BMC Bioinformatics

RESEARCH Open Access

An e□ective heuristic for developing hybrid feature selection in high dimensional and low sample size datasets

Hyunseok Shin[1] and Sejong Oh[2*]

*Correspondence:
sejongoh@dankook.ac.kr

[1] Department of Computer Science, Dankook University, Youngin, Gyeonggi, South Korea
[2] Department of Software Science, Dankook University, Youngin, Gyeonggi, South Korea

Abstract

Background: High-dimensional datasets with low sample sizes (HDLSS) are pivotal in the fields of biology and bioinformatics. One of core objective of HDLSS is to select most informative features and discarding redundant or irrelevant features. This is particularly crucial in bioinformatics, where accurate feature (gene) selection can lead to breakthroughs in drug development and provide insights into disease diagnostics. Despite its importance, identifying optimal features is still a significant challenge in HDLSS.

Results: To address this challenge, we propose an e□ective feature selection method that combines gradual permutation filtering with a heuristic tribrid search strategy, specifically tailored for HDLSS contexts. The proposed method considers inter-feature interactions and leverages feature rankings during the search process. In addition, a new performance metric for the HDLSS that evaluates both the number and quality of selected features is suggested. Through the comparison of the benchmark dataset with existing methods, the proposed method reduced the average number of selected features from 37.8 to 5.5 and improved the performance of the prediction model, based on the selected features, from 0.855 to 0.927.

Conclusions: The proposed method e□ectively selects a small number of important features and achieves high prediction performance.

Keywords: HDLSS, Feature selection, Machine learning, Filter method, Wrapper method

Background

In the current data era, information is recorded intricately across various dimensions. Consequently, the dimensionality of the data increases rapidly. High-dimensional datasets refer to data where the number of features (p) is considerably larger than the sample size (n). High-dimensional datasets have driven significant scientific discoveries in fields such as biology and bioinformatics by revealing previously unknown complex patterns. However, analyzing and utilizing high dimensional and low sample size (HDLSS) datasets remain a challenge for researchers [1–6].

Microarray data are prime examples of HDLSS datasets. With the capacity to capture the expression of tens of thousands of genes, the core aim is to identify genetic variations in specific cancers or other genetically-related diseases [3]. However, a notable proportion of these genes either lacks direct relevance to the disease or exhibits substantial overlap in feature information, leading to redundancy [7]. Furthermore, owing to the patient-centric nature of these data, sample size was inherently limited. ese aspects amplify risks such as overfitting and the curse of dimensionality [3, 4]. Consequently, developing cancer classification models and analyzing HDLSS datasets such as microarrays, pose intricate challenges for computer science researchers [7].

A fundamental approach for addressing these challenges in machine learning is dimensionality reduction via feature selection. e primary objective of feature selection is to filter out irrelevant and redundant features, thereby honing the most pertinent features relevant to the subject of interest. is enhances the clarity, generalizability, and predictive accuracy of classification models, concurrently reducing computational demands and boosting e ciency [1]. In bioinformatics, specifically, feature selection plays a critical role in uncovering potential avenues for drug development and o ering insights into disease diagnostics and causation [1–3, 8].

Identifying the optimal features is an NP-hard problem [3, 6, 8, 9]. Numerous feature-selection methods have been proposed to address this challenge. ese methods are typically classified into three categories based on their interactions with the classification model: filter, wrapper, and embedded methods [3, 4]. e filter method takes a model-independent stance, assigns rankings to features based on metrics such as statistical or information-theoretic properties, and subsequently selects those that exceed a predetermined threshold. In contrast, the wrapper method incorporates a model-dependent approach, by selecting feature subsets based on the performance of a specific model. e embedded method di ers from the wrapper method by integrating feature selection directly into the model-training process.

Recent advancements in feature selection include methods based on graphs and deep learning frameworks. ese methods aim to capture the intricate relationships among features. In the graph-based approach, features are visualized as nodes and their interrelationships as edges. is method determines the importance of features by analyzing the structural properties of a graph [6, 10–12]. Deep learning techniques for feature selection utilize neural networks with features as inputs [6–9, 12–14]. e significance of each feature is gauged by the magnitude of the gradients during network training.

For HDLSS data, several studies on feature selection have adopted a hybrid approach involving two stages, filtering and searching, as illustrated on the upper side of Fig. 1 [7]. is approach seeks to harness the computational e ciency of filter methods and the high performance of wrapper methods. e filter method can be viewed as a preprocessing step that significantly reduces the search scope by eliminating unnecessary features. By contrast, the wrapper method refines the search within this narrow scope, aiming to identify the most optimal features more precisely [3, 7].

From the collective insights of previous studies on feature selection, we can draw the following conclusions:

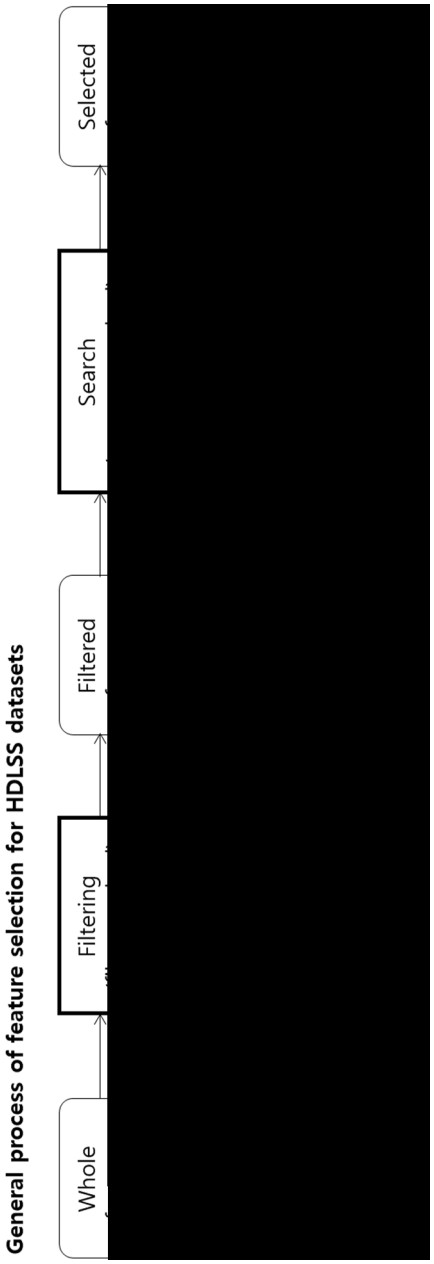


Fig. 1 General process of feature selection for HDLSS datasets and improved points with the proposed method

1. Understanding intricate relationships, such as the interactions between features, is pivotal. Previous methods often overlooked this aspect. For example, filter methods assume that each feature is independent and does not interact with any other feature.
2. Finding the optimal features for HDLSS is realistically di cult. An alternative is to find a better solution among several suboptimal features. erefore, avoiding local optima is essential, as wrapper methods are particularly prone to this issue.
3. ese two objectivess can be achieved by balancing the diversity and focus in the feature search space, introducing randomness during the search process, and employing a hybrid method that leverages both filter and wrapper techniques.

In this study, we propose a new metaheuristic method [15–19] to improve upon previous feature selection approaches for HDLSS datasets, aiming to achieve near-optimal performance with minimal features. By integrating gradual permutation filtering with a diverse search strategy, we designed our approach based on core principles. e key highlights of our method are as follows:

1. Filtering: Leveraging permutation importance, filtering accounts for feature interactions and establishes a correct threshold.
2. Search: rough "consolation matches," the nesting e ect can be overcome, wherein a once-selected (or excluded) feature remains unaltered. Additionally, by varying "first-choice features," the search range is expanded to approach a more global optimum.
3. Runtime optimization: Using ranked features, the method e ciently reduces the search space.
4. Assessing the fitness of the selected features: We crafted a unique performance metric for the HDLSS, providing a holistic perspective on feature quantity and quality.

e details are outlined in the next section.

Methods

Overall procedure

e proposed method also follows the general procedure of feature selection for HDLSS datasets. As illustrated on the lower side of Fig. 1 and elaborated in Algorithm 1, the proposed method comprises two primary phases: filtering and searching.

1. Gradual Permutation Filtering (GPF): is phase ingests all the HDLSS data features. It ranks features based on their permutation importance and subsequently eliminates irrelevant features (Algorithm 2).
2. Heuristic Tribrid Search (HTS): By leveraging the features ranked by the GPF, HTS employs a heuristic blend of forward search, consolation match, and backward elimination. is approach aims to identify the near-optimal feature set by considering both the number of features and classification performance using the features (see Algorithms 3–6).

e subsequent sections will delve into the specifics of each phase.

Algorithm 1 Overall procedure

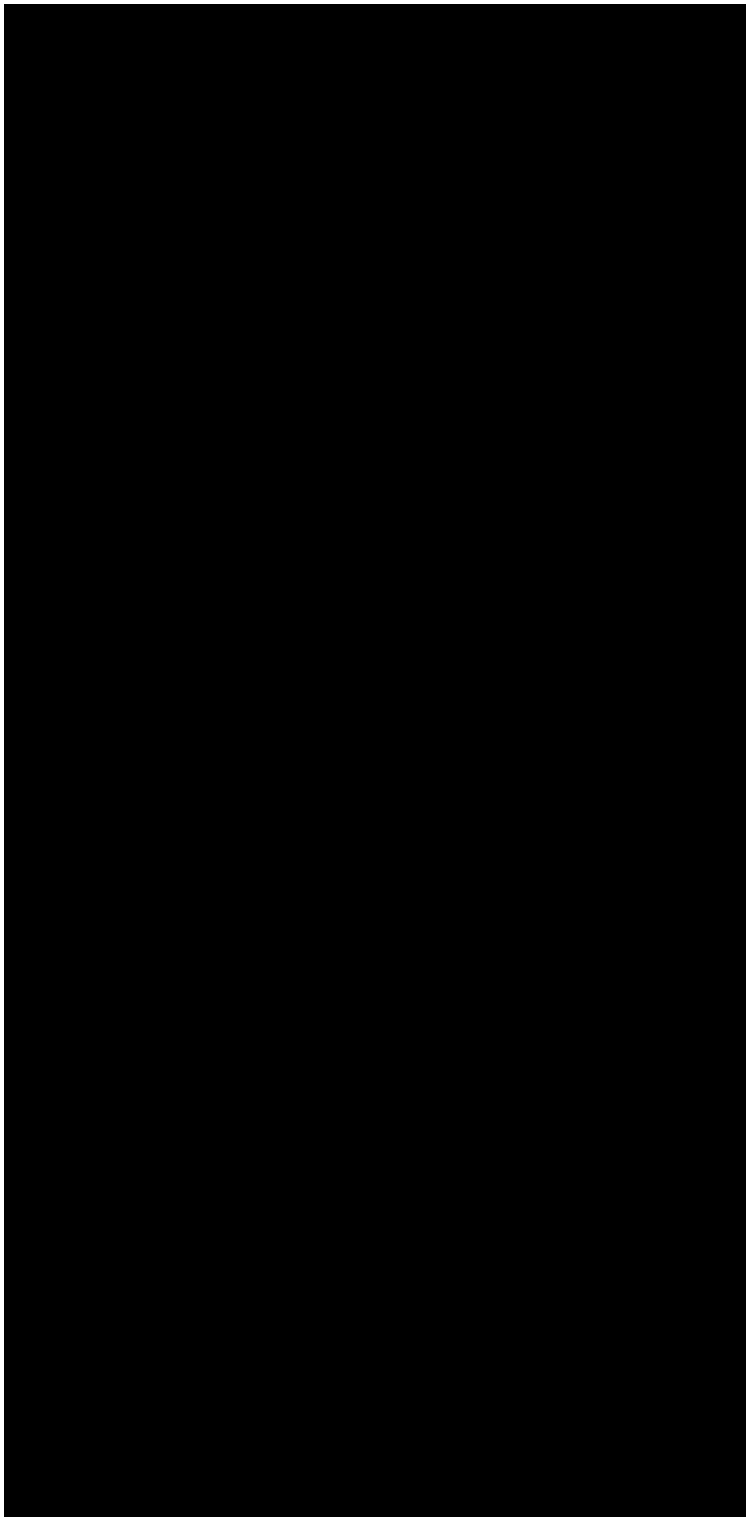


Table 1 Comparison of two measures, LRR and LCM

Number of selected features	Performance (AUC)	LRR	LCM
5	0.852	0.811	0.850
10	0.858	0.730	0.851
25	0.859	0.622	0.845
50	0.860	0.541	0.842

Gradual permutation filtering

The GPF stage prioritizes features based on their permutation importance and eliminates unimportant features. This method chooses features with an importance value greater than zero because values below this threshold often indicate redundancy or suggest potential noise.

This method offers a refined approach for evaluating feature importance by adopting a gradual process of eliminating noise and recalculating the importance, thereby minimizing the potential biases associated with single-step elimination. Considering that the permutation importance is evaluated within the evaluation group (features), the approach emphasizes filtering out irrelevant features and reevaluating the importance of enhancing purity, which is defined solely based on performance-impacting features. Notably, a gradual feature filtering method, as opposed to filtering all at once, provides a more precise selection of features with importance on the verge of 0.

To ensure robust feature selection, the GPF measures the permutation importance of each feature multiple times; i.e., 50 times in our case. In Algorithm 2, the constant variable M refers to the number of permutation test trials used for feature evaluation. Because permutation involves randomness, performing the test only once does not allow for an accurate assessment of feature importance. Based on our experiments, we determined that setting M to approximately 50 trials is appropriate. From these measurements, the features that exceed the importance of zero for a specific threshold number were selected. For the selected features, the permutation importance was recalculated by applying a progressively higher threshold each time. This iterative process, detailed in Algorithm 2, refines the ranking of significant features. The final ranking of the features was determined by averaging the last measured importance values of the features selected until the end.

Heuristic tribrid search

While the GPF is designed to filter out unimportant features, HTS focuses on identifying the most informative features from a pool refined by the GPF. HTS operates in three distinct stages, forward search, consolation match, and backward elimination, as outlined in Algorithm 3.

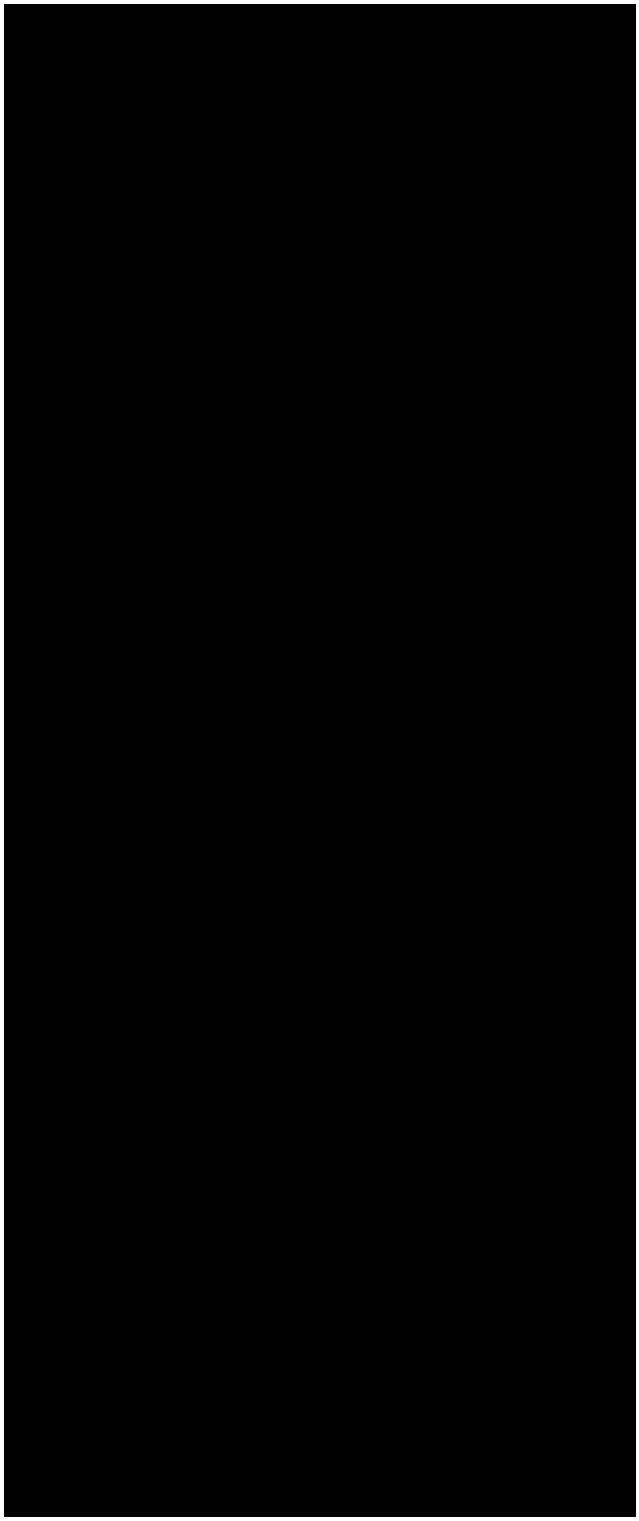
We slightly modified the original forward search technique. e original technique began with an empty feature set and determined its first feature. In contrast, the proposed method uses "first-choice feature" that is chosen from ranked feature list of GPF. is procedure incrementally adds to the feature set, guided by the performance increments detailed in Algorithm 4. e performance was measured using a classification metric based on the selected features.

When the performance increase caused by the selected features stops during the forward search, the process shifts to "consolation match." is stage aims to enhance the performance by swapping a single feature between selected and unselected feature pools, as depicted in Algorithm 5. is strategy o ers an escape route from potential local optima. Notably, the duration of this stage depends on the volume of features initially filtered by the GPF. Techniques for expediting this stage are explored in subsequent sections.

If the consolation match yields improved performance, the forward search resumes further enhancements. However, if no such gains are evident, HTS transitions to a backward elimination phase. is final stage discards any remaining unimportant features, thereby ensuring the relevance of the end feature set.

Importantly, at each stage of the HTS (Algorithms 4–6), the performance is gauged using a novel metric introduced in this study. is metric considers both the classification performance of the model and feature count. An elaboration of this metric is presented in the following section.

Algorithm 3 Heuristic_tribird_search (FILTERED, CL, i, N)



Algorithm 4 Forward_search (FILTERED, CL, SELECTED, max_fit, N)

```
Input:
  FILTERED - selected features sorted by ranking
  CL - class label vector
  SELECTED - selected feature list
  max_fit - maximum fit value
  N - Number of columns in original dataset
Output:
  SELECTED - selected feature list
  max_update - max fit update status
  max_fit - maximum fit value

DO
  max_update • FALSE
  CANDIDATE • {columns of FILTERED } – SELECTED

  FOR each feature F IN CANDIDATE DO
    TEMP • SELECTED ∪ {F}
    current_fit • fitness(CL, TEMP, N)
    IF current_fit > max_fit THEN
      max_fs • TEMP
      max_fit • current_fit
      max_update • TRUE
    END IF
  END FOR
  IF max_update THEN
    SELECTED • max_fs
  END IF
WHILE max_update

RETURN SELECTED, max_update, max_fit
```

Algorithm 5 Consolation_match (FILTERED, CL, SELECTED, max_fit, N)

```
Input:
   FILTERED - selected features sorted by ranking
   CL - class label vector
   SELECTED - selected feature list
   max_fit - maximum fit value
   N - Number of columns in original dataset
Output:
   max_fs - selected feature list with max fit value
   max_update - max fit update status
   max_fit - maximum fit value

max_update • FALSE
CANDIDATE • {columns of FILTERED } – SELECTED

// Iterate over each feature in SELECTED and CANDIDATE to find an optimal swap
FOR X IN SELECTED DO
 FOR Y IN CANDIDATE DO
    TEMP • (SELECTED - {X}) U {Y}
    current_fit • fitness(CL, TEMP, N)
 IF current_fit > max_fit THEN
    max_fs • TEMP
    max_fit • current_fit
    max_update • TRUE
 END IF
 END FOR
END FOR

RETURN max_fs, max_update, max_fit
```

Table 2 List of benchmark datasets

No	Dataset	Disease	Instances	Features	Dimensionality Index*	Balance Ratio**
1	ALLAML	Leukemia	72	7129	2.07	0.347
2	alon	Colon Cancer	62	2000	1.84	0.355
3	borovecki	Huntington	31	22,283	2.92	0.452
4	chiaretti	Leukemia	128	12,625	1.95	0.422
5	chin	Breast Cancer	118	22,215	2.1	0.364
6	chowdary	Breast Cancer	104	22,283	2.16	0.404
7	GLI_85	Gliomas	85	22,283	2.25	0.306
8	gordon	Lung Cancer	181	12,533	1.82	0.171
9	gravier	Breast Cancer	168	2905	1.56	0.339
10	pomeroy	CNS Tumor	60	7128	2.17	0.35
11	Prostate_GE	Prostate Cancer	102	5966	1.88	0.49
12	shipp	Lymphoma	77	7129	2.04	0.247
13	singh	Prostate Cancer	102	12,600	2.04	0.49
14	SMK_CAN_187	Lung cancer	187	19,993	1.89	0.481
15	subramanian	N/A	50	10,100	2.36	0.34
16	tian	Myeloma	173	12,625	1.83	0.208
17	west	Breast Cancer	49	7129	2.28	0.49
18	arcene		200	10,000	1.74	0.44
19	gisette		7000	5000	0.96	0.5
20	Hill_valley		1212	100	0.65	0.495
21	ionosphere		351	33	0.6	0.359
22	madelon		2600	500	0.79	0.5
23	sonar		208	60	0.77	0.466
24	wdbc		569	30	0.54	0.373

* Dimensionality index = log(*Numberoffeatures*)/log(*Numberofinstances*). It is a measure of how high-dimensional a given dataset is

** Balanced ratio: the proportion of the samples in lower class of the entire dataset. A value 0.5 is ideal for a binary class dataset

Log comprehensive metric for HDLSS datasets

Balancing the number of selected features with the classification performance is imperative when working with HDLSS datasets. erefore, this study proposes log-comprehensive metric (LCM). is metric, which is a refined iteration of the conventional fitness function, was meticulously designed for the HDLSS datasets, as elaborated in Algorithm 6.

Algorithm 6 Log_comprehensive_metric (CL, SELECTED, N)

Input:
CL - class label vector
SELECTED - selected feature list
N - Number of columns in original dataset
Output:
LCM - log comprehensive metric value

// Calculate the LCR metric using a weighted average of the LRR (Log Reduction Rate) derived from performance and selected feature count.
CONST C • 0.005 // C is trade-off constant for feature count and performance.
X • length of SELECTED// Number of selected features
Y • log(2)/log(N)

THETA • Y/(C+Y)
PERFORMANCE • the average performance obtained with 5-fold and XGBoost on SELECTED and CL
LRR • 1-log(X)/log(N)
LCM • THETA*PERFORMANCE + (1-THETA)*LRR

RETURN LCM

Table 3 List of benchmark feature selection methods

No	Feature selection method	Description
1	F-statistic	Filter method that measures the correlation between a feature and the class label using the F-test [23]
2	mRMR	Filter method that incorporates redundancy measurement [24]
3	Permutation importance	Filter method that considers feature interactions [25]
4	Infinite feature selection (INF)	Graph-based filtering method [10]
5	GRACES	Graph convolutional network-based feature selection [6]

e widely used fitness function is described by Eq. (1). It constitutes a weighted sum of the model error and the ratio of the selected features to the total. e aim is to minimize the following function:

$$fitness = \theta * Error + (1-\theta) * \left(\frac{Number\ of\ selected\ features}{Total\ number\ of\ features}\right), where\ 0 \leq \theta \leq 1 \tag{1}$$

By contrast, the LCM, expressed in Eq. (2), fuses the model's classification success with the log reduction rate (LRR) using weighted factors. e objective is to maximize.

$$LCM = \theta * Performance + (1-\theta) * LRR, where 0 \leq \theta \leq 1 \tag{2}$$

e parameter performance in Eq. (2) denotes the evaluation metric for the model used in the HTS stage, which is a variation of the wrapper method. In this study, AUC is used as the performance metric; however, it is not restricted to AUC. Other metrics, such as accuracy or sensitivity—both ranging between 0 and 1—can also be applied. Furthermore, the model for evaluating performance is not limited to XGBoost, o ering flexibility in the choice of algorithms. In Algorithm 6, performance values are computed as the average metric obtained from fivefold cross-validation.

LRR is further explained by Eq. (3). Its value fluctuates between 0 and 1 and approaches 1 as the number of selected features decreases.

$$LRR = 1 - \log_a b = 1 - \log b / \log a,$$

$$where: 0 \leq LRR \leq 1,$$

$$\mathbf{a} = \textbf{Total number of features}, \mathbf{b} = \textbf{Number of selected features} \tag{3}$$

In contrast to the reduction rate (RR), which is formulated *as RR=1—(Number of selected features / Total number of features)*, the advantage of LRR lies in its enhanced sensitivity, especially when the number of selected features is small.

Table 1 demonstrates the e ectiveness of the LCM measure. While the Performance (AUC) values are hypothetical, the LRR values are actual calculations based on the number of features, and the LCM values are derived from the Performance (AUC) and LRR. By focusing solely on the AUC performance, one might opt for the set with 50 features that presents the highest AUC. However, when evaluating based on the LCM, the most optimal feature set is that with only 10 features. is LCM-based selection results in a

minor AUC performance drop of 0.002, but with the benefit of reducing the selected feature count by 40.

Experimental procedure

Benchmark datasets

To evaluate the previous and proposed feature selection methods, we utilized 24 publicly available datasets designed for binary classification, as presented in Table 2. Of these, No. 1 to 17 are cancer-related microarray datasets. e features within these microarray datasets represent gene expression profiles, which are often referred to as probes. ese datasets were acquired from a widely-used R package, "datamicroarray" [20]. We also included additional datasets (No. 18 to 24) obtained from public databases [21, 22] to demonstrate that the proposed method is e ective for non-microarray or non- HDLSS data. No. 1 to 18 are HDLSS datasets, while No. 19 to 24 are not.

Compared feature selection methods and experimental environment

To assess the e cacy of the proposed method, we compared it with various established feature selection methods, as presented in Table 3.

Among these, the first four adopted a filter-based approach. For each technique, feature rankings were computed and the top 100 features were selected. In the case of GRACES, a fixed number of features were produced. Hence, to maintain parity with the number of features selected by the other methods, we set this count to 100 and used the resulting rankings. The rankings for both INF and GRACES were determined after parameter optimization using fivefold cross validation.

Direct comparisons between any filter method and the proposed approach are challenging because filter methods do not explicitly indicate a subset of features that yield optimal performance. Consequently, it is imperative to identify the optimal feature subsets for benchmarking methods. For this purpose, we employed two distinct strategies: 'simple sequential search' and 'forward search.' In simple sequential search, the performance of the classification model was evaluated at each step while increasing the number of features from 1 to 100 based on the ranking of each feature, and the subset of features with the highest performance was selected. Forward search is similar to simple sequential search, but it continues selecting good features until no further improvement in performance is observed.

To evaluate the quality of the selected features, we applied the eXtreme Gradient Boosting (XGB) classifier, equipped with specific parameters, nrounds = 5, objective = "multi:softmax", while retaining default configurations for the remaining parameters. Owing to its established potency and computational efficiency, XGB is an appropriate choice for evaluating feature selection methods.

For the performance evaluation, we incorporated two metrics: Area Under the Curve (AUC) and LCM. AUC is a useful measure for comparing the performance of prediction models, especially when the classes of the dataset are imbalanced. As explained previously, LCM is suitable for evaluating both the classification performance and the number of selected features in the HDLSS dataset. For our analysis, a θ value of 0.8 was adopted as the reference point. The choice of $\theta = 0.8$ was informed by a performance comparison between the proposed method and existing

methods across 24 datasets. Our method consistently outperformed Simple Sequential Search and Forward Search for θ values ranging from 0.7 to 0.9, with the optimal balance observed at 0.8. Figure 2 illustrates the overall process of the comparative experiments.

To orchestrate the efficacious experiments, we created two computational environments, R and Python, as shown in Table 4. Although the rankings for GRACE and INF were procured within a Python framework, the search and evaluation phases for all the techniques were uniformly executed in the R environment. The associated experimental code is available at https://bitldku.github.io/home/sw/heuristic_search.html.

Results

Number of selected features through GPF and HTS

e number of selected features is an important criterion for feature selection in the HDLSS datasets. e proposed method comprised two procedures for feature selection: GPF and HTS. e GPF filters useless features, and the HTS chooses informative features from GPF results. Table 5 summarizes the results of the GPF and HTS analyses. Details are provided in Supplementary Table S1. During the GPF process, approximately 98.2% of the total features were eliminated from the 18 HDLSS datasets. Conversely, in the six non-HDLSS datasets, the average feature reduction rate was 61.2%, indicating considerable variability among the datasets. Notably, datasets with fewer than 100 features were not filtered. In the HTS phase, an additional 97.6% of the previously reduced features in the HDLSS dataset were excluded. e non-HDLSS datasets also exhibited a significant elimination rate, with an average of 98.3%. In conclusion, after the integrated application of both the GPF and HTS methodologies, the HDLSS datasets consistently selected 12 or fewer features, representing an average of merely 0.07% of the initial feature set.

Performance comparison based on simple sequential search

In this section, we compare the classification performance the previous and proposed methods using the selected features. To determine the performance of the filter method, a simple sequential search was performed as described in the "Compared feature selection methods" section. AUC and LCM were used as performance criteria. A summary of the comparison results is presented in Table 6 and Fig. 3. Details are provided in Supplementary Table S2 and S3.

When compared using the AUC performance metric, the proposed method exhibited equivalent or superior performance in 16 out of 24 datasets, while maintaining the lowest performance variability among all methods, as detailed in Supplementary Table S2 and S3. e AUC of our method exceeded that of comparative approaches, with margins ranging from 2.2 to 13.1%. However, the number of selected features was markedly reduced, averaging only 5.5 features, which is between 1/4 and 1/10 that of the other methods selected.

Utilizing the LCM performance metric with a θ value of 0.8, our method displayed enhanced performance in 20 out of 24 datasets, representing 83.3%. is indicates

Fig. 2 Overall procedure of the comparative experiment

Table 4 Experimental environment

Task	Hardware	OS	Language	Package (Library)
- F-statistics - mRMR -Permutation Importance - Proposed Performance evaluation	Processor:AMD Ryzen 9 5900X 12-Core RAM:16GB, GPU:RTX3090	Windows 11	R 4.3.1	mlr 2.19.1 ranger 0.15.1 mRMRe 2.1.2.1 Parallel (base) xgboost 1.7.5.1 caret 6.0–94
-GRACES -INF	Processor:Intel i7=10700 16-Core RAM:32.0GB GPU:RTX3080	Ubuntu 20.04.6 LTS	Python 3.8.10	pytorch 2.0.1 torch_geometric 2.3 sklearn 1.2.2 xgboost 1.7.6 numpy 1.24.3 pandas 2.0.2 scipy 1.10.1 INF(https://pypi.org/project/PyIFS/) GRACES(https://github.com/canc1993/graces)

Table 5 Average number of features determined by proposed GPF and HTS procedures

Dataset	Original(A)	GPF(B)	HTS(C)	Reduction rate (1-B/A)	Selection rate (C/A)
HDLSS (1–18)	12,163.6	222.6	5.3	0.982	0.0004
Non-HDLSS (19–24)	954.8	370.8	6.3	0.612	0.0066

that when considering both the performance and influence of the number of selected features, the proposed method excels, highlighting its ability to achieve comparable or better results with fewer features.

Among the traditional methods, permutation exhibits the best performance, followed by F-statistics, mRMR, GRACES, and INF. Notably, GRACES and INF exhibited a tendency to select a greater number of features than the other methods. Furthermore, INF was unable to produce results for datasets exceeding 20,000 features, and GRACES also encountered di culties in generating results for specific datasets.

Performance comparison based on forward search

In this section, we compare the classification performances of previous methods and the proposed method. To determine the performance of the filter method, a forward and a simple sequential search was used as described in the “Compared feature selection methods” section. A summary of the results is presented in Table 7 and Fig. 4, and the complete experimental results are detailed in Supplementary Table S4 and S5.

Compared to the results in previous section, most filter methods showed an improvement in the AUC and significantly reduced the number of selected features.

is trend suggests that the top 100 features chosen using traditional filtering methods exhibit high redundancy. A forward search helped remove many overlapping features. In specific datasets subh as Alon, both the feature count and AUC decreased, which might indicate a fall into local optima.

Using the AUC metric, the proposed method matched or exceeded the performance of the other methods on 20 of the 24 datasets (83.3%). It also exhibited the lowest variability in performance. e AUC of the proposed method was between 2.2 and 11.1% higher than those of the other methods. Although the comparative methods selected fewer features on average than the sequence results, the proposed method consistently selected the least number of features. Furthermore, when using the LCM with θ at 0.8, the proposed method outperformed other methods in 21 of the 24 datasets (87.5%).

Execution time of the proposed method

Supplementary Table S6 lists the average runtimes of our method over 30 iterations for the HDLSS datasets. e Arcene dataset required approximately 12 min, the longest duration, whereas 16 out of the 18 datasets completed the feature selection within 5 min. Notably, the HTS phase constituted 77% of the total duration, with 94% spiking in some cases. e execution times for both GPF and HTS increased proportionally with instance size. Conversely, apart from SMK_CAN_187, the GPF duration remained relatively consistent irrespective of the feature count, whereas HTS showed a pronounced rise as the features increased. In essence, the duration of the HTS phase significantly shapes the total runtime, with the e ciency of the GPF in transmitting a reduced feature set being pivotal to the HTS timeframe. In conclusion, the average execution time of the proposed method is 169 s (2.8 min), which is reasonable.

Discussion and conclusions

e essence of feature selection lies in e ectively finding a subset that is close to the global optimum, where the relevance to the class is high and the redundancy between features is low. We proposed two methods, GPF for filtering and HTS for searching and verified their superior performance compared to those of other methods. Particularly, for microarray datasets with HDLSS properties, we selected a minimal subset of core features within a reasonable time. e advantages of the proposed method are summarized as follows:

Incorporating feature interactions for relevance measurement in the filtering stage: Oh [26] experimentally and theoretically proved that the importance of a feature can be decomposed into its intrinsic predictive power (feature power) and the e ect arising when combined with other features (interaction). e importance of a feature not only lies in its intrinsic power but also varies depending on the surrounding features.

Table 6 Comparison of the number of selected features, AUC, and LCM between the proposed method and previous works using simple sequential search

Method	Selected features	AUC	LCM
F-statistics (FS)	32.9 (±32.7)	0.891 (±0.092)	0.833 (±0.103)
mRMR (MR)	27.8 (±26.9)	0.874 (±0.116)	0.824 (±0.122)
Permutation (PE)	21.7 (±26.4)	0.905 (±0.089)	0.855 (±0.095)
INF	55.3 (±36.3)	0.796 (±0.125)	0.731 (±0.097)
GRACES (GE)	51.3 (±31.1)	0.816 (±0.122)	0.748 (±0.105)
Proposed (PRO)	5.5 (±3.3)	0.927 (±0.082)	0.900 (±0.079)

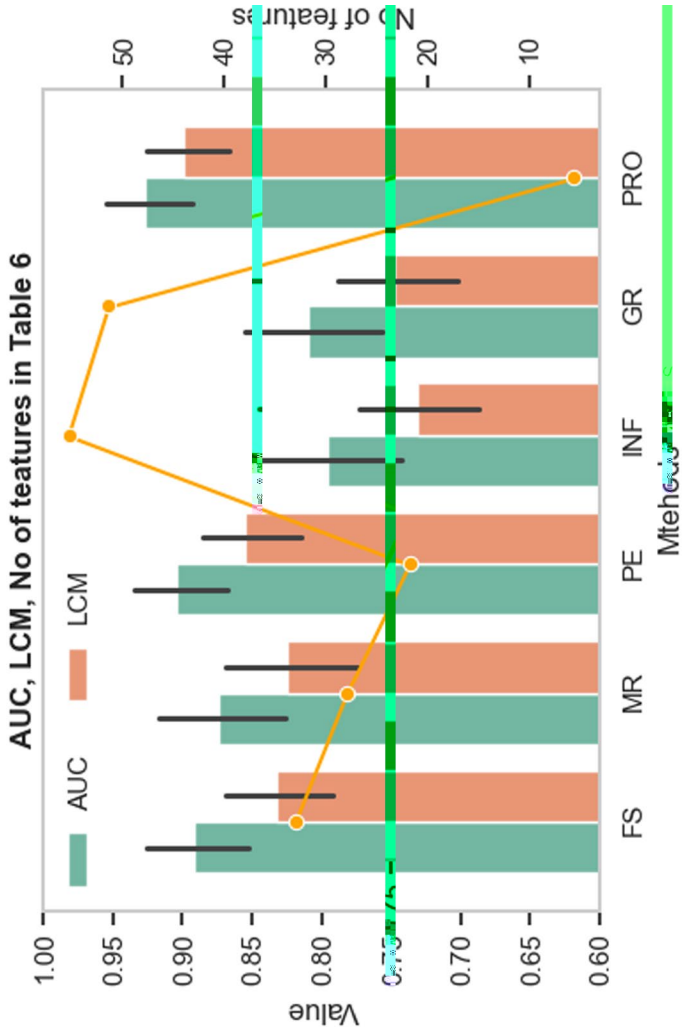


Fig. 3 Comparison result of AUC, LCM, and No of features in Table 6

erefore, the relevance between the subset and class can be evaluated more accurately by considering feature interactions. As we progressively filtered out features with low importance in the filtering stage and then measured their importance within the refined subset, we were able to accurately measure the interactions based on this subset. Although the permutation importance method we used does not explicitly distinguish between feature power and interaction, it is clear that the interaction e ect is reflected within the importance.

Elimination of redundancy among features during the search stage: While basic permutation importance can account for interactions between features, it does not capture redundancy. us, there is a possibility that highly redundant features exist within the filtered subset. By contrast, the proposed HTS can exclude redundancy among the fea-

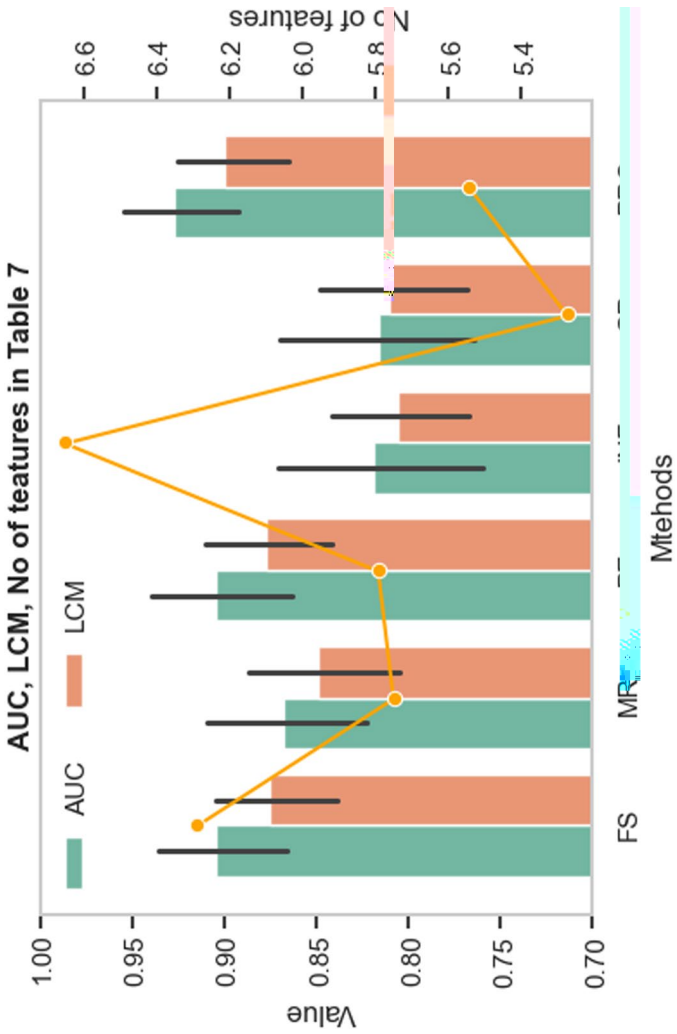


Fig. 4 Comparison result of AUC, LCM, and No of features in Table 7

Table 8 Studies demonstrating the association of selected genes with lymphoblastic leukemia

Gene	Paper title	Reference
IGF2R	IGF Signaling Predicts Outcomes and Is a Promising Target Therapy for Acute Myeloid Leukemia	[30]
FADS1	Fatty acid desaturase 1 (FADS1) is a cancer marker for patient survival and a potential novel target for precision cancer treatment	[31]
CDK14	Prediagnostic transcriptomic markers of Chronic lymphocytic leukemia reveal perturbations 10 years before diagnosis	[32]
PON2	PON2 subverts metabolic gatekeeper functions in B cells to promote leukemogenesis	[33]

only enhances the computational e ciency during the search phase but also avoids performance degradation.

Expended search space to get better features: Beyond "consolation match", we incorporated a semi-random approach to pinpoint the global optimum. At the start of the search, we considered multiple candidate features with high importance to alter the first-choice feature and selected the feature subset with the best performance. From our observations, the optimal feature set frequently emerged when the first-choice feature was not ranked as the top feature.

Selection of a su ciently good features for microarray datasets: To understand or diagnose complex diseases, it is essential to not only achieve high predictive (classification) performance but also identify pivotal biomarkers (features) [27]. For example, a predictive accuracy of 0.82 with 20 features is more meaningful than an accuracy of 0.85 with 120 features. Although the proposed method does not always result in a better predictive performance than the other methods, the derived features are very compact with a decent prediction performance. erefore, it would be helpful to identify novel genes associated with diseases.

Usefulness of selected features by the proposed method: Microarray-related papers typically present a list of important genes derived through simple statistical analysis, without experimental validation. erefore, it is di cult to demonstrate that the features (genes) identified by the proposed method are experimentally useful. In the case of chiaretti dataset, the proposed method selected five genes—ITM2A, IGF2R, FADS1, CDK14, and PON2—which are related to lymphoblastic leukemia. e original paper [29] identified 31 important genes, among which only ITM2A overlaps with the proposed method's results; the other four genes are newly discovered. rough a literature analysis, we confirmed that these four genes are also associated with lymphoblastic leukemia, as shown in Table 8. ese findings indirectly demonstrate that the proposed method produces biologically meaningful results.

In our proposed method, during the consolation match, we specifically operated on the latter half of the selected features and the former half of the non-selected features. Unexpectedly, when we expanded this search range during our experiments, a decline in the performance was observed. is phenomenon raises questions that require further exploration, particularly concerning feature interactions. is is a topic for further research.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-024-06017-9.

Additional file 1.

Acknowledgements
None.

Author contributions
Shin and Oh conceptualized the research. Shin contributed to development of the model. Shin collected and annotated data. All authors read and approved the final manuscript.

Funding
This work was supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grand funded by the Ministry of Science, ICT (MSIT), Korea (No. RS-2023–00222191, Development of data fabric technology to support logical data integration and compound analysis of distributed data).

Availability of data and materials
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interest
The authors declare no competing interests.

Received: 14 October 2024 Accepted: 17 December 2024
Published online: 26 December 2024

References

1. Ang JC, Mirzal A, Haron H, Hamed HNA. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. IEEE/ACM Trans Comput Biol Bioinform. 2016;13(5):971–89. https://doi.org/10.1109/TCBB.2015.2478454.
2. Lu H, Chen J, Yan K, Jin Q, Xue Y, Gao Z. A hybrid feature selection algorithm for gene expression data classification. Neurocomputing. 2017;256:56–62. https://doi.org/10.1016/j.neucom.2016.07.080.
3. Almugren N, Alshamlan H. A survey on hybrid feature selection methods in microarray gene expression data for cancer classification. IEEE Access. 2019;7:78533–48. https://doi.org/10.1109/ACCESS.2019.2922987.
4. Bommert A, Sun X, Bischl B, Rahnenführer J, Lang M. Benchmark for filter methods for feature selection in high-dimensional classification data. Comput Stat Data Anal. 2020;143:106839. https://doi.org/10.1016/j.csda.2019.106839.
5. Manikandan G, Abirami S. Feature selection is important: state-of-the-art methods and application domains of feature selection on high-dimensional data. In: Kumar R, Paiva S, editors. Applications in Ubiquitous Computing. Cham: Springer; 2021. p. 177–96.
6. Chen C, Weiss ST, Liu YY. Graph convolutional network-based feature selection for high-dimensional and low-sample size data. Bioinformatics. 2023. https://doi.org/10.1093/bioinformatics/btad135.
7. Alhenawi E, Al-Sayyed R, Hudaib A, Mirjalili S. Feature selection methods on gene expression microarray data for cancer classification: a systematic review. Comput Biol Med. 2022;140:105051. https://doi.org/10.1016/j.compbiomed.2021.105051.
8. Liu B, Wei Y, Zhang Y, Yang Q. Deep neural networks for high dimension, low sample size data. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. Melbourne: AAAI, 2017; 2287–2293.
9. Li K, Wang F, Yang L, Liu R. Deep feature screening: feature selection for ultra high-dimensional data via deep neural networks. Neurocomputing. 2023;538:126186. https://doi.org/10.1016/j.neucom.2023.03.047.
10. Ro□o G, Melzi S, Castellani U, Vinciarelli A, Cristani M. Infinite feature selection: a graph-based feature filtering approach. IEEE Trans Pattern Anal Mach Intell. 2020;43(12):4396–410. https://doi.org/10.1109/TPAMI.2020.3002843.
11. Rostami M, Forouzandeh S, Berahmand K, Soltani M, Shahsavari M, Oussalah M. Gene selection for microarray data classification via multi-objective graph theoretic-based method. Artif Intell in Med. 2022;123:102228. https://doi.org/10.1016/j.artmed.2021.102228.
12. Pati SK, Banerjee A, Manna S. Gene selection of microarray data using heatmap analysis and graph neural network. Appl Soft Comput. 2023;135:110034. https://doi.org/10.1016/j.asoc.2023.110034.
13. Li Y, Chen CY, Wasserman WW. Deep feature selection: theory and application to identify enhancers and promoters. J Comput Biol. 2016;23(5):322–36. https://doi.org/10.1089/cmb.2015.0189.

14. Chowdhury S, Dong X, Li X. Recurrent neural network based feature selection for high dimensional and low sample size micro-array data. In: Proceedings of 2019 IEEE International Conference on Big Data. Piscataway: IEEE, 2019; 4823–4828.
15. Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW. Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019). IEEE Access. 2021;9:26766–91. https://doi.org/10.1109/ACCESS.2021.3056407.
16. Dokeroglu T, Deniz A, Kiziloz HE. A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing. 2022;494:269–96. https://doi.org/10.1016/j.neucom.2022.04.083.
17. Ferri FJ, Pudil P, Hatef M, Kittler J. Comparative study of techniques for large-scale feature selection. Mach Intell Pattern Recognit. 1994;16:403–13. https://doi.org/10.1016/B978-0-444-81892-8.50040-7.
18. Wang L, Wang Y, Chang Q. Feature selection methods for big data bioinformatics: a survey from the search perspective. Methods. 2016;111:21–31. https://doi.org/10.1016/j.ymeth.2016.08.014.
19. Hsu HH, Hsieh CW, Lu MD. Hybrid feature selection by combining filters and wrappers. Expert Syst Appl. 2011;38(7):8144–50. https://doi.org/10.1016/j.eswa.2010.12.156.
20. Ramey J. Datamicroarray. *Available at* https://github.com/ramhiser/datamicroarray (accessed 19 Jan. 2024).
21. Arizona State University (ASU). Feature selection datasets. *Available at* https://jundongl.github.io/scikit-feature/datasets.html (accessed 19 Jan. 2024).
22. OpenML. A worldwide machine learning lab. *Available at* https://www.openml.org/ (accessed 19 May 2024).
23. Song Q, Jiang H, Liu J. Feature selection based on FDA and F-score for multi-class classification. Expert Syst Appl. 2017;81:22–7.
24. Berrendero JR, Cuevas A, Torrecilla JL. The mRMR variable selection method: a comparative study for functional data. J Stat Comput Simul. 2016;86(5):891–907.
25. Altmann A, Toloşi L, Sander O, Lengauer T. Permutation importance: a corrected feature importance measure. Bioinformatics. 2010;26(10):1340–7.
26. Oh S. Predictive case-based feature importance and interaction. Inform Sci. 2022;593:155–76. https://doi.org/10.1016/j.ins.2022.02.003.
27. Mi X, Zou B, Zou F, Hu J. Permutation-based identification of important biomarkers for complex diseases via machine learning models. Nature Commun. 2021;12(1):3008. https://doi.org/10.1038/s41467-021-22756-2.
28. Pudil P, Novovičová J, Kittler J. Floating search methods in feature selection. Pattern Recognit Lett. 1994;15(11):1119–25. https://doi.org/10.1016/0167-8655(94)90127-9.
29. Chiaretti S, Zini G, Bassan R. Diagnosis and subclassification of acute lymphoblastic leukemia. Mediterr J Hematol Infect Dis. 2014;6(1):e2014073.
30. Coelho-Silva JL, Machado-Neto JA, Fernandes JC, de Lima ASG, Scheucher PS, Rego EM, Traina F. IGF signaling predicts outcomes and is a promising target therapy for acute myeloid leukemia. Blood. 2017;130:3966.
31. Heravi G, Jang H, Wang X, Long Z, Peng Z, Kim S, Liu W. Fatty acid desaturase 1 (FADS1) is a cancer marker for patient survival and a potential novel target for precision cancer treatment. Front Oncol. 2022;12:942798.
32. Chadeau-Hyam M, Vermeulen RCH, Hebels DGAJ, Castagné R, Campanella G, Portengen L, et al. Prediagnostic transcriptomic markers of Chronic lymphocytic leukemia reveal perturbations 10 years before diagnosis. Ann Oncol. 2014;25(5):1065–72.
33. Pan L, Hong C, Chan LN, Xiao G, Malvi P, Robinson ME, et al. PON2 subverts metabolic gatekeeper functions in B cells to promote leukemogenesis. Proc Natl Acad Sci. 2021;118(7):e2016553118.

Publisher's Note