Enhancing Privacy-Preserving Cancer Classification with Convolutional Neural Networks

Aurora A. F. Colombo†, Luca Colombo, Alessandro Falcetta, and Manuel Roveri

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy

†*E-mail: auroraanna.colombo@mail.polimi.it*

Precision medicine significantly enhances patients prognosis, offering personalized treatments. Particularly for metastatic cancer, incorporating primary tumor location into the diagnostic process greatly improves survival rates. However, traditional methods rely on human expertise, requiring substantial time and financial resources. To address this challenge, Machine Learning (ML) and Deep Learning (DL) have proven particularly effective. Yet, their application to medical data, especially genomic data, must consider and encompass privacy due to the highly sensitive nature of data. In this paper, we propose OGHE, a convolutional neural network-based approach for privacy-preserving cancer classification designed to exploit spatial patterns in genomic data, while maintaining confidentiality by means of Homomorphic Encryption (HE). This encryption scheme allows the processing directly on encrypted data, guaranteeing its confidentiality during the entire computation. The design of OGHE is specific for privacy-preserving applications, taking into account HE limitations from the outset, and introducing an efficient packing mechanism to minimize the computational overhead introduced by HE. Additionally, OGHE relies on a novel feature selection method, VarScout, designed to extract the most significant features through clustering and occurrence analysis, while preserving inherent spatial patterns. Coupled with VarScout, OGHE has been compared with existing privacy-preserving solutions for encrypted cancer classification on the iDash 2020 dataset, demonstrating their effectiveness in providing accurate privacy-preserving cancer classification, and reducing latency thanks to our packing mechanism. The code is released to the scientific community.

Keywords: Computational genomics; Deep Learning; Homomorphic encryption; Privacy.

1. Introduction

Precision medicine is fundamentally changing the landscape of cancer treatment by tailoring medical care to individual genetic profiles, enhancing the efficacy of therapies.[1] This personalized approach not only targets treatments more effectively but also significantly improves patient outcomes and survival rates.[2] Nowadays, however, precision medicine mainly relies on human-performed processes, which require high expertise, lots of time, and finances.[3] From this perspective, the advancement of Machine Learning (ML) and Deep Learning (DL) techniques offers researchers the potential to improve cancer classification accuracy, particularly in identifying primary tumor sites from patients' genomic data,[4] which can lead to more precise and effective treatment strategies.[5] Medical clinics and hospitals often lack expertise in ML and DL and may struggle to afford the necessary computing infrastructure. To address this issue, third-party *as-a-service* solutions have emerged as a promising alternative.[6] However, exposing medical and personal data to third-party providers raises significant privacy concerns, especially when dealing with sensitive genomic information.[7] This vulnerability is a major obstacle to the widespread adoption of ML and DL-*as-a-service* (DLaaS) in healthcare.

In recent years, the application of Homomorphic Encryption (HE) within the DLaaS framework has gained considerable momentum in addressing privacy concerns. HE is an encryption method that encrypts data using a public key, making it unreadable to unauthorized entities. Only the holder of the corresponding private key can decrypt and access the original information. A key advantage of HE is its ability to perform computations on encrypted data without requiring decryption.[8] This enables the encrypted processing of patient genomic data by third-party ML and DL algorithms while maintaining data confidentiality, as the raw genomic data remains encrypted and inaccessible during analysis.[9]

With this method, healthcare institutions encrypt genomic data before transmitting it to a third-party ML and DL service, ensuring that the service provider remains unaware of the underlying data during processing. The service provider receives the keys needed to perform computations on encrypted data, and returns the encrypted results to the client for decryption. This privacy-preserving computation *as-a-service* not only addresses the shortage of ML and DL expertise while reducing costs, but also offers scalability and flexibility to meet the growing computational needs of medical research, data analysis, and clinical decision-making. In recent years, numerous privacy-preserving solutions have been developed for various healthcare applications, leveraging HE to protect sensitive data during analysis. For instance, studies have demonstrated the use of HE in securely processing medical images,[10] and conducting genome-wide association studies.[11] These advancements highlight the potential of HE to maintain data confidentiality while enabling valuable insights in the healthcare domain.

Nonetheless, the task of cancer classification on encrypted genomic data is quite new. Existing solutions have explored ML techniques such as Logistic Regressions (LR)[12] and Shallow Neural Networks (SNN).[13] Interestingly, despite their effectiveness in genomics,[14] Convolutional Neural Networks (CNNs) have received little attention due to their developmental complexity in the HE framework. Indeed, HE poses considerable limitations on the type and number of operations that can be performed on encrypted data. Since only addition and multiplication are supported by HE, several layers and activation functions commonly used in DL

models cannot be directly computed on encrypted inputs. Additionally, HE constraints the number of consecutive encrypted multiplications, thereby limiting the depth of DL models.[15]

In this perspective, our work introduces *Oncological Genomic analysis over HE and CNN* (OGHE), a CNN-based approach for cancer classification designed to operate on encrypted genomic data. Featuring parallel convolutional layers, OGHE separately analyzes Single Nucleotide Variants (SNVs) and Copy Number Variations (CNVs) to enhance accuracy and effectiveness. Additionally, OGHE employs a novel feature selection method, *Variant Scout* (VarScout), to extract the most significant features while preserving the inherent spatial patterns in genomic data. This approach effectively complements the characteristics of OGHE convolutional layers, while maintaining compatibility with HE limitations.

Overall, this work introduces the following innovations: (1) OGHE, a privacy-preserving CNN that incorporates parallel one-dimensional (1D) convolutional layers to independently capture SNVs and CNVs spatial patterns, as they provide distinct and uncorrelated information; (2) a novel feature selection technique, VarScout, which uses clustering and mutation frequency to identify key SNVs and CNVs, thereby reducing computational complexity; and (3) a novel packing mechanism to efficiently encrypt data, weights, and biases into ciphertexts, resulting in high computational performance and reduced latency. The efficacy and efficiency of OGHE and VarScout have been evaluated on the iDASH2020 competition dataset.[16] Compared to State-of-The-Art (SoTA) privacy-preserving cancer classification solutions, our approach achieves an accuracy improvement of 0.8% while reducing the inference time per sample to less than 30 seconds. The code has been made available to the scientific community.[a]

The paper is organized as follows. Sec. 2 presents the related literature. The background is given in Sec. 3. OGHE and VarScout are described in Sec. 4, whereas the experimental results are presented in Sec. 5. Conclusions are finally drawn in Sec. 6.

2. Related Works

In this section, we review the literature on cancer classification task on genomic data. We first discuss solutions for processing plain data and then those for encrypted data.

Over the past few years, both supervised and unsupervised learning techniques have been extensively explored for cancer classification based on genomic data.[17,18] However, the preference has leaned towards supervised classifiers as they result more reliable, interpretable, and precise. CNNs have largely conquered the genomic scenario thanks to their ability to extract spatial patterns.[14] For example, AlShibli et al.[19] proposed ResCNN6, a 6-layers Residual-CNN, to perform CNV-based cancer classification over six tumor types.[20] The architecture encompassed four 2D convolutional layers coupled with MaxPooling to extract relevant features, while two fully connected layers are exploited for classification. ResCNN6 presented shortcut connections to ensure the lowest training error possible by avoiding one or more convolutional layers. On the same task, Chen et al.[21] explored a simpler CNN architecture composed of two 1D convolutional layers. Each convolution is followed by MaxPooling and batch normalization, while a fully connected layer ended the processing pipeline. Despite their effectiveness, how-

[a]Code is available at `https://github.com/AI-Tech-Research-Lab/OGHE.git`.

ever, the reported solutions are not feasible for privacy-preserving computation based on HE due to the inability to compute several layers and activation functions on encrypted inputs. HE imposes stringent limitations, permitting only linear functions and operations. Additionally, the depth of these solutions would surpass the number of consecutive multiplications allowed by HE constraints, potentially leading to data corruption and unreliability.[22]

Due to the aforementioned HE limitations, ML solutions are still preferred over DL ones in privacy-preserving computation. In 2020, iDASH[16] competition challenged its competitors with the development of a cloud-based solution for privacy-preserving classification of eleven cancer locations exploiting genetic mutations and HE. Among the presented solutions, Sarkar et al.[12] developed a logistic regression approach, incorporating a feature engineering strategy to encode somatic mutations based on biological intuition and statistical tests. They advanced a technique to reduce the feature space from over $50,000$ features to $43,000$, implementing a HE-based model through an optimized matrix multiplication algorithm. Differently, Mağara et al.[23] investigated two ML algorithms, i.e., Support Vector Machine (SVM) and XGboost. Given that XGBoost internally utilizes comparisons not supported by the HE scheme, an efficient encoding method for encrypted comparison operations was devised for inference. Moreover, Hong et al.[13] proposed a Shallow Neural Network (SNN) consisting of one hidden layer with 64 nodes and a linear activation function. In the preprocessing of the input genomic data, the feature selection step incorporated both clustering and data filtering methods. Lastly, in 2024, Song et al.[24] introduced ReActHE, a family of CNNs characterized by a novel type of activation layer, i.e., the *Residue activation layer*, and a scaled power activation function. In particular, by selecting the $1,000$ most significant features by means of a L1 normalized logistic regression, they outperformed alternative privacy-preserving ML solutions, achieving low approximation errors in the cancer classification task.

Differently from the existing literature, our solution proposes two key aspects which are fundamental for privacy-preserving cancer classification. First, VarScout selects the most representative features in an effective way, helping in reducing input and model dimensions. Second, OGHE exploits spatial information from genomic data by employing only HE-compliant operations to allow encrypted computation.

3. Background

This section will present the basics needed to understand OGHE and VarScout implementation. Sec. 3.1 will provide a brief overview of the HE scheme employed, while Sec. 3.2 will present the characteristics of the genetic mutations analyzed.

3.1. *Homomorphic Encryption*

HE is a family of encryption schemes that enables a set of operations to be performed directly on encrypted data.[8] Mathematically, two functions $E(k_p, \cdot), D(k_s, \cdot)$ are said to be homomorphic with respect to a set of functions $\mathcal{F}$ if, for any $f \in \mathcal{F}$, a function g can be found that:

$$f(m) = D(k_s, g(E(k_p, m))) \tag{1}$$

for any set of input m.[25] In particular, $E(k_p, \cdot)$ and $D(k_s, \cdot)$ represent the encryption and

decryption functions, respectively, whereas k_p denotes the public key and k_s the secret key. The ability of HE to provide encrypted operations relies on the maintenance of the datum algebraic structure during the processing pipeline.[26] In this way, the result obtained from ciphertext computation is guaranteed to match the one from the same operation in plaintext. In this study, we adopted the Cheon–Kim–Kim–Song (CKKS) scheme,[27] which is based on the Ring Learning With Errors (RLWE) problem,[28] a computational problem commonly used in quantum-resistant cryptography.[29] The CKKS scheme supports encrypted additions and multiplications between real values. More in detail, it belongs to the family of leveled HE schemes, i.e., schemes that allow only a finite number of consecutive encrypted operations to be performed before the information is lost. This limit is called scheme level, denoted by l, and it is due to noise injection performed by the scheme itself in order to guarantee the probabilistic encryption properties.[30] In CKKS scheme, the algebraic structure of plaintexts and ciphertexts is defined through a set of encryption parameters $\Theta = \{N, q, \Delta\}$, where N is the polynomial modulus, q is the list of $l + 2$ coefficient modulus, and Δ is the scaling factor. More in detail, plaintexts are in the polynomial ring $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$, while ciphertexts are in the polynomial ring $\mathcal{R}_q = \mathbb{Z}_{q_0}[X]/(X^N + 1)$.

When dealing with the CKKS scheme, two key factors must be considered. The former deals with the choice of the encryption parameters Θ, defining the security level, which in this work is set to 128 bit, the polynomial order, and the encoding precision. They represent a trade-off between the scheme level l and the overhead added with respect to plain computation. The latter is called *batching* technique: it enables parallel processing through *Single Instruction, Multiple Data (SIMD)* operations.[27] By using *batching*, a single ciphertext can store up to $N/2$ values, reducing the computational overhead both in terms of time and memory requirements.

The adopted CKKS scheme supports two main operations. Let $\underline{a} = [\underline{a}_0, \underline{a}_1, \ldots, \underline{a}_n]$ and $\underline{b} = [\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_n]$ be two encrypted CKKS vectors. Then, the encrypted element-wise addition can be defined as follows:

$$\underline{A} + \underline{B} = [\underline{a}_0 + \underline{b}_0, \underline{a}_1 + \underline{b}_1, \ldots, \underline{a}_n + \underline{b}_n]. \tag{2}$$

Conversely, the encrypted element-wise product is defined as:

$$\underline{A} * \underline{B} = [\underline{a}_0 * \underline{b}_0, \underline{a}_1 * \underline{b}_1, \ldots, \underline{a}_n * \underline{b}_n]. \tag{3}$$

Additionally, matrices can be represented in ciphertext as their flattened forms. Aggregate operations that perform homomorphic sums across specific dimensions of encrypted data can be defined. Let $\underline{C}$ be an encrypted and flattened matrix with dimensions $M \times N$. The sum over columns is then expressed as:

$$\underline{S} = \left[\sum_{j=1}^{N} \underline{C}_{i,j} \right]_{i=1}^{M} \tag{4}$$

where $\underline{S}$ is an encrypted matrix containing the row sums of $\underline{C}$, repeated to match the dimensions of the input matrix. Similarly, this operation can be applied to the other dimension.

3.2. *Single Nucleotide Variants and Copy Number Variations*

Information from SNVs and CNVs is vital in the diagnostic process of metastatic cancer, as it helps in identifying the origin of the primary tumor mass. Being common for a certain population, SNVs, which involve the alteration of a single nucleotide in DNA strands, serve as biomarkers for specific diseases. When occurring in protein-coding regions, SNVs can lead to *missense variations*, i.e., the substitution of an amino acid altering protein structure and function, and *nonsense mutations*, i.e., the premature truncation of the protein-coding process.[31] SNVs are categorized as [LOW, MODERATE, MODIFIER, HIGH] based on their impact on disease onset, as determined by the Variant Effect Predictor software.

Conversely, CNVs are structural variations involving the rearrangement of more than 50 base pairs in the genome. Entire genes can be altered in the number of copies, compromising normal gene expression levels and affecting critical cellular processes like cell cycle regulation, apoptosis, and cell signaling. Thus, CNVs are strongly associated with genetic disorders and complex diseases such as cancer.[32] In this work, CNVs are represented by five mutation levels, i.e., $\{0.0, \pm 1.0, \pm 2.0\}$, where the absolute value indicates the number of strands involved, while the sign denotes either a positive duplication or a negative deletion. This information is directly inferred from the Copy Number Segmentations generated by the ASCAT software.

4. Proposed Solution

This section details the proposed solution, composed of VarScout and OGHE, designed to address the considered primary tumor location problem, formalized as follows. Let X_{CNV} and X_{SNV} be two vectors of size L_{CNV} and L_{SNV}, respectively. We define the primary tumor location as $\hat{y} = \arg\max_i \ y_i$, where

$$y = \varphi(X_{CNV}, X_{SNV}) \in \mathbb{R}^C \tag{5}$$

is the output vector, C is the number of classes, and $\varphi(\cdot)$ is the model. In the rest of the section, we will consider the encrypted version of this problem. In particular, Sec. 4.1 introduces VarScout, our proposed feature selection method designed to reduce the CNV and SNV feature space dimension, while Sec. 4.2 details OGHE, the model architecture specifically designed to provide encrypted primary tumor classification on encrypted SNV and CNV inputs.

4.1. *VarScout method*

VarScout has been designed to reduce CNV and SNV feature space dimension while keeping the highest data representation, which is crucial to design OGHE being HE compliant. Inspired by Hong et al.,[13] VarScout aims at enhancing OGHE accuracy by prioritizing the most impactful mutations while maintaining typical spatial patterns.

After organizing CNV and SNV mutations in chromosomal order, agglomerative cluster analysis is employed for CNV filtering to eliminate redundant information. Specifically, our method comprises three main steps: (1) similarity computation between adjacent genes (g_i, g_{i+1}) through the Hamming distance d, i.e., $d(g_i, g_{i+1})$, for each gene g_i in the original dataset; (2) formation of clusters O_l such that $O_l = \{g_i \,|\, min(d(g_i, g_{i+1}))\}$, i.e., genes character-

ized by the least distance are chosen to form a cluster; and (3) selection of the first in order gene g_i to represent the cluster l. These steps are repeated until l reaches L_{CNV}.

Conversely, SNVs are numerically encoded within the range $\{0.0, 0.20, 0.50, 0.90, 1.0\}$, as proposed by Hong et al.,[13] to denote the impact of the mutation on the disease insurgence. In particular, 0.0 represents the absence of genetic alteration, whereas 1.0 denotes the highest level of influence. To reduce SNV feature space, our feature selection method is based on mutation occurrences across different cancer types. Frequencies are calculated based on the impact of genetic alterations, defined as $F_j = \sum_i f_{ij}$, for $i = [1, ..., |Z_c|]$. More specifically, F_j denotes the weighted frequency of occurrence of a gene j within a sub-population $Z_c = \{x \,|\, y = c\} \, \forall c = [1, \ldots, C]$ characterized by a specific cancer class y, and f_{ij} represents the impact of the j-th gene for each individual sample i. The process ranks mutations by sequentially adding the most recurrent gene for each tumor type to the feature space until the desired feature dimension L_{SNV} is reached.

4.2. *OGHE Architecture*

To exploit spatial patterns in genomic data while addressing HE constraints, we propose OGHE. OGHE takes as input VarScout-selected CNV and SNV features encoded as two separate ciphertexts, namely $\underline{\widetilde{X}_{CNV}}$ and $\underline{\widetilde{X}_{SNV}}$, and outputs an encrypted vector $\underline{\widetilde{Y}_{pred}}$ of length C. Once decrypted, the output $Y_{pred} = D(k_s, \underline{\widetilde{Y}_{pred}})$, where $D(k_s, \cdot)$ is the decryption function described in Sec. 3, reveals the predicted cancer class, identified by the index of the highest value in Y_{pred}. OGHE architecture is designed to work within HE constraints while maintaining high accuracy and computational performance. The training of OGHE was performed on plain data, although it is specifically designed for encrypted inference.

As shown in Fig. 1, OGHE is a shallow CNN composed of two parallel 1D convolutional layers and a fully connected layer. Parallel convolutions are chosen to separate CNV and SNV information, ensuring independent processing of uncorrelated data until the fully connected layer. In its training configuration, a square activation function was chosen as commonly used in the privacy-preserving DL[15] framework, and a Spatial-Dropout layer is incorporated to mitigate the risk of overfitting. The feature maps resulting from the convolutional layers, i.e., $Y_{conv_{h,CNV}}$ and $Y_{conv_{h,SNV}}$, for each kernel $h = [1, \ldots, H]$, are then concatenated and passed to the flatten function. A fully connected layer ends the processing pipeline to provide the output vector Y_{pred}, where the index of the highest value indicates the predicted primary tumor mass location. This strategic design ensures compatibility with the limitations posed by HE while exploiting the available genetic information for accurate cancer classification.

Conversely, OGHE encrypted processing is designed to provide optimal computational performance by efficiently managing the ciphertext space through a well-defined packing mechanism. By strategically organizing data within ciphertexts, our approach enables efficient encrypted computations, and significantly enhances the performance of the network. OGHE includes *Encrypted Convolutional Blocks* (Sec. 4.2.1) and an *Encrypted Fully Connected Block* (Sec. 4.2.2). To streamline the notation we will consider the case where both the input data and OGHE model are encrypted. However, the following formulations can easily be extended to the scenario where the model is kept unencrypted by the service provider.

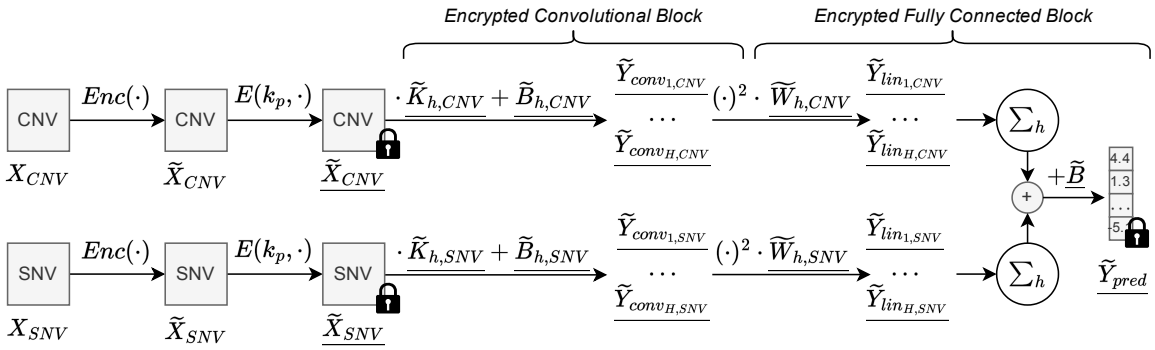

Fig. 1. OGHE encrypted pipeline. Each sample, composed of X_{CNV} and X_{SNV}, is encoded and encrypted into $\underline{\widetilde{X}_{SNV}}$ and $\underline{\widetilde{X}_{CNV}}$, respectively, before being processed by the *Encrypted Convolutional Block* and the *Encrypted Fully Connected Block*.

4.2.1. *Encrypted Convolutional Block*

This block proposes a 1D re-elaboration of the *im2col* method[33] to facilitate the computation of CNV and SNV convolutional layers. In our approach, data, weights, and feature maps are efficiently packed to be encrypted into single ciphertexts, to maximize computational efficiency.

Let $K = [k_1, \ldots, k_D]$ denote a 1D convolutional kernel of dimension D and stride S, and let $X = [x_1, \ldots, x_{L_x}]$ represent the 1D input vector. Our method encodes the input X into a matrix $\widetilde{X}$ of size $L_y \times V$, built as follows:

$$\widetilde{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{D-1} & x_D & 0 & \cdots & 0 \\ x_{(S+1)} & x_{(S+1)+1} & \cdots & x_{(S+1)+D-1} & x_{(S+1)+D} & 0 & \cdots & 0 \\ x_{(2S+1)} & x_{(2S+1)+1} & \cdots & x_{(2S+1)+D-1} & x_{(2S+1)+D} & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ x_{(iS+1)} & x_{(iS+1)+1} & \cdots & x_{(iS+1)+D-1} & x_{(iS+1)+D} & 0 & \cdots & 0 \end{bmatrix}. \tag{6}$$

Similarly, the convolutional kernel K is encoded into $\widetilde{K}$, a $L_y \times V$ matrix where each row contains a copy of K:

$$\widetilde{K} = \begin{bmatrix} k_1\, k_2 \cdots k_{D-1}\, k_D\, 0 \cdots 0 \\ k_1\, k_2 \cdots k_{D-1}\, k_D\, 0 \cdots 0 \\ k_1\, k_2 \cdots k_{D-1}\, k_D\, 0 \cdots 0 \\ \vdots \\ k_1\, k_2 \cdots k_{D-1}\, k_D\, 0 \cdots 0 \end{bmatrix}. \tag{7}$$

For computational reasons, both $\widetilde{X}$ and $\widetilde{K}$ are padded with zeros at the end of each row to maintain a power of 2 number of columns V, which will be set to $V = 2^{\lceil log_2(max(D,C)) \rceil}$. Thus, $\widetilde{X}$ and $\widetilde{K}$ share the same shape $L_y \times V$, where

$$L_y = \left\lfloor \frac{L_x - D}{S} \right\rfloor + 1, \tag{8}$$

being L_x the input length, D the kernel dimension, S the stride, and C the number of classes.

The matrices $\widetilde{X}$, $\widetilde{K}$ are then flattened and encrypted into the ciphertexts $\underline{\widetilde{X}}$ and $\underline{\widetilde{K}}$, respectively. This encoding ensures that both the input vector X and the convolutional kernel K are appropriately formatted for efficient encrypted computation.

Computing the h-th convolution $\underline{\widetilde{Y}_{conv_h}}$ is reduced to a single Hadamard multiplication between ciphertexts, followed by a sum over the columns, as described in Eq. (4):

$$\underline{\widetilde{Y}_{conv_h}} = \left[\sum_{j=1}^{V} (\underline{\widetilde{X}_h} \cdot \underline{\widetilde{K}_h})_{i,j} \right]_{i=1}^{L_y} \tag{9}$$

where i denotes the i-th row, j the j-th column, and L_y and V the dimensions of the output $\underline{\widetilde{Y}_{conv_h}}$. This operation is repeated for each of the $h = [1, \ldots, H]$ kernels of the convolutional block. The resulting ciphertext $\underline{\widetilde{Y}_{conv_h}}$ will be encoded as:

$$\underline{\widetilde{Y}_{conv_h}} = \begin{bmatrix} \underline{y_1} & \underline{y_1} & \cdots & \underline{y_1} & \underline{y_1} & \underline{y_1} & \cdots & \underline{y_1} \\ \underline{y_2} & \underline{y_2} & \cdots & \underline{y_2} & \underline{y_2} & \underline{y_2} & \cdots & \underline{y_2} \\ \underline{y_3} & \underline{y_3} & \cdots & \underline{y_3} & \underline{y_3} & \underline{y_3} & \cdots & \underline{y_3} \\ \vdots & & & & & & & \\ \underline{y_{L_y}} & \underline{y_{L_y}} & \cdots & \underline{y_{L_y}} & \underline{y_{L_y}} & \underline{y_{L_y}} & \cdots & \underline{y_{L_y}} \end{bmatrix} \tag{10}$$

in its flattened form, where $\underline{y_i}$ is the encrypted result of a single window convolution. Lastly, the bias is encoded to match $\underline{\widetilde{Y}_{conv_h}}$ packing, replicated $L_y * V$ times, and added to it.

It is worth noting that in OGHE, CNV and SNV inputs are processed in separate, parallel 1D convolutional layers. Eq. (9) is effectively used to compute *Encrypted Convolutions* for each kernel and parallel branch, after which the square activation is applied.

4.2.2. *Encrypted Fully Connected Block*

The output of each parallel convolutional branch is subsequently forwarded through the final fully connected layer. However, since CKKS ciphertexts cannot be concatenated, the operation has to be decomposed. The weight matrix W associated to the layer is split into $2H$ submatrices W_h, which are then flattened. Specifically, each W_h represents the portion of weights W that has to be multiplied by the h-th output channel per each parallel branch. This way, $2H$ reduced fully connected layers can be performed to compute the output:

$$\underline{\widetilde{Y}_{lin_{h,CNV}}} = \left[\sum_{i=1}^{L_y} (\underline{\widetilde{Y}_{conv_{h,CNV}}} \cdot \underline{\widetilde{W}_{h,CNV}})_{i,j} \right]_{j=1}^{V}, \underline{\widetilde{Y}_{lin_{h,SNV}}} = \left[\sum_{i=1}^{L_y} (\underline{\widetilde{Y}_{conv_{h,SNV}}} \cdot \underline{\widetilde{W}_{h,SNV}})_{i,j} \right]_{j=1}^{V}. \tag{11}$$

A Hadamard multiplication followed by a summation over the rows, described in Eq. (4), effectively emulates a vector-matrix multiplication. All the $2H$ values are then summed together along with the bias vector to provide the final prediction $\underline{\widetilde{Y}_{pred}}$, which will be encoded having the resulting vector repeated along the ciphertext.

Lastly, by multiplying the output $\underline{\widetilde{Y}_{pred}}$ by a binary mask, the output vector $\underline{Y_{pred}}$ will result in a single prediction vector of size C, where each element corresponds to a class. After decrypting with the secret key, i.e., $D(k_s, \cdot)$, the index of the element with the highest value will correspond to OGHE prediction.

5. Experimental Results

The experimental campaign is organized into two parts. First, Sec. 5.2 compares OGHE to SoTA solutions in terms of accuracy, micro Area Under the Curve (mAUC), and computational performance for privacy-preserving cancer classification tasks. Then, Sec. 5.3 shows the effectiveness of OGHE and VarScout when compared to Hong et al.[13] SNN and a baseline model, i.e., a single fully connected layer network, referred to as FCM. Our solution, implemented using OpenFHE-python[34] library, has been tested on a workstation equipped with 2 Intel Xeon Gold 5318 S CPUs and 384GBs of RAM.

5.1. *Procedure*

To evaluate stability and consistency, OGHE and VarScout were tested alongside the literature on the iDASH2020 dataset,[16] sourced from The Cancer Genome Atlas (TCGA). This dataset includes $3,622$ samples with CNV and SNV information for $25,128$ genes, and eleven cancer classes representing the primary tumor mass location.

Moreover, for the in-depth comparison of Sec. 5.3, we employed a 5-fold cross-validation technique to evaluate all the considered models. For each fold, we allocated data in a 7:1:2 ratio for training, validation, and testing, respectively. We also employed a hyperparameter selection based on the validation loss for all the considered models. In particular, we optimized the hyperparameters for OGHE considering the following ranges: kernel sizes of $\{16, 32, 64\}$, strides of $\{4, 8, 16\}$, and number of kernels $\{4, 8, 16\}$, along with activation functions either linear or square. The spatial dropout rate has been fixed to 0.5. We fixed an Adam optimizer with a weight decay of 0.0001, learning rate of 0.001 and cosine annealing, and batch size of 16. Instead, for the SNN[13] and FCM, the learning rate was evaluated in $\{0.001, 0.0001\}$, the batch size in $\{4, 8, 16, 32\}$, and the weight decay of the Adam optimizer in $\{0, 0.0001, 0.0005\}$. All the solutions were trained for 200 epochs using a weighted cross-entropy loss function, whose weights are inversely proportional to class frequencies.

For the encrypted computations, we employed the following CKKS encryption parameters: $\Theta = \{N = 32,768, q = [60, 50, 50, 50, 50, 50, 60], \Delta = 2^{50}\}$, yielding results that are consistent with those obtained from processing in plaintext, and ensuring a 128-bits security level.[35]

5.2. *Comparison with SoTA Solutions*

As a first analysis, OGHE accuracy was compared to privacy-preserving cancer classification SoTA solutions. Note that we compared OGHE only to models specifically designed for HE applications, as they share the same characteristics and limitations.

As demonstrated in Table 1, our solution outscores all other models in the literature in terms of accuracy, highlighting the exceptional capabilities of OGHE and VarScout. This accuracy improvement is attributed to the simultaneous learning from multiple sources, namely CNV and SNV, which enhances the model's robustness to variations and noise, thereby increasing its reliability. Nonetheless, OGHE shows a slight decrease in mAUC, which can be attributed to a more distributed error among the classes.

Furthermore, the computational performance of OGHE has been assessed in comparison to DL models in literature, as shown in Table 2. Since ReActHE[24] was evaluated in its orig-

Table 1. Accuracy and mAUC of our proposed solution compared to the existing literature.

	Model name	Model class	Accuracy	mAUC
Mağara et al.[23]	XGBoost	XGBoost	–	93.80%
Sarkar et al.[12]	LR	LR	83.61%	98.00%
Song et al.[24]	ReActHE	CNN	83.82%	–
Hong et al.[13]	SNN	NN	85.15%	98.82%
Ours	OGHE	CNN	85.94%	98.44%

Table 2. Comparison of computational performance with respect to FCM and SNN[13] in terms of encryption, computation, and decryption time, and in terms of latency per sample (L_1), and in the encrypted inference of 100 samples (L_{100}). All values are in seconds.

	Model name	Enc[s]	Comp[s]	Dec[s]	L_1[s]	L_{100}[s]
Hong et al.[13]	SNN	13.50	227.20	0.10	240.80	–
Song et al.[24]	ReActHE	–	–	–	–	685.35
Ours	OGHE	2.97	23.17	0.013	27.59	190.02

inal work by encrypting the model weights, the weights and biases of OGHE have also been encrypted to ensure a fair comparison. Additionally, single-sample inference utilized only 4 threads, whereas for the inference of 100 samples we limited our machine to use 40 threads, to align with the ReActHE[24] experimental setting. Table 2 proves the efficiency of our method both in single-sample and high-throughput inference. Notably, the computational times are the same for all the feature sizes up to [1024, 2286], given that the inputs, weights, and feature maps fit into a single ciphertext. For larger models, two ciphertexts must be used, leading to a slight increase in latency. However, the performance remains highly competitive, outperforming current state-of-the-art solutions. Additionally, the potential for further optimization through parallelization ensures scalability and efficiency in future implementations. Moreover, OGHE encryption time encompasses both model and data encryption. However, model encryption takes 2.88 seconds, making it the most time-consuming aspect of the encryption process. This evaluation considers the worst-case scenario where both the model and data are encrypted. If the model were in plaintext, a significant amount of time (around 15%) would be saved, highlighting the efficiency potential in less stringent encryption scenarios.

5.3. *VarScout and OGHE evaluation*

The aim of this part is to rigorously evaluate the effectiveness of VarScout and OGHE. To do so, we used the preprocessing method of Hong et al.[13] as feature extractor for OGHE, FCM, and the SoTA SNN model.[13] Subsequently, we applied our VarScout preprocessing method to our solution to determine if VarScout effectively enhances OGHE's performance. Note that Song et al.[24] is not included in this comparison, as they did not release the implementation.

Initially, we applied Hong et al.[13] feature selection method to all the models under consideration. The models were then tested with different input sizes, reflecting the number of CNV and SNV features after preprocessing. Along with the size [709, 1198], identified by Hong et al.[13]

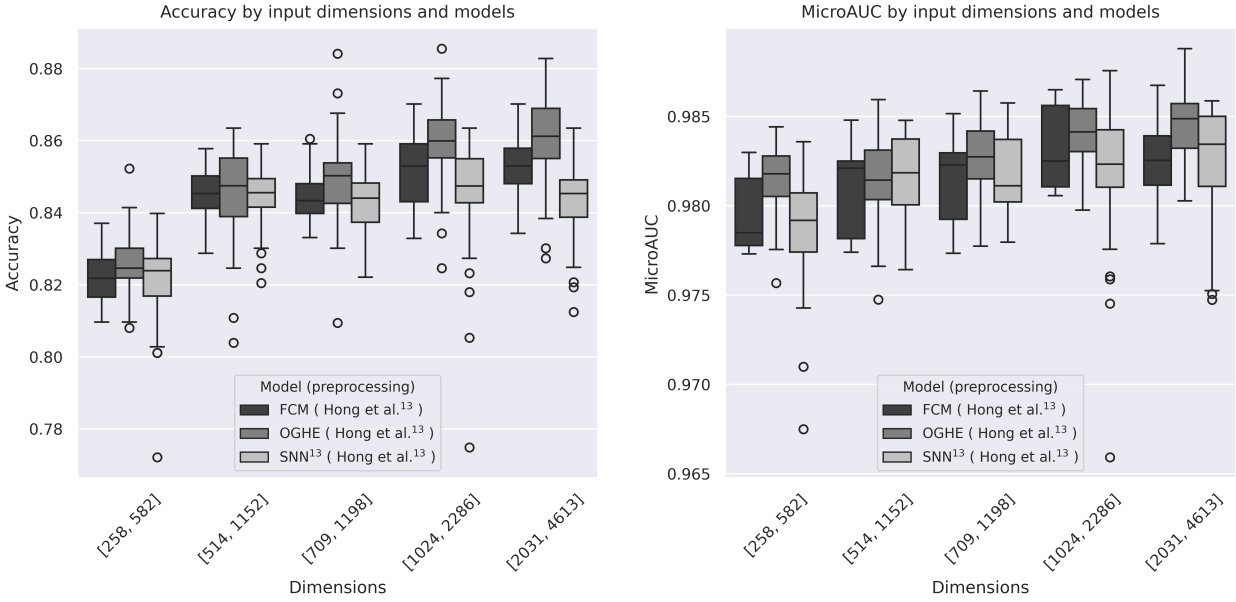

Fig. 2. Accuracy and MicroAUC boxplots of the considered models, for different input dimensions, i.e., the number of CNV and SNV features, respectively. The model names are followed by the preprocessing feature selection procedure in parenthesis. They show the metrics over 10 runs of the 5-fold cross validation.

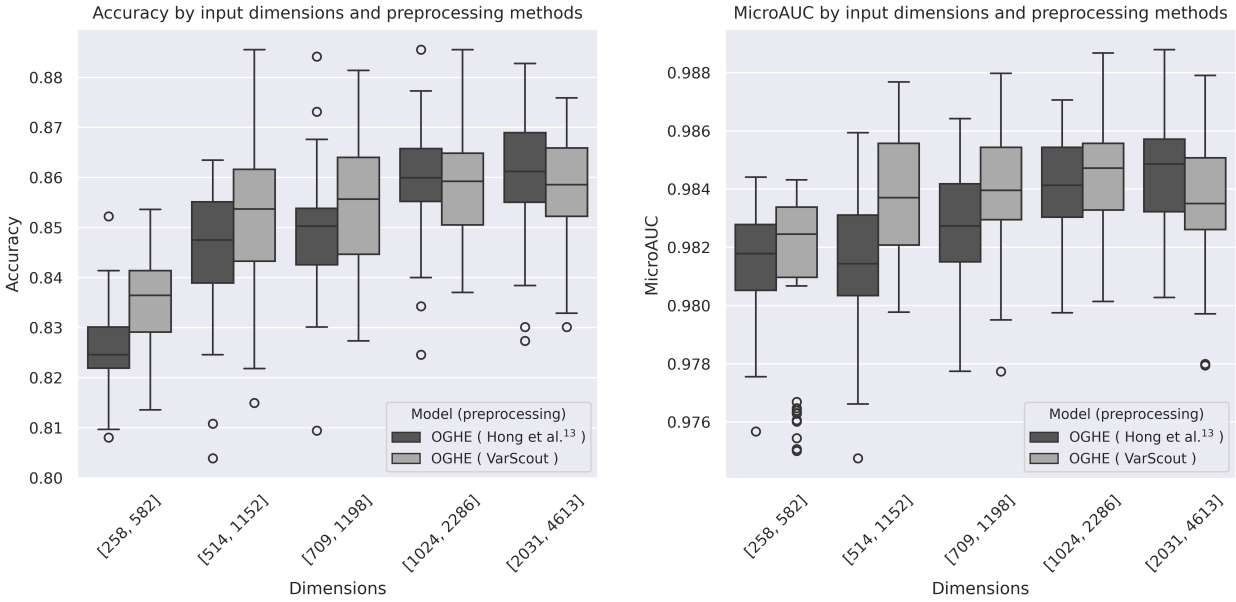

Fig. 3. Accuracy and MicroAUC boxplots of OGHE with different feature selection methods, for different input dimensions, i.e., the number of CNV and SNV features, respectively. The model names are followed by the preprocessing feature selection procedure in parenthesis. They show the metrics over 10 runs of the 5-fold cross validation.

as optimal, we also evaluated the input sizes $[258, 582]$, $[514, 1152]$, $[1024, 2286]$, and $[2031, 4613]$.

Fig. 2 demonstrates the effectiveness of OGHE, showing that it outperforms both FCM and SNN.[13] OGHE shows greater improvement over the other models as input feature size increases. Achieving a higher median accuracy with a narrower interquartile range, OGHE confirms that genomic data contains useful spatial and hierarchical information, effectively captured by the convolutional layers. Moreover, OGHE shows reduced variance and a fewer outliers across all input sizes when compared to the SNN,[13] indicating its robustness to variations in the input data and parameter initialization when dealing with complex tasks.

Furthermore, the statistical difference between SNN[13] and OGHE was evaluated using the McNemar-Bowker[36] test when both models were provided with inputs of size $[1024, 2286]$ as it ensures optimal performance for both models. The comparison was based on the run providing the highest test accuracy for each fold of the cross validation. The test indicated a statistically significant difference at a 5% confidence level between OGHE and the SNN[13] in four out of five folds, confirming the improvement our approach provides over existing literature.

Further improvements rise from the introduction of VarScout as feature selection method. To demonstrate its effectiveness, OGHE integrated with VarScout was compared to OGHE using the feature selection method proposed by Hong et al.[13] Fig. 3 shows that, for smaller input sizes, the model trained on VarScout-extracted features outperforms the one trained with Hong et al.[13] method, demonstrating that our feature selection method is superior in capturing spatial patterns and extracting the most important features first. This characteristic helps in maintaining a smaller network without sacrificing performance. Specifically, reducing the feature space to $[514, 1152]$, which is half the size of the configuration providing the best performance, results in only a 0.7% loss in accuracy. This is a key aspect when dealing with HE computations as it allows the use of ciphertexts characterized by smaller polynomial rings, resulting in a significant reduction in memory footprint and computation time.

6. Conclusion

This work proposes OGHE, a HE-friendly CNN for privacy-preserving cancer classification, and VarScout, a preprocessing method designed to maximize OGHE performance. OGHE architecture exploits spatial correlations in genomic data, separately processing the most relevant SNVs and CNVs extracted by VarScout, while preserving their spatial patterns. Together, these techniques achieve SoTA performance in encrypted cancer classification.

Despite advancements in privacy-preserving computing, HE introduces significant limitations in Artificial Intelligence applications, particularly regarding reduced computational efficiency. Future research will focus on minimizing the computational overhead and developing encrypted training, enabling researchers to analyze genomic data securely while preserving privacy, unlocking new possibilities for medical research and discovery.

Additionally, leveraging Neural Architecture Search (NAS) to optimize OGHE's architecture under HE constraints could further enhance its performance by automating the search for optimal architectures. Lastly, the release of new datasets will enable further validation and refinement of OGHE, expanding its potential applications.

References

1. M. R. Kosorok and E. B. Laber, Precision medicine, *Annual review of statistics and its application* **6**, 263 (2019).
2. M. Schwaederle, M. Zhao, J. J. Lee, A. M. Eggermont, R. L. Schilsky, J. Mendelsohn, V. Lazar and R. Kurzrock, Impact of precision medicine in diverse cancers: a meta-analysis of phase ii clinical trials, *Journal of clinical oncology* **33**, p. 3817 (2015).
3. Y.-F. Sun, X.-R. Yang, J. Zhou, S.-J. Qiu, J. Fan and Y. Xu, Circulating tumor cells: advances in detection methods, biological issues, and clinical relevance, *Journal of cancer research and clinical oncology* **137**, 1151 (2011).
4. A. Bhola and A. K. Tiwari, Machine learning based approaches for cancer classification using gene expression data, *Machine Learning and Applications: An International Journal* **2**, 01 (2015).
5. D. van Uden, M. van Maaren, L. Strobbe, P. Bult, M. Stam, J. van der Hoeven, S. Siesling, J. de Wilt and C. Blanken-Peeters, Better survival after surgery of the primary tumor in stage iv inflammatory breast cancer, *Surgical Oncology* **33**, 43 (2020).
6. S. Ghosh and R. Dasgupta, Cloud computing infrastructure in healthcare industry, in *Machine Learning in Biological Sciences: Updates and Future Prospects*, (Springer, 2022) pp. 169–176.
7. J. Santaló and M. Berdasco, Ethical implications of epigenetics in the era of personalized medicine, *Clinical epigenetics* **14**, p. 44 (2022).
8. A. Acar, H. Aksu, A. S. Uluagac and M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, *ACM Computing Surveys (Csur)* **51**, 1 (2018).
9. A. Wood, K. Najarian and D. Kahrobaei, Homomorphic encryption for machine learning in medicine and bioinformatics, *ACM Computing Surveys (CSUR)* **53**, 1 (2020).
10. A. Vizitiu, C. I. Niţă, A. Puiu, C. Suciu and L. M. Itu, Towards privacy-preserving deep learning based medical imaging applications, in *2019 IEEE international symposium on medical measurements and applications (MeMeA)*, (IEEE, 2019).
11. M. Blatt, A. Gusev, Y. Polyakov and S. Goldwasser, Secure large-scale genome-wide association studies using homomorphic encryption, *Proceedings of the National Academy of Sciences* **117**, 11608 (2020).
12. E. Sarkar, E. Chielle, G. Gursoy, L. Chen, M. Gerstein and M. Maniatakos, Privacy-preserving cancer type prediction with homomorphic encryption, *Scientific reports* **13**, p. 1661 (2023).
13. S. Hong, J. H. Park, W. Cho, H. Choe and J. H. Cheon, Secure tumor classification by shallow neural network using homomorphic encryption, *BMC genomics* **23**, 1 (2022).
14. C. Gunavathi, K. Sivasubramanian, P. Keerthika and C. Paramasivam, A review on convolutional neural network based deep learning methods in gene expression data for disease diagnosis, *Materials Today: Proceedings* **45**, 2282 (2021).
15. A. Falcetta and M. Roveri, Privacy-preserving deep learning with homomorphic encryption: An introduction, *IEEE Computational Intelligence Magazine* **17**, 14 (2022).
16. X. Jiang, A. O. Harmanci, M. Kim, H. Tang, X. Wang, T.-T. Kuo and L. Ohno-Machado, Idash privacy & security workshop 2020 - secure genome analysis competition (2020).
17. D. Wu, D. Wang, M. Q. Zhang and J. Gu, Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification, *BMC genomics* **16**, 1 (2015).
18. Y. Chen, J. Sun, L.-C. Huang, H. Xu, Z. Zhao *et al.*, Classification of cancer primary sites using machine learning and somatic mutations, *BioMed research international* **2015** (2015).
19. A. AlShibli and H. Mathkour, A shallow convolutional learning network for classification of cancers based on copy number variations, *Sensors* **19**, p. 4207 (2019).
20. J. Gao, B. A. Aksoy, U. Dogrusoz, G. Dresdner, B. Gross, S. O. Sumer, Y. Sun, A. Jacobsen, R. Sinha, E. Larsson *et al.*, Integrative analysis of complex cancer genomics and clinical profiles

using the cbioportal, *Science signaling* **6**, pl1 (2013).
21. H. Attique, S. Shah, S. Jabeen, F. G. Khan, A. Khan and M. ELAffendi, Multiclass cancer prediction based on copy number variation using deep learning, *Computational Intelligence and Neuroscience* **2022** (2022).
22. T. Lepoint and M. Naehrig, A comparison of the homomorphic encryption schemes fv and yashe, in *International Conference on Cryptology in Africa*, (Springer, 2014).
23. Ş. S. Mağara, C. Yıldırım, F. Yaman, B. Dilekoğlu, F. R. Tutaş, E. Öztürk, K. Kaya, Ö. Taştan and E. Savaş, Ml with he: Privacy preserving machine learning inferences for genome studies, *arXiv preprint arXiv:2110.11446* (2021).
24. C. Song and X. Shi, Reacthe: A homomorphic encryption friendly deep neural network for privacy-preserving biomedical prediction, *Smart Health* **32**, p. 100469 (2024).
25. F. Boemer, A. Costache, R. Cammarota and C. Wierzynski, ngraph-he2: A high-throughput framework for neural network inference on encrypted data, in *Proceedings of the 7th ACM workshop on encrypted computing & applied homomorphic cryptography*, (ACM, 2019).
26. M. Ogburn, C. Turner and P. Dahal, Homomorphic encryption, *Procedia Computer Science* **20**, 502 (2013).
27. J. H. Cheon, A. Kim, M. Kim and Y. Song, Homomorphic encryption for arithmetic of approximate numbers, in *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, (Springer, 2017).
28. V. Lyubashevsky, C. Peikert and O. Regev, On ideal lattices and learning with errors over rings, in *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, (Springer, 2010).
29. J.-P. Bossuat, R. Cammarota, J. H. Cheon, I. Chillotti, B. R. Curtis, W. Dai, H. Gong, E. Hales, D. Kim, B. Kumara *et al.*, Security guidelines for implementing homomorphic encryption, *Cryptology ePrint Archive* (2024).
30. G. J. Fuchsbauer, An introduction to probabilistic encryption, *Osječki matematički list* **6**, 37 (2006).
31. P. Katsonis, A. Koire, S. J. Wilson, T.-K. Hsu, R. C. Lua, A. D. Wilkins and O. Lichtarge, Single nucleotide variations: biological impact and theoretical interpretation, *Protein Science* **23**, 1650 (2014).
32. C. N. Henrichsen, E. Chaignat and A. Reymond, Copy number variants, diseases and gene expression, *Human molecular genetics* **18**, R1 (2009).
33. K. Chellapilla, S. Puri and P. Simard, High performance convolutional neural networks for document processing, in *Tenth international workshop on frontiers in handwriting recognition*, (Suvisoft, 2006).
34. A. Al Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee *et al.*, Openfhe: Open-source fully homomorphic encryption library, in *Proceedings of the 10th workshop on encrypted computing & applied homomorphic cryptography*, (ACM, 2022).
35. D. Joseph, R. Misoczki, M. Manzano, J. Tricot, F. D. Pinuaga, O. Lacombe, S. Leichenauer, J. Hidary, P. Venables and R. Hansen, Transitioning organizations to post-quantum cryptography, *Nature* **605**, 237 (2022).
36. A. Krampe and S. Kuhnt, Bowker's test for symmetry and modifications within the algebraic framework, *Computational statistics & data analysis* **51**, 4124 (2007).