SUPPLEMENTARY COMPUTATIONAL METHODS

MULTIGRAFT INTERFACE

The design of epitope-scaffolds that interact with the CDR H3 loop of 2F5 was done in the Rosetta Multigraft version of Rosetta++ (revision 45510) available for free to non-commercial users from the rosettacommons.org website.

The stages of the protocol and the command line options are as follows:

1. Matching to identify potential scaffolds in the PDB that may accommodate the epitope and interact with the CDR H3 loop. The matching is done with an Alanine/Glycine version of the CDR H3 loop (leave all glycine residues as glycine, mutate all others to alanine) and with the full-atom epitope:

```
rosetta_executable -epi_graft -match -nres_Ab $Ab_RESIDUES -Ab_first
-native_complex $NATIVE_COMPLEX -loop_ranges $EPITOPE_RANGES -keep_natro
$KEEP_NATRO -input_file $INPUT_FILE -paths $PATHS -max_closure_rms 3
-max_match_width_delta 30    -max_intra_clash 100  -max_inter_clash 100 -spatial_filter
$CDR_H3_ALA_GLY     -spatial_filter_min_cbeta 2    -spatial_filter_max_cbeta 10
-compute_cbeta_neighbors    -output_file $OUTPUT
-use_epitope_sidechain_during_clash_check
```

For the design of the epitope-scaffolds described in this paper:

$Ab_RESIDUES =238

$NATIVE_COMPLEX is defined as a pdb file containing the crystal structure of 2F5 mAb and its epitope on gp41 on HIV (PDBid:1tji). The mAb was described as the first chain in the pdb file (-Ab_first option).

$EPITOPE_RANGES is defined as a file containing the sub-ranges of the full epitope to be used during the matches stage with the following format:

```
loop: 1
full_range: 657 670
nranges: 12

range: 664 667
range: 663 667
range: 662 667
range: 661 667
```

```
range: 660 667
range: 663 666
range: 662 666
range: 661 666
range: 660 666
range: 660 669
range: 661 669
range: 662 669
```

$KEEP_NATRO is a text file specifying the epitope residues that are to be kept in their native rotomer conformation (otherwise the entire range of epitope loop is taken in its native rotamer form) with the following format:

```
# epitope residue number to keep NATRO
664
665
666
```

$INPUT_FILE is defined as a text file containing the PDBid of the PDB structures used as scaffold candidates.

$CDR_H3_ALA_GLY is defined as a pdb file containing an Alanine/Glycine version of the 2F5 CDR H3 loop defined by the ^{100A}LFGVPI^{100F} polypeptide region.

Command line options:

'-max_match_width_delta' : the maximum residue length of a polypeptide region on the scaffold that may be replaced by the epitope
'-max_intra_clash' : the highest accepted Rosetta energy value calculated between the epitope and the scaffold
'-max_inter_clash' : the highest accepted Rosetta energy calculated between the antibody and the scaffold
'-spatial_filter_min_cbeta': the minimum number of scaffold Cβ atoms neighboring the *$CDR_H3_ALA_GLY* accepted in a match
'-spatial_filter_max_cbeta' : the maximum number of scaffold Cβ atoms neighboring the *$CDR_H3_ALA_GLY* accepted in a match
'-compute_cbeta_neighbors' : calculate the number of Cβ neighbors
'-use_epitope_sidechain_during_clash_check' : include the epitope side chains for clash calculations

2. Screen the matches from above with repack to determine a correct inter-clash value between the epitope aligned on the scaffold as per the alignment determined at the matching stage and the rest of the scaffold.

```
rosetta_executable -epi_graft -match -nres_Ab $Ab_RESIDUES -native_complex
$NATIVE_COMPLEX    -loop_ranges $EPITOPE_RANGES -keep_natro
$KEEP_NATRO -use_input_from_match -input_file $EXISTING_MATCHES -paths
$PATHS  -max_closure_rms 5 -max_match_width_delta 30 -max_intra_clash 100
-max_inter_clash 100 -spatial_filter $CDR_H3_ALA_GLY -spatial_filter_min_cbeta 2
-spatial_filter_max_cbeta 10 -compute_cbeta_neighbors -output_file $OUTPUT
-use_epitope_sidechain_during_clash_check  -screen_with_repack
```

$EXISTING_MATCHES is a text file derived from the output file of the Matching stage in the following format:

```
#matching_scaffold_pdbid   loop_id   alignment_system   scaffold_match_begin
scaffold_match_end   epitope _begin   epitope_end

1dwu.pdb        1            N2C_N        90      100         662             669
```

- ' alignment_system' is one of: N2C_N, N2C_CA, N2C_C, C2N_N, C2N_CA, C2N_C, S, E (see Azoitei et. al., Science, 2011, for a full description of the alignment scheme)
- 'loop_id' should be internally consistent to the loop ranges file
- ‘scaffold_match_begin/end' are the residues on the scaffold where the epitope will be transplanted
- ‘epitope_begin/end' specifies the subrange of the epitope in native epitope numbering that will be transplanted onto the scaffold

3. Identify the matches to be used in the grafting design stage (e.g. sort on the numbers of scaffold Cβ atoms neighboring the CDR H3 loop, intra-clash, inter-clash etc.). A "predesign” structure can be generated for visual examination by running Rosetta Multigraft with the following command options:

```
rosetta_executable -epi_graft -match -nres_Ab $Ab_RESIDUES -native_complex
$NATIVE_COMPLEX -loop_ranges $EPITOPE_RANGES -keep_natro $KEEP_NATRO
-use_input_from_match -input_file $EXISTING_MATCHES
-output_predesign_structure_with_Ab
```

4. Grafting design to transfer and fully integrate the epitope with the scaffold followed by sequence design of the resulting epitope-scaffold.

```
rosetta_executable -epi_graft -multigraft -nres_Ab $Ab_RESIDUES -native_complex
$NATIVE_COMPLEX -loop_ranges $EPITOPE_RANGES -keep_natro $KEEP_NATRO
-input_file $INPUT_FILE_WITH_GRAFT_INFO -paths $PATHS
-use_non_monotone_line_search -ex1 -ex1aro -ex2 -extrachi_cutoff 0 -atom_vdw_set
highres -close_as_ALA -grow_as_ALA -graft_with_Ab -repack_Ab -build_loops
-refine_loops -refine_with_constraints -design_after_closure
-refine_with_Ab_after_design -closure_attempts 200 -store_n_best_closures 5
-design_attempts 3 -store_n_best_designs 5 -max_chainbreak_score 0.2
-max_local_rama 2.0 -design_after_closure -closure_reset_period 3 -output_file
$OUTPUT_FILE -export_resfile -export_blueprint -checkpoint $CHECKPOINT
-override_pdb_output_path $OPTIONAL_PDB_OUTPUT_OVERRIDE_DIR
-scan_randomize_moveable -scan_randomize_ss -scan_ss_minimum_content 0.25
-build_with_adaptive
```

$INPUT_FILE_WITH_GRAFT_INFO is a text file derived from the *$EXISTING_MATCHES* file containing the matching stage output and other additional options in the following format:

(*$EXISTING_MATCHES* columns) **$** keep/graft_bb keep/graft_sc **@** SSBM *instructions* & *AAI instructions*

Example:

1dwu.pdb 1 N2C_N 90 100 662 669 **$** graft_bb graft_sc **@** ...xxx | llH*HH. | hhHH **&** WDExxx | pA.?.n. | ??.*

The additional columns delimited by the '$', '@' and '&' characters in *$INPUT_FILE_WITH_GRAFT_INFO* control the type of grafting to be performed, including whether or not to transfer the backbone and/or the side chains of the epitope, how many residues to allow to "move", add or delete from the epitope or the scaffold during loop closure, as well as the secondary structure and amino acid identities of the residues designed to connect the epitope to the rest of the scaffold. Below we will describe in details all the options available for the *$INPUT_FILE_WITH_GRAFT_INFO* to ensure that the epitope is grafted and connected to the scaffold.

The '$' symbol is required to separate the match result columns and the graft info columns, since the number of match result columns and graft info columns can be variable.

The first two columns after the '$' indicate whether or not to graft backbone and side chain:

- for backbone valid settings are: 'graft_bb' and 'keep_bb'
- for sidechain valid settings are 'graft_sc' and 'keep_sc'

The second set of entries, called the SSBM section, is optional and is started by the '@' symbol. This section describes: i) the polypeptide breakpoints between the epitope and the scaffold; ii) what dihedrals on both the epitope and the scaffold can be altered while connecting the epitope to the scaffold at the loop closure stage (i.e. what residues can be “moved”); iii) where and how many residues can be added or deleted on the epitope and the scaffold during loop closure and iv) the secondary structure of these residues. The SSBM section is split into three parts via the breaks:

```
(running N -> C)      Scaffold_Left | Epitope | Scaffold_Right
```

The possible operations are then indicated via tether and linker residues. Tether residues describe usage of existing scaffold and epitope residues, while linker residues describe usage of completely new, added residues or deletion of existing residues. Therefore, the SSBM section can conceptually be split into the following parts where "s" refers to scaffold, "e" refers to epitope, "l" refers to linker and "t" refers to tether:

```
s_left_t  s_left_l  |  e_left_l  e_left_t  e_fixed  e_right_t  e_right_l  | s_right_l  s_right_t
```

Epitope linker residues can only be added, not deleted, therefore the 'x' character described below may not be used when dealing with the epitope. The breakpoints govern the type of loop closures for the epitope. A single loop closure may be used instead of a double loop closure, and rigid body movements may be used during double loop closures (see exposition below).

The third set of entries, named the AAI section, is optional and is started by the ‘&’ symbol. The AAI section describes the identity of the amino acids to be used in the SSBM section. Therefore, an AAI section cannot be defined independently and it requires the existence of a prior SSBM section. The AAI section is also split into three parts via breaks, identical to the SSBM section. If '-use_sequence_biased_fragments' is active, then the AAI will be used for sequence biased fragment picking. In that case, the characters '?', '*', and '.', will be ignored during sequence biased fragment picking (see below for character meanings).

The SSBM and AAI sections are independent of each other, which means that the total number of characters in the two sections may be of different lengths. The only requirement is that both sections MUST have at least two breakpoint indicators. It is not necessary to have instructions in all three parts (scaffold_left/epitope/scaffold_right) of the SSM and AAI sections, e.g. both ' | llH*HH. | hhh ' and ' | llH*HH | ' are valid. Spaces are ignored in both sections, so they may be used to enhance readability if desired. The program checks the validity of both SSBM and AAI sections and will exit if it finds a problem.

The full set of possible instructions is:

SSBM Section Instructions:

special:
@ start of SSBM characters
break types:
| open break
^ open break, use rigid-body
= closed break
general:
* freeze, no changes
. allow move, keep current secondary structure
linker (newly grown or deleted residues):
x delete
h grow helix
e grow sheet
l grow loop
d grow any ("degenerate")
g grow, but not moveable; meant to be used as an override when restarting from external file (-restart_from) to prevent design runs from treating the residue as historically moveable and designing around its neighborhood tether (usage of residues already existing on the epitope and the scaffold)
H use helix
E use sheet
L use loop
D use any ("degenerate")

AAI Section Instructions:

lower case or symbols:
& start of AAI characters
? keep protocol dictated default behavior
* freeze, no changes
. repack
d any amino acid ("degenerate")
r any amino acid except cys and pro ("restricted/regular degenerate")
n nonpolar
a aromatic
p polar
c charged
+ positively charged
- negatively charged
uppercase:
ACDEFGHIKLMNPQRSTVWY amino acid 1 letter codes
dummy/ignored characters for readability to keep 1-1 with ssbm: | ^ = x

SSBM - Defining different types of closures:

In the following we illustrate the different breaks.

```
.. | *** | ..   double break

.. ^ *** ^ ..   double break, use rigid-body moves (two '^' must be used)
```

.. = *** = ..   "double break" with both breaks artificially closed. If a side is indicated as artificially closed, the secondary structure settings are ignored since nothing is moving, However, the setting has an influence on whether a residue was originally "moveable", which can have an effect during design (moveable residues are designed)

```
.. = *** | ..   "double" break" with left break closed, right break open

.. | *** = ..   "double break" with right break closed, left break open

.. = *|** = ..  single break, with '|' marking the new open breakpoint
```

SSBM - Shortcuts:

A single '*' within the epitope region will be expanded as '*' to fill up any remaining residues with respect to the existing epitope length. For example, if a five residue epitope is given, then:

```
| * |      becomes     | ***** |
| ..* |    becomes     | ..*** |
| l*l |    becomes     | l*****l |
```

The analogue to this shortcut in the AAI is a single '?'.

AAI - Shortcuts:

A single '*' within the epitope region by itself with no other instructions will be expanded to fill the entire existing epitope residues as '*' (freeze) and any new epitope residues (linker) as '?' (default). For example, is a five residue epitope is given, then:

```
| * |      becomes     | ***** |
| ** |     no expansion
```

If we have a 5 residue epitope and | l*h | as the ssbm,

```
| * |      becomes     | ?*****? |
```

A single '.' within the epitope region by itself with no other instructions will be expanded to fill the entire epitope part as '.' (repack) and any new epitope residues (linker) as '?' (default). For example, if a five residue epitope is given, then:

```
| . |      becomes     | ..... |
| .* |     no expansion
```

A single '?' within the epitope region will be expanded to fill any remaining residues with respect to the existing epitope length to allow default behavior. For example, is a five residue epitope is given, then:

```
        | ? |      becomes     | ????? |
        | ..? |    becomes     | ..??? |
        | A?. |  becomes     | A?????. |
```

The analogue to this shortcut in the SSBM is a single '*'.

Important things to watch out for:

Remember that the instructions are taken literally. If we have

```
     ..l | * | ..  &  ... | ? | ..  <- the grown 'l' residue remains ala/gly
                                     because it is designated to only be
                                     repacked, typically not desired
     ..l | * | ..  &  ..? | ? | ..  <- the grown 'l' residue is designed
```

It is possible to override amino acid identities of all the epitope residues, meaning you can change "critical" residues for binding if so desired.

Since the length of the SSBM and AAI sections can be different, make sure the sections are describing the operations you want correctly. To reduce confusion, it may be good practice to ensure the two sections are the same length and formatted the same when performing non-simple operations.

Additional SSBM and AAI examples:

```
  @   HHHh | * | llLLL
    means...
      double break
      n-term scaffold tether = 3 moveable using helix
      n-term scaffold linker = grow 1 using helix
      entire epitope rigid
      c-term scaffold linker = grow 2 using loop
      c-term scaffold tether = 3 moveable using loop
         no AAI, so default amino acid behavior during design

  @   HHHh | ll.*****..d | llLLL   &   Y..? | ??.*****..? | ??rrr
    means...
      double break
      n-term scaffold tether = 3 moveable using helix
      n-term scaffold linker = grow 1 using helix
      n-term epitope linker  = grow 2 using loop
      n-term epitope tether  = 1 moveable using whatever ss is there
      5 residues rigid epitope
      c-term epitope tether  = 2 moveable using whatever ss is there
```

```
     c-term epitope linker  = grow 1 using degenerate (any ss)
     c-term scaffold linker = grow 2 using loop
     c-term scaffold tether = 3 moveable using helix
     repack all n-term scaffold tether
     design as tyrosine 1-residues n-term scaffold linker
     default design behavior 2-residues n-term scaffold linker
     default design behavior n-term epitope linker
     repack n-term epitope tether
     5 residues frozen, no repack
     repack c-term epitope tether
     default design behavior c-term epitope linker
     default design behavior c-term scaffold linker
     design with restricted degenerate alphabet for the 3 c-term
       scaffold tether residues

@   HHH = ll.*****..d | llLLL
     double break, but n-term (left) side is artificially closed
       so no backbone movement occurs on the left
     rest is similar to the prior example
     no aai, so default amino acid behavior during design

@   HHxx | ...******l = llEE   &    rrxx | ...******p = ....
     double break, but c-term (right) side is aritifically closed
       so no backbone movement occurs on the right
     n-term scaffold tether = 2 moveable using helix
     n-term scaffold linker = delete 2
     n-term epitope tether = 3 moveable using whatever ss is there
     6 residues rigid epitope
     c-term epitope linker = grow 1
     c-term scaffold linker = grow 2 using loop
     c-term scaffold tether = 2 moveable using sheet
     two residues n-term scaffold tether design w/ restricted degenerate
     two 'x' dummy spacers for readability
     three residues n-term epitope tether repack
     6 fixed epitope residues
     one residue c-term epitope linker design as polar
     2 residues c-term scaffold linker repack
     2 residues c-term scaffold tether repack

@   Hh = ....|..... = lLL   &   ?? = ????|????? = ???
     single break with breakpoint indicated by the "|" cut
     n-term scaffold tether = 1 moveable using helix
     n-term scaffold linker = grow 1 using helix
     4 residues moveable epitope
     5 residues moveable epitope
     c-term scaffold linker = grow 1 using loop
```

c-term scaffold tether = grow 2 using loop
AAI indicates default behavior for all residues

Command line options:

'-build_loops' : centroid level build
'-refine_loops' : full-atom refine after centroid level build
‘-design_after_closure' : design after successful closure attempt
'-refine_with_Ab_after_design' : attempt refine with antibody after design
'-max_chainbreak_score' : maximum chainbreak score that indicates closure (default 0.008)
'-max_local_rama' : maximum allowed rama score per moveable residue for a loop to be considered closed (default 5.0)
'-closure_attempts' : number of times to attempt centroid + fullatom closure (default 25)
'-closure_reset_period' : number of closure attempts before any remaining broken chainbreaks are reset to their original state (helps avoid getting trapped in minima) (default 3)
'-store_n_best_closures' : number of best closed epitope-scaffolds to store (default 5)
'-build_cycles' : number of cycles per build (default 10)
'-refine_cycles' : number of cycles per refine (default 3)
'-use_fast_refine' : decrease number of internal cycles during refine
'-refine_with_minrepack' : refine loops using min-repack procedure
'-graft_with_Ab' : attempt closure with Ab attached
'-close_as_GLY' : use all-glycine scaffold when closing (otherwise default is all-alanine scaffold with glycine at native glycine positions)
'-grow_as_GLY’ : use glycine when growing new residues at closure tips (otherwise default is all-alanine)
'-no_fragment_insertion' : do not use fragment insertion during the centroid level build step
'-use_sequence_biased_fragments' : use amino acid identity string in graft info when picking fragments
'-allow_any_ss_during_fragment_insertion' : during centroid level build step, moveable residues are markedwith 'D' (degenerate) secondary structure, and as a result fragments including these moveable residues are picked with disregard to the secondary structure of those positions
'-rotation_perturb_magnitude' : max rotation magnitude in degrees when using rigid-body perturbation of epitope component
'-translation_perturb_magnitude' : max translation magnitude in angstroms when using rigid-body perturbation of epitope component
'-scan_randomize_moveable' : Randomize number of moveable residues to the left and right of the cut.
'-scan_randomize_cutpoints' : Randomize cutpoints. This is incompatible with -scan_randomize_moveable.
'-scan_randomize_ss' : Randomize secondary structure in moveable residues of broken loops by choosing a random amount of ss content (either loop, 'L', or any, 'D') for a contiguous stretch. All remaining residues on the left and right are set to the secondary structure of the

first left/right non-moveable residue, assuming that second and third left/right non-moveable residues have the same secondary structure, otherwise ss is set to loop.
'-scan_ss_minimum_content' : minimum amount of secondary structure content for the contiguous stretch when scanning
'-scan_ss_use_any' : use any secondary structure when scan randomizing instead of loop

'-design_attempts' : number of times to attempt design of a closed epitope-scaffoldand take best attempt (default 3)
'-store_n_best_designs' : number of best designed epitope-scaffolds to store (default 1)
'-repack_epitope' : repack epitope during design
'-repack_Ab' : repack Ab during design
'-allow_AA_inter_design' : allow ALLAA for inter- design positions
'-intra_design_cutoff' : distance cutoff for deciding on intra- design positions (default 4.0)
'-inter_design_cutoff' : distance cutoff for deciding on inter- design positions (default 4.0)
'-Ab_repack_cutoff' : distance cutoff for deciding on Ab repack positions (default 4.5)

5. Design the scaffold region that interacts with the CDR H3 loop: (use -restart_from $PDB to restart from the design structures generated at the grafting design stage)

```
rosetta_executable -epi_graft -multigraft -nres_Ab $Ab_RESIDUES -native_complex
$NATIVE_COMPLEX        -loop_ranges $EPITOPE_RANGES -keep_natro
$KEEP_NATRO -input_file $INPUT_FILE_WITH_GRAFT_INFO -paths $PATHS
-use_non_monotone_line_search    -ex1 -ex1aro -ex2 -extrachi_cutoff 0 -atom_vdw_set
highres -close_as_ALA   -grow_as_ALA -graft_with_Ab -repack_Ab -design_after_closure
-design_attempts 1   -store_n_best_designs 5 -max_chainbreak_score 0.2 -max_local_rama
2.0 -output_file $OUTPUT -restart_from $PDB_RESTART_FILE
-complementarity_design_attempts 1     -complementarity_design_cycles 3
-inter_design_cutoff 0 -intra_design_cutoff 0
-complementarity_shell_repack  -complementarity_design
$COMPLEMENTARITY_TEXT_FILE -complementarity_rb
```

$COMPLEMENTARITY_TEXT_FILE is defined as a text file that lists the residues on the antibody. The format is one residue per line with residues in *native numbering* specified by chain-resnumber-insertion_code format, e.g.:

```
H98
H99
H100
H100A
H100B
H100C
H100D
```

** IMPORTANT, make sure and set intra/inter design cutoffs to be zero otherwise positions you want to remain constant will get designed!

```
-intra_design_cutoff 0 -inter_design_cutoff 0
```

Command line options and their defaults:

'-complementarity_design' <string>	[ ]	complementarity design text file
'-complementarity_rb'	{ false }	do rigid body moves?
'-complementarity_design_cutoff'	[4]	design distance cutoff (Å)
'-complementarity_design_attempts'	[1]	number of design attempts
'-complementarity_design_cycles'	[3]	number of cycles per attempt
'-complementarity_shell_repack'	{ false }	repack scaffold residues neighboring the scaffold region binding to the CDR H3 loop?
-complementarity_shell_redesign	{ false }	redesign scaffold residues neighboring scaffold's complementarity patch?

FIXED BACKBONE DESIGN

To generate models of the different 1dwu, 2F5 or epitope variants for computational ΔΔG calculations, the following "fixed backbone" command line was used in Rosetta 3.5:

```
$PATH/rosetta-3.5/rosetta_source/bin/fixbb.macosgccrelease @min_pack.flags -
database $PATH/rosetta-3.5/rosetta_database -in:file:s 1dwa_CM.pdb -packing:resfile
$RESFILE
```

'min_pack.flags' was defined as:

```
-min_pack
-packing
   -multi_cool_annealer 10
   -ex1
   -ex2
   -ex1aro
   -ex2aro
   -extrachi_cutoff 0
   -use_input_sc
-suffix min_pack
-nstruct 5
```

ROSETTA ΔΔG CALCULATIONS

The Rosetta ΔΔG binding energy for the different 1dwu_CM and 2F5 variants was computed by scoring the antibody and the epitope-scaffold separately and then subtracting from the sum of their total energies the energy score of the antibody-epitope-scaffold complex. The following command line was used to achieve this in Rosetta 3.5:

```
$PATH/rosetta-3.5/rosetta_source/bin/InterfaceAnalyzer.macosgccrelease -database
$PATH/rosetta-3.5/rosetta_database @interface.flags -in:file:s 1dwa_CM.pdb
```

The ‘interface.flags’ file was defined as:

```
-fixedchains H L
-out:file:scorefile interf.sc
```

The reported ΔΔG is outputted in the ‘dG_separated’ field of the output file resulting from this procedure.