Supporting Text S1: Command-line flags used for Rosetta comparative modeling.
As described in the main text, we tested four comparative modeling protocols. Command-line flags for each protocol are listed below. Variables that are meant to be supplied by the user are listed in the format {$variable_name}, and the definition of each variable is shown in **Supporting Table S2**.

Rebuild and Refine

```
-run:protocol threading
-run:shuffle
-database minirosetta_database

-in:file:alignment {$alignment_fn}
-cm:aln_format grishin

-frag3 {$frag3_fn}
-frag9 {$frag9_fn}
-loops:frag_sizes 9 3 1
-loops:frag_files {$frag9_fn} {$frag3_fn} none

-in:file:fasta {$fasta_fn}
-in:file:fullatom

-in:file:psipred_ss2   {$psipred_fn}
-in:file:fullatom

-idealize_after_loop_close
-out:file:silent_struct_type binary

-loops:extended
-loops:build_initial
-loops:remodel  quick_ccd
```

-loops:relax fastrelax
-relax:fastrelax_repeats 16
-silent_decoytime

-random_grow_loops_by 4
-select_best_loop_from 1

-in:detect_disulf false
-fail_on_bad_hbond false

-bGDT
-evaluation:gdtmm
-in:file:template_pdb {$template_pdb_fn_str}

Restrained Rebuild and Refine
Use the same flags as Rebuild and Refine and add the following:
-constraints:cst_file {$cst_file}
-constraints:cst_weight 0.1

Rebuild and Restrained Refine
Use the same flags as Rebuild and Refine and add the following:
-constraints:cst_fa_file {$cst_file}
-constraints:cst_fa_weight 0.1

Restrained Rebuild and Restrained Refine
Use the same flags as Rebuild and Refine and add the following:
-constraints:cst_file {$cst_file}
-constraints:cst_weight 0.1

-constraints:cst_fa_file {$cst_file}

-constraints:cst_fa_weight 0.1