doi.org/10.26434/chemrxiv.12781232.v1

Quantum Machine Learning for Drug Discovery

Kushal Batra, Kimberley M. Zorn, Daniel H. Foil, Eni Minerali, Victor O. Gawriljuk, Thomas R. Lane, sean ekins

Submitted date: 08/08/2020 • Posted date: 10/08/2020

Citation information: Batra, Kushal; Zorn, Kimberley M.; Foil, Daniel H.; Minerali, Eni; Gawriljuk, Victor O.; Lane, Thomas R.; et al. (2020): Quantum Machine Learning for Drug Discovery. ChemRxiv. Preprint. https://doi.org/10.26434/chemrxiv.12781232.v1

The growing public and private datasets focused on small molecules screened against biological targets or whole organisms [1] provides a wealth of drug discovery relevant data. Increasingly this is used to create machine learning models which can be used for enabling target-based design [2-4], predict on- or off-target effects and create scoring functions [5,6]. This is matched by the availability of machine learning algorithms such as Support Vector Machines (SVM) and Deep Neural Networks (DNN) that are computationally expensive to perform on very large datasets and thousands of molecular descriptors. Quantum computer (QC) algorithms have been proposed to offer an approach to accelerate quantum machine learning over classical computer (CC) algorithms, however with significant limitations. In the case of cheminformatics, one of the challenges to overcome is the need for compression of large numbers of molecular descriptors for use on QC. Here we show how to achieve compression with datasets using hundreds of molecules (SARS-CoV-2) to hundreds of thousands (whole cell screening datasets for plague and M. tuberculosis) with SVM and data re-uploading classifier (a DNN equivalent algorithm) on a QC benchmarked against CC and hybrid approaches. This illustrates a quantum advantage for drug discovery to build upon in future.

File list (4)

Article

Quantum Machine Learning for Drug Discovery

Kushal Batra[1], Kimberley M. Zorn[2], Daniel H. Foil[2], Eni Minerali[2], Victor O. Gawriljuk[3]

Thomas R. Lane[2] & Sean Ekins[2*]

[1] Computer Science, NC State University, Raleigh, NC 27606, USA.

[2]Collaborations Pharmaceuticals, Inc., 840 Main Campus Drive, Lab 3510, Raleigh, NC 27606, USA.

[3]São Carlos Institute of Physics, University of São Paulo, Av. João Dagnone, 1100 - Santa Angelina, São Carlos - SP, 13563-120, Brazil

*To whom correspondence should be addressed. E-mail: sean@collaborationspharma.com

Phone: 215-687-1320

The growing public and private datasets focused on small molecules screened against biological targets or whole organisms [1] provides a wealth of drug discovery relevant data. Increasingly this is used to create machine learning models which can be used for enabling target-based design [2-4], predict on- or off-target effects and create scoring functions [5,6]. This is matched by the availability of machine learning algorithms such as Support Vector Machines (SVM) and Deep Neural Networks (DNN) that are computationally expensive to perform on very large datasets and thousands of molecular descriptors. Quantum computer (QC) algorithms have been proposed to offer an approach to accelerate quantum machine learning over classical computer (CC) algorithms, however with significant limitations. In the case of cheminformatics, one of the challenges to overcome is the need for compression of large numbers of molecular descriptors for use on QC. Here we show how to achieve compression with datasets using hundreds of molecules (SARS-CoV-2) to hundreds of thousands (whole cell screening datasets for plague and *M. tuberculosis*) with SVM and data re-uploading classifier (a DNN equivalent algorithm) on a QC benchmarked against CC and hybrid approaches. This illustrates a quantum advantage for drug discovery to build upon in future.

The pharmaceutical industry and academic laboratories are increasingly using and exploring machine learning applications in drug discovery to mine and model their data generated from years of high throughput screening [7]. This is allowing rapid identification of molecules for neglected diseases such as Ebola [8] and Chagas disease [9], presenting lead compounds which can then be moved rapidly into *in vivo* models [10-12] and this approach can be more widely applied. Recent examples have illustrated the speed with which the machine learning combined with *in vitro* testing continuum can generate new leads compared to traditional efforts [13]. The availability of thousands of structure activity datasets, (some of which in turn contain data for hundreds of thousands of molecules screened against a single target or organism [14]), presents a computational challenge with machine learning methods such as classifiers like SVM on CC [15]. Recently the potential of quantum machine learning has been illustrated using two methods such as a variational quantum circuit and a quantum kernel estimator [16]. In addition, new machine learning methods for quantum computing continue to be developed [17,18] which creates opportunities for expanding useful applications. Like many emerging technologies, QC has been proposed as likely to transform early stage pharmaceutical research and development as well as provide a potential solution for computationally intractable problems that would be traditionally performed on a CC in order to learn from data [19,20]. One area of interest in early drug discovery is virtual screening and optimization where small molecules are frequently described by fingerprint descriptors which can lead to tens of thousands of vectors called multiple fingerprint features (MFF) [21]. While this may be important for many aspects of applying machine learning to chemistry [22], it also creates significant challenges when using these descriptors on a QC. Herein we describe how we have applied multiple approaches to compress the descriptors for QC while also demonstrating applications to drug discovery datasets on a range of scales that would be broadly applicable. We also describe hybrid approaches that merge QC with CC for machine

learning applied to these datasets curated from public sources for drug discovery. These include 132 small molecule inhibitors of SARS-CoV-2 in Vero cells [23], 18,886 inhibitors of *Mycobacterium tuberculosis* [24] as well as several larger datasets for inhibitors of Krabbe disease (β-Galactocerebrosidase, Pubchem Assay 1159614, 44,809 compounds), Cathepsin B (Pubchem Assay 453, 63,331 compounds), Plague (Yersinia pestis, Pubchem Assay 898, 139,861 compounds), a larger dataset for *M. tuberculosis* (293,937 compounds) [25] and hERG (306,587 compounds) [14]. All these datasets were curated and prepared using Assay Central™ [24] (See Methods).

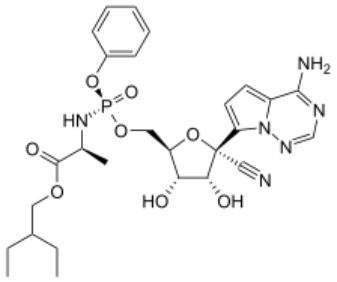
As the extended connectivity fingerprint radius 6 (ECFP6) has been widely applied in cheminformatics [26] we used the Morgan Fingerprint (with radius 3) which is equivalent to the ECFP6 fingerprint in RDkit [27]. The Morgan Fingerprint generates binary numbers whose default size is 2048 bits. While this is acceptable for use on CC, it is not acceptable for QC which requires a smaller number. Also, even with the increase in size/usage of qubits on the QC the network accuracy fails because of decoherence noise introduced in the qubit system with this big system [28]. Even having no linkage or communication between every qubit adds further noise to this system. We attempted two approaches to solve this utilizing QC alone or a hybrid where part of the code is run on the QC.

The initial algorithm we have compared is SVM implemented using the Qiskit library [29] which uses Least-Square SVM (LS-SVM) [30]. Qiskit chooses $M_{ij} = x_i\, x_j$ as the ansatz where M_{ij} is the kernel matrix and $x_i\, x_j$ are the datapoints in the dataset [31]. When working with QC we first formulate the ansatz and try to minimize it. This ansatz formulates the SVM hyperplane which divides the datasets. We have chosen the ansatz such that we get a line as an output. Changing the ansatz equation decides the shape the hyperplane will take. Within the QC, a

quantum algorithm is used to solve the direct product [30] and calculate the M(the kernel matrix), then a quantum algorithm can be used to transform to waveforms to solve the system of linear equations [31]. This approach solves the complete SVM on a QC. While SVM generates good results, it takes considerable time to solve linear equations to solve for Kernel Matrix, using feature maps (~1 minute for each datapoint) and hence, any time advantage for QC is not achievable.

In order to reduce the molecular descriptor features such that they can be represented and stored in our limited number of qubits (53 qubits for the ibmq-rochester, Fig. S1). We have initially demonstrated the application of QC with a SARS-CoV-2 Vero cell inhibition dataset consisting of just 132 compounds of which 66 were found to be active with the IC_{50} activity threshold of 6.65 μM [23]. We reduced the molecular descriptor dimensions to 2-3 and we have hence proposed multiple techniques to encode 2048 features. Method 1 used Principal Component Analysis (PCA) that is widely applied in data compression [32]. PCA resulted in accuracy rates on a CC of 37% using the kernel algorithm = RBF (Radial Basis Function) whereas on a QC accuracy was 33% (N= 3). Method 2 used a common dimension reduction technique of Linear Discriminant Analysis (LDA) which also considers the target class along with the predictors [33]. In this we simply take projection of points into some other hyperplane. LDA resulted in accuracy rates of 40% using a kernel algorithm on a CC while on QC this was 39% (N = 3). In Method 3, we designed an algorithm where we divide 2048 bits into groups of ‘k’ bits and convert these k bits into respective and convert it into decimal. This process is repeated until all the bits are converted to decimal such that k should completely divides 2048. An example of a molecule compressed using this algorithm is shown in Fig. 1. This resulted in a greatly improved accuracy rate of 61% on a CC using kernel algorithm=’poly’ and 59.6% on a QC (n=3).

Fig. 1. An example of a molecule and its compression using Method 3. 2D representation of the antiviral Remdesivir, and an illustration of the ECFP6 descriptor for Remdesivir and the compressed output.



An example: Take k= 128 bits (as 2048 completely divided by 128)

For Remdesivir ECPF6 =

[01000001000000000000000010000000010010000001011000000000000000000100000000000000
10000000000000000000000000000000100001000000000100000000000000000010000000000000
00000000000000000000000000000010000000000000000000000000000000000100000000
00000000000000000000000000000000000000010000000001000000000000000000000001
00000000000000000000000000000001000100000000000000001000000000000000010000000
00000000100010000000000100001010000000000000000000000000010000000000000000000
01001000000000000000000000000000010000000000000000000000000000000000000000000
00000000000000000000000000000000000000000010000000000000100000000000000000000
00010000010000000000000000100000000000000001001000000000000000000000000000010
00000000000000000000000000000000000000010000000000000000000100000000000010
00000000000000000000000000000000000000010000000000000000000000000000000000
00010000000000000000000000000000000000010000001000000000000000000000000000000
00000000000000001000000000000000000000000010000]

Using the above algorithm, we get the following compressed output =

[4683744163430531072, 9223512774343164416, 9223389629040820224,
9223372036858970112, 68753031168, 2305843009356300320, 2251836321452032,
281475014459392, 36028797018963968, 536879104, 9147937279969536, 2199023255553,
2199040032768, 70368744177668, 2164260864, 17179869200]

Method 4 uses an algorithm where we keep track of positions of 1 in the whole array. So, if we take the same example of Remdesivir, we store the position of 1's and 0's. we achieve this by iterating over the molecular descriptor and storing 1's value using **value = index * (-1^index) + value where index is the position of 1 or 0.** This way we get two values [-21086,-502690] where -21086 is storing the positions of 1's and -502690 is storing the positions of 0's. Using this approach, we get two dimensions. Even after feeding this data

from PCA we will get back the same results, as from PCA we were only concerned with the top two features. All the algorithms discussed above try to retain most of the information from the molecular descriptors as much as possible. This resulted in accuracy rates on CC of 59% using kernel algorithm='sigmoid' and on QC of 59.25% (with n=2).

These 4 methods were implemented on the QC and then tried as combinations and the accuracy was compared with CC (Table 1). The most promising results were obtained when combining Method 1 and Method 3 (k=128). The improved results for k=128 are due to the increased spread (more variance) compared to other k values when combined with PCA (Fig, 2). This makes it easier to obtain a hyperplane separating the two classes (active or not active). The Kernel matrix obtained from running the SVM model on a QC is shown in Fig. 3.

Table 1. The accuracy rates achieved with different combinations of methods. Method 1 (PCA), Method 2 (LDA), Method 3 and Method-4 using SVM on classical and quantum computer.

Combination	**Accuracy on Quantum Computer (%)**	**Accuracy on Classical Computer (%)**
PCA + Method 3(k=64)	<30	47
PCA + Method 3(k=128)	62.9	65
PCA + Method 3(k=256)	59.25	60
PCA + Method 3(k=512)	41	44
LDA + Method 3(k=64)	31	36
LDA + Method 3(k=128)	<30	35
LDA + Method 3(k=256)	<30	31
LDA + Method 3(k=512)	32	59
PCA + Method 4(2)	59.25	60

LDA + Method 4(2)	57	57

Fig. 2. The spread of the data for the SARS-CoV-2 dataset after applying Method 1 and Method 3. "x1" and "x2" are the top two features respectively (for all the compounds) obtained after applying these two methods.

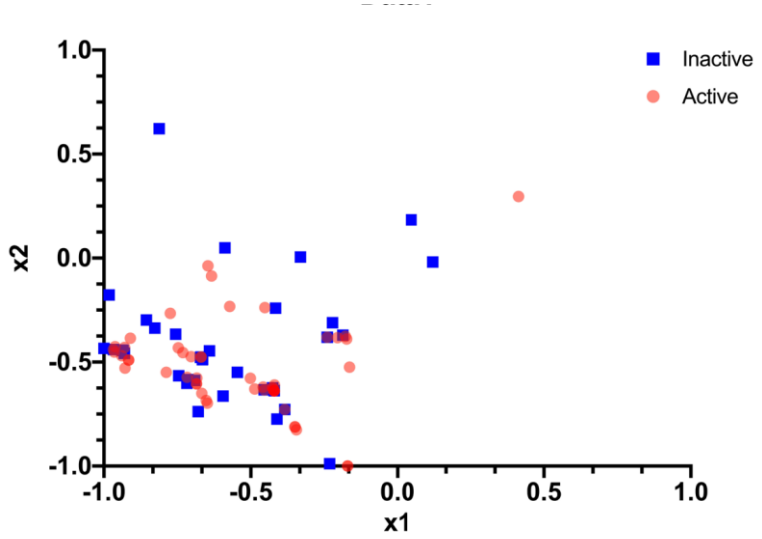
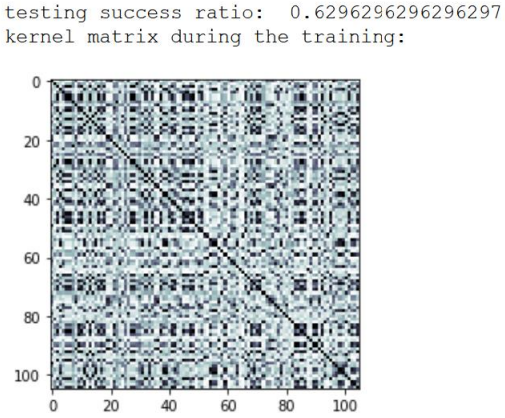


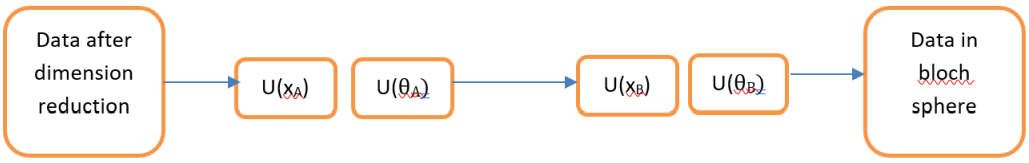
Fig. 3. Kernel Matrix obtained using PCA and Method 3 for the SARS-CoV-2 dataset.



Another approach we have applied is that of the hybrid approach in which the QC is used to perform part of the calculation while performing the remainder on a CC. This removes the storage limitation on QC as this is provided by the CC, hence leaving the processing to the QC. A data re-uploading classifier was used [34]. This DNN-like algorithm was implemented in the Pennylane tool [35] taking advantage of the preexisting data-reuploading classifier

algorithm. Here unlike SVM where we were reducing to 2-3 dimensions, we can reduce to 3-10 dimensions considering the qubit architecture. For our purpose, we reduce to 6 dimensions using the Method 1 and Method 3 as we discussed for SVM. In this approach, we load data into a single qubit by doing a simple unitary operation U (x1, x2, x3) where x1, x2, x3 is the coordinates of the point. When dimensions are greater than three, we can have U (x1, x2, x3, x4, x5, x6) to (U (x1, x2, x3) U (x4, x5, x6)). Using this approach, we can therefore store a datapoint which has 6 dimensions. We can have a similar unitary U (θ_1, θ_2, θ_3) for rotation of datapoint in bloch sphere (Fig. 4). Here we make a two-qubit connected layer to introduce non-linearity in our network.

Fig. 4. A high-level abstraction of how data is stored in the bloch sphere. The two layers - A and B, with U(x_A) denoting unitary applied on input vector, U(θ_A) representing the unitary applied to rotate the vector in the bloch sphere. The subscript denotes the layer name A or B.



The above algorithm will help in introducing non-linearity to the model which makes use of the Adam stochastic gradient optimizer [36]. We tuned different parameters using an iterative process for the following: Optimizer (Adagrad and Adam), Epochs, batch-size and the number of hidden layers (see Methods). For the same SARS-CoV-2 dataset, we obtained accuracy rates of 61%.

The same algorithm was then implemented on much larger datasets representative of high throughput screening which results in data for tens of thousands of compounds. For all these

datasets, we make use of the Method 1 and Method 3 for reducing the dimension first and then feeding it into our data re-uploading classifier. We also worked with the MFF descriptors [21] which generated a 71,375-descriptor vector (see Methods). A bigger dataset was for *M. tuberculosis* inhibitors which had three variants, namely with a cutoff at IC_{50}'s of 100nM, 1μM and 10 μM [24]. Fig. 5 shows a plot for the 100nM cutoff *M. tuberculosis* dataset when we reduce it into 2 dimensions, using Method 1 and Method 3 for plotting and visualization purposes. For these three datasets the data re-uploading classifier was run and then compared with the results obtained with CC (Table 2, Table S1). Another way of describing this graph is to identify how much advantage we can get on a QC if we are able to replicate the settings of a QC simulator.

Fig. 5. A 2D plot representing the spread of data for the *M. tuberculosis* (100nM) dataset. This takes the important features using dimension = 2 instead of 6. "x1" and "x2" are the top two features respectively (for all the compounds) that have been calculated using Method 1 and Method 3.

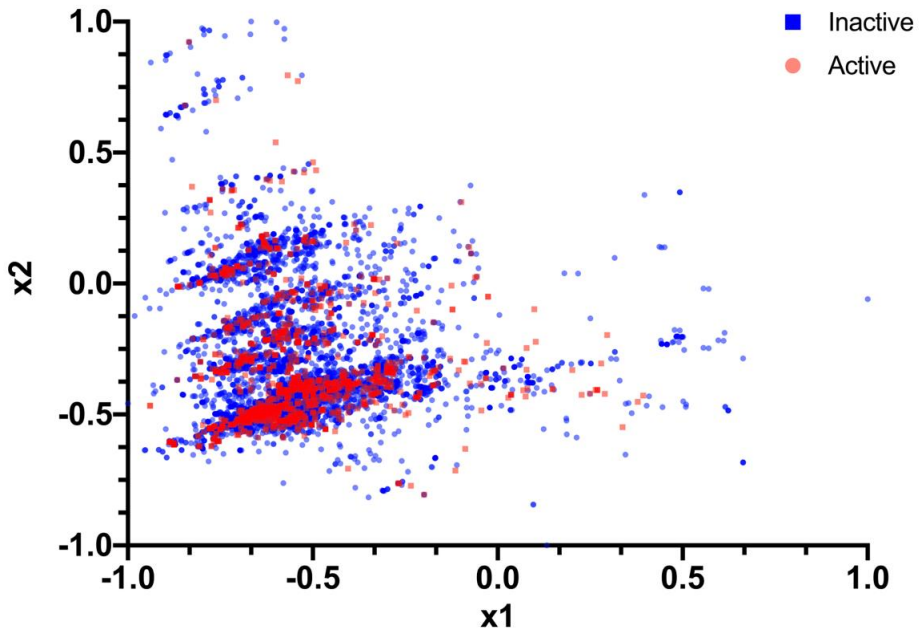


Table 2. Comparing accuracy and run time results for *M. tuberculosis* inhibition datasets (18,886 compounds) [24] using data re-uploading classifier on classical vs quantum computer with 5-fold cross validation.

Dataset potency cutoff (number of actives)	Time on classical computer (min)	Classical computer Accuracy (%)	Time on Quantum computer (min)	Quantum computer Accuracy (%)
100 nM (645)	125	97.1	104	90.5
1 μM (2351)	144	90.4	101	81.4
10 μM (7762)	229	75.6	153	54.9

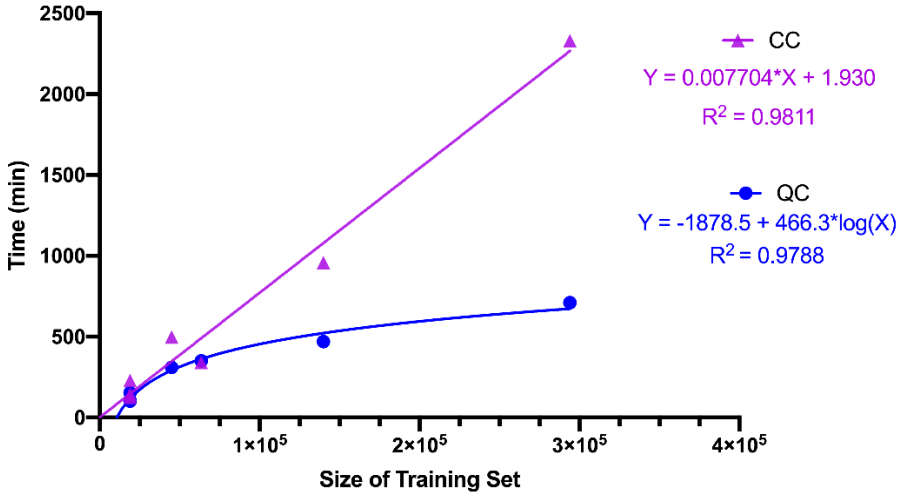
In Table 2 we see that the accuracy obtained on a QC is closer to that obtained on a CC with a slight time advantage over CC such that there is a trade-off of accuracy and speed with this dataset. All the above data reported are implemented on an actual QC (ibmq_rochester, see Methods).

The data re-uploading classifier algorithm [34] was implemented, with the ECFP6 molecular descriptor and tested for 5 considerably larger drug discovery datasets ranging from 44,000 – 293,000 molecules on a QC simulator (Table 3). Running them on QC with the transfers of data was the major overhead here for such big datasets. We find that the results obtained are very comparable with a slight time advantage for the QC simulator over CC. The linearity of calculation time on a CC with dataset size was apparent and this plateaued for the QC simulator (Fig. 6).

Table 3. Comparing large scale drug discovery datasets on a Quantum Computer Simulator and classical computer for data re-uploading classifier using the ECFP6 descriptor obtained from Method 1 and Method 2. Algorithms on QC Simulator were run for 10 epochs with 5 fold cross validation unless noted. ND = not determined.

Dataset Name	**Number of compounds (active)**	**Quantum computer training accuracy (%)**	**Quantum computer testing accuracy (%)**	**Time on Quantum computer (min/epoch)**	**Classical Testing accuracy (%)**	**Time on Classical computer**
Cathepsin B	63,331 (75)	99.4	99.1	35	99.8	341.25
Krabbe	44,809 (63)	91.2	92.4	31	99.9	497.91
Plague	139,861 (223)	92.9	93.1	47	99.8	958.59
M. tuberculosis	293,937 (6104)	90.4	91.3	71	97.9	2329.4
hERG	306,587 (233)	82.7	82.5	313 (ran for 5 epochs)	ND	ND

Fig. 6 : Comparing dataset size with run time for Quantum Computer (QC) simulator and Classical Computer (CC)



We have discussed four approaches and their combination for compression of the molecular descriptors and then calculation of the machine learning model on QC. We found that the results were optimal when combining Method 3 and Method 1 most likely due to the increased spread of data after employing these techniques. Further, we applied both the QC and hybrid approach to train our model. With the current QC hardware available, the option of choosing the hybrid approach for drug discovery is likely optimal. When dealing with bigger datasets on the order of tens of thousands of molecules, we found that the data communication overhead between computer and cloud-based QC was much larger than the actual time taken for a circuit to execute on QC. Plotting the time versus size of a dataset for CC vs QC simulator provides hope for the future of QC in terms of the likely advantage obtainable if we are able to replicate the same settings on a QC (Fig. 6). Now that we have optimized these steps for machine learning we have demonstrated that QC can handle 'massive' drug discovery datasets. Currently the available data for SARS-CoV-2 is in the low hundreds of molecules [23] and such datasets clearly do not require the performance of QC. However, the larger high throughput screening datasets for other targets and diseases that have been

amassing in public databases like PubChem [37] and ChEMBL [1] present significant challenges for SVM and other compute intensive tools. QC is therefore a viable approach to overcoming some of these limitations and allowing practical compute times. With thousands of such datasets now readily available, being able to curate them and update them quickly will be important as new screening data are added. This study demonstrates the clearly non-linear scaling of compute time on a QC with multiple independent dataset size, compared with the linearity observed on a CC. As quantum machine learning develops, the accessibility of QC will increase for drug discovery applications as we have demonstrated here.

Online content

References

1 Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucleic Acids Res* **45**, D945-D954, doi:10.1093/nar/gkw1074 (2017).

2 Bosc, N. *et al.* Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. *J Cheminform* **11**, 4, doi:10.1186/s13321-018-0325-4 (2019).

3 Nogueira, M. S. & Koch, O. The Development of Target-Specific Machine Learning Models as Scoring Functions for Docking-Based Target Prediction. *J Chem Inf Model* **59**, 1238-1252, doi:10.1021/acs.jcim.8b00773 (2019).

4 Imrie, F., Bradley, A. R., van der Schaar, M. & Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J Chem Inf Model* **58**, 2319-2330, doi:10.1021/acs.jcim.8b00350 (2018).

5 Shaikh, N., Sharma, M. & Garg, P. An improved approach for predicting drug-target interaction: proteochemometrics to molecular docking. *Mol Biosyst* **12**, 1006-1014, doi:10.1039/c5mb00650c (2016).

6 Mayr, A. *et al.* Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem Sci* **9**, 5441-5451, doi:10.1039/c8sc00148k (2018).

7 Ekins, S. *et al.* Exploiting machine learning for end-to-end drug discovery and development. *Nat Mater* **18**, 435-441, doi:10.1038/s41563-019-0338-z (2019).

8 Ekins, S. *et al.* Machine learning models identify molecules active against the Ebola virus in vitro. *F1000Res* **4**, 1091, doi:10.12688/f1000research.7217.3 (2015).

9 Ekins, S. *et al.* Machine Learning Models and Pathway Genome Data Base for Trypanosoma cruzi Drug Discovery. *PLoS Negl Trop Dis* **9**, e0003878, doi:10.1371/journal.pntd.0003878 (2015).

10 Lane, T. R. *et al.* Repurposing the antimalarial pyronaridine tetraphosphate to protect against Ebola virus infection. *PLoS Negl Trop Dis* **13**, e0007890, doi:10.1371/journal.pntd.0007890 (2019).

11 Lane, T. R., Comer, J. E., Freiberg, A. N., Madrid, P. B. & Ekins, S. Repurposing Quinacrine Against Ebola Virus Infection In vivo. *Antimicrob Agents Chemother*, doi:10.1128/AAC.01142-19 (2019).

12 Ekins, S. *et al.* Efficacy of Tilorone Dihydrochloride against Ebola Virus Infection. *Antimicrob Agents Chemother* **62**, doi:10.1128/AAC.01711-17 (2018).

13 Zhavoronkov, A. *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat Biotechnol* **37**, 1038-1040, doi:10.1038/s41587-019-0224-x (2019).

14 Du, F. *et al.* hERGCentral: a large database to store, retrieve, and analyze compound-human Ether-a-go-go related gene channel interactions to facilitate cardiotoxicity assessment in drug development. *Assay Drug Dev Technol* **9**, 580-588, doi:10.1089/adt.2011.0425 (2011).

15 Nalepa, J. & Kawulok, M. Selecting training sets for support vector machines: a review. *Artificial Intelligence Review* **52**, 857-900, doi:10.1007/s10462-017-9611-1 (2019).

16 Havlicek, V. *et al.* Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209-212, doi:10.1038/s41586-019-0980-2 (2019).

17 Fastovets, D. V., Bogdanov, Y. I., Bantysh, B. I. & Lukichev, V. F. *Machine learning methods in quantum computing theory*, <https://arxiv.org/abs/1906.10175> (2019).

18 Broughton, M. *et al. TensorFlow Quantum: A Software Framework for Quantum Machine Learning*, <https://arxiv.org/abs/2003.02989> (2020).

19 Langione, M., Bobier, J.-F., Meier, C., Hasenfuss, S. & Schulze, U. *Will quantum computing transform biopharma R&D?*, <https://www.bcg.com/publications/2019/quantum-computing-transform-biopharma-research-development.aspx> (2019).

20 Schuld, M. Machine learning in quantum spaces. *Nature* **567**, 179-181, doi:10.1038/d41586-019-00771-0 (2019).

21 Sandfort, F., Strieth-Kalthoff, F., Kühnemund, M., Beecks, C. & Glorius, F. A Structure-Based Platform for Predicting Chemical Reactivity. *Chem* **6**, 1379-1390, doi:https://doi.org/10.1016/j.chempr.2020.02.017 (2020).

22 Pattanaik, L. & Coley, C. W. Molecular Representation: Going Long on Fingerprints. *Chem* **6**, 1204-1207, doi:https://doi.org/10.1016/j.chempr.2020.05.002 (2020).

23 Gawriljuk, V. O. *et al.* Machine Learning Models Identify Inhibitors of SARS-CoV-2. *bioRxiv*, 2020.2006.2016.154765, doi:10.1101/2020.06.16.154765 (2020).

24 Lane, T. *et al.* Comparing and Validating Machine Learning Models for Mycobacterium tuberculosis Drug Discovery. *Mol Pharm* **15**, 4346-4360, doi:10.1021/acs.molpharmaceut.8b00083 (2018).

25 Ekins, S., Freundlich, J. S. & Reynolds, R. C. Are bigger data sets better for machine learning? Fusing single-point and dual-event dose response data for Mycobacterium tuberculosis. *J Chem Inf Model* **54**, 2157-2165, doi:10.1021/ci500264r (2014).

26 Clark, A. M. *et al.* Open Source Bayesian Models. 1. Application to ADME/Tox and Drug Discovery Datasets. *J Chem Inf Model* **55**, 1231-1245, doi:10.1021/acs.jcim.5b00143 (2015).

27 Landrum, G. *RDkit*, <https://www.rdkit.org> (2020).

28 Saki, A. A., Alam, M. & Ghosh, S. *Study of Decoherence in Quantum Computers: A Circuit-Design Perspective* <https://arxiv.org/abs/1904.04323> (2019).

29 Anon. *Qiskit: An Open-source Framework for Quantum Computing*, <https://github.com/Qiskit/qiskit/blob/master/Qiskit.bib> (2019).

30 Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys Rev Lett* **113**, 130503 (2014).

31 Adetokunbo, A. J. & et al. *Quantum algorithm implementations for beginners*, <https://arxiv.org/abs/1804.03719> (2018).

32 Paul, L. C., Suman, A. A. & Sultan, N. Methodological analysis of principal component analysis (PCA) method. *Int J Comp Eng Management* **16**, 32-38 (2013).

33 Tharwat, A., Gaber, T., Ibrahim, A. & Aboul Ella, H. Linear discriminant analysis: A detailed tutorial. *AI Communications* **30**, 169-190 (2017).

34 Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E. & Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **4**, 226 (2020).

35 Anon. *Pennylane*, <https://pennylane.ai/) > (2020).

36 Kingma, D. P. & Ba, J. *Adam: A method for stochastic optimization*, <https://arxiv.org/abs/1412.6980> (2014).

37 Wang, Y., Cheng, T. & Bryant, S. H. PubChem BioAssay: A Decade's Development toward Open High-Throughput Screening Data Sharing. *SLAS Discov* **22**, 655-666, doi:10.1177/2472555216685069 (2017).

38 Clark, A. M. & Ekins, S. Open Source Bayesian Models. 2. Mining a "Big Dataset" To Create and Validate Models with ChEMBL. *J Chem Inf Model* **55**, 1246-1260, doi:10.1021/acs.jcim.5b00144 (2015).

39 Wootton, J. R. *Benchmarking near-term devices with quantum error correction* <https://arxiv.org/abs/2004.11037> (2020).

Online Content

Article

Quantum Machine Learning for Drug Discovery

Kushal Batra[1], Kimberley M. Zorn[2], Daniel H. Foil[2], Eni Minerali[2], Victor O. Gawriljuk[3] Thomas R. Lane[2] & Sean Ekins[2*]

[1] Computer Science, NC State University, Raleigh, NC 27606, USA.

[2]Collaborations Pharmaceuticals, Inc., 840 Main Campus Drive, Lab 3510, Raleigh, NC 27606, USA.

[3]São Carlos Institute of Physics, University of São Paulo, Av. João Dagnone, 1100 - Santa Angelina, São Carlos - SP, 13563-120, Brazil

*To whom correspondence should be addressed. E-mail: sean@collaborationspharma.com Phone: 215-687-1320

Methods

Data curation

Our proprietary Assay Central™ software [26,38] is a framework for curating high-quality datasets and building Bayesian machine learning models. Each dataset was subjected to the same standardization processes (i.e. removing salts, metal complexes and mixtures) prior to building models. Duplicate parent compounds with finite concentration activities are merged into a single entry. Classification models such as these require a defined threshold of bioactivity. The SARS-CoV-2 model used a threshold of 6.65 μM; Cathepsin B used >20% inhibition; the three *M. tuberculosis* models used the thresholds as described; the Krabbe model used actives defined by the authors; plague used a threshold at ≥50% inhibition; the hERGcentral dataset was >50% inhibition; The large *M. tuberculosis* model used a cut off where IC_{90} < 10 μg/ml or 10μM and a selectivity index greater than 10 was used (where SI = CC_{50}/IC_{90}) [25]. These files were then output for machine learning.

Running our algorithms on QC:

1) All the accuracies reported are from a QC. The algorithm was run with shots=2048. Here we use IBM's ibmq_rochester for executing our algorithms. Figure 1 shows the architecture of ibmq_rochester. Colors represent error probabilities for controlled-NOTs and readout on qubits [39]. This architecture has 53 qubits linked in the network. These 53 qubits are assembled and connected following the property of hexagonal lattices which proves out to be advantageous when it comes to minimizing unwanted interactions (Supplemental Figure 1).

2) The error in reporting the accuracies can be ±3 depending on the time of day the code is ran. The Quantum Computer are recalibrated once in a day at unknown times. Also, the connection (the qubit architecture also plays an instrumental role in driving the accuracy

rates.) Therefore, the accuracy rates achieved are sometimes ~62% and sometimes ~59% for the best accuracy value. The above accuracy rates reported above are the best value achieved on QC or QC simulator.

The CC specifics on which the algorithms were executed:

Computational Server : Supermicro EATX DDR4 LGA 2011, Intel Computer CPU 2.1 8 BX80660E52620V4, Crucial 64GB Kit (16GBx4) DDR4 2133 (PC42133) DR x4 288 Pin Server Memory CT4K16G4RFD4213 / CT4C16G4RFD4213, 1-5 x EVGA GeForce GTX 1080 Ti FOUNDERS EDITION GAMING, 11GB GDDR5X, Intel 730 SERIES 2.5Inch Solid State Drive SSDSC2BP480G410, WD Gold 4TB Datacenter Hard Disk Drive 7200 RPM Class SATA 6 Gb/s 128MB Cache 3.5 Inch WD4002FYYZ, Supermicro 920 Watt 4U Server. 5 GPU's were used for the very large datasets in Table 3, 3 GPU's were for TB dataset in Table 2, and 1 GPU for COVID dataset.

Quantum SVM:

In the code for SVM the depth was set to 2. Entanglement was set to ‘full’ and the skip_qobj_validation parameter for quantum instance was set to False.

Data re-uploading classifier:

The MFF molecular descriptor [21] was also used with the four methods to achieve better accuracy rates. The data re-uploading classifier is very similar to a deep neural network [34]. With the above MFF descriptor we get 71,375 vectors. As the authors in the paper of MFF said this leads to overfitting and the same results were observed with MFF followed by our reduction algorithm.

The data re-uploading classifier had hyperparameters set to - Train size: (70% of dataset), Test size: (30% of dataset), train accuracy rate: 61.2%, test accuracy rate: 61% ,number of layers: 4, batch-size: 32, epochs: 10, optimizer: Adam, learning rate: 0.6, cross fold validation: 5. For all the datasets.

Acknowledgments

Professor Daniel D. Stancil (NC State) is kindly acknowledged for his guidance on quantum computing. Access to the IBM Q Network was obtained through the IBM Q Hub at NC State. We gratefully acknowledge our many colleagues and collaborators who assisted in curating these datasets used as examples for this study as well as Mr. Valery Tkachenko for hardware support and Dr. Alex M. Clark for Assay Central™ support. We kindly acknowledge NIH funding to develop the software from R44GM122196-02A1 “Centralized assay datasets for modelling support of small drug discovery organizations” from NIGMS and NIEHS for 1R43ES031038-01 “MegaTox for analyzing and visualizing data across different screening systems”. “Research reported in this publication was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under Award Number R43ES031038. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.”

Author contributions

The project was led, and experiments designed by S.E. The work on the descriptor compression and algorithm comparisons was implemented by K.B. The datasets employed were curated by K.M.Z, D.H.F., E.M. T.R.L. and V.O.G. T.R.L. provided graphics. D.D.S. assisted with access to the quantum computer. All authors contributed to the manuscript.

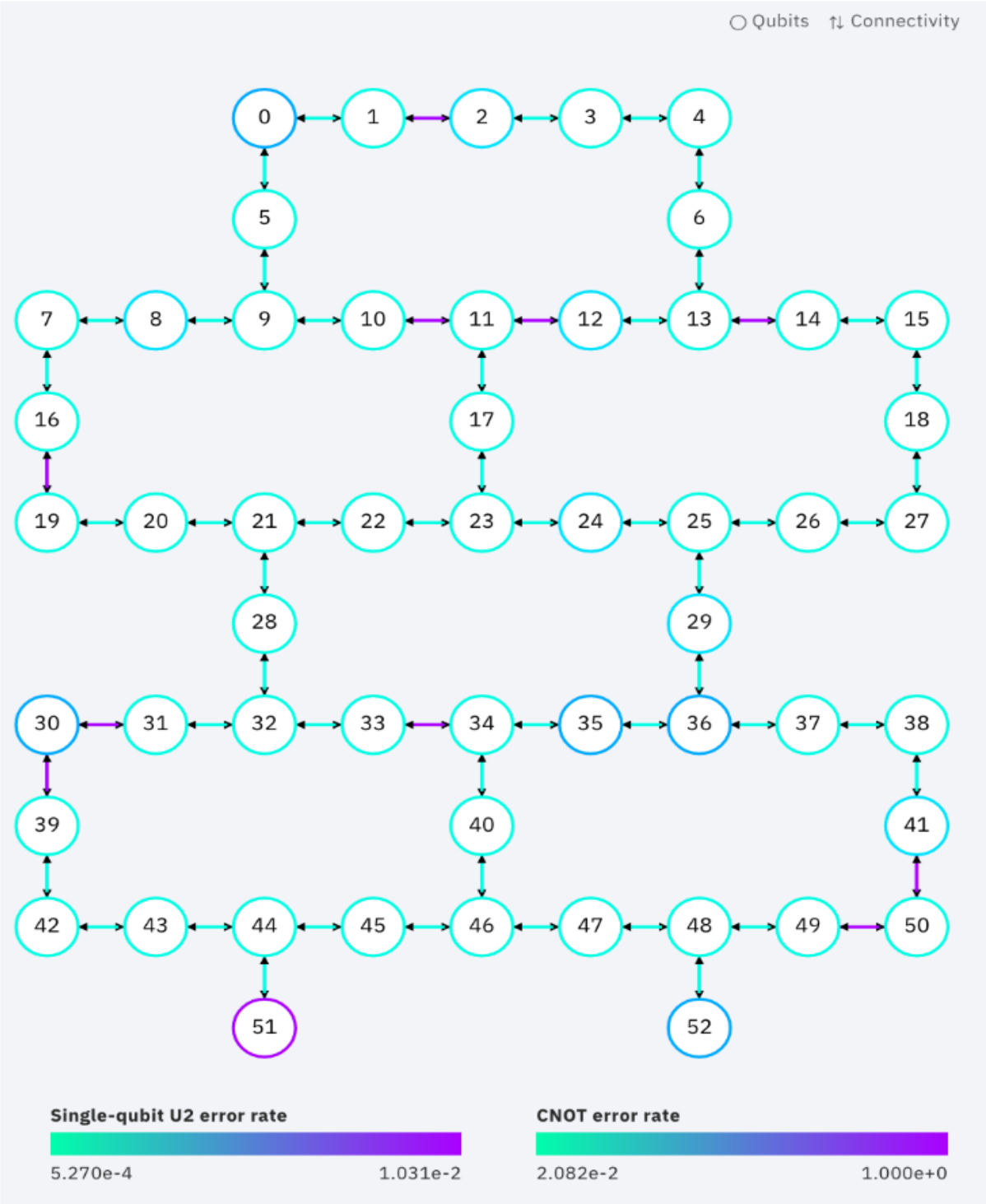
Competing interests

S.E., K.M.Z., D.H.F., E.M., and T.R.L. work for Collaborations Pharmaceuticals, Inc. K.B. and V.O.G. have no conflicts of interest.

Statements of data and code availability

Code and data are available upon request.

Supplemental Figure 1: IBM Rochester architecture map (source: IBM).



Supplemental Table 1: Confusion matrix for *M. tuberculosis* dataset (18,886 compounds) [24] obtained after running data re-uploading classifier algorithm.

Dataset potency cut-off	Confusion Matrix
100 nM	4652 677 203 34
1 μM	4453 395 655 63
10 μM	2077 1159 1477 853

Article

Quantum Machine Learning for Drug Discovery

Kushal Batra[1], Kimberley M. Zorn[2], Daniel H. Foil[2], Eni Minerali[2], Victor O. Gawriljuk[3]

Thomas R. Lane[2] & Sean Ekins[2*]

[1] Computer Science, NC State University, Raleigh, NC 27606, USA.

[2]Collaborations Pharmaceuticals, Inc., 840 Main Campus Drive, Lab 3510, Raleigh, NC 27606, USA.

[3]São Carlos Institute of Physics, University of São Paulo, Av. João Dagnone, 1100 - Santa Angelina, São Carlos - SP, 13563-120, Brazil

*To whom correspondence should be addressed. E-mail: sean@collaborationspharma.com

Phone: 215-687-1320

The growing public and private datasets focused on small molecules screened against biological targets or whole organisms [1] provides a wealth of drug discovery relevant data. Increasingly this is used to create machine learning models which can be used for enabling target-based design [2-4], predict on- or off-target effects and create scoring functions [5,6]. This is matched by the availability of machine learning algorithms such as Support Vector Machines (SVM) and Deep Neural Networks (DNN) that are computationally expensive to perform on very large datasets and thousands of molecular descriptors. Quantum computer (QC) algorithms have been proposed to offer an approach to accelerate quantum machine learning over classical computer (CC) algorithms, however with significant limitations. In the case of cheminformatics, one of the challenges to overcome is the need for compression of large numbers of molecular descriptors for use on QC. Here we show how to achieve compression with datasets using hundreds of molecules (SARS-CoV-2) to hundreds of thousands (whole cell screening datasets for plague and *M. tuberculosis*) with SVM and data re-uploading classifier (a DNN equivalent algorithm) on a QC benchmarked against CC and hybrid approaches. This illustrates a quantum advantage for drug discovery to build upon in future.

The pharmaceutical industry and academic laboratories are increasingly using and exploring machine learning applications in drug discovery to mine and model their data generated from years of high throughput screening [7]. This is allowing rapid identification of molecules for neglected diseases such as Ebola [8] and Chagas disease [9], presenting lead compounds which can then be moved rapidly into *in vivo* models [10-12] and this approach can be more widely applied. Recent examples have illustrated the speed with which the machine learning combined with *in vitro* testing continuum can generate new leads compared to traditional efforts [13]. The availability of thousands of structure activity datasets, (some of which in turn contain data for hundreds of thousands of molecules screened against a single target or organism [14]), presents a computational challenge with machine learning methods such as classifiers like SVM on CC [15]. Recently the potential of quantum machine learning has been illustrated using two methods such as a variational quantum circuit and a quantum kernel estimator [16]. In addition, new machine learning methods for quantum computing continue to be developed [17,18] which creates opportunities for expanding useful applications. Like many emerging technologies, QC has been proposed as likely to transform early stage pharmaceutical research and development as well as provide a potential solution for computationally intractable problems that would be traditionally performed on a CC in order to learn from data [19,20]. One area of interest in early drug discovery is virtual screening and optimization where small molecules are frequently described by fingerprint descriptors which can lead to tens of thousands of vectors called multiple fingerprint features (MFF) [21]. While this may be important for many aspects of applying machine learning to chemistry [22], it also creates significant challenges when using these descriptors on a QC. Herein we describe how we have applied multiple approaches to compress the descriptors for QC while also demonstrating applications to drug discovery datasets on a range of scales that would be broadly applicable. We also describe hybrid

approaches that merge QC with CC for machine learning applied to these datasets curated from public sources for drug discovery. These include 132 small molecule inhibitors of SARS-CoV-2 in Vero cells [23], 18,886 inhibitors of *Mycobacterium tuberculosis* [24] as well as several larger datasets for inhibitors of Krabbe disease (β-Galactocerebrosidase, Pubchem Assay 1159614, 44,809 compounds), Cathepsin B (Pubchem Assay 453, 63,331 compounds), Plague (Yersinia pestis, Pubchem Assay 898, 139,861 compounds), a larger dataset for *M. tuberculosis* (293,937 compounds) [25] and hERG (306,587 compounds) [14]. All these datasets were curated and prepared using Assay Central™ [24] (See Methods).

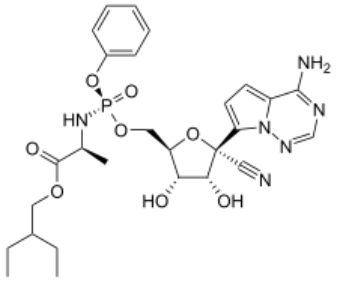
As the extended connectivity fingerprint radius 6 (ECFP6) has been widely applied in cheminformatics [26] we used the Morgan Fingerprint (with radius 3) which is equivalent to the ECFP6 fingerprint in RDkit [27]. The Morgan Fingerprint generates binary numbers whose default size is 2048 bits. While this is acceptable for use on CC, it is not acceptable for QC which requires a smaller number. Also, even with the increase in size/usage of qubits on the QC the network accuracy fails because of decoherence noise introduced in the qubit system with this big system [28]. Even having no linkage or communication between every qubit adds further noise to this system. We attempted two approaches to solve this utilizing QC alone or a hybrid where part of the code is run on the QC.

The initial algorithm we have compared is SVM implemented using the Qiskit library [29] which uses Least-Square SVM (LS-SVM) [30]. Qiskit chooses $M_{ij} = x_i x_j$ as the ansatz where M_{ij} is the kernel matrix and $x_i x_j$ are the datapoints in the dataset [31]. When working with QC we first formulate the ansatz and try to minimize it. This ansatz formulates the SVM hyperplane which divides the datasets. We have chosen the ansatz such that we get a line as an output. Changing the ansatz equation decides the shape the hyperplane will take. Within

the QC, a quantum algorithm is used to solve the direct product [30] and calculate the M(the kernel matrix), then a quantum algorithm can be used to transform to waveforms to solve the system of linear equations [31]. This approach solves the complete SVM on a QC. While SVM generates good results, it takes considerable time to solve linear equations to solve for Kernel Matrix, using feature maps (~1 minute for each datapoint) and hence, any time advantage for QC is not achievable.

In order to reduce the molecular descriptor features such that they can be represented and stored in our limited number of qubits (53 qubits for the ibmq-rochester, Fig. S1). We have initially demonstrated the application of QC with a SARS-CoV-2 Vero cell inhibition dataset consisting of just 132 compounds of which 66 were found to be active with the IC_{50} activity threshold of 6.65 μM [23]. We reduced the molecular descriptor dimensions to 2-3 and we have hence proposed multiple techniques to encode 2048 features. Method 1 used Principal Component Analysis (PCA) that is widely applied in data compression [32]. PCA resulted in accuracy rates on a CC of 37% using the kernel algorithm = RBF (Radial Basis Function) whereas on a QC accuracy was 33% (N= 3). Method 2 used a common dimension reduction technique of Linear Discriminant Analysis (LDA) which also considers the target class along with the predictors [33]. In this we simply take projection of points into some other hyperplane. LDA resulted in accuracy rates of 40% using a kernel algorithm on a CC while on QC this was 39% (N = 3). In Method 3, we designed an algorithm where we divide 2048 bits into groups of ‘k’ bits and convert these k bits into respective and convert it into decimal. This process is repeated until all the bits are converted to decimal such that k should completely divides 2048. An example of a molecule compressed using this algorithm is shown in Fig. 1. This resulted in a greatly improved accuracy rate of 61% on a CC using kernel algorithm=’poly’ and 59.6% on a QC (n=3).

Fig. 1. An example of a molecule and its compression using Method 3. 2D representation of the antiviral Remdesivir, and an illustration of the ECFP6 descriptor for Remdesivir and the compressed output.



An example: Take k= 128 bits (as 2048 completely divided by 128)

For Remdesivir ECPF6 =


```
[01000001000000000000000010000000010010000001011000000000000000001000000000000000
10000000000000000000000000000000100000100000000010000000000000000010000000000000
00000000000000000000000000000001000000000000000000000000000000000000000100000000
00000000000000000000000000000000000000010000000001000000000000000000000000000001
00000000000000000000000000000001000100000000000000010000000000000000010000000000
00000001000100000000000100001010000000000000000000000000010000000000000000000000
01001000000000000000000000000000010000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000010000000000000001000000000000000000
00010000010000000000000000010000000000000000100100000000000000000000000000000010
00000000000000000000000000000000000000010000000000000000000001000000000000000010
00000000000000000000000000000000000000010000000000000000000000000000000000000000
00010000000000000000000000000000000000010000001000000000000000000000000000000000
00000000000000000100000000000000000000000000010000]
```


Using the above algorithm, we get the following compressed output =


```
[468374416343053 1072, 9223512774343164416, 9223389629040820224,
9223372036858970112, 68753031168, 2305843009356300320, 2251836321452032,
281475014459392, 36028797018963968, 536879104, 9147937279969536, 2199023255553,
2199040032768, 70368744177668, 2164260864, 17179869200]
```


Method 4 uses an algorithm where we keep track of positions of 1 in the whole array. So, if we take the same example of Remdesivir, we store the position of 1's and 0's. we achieve this by iterating over the molecular descriptor and storing 1's value using **value = index * (-1^index) + value where index is the position of 1 or 0.** This way we get two values [-21086,-502690] where -21086 is storing the positions of 1's and -502690 is storing the positions of 0's. Using this approach, we get two dimensions. Even after feeding this data

from PCA we will get back the same results, as from PCA we were only concerned with the top two features. All the algorithms discussed above try to retain most of the information from the molecular descriptors as much as possible. This resulted in accuracy rates on CC of 59% using kernel algorithm='sigmoid' and on QC of 59.25% (with n=2).

These 4 methods were implemented on the QC and then tried as combinations and the accuracy was compared with CC (Table 1). The most promising results were obtained when combining Method 1 and Method 3 (k=128). The improved results for k=128 are due to the increased spread (more variance) compared to other k values when combined with PCA (Fig, 2). This makes it easier to obtain a hyperplane separating the two classes (active or not active). The Kernel matrix obtained from running the SVM model on a QC is shown in Fig. 3.

Table 1. The accuracy rates achieved with different combinations of methods. Method 1 (PCA), Method 2 (LDA), Method 3 and Method-4 using SVM on classical and quantum computer.

Combination	Accuracy on Quantum Computer (%)	Accuracy on Classical Computer (%)
PCA + Method 3(k=64)	<30	47
PCA + Method 3(k=128)	62.9	65
PCA + Method 3(k=256)	59.25	60
PCA + Method 3(k=512)	41	44
LDA + Method 3(k=64)	31	36
LDA + Method 3(k=128)	<30	35
LDA + Method 3(k=256)	<30	31
LDA + Method 3(k=512)	32	59
PCA + Method 4(2)	59.25	60
LDA + Method 4(2)	57	57

Fig. 2. The spread of the data for the SARS-CoV-2 dataset after applying Method 1 and Method 3. "x1" and "x2" are the top two features respectively (for all the compounds) obtained after applying these two methods.

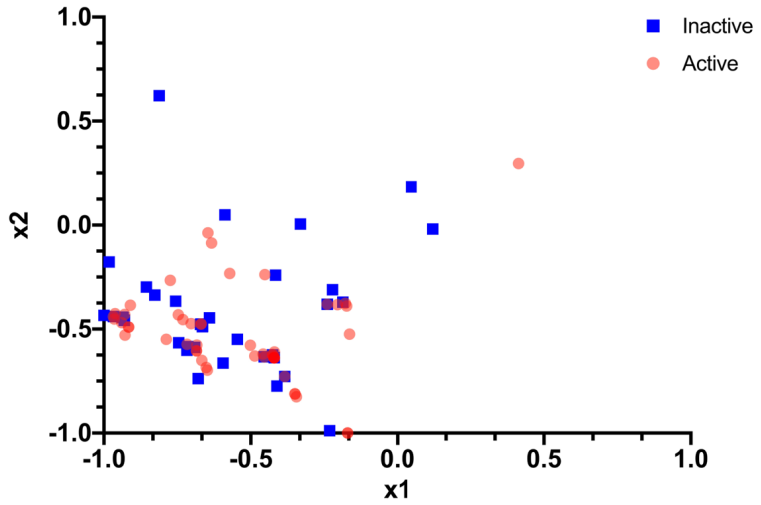
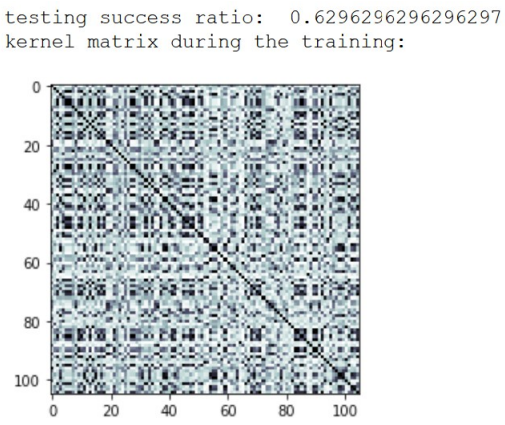


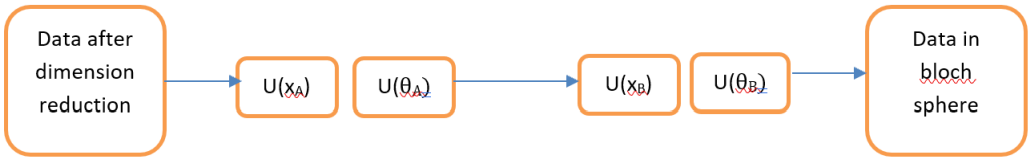
Fig. 3. Kernel Matrix obtained using PCA and Method 3 for the SARS-CoV-2 dataset.



Another approach we have applied is that of the hybrid approach in which the QC is used to perform part of the calculation while performing the remainder on a CC. This removes the storage limitation on QC as this is provided by the CC, hence leaving the processing to the QC. A data re-uploading classifier was used [34]. This DNN-like algorithm was implemented in the Pennylane tool [35] taking advantage of the preexisting data-reuploading classifier algorithm. Here unlike SVM where we were reducing to 2-3 dimensions, we can reduce to 3-10 dimensions considering the qubit architecture. For our purpose, we reduce to 6 dimensions using the Method 1 and Method 3 as we discussed for SVM. In this approach, we load data into a single qubit by doing a simple unitary operation U (x1, x2, x3) where x1,

x2, x3 is the coordinates of the point. When dimensions are greater than three, we can have U (x1, x2, x3, x4, x5, x6) to (U (x1, x2, x3) U (x4, x5, x6)). Using this approach, we can therefore store a datapoint which has 6 dimensions. We can have a similar unitary U (θ_1, θ_2, θ_3) for rotation of datapoint in bloch sphere (Fig. 4). Here we make a two-qubit connected layer to introduce non-linearity in our network.

Fig. 4. A high-level abstraction of how data is stored in the bloch sphere. The two layers - A and B, with U(x_A) denoting unitary applied on input vector, U(θ_A) representing the unitary applied to rotate the vector in the bloch sphere. The subscript denotes the layer name A or B.



The above algorithm will help in introducing non-linearity to the model which makes use of the Adam stochastic gradient optimizer [36]. We tuned different parameters using an iterative process for the following: Optimizer (Adagrad and Adam), Epochs, batch-size and the number of hidden layers (see Methods). For the same SARS-CoV-2 dataset, we obtained accuracy rates of 61%.

The same algorithm was then implemented on much larger datasets representative of high throughput screening which results in data for tens of thousands of compounds. For all these datasets, we make use of the Method 1 and Method 3 for reducing the dimension first and then feeding it into our data re-uploading classifier. We also worked with the MFF descriptors [21] which generated a 71,375-descriptor vector (see Methods). A bigger dataset was for *M. tuberculosis* inhibitors which had three variants, namely with a cutoff at IC_{50}'s of

100nM, 1μM and 10 μM [24]. Fig. 5 shows a plot for the 100nM cutoff *M. tuberculosis* dataset when we reduce it into 2 dimensions, using Method 1 and Method 3 for plotting and visualization purposes. For these three datasets the data re-uploading classifier was run and then compared with the results obtained with CC (Table 2, Table S1). Another way of describing this graph is to identify how much advantage we can get on a QC if we are able to replicate the settings of a QC simulator.

Fig. 5. A 2D plot representing the spread of data for the *M. tuberculosis* (100nM) dataset. This takes the important features using dimension = 2 instead of 6. "x1" and "x2" are the top two features respectively (for all the compounds) that have been calculated using Method 1 and Method 3.

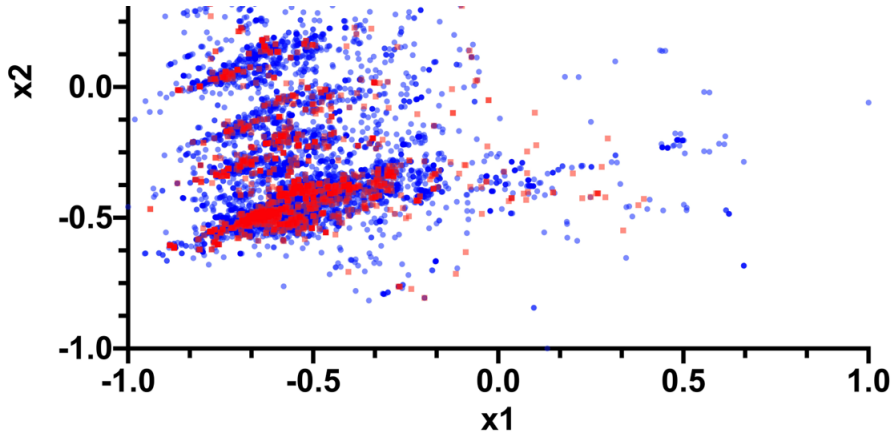


Table 2. Comparing accuracy and run time results for *M. tuberculosis* inhibition datasets (18,886 compounds) [24] using data re-uploading classifier on classical vs quantum computer with 5-fold cross validation.

Dataset potency cutoff (number of actives)	Time on classical computer (min)	Classical computer Accuracy (%)	Time on Quantum computer (min)	Quantum computer Accuracy (%)

100 nM (645)	125	97.1	104	90.5
1 μM (2351)	144	90.4	101	81.4
10 μM (7762)	229	75.6	153	54.9

In Table 2 we see that the accuracy obtained on a QC is closer to that obtained on a CC with a slight time advantage over CC such that there is a trade-off of accuracy and speed with this dataset. All the above data reported are implemented on an actual QC (ibmq_rochester, see Methods).

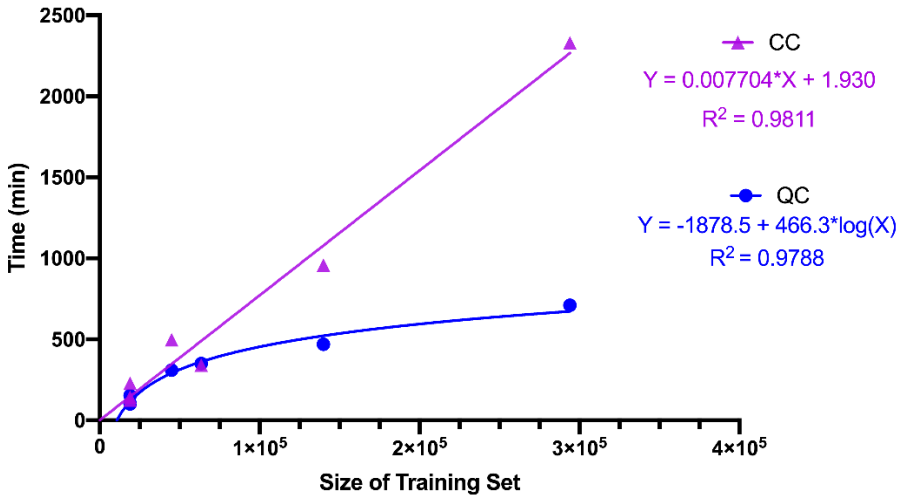
The data re-uploading classifier algorithm [34] was implemented, with the ECFP6 molecular descriptor and tested for 5 considerably larger drug discovery datasets ranging from 44,000 – 293,000 molecules on a QC simulator (Table 3). Running them on QC with the transfers of data was the major overhead here for such big datasets. We find that the results obtained are very comparable with a slight time advantage for the QC simulator over CC. The linearity of calculation time on a CC with dataset size was apparent and this plateaued for the QC simulator (Fig. 6).

Table 3. Comparing large scale drug discovery datasets on a Quantum Computer Simulator and classical computer for data re-uploading classifier using the ECFP6 descriptor obtained from Method 1 and Method 2. Algorithms on QC Simulator were run for 10 epochs with 5 fold cross validation unless noted. ND = not determined.

Dataset Name	Number of compounds (active)	Quantum computer training accuracy (%)	Quantum computer testing accuracy (%)	Time on Quantum computer (min/epoch)	Classical Testing accuracy (%)	Time on Classical computer
Cathepsin B	63,331 (75)	99.4	99.1	35	99.8	341.25

Krabbe	44,809 (63)	91.2	92.4	31	99.9	497.91
Plague	139,861 (223)	92.9	93.1	47	99.8	958.59
M. tuberculosis	293,937 (6104)	90.4	91.3	71	97.9	2329.4
hERG	306,587 (233)	82.7	82.5	313 (ran for 5 epochs)	ND	ND

Fig. 6 : Comparing dataset size with run time for Quantum Computer (QC) simulator and Classical Computer (CC)



We have discussed four approaches and their combination for compression of the molecular descriptors and then calculation of the machine learning model on QC. We found

that the results were optimal when combining Method 3 and Method 1 most likely due to the increased spread of data after employing these techniques. Further, we applied both the QC and hybrid approach to train our model. With the current QC hardware available, the option of choosing the hybrid approach for drug discovery is likely optimal. When dealing with bigger datasets on the order of tens of thousands of molecules, we found that the data communication overhead between computer and cloud-based QC was much larger than the actual time taken for a circuit to execute on QC. Plotting the time versus size of a dataset for CC vs QC simulator provides hope for the future of QC in terms of the likely advantage obtainable if we are able to replicate the same settings on a QC (Fig. 6). Now that we have optimized these steps for machine learning we have demonstrated that QC can handle 'massive' drug discovery datasets. Currently the available data for SARS-CoV-2 is in the low hundreds of molecules [23] and such datasets clearly do not require the performance of QC. However, the larger high throughput screening datasets for other targets and diseases that have been amassing in public databases like PubChem [37] and ChEMBL [1] present significant challenges for SVM and other compute intensive tools. QC is therefore a viable approach to overcoming some of these limitations and allowing practical compute times. With thousands of such datasets now readily available, being able to curate them and update them quickly will be important as new screening data are added. This study demonstrates the clearly non-linear scaling of compute time on a QC with multiple independent dataset size, compared with the linearity observed on a CC. As quantum machine learning develops, the accessibility of QC will increase for drug discovery applications as we have demonstrated here.

Online content

References

1 Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucleic Acids Res* **45**, D945-D954, doi:10.1093/nar/gkw1074 (2017).

2 Bosc, N. *et al.* Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. *J Cheminform* **11**, 4, doi:10.1186/s13321-018-0325-4 (2019).

3 Nogueira, M. S. & Koch, O. The Development of Target-Specific Machine Learning Models as Scoring Functions for Docking-Based Target Prediction. *J Chem Inf Model* **59**, 1238-1252, doi:10.1021/acs.jcim.8b00773 (2019).

4 Imrie, F., Bradley, A. R., van der Schaar, M. & Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J Chem Inf Model* **58**, 2319-2330, doi:10.1021/acs.jcim.8b00350 (2018).

5 Shaikh, N., Sharma, M. & Garg, P. An improved approach for predicting drug-target interaction: proteochemometrics to molecular docking. *Mol Biosyst* **12**, 1006-1014, doi:10.1039/c5mb00650c (2016).

6 Mayr, A. *et al.* Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem Sci* **9**, 5441-5451, doi:10.1039/c8sc00148k (2018).

7 Ekins, S. *et al.* Exploiting machine learning for end-to-end drug discovery and development. *Nat Mater* **18**, 435-441, doi:10.1038/s41563-019-0338-z (2019).

8 Ekins, S. *et al.* Machine learning models identify molecules active against the Ebola virus in vitro. *F1000Res* **4**, 1091, doi:10.12688/f1000research.7217.3 (2015).

9 Ekins, S. *et al.* Machine Learning Models and Pathway Genome Data Base for Trypanosoma cruzi Drug Discovery. *PLoS Negl Trop Dis* **9**, e0003878, doi:10.1371/journal.pntd.0003878 (2015).

10 Lane, T. R. *et al.* Repurposing the antimalarial pyronaridine tetraphosphate to protect against Ebola virus infection. *PLoS Negl Trop Dis* **13**, e0007890, doi:10.1371/journal.pntd.0007890 (2019).

11 Lane, T. R., Comer, J. E., Freiberg, A. N., Madrid, P. B. & Ekins, S. Repurposing Quinacrine Against Ebola Virus Infection In vivo. *Antimicrob Agents Chemother*, doi:10.1128/AAC.01142-19 (2019).

12 Ekins, S. *et al.* Efficacy of Tilorone Dihydrochloride against Ebola Virus Infection. *Antimicrob Agents Chemother* **62**, doi:10.1128/AAC.01711-17 (2018).

13 Zhavoronkov, A. *et al.* Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat Biotechnol* **37**, 1038-1040, doi:10.1038/s41587-019-0224-x (2019).

14 Du, F. *et al.* hERGCentral: a large database to store, retrieve, and analyze compound-human Ether-a-go-go related gene channel interactions to facilitate

cardiotoxicity assessment in drug development. *Assay Drug Dev Technol* **9**, 580-588, doi:10.1089/adt.2011.0425 (2011).

15 Nalepa, J. & Kawulok, M. Selecting training sets for support vector machines: a review. *Artificial Intelligence Review* **52**, 857-900, doi:10.1007/s10462-017-9611-1 (2019).

16 Havlicek, V. *et al.* Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209-212, doi:10.1038/s41586-019-0980-2 (2019).

17 Fastovets, D. V., Bogdanov, Y. I., Bantysh, B. I. & Lukichev, V. F. *Machine learning methods in quantum computing theory*, <https://arxiv.org/abs/1906.10175> (2019).

18 Broughton, M. *et al. TensorFlow Quantum: A Software Framework for Quantum Machine Learning*, <https://arxiv.org/abs/2003.02989> (2020).

19 Langione, M., Bobier, J.-F., Meier, C., Hasenfuss, S. & Schulze, U. *Will quantum computing transform biopharma R&D?*, <https://www.bcg.com/publications/2019/quantum-computing-transform-biopharma-research-development.aspx> (2019).

20 Schuld, M. Machine learning in quantum spaces. *Nature* **567**, 179-181, doi:10.1038/d41586-019-00771-0 (2019).

21 Sandfort, F., Strieth-Kalthoff, F., Kühnemund, M., Beecks, C. & Glorius, F. A Structure-Based Platform for Predicting Chemical Reactivity. *Chem* **6**, 1379-1390, doi:https://doi.org/10.1016/j.chempr.2020.02.017 (2020).

22 Pattanaik, L. & Coley, C. W. Molecular Representation: Going Long on Fingerprints. *Chem* **6**, 1204-1207, doi:https://doi.org/10.1016/j.chempr.2020.05.002 (2020).

23 Gawriljuk, V. O. *et al.* Machine Learning Models Identify Inhibitors of SARS-CoV-2. *bioRxiv*, 2020.2006.2016.154765, doi:10.1101/2020.06.16.154765 (2020).

24 Lane, T. *et al.* Comparing and Validating Machine Learning Models for Mycobacterium tuberculosis Drug Discovery. *Mol Pharm* **15**, 4346-4360, doi:10.1021/acs.molpharmaceut.8b00083 (2018).

25 Ekins, S., Freundlich, J. S. & Reynolds, R. C. Are bigger data sets better for machine learning? Fusing single-point and dual-event dose response data for Mycobacterium tuberculosis. *J Chem Inf Model* **54**, 2157-2165, doi:10.1021/ci500264r (2014).

26 Clark, A. M. *et al.* Open Source Bayesian Models. 1. Application to ADME/Tox and Drug Discovery Datasets. *J Chem Inf Model* **55**, 1231-1245, doi:10.1021/acs.jcim.5b00143 (2015).

27 Landrum, G. *RDkit*, <https://www.rdkit.org> (2020).

28 Saki, A. A., Alam, M. & Ghosh, S. *Study of Decoherence in Quantum Computers: A Circuit-Design Perspective* <https://arxiv.org/abs/1904.04323> (2019).

29 Anon. *Qiskit: An Open-source Framework for Quantum Computing*, <https://github.com/Qiskit/qiskit/blob/master/Qiskit.bib> (2019).

30 Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys Rev Lett* **113**, 130503 (2014).

31 Adetokunbo, A. J. & et al. *Quantum algorithm implementations for beginners*, <https://arxiv.org/abs/1804.03719> (2018).

32 Paul, L. C., Suman, A. A. & Sultan, N. Methodological analysis of principal component analysis (PCA) method. *Int J Comp Eng Management* **16**, 32-38 (2013).

33 Tharwat, A., Gaber, T., Ibrahim, A. & Aboul Ella, H. Linear discriminant analysis: A detailed tutorial. *AI Communications* **30**, 169-190 (2017).

34 Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E. & Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **4**, 226 (2020).

35 Anon. *Pennylane*, <https://pennylane.ai/) > (2020).

36 Kingma, D. P. & Ba, J. *Adam: A method for stochastic optimization*, <https://arxiv.org/abs/1412.6980> (2014).

37 Wang, Y., Cheng, T. & Bryant, S. H. PubChem BioAssay: A Decade's Development toward Open High-Throughput Screening Data Sharing. *SLAS Discov* **22**, 655-666, doi:10.1177/2472555216685069 (2017).

38 Clark, A. M. & Ekins, S. Open Source Bayesian Models. 2. Mining a "Big Dataset" To Create and Validate Models with ChEMBL. *J Chem Inf Model* **55**, 1246-1260, doi:10.1021/acs.jcim.5b00144 (2015).

39 Wootton, J. R. *Benchmarking near-term devices with quantum error correction* <https://arxiv.org/abs/2004.11037> (2020).

Online Content

Article

Quantum Machine Learning for Drug Discovery

Kushal Batra[1], Kimberley M. Zorn[2], Daniel H. Foil[2], Eni Minerali[2], Victor O. Gawriljuk[3] Thomas R. Lane[2] & Sean Ekins[2*]

[1] Computer Science, NC State University, Raleigh, NC 27606, USA.

[2]Collaborations Pharmaceuticals, Inc., 840 Main Campus Drive, Lab 3510, Raleigh, NC 27606, USA.

[3]São Carlos Institute of Physics, University of São Paulo, Av. João Dagnone, 1100 - Santa Angelina, São Carlos - SP, 13563-120, Brazil

*To whom correspondence should be addressed. E-mail: sean@collaborationspharma.com

Phone: 215-687-1320

Methods

Data curation

Our proprietary Assay Central™ software [26,38] is a framework for curating high-quality datasets and building Bayesian machine learning models. Each dataset was subjected to the same standardization processes (i.e. removing salts, metal complexes and mixtures) prior to building models. Duplicate parent compounds with finite concentration activities are merged into a single entry. Classification models such as these require a defined threshold of bioactivity. The SARS-CoV-2 model used a threshold of 6.65 μM; Cathepsin B used >20% inhibition; the three *M. tuberculosis* models used the thresholds as described; the Krabbe model used actives defined by the authors; plague used a threshold at ≥50% inhibition; the hERGcentral dataset was >50% inhibition; The large *M. tuberculosis* model used a cut off where IC_{90} < 10 μg/ml or 10μM and a selectivity index greater than 10 was used (where SI = CC_{50}/IC_{90}) [25]. These files were then output for machine learning.

Running our algorithms on QC:

1) All the accuracies reported are from a QC. The algorithm was run with shots=2048. Here we use IBM's ibmq_rochester for executing our algorithms. Figure 1 shows the architecture of ibmq_rochester. Colors represent error probabilities for controlled-NOTs and readout on qubits [39]. This architecture has 53 qubits linked in the network. These 53 qubits are assembled and connected following the property of hexagonal lattices which proves out to be advantageous when it comes to minimizing unwanted interactions (Supplemental Figure 1).

2) The error in reporting the accuracies can be ±3 depending on the time of day the code is ran. The Quantum Computer are recalibrated once in a day at unknown times. Also, the connection (the qubit architecture also plays an instrumental role in driving the accuracy

rates.) Therefore, the accuracy rates achieved are sometimes ~62% and sometimes ~59% for the best accuracy value. The above accuracy rates reported above are the best value achieved on QC or QC simulator.

The CC specifics on which the algorithms were executed:

Computational Server : Supermicro EATX DDR4 LGA 2011, Intel Computer CPU 2.1 8 BX80660E52620V4, Crucial 64GB Kit (16GBx4) DDR4 2133 (PC42133) DR x4 288 Pin Server Memory CT4K16G4RFD4213 / CT4C16G4RFD4213, 1-5 x EVGA GeForce GTX 1080 Ti FOUNDERS EDITION GAMING, 11GB GDDR5X, Intel 730 SERIES 2.5Inch Solid State Drive SSDSC2BP480G410, WD Gold 4TB Datacenter Hard Disk Drive 7200 RPM Class SATA 6 Gb/s 128MB Cache 3.5 Inch WD4002FYYZ, Supermicro 920 Watt 4U Server. 5 GPU's were used for the very large datasets in Table 3, 3 GPU's were for TB dataset in Table 2, and 1 GPU for COVID dataset.

Quantum SVM:

In the code for SVM the depth was set to 2. Entanglement was set to ‘full’ and the skip_qobj_validation parameter for quantum instance was set to False.

Data re-uploading classifier:

The MFF molecular descriptor [21] was also used with the four methods to achieve better accuracy rates. The data re-uploading classifier is very similar to a deep neural network [34]. With the above MFF descriptor we get 71,375 vectors. As the authors in the paper of MFF said this leads to overfitting and the same results were observed with MFF followed by our reduction algorithm.

The data re-uploading classifier had hyperparameters set to - Train size: (70% of dataset), Test size: (30% of dataset), train accuracy rate: 61.2%, test accuracy rate: 61% ,number of layers: 4, batch-size: 32, epochs: 10, optimizer: Adam, learning rate: 0.6, cross fold validation: 5. For all the datasets.

Acknowledgments

Professor Daniel D. Stancil (NC State) is kindly acknowledged for his guidance on quantum computing. Access to the IBM Q Network was obtained through the IBM Q Hub at NC State. We gratefully acknowledge our many colleagues and collaborators who assisted in curating these datasets used as examples for this study as well as Mr. Valery Tkachenko for hardware support and Dr. Alex M. Clark for Assay Central™ support. We kindly acknowledge NIH funding to develop the software from R44GM122196-02A1 “Centralized assay datasets for modelling support of small drug discovery organizations” from NIGMS and NIEHS for 1R43ES031038-01 “MegaTox for analyzing and visualizing data across different screening systems”. “Research reported in this publication was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under Award Number R43ES031038. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.”

Author contributions

The project was led, and experiments designed by S.E. The work on the descriptor compression and algorithm comparisons was implemented by K.B. The datasets employed were curated by K.M.Z, D.H.F., E.M. T.R.L. and V.O.G. T.R.L. provided graphics. D.D.S. assisted with access to the quantum computer. All authors contributed to the manuscript.

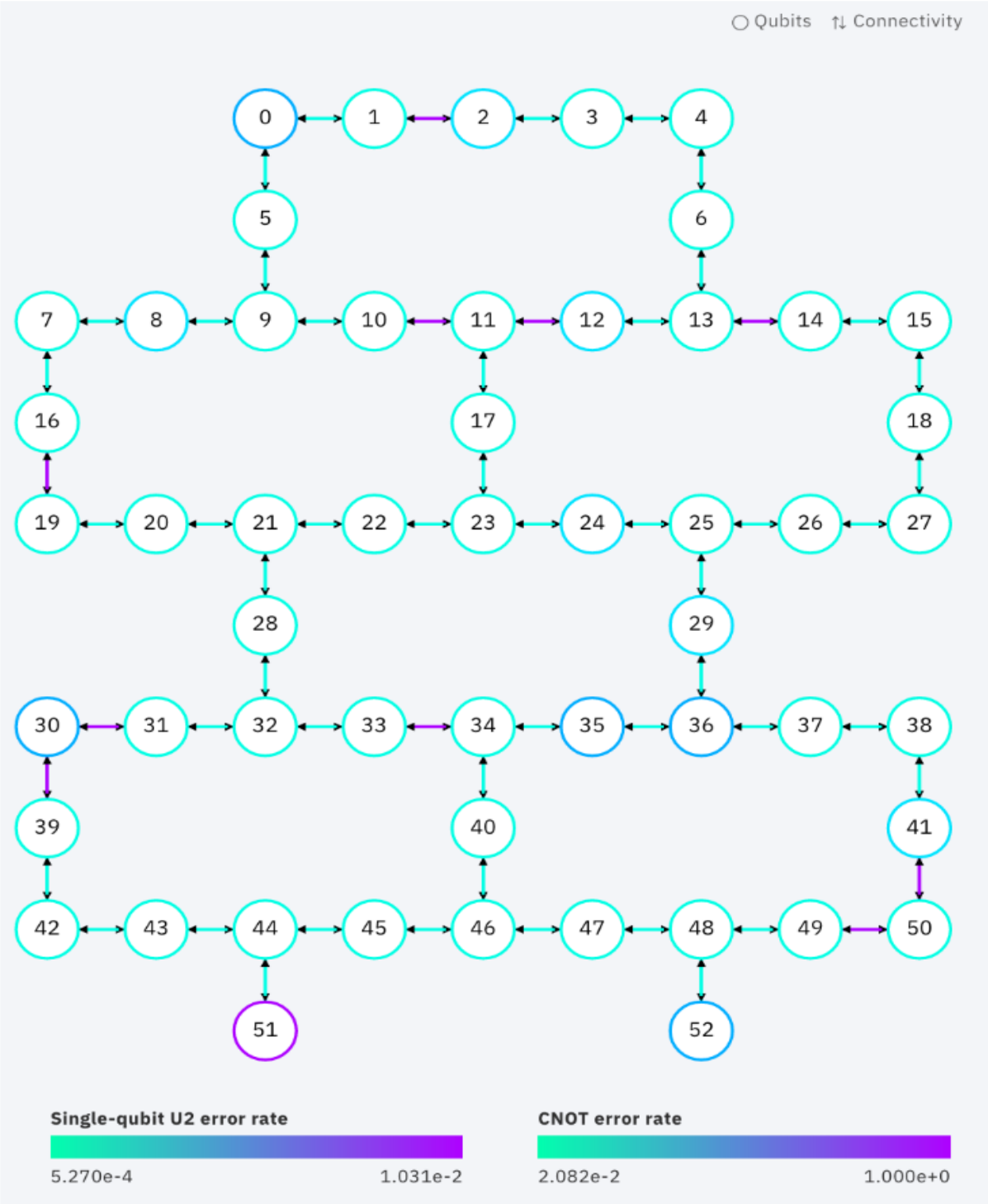
Competing interests

S.E., K.M.Z., D.H.F., E.M., and T.R.L. work for Collaborations Pharmaceuticals, Inc. K.B. and V.O.G. have no conflicts of interest.

Statements of data and code availability

Code and data are available upon request.

Supplemental Figure 1: IBM Rochester architecture map (source: IBM).



Supplemental Table 1: Confusion matrix for *M. tuberculosis* dataset (18,886 compounds) [24] obtained after running data re-uploading classifier algorithm.

Dataset potency cut-off	Confusion Matrix
100 nM	4652 677 203 34
1 μM	4453 395 655 63
10 μM	2077 1159 1477 853