Chromosome	Coding		Non-coding	
	COSMIC	1000G	COSMIC	1000G
1	1,463	3,506	5,760	5,427
2	897	1,847	5,888	5,436
3	1,194	1,373	4,567	4,358
4	565	1,026	4,365	4,574
5	580	1,228	4,310	4,129
6	812	2,137	3,446	3,820
7	790	1,401	4,455	4,490
8	470	1,003	3,970	3,992
9	596	1,234	2,441	2,226
10	495	1,028	3,258	3,460
11	905	2,365	3,300	3,445
12	727	1,339	2,689	2,609
13	247	370	1,935	2,044
14	344	1,077	2,039	1,926
15	408	888	1,902	1,832
16	506	1,760	2,522	2,593
17	1,321	1,818	1,821	1,753
18	180	365	1,601	1,691
19	1,107	4,215	1,704	1,843
20	339	817	1,611	1,591
21	163	451	1,530	1,257
22	280	783	972	1,132
Total	14,389	32,031	66,086	65,628

Supplementary Table S1. 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 2 to 5 times as many neutral (1000G) examples as positive (COSMIC) examples.

Features

Conservation and ENCODE data

As in previous studies[3,3,4] we evaluated over 30 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[5] and PhastCons[6] scores, derived from the multiple sequence alignment of 46 and 100 vertebrate genomes to the human genome[7].

- *Histone Modifications*: we used ChIP-Seq peak calls for 14 histone modifications across 45 cell lines from ENCODE[8] and narrow, broad and gapped regions of enrichment based on consolidated epigenomes from the NIH Roadmap project[9].

- *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.

We found just four of the ENCODE feature groups (Table S4, bold values) relevant to somatic mutations in noncoding regions, while none of these groups yielded good discrimination in coding regions. One reason may be the sparse coverage of these data: we found that many examples had no corresponding data in these feature groups. This sparsity limits the information available for training our models and for making novel predictions, and so can undermine the performance of classifiers that rely on these data[4].

Genomic context features

For our coding classifier 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[10]. 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. For non-coding SNVs we employ a simpler approach by measuring the proximity of each SNV to gene features such as the transcription start site (TSS), splice sides, and codons. In this method we establish a window w around each mutation and count the annotated features that fall within that window, or measure the distance to each feature (Supplementary Table S3).

Spectrum features

Our models should learn the sequence characteristics that are most susceptible to oncogenic mutations in both coding and non-coding regions. As a simple way to capture the disruption that may occur in the sequence surrounding a mutation, we use *spectrum* kernels[11] 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[11]), 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 $\mathscr{A}$ as follows:

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

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[12]). We found that these features perform competitively on their own, and improve balanced prediction accuracy in our merged-kernel tests (Supplementary Figures S6-S7). As this approach makes no assumptions about the kind of RNA binding proteins that may be impacted by a particular mutation, it eliminates the requirement 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 (Table S2). In both cases we found that a window size $w = 3$ and maximum k-mer size 2 yielded the best results: 59.5% balanced accuracy for the coding model and 56.9% for the non-coding model.

	Maximum k-mer size							
	coding				non-coding			
Window	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$
$w=1$	0.54	—	—	—	0.57	—	—	—
$w=3$	0.58	**0.60**	0.59	—	0.56	**0.57**	0.56	—
$w=5$	0.59	0.59	0.59	0.59	0.54	0.56	0.56	0.55
$w=7$	0.58	0.58	0.59	0.58	0.53	0.55	0.55	0.55
$w=9$	0.58	0.58	0.58	0.58	0.53	0.55	0.55	0.54

Supplementary Table S2. Spectrum kernel performance in coding regions using different values for window size w and maximum k-mer size k reveals that the highest average balanced accuracy occurs at a window size of 3 and maximum k-mer size of 2 for both coding (60%) and non-coding (57%) models (values in **bold**). Accuracy was averaged over 30 LOCO-CV runs with a balanced training set size of N=2,000 for each fold.

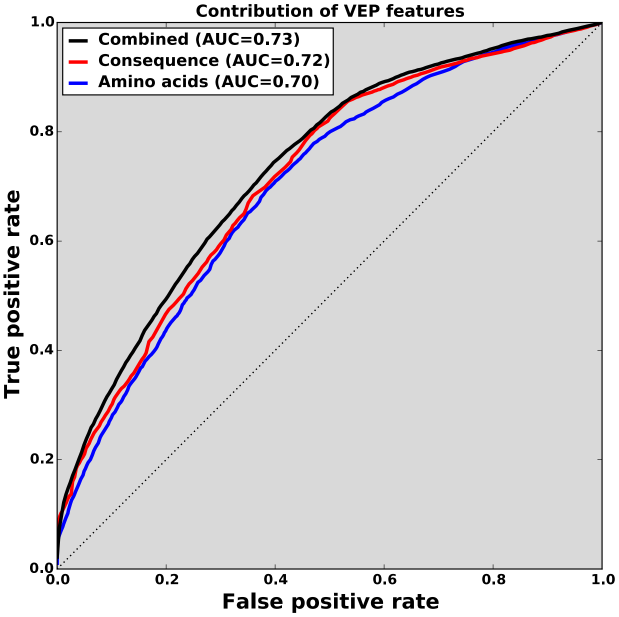
Variant Effect Predictor

We concatenate two vectors to encapsulate all of the Variant Effect Predictor (VEP) features used in the CScape 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 returned by the VEP (Table S3). 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.

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

Supplementary Table S3. 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.

We find that both sets of VEP features perform similarly (Figure S4). In LOCO cross-validation using balanced training sets of 2,000 examples for each chromosome, the *Consequences* features yield 66.0% balanced accuracy on average, while the *Amino acids* features yield 65.5%. When we combine these features we see a slight improvement overall, up to 67.4%, making it the most accurate group used for the coding region classifier (Table S4).



Supplementary Figure S4. ROC curves depict the contribution of distinct VEP feature sets in coding regions. The amino acid features encapsulate the possible amino acids represented by the wild-type and mutant sequences. Where a mutation falls within multiple transcripts several amino acids may be reported. The consequence features encapsulate locations within a gene that may be impacted by a mutation. Both sets of features are informative on their own; however the combination of the two yields slightly better performance.

Distance features

As noted in the main text, we apply the VEP features only to our coding predictor, since amino acid features are not relevant to SNVs in non-coding regions, and there are far more non-coding positions than coding positions. 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 either count each of the elements that fall within the window, or measure the distance to the nearest example of each element. We encapsulate counts features using a vector of six integer values. Distance features also comprise six features, but the distance to nearest feature is mapped into 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:

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

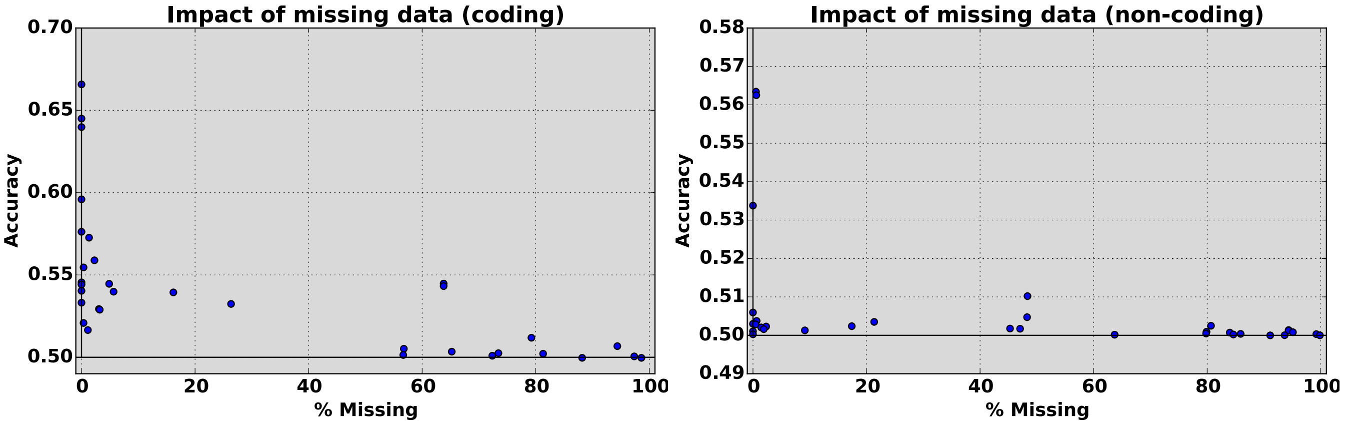
To identify the optimum setting for the window, w, we ran LOCO CV using values from $w = 1$ to $w = 10^6$ and found that $w = 10$ yielded the best performance for both coding and non-coding models. For coding models, the scoring representation given by Equation 2 worked best, while for non-coding models, the counts representation worked best. The balanced accuracy of both models are recorded as *Distance* in Table S4.

Feature groups

We used LOCO-CV to evaluate more than 30 distinct feature groups: two conservation-based groups, the spectrum kernel, two groups designed to encapsulate the genomic context (gene region, any corresponding amino acids, or proximity to gene features), and 27 groups based on ENCODE data (Table S4). Conservation groups yield relatively high balanced accuracy for both non-coding and coding examples, consistent with previous studies showing the importance of conservation in noncoding regions[3, 13]. The spectrum kernel scored equally well for 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 groups. ENCODE features generally perform poorly, due partly to the number of missing values (see Figure S5).

	Balanced accuracy	
Feature group	Non-coding	Coding
Conservation scores		
46-way cons.	**0.572**	**0.655**
100-way cons.	0.569	0.649
Spectrum kernel	**0.570**	**0.597**
Genomic context		
VEP	—	**0.674**
Distance	**0.515**	**0.605**
ENCODE features		
Mappability	**0.514**	0.541
BroadPeak	0.511	0.587
NarrowPeak	0.511	0.567
RNA	0.509	0.540
GappedPeak	0.507	0.575
BroadHMM	0.507	0.565
ChromHMM15	0.507	0.564
LongRnaSeq	0.507	0.553
Segmentation	0.507	0.538
ChromHMM18	0.506	0.567
ChiaPet	0.506	0.545
Repeats	0.505	0.502
RnaChip	0.504	0.542
GC	0.504	0.537
Histone ChipSeq	0.503	0.570
GeneSt	0.502	0.556
Riken	0.502	0.551
ShortRnaSeq	0.501	0.525
PeakSeq	0.501	0.510
TFBS Uniform	0.501	0.501
DNase Uniform	0.500	0.519
FDR peaks	0.500	0.518
RipSeq	0.500	0.512
FAIRE	0.500	0.507
SPP (TFBS)	0.500	0.501
Footprints	0.500	0.500
Tiling	0.500	0.500

Supplementary Table S4. Conservation, spectrum kernel, genomic context and ENCODE feature groups evaluated on COSMIC somatic mutations and 1000 Genomes neutral mutations. All feature groups were evaluated using LOCO cross-validation in which each training set was constructed from a balanced set of 2,000 examples. Feature groups are organized by category and shown in descending order of balanced accuracy for *non-coding* data. Shown in bold are the balanced accuracy statistics for the feature groups used in our final models. While none of the datasets individually yields high balanced accuracy, we find that as in previous studies, conservation scores are among the most informative features. Spectrum kernels also appear to provide some discrimination. In non-coding regions, we find that one of the ENCODE groups, *Mappability* provided enough discrimination power to improve performance in our final model. In coding regions, the ENCODE datasets generally do not perform as well as conservation or spectrum features, while the VEP features yield by far the highest balanced accuracy.



Supplementary Figure S5. Graphs depicting balanced accuracy as a function of the proportion of missing values show a direct correspondence between sparsity, measured as the proportion of missing values in a feature group, and performance, measured as balanced accuracy.

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 16 possible data sources for noncoding regions and 17 sources for coding regions, there were up to 131,000 possible combinations of feature groups, so exhaustive testing was impractical. Instead, we used an approach similar to previous work in which we found that greedy sequential learning could be an effective means to identify an optimal combination of groups[4]. To identify the data sources to include in the final model, we combine the two top-ranked data sources into a single kernel and record its balanced accuracy during LOCO-CV. We build subsequent models by adding data sources in descending order of balanced accuracy, constructing a kernel for each combination of data sources. We select as our final combination the data sources associated with the kernel where balanced accuracy reaches a plateau or declines thereafter.

For non-coding regions, the best model incorporated the top five feature groups: *46-way conservation*, *100-way conservation*, *Spectrum*, *Genomic context* and *Mappability* (Figure S6). At 58.0%, the kernel with two feature groups represents an improvement over the best individual kernel (*46-way conservation*, 57.2%, Table S4). Balanced LOCO-CV accuracy continues to increase with additional features until the top five feature groups are combined (60.6% balanced accuracy). From that point onward, additional feature groups provide no evident advantage, as average balanced accuracy declines. Hence our final noncoding-region model uses a single kernel that contains these top five feature groups.

For coding regions, our best model again included the top five data sets: *VEP* (including amino acid substitutions), *46-way conservation*, *100-way conservation*, *Genomic context* and *Spectrum* (Figure S6, right). The first two data sets yield a substantial improvement over either one individually: 69.2% balanced accuracy, compared with 67.4% for the top-ranked VEP kernel alone. Balanced LOCO-CV accuracy continues to increase with additional feature groups until the top five feature groups are combined (over 70% balanced accuracy). Our final coding-region model thus consists of a single kernel containing these top five feature groups.

Alternative classification models

For both coding and noncoding classifiers we investigated a variety of kernel-based models using the scikit-learn package (version 0.17.1)[14]. 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 balanced accuracy that could reflect potential overfitting (see Table S5). For each classifier we first used LOCO-CV to establish optimal parameters, then compared the balanced accuracy, averaged over 30 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 that gradient boosting[15], SVM models (see, e.g.,[12]) and Adaboost[16] yield the highest balanced accuracy, with no significant differences observed between them. In coding regions, gradient boosting and random forests[17] yielded the highest balanced accuracy. Based on these results we selected gradient boosting for our models in both regions.