Supplementary Information

FunSVPT: Function-based somatic variants prioritization tool for cancer whole-genome sequencing

Yao Fu[1], Zhu Liu[2], Shaoke Lou[3], Jason Bedford[1], Xinmeng Jasmine Mu[1,4], Kevin Yip[3], Ekta Khurana[1,5,*], Mark Gerstein[1,5,6,*]

[1] Program in Computational Biology and Bioinformatics, Yale University, New Haven, Connecticut, 06520,United States of America
[2] School of Life Science, Fudan University, Shanghai, 200433, P.R. China
[3] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
[4] Broad Institute of Harvard and MIT, Cambridge, MA 02142, USA
[5] Molecular Biophysics and Biochemistry Department, Yale University, New Haven, Connecticut, 06520, United States of America
[6] Department of Computer Science, Yale University, New Haven, Connecticut, 06520, United States of America

1. Overview

FunSVPT is an integrative tool to first build an organized data context from various resources and then prioritize somatic variants from cancer whole-genome sequencing. Supplementary Table 1 shows the comparison of different tools.

	Haploreg	RegulomeDB	ANNOVAR	GEMINI	FunciSNP	VEP	FunSVPT
Rare/novel variants		✓	✓	✓		✓	✓
Conservation	✓		✓			✓	✓
Motif gain	✓						✓
Motif breaking	✓	✓					✓
Functional annotations	✓	✓	✓	✓	✓	✓	✓
Non-coding association to genes							✓
Network analysis							✓
Scoring scheme		✓	✓				✓
Recurrent analysis							✓
Gene prioritization							✓
Differential gene expression analysis							✓
Web server	✓	✓				✓	✓

Supplementary Table 1. Comparison of various tools.

2. Material and Methods

Variants in functional annotations and conserved regions

User-input variants are first filtered against polymorphisms based on user-defined MAF (minor allele frequency) threshold. In current data context, polymorphisms are from 1000 Genomes Phase1 data and all coordinates are based on hg19. Currently, only SNVs will be analyzed (indels and SVs will be filtered out).

Whole genome GERP scores (Cooper, et al., 2005), ultra-conserved regions (Bejerano, et al., 2004) and sensitive/ultra-sensitive regions (Khurana, et al., 2013) are collected to examine whether a variant occurred in conserved regions. Each variant will be annotated with GERP score, in ultra-conserved or sensitive/ultra-sensitive regions. We also provide the scripts for users to define novel sensitive/ultra-sensitive regions (see **Building data context**).

We compiled transcription factor binding, DNase1 hypersensitive sites (Thurman, et al., 2012) and enhancer data (Ernst and Kellis, 2012; Hoffman, et al., 2012; Hoffman, et al., 2013; Yip, et al., 2012) from recent ENCODE release (Consortium, et al., 2012) together with GENCODE annotations to define functional regions. Hot regions (highly occupied by transcription factors) are obtained from (Yip, et al., 2012). Variants are intersected and

annotated with these functional annotations.

Association between regulatory elements and genes

Associating regulatory elements to genes provides a rich resource to interpret likely functional impact of noncoding variants. FunSVPT defines both proximal and distal associations to genes. For proximal associations, we assign variants in gene promoters (-2.5kb) to their nearby genes. For distal associations, in addition to the ones identified in (Yip, et al., 2012), we further expanded the method to all ENCODE non-coding elements and identified ~769K regulatory elements associated with ~17K genes (see 'Correlating histone marks with gene-expression data to identify likely target genes of distal regulatory elements').

Correlating histone marks with gene-expression data to identify likely target genes of distal regulatory elements

1. Definition of distal regulatory modules (DRMs)

We started with a list of regulatory regions from three different types, namely transcription factor binding peaks (TFP), DNase hypersensitive sites (DHS) and Segway/ChromHMM-predicted enhancers. All regulatory regions at least 1kb from the closest gene according to the Gencode v7 annotation (Harrow, et al., 2012) were defined as a distal regulatory module (DRM).

2. Identifying potential regulatory targets of each DRMs

We grouped different transcripts of a gene sharing the same transcription start site as a transcription start site expression unit (tssEU). For each DRM, we first considered all tssEUs within 1Mb from it as its candidate targets. We then correlated some activity/inactivity signals at a DRM and the expression of its candidate target tssEUs, and called the ones with significant correlation values as potential DRM-target pairs as follows.

At the DRMs, we considered the enhancer marks H3K4me1 and H3K27ac as two types of activity signals, and DNA methylation as an inactivity signal. The activity level of each DRM was defined as the number of sequencing reads aligned to the DRM from the corresponding ChIP-seq experiments. The methylation level of a DRM was defined as follows. For each CpG site i within a DRM, we counted the number of reads that support the methylation of it (m_i), and the total number of reads covering it (n_i). The methylation level of the DRM was then defined as the ratio of their sums across all CpG sites in the DRM, $\frac{\sum_i m_i}{\sum_i n_i}$. For each tssEU, we defined its expression level as the number of RNA-seq reads aligned to the [TSS-50, TSS+50] window. Both the activity signal levels and gene expression levels were normalized by the total reads, then multiplied by one million to keep them within an easily readable range of values.

We collected all bisulfite sequencing, ChIP-seq and RNA-seq data from the Roadmap Epigenomics project website (Bernstein, et al., 2010) (EDACC release 9[1]). We considered 19 tissue types with data for both the activity signals and gene expression, and 20 tissue types with data for both the inactivity signal and gene expression. For RNA-seq,

we used the paired-end 100bp Poly-A enriched data sets. For experiments with replicates, we used the mean value across the replicates as the expression level of a gene.

For each DRM-candidate target pair, we computed the correlations of their activity/inactivity and expression levels across the different tissue types. We computed both value-based Pearson correlation and rank-based Spearman correlation. The statistical significance of each correlation value was evaluated by computing a p-value based on one-tailed tests using the built-in functions in R. Briefly, for Pearson correlation, the correlation values would follow a t distribution with $n-2$ degrees of freedom (where n is the number of tissue types) if the samples were drawn independently from normal distributions. The Fisher's Z transformation was used to compute the p-values. For Spearman correlation, the p-value was computed based on a procedure proposed by Hollander and Wolfe (Wolfe, 1973). For activity signals, we considered the right-tail, which means we looked for correlations significantly more positive than would be expected by chance. For inactivity signals, we considered the left-tail, which means we looked for correlations significantly more negative (i.e., strong anti-correlations) than would be expected by chance. All p-values were then adjusted for multiple hypothesis testing using the Bonferroni, Holm, Benjamini-Hochberg (BH) or Benjamini-Yekutieli (BY) methods.

3. Software pipeline (see **Building data context**)
We have packaged our computer programs as a software pipeline for users to define DRMs and identify their potential targets according to the above procedure on their own data files. The pipeline involves the following three main steps:
a. Read user-defined regulatory regions, annotation file, tssEU expression, and meta-data of the data files (file names, total reads, etc.).
b. Calculate activity and inactivity levels at the DRMs based on the Roadmap Epigenomics data.
c. Correlate the activity/inactivity levels with the tssEU expression levels and determine their statistical significance, either using the pre-computed values or to compute the significance values on the fly based on the user-defined regulatory regions.

Detect differentially expressed genes
We incorporated a module to detect differentially expressed genes in cancer samples (relative to normal samples) from RNA-Seq data. When provided with gene expression files, our module calls NOISeq (Tarazona, et al., 2011) when having RPKMs and DESeq (Anders and Huber, 2010) with raw read counts from reads-mapping tools to detect differentially expressed genes. Genes that are up- or down- regulated with FDR < 0.05 (with biological replicates) and FDR < 0.1 (without replicates) in cancer samples are identified and annotated in the output.

Network analysis of variants associated with genes
For each variant associated with genes, we examine their network properties in various networks, such as protein-protein interaction, regulatory and phosphorylation networks. For each network, we calculated the cumulative probabilities of associated genes. Genes that are highly connected (higher cumulative probability) in biological networks are more

likely to be functional important. If a variant is associated with multiple genes or the associated gene participate in multiple networks, the highest cumulative probability is used as the continuous score for network centrality. Scripts are provided to calculate centrality in networks (see **Building data context**). User can easily incorporate other networks in this analysis.

Recurrence across multiple cancer samples

One important criterion to find cancer genes is to examine their recurrence in multiple samples. We extended the concept to non-coding regulatory elements. FunSVPT can detect recurrent mutations, genes and regulatory elements in multiple samples.

Running FunSVPT on 570 samples of 10 cancer types (Alexandrov, et al., 2013; Baca, et al., 2013; Berger, et al., 2011), we created the recurrence database for somatic mutations from cancer whole-genome sequencing. The summary table of the data is listed below.

Cancer Type	Sample Number	Somatic Mutations (SNV)	Recurrent Genes/Elements/Mutations
AML	7	271~1068	1
Breast	119	1043~67347	69140
CLL	28	522~3338	709
Liver	88	1348~25131	74144
Lung Adeno	24	9284~297569	162165
Lymphoma B cell	24	1502~37848	4233
Medulloblastoma	100	44~47440	2793
Pancreas	15	1096~14998	2591
Pilocytic Astrocytoma	101	2~926	58
Prostate	64	1430~18225	36327

Supplementary Table 2. Summary of currently collected cancer types

High-impact variants in motifs: Nucleotide resolution effects

1. Motif breaking

When variants hit transcription factor binding motifs in ENCODE Chip-Seq peak, we examined their motif breaking or conserving effect using position weight matrixes (PWM) (Mu, et al., 2011). Motif breaking events are defined as variants decreasing the PWM score. For somatic mutations, mutated allele is compared to the reference allele. We also provide the option for germline or personal genomes, which compared to the ancestral allele, since the functional impact of the variant reflects the historical event when the polymorphism was first introduced in the human population. Motif breaking events are reported in the output with the PWM changes. Transcription factor PWMs are obtained from ENCODE project, consisting TRANSFAC, JASPAR and *de novo* motifs.

2. Gain of motif

We developed an automated tool to detect gain-of-motif events. Whole genome motif scanning generally discovers millions of motifs, of which, a large fraction are false positives. To restrict motif scanning, we focused on variants occurred in promoters

(defined as -2.5kb from transcription starting site) or regulatory regions associated with genes (see 'Correlating histone marks with gene-expression data to identify likely target genes of distal regulatory elements'). For each variant, +/- 29bp are concatenated from human reference genome (motif length is generally <30bp). For each PWM, we scanned the 59bp sequence. For each candidate sequence encompassing the variant, we evaluated the sequence score with the mutated allele using TFM-Pvalue (Touzet and Varre, 2007) (with respect to the PWM). If the p-value of the mutated allele < 4e-8, whereas the reference allele is not, we define the variant creating a motif. The process is repeated for all PWMs. The sequence score changes are reported in the output. We applied this analysis to the two TERT promoter mutations and the result is in the following table (motif name # motif start position # motif end position # motif strand # variant position # alternative sequence score # reference sequence score).

Mutation Position	**Gain of Motif**
chr5 1295250	Ets_known10#1295246#1295252#+#4#5.743#2.472
chr5 1295228	Ets_known10#1295223#1295229#+#5#5.743#1.893

Supplementary Table 3. Gain of motif analysis on known TERT promoter mutations.

Weighted-sum scoring scheme to prioritize variants

FunSVPT has separate scoring schemes for coding and non-coding mutations.

1. Coding scoring scheme

Please refer to the (Khurana, et al., 2013). Here is the brief description. The effect of mutations occurred in coding regions (GENCODE 16 for current version. Users can replace this with other GENECODE annotations) are analyzed with VAT (variant annotation tool) (Habegger, et al., 2012). Mutations are ranked based on the following scheme (each criterion gets score 1): 1) non-synonymous; 2) premature stop; 3) is the gene under strong selection; 4) is the gene a network hub; 5) recurrent

2. Non-coding scoring scheme (weighted-sum scoring scheme)

Features used to score non-coding variants are shown in Supplementary Table 4. In general, features can be classified into two classes - discrete and continuous. Discrete features are binary, such as in ultra-conserved regions or not. Continuous features: 1. Gerp score of variant; 2. Motif-breaking value is the difference of relative frequencies between reference and mutated alleles in PWMs; 3. Motif-gaining value is the difference of sequence scores between mutated allele and reference allele; 4. Network centrality is the network position of the gene (e.g. cumulative probability of degree centrality). Note that for motif-gaining values, the PWMs (actually 'relative frequency matrix' used in this paper) are transformed using log likelihood and the sequence scores are calculated with the transformed PWMs. If variants possess multiple values of particular feature (e.g. participate in multiple networks), the largest value is used.

We then weight each feature based on the mutation pattern observed in 1000 Genomes polymorphisms. Features that are frequently observed in polymorphisms are less

important, thus should be weighted less. We randomly selected 10% of 1000 Genomes Phase 1 variants (~3.7M) and run through FunSVPT. The probabilities of observing each feature in polymorphisms are calculated as p_i. We chose to use 1- Shannon entropy *(1)* as our measure to assign weight. The value ranges from 0 to 1 and is monotonic increasing when the probability is between 0 and 0.5. a) For discrete features, we calculated *(1)* using the probability of observing features. b) For continuous features, taking 'motif-breaking' as an example, for each motif-breaking value v observed in 1000 Genomes, we calculated the probability of observing values $\geq v$ then used this probability to calculate *(1)*. The curve of motif-breaking values and 1-Shannon entropy reflects the uncertainly of observing the motif-breaking values in polymorphisms. This scheme is also applied to motif-gaining and network centrality features (Supplementary Figure 1).

The criterion of 'Gerp >2' has been commonly used to define conserved bases (Consortium, et al., 2012). For Gerp score, we decided to use sigmoid transformation to transform the scores to range 0 and 1. The parameters we chose make the sigmoid curve sharp at 'Gerp = 2' (Supplementary Figure 1). The sigmoid transformation preserves the cut-off of 'Gerp > 2' and makes the score continuous at the same time. We calculated *(1)* regarding 'Gerp > 2' as a discrete feature. Then we used $w_i * sigmoid\ transformed\ value$ to assign weight for continuous Gerp scores.

For discrete features, the weight calculated is shown in Supplementary Table 4. For each user-input variant, the score is the sum-up of weights of all its features. Because some features are subset of other features, to avoid overweighting similar features, we considered the dependency structure of features when calculating the sum-up scores (Supplementary Table 4). When observing leaf features, the weights of root features are ignored. For example, when variant occurs in ultra-sensitive regions, the weights of 'functional annotations' and 'sensitive regions' are not used in the sum-up. Other features are considered independent. Variants ranked on top of the output are those with higher scores and are most likely to be deleterious.

$$w_i = 1 + p_i log_2 p_i + (1 - p_i) log_2 (1 - p_i) \quad (1)$$

$$S = \sum w_i \quad (2)$$

$For\ discrete\ features, p_i\ is\ the\ probability\ of\ observing\ feature\ i;$
$For\ continuous\ features, p_i\ is\ the\ probability\ of\ observing\ values \geq v\ for\ feature\ i.$

Features	Class	Weight
Functional annotations	Discrete	0.18636650
Sensitive	Discrete	0.96918819
Ultra-sensitive	Discrete	0.99723918
Motif-Breaking	Continuous	-
Motif-Gaining	Continuous	-
Network centrality	Continuous	-
Gerp score > 2	Continuous/Discrete	0.62278676
Ultra-conserved	Discrete	0.99974654
HOT Regions	Discrete	0.79753934
Regulatory regions associated to genes	Discrete	0.003531882
Recurrent	Discrete	1

Dependency structure of features (leaf feature is a subset of root feature):

Functional annotations
- Sensitive
  - Ultra-sensitive

Functional annotations
- Motif-Breaking

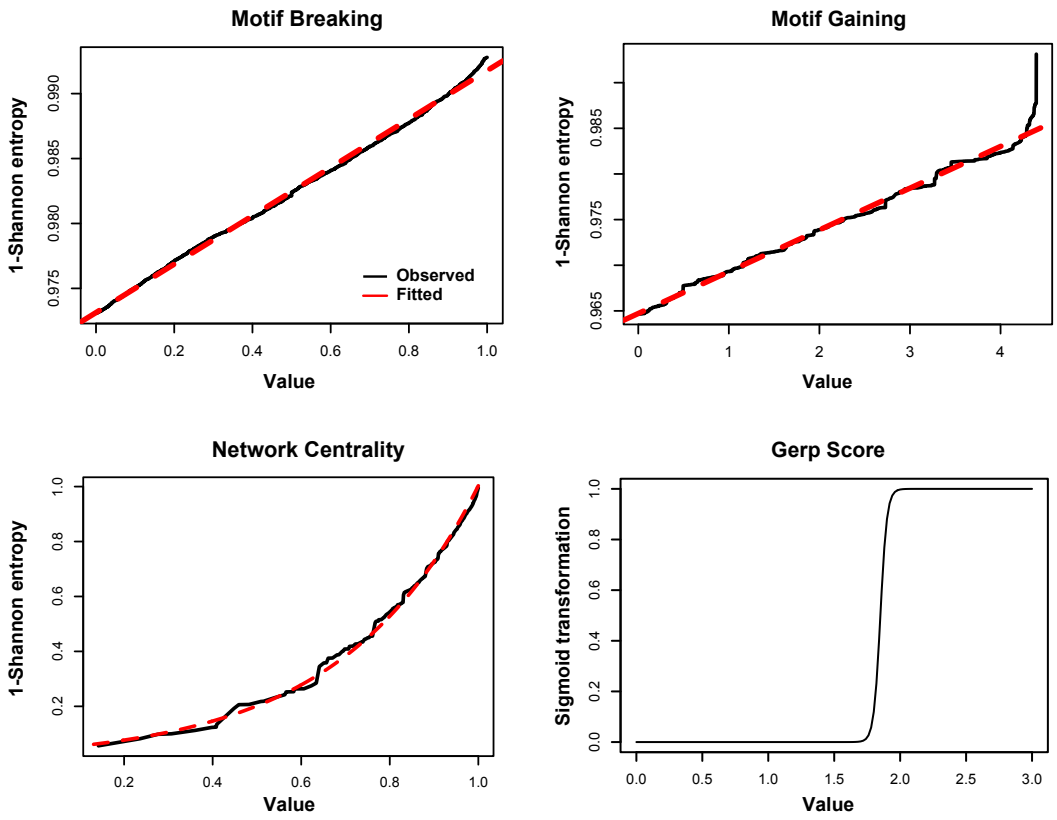
Functional annotations
- HOT regions

Regulatory regions associated to genes
- Network centrality

Regulatory regions associated to genes
- Motif-Gaining

Gerp score (>2)
- Ultra-conserved

Supplementary Table 4. Weighted scoring scheme



Supplementary Figure 1. Weight assignment for continuous features

3. FunSVPT manual

FunSVPT consists of two components: building data context and variants prioritization.

Building data context

Scripts:

* 0.define.proximal.distal.regions.pl

Generating distal/proximal regions for provided interval annotations (based on GENCODE data).

* Define conserved annotation categories using polymorphisms data.

1. Define conserved annotation categories from scratch (hg19).

1.1.Randomization.pl

The program using element-sliding method to generate null distributions for fraction of rare variants for each category.

1.2.FDR.r

FDR calculation for the randomization. This script can also output significant categories based on FDR.

2. Define novel conserved regions upon those defined in Khurana et al., (Science 2013).Only applicable to small number of categories, ~ 5.

2.sensitive.regions.delta.increment.pl

The input polymorphisms used in the paper is ' SNP.lowcov.noncodingGENCODE7.1kgMask.bed'.

* 3.gencode.process.pl

This script processes GENCDOE gtf file to obtain 'promoter','cds','intron' and 'utr' region files, which are used by variants prioritization part.

* 4.network.analysis.r

This script generates network centralities for input network (degree or betweenness).

* 5.PWM.score.cut.pl

This script generates the 'motif.score.cut' file for the variants prioritization part. TFMpvalue-pv2sc in 'TFM-pvalue' package is used to generate sequence score cut-offs for defined p-value cut-off. The file 'motif.score.cut' is used to speed up the gain-of-motif analysis.

Variants prioritization

FunSVPT code consists four parts. 1) The analysis module is "FUNSVPT/lib/FunSVPT.pm" containing all of the subroutines. 2) The executable file "FunSVPT" accepts the input parameters and passes it onto 3) "FunSVPT.pl", which stores the data path and organizes the modules in "FunSVPT.pm" to the pipeline. 4) "differential_gene_expression.r" is an R script detecting differentially expressed genes between cancer and benign samples.

1. Dependencies

The proper execution of FunSVPT depends on the following tools.

1. bedtools *(http://code.google.com/p/bedtools/downloads/list)*
   For intersection analysis and sequence retrieval.
2. tabix *(http://sourceforge.net/projects/samtools/files/tabix/)*
3. VAT (variant annotation tool - snpMapper Module) *(http://vat.gersteinlab.org/index.php)*
   If you are only interested in non-coding variants, you don't need to install VAT. But remember to use '-nc' option.
4. TFMpvalue-sc2pv *(http://bioinfo.lifl.fr/TFM/TFMpvalue/)*
   Calculate p value of motif scores regarding to its PWM.
5. bigWigAverageOverBed *(http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/)*
   Retrieve gerp scores. Note that gerp data file is huge. If you are not interested in gerp scores, gerp file and bigWigAverageOverBed are not needed.
6. R *(http://www.r-project.org)*
   Only needed for differential gene expression analysis.
7. Perl package Parallel::ForkManager *(http://search.cpan.org/~szabgab/Parallel-ForkManager-1.03/lib/Parallel/ForkManager.pm)*

Required for parallel running.
Please make sure you have Perl 5 and up.

2. FunSVPT tool installation

FunSVPT is a PERL- and Linux/UNIX-based tool. At the command-line prompt, unzip the tool and type the following. The purpose is to write path of FunSVPT.pm to your environment.

$ cd FunSVPT-0.2/
$ cd FUNSVPT/
$ perl Makefile.PL
$ make
$ make test
$ make install

3. Required data files

Please download the following data files from 'http://FunSVPT.gersteinlab.org' and put them in the folder '$FunSVPT-0.2/data/ '. If you use 'wget', please use 'wget http://FunSVPT.gersteinlab.org/static/data/data.tar.gz'. If you plan to use your own data, please prepare them following the described format. All data are based on hg19.

1. 1kg.phase1.snp.bed.gz (bed format)
   Contents: 1000 Genomes Phase I data with minor allele frequency in bed format.
   Columns: chromosome, start position (0-based), end position, MAF (minor allele frequency)
   Purpose: to filter out input SNVs based on user-defined allele-frequency threshold.
2. All_hg19_RS.bw
   Contents: binary file containing base-wise gerp score. Downloaded from http://hgdownload.cse.ucsc. edu/gbdb/hg19/bbi/All_hg19_RS.bw
   * Note: This file is ~7G. If you don't want to retrieve gerp score for variants, then no need to download this file.
3. HOT_region.bed (bed format)
   Contents: highly occupied region from Yip et al., (Yip, et al., 2012)
   Columns: chromosome, start position, end position, cell line info
   Purpose: to examine whether variants occur in hot regions.
4. ENCODE.annotation.gz (bed format)
   Contents: compiled annotation files from ENCODE, GENCODE v7 and others, including Dnase I hypersensitive sites, transcription factor binding peak, pseudo-genes, non-coding RNAs, enhancer regions (chromhmm, segway and distal regulatory modules (Yip, et al., 2012)).
   Columns: chromosome, start position, end position, annotation.
   Purpose: to annotate SNVs in ENCODE regions.
5. ENCODE.tf.bound.union.bed (bed format)
   Contents: transcription factor (TF) binding motifs under peak regions.
   Columns: chromosome, start position, end position, motif name, , strand, TF name
   Purpose: used for motif breaking analysis
6. gencode.v7.cds.bed (bed format)
   Contents: extracted CDS information from GENCODE v7.
   Columns: chromosome, start position, end position
   Purpose: locate coding SNVs.
7. gencode.v7.promoter.bed (bed format)

Contents: promoter regions, defined as -2.5kb from transcription start site (TSS)
Columns: chromosome, start position, end position, gene.
Purpose: to associate promoter SNVs with genes

8. gencode.v7.annotation.GRCh37.cds.gtpc.ttpc.interval
   Purpose: used by variant annotation tool (VAT).
9. gencode.v7.annotation.GRCh37.cds.gtpc.ttpc.fa
   Purpose: used by variant annotation tool (VAT).
10. drm.gene.bed (bed format)
    Contents: distal regulatory module linked to genes.
    Columns: chromosome, start position, end position, gene, p-value, cell-lines
    Purpose: to associate enhancer SNVs with genes
11. motif.PFM
    Contents: position frequency matrix (PFM) for ENCODE TFs.
    Purpose: used for motif breaking and gain of motif calculation
12. PPI.hubs.txt
    Purpose: defined hub genes in protein-protein interaction network
13. REG.hubs.txt
    Purpose: defined hub genes in regulatory network
14. GENE.strong_selection.txt
    Purpose: genes under strong negative selection (fraction of rare SNVs among non-synonymous variants).
15. human_ancestor_GRCh37_e59.fa
    Contents: contains human ancestral allele in hg19, Ch37.
    Purpose: for motif breaking calculation in personal or germline genome.
    * Note: for somatic analysis, this file is not needed.
16. human_g1k_v37.fasta
    Contents: human reference genome
    Purpose: for gain-of-motif analysis
17. sensitive.nc.bed (bed format)
    Contents: coordinates of sensitive/ultra-sensitive regions.
    Purpose: to find SNVs in sensitive/ultra-sensitive regions.
18. ultra.conserved.hg19.bed
    Contents: ultra-conserved region in (Bejerano, et al., 2004).
19. motif.score.cut
    Contents: pre-calculated PWM scores corresponding to 4e-8.
    Purpose: to speed up the gain-of-motif analysis
20. regulatory.network
    Contents: human regulatory network from (Gerstein, et al., 2012)
21. cancer.genes
    Contents: cancer genes from Cancer Gene Census (Futreal, et al., 2004)
22. actionable.gene
    Contents: actionable genes from (Wagle, et al., 2012)

4. Running FunSVPT

To display the usage of FunSVPT, type "./FunSVPT". It will show the following instructions.

** Usage: ./FunSVPT –f file -maf MAF -m <1/2> -inf <bed/vcf> -outf <bed/vcf> -nc -o path -g file -exp file -cls file -exf <rpkm/raw>*

Options:

-f	*User Input SNVs File*
-maf	*Minor Allele Frequency Threshold to filter 1KG SNVs*
-m	*1 - Somatic Genome; 2 - Germline or Personal Genome*
-inf	*input format - BED or VCF*
-outf	*output format - BED or VCF*
-nc	*[Optional] Only do non-coding analysis, no need of VAT (variant annotation tool)*
-o	*[Optional] Output path, default is the directory 'out'*
-g	*[Optional] gene list, only output variants associated with selected genes.*
-exp	*[Optional] gene expression matrix*
-cls	*[Optional] class file for samples in gene expression matrix*
-exf	*[Optional] gene expression format - rpkm / raw*

Default Options: -maf 0 -m 1 -outf vcf -o out

** Multiple Genomes with Recurrent Output*

Option 1: Separate multiple files by ','
Example: ./FunSVPT -f file1,file2,file3,... -maf MAF -m <1/2> -inf <bed/vcf> -outf <bed/vcf>...

Option 2: Use the 6th column of BED file to specify samples
Example: ./FunSVPT -f file -maf MAF -m <1/2> -inf bed -outf <bed/vcf> ...

NOTE: Please make sure you have sufficient memory, at least 3G.

Options:

-maf : should be a number between 0~1
-nc : when using this option, users don't need to install VAT (variant annotation tool)
-exp, -cls, -exf : if used, should be specified together.

5. Input format

3.5.1 User input file (-f): could be either BED format or VCF format.

* BED format

In addition to the three required BED fields, please prepare your file as follows (5 required fields, tab delimited; **the 6th column is reserved for sample names, do not put other information there**): chromosome, start position, end position, reference allele, and alternative allele.

Chromosome - name of the chromosome (e.g. chr3, chrX)
Start position - start position of variants. (0-based)
End position - ending position of variants. (end exclusive)
e.g., chr1 0 100 spanning bases numbered 0-99
Reference allele - reference allele of variants
Alternative allele - alternative allele of variants

* VCF format

The header line names the 8 fixed, mandatory columns. These columns are as follows (tab-delimited): #CHROM POS ID REF ALT QUAL FILTER INFO

* Recurrent analysis input format
Option 1: separate files for each genome (BED or VCF). Use *"-f file1, file2, file3*" separated by comma.
Option 2: put all variants in one file (only for BED format, use the 6th column labeling sample names). Use "*-f file*".

3.5.2 Gene list format (-g): If you are interested in particular set of genes, you can put your genes in one file (one gene per row) and use "*-g file*" to instruct the program to only analyze variants in or linked to those genes. Please use **gene symbols**.

3.5.3 Gene expression format (-exp): Users can also upload gene expression data for the program to detect differentially expressed genes between cancer and benign samples and highlight variants linked to these genes. The gene expression data should be prepared as a matrix with first column stores gene names (use **gene symbols**) and first row as sample names. Other fields are gene expression data either in rpkm or raw read counts. Tab delimited.
e.g.,

Gene	*Sample1*	*Sample2*	*Sample3*	*Sample4*	*...*
A1BG	*1*	*5*	*40*	*0*	*...*
A1CF	*20*	*9*	*0*	*23*	*...*
...	*...*	*...*	*...*	*...*	*...*

3.5.4 Sample class format (-cls): In addition to the expression data, users need to upload annotations of samples as "cancer" or "benign" (only two classes "**cancer**" or "**benign**"). The number of samples in this file should equal to that in expression data. And sample names should match.
e.g.,

Sample1	*benign*
Sample2	*cancer*
Sample3	*cancer*
Sample4	*benign*
...	*...*

6. Output file
FunSVPT will generate four outputs: 1) "*FunSVPT.Output.format*", 2) "*FunSVPT.recur.Summary*", 3) "*FunSVPT.candidates.Summary*" and 4) "*FunSVPT.err*". *FunSVPT.Output.format:* stores detail results from all samples; *FunSVPT.recur.Summary:* the recurrent elements with sample information; *FunSVPT. candidates.Summary:* brief output of potential candidates (coding nonsynonymous/prematurestop mutations, non-coding mutations with score >= 5 and mutations in or linked to known cancer genes); "*FunSVPT.err*" stores the error information. For a single genome, the "*FunSVPT.recur.Summary*" is empty.

When providing gene_expression data, FunSVPT produces two additional files - "*DE.gene.txt*" is the differentially expression genes from RNA-Seq analysis and "*DE.pdf*" is the differential gene expression plot.

3.6.1 Sample BED format output

Header:

chr start end ref alt sample gerp;cds;variant.annotation.cds;network.hub;gene.under.negative.selection;ENCODE.annotated;hot.region;motif.analysis;sensitive;ultra.sensitive;ultra.conserved;target.gene[known_cancer_gene/TF_regulating_known_cancer_gene;differential_expressed_in_cancer;actionable_gene];coding.score;noncoding.score;noncoding.recurrent

Coding variants:

chr1 36205041 36205042 C A PR2832 5.6;Yes;VA=1:CLSPN:ENSG00000092853.8:-:prematureStop:4/5:CLSPN-001:ENST00000251195.5:3999_3232_1078_E->:CLSPN-005:ENST00000318121.3:4020_3232_1078_E->*:CLSPN-003:ENST00000373220.3:3828_3040_1014_E->*:CLSPN-004:ENST00000520551.1:3861_3073_1025_E->*;PPI;Yes;.;.;.;.;.;.;CLSPN;4;.;.*

Non-coding variants:

chr1 48306315 48306316 T G PR1783 4.53;No;.;PPI&REG(TAL1);.;Enhancer(drm|chr1:48305900-48307500),TFP(HDAC2|chr1:48306080-48307045),TFP(HNF4A|chr1:48306228-48306944);.;.;.;.;.;TAL1(DRM)[known_cancer_gene];.;3;.

3.6.2 Sample VCF format output

Header:

##fileformat=VCFv4.0
##INFO=<ID=OTHER,Number=.,Type=String, Description = "Other Information From Original File">
##INFO=<ID=SAMPLE,Number=.,Type=String,Description="Sample id">
##INFO=<ID=CDS,Number=.,Type=String,Description="Coding Variants or not">
##INFO=<ID=VA,Number=.,Type=String,Description="Coding Variant Annotation">
##INFO=<ID=HUB,Number=.,Type=String,Description="Network Hubs, PPI (protein protein interaction network), REG (regulatory network)">
##INFO=<ID=GNEG,Number=.,Type=String,Description="Gene Under Negative Selection">
##INFO=<ID=GERP,Number=.,Type=String,Description="Gerp Score">
##INFO=<ID=NCENC,Number=.,Type=String,Description="NonCoding ENCODE Annotation">
##INFO=<ID=HOT,Number=.,Type=String,Description="Highly Occupied Target Region">
##INFO=<ID=MOTIFBR,Number=.,Type=String,Description="Motif Breaking">
##INFO=<ID=MOTIFG,Number=.,Type=String,Description="Motif Gain">
##INFO=<ID=SEN,Number=.,Type=String,Description="In Sensitive Region">
##INFO=<ID=USEN,Number=.,Type=String,Description="In Ultra-Sensitive Region">
##INFO=<ID=UCONS,Number=.,Type=String,Description="In Ultra-Conserved Region">
##INFO=<ID=GENE,Number=.,Type=String,Description="Target Gene (For coding - directly affected genes ; For non-coding - promoter or distal regulatory module)">
##INFO=<ID=CANG,Number=.,Type=String,Description="Cancer related info [known_cancer_gene/TF_regulating_known_cancer_gene;differential_expressed_in_cancer;actionable_gene]";
##INFO=<ID=CDSS,Number=.,Type=String,Description="FunSVPT Coding Score">
##INFO=<ID=NCDS,Number=.,Type=String,Description="FunSVPT NonCoding Score">
##INFO=<ID=RECUR,Number=.,Type=String,Description="Recurrent elements / variants">
#CHROM POS ID REF ALT QUAL FILTER INFO

Coding variants:

chr1 36205042 . C A . . SAMPLE=PR2832;GERP=5.6;CDS=Yes;VA=1:CLSPN:ENSG00000092853.8:-:prematureStop:4/5:CLSPN-001:ENST00000251195.5:3999_3232_1078_E->:CLSPN-005:ENST00000318121.3:4020_3232_1078_E->*:CLSPN-003:ENST00000373220.3:3828_3040_1014_E->*:CLSPN-004:ENST00000520551.1:3861_3073_1025_E->*;HUB=PPI;GNEG=Yes;GENE=CLSPN;CDSS=4*

Non-coding variants:

chr1 48306316 . T G . .

SAMPLE=PR1783;GERP=4.53;CDS=No;HUB=PPI&REG(TAL1);NCENC=Enhancer(drm|chr1:48305900-48307500),TFP(HDAC2|chr1:48306080-48307045),TFP(HNF4A|chr1:48306228-48306944);GENE=TAL1(DRM);CANG=TAL1(DRM)[known_cancer_gene];NCDS=3

3.6.3 **Output description (VCF format as an example)**

** VA (variants annotation)*

This is the output produced from VAT (variant annotation tool) for coding variations.

** NCENC (Non-coding ENCODE annotation)*

This is formatted as "*category(element_name|chromosome:position)*" (0-based, end exclusive).

TFP - transcription factor binding peak.

TFM - transcription factor bound motifs in peak regions.

DHS - DNase1 hypersensitive sites, with number of cell lines (MCV, total 125 cell lines).

ncRNA - non-coding RNA

Pseudogene

Enhancer - chmm/segway (genome segmentation), drm (distal regulatory module)

** MOTIFBR*

This field is a hash-delimited tag, defined as follows: *TF name # motif name # motif start # motif end # motif strand # mutation position # alternative allele frequency in PFM # reference allele frequency in PFM* . (0-based, end exclusive)

e.g., MOTIFBR=TAF1#TATA_known1_8mer#85913478#85913493#+#3#0.02#0.4

** MOTIFG*

Hash-delimited. *motif name # motif start # motif end # motif strand # mutation position # motif PWM score with alternative allele # motif PWM score with reference allele*

e.g., MOTIFG=HNF4_known6#49357253#49357259#-#1#4.893#0.606

** HOT (highly occupied region)*

If a mutation occurs in HOT regions, the corresponding cell lines (5 in total) are shown.

** CANG (cancer related information)*

This field stores all the cancer related information. Five possible tags:

[known_cancer_gene]: the gene have been annotated as an cancer gene.

[TF_regulating_known_cancer_gene]: the gene is a transcription factor regulating known cancer genes.

[actionable_gene]: the gene is potentially actionable ("druggable").

[up_regulated]: the gene is up-regulated in cancers, if providing RNA-Seq gene expression data.

[down_regulated]: the gene is down-regulated in cancers, if providing RNA-Seq gene expression data.

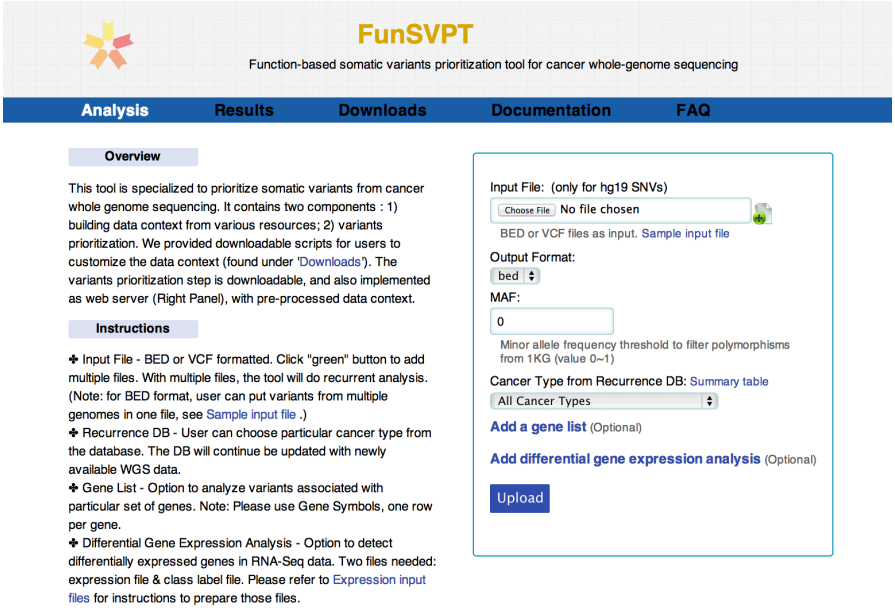
** RECUR (recurrent genes, regulatory elements and mutations)*

If a mutation occurs in recurrent genes or regulatory elements, it is annotated as "*gene/regulatory element name: recurrent samples (mutations in corresponding samples (position is 1-based))*". If it is a recurrent mutation, "*" is tagged.

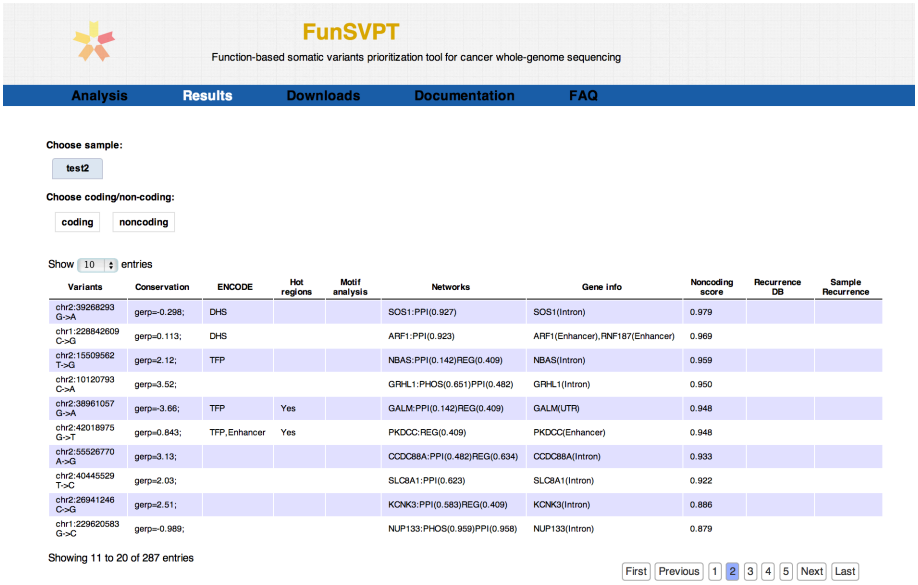
e.g., RECUR=Pseudogene(ENST00000467115.1|chr1:568914-569121):PR1783(chr1:568941,chr1:569004*),PR2832(chr1:569004*)

4. FunSVPT webserver

FunSVPT is implemented as a webserver using django web framework. The screenshot is shown in Supplementary Figure 2. User can download the results or browser them in interactive tables (Supplementary Figure 3).



Supplementary Figure 2. Web interface.



Supplementary Figure 3. Interactive table for results.

References

Alexandrov, L.B.*, et al.* (2013) Signatures of mutational processes in human cancer, *Nature*, **500**, 415-421.

Anders, S. and Huber, W. (2010) Differential expression analysis for sequence count data, *Genome biology*, **11**, R106.

Baca, S.C.*, et al.* (2013) Punctuated evolution of prostate cancer genomes, *Cell*, **153**, 666-677.

Bejerano, G.*, et al.* (2004) Ultraconserved elements in the human genome, *Science*, **304**, 1321-1325.

Berger, M.F.*, et al.* (2011) The genomic complexity of primary human prostate cancer, *Nature*, **470**, 214-220.

Bernstein, B.E.*, et al.* (2010) The NIH Roadmap Epigenomics Mapping Consortium, *Nature biotechnology*, **28**, 1045-1048.

Consortium, E.P.*, et al.* (2012) An integrated encyclopedia of DNA elements in the human genome, *Nature*, **489**, 57-74.

Cooper, G.M.*, et al.* (2005) Distribution and intensity of constraint in mammalian genomic sequence, *Genome research*, **15**, 901-913.

Ernst, J. and Kellis, M. (2012) ChromHMM: automating chromatin-state discovery and characterization, *Nature methods*, **9**, 215-216.

Futreal, P.A.*, et al.* (2004) A census of human cancer genes, *Nature reviews. Cancer*, **4**, 177-183.

Gerstein, M.B.*, et al.* (2012) Architecture of the human regulatory network derived from ENCODE data, *Nature*, **489**, 91-100.

Habegger, L.*, et al.* (2012) VAT: a computational framework to functionally annotate variants in personal genomes within a cloud-computing environment, *Bioinformatics*, **28**, 2267-2269.

Harrow, J.*, et al.* (2012) GENCODE: the reference human genome annotation for The ENCODE Project, *Genome research*, **22**, 1760-1774.

Hoffman, M.M.*, et al.* (2012) Unsupervised pattern discovery in human chromatin structure through genomic segmentation, *Nature methods*, **9**, 473-476.

Hoffman, M.M.*, et al.* (2013) Integrative annotation of chromatin elements from ENCODE data, *Nucleic acids research*, **41**, 827-841.

Khurana, E.*, et al.* (2013) Integrative Annotation of Variants from 1092 Humans: Application to Cancer Genomics, *Science*, **342**, 1235587.

Mu, X.J.*, et al.* (2011) Analysis of genomic variation in non-coding elements using population-scale sequencing data from the 1000 Genomes Project, *Nucleic acids research*, **39**, 7058-7076.

Tarazona, S.*, et al.* (2011) Differential expression in RNA-seq: a matter of depth, *Genome research*, **21**, 2213-2223.

Thurman, R.E.*, et al.* (2012) The accessible chromatin landscape of the human genome, *Nature*, **489**, 75-82.
Touzet, H. and Varre, J.S. (2007) Efficient and accurate P-value computation for Position Weight Matrices, *Algorithms for molecular biology : AMB*, **2**, 15.
Wagle, N.*, et al.* (2012) High-throughput detection of actionable genomic alterations in clinical tumor samples by targeted, massively parallel sequencing, *Cancer discovery*, **2**, 82-93.
Wolfe, M.H.a.D.A. (1973), *John Wiley and Sons*, pages 185–194.
Yip, K.Y.*, et al.* (2012) Classification of human genomic regions based on experimentally determined binding sites of more than 100 transcription-related factors, *Genome biology*, **13**, R48.