Supplementary Material

Section S1 CI and LOR features

For each protein for which we want to predict the outcome of the mutations (the query protein) we obtained the MSAs using hhBlits with 1 iteration and E-value = 10^{-4}.

In order to compute the Conservation Index (CI) score, originally proposed in [2], we filtered the MSAs discarding the sequences with less than 10% coverage and less than 10% sequence identity (SI) with the target sequence. The CI equation is the following (see also [2, 8]): A is the set of the possible 20 amino-acids and i is the position of the MSA in which the mutation occurs. $f_a(i)$ is thus the frequency of occurrence of amino-acid a at position i while f_a is the frequency of a as observed in the entire alignment:

$$CI(i) = \sqrt{\sum_{a \in A} (f_a(i) - f_a)^2} \tag{1}$$

The LOR score is the log-odd ratio of observing the wild-type amino acid w with respect to the mutated amino acid m at the target position. To compute the LOR score, we filtered out from the MSAs the sequences with less than 30% coverage and 30% SI with respect to the target protein.

$$LOR = \log\left(\frac{f_w(i)}{1 - f_w(i)}\right) - \log\left(\frac{f_m(i)}{1 - f_m(i)}\right) \tag{2}$$

As explained in our previous work [8], $f_w(i)$ is the frequency of occurrence of wildtype (w) amino-acid at position i, while $f_m(i)$ is the frequency of the mutated (m) amino-acid in the same column of the MSA.

Section S2 Domain log-odd score

The PF feature, indicating the *sensitivity* of the PFAM domains to deleterious variants is computed in the following way:

$$PF_d = \log(\frac{N_{\text{del}}(d) + 1}{N_{\text{del}}(d) + N_{\text{neut}}(d) + 2}) - \log(\frac{N_{\text{neut}}(d) + 1}{N_{\text{del}}(d) + N_{\text{neut}}(d) + 2}) \tag{3}$$

where $N_{\text{del}}(d)$ and $N_{\text{neut}}(d)$ indicate, respectively, the number of deleterious and neutral variants mapped on the domain d. The +1 and +2 terms are the pseudo-counts added to avoid mathematical errors with the logarithm computation.

Section S3 Opening the random forest black-box: analysis of the decision profiles with treeinterpreter library

For the decomposition and the interpretation of the trained Random Forest (RF) model, we relied on the treeinterpreter python library (https://github.com/andosa/treeinterpreter)

by Ando Saabas (unpublished work). The method is described here:
http://blog.datadive.net/interpreting-random-forests/ but we briefly summarize the mathematical aspects.

Given a target feature vector x, `treeinterpreter` library visits each tree t in the trained forest T and analyzes which clauses are *activated* following the path of decisions leading from the root of t to the final leaf (corresponding to the prediction $t(x)$). While traversing t, the algorithm records whether the features guarding the splitting nodes *pushed* the final prediction towards the "1" or "0" class. This method produces two major advances with respect to the classical "feature relevance" computed during the training of the RF models. The first one is that the feature relevance is computed over the entire dataset while `treeinterpreter` acts on each feature vector x at a time, explaining *why* the model T made the particular decision $T(x) \in [0, 1]$. Second, if the single features contributions are averaged over the entire dataset instead of just focusing on x, `treeinterpreter` tells, for each feature, not only its *absolute* relevance but also whether it is used to discriminate one class better than the other. The classical feature relevance focuses necessarily on the entire dataset and gives little insight on understanding how *individual* decisions are made.

More formally, for each prediction in a decision tree t there is a path going from the root to the final leaf, which corresponds to the final prediction of the tree. Every node in the tree correspond to a decision and is guarded by a clause over a feature. Each decision influences the final prediction $t(x)$ as it *changes* the set of the reachable leaves, altering the path from the root to the final leaf.

As explained in `treeinterpreter` documentation, a tree t with M leaves divides the feature space in M regions R_m ($1 \leq m \leq M$) and the final prediction of the tree t can be written as

$$t(x) = \sum_{m=1}^{M} c_m I(x, R_m)$$

where c_m is a value learned during training and $I(x, R_m) = 1$ iff $x \in R_m$. This formula considers only the leaves and thus ignores the path of decisions from the root to the leaf, that leads to the final leaf and the corresponding c_m. This path starts with the value associated with largest possible region (the entire dataset class balancement in the entire dataset, called *bias*) and each step towards the leaves adds or subtracts a specific c_k from the value *inherited* from the parent node: the result of this summation corresponds indeed to c_m. Since each decision in an internal node is associated to the feature guarding it, we can rewrite the previous formula as $t(x) = \mathit{bias} + \sum \mathit{feature\_contributions}$. More precisely,

$$t(x) = c_{all} + \sum_{k=1}^{K} \mathit{contrib}(x, k)$$

where c_{all} is the bias term (the value based on the balancement of the classes in the entire dataset), K is the number of features and $\mathit{contrib}(x, k)$ is the contribution of the k-th feature with respect to the prediction of the instance x. `treeinterpreter` gets these $\mathit{contrib}(x, k)$ scores for each $t \in T$ by inspecting each tree in the the `scikit-learn` [9] implementation of the RF classifier while it performs the prediction for x. It is important to notice that the value of $\mathit{contrib}(x, k)$ depends on the values assumed by all the features in x because each feature influences the entire path from the root to the final prediction in the decision tree.

The final step in the RF algorithm is to compute the final predictions $T(x)$ of the entire forest by averaging the predictions $t(x)$ of the single trees:

$$T(x) = \frac{1}{|T|}\sum_{j=1}^{|T|} t_j(x)$$

where $|T|$ is the number of decision trees in the forest T. Expanding this summation with the `treeinterpreter` formulation of the RF problem, we obtain the final formula as:

$$T(x) = \frac{1}{|T|}\sum_{j=1}^{|T|} c_{all} + \sum_{k=1}^{K}\left(\frac{1}{|T|}\sum_{j=1}^{|T|} contrib_j(x,k)\right).$$

showing how the entire RF can be thought as the summation of feature contributions. `treeinterpreter` provides scores for the contributions of the features in x by analyzing the forest and averaging the observed contributions over the trees. The full explanation is available at http://blog.datadive.net/interpreting-random-forests/ .

Section S4 Supplementary Tables

Table S1: Table showing the incremental contributions of the features used in DEOGEN2. We added the features conceptually following the *biological scales* represented by each feature, from the molecular aspects (PROV, CI, LOR, EF), to the domain sensitivity to deleterious variants (PF), the interaction between proteins (INT), the relevance of the gene for the human health (RVIS, GDI, REC, ESS) and the pathway level (PATH), that is the broader biological aspect considered in our model.

Features	**Sen**	**Spe**	**Bac**	**Pre**	**MCC**	**AUC**
PROV	78.3	79.6	78.9	73.3	57.3	85.1
+CI	78.1	80.5	79.3	74.1	58.1	86.7
+LOR	76.6	83.2	79.9	76.5	59.6	87.8
+EF	76.5	83.6	80.0	76.8	59.9	88.0
+PF	80.6	88.2	84.4	83.0	69.2	92.4
+INT	81.0	88.4	84.7	83.3	69.8	92.7
+RVIS	81.1	89.5	85.3	84.7	71.2	93.0
+GDI	81.0	89.5	85.3	84.8	71.2	93.1
+REC	82.2	90.1	86.2	85.7	72.9	94.0
+ESS	83.5	90.3	86.9	86.2	74.3	94.5
+PATH	83.8	90.9	87.4	87.0	75.3	94.9

Section S5 Supplementary Figures

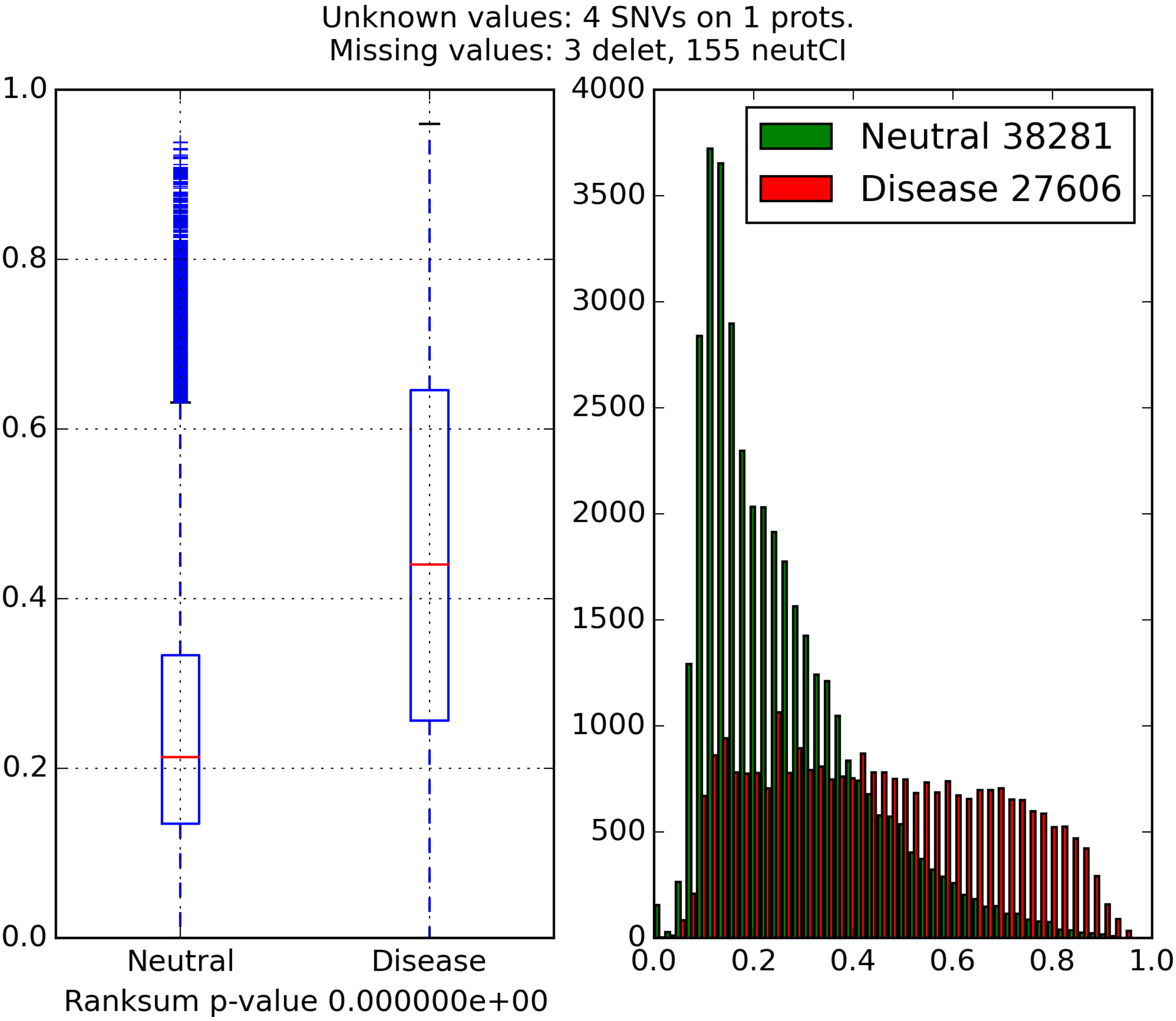

Figure S1: Plot showing the distribution of the Conservation Index (CI) computed from hhBlits alignments. Before computing the CI, the alignments have been filtered by removing sequences with less than 30% of coverage and less than 30% of sequence identity with the target sequence. The Wilcoxon's ranksums p-value between the deleterious and neutral classes is $< 10^{-300}$.

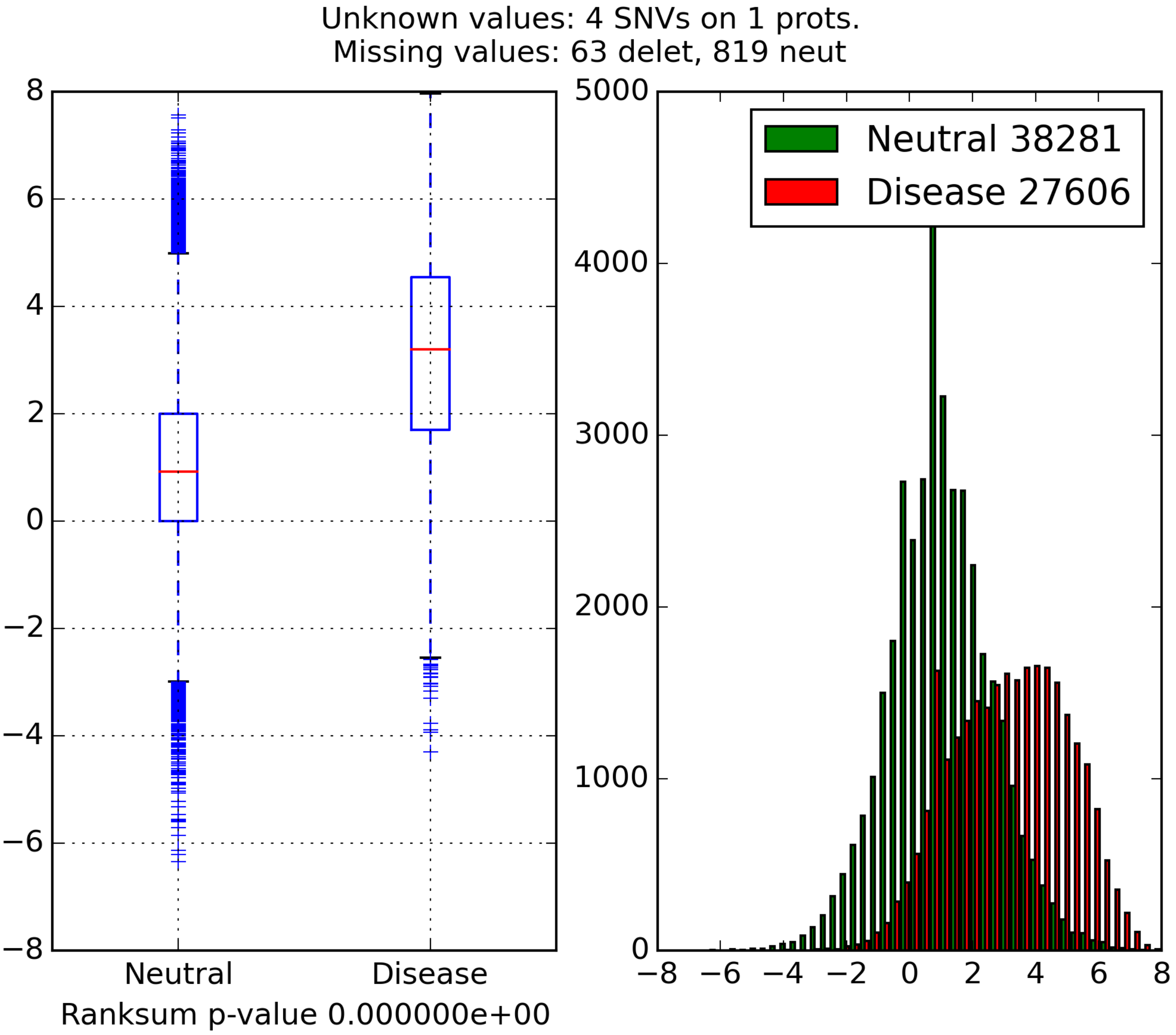

Figure S2: Plot showing the distribution of the Log-odd ratio (LOR) computed from hhBlits alignments. Before computing the CI, the alignments have been filtered by removing sequences with less than 10% of coverage and less than 10% of sequence identity with the target sequence. The Wilcoxon's ranksums p-value between the deleterious and neutral classes is $< 10^{-300}$.

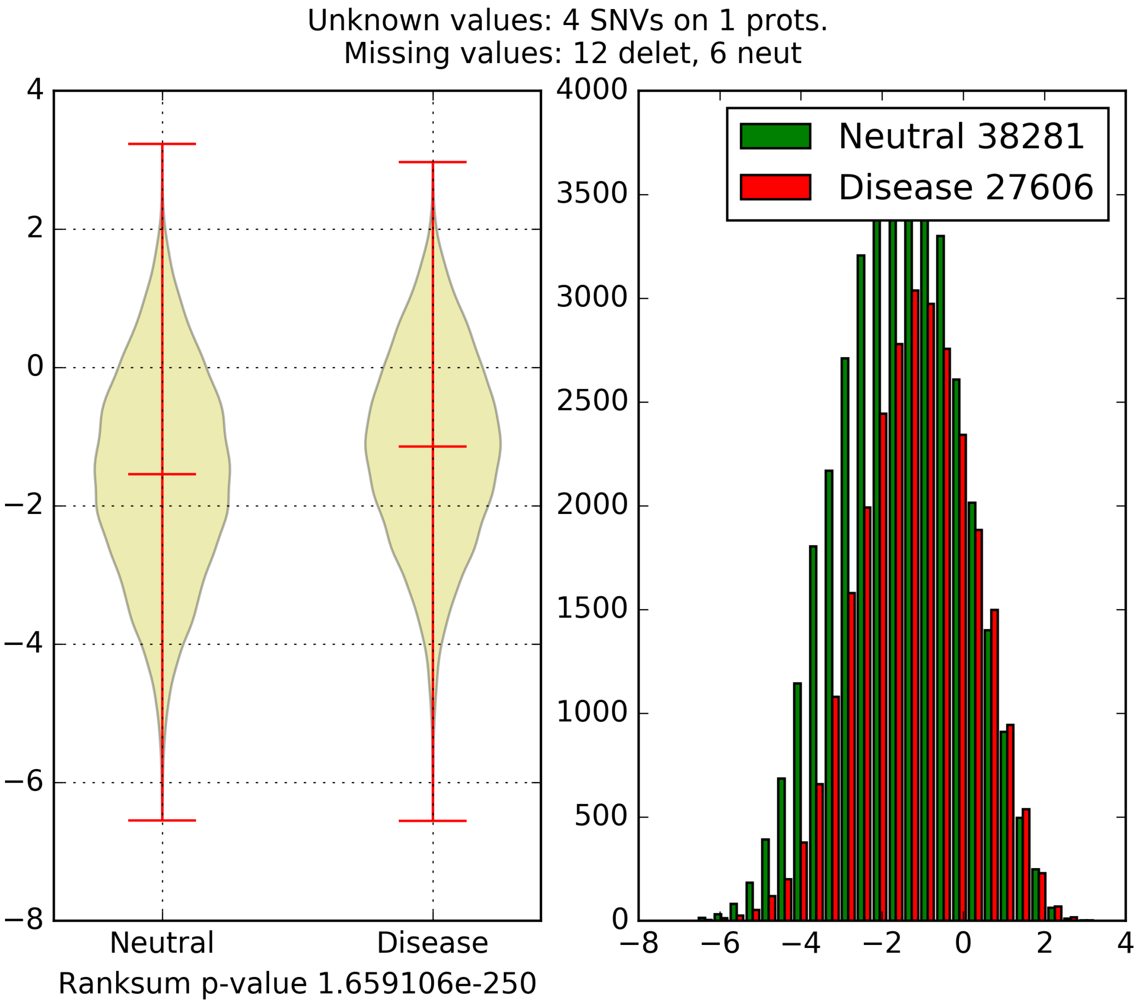

Figure S3: Plot showing the distribution of the early folding (EF) feature with respect to the neutral and deleterious classes of variants. On the left, the distributions are shown as violin plots, on the right as histogram. These EF scores have been obtained from an in-house SVM-based predictor of early folding residues, which are defined as the residues that are more crucial for the initiation of the protein folding process (paper currently under review). The predictor has been trained on data taken from the Start2Fold dataset [11] and uses DynaMine scores as inputs [10]. In this plot, positive scores indicate that the residue is likely to be involved in the early folding process, while negative scores indicate the opposite. Early folding residues are generally rare (5-10% of the residues in the protein sequence) and thus the scores are generally negative. From these plots it appears that deleterious variants are significantly enriched with EF residues (Wilcoxon's ranksums p-value $= 1.66 \times 10^{-250}$), explaining a possible structural reason for their deleteriousness.

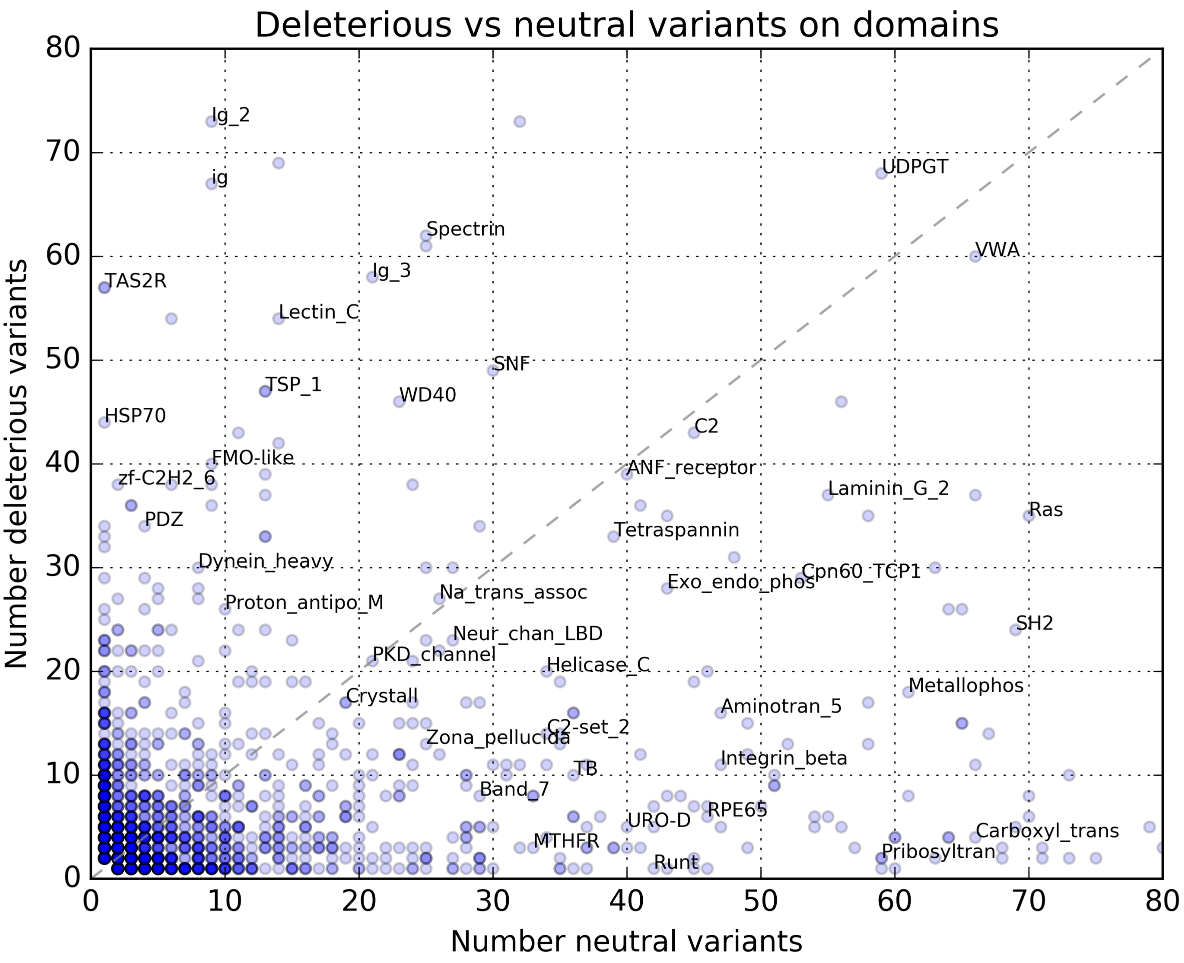

Figure S4: Scatter plot showing the distribution of the ratio between deleterious and neutral variants mapped on PFAM domains. The grey dotted line corresponds to the 1:1 ratio.

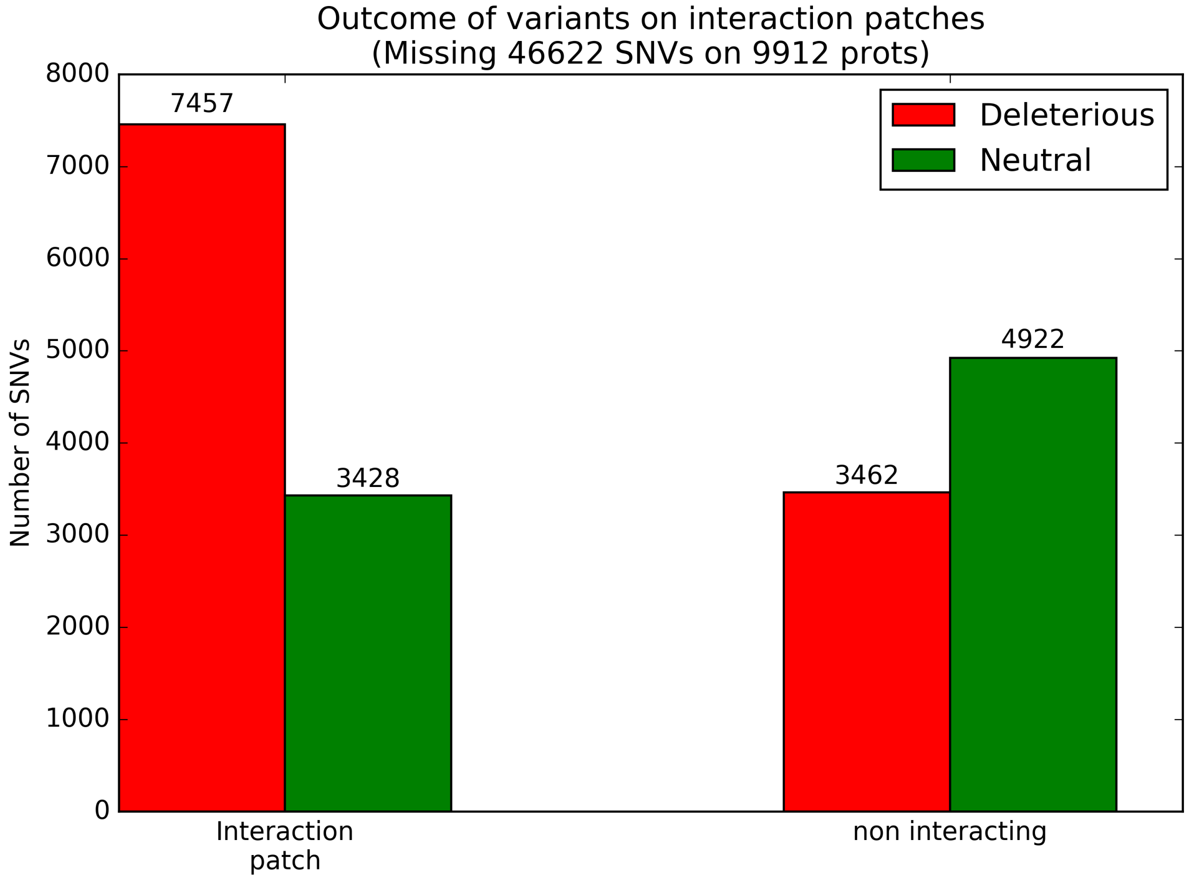

Figure S5: Plot showing how deleterious (red) and neutral (green) variants are mapped on the protein-protein interaction patches retrieved from INstruct database. As expected from literature, interaction patches appear to be enriched with deleterious variants (bars on the left).

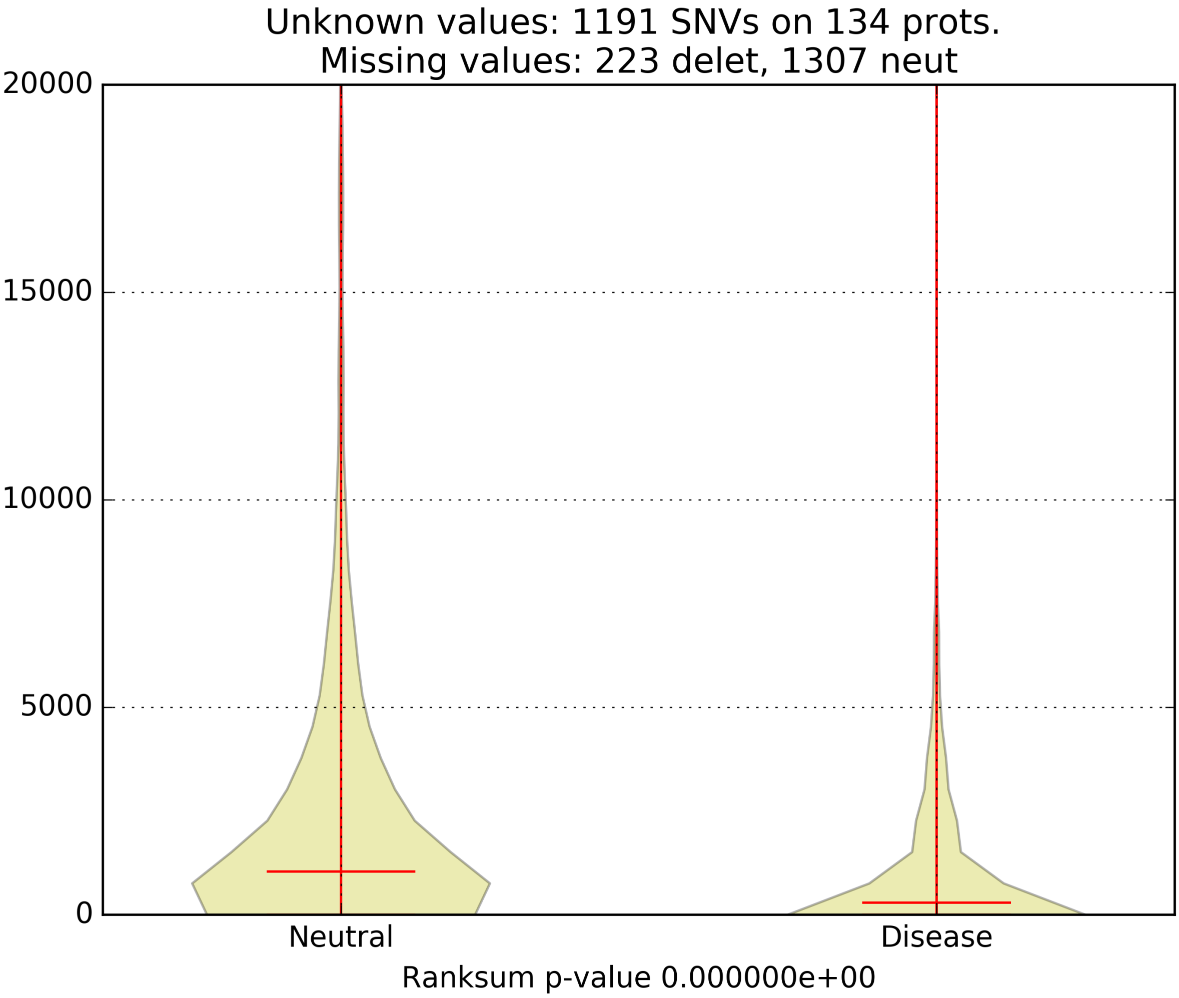

Figure S6: Violin plot showing the distributions of the GDI scores for the neutral and deleterious classes. Neutral variants tend to be mapped on genes with an higher mutational burden (and thus less affected by purifying selection). The Wilcoxon's ranksums test p-value is $< 10^{-300}$.

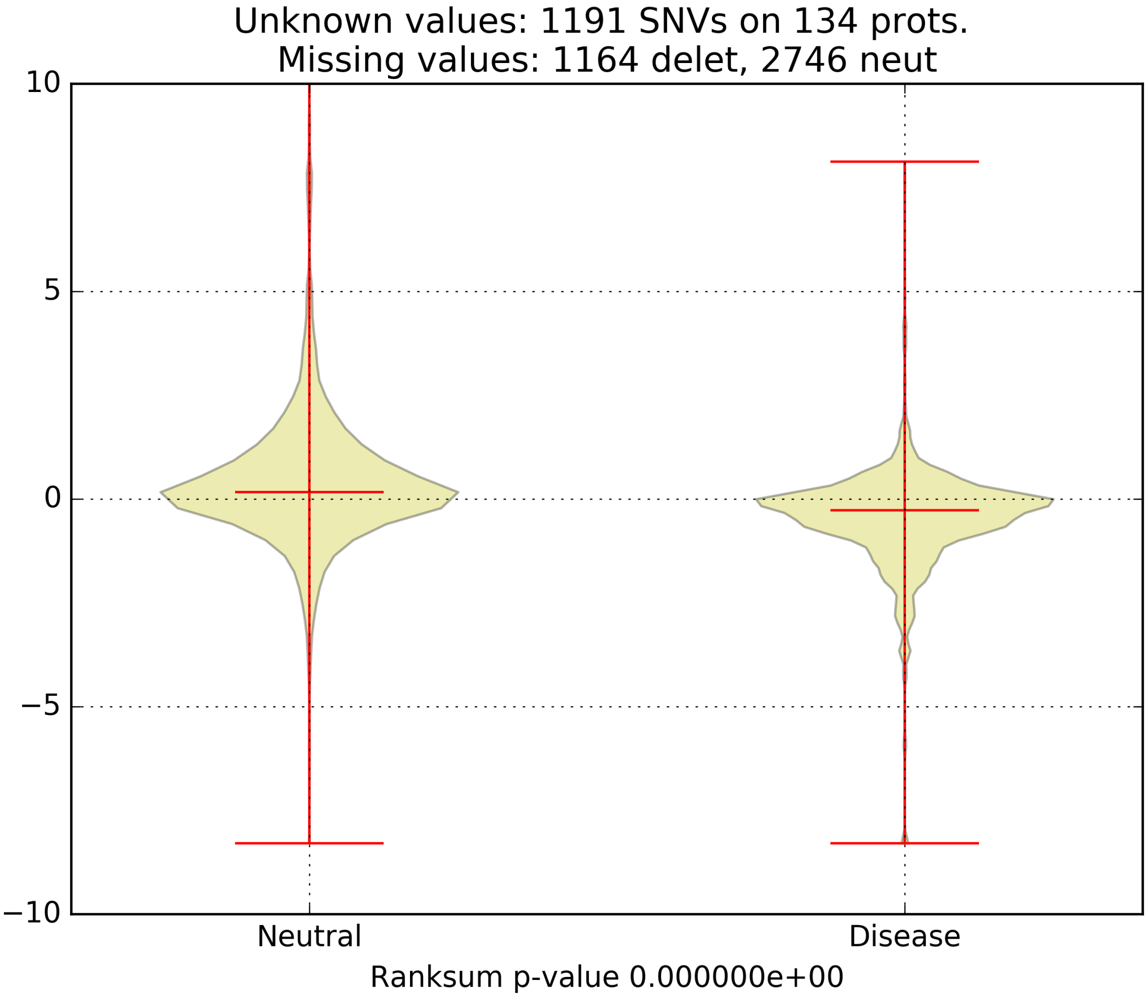

Figure S7: Violin plot showing the distributions of the RVIS scores for neutral and deleterious variants in Humsavar16. Neutral variants are significantly more likely to map on genes with higher variation in the general population (genes under less strong purifying selection). Wilcoxon's ranksums p-value $< 10^{-300}$.

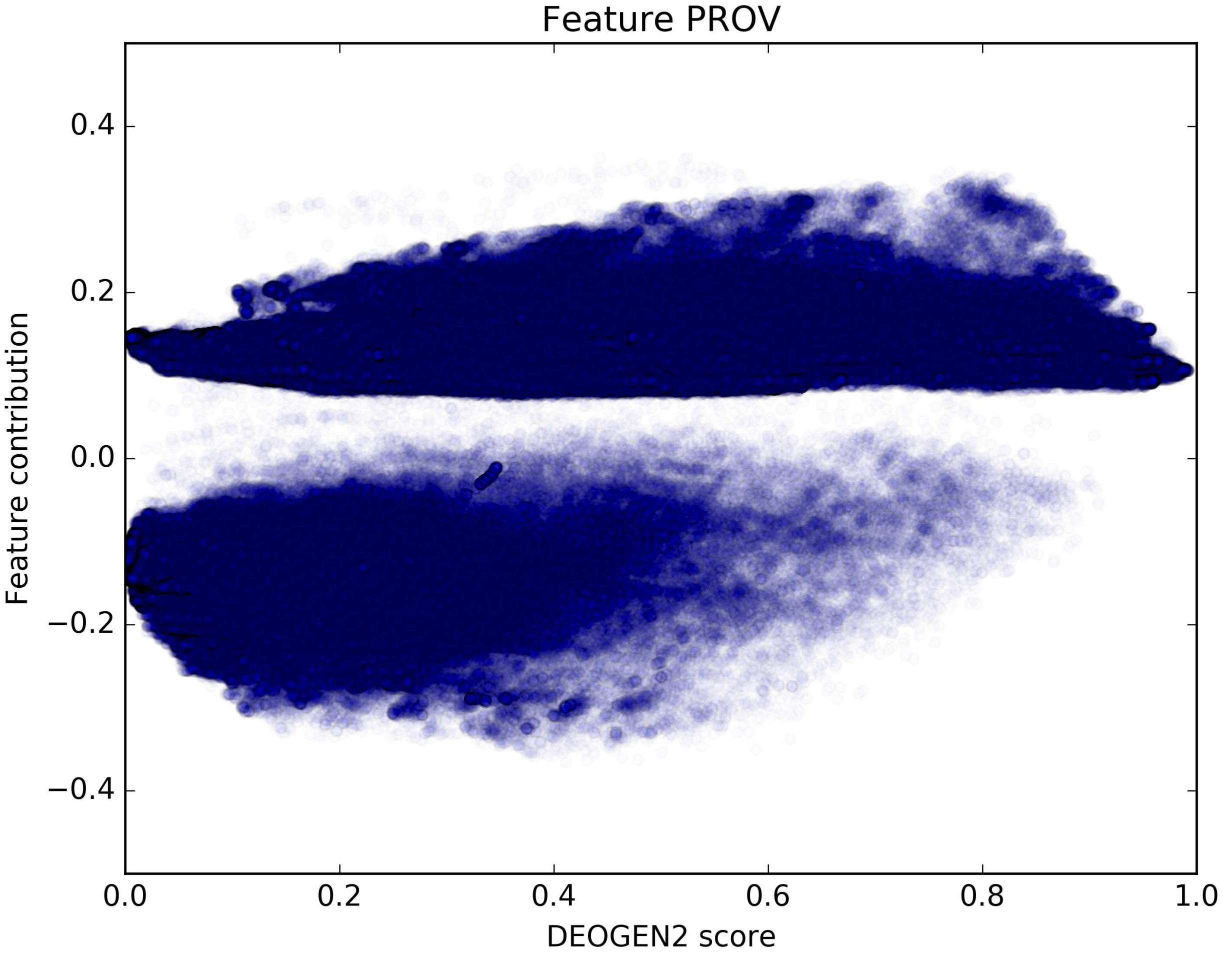

Figure S8: Scatter plot showing the contributions of the feature PROV (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

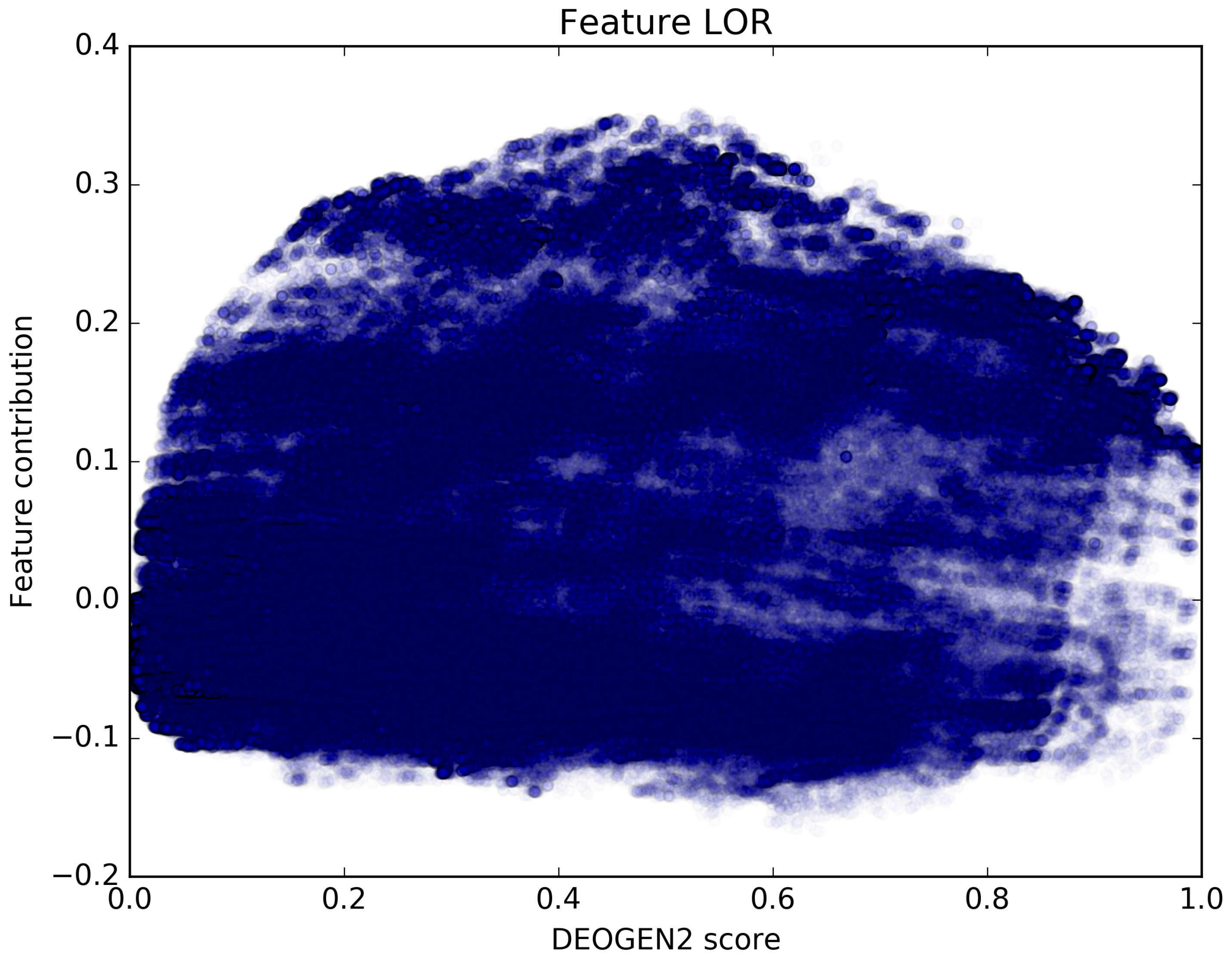

Figure S9: Scatter plot showing the contributions of the feature LOR (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

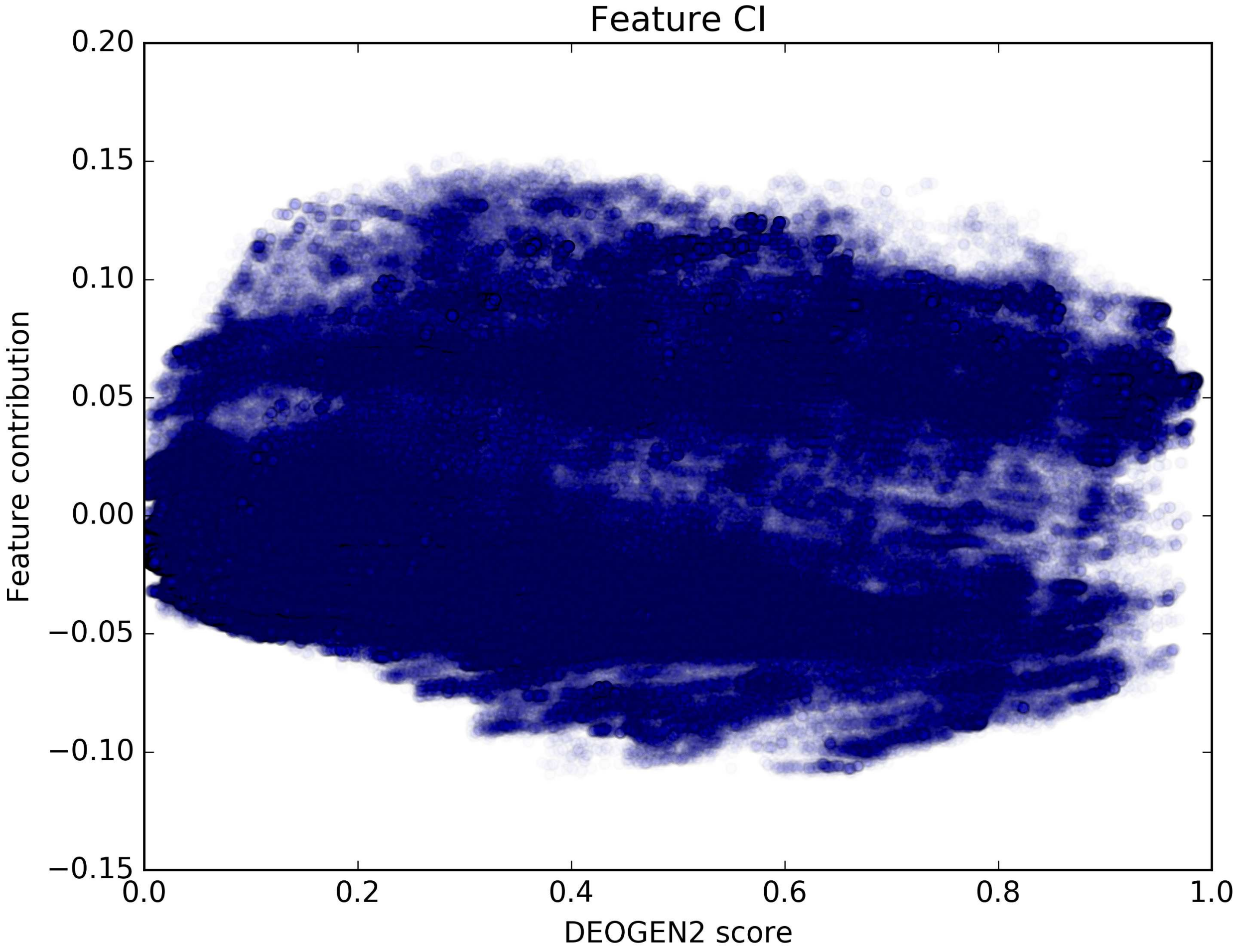

Figure S10: Scatter plot showing the contributions of the feature CI (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

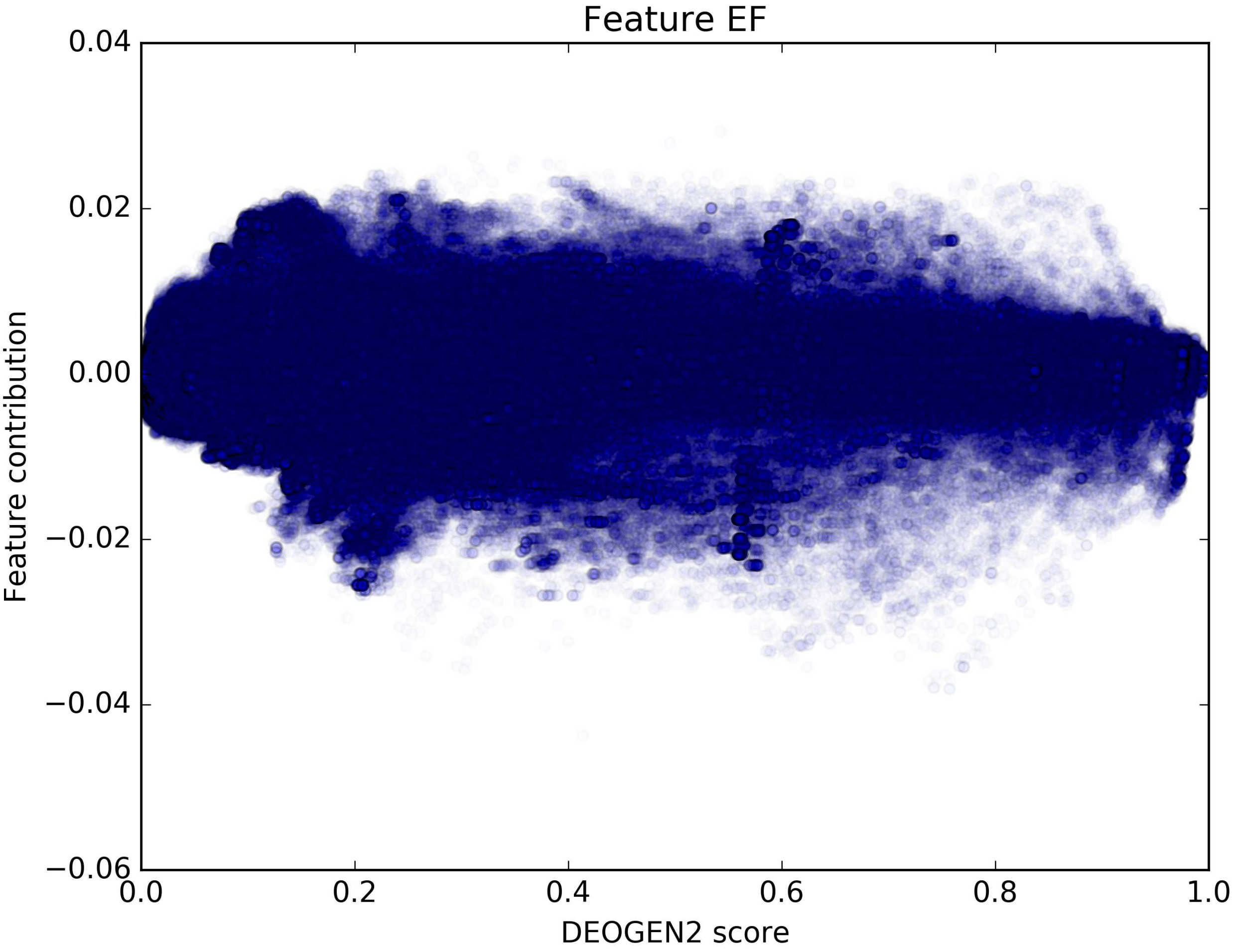

Figure S11: Scatter plot showing the contributions of the feature EF (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

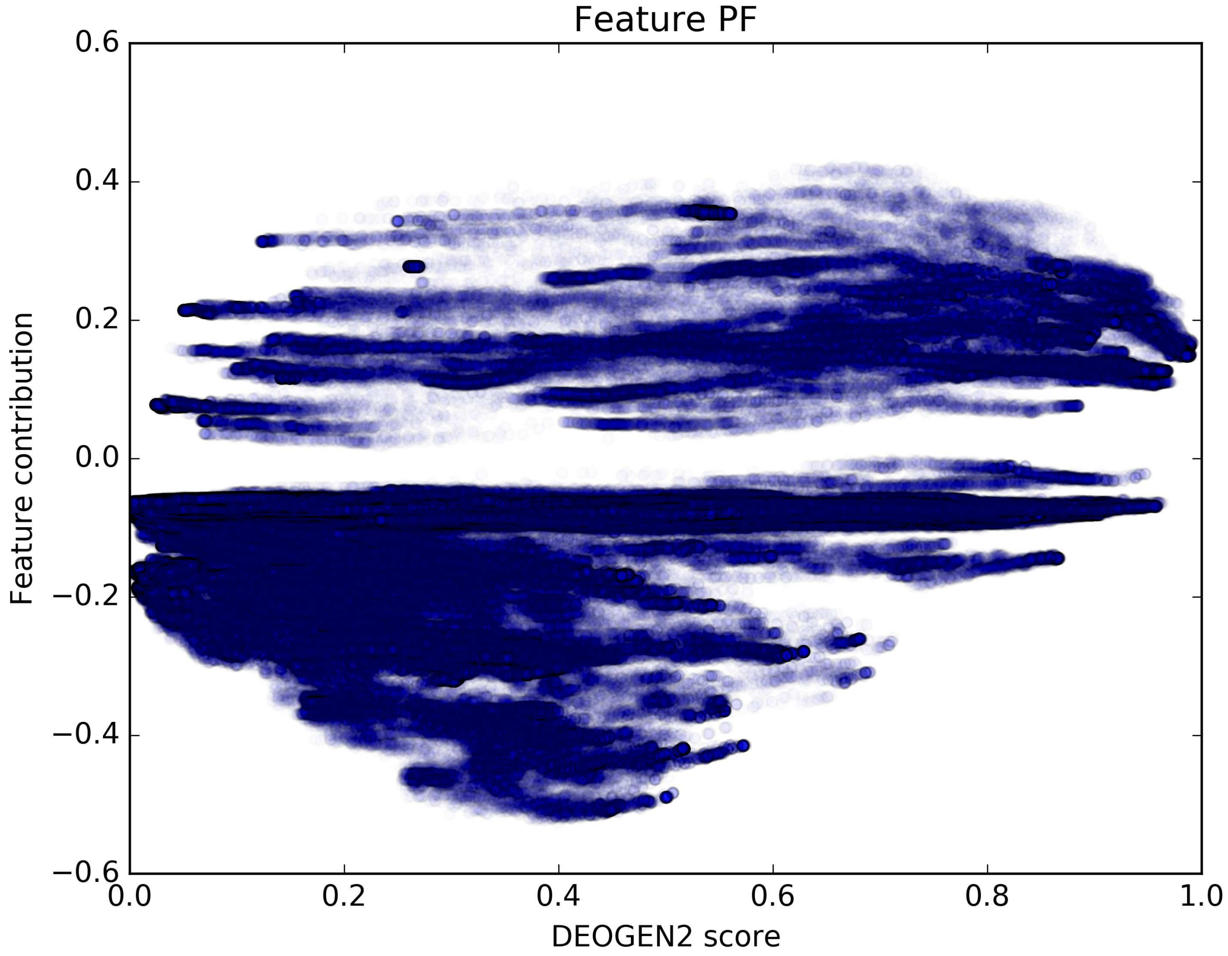

Figure S12: Scatter plot showing the contributions of the feature PF (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

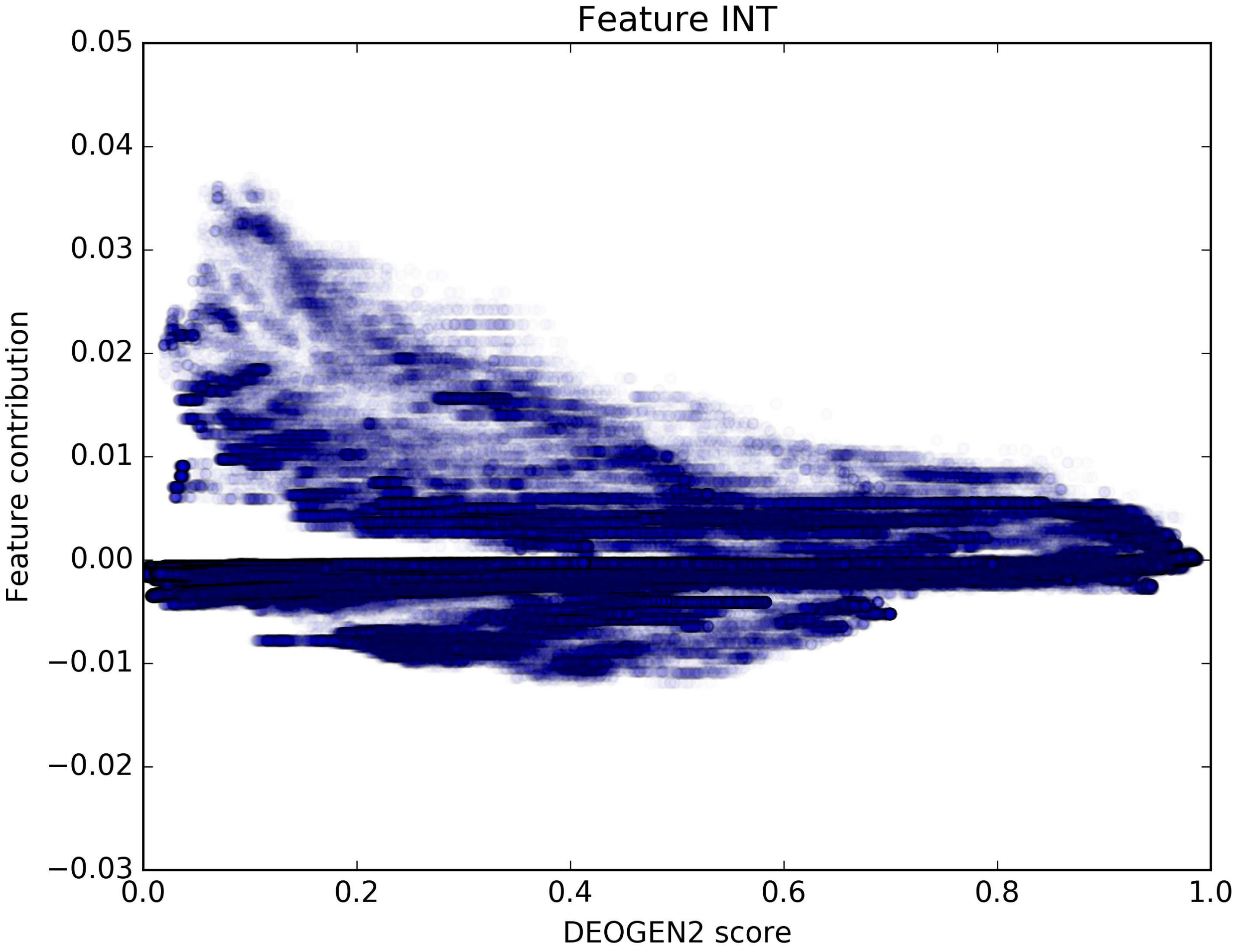

Figure S13: Scatter plot showing the contributions of the feature INT (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

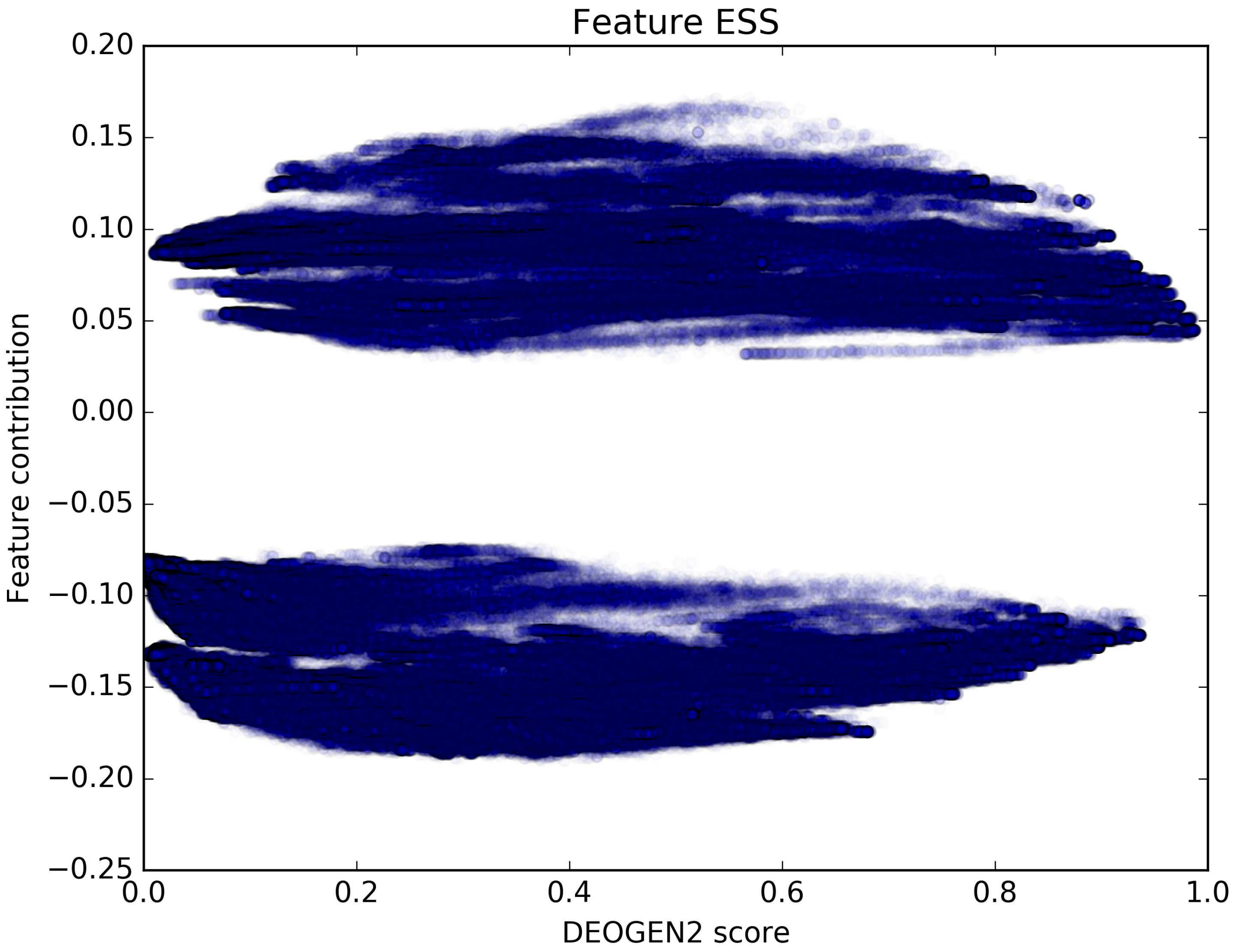

Figure S14: Scatter plot showing the contributions of the feature ESS (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

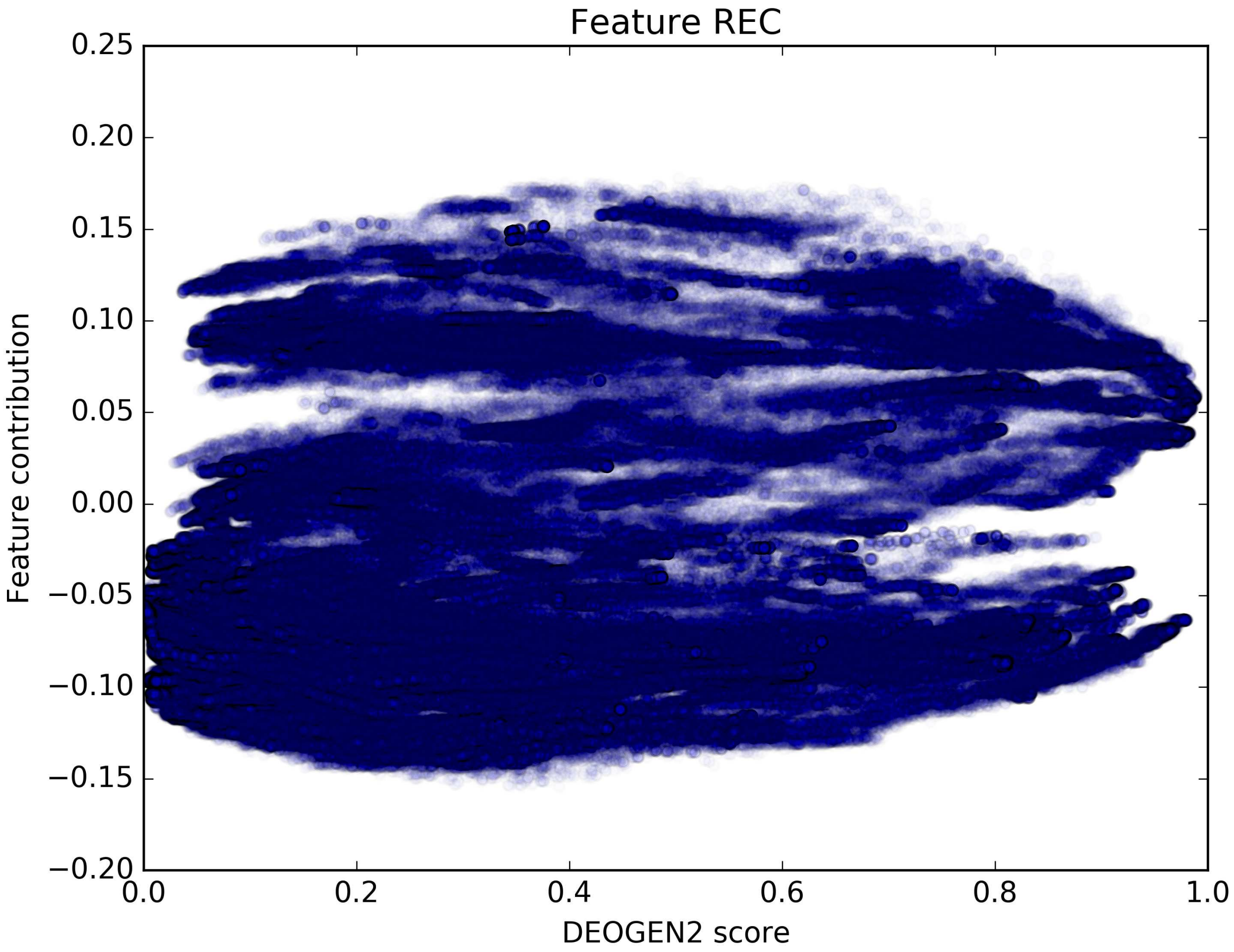

Figure S15: Scatter plot showing the contributions of the feature REC (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

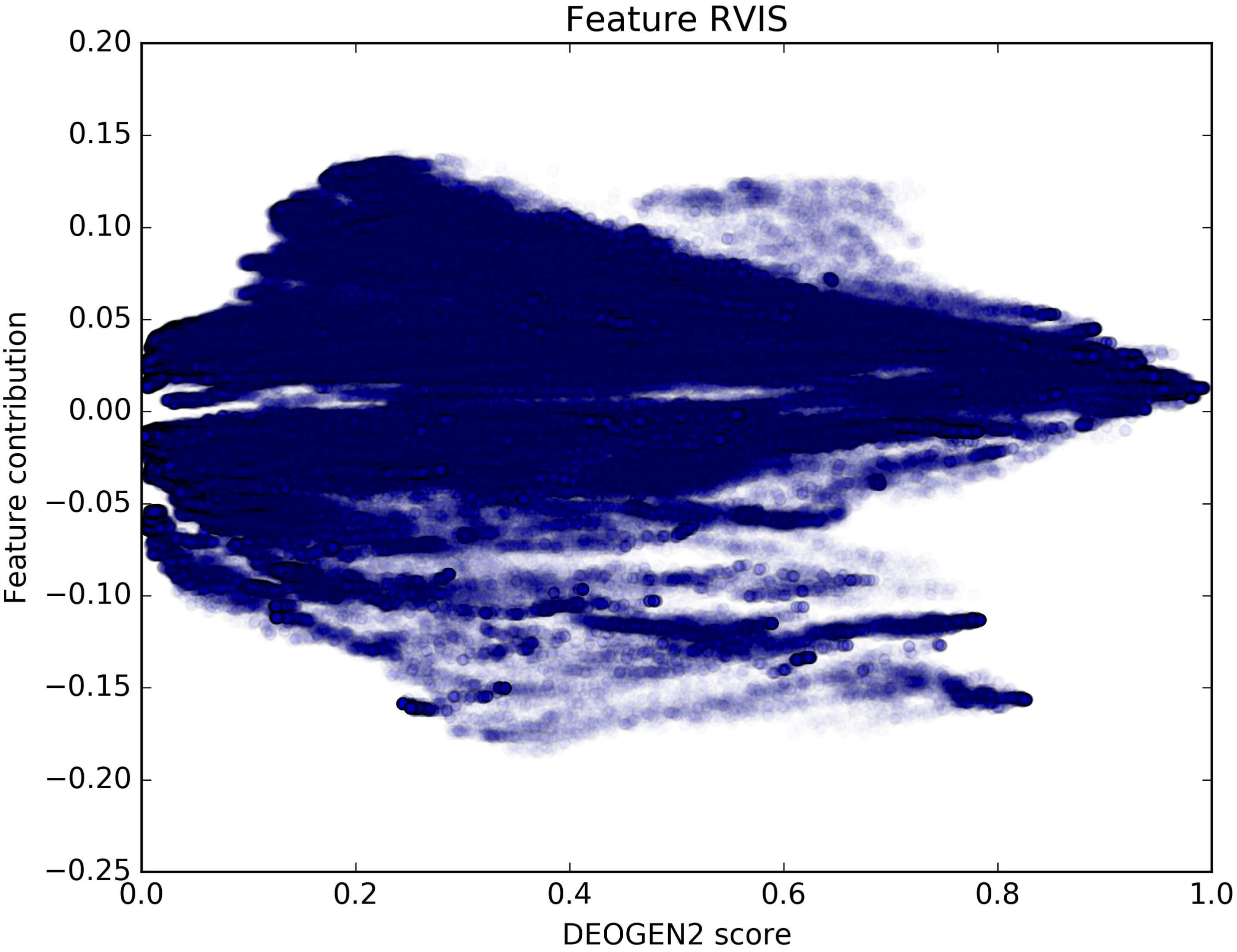

Figure S16: Scatter plot showing the contributions of the feature RVIS (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

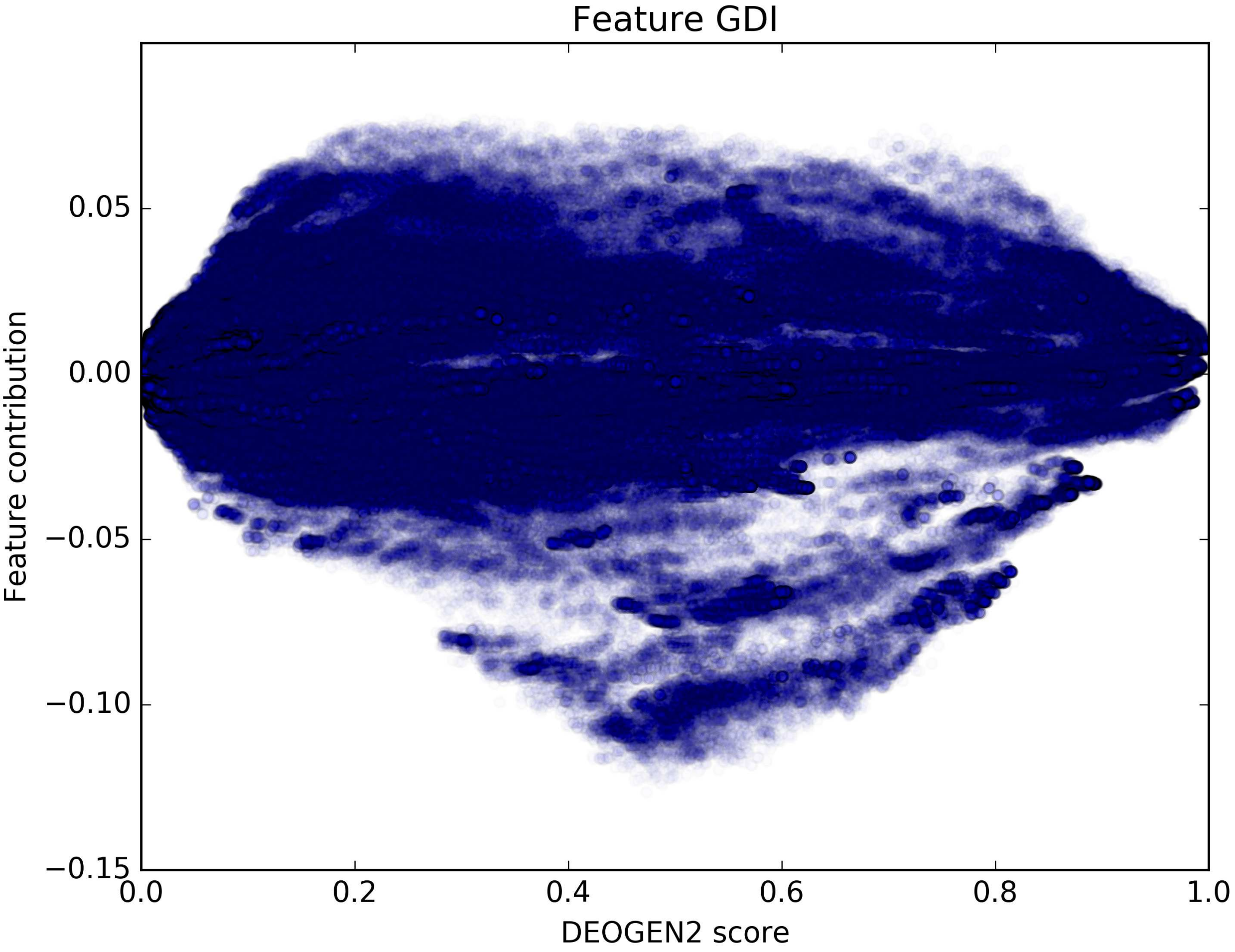

Figure S17: Scatter plot showing the contributions of the feature GDI (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

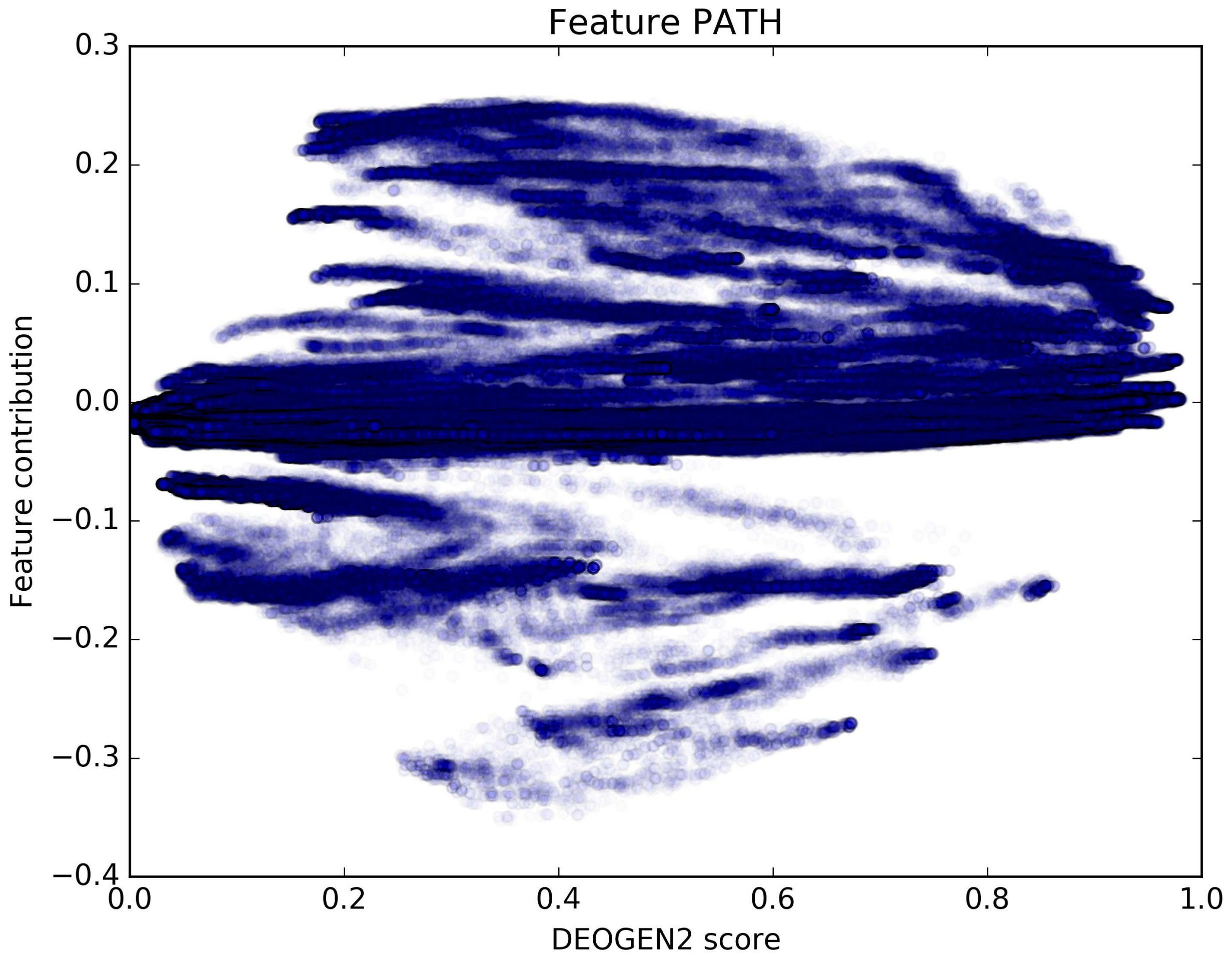

Figure S18: Scatter plot showing the contributions of the feature CI (on the y axis) with respect to the final DEOGEN2 prediction (on the x axis). This plot indicates whether there is a relationship between a specific feature's tendency to *push* towards the deleterious or neutral classes and the overall prediction obtained from DEOGEN2. Moreover, it shows the expected range of the contributions for each feature, in function of the final predictions from DEOGEN2.

References

[1] Remmert, M. et al. (2012). HHblits: lightning-fast iterative protein sequence searching byHMM-HMMalignment. Nat. Methods, 9, 173175

[2] Calabrese, R., Capriotti, E., Fariselli, P., Martelli, P. L. & Casadio, R. Functional annotations improve the predictive score of human disease-related mutations in proteins. *Hum. Mutat.* 30, 123744 (2009).

[3] Michael J. Meyer, Jishnu Das, Xiujuan Wang and Haiyuan Yu. INstruct: a database of high-quality 3D structurally resolved protein interactome networks. Bioinformatics, Vol. 29 no. 12 (2013), pages 1577-1579

[4] Georgi, B., Voight, B. F. & Buan, M. (2013) From mouse to human: evolutionary genomics analysis of human orthologs of essential genes. PLoS Genet. 9, e1003484.

[5] Daniel G. MacArthur *et al.* A Systematic Survey of Loss-of-Function Variants in Human Protein-Coding Genes Science 17 February 2012: 335 (6070), 823-828.

[6] Itan, Y., Shang, L., Boisson, B., Patin, E., Bolze, A., Moncada-Vlez, M., Casanova, J.-L. (2015). The human gene damage index as a gene-level approach to prioritizing exome variants. Proceedings of the National Academy of Sciences of the United States of America, 112(44), 1361520. https://doi.org/10.1073/pnas.1518646112

[7] Petrovski, S., Wang, Q., Heinzen, E. L., Allen, A. S., & Goldstein, D. B. (2013). Genic intolerance to functional variation and the interpretation of personal genomes. PLoS Genetics, 9(8), e1003709. https://doi.org/10.1371/journal.pgen.1003709

[8] Raimondi, D., Gazzo, A. M., Rooman, M. & Vranken, W. F. Multi-level biological characterization of exomic variants at the protein level significantly improves the identification of their deleterious effects. Bioinformatics, 18 (2016). doi:10.1093/bioinformatics/btw094

[9] Pedregosa et al. (2011) Scikit-learn: Machine Learning in Python. *JMLR* 12, pp. 2825-2830.

[10] Elisa Cilia, Rita Pancsa, Peter Tompa, Tom Lenaerts, and Wim Vranken From protein sequence to dynamics and disorder with DynaMine Nature Communications 4:2741 doi: 10.1038/ncomms3741 (2013)

[11] R. Pancsa and M. Varadi, P. Tompa, W. Vranken: Start2Fold: a database of hydrogen/deuterium exchange data on protein folding and stability Nucleic Acids Research, November 2015, Database issue