**Data Distribution Roadmap for**

Doria-Rose *et al.*, “Mapping Polyclonal HIV-1 Antibody Responses via Next-Generation Neutralization Fingerprinting”

This archive contains data and program files for the work presented in Doria-Rose *et al.*, “Mapping Polyclonal HIV-1 Antibody Responses via Next-Generation Neutralization Fingerprinting.” All data and programs are distributed without any stated or implied warranty.

The main folder in this archive contains three subfolders related to three different types of analysis performed in the publication:

* **panels\_scores**: This folder contains data and program files related to the panel optimization algorithms and the analysis of panel size vs. prediction accuracy.
* **sim\_population**-**level**: This folder contains data and program files related to the algorithm for population-level analysis.
* **large**-**scale\_cohort**: This folder contains data and program files related to the application of the population-level NFP algorithm to a large collection of HIV-infected samples.

The contents of each of these folders are described in the following sections.

1. **panel\_scores:** There are two subfolders in this folder:
   1. **1-gen\_sera:** This folder contains data and program files related to the generation of simulated sera. The two subfolders relate to the two sets of sera: for training and for testing. With each of these subfolders are several input files (.in) and a Mathematica script for performing the simulations (“gen\_theoretical\_comb\_w-var.m”), as well as output files (.out) generated by that script. The script can be executed from within a Mathematica environment. The file “sera\_neut.out” contains the neutralization values for the generated simulated sera, while the file “sera\_classWeights.out” contains the pairwise combination of antibody specificities for the given simulated serum (with 0.5 for specificities that are part of the pair and 0 for specificities that are not). These two output files are used for evaluation of the accuracy of candidate virus panels.
   2. **2-panel\_search:** This folder contains data and program files related to the different panel search experiments, including the 20-strain panel f61 and the evaluation of the effect of virus panel size on prediction accuracy. The subfolders include the data for panels of size 10-50 at increments of 5 (in subfolders “s*X*”, where *X* is 10-50), as well as the data for the previously published 21-strain panel and the large 132-strain panel (in subfolder “std\_panels”). The “s20” subfolder contains the same experiments as the subfolders for the other panel sizes, and in addition, contains data related to panel f61. Within “s20”, there are three subfolders related to the three methods used for panel selection: random panel generation (folder “rand”), sequence diversity optimization (both for epitope-specific and full sequences: folder “seq\_id”), and the best-performing Monte Carlo-based optimization (folder “opt”).
      1. **Random panel generation (folder “rand”):** The first step here is to generate a set of random panels of a specified size (in this case, 20); this can be achieved within environments such as Mathematica or Matlab. The full set of strains from which the 20-strain panels are randomly selected is given by the “mab\_neut\_vStrains.in” file in the “1-get\_sel-strain\_pos” folder. An example set of random 20-strain panels is given by the “sel\_subsets.in” file in that folder. The “get\_sel-subset\_list-pos.m” Mathematica script is used to generate the positions of the strains from each of the random panels into the full list of strains. The output file from this script is then used as input for the NFP algorithm, which is found in the “2-ser\_del” folder. The neutralization data for the reference set of bNAbs is stored in the three “mab\_neut\_x.in” files (bNAb names, strain names, and corresponding bNAb-strain neutralization data matrix values). All these input files are used as input to an Octave script (“octave\_genSel.oct”) that can be executed within an Octave environment (e.g., “octave < octave\_genSel.oct”). The script outputs a “virusSelSubsetsSerDel.out” in which each line corresponds to a single selected virus panel of the appropriate size (in this case, 20 strains). The file starts with six values associated with panel performance. The serum delineation error scores described in the publication (computed as the difference between the predicted prevalence of the component antibody specificities and the actual prevalence from “sera\_classWeights.in”) is the second of these six values. This is followed by a list of the strain names for the given virus panel. This output file can then be sorted using standard Unix commands based on different properties (e.g., serum delineation error).
      2. **Sequence diversity (folder “seq\_id”):** The two folders contain the data/code for using the full gp140 sequences (“full\_seq”) or only epitope residues (“ep\_res”). We will describe the “ep\_res” folder since both are analogous, with the exception of the sequence alignment (full vs. epitope residues only) used for the analysis.

The first step of the analysis is to generate a set of virus panels based on sequence diversity optimization, and these results are given in the “1-astar\_search” folder. Within that folder, the “1-clust” subfolder relates to generating a hierarchical clustering of the full list of strains based on the input sequence alignment (given by the .fasta file). The “parms.cfg” file contains a single line with input parameters for the Mathematica script, specifying (in order) the smallest and largest number of subclusters, the input fasta file, and two output files. The first output file shows the pairwise sequence identity between each pair of strains. The second file contains the hierarchical clusterings. The first line is the clustering with all strains in one cluster, followed by a line for the clustering when the hierarchical clustering is split to obtain two subclusters, etc. Next, these two output files are used as input to an A\* branch-and-bound search for identifying virus panels based on sequence diversity (folder “2-astar\_search”).

The code for the A\* search is implemented in Java and was adapted from an implementation part of the OSPREY protein design software suite. A\* uses the pairwise sequence identities between all pairs of strains, in order to enumerate a list of virus panels in order of sequence diversity. The sequence-based strain clustering (e.g., into 20 subclusters) is used for partitioning the full set of strains, so that the resulting virus panels are constrained only contain one strain per subcluster (e.g., for a total of 20 strains per panel). The A\* search uses the parameters specified in the “parms.cfg” parameter file. The first two lines in “parms.cfg” are the output files from the clustering procedure. numTotalStrains specifies the full set of strains, while minStrains and maxStrains specify the min and max size of virus panels that should be identified. The parameter lambda relates to pruning potential combinations that may not be optimal and should generally not be modified. numSubsets specifies how many panels should be generated, and outFile specifies the name of the output file. The A\* search can be executed using a command such as “java SelDiverseStrains parms.cfg”.

The generated list(s) of virus panels are output to the “sel\_subsets” folder. One a list of strains is selected, the application of the NFP algorithm is similar to that described for the random search above, and can be found in the “2-ser\_del” subfolder.

* + 1. **Optimization search (folder “opt”):** Within folder “opt”, the source for panel f61, there are four subfolders:
       1. **1-search:** This folder contains data and program files for performing the optimization search. The “sera\_classWeights.in” and “sera\_neut.in” input files are obtained from the output of the serum simulation experiments described above. The neutralization data for the reference set of bNAbs is stored in the three “mab\_neut\_x.in” files (bNAb names, strain names, and corresponding bNAb-strain neutralization data matrix values). All these input files are used as input to an Octave script (“octave\_genSel\_mc.oct”) that can be executed within an Octave environment. The script outputs a “virusSelSubsetsSerDel.out” in which each line corresponds to a single selected virus panel of the appropriate size (in this case, 20 strains). The file starts with two indexing values for bookkeeping during the search, followed by six values associated with panel performance. The serum delineation error scores described in the publication (computed as the difference between the predicted prevalence of the component antibody specificities and the actual prevalence from “sera\_classWeights.in”) is the second of these six values. This is followed by a list of the strain names for the given virus panel. This output file can then be sorted using standard Unix commands based on different properties (e.g., serum delineation error).
       2. **2-unique\_subsets:** The generated list of panels from “1-search” is then filtered to only include unique sequences. This is performed in two steps, in which the first step is to sort the strains (according to their names) within each panel (e.g., by using the Mathematica script in “1-sort\_strains”), and the second step is to search through the sorted panels from step 1 and only keep the unique panels (e.g., by using the “get\_unique.bash” script in “2-get\_unique\_subsets”).
       3. **3-test\_sera:** To further evaluate the performance of candidate virus panels, the test set of sera (as opposed to the sera used for the panel search in “1-search”) can be used. The application of the NFP algorithm to the supplied list of virus panels is analogous to that described for the random panel generation (section 1.2.1 above).
       4. **4-mis\_class:** This folder contains code (Mathematica script) for checking whether a list of virus panels (in this example, “sel\_subsets\_breadth\_0.25.in”) is capable of successfully clustering antibody neutralization fingerprints (“mab\_neut\_22\_132.in”) according to a pre-defined grouping based on known epitope specificity (“ab\_ep\_cl\_22.in”). The script computes a neutralization fingerprint-based clustering of the input antibodies and compares it to the pre-defined epitope-specific grouping. The first line of the output file contains: True/False for whether there has been misclassification of antibodies when the entire set of viruses from the input file is used (i.e., False if the fingerprint-based clustering is correct for the full virus panel), followed by the actual resulting antibody clustering. Each following line contains the output for a single virus panel from “sel\_subsets\_breadth\_0.25.in”, with the following format: index of the virus panel from the input file, True/False for misclassification, list of strains in the virus panel, and resulting fingerprint-based clustering of the antibodies.
       5. **5-panel\_eval\_f61:** This folder contains data/code related to the analysis of virus panel f61.
          - **test\_sera:** This is the analysis for panel f61 against the set of test simulated sera.

The application of the NFP algorithm is found in the “1-ser\_del” folder, with the “1-get\_sel-strain\_pos” and “2-ser\_del” subfolders executed as described above, with the following highlighted additional step. The Octave script “ser-del\_fixed-subset.oct” is used for generating detailed analysis of the NFP results for a given panel (in this case, f61). The output of this script is two files:

“serum\_delineation.out” outputs the delineation scores for each of the 10 reference-set specificities for each of the test set of sera. In other words, these are the predictions for the prevalence of each of the 10 specificities (columns) for each of the 4,500 simulated sera (rows).

“serum\_delineation\_rss.out” outputs the residuals from the fitting procedure for each of the test set of sera. These are used for computing the residual scores reported in the publication.

The “3-t\_scores” subfolder within “1-ser\_del” contains the data/code for computing residual scores. The “known\_rss.in” file contains the residuals for the simulated sera with known specificities, whereas the “unknown\_rss.in” file contains the residuals for the simulated sera with unknown specificities (see below for folder “unk\_sera”); “query\_rss.in” would contain the residuals for the set of sera that are being interrogated. The output of the Mathematica script is a single file that transforms the values from “query\_rss.in” into residual scores, as described in the publication.

In the main “test\_sera” folder, the only other folder is “2-confidence\_score”. This folder contains data/code for computing the frequency of random fingerprint signals within each serum in a given set. In that folder, the “1-mab\_neut\_split” subfolder contains code to generate a set of files, each containing one random fingerprint (from a large set of such fingerprints, stored in “mab\_neut\_lt-0.2-corr.in”) for the given virus panel. These files are then used as input for the NFP delineation in “2-ser\_del”, which separately adds each of the random fingerprints to the reference antibody set, and determines how frequently there is a signal for random fingerprints for a given serum. The code in “2-ser\_del” can be run using a script, such as the included “run\_nfp-for-rand.bash”. In essence, the experiments in this folder are simply multiple executions of the NFP delineation for a set of sera, where each execution differs by the inclusion of a different random fingerprint within the reference antibody set, as described in the publication. In the “3-summary” folder, the results from the NFP runs are copied over, extracted, and summarized using the included Mathematica script. The output file contains results for each input serum, with one line per serum. The frequency of random signals for a given serum is computed as the value in the next-to-last column (number of random signals) divided by the total number of random fingerprints (in this case, 10,000).

* **nfp\_sera:** This is the analysis for panel f61 against real donor sera. The application of the methods and data are analogous to those described for the “test\_sera” folder above.
* **mab\_comb\_ser-del:** This is the analysis for panel f61 against experimental monoclonal antibody combinations. The application of the methods and data are analogous to those described for the “test\_sera” folder above.
* **unk\_sera:** This is the analysis for panel f61 against sera with unknown specificities. The sera are generated in folder “1-gen\_sera”, by using a 0.2 correlation cutoff against any of the monoclonal antibodies in the reference set, as described in the publication. The generated sera are then used as input for the NFP delineation in “2-ser\_del”, which outputs the predicted reference-set antibody specificities for each unknown serum, as well as the residuals from the fitting procedure.

1. **sim\_population-level:** There are two subfolders in this folder:

**1-eval\_sim-sera:** This folder contains data and program files related to the evaluation of simulated sera for testing the population-level NFP algorithm. The two subfolders here contain the data for the original 21-strain panel (“s21”) and the new 50-strain panel used in the current publication (“c632”). In “c632”, the “1-get\_sel-strain\_pos”, “2-ser\_del” and “3-confidense\_score” are analogous to those described in sections 1.2.1 and 1.2.3.5 above, with the following difference: the NFP analysis was performed for simulated sera with 0, 1, or 2 component antibodies with known epitope specificities (given in “zero\_comb”, “single\_comb”, and “pair\_comb”).

**2-sim\_pop:** The output from the “1-eval\_sim-sera” folder is then used as input for this folder. This input contains the serum delineation scores, residual scores, frequency of random signals, and actual antibody composition for each of the simulated sera, for panels s21 and c632. The Mathematica script is then applied to these output files to generate different “cohorts” of sera and compute the accuracy of the NFP predictions at the “population” level, as described in the publication. In the main folder, the two output files from this script are:

“pred\_acc\_summary\_ext.out”: This file specifies the prediction accuracy for the two virus panels given different fractions of sera with known vs. uknown specificities. Each line gives the following information:

Name of virus panel.

Is filtering used based on the confidence scores for each serum (y/n).

Is the population-level adjustment method applied (y/n).

A list of four values for the fractions of sera composed of: 0, 1, or 2 antibodies (the last value is not used and should always be 0).

A list of ten values for the median cohort-level prediction accuracy (as described in the publication) for each of the ten reference antibody specificities for the simulated cohorts of sera.

“pred\_acc\_summary.out”: Each line in this file gives the ratio of prediction accuracies between the two virus panels, for each combination of fractions of simulated sera (0, 1, and 2 component antibody specificities), for the ten reference antibody specificities.

In addition, detailed output for each cohort simulation for a given virus panel, algorithm filters, and combination of fractions of simulated sera (0, 1, and 2 component antibody specificities) can be found in the “pred\_acc\_out” subfolder. Example output is given in the file “pred\_acc\_s21\_{0.5, 0.25, 0.25, 0.}.out”, which contains the data for simulations with 50% sera with zero known specificities, and 25% each of sera with 1 and 2 known specificities, for virus panel s21. Each line corresponds to a single simulation (a selected cohort of sera) that shows:

Name of virus panel.

Is filtering used based on the confidence scores for each serum (y/n).

Is the population-level adjustment method applied (y/n).

Simulation number.

A list of ten values for the cohort-level prediction accuracy for the ten reference antibody specificities.

Lists of the indices of the simulated sera selected for the current cohort for the sera with 0, 1, or 2 known antibody specificities (if filtering based on confidence scores is used, then this list will include the sera that pass those filters). These serum indices can be used to look up the respective sera from the input files.

1. **large-scale\_cohort:** There are three subfolders in this folder:

**1-c632:** This folder contains data and program files related to the delineation of serum specificities for the simulated sera with virus panel c632. The execution of the NFP analysis is analogous to that described in sections 1.2.1 and 1.2.3.5 above.

**2-ser\_del:** This folder contains data and program files related to the delineation of serum specificities for the large cohort of HIV-infected donors, with virus panel c632. The execution of the NFP analysis is analogous to that described in sections 1.2.1 and 1.2.3.5 above. Of note, the “sera\_neut.in” file contains the neutralization data for the set of HIV-infected donors used in the analysis (as described in the publication), for the 69 strains that are in common between the serum data and the full 132-strain mAb-virus panel; the two “.txt” files contain a more user-friendly version of the overlapping monoclonal and polyclonal neutralization data for these 69 strains.

**3-confidence\_score:** This folder contains the data/code for computing the confidence scores for the NFP predictions on the large HIV-infected cohort. The execution of this analysis is analogous to that described in section 1.2.3.5 above.