**D-1-b-v We will use a unified weighted scoring scheme for combining all eleVAR features to prioritize variants**

To integrate the various features mentioned above, we plan to elaborate the weighting system in FunSeq.4. Constrained by selective pressure, common variations tend to arise in functionally unimportant regions. Thus, features that are enriched with common polymorphisms are less likely to contribute to the deleteriousness of variants and are weighted less. In general, features can be classified into two classes: discrete (e.g., within or outside of a given functional annotation) and continuous (e.g., the PWM change in ‘motif-breaking’). We will weight these two sets of features with different strategies.

For each discrete feature , such as sentitive region overlap, ultraconserve region overlap, and HOT region overlap, we calculate the probability that it overlaps with common polymorphisms. We then calculate the information content to denote the value of discret features + , where and can be used for score optimization.

The situation is more complex for continuous features, as different feature values have different probabilities of being observed in natural polymorphisms. Thus, one weight cannot suffice for varied feature values. For a continuous feature , such as motif gain, motif break and GERP etc, which is associated with a value , the probability is firstly estimated using common variants: . The score of continuous feature is defined as . We then fit a smoothing curve and estimate parameters ’s according to empirical distribution .

The eleVAR score (eS) is calculated by summing up the values of all its features. We will also consider the feature dependency structure when calculating the scores (e.g., removing redundant features or performing dimension reduction techniques).

**D-2 Approach Aim 2: Implement an efficient eleVAR pipeline & develop a workflow for tuning model parameters & assessing performance**

**D-2-a Overall workflow for the project in relation to Aim 2**

As shown in the timeline (Fig 6), we will take our features and scoring scheme (from Aim 1) and construct a practical software pipeline that can be applied on many genomic variants in a high-throughput fashion. We will then collect genomic variants from the existing cancer genomics data. We will run the pipeline on these variants to prioritize many of them. We will then compare the prioritization of the variants to publicly accessible validated variants and elements to readjust the parameters in our prioritization scheme. Finally, we will compare the newly-prioritized variants after this first round with the results of our high-throughput experimental characterization. Finally, we will perform an unbiased testing and pick a number of variants for in-depth validation.

**D-2-b Research plan for Aim 2**

**D-2-b-i Statistical framework for parameter tuning using Bayesian updates**

The initial feature parameter () (given number of features) assigned in D-1-b-v will be further optimized with newly available “gold standard” datasets. We plan to tune these parameters using an incremental Bayesian learning strategy. For a variant , given eleVAR score (equation 3 in D-1-b-v), the probability that is functional ( designates a positive result, whereas denotes a negative result) follows a logistic function ( are scaling parameters). To update with training data , we implement Bayes’ rule:.

The likelihood ratio is defined as: , and then MCMC (Monte Carlo Markov Chain) will be used to find the most probable . The updated will then be used as tuned parameters in eleVAR to prioritize variants. The procedure will be iterated in several rounds. In the first round of tuning, feature weights obtained in D-1-b-v will be used to construct priors . In subsequent rounds, the updated weights will be set as new priors.