Computational and Structural Biotechnology Journal 21 (2023) 3124–3135

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research article

DriverGenePathway: Identifying driver genes and driver pathways in cancer based on MutSigCV and statistical methods

Xiaolu Xu [a,1], Zitong Qi [b,1], Dawei Zhang [a], Meiwei Zhang [c], Yonggong Ren [a,*], Zhaohong Geng [d,*]

[a] School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China
[b] Department of Statistics, University of Washington, Seattle, WA 98195, USA
[c] Center for Reproductive and Genetic Medicine, Dalian Women and Children's Medical Group, Dalian 116037, China
[d] Department of Cardiology, Second Affiliated Hospital of Dalian Medical University, Dalian 116023, China

ARTICLE INFO

Article history:
Received 18 February 2023
Received in revised form 18 May 2023
Accepted 18 May 2023
Available online 26 May 2023

Keywords:
Cancer research
Driver gene
Driver pathway
MutSigCV
Statistical methods

ABSTRACT

Although computational methods for driver gene identification have progressed rapidly, it is far from the goal of obtaining widely recognized driver genes for all cancer types. The driver gene lists predicted by these methods often lack consistency and stability across different studies or datasets. In addition to analytical performance, some tools may require further improvement regarding operability and system compatibility. Here, we developed a user-friendly R package (DriverGenePathway) integrating MutSigCV and statistical methods to identify cancer driver genes and pathways. The theoretical basis of the MutSigCV program is elaborated and integrated into DriverGenePathway, such as mutation categories discovery based on information entropy. Five methods of hypothesis testing, including the beta-binomial test, Fisher combined *p*-value test, likelihood ratio test, convolution test, and projection test, are used to identify the minimal core driver genes. Moreover, de novo methods, which can effectively overcome mutational heterogeneity, are introduced to identify driver pathways. Herein, we describe the computational structure and statistical fundamentals of the DriverGenePathway pipeline and demonstrate its performance using eight types of cancer from TCGA. DriverGenePathway correctly confirms many expected driver genes with high overlap with the Cancer Gene Census list and driver pathways associated with cancer development. The DriverGenePathway R package is freely available on GitHub: https://github.com/bioinformatics-xu/DriverGenePathway.

1. Introduction

Cancer is a severe and global disease. Specifically, accumulated genetic mutations inducing an imbalance between cell differentiation and apoptosis deregulation is the leading cause of cancer. During this process, genes are classified into two categories. Genes with mutations responsible for cancer development and progression are defined as driver genes [1]. Conversely, passenger genes are defined as those with mutations that are coincidentally or subsequently acquired from the driver genes. Adequate identification of driver genes is the key to gene-targeted therapy for cancers. Nevertheless, identifying driver genes directly from the randomly mutated genes is a significant challenge since individuals with the same cancer type often have different diver genes, i.e. mutational heterogeneity. With reliable sequencing genomic data provided by projects such as TCGA (http://cancergenome.nih.gov) and ICGC (http://www.icgc.org), these observed mutational heterogeneities in cancer have motivated the development of driver gene identification tools.

MutSigCV[2] and MuSiC[3] are two classical and well-known frequency methods that have been cited thousands of times and used for performance comparisons by almost all driver gene identification tools[4–7,8,9]. MutSigCV proposed an effective method to overcome the mutational heterogeneity by estimating the background mutation rate (BMR) for each gene-patient-category

* Corresponding authors.
E-mail addresses: ygren@lnnu.edu.cn (Y. Ren), gengzhaohong@dmu.edu.cn (Z. Geng).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.csbj.2023.05.019

combination based on the observed silent mutations in the gene and non-coding mutations in the surrounding genes. MuSiC separated the drivers from the passenger mutations using various statistical methods based on the mapped reads in BAM format, mutation annotation format (MAF), a set of regions of interest, and relevant clinical data. Such recently published methods for driver gene identification also include DriverML[10], PanCancer and PanSoftware analysis[11], and nucleotide context-based method[12]. On the other hand, since driver mutations typically target genes in a few key pathways, several methods examining the combination of mutations have been proposed to overcome the mutational heterogeneity. A class of de novo methods exploited two combination properties, coverage and mutual exclusivity, to effectively distinguish gene sets containing driver mutations (driver pathway)[13–16]. The advantage of de novo methods is that the input data are easily accessible and are not limited by the incompleteness and inaccuracy of the prior knowledge database. However, while most tools for identifying driver genes have their source codes publicly available, many are not user-friendly, and some are only compatible with Linux systems. Moreover, the complexity of the input data poses limitations to the practical application of these tools. Table 1 shows the details of some well-known identification methods in terms of language, supported system, availability of code, availability of package and whether single data input is supported (Single input data).

To popularize and improve the above identification methods, especially MutSigCV, we developed a user-friendly R software package (DriverGenePathway) which includes two main functions, DriverGene and DriverPathway, to identify driver genes and pathways, respectively. Identification of driver genes consists of (i) preprocessing the MAF file, (ii) BMR calculation, and (iii) identifying driver genes based on five methods of hypothesis testing. Identification of driver pathways consists of (i) preprocessing mutation data into (0−1)-mutation matrix(if MAF is input), (ii) identifying driver gene sets using (0,1)-mutation matrix based on adaptively weighted and robust mathematical programming, which is a de novo method, and (iii) evaluating the significance of mutual exclusivity for the identified driver gene sets using a permutation test. The usage of DriverGenePathway is straightforward. With a single command, users can (1) apply MutsigCV more effectively to identify driver genes; (2) identify driver genes using the BMR for each gene-patient-category and other four methods of hypothesis testing including beta-binomial test, Fisher combined *p*-value test, likelihood ratio test, and convolution test; (3) identify significantly driver gene sets (pathways). DriverGenePathway has been shown to be highly effective and advantageous in analyzing mutation data obtained from the TCGA project (Section 3.2).

2. Methods

The statistical methods of DriverGenePathway workflow (Fig. 1) are introduced in the following sections. Additionally, users can easily access a detailed guide by referring to the vignette file.

2.1. Identification of driver genes

2.1.1. Input Data

The input data of the DriverGene function for identifying driver genes include mutation data (mandatory), coverage data, covariates, mutation dictionary, and the reference genome. Specifically, mutation data should be MAF, including columns of the gene, chromosome, start position, end position, variant classification, Tumor_Seq_Allele, and mutation category. It is recommended that users download and input coverage, covariates, mutation dictionary and reference genome applicable to the mutation data for preprocessing. Besides, the default four inputs (if NULL) can be automatically downloaded from MutSigCV, which may take some time. The default covariate (gene expression level, DNA replication time, and HiC compartment) and coverage data can be applied to all cancer types. Reference genome sequence defaulted to hg19 should be chosen according to the DNA sequencing from MAF.

2.1.2. Preprocessing

Preprocessing is an essential step for identifying driver genes. Gene, patient, mutation effect, mutation category, and covariates will be processed to generate a uniform format. Columns “gene” and “Hugo_Symbol” in MAF are unified as “gene”. Columns “patient” and “Tumor_Sample_Barcode” are unified as “patient”. In preprocessing the mutation effect, variant classifications (frameshift insertion, missense mutation, intron silent, nonsense mutation, etc.) are projected to the corresponding mutation effects (silent, nonsilent, noncoding, null) for future mutation category discovery. Missing data in covariates may lead to inaccurate estimates of mutation rates and false positives in the list of identified driver genes. To this end, we employed a cluster-based approach that begins by clustering genes according to their mutation counts across the above four effects. Once genes are clustered based on this criterion, any missing covariate values for a given gene are then estimated by taking the mean covariate value of other genes within the same cluster. Fig. 2 illustrates the preprocessing workflow utilized by DriverGenePathway for analyzing mutation data.

Mutation category discovery is a complicated and crucial step of preprocessing. We performed the mutation category discovery process based on a detailed analysis and summary of the “CATEGORY

Table 1
Summary of well-known methods to identify driver genes and pathways.

Methods	Languages	Supported systems	Availability of code	Availability of package	Single input data
AWRMP[15]	-	-			✓
Dendrix[13]	Python	Windows & Linux & IOS	✓		✓
DiSCaGe[17]	-	-			
DriverML[10]	R & Perl & Shell & C+ +	Linux	✓		✓
GeNWeMME[18]	Python	Windows & Linux & IOS	✓		
Hier. HotNet[19]	Python	Linux	✓		
LOTUS[20]	R	Windows & Linux & IOS	✓		
MoPRO[21]	Python	Windows & Linux & IOS	✓		
MutSigCV[2]	Matlab	Windows & Linux & IOS	✓		✓
MuSiC[3]	R & Perl & Python	Linux	✓	✓	
OMEN[16]	Prolog	Linux	✓		

DISCOVERY" in the MutSigCV program(refer to lines 266 ~ 636 in Matlab code). For each mutation, there are four possible base types (A, T, G, C) in the mutation site (where the mutation occurs), the left site (the left adjacent site to the mutation site), and the right site (the right adjacent site to mutation site). Due to the principle of complementary base pairing, mutations that occur at a given site can be converted to either A or C. the mutation site can be converted to either A or C. Mutation types can be (i) transitions in A's or C's; (ii) transversions in A's or C's; (iii) small insertions/deletions, nonsense and splice site mutations. In other words, there are thousands of possible mutation categories. Only the most valuable mutation categories can help to identify the core driver genes. Methods for selecting mutation categories are determined by mutation data, coverage data, and the usability of the reference genome sequence. Mutation categories discovery for different category numbers (*categ_flag*) and methods are as follows.

1. *categ_flag*: there are four cases for the number of mutation categories as follows:
   (1) *categ_flag* = 0: If the mutation categories in the mutation data and coverage data correspond one-to-one, then Method1 will be applied; otherwise, the program stops.
   (2) *categ_flag* = 1: Method2 will be applied.
   (3) *categ_flag* = 2 ~ 6: If the reference genome file is available, then Method3 will be applied; otherwise, program stops.
   (4) *categ_flag* > 6: Set *categ_flag* = 6 and Method3 will be applied.
   (5) *categ_flag* = NaN (default value): If the reference genome file is available, then set *categ_flag* = 4 and Method3 will be applied; else, if the reference genome file is not available and mutation categories in mutation data and coverage data correspond one-to-one, then Method1 will be used; otherwise, Method2 will be applied.
2. Methods: three Methods corresponding to the number of mutation categories (*categ_flag*) are as follows:
   (1) Method1: mutation category in the mutation data is directly used for identifying driver genes.
   (2) Method2: mutation categories are classified into only "missense" and "null+indel". In this method, the coverage count of "missense" and "null+indel" equals the total mutation count of all 192 categories of the four effects in the coverage data. For the mutation data, mutations with the effect of "null" are set to the "null+indel" category, and the rest are set to the "missense" category.
   (3) Method3: steps of Method3 to select mutation categories are as follows:
      1. The mutation counts that 64 (4 *4 *4) triplet bases in the coding regions mutate to four bases in the coverage data is calculated. The triplet bases are the mutation site, the left neighbour, and the right neighbour. For example, A in C_G means the mutation site is A with left neighbour C and right neighbour G. Then, the count of non-coding mutations in each gene is calculated and divided by three since the mutation site in the middle of each triplet base may mutate to three other bases.
      2. The mutation counts that 64 types of triplet bases mutate to four bases in the mutation data is calculated. The mutation count is set to 0 for the identical base before and after the mutation. For point mutations, the left and right of each mutation site are typically determined by the reference genome being used for comparison.

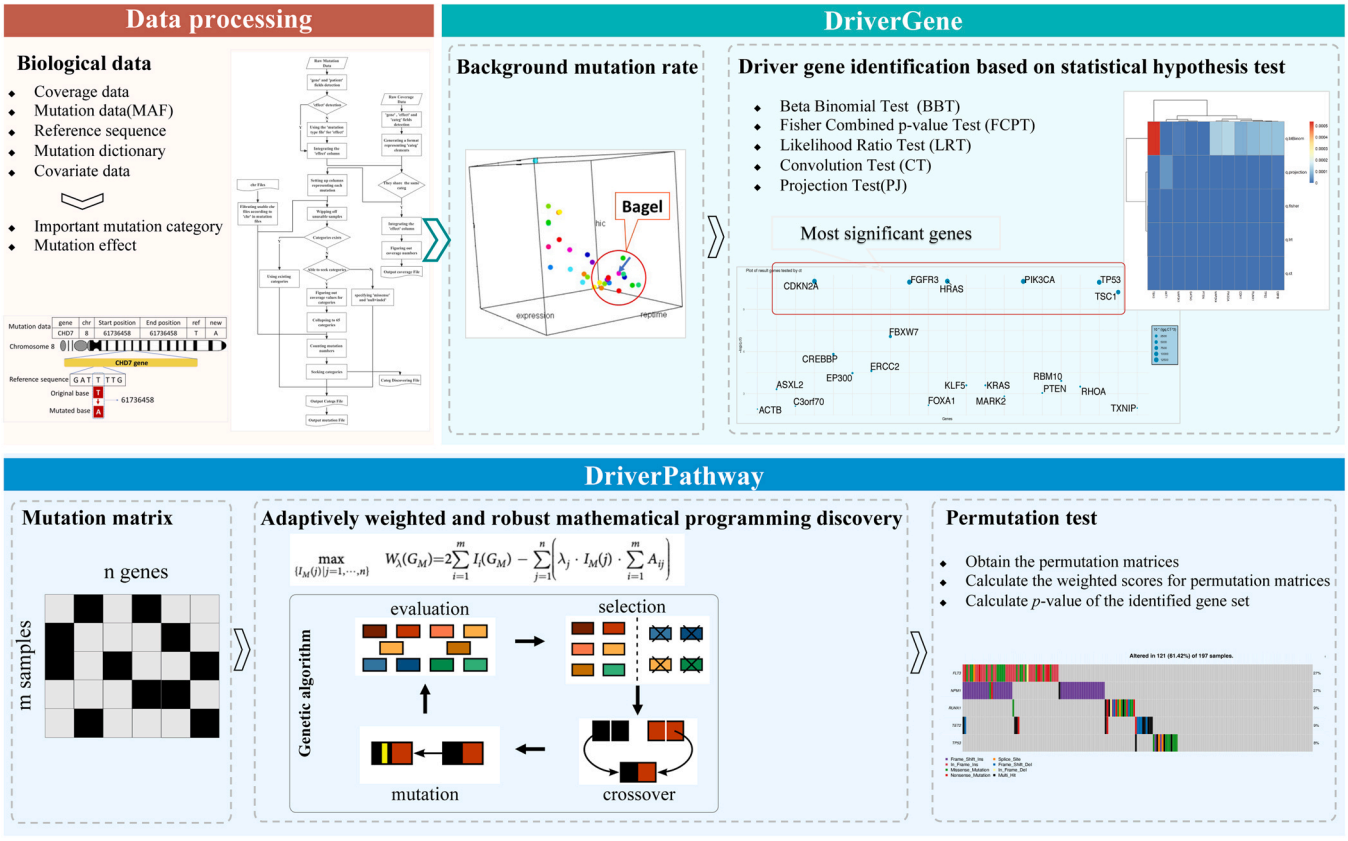

Fig. 1. Workflow of DriverGenePathway package.

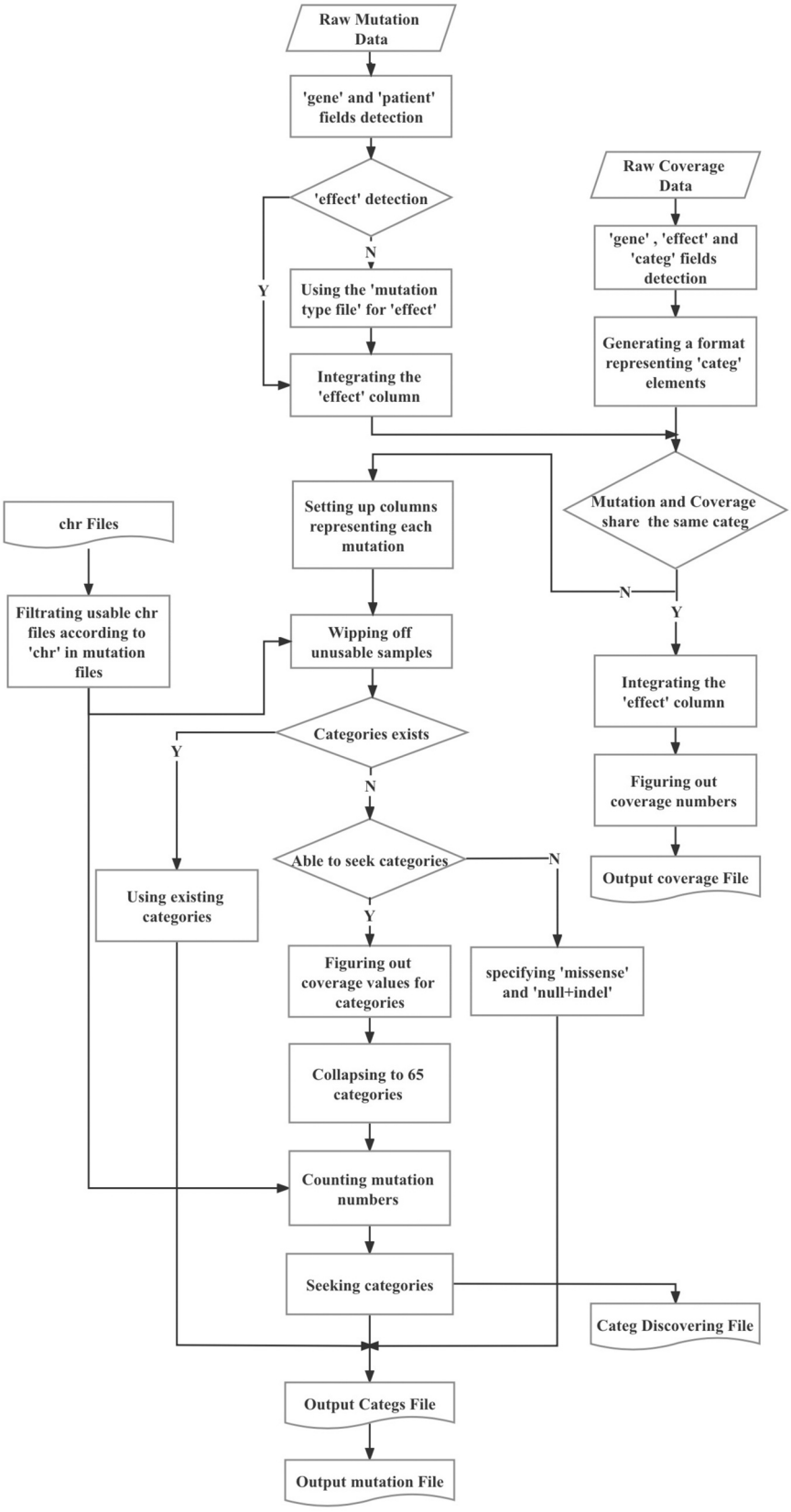

Fig. 2. Workflow of preprocessing.

3. The optimized mutation categories are selected by maximizing the entropy (negative-entropy) calculated based on the coverage count and the mutation count from Step 1 and Step 2. The mutation information entropy of a set of mutation categories with a category number of *ncat* is calculated by

$$\text{Entropy} = \sum_{i=1}^{ncat} \frac{N^i}{N_{tot}} * \left(-\left(\frac{n^i}{N^i} * \log_2 \frac{n^i}{N^i} + \left(1 - \frac{n^i}{N^i}\right) * \log_2 \left(1 - \frac{n^i}{N^i}\right) \right) \right) \quad (1)$$

where N_{tot} is the coverage count of all categories; N^i is the coverage count of ith category; and n^i is the mutation count of ith category. Multiple sets of mutation categories are randomly initialized, and the corresponding information entropy is calculated. The optimized category set is the one with the highest information entropy.

4. Mutations in mutation data are projected into the selected categories in Step 3. Then, the "null+indel" category is added to the mutation category matrix where the coverage count of "null+indel" is the sum of the mutation count in all categories selected in Step 3. The actual mutation count of "null+indel" is the count with the "null" effect in mutation data.
5. Mutations in coverage data are projected into the selected categories, including "null+indel".

2.1.3. Background mutation rate calculation

The calculation of background mutation rate (BMR) is based on the process used in MutSigCV, which involves obtaining both the observed count ($x_{g,c,p}$) and coverage count ($X_{g,c,p}$) of mutations per gene, mutation category, and patient, using the preprocessed mutation data, coverage data, and covariate data. The BMR of each gene is estimated from the gene's silent and noncoding mutations and those of its neighbour genes in the covariate space called the bagel method. For each gene, a bagel of the closest neighbouring genes in the covariate space is built so that all genes in the bagel do not disagree with BMR estimated for the gene. The n_B^{max} genes within the bagel of gene g were similar to gene g in terms of covariates, observed mutation counts, and coverage counts by hypothesis testing. Thus, the total background counts for gene g can be estimated by the background counts in the gene and its bagel.

To improve computational efficiency, we first conducted the binomial test, binomial($n_g^{nonsilent}$, $N_g^{nonsilent}$, bmr), to calculate p-values for all genes. Therein, $n_g^{nonsilent}$ is the observed count of nonsilent mutation of gene g, $N_g^{nonsilent}$ is the count of covered nonsilent sequenced bases in gene g, bmr is the background mutation rate which is a user-defined parameter. Then, q-values are calculated using the Benjamini-Hochberg procedure [22]. Subsequently, genes with q-value≤ 0.05 are selected as candidate genes for the future identification.

2.1.4. Identifying driver genes based on five methods of hypothesis testing

Based on the BMR of each gene, driver genes are identified through five methods of hypothesis testing, including Beta Binomial Test (BBT), Fisher Combined p-value Test (FCPT), Likelihood Ratio Test (LRT), Convolution Test (CT), and Projection Test (PT).

1. Beta Binomial Test
   BBT supposes that mutation parameters N_g, x_g, *and* X_g of gene g follow the beta-binomial distribution. Then, the p-value of gene g is calculated by

$$p_g = 1 - \sum_{k=0}^{n_g^{obs}} f(k|N_g, x_g + 1, X_g + 1) \quad (2)$$

where

$$f(k|N_g, x_g + 1, X_g + 1) = \frac{\Gamma(N_g + 1)\Gamma(k + x_g + 1)\Gamma(N_g - k + X_g - x_g + 1)\Gamma(X_g + 2)}{\Gamma(k + 1)\Gamma(N_g - k + 1)\Gamma(N_g + X_g + 2)\Gamma(x_g + 1)\Gamma(X_g - x_g + 1)}$$

n_g^{obs} is the observed count of nonsilent mutation in gene g, N_g is the count of covered sequenced bases in gene g, x_g is the background mutation count of g, and X_g is the background coverage count of g. $f(k|N_g, x_g + 1, X_g + 1)$ is the normalized probability density function of the beta-binomial distribution such that $\sum_{k=0}^{N_g} f(k|N_g, x_g + 1, X_g + 1) = 1$, and $\Gamma(\cdot)$ is the gamma function.

2. Fisher Combined p-value Test
3. FCPT performs binomial hypothesis testing for different mutation categories to obtain p-values. According to Fisher's method [23],

$$\chi_g = -2 \sum_{i=1}^{n_c} log(p_g^i) \quad (3)$$

where p_g^i is the p-value from hypothesis testing on the ith mutation category of gene g, n_c is the number of mutation categories. The final p-value for the entire gene is calculated as the probability of observing a value no less than χ_g, based on a χ^2 distribution with $2n_c$ degrees of freedom.

4. Likelihood Ratio Test
5. LRT constructs a likelihood ratio-based statistic (χ_g) for a gene, denoted as follows:

$$\chi_g = -2 \sum_{i=1}^{n_c} \log \left(\frac{L\left(n_{g,i}^{obs}, N_{g,i} | \frac{x_{g,i}}{X_{g,i}}\right)}{L\left(n_{g,i}^{obs}, N_{g,i} | \frac{b_{g,i}}{B_{g,i}}\right)} \right) \quad (4)$$

where $n_{g,i}^{obs}$ and $N_{g,i}$ are the observed count and coverage count of mutations in gene g and category i, respectively; $x_{g,i}$ and $X_{g,i}$ are the background mutation count and background coverage count in gene g and category i, respectively; $b_{g,i}$ and $B_{g,i}$ are the sum of the observed count and sum of the coverage count in gene g and category i for nonsilent, noncoding, and silent. $L(\cdot)$ is the likelihood of observed mutation count for the i-th mutation category, defined as the point probability of observing $n_{g,i}^{obs}$ mutations given a coverage count of $N_{g,i}$ and a mutation rate of $\frac{x_{g,i}}{X_{g,i}}$ or $\frac{b_{g,i}}{B_{g,i}}$. The final p-value for the entire gene is calculated as the probability of observing a value no less than χ_g, based on an approximate χ^2 distribution with n_c degrees of freedom.

6. Convolution Test
7. Similar to FCPT and LRT, CT calculates the logarithm (with base 10) of the sum of the single-point binomial probability density of gene g,

$$S_g = -\sum_{i=1}^{n_c} \log \left(L\left(n_{g,i}^{obs}, N_{g,i} | \frac{x_{g,i}}{X_{g,i}}\right) \right) \quad (5)$$

where n_c, $n_{g,i}^{obs}$, $x_{g,i}$, and $X_{g,i}$ are the same as the variables in LRT. Hence, the p-value of gene g is

$$p_g^{s > S_g^{obs}} = \sum_{k \geq S_g^{obs}} \exp(\text{hist}(k)) \quad (6)$$

where

$$S_g^{obs} = -\sum_{i=1}^{n_c} \log \left(\binom{N_{g,i}}{n_{g,i}^{obs}} \left(\frac{x_{g,i}}{X_{g,i}}\right)^{n_{g,i}^{obs}} \left(\frac{1 - x_{g,i}}{X_{g,i}}\right)^{N_{g,i} - n_{g,i}^{obs}} \right)$$

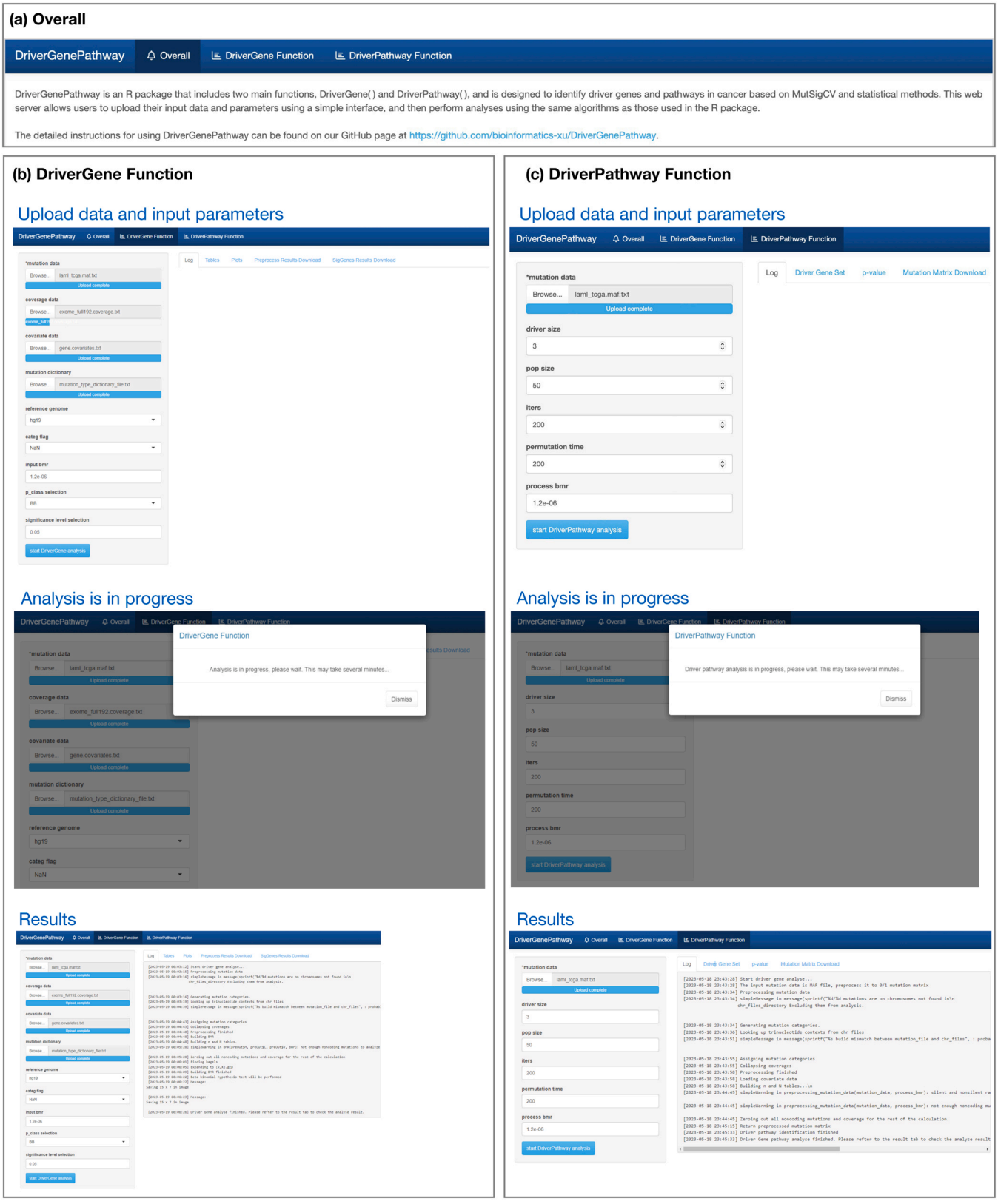

Fig. 3. DriverGenePathway webserver including DriverGene and DriverPathway functions.

hist is the histogram constructed based on convolution [3].

8. Projection Test
9. Projection Test (refer to Section 2.5 on page 27 in the 'SUPPLEMENTARY INFORMATION' of MutSigCV) compares the mutational signal from the observed nonsilent count with the mutational background count estimated above for each gene. The probability that in gene g, category c, and patient p, has zero mutations ($P^{(0)}_{g,c,p}$), one mutation ($P^{(1)}_{g,c,p}$), two or more mutations ($P^{(2+)}_{g,c,p}$) are first cal-

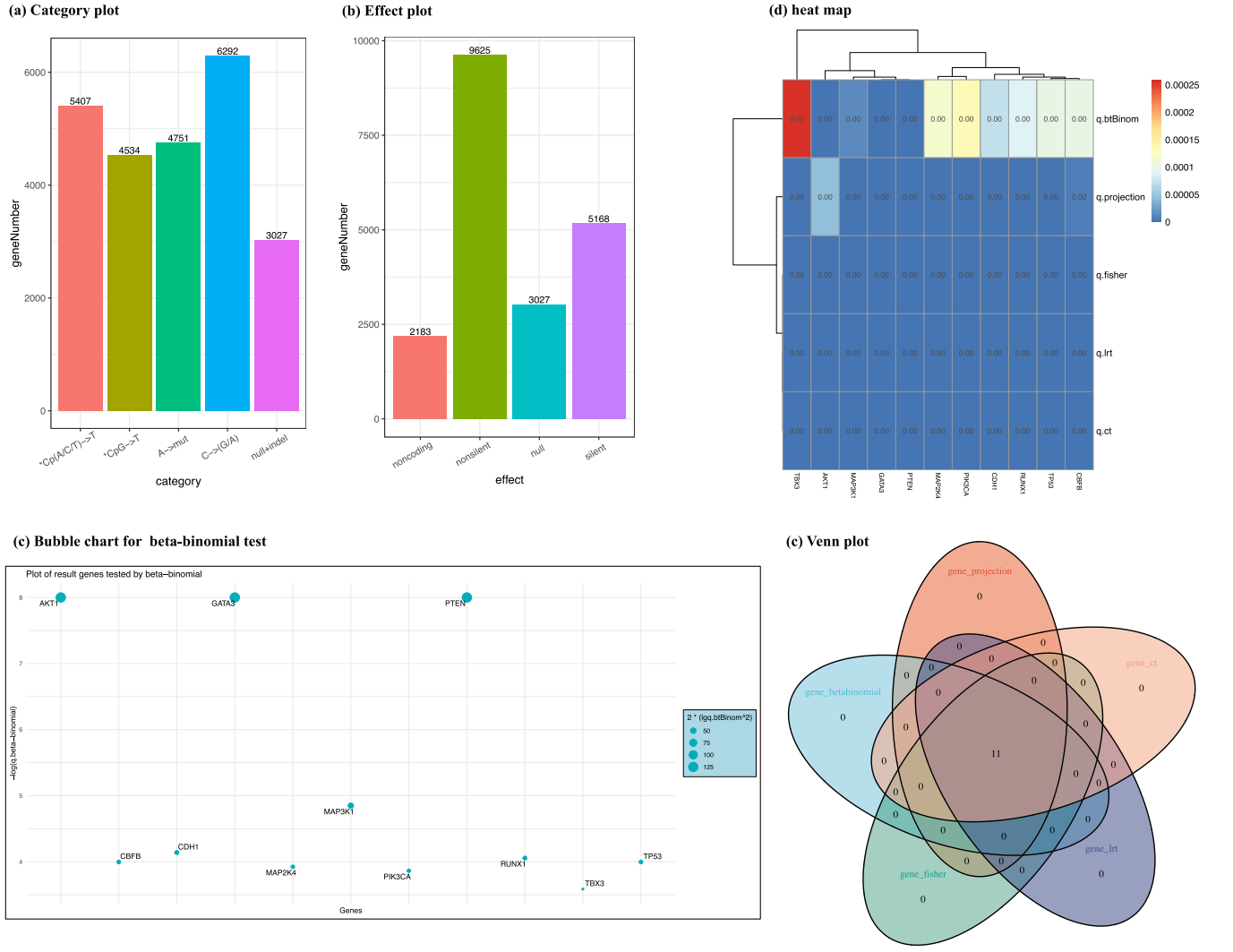

Fig. 4. Output figures of DriverGene function in cancer BRCA. (a) gene number distribution of the selected important categories; (b) gene number distribution of effects including noncoding, nonsilient, null, and silent; (c) heat map on *q*-values of each identified driver gene; (d) Bubble chart for the beta-binomial test; (e) Venn diagram of the overlapping driver genes from five methods of hypothesis testing.

culated by beta-binomial probability mass function. Only the first two mutation categories with the highest priority (d_1,d_2) are considered according to the order of $P^{(1)}$. Each patient is then projected to a two-dimensional space of degrees $D_{g,p} = (d_1, d_2)$, taking into account up to two of its mutations, with the mutations prioritized by two categories with the highest priorities ($d_1 \geq d_2$). In order to compute the distribution of patient degrees expected under the estimated model of background mutation, the probability for each patient to be of each degree by chance ($P_{g,p}^{(d_1,d_2)}$) is calculated based on $P_{g,c,p}^{(0)}$, $P_{g,c,p}^{(1)}$, and ($P_{g,c,p}^{(2+)}$). Furthermore, each degree is also associated with a score S ($S_{g,p}^{(d_1,d_2)}$). By summing the scores associated with each patient's observed degree D, gene g is assigned a total overall score for the observed configuration of patient degrees S_g^{obs}. To determine the probability of obtaining a given score by chance, i.e., from background mutation alone, a null distribution of scores $P_g^{(S=x)}$ is calculated by convolution. Hence, the p-value of gene g is calculated by

$$p_g = 1 - \int_0^{S_g^{obs}} P_g^{(S=x)}\, dx \tag{7}$$

We further applied the Benjamini-Hochberg procedure to the results of the above five methods of hypothesis testing to calculate the false discovery rate, i.e., the q-value. Finally, genes with $q \leq sigThreshold$ will be identified as driver genes. Therein, *sigThreshold* is a parameter of DriverGene function to determine the significance level.

2.2. Identification of driver pathway

2.2.1. Adaptively weighted and robust mathematical programming (AWRMP) for identifying driver pathways

High coverage and mutual exclusivity are two critical biological properties of driver mutations in pathways, which are widely used in driver gene set (pathway) identification[13]. We developed the DriverPathway function based on our previous research AWRMP that adaptively balances the coverage and mutual exclusivity of gene sets using mutation frequencies [15].

The input data for DriverPathway is (0–1)-mutation matrix where rows represent patients and columns represent genes or MAF. For mutation matrix A, if gene j of patient i is mutated, then $A_{ij} = 0$, otherwise $A_{ij} = 1$. The mutation matrix A is defined as the following:

Table 2
Summary of cancer mutation datasets.

Datasets	Project names in cBioPortal	Reference	Number of samples	Number of genes
BLCA	Bladder Urothelial Carcinoma (TCGA, Nature 2014)	[25]	130	13,421
BRCA	Breast Invasive Carcinoma (TCGA, Nature 2012)	[26]	507	13,415
COADREAD	Colorectal Adenocarcinoma (TCGA, Nature 2012)	[24]	224	15,998
COLON	Colorectal Adenocarcinoma (TCGA, Nature 2012)	[24]	155	15,038
LAML	Acute Myeloid Leukemia (TCGA, Firehose Legacy)	–	150	1887
PAAD	Pancreatic Adenocarcinoma (TCGA, Firehose Legacy)	–	186	11,618
RECTUM	Colorectal Adenocarcinoma (TCGA, Nature 2012)	[24]	69	10,092
UCS	Uterine Carcinosarcoma (TCGA, Firehose Legacy)	–	57	7084

Table 3
Genes identified by DriverGene.

Cancer	Genes identified by DriverGene	Accuracy
BLCA	***ARID1A***, ***KDM6A***, ***TP53***, ***CDKN1A***, ***RB1***, ***ELF3***, ***STAG2***, *TXNIP*, ***FBXW7***, ***CDKN2A***, *FOXQ1*, ***PIK3CA***, ***ERCC2***, ***FGFR3***	85.71%
BRCA	***GATA3***, ***MAP2K4***, ***PTEN***, ***TBX3***, ***TP53***, ***PIK3CA***, ***CDH1***, ***RUNX1***, ***MAP3K1***, ***CBFB***, ***AKT1***	100%
COADREAD	***ACVR2A***, *AIM2*, ***APC***, ***FBXW7***, ***KRAS***, ***NRAS***, ***TP53***, *SEC63*, ***TGFBR2***, ***SMAD2***, ***ACVR1B***, *MIER3*, *ARL2BP*, ***B2M***	71.43%
COLON	***ACVR2A***, ***KRAS***, ***TGFBR2***, ***TP53***, ***APC***, ***FBXW7***, ***SMAD4***, *AIM2*, ***NRAS***, *SEC63*, ***SMAD2***, *PSG8*, ***CASP8***	76.92%
LAML	***CEBPA***, ***NRAS***, ***DNMT3A***, ***WT1***, ***IDH1***, ***KRAS***	100%
PAAD	***KRAS***, ***SMAD4***, ***CDKN2A***, ***TP53***, ***RNF43***	100%
RECTUM	***APC***, ***KRAS***, ***TP53***, ***TCF7L2***, ***PIK3CA***, ***FBXW7***	100%
UCS	***TP53***, ***FBXW7***, ***PIK3CA***, ***PTEN***, ***PPP2R1A***, ***PIK3R1***, ***KRAS***, ***RB1***, ***CHD4***, ***ARID1A***, *ZNF814*, *TPTE*	83.33%

The bolded genes are the genes that are also in the CGC list

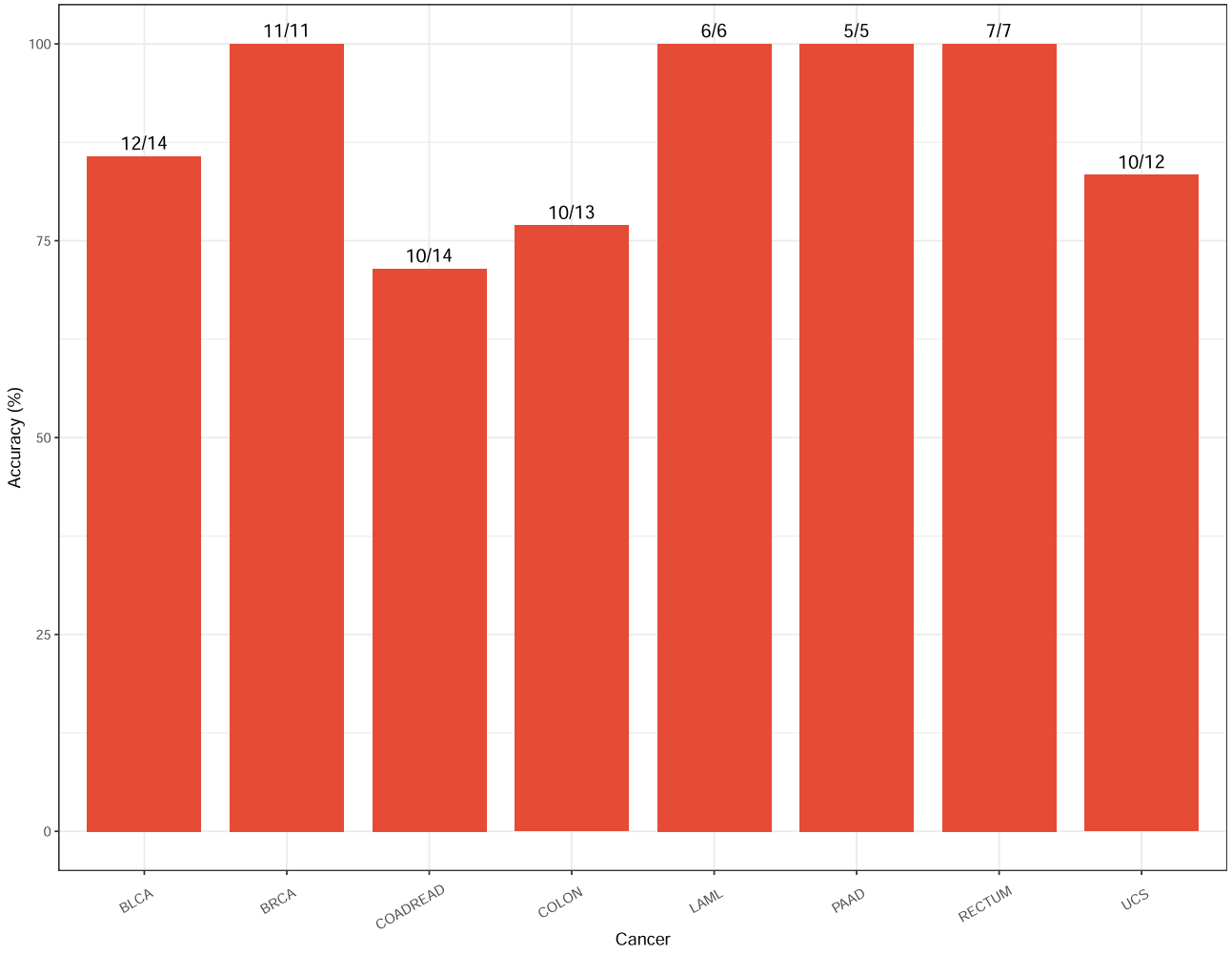

Fig. 5. The proportion of identified driver genes that are included in CGC to all identified driver genes.

$$A_{ij} = \begin{cases} 1 & \text{if gene } j \text{ mutated for patient } i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

If MAF is input, DriverPathway function will preprocess MAF and generate a (0–1)-mutation matrix using coverage data, covariates, mutation dictionary, and the reference genome, which will be downloaded automatically if NULL. Specifically, binomial hypothesis, binomial ($n_g^{nonsilent}$, $N_g^{nonsilent}$, bmr), and Benjamini-Hochberg procedure (refer to Section 2.1.3) will be conducted to select candidate genes in mutation matrix. To ensure efficiency, the number of genes included in the preprocessed mutation matrix is controlled to between 50 and 500 genes, using the significance level (0.001 ~ 0.1) as a guide wherever possible.

2.2.2. Permutation Test

We further utilized the permutation test from Dendrix [13] to test the statistical significance of the identified driver gene set with the following three steps.

1. Obtain the permutation matrices
2. Apply P_n times permutation on the mutation matrix A to get P_n matrices $\{A_t^{'}|t = 1, 2, ...,P_n\}$. Permutation only replaces the sub-matrix containing the identified driver gene set. Specifically, patient IDs with mutations are randomly replaced while ensuring that the number of mutations in each gene of the sub-matrix remains the same.
3. Calculate the weighted scores for permutation matrices
4. P_n weighted scores $\{W_t^{'}|t = 1, 2, ...,P_n\}$ of coverage and mutual exclusivity corresponding to P_n permutation matrices are then calculated by

$$W_\lambda(G_M) \equiv |\Gamma(G_M)| - \omega(M) = 2|\Gamma(G_M)| - \sum_{g \in G_M} |\Gamma(g)| \quad (9)$$

5. Calculate p-value of the identified gene set
6. Denote the score for coverage and mutual exclusivity of the original mutation matrix as W, then p-value of the identified gene set in the permutation test is

$$P \equiv Pr(W_t^{'} \text{ is greater than } W) = \frac{\sum_{t=1}^{P_n} \mathbb{1}(W_t^{'})}{P_n} \quad (10)$$

where $W_t^{'}$ is the score for coverage and mutual exclusivity of the permutation mutation matrix t, and $\mathbb{1}(W_t^{'})$ is defined as

$$\mathbb{1}(W_t^{'}) = \begin{cases} 1 & \text{if } W_t^{'} \text{ is greater than } W \\ 0 & \text{otherwise} \end{cases}$$

2.3. Implementation and Installation

DriverGenePathway R package is ready and published on Github (https://github.com/bioinformatics-xu/DriverGenePathway). Users can install the package using devtools::install_github("bioinformatics-xu/DriverGenePathway"), and identify driver genes with a single line code DriverGene(…) and driver pathway with DriverPathway(…). To enhance the accessibility of DriverGenePathway for users without expertise in R programming, we have also developed the DriverGenePathway web server (see GitHub page for address, Fig. 3). This web server provides a user-friendly interface that enables users to easily upload their data and input parameters and perform analyses using the same algorithms as those used in the R package.

The unique mandatory input of the DriverGene function is Mutation, i.e., mutation data which is a MAF format. Besides, other default parameters are *Coverage* = NULL, *Covariate* = NULL, *MutationDict* = NULL, *chr_files_directory* = NULL, *categ_flag* = NaN (*categ_flag* = 4 and Method3 are adopted in Section 2.1.2), *bmr* = 1.2*e* − 6, *p_class* = allTest, and *sigThreshold* = 0.05. Coverage, Covariate, MutationDict, and chr_files_directory can be downloaded automatically if NULL. The unique mandatory parameter of the DriverPathway function is *mutation_data* which is a (0–1)-mutation matrix or MAF data. The other default parameters are *driver_size* = 3, *pop_size* = 200, *iters* = 1000, *permut_time* = 1000. The specific options and configurations for the parameters can be found in the vignette file. During the analysis, DriverGenePathway saves the preprocessed data and result files in the current working directory for further reference and downstream analysis. DriverGenePathway has been rigorously tested on Windows, iOS, and Linux operating systems, using input data from various types of cancer.

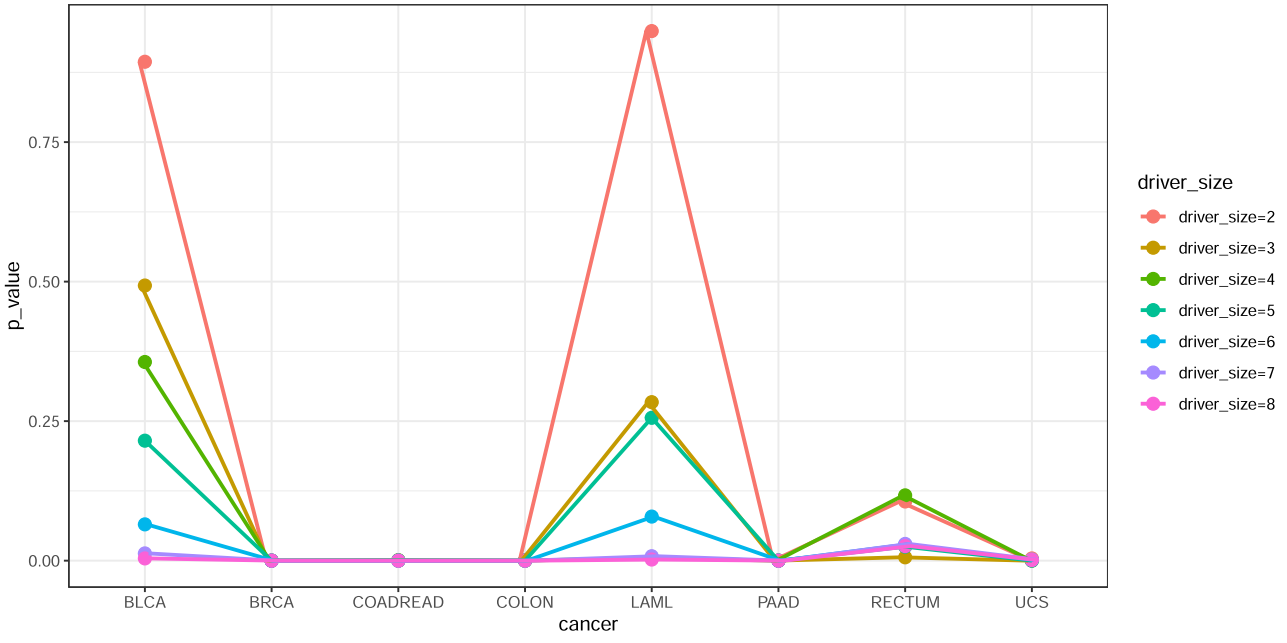

Fig. 6. *p*-values of permutation test for different driver gene sets.

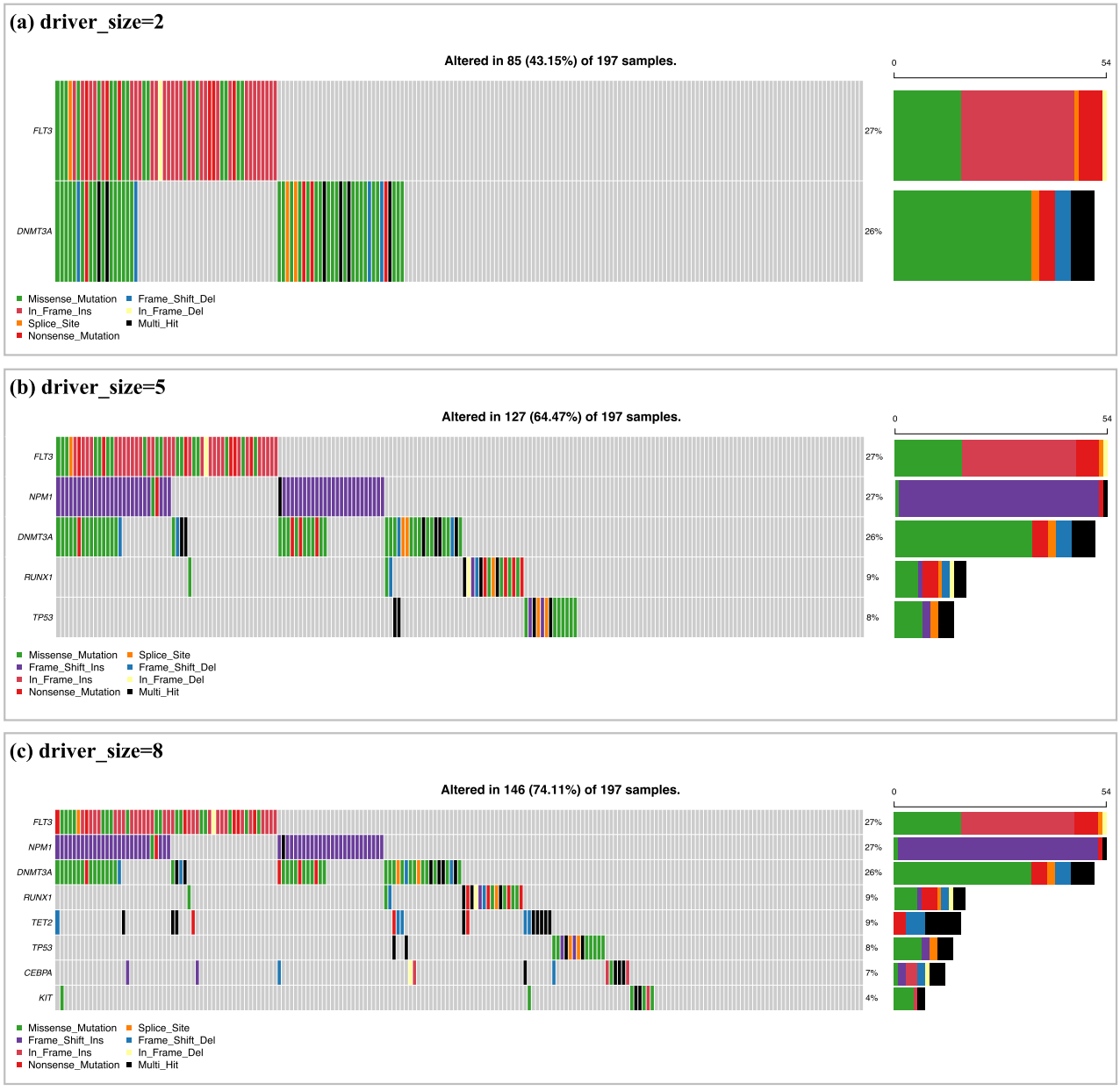

Fig. 7. Waterfall of LAML for dirver_size is 2, 5, and 8.

3. Results

3.1. The description for result files

The preprocessing of the DriverGene function outputs preprocessed mutation information about coverage and categories, including five text files ("MutationCategories.txt" and four other files in the format of the input parameter *output_filestem* + "_covaraite.txt"; *output_filestem* + "_coverage.txt"; *output_filestem* + "_mutations.txt"; *output_filestem* + "_mutcateg_discovery.txt"), and visualization files. These text files include the preprocessed mutation categories, covariate, mutation data, and coverage data corresponding to the selected categories. Visualization files "CategoryPlot.pdf" and "EffectPlot.pdf" in Fig. 4(a)(b) show the mutation number for different categories and effects, respectively. BMR calculation of DriverGene outputs observed mutation count and covered sequence count of each mutation effect as matrices in the console plan. DriverGene function finally outputs the driver gene list file corresponding to the specified hypothesis testing method (named *output_filestem* + *p_class* + "_sigGenes.csv") in *output_filestem* + "_sigGenes" directory. Driver gene list file includes columns of gene name, *p*-value, and *q*_value from the Benjamini-Hochberg procedure. Bubble chart (Fig. 4(c)) shows the identified significant driver genes, where the size of the gene bubble is inversely proportional to *q*-value. For the case of *p_class* = "allTest", *q*-value heatmap (Fig. 4(d)) and Venn diagram (Fig. 4(e)) are generated to show the degree of overlap between the identified driver genes.

DriverPathway function returns a list of the identified driver gene set along with the statistical significance of the permutation test and preprocessed mutation matrix of MAF input. Users can choose to save the preprocessed mutation matrix for further analysis, which can be of great assistance when identifying driver pathways with other *driver_size*.

3.2. Performance evaluation of DriverGenePathway

3.2.1. Datasets

We used the MAF files from TCGA (available at http://www.cbioportal.org) to evaluate the performance of DriverGenePathway, including bladder cancer (BLCA), breast cancer (BRCA), colorectal adenocarcinoma (COADREAD), Colorectal cancer (COLON), acute myeloid leukemia (LAML), Pancreatic adenocarcinoma (PAAD), Rectal cancer (RECTUM), and Uterine carcinosarcoma (UCS). Details of datasets, including project names in cBioPortal, reference, the number of samples, and the number of genes, are shown in Table 2. Therein, COLON, RECTUM, and COADREAD are from the same project [24]. COADREAD is the combined dataset of COLON and RECTUM, comprising 155 colon and 69 rectum samples.

3.2.2. Identification of driver genes

For the performance analysis of DriverGene, parameters *categ_flag*, *bmr*, *output_filestem*, *p_class*, and *sigThreshold* were set as defaults.The minimal core driver genes are defined as the genes that are statistically significant in all five methods of hypothesis testing ($q \leq sigThreshold$). According to the previous studies[27,28], the validity of DriverGene was evaluated by the overlap rate between the identified driver genes and the Cancer Gene Census (CGC, https://cancer.sanger.ac.uk/census/) gene list downloaded on Jan 1, 2023. Therefore, the accuracy of DriverGene is defined as the proportion of genes in the CGC list to all identified genes as follows

$$Acc_i = \frac{|C_i|}{|A_i|}$$

where Acc_i is the accuracy of cancer i, C_i is the set of genes identified by DriverGene in cancer i and also in CGC, A_i is the set of all genes that identified by DriverGene in cancer i, $|\cdot|$ is the cardinality of gene set.

As shown in Table 3 and Fig. 5, the number of driver genes identified by DriverGene in eight types of cancer range from 5 (LAML) to 14 (BLCA and COADREAD), which is in a reasonable range. Accuracy range from 71.43%(COADREAD) to 100% (BRCA, LAML, PAAD, and RECTUM) with a mean value of 89.67%. Furthermore, through a combination of hypothesis testing methods, DriverGene has been successful in identifying critical genes that play an important role in cancer development and progression, including well-known oncogenes and tumor suppressor genes such as *TP53*, *KRAS*, *RB1*, and *APC* [29].

3.2.3. Identification of driver pathways

As described in Section 2.2.1, the mutation matrices used for evaluating the performance of the DriverPathway were derived from preprocessing the input MAF files. Previous epidemiological studies and sequencing data analysis have indicated that a typical tumour generally contains 2–8 driver mutations [30]. In light of this, *driver_size* was set to range from 2 to 8. Besides, parameters *pop_size*, *iters*, and *permut_time* were set as defaults.

Experimental results show that the identified gene sets exhibit a nested nature with increasing values of *driver_size*. The magnitude of *p*-values in the permutation test decrease with the increasing of *driver_size* for the same cancer type (Fig. 6). Specifically, the mutual exclusivity of gene sets with *driver_size* = 2 is less significant in BLCA, LAML, RECTUM (*p*-values ≥ 0.05). (*MUC16*, *TP53*), (*DNMT3A*, *FLT3*), and (*APC*, *PZP*) were identified in the above three cancer types. Since the requirement of AWRMP for mutual exclusivity is inversely proportional to coverage, the coverage of the above three gene sets (88/130, 87/173, 61/69) dominates the optimization process. As *driver_size* increases from 2 to 8, the significance of mutual exclusivity in the driver sets gradually increases, and the *p*-values decrease to below 0.05. For example (Fig. 7), driver gene sets on LAML (driver size = 2, 5, 8) showcases such a relation. As the driver size increases, the coverage of genes gradually increases, and the mutual exclusivity becomes more and more apparent. For *driver_size* = 8, *p*-values for all cancer types are less than 0.05.

Furthermore, DriverPathway has also been successful in identifying gene sets that are enriched in many important signaling pathways. For BRCA, gene set (*AKT1*, *BRCA2*, *CDH1*, *GATA3*, *MAP2K4*, *MAP3K1*, *PIK3CA*, *TP53*) was identified for *driver_size* = 8. Through annotation using DAVID[31], these eight genes were found to be involved in breast cancer(*q*-value = 2.8e-3), pathway in cancer(5.7e-3), and Human T-cell leukemia virus 1 infection(7.4e-4) pathways, which are known to be critical in breast cancer. Therein, (*AKT1*, *BRCA2*, *PIK3CA*, *TP53*), (*AKT1*, *BRCA2*, *CDH1*, *PIK3CA*, *TP53*), and (*AKT1*, *MAP2K4*, *MAP3K1*, *PIK3CA*, *TP53*) act in breast cancer, pathway in cancer and Human T-cell leukemia virus 1 infection respectively.

4. Conclusions

In summary, DriverGenePathway is a user-friendly R package that integrates and improves upon several well-known driver gene identification tools, including MutSigCV (the primary reference), MuSiC, and de novo methods. As demonstrated, the initial filtering of genes that we implemented significantly improves the efficiency of basic MutSigCV. The combined utilization of five methods of hypothesis testing, namely BBT, FCPT, LRT, CT, and PJ, allows the proposed algorithm to identify genes that are critical to cancer development. Through the simultaneous analysis of mutation rate, coverage, mutual exclusivity, and pathway enrichment for identified genes and gene sets, users can gain a comprehensive understanding of the similarities and differences between the two types of methods. DriverGenePathway, featured by multi-system compatibility and accessibility of input data, is expected to highly drive the development of driver gene identification tools and precision medicine for cancer.

Funding

The authors would like to respect and thank all reviewers for their constructive and helpful review. This research is funded by the National Natural Science Foundation of China (No. 61976109), Liaoning Revitalization Talents Program (No. XLYC2006005), Scientific Research Project of Liaoning Province (No. LJKZ0963), Key Research and Development projects of Liaoning Provincial Department of Science and Technology, Liaoning Provincial Key Laboratory Special Fund.

CRediT authorship contribution statement

Xiaolu Xu: Conceptualization, Methodology, Software. **Zitong Qi**: Data curation, Writing – original draft preparation, Software. **Dawei Zhang**: Visualization, Investigation. **Meiwei Zhang**: Supervision. **Yonggong Ren**: Editing and Validation. **Zhaohong Geng**: Writing – Review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Malebary SJ, Khan YD. Evaluating machine learning methodologies for identification of cancer driver genes. Sci Rep 2021;11(1):1–13.
[2] Lawrence MS, Stojanov P, Polak P, Kryukov GV, Cibulskis K, Sivachenko A, Carter SL, Stewart C, Mermel CH, Roberts SA, et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature 2013;499(7457):214–8.

[3] Dees ND, Zhang Q, Kandoth C, Wendl MC, Schierding W, Koboldt DC, Mooney TB, Callaway MB, Dooling D, Mardis ER, et al. MuSiC: identifying mutational significance in cancer genomes. Genome Res. 2012;22(8):1589–98.
[4] Hofree M, Carter H, Kreisberg JF, Bandyopadhyay S, Mischel PS, Friend S, Ideker T. Challenges in identifying cancer genes by analysis of exome sequencing data. Nat Commun 2016;7(1):1–9.
[5] Braun DA, Hou Y, Bakouny Z, Ficial M, Sant'Angelo M, Forman J, et al. Interplay of somatic alterations and immune infiltration modulates response to pd-1 blockade in advanced clear cell renal cell carcinoma. Nat Med 2020;26(6):909–18.
[6] Chan-Seng-Yue M, Kim JC, Wilson GW, Ng K, Figueroa EF, O'Kane GM, et al. Transcription phenotypes of pancreatic cancer are driven by genomic events during tumor evolution. Nat Genet 2020;52(2):231–40.
[7] Chao JY-c, Chang H-C, Jiang J-K, Yang C-Y, Chen F-H, Lai Y-L, Lin W-J, Li C-Y, Wang S-C, Yang M-H, et al. Using bioinformatics approaches to investigate driver genes and identify bcl7a as a prognostic gene in colorectal cancer. Comput Struct Biotechnol J 2021;19:3922–9.
[8] Wang T, Ruan S, Zhao X, Shi X, Teng H, Zhong J, et al. OncoVar: an integrated database and analysis platform for oncogenic driver variants in cancers. Nucleic Acids Res 2021;49(D1):D1289–301.
[9] Guo L, Li S, Yan X, Shen L, Xia D, Xiong Y, Dou Y, Mi L, Ren Y, Xiang Y, et al. A comprehensive multi-omics analysis reveals molecular features associated with cancer via rna cross-talks in the notch signaling pathway. Comput Struct Biotechnol J 2022;20:3972–85.
[10] Han Y, Yang J, Qian X, Cheng W-C, Liu S-H, Hua X, et al. DriverML: a machine learning algorithm for identifying driver genes in cancer sequencing studies. Nucleic Acids Res 2019;47(8):e45.
[11] Bailey MH, Tokheim C, Porta-Pardo E, Sengupta S, Bertrand D, Weerasinghe A, Colaprico A, Wendl MC, Kim J, Reardon B, et al. Comprehensive characterization of cancer driver genes and mutations. Cell 2018;173(2):371–85.
[12] Dietlein F, Weghorn D, Taylor-Weiner A, Richters A, Reardon B, Liu D, Lander ES, Van Allen EM, Sunyaev SR. Identification of cancer driver genes based on nucleotide context. Nat Genet 2020;52(2):208–18.
[13] Vandin F, Upfal E, Raphael BJ. De novo discovery of mutated driver pathways in cancer. Genome Res 2012;22(2):375–85.
[14] Leiserson MD, Wu H-T, Vandin F, Raphael BJ. Comet: a statistical approach to identify combinations of mutually exclusive alterations in cancer. Genome Biol 2015;16(1):1–20.
[15] Xu X, Qin P, Gu H, Wang J, Wang Y. Adaptively weighted and robust mathematical programming for the discovery of driver gene sets in cancers. Sci Rep 2019;9(1):1–12.
[16] Van Daele D, Weytjens B, De Raedt L, Marchal K. OMEN: network-based driver gene identification using mutual exclusivity. Bioinformatics 2022.
[17] Cutigi JF, Evangelista AF, Reis RM, Simao A. A computational approach for the discovery of significant cancer genes by weighted mutation and asymmetric spreading strength in networks. Sci Rep 2021;11(1):23551.
[18] J.F. Cutigi, A.F. Evangelista, A. Simao, GeNWeMME: a network-based computational method for prioritizing groups of significant related genes in cancer, in: Advances in Bioinformatics and Computational Biology: 12th Brazilian Symposium on Bioinformatics, BSB 2019, Fortaleza, Brazil, October 7–10, 2019, Revised Selected Papers 12, Springer, 2020, pp. 29–40.
[19] Reyna MA, Leiserson MD, Raphael BJ. Hierarchical HotNet: identifying hierarchies of altered subnetworks. Bioinformatics 2018;34(17):i972–80.
[20] Collier O, Stoven V, Vert J-P. LOTUS: A single-and multitask machine learning algorithm for the prediction of cancer driver genes. PLoS Comput Biol 2019;15(9):e1007381.
[21] Gumpinger AC, Lage K, Horn H, Borgwardt K. Prediction of cancer driver genes through network-based moment propagation of mutation scores. Bioinformatics 2020;36(Supplement_1):i508–15.
[22] Martin BD, Witten D, Willis AD. Modeling microbial abundances and dysbiosis with beta-binomial regression. Ann Appl Stat 2020;14(1):94.
[23] Zeng P, Dai J, Jin S, Zhou X. Aggregating multiple expression prediction models improves the power of transcriptome-wide association studies. Hum Mol Genet 2021;30(10):939–51.
[24] D.M. Muzny, M.N. Bainbridge, K. Chang, H.H. Dinh, J.A. Drummond, G. Fowler, C. L. Kovar, L.R. Lewis, M.B. Morgan, I.F. Newsham, et al., Comprehensive molecular characterization of human colon and rectal cancer (2012).
[25] John NW, Rehan A, Bradley MB, Wenyi W, Roeland GWV, David M, Seth L, Margaret M, Chad JC, Carolyn S. Comprehensive molecular characterization of urothelial bladder carcinoma. Nature 2014;507(7492):315.
[26] Daniel CK, Robert SF, Michael DM, Heather S, Joelle K-V, Joshua FM, Lucinda LF, David JD, Li D, Elaine RM, Wilson RK, Ding L, Mardis ER. Comprehensive molecular portraits of human breast tumours. Nature 2012;490(7418):61–70.
[27] Gu H, Xu X, Qin P, Wang J. Fl-net: identification of cancer driver genes by using functional impact prediction neural network. Front Genet 2020;11:564839.
[28] Martínez-Jiménez F, Muiños F, Sentís I, Deu-Pons J, Reyes-Salazar I, Arnedo-Pac C, Mularoni L, Pich O, Bonet J, Kranas H, et al. A compendium of mutational cancer driver genes. Nat Rev Cancer 2020;20(10):555–72.
[29] van Riet J, van de Werken HJ, Cuppen E, Eskens FA, Tesselaar M, van Veenendaal LM, Klümpen H-J, Dercksen MW, Valk GD, Lolkema MP, et al. The genomic landscape of 85 advanced neuroendocrine neoplasms reveals subtype-heterogeneity and potential therapeutic targets. Nat Commun 2021;12(1):4612.
[30] Vogelstein B, Papadopoulos N, Velculescu VE, Zhou S, Diaz Jr. LA, Kinzler KW. Cancer genome landscapes. science 2013;339(6127):1546–58.
[31] Dennis G, Sherman BT, Hosack DA, Yang J, Gao W, Lane HC, et al. DAVID: database for annotation, visualization, and integrated discovery. Genome Biol 2003;4(9):1–11.