OXFORD (GIGA)n SCIENCE

GigaScience, 2023, **12**, 1–11
DOI: 10.1093/gigascience/giad081
Technical Note

Katdetectr: an R/bioconductor package utilizing unsupervised changepoint analysis for robust kataegis detection

Daan M. Hazelaar [1,†], Job van Riet [1,2,5,†], Youri Hoogstrate [3] and Harmen J. G. van de Werken [2,4,*]

[1]Department of Medical Oncology, Erasmus MC Cancer Institute, University Medical Center, 3015 GD, Rotterdam, the Netherlands
[2]Department of Urology, Erasmus MC Cancer Institute, University Medical Center, 3015 GD, Rotterdam, the Netherlands
[3]Department of Neurology, Erasmus MC Cancer Institute, University Medical Center, 3015 GD, Rotterdam, the Netherlands
[4]Department of Immunology, Erasmus MC Cancer Institute, University Medical Center, 3015 GD, Rotterdam, the Netherlands
[5]Current Address: Division of AI in Oncology, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, 69120, Heidelberg, Germany
*Correspondence address. Harmen J. G. van de Werken, Erasmus MC Department of Immunology (Room, Na-1218) P.O. box 2040 3000 CA Rotterdam, the Netherlands. E-mail: h.vandewerken@erasmusmc.nl
†Shared first-authorship

Abstract

Background: Kataegis refers to the occurrence of regional genomic hypermutation in cancer and is a phenomenon that has been observed in a wide range of malignancies. A kataegis locus constitutes a genomic region with a high mutation rate (i.e., a higher frequency of closely interspersed somatic variants than the overall mutational background). It has been shown that kataegis is of biological significance and possibly clinically relevant. Therefore, an accurate and robust workflow for kataegis detection is paramount.

Findings: Here we present Katdetectr, an open-source R/Bioconductor-based package for the robust yet flexible and fast detection of kataegis loci in genomic data. In addition, Katdetectr houses functionalities to characterize and visualize kataegis and provides results in a standardized format useful for subsequent analysis. In brief, Katdetectr imports industry-standard formats (MAF, VCF, and VRanges), determines the intermutation distance of the genomic variants, and performs unsupervised changepoint analysis utilizing the Pruned Exact Linear Time search algorithm followed by kataegis calling according to user-defined parameters.
We used synthetic data and an *a priori* labeled pan-cancer dataset of whole-genome sequenced malignancies for the performance evaluation of *Katdetectr* and 5 publicly available kataegis detection packages. Our performance evaluation shows that Katdetectr is robust regarding tumor mutational burden and shows the fastest mean computation time. Additionally, Katdetectr reveals the highest accuracy (0.99, 0.99) and normalized Matthews correlation coefficient (0.98, 0.92) of all evaluated tools for both datasets.

Conclusions: Katdetectr is a robust workflow for the detection, characterization, and visualization of kataegis and is available on Bioconductor: https://doi.org/doi:10.18129/B9.bioc.katdetectr.

Keywords: kataegis, R-package, Bioconductor, changepoint analysis, cancer

Introduction

Large-scale next-generation sequencing of malignancies has revealed that a myriad of mutational mechanisms and mutational rates are at play within even a single tumor genome. Moreover, it has been shown that mutations can cluster together, that is, the acquired mutations are found in proximity to one another, much closer than expected if each base pair had an equal probability of being mutated. This phenomenon was termed *kataegis* and its respective genomic location was termed a *kataegis locus* [1, 2].

Kataegis, Greek for thunderstorm or shower, was first observed and visualized in whole-genome sequencing (WGS) data of 21 primary breast cancers [1]. Alexandrov and colleagues [2] subsequently detected 873 kataegis loci in a pan-cancer dataset containing 507 WGS samples from primary malignancies.

Extensive exploration of the etiology of kataegis revealed a significant positive association between kataegis and 2 distinct mutational signatures (COSMIC signatures SBS2 and SBS13) both attributed to the APOBEC enzyme family [3, 4]. Subsequently, multiple studies confirmed the importance of the APOBEC enzymes in cancer, showing that APOBEC enzymes are a major cause of mutagenesis, grouped in clusters, dispersed throughout the cancer genome and in extrachromosomal DNA [5–7]. Additionally, kataegis has been ascribed in lymphomas to 2 other mutational signatures (COSMIC signatures SBS84 and SBS85) related to the APOBEC family member activation-induced cytidine deaminase (AID) enzyme [8].

Moreover, the locations of kataegis loci have been associated with locations of somatic structural variant breakpoints. Kataegis loci have been observed most frequently within the proximity of deletions and complex rearrangement breakpoints [3, 9]. Furthermore, kataegis can occur within known cancer driver genes, including *TP53*, *EGFR*, and *BRAF*, which are associated with overall survival in some cancer types [5, 18]. However, the clinical relevance of kataegis remains to be validated and therefore obfuscates kataegis as a clinical biomarker for prognosis. Moreover, future insight into kataegis etiology and clinical applications requires accurate and robust detection of kataegis.

Since the discovery of kataegis, different computational detection tools using genomic variant data have been developed and are publicly available, including MafTools [10],

Received: February 24, 2023. **Revised:** June 15, 2023. **Accepted:** September 12, 2023

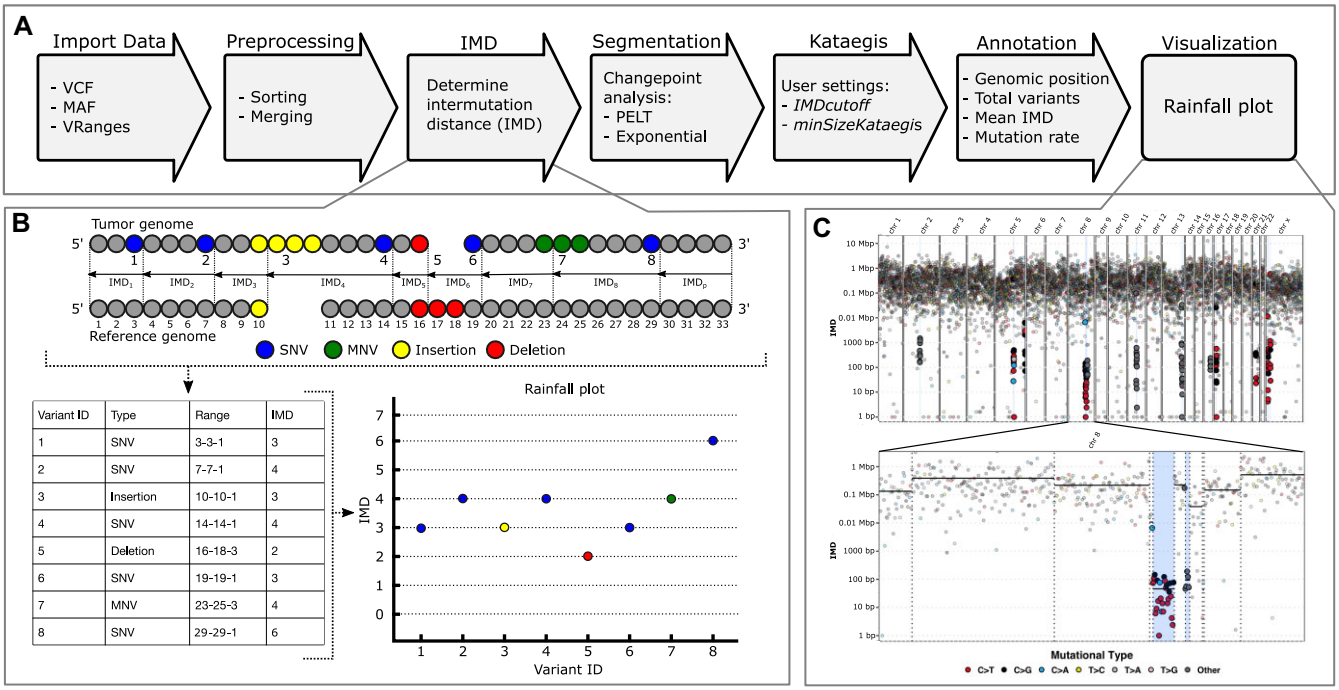


Figure 1: Overview of the Katdetectr workflow, intermutation distance, and rainfall plots. (A) General workflow of Katdetectr from data import to data visualization represented by arrows. (B) The intermutation distance (IMD) is determined for all genomic variants in each chromosome, and rainfall plots are used to visualize the IMDs. Single-nucleotide variant (SNV), multinucleotide variant (MNV). (C) Rainfall plot of WGS breast cancer sample PD7049a as interrogated by Katdetectr with IMDcutoff = 1,000 and minSizeKataegis = 6 [2]. Y-axis: IMD, x-axis: variant ID ordered on genomic location, light blue rectangles: kataegis loci with genomic variants within kataegis loci shown in bold. The color depicts the mutational type. The vertical lines represent detected changepoints, while black horizontal solid lines show the mean IMD of each segment.

ClusteredMutations [11], kataegis [12], SeqKat [13], and SigProfilerClusters [14]. These packages employ distinct statistical methods for kataegis detection and differ in their ease of use and computational feasibility. Therefore, a comparison of their performances is currently needed.

Here, we introduce Katdetectr, an R-based Bioconductor package that contains a suite for the detection, characterization, and visualization of kataegis. Additionally, we have evaluated and compared the performance of Katdetectr to the 5 commonly used and publicly available kataegis detection packages.

Results

The principle of Katdetectr is to assess the variation in the mutation rate of a cancer genome. To achieve this, Katdetectr starts by importing and preprocessing industry-standard variant calling formats (VCF, MAF, VRanges) (Fig. 1A). Next, the intermutation distance (IMD) is determined, which denotes the distance between variants in base pairs (Fig. 1B; see Methods). Unsupervised changepoint analysis is performed, using the IMD as input, which results in detected changepoints. The changepoints, which denote the points at which the distribution of the IMD changes, are used to segment the genomic sequence. Finally, segments are annotated and labeled as a putative kataegis locus if a segment fits the user-defined settings: the mean IMD of the segment $\leq$ *IMDcutoff* and the number of variants in the segment $\geq$ *minSizeKataegis*. The IMD, segmentation, and detected kataegis loci can be visualized by *Katdetectr* in a rainfall plot (Fig. 1C).

Katdetectr search algorithm selection

To optimize Katdetectr for kataegis detection, we generated a synthetic dataset to test 4 changepoint search algorithms: pruned exact linear time (PELT) [15], binary segmentation (BinSeg) [15], segment neighborhoods (SegNeigh) [17], and at most one change (AMOC). The synthetic dataset contains 1,024 samples with a varying number of kataegis loci and tumor mutational burden (TMB) (see Methods). All variants in this dataset were binary labeled for kataegis, as a variant either lies within a kataegis locus (TRUE) or not (FALSE). This dataset was considered ground truth and was used for computing performance metrics. We analyzed the synthetic dataset separately for each search algorithm showing that the PELT algorithm outperformed the alternatives (Supplementary Table 1, Supplementary Figs. S1, S2). Therefore, we set PELT as the default search algorithm in Katdetectr.

Performance Evaluation

We utilized the synthetic dataset to evaluate the performances of Katdetectr and 5 publicly available kataegis detection packages: MafTools, ClusteredMutations, Kataegis, SeqKat, and SigProfilerClusters (Table 1, Supplementary Table S1). Katdetectr revealed the highest overall accuracy (0.99), normalized Matthews correlation coefficient (nMCC: 0.98), and F1 score (0.97), whereas ClusteredMutations showed the highest true-positive rate (TPR: 0.99) and Kataegis showed the highest true-negative rate (TNR: 0.99). Most packages showed a high nMCC for samples with a TMB ranging from 0.1 to 50. However, the performance of all packages dropped for samples with a TMB $\geq$100 (Fig. 2A). More specifically, for Katdetectr and Kataegis, this is due to an increase in false negatives. For SeqKat, MafTools, ClusteredMutations, and SigProfilerClusters, this performance drop is due to an increase in false positives in samples with a TMB of 100 and 500 (Supplementary Fig. S1).

Next to the synthetic dataset, we evaluated the performance of the kataegis detection packages on a dataset containing 507 *a*

Table 1: Summary and performance of kataegis detection packages

					Synthetic dataset				WGS dataset			
Package	**Reference**	**Available on**	**Language**	**Method**	**Accuracy**	**nMCC**	**F1**	**TPR**	**TNR**	**Accuracy**	**nMCC**	**F1**	**TPR**	**TNR**
Katdetectr	[21]	Bioconductor	R	Changepoint analysis (PELT)	0.99	0.98	0.97	0.94	0.99	0.99	0.92	0.83	0.91	0.99
SeqKat	[13]	CRAN	R	Sliding window/exact binomial test	0.84	0.54	0.02	0.93	0.84	0.99	0.85	0.69	0.59	0.99
MafTools	[10]	Bioconductor	R	Sliding window/piecewise constant fit (PCF)	0.74	0.53	0.01	0.96	0.74	0.99	0.85	0.66	0.93	0.99
SigProfilerClusters	[14]	GitHub	Python	Model sample-specific IMD cutoff	0.65	0.52	0.01	0.88	0.65	0.99	0.84	0.68	0.66	0.99
ClusteredMutations	[11]	CRAN	R	Anti-Robinson matrix	0.70	0.53	0.01	0.99	0.74	0.99	0.83	0.61	0.99	0.99
Kataegis	[12]	GitHub	R	Piecewise constant fit (PCF)	0.99	0.80	0.52	0.36	0.99	0.99	0.56	0.03	0.02	0.99

Summary: information of all evaluated kataegis detection packages and their respective performance metrics regarding kataegis classification on 1,024 synthetic samples and 507 *a priori* labeled whole-genome sequenced (WGS) samples. Accuracy, normalized Matthews correlation coefficient (nMCC), F1-score, true-positive rate (TPR), and true-negative rate (TNR), pruned exact linear time (PELT), piecewise constant fit (PCF), and intermutation distance (IMD).
Note: Highest value per column is underscored.

priori labeled WGS samples from Alexandrov et al. [2] (see Methods). Katdetectr revealed the highest overall accuracy (0.99), nMCC (0.92), and F1 score (0.83), whereas ClusteredMutations showed the highest TPR (0.99) and SigProfilerClusters showed the highest TNR (0.99) (Table 1, Supplementary Fig. S1). Katdetectr, ClusteredMutations, and MafTools showed a high nMCC (>0.92) on the samples with a low or middle TMB. However, the performance of all packages drops for samples with a TMB >10 ($n = 20$) (Fig. 2A). This is due to an increase in false negatives by Kataegis and SeqKat and false positives by Katdetectr, MafTools, ClusteredMutations, and SigProfilerClusters.

We visualized the concordance regarding per sample kataegis classification and kataegis locus between Katdetectr, SigProfilerClusters, ClusteredMutations, MafTools, and the original authors of the WGS dataset: Alexandrov et al. [2] (Fig. 2B). In total, 451 kataegis loci were detected in 127 WGS samples by all the packages and the original publication. Interestingly, Katdetectr, SigProfilerClusters, ClusteredMutations, and MafTools concordantly detected 102 previously unannotated kataegis loci within the original publication.

The runtimes of all packages were recorded to give insight into the computational feasibility of these packages. Katdetectr showed the lowest mean runtime on both the synthetic and the WGS datasets (Fig. 2C).

Katdetectr examples with different TMBs

We highlight 4 samples from the datasets that illustrate how Katdetectr accurately detects kataegis loci regardless of the TMB of the respective sample (Fig. 3). The synthetic sample 124625_1_50_100 (TMB: 500) harbors 1 kataegis locus, containing 57 variants, which is detected by Katdetectr (Fig. 3A). This kataegis locus is also detected by SeqKat, MafTools, ClusteredMutations, and SigProfilerClusters, in addition to numerous false positives. The package Kataegis did not detect any kataegis loci in this synthetic sample.

In lung adenocarcinoma sample LUAD-E01014 (TMB: 7.6), Katdetectr detected 37 kataegis loci containing 449 variants (Fig. 3B). MafTools, ClusteredMutations, and SeqKat detected similar kataegis loci in this sample, whereas Kataegis and SigProfilerClusters did not detect any kataegis loci in this sample. In breast cancer sample PD7207a (TMB: 0.8), 2 kataegis loci were detected by Katdetectr MafTools, ClusteredMutations, and SigProfilerClusters (Fig. 3C). Kataegis and SeqKat did not detect any kataegis loci in this sample. Lastly, in the breast cancer sample PD4086a (TMB: 0.6), 1 kataegis locus was detected by all packages except for Kataegis (Fig. 3D).

Methods

Implementation of Katdetectr

Katdetectr (v1.2.0, git commit 5a6e5d04109eb082cbea040049dca34237b6c8f5) was developed in the R statistical programming language (v4.2.0) [23]. Katdetectr imports genomic variants through generic, standardized file formats for variant calling: MAF, VCF, or Bioconductor-standard VRanges objects. Within Katdetectr, the imported variants are preprocessed such that, per chromosome, all variants (all rows in variant file, including indels or structural variations) are sorted in ascending order based on their genomic position. Overlapping variants are merged into a single record as phasing and clonality are not considered by katdetectr. Following, per $chromosome_j$, the intermutation distance ($IMD_{i,j}$) of each

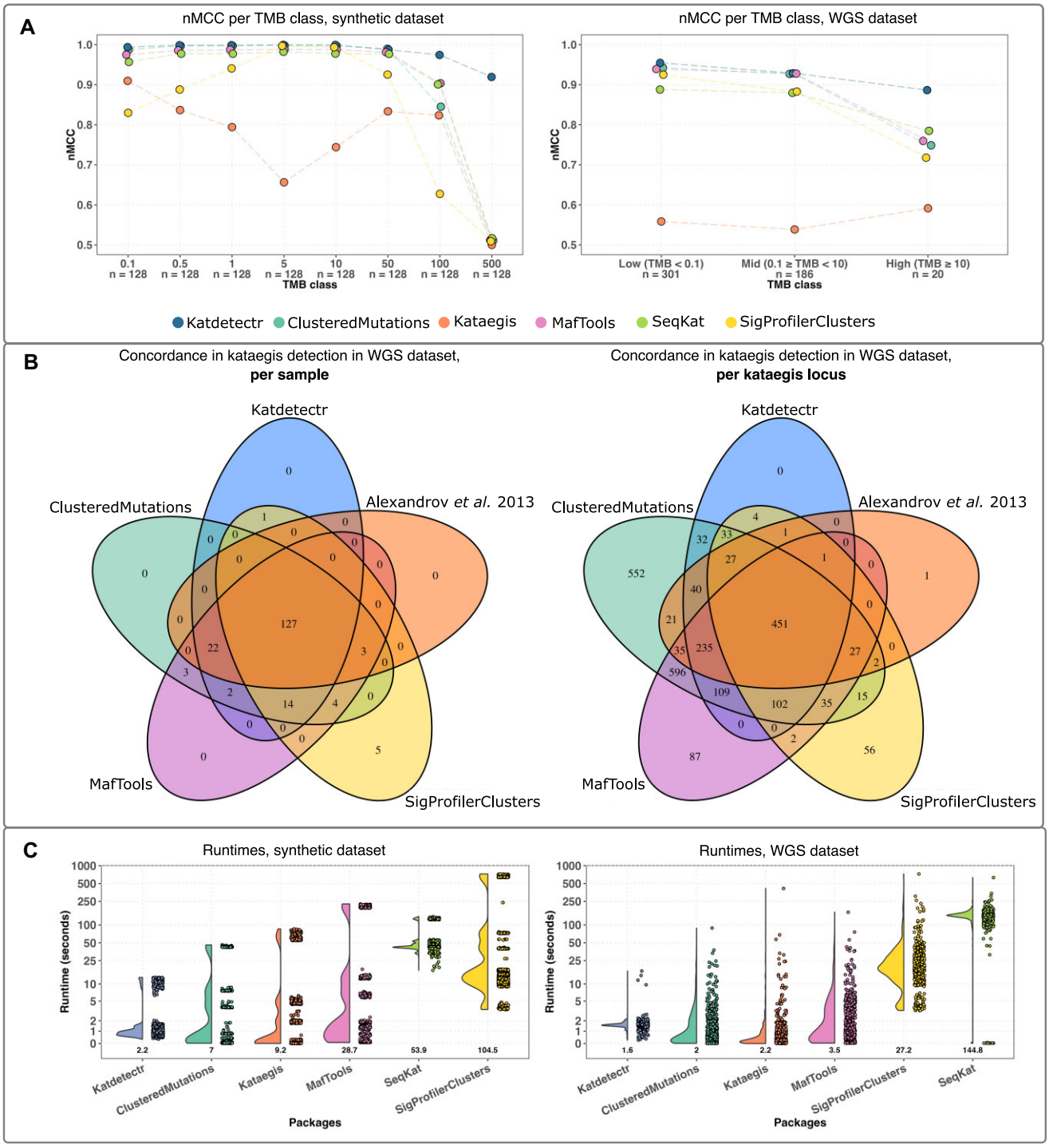


Figure 2: Performance evaluation of kataegis detection tools. (A) The normalized Matthews correlation coefficient (nMCC) per package and tumor mutational burden (TMB) class are depicted by individual data points connected with a dashed line (colored per package). (B) Venn diagrams showing the concordance between Katdetectr, SigProfilerClusters, MafTools, ClusteredMutations, and Alexandrov et al. regarding kataegis classification per sample (i.e., does a sample contain 1 or more kataegis loci) and per kataegis loci (i.e., does a detected kataegis locus overlap with a kataegis locus detected by another package). (C) Boxplots with individual data points represent the per sample runtimes of kataegis detection packages on the synthetic and whole-genome sequence datasets. Boxplots were sorted in ascending order based on mean runtime (depicted in the text below the boxplot). Y-axis is $\log_{10}$-scaled. Boxplots depict the interquartile range, with the median as a black horizontal line.

$variant_{i,\,j}$ and its closest upstream $variant_{i\,-\,1,\,j}$ is calculated according to

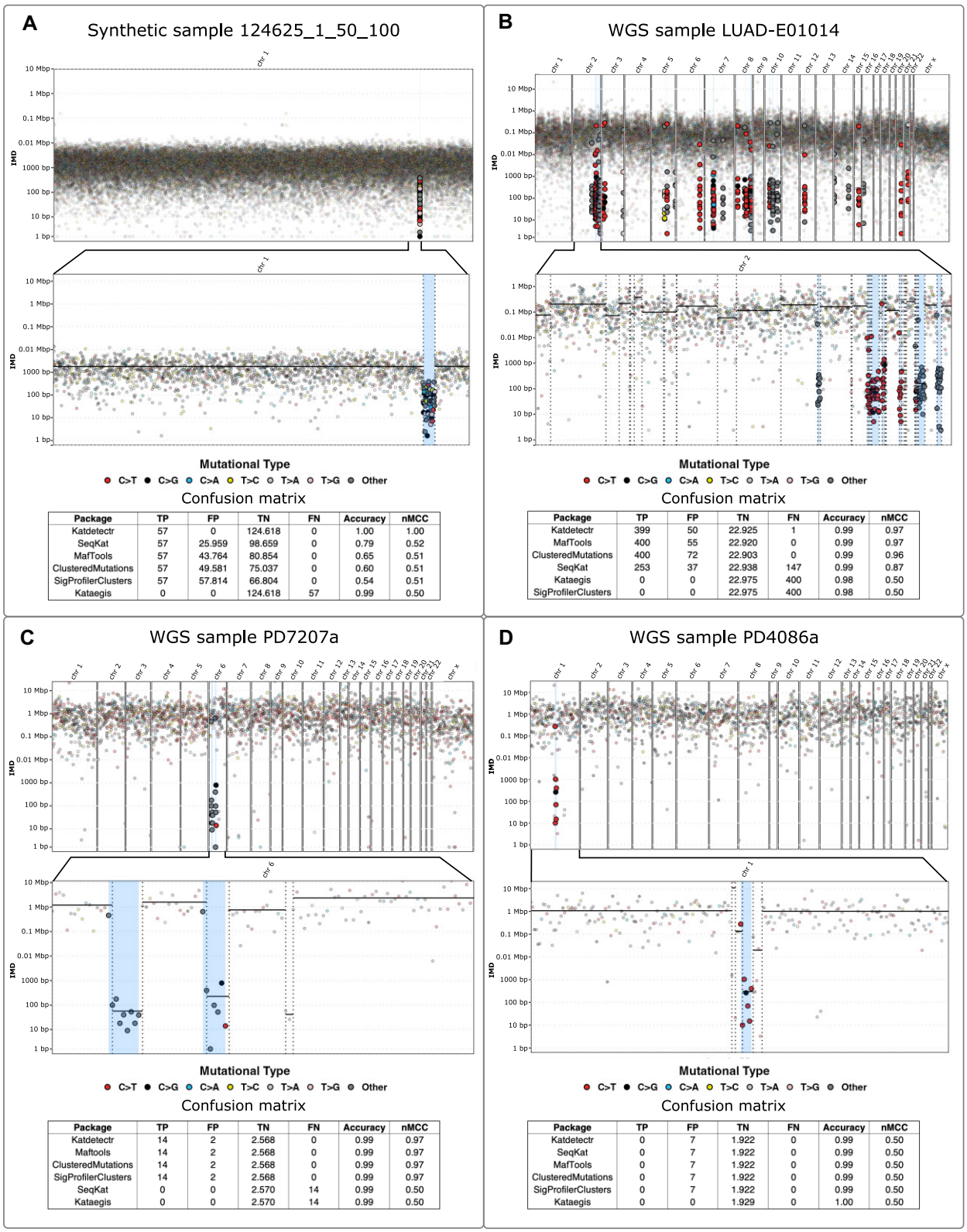
$$IMD_{i,j} = \begin{cases} i = 1\ s_{i,j} \\ i > 1\ s_{i,j} - s_{i-1,j} \end{cases} \quad i = \{1, 2, \ldots, k_j\} \qquad (1)$$

with i as the variant number, j as the chromosome number, s as the genomic location of the first base pair of a $variant_{i,j}$, and k_j as the total number of variants in $chromosome_j$ (Fig. 1B). Additionally, for each $chromosome_j$ one pseudo-IMD, $IMD_{p,j}$, is added such that

$$n_j = IMD_{p,j} + \sum_{i=1}^{k_j} IMD_{i,j} \qquad (2)$$

with n_j as the total number of base pairs in $chromosome_j$.

Katdetectr aims to identify genomic regions characterized by specific mutation rates. An unsupervised technique called



Confusion matrix

Package	TP	FP	TN	FN	Accuracy	nMCC
Katdetectr	57	0	124.618	0	1.00	1.00
SeqKat	57	25.959	98.659	0	0.79	0.52
MafTools	57	43.764	80.854	0	0.65	0.51
ClusteredMutations	57	49.581	75.037	0	0.60	0.51
SigProfilerClusters	57	57.814	66.804	0	0.54	0.51
Kataegis	0	0	124.618	57	0.99	0.50

Confusion matrix

Package	TP	FP	TN	FN	Accuracy	nMCC
Katdetectr	399	50	22.925	1	0.99	0.97
MafTools	400	55	22.920	0	0.99	0.97
ClusteredMutations	400	72	22.903	0	0.99	0.96
SeqKat	253	37	22.938	147	0.99	0.87
Kataegis	0	0	22.975	400	0.98	0.50
SigProfilerClusters	0	0	22.975	400	0.98	0.50

Confusion matrix

Package	TP	FP	TN	FN	Accuracy	nMCC
Katdetectr	14	2	2.568	0	0.99	0.97
Maftools	14	2	2.568	0	0.99	0.97
ClusteredMutations	14	2	2.568	0	0.99	0.97
SigProfilerClusters	14	2	2.568	0	0.99	0.97
SeqKat	0	0	2.570	14	0.99	0.50
Kataegis	0	0	2.570	14	0.99	0.50

Confusion matrix

Package	TP	FP	TN	FN	Accuracy	nMCC
Katdetectr	0	7	1.922	0	0.99	0.50
SeqKat	0	7	1.922	0	0.99	0.50
MafTools	0	7	1.922	0	0.99	0.50
ClusteredMutations	0	7	1.922	0	0.99	0.50
SigProfilerClusters	0	7	1.922	0	0.99	0.50
Kataegis	0	0	1.929	0	1.00	0.50

Figure 3: Rainfall plots constructed by Katdetectr and confusion matrices, accuracy, and nMCC for 4 samples. (A) Synthetic sample 124625_1_50_100 with tumor mutational burden (TMB): 500. (B) Lung adenocarcinoma whole-genome sequenced (WGS) sample LUAD-E01014 with TMB: 7.6. (C) Breast cancer WGS sample PD7207a with TMB: 2.5. (D) Breast cancer WGS sample PD4086a with TMB: 0.62. The WGS samples were collected and labeled for kataegis by Alexandrov et al. [2]; their results were used as ground truth to construct the confusion matrices and performance metrics. Rainfall plot: y-axis: IMD, x-axis: variant ID ordered on genomic location, light blue rectangles: kataegis loci with genomic variants within kataegis loci shown in bold. The color depicts the mutational type. The vertical lines represent detected changepoints, while black horizontal solid lines show the mean IMD of each segment. Confusion matrix: true positive (TP), false positive (FP), true negative (TN), false negative (FN), accuracy, and normalized Matthews correlation coefficient (nMCC).

changepoint analysis is performed per chromosome on the IMDs to assess the variability in mutation rate across each chromosome. Changepoint analysis refers to the process of detecting points in a sequence of observations where the statistical properties of the sequence significantly change. Subsequently, the detected changepoints are used to segment the input sequence into segments. For a detailed description of the changepoint analysis, see the work of Killick et al. [15]

We implemented the `cpt.meanvar()` function from the commonly used R changepoint package (v2.2.3) in Katdetectr for the unsupervised segmentation of IMDs, as detailed by [15, 24, 25]. We set the following parameter settings as default settings in Katdetectr: method, PELT; minimal segment length, 2; test statistic, exponential; and penalty, Bayesian information criterion (BIC).

After changepoint analysis, each segment is annotated with its respective genomic start and end positions, its mean IMD, and the total number of included variants. Since we use an exponential distribution as the test statistic in changepoint analysis, each segment has a corresponding rate parameter of the fitted exponential distribution. Whereas each segment is annotated with its corresponding mutation rate, the mutation rate of an entire sample can be expressed as the weighted arithmetic mean of the mutation rate of the segments:

$$\lambda_t = \frac{k_t}{n_t} = \sum_{s=1}^{m} \frac{\lambda_s \, n_s}{n_t} \tag{3}$$

with λ_t as the mutation rate of the entire sample, k_t as the total number of variants present in the sample, n_t as the total number of base pairs in the genome, m as the total number of segments in the sample, and λ_s and n_s as the mutation rate and the number of base pairs in $segment_s$.

To call a segment a putative kataegis locus, it has to adhere to 2 user-defined parameters: the maximum mean IMD of the segment (*IMDcutoff*) and the minimum number of included variants (*minSizeKataegis*). These parameters can be provided as static integer values or as a custom R function determining the IMD cutoff for each segment. For example, the following function for annotation of kataegis events, as used by the ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium, can be easily implemented in Katdetectr [3]:

$$\begin{aligned} IMDcutoff \;&\leq\; \frac{-log\left(1-\sqrt[k_s-1]{\frac{0.01}{L_s}}\right)}{\lambda_{med}} \\ with;\, [IMDcutoff] \;&=\; 1000 \end{aligned} \tag{4}$$

with $IMDcutoff_s$ as the IMD cutoff value, k_s as the number of mutations, and L_s as the length of $segment_s$ in base pairs. For this function, the rate of the whole sample is modeled assuming an exponential distribution with

$$\lambda_{med} = \frac{log\,(2)}{median\,(IMD)} \tag{5}$$

Henceforth, all segments satisfying these user-specified parameters are considered putative kataegis loci and stored appropriately. Two or more adjacent kataegis loci are merged and stored as a single record.

The output of Katdetectr consists of an S4 object of class "KatDetect" that stores all relevant information regarding kataegis detection and characterization. A KatDetect object contains 4 slots: (i) the putative kataegis loci (Granges), (ii) the detected segments (Granges), (iii) the inputted genomic variants with annotation (Vranges), and (iv) the parameters settings (List). These data objects can be accessed using accessor functions.

In addition, we implemented 3 methods for the KatDetect class: *summary*, *show*, and *rainfallPlot*. In concordance with R standards, the *summary* function prints a synopsis of the performed analysis, including the number of detected kataegis loci, and the number of variants inside a kataegis loci. The *show* function displays information regarding the S4 class and the synopsis.

The method *rainfallPlot* is a function for generating rainfall plots. These rainfall plots display the genomic ordered IMDs (from all genomic variants) within a sample and highlight putative kataegis loci and associated genomic variants. This function has additional arguments: *showSequence*, which allows the user to display specific chromosomes, and *showSegmentation*, for displaying the changepoints and the mean IMD of all segments.

For additional examples and more hands-on technical instructions, we refer to the accompanying vignette (Supplemental Vignette) or the online Bioconductor repository [21].

Performance evaluation

As multiple packages for kataegis detection are publicly available, we compared Katdetectr against MafTools (v2.13.0), ClusteredMutations (v1.0.1), kataegis (v0.99.2), SeqKat (v0.0.8), and SigProfilerClusters (v1.0.11) [6–10]. For benchmarking, we used an in-house generated synthetic dataset and an *a priori* labeled pan-cancer dataset of whole-genome sequenced malignancies. As not all evaluated packages accepted indels

We used the following definition of kataegis as postulated by Alexandrov and colleagues [2]: a kataegis locus is (i) a continuous segment harboring ≥6 variants and (ii) the captured IMDs within the segment have a mean IMD of ≤1,000 bp. To quantify and compare performances, the task of kataegis detection was reduced to a binary classification problem. The task of the kataegis detection packages was to correctly label each variant for kataegis (i.e., whether or not a genomic variant lies within a kataegis locus).

Performance metrics

Only a small fraction of all observed variants is located within kataegis loci, which results in a large class imbalance that renders the interpretation of performance metrics, such as accuracy, F1, TPR, and TNR, counterintuitive and possibly unrepresentative (Equation 3). Therefore, the nMCC was used as the primary metric for performance evaluation. The nMCC considers performance proportionally to both the size of positive and negative elements in a dataset [26].

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FP+TN+FN} \\ MCC &= \frac{TP{\cdot}TN - FP{\cdot}FN}{\sqrt{(TP+FP){\cdot}(TP+FN){\cdot}(TN+FP){\cdot}(TN+FN)}} \\ nMCC &= \frac{MCC+1}{2} \\ F1 &= \frac{TP}{TP+\frac{1}{2}(FP+FN)} \\ TPR &= \frac{TP}{TP+FN} \\ TNR &= \frac{TN}{TN+FP} \end{aligned} \quad . \tag{6}$$

Performance metrics. Accuracy, Matthews correlation coefficient (MCC), normalized Matthews correlation coefficient (nMCC), F1 score, true-positive rate (TPR), and true-negative rate (TNR).

1. True positive (TP): Predicted: variant in kataegis locus. Truth set: variant in kataegis locus.
2. False positive (FP): Predicted: variant in kataegis locus. Truth set: variant not in kataegis locus.

3. True negative (TN): Predicted: variant not in kataegis locus. Truth set: variant not in kataegis locus.
4. False negative (FN): Predicted: variant not in kataegis locus. Truth set: variant in kataegis locus.

We utilized Venn diagrams to display the concordance of the kataegis detection packages. We showed in which samples the packages detected 1 or more kataegis loci and which kataegis loci were detected by the packages. Two packages are said to detect the same kataegis locus if the genomic locations of their respective kataegis locus overlap by at least 1 base pair.

To give insight into the package's computation time, the package's runtime performance was recorded using the `proc.time()` function from the base R package. All packages and comparisons were run on the same server utilizing an AMD EPYC 7742 64-Core Processor. The packages Katdetectr and SigProfilerClusters contained options for parallel processing and used at most 4 cores per sample during the analyses. All other packages used a single processing core per sample.

All scripts necessary for running and visualizing the performance evaluation of all evaluated packages are available on GitHub [22]. All data used for the performance evaluation are available at Zenodo [27].

Synthetic data generation

The synthetic dataset was generated using the `generateSyntheticData()` function within the Katdetectr package. Mutations were randomly sampled on a reference genome such that each base has an equal probability, p, of being mutated (except for N bases for which $p = 0$). This reduces the occurrence of mutations on the reference genome to a sequence of $X_1, X_2, \ldots, X_n$, independent Bernoulli trials, X_i (i.e., a Bernoulli process), where

$$\begin{aligned} \mathbf{P}(X_i = 1) &= \mathbf{P}(\text{Mutation at } i\text{th base}) = p \\ \mathbf{P}(X_i = 0) &= \mathbf{P}(\text{No mutation at } i\text{th base}) = 1 - p \end{aligned} \tag{7}$$

with probability mass function (PMF), expectation, and variance:

$$\begin{aligned} p_s(k) &= \binom{n}{k} p^k (1-p)^{n-k}, \qquad k = 0, 1, \ldots, n \\ \mathbf{E}(S) &= np \\ var(S) &= np(1-p) \end{aligned} \tag{8}$$

with p as the probability of success (i.e., mutation), n as the number of independent trials (i.e., length of the genome in base pairs), and k as the number of successes (i.e., number of occurred mutations). The IMD now reduces to geometric random variable T, with PMF, expectation, and variance:

$$\begin{aligned} p_T(t) &= (1-p)^{-1} p \\ \mathbf{E}(T) &= \tfrac{1}{p} \\ var(T) &= \tfrac{1-p}{p^2} \end{aligned} \tag{9}$$

The genomic start location of a kataegis locus was sampled as an independent Bernoulli trial. The genomic end location of a kataegis locus was calculated using

$$end_i = start_i + \mathbf{E}(T)_i (k_i + 1) - 1 \tag{10}$$

Synthetic dataset description

The synthetic data consist of 1,024 samples with a total of 21,299,360 SNVs (Table 2). All mutations were generated on chromosome 1 on the human reference genome hg19. These samples were generated such that 8 different TMB classes (0.1, 0.5, 1, 5, 10, 50, 100, 500) were considered.

$$TMB = \frac{total\ number\ of\ variants\ in\ sample}{length\ of\ genome\ in\ bp} * 10^6 \tag{11}$$

For each TMB class, a sample was generated for all combinations of the following parameters: the number of kataegis loci (1, 2, 3, 5), the number of variants within each kataegis loci (6, 10, 25, 50), and the expected IMD of the variants in kataegis loci (100, 250, 500, 750). This resulted in 64 kataegis samples per TMB class. To balance the dataset, 64 samples without kataegis loci were generated for each TMB class. The synthetic dataset contained 1,232 kataegis loci and 33,245 variants within kataegis loci.

WGS dataset description

The WGS dataset (as used in this study; Table 3) is publicly available in .txt format [2]. This dataset contained 7,042 primary cancer samples from 30 different tissues, of which 507 originated from WGS and 6,535 from whole-exome sequencing (WES). Only the WGS samples ($n = 507$) were originally labeled using a piecewise constant fit (PCF) model and manually curated for kataegis presence (or absence) by the original study. Only the respective WGS samples, with a total of 3,382,751 SNVs, were reinterrogated within our performance evaluation. Additionally, we binned this dataset into 3 TMB classes (low: TMB <0.1, middle: 0.1 ≥ TMB < 10, high: TMB ≥10) and filtered it such that it only contained SNVs.

Preprocessing and parameter settings of alternative kataegis detection packages

Both the synthetic and the Alexandrov et al. [2] datasets were converted to MAF format for use in MafTools [10] ClusteredMutations [11], and kataegis [12] and to BED format for use in SeqKat [13]. All other parameter settings for MafTools, kataegis, ClusteredMutations, and SeqKat were set to the default values as specified in their respective manuals and vignettes.

For SigProfilerClusters [14], both the synthetic and the Alexandrov et al. [2] datasets were converted to a .txt file with column names as specified in the manual of SigProfilerClusters. We set the following parameters for SigProfilerSimulator(): genome="GRCh37," contexts = ["288"], simulations=100, overlap=True. For subsequent cluster detection, we set the following parameters for SigProfilerClusters.analysis(): genome="GRCh37," contexts="96," simContext=["288"], analysis="all," sortSims=True, subClassify=True, correction=True, calculateIMD=True, max_cpu=4, includedVAFs=False.

From the output of SigProfilerClusters, we selected the class 2 (kataegis) clusters for further analysis. The definition of kataegis used by SigProfilerClusters differs from the one used in our performance evaluation. SigProfilerClusters defines kataegis as a cluster of ≥4 genomic variants, of which the mean IMD is statistically different from the sample specific IMD cutoff. To include SigProfilerClusters in our performance evaluation, we only selected clusters detected by SigProfilerClusters that fit the definition of kataegis we used for the performance evaluation (i.e., a kataegis locus contains ≥6 genomic variants with a mean IMD ≤1,000 bp).

Table 2: Descriptive statistics of synthetic dataset

TMB class (no. of background mutations)	No. of samples (with kataegis)	No. of kataegis loci	No. of variants in kataegis loci
0.1 (25)	128 (64)	176	4,005
0.5 (125)	128 (64)	176	4,006
1 (249)	128 (64)	176	4006
5 (1,246)	128 (64)	176	4,014
10 (2,493)	128 (64)	176	4,029
50 (12,463)	128 (64)	176	4,077
100 (24,925)	128 (64)	176	4,183
500 (124,625)	128 (64)	176	4,925

Showing, per tumor mutational burden (TMB) class: TMB, number of generated background mutations per sample, the total number of samples, total number of samples with kataegis, total number of kataegis loci, and total number of variants within a kataegis loci of 1,024 synthetic samples.

Table 3: Descriptive statistics of the WGS dataset

TMB class	No. of samples (with kataegis)	No. of kataegis loci	No. of variants in kataegis loci
Low: TMB <0.1	301 (45)	93	946
Middle: 0.1 ≥ TMB < 10	186 (89)	444	5,058
High: TMB ≥10	20 (18)	336	3,107

Showing, per tumor mutational burden (TMB) class: TMB range, the total number of samples, total number of samples with kataegis, total number of kataegis loci, and total number of variants within a kataegis loci of 507 whole-genome sequenced (WGS) samples labeled by Alexandrov et al. [2].

Discussion

Here, we described Katdetectr, an R/Bioconductor package for the detection, characterization, and visualization of kataegis in genomic variant data by utilizing unsupervised changepoint analysis.

First, we tested 4 search algorithms for changepoint analysis, which revealed that the PELT [15] algorithm outperformed the BinSeg [16], SegNeigh [17], and AMOC algorithms in terms of prediction accuracy and computational feasibility. The BinSeg algorithm performed reasonably well, but it underfitted the data, which resulted in many false negatives. The SegNeigh algorithm performed well on samples with a TMB <5; however, this algorithm is computationally expensive, as it scales exponentially with the size of the data and cannot reasonably be used for the analysis of samples with a TMB >10. Unsurprisingly, the AMOC algorithm cannot detect kataegis as a kataegis locus is generally defined by 2 changepoints.

Besides testing search algorithms, we benchmarked Katdetectr using PELT and 5 publicly available kataegis detection packages that were recently published and used for supporting kataegis research [2, 5, 14, 15, 19]. Since no consensus benchmark was available, we aimed to get insight into the performance of these tools. The complexity of kataegis detection is to separate genomic regions of higher-than-expected mutational density from the background of somatic mutations. Therefore, we argued that generating a synthetic dataset containing samples of varying TMB (0.1–500) would provide a good measure for algorithmic solvability of the kataegis detection problem. Benchmarking on this synthetic dataset revealed that the accuracy of kataegis detection for all evaluated packages drops when the TMB increases. Performance evaluation per TMB-binned class revealed that Katdetectr is on par with alternative packages for samples with low or middle TMB. However, in contrast to alternative packages, Katdetectr remained robust when analyzing samples with a high TMB. This could be an important feature when analyzing late-stage (metastatic) malignancies or malignancies with a known predisposition of acquiring many somatic mutations such as skin or lung malignancies [20].

Additionally, the computation times of Katdetectr are feasible for samples with a TMB ranging from 0.1 to 500 as PELT scales linearly with the size of the data [15]. This shows that kataegis detection using Katdetectr is feasible on reasonably modern computer hardware.

The presented performance evaluation depends on the truth labels provided by the datasets. Both the synthetic and the WGS datasets have their limitations. We constructed the synthetic dataset by modeling mutations on a genome as a Bernoulli process, which is a common approach for modeling events that occur in a sequence. However, we did not incorporate prior biological knowledge in the synthetic dataset generation. Both SeqKat and SigProfilerClusters incorporate biological assumptions regarding kataegis (e.g., mutation context), which possibly negatively influenced their performance regarding the synthetic dataset. Additionally, the distance between events generated by a Bernoulli process is a geometric random variable. For a large n, which is the case for a human genome, a geometric random variable approximates an exponential random variable. Since we constrain Katdetectr to only fit exponential distributions, it is unsurprising that Katdetectr performs well on the synthetic dataset. Nevertheless, MafTools, ClusteredMutations, SeqKat, and SigProfilerClusters are less robust when analyzing the synthetic samples with a TMB of 100 and 500 as they classify many false-positive kataegis loci.

In addition to the synthetic dataset, we used the *a priori* labeled pan-cancer WGS dataset from the groundbreaking work of Alexandrov et al. [2] to evaluate the kataegis detection tools. However, the field of kataegis has grown and evolved since the publication of this dataset. Therefore, we want to emphasize that this dataset should not be considered an unequivocal truth, and the performance metrics should not be taken at face value. The annotation of this dataset likely contains several false positives and false negatives, as highlighted by the concordant discovery of 102 additional kataegis loci by several packages. Nevertheless, we

believe that the current benchmarking results give insight into the behavior of the evaluated packages regarding kataegis classification in samples with varying TMB. Additionally, the dataset published by Alexandrov et al. [2] and the predictions by all tools evaluated here are publicly available, which facilitates benchmarking of future endeavors regarding kataegis loci detection methods.

Our benchmarking showed that, for the WGS dataset, Katdetectr, MafTools, ClusteredMutations, and, SigProfilerClusters have a high concordance in classifying a whole sample as kataegis positive or negative. However, when concerning distinct kataegis loci, we observed more differences. ClusteredMutations reported the overall largest number of loci (n = 2,360), indicating it has the highest sensitivity. Conversely, kataegis (n = 8) and SeqKat (n = 528) reported the overall smallest number of loci, which we deem too small based on visual inspection. The third smallest number of kataegis loci is reported by SigProfilerClusters (n = 764), indicating it has the highest specificity. Katdetectr appears to balance sensitivity and specificity as it only detects kataegis loci detected by 1 or more alternative packages (n = 1,050).

We have sought to test the performance of all alternative tools utilizing their hard-coded or otherwise suggested default settings as mentioned by the authors in their respective manuscripts or manuals. Katdetectr was likewise performed with its default settings as described within this article. We have not performed additional parameter sweeps for the alternative packages as we argue that the default settings will be used by the majority of users. We therefore cannot discard that fine-tuning the parameters would have had an influence on our performance evaluation.

Kataegis is the most commonly used term for local hypermutations and has historically been defined as a cluster of at least 6 variants, of which the mean IMD is less than or equal to 1,000 base pairs [1, 16]. However, this definition has been altered recently, making the formal definition of kataegis ambiguous [2, 4, 5, 14]. For instance, another type of clustered mutations is called Omikli, which refers to clusters smaller than kataegis, generally containing 3 or 4 variants [7]. Although different types of clustered variants can be detected using Katdetectr by supplying the correct parameters, we only evaluated Katdetectr for the detection of kataegis.

We made Katdetectr publicly available on the Bioconductor platform, which requires peer-reviewed open-source software and high standards regarding development, documentation, and unit testing. Furthermore, Bioconductor ensures reliability and operability on common operating systems (Windows, macOS, and Linux). We designed Katdetectr to fit well in the Bioconductor ecosystem by incorporating common Bioconductor object classes. This allows Katdetectr to be used reciprocally with the plethora of statistical software packages available in Bioconductor for preprocessing and subsequent analysis. Lastly, we implemented Katdetectr flexibly, allowing Katdetectr to be used in an *ad hoc* manner for quick assessment of clustered variants and extensive research of the mutation rates across a tumor genome.

Conclusion

Katdetectr is a free, open-source R package available on Bioconductor that contains a suite for the detection, characterization, and visualization of kataegis. Katdetectr employs the PELT search algorithm for unsupervised changepoint analysis, resulting in robust and fast kataegis detection. Additionally, Katdetectr has been implemented in a flexible manner, which allows Katdetectr to expand in the field of kataegis. Katdetectr is available on Bioconductor [21] and on GitHub [22].

Availability of Supporting Source Code and Requirements

- Project name: Katdetectr
- RRID: SCR_023506
- BiotoolsID: katdetectr
- Workflowhub: 10.48546/workflowhub.workflow.463.1
- Project homepage:
- https://bioconductor.org/packages/release/bioc/html/katdetectr.html
- https://github.com/ErasmusMC-CCBC/katdetectr
- Operating system(s): Platform independent
- Programming language: R (>= 4.2)
- Other requirements: BiocParallel (>= 1.26.2), changepoint (>= 2.2.3), checkmate (>= 2.0.0), dplyr (>= 1.0.8), GenomicRanges (>= 1.44.0), GenomeInfoDb (>= 1.28.4), IRanges (>= 2.26.0), maftools (>= 2.10.5), methods (>= 4.1.3), rlang (>= 1.0.2), S4Vectors (>= 0.30.2), tibble (>= 3.1.6), VariantAnnotation (>= 1.38.0), Biobase (>= 2.54.0), Rdpack (>= 2.3.1), ggplot2 (>= 3.3.5), tidyr (>= 1.2.0), BSgenome (>= 1.62.0), ggtext (>= 0.1.1), BSgenome.Hsapiens.UCSC.hg19 (>= 1.4.3), BSgenome.Hsapiens.UCSC.hg38 (>= 1.4.4), plyranges (>= 1.17.0)
- License: GPL-3
- Project name: Evaluation of Katdetectr and alternative kataegis detection packages
- Workflowhub: 10.48546/workflowhub.workflow.500.1
- Project homepage: https://github.com/ErasmusMC-CCBC/evaluation_katdetectr
- Operating system(s): Platform independent
- Programming language: R (>= 4.2)
- Other requirements: katdetectr (1.1.2), MafTools (2.13.0), ClusteredMutations (1.0.1), kataegis (0.99.2), SeqKat (0.0.8), SigProfilerClusters (1.0.11), dplyr (1.0.10), tidyr (1.2.1), ggplot2 (3.4.0), variantAnnotation (1.44.0), mltools (0.3.5)
- License: GPL-3

Data Availability

All data used in the performance evaluation can be found on Zenodo [27]. All supporting data and materials are available in the *GigaScience* GigaDB database [28].

Additional Files

Supplemental Fig. S1. Heatmap showing performance of kataegis detection packages on synthetic data. Accuracy, normalized Matthews correlation coefficient (nMCC), F1 score, true-positive rate (TPR), and true-negative rate (TNR) for each of the tumor mutational burden (TMB) classes.

Supplemental Fig. S2. Violin plots with individual data points representing the per sample runtimes of katdetectr using different search algorithms on the synthetic dataset. Boxplots were sorted in ascending order based on mean runtime (depicted in text below boxplot).

Supplemental Fig. S3. Heatmap showing performance of kataegis detection packages on WGS data. Accuracy, normalized Matthews correlation coefficient (nMCC), F1 score, true-positive rate (TPR), and true negative rate (TNR) for each of the tumor mutational burden (TMB) classes.

Supplementary Table S1. Confusion matrix for the synthetic dataset.

Supplementary Table S2. Confusion matrix for the WGS dataset.

supplementary_material_vignette_general_overview

Abbreviations

AMOC: at most one change; bp: base pair; BinSeg: binary segmentation; IMD: intermutation distance; MAF: mutation annotation format; MNV: multinucleotide variant; nMCC: normalized Matthews correlation coefficient; PCF: piecewise constant fit; PELT: pruned exact linear time; SNV: single-nucleotide variant; SegNeigh: segment neighborhoods; TMB: tumor mutational burden; TNR: true-negative rate; TPR: true-positive rate; VCF: variant calling format; WES: whole-exome sequencing; WGS: whole-genome sequencing.

Competing interests

The authors declare no competing interests.

Funding

This research received funding from the Daniel den Hoed Fonds—Cancer Computational Biology Center (DDHF-CCBC) grant.

Authors' Contributions

D.M.H.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft. J.v.R.: conceptualization, methodology, investigation, software, visualization, writing—review & editing. Y.H.: conceptualization, methodology, software, writing—review & editing. H.J.G.v.d.W.: conceptualization, funding acquisition, investigation, methodology, project administration, resources, supervision, writing—review & editing.

Acknowledgments

We thank Martijn Lolkema, John Martens, Marcel Smid, Guido Jenster, and Stavros Makrodimitris for their discussions, input, and support. Additionally, we thank Coen Berns and Yi Ping for their initial efforts in detecting kataegis.

References

1. Nik-Zainal S, Alexandrov LB, Wedge DC, et al. Mutational processes molding the genomes of 21 breast cancers. Cell 2012;149(5):979–93. https://doi.org/10.1016/j.cell.2012.04.024.
2. Alexandrov LB, Nik-Zainal S, Wedge DC, et al. Signatures of mutational processes in human cancer. Nature 2013;500(7463):415–21. https://doi.org/10.1038/nature12477.
3. Campbell PJ, Getz G, Korbel JO, et al. Pan-cancer analysis of whole genomes. Nature 2020;578(7793):82–93. https://doi.org/10.1038/s41586-020-1969-6.
4. Alexandrov LB, Kim J, Haradhvala NJ, et al. The repertoire of mutational signatures in human cancer. Nature 2020;578(7793):94–101. https://doi.org/10.1038/s41586-020-1943-3.
5. Bergstrom EN, Luebeck J, Petljak M, et al. Mapping clustered mutations in cancer reveals APOBEC3 mutagenesis of ecDNA. Nature 2022;602(7897):510–7. https://doi.org/10.1038/s41586-022-04398-6.
6. Burns MB, Lackey L, Carpenter MA, et al. APOBEC3B is an enzymatic source of mutation in breast cancer. Nature 2013;494(7437):366–70. https://doi.org/10.1038/nature11881.
7. Mas-Ponte D, Supek F. DNA mismatch repair promotes APOBEC3-mediated diffuse hypermutation in human cancers. Nat Genet 2020;52(9):958–68. https://doi.org/10.1038/s41588-020-0674-6.
8. Lee S-Y, Wang H, Cho HJ, et al. The shaping of cancer genomes with the regional impact of mutation processes. Exp Mol Med 2022;54(7):1049–60. https://doi.org/10.1038/s12276-022-00808-x.
9. Roberts SA, Sterling J, Thompson C, et al. Clustered mutations in yeast and in Human cancers can arise from damaged long single-strand DNA regions. Mol Cell 2012;46(4):424–35. https://doi.org/10.1016/j.molcel.2012.03.030.
10. Mayakonda A, Lin D-C, Assenov Y, et al. Maftools: efficient and comprehensive analysis of somatic variants in cancer. Genome Res 2018;28(11):1747–56. https://doi.org/10.1101/gr.239244.118.
11. Lora D. ClusteredMutations: location and visualization of clustered somatic mutations. 2016. https://CRAN.R-project.org/package=ClusteredMutations. Accessed 28 November 2022.
12. Lin X, Hua Y, Gu S, et al. kataegis: an R package for identification and visualization of the genomic localized hypermutation regions using high-throughput sequencing. BMC Genomics 2021;22(1):440. https://doi.org/10.1186/s12864-021-07696-x.
13. Yousif F, Lin X, Fan F, et al. SeqKat: detection of kataegis. 2020. https://CRAN.R-project.org/package=SeqKat. Accessed 28 November 2022.
14. Bergstrom EN, Kundu M, Tbeileh N, et al. Examining clustered somatic mutations with SigProfilerClusters. Bioinformatics 2022;38(13):3470–3. https://doi.org/10.1093/bioinformatics/btac335.
15. Killick R, Fearnhead P, Eckley IA. Optimal detection of changepoints with a linear computational cost. J Am Statist Assoc 2012;107(500):1590–8. https://doi.org/10.1080/01621459.2012.737745.
16. Scott AJ, Knott M. A cluster analysis method for grouping means in the analysis of variance. Biometrics 1974;30(3):507–12. https://doi.org/10.2307/2529204.
17. Auger IE, Lawrence CE. Algorithms for the optimal identification of segment neighborhoods. Bull Math Biol 1989;51(1):39–54. https://doi.org/10.1016/S0092-8240(89)80047-3.
18. Selenica P, Marra A, Choudhury NJ, et al. APOBEC mutagenesis, kataegis, chromothripsis in EGFR-mutant osimertinib-resistant lung adenocarcinomas. Ann Oncol 2022;33(12):1284–95. https://doi.org/10.1016/j.annonc.2022.09.151.
19. Stenman A, Yang M, Paulsson JO, et al. Pan-genomic sequencing reveals actionable CDKN2A/2B deletions and kataegis in anaplastic thyroid carcinoma. Cancers 2021;13(24):6340. https://doi.org/10.3390/cancers13246340.
20. Priestley P, Baber J, Lolkema MP, et al. Pan-cancer whole-genome analyses of metastatic solid tumours. Nature 2019;575(7781):210–6. https://doi.org/10.1038/s41586-019-1689-y.
21. Hazelaar DM, van Riet J. Characterization and visualization of Kataegis in sequencing data. R package version 1.2.0. https://doi.org/10.18129/B9.bioc.katdetectr. Accessed 14 February 2023.
22. Van Riet J, Hazelaar D. ErasmusMC-CCBC/evaluation_katdetectr: publication. *Zenodo*. 2023. https://doi.org/10.5281/ZENODO.8328463.
23. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing; 2022. https://www.R-project.org/.
24. Killick R, Eckley I. changepoint:an R package for changepoint analysis. J Stat Soft 2014;58(3): 3. https://doi.org/10.18637/jss.v058.i03.

25. Killick R, Haynes K, Eckley IA. changepoint: an R Package for changepoint Analysis Software Reference. 2022. https://CRAN.R-project.org/package=changepoint. Accessed 14 March 2022.
26. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 2020;21(1):6. https://doi.org/10.1186/s12864-019-6413-7.
27. Hazelaar D, van Riet J, van de Werken H. Datasets used for the performance evaluation of kataegis detection tools. *Zenodo*. 2022. https://doi.org/10.5281/ZENODO.8046959.
28. Hazelaar DM, van Riet J, Hoogstrate Y, et al. Supporting data for "Katdetectr: An R/bioconductor Package Utilizing Unsupervised Changepoint Analysis for Robust Kataegis Detection." GigaScience Database. 2023. https://doi.org/10.5524/102445.

Received: February 24, 2023. **Revised:** June 15, 2023. **Accepted:** September 12, 2023