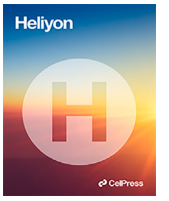
Heliyon 10 (2024) e33582

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

DriverDetector: An R package providing multiple statistical methods for cancer driver genes detection and tools for downstream analysis

Zeyuan Wang [a], Hong Gu [a], Pan Qin [a,*], Jia Wang [b,*]

[a] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Lingshui Street, Dalian, 116024, Liaoning, China
[b] Department of Breast Surgery, Institute of Breast Disease, Second Hospital of Dalian Medical University, Zhongshan Road, Dalian, 116023, Liaoning, China

ARTICLE INFO

Dataset link: https://tcga-data.nci.nih.gov/tcga/

Dataset link: ftp://ftp.broadinstitute.org/pub/genepattern/example_files/MutSigCV_1.3/

Dataset link: https://github.com/FrancisWang96/DriverDetector

Keywords:
Cancer driver genes
Genome analysis software
Background mutation rate

ABSTRACT

Identifying driver genes in cancer is a difficult task because of the heterogeneity of cancer as well as the complex interactions among genes. As sequencing data become more readily available, there is a growing need for detecting cancer driver genes based on statistical and mathematical modeling methods. Currently, plenty of driver gene identification algorithms have been published, but they fail to achieve consistent results. In order to obtain gene sets with high confidence, we present DriverDetector, an R package providing a convenient workflow for cancer driver genes detection and downstream analysis. We develop the background mutation rate calculating module based on the distance between genes in covariate space and binomial test, followed by the driver gene selection module which integrates 11 methods, including two already recognized approaches, a de novo method, and five variants of Fisher's method which are applied to driver gene identification for the first time. Through verification on 12 TCGA datasets, each method is able to identify a set of confirmed driver genes while the number of resulting genes vary significantly across different methods. For robust driver genes detection, a voting strategy based on 10 of the statistical methods is further applied. Results show that the collective prediction based on the voting strategy demonstrates superiority in achieving the consistency of prediction while ensuring a reasonable number of predicted genes and confirmed drivers. By comparing the results of each cancer dataset, we also find that sample size has a huge impact on the number of predicted genes. For downstream analysis, DriverDetector automatically generates plenty of plots and tables to elaborate the results. We propose DriverDetector as a user-friendly tool promoting early diagnosis of cancer and the development of targeted drugs.

1. Introduction

Next generation sequencing (NGS) technologies are facilitating to achieve the human genomic data. Consequently, large-scale databases, like The Cancer Genome Atlas (TCGA) and International Cancer Genome Consortium (ICGC), occur to contribute cancer genomic data to worldwide researchers by the Mutation Annotation Format (MAF) files [1,2]. These advances ensure massive amounts of cancer genomics data for the investigation of cancer driver genes directly responsible for promoting tumorigenesis [3]. There are

* Corresponding authors.
E-mail addresses: qp112cn@dlut.edu.cn (P. Qin), wangjia77@hotmail.com (J. Wang).

https://doi.org/10.1016/j.heliyon.2024.e33582
Received 30 January 2024; Received in revised form 17 June 2024; Accepted 18 June 2024

Available online 1 July 2024

generally two types of computing methods for searching driver gene sets based on somatic mutation data. The first type of methods identifies recurrent mutations whether at the same genes in different patients based on the background mutation rates of genes and statistical hypothesis tests [4–13]. The representative of this type of methods is MutsigCV [8], which uses the 2D-projection method to calculate the probability that the observed mutation frequency of a given gene is greater than the expectation obtained by the nonfunctional background mutation rate. Other ratio-metric approaches focus on the functional impact bias, clustering patterns or composition patterns of mutations [6,12]. The second type of methods based on somatic mutation data identifies a collection of genes in a batch based on high coverage and mutual exclusivity, which are combinatorial mutation patterns in signaling and regulatory pathways [14–17]. In addition, newly presented methods based on multi-omics data or machine learning are gradually becoming research hotspots [11,18–20]. The limitations of existing driver gene identifying methods mainly exist in two aspects. First, the total number of predicted genes cannot be effectively adjusted. According to [6], the number of driver genes predicted by 8 methods (q-value$\leq$0.1) on the same pan-cancer dataset ranges from 158 to 2,600, indicating a huge variation exists across methods. A more recent research [18] tested 21 methods and reached the same conclusion. Since the main purpose is to explore potential driver genes, a gold standard is lacking for this task [6]. One approach to benchmark driver prediction is to measure the overlap with the Cancer Gene Census (CGC) [21], which is a list containing already confirmed and suspected driver genes. To ensure credibility, the ideal result should include enough confirmed driver genes as well as new potential drivers. The second limitation is the consistency of prediction, which is an important criterion for measuring robustness. Specifically, when the dataset is randomly divided by samples into two subsets, an ideal method would achieve the same gene set for each subset [6]. Therefore, the main goal is to find a balance among the predicted gene number, the overlap with known drivers, and the consistency of prediction. To our best knowledge, few methods have managed to fully achieve the balance of results [6,18,19,22]. Besides, as cancer datasets continue to expand, flexible and easy-to-use systematic frameworks are being increasingly required.

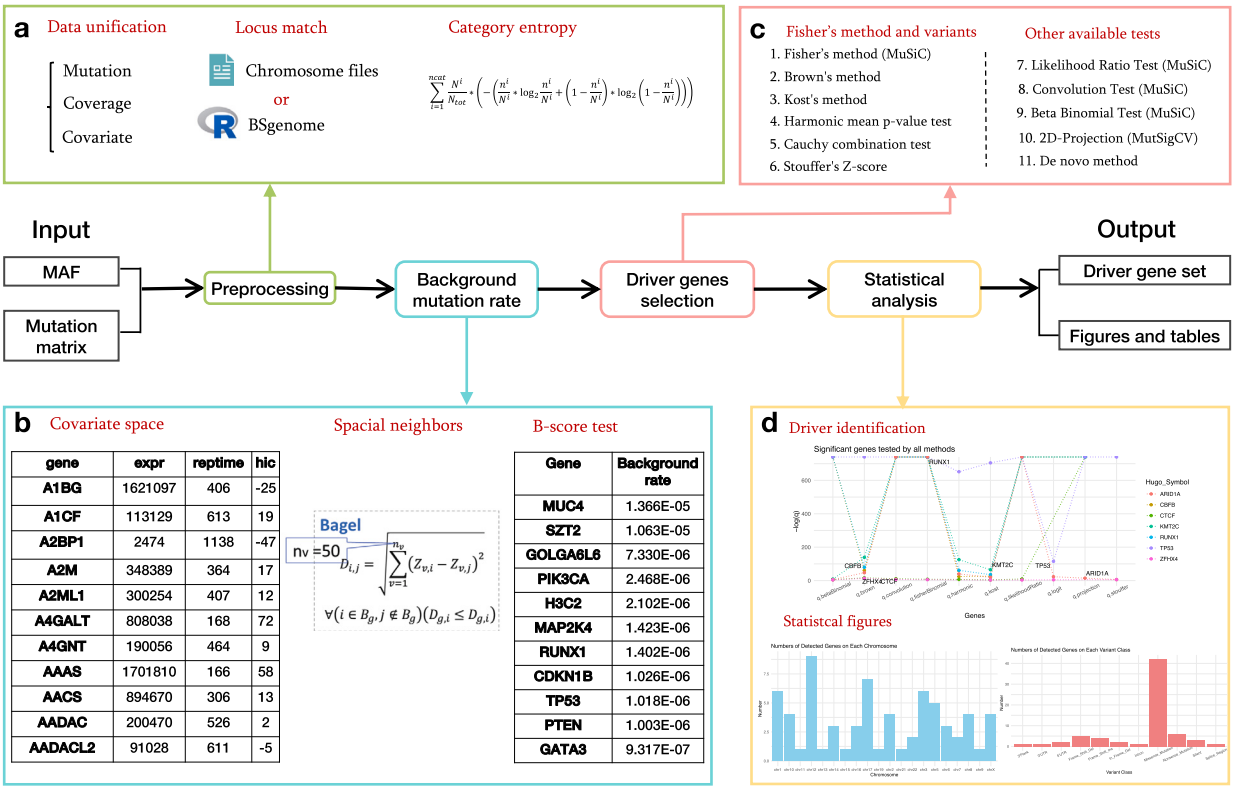
In this work, we present an R package called DriverDetector, which is a powerful toolbox for robust identification of cancer driver genes. The workflow is shown in (**Fig. 1**). Our research is mainly inspired by MuSiC [12], where we find different hypothesis tests using the same significance threshold lead to considerable differences in results. The main hypothesis is that it is hard for methods based on single hypothesis test or gene functionalities to fully distinguish driver genes from the complex background. To this end, we introduce a voting strategy based on 10 statistical methods, which can significantly increase the consistency of results and remove most unlikely genes. In addition, we incorporate a de novo method to identify gene sets with high coverage and mutual exclusivity. DriverDetector is a user-friendly toolbox, for which both MAF and binary mutation matrix can be the input. The mutation data first go through the preprocessing module which washes out trivial genes by using the maximum entropy method and assigning mutation effects and categories (**Fig. 1a**). Next, the background mutation rate for each gene is calculated based on its location in the covariate space. A B-score test is processed to obtain genes with high driving possibilities (**Fig. 1b**). Then, the candidate genes enter the voting system by multiple statistical methods which can be used separately as well (**Fig. 1c**). The significant genes identified by most methods are then go through the statistical analysis module, where the p-values by different methods and the distribution of mutation categories or chromosomes are further analyzed (**Fig. 1d**). We test DriverDetector on 12 type-specific cancer datasets from TCGA, results show that DriverDetector can not only identify a large number of known driver genes but also discover potential driver genes with high confidence. We further evaluate the CGC overlap and consistency for each statistical method, results show that by applying the voting strategy, the consistency of predicted genes is significantly improved. Finally, we compare DriverDetector with existing methods based on different principles on the BRCA dataset. It can be concluded that DriverDetector is an effective tool for driver genes identification. Our proposed software is available at https://github.com/FrancisWang96/DriverDetector.

2. Materials and methods

2.1. Data input and preprocessing

DriverDetector is an easy-to-use R package which requires three materials as input. The first is the mutation data, which can either be MAF or a binary-valued mutation matrix. The second input is the coverage data, which contains universal coverage information for all cancer types by default. The covariate data covering genetic indicators, such as gene expression level, replication time, and chromosomal status, are also necessary for the calculation of background mutation rates. A reference genome list is optional for discovering mutation categories, which can either be a folder path or the BSgenome format [23]. Incorrect input may cause the program to fail or affect the results. DriverDetector automatically checks and unifies each input before further calculation. The preprocessing module first checks that all the required variables are present in each data. For mutation data, the columns should include Hugo_Symbol or gene, Tumor_Sample_Barcode, Chromosome, Start_Position, End_Position, Variant_Classification, and Tumor_Seq_Allele. For coverage data, the required columns are Hugo_Symbol or gene, effect, category, and coverage. The covariate data should include genes and the values of corresponding covariates. Next, the intersection of genes in all data is extracted, and the rows containing out-of-intersection genes are eliminated.

Mutation category discovery is a crucial preprocessing step to determine which mutation categories are most influential for searching driver genes. For each mutation, considering the triplex base group of the mutation site, there are a total of $\left(\binom{4}{1}+\binom{4}{2}+\binom{4}{3}+\binom{4}{4}\right)\times 4\times\left(\binom{4}{1}+\binom{4}{2}+\binom{4}{3}+\binom{4}{4}\right)\times 3=2700$ possibilities. Therefore, only by selecting the most valuable mutation categories and searching for driver genes by the number of mutations in these categories can we find genes that play a significant role in the development and progression of cancer. According to the value of the category_num parameter, which should be set to non-negative integers no greater than 6, the following method is applied. When category_num is set to 0, the program checks for a one-to-one correspondence between mutation categories in the input mutation data and those in the coverage data. If they match,



gene	expr	reptime	hic
A1BG	1621097	406	-25
A1CF	113129	613	19
A2BP1	2474	1138	-47
A2M	348389	364	17
A2ML1	300254	407	12
A4GALT	808038	168	72
A4GNT	190056	464	9
AAAS	1701810	166	58
AACS	894670	306	13
AADAC	200470	526	2
AADACL2	91028	611	-5

Gene	Background rate
MUC4	1.366E-05
SZT2	1.063E-05
GOLGA6L6	7.330E-06
PIK3CA	2.468E-06
H3C2	2.102E-06
MAP2K4	1.423E-06
RUNX1	1.402E-06
CDKN1B	1.026E-06
TP53	1.018E-06
PTEN	1.003E-06
GATA3	9.317E-07

Fig. 1. Workflow of DriverDetector. (a) Preprocessing first ensures the data is available and gene names are appropriate. Then trivial genes are washed out using the maximum entropy method. Based on the reference genome file, the triplets of mutation sites are found in order to generate mutation effects and categories. (b) The background mutation rate is calculated by considering not only the gene itself, but its neighbors in the covariate space as well. (c) Using 11 methods, the significance of genes is determined by their q-values. (d) Statistical analysis for driver identification and multiple variables.

then these categories are directly accepted. Otherwise, the mutations are categorized into "Missense" and "null+indel" based on mutation effects. For coverage data, the corresponding "missense" and "null+indel" coverage numbers for each mutation effect are both equal to the sum of the coverage numbers for all mutation categories. When the parameters are set in the range of 1 to 6, the mutation categories are identified using a reference genome based on the following five steps.

Step 1: Count the number of triple base mutations in all coding regions in the coverage data. Since there are four types of bases (A,T,G,C), the total number of triplets is 4 (left site)×4 (mutation site)×4 (right site) =64. The number of noncoding mutations in each gene is then summed and divided by three due to the mutations in the middle of each triplet can mutate to three other bases.

Step 2: Count the number of all 64 triple base mutations in the mutation data. For each point mutation, the left and right adjacent sites are obtained according to the reference genome. Then the number of mutations of every triplet is calculated. (if the site bases before and after mutation are the same, the number of mutations is 0).

Step 3: Based on the mutation number and coverage number obtained above, the optimal mutation category is found by maximizing the mutation information entropy (negative entropy). The information entropy of a group of mutations, including n members, is calculated as **Eq. (1)**:

$$\mathrm{IE} = \sum_{i=1}^{n} \frac{N^i}{N_{\mathrm{tot}}} * \left(-\left(\frac{n^i}{N^i} * \log_2 \frac{n^i}{N^i} + \left(1 - \frac{n^i}{N^i}\right) * \log_2 \left(1 - \frac{n^i}{N^i}\right)\right)\right) \tag{1}$$

where N_{tot} is the coverage number of all categories, N^i is the coverage number of category i, and n^i is the mutation number of category i.

In order to search mutation categories, multiple groups of mutation categories are first randomly initialized according to a certain rule, then the mutation category information entropies of all groups are calculated respectively. The one with the largest information entropy should be selected as the optimal category.

Step 4: According to the results of step 3, each mutation is assigned to the corresponding categories. Furthermore, a "null+indel" category is appended, whose mutation number is determined by the count of mutations with the effect "null" in the mutation data. The "null+indel" coverage number is the sum of all point mutations in the categories determined by **Step 1-3**.

Step 5: Identify the category for each mutation in the coverage data based on the obtained categories from **Step 1-4** (including the "null+indel" category).

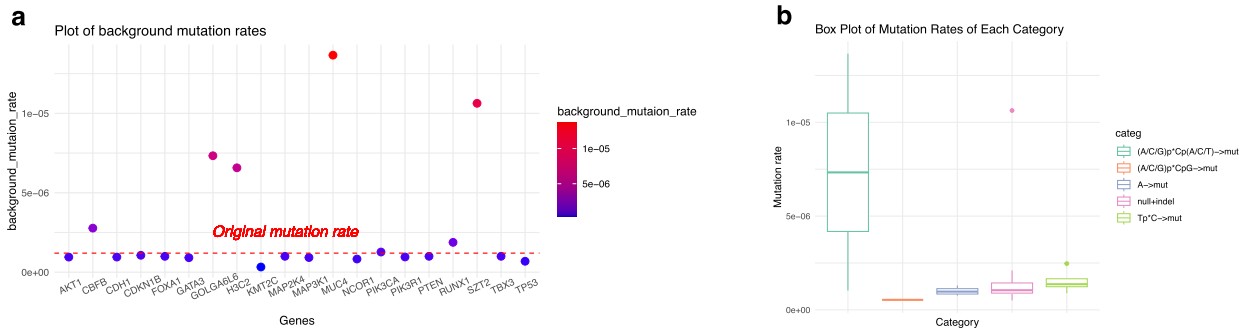


Fig. 2. Outputs of the background mutation rate calculation module run on the BRCA dataset. (a) Background mutation rates of candidate genes selected by B_{score}. (b) Box plot of background mutation rates of each mutation category.

When the category searching finishes, the program outputs the preprocessed results as txt files, including the resulting categories, categorized mutation and coverage data, and the processed covariate data.

2.2. Calculation of background mutation rates

The background mutation rates are calculated using the preprocessed mutation and coverage data, as well as the gene covariate data, which is used to calculate the expected mutation and coverage numbers for each gene, category, and patient. Using the neighborhood construction method based on gene covariates [8], a background mutation rate is calculated for each gene. The default covariates (including gene expression level, replication time and chromosome status) data are from the MutsigCV website and can also be customized by users. To accelerate the process, the default initial background mutation rate is set to 1.2e-6 [24] for all genes.

Binomial test is widely used in driver gene predicting methods [6,8–11] for evaluating whether the mutation rates are statistically significant higher than the background rates. In DriverDetector, the p-value of the binomial test, namely B_{score}, is calculated for each gene by **Eq. (2)**:

$$B_{\text{score}} = \sum_{k=n+1}^{N} \binom{N}{k} p^k (1-p)^{N-k} \tag{2}$$

where p is the mutation rate, k is the number of observed mutation of a given type at a particular nucleotide, n is the number of base pairs in the gene transcript, and N is a total number of cancer samples in a cohort. The threshold of B_{score} is set to 0.05 by default, and the genes which meet the threshold are selected as candidates for subsequent tests.

In this module, the output files include a text file of candidate genes with their background mutation rates, a plot showing the values of background mutation rates and the original mutation rate (**Fig. 2a**), and a box plot presenting the mutation rates of each mutation category (**Fig. 2b**).

2.3. Collective identification of significant genes

In this module, we integrate 11 methods for driver genes identification and each of them can be used separately. By default, all methods are run on the mutation data after category assignment and background mutation rate calculation. In DriverDetector, a voting strategy is applied where the genes collectively predicted by multiple methods are selected. According to previous studies, the results vary hugely by different methods [6,18]. Therefore, collective prediction is able to enhance the robustness of the results. The specific methods are as follows. Based on the previous progress achieved by Fisher's method in driver prediction [6,12], we first integrate Fisher's method as an independent approach:

Fisher's Method

According to [25], the test statistic of Fisher's Method is calculated by **Eq. (3)**

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

where p_g^i is the p-value obtained by testing the binomial distribution hypothesis for the ith mutation category of gene g. n_c is the number of mutation categories. χ_g follows a χ^2 distribution with degrees of freedom $2n_c$. Therefore, the final p-value of gene g is calculated by **Eq. (4)**:

$$p_g = 1 - \int_0^{\chi_g^{obs}} \chi^2_{2n_c}(t)dt \tag{4}$$

$$\chi_g^{obs} = -2 \cdot \sum_{i=1}^{n_c} \log\left(1 - \sum_{n=0}^{n_{g,c}^{obs}} \binom{N_{g,i}}{n} \left(x_{g,i}/X_{g,i}\right)^n \left(1 - x_{g,i}/X_{g,i}\right)^{N_{g,i}-n}\right)$$

where n_c is the total number of mutation categories, $n_{g,i}^{obs}$ is the actual number of mutation in category i of gene g, and $N_{g,i}$ is the coverage number of category i of gene g.

When facing large gene mutation data, Fisher's method has some limitations such as ignoring the internal correlation of data and high computational complexity [26,27]. Such limitations may lead to a negative impact on the results. To this end, a group of extensions of Fisher's method, namely, empirical Brown's method, Kost's method, harmonic mean p-value method, Cauchy's method, and Stouffer's method, are integrated in DriverDetector to improve the robustness of results:

Empirical Brown's Method

As an extension to Fisher's method, Brown's method [28] uses a re-scaled χ^2 distribution as **Eq. (5)**:

$$\chi_g \sim c\chi_{2f}^2 \tag{5}$$

where c is the scale factor and f denotes a re-scaled number of degrees of freedom. Brown calculated these constants by **Eq. (6)**:

$$f = \frac{\mathrm{E}[\chi_g]^2}{\mathrm{var}[\chi_g]} \quad \text{and} \quad c = \frac{\mathrm{var}[\chi_g]}{2\mathrm{E}[\chi_g]} = \frac{n_c}{f} \tag{6}$$

Poole et al. [26] proposed an adaptation of Brown's method using the empirical cumulative distribution function derived directly from the data which can efficiently be applied to large intra-correlated biological datasets. The method aims to calculate the covariance empirically base on **Eq. (7)**:

$$\mathrm{var}[\chi_g] = 4n_c + 2\sum_{i<j} \mathrm{cov}\left(\vec{w}_i, \vec{w}_j\right) \tag{7}$$

where $\vec{w}_i$ and $\vec{w}_j$ denotes the right-sided empirical cumulative distribution function calculated from the sample $\vec{x}_i$ and $\vec{x}_j$.

Kost's Method

Considering Brown's method calculates the expected value and variance of χ_g directly via numerical integration to obtain the covariance, for large datasets, due to computational complexity, numerical integration is not feasible. Kost and McDermott [27] fit a third-order polynomial to approximate this covariance by **Eq. (8)**:

$$\mathrm{cov}\left(-2\log P_i, -2\log P_j\right) \approx 3.263\rho_{ij} + 0.710\rho_{ij}^2 + 0.027\rho_{ij}^3 \tag{8}$$

where P_i and P_j denote the p-values, ρ_{ij} is the correlation between random variables X_i and X_j. The combined p-value is given by **Eq. (9)**:

$$P_{\text{combined}} = 1.0 - \Phi_{2f}(\psi/c) \tag{9}$$

where $\psi = -2\sum_{i=1}^{n_c} \log P_i$ and Φ_{2f} is the cumulative distribution function of χ_{2f}^2

Harmonic Mean p-value Method

The harmonic mean p-value (HMP) [29] is a statistical technique similar to Fisher's method in certain aspects that they both test whether groups of p-values are statistically significant. However, unlike Fisher's method, HMP avoids the restrictive assumption that the p-values are independent. The weighted harmonic mean of p-values $p_1, \ldots, p_L$ is defined as **Eq. (10)**:

$$\overset{\circ}{p} = \frac{\sum_{i=1}^{n_c} w_i}{\sum_{i=1}^{n_c} w_i/p_i} \tag{10}$$

where $w_1, \ldots, w_{n_c}$ are weights that sum to one, i.e. $\sum_{i=1}^{n_c} w_i = 1$. Generalized central limit theorem shows that an asymptotically exact p-value $p_{\overset{\circ}{p}}$ can be calculated using **Eq. (11)**:

$$p_{\overset{\circ}{p}} = \int_{1/\overset{\circ}{p}}^{\infty} f_{\text{Landau}}\left(x \mid \log n_c + 0.874, \frac{\pi}{2}\right) \mathrm{d}x \tag{11}$$

where f_{Landau} denotes the Landau distribution, whose density function can be written as **Eq. (12)**:

$$f_{\text{Landau}}(x \mid \mu, \sigma) = \frac{1}{\pi\sigma} \int_0^{\infty} \mathrm{e}^{-t\frac{(x-\mu)}{\sigma} - \frac{2}{\pi}t\log t} \sin(2t)\mathrm{d}t \tag{12}$$

Cauchy's Method

Cauchy's method [30] is a variant of Fisher's method which uses a tan transformation to obtain a test statistic whose tail is asymptotic to that of a Cauchy distribution under the null. The test statistic can be written as **Eq. (13)**:

$$\chi_g = \sum_{i=1}^{n_c} \omega_i \tan\left[\left(0.5 - p_i\right)\pi\right] \tag{13}$$

where $w_1, \ldots, w_{n_c}$ are weights that sum to one. Under the null, p_i are uniformly distributed, therefore $\tan\left[\left(0.5 - p_i\right)\pi\right]$ are Cauchy distributed. Let W denote a standard Cauchy random variable as **Eq. (14)**:

$$\lim_{t\to\infty} \frac{P[\chi_g > t]}{P[W > t]} = 1 \tag{14}$$

leads to a combined hypothesis test, in which χ_g is compared to the quantiles of the Cauchy distribution.

Stouffer's Method

Stouffer's method [31] serves as a compromise between Fisher's method which is sensitive to the smallest p-value and Pearson's method which is sensitive to the largest p-value. Letting $p_1, p_2, \ldots, p_{n_c}$ denote the individual (one- or two-sided) p-values of the k hypothesis tests to be combined, the test statistic is then computed with $z = \sum_{i=1}^{n_c} z_i/\sqrt{k}$ where $z_i = \Phi^{-1}(1 - p_i)$ and $\Phi^{-1}(\cdot)$ denotes the inverse of the cumulative distribution function of a standard normal distribution. Under the joint null hypothesis, the test statistic follows a standard normal distribution which is used to compute the combined p-value. Stouffer's method assumes that the p-values to be combined are independent. If this is not the case, the method can either be conservative (not reject often enough) or liberal (reject too often), depending on the dependence structure among the tests. In this case, one can adjust the method to account for such dependence.

Next, we incorporate four methods from two influential articles [8] and [12] to increase confidence in the resulting gene set. Specifically, the beta binomial test, likelihood ratio test and convolution test are from MuSiC, while the 2D-projection method is ported from MutSigCV.

Beta Binomial Test

Assume the number of mutations of gene g follows a beta binomial distribution with parameters N_g, x_g, x_g. The p-value of g is calculated as **Eq. (15)**:

$$\begin{aligned} p_g &= 1 - \sum_{k=0}^{n_g^{obs}} f\left(k \mid N_g, x_g + 1, X_g + 1\right) \\ &= 1 - \sum_{k=0}^{n_g^{obs}} \frac{\Gamma\left(N_g + 1\right)\cdot\Gamma\left(k + x_g + 1\right)\cdot\Gamma\left(N_g - k + X_g - x_g + 1\right)\cdot\Gamma\left(x_g + 2\right)}{\Gamma(k+1)\Gamma\left(N_g - k + 1\right)\cdot\Gamma\left(N_g + X_g + 2\right)\cdot\Gamma\left(x_g + 1\right)\Gamma\left(x_g - x_g + 1\right)} \end{aligned} \tag{15}$$

where n_g^{obs} is the number of non-silent mutations actually observed of the gene, N_g is the number of non-silent coverage of g, x_g is the number of background mutations, x_g is the number of background coverage. $f\left(k \mid N_g, x_g + 1, X_g + 1\right)$ is the standardized beta binomial probability density function with $\sum_{k=0}^{n_g^N} f\left(k \mid N_g, x_g + 1, X_g + 1\right) = 1$, and Γ is the gamma function.

Likelihood Ratio Test

A likelihood ratio statistic is set up for each gene by **Eq. (16)**:

$$\chi_g = -2\sum_{i=1}^{n_c} \log\left(\frac{L\left(n_{g,i}^{obs}, N_{g,i} \mid x_{g,i}/X_{g,i}\right)}{L\left(n_{g,i}^{obs}, N_{g,i} \mid b_{g,i}/B_{g,i}\right)}\right) \tag{16}$$

where $n_{g,i}^{obs}$ is the actual mutation number of the ith mutation category of gene g and $N_{g,i}$ is the coverage number of category i of gene g. x_g^i is the number of background mutations in the ith mutation category of gene g and X_g^i is the number of background coverage in the ith category of gene g. b_g^i is the sum of the non-silent, noncoding and silent actual mutation number of the ith mutation category of gene g, and B_g^i is the sum of the non-silent, noncoding and silent coverage number of the ith category of gene g. L is the probability density function of the binomial distribution. χ_g follows a chi-square distribution with degrees of freedom n_c. Therefore, the final p-value of gene g is calculated by **Eq. (17)**:

$$\begin{aligned} p_g &= 1 - \int_0^{\chi_g^{obs}} \chi_{n_c}^2(t)dt \\ \chi_g^{obs} &= -2\cdot\sum_{i=1}^{n_c} \log\left(\frac{\binom{N_{g,i}}{n_{g,i}^{ob,i}}\left(x_{g,i}/X_{g,i}\right)^{n_{g,i}^{obs}}\left(1 - x_{g,i}/X_{g,i}\right)^{N_{g,i} - n_{g,i}^{obs}}}{\binom{N_{g,i}}{n_{g,i}^{ob}}\left(r_{g,i}/R_{g,i}\right)^{n_{g,i}^{obs}}\left(1 - r_{g,i}/R_{g,i}\right)^{N_{g,i} - n_{g,i}^{obs}}}\right) \end{aligned} \tag{17}$$

Convolution Test

Similar to Fisher's Combined p-value Test and Likelihood Ratio Test, Convolution Test calculates the logarithm base 10 of the sum of the probability densities of the single-point binomial distribution for all mutation categories for each gene g by **Eq. (18)**:

$$S_g = -\sum_{i=1}^{n_c} \log\left(L\left(n_{g,i}^{obs}, N_{g,i} \mid x_{g,i}/X_{g,i}\right)\right) \tag{18}$$

where n_c, $n_{g,i}^{obs}$, $N_{g,i}$, $x_{g,i}$, and $X_{g,i}$ are defined in the same way as above. The final p-value of each gene is calculated by **Eq. (19)**:

$$P_g^{\left(s>S_g^{obs}\right)} = \sum_{k=S_g^{obs}}^{S_{bin}^{\max}} \exp(\mathrm{hist}(k)) \tag{19}$$

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

where hist() is the histogram function constructed based on convolution [32].

2D-Projection Method

The 2D-projection method searches for driver genes by mapping each patient into a two-dimensional space, which is proposed by MutsigCV. First, the probability that the mutation of gene g in mutation category c and patient p occurs only once is calculated based on the beta binomial distribution. According to the probability value, only the first two mutation categories with the highest priority ($d1$, $d2$) are considered. Then, an S-score is calculated by the mutation distribution of each gene and each patient according to $d1$ and $d2$. Furthermore, the background distribution and the observed value S_g^{obs} of the S-score of each gene g is calculated. The final p-value of gene g is calculated by **Eq. (20)**:

$$p_g = 1 - \int_0^{S_g^{obs}} P_g^{(S=x)} dx \tag{20}$$

We also include a method based on high coverage and mutual exclusivity for driver genes identification, which is called the de novo method.

De Novo Method

De novo method selects genes with mutation frequencies greater than the threshold and builds the mutation matrix which relies on a binary mutation matrix, of which the rows represent patients and the columns represent genes. In addition to taking a mutation matrix as input, it is also allowed to input a MAF file with a threshold for gene mutation frequency. The aim is to find a sub-matrix G_M of k genes, which maximizes the following function **Eq. (21)**:

$$W_\lambda\left(G_M\right) \equiv \left|\Gamma\left(G_M\right)\right| - \omega(M) = 2\left|\Gamma\left(G_M\right)\right| - \sum_{g \in G_M} |\Gamma(g)| \tag{21}$$

where G_M denotes the mutation matrix composed of elements in the driver gene set. $\left|\Gamma\left(G_M\right)\right|$ is the measure of coverage, $\Gamma(G_M) \equiv \cup_{g \in G_M} \Gamma(g)$ is the set of patients with mutations in the gene set corresponding to M. $\omega(M)$ represents repeat coverages, whose opposite measures the exclusivity of the gene set. $\Gamma(g) \equiv \{i : A_{ig} = 1\}$ refers to the set of patients with gene g mutations.

In our previous research, an algorithm called AWRMP [15] was proposed which improves the above optimization model by adding different weights to each gene in the mutation matrix to obtain the following optimization objective function **Eq. (22)**:

$$\begin{aligned} & W_\lambda\left(G_M\right) \equiv \left|\Gamma\left(G_M\right)\right| - \omega_\lambda(M) \\ & = \sum_{i=1}^{m} I_i\left(G_M\right) - \left(\sum_{j=1}^{n}\left(\lambda_j \cdot I_M(j) \cdot \sum_{i=1}^{m} A_{ij}\right) - \sum_{i=1}^{m} I_i\left(G_M\right)\right) \\ & = 2\sum_{i=1}^{m} I_i\left(G_M\right) - \left(\sum_{j=1}^{n}\left(\lambda_j \cdot I_M(j) \cdot \sum_{i=1}^{m} A_{ij}\right)\right) \\ & \text{s.t.} \begin{cases} I_i\left(G_M\right) \le \left(\sum_{j=1}^{n} A_{ij} \cdot I_M(j)\right) & \text{for } i = 1, \cdots, m; j = 1, \cdots, n \\ \sum_{j=1}^{n} I_M(j) = k \end{cases} \end{aligned} \tag{22}$$

where,

$$I_M(j) \equiv \begin{cases} 1 & j \in G_M \\ 0 & \text{otherwise} \end{cases}$$

$$I_i\left(G_M\right) \equiv \begin{cases} 1 & \text{genes in } G_M \text{ mutate in patient } i \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda_j \equiv \begin{cases} \dfrac{\exp(-|\Gamma(j)|)}{\sum_{r\in G_M}\exp(-|\Gamma(r)|)} & j \in G_M \\ 0 & \text{otherwise} \end{cases}$$

An adaptive weight λ_j is introduced to balance the coverage and exclusivity of gene sets. For genes with high mutation frequency, the condition of exclusivity is not strictly required, making the result more natural. The introduction of λ_j changes the original binary linear programming model into a nonlinear model. Therefore, the objective function is optimized based on the genetic algorithm.

The robustness of the algorithm is guaranteed by adopting the leave-one-out sampling strategy, which assigns the driver gene set a high sampling rate corresponding to the input data. For an input matrix with m patients and n genes, the subsampling process runs the genetic algorithm m times to search the driver gene set. First, m sub matrices $A_{i-}(i = 1, 2, \cdots, m)$ are generated by the leave-one-out strategy, where A_{i-} is the sub matrix obtained by removing the ith row of input matrix A. Next, for each sub matrix, the genetic algorithm is operated to search the corresponding driver gene set, obtaining m gene sets $\{G_k | k = 1, 2, \cdots, m\}$. Thus, the probability (subsampling rate) for a gene set G_k to be selected as a driver gene set is defined as **Eq. (23)**:

$$SSR_{G_k} \equiv \Pr\left(G_k \text{ is selected}\right) = \frac{m_k}{m} \tag{23}$$

where m_k is the sum of times G_k is selected after m runs of the genetic algorithm. Similarly, the subsampling rate of each gene is defined as **Eq. (24)**:

$$SSR_g \equiv P_r(g \text{ is selected in the driver gene set}) = \frac{\sum_{i=1}^{m} I_i(g)}{m} \tag{24}$$

where,

$$I_i(g) \equiv \begin{cases} 1 & \text{gene } g \text{ is selected in the } i\text{th subsampling} \\ 0 & \text{otherwise} \end{cases}$$

Based on the statistical results, the parsimonious gene set is obtained by **Eq. (25)**:

$$\text{Parsimonious set} \equiv \{g | SSR_g = 1\} \tag{25}$$

For all hypothesis tests, a false discovery rate (q-value) for each gene is calculated by the Benjamini-Hochberg method. Comparing to p-values, q-values are more effective in identifying important genes from numerous trivial ones [33,8,22]. The significance threshold for gene q-values is set to 0.1.

2.4. Statistical analysis based on searching results

In this module, the genes with highest possibilities to cause cancer are screened out followed by outputting a series of analysis figures. First, the significant genes with their and q-values obtained by each driver identifying method are output as csv files. In order to show more intuitively, a scatter plot for each result is generated, of which the horizontal axis represents gene names and the vertical axis represents the negative logarithm of the q-values (**Fig. 4a**). Next, the genes voted by most methods are further selected to increase robustness. In particular, all statistical methods except the de novo method are used for voting, since the number of genes in a set is chosen manually for the de novo method. Then, for resulting genes, a box plot showing the distribution of q-values for each gene (**Fig. 4b**) and a scatter plot showing the q-values of each method (**Fig. 4c**) are output to reveal the consistency of both gene significance and methods.

Then, based on the information in MAF, we construct a statistical function to analyze the distributions of drivers on certain variables. Specifically, the numbers of detected genes on each chromosome, variant class, variant type, and nucleotide mutation type are shown by bar plots (**Fig. 4d-g**).

2.5. Application and implementation

The DriverDetector R package runs on mainstream operating systems such as Windows, macOS, and Ubuntu. The input formats of MAF, coverage, covariate and mutation matrix can be txt or Rdata. The reference genome such as hg19 and hg38 can either be a folder path or the BSgenome format. The user guide is available in the package vignette. DriverDetector is available at https://github.com/FrancisWang96/DriverDetector.

3. Results

3.1. Analysis of predicted genes by DriverDetector on TCGA datasets

To evaluate the effectiveness of our proposed R package, we run DriverDetector on MAF files of multiple cancers from TCGA, namely, breast cancer (BRCA), cervical cancer (CESC), esophageal cancer (ESCA), glioblastoma (GBM), kidney clear cell carcinoma (KIRC), liver cancer (LIHC), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), pancreatic cancer (PAAD), rectal

cancer (READ), stomach cancer (STAD), and uterine carcinosarcoma (UCS). We first examine the number of predicted genes and the overlap with CGC genes (**Fig. S1,S2**). According to **Table S2** and **S3**, the number of genes derived from 10 individual methods ($q \leq 0.1$) ranges from 1 to 1384 across all datasets, and the overlap with CGC genes ranges from 0.08 to 1, which illustrates the significant differences in results of different methods. By applying the voting mechanism based on 10 methods, we find that the number of genes and the CGC overlap can both be well controlled. In **Table S4** and **S5**, the number of genes collectively identified by at least 1 to 10 methods and the CGC overlaps are shown. As the votes start increasing, the number of predicted genes immediately decreases to a reasonable level (<1000), while the overlap with CGC gradually increases (**Fig. 3a**). We also find that the CGC overlap slightly decreases for BRCA, CESC, and LUSC when the required number of votes is 8. Thus, we recommend to set the minimum number of votes to 7, which can lead to more resulting genes while ensuring prediction confidence. The genes predicted by at least 7 methods for 12 cancers are shown in **Table S1**, it can be seen that the number of predicted genes is less than 100 for all datasets, and an average CGC overlap of 0.67 is achieved. To further illustrate the effectiveness of DriverDetector, the genes predicted by at least 7 methods on the BRCA dataset are shown as an example. The resulting genes are *CASP8, CBFB, CDH1, CTCF, KMT2C, MAP3K1, PIK3CA, RUNX1, TBX3, TP53, ARID1A, NCOR1, NF1,* and *ZFHX4*. All of them except *ZFHX4* are known drivers in CGC. Although *ZFHX4* has not been formally certified as a cancer driver, there are biological evidences showing that *ZFHX4* has a potential oncogenic function [34,35]. Taking the CESC results as another example, according to **Table S1**, the newly identified drivers for CESC are *FLG, ADGRV* and *HLA-B*. Clinical results show evidence that the mutations of *FLG* and *HLA-B* are risk factors of cancer [36–38]. The oncogenic mechanism of *ADGRV* needs to be further confirmed through clinical studies.

3.2. Consistency evaluation for statistical methods

To evaluate the consistency of each method, we implement five trials for each dataset. In each trial, the dataset is randomly divided into two subsets by samples, and the predicted genes based on each half are compared. The consistency is calculated by 2× (number of intersection)/(sum of number of predicted genes by each method). The average consistencies of five trials are shown in **Table S6**. In **Table S7**, the rank of the consistency on each dataset is calculated for each method, along with the average consistency rank and the average CGC overlap rank. According to (**Fig. 3b**), the method achieving the highest consistency rank is the harmonic mean p-value method, followed by Kost's method, Brown's method, and Stouffer's method, indicating that the group of variants of Fisher's method are outstanding in consistency. On the other hand, the highest CGC overlap rank is achieved by the 2D-projection method from MutsigCV, followed by the harmonic mean p-value method, Stouffer's method, Kost's method, and Cauchy's method. The overall highest rank in consistency and CGC overlap is achieved by the harmonic mean p-value method. For convolution and likelihood ratio tests from MuSiC, the main reason of their low consistency and CGC overlap is that the numbers of resulting genes are too large (**Table S3**). However, according to **Fig. S1** and **S2**, among all CGC genes identified by 10 methods, most of them are covered by methods from MuSiC, indicating that the rest of genes are also worth investigating. From **Fig. S2** and **Fig. 3b**, it can also be seen that Brown's method identifies a number of CGC genes that are not predicted by other methods while maintaining a decent consistency. Thus, the reflection of Brown's method on genetic mechanisms deserves to be further evaluated. We also test the consistency of genes collectively predicted by 7 methods. According to **Fig. 3c** and **Table S6**, the overall consistency by applying the voting strategy is the highest.

Furthermore, we compared the difference in prediction performance among DriverDetector and several representative methods based on different principles. Specifically, a method based on multi-omics data called Rdriver [18], a method based on machine learning called DriverML [19], a method based on joint prediction called DriverGenePathway [13], along with MutSigCV [8] and MuSiC [12] are used for comparisons. The minimal required votes for DriverDetector are set from 6 to 10. All methods are tested on the BRCA dataset, which contains the most samples among all 12 datasets. The measures include the number of predicted genes, the overlap with CGC, if new genes outside the CGC list are found, and consistency. The results are summarized in **Table 1**, where the methods are ranked by their consistency. It needs to emphasized that based on the purpose of driver gene prediction methods, new genes outside the CGC list are expected to be found. According to **Table 1**, by setting the minimal required votes to 6,7, and 8, DriverDetector achieves a high consistency and CGC overlap while predicting new genes. Among all methods that predict new genes, MutSigCV achieves the highest overlap with CGC. However, the consistency of MutSigCV is quite low. On the other hand, by applying joint prediction, DriverGenePathway also predicts genes with high consistency. However, since all predicted genes are already known drivers, its practicality is highly reduced. Overall, we find that DriverDetector is a more effective method for driver genes identification.

3.3. Evaluation of the impact of sample size on results

According to **Table S2** and **S3**, huge differences exist in the number of genes identified for different cancers. Specifically, the predicted gene number based on the LUAD dataset is far more than the others. Considering that the LUAD dataset contains 567 samples, which is the second highest among all cancer sets. Therefore, we assess whether sample size has an impact on the results. In **Table S8**, the sample size of each dataset, along with the number of predicted genes by at least one method and the overlap with CGC is listed. Based on the ranks of sample size, predicted genes, CGC overlap gene number, and CGC overlap percentage, 6 plots in **Fig. S3** are generated to show their relationship. **Fig. S3 (a-c)** show that with the increase of sample size, the number of predicted genes and the ones in CGC both increases, while the CGC overlap percentage drops. This indicates that it is hard to achieve a balance between the number of genes and the overlap percentage of CGC, which is shown more clearly in **Fig. S3 (e)**. Our solution in DriverDetector

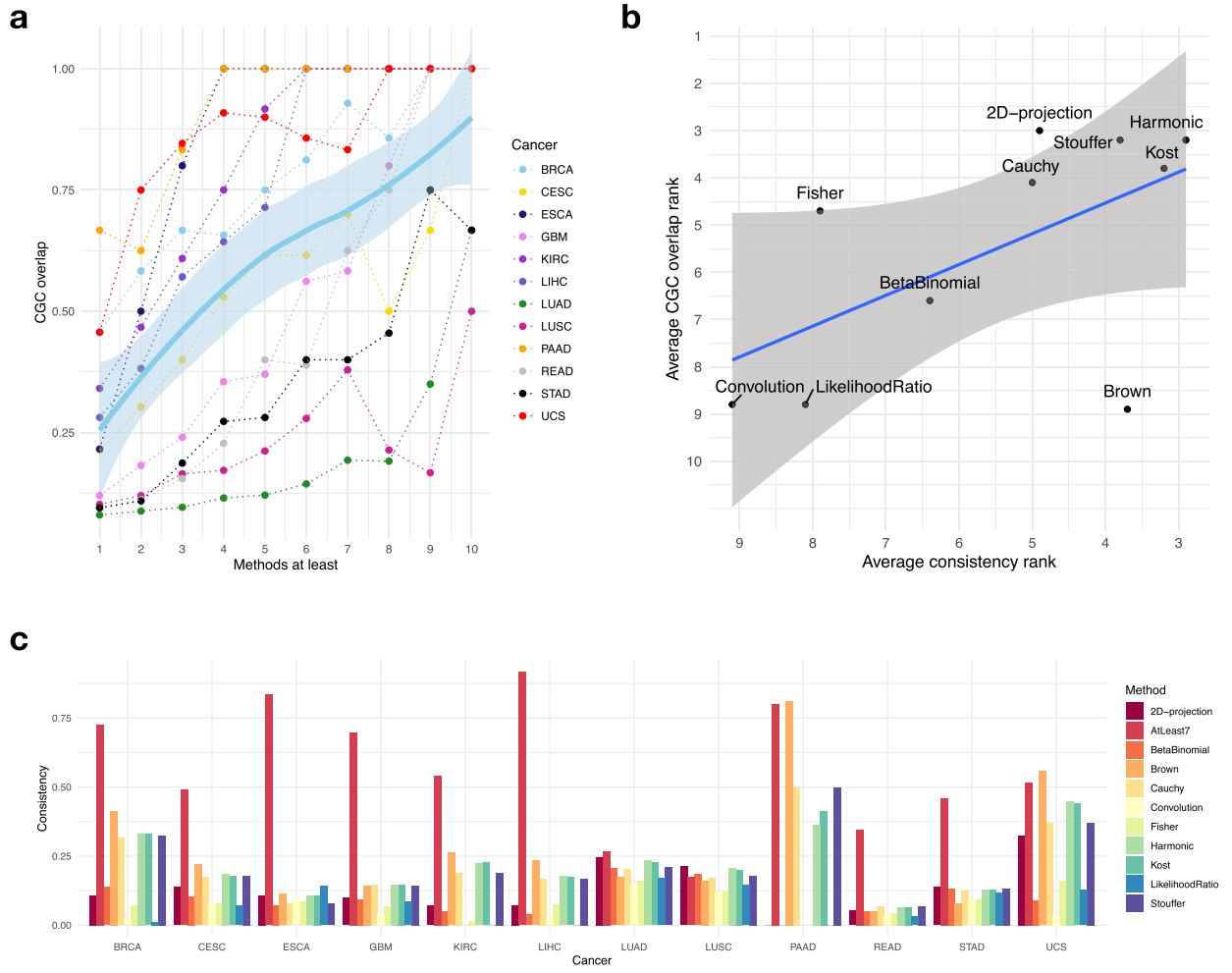


Fig. 3. Performance evaluation of statistical methods. (a) Overlap of predicted genes by at least 1 to 10 methods with CGC driver list on 12 cancer datasets. (b) Rank of 10 methods based on CGC overlap percent and consistency. (c) Consistency comparison among genes predicted by at least 7 methods and each individual method.

Table 1
Performance of cancer driver gene prediction methods on the BRCA dataset.

Method	No. genes	CGC overlap	Consistency	New genes identified
DriverDetector_atleast8	7	0.86	0.8	✓
DriverDetector_atleast6	16	0.81	0.79	✓
DriverDetector_atleast7	14	0.93	0.72	✓
Rdriver	30	0.43	0.4	✓
DriverML	168	0.21	0.32	✓
MuSiC	36	0.67	0.14	✓
MutSigCV	23	0.91	0.1	✓
DriverDetector_all10	2	1	1	✗
DriverGenePathway	11	1	0.83	✗
DriverDetector_atleast9	2	1	0.67	✗

is to control the number of genes and the overlap percent with CGC by adjusting the minimum number of votes, which is a more suitable and flexible way considering the impact of sample size.

3.4. Interpretation of the charts generated by DriverDetector

To illustrate the statistical analysis module, the results collectively predicted by all methods on the BRCA dataset are shown in **Fig. 4** as an example. The resulting genes are *ARID1A, CBFB, CTCF, KMT2C, RUNX1, TP53*, and *ZFHX4*. As shown in **Fig. 4b**, the confidence levels of these genes are slightly different, where *ARID1A, CBFB, CTCF, KMT2C, RUNX1*, and *TP53* are identified with

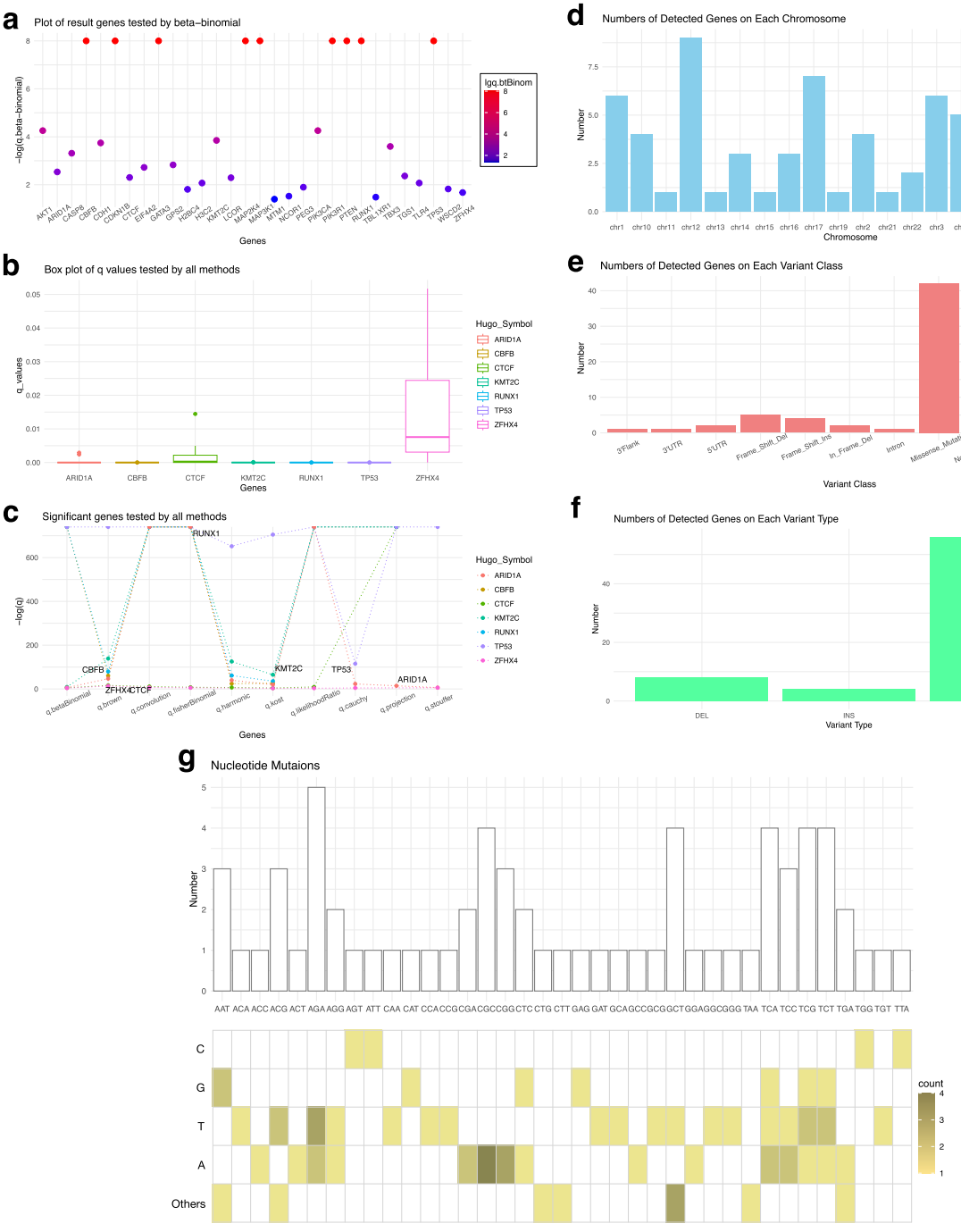


Fig. 4. Results for the BRCA dataset. (a) The q-values after taking the negative logarithms of significant genes identified by the beta-binomial method. (b) A box plot of q-values of the result gene set identified by different methods. (c) The q-values of each method. (d) Number of all result genes on each chromosome. (e) Number of all result genes on each variant class. (f) Number of all result genes on each variant type. (g) Number of all result genes on each nucleotide mutation type.

high confidence, while *ZFHX4* has a relatively lower confidence level than the other genes. **Fig. 4c** shows more details of the q-values obtained by each method, from which we can find that *CBFB, KMT2C, RUNX1*, and *TP53* are the most consistent genes. In bar plots **Fig. 4d-g**, we calculate the distribution of all genes below the q-value threshold on major variables, which gives further information for analysis. The plots show that the factors such as chr 12, missense mutation, and single-nucleotide polymorphism contain more resulting genes than others. Except for searching gene individuals, DriverDetector also has the potential to reveal signaling pathways based on the de novo method. By setting the number of genes in each set to 3, the results for 12 cancer datasets are summarized in **Table S9**. Here we take the result of LUAD as an example, where a gene set {*KRAS*, *TTN*, *EGFR*} is found. According to the

KEGG (Kyoto Encyclopedia of Genes and Genomes) database [39], the gene set is a subset of the Proteoglycans in cancer pathway. Proteoglycans are key molecular effectors of cell surface and pericellular micro-environments, which perform multiple functions in cancer and angiogenesis by virtue of their polyhedric nature and their ability to interact with both ligands and receptors that regulate neoplastic growth and neovascularization [40,41]. Therefore, it can be concluded that DriverDetector is able to effectively identify potential driver genes as well as gene sets of high coverage and exclusivity.

4. Discussion

Heterogeneity is a major factor that makes the early diagnosis of cancer difficult. For most cancers, the cause lies in the mutations of a group of genes. Therefore, identifying driver genes based on genomic data is an important way to accelerate the research on the pathological mechanisms of cancer. However, compared with the huge number of genes, the insufficient number of samples greatly increases the probability of random errors. As the application of genomics analyses in biology and biomedicine continues to increase, several computational analysis strategies have been developed, while few methods manage to maintain the balance among the predicted gene number, the overlap with known drivers, and the consistency of prediction. In the result section, we evaluate the results of several representative methods based on different principles. Although methods based on multi-omics data and machine learning have become a research hotspot, we believe that when the samples are not sufficient, including more mutational features or multi-omics data will further deepen the data imbalance and thus affect the consistency of the results. Similarly, the performance of machine learning is strongly affected by sample size, which easily leads to overfitting or insufficient training.

To address existing problems, we develop the DriverDetector R package for robust prediction of cancer driver genes and downstream analysis. DriverDetector integrates two widely influential driver gene identification methods (MutSigCV and MuSiC), five variants of Fisher's method, and a de novo method. By running DriverDetector on multiple cancer datasets, we first verify the limitations of individual methods in achieving consistent results. Then, we prove that by applying the voting strategy, the consistency of predicted genes is significantly improved, and the circumstances of obtaining too many genes or too low overlap with known drivers can be avoided at the same time. We further assess the impact of sample size on results and provide recommendations on the usage of DriverDetector. The main highlight of DriverDetector is the collective prediction based on multiple statistical methods, which has been verified to be a promising strategy for robust and consistent prediction of cancer driver genes. By adjusting the minimal required votes and applying individual methods, DriverDetector is also able to guarantee certain flexibility for various datasets. In addition, the extensions of Fisher's methods demonstrate superiority in achieving consistent results. As sequencing data of various cancers increases, DriverDetector can be used to identify new driver genes with robustness, which assists in early diagnosis of cancer and development of targeted drugs.

However, DriverDetector also has some shortcomings. First of all, DriverDetector is not applicable to all mutation types, such as copy number variations (CNV) and chromosomal structural variations (CSV). Secondly, certain driver genes predicted by a few methods could be discarded as a cost for pursuing robustness and consistency. Lastly, some parameters need to be adjusted manually based on experience. In future work, other mutation types such as CNV and CSV need to be considered. Automatic parameter selection will also be the direction of efforts. In addition, more technics and methods can be combined to uncover the mechanism of cancer. We implement DriverDetector as an open-source R package for researchers to study and utilize.

Funding

This work was supported by the "1+X" Program Cross-Disciplinary Innovation Project under Grant 2022JCXKYB07, and in part by Wu Jieping Medical Foundation under Grant 320.6750.2022-19-81.

CRediT authorship contribution statement

Zeyuan Wang: Writing – original draft, Software, Methodology, Data curation. **Hong Gu:** Writing – review & editing, Supervision. **Pan Qin:** Writing – review & editing, Methodology, Conceptualization. **Jia Wang:** Writing – review & editing, Visualization, Validation, Conceptualization.

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.

Data availability

Publicly available datasets are analyzed in this study. The mutation data from TCGA can be found at: https://tcga-data.nci.nih.gov/tcga/. The MutSigCV files can be found at ftp://ftp.broadinstitute.org/pub/genepattern/example_files/MutSigCV_1.3/. The DriverDetector software can be found at https://github.com/FrancisWang96/DriverDetector. Additional results can be found at Supplementary.

Acknowledgements

We express our gratitude to Dalian University of Technology and Second Hospital of Dalian Medical University to carry out this work successfully.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.heliyon.2024.e33582.

References

[1] J.N. Weinstein, E.A. Collisson, G.B. Mills, K.R. Shaw, B.A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J.M. Stuart, The cancer genome atlas pan-cancer analysis project, Nat. Genet. 45 (2013) 1113–1120.
[2] L. Chin, J.N. Andersen, P.A. Futreal, Cancer genomics: from discovery science to personalized medicine, Nat. Med. 17 (2011) 297–303.
[3] B. Vogelstein, N. Papadopoulos, V.E. Velculescu, S. Zhou, L.A. Diaz Jr, K.W. Kinzler, Cancer genome landscapes, Science 339 (2013) 1546–1558.
[4] Z. Waks, O. Weissbrod, B. Carmeli, R. Norel, F. Utro, Y. Goldschmidt, Driver gene classification reveals a substantial overrepresentation of tumor suppressors among very large chromatin-regulating proteins, Sci. Rep. 6 (2016) 38988.
[5] K.D. Korthauer, C. Kendziorski, Madgic: a model-based approach for identifying driver genes in cancer, Bioinformatics 31 (2015) 1526–1535.
[6] C.J. Tokheim, N. Papadopoulos, K.W. Kinzler, B. Vogelstein, R. Karchin, Evaluating the evaluation of cancer driver genes, Proc. Natl. Acad. Sci. 113 (2016) 14330–14335.
[7] F. Vandin, E. Upfal, B.J. Raphael, De novo discovery of mutated driver pathways in cancer, Genome Res. 22 (2012) 375–385.
[8] M.S. Lawrence, P. Stojanov, P. Polak, G.V. Kryukov, K. Cibulskis, A. Sivachenko, S.L. Carter, C. Stewart, C.H. Mermel, S.A. Roberts, et al., Mutational heterogeneity in cancer and the search for new cancer-associated genes, Nature 499 (2013) 214–218.
[9] B.J. Raphael, J.R. Dobson, L. Oesper, F. Vandin, Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine, Gen. Med. 6 (2014) 5.
[10] F. Dietlein, D. Weghorn, A. Taylor-Weiner, A. Richters, B. Reardon, D. Liu, E.S. Lander, E.M. Van Allen, S.R. Sunyaev, Identification of cancer driver genes based on nucleotide context, Nat. Genet. 52 (2020) 208–218.
[11] R.I. Juul, M.M. Nielsen, M. Juul, L. Feuerbach, J.S. Pedersen, The landscape and driver potential of site-specific hotspots across cancer genomes, npj Genom. Med. 6 (2021) 33.
[12] N.D. Dees, Q. Zhang, C. Kandoth, M.C. Wendl, W. Schierding, D.C. Koboldt, T.B. Mooney, M.B. Callaway, D. Dooling, E.R. Mardis, et al., Music: identifying mutational significance in cancer genomes, Genome Res. 22 (2012) 1589–1598.
[13] X. Xu, Z. Qi, D. Zhang, M. Zhang, Y. Ren, Z. Geng, Drivergenepathway: identifying driver genes and driver pathways in cancer based on mutsigcv and statistical methods, Comput. Struct. Biotechnol. J. 21 (2023) 3124–3135.
[14] B. Vogelstein, K.W. Kinzler, Cancer genes and the pathways they control, Nat. Med. 10 (2004) 789–799.
[15] X. Xu, P. Qin, H. Gu, J. Wang, Y. Wang, Adaptively weighted and robust mathematical programming for the discovery of driver gene sets in cancers, Sci. Rep. 9 (2019) 5959.
[16] S. Constantinescu, E. Szczurek, P. Mohammadi, J. Rahnenführer, N. Beerenwinkel, Timex: a waiting time model for mutually exclusive cancer alterations, Bioinformatics 32 (2016) 968–975.
[17] M.D. Leiserson, H.-T. Wu, F. Vandin, B.J. Raphael, Comet: a statistical approach to identify combinations of mutually exclusive alterations in cancer, Genome Biol. 16 (2015) 1–20.
[18] Y. Han, J. Yang, X. Qian, W.-C. Cheng, S.-H. Liu, X. Hua, L. Zhou, Y. Yang, Q. Wu, P. Liu, et al., Driverml: a machine learning algorithm for identifying driver genes in cancer sequencing studies, Nucleic Acids Res. 47 (2019) e45.
[19] Z. Wang, K.-S. Ng, T. Chen, T.-B. Kim, F. Wang, K. Shaw, K.L. Scott, F. Meric-Bernstam, G.B. Mills, K. Chen, Cancer driver mutation prediction through Bayesian integration of multi-omic data, PLoS ONE 13 (2018) e0196939.
[20] Y. You, X. Lai, Y. Pan, H. Zheng, J. Vera, S. Liu, S. Deng, L. Zhang, Artificial intelligence in cancer target identification and drug discovery, Signal Transduct. Targeted Ther. 7 (2022) 156.
[21] Z. Sondka, S. Bamford, C.G. Cole, S.A. Ward, I. Dunham, S.A. Forbes, The cosmic cancer gene census: describing genetic dysfunction across all human cancers, Nat. Rev. Cancer 18 (2018) 696–705.
[22] H. Gu, X. Xu, P. Qin, J. Wang, Fi-net: identification of cancer driver genes by using functional impact prediction neural network, Front. Genet. 11 (2020) 564839.
[23] H. Pagès, BSgenome: Software infrastructure for efficient representation of full genomes and their SNPs, 2024.
[24] T. Sjoblom, S. Jones, L.D. Wood, D.W. Parsons, J. Lin, T.D. Barber, D. Mandelker, R.J. Leary, J. Ptak, N. Silliman, et al., The consensus coding sequences of human breast and colorectal cancers, Science 314 (2006) 268–274.
[25] R.A. Fisher, et al., Statistical Methods for Research Workers, 1936.
[26] W. Poole, D.L. Gibbs, I. Shmulevich, B. Bernard, T.A. Knijnenburg, Combining dependent p-values with an empirical adaptation of brown's method, Bioinformatics 32 (2016) i430–i436.
[27] J.T. Kost, M.P. McDermott, Combining dependent p-values, Stat. Probab. Lett. 60 (2002) 183–190.
[28] M.B. Brown, 400: a method for combining non-independent, one-sided tests of significance, Biometrics (1975) 987–992.
[29] D.J. Wilson, The harmonic mean p-value for combining dependent tests, Proc. Natl. Acad. Sci. 116 (2019) 1195–1200.
[30] Y. Liu, J. Xie, Cauchy combination test: a powerful test with analytic p-value calculation under arbitrary dependency structures, J. Am. Stat. Assoc. (2019).
[31] S. Stouffer, A study of attitudes, Sci. Am. 180 (1949) 11–15.
[32] G. Getz, H. Hofling, J.P. Mesirov, T.R. Golub, M. Meyerson, R. Tibshirani, E.S. Lander, Comment on "The consensus coding sequences of human breast and colorectal cancers", Science 317 (2007) 1500.
[33] Q.-H. Nguyen, D.-H. Le, Improving existing analysis pipeline to identify and analyze cancer driver genes using multi-omics data, Sci. Rep. 10 (2020) 20521.
[34] S. Zong, P.-p. Xu, Y.-h. Xu, Y. Guo, A bioinformatics analysis: Zfhx4 is associated with metastasis and poor survival in ovarian cancer, J. Ovarian Res. 15 (2022) 90.
[35] T. Qing, S. Zhu, C. Suo, L. Zhang, Y. Zheng, L. Shi, Somatic mutations in zfhx4 gene are associated with poor overall survival of Chinese esophageal squamous cell carcinoma patients, Sci. Rep. 7 (2017) 4951.
[36] P. Zhang, Z. An, C. Sun, Y. Xu, Z. Zhang, Flg gene mutation up-regulates the abnormal tumor immune response and promotes the progression of prostate cancer, Curr. Pharm. Biotechnol. 23 (2022) 1658–1670.
[37] M.M. Madeleine, L.G. Johnson, A.G. Smith, J.A. Hansen, B.B. Nisperos, S. Li, L.-P. Zhao, J.R. Daling, S.M. Schwartz, D.A. Galloway, Comprehensive analysis of hla-a, hla-b, hla-c, hla-drb1, and hla-dqb1 loci and squamous cell cervical cancer risk, Cancer Res. 68 (2008) 3532–3539.

[38] T. Michelakos, F. Kontos, T. Kurokawa, L. Cai, A. Sadagopan, D. Krijgsman, W. Weichert, L.G. Durrant, P.J. Kuppen, C.R. Ferrone, et al., Differential role of hla-a and hla-b, c expression levels as prognostic markers in colon and rectal cancer, J. ImmunoTher. Cancer 10 (2022).
[39] M. Kanehisa, S. Goto, Kegg: Kyoto encyclopedia of genes and genomes, Nucleic Acids Res. 28 (2000) 27–30.
[40] R.V. Iozzo, R.D. Sanderson, Proteoglycans in cancer biology, tumour microenvironment and angiogenesis, J. Cell. Mol. Med. 15 (2011) 1013–1031.
[41] M. Mellai, C. Casalone, C. Corona, P. Crociara, A. Favole, P. Cassoni, D. Schiffer, R. Boldorini, Chondroitin sulphate proteoglycans in the tumour microenvironment, in: Tumor Microenvironment: Extracellular Matrix Components–Part B, 2020, pp. 73–92.