Genome analysis

Supplementary information for FATHMM-XF: accurate prediction of pathogenic point mutations via extended features

Mark F. Rogers [1,*], Hashem A. Shihab [2], Matthew Mort [3], David N. Cooper [3], Tom R. Gaunt [2] and Colin Campbell [1,*]

[1] Intelligent Systems Laboratory, University of Bristol, Bristol, BS8 1UB, UK.
[2] MRC Integrative Epidemiology Unit (IEU), University of Bristol, Bristol, BS8 2BN, UK.
[3] Institute of Medical Genetics, Cardiff University, Cardiff, CF14 4XN, UK.

* To whom correspondence should be addressed.

1 Leave-one-chromosome-out cross-validation (LOCO-CV)

We evaluated all models using LOCO-CV. For each fold we leave out one test chromosome and use the remaining chromosomes to train the model, applying the same model parameters for all folds. Except where noted, we trained models using balanced sets of 1,000 positive and 1,000 negative examples. These relatively small training subsets yield accuracy nearly as high as the final model but take less time to train, and this approach allowed us to estimate variability that arises from using randomised training sets. For testing we can use all available examples for the left-out chromosome, resulting in unbalanced test sets in most cases. For this reason we report balanced accuracy for all tests.

2 Labeled examples

We constructed our pathogenic (positive) dataset using somatic point mutations from the HGMD database (Stenson *et al.*, 2017) while we used SNVs from the 1,000 Genomes Project (The 1000 Genomes Project Consortium, 2012) as neutral examples. Bias may be introduced if benign mutations are located in genomic regions that differ substantially from regions containing pathogenic mutations, such as examples from different genes. To ensure the locations of putative benign mutations approximate those of pathogenic mutations, we select only those putative benign mutations found within a 1000-nt window of some pathogenic mutation. This issue has been evaluated more thoroughly in previous studies (Ritchie *et al.*, 2014; Kircher *et al.*, 2014; Shihab *et al.*, 2015). In addition, we restrict our data to the autosomes (Table 1), as allosomes (sex chromosomes X and Y) have been shown to evolve at a different rate than the autosomes (see, e.g., (Charlesworth *et al.*, 1987)) and hence may have different sequence conservation profiles. As a result, we are wary of models or predictions that do not consider this distinction.

We used the Variant Effect Predictor (McLaren *et al.*, 2016), to assess the distribution of examples in our training data and compared them with our ClinVar test sets (Supplementary Figures 1 and 2). In coding regions, we found that the majority of pathogenic examples are missense mutations, causing a change in the resulting amino acid, whilst the neutral examples are balanced between missense and synonymous mutations. In non-coding regions, a large proportion of pathogenic examples reside near splice sites, whilst neutral examples tend to be spread more evenly across introns. It is important to note that the overwhelming majority of human genes undergo alternative splicing, hence mutations tend to fall within multiple transcripts, hence receive multiple annotations. As a result, the proportions shown in these graphs do not sum to 1. In addition, many examples in coding regions may also function as non-coding (*Noncoding alternatives*), for example, when they fall within a cassette exon that is omitted from some transcripts.

3 Feature groups and kernels

Subsequent to our previous work on *FATHMM-MKL*, we obtained a variety of additional data sets from ENCODE (The ENCODE Project Consortium, 2012) and from NIH Roadmap Epigenomics (Bernstein *et al.*, 2010). These resources provided us with nearly 30 data sources that have proved informative in predicting haploinsufficient genes, for example (Shihab *et al.*, 2017b). Recently we have also derived new features from the Variant Effect Predictor (McLaren *et al.*, 2016), from annotated gene models, and from patterns in nucleotide sequence. We converted each of these feature groups into a set of kernels consisting of: two conservation-based kernels; a nucleotide spectrum kernel; two kernels designed to encapsulate the genomic context (gene region, amino acid changes, and proximity to gene features), and 27 kernels based on ENCODE data.

	Coding		Non-coding	
Chromosome	1000G	HGMD	1000G	HGMD
1	11,240	5,484	1,909	590
2	7,100	3,674	1,510	611
3	5,540	3,503	909	445
4	4,236	1,450	932	231
5	4,826	1,637	939	337
6	6,358	1,715	1,109	237
7	5,096	3,286	872	406
8	3,609	1,136	712	251
9	4,478	1,697	976	292
10	4,164	1,449	719	180
11	7,099	4,382	1,243	596
12	5,344	2,962	920	371
13	1,682	1,384	376	257
14	4,016	1,639	481	121
15	3,416	1,940	695	281
16	4,984	2,504	1,083	340
17	6,208	3,255	1,008	595
18	1,700	843	500	121
19	9,173	3,131	860	230
20	2,875	953	381	99
21	1,403	591	370	57
22	2,814	799	440	128
Total	107,361	49,414	18,944	6,776

Table 1. **Distributions of training examples by chromosome** shows the number of examples available for testing and training in LOCO cross-validation. These data are unbalanced, with up to 3.7 times as many neutral (1000G) examples as positive (HGMD) examples in coding regions and up to 6 times as many in non-coding regions. Note that for FATHMM-XF, only autosomal chromosomes were used in training and testing the method.

3.1 Conservation and ENCODE data

As in previous studies (Shihab *et al.*, 2015; Rogers *et al.*, 2015; Shihab *et al.*, 2017b) we evaluated distinct ENCODE datasets as potential feature groups for these classifiers. Broadly speaking, we divide these datasets into eight categories:

- *Genomic and Evolutionary*: where appropriate, we used a number of genomic properties such as gene length, number of transcripts and the average number of predicted protein domains across transcripts. In addition, we used a comprehensive set of conservation-based measures, such as dN/dS ratios between human and 65 different species (one-to-one orthologues). We also used several conservation based measures, e.g., PhyloP (Pollard *et al.*, 2010) and PhastCons (Siepel *et al.*, 2005) scores, derived from the multiple sequence alignment of 46 and 100 vertebrate genomes to the human genome (Blanchette *et al.*, 2004).
- *Histone Modifications*: we used ChIP-Seq peak calls for 14 histone modifications across 45 cell lines from ENCODE (The ENCODE Project Consortium, 2012) and narrow, broad and gapped regions of enrichment based on consolidated epigenomes from the NIH Roadmap project (Kundaje *et al.*, 2015).
- *Open Chromatin*: we used DNase-Seq and Formaldehyde-Assisted Isolation of Regulatory Elements (FAIRE) peak calls across 119 cell lines from ENCODE and narrow, broad and gapped regions of enrichment based on consolidated epigenomes from the NIH Roadmap project.
- *Transcription Factor Binding Sites*: based on PeakSeq and SPP peak calls for 119 transcription factors across 77 cell lines from ENCODE.
- *Gene Expression*: based on RNA-seq signal coverage using consolidated epigenomes from NIH Roadmap Epigenomics.
- *Methylation*: based on whole genome bisulphite sequencing (WGBS) from NIH Roadmap Epigenomics.
- *Digital Genomic Footprinting Sites*: for transcription factor recognition sequences within DNase-hypersensitive sites using consolidated epigenomes from the NIH Roadmap Epigenomics Project.
- *Networks*: we used measures of centrality from cell-type specific interactome and tissue-specific co-expression networks.

3.2 Sequence features

One goal for our models is to learn the sequence characteristics that are most susceptible to pathogenic mutations in both coding and non-coding regions. As a simple method for capturing the disruption that may occur in the sequence surrounding a mutation, we use *spectrum* kernels (Leslie *et al.*, 2002) to compare the composition of k-mers within a region before and after a mutation is applied to a sequence. Given a mutation and its flanking sequence, we obtain the k-spectra for the wild-type and mutated versions of the sequence and concatenate these features to provide a picture of the region before and after mutation. Formally, borrowing notation from (Leslie *et al.*, 2002), if the k-spectrum of an input sequence s is the set of all k-length contiguous subsequences, then we define a feature map of all possible subsequences a of length k from alphabet $\mathcal{A}$ as follows:

$$\Phi_k(s) = \left(\phi_a(s)\right)_{a\in\mathcal{A}^k} \tag{1}$$

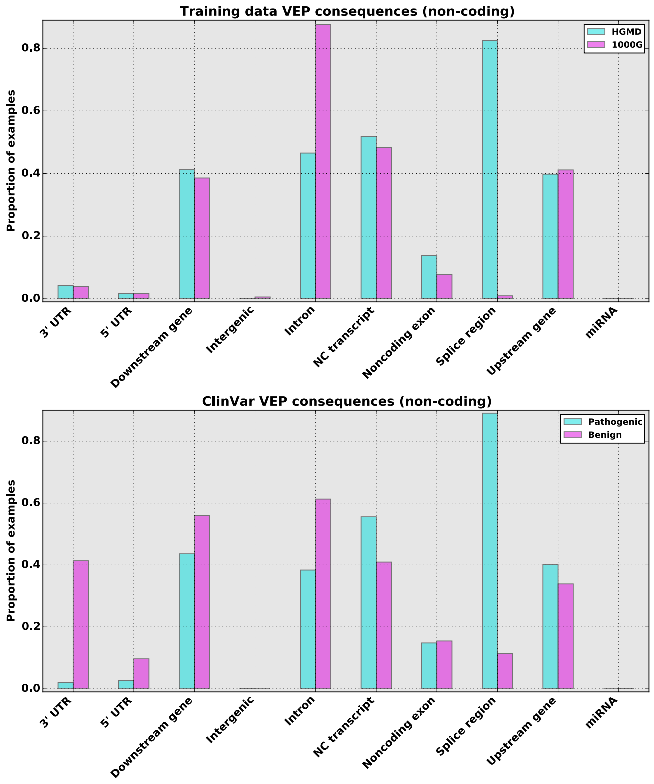

Fig. 1. Distributions of consequence annotations from the Variant Effect Predictor highlight differences between pathogenic and neutral examples in non-coding regions within the *FATHMM-XF* training data (top) and the ClinVar benchmark (bottom). In both data sets, splicing regions are by far the most prevalent group within positive (Pathogenic/HGMD) examples. In the *FATHMM-XF* training set, intronic variants are prominent amongst the neutral (1000G) data, whilst the ClinVar neutral (Benign) data also feature UTRs and locations downstream of genes.

where $\phi_a(s)$ is the count of the number of times sequence a occurs in sequence s, from which a kernel matrix can be readily derived (Campbell and Ying, 2011). We found that these features perform competitively on their own, and improve prediction accuracy in our merged-kernel tests. As this approach makes no assumptions about the kind of RNA binding proteins that may be impacted by a particular mutation, it may obviate the need to find and assess known motifs.

For these features we optimise two relevant parameters: the size of the window w flanking each mutation, and the maximum k-mer size, k. For a single-point mutation we expect the disruption to be confined to a relatively small region around the mutation. This restricts the useful window size and in turn, the maximum k-mer sizes that will be relevant. For both coding and non-coding models we performed a grid search over these sizes, for $k \in [1, \ldots, 5]$ and $w \in [1, \ldots, 10]$.

3.3 Genomic context features

We also include features that describe the genomic context where a mutation occurs. For coding regions we base these features on information from the ENSEMBL Variant Effect Predictor (VEP). The VEP provides characteristics for specific genomic locations that we can exploit to predict the likely impact of a SNV. These may include transcript features and amino acids impacted by a mutation, relative allele frequencies, and scores from pathogenic variant predictors such as SIFT and PolyPhen (Adzhubei *et al.*, 2010). To mitigate potential bias we are careful not to include these other scores, nor do we include features such as PubMed IDs that may have been used to curate SNV databases. Hence our features include only the following elements:

- *Consequence:* the VEP provides these as annotations of 35 types of changes to associated transcripts, such as *3' UTR variant*, *missense variant* or *TF binding site variant*. We represent these using 35-element binary vectors (one bit per annotation), plus a count of the number of transcripts possibly impacted. Note that we do not encode the impact levels provided on the VEP website (HIGH, MODERATE, MODIFIER, LOW); instead we allow our model to learn priorities in training.
- *Amino acid:* the amino acids inferred by the reference and allele nucleotides. To capture the change in amino acid composition, we construct two sets of features: two 20-element binary vectors that reflect the reference and allele residues, respectively, and two real-valued vectors that represent specific residue characteristics: molecular weight, hydrophobicity, occurrence frequency, dissociation constants for the $COOH$ and NH_3^+ groups, and the pH at the isoelectric point.

We apply the VEP features only to our coding predictor, as amino acid features are not relevant to SNVs in non-coding regions, and there are far more non-coding positions than coding positions: such a vast number of VEP queries would be impractical for creating a genome-wide database.

We concatenate two vectors to encapsulate all of the VEP features used in the model for coding regions. The *Consequences* features consist of a transcript count (the number of transcripts that may be impacted by a mutation) and a vector of binary flags that represent the possible consequences

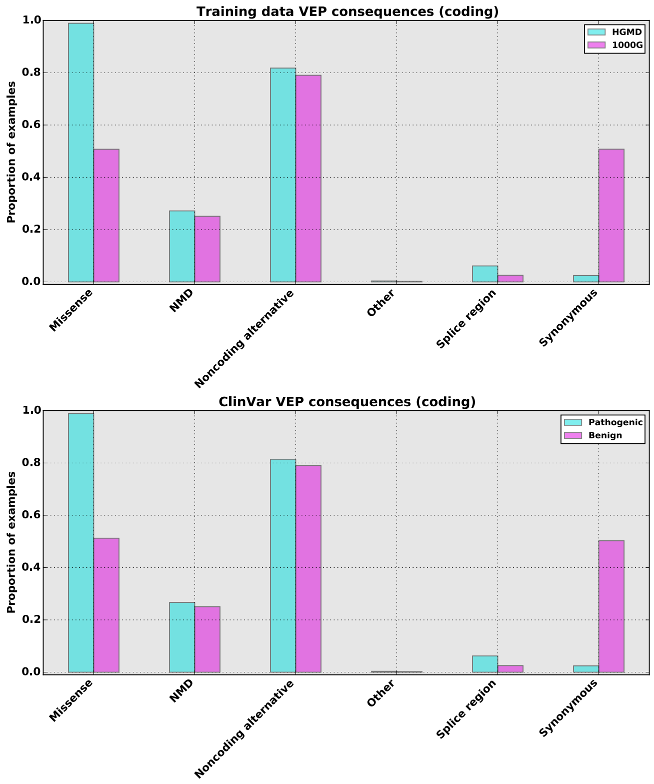

Fig. 2. The distributions of consequence annotations from the Variant Effect Predictor highlight differences between pathogenic and neutral examples in coding regions within the *FATHMM-XF* training data (top) and the ClinVar benchmark (bottom). Here we see that both datasets have nearly identical distributions of features.

returned by the VEP (Table 2). The *Amino acids* features consist of two vectors that encapsulate the possible amino acids associated with each mutation. The first amino acid vector represents the amino acids associated with the wild-type sequence, and the second represents the amino acids associated with the mutation. Instead of using a simple binary flag, we use counts to record the number of transcripts impacted by each amino acid change. The accuracy of this model is recorded as *Distance* in Table 4.

transcript ablation	*splice region variant*	*start lost*	*5 prime UTR variant*
upstream gene variant	*splice acceptor variant*	*regulatory region amplification*	*transcript amplification*
incomplete terminal codon variant	*downstream gene variant*	*3 prime UTR variant*	*feature elongation*
splice donor variant	*stop retained variant*	*inframe insertion*	*non coding transcript exon variant*
TFBS ablation	*stop gained*	*regulatory region variant*	*inframe deletion*
synonymous variant	*TFBS amplification*	*intron variant*	*feature truncation*
frameshift variant	*coding sequence variant*	*missense variant*	*NMD transcript variant*
TF binding site variant	*stop lost*	*intergenic variant*	*protein altering variant*
mature miRNA variant	*regulatory region ablation*	*non coding transcript variant*	

Table 2. **Consequence codes encapsulated in the VEP features.** The coding region classifier uses a binary vector to identify which of these consequences is annotated for a particular mutation.

3.4 Distance features

For non-coding SNVs we employ a related, but simpler approach: we measure the distance from each SNV to gene features annotated in the ENSEMBL gene models: *start codon*, *stop codon*, *gene*, *UTR*, *CDS* and *exon*. This approach is simple, yet should enable our models to learn relationships between SNVs and important gene elements. For example, exon boundaries help to identify mutations close to splice sites. Similarly, 5' gene boundaries identify mutations close to transcription start sites or promoter regions. To capture this information, we establish a window w around each mutation and measure the distance to the nearest example of each element. The features are then the distances to six element types, mapped onto the range $[0, 1]$ as follows. If a mutation is d positions away from the nearest element, $0 \leq d \leq w$, then the score s is given as:

$$s = \begin{cases} \frac{1}{d+1}, & d \leq w. \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

To identify the optimum setting for the window, w, we ran LOCO CV using values from $w = 1$ to $w = 10^6$ and found that $w = 10$ yielded the best performance (*Distance*, Table 3).

3.5 Kernel performance

We evaluated each of these kernels using LOCO-CV (Table 3,4). Conservation kernels yield relatively high accuracy for both non-coding and coding examples, consistent with previous studies showing the importance of conservation in non-coding regions (Kircher *et al.*, 2014; Shihab *et al.*, 2015). The spectrum kernel also scored well for both coding and non-coding regions, suggesting that it may learn patterns of regulatory motifs gained or lost through mutation. Genomic context features also appear amongst the top-performing kernels.

3.5.1 Kernel performance for non-coding variants

Features	Accuracy	Features	Accuracy
100-way cons.	0.91	GappedPeak	0.56
46-way cons.	0.91	BroadHMM	0.55
Distance	0.91	Histone Chip-Seq	0.55
Long RNA-Seq	0.78	Segmentation	0.55
RnaChip	0.72	RIP-Chip GeneST	0.54
Spectrum	0.71	TFBS Peak-Seq	0.53
RNA	0.66	Chia PET	0.53
Riken CAGE	0.68	DNase Uniform	0.52
Repeats	0.61	FDR peaks	0.52
NarrowPeak	0.60	TFBS Uniform	0.52
BroadPeak	0.59	FAIRE	0.51
Mapability	0.58	Footprints	0.51
GC content	0.57	SPP	0.51
Chromatin HMM (15)	0.56	RipSeq	0.50
Chromatin HMM (18)	0.56	Tiling	0.50
Short RNA-Seq	0.56		

Table 3. **Accuracy of individual feature groups in non-coding regions** shows that the conservation groups and distance to genomic features yield by far the highest accuracy of any group. Groups are sorted in descending order by balanced accuracy in LOCO-CV.

For non-coding regions, conservation scores and gene element distance features yielded by far the highest accuracy. Gene expression estimates given by RNA-Seq also provided good discrimination, as did evidence for RNA binding proteins and the nucleotide spectrum kernel. The final model also leverages evidence for chromatin interactions via the Chia PET assay, which is not the most accurate of the chromatin assays on its own, but improves accuracy in the context of these other groups (Section 3.6).

3.5.2 Kernel performance for coding variants

Features	Accuracy	Features	Accuracy
100-way cons.	0.84	Segmentation	0.53
46-way cons.	0.83	RIP-Chip GeneST	0.53
VEP features	0.75	TFBS Uniform	0.52
RNA	0.62	Chia PET	0.52
Long RNA-Seq	0.59	Mapability	0.52
Spectrum	0.59	TFBS Peak-Seq	0.52
RnaChip	0.58	DNase Uniform	0.51
BroadPeak	0.57	FAIRE	0.51
NarrowPeak	0.57	FDR peaks	0.51
GappedPeak	0.56	GC content	0.51
BroadHMM	0.56	SPP (TFBS)	0.51
Chromatin HMM (15)	0.56	Footprints	0.50
Chromatin HMM (18)	0.56	Repeats	0.50
Histone Chip-Seq	0.55	RipSeq	0.50
Riken CAGE	0.55	Tiling	0.50
Short RNA-Seq	0.54		

Table 4. **Accuracy of individual feature groups in coding regions** shows that the conservation scores and VEP features far outperform other feature groups. Groups are sorted in descending order by balanced accuracy in LOCO-CV.

In coding regions, conservation scores again yielded the highest accuracy, followed by VEP features (amino acid changes and consequence), gene expression estimates and RNA interaction features. Again the nucleotide spectrum kernel is among the top-performing groups. Interestingly, the final model improved when we added the *Segmentation* features that provide gene element information similar to the VEP *consequence* features (Section 3.6).

3.6 Data-level integration

The simplest kernel method for integrating different data sources is to combine the features from all sources into a single kernel. This allows a model to discriminate between classes by learning how features from one source may interact with features from other sources. Given more than 30 possible data sources, there are billions of possible combinations of feature groups, making exhaustive testing impractical. Instead, we used an approach similar to previous work in which we found that sequential learning could be an effective means to identify an optimal combination of groups (Rogers *et al.*, 2015). In this work, we use a forward selection method, in which we try all combinations of the top-performing kernel with the remaining kernels to identify the best two-kernel model. Next, we try all combinations of this two-kernel model with remaining kernels, and continue the process until accuracy plateaus or starts to decline (Figure 3).

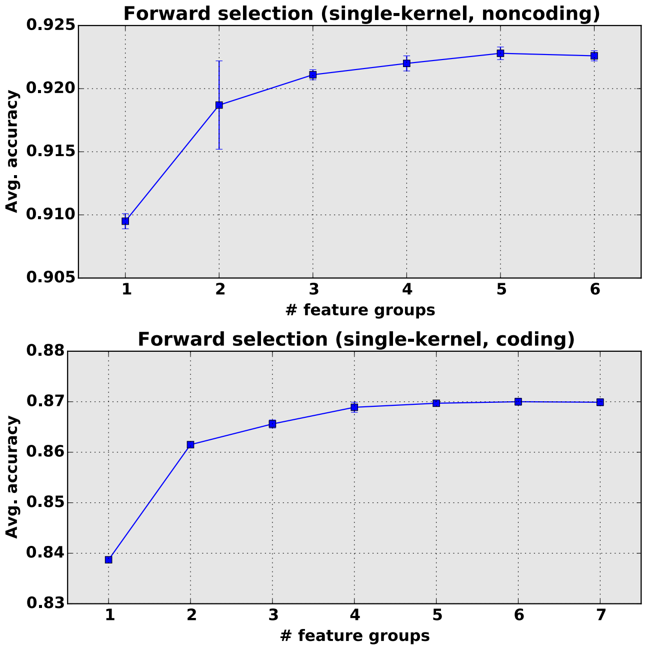

Fig. 3. Data-level integration for non-coding examples and coding examples: for data-level sequential learning we used balanced sets of 2,000 examples in LOCO-CV for each combination of features, and average accuracy over 10 LOCO-CV runs. **Top:** For non-coding regions, accuracy peaks at five feature groups (92.3% accuracy) after which it declines slightly. **Bottom:** For coding regions, accuracy reaches a nominal peak at six feature groups (87.0% accuracy).

For non-coding regions, the best model incorporated five feature groups (Figure 3, top): starting at 91.0% accuracy for the first group (*100-way conservation*), we see rapid improvement up to 92.1% accuracy for three groups, after which performance plateaus, reaching 92.3% accuracy with five groups. From that point onward, additional feature groups provide no evident advantage, as average accuracy declines. Hence our final non-coding region model uses a single kernel constructed from five feature groups.

For coding regions, our best model included six data sets (Figure 3, bottom): accuracy for the best feature group (again, *100-way conservation*) is 83.9% and climbs rapidly to 86.9% with four feature groups, reaching its peak with six feature groups at 87.0% accuracy. Our final coding-region model thus consists of a single kernel constructed from six feature groups.

3.7 Kernel-level integration

In previous studies we found that integration at the kernel level, using multiple kernel learning (MKL) and sequential learning strategies, tended to outperform integration at the data level (Rogers *et al.*, 2015; Shihab *et al.*, 2015, 2017b). We applied our sequential learning pipeline (Rogers *et al.*, 2015) to examples in both coding and non-coding regions. For non-coding regions, the best MKL model achieved 89% accuracy; for coding examples, the best MKL model achieved 85% accuracy. Both models thus yielded accuracy slightly below the best models from data-level integration.

3.7.1 Alternative classification models

For both coding and non-coding classifiers we investigated a variety of kernel-based models using the scikit-learn package (version 0.17.1) (Pedregosa *et al.*, 2011). We selected the package for its relatively robust performance and for the variety of models available. We evaluated seven different classification models to select the one yielding the best performance, and to observe changes in accuracy that could reflect potential overfitting (see Table 5). For each classifier we first used LOCO-CV to establish optimal parameters, then compared the accuracy, averaged over 10 runs, to identify the strongest performers in non-coding and coding regions. For each LOCO fold we used balanced training sets of 2,000 examples. In non-coding regions we found