.9×10^{5}	1.3×10^{-2}	22	17
	SARS-CoV-2_R408I	5.1×10^{5}	1.7×10^{-2}	33	26
	SARS-CoV-2_W436R	6.8×10^{5}	8.7×10^{-2}	13	8
CTC-445.2d	SAEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAKDGDAAR NAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAGGGGSGGSGSGGSG GGSPGSAEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAK DGDAARNAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAGGSGWGS				
	SARS-CoV-1	3.7×10^{5}	4.7×10^{-1}	1.2×10^{3}	587
	SARS-CoV-2	3.0×10^{5}	≤ 2.0×10^{-3}	≤ 7.0	3.5
CTC-445.2t	SAEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAKDGDAAR NAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAGGGGSGGSGSGGSG GGSPGSAEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAK DGDAARNAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAGGGGSGGSG SGGSGGGSPGSAEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEIIREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIA ARAAKDGDAARNAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDRFYQAALAVFEAAEAAAGGSG WGSG				
	SARS-CoV-1	5.4×10^{5}	0.1	188	212
	SARS-CoV-2	1.4×10^{6}	≤ 9.1×10^{-5}	≤ 0.065	0.27
CTC-445.3d	SVEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEILREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAKDGDAA RNAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDLFYQAALAVFEAAEAAAGGGGSGGSGSGGSG GGSPGSVEIDLGKGDFREIRASEDAREAAEALAEAARAMKEALEILREIAEKLRDSSRASEAAKRIAKAIRKAADAIAEAAKIAARAAK DGDAARNAENAARKAKEFAEEQAKLADMYAELAKNGDKSSVLEQLKTFADKAFHEMEDLFYQAALAVFEAAEAAAGGSGWGSG				
	SARS-CoV-1	ND	ND	ND	ND
	SARS-CoV-2	4.2×10^{5}	1.4×10^{-2}	3	1

Table S2. Characterization of the expression, solubility and binding of CTC-445 variants. CTC-445 variants were generated by rationally combining mutations discovered by directed evolution and characterized. Columns indicate the qualitative expression level (0-5), solubility (0-5), and binding affinity (0-5). Propensity for aggregation indicates whether large molecular weight aggregates or oligomers were present during size exclusion chromatography, and the fraction of non-monomeric protein is measured by comparing peak areas for monomeric and oligomeric species by analytical size exclusion chromatography.

Design ID	Description	Expression level (Cat. 0-3)	Solubility (Cat. 0-3)	Binding strength (Cat. 0-5)	Propensity for aggregation	SEC monomeric fraction (%)
CTC-00445	CTC-445					
CTC-00613	CTC-445_S1P	2	1	1		
CTC-00614	CTC-445_M6L	3	2	3		
CTC-00615	CTC-445_F152W	3	2	1		
CTC-00616	CTC-445_F11Y	3	2	1		
CTC-00617	CTC-445_F11K	2	3	2		
CTC-00618	CTC-445_R42S	2	1	1		
CTC-00619	CTC-445_E46S	2	2	1		
CTC-00620	CTC-445_A88N	3	2	N.D.		
CTC-00621	CTC-445_F116N	2	3	1		
CTC-00622	CTC-445_R124S	3	3	2		
CTC-00623	CTC-445_F137V	3	3	N.D.		
CTC-00624	CTC-445_R143L	2	3	3		
CTC-00625	CTC-445_S1P_F11Y_R42S_E46S_A88N_F116N_R124S_F137V_R143L	3	3	N.D.		
CTC-00626	CTC-445_S1P_M6L_F11Y_R42S_E46S_A88N_F116N_R124S_F137V_R143L_F152W	2	3	0.5		
CTC-00627	CTC-445_S1P_F11K_R42S_E46S_A88N_F116N_R124S_F137V_R143L	2	3	N.D.		
CTC-00628	CTC-445_S1P_M6L_F11K_R42S_E46S_A88N_F116N_R124S_F137V_R143L_F152W	2	3	2		
CTC-00629	CTC-445_E74C	3	2	1		
CTC-00630	CTC-445_K63C	2	1	Not screened		
CTC-00631	CTC-445_F11C	2	1	N.D.		
CTC-00632	CTC-445_E74C_F11Y	2	3	1	0	17%
CTC-00633	CTC-445_K63C_F11Y	2	1	1	1	
CTC-00634	CTC-445_E74C_S1P_M6L_F11Y_R42S_E46S_A88N_F116N_R124S_F137V_R143L_F152W	2	3	N.D.	0	82%
CTC-00635	CTC-445_K63C_S1P_M6L_F11Y_R42S_E46S_A88N_F116N_R124S_F137V_R143L_F152W	2	3	N.D.	0	86%
CTC-00636	CTC-445_S1P_M6L_F11C_R42S_E46S_A88N_F116N_R124S_F137V_R143L_F152W	0	1	N.D.	1	

CTC-00637	{PG}.CTC-445_M6L_R126L	1.5	2	4	0	97%
CTC-00638	{PG}.CTC-445_F116L_R124S	0.5	3	Not screened	1	
CTC-00639	{PG}.CTC-445_M6L_F116L_R124S_R126L	2	2	4.5	0	92%
CTC-00640 (CTC-445.2)	{PG}.CTC-445_M6L_E86G_F116L_R124S_R126L	2	3	4.5	0	99%
CTC-00641	{PG}.CTC-445_A2V_E3V_M6L_I61V_K63R_A73V_K84T_E86G_A92V_A95E_K100Q_F116L_R124S_R126L_A136T_R143S	2	3	3.5	0	95%
CTC-00642	{PG}.CTC-445_M6L_F11C_R126L	3	2	4.5	1	
CTC-00643	{PG}.CTC-445_M6L_K63C_R126L	3	2	4.5	0	59%
CTC-00644	{PG}.CTC-445_M6L_E74C_R126L	3	2	4.5	0	89%
CTC-00645	{PG}.CTC-445_F11C_F116L_R124S	3	2	Not screened		
CTC-00646	{PG}.CTC-445_K63C_F116L_R124S	3	2	3.5	1	
CTC-00647	{PG}.CTC-445_E74C_F116L_R124S	3	2	3	1	
CTC-00648	{PG}.CTC-445_M6L_F11C_F116L_R124S_R126L	2	3	5	0	85%
CTC-00649	{PG}.CTC-445_M6L_K63C_F116L_R124S_R126L	2	3	4.5	0	87%
CTC-00650	{PG}.CTC-445_M6L_E74C_F116L_R124S_R126L	0.5	3	4.5	0	90%
CTC-00651	{PG}.CTC-445_A2V_E3V_M6L_F11C_I61V_K63R_A73V_K84T_E86G_A92V_A95E_K100Q_F116L_R124S_R126L_A136T_R143S	2	3	4	0	97%
CTC-00652	{PG}.CTC-445_A2V_E3V_M6L_I61V_K63R_K63C_A73V_K84T_E86G_A92V_A95E_K100Q_F116L_R124S_R126L_A136T_R143S	2	3	4	0	89%
CTC-00653	{PG}.CTC-445_A2V_E3V_M6L_I61V_K63R_A73V_E74C_K84T_E86G_A92V_A95E_K100Q_F116L_R124S_R126L_A136T_R143S	2	3	4	0	86%
CTC-00654 **(CTC-445.2d)**	{PG}.CTC-445_M6L_E86G_F116L_R124S_R126L_X2	2	3	5		
CTC-00655	{PG}.CTC-445_A2V_E3V_M6L_I61V_K63R_A73V_K84T_E86G_A92V_A95E_K100Q_F116L_R124S_R126L_A136T_R143S_X2	2	3	5		

Table S3. SARS-CoV-2 neutralization by CTC-445.2 and CTC-445.2d in a time-of-addition experiment. *Recombinant protein was present before (pre-incubating with virus for 1h), during (1 h) and post SARS-CoV-2 infection (72 h) in Vero E6 cells. #Recombinant protein was added post SARS-CoV-2 infection. @Recombinant protein was pre-incubated with virus for 1 h and the virus-recombinant protein mix was added to Vero E6 cells and incubated for 1 h. The cells were washed with PBS after infection and overlaid with media without recombinant protein.

	Inhibitory concentration at MOI= 1.0 (µM)			**Inhibitory concentration at MOI= 0.01 (µM)**	
Proteins	**CPE formation**	**Reduction in viral RNA copy numbers (EC_{50})**		**CPE formation**	**Reduction in viral RNA copy numbers (EC_{50})**
Experiment 1: Inhibitory effect throughout the course of infection*					
CTC-445.2	10	1.807		2.31	0.753
CTC-445.2d	0.37	0.0235		0.01	<0.005
Experiment 2: Post-infection inhibitory effect#					
CTC-445.2	30	~10		10	2.947
CTC-445.2d	3.33	0.2369		0.12	0.0139
Experiment 3: Pre-infection inhibitory effect@					
CTC-445.2	>30	>30		~30	~29.33
CTC-445.2d	>10	~3.903		3.33	2.688

Table S4. CryoEM data collection, refinement, and validation statistics.

	CTC-445.2 SARS-CoV-2 S 6P (state 4)	CTC-445.2 SARS-CoV-2 S 6P (state 1)	CTC-445.2 SARS-CoV-2 S 6P (state 2)	CTC-445.2 SARS-CoV-2 S 6P (state 3)
PDB				
EMD				
Microscope	Talos Arctica	Talos Arctica	Talos Arctica	Talos Arctica
Camera	Gatan K3 Summit	Gatan K3 Summit	Gatan K3 Summit	Gatan K3 Summit
Magnification	45,000x	45,000x	45,000x	45,000x
Voltage (kV)	200	200	200	200
Recording mode	counting	counting	counting	counting
Dose rate (e^-/pixel/s)	15.6	15.6	15.6	15.6
Electron dose ($e^-/Å^2$)	60	60	60	60
Defocus range (μm)	0.8 - 2.0	0.8 - 2.0	0.8 - 2.0	0.8 - 2.0
Pixel size (Å)	0.869	0.869	0.869	0.869
Micrographs collected	2,250	2,250	2,250	2,250
Micrographs used	1,610	1,610	1,610	1,610
Extracted particles	420,998	420,998	420,998	420,998
Total refined particles	187,629	187,629	187,629	187,629
Particles in final volume	26,038	39,795	47,013	37,342
Symmetry imposed	C1	C1	C1	C1
Nominal Resolution (Å)				
FSC 0.5 (masked/unmasked)	4.5/8.8	4.4/8.6	4.3/9.4	6.6/9.8
FSC 0.143 (masked/unmasked)	4.1/6.5	3.8/7.5	3.8/6.8	4.2/7.8
Map sharpening B-factor ($Å^2$)	-70.9	-73.2	-68.9	-100.3
Refinement and Validation				
Model-to-map resolution (FSC 0.143/0.5)	4.0/4.3			
Number of atoms				
Protein	26,857			
Ligand	518			
MapCC (global/local)	0.79/0.81			
R.m.s. deviations				
Bond lengths (Å)	0.01			
Bond angles (°)	0.7			
MolProbity score	2.8			
CaBLAM outliers (%)	2.79			
Clashscore (all atom)	13.6			
Poor rotamers (%)	0.1			
Ramachandran plot				
Favored (%)	93.8			
Allowed (%)	6.2			
Disallowed (%)	0			

Table S5. Mean clinical scores for the Syrian hamsters data presented in Fig. 5C: A) SARS-CoV-2 challenged and CTC-445.2d treated, as well as the control groups: B) SARS-CoV-2 challenged and PBS treated, and C) unchallenged and untreated. The scoring system for assessing clinical symptoms is: lethargy (1 point), rough coat (1 point), moderate dyspnea (2 points), hunched back (1 point), and body weight loss 10-20% (1 point). Hamsters that accumulated a clinical score of 3 or higher were euthanized.

Day	A) Challenged, CTC-445.2d-treated	B) Challenged, PBS-treated	C) Unchallenged, untreated
0	0	0	0
1	0	0	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	1.3	0
7	1	3	0
10	0.25	-	0

Appendix A. Python/PyRosetta code to generate multiple initial perturbations for mobile secondary structure elements.

```
#########################################################################################################################
#############################################START#######################################################################
#########################################################################################################################

import argparse
import os
import numpy as np
import uuid
import pyrosetta

def _gaussian():
          return pyrosetta.rosetta.numeric.random.rg().gaussian()

def extract_subpose(pose_in, selector):
          pose = pose_in.clone()

          not_selector=pyrosetta.rosetta.core.select.residue_selector.NotResidueSelector()
          not_selector.set_residue_selector(selector)
          subset = not_selector.apply(pose)

          ranges = pyrosetta.rosetta.core.select.residue_selector.ResidueRanges(subset)
          intervals = []
          for rang in ranges:
          intervals.append((rang.start(), rang.stop()))

          for start, stop in reversed(intervals):
          pose.delete_residue_range_slow(start, stop)

          return pose

def extract_chains(pose_in, chains_str):
          '''
          Convenience function to extract a list of chains from a pose
          '''
          selector = pyrosetta.rosetta.core.select.residue_selector.ChainSelector(chains_str)
          return extract_subpose(pose_in, selector)

def split_pose(inPose, selectors):
          inPoseByChain=pyrosetta.rosetta.utility.vector1_core_pose_Pose()
          for selector in selectors:
          sel_pose=extract_subpose(inPose, selector)
          inPoseByChain.append(sel_pose)
          return inPoseByChain

def join_poses(poses):
          assert len(poses) >= 1
          joined = None
          for pose in poses:
          if joined is None:
          joined = pose.clone()
          else:
          offset = joined.size()
          joined.append_pose_by_jump(pose, 1)
          # copy pdb info
          for seqpos in range(1, pose.size()+1):
                    joined_seqpos = seqpos + offset
                    joined.pdb_info().chain(joined_seqpos, pose.pdb_info().chain(seqpos))
                    joined.pdb_info().number(joined_seqpos, pose.pdb_info().number(seqpos))
                    for label in pose.pdb_info().get_reslabels(seqpos):
                    joined.pdb_info().add_reslabel(joined_seqpos, label)
          joined.pdb_info().obsolete(False)
          return joined

def rodrigues_rotation(a,
                    angle,
                    verbose=False):
          #from math import cos, sin, acos, asin, sqrt, pi, radians, degrees
          "use Rodrigues' rotation formula to get rotation matrix"
          a = np.array(a, dtype=float)
          a /= np.sqrt(np.inner(a, a)) # make unit vector
          omega = np.array([[   0., -a[2],  a[1]],
                    [ a[2],   0., -a[0]],
                    [-a[1],  a[0],      0.]], dtype=float)
          rm = (np.identity(3) + omega * np.sin(angle)
                              + np.dot(omega, omega) * (1 - np.cos(angle)))
          return rm

class PackstatSimpleMetric(pyrosetta.rosetta.core.simple_metrics.RealMetric):
```

```
        def __init__(self, nrepeats=5):
        super(PackstatSimpleMetric, self).__init__()
        self.nrepeats = nrepeats

        def name(self):
        return self.__class__.__name__

        def metric(self):
        return 'packstat'

        def calculate(self, pose):
        result_array=[]
        for i in range(self.nrepeats):
        result_array.append(pyrosetta.rosetta.core.scoring.packstat.compute_packing_score(pose))
        value = np.average(result_array)
        return value

class CentroidScoreSimpleMetric(pyrosetta.rosetta.core.simple_metrics.RealMetric):
        def __init__(self, weights=['cen_std.wts', 'interchain_cen.wts'], min_score=-99999):
        super(CentroidScoreSimpleMetric, self).__init__()
        self.min_score = min_score
        self.sfxn = pyrosetta.rosetta.core.scoring.ScoreFunction()
        for filename in weights:
        self.sfxn.add_weights_from_file(filename)
        self.sfxn.set_weight(pyrosetta.rosetta.core.scoring.ScoreType.atom_pair_constraint, 1.0)

        def name(self):
        return self.__class__.__name__

        def metric(self):
        return 'centroid_score'

        def calculate(self, in_pose):
        return self.sfxn(in_pose)

class ClashCheckFilter(pyrosetta.rosetta.protocols.filters.Filter):
        def __init__(self, global_context_pose,
        pick_mode=pyrosetta.rosetta.core.pose.PoseCoordPickMode_N_CA_C_CB,
        radius=2.0):
        self.global_context_pose = global_context_pose
        self.pick_mode = pick_mode
        self.radius = radius
        self.global_context_pose_hash = pyrosetta.rosetta.core.pose.xyzStripeHashPose(self.global_context_pose,
                self.pick_mode, self.radius)

        def clash_map_to_str(self, pose, clash_map):
        newmap = {}
        for rsd1, rsd2 in clash_map.items():
        pdb_info = pose.pdb_info()
        r1 = '{}{}'.format(pdb_info.number(rsd1), pdb_info.chain(rsd1))
        pdb_info = self.global_context_pose.pdb_info()
        r2 = '{}{}'.format(pdb_info.number(rsd2), pdb_info.chain(rsd2))
        newmap[r1] = r2
        assert len(newmap) == len(clash_map)
        return newmap

        def apply(self, pose):
        # Extract balls
        target_balls = pyrosetta.rosetta.utility.vector1_numeric_geometry_hashing_Ball()
        pyrosetta.rosetta.core.pose.xyzStripeHashPose.extract_pose_balls(pose,
                target_balls,
                self.pick_mode)

        # Get clash map
        target_clash_map = pyrosetta.rosetta.std.map_unsigned_long_unsigned_long()
        self.global_context_pose_hash.clash_check_residue_pairs(target_balls,
                target_clash_map)

        if len(target_clash_map) > 0:
        return False

        return True

class RandomRoller(pyrosetta.rosetta.protocols.moves.Mover):
        def __init__(self, fixed_residue_selector, movable_residue_selectors,
        distance_step, angle_step):
        super(RandomRoller, self).__init__()
        self.fixed_residue_selector = fixed_residue_selector
        self.movable_residue_selectors = movable_residue_selectors
        self.angle_step = angle_step
        self.distance_step = distance_step

        def get_name(self):
```

```
        return self.__class__.__name__

        def apply(self, pose):
        movable_residue_sets, coord_sets, coms = self.generate_cached_data(pose)
        for ix, mset in enumerate(movable_residue_sets):
        #Calculate translation
        tVec = np.zeros((3, ), dtype=float)
        for iaxis in range(3):
                  tVec[iaxis] = _gaussian()*self.distance_step

        #Calculate rotation
        angle = _gaussian()*self.angle_step
        rMat1 = rodrigues_rotation([pyrosetta.rosetta.numeric.random.gaussian(),
                                    pyrosetta.rosetta.numeric.random.gaussian(),
                                    pyrosetta.rosetta.numeric.random.gaussian()],
                                    np.radians(angle))

        #Apply -T,R,+T
        coor_array = coord_sets[ix] - coms[ix]
        coor_array = np.dot(coor_array, rMat1)
        coor_array += coms[ix]
        coor_array += tVec

        #Apply_coordinates_to_pose
        tmp_cntr = 0 #It is easier just to keep a counter
        for ires in mset:
                  rsd = pose.residue(ires)
                  for jatom in range(1, rsd.natoms()+1):
                  tmp_xyz = pyrosetta.rosetta.numeric.xyzVector_double_t(coor_array[tmp_cntr][0],
                            coor_array[tmp_cntr][1],
                            coor_array[tmp_cntr][2])
                  pose.set_xyz(pyrosetta.rosetta.core.id.AtomID(jatom, ires), tmp_xyz)
                  tmp_cntr += 1

        def get_coordinates(self, pose, residues):
        coords = []
        for seqpos in residues:
        rsd = pose.residue(seqpos)
        for atom_ix in range(1, rsd.natoms()+1):
                  xyz = pose.residue(seqpos).xyz(atom_ix)
                  coords.append([xyz[0], xyz[1], xyz[2]])
        return np.array(coords)

        def generate_cached_data(self, pose):
        fixed_residue_subset = self.fixed_residue_selector.apply(pose)
        fixed_residues = pyrosetta.rosetta.core.select.residue_selector.ResidueVector(fixed_residue_subset)

        movable_residue_sets = []
        for sel in self.movable_residue_selectors:
        movable_residue_subset = sel.apply(pose)
        movable_residues = pyrosetta.rosetta.core.select.residue_selector.ResidueVector(movable_residue_subset)
        movable_residue_sets.append(movable_residues)

        # check for duplicate assignments
        residue_set = set(fixed_residues)
        for mset in movable_residue_sets:
        for m in mset:
                  if m in residue_set:
                  raise Exception('Residue {} is present in more than one Rockpack residue selection'.format(m))
                  residue_set.add(m)
        for seqpos in range(1, pose.size()+1):
        if seqpos not in residue_set:
                  raise Exception('Residue {} is not present in any Rockpack residue selection'.format(m))

        # find coordinates
        coord_sets = [self.get_coordinates(pose, mset) for mset in movable_residue_sets]
        coms = [np.mean(x, axis=0) for x in coord_sets]
        return movable_residue_sets, coord_sets, coms

class MonteCarlo(pyrosetta.rosetta.protocols.moves.Mover):
          def __init__(self, mover, metric, unique_run_id,
          out_file_dir='.',
          filters=[],
          reporting_frequency=100,
          num_perturbations=1000,
          num_trajectories=1,
          ):
          self.mover = mover
          self.metric = metric
          self.filters = filters
          self.reporting_frequency = reporting_frequency
          self.num_perturbations = num_perturbations
          self.num_trajectories = num_trajectories
```

```
        self.unique_run_id = unique_run_id
        self.out_file_dir = out_file_dir

        def score(self, pose):
        return self.metric.calculate(pose)

        def check_filters(self, pose):
        for f in self.filters:
        if not f.apply(pose):
                  return False
        return True

        def run(self, pose):
        ipose_start = pose
        results = []
        for trajectory_id in range(self.num_trajectories):
        ipose = self.run_trajectory(ipose_start, trajectory_id)
        results.append((self.score(ipose), ipose))
        if not self.check_filters(ipose):
                  continue

        if len(results) == 0:
        raise Exception('ERROR: No trajectories produced any poses that passed the filters.')

        results.sort()
        return results[0][1]

        def write_pdb(self, pose, traj, pert):
        pdb_file = '{}_rockpack_trajectory{:03d}_iteration{:06d}.pdb'.format(self.unique_run_id, traj, pert)
        pdb_path = os.path.join(self.out_file_dir, 'debug', pdb_file)
        pose.dump_pdb(pdb_path)

        def run_trajectory(self, ipose, traj_id):
        mover = self.mover
        metric = self.metric

        best_score = metric.calculate(ipose)
        best_pose = ipose.clone()
        if debug:
        self.write_pdb(best_pose, traj_id, 0)
        for icounter in range(1, self.num_perturbations+1):
        test_pose = ipose.clone()
        mover.apply(test_pose)
        test_score = metric.calculate(test_pose)

        if icounter%self.reporting_frequency == 0:
                  print("\n\nWorking on scoring RockPerturbation %d"%icounter)
                  print("Best score", best_score, ipose)

        if test_score < best_score:
                  if self.check_filters(test_pose):
                  if debug:
                            print('Found better score: {}, {}, {}'.format(icounter, best_score, test_score))
                            self.write_pdb(test_pose, traj_id, icounter)
                  best_pose = test_pose
                  best_score = test_score

        if debug:
        self.write_pdb(best_pose, traj_id, self.num_perturbations+1)
        return best_pose

class CentroidRocker(MonteCarlo):
          def __init__(self, fixed_residue_selector, **kwargs):
          super(CentroidRocker, self).__init__(**kwargs)
          self.fixed_residue_selector = fixed_residue_selector

          def switch_to_centroid_mode(self, pose):
          switch_to_centroid = pyrosetta.SwitchResidueTypeSetMover('centroid')
          switch_to_centroid.apply(pose)

          def switch_to_fullatom_mode(self, pose):
          switchToFA = pyrosetta.SwitchResidueTypeSetMover("fa_standard")
          switchToFA.apply(pose)

          def constrain_atom_pairs(self, pose):
          pairs = self._get_atom_pairs(pose)
          func = pyrosetta.rosetta.core.scoring.constraints.BoundFunc(5.5, 9999, 0.1, 'CA-CA')
          for iatom, jatom in pairs:
          cst = pyrosetta.rosetta.core.scoring.constraints.AtomPairConstraint(
                  iatom,
                  jatom,
                  func)
          pose.add_constraint(cst)
```

```
        def apply(self, pose):
        # Switch over to centroid mode
        self.switch_to_centroid_mode(pose)

        # Constrain atom pairs
        self.constrain_atom_pairs(pose)

        newpose = self.run(pose)

        # Switch back to full atom mode
        self.switch_to_fullatom_mode(newpose)
        pose.assign(newpose)

        def _get_atom_pairs(self, pose):
        fixed_residue_subset = self.fixed_residue_selector.apply(pose)
        pairs = []
        for ires in pose.residues:
        ## ires must be a rock pack chain
        if fixed_residue_subset[ires.seqpos()]:
                  continue

        iatom_index = ires.atom_index('CA')
        iatom = pyrosetta.rosetta.core.id.AtomID(iatom_index, ires.seqpos())

        for jres in pose.residues:
                  ## jres must be a different chain
                  if jres.chain() == ires.chain():
                  continue

                  jatom_index = jres.atom_index('CA')
                  jatom = pyrosetta.rosetta.core.id.AtomID(jatom_index, jres.seqpos())

                  pairs.append((iatom, jatom))
        return pairs

def pack_pose_generic(in_pose,
                  packaas=["A","L","I","V","F"],
                  scorefxn=None):
        packstatfilter=pyrosetta.rosetta.protocols.simple_filters.PackStatFilter()
        pack_task={}
        for ires in range(1,in_pose.total_residue()+1):
        pack_task[ires]=packaas
        packed_pose=pack_rotamers_fast(in_pose=in_pose,
                                  pack_task=pack_task,
                                  pack_scorefxn=scorefxn)
        return packed_pose

def pack_rotamers_fast(in_pose,
                  pack_task={},
                  pack_scorefxn=None
                  ):
        test_pose = in_pose.clone()
        pack_scorefxn(test_pose)
        task = pyrosetta.standard_packer_task( test_pose )
        for ipos in range(1,in_pose.total_residue()+1):
        raa_bool = pyrosetta.rosetta.utility.vector1_bool(20)
        for jndx in range( 20 ):
                  raa_bool[jndx+1]=False
        if ipos in pack_task:
        for jaa in pack_task[ipos]:
                  mutant_aa = pyrosetta.rosetta.core.chemical.aa_from_oneletter_code( jaa )
                  raa_bool[int(mutant_aa)]=True
        task.nonconst_residue_task( ipos
                  ).restrict_absent_canonical_aas( raa_bool )

        #Pack
        pyrosetta.rosetta.core.pack.pack_rotamers(test_pose,
                                                  pack_scorefxn,
                                                  task)
        return test_pose

def msd_bb_2_poses(pose1,
                  init_res1,
                  end_res1,
                  pose2,
                  init_res2,
                  end_res2):
        numRes=(end_res1-init_res1+1)
        coorA=np.zeros(((4*numRes),3), float)
        coorB=np.zeros(((4*numRes),3), float)

        counter=0
```

```
            for res in range (init_res1, (end_res1+1)):
            for dim in range(0,3):
            coorA[counter,dim]=(pose1.residue(res).xyz("CA")[dim])
            counter+=1
            for dim in range(0,3):
            coorA[counter,dim]=(pose1.residue(res).xyz("C")[dim])
            counter+=1
            for dim in range(0,3):
            coorA[counter,dim]=(pose1.residue(res).xyz("O")[dim])
            counter+=1
            for dim in range(0,3):
            coorA[counter,dim]=(pose1.residue(res).xyz("N")[dim])
            counter+=1

            counter=0
            for res in range (init_res2, (end_res2+1)):
            for dim in range(0,3):
            coorB[counter,dim]=(pose2.residue(res).xyz("CA")[dim])
            counter+=1
            for dim in range(0,3):
            coorB[counter,dim]=(pose2.residue(res).xyz("C")[dim])
            counter+=1
            for dim in range(0,3):
            coorB[counter,dim]=(pose2.residue(res).xyz("O")[dim])
            counter+=1
            for dim in range(0,3):
            coorB[counter,dim]=(pose2.residue(res).xyz("N")[dim])
            counter+=1

            msdVal=0.0
            for i in range(len(coorA)):
            tmpVal_buff= np.linalg.norm(coorA[i] - coorB[i])
            msdVal+=(tmpVal_buff*tmpVal_buff)
            msdVal=msdVal/len(coorA)
            msdVal=np.sqrt(msdVal)

            return msdVal

class RockPackRMSDFilter(pyrosetta.rosetta.protocols.filters.Filter):
            def __init__(self, reference_pose, residue_selectors, max_rmsd):
            super(RockPackRMSDFilter, self).__init__()
            self.residue_selectors = residue_selectors
            self.reference_pose = reference_pose.clone()
            self.reference_pose_by_chain = None
            self.max_rmsd = max_rmsd

            def apply(self, pose):
            if self.reference_pose_by_chain is None:
            self.reference_pose_by_chain = split_pose(self.reference_pose, self.residue_selectors)

            tmp_test_ipose_by_chain = split_pose(pose, self.residue_selectors)
            assert len(self.reference_pose_by_chain) == len(tmp_test_ipose_by_chain)

            for kndx, (reference_pose, chain_pose) in enumerate(zip(self.reference_pose_by_chain, tmp_test_ipose_by_chain)):
            assert reference_pose.size() == chain_pose.size()
            tmp_this_msd_to_starting_point=msd_bb_2_poses(
                        reference_pose, 1, reference_pose.size(),
                        chain_pose, 1, chain_pose.size())
            self.last_msd_to_reference_pose = tmp_this_msd_to_starting_point

            if tmp_this_msd_to_starting_point>self.max_rmsd:
                        return False
            return True

class RockPackMover(pyrosetta.rosetta.protocols.moves.Mover):
            def __init__(self, packaas=['V', 'I', 'L'], scorefxn=None):
            super(RockPackMover, self).__init__()
            if scorefxn is None:
            self.scorefxn = pyrosetta.get_score_function()
            else:
            self.scorefxn = scorefxn
            self.packaas = packaas

            def apply(self, in_pose):
            pack_task={}
            for ires in  range(1,in_pose.total_residue()+1):
            pack_task[ires]=self.packaas
            self.pack_rotamers_fast(in_pose=in_pose, pack_task=pack_task)

            def pack_rotamers_fast(self, in_pose, pack_task):
            self.scorefxn(in_pose)

            # Create task
```

```
        task = pyrosetta.standard_packer_task(in_pose)
        for ipos in range(1, in_pose.total_residue()+1):
        raa_bool = pyrosetta.rosetta.utility.vector1_bool(20)
        for jndx in range( 20 ):
                raa_bool[jndx+1]=False
        if ipos in pack_task:
                for jaa in pack_task[ipos]:
                mutant_aa = pyrosetta.rosetta.core.chemical.aa_from_oneletter_code( jaa )
                raa_bool[int(mutant_aa)]=True
        task.nonconst_residue_task(ipos).restrict_absent_canonical_aas(raa_bool)

        # Pack
        pyrosetta.rosetta.core.pack.pack_rotamers(in_pose, self.scorefxn, task)

class CompoundMover(pyrosetta.rosetta.protocols.moves.Mover):
        def __init__(self, movers):
        super(CompoundMover, self).__init__()
        self.movers = movers

        def apply(self, pose):
        for m in self.movers:
        m.apply(pose)

def rmsd_and_rosettaRotations_2_np_arrays(crds1,
                                          crds2):
        """Returns RMSD between 2 sets of [nx3] numpy array"""
        COM1 = np.sum(crds1,axis=0) / crds1.shape[0]
        COM2 = np.sum(crds2,axis=0) / crds2.shape[0]
        crds1-=COM1
        crds2-=COM2
        n_vec = np.shape(crds1)[0]
        correlation_matrix = np.dot(np.transpose(crds1), crds2)
        v, s, w_tr = np.linalg.svd(correlation_matrix)
        is_reflection = (np.linalg.det(v) * np.linalg.det(w_tr)) < 0.0
        if is_reflection:
                s[-1] = - s[-1]
                v[:,-1] = -v[:,-1]
        E0 = sum(sum(crds1 * crds1)) + sum(sum(crds2 * crds2))
        rmsd_sq = (E0 - 2.0*sum(s)) / float(n_vec)
        rmsd_sq = max([rmsd_sq, 0.0])
        rMtx=np.dot(v, w_tr)
        tVec=COM1-(np.dot(COM2, np.linalg.inv(rMtx)))
        rMtx_xyzM=pyrosetta.rosetta.numeric.xyzMatrix_double_t()
        rMtx_xyzM.xx=rMtx[0,0]
        rMtx_xyzM.xy=rMtx[0,1]
        rMtx_xyzM.xz=rMtx[0,2]
        rMtx_xyzM.yx=rMtx[1,0]
        rMtx_xyzM.yy=rMtx[1,1]
        rMtx_xyzM.yz=rMtx[1,2]
        rMtx_xyzM.zx=rMtx[2,0]
        rMtx_xyzM.zy=rMtx[2,1]
        rMtx_xyzM.zz=rMtx[2,2]
        tVec_xyzV=pyrosetta.rosetta.numeric.xyzVector_double_t()
        tVec_xyzV.x=tVec[0]
        tVec_xyzV.y=tVec[1]
        tVec_xyzV.z=tVec[2]
        return np.sqrt(rmsd_sq), rMtx_xyzM, tVec_xyzV

def align_atoms_by_ndxs(pose1,
                            init_res1,
                            end_res1,
                            pose2,
                            init_res2,
                            end_res2,
                            atoms=["CA","C","O","N"]):
        numRes=(end_res1-init_res1+1)
        coorA=np.zeros(((len(atoms)*numRes),3), float)
        coorB=np.zeros(((len(atoms)*numRes),3), float)

        counter=0
        for res in range (init_res1, (end_res1+1)):
        for atom in atoms:
                for dim in range(0,3):
                coorA[counter,dim]=(pose1.residue(res).xyz(atom)[dim])
                counter+=1

        counter=0
        for res in range (init_res2, (end_res2+1)):
        for atom in atoms:
                for dim in range(0,3):
                coorB[counter,dim]=(pose2.residue(res).xyz(atom)[dim])
                counter+=1
```

```
        rmsdVal, rMtx, tVec = rmsd_and_rosettaRotations_2_np_arrays(coorB, coorA)
        pose1.apply_transform_Rx_plus_v(rMtx, tVec)

        return rmsdVal


class FullAtomRockPacker(pyrosetta.rosetta.protocols.moves.Mover):
        def __init__(self, rock_mover, pack_mover, scorefxn, filters, num_perturbations, max_rmsd, out_file_dir, unique_run_id,
        core_max_sasa,
        is_underpacked_func=lambda x: x < 0.10):
        super(FullAtomRockPacker, self).__init__()
        self.score_metric = pyrosetta.rosetta.core.simple_metrics.metrics.TotalEnergyMetric()
        self.score_metric.set_scorefunction(scorefxn)
        self.packing_metric = PackstatSimpleMetric()
        self.is_underpacked = is_underpacked_func
        self.rock_mover = rock_mover
        self.pack_mover = pack_mover
        self.max_rmsd = max_rmsd
        self.filters = filters
        self.num_perturbations = num_perturbations
        self.out_file_dir = out_file_dir
        self.unique_run_id = unique_run_id
        self.small_rmsd_pert_num = 0.01

        def apply(self, pose):
        in_pose_ss_contigous=pose.clone()
        num_rockPack_randPert=self.num_perturbations
        out_file_dir = self.out_file_dir
        unique_run_id = self.unique_run_id
        min_allowed_rock_packstat_score = 0.10
        rockpack_reporting_frequency = 10

        # for checking RMSD; needs to be created at apply time because it depends
        # on the reference pose being right
        rmsd_filter = RockPackRMSDFilter(reference_pose=pose,
                  residue_selectors=self.rock_mover.movable_residue_selectors,
                  max_rmsd=self.max_rmsd)

        best_packed_perturbation=[]
        ipose = in_pose_ss_contigous.clone()
        self.pack_mover.apply(ipose)

        max_allowed_rock_energy_score=self.score_metric.calculate(ipose) #Set the maximum score to the initial score
        if debug:
        ipose.dump_pdb("%s/debug/%s_rockpackInit_test_optSSPosition.pdb"%(out_file_dir,
                  unique_run_id))

        for icounter in range(num_rockPack_randPert):
        tmp_test_ipose=ipose.clone()

        # Random rigid body perturbation
        self.rock_mover.apply(tmp_test_ipose)

        # Assess RMSD by SS element and skip the perturbations if it is too large
        if not rmsd_filter.apply(tmp_test_ipose):
                  continue

        # Repack with small set of residues to evaluate packing
        self.pack_mover.apply(tmp_test_ipose)

        # Calc score after packing
        tmp_cen_score=self.score_metric.calculate(tmp_test_ipose)

        if (icounter%rockpack_reporting_frequency==0):
                  print("\n\nRockPerturbation %d"%icounter)
                  print("Best_packed_perturbation: ", best_packed_perturbation)

        if ((len(best_packed_perturbation)==0) or
                  (tmp_cen_score<best_packed_perturbation[0])):
                  tmp_packstat_score=self.packing_metric.calculate(tmp_test_ipose)
                  if ((len(best_packed_perturbation)>0) and
                            self.is_underpacked(tmp_packstat_score)):
                  continue
                  if ((len(best_packed_perturbation)>0) and
                            (tmp_packstat_score<best_packed_perturbation[1])):
                  continue

                  print("Accepted: ", tmp_cen_score, tmp_packstat_score)
                  best_packed_perturbation = [tmp_cen_score, tmp_packstat_score, tmp_test_ipose.clone()]
                  if debug:
                  tmp_test_ipose.dump_pdb("%s/debug/%s_rockpack_%05d_%s_test_optSSPosition_best.pdb"%(out_file_dir,
                            unique_run_id,
                            icounter,
```

```
                        unique_run_id))
                ipose=tmp_test_ipose.clone()

        rolled_poses=None

        if len(best_packed_perturbation)==0:
        return

        target_rockedPose_best = best_packed_perturbation[2]
        if debug:
        target_rockedPose_best.dump_pdb("%s/debug/%s_rockpackBest_test_optSSPosition.pdb"%(out_file_dir,
                unique_run_id))

        chain_selectors = [self.rock_mover.fixed_residue_selector] + list(self.rock_mover.movable_residue_selectors)
        inpose_split_by_ss=split_pose(pose, chain_selectors)
        target_rockedPose_split_by_ss=split_pose(target_rockedPose_best, chain_selectors)
        largest_chains_for_rocking = [(None, 1, target_rockedPose_split_by_ss[x].size(), 0) for x in range(1,
1+len(target_rockedPose_split_by_ss))]

        inpose_split_by_ss_optimized_position=[]
        for indx in range(len(inpose_split_by_ss)):
        inpose_split_by_ss_optimized_position.append(inpose_split_by_ss[indx+1].clone())

        if indx > 0:
                assert(inpose_split_by_ss[indx+1].total_residue()==
                        (largest_chains_for_rocking[indx][2]-largest_chains_for_rocking[indx][1]+1 ))

                tmp_rmsd=align_atoms_by_ndxs(inpose_split_by_ss_optimized_position[-1],
                                        1,
                                        inpose_split_by_ss_optimized_position[-1].total_residue(),
                                        target_rockedPose_split_by_ss[indx+1],
                                        largest_chains_for_rocking[indx][1],
                                        largest_chains_for_rocking[indx][2],
                                        atoms=["CA","C","O","N"])

                tmp_rmsd=tmp_rmsd-largest_chains_for_rocking[indx][3]
                assert(tmp_rmsd<self.small_rmsd_pert_num)

        in_pose_ss_rockOptimized_contigous=join_poses(inpose_split_by_ss_optimized_position)
        if debug:
        in_pose_ss_rockOptimized_contigous.dump_pdb("%s/debug/%s_test_optSSPosition.pdb"%(out_file_dir,
                unique_run_id))

        pose.assign(in_pose_ss_rockOptimized_contigous)
        print('Assigned to {}'.format(in_pose_ss_rockOptimized_contigous))


def parse_flags():
        parser = argparse.ArgumentParser(description='Refines secondary structure position')

        parser.add_argument("--in_pdb_name", required=True,
                        type=str, help='Input PDB containing multiple chains to work on')

        parser.add_argument('--global_context_pdb', type=str, default=None, help='PDB containing the global context (e.g.
receptor)')

        parser.add_argument("--num_rockPack_randPert",
                        type=int, dest="num_rockPack_randPert", default=1000)

        parser.add_argument("--max_rockPack_rmsd",
                        type=float, dest="max_rockPack_rmsd", default=3.0)

        parser.add_argument("--rockPack_chains",
                        type=str, nargs='*', dest="rockPack_chains", default=[])

        parser.add_argument("--rockPack_low_resolution_refinement",
                        action='store_true', dest='rockPack_low_resolution_refinement', default=False,
                        help='Activates low resolution refinement of specific SS elements')

        parser.add_argument("--rockPack_disable_high_resolution_refinement",
                        action='store_false', dest='rockPack_high_resolution_refinement',
                        help='Deactivates high resolution refinement of specific SS elements')

        return parser.parse_args()

# Constants
debug = False
layer_sasa_definitions={}
layer_sasa_definitions["core"]=10.0 #% <=10.0-limbo
layer_sasa_definitions["limbo"]=25.0 #% <=25.5
layer_sasa_definitions["surf"]=40.0 #% <=40.0
layer_sasa_definitions["interface"]=1.0 #% <=, This is chainge in SASA with and without context
unique_run_id = str(uuid.uuid4())
```

```
args = parse_flags()

# Prepare debug folder
if debug and not os.path.exists('debug'):
          os.mkdir('debug')

# Initialize pyrosetta
pyrosetta.init('-beta 1 -mute all')

# Residue Selectors
rockpack_selectors = []
for chain_str in args.rockPack_chains:
          selector = pyrosetta.rosetta.core.select.residue_selector.ChainSelector(chain_str)
          rockpack_selectors.append(selector)

rockpack_all = pyrosetta.rosetta.core.select.residue_selector.OrResidueSelector()
for sel in rockpack_selectors:
          rockpack_all.add_residue_selector(sel)

fixed_selector = pyrosetta.rosetta.core.select.residue_selector.NotResidueSelector()
fixed_selector.set_residue_selector(rockpack_all)

# Load the pose
pose = pyrosetta.pose_from_pdb(args.in_pdb_name)

# Debug, print out residues
residues = pyrosetta.rosetta.core.select.residue_selector.ResidueVector(fixed_selector.apply(pose))
for ix, selector in enumerate(rockpack_selectors):
          rockpack_residues = pyrosetta.rosetta.core.select.residue_selector.ResidueVector(selector.apply(pose))

if args.rockPack_low_resolution_refinement:
          print('Running low resolution rockPack ({} iterations)'.format(args.num_rockPack_randPert))
          mover = RandomRoller(fixed_residue_selector=fixed_selector, movable_residue_selectors=rockpack_selectors,
          distance_step=1.2, angle_step=8.0)

          # filters
          if args.global_context_pdb is None:
          filters = []
          else:
          global_context_pose = pyrosetta.pose_from_file(args.global_context_pdb)
          filters = [ClashCheckFilter(global_context_pose)]

          # Create the mover
          rocker = CentroidRocker(fixed_residue_selector=fixed_selector,
          filters=filters,
          mover=mover,
          metric=CentroidScoreSimpleMetric(),
          out_file_dir='.',
          unique_run_id=str(uuid.uuid4()),
          num_perturbations=args.num_rockPack_randPert)
          rocker.apply(pose)

if args.rockPack_high_resolution_refinement:
          print('Running high resolution rockPack ({} iterations)'.format(args.num_rockPack_randPert))

          # Full atom rockpack objects
          scorefxn_fa = pyrosetta.get_fa_scorefxn()

          # Movers
          fast_pack_mover = RockPackMover(packaas=['V', 'I', 'L'], scorefxn=scorefxn_fa)
          rock_mover = RandomRoller(fixed_residue_selector=fixed_selector, movable_residue_selectors=rockpack_selectors,
          distance_step=1.2, angle_step=8.0)

          # Full atom rockpack
          rocker_fa = FullAtomRockPacker(filters=[], rock_mover=rock_mover,
          pack_mover=fast_pack_mover, scorefxn=scorefxn_fa,
          num_perturbations=args.num_rockPack_randPert, out_file_dir='.', unique_run_id=unique_run_id,
          max_rmsd=args.max_rockPack_rmsd, core_max_sasa=layer_sasa_definitions['core'])

          # Run
          rocker_fa.apply(pose)

pose.dump_pdb('output.pdb')

#########################################################################################################################
#################################################END#####################################################################
#########################################################################################################################
```

References and Notes

1. D. E. Gordon, G. M. Jang, M. Bouhaddou, J. Xu, K. Obernier, K. M. White, M. J. O'Meara, V. V. Rezelj, J. Z. Guo, D. L. Swaney, T. A. Tummino, R. Hüttenhain, R. M. Kaake, A. L. Richards, B. Tutuncuoglu, H. Foussard, J. Batra, K. Haas, M. Modak, M. Kim, P. Haas, B. J. Polacco, H. Braberg, J. M. Fabius, M. Eckhardt, M. Soucheray, M. J. Bennett, M. Cakir, M. J. McGregor, Q. Li, B. Meyer, F. Roesch, T. Vallet, A. Mac Kain, L. Miorin, E. Moreno, Z. Z. C. Naing, Y. Zhou, S. Peng, Y. Shi, Z. Zhang, W. Shen, I. T. Kirby, J. E. Melnyk, J. S. Chorba, K. Lou, S. A. Dai, I. Barrio-Hernandez, D. Memon, C. Hernandez-Armenta, J. Lyu, C. J. P. Mathy, T. Perica, K. B. Pilla, S. J. Ganesan, D. J. Saltzberg, R. Rakesh, X. Liu, S. B. Rosenthal, L. Calviello, S. Venkataramanan, J. Liboy-Lugo, Y. Lin, X.-P. Huang, Y. Liu, S. A. Wankowicz, M. Bohn, M. Safari, F. S. Ugur, C. Koh, N. S. Savar, Q. D. Tran, D. Shengjuler, S. J. Fletcher, M. C. O'Neal, Y. Cai, J. C. J. Chang, D. J. Broadhurst, S. Klippsten, P. P. Sharp, N. A. Wenzell, D. Kuzuoglu-Ozturk, H.-Y. Wang, R. Trenker, J. M. Young, D. A. Cavero, J. Hiatt, T. L. Roth, U. Rathore, A. Subramanian, J. Noack, M. Hubert, R. M. Stroud, A. D. Frankel, O. S. Rosenberg, K. A. Verba, D. A. Agard, M. Ott, M. Emerman, N. Jura, M. von Zastrow, E. Verdin, A. Ashworth, O. Schwartz, C. d'Enfert, S. Mukherjee, M. Jacobson, H. S. Malik, D. G. Fujimori, T. Ideker, C. S. Craik, S. N. Floor, J. S. Fraser, J. D. Gross, A. Sali, B. L. Roth, D. Ruggero, J. Taunton, T. Kortemme, P. Beltrao, M. Vignuzzi, A. García-Sastre, K. M. Shokat, B. K. Shoichet, N. J. Krogan, A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature* **583**, 459–468 (2020). doi:10.1038/s41586-020-2286-9 Medline

2. C. Liu, Q. Zhou, Y. Li, L. V. Garner, S. P. Watkins, L. J. Carter, J. Smoot, A. C. Gregg, A. D. Daniels, S. Jervey, D. Albaiu, Research and development on therapeutic agents and vaccines for COVID-19 and related human coronavirus diseases. *ACS Cent. Sci.* **6**, 315–331 (2020). doi:10.1021/acscentsci.0c00272 Medline

3. W.-H. Chen, U. Strych, P. J. Hotez, M. E. Bottazzi, The SARS-CoV-2 vaccine pipeline: An overview. *Curr. Trop. Med. Rep.* **7**, 1–4 (2020). doi:10.1007/s40475-020-00201-6 Medline

4. K. K. Chan, D. Dorosky, P. Sharma, S. A. Abbasi, J. M. Dye, D. M. Kranz, A. S. Herbert, E. Procko, Engineering human ACE2 to optimize binding to the spike protein of SARS coronavirus 2. *Science* **369**, 1261–1265 (2020). doi:10.1126/science.abc0870 Medline

5. J. D. Walter, C. A. J. Hutter, I. Zimmermann, J. Earp, P. Egloff, M. Sorgenfrei, L. M. Hürlimann, I. Gonda, G. Meier, S. Remm, S. Thavarasah, P. Plattet, M. A. Seeger, Sybodies targeting the SARS-CoV-2 receptor-binding domain. bioRxiv 045419 [Preprint]. 16 May 2020. https://doi.org/10.1101/2020.04.16.045419.

6. L. Cao, I. Goreshnik, B. Coventry, J. B. Case, L. Miller, L. Kozodoy, R. E. Chen, L. Carter, A. C. Walls, Y.-J. Park, E.-M. Strauch, L. Stewart, M. S. Diamond, D. Veesler, D. Baker, De novo design of picomolar SARS-CoV-2 miniprotein inhibitors. *Science* **370**, 426–431 (2020). doi:10.1126/science.abd9909 Medline

7. L. Zhang, D. Lin, X. Sun, U. Curth, C. Drosten, L. Sauerhering, S. Becker, K. Rox, R. Hilgenfeld, Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Science* **368**, 409–412 (2020). doi:10.1126/science.abb3405 Medline

8. E. C. Smith, M. R. Denison, Coronaviruses as DNA wannabes: A new model for the regulation of RNA virus replication fidelity. *PLOS Pathog.* **9**, e1003760 (2013). doi:10.1371/journal.ppat.1003760 Medline

9. S. Duffy, Why are RNA virus mutation rates so damn high? *PLOS Biol.* **16**, e3000003 (2018).

doi:10.1371/journal.pbio.3000003 Medline

10. Z. Zhao, H. Li, X. Wu, Y. Zhong, K. Zhang, Y.-P. Zhang, E. Boerwinkle, Y.-X. Fu, Moderate mutation rate in the SARS coronavirus genome and its implications. *BMC Evol. Biol.* **4**, 21 (2004). doi:10.1186/1471-2148-4-21 Medline

11. M. B. Doud, S. E. Hensley, J. D. Bloom, Complete mapping of viral escape from neutralizing antibodies. *PLOS Pathog.* **13**, e1006271 (2017). doi:10.1371/journal.ppat.1006271 Medline

12. T. Phan, Genetic diversity and evolution of SARS-CoV-2. *Infect. Genet. Evol.* **81**, 104260 (2020). doi:10.1016/j.meegid.2020.104260 Medline

13. J. Hu, C.-L. He, Q.-Z. Gao, G.-J. Zhang, X.-X. Cao, Q.-X. Long, H.-J. Deng, L.-Y. Huang, J. Chen, K. Wang, N. Tang, A.-L. Huang, D614G mutation of SARS-CoV-2 spike protein enhances viral infectivity. bioRxiv 161323 [Preprint]. 6 July 2020. https://doi.org/10.1101/2020.06.20.161323.

14. A. Baum, B. O. Fulton, E. Wloga, R. Copin, K. E. Pascal, V. Russo, S. Giordano, K. Lanza, N. Negron, M. Ni, Y. Wei, G. S. Atwal, A. J. Murphy, N. Stahl, G. D. Yancopoulos, C. A. Kyratsous, Antibody cocktail to SARS-CoV-2 spike protein prevents rapid mutational escape seen with individual antibodies. *Science* **369**, 1014–1018 (2020). doi:10.1126/science.abd0831 Medline

15. T. N. Starr, A. J. Greaney, S. K. Hilton, K. H. D. Crawford, M. J. Navarro, J. E. Bowen, M. A. Tortorici, A. C. Walls, D. Veesler, J. D. Bloom, Deep mutational scanning of SARS-CoV-2 receptor binding domain reveals constraints on folding and ACE2 binding. *Cell* **182**, P1295–P1310.E20 (2020). doi:10.1016/j.cell.2020.08.012 Medline

16. J. ter Meulen, E. N. van den Brink, L. L. M. Poon, W. E. Marissen, C. S. W. Leung, F. Cox, C. Y. Cheung, A. Q. Bakker, J. A. Bogaards, E. van Deventer, W. Preiser, H. W. Doerr, V. T. Chow, J. de Kruif, J. S. M. Peiris, J. Goudsmit, Human monoclonal antibody combination against SARS coronavirus: Synergy and coverage of escape mutants. *PLOS Med.* **3**, e237 (2006). doi:10.1371/journal.pmed.0030237 Medline

17. L. Enjuanes, S. Zuñiga, C. Castano-Rodriguez, J. Gutierrez-Alvarez, J. Canton, I. Sola, "Molecular basis of coronavirus virulence and vaccine development," in *Advances in Virus Research*, J. Ziebuhr, Ed. (Elsevier, 2016), vol. 96, pp. 245–286.

18. S. Belouzard, V. C. Chu, G. R. Whittaker, Activation of the SARS coronavirus spike protein via sequential proteolytic cleavage at two distinct sites. *Proc. Natl. Acad. Sci. U.S.A.* **106**, 5871–5876 (2009). doi:10.1073/pnas.0809524106 Medline

19. J. Shang, Y. Wan, C. Luo, G. Ye, Q. Geng, A. Auerbach, F. Li, Cell entry mechanisms of SARS-CoV-2. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 11727–11734 (2020). doi:10.1073/pnas.2003138117 Medline

20. J. K. Millet, G. R. Whittaker, Host cell entry of Middle East respiratory syndrome coronavirus after two-step, furin-mediated activation of the spike protein. *Proc. Natl. Acad. Sci. U.S.A.* **111**, 15214–15219 (2014). doi:10.1073/pnas.1407087111 Medline

21. X. Ou, Y. Liu, X. Lei, P. Li, D. Mi, L. Ren, L. Guo, R. Guo, T. Chen, J. Hu, Z. Xiang, Z. Mu, X. Chen, J. Chen, K. Hu, Q. Jin, J. Wang, Z. Qian, Characterization of spike glycoprotein of SARS-CoV-2 on virus entry and its immune cross-reactivity with SARS-CoV. *Nat. Commun.* **11**, 1620 (2020). doi:10.1038/s41467-020-15562-9 Medline

22. Z. Song, Y. Xu, L. Bao, L. Zhang, P. Yu, Y. Qu, H. Zhu, W. Zhao, Y. Han, C. Qin, From SARS to MERS, thrusting coronaviruses into the spotlight. *Viruses* **11**, 59 (2019). doi:10.3390/v11010059 Medline

23. R. Yan, Y. Zhang, Y. Li, L. Xia, Y. Guo, Q. Zhou, Structural basis for the recognition of SARS-CoV-2 by full-length human ACE2. *Science* **367**, 1444–1448 (2020). doi:10.1126/science.abb2762 Medline

24. Q. Wang, Y. Zhang, L. Wu, S. Niu, C. Song, Z. Zhang, G. Lu, C. Qiao, Y. Hu, K.-Y. Yuen, Q. Wang, H. Zhou, J. Yan, J. Qi, Structural and functional basis of SARS-CoV-2 entry by using human ACE2. *Cell* **181**, 894–904.e9 (2020). doi:10.1016/j.cell.2020.03.045 Medline

25. J. Lan, J. Ge, J. Yu, S. Shan, H. Zhou, S. Fan, Q. Zhang, X. Shi, Q. Wang, L. Zhang, X. Wang, Structure of the SARS-CoV-2 spike receptor-binding domain bound to the ACE2 receptor. *Nature* **581**, 215–220 (2020). doi:10.1038/s41586-020-2180-5 Medline

26. D.-A. Silva, S. Yu, U. Y. Ulge, J. B. Spangler, K. M. Jude, C. Labão-Almeida, L. R. Ali, A. Quijano-Rubio, M. Ruterbusch, I. Leung, T. Biary, S. J. Crowley, E. Marcos, C. D. Walkey, B. D. Weitzner, F. Pardo-Avila, J. Castellanos, L. Carter, L. Stewart, S. R. Riddell, M. Pepper, G. J. L. Bernardes, M. Dougan, K. C. Garcia, D. Baker, De novo design of potent and selective mimics of IL-2 and IL-15. *Nature* **565**, 186–191 (2019). doi:10.1038/s41586-018-0830-7 Medline

27. F. Sesterhenn, C. Yang, J. Bonet, J. T. Cramer, X. Wen, Y. Wang, C.-I. Chiang, L. A. Abriata, I. Kucharska, G. Castoro, S. S. Vollers, M. Galloux, E. Dheilly, S. Rosset, P. Corthésy, S. Georgeon, M. Villard, C.-A. Richard, D. Descamps, T. Delgado, E. Oricchio, M.-A. Rameix-Welti, V. Más, S. Ervin, J.-F. Eléouët, S. Riffault, J. T. Bates, J.-P. Julien, Y. Li, T. Jardetzky, T. Krey, B. E. Correia, De novo protein design enables the precise induction of RSV-neutralizing antibodies. *Science* **368**, eaay5051 (2020). doi:10.1126/science.aay5051 Medline

28. J. Zhou, A. E. Panaitiu, G. Grigoryan, A general-purpose protein design framework based on mining sequence-structure relationships in known protein structures. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 1059–1068 (2020). doi:10.1073/pnas.1908723117 Medline

29. D.-A. Silva, B. E. Correia, E. Procko, "Motif-driven design of protein–protein interfaces," in *Computational Design of Ligand Binding Proteins*, B. L. Stoddard, Ed. (Humana, 2016), vol. 1414, pp. 285–304.

30. A. Quijano-Rubio, U. Y. Ulge, C. D. Walkey, D.-A. Silva, The advent of de novo proteins for cancer immunotherapy. *Curr. Opin. Chem. Biol.* **56**, 119–128 (2020). doi:10.1016/j.cbpa.2020.02.002 Medline

31. A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y.-E. A. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popović, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker, P. Bradley, ROSETTA3: An object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* **487**, 545–574 (2011). doi:10.1016/B978-0-12-381270-4.00019-6 Medline

32. E. Marcos, D.-A. Silva, Essentials of de novo protein design: Methods and applications. *WIREs Comput Mol Sci* **8**, e1374 (2018). doi:10.1002/wcms.1374

33. A. Chevalier, D.-A. Silva, G. J. Rocklin, D. R. Hicks, R. Vergara, P. Murapa, S. M. Bernard, L. Zhang, K.-H. Lam, G. Yao, C. D. Bahl, S.-I. Miyashita, I. Goreshnik, J. T. Fuller, M. T. Koday, C. M. Jenkins, T. Colvin, L. Carter, A. Bohn, C. M. Bryan, D. A. Fernández-Velasco, L. Stewart, M. Dong, X. Huang, R. Jin, I. A. Wilson, D. H. Fuller, D. Baker, Massively parallel de novo protein design for targeted therapeutics. *Nature* **550**, 74–79 (2017). doi:10.1038/nature23912 Medline

34. T. A. Whitehead, A. Chevalier, Y. Song, C. Dreyfus, S. J. Fleishman, C. De Mattos, C. A. Myers, H.

Kamisetty, P. Blair, I. A. Wilson, D. Baker, Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nat. Biotechnol.* **30**, 543–548 (2012). doi:10.1038/nbt.2214 Medline

35. M. Hoffmann, H. Kleine-Weber, S. Schroeder, N. Krüger, T. Herrler, S. Erichsen, T. S. Schiergens, G. Herrler, N.-H. Wu, A. Nitsche, M. A. Müller, C. Drosten, S. Pöhlmann, SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor. *Cell* **181**, 271–280.e8 (2020). doi:10.1016/j.cell.2020.02.052 Medline

36. W. Sungnak, N. Huang, C. Bécavin, M. Berg, R. Queen, M. Litvinukova, C. Talavera-López, H. Maatz, D. Reichart, F. Sampaziotis, K. B. Worlock, M. Yoshida, J. L. Barnes; HCA Lung Biological Network, SARS-CoV-2 entry factors are highly expressed in nasal epithelial cells together with innate immune genes. *Nat. Med.* **26**, 681–687 (2020). doi:10.1038/s41591-020-0868-6 Medline

37. M. P. Baker, H. M. Reynolds, B. Lumicisi, C. J. Bryson, Immunogenicity of protein therapeutics: The key causes, consequences and challenges. *Self Nonself* **1**, 314–322 (2010). doi:10.4161/self.1.4.13904 Medline

38. N. Casadevall, J. Nataf, B. Viron, A. Kolta, J.-J. Kiladjian, P. Martin-Dupont, P. Michaud, T. Papo, V. Ugo, I. Teyssandier, B. Varet, P. Mayeux, Pure red-cell aplasia and antierythropoietin antibodies in patients treated with recombinant erythropoietin. *N. Engl. J. Med.* **346**, 469–475 (2002). doi:10.1056/NEJMoa011931 Medline

39. M. G. Tovey, C. Lallemand, Immunogenicity and other problems associated with the use of biopharmaceuticals. *Ther. Adv. Drug Saf.* **2**, 113–128 (2011). doi:10.1177/2042098611406318 Medline

40. J. Li, C. Yang, Y. Xia, A. Bertino, J. Glaspy, M. Roberts, D. J. Kuter, Thrombocytopenia caused by the development of antibodies to thrombopoietin. *Blood* **98**, 3241–3248 (2001). doi:10.1182/blood.V98.12.3241 Medline

41. M. P. Ettinger, T. W. Littlejohn, S. L. Schwartz, S. R. Weiss, H. H. McIlwain, S. B. Heymsfield, G. A. Bray, W. G. Roberts, E. R. Heyman, N. Stambler, S. Heshka, C. Vicary, H.-P. Guler, Recombinant variant of ciliary neurotrophic factor for weight loss in obese adults: A randomized, dose-ranging study. *JAMA* **289**, 1826–1832 (2003). doi:10.1001/jama.289.14.1826 Medline

42. M. Findeisen, T. L. Allen, D. C. Henstridge, H. Kammoun, A. E. Brandon, L. L. Baggio, K. I. Watt, M. Pal, L. Cron, E. Estevez, C. Yang, G. M. Kowalski, L. O'Reilly, C. Egan, E. Sun, L. M. Thai, G. Krippner, T. E. Adams, R. S. Lee, J. Grötzinger, C. Garbers, S. Risis, M. J. Kraakman, N. A. Mellet, J. Sligar, E. T. Kimber, R. L. Young, M. A. Cowley, C. R. Bruce, P. J. Meikle, P. A. Baldock, P. Gregorevic, T. J. Biden, G. J. Cooney, D. J. Keating, D. J. Drucker, S. Rose-John, M. A. Febbraio, Treatment of type 2 diabetes with the designer cytokine IC7Fc. *Nature* **574**, 63–68 (2019). doi:10.1038/s41586-019-1601-9 Medline

43. G. Gao, C. Lebherz, D. J. Weiner, R. Grant, R. Calcedo, B. McCullough, A. Bagg, Y. Zhang, J. M. Wilson, Erythropoietin gene therapy leads to autoimmune anemia in macaques. *Blood* **103**, 3300–3302 (2004). doi:10.1182/blood-2003-11-3852 Medline

44. H. Schellekens, N. Casadevall, Immunogenicity of recombinant human proteins: Causes and consequences. *J. Neurol.* **251** (Suppl 2), II4–II9 (2004). doi:10.1007/s00415-004-1202-9 Medline

45. I. Mukovozov, T. Sabljic, G. Hortelano, F. A. Ofosu, Factors that contribute to the immmunogenicity of therapeutic recombinant human proteins. *Thromb. Haemost.* **99**, 874–882 (2008). doi:10.1160/TH07-11-0654 Medline

46. M. Sauerborn, V. Brinks, W. Jiskoot, H. Schellekens, Immunological mechanism underlying the immune response to recombinant human protein therapeutics. *Trends Pharmacol. Sci.* **31**, 53–59 (2010). doi:10.1016/j.tips.2009.11.001 Medline

47. W. B. Alsoussi, J. S. Turner, J. B. Case, H. Zhao, A. J. Schmitz, J. Q. Zhou, R. E. Chen, T. Lei, A. A. Rizk, K. M. McIntire, E. S. Winkler, J. M. Fox, N. M. Kafai, L. B. Thackray, A. O. Hassan, F. Amanat, F. Krammer, C. T. Watson, S. H. Kleinstein, D. H. Fremont, M. S. Diamond, A. H. Ellebedy, A potently neutralizing antibody protects mice against SARS-CoV-2 infection. *J. Immunol.* **205**, 915–922 (2020). doi:10.4049/jimmunol.2000583 Medline

48. M. Schoof, B. Faust, R. A. Saunders, S. Sangwan, V. Rezelj, N. Hoppe, M. Boone, C. Bache Billesbølle, M. Zimanyi, I. Deshpande, J. Liang, A. A. Anand, N. Dobzinski, B. S. Zha, B. Barsi-Rhyne, V. Belyy, A. W. Barile-Hill, S. Gupta, C. R. Simoneau, K. Leon, K. M. White, S. Nock, Y. Liu, N. J. Krogan, C. Y. Ralston, D. L. Swaney, A. García-Sastre, M. Ott, M. Vignuzzi, QCRG Structural Biology Consortium, P. Walter, A. Manglik, An ultra-high affinity synthetic nanobody blocks SARS-CoV-2 infection by locking Spike into an inactive conformation. bioRxiv 238469 [Preprint]. 17 August 2020. https://doi.org/10.117101/2020.08.08.238469.

49. E. Marcos, B. Basanta, T. M. Chidyausiku, Y. Tang, G. Oberdorfer, G. Liu, G. V. T. Swapna, R. Guan, D.-A. Silva, J. Dou, J. H. Pereira, R. Xiao, B. Sankaran, P. H. Zwart, G. T. Montelione, D. Baker, Principles for designing proteins with cavities formed by curved β sheets. *Science* **355**, 201–206 (2017). doi:10.1126/science.aah7389 Medline

50. National Center for Biotechnology Information, "VAST: Non-redundant PDB chain set" (2016); https://structure.ncbi.nlm.nih.gov/Structure/VAST/nrpdb.html.

51. S. Minami, K. Sawada, G. Chikenji, MICAN: A protein structure alignment algorithm that can handle Multiple-chains, Inverse alignments, C(α) only models, Alternative alignments, and Non-sequential alignments. *BMC Bioinformatics* **14**, 24 (2013). doi:10.1186/1471-2105-14-24 Medline

52. J. Xu, Y. Zhang, How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics* **26**, 889–895 (2010). doi:10.1093/bioinformatics/btq066 Medline

53. Y. Zhang, J. Skolnick, Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702–710 (2004). doi:10.1002/prot.20264 Medline

54. D. G. Gibson, L. Young, R.-Y. Chuang, J. C. Venter, C. A. Hutchison 3rd, H. O. Smith, Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* **6**, 343–345 (2009). doi:10.1038/nmeth.1318 Medline

55. C. O. Barnes, A. P. West, K. E. Huey-Tubman, M. A. G. Hoffmann, N. G. Sharaf, P. R. Hoffman, N. Koranda, H. B. Gristick, C. Gaebler, F. Muecksch, J. C. Cetrulo Lorenzi, S. Finkin, T. Hagglof, A. Hurley, K. G. Millard, Y. Weisblum, F. Schmidt, T. Hatziioannou, P. D. Bieniasz, M. Caskey, D. F. Robbiani, M. C. Nussenzweig, P. J. Bjorkman, Structures of human antibodies bound to SARS-CoV-2 spike reveal common epitopes and recurrent features of antibodies. *Cell* **182**, P828–P842.E16 (2020). doi:10.1016/j.cell.2020.06.025 Medline

56. C.-L. Hsieh, J. A. Goldsmith, J. M. Schaub, A. M. DiVenere, H.-C. Kuo, K. Javanmardi, K. C. Le, D. Wrapp, A. G. Lee, Y. Liu, C.-W. Chou, P. O. Byrne, C. K. Hjorth, N. V. Johnson, J. Ludes-Meyers, A. W. Nguyen, J. Park, N. Wang, D. Amengor, J. J. Lavinder, G. C. Ippolito, J. A. Maynard, I. J. Finkelstein, J. S. McLellan, Structure-based design of prefusion-stabilized SARS-CoV-2 spikes. *Science* **369**, 1501–1505 (2020). doi:10.1126/science.abd0826 Medline

57. D. N. Mastronarde, Automated electron microscope tomography using robust prediction of specimen movements. *J. Struct. Biol.* **152**, 36–51 (2005). doi:10.1016/j.jsb.2005.07.007 Medline

58. A. Punjani, J. L. Rubinstein, D. J. Fleet, M. A. Brubaker, cryoSPARC: Algorithms for rapid unsupervised cryo-EM structure determination. *Nat. Methods* **14**, 290–296 (2017). doi:10.1038/nmeth.4169 Medline

59. T. D. Goddard, C. C. Huang, T. E. Ferrin, Visualizing density maps with UCSF Chimera. *J. Struct. Biol.* **157**, 281–287 (2007). doi:10.1016/j.jsb.2006.06.010 Medline

60. T. C. Terwilliger, P. D. Adams, P. V. Afonine, O. V. Sobolev, A fully automatic method yielding initial models from high-resolution cryo-electron microscopy maps. *Nat. Methods* **15**, 905–908 (2018). doi:10.1038/s41592-018-0173-1 Medline

61. P. Emsley, K. Cowtan, Coot: Model-building tools for molecular graphics. *Acta Crystallogr. D Biol. Crystallogr.* **60**, 2126–2132 (2004). doi:10.1107/S0907444904019158 Medline

62. V. B. Chen, W. B. Arendall 3rd, J. J. Headd, D. A. Keedy, R. M. Immormino, G. J. Kapral, L. W. Murray, J. S. Richardson, D. C. Richardson, MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 12–21 (2010). doi:10.1107/S0907444909042073 Medline

63. E.-M. Strauch, S. M. Bernard, D. La, A. J. Bohn, P. S. Lee, C. E. Anderson, T. Nieusma, C. A. Holstein, N. K. Garcia, K. A. Hooper, R. Ravichandran, J. W. Nelson, W. Sheffler, J. D. Bloom, K. K. Lee, A. B. Ward, P. Yager, D. H. Fuller, I. A. Wilson, D. Baker, Computational design of trimeric influenza-neutralizing proteins targeting the hemagglutinin receptor binding site. *Nat. Biotechnol.* **35**, 667–671 (2017). doi:10.1038/nbt.3907 Medline

64. L. Benatuil, J. M. Perez, J. Belk, C.-M. Hsieh, An improved yeast transformation method for the generation of very large human antibody libraries. *Protein Eng. Des. Sel.* **23**, 155–159 (2010). doi:10.1093/protein/gzq002 Medline

65. M. T. Koday, J. Nelson, A. Chevalier, M. Koday, H. Kalinoski, L. Stewart, L. Carter, T. Nieusma, P. S. Lee, A. B. Ward, I. A. Wilson, A. Dagley, D. F. Smee, D. Baker, D. H. Fuller, A computationally designed hemagglutinin stem-binding protein provides in vivo protection from influenza independent of a host immune response. *PLOS Pathog.* **12**, e1005409 (2016). doi:10.1371/journal.ppat.1005409 Medline

66. D. K. W. Chu, Y. Pan, S. M. S. Cheng, K. P. Y. Hui, P. Krishnan, Y. Liu, D. Y. M. Ng, C. K. C. Wan, P. Yang, Q. Wang, M. Peiris, L. L. M. Poon, Molecular diagnosis of a novel coronavirus (2019-nCoV) causing an outbreak of pneumonia. *Clin. Chem.* **66**, 549–555 (2020). doi:10.1093/clinchem/hvaa029 Medline

67. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt; SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020). doi:10.1038/s41592-019-0686-2 Medline