

## Supplementary information for:

# A ‘fuzzy’-logic language for encoding multiple physical traits in biomolecules

Shira Warszawski<sup>1</sup>, Ravit Netzer<sup>1</sup>, Dan S. Tawfik<sup>1</sup>, Sarel J. Fleishman<sup>1,\*</sup>

<sup>1</sup>Department of Biological Chemistry, Weizmann Institute of Science, Rehovot 76100, Israel

\*To whom correspondence should be addressed: sarel@weizmann.ac.il

## Table of contents

<b>Supplemental Data</b> .....	<b>2</b>
<b>Parameterization of sigmoidal functions</b> .....	<b>2</b>
<b>Physical traits in biomolecular systems plateau</b> .....	<b>2</b>
<b>Multistate design with two states is a special case of the fuzzy-logic design</b> .....	<b>3</b>
<b>Supplemental Experimental Procedures</b> .....	<b>4</b>
<b>Multispecificity task</b> .....	<b>5</b>
<b>Refinement of PDB structures</b> .....	<b>5</b>
<u>Execution example</u> .....	<b>6</b>
<b>Positive design of H-RAS separately towards each of its partners</b> .....	<b>6</b>
<u>Execution example</u> .....	<b>7</b>
<b>Fuzzy-logic design</b> .....	<b>8</b>
<u>Execution example</u> .....	<b>10</b>
<u>Command line options (can be wrapped in a “flags” file)</u> .....	<b>12</b>
<b>Specificity task</b> .....	<b>12</b>
<b>Refinement of PDB structures</b> .....	<b>13</b>
<u>Execution example</u> .....	<b>13</b>
<b>Fuzzy-logic design</b> .....	<b>14</b>
<u>Execution example</u> .....	<b>16</b>
<u>Command line options (can be wrapped in a “flags” file)</u> .....	<b>16</b>
<b>Supporting Table</b> .....	<b>17</b>
<b>Supporting Figure</b> .....	<b>18</b>
<b>Supplemental References</b> .....	<b>19</b>

## Supplemental Data

### Parameterization of sigmoidal functions

To determine  $s$  and  $o$  parameters for each individual fractional-occupancy function (e.g., for binding and stability) of Eq. 6, we linearize Eq. 6 as:  $\ln\left(\frac{1-f}{f}\right) = s(x - o)$  [Eq. S1]. The linear relationship between  $\ln\left(\frac{1-f}{f}\right)$  and  $x$  can be used to determine  $s$  and  $o$  uniquely when at least two constraints relating  $x$  to  $f$  are known (examples below). In a majority of cases the  $s$  and  $o$  parameters are not known *a priori* and it might not be possible to compute them from natural structures. In those cases we recommend using  $o$  values that are within 3 – 7 Rosetta energy units (R.e.u.) of the binding energy computed for the target state and values ranging from 5 to 20 R.e.u. for stability. Assuming 1R.e.u.=0.57kcal/mole [1] a slope ( $s$ ) value of 1R.e.u.<sup>-1</sup> is equivalent to 1/RT at room temperature, and depending on the application, slopes in the range 0.4-2.5 are well behaved in simulations. As noted in the main text, in some scenarios, where the biological response is a sharp function of fractional occupancy, such as when the molecule has irreversible effects on cell fate, a step-function is desirable, and can be implemented using large slope values; yet in other cases a more gradual response discriminates more finely between models and is ultimately more useful in choosing the ones that best conform to the constraints embodied in the fuzzy-logic function.

### Physical traits in biomolecular systems plateau

Receptor-ligand affinities evolve up to a limit and protein engineering has been used to increase affinities by several orders of magnitude relative to their natural affinities [2]. Antibodies provide a particularly striking example: the dissociation constants of many naturally evolved antibodies for their targets typically peak at ~0.1nM leading to the so-called concept of an ‘affinity ceiling’ for antibodies [3,4]; yet natural antibodies can be artificially evolved in the laboratory for tighter binding, in one case by more than 3 orders of magnitude relative to the highest-affinity natural binder [5]. This contrast between natural and laboratory evolution indicates that higher antibody affinities are physico-chemically possible, but that natural affinity maturation reaches

saturation of fractional occupancy (or biological activity), above which the immune system experiences no selective pressures to increase affinity [4]. Conversely, the sigmoidal relationship of Eqs. 5 and 6 predicts that destabilizing mutations could accumulate without immediately harming fitness, whereas close to the sigmoid-inflection point ( $\Delta G = \Delta G_0$ ) any additional destabilizing mutations would severely undermine fitness (Fig. 1a); such nonlinear relationships have indeed been observed in laboratory-evolution experiments (Bloom et al., 2005; Tokuriki and Tawfik, 2009; Tokuriki et al., 2008). Thus sensitivity and robustness to mutation in diverse biomolecular systems can be readily explained by the properties of fractional occupancy at thermodynamic equilibrium.

### **Multistate design with two states is a special case of the fuzzy-logic design**

Multistate design is a widely used strategy to design energy gaps between a target state and competitor states [9]. Here we show that in certain cases the fuzzy-logic strategy described in the main text can be used to define exactly the energy gap between two states as in multistate design. Assuming two states, where one is the target and the other is its competitor, let  $x$  and  $y$  represent their free energies, respectively.

According to the multistate design the probability that a protein will adopt a target conformation or state is given by:

$$P_{\text{target}} = \frac{e^{-x/RT}}{e^{-x/RT} + e^{-y/RT}} = \frac{1}{1 + e^{(x-y)/RT}} \quad [\text{Eq. S2}]$$

Which is equivalent to the fuzzy-logic definition using Eq. 6 of the main text to define a sigmoidal transformation of the energy gap  $x - y$ :

$$f(x-y) = \frac{1}{1 + e^{(x-y)/RT}} \quad [\text{Eq. S3}]$$

Typical applications of multistate design define one positive and several negative states. If the negative states have largely different energies, only the negative state with the lowest energy makes an appreciable contribution to the denominator of the Boltzmann distribution. Hence, multistate design with more than one negative state

often resembles the two-state scenario described here. As noted in the main text, the fuzzy-logic framework provides advantages over multistate design.

## **Supplemental Experimental Procedures**

We present here all of the computational protocols used to generate the results described in the manuscript. Each script is accompanied by a short explanation, command-line, flags, and a usage example. For additional information regarding RosettaScripts and implementation we refer the reader to the RosettaScripts documentation page on the RosettaCommons website. (<https://www.rosettacommons.org/docs/latest/RosettaScripts.html>)

### ***Source code availability***

The methods have been implemented within the Rosetta macromolecular modeling software suite [10] and are available through the Rosetta Commons agreement. All of the methods have been implemented through RosettaScripts [11], and all scripts are available here.

### ***Protein Data Bank structures***

Design of multispecificity of H-RAS started with Protein Data Bank (PDB) entries: 1BKD (H-RAS,Son of Sevenless-I) , 1WQ1 (H-RAS,RAS-GAP), 1HE8 (H-RAS,PI-3 Kinase), 1K8R (H-RAS,BRY-2RBD) and 1LFD (H-RAS,Ral-GDS).

Analysis of specificity in the colicin immunity-endonuclease complexes used PDB entries: cognate pairs: 1EMV (E9,Im9) and 3U43 (E2,Im2) and the non-cognate pair 2WPT (E9,Im2).

### ***Energy function***

All energy calculations were done using the Talaris 2013 all-atom energy function encoded in Rosetta, which is dominated by contributions from hydrogen bonding, van der Waals packing, and solvation. This energy function is distributed in Rosetta (git RosettaCommons/main @ 2fac63a) and used in combination with the customized “talaris.wts” score weights.

### ***Fuzzy-logic multispecificity design***

Two design structures with different folds, designed H-RAS,SOS1 and designed H-RAS,Ral-GDS were used as two independent starting points. The designed H-RAS sequence was subjected to 4000 design trajectories each comprising 3 trials at high temperature, followed by 100 trials at medium temperature, and ending with 50 trials at low temperature, producing a simulated annealing effect. In this process a random point mutation was introduced in one of the eight allowed positions. Then this new sequence was threaded on the backbone of each complex state, followed by rigid-body minimization and sidechain combinatorial packing and minimization of surrounding residues. Next, the binding energy of the complex and the stability energy of mutated H-RAS monomer for each one of the states were evaluated. These were transformed to a sigmoidal function (offset ( $o$ ) and slope ( $s$ ) values are specified at the [fuzzy-logic design](#) of multispecificity) that were then combined using the logic operator AND to generate an optimization objective function (Eq. 9). The logarithm of the objective function was subjected to simulated annealing Monte Carlo optimization of sequence where a mutation is accepted according to the Metropolis criterion (Figure S1).

### ***Fuzzy-logic specificity design***

The relaxed 1EMV structure was subjected to a similar procedure as was described above for the fuzzy-logic design of multispecificity.

The sigmoidal function parameters (offset and slope values are specified at the [fuzzy-logic design](#) of specificity) were then combined using the logic operator AND and NOT to generate an optimization objective function (e.g. Eq. 10).

## **Multispecificity task**

We substituted the oncogenic valine at position 12 in H-RAS,PI3K (PDB entry 1HE8) to the consensus amino acid glycine.

### **Refinement of PDB structures**

A preliminary step in the protocol starts by relieving minor clashes that are often observed in molecular structures in the PDB.

This script is used for packing and minimizing the rigid-body orientation and sidechains in a two-chain PDB structure.

### **refine\_two\_chain\_pdb.xml (explanation)**

```
<dock_design>
  <TASKOPERATIONS>
    <InitializeFromCommandline name=init/>
  </TASKOPERATIONS>
  <SCOREFXNS>
    <tal weights=talaris2013_calibrated/>
  </SCOREFXNS>
  <FILTERS>
    <Ddg name=ddg confidence=0 scorefxn=tal/>
    <Sasa name=sasa confidence=0/>
    <Rmsd name=rms confidence=0/>
  </FILTERS>
  <MOVERS>
    <Prepack name=prepack jump_number=0 scorefxn=tal task_operations=init/>
    <MinMover name=min bb=0 chi=1 jump=1 scorefxn=tal/>
  </MOVERS>
  <PROTOCOLS>
    <Add mover_name=prepack/>
    <Add mover_name=min/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
    <Add filter_name=rms/>
  </PROTOCOLS>
</dock_design>
```

### **Execution example**

```
rosetta_scripts.default.linuxgccrelease -s 1bkd.pdb
-parser:protocol refine_two_chain_pdb.xml @flags
```

### **Positive design of H-RAS separately towards each of its partners**

Positive design of H-RAS separately towards each of its partners leads to sequences that increase stability and binding energy towards that particular binder. For each relaxed PDB structure 8 residues on the binding surface were subjected to design. In the design process these amino acids were allowed to vary to any amino acid except for proline, cysteine and histidine, these and the neighboring residues were subjected to combinatorial packing, rotamer optimization and minimization. This procedure was repeated 10 times, producing 10 independent trajectories, and the structure with the lowest binding and stability energies was selected to the next step of fuzzy-logic design.

## **positive\_design.xml (explanation)**

```
<dock_design>
  <TASKOPERATIONS>
    <InitializeFromCommandline name=init/>
    <ProteinInterfaceDesign name=pido design_chain1=1 design_chain2=0
interface_distance_cutoff=6/>
    <RestrictAbsentCanonicalAAS name= no_cys_his
keep_aas="ADEFGIKLMNQRSTVWY"/>
    <DesignAround name=design_residues resnums=21,25,36,37,38,39,61,63
design_shell=0 repack_shell=8.0/>
    <RestrictToRepacking name=rtr/>
  </TASKOPERATIONS>
  <SCOREFXNS>
    <tal weights=talaris2013_calibrated/>
  </SCOREFXNS>
  <FILTERS>
    <Ddg name=ddg confidence=0 scorefxn=tal repeats=3/>
    <Rmsd name=rmsd superimpose=1 threshold=100/>
    <ScoreType name=score_f scorefxn=tal score_type=total_score threshold=1000/>
  </FILTERS>
  <MOVERS>
    <PackRotamersMover name=design task_operations=design_residues,init,
no_cys_his scorefxn=tal/>
    <MinMover name=min scorefxn=tal bb=0 jump=1 chi=1/>
    <AtomTree name=docking_ft docking_ft=1/>
    <RotamerTrialsMinMover name=rtmin scorefxn=tal
task_operations=rtr,design_residues,init, no_cys_his />
  </MOVERS>
  <PROTOCOLS>
    <Add mover=docking_ft/>
    <Add mover=design/>
    <Add mover=rtmin/>
    <Add mover=rtmin/>
    <Add mover=rtmin/>
    <Add mover=min/>
    <Add filter=ddg/>
    <Add filter=rmsd/>
    <Add filter=score_f/>
  </PROTOCOLS>
</dock_design>
```

### **Execution example**

```
rosetta_scripts.default.linuxgccrelease -s 1bkd\_refined.pdb
-parser:protocol positive_design.xml @flags
```

## Fuzzy-logic design

See [Figure S1](#).

### fuzzyDesign\_1bkd.xml (explanation)

```
<dock_design>
  <SCOREFXNS>
    <tal weights=talaris2013_calibrated/>
  </SCOREFXNS>
  <TASKOPERATIONS>
    <InitializeFromCommandline name=init/>
    <DesignAround name=design_positions
resnums=21A,25A,36A,37A,38A,39A,61A,63A design_shell=0.01/>
    <RestrictAbsentCanonicalAAS name=no_cys_his
keep_aas="ADEFIJKLMNQRSTVWY"/>
    <RestrictToRepacking name=rtr/>
  </TASKOPERATIONS>
  <MOVERS>
    <AtomTree name=docking_ft docking_ft=1/>
    <MinMover name=hard_min scorefxn=tal bb=0 jump=1 chi=1 scorefxn=tal/>
    <RotamerTrialsMinMover name=rtmin task_operations=rtr,design_positions
scorefxn=tal/>
    <SwitchChainOrder name=chain1 chain_order=1 scorefxn=tal/>
    <ParsedProtocol name=refine_bound>
      <Add mover=docking_ft/>
      <Add mover=rtmin/>
      <Add mover=rtmin/>
      <Add mover=hard_min/>
    </ParsedProtocol>
    <ParsedProtocol name=refine_unbound>
      <Add mover=rtmin/>
      <Add mover=rtmin/>
      <Add mover=hard_min/>
    </ParsedProtocol>
  </MOVERS>
  <FILTERS>
    <ScoreType name=total_score scorefxn=tal score_type=total_score threshold=100/>
    <Ddg name=ddg scorefxn=tal repeats=3 confidence=0/>
    <MoveBeforeFilter name=score_chain1 mover=chain1 filter=total_score/>
    <RelativePose name=ddg_1k8r pdb_name="1k8r_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
    <RelativePose name=ddg_1lfd pdb_name="1lfd_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
    <RelativePose name=ddg_1he8 pdb_name="1he8_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
    <RelativePose name=ddg_1wq1 pdb_name="1wq1_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
    <RelativePose name=score_1k8r pdb_name="1k8r_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
    <RelativePose name=score_1lfd pdb_name="1lfd_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
    <RelativePose name=score_1he8 pdb_name="1he8_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
```



```

    <RelativePose name=score_1wq1 pdb_name="1wq1_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
    <Sigmoid name=1bkd_ddg_sig steepness=0.7 offset=-45 negate=0 filter=ddg/>
    <Sigmoid name=1he8_ddg_sig steepness=1 offset=-9 negate=0 filter=ddg_1he8/>
    <Sigmoid name=1k8r_ddg_sig steepness=1.5 offset=-13 negate=0 filter=ddg_1k8r/>
    <Sigmoid name=1fd_ddg_sig steepness=0.6 offset=-15 negate=0 filter=ddg_1fd/>
    <Sigmoid name=1wq1_ddg_sig steepness=0.6 offset=-16 negate=0
filter=ddg_1wq1/>
    <Sigmoid name=sig_score_1bkd filter=score_chain1 steepness=0.6 offset=-190
negate=0/>
    <Sigmoid name=sig_score_1he8 filter=score_1he8 steepness=0.4 offset=-120
negate=0/>
    <Sigmoid name=sig_score_1k8r filter=score_1k8r steepness=0.6 offset=-120
negate=0/>
    <Sigmoid name=sig_score_1fd filter=score_1fd steepness=0.5 offset=-215
negate=0/>
    <Sigmoid name=sig_score_1wq1 filter=score_1wq1 steepness=0.3 offset=-140
negate=0/>
    <Operator name=objective_function
filters=1he8_ddg_sig,1k8r_ddg_sig,1fd_ddg_sig,1wq1_ddg_sig,1bkd_ddg_sig,sig_score_1he8,sig_sco
re_1bkd,sig_score_1k8r,sig_score_1fd,sig_score_1wq1 operation=PRODUCT negate=1 logarithm=1
threshold=10000000/>
    </FILTERS>
    <MOVERS>
    <RandomMutation name=mut task_operations=design_positions,init, no_cys_his
scorefxn=tal/>
    <ParsedProtocol name=mut_min>
    <Add mover=mut/>
    <Add mover=rtmin/>
    <Add mover=hard_min/>
    </ParsedProtocol>
    <GenericMonteCarlo name=mut_mc1 mover_name=mut_min preapply=0 drift=1
filter_name=objective_function reset_baselines=0 trials=3 temperature=0.2/>
    <GenericMonteCarlo name=mut_mc2 mover_name=mut_min preapply=0 drift=1
filter_name=objective_function reset_baselines=0 trials=100 temperature=0.05/>
    <GenericMonteCarlo name=mut_mc3 mover_name=mut_min preapply=0 drift=1
filter_name=objective_function reset_baselines=0 trials=50 temperature=0.015/>
    </MOVERS>
    <PROTOCOLS>
    <Add mover=docking_ft/>
    <Add mover=mut_mc1/>
    <Add mover=mut_mc2/>
    <Add mover=mut_mc3/>
    <Add filter=objective_function/>
    <Add filter=score_chain1/>
    <Add filter=ddg/>
    <Add filter=ddg_1k8r/>
    <Add filter=ddg_1wq1/>
    <Add filter=ddg_1fd/>
    <Add filter=ddg_1he8/>
    <Add filter=1k8r_ddg_sig/>
    <Add filter=1wq1_ddg_sig/>
    <Add filter=1fd_ddg_sig/>
    <Add filter=1bkd_ddg_sig/>
    <Add filter=1he8_ddg_sig/>

```

```

        <Add filter=sig_score_1he8/>
        <Add filter=sig_score_1wq1/>
        <Add filter=sig_score_1lfd/>
        <Add filter=sig_score_1bkd/>
        <Add filter=sig_score_1k8r/>
        <Add filter=score_1he8/>
        <Add filter=score_1wq1/>
        <Add filter=score_1k8r/>
        <Add filter=score_1lfd/>
    </PROTOCOLS>
</dock_design>

```

### **Execution example**

rosetta\_scripts.default.linuxgccrelease -s [1bkd\\_designed.pdb](#) -parser:protocol fuzzyDesign\_1bkd.xml @[flags](#)

### **fuzzyDesign\_1lfd.xml (explanation)**

```

<dock_design>
  <SCOREFXNS>
    <tal weights=talaris2013_calibrated/>
  </SCOREFXNS>
  <TASKOPERATIONS>
    <InitializeFromCommandline name=init/>
    <DesignAround name=design_positions
resnums=21A,25A,36A,37A,38A,39A,61A,63A design_shell=0.01/>
    <RestrictAbsentCanonicalAAS name= no_cys_his
keep_aas="ADEFIJKLMNQRSTVWY"/>
    <RestrictToRepacking name=rtr/>
  </TASKOPERATIONS>
  <MOVERS>
    <AtomTree name=docking_ft docking_ft=1/>
    <MinMover name=hard_min scorefxn=tal bb=0 jump=1 chi=1 scorefxn=tal/>
    <MinMover name=hard_bb_min scorefxn=tal bb=1 jump=1 chi=1 scorefxn=tal/>
    <RotamerTrialsMinMover name=rtmin task_operations=rtr,design_positions
scorefxn=tal/>
    <SwitchChainOrder name=chain1 chain_order=1 scorefxn=tal/>
    <ParsedProtocol name=refine_bound>
      <Add mover=docking_ft/>
      <Add mover=rtmin/>
      <Add mover=rtmin/>
      <Add mover=hard_min/>
    </ParsedProtocol>
    <ParsedProtocol name=refine_unbound>
      <Add mover=rtmin/>
      <Add mover=rtmin/>
      <Add mover=hard_min/>
    </ParsedProtocol>
  </MOVERS>
</FILTERS>
  <ScoreType name=total_score scorefxn=tal score_type=total_score threshold=100/>
  <Ddg name=ddg scorefxn=tal repeats=3 confidence=0/>
  <MoveBeforeFilter name=score_chain1 mover=chain1 filter=total_score/>

```

```

                <RelativePose name=ddg_1bkd pdb_name="1bkd_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
                <RelativePose name=ddg_1k8r pdb_name="1k8r_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
                <RelativePose name=ddg_1he8 pdb_name="1he8_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
                <RelativePose name=ddg_1wq1 pdb_name="1wq1_refined.pdb"
relax_mover=refine_bound alignment=A:A scorefxn=tal baseline=0 filter=ddg/>
                <RelativePose name=score_1bkd pdb_name="1bkd_refined.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
                <RelativePose name=score_1k8r pdb_name="1k8r_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
                <RelativePose name=score_1he8 pdb_name="1he8_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
                <RelativePose name=score_1wq1 pdb_name="1wq1_refined_ras.pdb"
relax_mover=refine_unbound alignment=A:A scorefxn=tal baseline=0 filter=total_score/>
                <Sigmoid name=1bkd_ddg_sig steepness=0.7 offset=-45 negate=0 filter=ddg_1bkd/>
                <Sigmoid name=1he8_ddg_sig steepness=2.9 offset=-11 negate=0 filter=ddg_1he8/>
                <Sigmoid name=1k8r_ddg_sig steepness=1 offset=-12 negate=0 filter=ddg_1k8r/>
                <Sigmoid name=1fd_ddg_sig steepness=1.5 offset=-22 negate=0 filter=ddg/>
                <Sigmoid name=1wq1_ddg_sig steepness=1.5 offset=-14 negate=0
filter=ddg_1wq1/>
                <Sigmoid name=sig_score_1bkd filter=score_1bkd steepness=0.4 offset=-180
negate=0/>
                <Sigmoid name=sig_score_1he8 filter=score_1he8 steepness=0.4 offset=-125
negate=0/>
                <Sigmoid name=sig_score_1k8r filter=score_1k8r steepness=0.4 offset=-125
negate=0/>
                <Sigmoid name=sig_score_1fd filter=score_chain1 steepness=0.6 offset=-235
negate=0/>
                <Sigmoid name=sig_score_1wq1 filter=score_1wq1 steepness=0.5 offset=-145
negate=0/>
                <Operator name=objective_function
filters=1he8_ddg_sig,1k8r_ddg_sig,1fd_ddg_sig,1wq1_ddg_sig,1bkd_ddg_sig,sig_score_1he8,sig_sco
re_1bkd,sig_score_1k8r,sig_score_1fd,sig_score_1wq1 operation=PRODUCT negate=1 logarithm=1
threshold=10000000/>
                </FILTERS>
                <MOVERS>
                <RandomMutation name=mut task_operations=design_positions,init, no_cys_his
scorefxn=tal/>
                <ParsedProtocol name=mut_min>
                <Add mover=mut/>
                <Add mover=rtmin/>
                <Add mover=hard_min/>
                </ParsedProtocol>
                <GenericMonteCarlo name=mut_mc1 mover_name=mut_min preapply=0 drift=1
filter_name=objective_function reset_baselines=0 trials=3 temperature=0.2/>
                <GenericMonteCarlo name=mut_mc2 mover_name=mut_min preapply=0 drift=1
filter_name=objective_function reset_baselines=0 trials=100 temperature=0.05/>
                <GenericMonteCarlo name=mut_mc3 mover_name=mut_min preapply=0 drift=1
filter_name=objective_function reset_baselines=0 trials=50 temperature=0.015/>
                </MOVERS>
        <PROTOCOLS>
                <Add mover=docking_ft/>
                <Add mover=mut_mc1/>

```

```

    <Add mover=mut_mc2/>
    <Add mover=mut_mc3/>
    <Add filter=objective_function/>
    <Add filter=score_chain1/>
    <Add filter=ddg/>
    <Add filter=ddg_1k8r/>
    <Add filter=ddg_1wq1/>
    <Add filter=ddg_1bkd/>
    <Add filter=ddg_1he8/>
    <Add filter=1k8r_ddg_sig/>
    <Add filter=1wq1_ddg_sig/>
    <Add filter=1lfd_ddg_sig/>
    <Add filter=1bkd_ddg_sig/>
    <Add filter=1he8_ddg_sig/>
    <Add filter=sig_score_1he8/>
    <Add filter=sig_score_1wq1/>
    <Add filter=sig_score_1lfd/>
    <Add filter=sig_score_1bkd/>
    <Add filter=sig_score_1k8r/>
    <Add filter=score_1he8/>
    <Add filter=score_1wq1/>
    <Add filter=score_1k8r/>
    <Add filter=score_1bkd/>
  </PROTOCOLS>
</dock_design>

```

Command line options (can be wrapped in a “flags” file)

```

-ex1
-ex2
-use_input_sc
-extrachi_cutoff 0
-ignore_unrecognized_res
-database Rosetta/main/database
-chemical:exclude_patches LowerDNA UpperDNA Cterm_amidation
SpecialRotamer protein_cutpoint_upper protein_cutpoint_lower VirtualBB ShoveBB
VirtualDNAPhosphate VirtualNTerm CTermConnect sc_orbitals
pro_hydroxylated_case1 pro_hydroxylated_case2 ser_phosphorylated
thr_phosphorylated tyr_phosphorylated tyr_sulfated lys_dimethylated
lys_monomethylated lys_trimethylated lys_acetylated glu_carboxylated
cys_acetylated tyr_diiodinated N_acetylated C_methylamidated
MethylatedProteinCterm
-out:file:silent_struct_type binary
-score:weights talaris2013_calibrated
-mute all

```

## Specificity task

Only the aligned region of endonuclease-immunity proteins 9 and 2 structures (PDB entries: 1EMV and 3U43, respectively) was modeled comprising residues 5-85 in chain A, and 3-131 in chain B.

## **Refinement of PDB structures**

A preliminary step in the protocol starts by relieving minor clashes that are often observed in molecular structures in the PDB.

This script is used for packing, local docking and minimizing the rigid-body orientation and sidechains in a two-chain PDB structure.

### **refine two chain pdb.xml (explanation)**

```
<dock_design>
  <SCOREFXNS>
    <tal weights=talaris2013_calibrated/>
  </SCOREFXNS>
  <FILTERS>
    <Ddg name=ddg confidence=0 scorefxn=tal/>
    <Sasa name=sasa confidence=0/>
    <Rmsd name=rms confidence=0/>
  </FILTERS>
  <MOVERS>
    <Prepack name=prepack jump_number=0 scorefxn=tal/>
    <Docking name=dock1 fullatom=1 local_refine=1 score_high=tal/>
    <MinMover name=min bb=1 chi=1 jump=1 scorefxn=tal>
      <MoveMap>
        <Chain number=1 bb=1 chi=1/> minimize immunity backbone
        <Chain number=2 bb=0 chi=1/>
      </MoveMap>
    </MinMover>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=prepack/>
    <Add mover_name=dock1/>
    <Add mover_name=prepack/>
    <Add mover_name=min/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
    <Add filter_name=rms/>
  </PROTOCOLS>
</dock_design>
```

### **Execution example**

```
rosetta_scripts.default.linuxgccrelease -s 1emv_cut_AB.pdb
```

-parser:protocol refine\_two\_chain\_pdb.xml @[flags](#)

### **Fuzzy-logic design**

A total of nine positions at the binding surfaces of the immunity and endonuclease 9 structure (PDB entry: 1EMV), four on the immunity (Thr27, Leu33, Val34, Thr38) and five on the endonuclease (Ser77, Ser78, Tyr83, Lys97, Val98) were subjected to design.

### **fuzzyDesign.xml (explanation)**

```
<dock_design>
  <SCOREFXNS>
    <tal weights=talaris2013_calibrated/>
  </SCOREFXNS>
  <TASKOPERATIONS>
    <InitializeFromCommandline name=init/>
    <DesignAround name=design_positions design_shell=0.01
resnums=27A,33A,34A,38A,77B,78B,83B,97B,98B/>
    <RestrictToRepacking name=rtr/>
    <RestrictAbsentCanonicalAAS name=no_cys_his
keep_aas="ADEFGIKLMNQRSTVWY"/>
    <RestrictAbsentCanonicalAAS name=small_aas27 resnum=23
keep_aas="AGSTV"/>
  </TASKOPERATIONS>
  <MOVERS>
    <AtomTree name=docking_tree docking_ft=1/>
    <MinMover name=min_all scorefxn=tal chi=1 bb=0 jump=1>
      <MoveMap name=minbb_chain1>
        <Chain number=1 chi=1 bb=1/>
      </MoveMap>
    </MinMover>
    <RotamerTrialsMinMover name=RTmin scorefxn=tal
task_operations=design_positions,rtr/>
    <ParsedProtocol name=relax_before_baseline>
      <Add mover=docking_tree/>
      <Add mover=RTmin/>
      <Add mover=RTmin/>
      <Add mover=min_all/>
    </ParsedProtocol>
    <SwitchChainOrder name=chain1 chain_order=1 scorefxn=tal/>
    <SwitchChainOrder name=chain2 chain_order=2 scorefxn=tal/>
  </MOVERS>
  <FILTERS>
```

```

        <ScoreType name=total_score scorefxn=tal score_type=total_score
threshold=0.0/>
        <Ddg name=ddg scorefxn=tal repeats=1 confidence=0/>
        <Ddg name=ddg_I9mut_E9mut scorefxn=tal repeats=1 confidence=0/>
        <RelativePose name=ddg_I9mut_E9 baseline=0
pdb_name="1emv_cut_AB_refined.pdb" filter=ddg relax_mover=relax_before_baseline
scorefxn=tal alignment=A:A/>
        <RelativePose name=ddg_I9_E9mut baseline=0
pdb_name="1emv_cut_AB_refined.pdb" filter=ddg relax_mover=relax_before_baseline
scorefxn=tal alignment=B:B/>
        <MoveBeforeFilter name=score_chain1 mover=chain1 filter=total_score/>
        <MoveBeforeFilter name=score_chain2 mover=chain2 filter=total_score/>
        <MoveBeforeFilter name=score_chain1_confidence0 mover=chain1
filter=total_score confidence=0/>
        <MoveBeforeFilter name=score_chain2_confidence0 mover=chain2
filter=total_score confidence=0/>
        <Sigmoid name=I9mut_stability filter=score_chain1 steepness=1 offset=10
negate=0/>
        <Sigmoid name=E9mut_stability filter=score_chain2 steepness=1 offset=6
negate=0/>
        <Sigmoid name=I9mut_E9mut_binding filter=ddg_I9mut_E9mut steepness=1
offset=4 negate=0/>
        <Sigmoid name=I9mut_E9_binding filter=ddg_I9mut_E9 steepness=1 offset=5
negate=1/>
        <Sigmoid name=I9_E9mut_binding filter=ddg_I9_E9mut steepness=1 offset=5
negate=1/>
        <Operator name=objective_function
filters=I9mut_stability,E9mut_stability,I9mut_E9mut_binding,I9mut_E9_binding,I9_E9mut_bindi
ng operation=PRODUCT negate=1 logarithm=1 threshold=100000/>
    </FILTERS>
    <MOVERS>
        <RandomMutation name=mut
task_operations=design_positions,init,no_cys_his,small_aas27 scorefxn=tal/>
        <ParsedProtocol name=mut_min>
            <Add mover=mut/>
            <Add mover=RTmin/>
            <Add mover=min_all/>
        </ParsedProtocol>
        <GenericMonteCarlo name=mut_mc1 mover_name=mut_min preapply=0
drift=1 filter_name=objective_function trials=3 temperature=0.2/>
        <GenericMonteCarlo name=mut_mc2 mover_name=mut_min preapply=0
drift=1 filter_name=objective_function reset_baselines=0 trials=100 temperature=0.05/>
        <GenericMonteCarlo name=mut_mc3 mover_name=mut_min preapply=0
drift=1 filter_name=objective_function reset_baselines=0 trials=50 temperature=0.00/>
    </MOVERS>
    <PROTOCOLS>
        <Add mover=docking_tree/>
        <Add mover=mut_mc1/>
        <Add mover=mut_mc2/>
        <Add mover=mut_mc3/>

```

```

    <Add mover=RTmin/>
    <Add mover=RTmin/>
    <Add mover=min_all/>
    <Add filter=ddg_I9mut_E9mut/>
    <Add filter=ddg_I9mut_E9/>
    <Add filter=ddg_I9_E9mut/>
    <Add filter=score_chain1_confidence0/>
    <Add filter=score_chain2_confidence0/>
    <Add filter=I9mut_E9mut_binding/>
    <Add filter=I9mut_E9_binding/>
    <Add filter=I9_E9mut_binding/>
    <Add filter=I9mut_stability/>
    <Add filter=E9mut_stability/>
    <Add filter=objective_function/>
  </PROTOCOLS>
</dock_design>

```

### ***Execution example***

```

rosetta_scripts.default.linuxgccrelease -s 1emv_cut_AB_refined.pdb
-parser:protocol fuzzyDesign.xml @flags

```

Command line options (can be wrapped in a “flags” file)

```

-use_input_sc
-extrachi_cutoff 5
-ignore_unrecognized_res
-database /home/labs/fleishman/sarel/rosetta/main/database/
-chemical:exclude_patches LowerDNA UpperDNA Cterm_amidation
SpecialRotamer VirtualBB ShoveBB VirtualDNAPhosphate VirtualNTerm
CTermConnect sc_orbitals pro_hydroxylated_case1 pro_hydroxylated_case2
ser_phosphorylated thr_phosphorylated tyr_phosphorylated tyr_sulfated
lys_dimethylated lys_monomethylated lys_trimethylated lys_acetylated
glu_carboxylated cys_acetylated tyr_diiodinated N_acetylated C_methylamidated
MethylatedProteinCterm
-out:file:silent_struct_type binary
-score:weights talaris2013_calibrated
-mute all

```

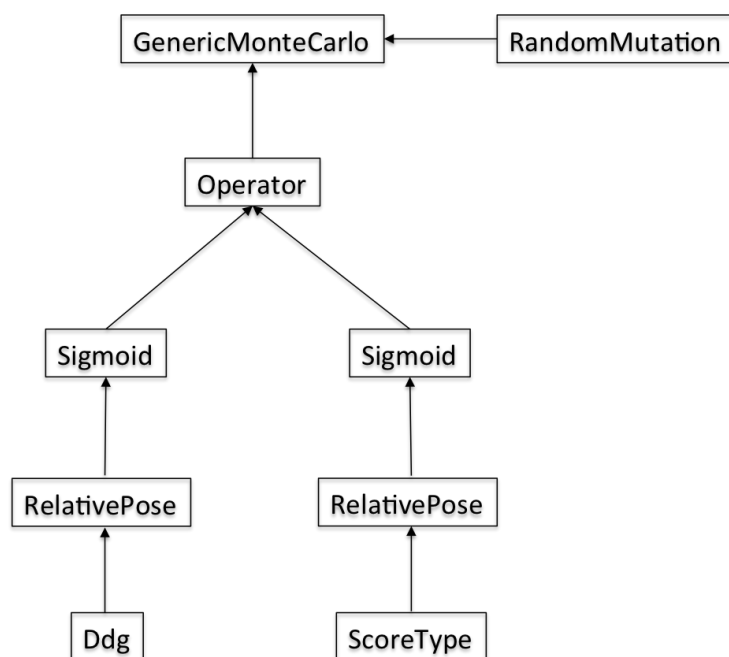


## Supporting Table

Threaded on Designed protein sequence		<b>SOS1</b>	<b>RAS-GAP</b>	<b>PI3K</b>	<b>BRY-2RBD</b>	<b>Ral-GDS</b>
<b>Natural</b>	<b>Stability</b>	<b>-195</b>	<b>-149</b>	<b>-127</b>	<b>-125</b>	<b>-221</b>
	<b>Binding</b>	<b>-49</b>	<b>-16</b>	<b>-12</b>	<b>-15</b>	<b>-20</b>
<b>SOS1</b>	$\Delta$ Stability	<b>-6</b>	+1	+3	+1	+2
	$\Delta$ Binding	<b>0</b>	-5	+3	+2	+5
<b>RAS-GAP</b>	$\Delta$ Stability	+24	<b>-6</b>	-6	-2	-15
	$\Delta$ Binding	+1	<b>-7</b>	-1	0	+2
<b>PI3K</b>	$\Delta$ Stability	+173	-1	<b>-6</b>	-8	-5
	$\Delta$ Binding	+11	-3	<b>-4</b>	-4	+2
<b>BRY-2RBD</b>	$\Delta$ Stability	+126	-5	-3	<b>-11</b>	-5
	$\Delta$ Binding	+9	-7	-3	<b>-4</b>	0
<b>Ral-GDS</b>	$\Delta$ Stability	+8	-2	-6	-8	<b>-19</b>
	$\Delta$ Binding	+15	+1	+1	0	<b>-2</b>

**Table S1:** Designing H-RAS to bind any of its natural partners separately leads, in almost all cases, to an improvement in both stability and binding affinity in the target complex but to destabilization of one or more of the others. All values are given in Rosetta energy units (R.e.u). For each complex the H-RAS binding surface was sequence optimized to bind to one of its partners using RosettaDesign [10]. This sequence was then threaded on to the other H-RAS complexes and the stability and binding energies were compared to the values of the original complex structure, namely  $\Delta$ Stability and  $\Delta$ Binding respectively.

## Supporting Figure



**Figure S1: Code architecture for the fuzzy-logic design framework.** The fuzzy-logic framework is implemented as a set of filters within RosettaScripts [11] and can interface with existing filters or with new filters yet to be written. Most filters in RosettaScripts report rational numbers, such as free energies. In the example provided in the figure, the bottom-most layer comprises the RosettaScripts filters, Ddg and ScoreType, which report on the binding energy and the protein stability, respectively. The RelativePose filter keeps a static record of a starting pose (read from disk in PDB format); it then applies to this copy the mutations observed in the pose currently being optimized (for instance, mutations at the binding surface), relaxes the copy-pose, and applies the relevant filter (Ddg or ScoreType) to this copy-pose; values from these filters are then transferred to the Sigmoid filters, which implement the fractional occupancy (Eq. 6 of the main text). Each Sigmoid filter is defined with user-determined slope and offset parameters. The Sigmoid filter further defines a parameter, negate, which if applied, takes the Boolean complement of the Sigmoid value, e.g., if the Sigmoid evaluates to  $x$  then negate produces  $1 - x$ ; Negate is used to

implement the NOT operator in the design of specificity. Values from the Sigmoid filters are combined into a fuzzy-logic statement using the Operator filter, which defines the operators (AND and OR) to be applied to the combined Sigmoid values. Operator filters can be combined hierarchically within other Operator filters, providing a complete framework to define any combination of Boolean operators acting on computed fractional occupancies. Finally, the Operator filter is fed into the GenericMonteCarlo mover. At each iteration, GenericMonteCarlo invokes the RandomMutation mover, which defines the set of allowed positions and the amino acid identities allowed at each position. When RandomMutation Mover is invoked a random point mutation is introduced according to the user-defined definitions for allowed positions and identities and side chains with 6Å of the mutation are subjected to combinatorial repacking. GenericMonteCarlo then calls the Operator filter, and compares the value obtained from Operator filter with the value of the currently accepted pose. GenericMonteCarlo also keeps in memory the pose with the best Operator evaluation and at the end of the trajectory returns this best pose. To implement a simulated annealing policy we break the Monte Carlo simulation into three separate Monte Carlo steps starting with high temperature and ending with zero temperature. Complete RosettaScripts implementations for the trajectories reported in the paper are available below. Further implementation details and updates are available in the RosettaCommons website.

## Supplemental References

- [1] Kellogg EH, Leaver-Fay A, Baker D. Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins* 2011;79:830–8.
- [2] Lowman HB WJ. Affinity maturation of human growth hormone by monovalent phage display. *J Mol Biol* 1993;234:564–78.
- [3] Batista FD, Neuberger MS. Affinity dependence of the B cell response to antigen: a threshold, a ceiling, and the importance of off-rate. *Immunity* 1998;8:751–9.
- [4] Foote J, Eisent HN. Commentary Kinetic and affinity limits on antibodies produced during immune responses. *Proc Natl Acad Sci U S A* 1995;92:1254–6.
- [5] Midelfort KS, Hernandez HH, Lippow SM, Tidor B, Drennan CL, Witttrup KD. Substantial energetic improvement with minimal structural perturbation in a high affinity mutant antibody. *J Mol Biol* 2004;343:685–701.

- [6] Tokuriki N, Tawfik DS. Stability effects of mutations and protein evolvability. *Curr Opin Struct Biol* 2009;19:596–604.
- [7] Tokuriki N, Stricher F, Serrano L, Tawfik DS. How protein stability and new functions trade off. *PLoS Comput Biol* 2008;4:e1000002.
- [8] Bloom JD, Silberg JJ, Wilke CO, Drummond DA, Adami C, Arnold FH. Thermodynamic prediction of protein neutrality. *Proc Natl Acad Sci U S A* 2005;102:606–11.
- [9] Havranek JJ, Harbury PB. Automated design of specificity in molecular recognition. *Nat Struct Biol* 2003;10:45–52.
- [10] Das R, Baker D. Macromolecular modeling with rosetta. *Annu Rev Biochem* 2008;77:363–82.
- [11] Fleishman SJ, Leaver-Fay A, Corn JE, Strauch E-M, Khare SD, Koga N, et al. RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite. *PLoS One* 2011;6:e20161.