Supplementary Information for

Improved mutant function prediction via PACT: Protein Analysis and Classifier Toolkit

Justin R. Klesmith[1], and Benjamin J. Hackel[1,*]

[1]University of Minnesota – Twin Cities, Department of Chemical Engineering and Materials Science, 421 Washington Avenue SE, Minneapolis, MN 55455

*Corresponding author: Benjamin J. Hackel, 356 Amundson Hall, 421 Washington Avenue SE, Minneapolis, MN 55455; 612-624-7102; hackel@umn.edu

This PDF file includes:

Other supplementary materials for this manuscript include the following:

Python scripts available at https://github.com/JKlesmith/PACT

Note S1: Workflow explanation for the *fitness* protocol to convert sequencing to fitness metric measurements. To run the protocol enter the config file at the command line >python pact.py –c ./fitness.ini.

This protocol is capable of:
1) Merging FASTQ files.
2) Translating FASTQ files to amino acid sequences and filtering on read quality.
3) Filtering translated sequences based on expected library design.
4) Counting accepted and filtered synonymous and non-synonymous mutations.
5) Calculating the $\log_2$ enrichment of mutations above a certain read count threshold.
6) Calculating the fitness metric of mutations and wild-type synonymous mutations.
7) Calculating the distribution of wild-type synonymous mutations.
8) Calculating the error on fitness metrics.
9) Calculating a probability value that a non-synonymous mutation is functionally indistinguishable from wild-type.
10) Calculating the per-residue amino acid frequency change for multiple-codon libraries.
11) Calculating mutual information for amino acid pairs in multiple-codon libraries.
12) Outputting tile library statistics (total read counts, amino acid and codon coverage, median library read counts, wild-type synonymous codon enrichment mean and standard deviation) for publication.

Comparison to existing packages:

Alternative software packages for the *fitness* protocol are Enrich 0.2 (Fowler, et al., 2011), Enrich2 (Rubin, et al., 2017), and dms_tools (Bloom, 2015). While Enrich 0.2 is no longer supported nor recommended by the authors, it still has a strong presence as it was one of the early software packages to calculate $\log_2$ enrichments of mutations in deep mutational scanning experiments. The PACT *fitness* protocol replicates the core functionality of these existing packages with some improvements and differences; however, PACT goes beyond their scope (**Fig. 1**). In short, existing software packages process deep sequencing reads then calculate a metric of per-variant function. For example, *Enrich2* provides an implementation of a random-effects model that is geared for calculating variant enrichment over multiple time points while combining multiple replicates. The experiments targeted by the PACT *fitness* protocol are end-point with a reference library and a selected library where replicates are processed separately and compared via correlation statistics. More importantly, other protocols within the PACT platform are not included within existing packages. The PACT *fitness* protocol offers several advantages over existing packages. First, it provides stricter FASTQ filtering by offering minimum Q and average Q quality filtering. *fitness* also filters reads with incorrectly mutated locations and total number of mutations based on library design. The six other advantages PACT fitness has over

Enrich 0.2 is native multi-processing support, calculation of synonymous wild-type codon fitness to guide mutation classification and evaluation of statistical significance, the conversion of $\log_2$ enrichment values into fitness metrics, the support of 'tiles,' and the analysis of designed multi-site comprehensive libraries.

Workflow and config file options:

```
[pact]
pact_config_version: 2018.6
pact_protocol: fitness
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
fastq_merge_sel: False
fastq_merge_ref: False
fastq_filter_translate_ref: False
fastq_filter_translate_sel: False
filter_counter_sel: False
filter_counter_ref: False
enrichment: False
fitness: False
multiple_freq_mi: True
library_stats: False
```

The [workflow] section defines which steps to perform by either True or False. For this protocol, the order listed here is the order of the workflow.

```
[global]
wtdna: CCCGAG…
wtaa: PE…
processes: 12
directory: ./runs/loops/loop2/
output_prefix: Loop2
firstaamutated: 229
lastaamutated: 256
mutationtype: multiple
mutthreshold: 7
mutcodons: [[243, 244, 245, 246, 247, 248, 249]]
```

The [global] section includes shared information for multiple steps. The wtdna and wtaa should be the entire sequence of the gene (neither the entire vector nor just the tile). The first and last aa mutated is the starting and ending amino acid of the tile. Mutcodons defines which residues are mutated [1,'n',92]] would be 1 to 92, [[1,3,5], [7,9,11]] would be 1,3,5 in one group and 7,9,11 in a second group. Mutationtype is either single or multiple. The number of processes can also be defined here. Mutthreshold defines the maximum number of amino acid non-synonymous mutations to accept.

```
[fastq_merge_sel]
forward_fastq: Sel_R1.fastq
reverse_fastq: Sel_R2.fastq
directory:
min_coverage: 0.2
[fastq_merge_ref]
forward_fastq: Ref_R1.fastq
reverse_fastq: Ref_R2.fastq
directory:
min_coverage: 0.2
```

Illumina MiSeq reads are merged by [fastq_merge]. Reads are merged by a pairwise analysis of the forward and reverse read. The relative position with the most number of matches is used to merge the reads. The software will select 100 reads and perform statistics on the median position value to speed up the rest of the file. Mismatches in the read overlaps are handled by taking the base with the higher Q score or marked as unresolvable in the case of equal quality. The forward and the reverse reads must be of equal length and have at minimum 20% overlap. Directory provides the opportunity to define a custom directory to search for the fastq file otherwise the software will search the directory defined in [global].

```
[fastq_filter_translate_sel]
fiveprimeanchor:
enable_anchors: True
qaverage: 20
qlimit: 10
fastq_file: <custom file if needed, leave blank otherwise>
[fastq_filter_translate_ref]
fiveprimeanchor:
enable_anchors: True
qaverage: 20
qlimit: 10
fastq_file: <custom file if needed, leave blank otherwise>
```

The software will select 100 reads and perform a non-gapping alignment to find the nine bases immediately preceding the tile as the '5' anchor' sequence. This sequence can also be defined in this section if multiple repeats are present. If the reads have been merged and trimmed by other software then the requirement for anchors can be disabled by enable_anchors. The reads are then translated and filtered by Q score (version 1.8, phred +33) by [fastq_filter_translate]. Output from other programs that merge Illumina reads (Magoč and Salzberg, 2011; Masella, et al., 2012) can be used as the standard FASTQ file format is used as the input. The read quality filter offers the option of both filtering on the average Q score of the entire read and a minimum lower limit of Q for any one base. The default settings are Q of 20 averaged across the entire read, a minimum Q of 10 for any one base, and no unresolvable 'N' bases.

```
[filter_counter_sel]
read_file: <custom file if needed, leave blank otherwise>
[filter_counter_ref]
read_file: <custom file if needed, leave blank otherwise>
```

Synonymous mutations and the location and amount of non-synonymous mutations are filtered by and then counted by [filter_counter]. Non-synonymous mutations within the expected mutational library design and the total amount within an expected threshold are accepted.

```
[enrichment]
ref_count_wildtype: <custom file if needed, leave blank otherwise>
sel_count_wildtype: <custom file if needed, leave blank otherwise>
ref_count: <custom file if needed, leave blank otherwise>
sel_count: <custom file if needed, leave blank otherwise>
ref_count_rejected: <custom file if needed, leave blank otherwise>
sel_count_rejected: <custom file if needed, leave blank otherwise>
ref_count_threshold: 12
sel_count_threshold: 12
strict_count_threshold: False
```

The $\log_2$ enrichments for synonymous, rejected, and accepted non-synonymous mutations are calculated by [enrichment]. The $\log_2$ enrichments for the rejected and accepted non-synonymous mutations are then calculated for mutations above a user-defined read count threshold (default 12 counts for the reference and selected population) as mutations with low counts are affected by counting noise (**Fig S1**). If a mutation is above this threshold in one of the two populations then the original count or a count of one in the case of zero is added to the other population to capture mutations that fell out of the population or had a dramatic enrichment. If strict_count_threshold is set as True, then the variant must have that count defined in the ref or sel count threshold values. The $\log_2$ enrichment variance (Klesmith, et al., 2015) is calculated for each variant.

```
[fitness]
pact_enrichment_summary: <custom file if needed, leave blank otherwise>
pact_enrichment_accept_nonsynon: <custom file if needed, leave blank otherwise>
pact_enrichment_wtsynon: <custom file if needed, leave blank otherwise>
manual_log2:
metric: e-wt
evalue_type: facs
evalue_facs_cellcount: 10389351
growth_gp: 10
facs_sd: 0.6
facs_pc: 0.05
```

Several fitness metrics are available to be applied to the $\log_2$ enrichments (e-wt, facs, or growth) (**Table S1**). If a manual wild-type enrichment value is desired then it can be defined in manual_log2. The standard deviation of the fitness metric values at varying read depths for synonymous wild-type mutations within the designed library is calculated and used to define functionally neutral mutations (Klesmith, et al., 2017).

To calculate the expectation value for the number of experiments per variant (**Note S13 and S14**) the type of analysis is selected on evalue_type (either FACS or growth). If the FACS expectation value is selected then the number of total cells that were able to be collected is entered on the evalue_facs_cellcount line. For the growth expectation value, growth_gp is used for the number of generations (converted to the nearest integer).

For the growth fitness metric the number of generations of growth is entered on the growth_gp line.

For the FACS fitness metric the standard deviation and percent collected are entered on the facs_sd and facs_pc lines, respectively.

In the case of a single-site saturation mutagenesis library a csv heatmap of fitness metrics is saved, while in the case of a multiple codon library a frequency-based heatmap is saved.

All results for the location, mutation, counts, enrichment, and fitness are saved as a column based tsv file. The internal Python dictionary with the mutation/count/enrichment/fitness data is saved as a .pact file for use by other protocols within the software distribution.

```
[multiple_freq_mi]
pact_fitness_nonsynon: <custom file if needed, leave blank otherwise>
frequency_log2_filter: False
```

Only for libraries with multiple codons will a per-residue codon enrichment heatmap and mutual information will be calculated and output as csv files. While not recommended, if only mutations with $\log_2$ enrichments are desired to be processed then frequency_log2_filter can be set to True (this is only recommended if strict_count_threshold is enabled in the [enrichment] section to avoid mutations that 1 was added in for).

```
[library_stats]
pact_enrichment_summary: <custom file if needed, leave blank otherwise>
pact_fitness_nonsynon: <custom file if needed, leave blank otherwise>
pact_fitness_wtsynon: <custom file if needed, leave blank otherwise>
codon_type: {'NNK':[229,'n',256]}
```

The library_stats module is responsible for reporting the number of read counts, amino acid coverage, and codon coverage for a given input library design. The total number of reads and the percentages of synonymous, accepted and rejected non-synonymous mutations for both the reference and selected populations is reported. The theoretical size of the library for single or multiple codon libraries is calculated and then used to determine the total coverage of non-synonymous amino acid with $\log_2$ enrichments. The theoretical codon coverage is calculated by a user defined location to codon setting (all base and degenerate (GATCRYMKSWHBVDN) codes are supported) then the fold oversampling of the two populations is reported. If different codons were used, then codon_type can be defined as, for example: {'NNK':[229,'n',256], ‘NNN’:[257,’n’300]}. Where the nomenclature is that for residues from 229 to 256 are NNK and 257 to 300 are NNN.

Note S2: Workflow explanation for the *classification_features* protocol to combine PACT fitness datasets with sequence and structural features for model generation.

This protocol is capable of:
1) Combining PACT fitness datasets.
2) Calculating PSSM and frequency observed sequence homology for each mutation.
3) Calculating distance from active site, contact number, and fraction burial of residues.
4) Classifying mutations based on size and chemical properties change.
5) Calculating site-wise consensus information (see **Note S8**).

```
[pact]
pact_config_version: 2018.6
pact_protocol: classification_features
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
combinepact: True
basal_count: True
blastp_align_filter: False
pssm: False
pssm_reader: True
pdb_import: True
distance_to_active: True
contact_number: True
residue_chemical_size: True
consensus: True
```

This will enable and disable individual sections. It is possible to run the protocol without PACT fitness datasets if combinepact is disabled. This would be potentially used with the *function_filter* protocol to identify mutations that are not deleterious.

```
[global]
wtaa: MPIAT…
directory: ./pact/tests/classification_features/
output_prefix: enzyme_filter_lgk
```

This defines the wild-type amino acid sequence, working directory, and output prefix of files.

```
[pdb_import]
numpdb: 1
file_1: 4zlu_cleanf2.pdb
```

This will import a PDB file.

```
[distance_to_active]
pdb_file: 4zlu_cleanf2.pdb
atoms: CA
chains: A
ligands: ADP,MG,4PW
active_residues:
```

```
ligand_chains: A
report_chain: A
```

This section will calculate the distance to active site (see **Note S6**).

```
[contact_number]
pdb_file: 4zlu_cleanf2.pdb
atoms: CA
distance: 10
chains: A
report_chain: A
```

This section will calculate the contact number (see **Note S6**).

```
[combinepact]
numdatasets: 2
dataset_1: LGK_Triple
dataset_2: LGK_wt
[LGK_Triple]
file1: ./pact/tests/datasets/lgk.1/LGK_Triple_12_SSM_fitness.pact
file2: ….
[LGK_wt]
file1: ./pact/tests/datasets/lgk_wt/LGK_WT_1_SSM_fitness.pact
file2: ….
```

This section will import the PACT fitness datasets.

```
[variant_classification]
class_column: sd_from_wt
class_threshold: 1.5
```

This section will classify mutations based on the ‘column’ (fitness or sd_from_wt) and threshold.

```
[classification_analysis]
pdb_file: 4zlu_cleanf2.pdb
chain: A
```

This section defines which PDB file to use and which chain to use when preparing the output CSV and .pact dataset.

```
[blastp_align_filter]
processes: 2
cdhit_clustering_threshold: 0.98
ncbi_xml: J88W49S1014-Alignment.xml
minquerylen: 0.6
minseqid: 0.35
nummaxhits: 500
[pssm]
region_size: 20
manual_regions: [[0, 19], [20, 39], [40, 59], [60, 79], [80, 99], [100, 119], [120, 139], [140, 159], [160, 179], [180, 199], [200, 219], [220, 239], [240, 259], [260, 271]]
```

This section will calculate the PSSM and per-residue frequency (see **Note S5**). This section is required for the 'consensus' calculations within this protocol.

Note S3: Workflow explanation for the *function_filter* protocol.

This protocol has the capacity to:
1) Count input mutation classification for different analyses (**Table S4**) to be used to calculate naïve Bayesian probabilities.
2) Score input mutations against stored LGK-WT/LGK.1 Bayesian feature probabilities and be classified as either beneficial, neutral, or deleterious.
3) Test all combinations of stored Bayesian probabilities to optimize false positive rates for a given dataset.
4) Filter mutations from an input dataset on an old filter and new filter.

```
[pact]
pact_config_version: 2018.6
pact_protocol: function_filter
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
import_classifiers: True
bayes_count: False
bayes_model_score: False
strict_filter_old: False
strict_filter_new: True
bayes_combo: False
```

Import_classifiers is required as that will import the dataset. Bayes_count will count mutation classifications per analysis bin (i.e. PSSM, contact number, etc). Bayes_model_score will import a dataset and classify mutations based on the LGK-WT/LGK.1 probabilities (individual probabilities can be turned on or off in the [bayes_model_score] section). Strict_filter new and old will filter mutations based off of the old and new filters. Bayes_combo will score a dataset by testing all combinations of classifier.

```
[global]
wtaa: MPIAT…
directory: ./pact/tests/function_filter/
output_prefix: enzyme_filter_lgk
```

This defines the wild-type amino acid sequence, working directory, and output prefix of files.

```
[import_classifiers]
file: enzyme_filter_lgk_dataset
```

This is the input dataset (from the *classification_features* protocol).

```
[bayes_count]
classification_key: LGK_Triple_classified
classifiers: BEN,NEU,DEL

[variant_classification]
```

[bayes_model_score]
pssm_variant: false
frac_burial: false
contact_number: false
wt_cons: true
variant_cons: true
d_to_a: false
mut_percent: true
wt_percent: false
max_percent: false
pro_v_contactnum: false

Note S4: Workflow explanation for the *sequence_homology* protocol which will input a XML file from a blastp search and then produce a PSSM and observed frequency tables. Sequences are handled as described in (Goldenzweig, et al., 2016). In short, a BLASTP search with an expect value of 0.0001 is imported, clustered on similarity by CD-Hit, aligned using Muscle, and PSSM and frequency data calculated by PsiBLAST.

This protocol has the capacity to:
1) Filter XML from a blastp search for length and sequence identity.
2) Call CD-Hit to filter reads based on similarity.
3) Call MUSCLE to perform the multiple sequence alignment.
4) Use PSIBlast to produce PSSM and observed percentages heatmap and csv files.
5) Count mutation classifications based on different sequence homology thresholds.

```
[pact]
pact_config_version: 2018.6
pact_protocol: sequence_homology
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
blastp_align_filter: False
pssm: False
pssm_reader: True
site_frequencies: True
combinepact: True
analysis_sitefitness_homology: True
```

Blastp_align_filter and pssm will perform the multiple sequence alignment (MSA), MSA similarity filtering, PSSM generation, and weighted amino acid frequencies calculation. PSSM_reader will read a stored sequence homology dataset (i.e. blastp_align_filter and pssm steps can be disabled after generation). Site_frequencies

```
[global]
wtaa: MPIAT…
directory: ./pact/tests/sequence_homology/
output_prefix: enzyme_homology_lgk
```

The wild-type amino acid sequence, the working directory, and the output prefix is defined here.

```
[blastp_align_filter]
processes: 2
cdhit_clustering_threshold: 0.98
ncbi_xml: J88W49S1014-Alignment.xml
minquerylen: 0.6
minseqid: 0.35
nummaxhits: 500
[pssm]
region_size: 20
manual_regions: [[0, 19], [20, 39], [40, 59], [60, 79], [80, 99], [100, 119], [120, 139], [140, 159], [160, 179], [180, 199], [200, 219], [220, 239], [240, 259], [260, 271]]
```

This section defines parameters for the MSA generation, CD-Hit clustering, minimum length of query hit, minimum sequence identity, max number of hits to consider, pssm region size (if region_size has a value then the manual_regions will be ignored).

```
[combinepact]
numdatasets: 1
dataset_1: LGK_Triple
[LGK_Triple]
file1: ./pact/tests/datasets/lgk.1/LGK_Triple_12_SSM_fitness.pact
file2: …
```

The [combinepact] section defines datasets constituted of multiple .pact files. The dataset names must match the [name] of the section with the files listed.

```
[analysis_sitefitness_homology]
dataset_x: site_frequencies
scatter: True
dataset_y: LGK_Triple
y_column: sd_from_wt
y_threshold: 2
x_axis_label: Frequency
y_axis_label: Number of 2SD Mutations
x_axis_min: -0.1
x_axis_max: 1.1
y_axis_min: -0.1
y_axis_max: 1.1
regression: True
```

The protocol will count beneficial, neutral, and deleterious classifications based on the y_column (fitness or sd_from_wt) and y_threshold for observed frequencies of 1, ≥0.9, ≥0.75, ≥0.5, ≥0.25, ≤0.5, and ≤0.25. It will then plot a scatterplot of the highest site frequency on the x-axis and the number of mutations that are ≥ the threshold value.

Note S5: Workflow explanation for the *structure_analysis* protocol to calculate the burial distance, distance to active site, contact number, and fraction burial of a residue for a given pdb structure.

This protocol is capable of calculating:
1) The per-residue fraction burial based off RASA.
2) The per-residue burial distance to surface.
3) The per-residue distance to active site ligands or residues.
4) The per-residue contact number.
5) The per-residue distance to interface.

```
[pact]
pact_config_version: 2018.6
pact_protocol: structure_analysis
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
pdb_import: True
burial_distance: False
distance_to_active: True
contact_number: True
interface_distance: False
```

The [workflow] section defines which steps to perform by either True or False. For this protocol, burial_distance is disabled and will be enabled in the next release of PACT pending a re-write to use vectorization within numpy. Pdb_import is required and cannot be disabled.

```
[global]
directory: ./pact/tests/structure_analysis/
output_prefix: lgk
```

The [global] section defines the working directory and the output prefix.

```
[pdb_import]
numpdb: 1
file_1: 4zlu.pdb
```

This section lists the filenames of pdb files within the working directory. This automatically calls DSSP and the ASA is converted to RASA using values by (Tien, et al., 2013). Fraction burial is 1-RASA and negative values are set to zero.

```
[interface_distance]
pdb_file: 4zlu_cleanf2.pdb
main_chain: A
secondary_chains: B
```

The minimum Euclidean distance from any residue is calculated to atoms in the secondary chain as defined in the config file.

```
[burial_distance]
pdb_file: 4zlu_cleanf2.pdb
chains: A
classifer_chain: A
num_points: 30
```

For each atom within the PDB file, a Fibonacci sphere is created with the number of points given as input. Atoms with points that are not within the radius of any neighboring atom's van der Waals radius plus the radius of water approximate the surface. The average minimum side-chain distance to any surface point is defined as the burial distance.

```
[distance_to_active]
pdb_file: 4zlu_cleanf2.pdb
atoms: CA, CB
chains: A
active_type: ligands
active_ligands: ADP,MG,4PW
active_residues: 212
active_chains: A
classifer_chain: A
```

The Euclidean distance from any active site ligand or residue is calculated to residue atoms defined in the config file.

```
[contact_number]
pdb_file: 4zlu_cleanf2.pdb
atom: CA
distance: 10
chains: A
classifer_chain: A
```

The contact number is the number of residues within a certain distance. This approximates the packing density of the residue as located within the structure.

```
[structure_analysis]
pdb_file: 4zlu.pdb
chain: A
```

This section is used to define which pdb file to use for the report and which chain to use.

Note S6: Workflow explanation for the *Shannon_entropy* protocol to measure the residue-specific entropy based off the log_2 enrichment values. The Shannon entropy calculation and modifications for the fraction of observed variants per site is outlined in (Kowalsky, et al., 2015).

This protocol is capable of calculating:
1) The per-residue Shannon entropy.
2) The theoretical per-residue Shannon entropy.
3) The ratio of total Shannon entropy.

```
[pact]
pact_config_version: 2018.6
pact_protocol: shannon_entropy
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
combinepact: True
shannon_entropy: True
```

This section has no effect as both are required.

```
[global]
wtaa: MPIATS…
directory: ./pact/tests/shannon_entropy/
output_prefix: lgk-wt
```

This section defines the protein amino acid sequence, the output directory, and the output prefix.

```
[combinepact]
numdatasets: 1
dataset_1: LGK_wt
```

This section defines the names of the pact file collections.

```
[LGK_wt]
file1: Tile1_SSM_fitness.pact
file2: …
```

The name of this section in brackets must match whatever is listed in the [combinepact] section. Each line provides the path to a .pact file.

```
[shannon_entropy]
dataset: LGK_wt
mutation_type: single
```

The dataset decides which dataset to use and the mutation type (single or multiple) is used to calculate the theoretical entropy.

Note S7: Workflow explanation for the *back_to_consensus* protocol to calculate the probability of a mutation classification versus the degree of wild-type conservation in sequence homologs.

This protocol is capable of calculating:
1) Basal rates of pact fitness datasets for given mutation classifications.
2) The quantity of mutation classifications per PSSM bin (<0, 0 to 2, and ≥3).
3) CSV dataset of location, wild-type residue, wild-type PSSM and percentage observed, maximum residue PSSM and percentage value, number of mutations above a PSSM value of 0 and the number of mutations with a non-zero observed percentage, and binary output if wild-type is conserved.
4) The wild-type PSSM and percentage observed value at the residue for each dataset mutation class (i.e. for mutation classification BEN the wild-type PSSM and percentage observed at that residue).
5) Same as above but cross-comparing two different datasets.
6) The count of mutation classifications at sites where wild-type is not conserved and the mutation is the conserved or any observed in sequence homologs.
7) The count of mutation classifications at sites where wild-type is not conserved and the mutation is not observed in sequence homologs.
8) Same as #6 above but separated by fraction burial of the residue.

```
[pact]
pact_config_version: 2018.6
pact_protocol: back_to_consensus
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
combinepact: True
basal_count: True
blastp_align_filter: False
pssm: False
pssm_reader: True
pdb_import: True
consensus: True
```

Combinepact is required and cannot be turned on or off. Basal_count will calculate the basal library counts of classification labels. Blastp_align_filter and pssm will perform the multiple sequence alignment (MSA), MSA similarity filtering, PSSM generation, and weighted observed percentages calculation. PSSM_reader will read a stored sequence homology dataset (i.e. blastp_align_filter and pssm steps can be disabled after generation). Pdb_import is not required but if enabled will calculate the sequence homology at surface versus buried residues. Consensus is required and constitutes the major workflow.

```
[global]
wtaa: MPIAT…
directory: ./pact/tests/back_to_consensus/
output_prefix: enzyme_homology_lgk
```

The wild-type amino acid sequence, the working directory, and the output prefix are defined here.

```
[combinepact]
numdatasets: 2
dataset_1: LGK_Triple
dataset_2: LGK_wt
[LGK_Triple]
file1: fitness.pact
file2: …
[LGK_wt]
file1: fitness.pact
file2: …
```

The [combinepact] section defines datasets constituted of multiple .pact files. The dataset names must match the [name] of the section with the files listed.

```
[variant_classification]
class_column: sd_from_wt
class_threshold: 1.5
```

Currently this protocol will classify mutations as BEN (beneficial), NEU (neutral), and DEL (deleterious) based on a 'column' output from the fitness dataset (for column names see the .tsv output however the two major ones are 'fitness' and 'sd_from_wt'). The threshold defines the value from the column as a threshold (in the above example neutral is +/- 1.5 SD from zero).

```
[blastp_align_filter]
processes: 2
cdhit_clustering_threshold: 0.98
ncbi_xml: J88W49S1014-Alignment.xml
minquerylen: 0.6
minseqid: 0.35
nummaxhits: 500
[pssm]
region_size: 20
manual_regions: [[0, 19], [20, 39], [40, 59], [60, 79], [80, 99], [100, 119], [120, 139], [140, 159], [160, 179], [180, 199], [200, 219], [220, 239], [240, 259], [260, 271]]
```

This section defines parameters for the MSA generation, CD-Hit clustering, minimum length of query hit, minimum sequence identity, max number of hits to consider, pssm region size (if region_size has a value then the manual_regions will be ignored).

```
[consensus]
dataset_x: LGK_Triple
dataset_y: LGK_wt
pdb_file: 4zlu_cleanf2.pdb
frac_burial: 0.85
chain: A
```

The consensus section defines which pdb file, chain, and fraction burial value (if pdb_import is enabled). This section also defines which two datasets to use when cross-comparing dataset mutation classifications and consensus values.

```
[pdb_import]
numpdb: 1
file_1: 4zlu_cleanf2.pdb
```

This will import a pdb file.

Note S8: Workflow explanation for the *pact_vs_pact* protocol.

This protocol is capable of:
1) Plotting two PACT per-mutation datasets against each other.
2) Color-coding plotted data points based on an analysis (currently fraction burial of residue).

```
[pact]
pact_config_version: 2018.6
pact_protocol: pact_vs_pact
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
combinepact: True
pdb_import: True
classifier_color: True
setvsset: True
```

Neither combinepact nor setvsset section cannot be disabled. If coloring based on a structural features then pdb_import and classifier_color must be enabled.

```
[global]
wtaa: MPIATS…
directory: ./pact/tests/pact_vs_pact/
output_prefix: pact_vs_pact
```

This section defines the wild-type amino acid sequence, working directory, and the output prefix.

```
[combinepact]
numdatasets: 2
dataset_1: LGK_Triple
dataset_2: LGK_wt
[LGK_Triple]
file1: ./pact/tests/datasets/lgk.1/LGK_Triple_12_SSM_fitness.pact
file2: ….
[LGK_wt]
file1: ./pact/tests/datasets/lgk_wt/LGK_WT_1_SSM_fitness.pact
file2: ….
```

The [combinepact] section defines datasets constituted of multiple .pact files. The dataset names must match the [name] of the section with the files listed.

```
[pdb_import]
numpdb: 1
file_1: 4zlu_cleanf2.pdb
```

This will import a pdb file.

```
[classifier_color]
dataset: LGK_wt
classifier: pdb
```

```
pdb_file: 4zlu_cleanf2.pdb
pdb_chain: A
classifer_key: frac_burial
burial_color: red
burial_value: 0.85
burial_equality: >=
burial_othercolor: blue
```

This section defines which PDB file to use, what property to classify on, the value and color for that value and the opposite value, and equality of that value. Future PACT releases will support other analyses offered by the distribution (such as contact number, burial distance, type of mutation, etc).

```
[setvsset]
dataset_x: LGK_Triple
dataset_y: LGK_wt
x_column: sd_from_wt
y_column: sd_from_wt
ref_threshold: 0
sel_threshold: 0

output_csv: false
shared_counts: false
regression: false

xy_scatter: standard
xy_scatter_type: standard
x_axis_label: LGK_Triple FM
x_axis_min: -10
x_axis_max: 10
y_axis_label: LGK_wt FM
y_axis_min: -10
y_axis_max: 10
1to1line: true
sd_boundaries: 1.5

outlier_threshold: 2
winner_threshold: 2
amino_acid_highlight: *
point_color: classifier_color

headless: false
```

This section defines how the scatterplot will be plotted. The X and Y datasets, data type, and count threshold can be modified. A CSV file of the (X,Y) data is able to be output. A regression line and 1-to-1 line can be plotted. SD_Boundaries will plot lines +/- zero to indicate neutral mutations.

The option “point_color” will color the figure if the option is:

- ‘classifier_color’ defined in the section [classifier_color].
- ‘amino’ will color the amino acids listed in ‘amino_acid_highlight’ red.
- ‘outlier_sign’ will color mutations that change the sign of their fitness value red while no sign change will be blue, ‘winner’.

- ‘winner’ will color mutations red if the shared mutation is above the value in ‘winner_threshold’.
- ‘outlier’ will color mutations red if the difference in fitness values is greater than the threshold in ‘outlier_threshold’.

If ‘xy_scatter_type’ is set to “grouped_location_outlier” will plot each point as the mean of fitness values at a location that the difference between the selections are greater than what is defined in ‘outlier_threshold.’

Note S9: Workflow explanation for the *pact_vs_feature* protocol.

This protocol is capable of calculating:
1) Perform a T-Test of groups of amino acids.
2) Count the number of mutations above a certain threshold within a dataset.

```
[pact]
pact_config_version: 2018.6
pact_protocol: pact_vs_feature
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
combinepact: True
aa_compare_ttest: True
threshold_count: True
```

The combinepact cannot be disabled.

```
[global]
wtaa: MPIAT…
directory: ./pact/tests/pact_vs_feature/
output_prefix: pact_vs_feature
```

This section defines the wild-type amino acid sequence, working directory, and the output prefix.

```
[combinepact]
numdatasets: 1
dataset_1: LGK_Triple
[LGK_Triple]
file1: ./pact/tests/datasets/lgk.1/LGK_Triple_12_SSM_fitness.pact
file2: ./pact/tests/datasets/lgk.1/LGK_Triple_34_SSM_fitness.pact
file3: ./pact/tests/datasets/lgk.1/LGK_Triple_56_SSM_fitness.pact
file4: ./pact/tests/datasets/lgk.1/LGK_Triple_78_SSM_fitness.pact
file5: ./pact/tests/datasets/lgk.1/LGK_Triple_910_SSM_fitness.pact
file6: ./pact/tests/datasets/lgk.1/LGK_Triple_11_SSM_fitness.pact
```

The [combinepact] section defines datasets constituted of multiple .pact files. The dataset names must match the [name] of the section with the files listed.

```
[threshold_count]
dataset: LGK_Triple
column: fitness
cutoff: 0.22
```

This section is used to count the number of mutations above a cutoff value for a given ‘column’ (fitness or sd_from_wt).

```
[aa_compare_ttest]
dataset: LGK_Triple
group_a: *
```

group_b: FWYPMILVAGCSTNQDEHKR
group_a_title: Nonsense
group_b_title: Missense
exclude_wt: True
y_axis_label: Fitness metric of variant
column: fitness
headless: false

This section will perform a unpaired parametric t-Test with Welch's correction on two groups of amino acids. This is typically used to measure the probability of nonsense versus missense mutations distributions.

Note S10: Workflow explanation for the *tools* protocol for the additional tools included with the PACT software distribution.

This protocol is capable of:
1) Calculating degenerate codons that have the lowest amount of unwanted amino acids and stop codons.
2) Swapping codons to optimized synonymous codons.
3) Splitting FASTQ files based on a shared starting 10 amino acid sequence.
4) Converting FASTQ files to FASTA format.
5) Designing mutagenic primers for nicking mutagenesis.
6) Converting CSV files with previously calculated fitness data to .pact format for analysis.

```
[pact]
pact_config_version: 2018.6
pact_protocol: tools
```

The [pact] section has the version of the software (the version of the config file must match the software), and which protocol to use (this list is kept in ./pact/pact_protocols.ini).

```
[workflow]
codon_condenser: True
codon_swap: False
fastq_split: False
fastq_to_fasta: False
primer_design: False
convert_csv_to_pact: True
```

The [workflow] section defines which steps to perform by either True or False.

```
[global]
directory: ./pact/tests/tools/
output_prefix: tools
```

The [global] section defines the working directory and what the output prefix is.

```
[codon_condenser]
list_aminos: HKD
codon: NNN
```

Codon_condenser takes in two different inputs: 1) a specific codon (all base and degenerate codes are supported: GATCRYMKSWHBVDN), or 2) a list of amino acids desired at a particular site. If a codon is given then the script will output the total number of codons, amino acid encoding codons, and stop codons. It will then break down by amino acid and output the number, percentage of all possible codons, and the specific non-degenerate codon. If a list of amino acids is given then it will return the potential degenerate codons sorted by the highest percentage of codons with wanted amino acids and lowest percentage of stop codons.

If a codon is given, the script will display the bases at each position of the codon, the total number of codons, and then list the amino acid to codon possibilities.

The Command line for codon_condenser when looking at a NNK codon.
>python codon_condenser.py -c NNK

If a list of amino acids is given, the list of degenerate codons are given that provides the lowest number of non-desired amino acids and stop codons. If a single amino acid is given then the fraction usage by yeast and human cells is given in the far right two columns. A tab character separates each column, and the columns are sorted by the highest percentage of codons of wanted amino acids and the lowest percentage of stop codons.

Command line for codon_condenser when looking at either a single amino acid or multiple amino acids.
python codon_condenser.py -a ASDK

[codon_swap]
dna_sequence:

A DNA sequence is given as the input then a synonymous sequence is given as the output. Currently, the synonymous codon used is optimized for *E. coli*.

[fastq_split]
forward_fastq: R1.fastq
reverse_fastq: R2.fastq
directory:
cutoff: True

A forward and reverse FASTQ file is given as the input. The first 10 bases are used to split each read. If the cutoff is enabled, only 10-mer sequences with at least 400 reads will be saved in new FASTQ files.

[fastq_to_fasta]
fastq_file: File.fastq

Each FASTQ read is converted to a uniquely numbered FASTA formatted read. No other quality filtering is performed.

[primer_design]
processes: 4
dna_sequence:
mutated_codons: [[1,'n',439]]
constant_length: 60

A training set of 750 primers (with NNN and NNK degenerate codons) successfully incorporated by nicking mutagenesis (Klesmith, et al., 2017; Wrenbeck, et al., 2016) was used to calculate classifiers for: overall primer length, overall GC content, length and GC content on each side of the degenerate codon, melting temperature and Phusion corrected melting temperature. Sequences that pass a filter and have the highest score based on the ideal primer properties are then accepted as the design. Primers with tied scores are then scored on the free energy cost of a mismatch versus a perfectly matched template based on the nearest neighbor approach (SantaLucia, 1998). Classifier equations, classifier range and

averages, and scoring weights are listed in **Table S9**. Within the config file the number of processes to use, the DNA sequence, the range of mutated codons can be defined, and a constant length if required (if defined it will set the required length to the defined value).

```
[convert_csv_to_pact]
numdatasets: 1
dataset_1: dataset_name

[dataset_name]
file: <filename.csv>
wtaa:
location: location
mutation: mutation
fitness: normalized_fitness
starting_index: 1
```

To convert a CSV file with location/mutation/fitness data to .pact format the convert_csv_to_pact tool can be used. Similar to combinepact in other protocols, this protocol defines the number of datasets and names then looks at the [section] with the name for the details. The file is the full or relative path to the csv file; wtaa is the amino acid sequence; location/mutation/fitness is the column names to associate with; and starting_index is the first numbered location in the dataset.

Note S11: A list of external software packages required for individual classifiers and protocols. The path to external programs is kept within the pact_external_programs.ini file within the ./pact/ folder.

Protocols: Any protocol that requires sequence homology.
Programs: psiblast version 2.6.0+ (Altschul, et al., 2009)
cd-hit version 4.6.7 (Li and Godzik, 2006)
muscle version 3.8.31 (Edgar, 2004)

Protocols: Any protocol that requires structural information.
Programs: DSSP 2.0.4 (Kabsch and Sander, 1983)

Example config file: ./pact/pact_external_programs.ini
[programs]
dssp: ./external/Windows/dssp-2.0.4-win32.exe
psiblast: ./external/Windows/ncbi-blast-2.6.0+/bin/psiblast.exe
cdhit: ./external/Windows/cd-hit-v4.6.7-2017-0501/cd-hit.exe
muscle: ./external/Windows/muscle3.8.31_i86win32.exe

Note S12: Explanation of mathematical equations incorporated by the *fitness* protocol.

Fitness metrics

The base metric of mutational function in most deep mutational scanning experiments is the $\log_2$ enrichment of the frequency change of a variant in the selected (or final) population relative to the reference (or initial) population. Equation (1) is the frequency calculation for variant *i* where x_{fi} is the number of sequenced counts of variant *i* in the selected population over the total sequenced counts ($\sum x_{fi}$) of the selected population.

$$f_{fi} = \frac{x_{fi}}{\sum x_{fi}} \tag{1}$$

Equation (2) calculates the $\log_2$ enrichment of variant *i* (ε_i) by taking the $\log_2$ ratio of the frequency of variant *i* in the selected population (f_{fi}) over the frequency in the reference population (f_{oi}).

$$\varepsilon_i = \log_2 \frac{f_{fi}}{f_{oi}} \tag{2}$$

An alternate form of this equation can be written as:

$$\varepsilon_i = \log_2 \left(\frac{x_{fi}}{x_{oi}}\right) - \log_2 \left(\frac{\Sigma x_{fi}}{\Sigma x_{oi}}\right) \tag{3}$$

Three fitness metrics (ζ) are included within the PACT workflow. The fitness metric in Equation (4) is the is the $\log_2$ enrichment of a variant *i* (ε_i) minus the $\log_2$ enrichment of the wild-type (ε_{wt}) sequence (Klesmith, et al., 2017). This can be applied to all selections including growth or FACS based screens if the other fitness metrics are not desired.

$$\zeta_i = \varepsilon_i - \varepsilon_{wt} \tag{4}$$

For growth selections we can express the fitness metric as the growth rate of variant *i* in the population normalized to the growth rate of the wild-type variant in the population (Kowalsky, et al., 2015)

$$\zeta_i = \log_2 \left(\frac{\mu_i}{\mu_{wt}}\right) \tag{5}$$

where the specific growth rate (μ) for variant *i* can be defined in Equation (6) as the natural log of the ratio of the final (x_{fi}) to initial (x_{oi}) variant cell concentration divided by the total time (*t*) of growth.

$$\mu_i = \ln \left(\frac{x_{fi}}{x_{oi}}\right) \frac{1}{t} \tag{6}$$

Using the alternate form of the enrichment equation defined in (3) we can write the change of culture density as the number of average culture doublings (g_p):

$$g_p = \log_2\left(\frac{\Sigma x_{fi}}{\Sigma x_{oi}}\right) \quad (7)$$

Combining Equations (6) and (3) and redefining time (*t*) as a function of g_p (7) and the bulk average growth rate of the population (μ_p) we can write the specific growth rate for a variant as a function of its enrichment ratio:

$$\mu_i = \mu_p\left(\frac{\varepsilon_i}{g_p} + 1\right) \quad (8)$$

Therefore, we can write metric (5) as a function of ε_i and ε_{wt} (the $\log_2$ enrichments of the variant and wild-type respectively) and g_p.

$$\zeta_i = \log_2\left(\frac{\left(\frac{\varepsilon_i}{g_p}\right) + 1}{\left(\frac{\varepsilon_{wt}}{g_p}\right) + 1}\right) \quad (9)$$

PACT will assign a fitness metric of -10 if ε_i is equal or less than $-g_p$ because the calculated growth rate of the variant is zero or negative under this condition and the degree of deleteriousness of the mutation cannot be resolved.

For FACS screens, the fitness metric (10) relates the mean fluorescence of variant *i* (F_i) to the mean fluorescence of the wild-type sequence (Kowalsky, et al., 2015).

$$\zeta_i = \log_2\left(\frac{F_i}{F_{wt}}\right) \quad (10)$$

In metric (6), (ε_i) is the enrichment ratio of the variant *i*, (ε_{wt}) is the wild-type enrichment ratio, (σ) is the standard deviation of the population, and ϕ is the percentage of cells collected of the gating population.

$$\zeta_i = \log_2(e)\sqrt{2}\sigma'\left[erf^{-1}(1 - \phi 2^{\varepsilon wt+1}) - erf^{-1}\left(1 - \phi 2^{\varepsilon i+1}\right)\right] \quad (11)$$

For libraries with multiple mutations, it is more convenient to express the percent frequency change ($\%x_{ij}$) of variant *i* between the selected (*f*) and reference (*o*) populations. Where ($\sum x_{f,ij}$) is the total count of mutation *i* at residue *j* and ($\sum x_{f,kj}$) is the total counts of all mutations *k* (from stop * to Tyr (Y)) at residue *j*.

$$\%x_{ij} = \left(\frac{\sum x_{f,ij}}{\sum_{k=*}^{Y} x_{f,kj}}\right) - \left(\frac{\sum x_{o,ij}}{\sum_{k=*}^{Y} x_{o,kj}}\right) \quad (12)$$

Alternately, the site-wise $\log_2$ enrichment for each amino acid mutation will be calculated from the sum of the mutation combinations. A normalized value using the metric in (4) will be reported where the per-site wild-type $\log_2$ enrichment is utilized instead of the full-length wild-type enrichment. The reported output is the percent frequency change for all 20 amino acids and stop codons at each residue as a heatmap, and the unique mutation $\log_2$ enrichment from equation (2) in a TSV file. Additionally, pairwise analysis using mutual information will be calculated for mutation combination pairs (Dunn, et al., 2008):

$$MI(X,Y) = \Sigma_i \Sigma_j f(x_i y_j) \log_2 \frac{f(x_i y_j)}{f(x_i) f(y_j)} \quad (13)$$

To reduce the bias associated with the entropy with this metric the average product correction (APC) is applied in equation 13 (Dunn, et al., 2008) where MI(x,*) is the average mutual information for mutation x versus the rest of the dataset and MI(*,*) is the overall dataset average MI.

$$MI_p = MI(x_i y_j) - \frac{MI(x,*)MI(*,y)}{MI(*,*)} \quad (14)$$

While approaches that look at the global correlations between residues (Hopf, et al., 2017) may avoid improper transitive conclusions from mutual information (A → B and B → C can improperly imply A → C); however, mutual information is appropriate given the limited size of the library sampling.

Estimation of fitness metric variance

The variance for any fitness metric can be defined as (Klesmith, et al., 2015):

$$\sigma_{\zeta i}^2 = \sigma_{\varepsilon i}^2 \left(\frac{\partial \zeta_i}{\partial \varepsilon_i}\right)^2 + \sigma_{\varepsilon wt}^2 \left(\frac{\partial \zeta_i}{\partial \varepsilon_{wt}}\right)^2 \quad (15)$$

Where ε_i is the $\log_2$ enrichment ratio for variant *i* and ε_{wt} for the wild-type variant. The variance for ε_i can be estimated from Poisson noise:

$${\sigma_{\varepsilon i}}^2 = (\log_2 e)^2 \left(\frac{1}{x_{fi}} + \frac{1}{x_{oi}}\right) \quad (16)$$

Where x_{oi} and x_{fi} are the number of counts in the reference and selected populations respectively. If we combine these two equations into the fitness metric in equation (4) where the enrichment of the wild-type variant is subtracted from the enrichment of the variant *i* we get:

$$\sigma_{\zeta i}^2 = \sigma_{\varepsilon i}^2(1)^2 + \sigma_{\varepsilon wt}^2(-1)^2 \tag{17}$$

For the growth fitness metric in equation (9), we can write the partial derivative with respect to ε_i as:

$$\frac{\partial \zeta_i}{\partial \varepsilon_i} = \frac{\partial}{\partial \varepsilon_i}\left(\log_2\left(\frac{\varepsilon_i}{g_p}+1\right)\right)^2 - \frac{\partial}{\partial \varepsilon_i}\left(\log_2\left(\frac{\varepsilon_{wt}}{g_p}+1\right)\right)^2 \tag{18}$$

Then solve:

$$\frac{\partial \zeta_i}{\partial \varepsilon_i} = \frac{1}{\ln 2}\left(\frac{1}{\varepsilon_i + g_p}\right) \tag{19}$$

This can be repeated for the partial derivative with respect to ε_{wt} and joined back into the variance equation. Similarly, for the FACS normalization equation the variance can be calculated for the metric in equation (11) (Kowalsky, et al., 2015):

$$\sigma_{\zeta i}^2 = \pi\phi^2\sigma'^2\left\{\sigma_{\varepsilon i}^2\left[\left(2^{\varepsilon_i+\frac{1}{2}}\right)e^{erf^{-1}\left(1-\phi 2^{\varepsilon_i+1}\right)^2}\right]^2 + \sigma_{\varepsilon wt}^2\left[\left(-2^{\varepsilon_{wt}+\frac{1}{2}}\right)e^{erf^{-1}\left(1-\phi 2^{\varepsilon_{wt}+1}\right)^2}\right]^2\right\} \tag{20}$$

Note S13: Approximation of the number of experiments for a variant within a FACS screen. We wish to calculate the significance probability value of a variant from the synonymous wild-type codon enrichment distribution to determine if a variant is functionally neutral. Our main assumption is all wild-type synonymous genes have equal protein phenotype. Therefore, the distribution of synonymous codon enrichments can approximate functional neutrality. A useful calculation is an unpaired t-test with Welch's correction of the distribution of the wild-type synonymous codon enrichments versus any given nonsynonymous variant. The wild-type population has a mean ($\bar{x}$), standard deviation (σ), and size (N) calculated from the list of synonymous codon enrichments. For the non-synonymous variant, we can use the enrichment for the mean ($\bar{x}$) and the standard deviation (σ) from the enrichment variance (**Note: S12**). However, for the non-synonymous population we will calculate the expectation value for (N) via the number of experiments for any given variant. In this case, we are defining the number of experiments as the number of chances the variant has the opportunity of being collected.

In any given FACS experiment, cells are typically gated on a given parameter such as binding or display. For example, we have one gate and two variants (A and B). Within the population the actual distribution of counts within and outside the gate were:

	Within Gate	Outside Gate	Total
Variant A	167	3	170
Variant B	2	72	74
Total	169	75	

Each variant has an associated frequency within the collected population:

	Counts	Frequency (f_{oi})
Variant A	167/169	0.99
Variant B	2/169	0.01

If we know the frequency of A and B within the starting population from deep sequencing ($x_{oi}/\sum x_{oi}$) and the total number of cells that passed through the detector that had the opportunity of being collected, we can calculate the expectation value for any variant. Alternately, we could approximate the expectation value for a given variant by the input library design codon frequency and the total number of cells that had the opportunity of being collected. Therefore, the expectation value for FACS (21) can be defined as either:

$$e_{facs,i} = (Number\ cells\ could\ have\ been\ collected) * f_{oi} \quad (21)$$

Where f_{oi} is either:

$$f_{oi} = \frac{x_{fi}}{\sum x_{fi}} \quad (22)$$

$$f_{oi} = frequency\ of\ i\ in\ the\ designed\ library \quad (23)$$

Note S14: Approximation of the number of experiments for a variant within a growth selection. We wish to calculate the probability of observing the data by random chance rather than an underlying difference (*i.e.* statistical significance) in the variant versus synonymous wild-type genes to determine if a variant is functionally neutral. Our main assumption is that the all wild-type synonymous genes have equal protein phenotype. Therefore, the distribution of synonymous codon enrichments can approximate functional neutrality. A useful calculation is an unpaired t-test with Welch's correction of the distribution of the wild-type synonymous codon enrichments versus any given non-synonymous variant. The wild-type population has a mean ($\bar{x}$), standard deviation (σ), and size (N) from the list of synonymous codon enrichments. For the non-synonymous variant, we can use the enrichment for the mean and the standard deviation from the enrichment variance (**Note: S12**). However, for the nonsynonymous population we will calculate the expectation value for (N) via the number of experiments for any given variant. In this case, we are defining the number of experiments as the number of chances the variant has the opportunity of doubling (probability of an experiment or p(e)).

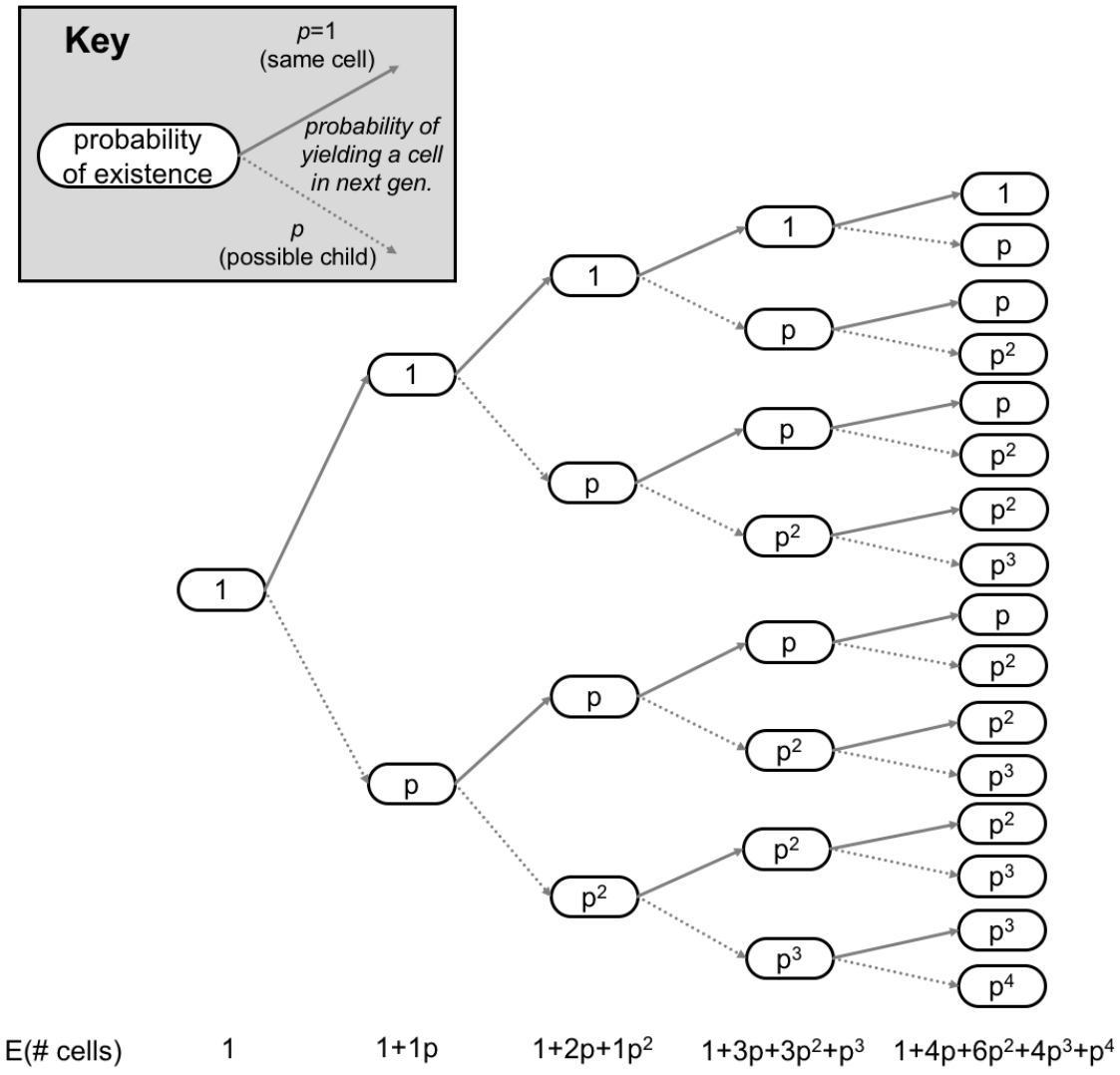
The probability of a cell doubling in one generation is defined as *p*. This can be written as a function of the final cell count (x_f) and the number of generations (g_p):

$$p = x_f^{\frac{1}{g_p}} - 1 \tag{25}$$

The likelihood of the cell existing in the next generation is the product of *p* and the cell's likelihood of existing in the previous generation.

$$p_{gen\ n+1} = p \cdot p_{gen\ n}$$

Thus, the expected numbers of cells can be tracked at each generation.



Interestingly, the coefficients of probabilities of cells existing follow Pascal's triangle. Therefore, because the number of cells in each generation equals the number of experiments for that generation, we can write our expected number of total experiments as:

$$e_{growth,i} = x_{oi} \sum_{m=1}^{g_p} \sum_{k=0}^{m-1} P_{m,k+1} p^k \tag{26}$$

Where the starting number of cells is (x_{oi}), the number of generations is (g_p), and Pascal's number is (P). Where Pascal's number is:

$$P_{m,k+1} = \binom{g_p - 1}{k} = \frac{(g_p - 1)!}{k!\,(g_p - 1 - k)!} \tag{27}$$

Fig. S1: Poisson statistics can help guide deep sequencing and FACS methods. Poisson statistics can be used to calculate the standard deviation of the count of DNA sequences (Whitehead, et al., 2012) or cells collected within a gate (Roederer, 2008). The standard deviation for any single variant is the square root of the number of events of that variant. The relative precision of the frequency ($N^{1/2}$ / N) is plotted below (diamonds). Assay variation is typically greater than ±30% (Roederer, 2008) (dashed line), thus with 12 and greater counts for a given variant the error is more of a consequence of assay errors and less of counting errors.

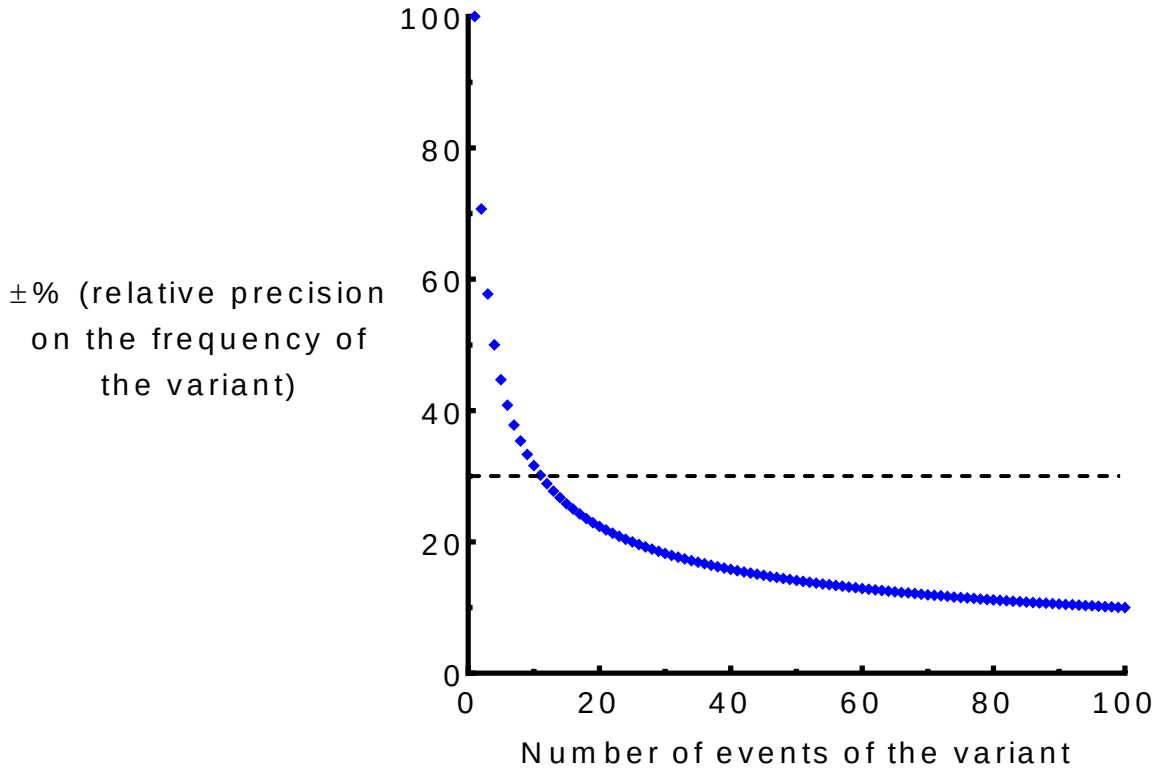


Fig. S2: The percent change of the growth rate of a variant in the sequenced population relative to the wild-type sequence for a given growth fitness metric value. A locally-weighted scatterplot smoothed (LOWESS) curve is fitted to guide the reader.

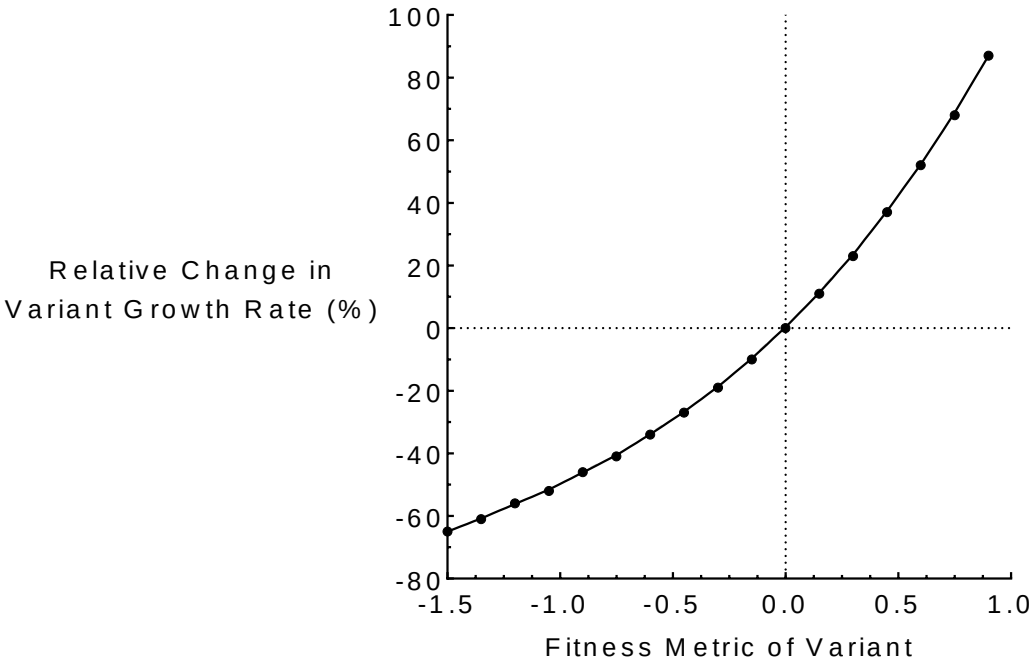


Fig. S3: The per-mutation number of standard deviations from wild-type (z-score) of LGK-WT versus LGK.1 selections. Mutations that have a fraction burial of ≥ 0.85 (core) are in red while others (surface) are in blue. Thresholds at ±1.5 wild-type synonymous SDs are in purple dashed lines. A y = x line is in gray. Mutations are classified as beneficial (>+1.5 z-score, BEN), neutral (within ±1.5 z-score, NEU), or deleterious (<–1.5 z-score, DEL). Deep sequencing datasets on LGK-WT and LGK.1 from (Klesmith, et al., 2015) was reprocessed with the *fitness* protocol followed by the *pact_vs_pact* protocol for graphing and color coding fraction burial. Each point is the z-score an individual mutation shared between the two selections. The z-scores were calculated from the neutral variance as calculated from a Gaussian distribution of synonymous wild-type genes.

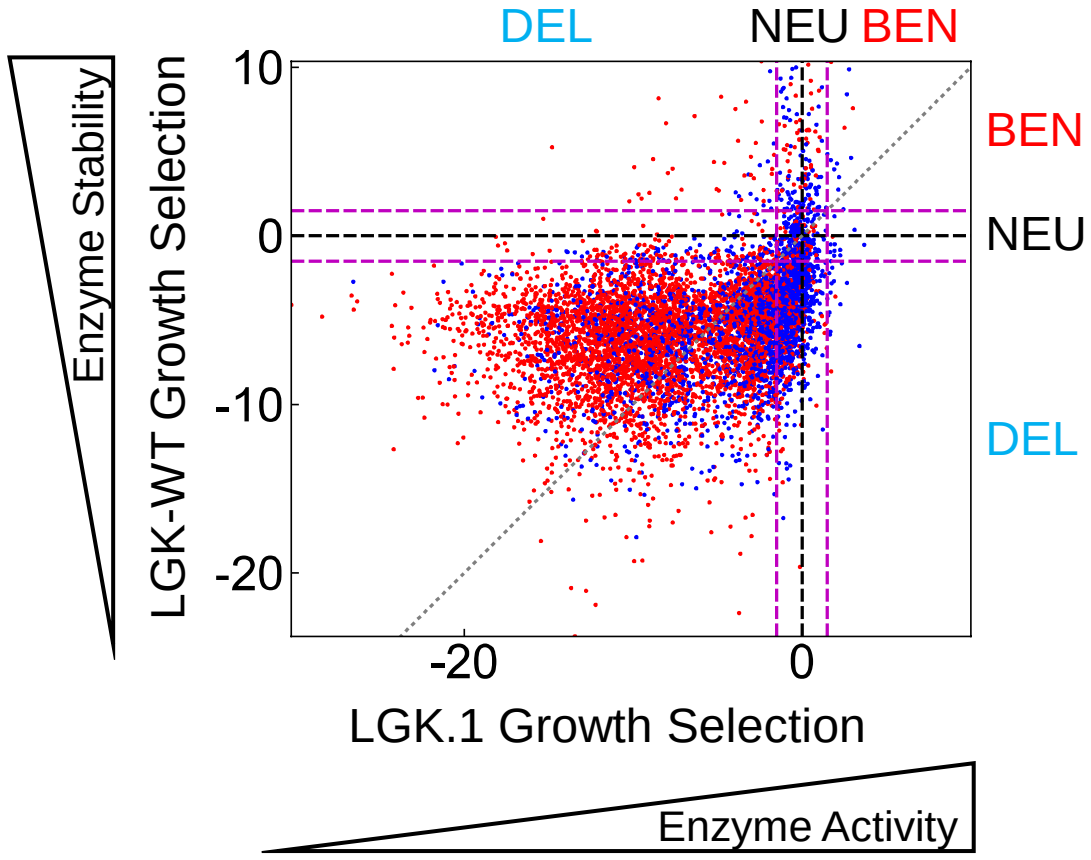


Fig. S4: Frequency of beneficial (BEN, red), neutral (NEU, gray), or deleterious (DEL, blue) versus different consensus features. Deep sequencing datasets on LGK-WT and LGK.1 from (Klesmith, et al., 2015) was reprocessed with the *fitness* protocol. Homologous LGK sequences from a BlastP search (e-value 10^{-4}, CD-Hit clustering threshold 0.98, minimum query length of 0.6, and minimum sequence identity of 0.35) were processed using the *back_to_consensus* protocol to calculate and build a PSSM and sequence frequency dataset of all mutations. The first column lists the results for all mutants. The next column pair are the mutation type frequency for all mutants with PSSM value ≥ 0 or non-zero natural sequence frequency. The following column pairs are limited to sites where wild-type is not consensus. The final column pair considers only the consensus mutations.

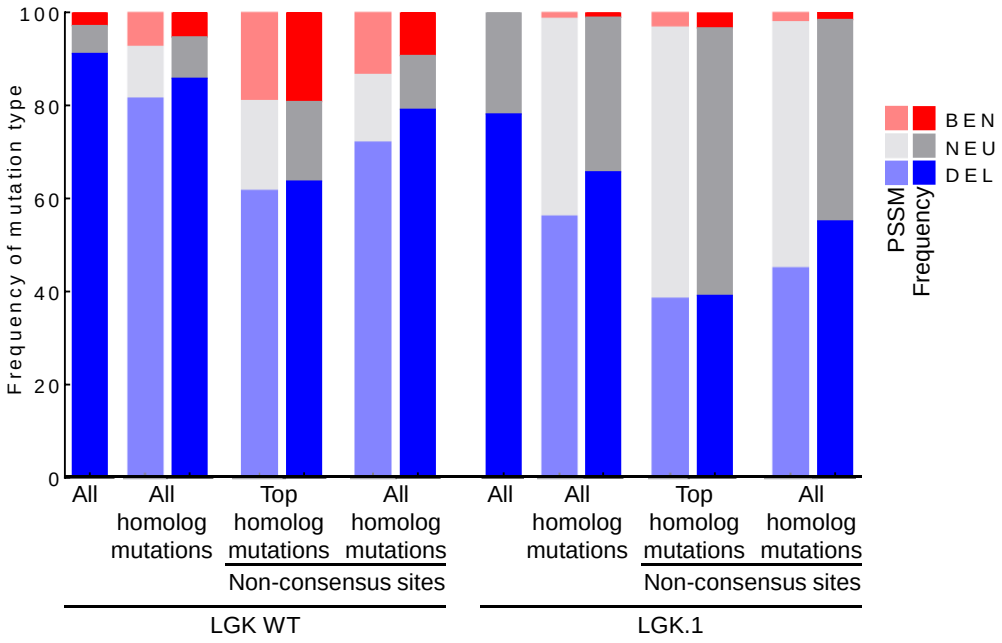


Fig. S5: Frequency of beneficial (BEN, red), neutral (NEU, gray), or deleterious (DEL, blue) versus different consensus features. Deep sequencing datasets on LGK-WT and LGK.1 from (Klesmith, et al., 2015) was reprocessed with the *fitness* protocol. Homologous LGK sequences from a BlastP search (e-value 10^{-4}, CD-Hit clustering threshold 0.98, minimum query length of 0.6, and minimum sequence identity of 0.35) were processed using the *back_to_consensus* protocol to calculate and build a PSSM and sequence frequency dataset of all mutations. The first column lists the results for all mutants. The next column pair are the mutation type frequency for all mutants with PSSM value ≥ 0 or non-zero natural sequence frequency. The following column pairs are limited to sites where wild-type is not consensus. The final column pair considers only the consensus mutations. The top figure is at residues with fraction burial < 0.85 (surface exposed) and the bottom are residues ≥ 0.85 (core).

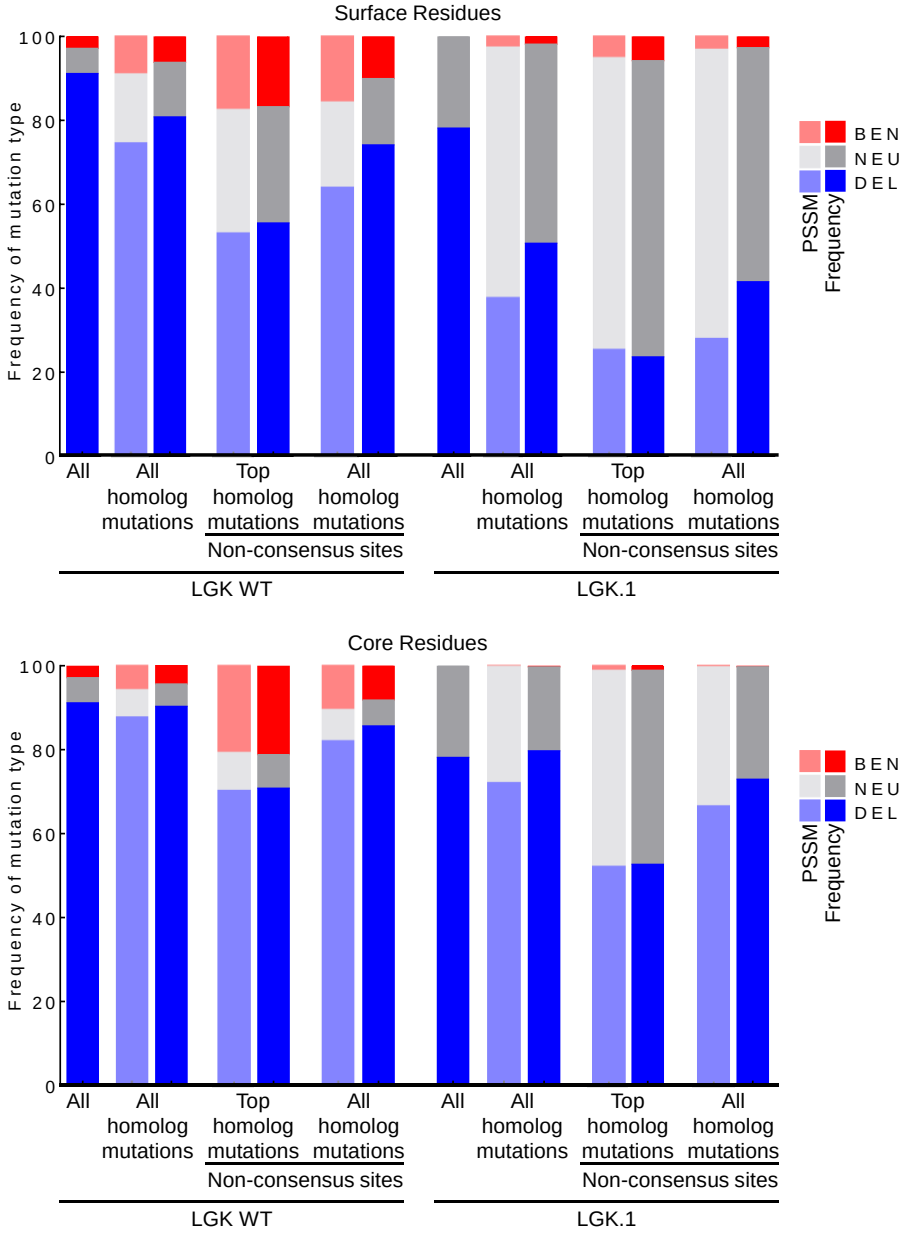


Fig. S6: Optimizing combinations of Bayesian feature probabilities. The predictive performance of all 2^{10} combinations of feature Bayesian probabilities. Each point is a combination of Bayesian feature probabilities (**Table S4**). The combinations in red squares are shared between all subfigures and in **Fig. 2** and **Tables S6-8**. A) The fraction of finding a truly beneficial mutation versus the fraction of a truly deleterious mutation for propionamide if a predicted beneficial or neutral predicted mutation is selected. B) The fraction of truly beneficial mutations if a predicted beneficial or neutral mutation is selected for both selections.

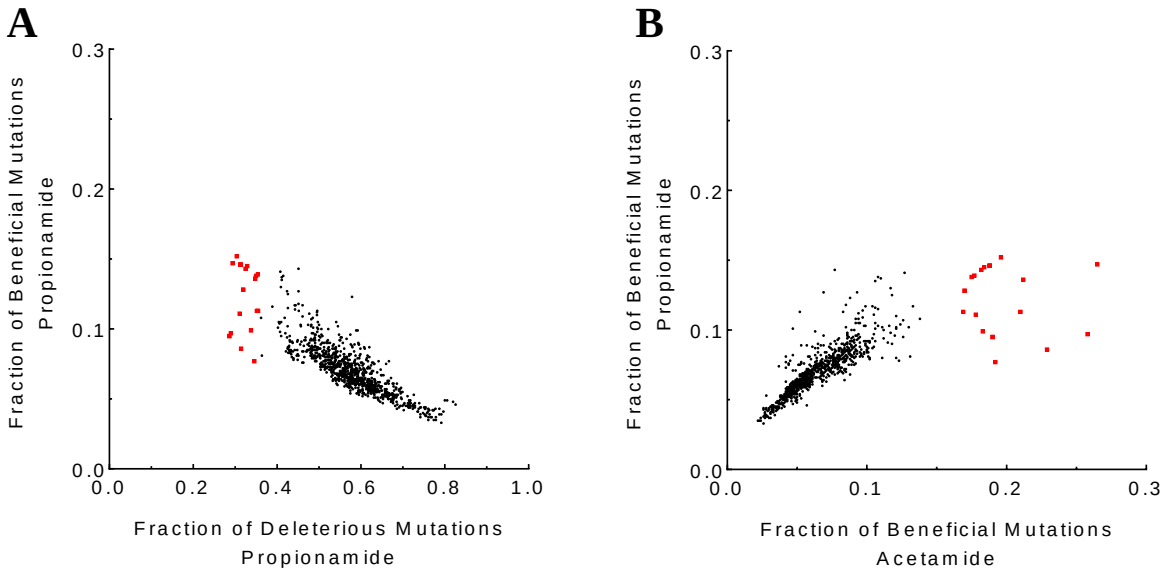


Fig. S7: The number of beneficial mutations found within Bayesian classifications from combinations of features. Each data point is a combination of feature Bayesian probabilities. The two amidase datasets are binned on if a mutation is predicted to be beneficial (BEN) and beneficial or neutral (BEN + NEU). The 5 to 95 percentile is plotted.

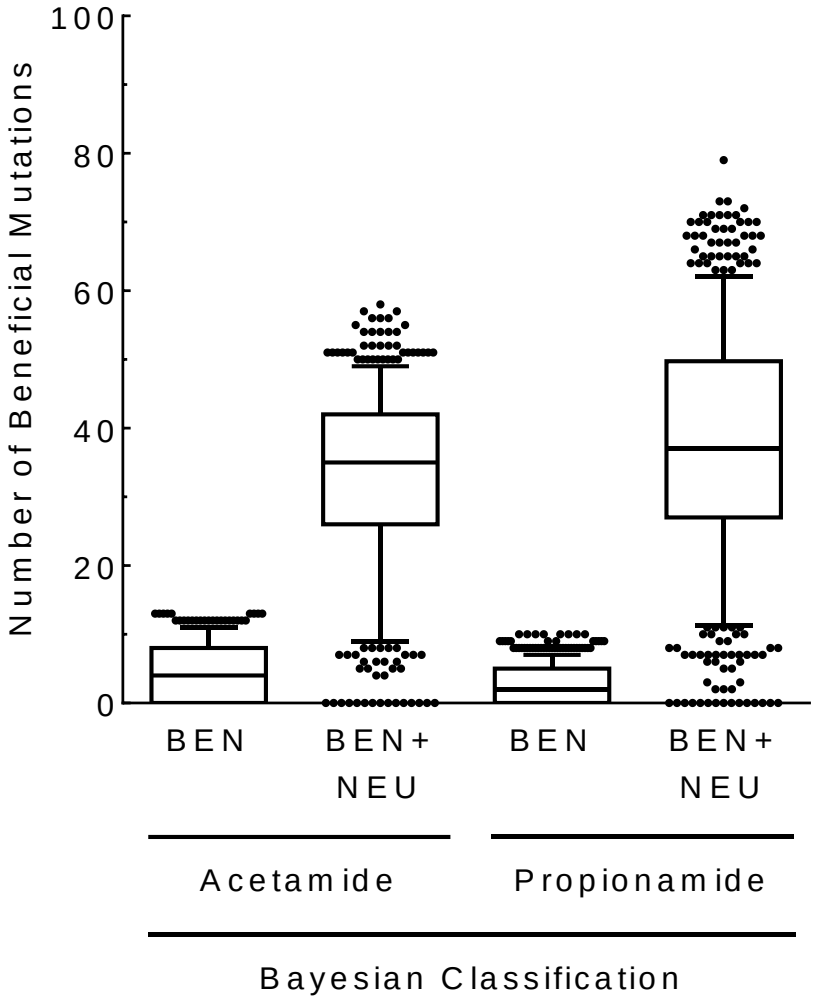


Table S1: Summary of fitness metrics included within the *fitness* protocol and multi-site frequency change. Three fitness metrics (ζ) are included within the pact *fitness* protocol and can be used on single-site and multiple-site library types. For multiple-site libraries a second calculation is utilized to get the percent frequency change of a mutation using all mutation combinations.

Equation	Inputs
(Any screen type) The $\log_2$ enrichment of variant *i* normalized to the wild-type enrichment. $$\zeta_i = \varepsilon_i - \varepsilon_{wt}$$ Citation: (Klesmith, et al., 2017)	ε_i = $\log_2$ enrichment of variant *i* ε_{wt} = $\log_2$ enrichment of wild-type
(Growth selections) The growth rate of variant *i* normalized to the wild-type growth rate. $$\zeta_i = \log_2\left(\frac{\left(\frac{\varepsilon_i}{g_p}\right)+1}{\left(\frac{\varepsilon_{wt}}{g_p}\right)+1}\right)$$ Citation: (Kowalsky, et al., 2015)	ε_i = $\log_2$ enrichment of variant *i* ε_{wt} = $\log_2$ enrichment of wild-type g_p = number of culture doublings
(Flow cytometry screens) The mean fluorescence of variant *i* versus the mean fluorescence of the wild-type sequence. $$\zeta_i = \log_2(e)\sqrt{2}\sigma'\begin{bmatrix} erf^{-1}(1-\phi 2^{\varepsilon wt+1}) \\ -erf^{-1}(1-\phi 2^{\varepsilon i+1}) \end{bmatrix}$$ Citation: (Kowalsky, et al., 2015)	ε_i = $\log_2$ enrichment of variant *i* ε_{wt} = $\log_2$ enrichment of wild-type σ = standard deviation of collected population ϕ = percentage of cells collected of the gating population
(Multi-site only) The percent frequency change of variant *i* between the selected and reference population. $$\%x_{ij} = \left(\frac{\sum x_{f,ij}}{\sum_{k=*}^{Y} x_{f,kj}}\right) - \left(\frac{\sum x_{o,ij}}{\sum_{k=*}^{Y} x_{o,kj}}\right)$$	$\sum x_{f,ij}$ = Sum of mutation *i* counts at location *j* in the selected population $\sum x_{f,kj}$ = Sum of all mutations *k* counts at location *j* in the selected population x_f = final population x_o = reference population

Table S2: Deep sequencing coverage of LGK-WT and LGK.1 for the reference and selected libraries for synonymous and nonsynonymous variants. LGK-WT tile sizes are in 40 amino acid lengths and LGK.1 in 80 amino acid lengths. Datasets were taken from (Klesmith, et al., 2015) and processed using the *fitness* protocol.

		SD of Synonymous Wild-Type Codons at different reference read thresholds			Synonymous Reads		Nonsynonymous Reads		Fold Nonsynonymous Codon Coverage	
Tile	Log2 WT	All	≥12 Reads	≥30 Reads	Ref	Sel	Ref	Sel	Ref	Sel
LGK-WT										
1	-1.256	0.15	0.10	0.10	33,755	34,537	97,615	288,101	38.7	114.3
2	-0.724	0.11	0.10	0.09	30,494	48,542	84,992	254,433	33.7	101.0
3	-0.776	0.12	0.11	0.08	33,789	41,599	104,585	266,130	41.5	105.6
4	-0.444	0.12	0.12	0.08	30,401	50,332	97,341	238,073	38.6	94.5
5	-1.013	0.14	0.11	0.08	19,580	29,353	74,293	243,789	29.5	96.7
6	-0.828	0.12	0.10	0.09	42,613	47,640	119,690	267,040	47.5	106.0
7	-0.147	0.10	0.11	0.04	29,296	29,990	104,307	115,809	41.4	46.0
8	-0.537	0.08	0.08	0.07	35,675	46,299	110,760	223,323	44.0	88.6
9	-0.708	0.07	0.07	0.06	40,606	45,289	90,307	185,031	35.8	73.4
10	-0.959	0.09	0.09	0.07	40,275	43,723	120,769	283,344	47.9	112.4
11	-0.555	0.10	0.09	0.08	44,043	66,444	119,270	277,891	48.5	113.1
LGK.1										
1	0.835	0.10	0.07	0.07	96,126	396,858	160,933	197,692	31.9	39.2
2	1.253	0.15	0.07	0.07	78,849	556,319	194,698	252,631	38.6	50.1
3	1.033	0.09	0.06	0.05	206,635	604,288	279,183	91,998	55.4	18.3
4	1.142	0.13	0.08	0.07	144,633	648,652	248,378	150,906	49.3	29.9
5	1.184	0.11	0.08	0.08	123,522	547,553	213,408	110,968	42.3	22.0
6	0.659	0.11	0.08	0.07	87,393	453,643	142,253	301,597	57.9	122.8

Table S3: Thermal and catalytic measurements of published LGK mutants individually produced and tested (Klesmith, et al., 2015). Mutations that are included in LGK.1 are marked with an asterisk. The active site residue is D212 for this enzyme. Mutations are classified as beneficial (BEN), neutral (NEU), or deleterious (DEL) depending on the mutation's z-score with neutral defined within ± 1.5 SD. Fitness metric values and number of standard deviations from wild-type are listed. Mutations indicated by asterisks are the three mutations that form the LGK.1 variant.

Variant	$\Delta T_{m,app}$ (°C)	Catalytic Efficiency Rel. To LGK-WT	LGK-WT fitness metric	LGK-WT z-score	LGK.1 fitness metric	LGK.1 z-score	LGK-WT Class	LGK.1 Class	Matches categorical expectations? (**Fig. S3**)
LGK-WT	0	1							
LGK.1	5.1	1.07 ± 0.08							
Predicted mutations that improve stability and activity									
G359R	1.1	1.86 ± 0.10	0.74	9.88	0.31	2.79	BEN	BEN	YES
Predicted mutations that improve stability at the cost of activity									
H113G	4.9	0.01 ± 0.00	0.49	4.07	-1.52	-9.93	BEN	DEL	YES
I167H	9.8	0.17 ± 0.01	0.99	7.10	-0.57	-6.36	BEN	DEL	YES
T268C	4	0.35 ± 0.04	0.85	8.25	-0.61	-4.79	BEN	DEL	YES
Q369L	3.4	0.09 ± 0.01	0.69	7.52	-0.43	-3.94	BEN	DEL	YES
Predicted mutations that improve stability that are neutral on activity									
V11P	0.1	1.06 ± 0.19	0.90	6.10	-0.03	-0.28	BEN	NEU	YES
R94H	1.9	1.10 ± 0.12	0.92	7.66	0.15	0.99	BEN	NEU	YES
L140I*	2.2	1.11 ± 0.15	0.86	7.14	0.00	0.00	BEN	NEU	YES
S142A*	0.8	1.26 ± 0.14	0.86	7.12	0.00	0.00	BEN	NEU	NO
C194T	6	0.70 ± 0.11	0.91	6.53	-0.09	-1.03	BEN	NEU	NO
M257H	-0.4	1.43 ± 0.08	0.91	8.78	-0.14	-1.09	BEN	NEU	NO
A373C*	0.5	1.33 ± 0.09	0.83	9.01	0.00	0.00	BEN	NEU	NO
Predicted mutations that are neutral on stability but are deleterious on activity									
I167N	2.1	0.01 ± 0.00	-0.12	-0.87	-0.44	-4.96	NEU	DEL	NO
Predicted mutations that are deleterious on stability or deleterious on activity									
D212A	1.4	0.00 ± 0.00	-0.64	-5.20	-0.85	-9.47	DEL	DEL	YES
N217S	0.6	0.44 ± 0.04	-0.20	-1.60	-0.46	-5.13	DEL	DEL	YES

Table S4: Feature counts for LGK-WT versus LGK.1. The basal rate was used for the prior, p(evidence) and p(likelihood) was calculated from the counts for each feature to form the naïve Bayes classifier. We excluded analysis on residues M1 to 9D as these residues were not part of the crystal structure nor potentially the main structure therefore fitness values could be affected by 5' portions of mRNA transcripts (Firnberg, et al., 2014) and not protein characteristics.

	N	BEN	NEU	DEL
Basal Rate	7324	167	1368	5789
Percentage		2%	19%	79%
Contact Number				
≥0 <10	358	22	187	149
Percentage		6%	52%	42%
≥11 <20	3297	102	933	2262
Percentage		3%	28%	69%
20+	3669	43	248	3378
Percentage		1%	7%	92%
Contact Number (20+)				
Non PRO	3494	167	234	3093
Percentage		5%	7%	89%
To/From PRO	300	1	14	285
Percentage		0%	5%	95%
Contact Number (11-20)				
Non PRO	2809	92	797	1920
Percentage		3%	28%	68%
To/From PRO	311	5	24	282
Percentage		2%	8%	91%
Contact Number (0-10)				
Non PRO	472	21	276	175
Percentage		4%	58%	37%
To/From PRO	63	6	23	34
Percentage		10%	37%	54%

	N	BEN	NEU	DEL
Residue Fraction Burial (%)				
≥0 <30	369	10	215	144
Percentage		3%	58%	39%
≥30 <60	851	48	387	416
Percentage		6%	45%	49%
≥60 <90	1660	60	432	1168
Percentage		4%	26%	70%
90 to 100	4444	49	334	4061
Percentage		1%	8%	91%
Distance to Active Site (Å)				
≥0 <5	294	3	37	254
Percentage		1%	13%	86%
≥5 <10	1063	18	86	959
Percentage		2%	8%	90%
≥10 <15	1375	20	198	1157
Percentage		1%	14%	84%
≥15 <20	1812	41	324	1447
Percentage		2%	18%	80%
≥20 <25	1467	34	382	1051
Percentage		2%	26%	72%
≥25 <30	768	41	294	433
Percentage		5%	38%	56%
≥30	145	10	47	88
Percentage		7%	32%	61%

	N	BEN	NEU	DEL
PSSM				
≥ 3	338	35	138	165
Percentage		10%	41%	49%
< 3 & ≥ 0	1184	53	432	699
Percentage		4%	36%	59%
< 0	5465	56	686	4723
Percentage		1%	13%	86%
WT Consensus at Site?				
FALSE	2490	124	664	1702
Percentage		5%	27%	68%
TRUE	4834	43	704	4087
Percentage		1%	15%	85%
Mutation Consensus at Site?				
FALSE	7182	142	1307	5733
Percentage		2%	18%	80%
TRUE	142	25	61	56
Percentage		18%	43%	39%
Homolog Wild-Type Percentage				
0-15	1313	100	417	796
Percentage		8%	32%	61%
16-30	1720	42	501	1177
Percentage		2%	29%	68%
31-45	966	6	131	829
Percentage		1%	14%	86%
46-60	879	7	100	772
Percentage		1%	11%	88%
60+	2446	12	219	2215
Percentage		0%	9%	91%

	N	BEN	NEU	DEL
Homolog Mutation Percentage				
0	4247	33	474	3740
Percentage		1%	11%	88%
1-15	2726	91	747	1888
Percentage		3%	27%	69%
16-30	253	24	111	118
Percentage		9%	44%	47%
31-45	57	10	23	24
Percentage		18%	40%	42%
46-60	24	6	8	10
Percentage		25%	33%	42%
60+	17	3	5	9
Percentage		18%	29%	53%
Max Site Homolog Percentage				
0-15	48	3	14	31
Percentage		6%	29%	65%
16-30	1695	83	644	968
Percentage		5%	38%	57%
31-45	1619	34	285	1300
Percentage		2%	18%	80%
46-60	1226	31	166	1029
Percentage		3%	14%	84%
60+	2736	16	259	2461
Percentage		1%	9%	90%

Table S5: Basal rate of library mutations for all single selections. For LGK-WT and LGK.1 the approximation of neutral (NEU) is ± 1.5 SD (therefore >+1.5SD is beneficial (BEN), <-1.5SD is deleterious (DEL)). While for AmiE neutral is approximated for +/- 10% change in growth which is ± 0.15 fitness metric. Alternately, mutations can be ranked as ≥80% of wild-type as neutral (NEU: ≥ 80% of wild-type), ≥50% of wild-type to be classified as slightly deleterious (S. DEL: ≥ 50% of wild-type), or deleterious (DEL < 50% of wild-type). This classification is included for comparison to previous work (Klesmith, et al., 2017).

	N	BEN	NEU	DEL			N	NEU	S.DEL	DEL
LGK.1	7478	56	1610	5812		LGK.1	7478	2350	2550	2578
Percentage		1%	22%	78%		Percentage		31%	34%	34%
	N	BEN	NEU	DEL			N	NEU	S.DEL	DEL
LGK-WT	6783	209	416	6158		LGK-WT	6783	1066	5270	447
Percentage		3%	6%	91%		Percentage		16%	78%	7%
	N	BEN	NEU	DEL			N	NEU	S.DEL	DEL
AmiE-ACE	6289	88	646	5555		AmiE-ACE	6289	1291	1977	3021
Percentage		1%	10%	88%		Percentage		21%	31%	48%
	N	BEN	NEU	DEL			N	NEU	S.DEL	DEL
AmiE-PRO	6291	120	678	5493		AmiE-PRO	6291	1412	2186	2693
Percentage		2%	11%	87%		Percentage		22%	35%	43%

Table S6: Bayesian feature combinations identified as putatively higher performing. The feature columns match the features listed in **Table S4**. These combinations were identified as potentially higher performing based on low rate of deleterious mutation and high fraction of true beneficial mutations (**Fig. 2**). The combinations listed in this table match the combinations listed in **Table S7**. True or False is if the probability for that feature was considered.

Feature Combinations from **Table S4** Identified as Higher Performing (**Fig. 2**)
(TRUE = Enabled, FALSE = Not Enabled)

Combination Number	PSSM	Fraction Burial	Contact Number	Wild-Type Consensus?	Mutation Consensus?	Distance to Active Site	Mutation Percent	Max Site Percent	Wild-Type Percent	Contact Number/ Prolines
1	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE
2	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
3	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE
5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE
6	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE
7	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
8	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
9	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
10	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
11	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
12	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE

Table S7: Naïve Bayes optimization via testing of combinations of features. Feature combinations that had zero correctly predicted beneficial mutations in the Bayesian predicted BEN class were discarded (N = 11 out of 23). The combination number matches the ones in **Table S6**. All tables are sorted on the number of beneficial mutations per deleterious mutation if a predicted BEN and NEU mutation is selected for the acetamide dataset.

	Dataset													
	Beneficial			Neutral			Deleterious							
	Bayesian Prediction													
Combination Number	BEN	NEU	DEL	BEN	NEU	DEL	BEN	NEU	DEL	# Beneficial per Deleterious in BEN+NEU	% Beneficial in BEN+NEU	% Deleterious in BEN+NEU	% Beneficial in BEN	% Deleterious in BEN
									Acetamide					
1	7	2	79	10	15	621	4	8	5543	0.75	20%	26%	33%	19%
2	8	1	79	20	6	620	11	2	5542	0.69	19%	27%	21%	28%
3	1	8	79	1	25	620	1	12	5542	0.69	19%	27%	33%	33%
4	4	10	74	6	25	615	3	18	5534	0.67	21%	32%	31%	23%
5	4	9	75	6	23	617	3	17	5535	0.65	21%	32%	31%	23%
6	7	1	80	11	15	620	5	8	5542	0.62	17%	28%	30%	22%
7	9	5	74	25	14	607	12	11	5532	0.61	18%	30%	20%	26%
8	9	5	74	26	14	606	13	10	5532	0.61	18%	30%	19%	27%
9	2	11	75	8	28	610	7	15	5533	0.59	18%	31%	12%	41%
10	3	9	76	11	26	609	7	15	5533	0.55	17%	31%	14%	33%
11	9	5	74	15	25	606	7	19	5529	0.54	18%	33%	29%	23%
12	8	6	74	13	26	607	8	18	5529	0.54	18%	33%	28%	28%
									Propionamide					
1	2	5	113	13	12	653	6	8	5479	0.50	15%	30%	10%	29%
2	6	1	113	21	5	652	12	3	5478	0.47	15%	31%	15%	31%
3	2	5	113	0	26	652	1	14	5478	0.47	15%	31%	67%	33%
4	2	7	111	7	27	644	4	19	5470	0.39	14%	35%	15%	31%
5	2	5	113	7	26	645	4	18	5471	0.32	11%	35%	15%	31%
6	3	3	114	13	13	652	7	8	5478	0.40	13%	32%	13%	30%
7	7	4	109	25	15	638	14	11	5468	0.44	14%	33%	15%	30%
8	7	4	109	26	15	637	15	10	5468	0.44	14%	32%	15%	31%
9	2	5	113	7	33	638	8	16	5469	0.29	10%	34%	12%	47%
10	2	6	112	10	28	640	9	16	5468	0.32	11%	35%	10%	43%
11	4	7	109	18	23	637	9	19	5465	0.39	14%	35%	13%	29%
12	4	7	109	16	24	638	9	19	5465	0.39	14%	35%	14%	31%

Table S8: Number of variants predicted, input classification, and false positive rate of finding a deleterious mutation. For LGK-WT and LGK.1 the approximation of neutral (NEU) is ±1.5 SD (therefore >+1.5SD is beneficial (BEN), <-1.5SD is deleterious (DEL)). While for AmiE neutral is approximated for ±10% change in growth which is ±0.15 fitness metric. Alternately, mutations can be ranked as ≥80% of wild-type as neutral (NEU: ≥ 80% of wild-type), ≥50% of wild-type to be classified as slightly deleterious (S. DEL: ≥ 50% of wild-type), or deleterious (DEL < 50% of wild-type). This classification is included for comparison to previous work (Klesmith, et al., 2017). The deleterious mutation rate p(DEL) is based off of finding a deleterious mutation predicted to be beneficial or neutral (BEN+NEU), just beneficial (BEN), or accepted by the filter (Accept). The naïve Bayes classifier assesses the feature probabilities from combination number 1 (**Table S6**).

	Dataset Classification	Neutral (NEU)			Slight DEL (S. DEL)			Deleterious (DEL)			p(DEL) BEN+NEU	p(DEL) BEN
Classifier	Predicted Classification	BEN	NEU	DEL	BEN	NEU	DEL	BEN	NEU	DEL		
Naïve Bayes	AmiE ACE	18	22	1251	3	2	1972	0	1	3020	2.2%	0.0%

	Dataset Classification	Beneficial (BEN)		Neutral (NEU)		Deleterious (DEL)		p(DEL) Accept
Filter	Filter Classification	Accept	Reject	Accept	Reject	Accept	Reject	
Old Filter	LGK.1/LGK-WT	59	108	354	1014	225	5564	35%
Old Filter	AmiE ACE	26	62	155	491	249	5306	58%
New Filter	LGK.1/LGK-WT	9	158	55	1313	7	5782	9.9%
New Filter	AmiE ACE	6	82	32	614	17	5538	31%

	Dataset Classification	Neutral (NEU)		Slight Del (S. DEL)		Deleterious (DEL)		p(DEL) Accept
Filter	Filter Classification	Accept	Reject	Accept	Reject	Accept	Reject	
Old Filter	LGK.1	496	1712	119	2423	23	2551	3.6%
Old Filter	AmiE ACE	265	1026	126	1851	39	2982	9.1%
New Filter	LGK.1	68	2140	2	2540	1	2573	1.4%
New Filter	AmiE ACE	47	1244	7	1970	1	3020	1.8%

Table S9: Feature equations, feature range and averages, and scoring weights used in the primer_design script. 750 primers used with Nicking mutagenesis (Wrenbeck, et al., 2016) with NNN, NNK degenerate codons was used as a training set. A filter is a value that the script will reject the sequence if crossed. A weight is a value given to the sequence if at the mean or within a certain SD. Primers with tied scores are ranked by the nearest neighbor lowest free energy cost of the mismatches versus a perfect matched template calculated at 68°C (SantaLucia, 1998).

Feature and Equation	Min Max Mean (SD)	Filter (F) or Weight (W)
Total length of oligo (N_{total})	23 60* 36 ± 5	F: Must be at or greater F: Must be at or lower W: +1 if w/in -1SD and +1.5SD
Length of the side 5' Side 3' Side	10‡ 39 16 ± 4 17 ± 3	F: Must be at or greater F: Must be at or lower W: +0.5 w/in 1SD W: +0.5 w/in 1SD
Difference in length between 5' and 3' sides	0 ± 4#	W: +2 w/in 1SD
% total GC content $(N_G + N_C)/N_{total} * 100$	30 75 51 ± 7	F: Must be at or greater F: Must be at or lower W: +1 if w/in -1SD and +1.5SD
% GC of the side 5' Side 3' Side	21 100 57 ± 13 56 ± 12	F: Must be at or greater F: Must be at or lower W: +0.5 w/in 1SD W: +0.5 w/in 1SD
Total sequence T_m (°C) 81.5+0.41*%GC-(675/N_{total})-%Mismatch	63 82 Mode: 78 ± 3	F: Must be at or greater F: Must be at or lower W: +1 if w/in 1SD
% total GC content lower bound (**Fig. S2**) (185.7 * e^(-0.06797 * N_{total}) + 21.65	-	F: Must be at or greater
% total GC content upper bound (**Fig. S2**) (205.1 * e^(-0.05125 * N_{total}) + 28.64	-	F: Must be at or lower
% side GC content lower bound (148.9 * e^(-0.1929 * N_{total}) + 18.11	-	F: Must be at or greater
% side GC content upper bound (193.4 * e^(-0.03400 * N_{total}) - 24.26	-	F: Must be at or lower
Phusion T_m (°C) Breslauer Equation (Breslauer, et al., 1986) -10.8 cal/Mol helix initiation correction Nicking primer equation of 0.0003524 uM †Monovalent cation concentration 0.28 M	 60^ 78 Median: 68 ± 4	 F: Must be at or greater - W: +1 if w/in 1SD
Last base in the primer is G or C	-	W: +0.5 if G or C

*59 bases was observed but 60 is set as the theoretical.
‡6 bases was observed in rare high GC cases but 10 is set for safety.
^50°C observed however for safety the lower filter is set at 60 as the Nicking reaction anneals at 55°C.
†Value calculated from the NEB website from a test set of primer sequences.
#Actual value is -1 ± 4.6 bases but set at zero as centering the degenerate codon is important for stability.
The T_m values are measured at the mode or median as they are left skewed.

References:

Altschul, S.F., *et al.* PSI-BLAST pseudocounts and the minimum description length principle. *Nucleic Acids Research* 2009;37(3):815-824.

Bloom, J.D. Software for the analysis and visualization of deep mutational scanning data. *BMC Bioinformatics* 2015;16(1):168.

Breslauer, K.J., *et al.* Predicting DNA duplex stability from the base sequence. *Proceedings of the National Academy of Sciences* 1986;83(11):3746-3750.

Dunn, S.D., Wahl, L.M. and Gloor, G.B. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics* 2008;24(3):333-340.

Edgar, R.C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* 2004;32(5):1792-1797.

Firnberg, E., *et al.* A Comprehensive, High-Resolution Map of a Gene's Fitness Landscape. *Molecular Biology and Evolution* 2014;31(6):1581-1592.

Fowler, D.M., *et al.* Enrich: software for analysis of protein function by enrichment and depletion of variants. *Bioinformatics* 2011;27(24):3430-3431.

Goldenzweig, A., *et al.* Automated Structure- and Sequence-Based Design of Proteins for High Bacterial Expression and Stability. *Molecular Cell* 2016;63(2):337-346.

Hopf, T.A., *et al.* Mutation effects predicted from sequence co-variation. *Nature Biotechnology* 2017;35:128.

Kabsch, W. and Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 1983;22(12):2577-2637.

Klesmith, J.R., *et al.* Comprehensive Sequence-Flux Mapping of a Levoglucosan Utilization Pathway in E. coli. *ACS Synthetic Biology* 2015;4(11):1235-1243.

Klesmith, J.R., *et al.* Trade-offs between enzyme fitness and solubility illuminated by deep mutational scanning. *Proceedings of the National Academy of Sciences* 2017;114(9):2265-2270.

Kowalsky, C.A., *et al.* Rapid Fine Conformational Epitope Mapping Using Comprehensive Mutagenesis and Deep Sequencing. *Journal of Biological Chemistry* 2015.

Kowalsky, C.A., *et al.* High-Resolution Sequence-Function Mapping of Full-Length Proteins. *PLOS ONE* 2015;10(3):e0118193.

Li, W. and Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 2006;22(13):1658-1659.

Magoč, T. and Salzberg, S.L. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics* 2011;27(21):2957-2963.

Masella, A.P., *et al.* PANDAseq: paired-end assembler for illumina sequences. *BMC Bioinformatics* 2012;13(1):31.

Roederer, M. How many events is enough? Are you positive? *Cytometry Part A* 2008;73A(5):384-385.

Rubin, A.F., *et al.* A statistical framework for analyzing deep mutational scanning data. *Genome Biology* 2017;18(1):150.

SantaLucia, J. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences* 1998;95(4):1460-1465.

Tien, M.Z., *et al.* Maximum Allowed Solvent Accessibilites of Residues in Proteins. *PLOS ONE* 2013;8(11):e80635.
Whitehead, T.A., *et al.* Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nature biotechnology* 2012;30(6):543-548.
Wrenbeck, E.E., *et al.* Plasmid-based one-pot saturation mutagenesis. *Nat Meth* 2016;13(11):928-930.