MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
3 October 2022

REVISED
13 January 2023

ACCEPTED FOR PUBLICATION
3 February 2023

PUBLISHED
17 February 2023

PAPER

Quantum machine learning framework for virtual screening in drug discovery: a prospective quantum advantage

Stefano Mensa[1,3], Emre Sahin[1,3], Francesco Tacchino[2], Panagiotis Kl Barkoutsos[2] and Ivano Tavernelli[2,*]

[1] The Hartree Centre, STFC, Sci-Tech Daresbury, Warrington WA4 4AD, United Kingdom
[2] IBM Quantum, IBM Research—Zurich, Säumerstrasse 4, Rüschlikon CH-8803, Switzerland
[3] These authors contributed equally.
[*] Author to whom any correspondence should be addressed.

E-mail: ita@zurich.ibm.com

Keywords: quantum machine learning, quantum kernel methods, machine learning, support vector classifier, drug discovery, ligand based, virtual screening

Abstract

Machine Learning for ligand based virtual screening (LB-VS) is an important *in-silico* tool for discovering new drugs in a faster and cost-effective manner, especially for emerging diseases such as COVID-19. In this paper, we propose a general-purpose framework combining a classical Support Vector Classifier algorithm with quantum kernel estimation for LB-VS on real-world databases, and we argue in favor of its prospective quantum advantage. Indeed, we heuristically prove that our quantum integrated workflow can, at least in some relevant instances, provide a tangible advantage compared to state-of-art classical algorithms operating on the same datasets, showing strong dependence on target and features selection method. Finally, we test our algorithm on IBM Quantum processors using ADRB2 and COVID-19 datasets, showing that hardware simulations provide results in line with the predicted performances and can surpass classical equivalents.

1. Introduction

The early stages in a drug discovery process are notoriously expensive and time-consuming, usually coupling wet lab with *in-silico* technologies. virtual screening (VS) is an important computational technique used to provide a quick and economical method for the discovery of novel therapeutics. In practice, VS searches digital libraries of small molecules to identify compounds that are most likely to bind to a target (e.g. a protein) and therefore exert a pharmacological effect (i.e. activity). In an ideal scenario, the pharmacological binding site is known, and the structure of the target is well characterised. Therefore, the VS could—in principle—be performed directly using a digital representation of the binding site.

However, the structure of the target is generally unknown, especially for emerging diseases such as COVID-19, caused by SARS-CoV-2, the coronavirus that arose in December 2019. In such cases, the information available for known active molecules (ligands) against a given target is commonly used to drive the design process. Starting from the chemical structure of known ligands, it is possible to infer hundreds of cheminformatic features that can be coupled with experimental evidence, when available, and then used to perform VS of a digital database. Such methods are known as ligand based VS (LB-VS). A conventional LB-VS strategy consists in using machine learning (ML) algorithms to train a classifier that can identify potential drug candidates from a digitalised library of compounds. The latter are usually characterized by means of a combination of theoretical and experimental features. Thanks to their proven efficiency in identifying patterns in unstructured data, ML algorithms such as Support Vector Classifiers (SVCs) became an effective tool for performing LB-VS tasks [1–10].

Recent progress in the field of quantum machine learning (QML) [11, 12] identified quantum kernel (QK) methods [13], and specifically Quantum Support Vector Classifiers (QSVC) [14], among the most promising candidates for extending the reach of artificial intelligence beyond classical boundaries. By mapping classical inputs into high-dimensional complex Hilbert spaces, QK methods may efficiently

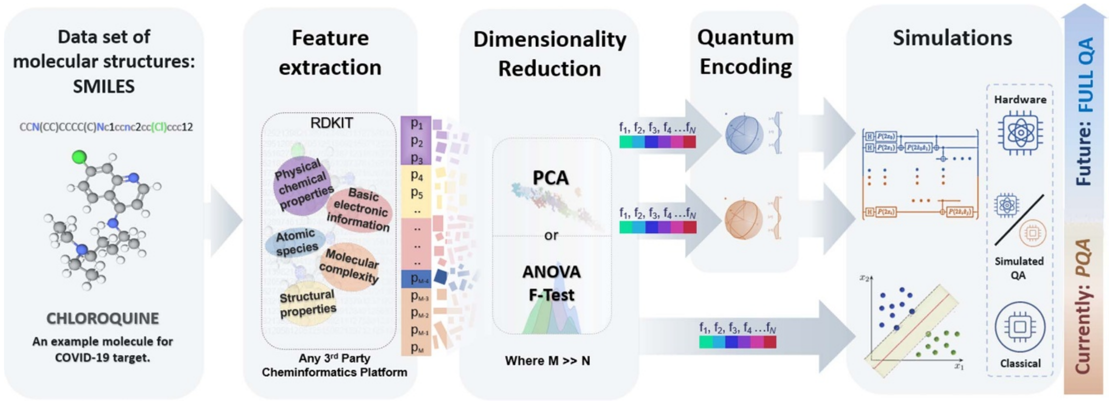

Data set of molecular structures: SMILES
CCN(CC)CCCC(C)Nc1ccnc2cc(Cl)ccc12
CHLOROQUINE
An example molecule for COVID-19 target.
Feature extraction
RDKIT
Physical chemical properties
Basic electronic information
Atomic species
Molecular complexity
Structural properties
Any 3rd Party Cheminformatics Platform
p_1 p_2 p_3 p_4 p_5 .. .. p_{M-4} p_{M-3} p_{M-2} p_{M-1} p_M
Dimensionality Reduction
PCA
or
ANOVA F-Test
Where M >> N
$f_1, f_2, f_3, f_4 \dots f_N$
Quantum Encoding
Simulations
Hardware
Simulated QA
Classical
x_2
x_1
Future: FULL QA
Currently: PQA

Figure 1. Hybrid classical-quantum integrated Machine Learning framework for Ligand-Based Virtual Screening in drug discovery. A database of SMILES encoded molecules is used to extract a variety of molecular features using RDKit. Through different feature reduction and selection methods, we refine the feature vectors, which are then passed on to train and test a Support Vector Classifier algorithm. We compare the performance of the SVC algorithm trained using a classical and a quantum kernel on both classical and quantum hardware, and we find cases in which the quantum approach outperforms classical equivalents in a consistent way when the problem size, as measured by the reduced number of features, goes beyond a certain threshold. We name this *Prospective Quantum Advantage* (PQA) in drug discovery. With the advent of larger and better performing near-term quantum computers, PQA may lead to full quantum advantage on relevant problem instances.

produce atypical patterns, thus potentially leading to quantum advantage in training speed, prediction accuracy and classification [15, 16]. While many key challenges [17, 18], and particularly the quest for robust quantum advantage on naturally occurring data sets, remain open, QML protocols appear to be very well suited in situations dealing with complex data and large amounts of available information, such as the LB-VS case [19].

At the same time, the application of QML to VS comes with complications, and guidance on how to port the LB-VS problem to a quantum computing framework is currently missing. The first practical issue consists in identifying the most suited approach to efficiently load data (i.e. the database of molecules with features) into a quantum register. Second, there is the need to optimize the available QML algorithms so that to exploit—in the most efficient way—the potential of the quantum processor for the specific LB-VS task, providing a path for achieving quantum advantage over classical counterparts.

In this paper, we propose a general framework for the integration of quantum computing in the LB-VS workflow for drug discovery, see figure 1. We demonstrate that the construction of a quantum classifier trained with a small number of cheminformatics descriptors can, in some cases, significantly outperform classical state-of-the-art ML and deep learning (DL) algorithms, when working using a specific benchmarking dataset for ML/DL-based VS. The performance of our quantum-powered LB-VS methodology is validated on superconducting quantum processors, using an ADRB2 benchmarking dataset as well as a novel dataset containing COVID-19 inhibitors [20] and reporting results in line with the numerical predictions obtained with classical numerical simulations. The proposed approach is not tied to a specific use case and can be applied to screen any digital database of active/inactive compounds against a desired target, using any third-party cheminformatics package.

Inspired by these promising results and confident in the scalability of the quantum approach, we also introduce the concept of 'prospective quantum advantage' (PQA) obtained when a quantum algorithm simulated on a classical computer or executed on state-of-the-art hardware (or any combination of the two) can provide – *at least in some relevant instances*—a tangible advantage compared to the best known classical algorithm operating with the same data sets (input information). In addition, PQA requires the absence of evident restrictions on extensions to larger problem sizes for which the quantum algorithms cannot be efficiently simulated on classical computers, while keeping the same—or even extend—the observed advantage. When demonstrations on larger quantum processors will become feasible, PQA could hence lead to a full effective Quantum Advantage.

The demonstration of PQA for a drug discovery workflow based on LB-VS is one of the main outcomes of the results presented in this work.

2. Methodology

We design a target-agnostic computational workflow to provide heuristic algorithmic evidence of quantum advantage over classical methods in LB-VS. More specifically, we use classical data to perform VS using a quantum SVC algorithm, hence evaluating its performance with respect to its classical counterparts. A like-for-like comparison between classical and quantum methods is fundamental to be as fair as possible in assessing the actual potential for quantum advantage.

2.1. Dataset

The general nature and the purpose of our classification workflow both require a realistic benchmarking dataset specifically dedicated to applied ML in VS. Specifically, the dataset should [21–24] (a) imitate real-world screening libraries and guide ML methods to discriminate active from inactive compounds; (b) contain molecules with unambiguous experimentally measured activity against a target, i.e. with either potent/moderate activity or inactivity; (c) have a realistic ratio between active and inactive molecules; (d) contain active and inactive compounds with comparable molecular properties; (e) be chemically unbiased.

The LIT-PCBA dataset [22] encompasses all the previous points for fifteen well-characterised biological targets, and it is a natural choice for this project. Furthermore, and with respect to prior efforts [21, 25–27], the LIT-PCBA dataset was created specifically to test the performance of ML tasks in VS of molecules. The dataset was prepared by removal of potential false positives as well as including experimentally confirmed inactive molecules rather than seeding the database with decoy molecules (i.e. molecules *assumed* to be inactive).

The targets in the LIT-PCBA datasets have a real-world active-to-inactive molecules ratio, meaning that in some cases the number of confirmed active molecules is strongly imbalanced. As ML methods can be severely affected by class imbalance in a dataset [28, 29], we balanced active-to-inactive ratios of each target dataset. However, for some targets in the LIT-PCBA dataset, the number of active molecules is astoundingly small (less than 30), with thousands of confirmed inactive molecules. In such cases, we padded the dataset with a 1:6 active-to-inactive ratio to balance the dataset (see appendix A).

Following the above procedure and considerations, we also choose to assess the performance of our method by screening a novel dataset containing known active and inactive COVID-19 inhibitors [20]. This dataset is particularly challenging for ML classification tasks, and specifically for those based on a purely structural basis (i.e. using a fingerprint-based approach), due to the fact that more than 30% of the active or inactive molecules do not share a common scaffold or other recurrent structural features, showing how different and diverse the active molecules are. In summary, while LIT-PCBA was chosen to provide a realistic benchmark of our methodology against a dataset specifically dedicated to applied ML in VS for very well known and studied diseases, the COVID-19 case study is instead meant as a less standardised but paradigmatic example in which lack of good quality experimental data and high class imbalance result in a difficult problem for classical support vector machine (SVM) kernel methods.

2.2. SVC and QK

SVMs [1, 3–5] are among the most widely used supervised ML methods for LB-VS. In its basic version, an SVC [30], is an SVM-based linear model that, given a training set of the form $\{(\vec{x}_i, y_i) | \vec{x}_i \in \mathbb{R}^m, y_i = \pm 1\}$, finds an optimal separating hyperplane in the data space to distinguish between two given classes, labelled by y_i. SVMs can be turned into non-linear classifiers by lifting the input data into a higher dimensional feature space $\mathcal{F}$ through a feature map $\phi(\vec{x})$. Indeed, while in its elementary form a SVC relies on the computation of similarities between data points through inner products $\langle \vec{x}_i, \vec{x}_j \rangle$ in $\mathbb{R}^m$, in the more general case this is replaced by the corresponding feature space quantity $\langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle$, thus allowing for complex non-linear representations of the original inputs. The SVC is usually trained by solving a constrained quadratic optimization problem, with a target function of the form

$$\mathcal{C}(\vec{\alpha}) = \sum_i \alpha_i - \sum_{ij} y_i y_j \alpha_i \alpha_j \langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle \tag{1}$$

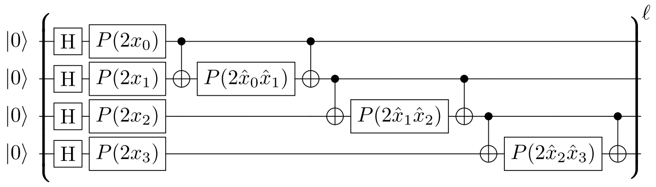
subject to $\sum_i \alpha_i y_i = 0$ and $0 \leqslant \alpha_i \leqslant 1/(2nC)$. Here n is the cardinality of the training set and C is a regularization parameter controlling the hardness of the margin. Moreover, the so-called kernel trick [31] allows one to replace all scalar products with a symmetric positive definite kernel function $k(\vec{x}_i, \vec{x}_j)$, such that, in general, the feature map ϕ is only specified in an implicit manner. Some popular choices for classical SVCs are represented by polynomial kernels $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + r)^d$, which reduce to the linear case for $d = 1$, and the Gaussian radial basis function (RBF) kernel $k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma |\vec{x}_i - \vec{x}_j|^2)$.

The family of QK methods [13, 14] employs a quantum computer to extend the class of available kernel functions, particularly to those instances which are believed to be hard to define and compute classically. Here, the complex-valued Hilbert space of quantum mechanical states plays the role of the feature space through a mapping of form $\vec{x} \mapsto |\phi(\vec{x})\rangle\langle\phi(\vec{x})|$ and a natural kernel choice is provided by

$k(\vec{x}_i, \vec{x}_j) = |\langle\phi(\vec{x}_i)|\phi(\vec{x}_j)\rangle|^2$. The specific nature and properties of the QK depend on the explicit choice of the embedding function $|\phi(\vec{x})\rangle$, which is typically realized as $|\phi(\vec{x})\rangle = U(\vec{x})|0\rangle$ with a unitary operation U acting on a reference state $|0\rangle$ and depending parametrically on the classical input data.

Some important remarks are in order here: on the one hand, it is not difficult to see how carefully designed U can in principle lead to classically intractable kernels. On the other hand, it has also been argued that the mere computational hardness of U does not *a priori* guarantee quantum advantage in speed or accuracy for ML tasks. In fact, complexity considerations in this context are influenced by both the availability of training data [17] and the inherent structure of the data sets [16, 32]. Although a rigorous quantum speed-up in supervised learning with QK methods has been proven theoretically in at least one specific example [16], the quest for quantum advantage on naturally occurring data sets remains open. Here we provide strong empirical evidence towards such a goal, working from the assumption that good candidate models for practical quantum advantage are those that, using a classically hard QK, exhibit superior performances with respect to all classical counterparts in at least some instances of the problem under study.

In the following, we will make use of a unitary template, known as the ZZ feature map with linear entanglement [14, 33], to embed classical feature vectors into quantum states. For a 4-dimensional classical input $\vec{x} = (x_0, x_1, x_2, x_3)$ this reads



(2)

where $P(\theta)$ is the single-qubit phase gate, $\hat{x}_i \equiv \pi - x_i$ and ℓ is called the depth of the feature map. Notice that the number of qubits in the register matches the size of the classical inputs, and the generalization of the parameterized unitary transformation to larger classical inputs is straightforward. This design is derived from the one presented in [14] and yields computationally hard kernels under suitable complexity theory assumptions. In the following, we will set $\ell = 2$.

For every pair of classical inputs $\vec{x}$, $\vec{x}'$, the QK is evaluated on a quantum processor according to the definition

$$k(\vec{x}, \vec{x}') = |\langle\phi(\vec{x})|\phi(\vec{x}')\rangle|^2 = |\langle 0|U^\dagger(\vec{x})U(\vec{x}')|0\rangle|^2. \tag{3}$$

In practice, after initializing the quantum register in the reference state $|0\rangle$, one applies the unitary $U(\vec{x}')$ and the inverse $U^\dagger(\vec{x}) = U^{-1}(\vec{x})$, which is easily obtained by reversing the order of operations appearing in equation (2) while also replacing $P(\theta) \to P(-\theta)$ in all single-qubit phase gates. Finally, the value of the QK corresponds to the probability of observing the register in the reference state $|0\rangle$ after a measure in the computational basis.

2.3. Molecular descriptors

Rather than performing VS based on a similarity search of molecules in a database—using for example molecular fingerprints, like most conventional VS methods—we trained the classifier using generic features (chemical descriptors), which can be computed for every dataset, regardless of the target type and experimental features. Chemical descriptors exist in their thousands and can be quickly obtained using specific third-party cheminformatics software tools, both open-source and commercial. Our methodology can be coupled with any third-party package capable of producing such descriptors. Here, we use RDKit [34] to extract molecular properties starting from molecular structures in a SMILES format. As also described in [7, 35–38], the chosen features can be grouped as (a) atomic species; (b) structural properties; (c) physical-chemical properties; (d) basic electronic information; (e) molecular complexity - i.e. graph descriptors. The full list of descriptors can be found in appendix B.

2.4. Descriptors selection methods and quantum encoding

As described in section 2.2, employing a ZZ feature map implies the use of a parametric type of encoding. Under this scheme, each classical data point must be represented as a vector of real numbers—in our case the chemical features or molecular descriptors—that, after a suitable normalization procedure, are used as angles in single-qubit quantum gates (see equation (2)). These operations, in combination with 2-qubit entangling operations, eventually create a quantum superposition state, i.e. a linear combination of multi-qubit basis states or strings, whose complex coefficients have a non-trivial dependence on the classical input and represent its embedding into the feature Hilbert space.

While in principle very powerful and versatile, this data encoding scheme is rather qubit-intensive, with typically one assigned feature per qubit. As a matter of fact, and despite continued and steady progress in quantum computing technology, implementing a quantum LB-VS workflow featuring hundreds of descriptors is currently demanding even for the largest publicly available quantum processors (127 qubits, IBM Eagle—late 2021 [39]). Similarly, this precludes the use of conventional 2048-bit molecular fingerprints, although reduced representations such as the ones proposed by Batra *et al* [40] may offer viable intermediate-scale solutions.

To achieve a meaningful yet manageable problem size, we therefore extracted, starting from SMILES representations, 47 chemical descriptors as described in section 2.3. First, each descriptor was normalised using standardization methods. Second, the descriptors were compressed into N variables using principal component analysis (PCA) [41] to match a set of N qubits used by the QSVC algorithm. Alternatively, the best N features were selected using the analysis of variance methodology (ANOVA) [42]. More in detail, PCA—which is widely adopted for feature reduction in drug discovery [36, 43] – was applied via singular value decomposition to obtain a lower-dimensional representation of the data as described in [44]. Conversely, the ANOVA f-test [44] enabled us to pick the most descriptive N features, as also suggested in [45], leveraging the numerical structure of the inputs and the binary categorical nature (i.e. active/inactive) of the classification output. Finally, the resulting feature vectors were used to parametrically initialise the corresponding quantum states on which the QK entries were evaluated.

2.5. Evaluation of performance (ROC)

The most intuitive way of assessing the performances of binary classification algorithms in both classical and quantum cases is to calculate the AUC-ROC (receiver operating characteristic) metric [44, 46–48]. The ROC score provides an indication of the capability of the method to distinguish accurately between true positives and false positives. The score ranges between 0 and 1, where 1 is a perfect classifier capable of retrieving true positives only.

3. Results

To demonstrate the theoretical benefits of a quantum-inspired LB-VS method, first we ran our experiments simulating noiseless quantum hardware with the python-based Qiskit [33] statevector simulator. Then, we performed simulations introducing statistical and gate noise of actual IBM processors (Qiskit qasm and mock hardware simulators). In parallel, we ran identical experiments using the best classical SVC (CSVC) kernel method with optimal hyperparameters (i.e. the regularization factor C and the RBF kernel non-linearity γ) obtained via a thorough grid search using the scikit-learn package (see appendix A) [44]. To further substantiate our heuristic claims, we performed LB-VS of the same benchmarking datasets using two other state-of-the-art classical methodologies whose performances for generalised VS tasks have been acknowledged in the literature [49–54]. Specifically, we used a random matrix discriminant (RMD) algorithm [55], as implemented in PyRMD [56], and a DL approach called Directed message passing neural network (MPNN) [57]. We compared the results of the following methodologies ‘out-of-the-box’ to establish a reference baseline that can be easily reproduced:

- D-MPNN with 2D CDF-normalized 200 additional features from RDKIT or with binary 2048-bit Morgan fingerprints [58];
- PyRMD with 2048-bit MHFP6 [59], ECFP6 [60] and RDKIT fingerprints.

Finally, we measured the performance of our QSVC method using actual quantum hardware (IBM Quantum Montreal and IBM Quantum Guadalupe) screening the ADBR2 and the COVID-19 datasets.

3.1. Results from numerical simulations

Following the methodology described in section 2, all targets of the LIT-PCBA and the COVID-19 datasets were screened using our algorithm. To assess the average method performance, each simulation involving both classical and quantum SVC algorithms was run ten times for each feature selection method and per number of features. Similarly, the RMD and DL classical simulations were also run 10 times. In the interest of fairness, we quantified the standard deviation of the results using ten-fold cross-validation for all simulations. In addition, we used identical test and train subsets for all SVC methods, in order to have a true like-for-like comparison between quantum and classical algorithms. The full set of numerical results is reported in tables 2 and 3 of appendix C.

Consistently with classical SVC methods, QSVC statevector results suggest that the overall performance of our method is influenced by (a) the class balance and the number of the actives in the dataset, (b) the feature selection method, (c) number of features and (d) the value of regularization parameter *C*. This trend is captured in figures 6 and 7 of appendix C. The class balance of the target is the first major factor to affect the performance of our method, similarly to other classical ML/DL approaches. A small number of active molecules in a dataset (e.g. 17 actives *vs* 312 483 inactives for ADBR2) leads to large standard deviation values (up to ca. ±0.3 of AUC ROC). On the other hand, a near negligible standard deviation for target ALDH1, which has the largest number of actives (5386 actives plus 103 474 inactives), confirms the importance of class balance in terms of stability and performance of all methods. The feature selection method is the second major factor to have an effect on the overall performance of all SVC methods described in this paper. Due to its very essence, feature selection involves the inevitable loss of information, and this may a positive or negative effect depending on the dataset/target and the number of features selected. Throughout this work, we observe that our QSVC method tends to outperform CSVC and the other classical methods when then number of features is 8 or higher, which also corresponds to a higher number of qubits. In some cases, such as the ALDH1 target shown in figure 2(a), there is a near-linear correlation between performance and the number of features/qubits used. Finally, the performance of QSVC method is also affected by the choice of the regularization factor *C*. This is expected as SVC based methods have an inherent dependency on the *C* value to determine the misclassification rate of the method. Our results suggest that a better QSVC performance is achieved when using a default value $C = 1$.

A more detailed overview is represented in figure 2, where the comparison of ANOVA and PCA QSVC(statevector)/CSVC, RMD and DL trends is reported for the ALDH1, COVID-19 (see section 2.1) and GBA (166 active *vs* 296 052 inactive molecules) targets. Notice that the trend lines of RMD and DL methods are flat since the methods do not make use of an explicit number of features or feature selection methods. Overall, the performances of both classical and quantum SVC classifiers tend to increase with an incremented number of features used for training, with the steepness of increase and maximum ROC values depending on the features selection method used. In other words, the amount of information used to train the algorithm has a major impact on determining the quality of the classifier. figure 2(a) shows that the QSVC tends to perform significantly better with 8+ features if compared to its classical equivalent (up to 13% better when using 16 features), suggesting that after this threshold the effect of dimensionality reduction starts to vanish. This results in a better performance of the QSVC algorithm, possibly due to the larger size of the quantum feature space and a better expressivity of the quantum feature map, leading to a more favourable separation hyperplane between active and inactive classes. Importantly, comparing the classical and quantum methodologies, we notice that the performance of QSVC either remains stable or keeps increasing with an increasing number of features, thus hinting towards potential for quantum advantage. As previously discussed, the stability of the method on datasets with optimal class balance such as ALDH1 is suggested by little standard deviation values. Conversely, class-imbalanced datasets such as COVID-19 and GBA (figures 2(c) and (d)) typically show higher standard deviations. Nevertheless, the QSVC method consistently keeps higher average accuracy with increasing number of features for at least one choice of the feature selection method (PCA for COVID-19 and ANOVA for GBA).

In figure 2 we also report the mean ROC values obtained following VS of the ALDH1, COVID-19 and GBA targets using the RMD and MPNN methodologies, trained using various fingerprint encoding. The performance of our quantum classifier seems to outperform the classical VS methodology of RMD with up to ca. 20% better mean quantum ROC values when compared to the best value obtained with PyRMD (trained with RDKIT or ECFP6 fingerprints). Conversely, the MPNN method generally provides better mean ROC values if compared to RMD, and it seems to be outperformed by our methodology only when using a larger number of features. The standard deviation values of both RMD and MPNN methodologies are in agreement with our CSVC and QSVC methods, confirming the role of class balance in defining the stability of the method. In figures 2(c) and (d) we include data points obtained for 8 features using two different quantum simulator backends as implemented in Qiskit, namely the noiseless qasm and the mock Montreal backend simulators. The former adds statistical noise to the simulation, mimicking the probabilistic quantum measurement process, while the latter simulates both the statistical and hardware noise that one would get on the actual IBM Quantum Montreal processor, whose calibration data are used as parameters in the noise model. Due to memory and time limitations, we run simulations with qasm and/or gate noise for 8 features/qubits. These results show that the addition of noise is not significantly detrimental to the performance of the QSVC method for the COVID-19 and GBA targets. Furthermore, the standard deviation for the simulations are also in line with the noiseless statevector case.

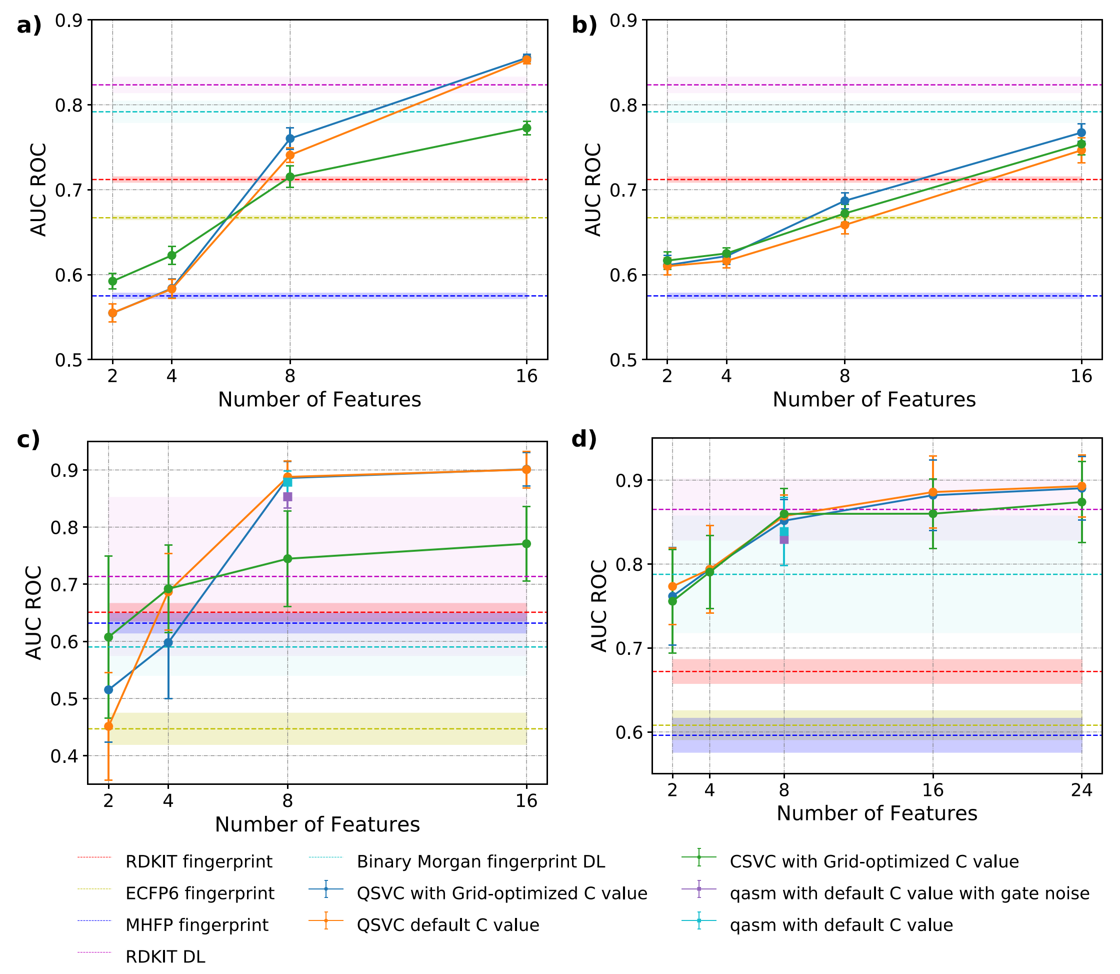


Figure 2. Mean AUC ROC values plotted against the number of features used to train the model. (a) ALDH1 target with PCA feature reduction method, (b) ALDH1 target with ANOVA feature reduction method, (c) COVID-19 target with PCA feature reduction method, (d) GBA target with ANOVA feature reduction method. In all panels, we compare the results obtained with CSVC, QSVC and the other classical methods, including standard deviations (bars for our CSVC and QSVC, shaded lines for RMD and MPNN). Flat lines represent the mean AUC ROC values of classical methods RMD and MPNN, which are independent of the features number and the features selection method. Panels (c) and (d) also report average AUC ROC using QSVC algorithm trained with eight features and using a quantum backend numerical simulator, accounting for statistical noise (Qiskit qasm simulator, cyan) and hardware noise (Montreal backend simulator, purple).

Summarizing, the simulation of our quantum LB-VS methodology provides, in a like-for-like comparison, a performance that either compares to other existing classical methodologies or that significantly outperforms them for a sufficiently large number of selected features, sometimes showing evidence of potential linear scale-up. This implies that an increase in the number of features/qubits used to train the algorithm can, in principle, further improve the performance of the method. Finally, it is important to mention that the PQA suggested by this study is also supported by the geometric test recently introduced in [17]. In fact, the data sets shown in figure 2 noticeably passed such test, yielding $g_{CQ} \propto \sqrt{n}$, where g_{CQ} is the geometric difference between the QK and the best classical CSVC counterpart, and n is the number of the data points. Overall, the observed win-rate of QSVC compared to classical counterparts is 43.7% (16 datasets), with variability depending on the features selection method and learning rate.

3.2. Results from quantum hardware

In this work, we have so far established that the QSVC method in some cases outperforms an equivalent binary classifier or other classical methods used in VS of databases of molecules, highlighting its dependencies from dataset size, number of features and so forth. We also showed that the addition of statistical and simulated noise to the quantum algorithm does not considerably reduce the performance of the method, and it lies within the standard deviation of statevector (i.e. noise-free) simulations. The next logical step is therefore to verify whether the quantum algorithm can indeed perform well on actual quantum devices and if the numerical performance is a reliable tool for assessing the potential benefits coming from our method applied to LB-VS on quantum computers. To this aim, we run simulations on IBM Quantum

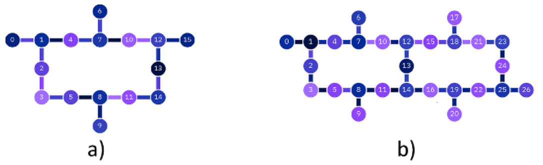


Figure 3. IBM Quantum processors layout, (a) IBM Quantum Guadalupe : 16 Qubits, 32 QV, 2.4 K CLOPS Falcon r4P processor, (b) IBM Quantum Montreal: 27 Qubits, 128 QV, 2 K CLOPS Falcon r4 processor.

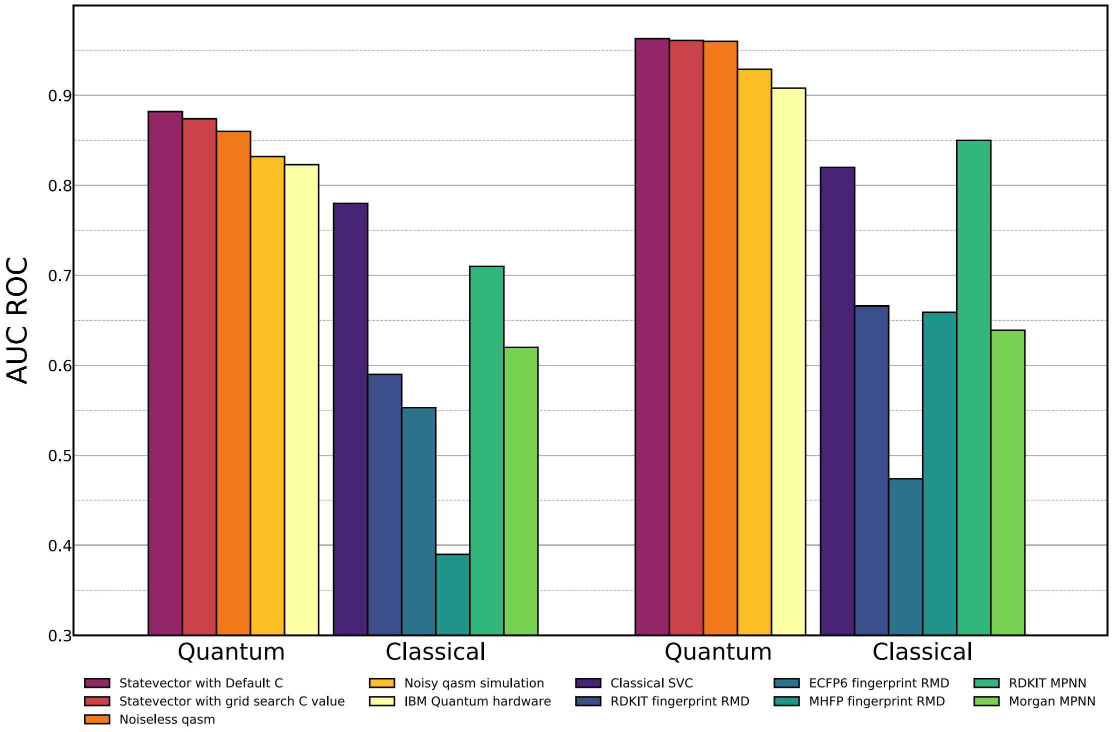


Figure 4. Quantum hardware results (single AUC-ROC measurement, in brown), obtained using the IBM Quantum Montreal (ADRB2 - ANOVA, left) and Guadalupe (COVID-19 - PCA, right) quantum processors using 8 qubit and corresponding 8-dimensional feature vectors. Other mean AUC-ROC results were obtained via classical and quantum QSVC, RMD and MPNN numerical simulations are also reported, following the trends discussed in section 3.1.

processors using 8 qubits for ADRB2 on IBM Quantum Montreal (figure 3(b)) and for the COVID-19 dataset on IBM Quantum Guadalupe (figure 3(a)).

The two datasets underwent the same preparation and simulation methodology as the other datasets described in the previous section. In figure 4 we report the best results obtained with the optimal feature selection methods (ANOVA for ADRB2, PCA for COVID-19) along with qasm, mock hardware (Montreal and Guadalupe backend) and hardware results. Also in this case, QSVC statevector numerical simulations for both ADRB2 and COVID-19 datasets provided better results if compared to classical equivalents, including RMD and MPNN methods. Remarkably, in our experiments we found that training and testing our QSVC algorithm on IBM Quantum Montreal provided an AUC ROC result for ADRB2 comparable to the results of all the other (quantum) numerical simulation methods, with outcomes lying well within the error bar of statevector simulations (see table 2). Similarly, running the QSVC algorithm on IBM Quantum Guadalupe for the COVID-19 dataset also provided an AUC ROC that is comparable to the best (quantum) numerical simulations. Most importantly, quantum hardware results for both targets greatly outperform the other classical methodologies, thus confirming that QK methods are indeed a well suited tool for potential active/inactive molecules classification against ADRB2 and COVID-19 targets.

We can conclude that, for these two instances, simulated backends have correctly predicted the respective hardware results, hence providing a truthful insight into what to expect from real quantum hardware. These results are encouraging, suggesting that although it might not yet be possible to perform large scale simulations on currently available quantum hardware, promising proof-of-principle demonstrations can be achieved, confirming the reliability and predictive power of numerical simulations.

4. Discussion and conclusions

In this paper, we have shown that, when using the same data sets and training conditions, a quantum classifier can in some cases outperform the best available classical counterparts. This implies that the analysis of cheminformatics data can already benefit from the application of quantum technologies, which have the potential of enhancing the success rate of, e.g. structure classification for material design and drug discovery. We need to stress that for this particular investigation, we reduced the problem size (number of features) to a level that is currently affordable for quantum calculations, using both classical simulations of quantum algorithms and the execution on state-of-the-art quantum hardware. This implies for the moment the use of relatively small data sets and the selection of a limited number of representative features for the characterization of the compound properties. In doing that, the proposed quantum classifiers are still classically accessible in simulations, hence our approach cannot yet lead to direct quantum advantage—at best, it can be qualified as an interesting new quantum-inspired classical algorithm. However, and this is the main message of this paper, there are currently no evident reasons to doubt that the same advantage observed in the proof-of-principle applications exposed in this work will not withstand the scaling-up to a larger number of descriptors (features). These will correspond to a number of qubits (more than 30–50) that will remain inaccessible to classical simulators while becoming manageable on the near-term quantum computers of the coming generations [39]. In fact, we not only observe instances where a quantum approach outperforms classical counterparts for a single choice of reduced problem size, but we also report that such advantage shows a persisting trend when the number of features is increased, approaching the original full dimension. To refer to the class of algorithms featuring this property, we introduce the concept of PQA, as an impactful, although fully heuristic, intermediate step—with potential value for small and intermediate size instances—towards full quantum advantage in large scale applications. In this sense, we conclude that quantum information algorithms have already an attractive potential for the chemical and pharmaceutical industry, as they bring new value to research workflows and provide solutions superior to previously accessible classical ones.

Data availability statement

The data and software that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by the Hartree National Centre for Digital Innovation, a UK Government-funded collaboration between STFC and IBM. IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at www.ibm.com/legal/copytrade.

Appendix A. Algorithm workflow

The algorithm is divided as follows (see also figure 5).

(a) *Preprocessing step.* The dataset of molecules (encoded in a SMILES format) is loaded and 47 molecular descriptors are extracted using RDKit. A new dataset (80% training and 20% validating) is then created and normalised, containing the extracted features per molecule. A novel dataset is created for each new set of reduced features and per reduction method. In order to overcome over-fitting we balanced the dataset. The balancing of the dataset has been done by randomly under-sampling the data points from the majority class (in our case the inactive molecules) for the sake of generalisation, whilst repeating the measurements with new random subsets for repeatability of results. We avoided up-sampling techniques for the active molecules since the addition of artificial data-points, with similar chemical structure to existing active molecules can lead to opposite desired effects, e.g. single addition or substitution of atoms in the molecular structure can lad to dramatically different biological effects. To overcome bias towards one class, we padded the dataset aiming to have equal amounts of active and inactive molecules. However, when the number of actives is very small (e.g. 30 or less), a 1:1 padding resulted in a very small number of data points, hence generally resulting in less powerful and less accurate ML algorithms. In such cases we choose to balance the number of inactive molecules to six times the number of actives,

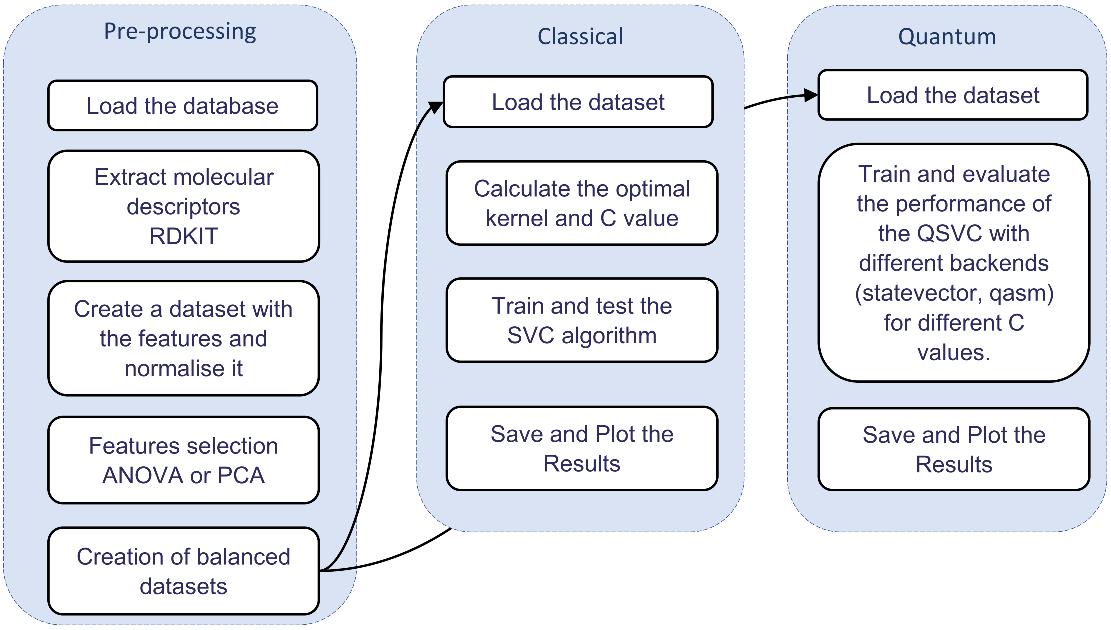

Pre-processing
Load the database
Extract molecular descriptors RDKIT
Create a dataset with the features and normalise it
Features selection ANOVA or PCA
Creation of balanced datasets
Classical
Load the dataset
Calculate the optimal kernel and C value
Train and test the SVC algorithm
Save and Plot the Results
Quantum
Load the dataset
Train and evaluate the performance of the QSVC with different backends (statevector, qasm) for different C values.
Save and Plot the Results

Figure 5. Schematic representation of the algorithm workflow. In light blue are represented the three main steps of the workflow: preprocessing, classical SVC and Quantum SVC.

therefore giving us a number of data points in the order of hundreds. The 1:6 ratio was selected empirically as the smallest amount of padding producing stable results (i.e. with a limited sensitivity to variations in the training dataset) whilst avoiding overfitting.

(b) *Classical SVC step.* Using the training portion, a Grid-Search is performed over C and γ parameters as well as kernel type (Poly, RFB, Linear). The best C, γ and kernel type are then used to train the SVC algorithm. Therefore, the algorithm is validated over the validating portion and the AUC ROC is measured.

(c) *Quantum SVC.* The training set for each reduced features dataset obtained in step 2 is re-used to compare Quantum and Classical SVC results. Here, the training is performed by the calculation of an QK via the ZZ feature Map. Two computations are run in parallel. The first uses the same C and γ parameters obtained in step 2 and used by the Classical SVC algorithm (a like-for-like comparison). The second uses default Qiskit C and γ values. Validation of algorithm performance follows the training and AUC ROC is measured. Steps 2 and 3 have been repeated ten times per dataset to verify the stability of our method.

Appendix B. Molecular descriptors

The classifier has been trained using generic chemical descriptors as features. A generalised description of a molecule can be done by using as little as five descriptors classes. The atomic species are a collection of descriptors that simply identify the number and kind of atoms composing a molecule. The structural properties family of descriptors provides rough information about the molecular scaffold, that is total number of single and double bonds, presence of aromatic rings and so forth. Physical-chemical descriptors provide information about properties such as molecular weight and lipophilicity, both fundamental to determine the drug-like properties of small molecules. The descriptor family providing basic electronic information aims at roughly describing the electronic behaviour of the molecule, such as reactivity, polarity and rough electron density (i.e. not determined using accurate electronic structure methods such as Density Functional Theory). Finally, molecular complexity descriptors measure the complexity of every atom environment in a molecule and is an intrinsic feature that ultimately depends only on the molecular structure but, nevertheless, provides an estimator for the synthetic effort. More information about each descriptor can be found here [38]. All the molecular descriptors used in our work are presented in table 1.

Table 1. Table containing all the molecular descriptors used in this work. All the descriptors were extracted using the RDKit package.

Category	Molecular descriptor
Atomic species	C
	N
	O
	P
	S
	F
	Cl
	Br
	I
Structural properties	Single_Bonds
	Double_Bonds
	NumStereoE
	Num_Aromatic_Atoms
	Aromatic_Proportion
	NumRotatableBonds
	Total_NH_OH
	Total_N_O
	NumHydrogenAcceptors
	NumHydrogenDonors
	NumofHeteroatoms
Physical-chemical properties	MolLogP
	MolWt
Basic electronic information	FpDensityMorgan1
	FpDensityMorgan2
	FpDensityMorgan3
	MaxAbsPartialCharge
	MinAbsPartial Charge
	NumValenceElectrons
Molecular complexity	BertzCT
	BalabanJ
	Chi0
	Chi0n
	Chi0v
	Chi1
	Chi1n
	Chi1v
	Chi2n
	Chi2v
	Chi3n
	Chi3v
	Chi4n
	Chi4v
	HallKierAlpha
	Ipc
	Kappa1
	Kappa2
	Kappa3

Appendix C. LIT-PCBA screening results

In this section, we collect numerical results for all LIT-PBA sub-sets and COVID-19 dataset, calculated using CSVC, QSVC via Qiskit statevector simulator, PyRMD and MPNN. Results are divided by features selection method (ANOVA/PCA).

Table 2. Average AUC ROC values for 15 targets of the LIT-PCBA dataset and the COVID-19 dataset, obtained using our classical and quantum SVC method (with **ANOVA** features selection) and fingerprint similarity using PyRMD.

	CSVC					QSVC ANOVA (Default C)					QSVC ANOVA (CSVC C values)					PyRMD			ChemProp (MPNN)	
Target	2	4	8	16	24	2	4	8	16	24	2	4	8	16	24	RDKIT	ECFP6	MHFP	RDKIT	Morgan
ADRB2	0.508 (±0.223)	0.661 (±0.172)	0.652 (±(0.138)	0.632 (±0.228)	0.549 (±0.214)	0.465 (±0.305)	0.732 (±0.127)	0.777 (±0.0874)	0.815 (±0.0944)	0.546 (±0.302)	0.507 (±0.242)	0.759 (±0.109)	0.746 (±0.121)	0.812 (±0.089)	0.617 (±0.282)	0.524 (±0.061)	0.453 (±0.057)	0.359 (±0.031)	0.498 (±0.218)	0.497 (±0.127)
ALDH1	0.617 (±0.01)	0.625 (±0.007)	0.672 (±0.011)	0.754 (±0.013)	n/a	0.61 (±0.01)	0.616 (±0.009)	0.659 (±0.01)	0.747 (±0.015)	n/a	0.611 (±0.011)	0.622 (±0.01)	0.687 (±0.009)	0.767 (±0.01)	n/a	0.712 (±0.003)	0.667 (±0.002)	0.575 (±0.003)	0.824 (±0.009)	0.792 (±0.012)
ESR1_ago	0.346 (±0.210)	0.425 (±0.278)	0.557 (±0.227)	0.497 (±0.234)	0.486 (±0.177)	0.362 (±0.267)	0.382 (±0.155)	0.415 (±0.145)	0.402 (±0.18)	0.493 (±0.22)	0.406 (±0.211)	0.455 (±0.269)	0.488 (±0.174)	0.414 (±0.164)	0.483 (±0.231)	0.528 (±0.041)	0.496 (±0.05)	0.585 (±0.026)	0.618 (±0.315)	0.344 (±0.203)
ESR1_ant	0.646 (±0.109)	0.636 (±0.072)	0.715 (±0.046)	0.688 (±0.037)	0.729 (±0.055)	0.693 (±0.086)	0.635 (±0.076)	0.78 (±0.041)	0.816 (±0.033)	0.822 (±0.047)	0.689 (±0.088)	0.557 (±0.108)	0.75 (±0.059)	0.797 (±0.047)	0.818 (±0.048)	0.661 (±0.013)	0.477 (±0.022)	0.596 (±0.011)	0.768 (±0.071)	0.700 (±0.104)
FEN1	0.698 (±0.038)	0.787 (±0.027)	0.835 (±0.024)	0.843 (±0.028)	0.880 (±0.031)	0.705 (±0.044)	0.795 (±0.031)	0.826 (±0.027)	0.855 (±0.016)	0.871 (±0.03)	0.697 (±0.038)	0.772 (±0.034)	0.761 (±0.047)	0.85 (±0.016)	0.869 (±0.028)	0.755 (±0.012)	0.735 (±0.01)	0.686 (±0.01)	0.952 (±0.014)	0.893 (±0.015)
GBA	0.756 (±0.062)	0.790 (±0.043)	0.859 (±0.030)	0.860 (±0.041)	0.874 (±0.048)	0.773 (±0.0455)	0.794 (±0.052)	0.857 (±0.025)	0.886 (±0.043)	0.893 (±0.037)	0.762 (±0.058)	0.794 (±0.052)	0.852 (±0.025)	0.882 (±0.042)	0.890 (±0.038)	0.672 (±0.014)	0.608 (±0.017)	0.596 (±0.02)	0.865 (±0.036)	0.788 (±0.069)
IDH1	0.491 (±0.138)	0.482 (±0.134)	0.553 (±0.084)	0.752 (±0.085)	0.708 (±0.081)	0.623 (±0.133)	0.563 (±0.168)	0.658 (±0.088)	0.427 (±0.16)	0.678 (±0.08)	0.628 (±0.166)	0.547 (±0.17)	0.557 (±0.161)	0.426 (±0.141)	0.673 (±0.086)	0.692 (±0.015)	0.484 (±0.028)	0.504 (±0.031)	0.725 (±0.12)	0.669 (±0.113)
KAT2A	0.587 (±0.031)	0.629 (±0.043)	0.647 (±0.053)	0.684 (±0.064)	0.653 (±0.056)	0.594 (±0.025)	0.621 (±0.051)	0.666 (±0.062)	0.683 (±0.047)	0.616 (±0.044)	0.588 (±0.028)	0.57 (±0.094)	0.62 (±0.054)	0.684 (±0.034)	0.624 (±0.039)	0.568 (±0.022)	0.546 (±0.014)	0.475 (±0.011)	0.729 (±0.048)	0.636 (±0.098)
MAPK1	0.693 (±0.052)	0.71 (±0.028)	0.72 (±0.033)	0.716 (±0.026)	0.699 (±0.039)	0.674 (±0.064)	0.690 (±0.034)	0.741 (±0.043)	0.738 (±0.034)	0.667 (±0.024)	0.675 (±0.059)	0.653 (±0.056)	0.717 (±0.071)	0.73 (±0.042)	0.663 (±0.025)	0.578 (±0.006)	0.565 (±0.01)	0.524 (±0.009)	0.768 (±0.043)	0.709 (±0.057)
MTORC1	0.586 (±0.077)	0.611 (±0.098)	0.702 (±0.079)	0.652 (±0.068)	0.678 (±0.077)	0.575 (±0.119)	0.633 (±0.045)	0.628 (±0.085)	0.604 (±0.14)	0.600 (±0.112)	0.613 (±0.074)	0.574 (±0.097)	0.615 (±0.081)	0.605 (±0.14)	0.609 (±0.095)	0.62 (±0.014)	0.576 (±0.02)	0.547 (±0.015)	0.729 (±0.102)	0.682 (±0.13)
OPRK1	0.801 (±0.07)	0.809 (±0.116)	0.897 (±0.04)	0.887 (±0.101)	0.914 (±0.087)	0.693 (±0.184)	0.693 (±0.192)	0.789 (±0.125)	0.867 (±0.095)	0.93 (±0.053)	0.789 (±0.094)	0.771 (±0.157)	0.866 (±0.043)	0.891 (±0.079)	0.929 (±0.052)	0.778 (±0.024)	0.594 (±0.046)	0.55 (±0.007)	0.790 (±0.246)	0.728 (±0.275)
PKM2	0.729 (±0.033)	0.738 (±0.033)	0.747 (±0.028)	0.788 (±0.035)	0.785 (±0.037)	0.73 (±0.029)	0.731 (±0.032)	0.737 (±0.028)	0.783 (±0.024)	0.761 (±0.022)	0.594 (±0.043)	0.654 (±0.03)	0.677 (±0.034)	0.742 (±0.022)	0.767 (±0.019)	0.678 (±0.005)	0.647 (±0.008)	0.55 (±0.007)	0.813 (±0.036)	0.786 (±0.042)
PPARG	0.508 (±0.194)	0.702 (±0.07)	0.829 (±0.052)	0.767 (±0.075)	0.836 (±0.035)	0.544 (±0.265)	0.701 (±0.078)	0.775 (±0.072)	0.749 (±0.06)	0.666 (±0.197)	0.665 (±0.157)	0.692 (±0.056)	0.794 (±0.055)	0.739 (±0.059)	0.661 (±0.195)	0.654 (±0.023)	0.701 (±0.019)	0.692 (±0.018)	0.837 (±0.209)	0.728 (±0.211)
TP53	0.64 (±0.14)	0.654 (±0.101)	0.725 (±0.102)	0.756 (±0.082)	0.766 (±0.063)	0.693 (±0.104)	0.649 (±0.088)	0.827 (±0.078)	0.858 (±0.035)	0.844 (±0.052)	0.66 (±0.118)	0.541 (±0.144)	0.826 (±0.076)	0.849 (±0.044)	0.849 (±0.048)	0.609 (±0.026)	0.536 (±0.02)	0.503 (±0.018)	0.717 (±0.144)	0.712 (±0.116)
VDR	0.684 (±0.03)	0.683 (±0.032)	0.709 (±0.034)	0.765 (±0.027)	0.808 (±0.03)	0.675 (±0.026)	0.684 (±0.03)	0.7 (±0.036)	0.75 (±0.023)	0.754 (±0.027)	0.68 (±0.027)	0.676 (±0.032)	0.684 (±0.036)	0.733 (±0.028)	0.755 (±0.018)	0.747 (±0.009)	0.674 (±0.009)	0.634 (±0.007)	0.888 (±0.027)	0.829 (±0.028)
COVID-19	0.476 (±0.091)	0.613 (±0.147)	0.65 (±0.068)	0.693 (±0.046)	n/a	0.517 (±0.137)	0.639 (±0.129)	0.683 (±0.058)	0.804 (±0.067)	n/a	0.532 (±0.096)	0.703 (±0.063)	0.700 (±0.088)	0.755 (±0.09)	n/a	0.651 (±0.015)	0.447 (±0.027)	0.632 (±0.017)	0.713 (±0.137)	0.590 (±0.049)

Table 3. Average AUC ROC values for 15 targets of the LIT-PCBA dataset and the COVID-19 dataset, obtained using our classical and quantum SVC method (with **PCA** features selection) and fingerprint similarity using PyRMD and ChemProp (MPNN).

	CSVC					QSVC PCA (Default C)					QSVC PCA (CSVC C values)					PyRMD			ChemProp (MPNN)	
Target	2	4	8	16	24	2	4	8	16	24	2	4	8	16	24	RDKIT	ECFP6	MHFP	RDKIT	Morgan
ADRB2	0.483 (±0.168)	0.499 (±0.181)	0.752 (±0.188)	0.634 (±0.234)	0.682 (±0.115)	0.412 (±0.122)	0.435 (±0.175)	0.364 (±0.124)	0.380 (±0.235)	0.424 (±0.297)	0.436 (±0.096)	0.440 (±0.198)	0.370 (±0.124)	0.380 (±0.235)	0.425 (±0.297)	0.524 (±0.061)	0.453 (±0.057)	0.359 (±0.031)	0.498 (±0.218)	0.497 (±0.127)
ALDH1	0.592 (±0.009)	0.623 (±0.011)	0.715 (±0.013)	0.773 (±0.008)	n/a	0.555 (±0.011)	0.583 (±0.011)	0.741 (±0.009)	0.853 (±0.005)	n/a	0.555 (±0.011)	0.584 (±0.011)	0.76 (±0.012)	0.855 (±0.004)	n/a	0.712 (±0.003)	0.667 (±0.002)	0.575 (±0.003)	0.824 (±0.009)	0.792 (±0.012)
ESR1_ago	0.519 (±0.207)	0.458 (±0.224)	0.607 (±0.161)	0.549 (±0.168)	0.44 (±0.154)	0.486 (±0.199)	0.457 (±0.164)	0.591 (±0.196)	0.539 (±0.133)	0.585 (±0.200)	0.470 (±0.216)	0.485 (±0.169)	0.581 (±0.204)	0.526 (±0.136)	0.585 (±0.200)	0.528 (±0.041)	0.496 (±0.05)	0.585 (±0.026)	0.618 (±0.315)	0.344 (±0.203)
ESR1_ant	0.603 (±0.086)	0.707 (±0.06)	0.79 (±0.073)	0.819 (±0.065)	0.813 (±0.062)	0.463 (±0.069)	0.618 (±0.122)	0.734 (±0.061)	0.763 (±0.054)	0.824 (±0.048)	0.489 (±0.067)	0.620 (±0.122)	0.735 (±0.061)	0.762 (±0.054)	0.824 (±0.048)	0.661 (±0.013)	0.477 (±0.022)	0.596 (±0.011)	0.768 (±0.071)	0.700 (±0.104)
FEN1	0.619 (±0.042)	0.808 (±0.025)	0.835 (±0.029)	0.888 (±0.026)	0.915 (±0.027)	0.612 (±0.025)	0.629 (±0.051)	0.763 (±0.019)	0.799 (±0.024)	0.809 (±0.035)	0.609 (±0.025)	0.629 (±0.051)	0.763 (±0.019)	0.798 (±0.024)	0.810 (±0.035)	0.755 (±0.012)	0.735 (±0.01)	0.686 (±0.01)	0.952 (±0.014)	0.893 (±0.015)
GBA	0.756 (±0.052)	0.785 (±0.044)	0.882 (±0.033)	0.899 (±0.032)	0.900 (±0.028)	0.717 (±0.054)	0.654 (±0.054)	0.781 (±0.047)	0.804 (±0.039	0.831 (±0.034)	0.717 (±0.054)	0.660 (±0.047)	0.781 (±0.046)	0.805 (±0.037)	0.831 (±0.034)	0.672 (±0.014)	0.608 (±0.017)	0.596 (±0.02)	0.865 (±0.036)	0.788 (±0.069)
IDH1	0.523 (±0.084)	0.563 (±0.074)	0.718 (±0.082)	0.728 (±0.069)	0.824 (±0.078)	0.505 (±0.096)	0.507 (±0.152)	0.616 (±0.127)	0.705 (±0.082)	0.575 (±0.180)	0.539 (±0.074)	0.509 (±0.119)	0.595 (±0.123)	0.701 (±0.08)	0.573 (±0.178)	0.692 (±0.015)	0.484 (±0.028)	0.504 (±0.031)	0.725 (±0.12)	0.669 (±0.113)
KAT2A	0.532 (±0.081)	0.579 (±0.051)	0.627 (±0.095)	0.677 (±0.044)	0.668 (±0.054)	0.478 (±0.054)	0.464 (±0.061)	0.602 (±0.051)	0.627 (±0.043)	0.632 (±0.06)	0.489 (±0.064)	0.53 (±0.092)	0.601 (±0.052)	0.627 (±0.043)	0.632 (±0.061)	0.568 (±0.022)	0.546 (±0.014)	0.475 (±0.011)	0.729 (±0.048)	0.636 (±0.098)
MAPK1	0.675 (±0.055)	0.682 (±0.067)	0.701 (±0.032)	0.728 (±0.041)	0.741 (±0.038)	0.615 (±0.042)	0.630 (±0.048)	0.600 (±0.035)	0.633 (±0.038)	0.654 (±0.046)	0.611 (±0.046)	0.591 (±0.069)	0.599 (±0.033)	0.633 (±0.038)	0.651 (±0.045)	0.578 (±0.006)	0.565 (±0.01)	0.524 (±0.009)	0.768 (±0.043)	0.709 (±0.057)
MTORC1	0.559 (±0.12)	0.557 (±0.077)	0.641 (±0.072)	0.667 (±0.075)	0.677 (±0.076)	0.437 (±0.055)	0.418 (±0.052)	0.604 (±0.141)	0.468 (±0.095)	0.570 (±0.078)	0.429 (±0.039)	0.439 (±0.067)	0.663 (±0.058)	0.468 (±0.095)	0.577 (±0.081)	0.62 (±0.014)	0.576 (±0.02)	0.547 (±0.015)	0.729 (±0.102)	0.682 (±0.13)
OPRK1	0.822 (±0.088)	0.782 (±0.106)	0.811 (±0.099)	0.852 (±0.082)	0.840 (±0.074)	0.743 (±0.147)	0.646 (±0.15)	0.801 (±0.107)	0.763 (±0.143)	0.65 (±0.13)	0.722 (±0.195)	0.65 (±0.161)	0.801 (±0.115)	0.756 (±0.14)	0.653 (±0.125)	0.778 (±0.024)	0.594 (±0.046)	0.55 (±0.007)	0.790 (±0.246)	0.728 (±0.275)
PKM2	0.646 (±0.039)	0.75 (±0.033)	0.784 (±0.033)	0.801 (±0.038)	0.787 (±0.023)	0.59 (±0.044)	0.659 (±0.033)	0.68 (±0.032)	0.743 (±0.023)	0.768 (±0.019)	0.594 (±0.043)	0.654 (±0.03)	0.677 (±0.034)	0.742 (±0.022)	0.767 (±0.019)	0.678 (±0.005)	0.647 (±0.008)	0.55 (±0.007)	0.813 (±0.036)	0.786 (±0.042)
PPARG	0.602 (±0.101)	0.691 (±0.096)	0.834 (±0.037)	0.778 (±0.062)	0.797 (±0.052)	0.643 (±0.072)	0.559 (±0.114)	0.577 (±0.222)	0.591 (±0.25)	0.504 (±0.259)	0.559 (±0.144)	0.561 (±0.11)	0.571 (±0.22)	0.594 (±0.251)	0.495 (±0.257)	0.654 (±0.023)	0.701 (±0.019)	0.692 (±0.018)	0.837 (±0.209)	0.728 (±0.211)
TP53	0.581 (±0.133)	0.69 (±0.074)	0.738 (±0.055)	0.724 (±0.092)	0.771 (±0.063)	0.738 (±0.103)	0.705 (±0.071)	0.821 (±0.035)	0.795 (±0.055)	0.842 (±0.057)	0.745 (±0.108)	0.709 (±0.064)	0.821 (±0.04)	0.795 (±0.055)	0.844 (±0.056)	0.609 (±0.026)	0.536 (±0.02)	0.503 (±0.018)	0.717 (±0.144)	0.712 (±0.116)
VDR	0.649 (±0.037)	0.712 (±0.026)	0.754 (±0.021)	0.817 (±0.032)	0.833 (±0.015)	0.578 (±0.028)	0.652 (±0.03)	0.717 (±0.033)	0.786 (±0.025)	0.809 (±0.027)	0.568 (±0.03)	0.644 (±0.036)	0.716 (±0.034)	0.786 (±0.025)	0.811 (±0.026)	0.747 (±0.009)	0.674 (±0.009)	0.634 (±0.007)	0.888 (±0.027)	0.829 (±0.028)
COVID-19	0.608 (±0.142)	0.692 (±0.077)	0.745 (±0.084)	0.771 (±0.065)	n/a	0.451 (±0.094)	0.687 (±0.067)	0.888 (±0.028)	0.901 (±0.032)	n/a	0.515 (±0.091)	0.598 (±0.098)	0.886 (±0.029)	0.901 (±0.029)	n/a	0.651 (±0.015)	0.447 (±0.027)	0.632 (±0.017)	0.713 (±0.137)	0.590 (±0.049)

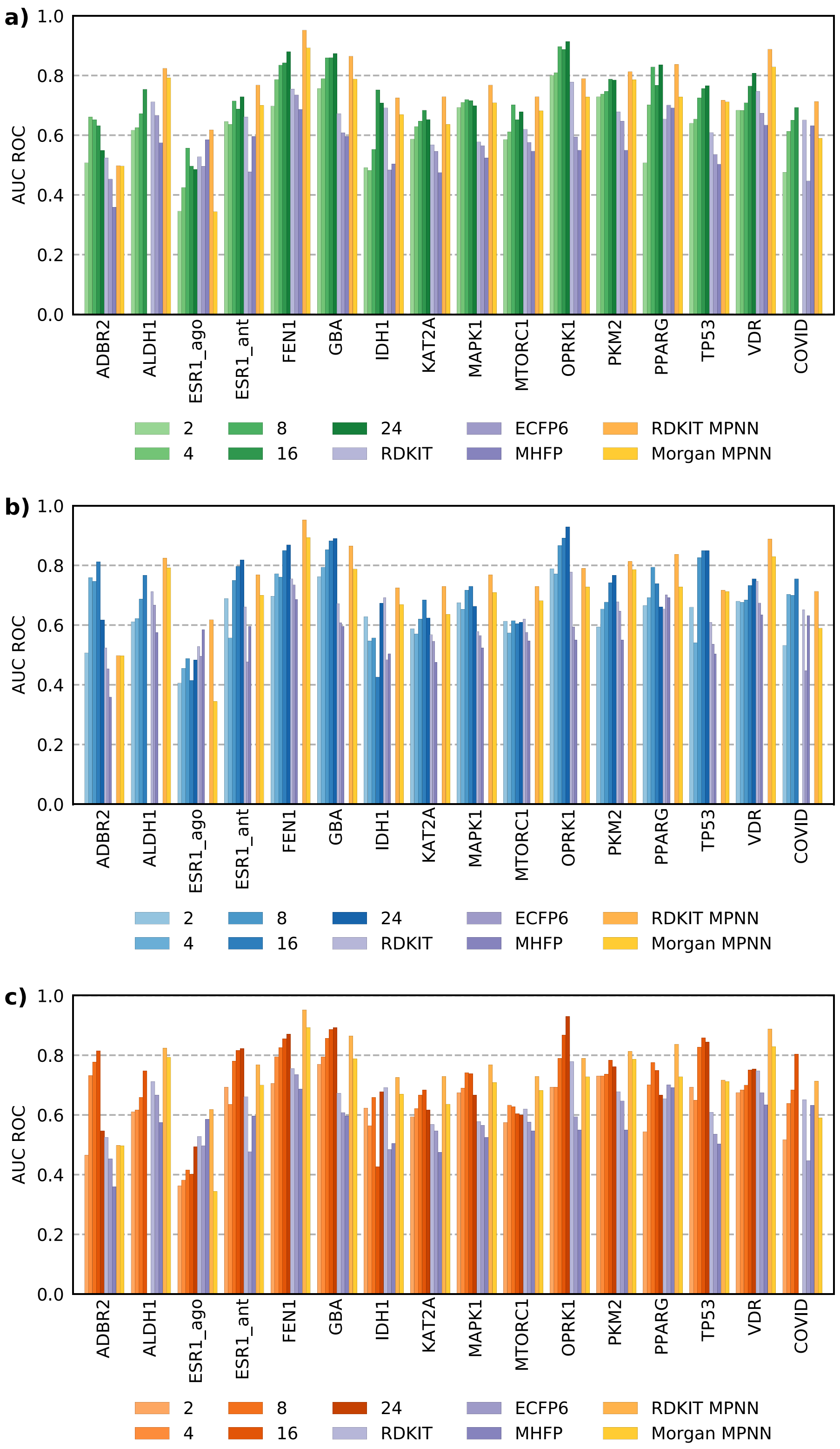


Figure 6. Graphical representation of mean AUC ROC results (standard deviation omitted for ease of representation) for benchmark LIT-PBCA and COVID-19 dataset from table 2 using **ANOVA** features selection. (a) Mean CSVC AUC ROC, per feature (green) and against RMD (purple) and MPNN (yellow) results. (b) Mean QSVC tuned with CSVC *C* parameter (blue). (c) Mean QSVC tuned with default *C* parameter (red).

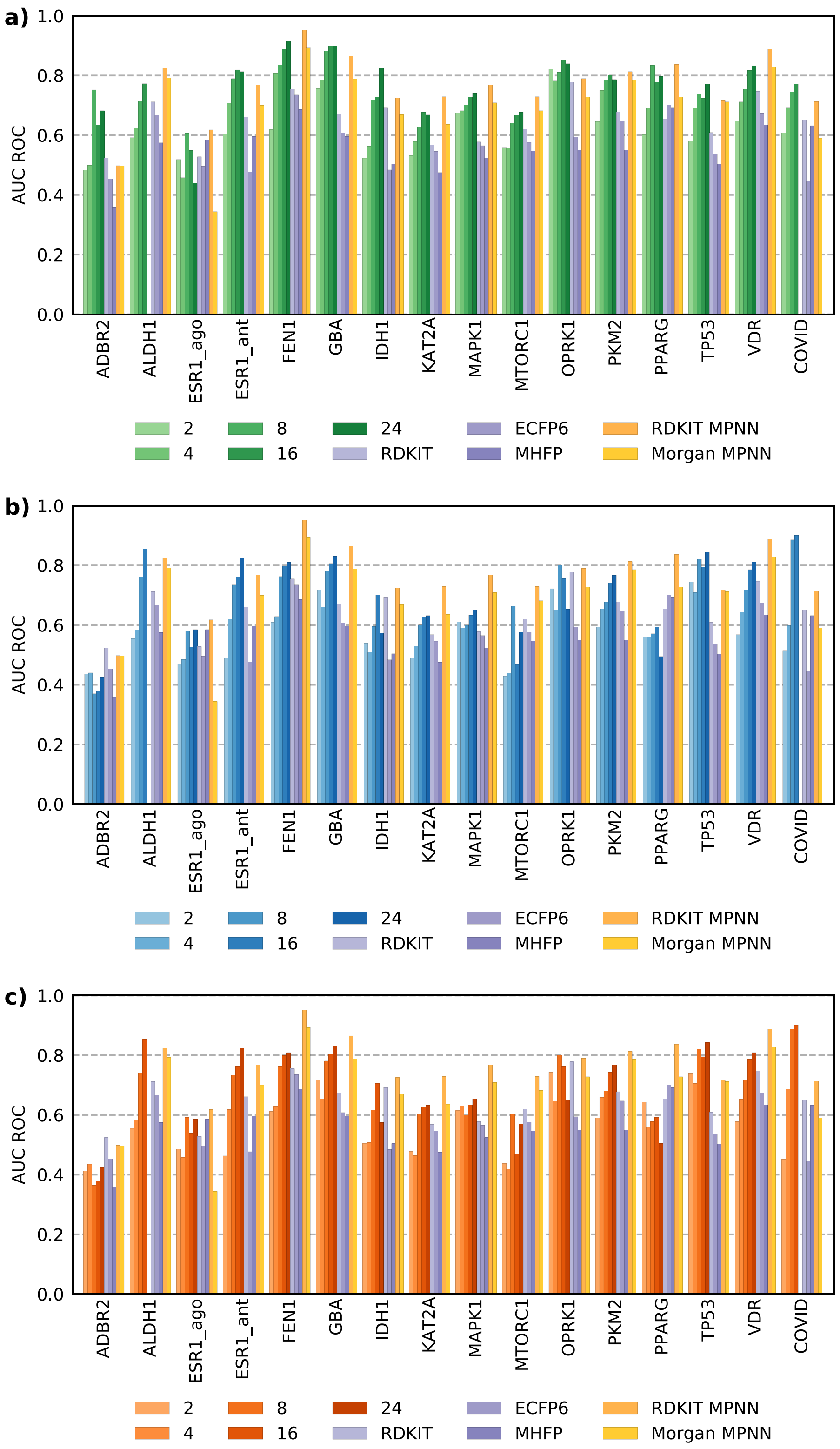


Figure 7. Graphical representation of mean AUC ROC results (standard deviation omitted for ease of representation) for benchmark LIT-PBCA and COVID-19 datasets from table 3, using **PCA** features selection. (a) Mean CSVC AUC ROC, per feature (green) and against RMD (purple) and MPNN (yellow) results. (b) Mean QSVC tuned with CSVC *C* parameter (blue). (c) Mean QSVC tuned with default *C* parameter (red).

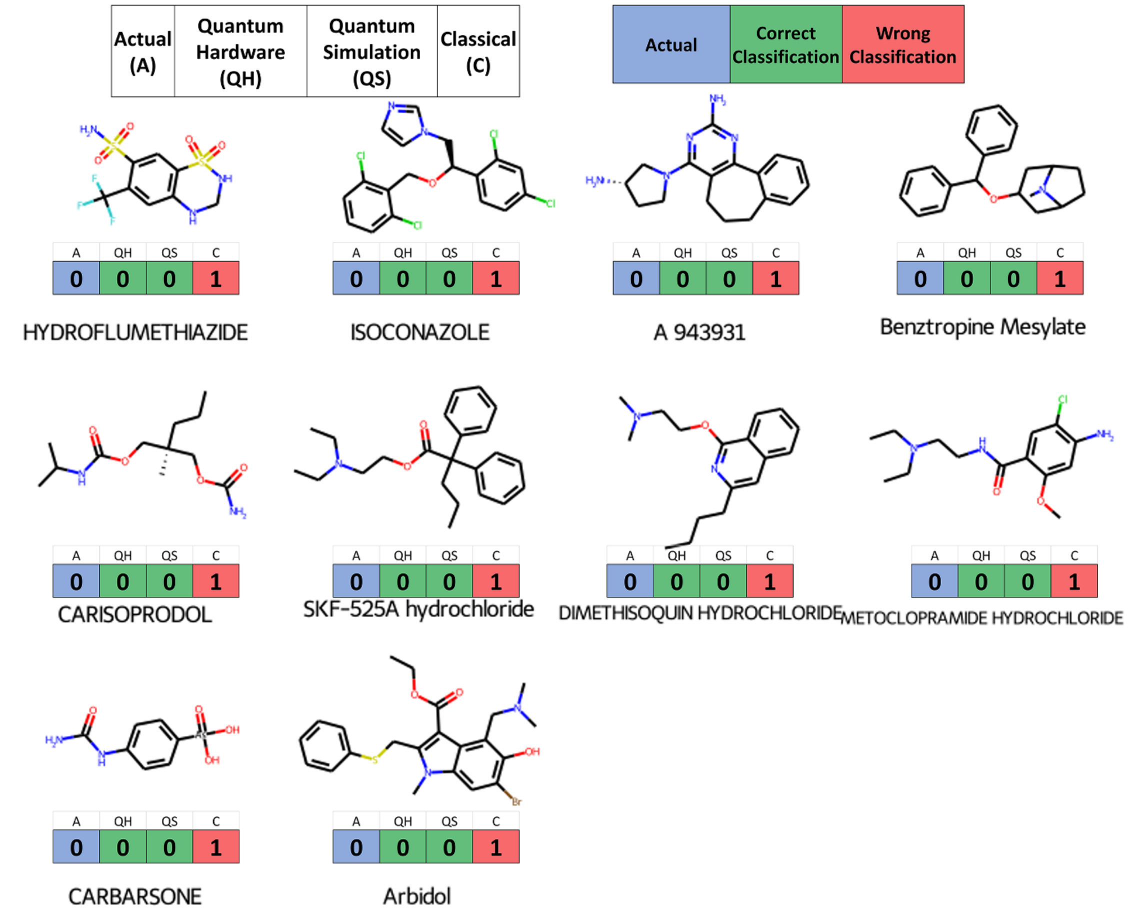


Figure 8. Inactive molecules from the COVID-19 dataset wrongly labelled at screening time by either QSVC or CSVC.

Appendix D. Example of screened molecules from COVID-19 dataset

In figures 8 and 9, we present a sub-portion of the screened COVID-19 dataset using QSVC and CSVC methods (8 features used for training the models, as seen in figures 2 and 4). This sub-portion is particularly insightful in showing how quantum and classical SVC methods differ in classifying molecules from the same dataset. Molecules not reported here (475) have been correctly classified by both methods. We remind the reader that the COVID-19 dataset used in this work is a preliminary one, prone to errors. For example, active molecules tested *in vitro* have been later on proven to not be effective drugs against the disease, such as Ivermectin [61, 62], whereas other such as Remdesivir are being currently used to treat COVID-19 patients [63]. Interestingly, both our quantum and classical methods considered Ivermectin to be inactive, although according to the dataset Ivermectin is considered as active. For the Remdesivir case, the quantum method has selected the molecule as an active candidate whereas the classical method has failed. Finally, we want to stress that all the misclassification of inactive molecules occurred at screening time, were made by the classical algorithm, meaning that in this instance the quantum algorithm is less prone to give false positives if compared to the classical one. We would like to stress that these results are highly dependent on the dataset, the methodologies employed and the explored region in parameters space. Here we only report the data for our quantum algorithm and the best classical SVC method used in this work.

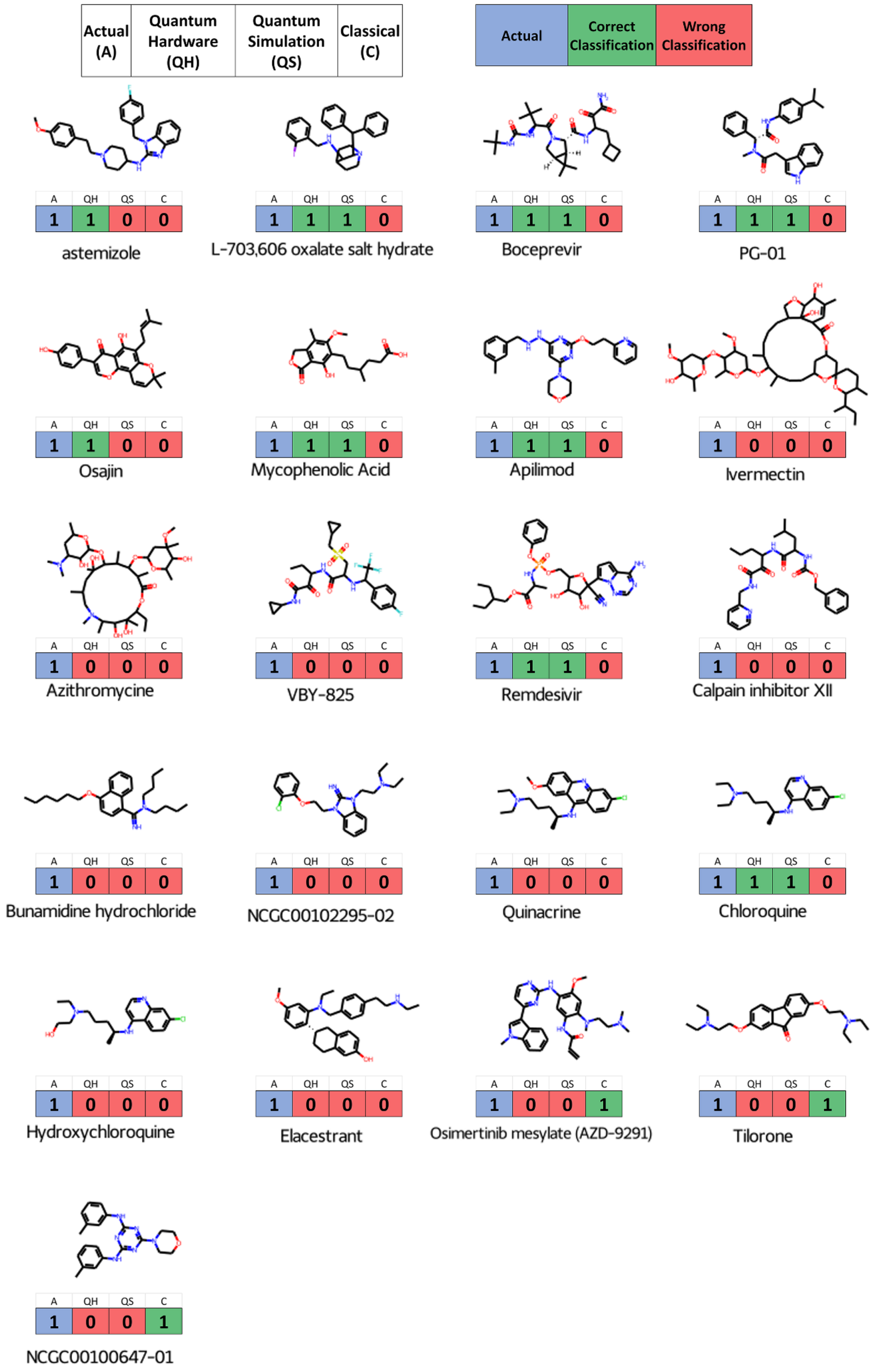


Figure 9. Active molecules from the COVID-19 dataset wrongly labelled at screening time by either QSVC or CSVC.

ORCID iDs

Stefano Mensa https://orcid.org/0000-0002-0938-144X
Emre Sahin https://orcid.org/0000-0002-5996-0407
Francesco Tacchino https://orcid.org/0000-0003-2008-5956
Panagiotis Kl Barkoutsos https://orcid.org/0000-0001-9428-913X
Ivano Tavernelli https://orcid.org/0000-0001-5690-1981

References

[1] Jorissen R N and Gilson M K 2005 *J. Chem. Inf. Model.* **45** 549
[2] Hert J, Willett P, Wilton D J, Acklin P, Azzaoui K, Jacoby E and Schuffenhauer A 2006 *J. Chem. Inf. Model.* **46** 462
[3] Plewczynski D, Spieser S and Koch U 2009 *Comb. Chem. High Throughput Screen.* **12** 358
[4] Sato T, Yuki H, Takaya D, Sasaki S, Tanaka A and Honma T 2012 *J. Chem. Inf. Model.* **52** 1015
[5] Lavecchia A 2015 Machine-learning approaches in drug discovery: Methods and applications *Drug Discov. Today* **20** 318–31
[6] Korotcov A, Tkachenko V, Russo D P and Ekins S 2017 *Mol. Pharm.* **14** 4462
[7] Lo Y C, Rensi S E, Torng W and Altman R B 2018 Machine learning in chemoinformatics and drug discovery *Drug Discov. Today* **23** 1538–46
[8] Yang X, Wang Y, Byrne R, Schneider G and Yang S 2019 Concepts of artificial intelligence for computer-assisted drug discovery *Chem. Rev.* **119** 10520–94
[9] Walters W P and Wang R 2020 *J. Chem. Inf. Model.* **60** 4109
[10] Patel L, Shukla T, Huang X, Ussery D W and Wang S 2020 *Molecules* **25** 5277
[11] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195
[12] Mangini S, Tacchino F, Gerace D, Bajoni D and Macchiavello C 2021 *Europhys. Lett.* **134** 10002
[13] Schuld M and Killoran N 2019 *Phys. Rev. Lett.* **122** 040504
[14] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 *Nature* **567** 209
[15] Wu S L *et al* 2021 *Phys. Rev. Res.* **3** 033221
[16] Liu Y, Arunachalam S and Temme K 2021 *Nat. Phys.* **17** 1013
[17] Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 *Nat. Commun.* **12** 2631
[18] Kübler J M, Buchholz S and Schölkopf B 2021 arXiv:2106.03747 [quant-ph]
[19] Cao Y, Romero J and Aspuru-Guzik A 2018 *IBM J. Res. Dev.* **62** 1
[20] Gawriljuk V O *et al* 2021 Machine learning models identify inhibitors of SARS-CoV-2 *J. Chem. Inf. Model.* **61** 4224–35
[21] Sieg J, Flachsenberg F and Rarey M 2019 *J. Chem. Inf. Model.* **59** 947
[22] Tran-Nguyen V K, Jacquemard C and Rognan D 2020 *J. Chem. Inf. Model.* **60** 4263
[23] Rohrer S G and Baumann K 2009 *J. Chem. Inf. Model.* **49** 169
[24] Xia J, Tilahun E L, Reid T E, Zhang L and Wang X S 2015 *Methods* **71** 146
[25] Huang N, Shoichet B K and Irwin J J 2006 *J. Med. Chem.* **49** 6789
[26] Chaput L, Martinez-Sanz J, Saettel N and Mouawad L 2016 *J. Cheminform.* **8** 1
[27] Wallach I and Heifets A 2018 *J. Chem. Inf. Model.* **58** 916
[28] Bishop C M 2006 *Pattern Recognition and Machine Learning* (New York: Springer) p 738
[29] Murphy K P 2012 *Machine Learning, a Probabilistic Perspective* 1st edn (Cambridge, MA: The MIT Press) p 1096
[30] Cortes C and Vapnik V 1995 *Mach. Learn.* **20** 273
[31] Aizerman M A, Braverman E A and Rozonoer L 1964 *Autom. Remote Control* **25** 821–37
[32] Glick J R, Gujarati T P, Corcoles A D, Kim Y, Kandala A, Gambetta J M and Temme K 2021 arXiv:2105.03406 [quant-ph]
[33] Anis M S 2021 Qiskit: an open-source framework for quantum computing https://doi.org/10.5281/zenodo.2573505
[34] RDKit: open-source cheminformatics software (available at: www.rdkit.org)
[35] Todeschini R and Consonni V 2000 *Handbook of Molecular Descriptors* (*Methods and Principles in Medicinal Chemistry* vol 11) (Weinheim: Wiley) p 688
[36] Bajorath J 2002 *Nat. Rev. Drug Discovery* **1** 882
[37] Hong H, Xie Q, Ge W, Qian F, Fang H, Shi L, Su Z, Perkins R and Tong W 2008 *J. Chem. Inf. Model.* **48** 1337
[38] Sawada R, Kotera M and Yamanishi Y 2014 *Mol. Inform.* **33** 719
[39] Gambetta J 2020 Ibm's roadmap for scaling quantum technology (available at: https://research.ibm.com/blog/ibm-quantum-roadmap)
[40] Batra K, Zorn K M, Foil D H, Minerali E, Gawriljuk V O, Lane T R, and Ekins S 2021 Quantum machine learning algorithms for drug discovery applications *J. Chem. Inf. Model.* **61** 2641–7
[41] Wold S, Esbensen K and Geladi P 1987 *Chemometr. Intell. Lab. Syst.* **2** 37
[42] Sthle L and Wold S 1989 Analysis of variance (ANOVA) *Chemometr. Intell. Lab. Syst.* **2** 37–52
[43] Elbadawi M, Gaisford S and Basit A W 2021 *Drug Discov. Today* **26** 769
[44] Pedregosa F *et al* 2011 *J. Mach. Learn. Res.* **12** 2825 (available at: http://jmlr.org/papers/v12/pedregosa11a.html)
[45] Kuhn M and Johnson K 2019 *Chapman and Hall/CRC* (Abingdon: Taylor& Francis) p 310
[46] Fawcett T 2006 *Pattern Recognit. Lett.* **27** 861
[47] Hand D J and Till R J 2001 *Mach. Learn.* **45** 171
[48] McClish D K 1989 *Med. Decis. Mak.* **9** 190
[49] Heid E and Green W H 2021 *J. Chem. Inf. Model.* **16** 15
[50] Stokes J M *et al* 2020 *Cell* **180** 688
[51] Bartók A P, De S, Poelking C, Bernstein N, Kermode J R, Csányi G and Ceriotti M 2017 *Sci. Adv.* **3** e1701816
[52] Nguyen D D, Xiao T, Wang M and Wei G W 2017 *J. Chem. Inf. Model.* **57** 1715
[53] David L, Thakkar A, Mercado R and Engkvist O 2020 Molecular representations in AI-driven drug discovery: a review and practical guide *J. Cheminform.* **12** 56
[54] Coley C W, Eyke N S and Jensen K F 2020 Autonomous discovery in the chemical sciences part II: outlook (arXiv:2003.13755)

[55] Lee A A, Brenner M P and Colwell L J 2016 *Proc. Natl Acad. Sci. USA* **113** 13564

[56] Amendola G and Cosconati S 2021 PyRMD: a new fully automated ai-powered ligand-based virtual screening tool *J. Chem. Inf. Model.* **61** 3835–3845

[57] Yang K *et al* 2019 *J. Chem. Inf. Model.* **59** 3370

[58] Morgan H L 1965 *J. Chem. Doc.* **5** 107

[59] Probst D and Reymond J L 2018 *J. Cheminform.* **10** 1

[60] Rogers D and Hahn M 2010 *J. Chem. Inf. Model.* **50** 742

[61] Bramante C T *et al* 2022 *New Engl. J. Med.* **387** 599

[62] Roman Y M, Burela P A, Pasupuleti V, Piscoya A, Vidal J E and Hernandez A V 2022 *Clin. Infect. Dis.* **74** 1022

[63] Gottlieb R L *et al* 2022 *New Engl. J. Med.* **386** 305