



PAPER • OPEN ACCESS

Investigating the potential for a limited quantum speedup on protein lattice problems

To cite this article: Carlos Outeiral *et al* 2021 *New J. Phys.* **23** 103030

View the [article online](#) for updates and enhancements.

You may also like

- [Operation and performance of the ATLAS semiconductor tracker in LHC Run 2](#)
The ATLAS collaboration, Georges Aad, Brad Abbott et al.
- [Black holes, gravitational waves and fundamental physics: a roadmap](#)
Abbas Askar, Chris Belczynski, Gianfranco Bertone et al.
- [Domain wall encoding of discrete variables for quantum annealing and QAOA](#)
Nicholas Chancellor



PAPER

Investigating the potential for a limited quantum speedup on protein lattice problems

Carlos Outeiral^{1,2} , Garrett M Morris¹ , Jiye Shi³ , Martin Strahm⁴,
Simon C Benjamin^{2,*}  and Charlotte M Deane^{1,*} ¹ Department of Statistics, University of Oxford, 24–29 St Giles', Oxford OX1 3LB, United Kingdom² Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom³ Computer-Aided Drug Design, UCB Pharma, 216 Bath Road, Slough SL1 3WE, United Kingdom⁴ Pharma Research & Early Development, F Hoffmann-La Roche, Grenzacherstrasse 4058, Basel, Switzerland

* Authors to whom any correspondence should be addressed.

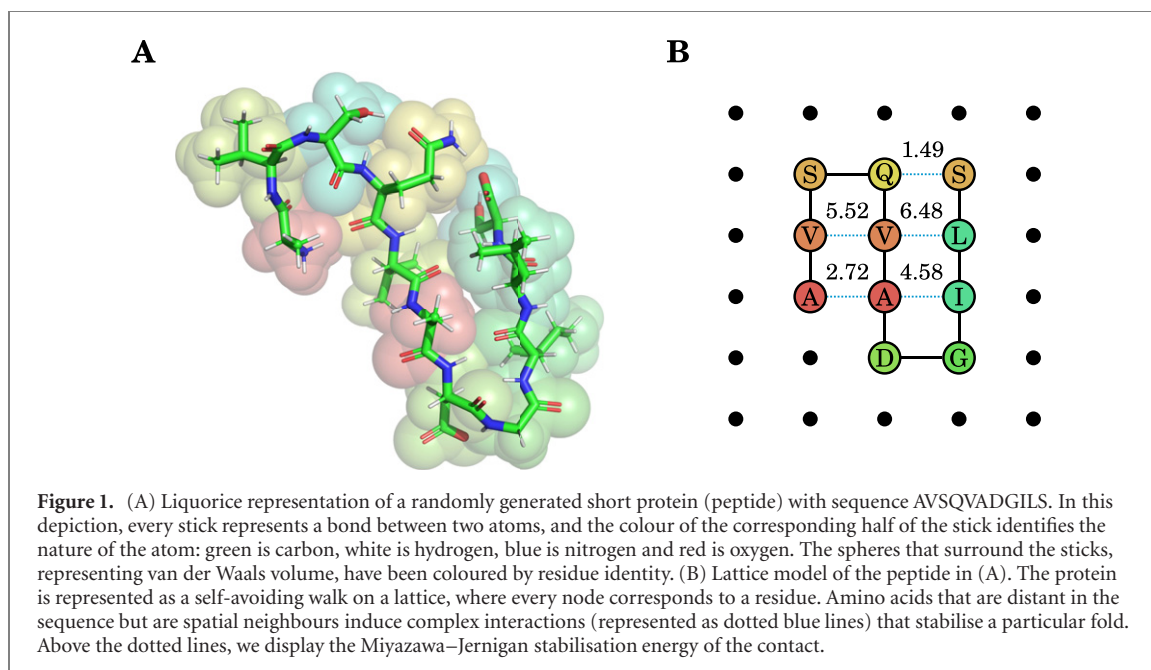
E-mail: simon.benjamin@materials.ox.ac.uk and deane@stats.ox.ac.uk**Keywords:** quantum annealing, protein folding, quantum optimisation, biophysics, NISQRECEIVED
18 May 2021REVISED
14 September 2021ACCEPTED FOR PUBLICATION
24 September 2021PUBLISHED
25 October 2021Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).Any further distribution
of this work must
maintain attribution to
the author(s) and the
title of the work, journal
citation and DOI.**Abstract**

Protein folding is a central challenge in computational biology, with important applications in molecular biology, drug discovery and catalyst design. As a hard combinatorial optimisation problem, it has been studied as a potential target problem for quantum annealing. Although several experimental implementations have been discussed in the literature, the computational scaling of these approaches has not been elucidated. In this article, we present a numerical study of quantum annealing applied to a large number of small peptide folding problems, aiming to infer useful insights for near-term applications. We present two conclusions: that even naïve quantum annealing, when applied to protein lattice folding, has the potential to outperform classical approaches, and that careful engineering of the Hamiltonians and schedules involved can deliver notable relative improvements for this problem. Overall, our results suggest that quantum algorithms may well offer improvements for problems in the protein folding and structure prediction realm.

1. Introduction

The structure of a protein captures crucial information about its biological function and therapeutic potential [1]. Knowledge of a proteins' structure unlocks valuable biological information, ranging from the ability to predict protein–protein interactions [2], to structure-based discovery of new drugs [3] and catalysts [4]. Unfortunately, experiments to determine protein structure are challenging and require extensive time and resources [1, 5, 6]. As of May 2021, the TrEMBL database [7] contained 214 million protein sequences, while only 177 000 protein structures have been deposited in the Protein Data Bank [8]. A reliable computational algorithm for the template-free—that is, when a structurally similar protein, perhaps an evolutionary-related specimen from a different organism, is not available to use as a reference [9]—prediction of a protein's structure and its folding pathway from sequence information alone would enable annotation of millions of proteins and could stimulate major advances in biological research. However, despite steady improvements in the past six decades [10–12], and recent advances in the past year [13, 14], a consistent and accurate algorithm for protein folding from sequence has remained elusive.

Over the past decade, there have been attempts to leverage quantum computing for protein structure prediction [15]. The biological structure of a protein is thought to correspond to the minimum of a free energy hypersurface, which for even small peptides is too vast to explore exhaustively [11]. A type of quantum computation that may be appropriate to help is quantum annealing, an approach to exploiting the physics of a controlled quantum system that is considered to be of potential use in optimisation problems (whether classical or quantum in nature) [16]. Typically, the set of possible solutions is mapped to a register of qubits with a binary encoding, and the objective function is represented as a physical Hamiltonian,



H_{problem} , whose eigenvectors and eigenvalues are respectively problem solutions and their scores. In particular, the ground state $|\Phi\rangle$ (or the respective ground eigenspace) corresponds to the global minimum of the problem. The algorithm starts by initialising the register of qubits in the ground state $|\Psi(0)\rangle$ of a given Hamiltonian, H_{trivial} , whose ground state is easy to prepare, and gently transforming into the problem Hamiltonian, H_{problem} . If the evolution is slow enough, the adiabatic theorem of quantum mechanics [17] ensures that the final state $|\Psi(T)\rangle$ will be infinitesimally close to $|\Phi\rangle$.

Protein chemistry applications of quantum computing have concentrated on a simplified prototype known as the protein lattice model [18], which has been used as a coarse-grained proxy for structure prediction [19, 20] (see figure 1) as well as an invaluable testbed in early theoretical protein folding studies [39]. In this model, the protein is described by a self-avoiding walk on a lattice, whose energy is determined by the contacts between adjacent amino acids, and the minimum energy is identified with the biologically active form of the protein. Unfortunately, the problem of finding the protein configuration that minimises the energy is known to be NP-hard [21, 22]. In the context of quantum computing, several encodings (i.e. instructions to map the problem to a Hamiltonian operator and the solutions to a binary string) have been proposed [23–25], some of which have also been tested experimentally in D-Wave processors [25, 26]. Recent work has attempted extensions of the protein lattice model [27], and even off-lattice models [28, 29]. Although multiple algorithmic approaches have been suggested, there is not, to our knowledge, any numerical or analytical study establishing the computational scaling of quantum annealing for protein folding applications.

Protein sequences are constrained to a small range of hundreds, or at most thousands of amino acids. This idiosyncrasy renders considerations of asymptotic complexity, a typical focus of the quantum algorithms literature, inadequate for a practical evaluation of usefulness. The median length of a human protein is 375 amino acids [30], while the median clinical target is about 414 amino acids long [31]—and the effective length may be made smaller by considering independently-folding domains, an ubiquitous strategy in computational structure prediction pipelines. Hence, most interesting problems could be encoded with just a few hundred qubits. If there exists a relative, not necessarily asymptotic, speedup with respect to classical heuristics, this will be of interest to both basic research in molecular biology and the pharmaceutical and biotechnological industries, which dedicate significant resources to structural protein studies. Establishing the scaling of quantum annealers in problems of modest size is a crucial baseline for further exploration as near-term quantum annealers capable of addressing larger problems are made available [38].

In this article, we present an extensive numerical study of protein lattice folding in idealised, error-free, closed-system quantum annealing, such as might be achievable in future devices with long coherence times in the presence of error correction [32–36], or with fault-tolerant universal quantum computers employing Hamiltonian simulation [37]. In particular, we have computed the minimum spectral gaps and optimised time-to-solution (TTS) metrics for a large dataset of hard problems. The spectral gaps display a strongly vanishing behaviour, which according to the adiabatic theorem leads one to anticipate a quadratically

stronger upper bound in runtime. However, our simulations of unitary evolution reveal a scaling that is several orders of magnitude milder, and that can be optimised further via simple heuristics based on average gap positions. When comparing the quantum method to a purely classical heuristic that minimises a classical energy function we find that, in the conditions examined, the quantum annealing method is more efficient than classical simulated annealing, although further research will need to be conducted to establish if this potential advantage holds under different conditions or when compared against other potential methods.

2. Methodology

Throughout this article, we use a large dataset of peptide (i.e. short protein) sequences, each with a unique global minimum (UGEM), which have been mapped to Ising-like problems with the methods described by Babbush *et al* [24]. Peptide sequences with UGEMs have been shown to display the properties of real proteins, even for small sequences (e.g. [39]). The dataset is examined using numerical simulations to assess the gap and expected TTS, studied in terms of the relationship between conditions and results, and compared against an off-the-shelf classical algorithm.

2.1. Benchmark dataset generation

We examined a total of 29 503 peptide sequences, an approximately equal number of cases in both dimensions (15 173 in 2D and 14 330 in 3D) and lengths (approximately 4500 cases per length at a given dimension, with the exception of 2D length 7, where it is challenging to generate UGEM cases, and we considered only 1700 cases). To produce this dataset, we generated protein sequences by random sampling with replacement of the 20 standard amino acids (ARNDCQEGHILKMFPSTWYV). The states of these instances were enumerated by a brute force algorithm, scoring the energies using the Miyazawa–Jernigan 20-amino acid model [68], and all the sequences with two or more non-equivalent minimum energy conformations were rejected.

These sequences were mapped to an algebraic expression representing the couplings between individual spins in a programmable Ising model. This expression is often known as a polynomial unconstrained binary optimisation (PUBO) [69]. We employ the *turn encoding* approach by Babbush *et al* [24], which displays the highest reported efficiency in the number of qubits. To perform the mapping in practice, we developed a modified version of *SymEngine* [70], a computer algebra system written in C++, which exploits the idempotency of binary variables leading to up to a five orders of magnitude speedup. We consider protein Hamiltonians with 10 (6 aa peptide in 2D) to 16 qubits (8 aa peptide in 3D), meaning Hilbert spaces of size 1024 to 65 536.

2.2. Numerical simulations

2.2.1. Representation of the Hamiltonian

Every quantum annealing process considered in this article may be represented in the following general form:

$$H(s) = (1 - s) H_{\text{start}} + s H_{\text{protein}} + \lambda s(1 - s) H_{\text{catalyst}}, \quad (1)$$

where H_{protein} is the Hamiltonian expressing a given protein lattice problem,

$$H_{\text{start}} = \frac{1}{2} \sum_{i=1}^N (I - \sigma_i^x) \quad (2)$$

and H_{catalyst} is one of the following:

$$H_{\text{nonstq}} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \sigma_i^x \sigma_j^x \quad (3)$$

$$H_{\text{stq}} = \sum_{i=1}^N \sigma_i^x. \quad (4)$$

In these equations, N is the number of qubits; λ is the relative strength of the catalyst; s is the annealing progress, defined in the interval $[0, 1]$, and generated by some interpolation function $s = f(t/T)$; I is the identity operator in the N -spin space i.e. $I = I_1 \otimes I_2 \otimes \dots \otimes I_N$; and σ_i^x is the Pauli X matrix applied to the i th spin in N -spin space, i.e. $\sigma_i^x = \left(\bigotimes_{k=1}^{i-1} I \right) \otimes \sigma_x \otimes \left(\bigotimes_{k=i+1}^N I \right)$. Note that the non-stoquastic catalyst in equation (3) includes a $1/N$ factor to ensure that it is in the same energy scale as the rest of the terms. It is

worth pointing out that one could compare this catalyst to its stoquastic counterpart by making the interaction ferromagnetic.

In the computational basis, H_{protein} can be represented as a diagonal matrix, and both H_{start} and H_{stq} will be matrices with off-diagonal elements in defined positions. It is easy to prove that the number of null elements in a linear combination of these Hamiltonian matrices grows as $\mathcal{O}(2^N)$, hence significant savings in memory and processing time can be made by exploiting the sparsity of the matrix representation. In contrast, however, H_{nonstq} has a different sparsity pattern and can only be simulated at a much higher computational cost. Our numerical simulations relied on the PETSc library [72, 73], where the Hamiltonians were represented in the compressed sparse row or Yale format.

2.2.2. Gap evaluation

We estimated the minimum spectral gap between the ground state and the first excited state via numerical diagonalisation of the Hamiltonian matrices, using the Krylov–Schur method [71] as implemented in the SLEPC package [71]. We computed the eigenvalues in increments of $\Delta s = 0.1$, and interpolated the results using cubic splines. The gap was found as the minimum energy difference between every curve that ran into the ground state at the end of the evolution, and any other curve.

We considered only Hamiltonians without a catalyst term (i.e. $\lambda = 0$) for the evaluation of the spectral gap.

2.2.3. Quantum annealing simulation

We estimated the expected TTS by simulating the quantum dynamics of the annealer. We assumed an idealised quantum annealer at zero temperature, in the absence of noise, and with perfect control over couplings, which was simulated by numerical integration of the time-dependent Schrödinger equation:

$$\frac{d|\Psi\rangle}{dt} = -iH(t)|\Psi\rangle \quad (5)$$

where $H(t)$ is the time-dependent Hamiltonian defined in equation (1). We integrated this equation using the Runge–Kutta 5th order method with adaptive timestepping, as implemented in the PETSc package [72, 73]. Runge–Kutta methods have been previously validated for studying quantum annealing [74]. At the end of the evolution, the final state $|\Psi(t = T)\rangle$ is a vector of amplitudes Ψ_i whose square modulus $|\Psi_i|^2$ is the probability of measuring a particular binary string in the device.

Several authors have attested the need to use optimal TTS metrics to assess the scaling of quantum annealing algorithms, e.g. [47, 64]. We employed the Bayesian optimisation package GPyOpt [57] to optimise the annealing time T . We defined an optimisation domain between 0.1 and 1000 a.u., which was considered acceptable after initial exploration.

We set up a maximum of 50 iterations for annealing programs involving a single parameter (the sample time), and a maximum of 500 iterations for trajectories including a catalyst, where the strength of the catalyst (including the energy scale multiplicative factor $1/N$, i.e. $\lambda' = \lambda/N$) has to be optimised alongside the sample time; in the case of the non-stoquastic catalyst, given the loss of sparsity of the Hamiltonian matrix, the optimisation was stopped after 30 iterations for some of the length 9 peptides. Default parameters were used otherwise.

2.3. Optimal trajectory design

We considered the design of tailored trajectories that increase the efficiency of the algorithm. These trajectories were designed by considering the tendency of gaps to appear more often at certain stages of the annealing trajectory (see figure S8). We developed a heuristic that accounts for the relative probability of encountering a minimum gap, considering the magnitudes of all the gaps found at that position.

Our heuristic is based in dividing the annealing trajectory in small regions ($\Delta s = 0.05$) and estimating the following magnitude for each bin, which we denote ‘ R -score’:

$$R_k = \sum_j \frac{P_{kj}}{f(\Delta_{kj})}, \quad (6)$$

where k is the index of the bin; the sum extends to all peptides whose minimal gap falls in that bin, indexed by j ; P_{kj} is the normalised probability that there is a gap at that position (estimated using kernel density estimation with the Epanechnikov kernel and 0.01 bandwidth); Δ_{kj} is the magnitude of said gap; and f is a function that weighs the magnitude of the gap into the R -score. The values of the R -score are normalised and used to define a piecewise linear interpolation function that slows down in regions likely to contain a vanishing gap, and anneals at a faster rate otherwise.

We consider two types of optimisation: one employing peptides of all lengths, and a different one designing an optimal program for each length. We also consider several functional forms $f(\Delta_{kj})$ for the R -score: linear (x), square root (\sqrt{x}) and cubic root ($\sqrt[3]{x}$). Performance comparisons for these functions are reported in figures S9–S12.

2.4. Classical simulated annealing

We compared the performance of the quantum annealing to classical simulated annealing, one of the most common optimisation algorithms in computational biology, which is used in many modern structure prediction packages like Rosetta [62]. We employ an off-the-shelf classical simulated annealing subroutine, the `gsl_siman.h` module of the GNU Scientific Library [63]. This module requires a definition of the optimisation space, and a function that returns the energy of a given configuration. We implemented a simple subroutine to compute the energy of a sequence of lattice moves and, in order to avoid biases deriving from our implementation, we analyse the results in term of Monte Carlo moves instead of times. We note that simulated annealing is not applied to the problem in its PUBO form, but to the energy function encoded therein, as we aim for a head-to-head comparison between quantum and classical approaches.

The classical simulated annealing subroutine employs several parameters, including the number of trials per step, the number of iterations per temperature, the maximum step size of the random walk and the parameters of the Boltzmann distribution (initial and final temperature, Boltzmann constant and damping factor). All of these parameters, except for the Boltzmann constant (set to 1) and the step size (set to 1), were optimised using Bayesian optimisation in the same way parameters were tuned in numerical simulations of quantum annealing.

2.5. Statistical analysis of scaling

We assessed the significance of these results using a statistical analysis. We performed a non-linear least squares fit of our data using the `lmfit` library [75]. We considered four functional models: polynomial (x^α), exponential ($e^{\alpha x}$), square exponential ($e^{\alpha x^2}$) and cubic exponential ($e^{\alpha x^3}$). The function was augmented by a constant β , which was set to the average value of the dependent variable for a given length.

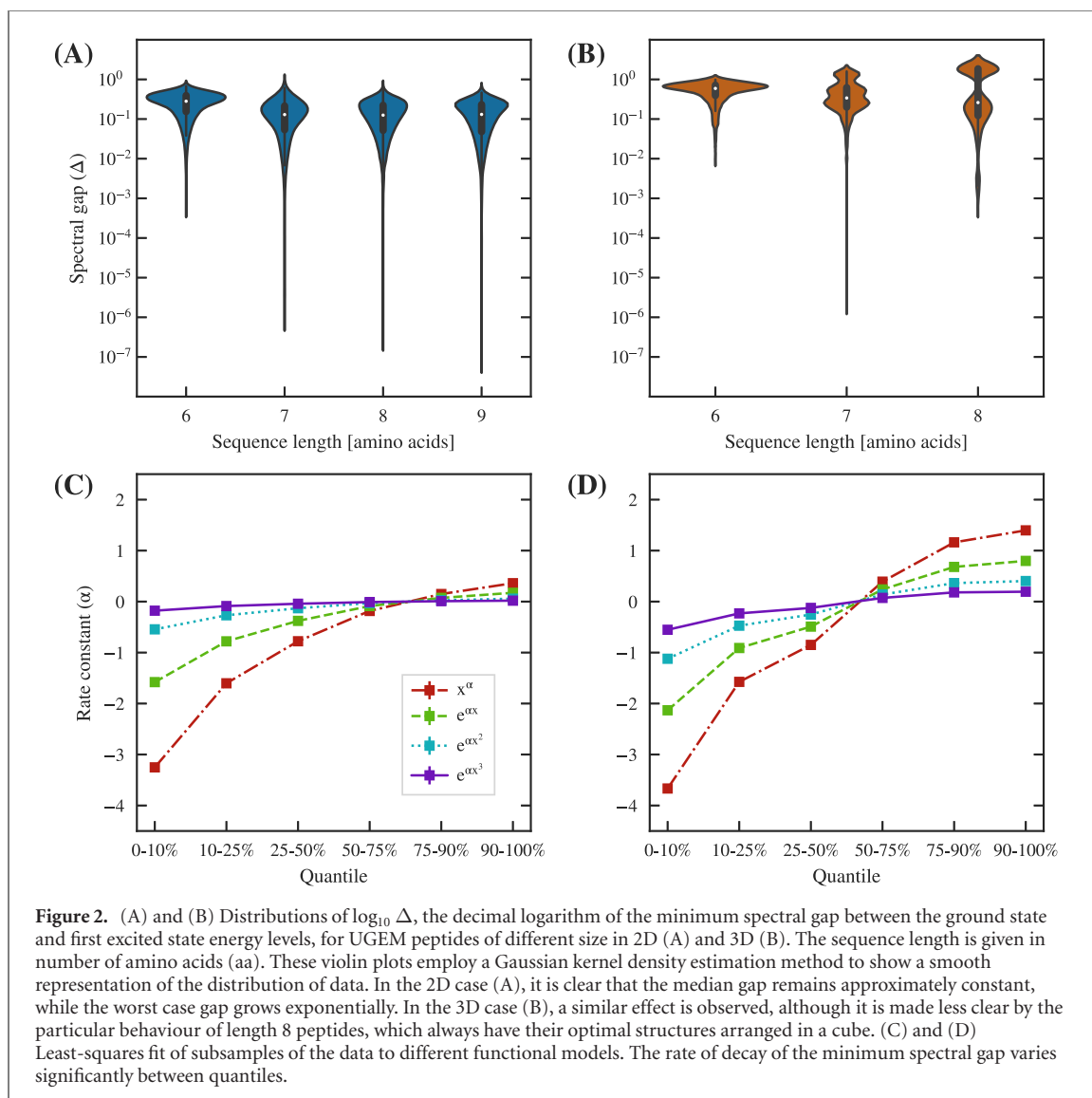
We employed three statistical model selection criteria to decide the function that provided the best explanation of the data: the Akaike information criterion (AIC), the Bayesian information criterion (BIC), and the mean squared error (MSE) of the means. The last method was chosen because of the nature of the dataset: the independent variable is discrete, hence the model only provides the prediction for 4 values (3 in 3D). Thus, understanding how the real means differ from the predicted means provides useful information for model selection. These methods are introduced in more detail in appendix B.

3. Spectral gap

We start our analysis by studying the minimum spectral gap, Δ , between the ground state and the first excited state. It is often stated (based on theoretical arguments) that the runtime of quantum annealing algorithms is proportional to $\mathcal{O}(\Delta^{-2})$ [40]. Unfortunately, many problems exhibit an exponentially vanishing gap with increasing problem size [41–43], and in particular it is believed that no form of quantum computing is able to efficiently solve NP-complete problems, or at least no such report has withstood scrutiny [40, 44]. We note that there exist numerical problems other than the gap that may impair the performance of quantum annealing, for example the precision needed to specify the PUBO, which in the Hamiltonians considered here grows rapidly (see figures S15 and S16). The distribution of gaps for the protein lattice problem in 2D and 3D, considering a stoquastic process without a catalyst, is shown in figure 2.

The distribution of gaps at a given length resembles a skewed Gaussian distribution: the majority of gaps are concentrated around a narrow centre spanning two orders of magnitude, and there are long, thick tails (containing 5%–10% of the data) that spread away several more orders of magnitude to one side. The extent of these tails presents a severe decrease. In the dataset of 2D peptides, the size of the worst-case gaps decreases by five orders of magnitude within a length increase of four amino acids, but only an order of magnitude and a half within the last three lengths considered. Similar results are seen for the 3D peptides. Length 8 three-dimensional peptides show smaller gaps because of their distinct distribution of energetic levels due to cubic symmetry.

This steep decline for the hardest instances is in contrast with the small decrease experienced by an average peptide. The vast majority of the examples populate the area around the median gap, exhibiting a steady but far slower decline. A close examination of the violin plots for the 2D examples also reveals that the position of the peak of the distribution tends to rise as the sequence length increases, and in fact, if we



ignore the tails of the distribution, the average gap increases rather than decreases. In interpreting this finding, it is important to keep in mind that our dataset is composed of peptide folding problems that are hard by design, since they have only one ground state solution (plus, in some cases, symmetry-equivalent configurations). These problems are known to be classically very hard [21, 22]. In addition to the lack of structure of NP-hard problems, the proportion of lowest-energy solutions in the solution space is minimal, so randomised algorithms will find it very challenging to find the ground state.

We characterised the scaling of the gap using regression analysis by maximum likelihood estimation (see section 2 for details)⁵. Four functional forms were considered: polynomial, x^α , exponential, $e^{\alpha x}$, square exponential, $e^{\alpha x^2}$ and cubic exponential, $e^{\alpha x^3}$. We then employed several standard model selection criteria, detailed in appendix B, to select the model that better explains the data. The polynomial model x^α , with $\alpha \approx -0.75$ in 2D and $\alpha \approx -0.4$ in 3D, is selected by all criteria, and is significantly better than the second best model.

We also binned the data into different quantiles and repeated the inference, to account for the inhomogeneity of the results (see tables S2 and S3 in appendix B). In 2D, the polynomial model is consistently selected across all quantiles, and in 3D, there are some quantiles where the exponential and cubic exponential are selected, which is probably due to the limited range of the dataset and the symmetry effect discussed above. We observed a notable variation of the coefficients across quantiles, as depicted in figures 2(C) and (D). For example, the first quantile, 0%–10%, with $\alpha = -3.23$ in 2D and $\alpha = -3.65$ in

⁵ In this section and elsewhere, we fit the data to the length of the protein problem, which we think is the most informative metric from a biological standpoint. We note that there is a linear relation between this parameter and the number of variables in the PUBO (see table S1), hence the goodness-of-fit estimations are unchanged, and the parameters resulting of the fit can be transformed trivially.

3D for the polynomial model, contains examples whose gaps vanish at a notably larger rate. On the other hand, the two upper quartiles (75%–90% and 90%–100%) display positive α values, showing that the gap actually widens with increasing size. Only a portion of the problem instances exhibits fast gap vanishing.

This data does not allow us to conclude that the gap vanishes polynomially. Our results are reminiscent of a previous study on the exact cover problem by Young *et al* [45, 46]. In that study, it was described that, while the scaling of the spectral gap at small problem sizes is consistent with a polynomial [44], at large problem sizes the scaling turns exponential. Similarly, the fraction of problem instances exhibiting small gaps increases at large problem size [46]. We have been unable to study the behaviour of the protein lattice problem at greater sizes, given the large number of qubits and the difficulty of obtaining Hamiltonian expressions beyond 9 amino acids. However, we hypothesise that the protein lattice problem presents exponentially vanishing spectral gaps that will hinder a general polynomial-time solution by quantum annealing for large sizes, while still exhibiting reasonable advantage for problems of modest size, such as may arise in high-throughput peptide discovery experiments.

4. Simulations of stoquastic dynamics

The minimum spectral gap imposes an upper bound on the running time of quantum annealing, but in order to understand the behaviour of the process we need to access the evolution of the quantum state during the computational process. We investigate this regime by numerical integration of the Schrödinger equation. Since this procedure is costly, the assessment of our entire dataset of *circa* 30 000 peptides is beyond our resources and, instead, we selected two samples based on spectral gap values. The first sample contains the set of peptides with the smallest gaps (worst-case set), while the second sample is a random selection of peptides (random set). In both cases, each sample contains 100 peptides per chain length of a given dimension, giving a total of 1400 instances. We report the main features of these Hamiltonians in figures S13–S16.

A comparison of this sort requires optimising the annealing time to maximise the probability of success. As described by Rønnow *et al* [47], a short run can provide a small, but sizeable probability that can be amplified by repetition. In many cases, the repetitions amount to a much shorter runtime than a longer, quasi-adiabatic runtime. We employed Bayesian optimisation [56, 57] to find the optimal runtime, as detailed in section 2. The optimised TTS metric, corresponding to the expected runtime to find the correct solution with probability 50%, is shown in figures 3(A) and (B).

This approach connects with recent developments in what has come to be known as *diabatic quantum computing* [58, 59]. In this procedure, the system is allowed excitations to higher-energy states, which can result in significant speedups and is considered one of the most promising approaches for useful quantum annealing [60]. For example, numerical studies on the problem of maximum two-satisfiability (MAX-2-SAT) have demonstrated that for the hardest random instances, the optimal annealing times were significantly lower (often, several orders of magnitude) when the annealing was allowed to proceed diabatically [61].

The optimisation of the annealing time of an individual run has a significant impact. We simulated a baseline experiment in which the quantum annealer was run for 1000 a.u., and found an average improvement in the expected total runtime of 15 orders of magnitude in 2D and 10 orders of magnitude in 3D (see figure S1). We also find a small, but appreciable difference on the dependence on the gap, as depicted in table 1.

For both the 2D and 3D peptides, the worst-case set containing the smallest gaps does not require significantly longer expected runtimes than the random set. We performed a two-tailed Welch's *t* test, and found that the random and worst-case sets could not be distinguished (*p*-value 0.34, average *p*-value of subgroup analysis 0.31). In other words, the fact that a problem presents a very small minimum spectral gap does not necessarily indicate it will require a long runtime. This apparently contradictory statement might be explained by the fact that the Δ^{-2} scaling in relation to the spectral gap is only valid in the asymptotic limit, and that other terms may govern the dynamics at smaller sizes; and that this rule is above and after all an upper bound, and it is very possible that an instance succeeds even if the annealing is conducted diabatically. In practical terms it is not necessary to guarantee that the final state has a large fidelity with the ground state, but rather to ensure that it has a sizeable amplitude in order to obtain the expected result after enough trials. Another possibility to note is that within the range of problem sizes we inspect there may be only one (or a few) instances of the minimum (or near-minimum) gap occurring during the adiabatic sweep, whereas for large problems a near-minimum gap may occur at multiple points. In addition, the decrease by five orders of magnitude in 2D gaps discussed earlier is also markedly absent from figures 3(A) and (B).

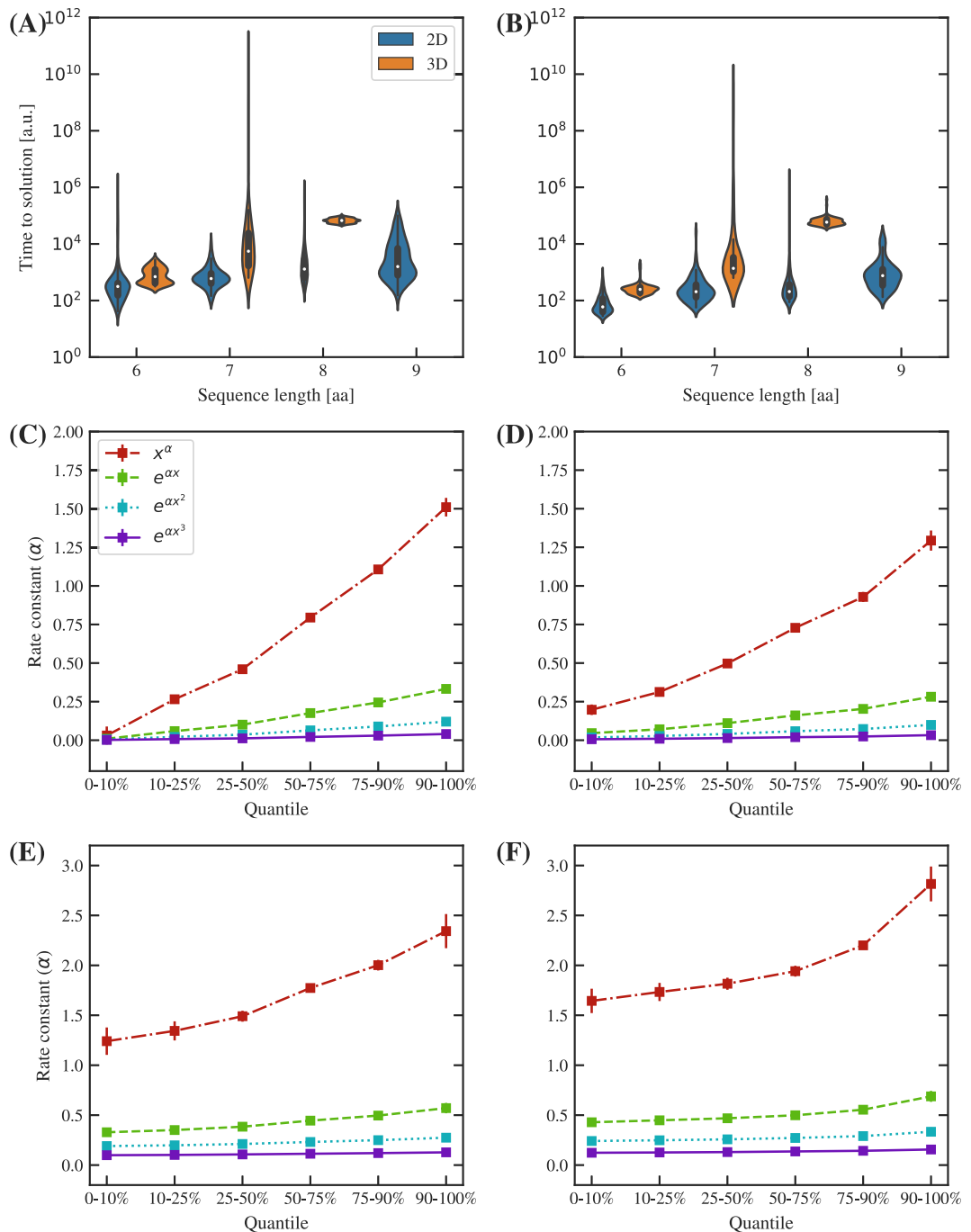


Figure 3. (A) and (B) Distributions of the expected quantum annealing runtime T of the worst-case (A) and random (B) sets. (C) and (D) Least-squares fit of subsamples of the data to different functional models for the worst-case (C) and random (D) sets. (E) and (F) Least-squares fit of subsamples of the data to different functional models for the worst-case (E) and random (F) sets.

We performed a scaling analysis identical to the previous section, finding that, for all cases, either the exponential model is selected over the polynomial, or there is not a significant difference between both of them (see tables S4–S7 in appendix B). In particular, in 2D the polynomial model x^α (with $\alpha \approx 0.65$) and the exponential model $e^{\alpha x}$ (with $\alpha \approx 0.15$) cannot be separated. In 3D, an exponential model $e^{\alpha x}$ with $\alpha \approx 0.45$ is selected with high significance.

Our findings suggest that the protein lattice problem is not as severely affected by the vanishing of the spectral gap, as might have been expected. Problems with gaps smaller than 10^{-2} a.u. (and down to 10^{-8} a.u.) do not take significantly longer than problems with a median gap of 0.22 a.u. Moreover, our results hint at an exponential scaling, albeit with a potentially small rate constant. This analysis suggests that this quantum annealing application has a milder scaling than previously expected.

Table 1. Correlation coefficients between expected runtime and gap, for the results obtaining optimising the individual runtime and the results obtained with a fixed runtime of 1000 a.u. ρ is Spearman's rank correlation coefficient between Δ and T , which describes the monotonic relationship between the two variables ($\rho = +1$ is perfectly monotonic, $\rho = -1$ is perfectly inverse monotonic). R is Pearson's correlation coefficient between $\log_{10}(\frac{1}{\Delta^2})$ and $\log_{10} T$.

	ρ 2D	ρ 3D	R 2D	R 3D
Optimised time	−0.66	−0.44	0.52	0.38
Baseline	−0.73	−0.34	0.62	0.30

5. Improvement of the annealing pathway

In the previous section, we found that optimising the sample time can dramatically enhance the performance of quantum annealing, often by several orders of magnitude. This raises the question of whether other simple changes may be able to similarly increase efficiency. In this section we explore several approaches that deliver increasing improvements to the performance of the algorithm.

We first considered whether the linear interpolation function is the most appropriate choice, or if conversely a non-linear interpolation function increases the efficiency of the algorithm. As a first approach, we experimented with three alternative functions: quadratic (x^2), cubic (x^3) and, in order to include a rapidly changing schedule, sigmoid ($1/(1 + e^x)$). We optimised these functions following the same procedure as in the previous section. The results are summarised in figures S2 and S3.

Our results suggest that none of the functions considered is able to deliver a significant improvement. The quadratic function displays only minor deviations, and the cubic and sigmoid functions lead to markedly worse performance 99% of the cases. Moreover, when the runtime was reduced, the magnitude of the change was much smaller (roughly 80% of the original time to solution) than when the runtime was slowed down (140%–150% in the cubic case and 230%–300% in the sigmoid). This supports our initial choice of the linear annealing schedule to establish general trends.

We then considered the introduction of a non-stoquastic catalyst, which has been considered as a potential technique to improve quantum annealing [48–50]. As detailed in the methods section, this catalyst incorporates terms $\sigma_i^x \sigma_j^x$, and is multiplied by a tunable parameter λ that was optimised alongside the sample time. Introducing a catalyst of this form alters the sparsity of the Hamiltonian matrix that our code was optimised for, and therefore we could only consider one of the interpolation functions. This is adequate since our previous examination reveals that the choice of interpolation function has a small impact on efficiency. The results of these simulations are reported in figures S4 and S5.

The introduction of a non-stoquastic catalyst improves the runtime in about 5% of the worst-case examples, showing a notable average slowdown in the random sample. The magnitude of the deterioration seems to increase with the size of the sequence, while the proportion of sequences that are improved, as well as the magnitude of said improvement, decrease with the size of the sequence. These results suggest that while the non-stoquastic catalyst may well be useful in a fraction of the hardest problems, it is not a general solution. Incidentally, our results agree with those in a recent article by Crosson *et al* [51], providing significant analytical and numerical evidence that, in general, stoquastic Hamiltonians are more adept at optimisation than their non-stoquastic counterparts, while allowing however for specific, cleverly engineered non-stoquastic catalysts to provide excellent (sometimes, even exponential) speedups.

The theoretical arguments in [51] led us to consider a *stoquastic* catalyst that perturbs the annealing trajectory while maintaining stoquasticity. We conceived a catalyst that rotates the transverse field throughout annealing with a sum of terms σ_i^x . Incidentally, this choice of rotation axis preserves the sparse structure of the matrix and enables rapid simulation. As in the previous case, we introduced a tunable parameter λ that was optimised alongside the sample time.

The stoquastic catalyst also improves the runtime in about 5% of the examples. In this case, however, both the proportion of sequences and the magnitude of the improvement seem to increase with the size of the sequence, and while the improvement is generally more notable in the worst-case dataset than in the random dataset, both present some sort of improvement. The magnitude of the deterioration is significantly smaller than in the non-stoquastic case, and is smaller than an order of magnitude, compared to the 5–6 orders of magnitude introduced before. These results may indicate that a stoquastic catalyst is generally preferable to a non-stoquastic one.

After exploring these alternatives, we considered the design of *tailored* annealing trajectories: schedules designed to slow down near the spectral gap, and therefore reduce the ratio of excitation to local minima.

This improvement has been considered in several works [52–55], and in this particular problem it is motivated by the tendency of spectral gaps to concentrate around certain particular positions (see figure S8). In practice, a tailored trajectory is simply a piecewise linear function, defined on equally spaced bins, where the slope of the function at a given bin is inversely proportional to the probability of finding a gap of certain magnitude. We report the heuristic employed to design these schedules in the methods section. The results of these experiments are summarised in figures S9 and S10.

Tailored trajectories⁶ display better results than any of the previous approaches. Nearly half of the sequences in the random dataset, and about 80% of the worst-case dataset, experience some improvement. The magnitude of the speedup is also significantly larger, with many cases achieving accelerations between $10\times$ and $100\times$, and we observe an increasing trend where longer peptides improve more than shorter ones. Unfortunately, this technique has a significant caveat: that, when the strategy fails, the deterioration can be far worse (10^4 to 10^6 slow down in some cases).

A primary reason why the tailored approach can result in such slow annealing processes is because our heuristic has to balance the positions where there is a high probability of encountering the gap, and the magnitude of said energy difference. In some cases, particularly in the longer sequences, this balance is failing. A potential solution is to employ the distribution of gaps for every peptide length, instead of considering all the data at once. We therefore designed one annealing schedule per peptide length, and simulated the results as in the previous sections. The results of this experiment are reported in figures S11 and S12.

This approach is unsurprisingly the most successful of all, improving 60% of the sequences in the random dataset, and an impressive 95% of the sequences in the bad dataset. The magnitude of the improvement is similar, although the average improvement is higher; and we do not see the significant deterioration observed in the previous approach. This results suggest a potential strategy to optimise quantum annealing further.

In practical applications, the position of the gap will not be readily available. However, we have observed that the position of the gaps can be estimated by running short annealing runs with a sudden change at different parts of the schedule. We considered a piecewise linear function, similar to the one used in the tailored trajectories, but changing the slope only at the point where we probe for the gap. When applied to our dataset, this method shows some correlation (full dataset: $R = 0.17$, $p = 10^{-6}$; worst-case dataset: $R = 0.27$, $p = 5 \times 10^{-8}$), suggesting that more involved programs may be able to produce reasonable estimations of the gap position. We acknowledge, however, that this conclusion may not readily translate to larger sizes if the gaps become exponentially small in an exponentially small area, and that further experimentation is necessary.

The results in this section suggest that simple engineering of the Hamiltonians and trajectories involved in the annealing process can deliver a substantial increase in performance. In this sense, the results presented earlier represent a ‘sensible baseline’ of what is achievable with quantum annealing; and the experiments described in this section propose a pathway to improve the performance further. This also allows us to postulate that further experimentation in this domain may deliver better scalings (in relative terms, but potentially also in complexity) than previously reported.

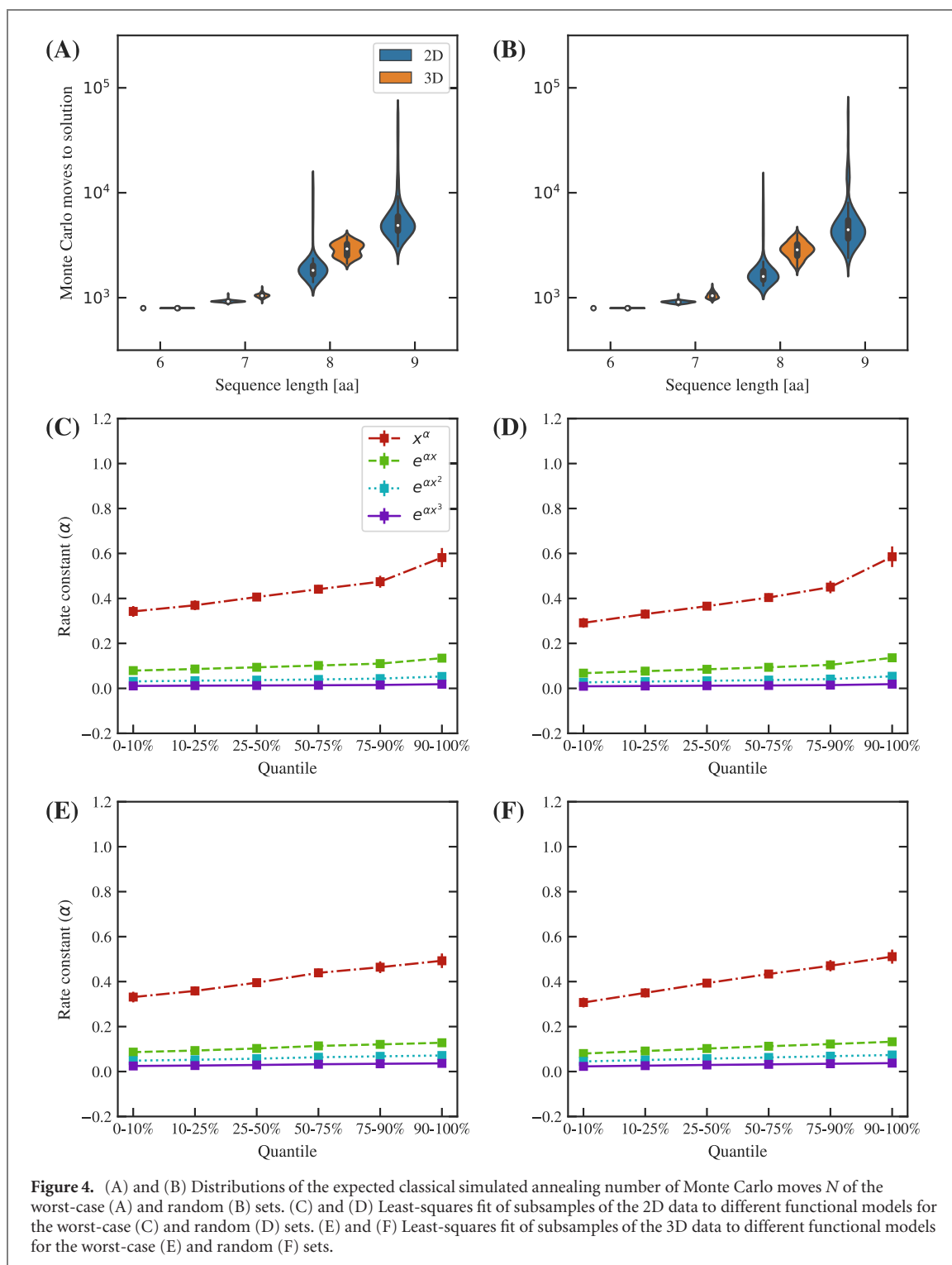
6. Comparison with simulated annealing

We have observed that quantum annealing requires exponentially growing runtimes to find the ground state of a protein lattice model, even in the small range of lengths explored in our dataset (see figure 3). Assuming this remains true of larger systems, as one would expect, it precludes an exponential speedup, since enumerating all possible conformations of a lattice model has the same asymptotic complexity. However, an algorithm which scales significantly better, that is, with a far smaller exponential rate α than the classical case could still be useful for practical applications.

In this section, we compare quantum annealing with classical simulated annealing using the data displayed in figure 4. Unlike other comparisons of quantum annealing and classical simulated annealing (e.g. [64]), we have not constrained the classical approach to solve a problem in the Ising form. More importantly, we consider an NP-hard problem, as opposed to previous work by Albash *et al* [64] that considered simpler problems.

The distributions presented in figures 4(A) and (B) show a rapidly growing number of Monte Carlo moves. Visually, the runtime appears to display worse-than-exponential growth. Our model comparison

⁶ Here we refer to tailored trajectories employing the cubic root function in the R -score (see section 2), which are the best performing in our benchmark.



analysis (see tables S8–S11 in appendix B) finds that, in all cases, the model fits to a square exponential $e^{\alpha x^2}$ with a high level of significance (and this behaviour is reproduced at every quantile).

There are some theoretical arguments (e.g. [65]) which conjecture that simulated annealing converges in exponential time. The square exponential fit found by our statistical analysis could be an artefact of parameter optimisation (note that we optimise four parameters for a single simulated annealing run, as opposed to only one in quantum annealing). This anticipates that quantum annealing provides a better scaling. Our results are made stronger by the fact that in this analysis we have considered only the number of Monte Carlo moves, i.e. the number of evaluations of the energy function. The cost of evaluating this energy is approximately quadratic on the size of the peptide, and the actual performance will be slightly worse than as depicted in figure 4. Of course, since this cost is polynomial, there will be no changes to the complexity of the algorithm, albeit the practical scaling will be worse.

These findings suggest that quantum annealing has an improved performance over simulated annealing. Over the system sizes we examined, the runtime of quantum annealing scales approximately exponentially, while simulated annealing shows a rapidly growing function that fits better to a square exponential.

7. Discussion

In this article, we have presented a numerical investigation of quantum annealing applied to protein lattice models. We have considered nearly 30 000 protein sequences, each with a unique global energy minimum, which represent realistic protein problems displaying a folded state, but are also high difficulty instances of an NP-hard problem.

We first turned our attention to the minimum spectral gap, a quantity connected theoretically to the runtime of a perfect annealer. We have observed that the gap for these protein sequences can decrease quickly in magnitude, although the scaling appears to be polynomial in the range of sizes considered. The polynomial scaling was confirmed by several statistical selection criteria (detailed in the methods section and appendix B), although comparison with prior results reported in the literature leads us to hypothesise that the gap vanishes exponentially. We have also observed that the worst cases decrease by five orders of magnitude between 6 and 9 amino acids. This numerical evidence shows that adiabatic evolution of the computer, where the probability of success nears 100%, will require rapidly growing runtimes and likely be infeasible for worst-case problems.

We then considered optimal annealing runs, where the computer is run for a shorter time to produce a small, but sizeable probability of success that is amplified by repetition. We have established that this runtime grows exponentially with peptide length, although the rate of growth was far smaller than the gap analysis suggested. The scaling was found to be approximately equal to $e^{0.15L}$ for 2D examples and approximately $e^{0.75L}$ for 3D examples within the range of problem sizes studied. We also found statistical evidence that peptides with very small gaps are not significantly harder than average cases, and that the exponential rates are almost identical for these two datasets.

We experimented with the parameters of the annealing process in pursuit of strategies to increase efficiency. We considered non-linear interpolation functions, non-stoquastic catalysts, and two adaptive programs to slow down the annealing schedule near regions where a potential gap is expected. Our results demonstrate that the last approach is able to deliver 10–100× speedups. Furthermore, we observe a trend of increasing relative improvements as the size increases: the vast majority of the peptides in the worst-case dataset were improved by this strategy, suggesting that potential decreases in performance can be addressed via careful engineering. We also suggest a method to estimate the gap position when this is not readily available.

A comparison with classical simulated annealing on our dataset shows that the quantum annealing approach is preferable. Statistical modelling seems to suggest that the scaling of simulated annealing fits best to a square exponential, $e^{\alpha x^2}$ although theoretical arguments lead us to expect this behaviour to become exponential with a large rate as problem size increases. This implies that for large peptide sizes, a quantum annealer may take significantly less time than a classical machine running a stochastic algorithm. We note, however, that although classical simulated annealing is the state-of-the-art for physics-based protein structure prediction, other classical algorithms may exhibit superior performance either minimising a classical energy function, or finding the minimum of the PUBO problem that we have considered with quantum annealing.

One of the reasons why quantum annealing may prove useful for protein folding and structure prediction is the limited size of interesting problems. More than half the structures deposited in the Protein DataBank contain fewer than 500 residues, and 80% of the domains in the CATH database [66] are smaller than 200 residues. Similarly, the median length of a human protein is 375 amino acids long [30]. Even if the scaling of quantum annealing is exponential, as long as the exponential rate is low enough to fold small proteins or domains in a timely fashion, this approach will be useful for a multitude of practical problems.

There are other advantages to quantum annealing that could be explored further. When the algorithm fails, the system has been excited to a higher energy state, but although this will only be a local minimum, it may still be useful. For example, if this result is used in a bottom-up approach to explore the conformational space of a protein, it may still be a good starting point for more complex simulations. In contrast, classical simulated annealing will often fail to provide a solution that is close to the global minimum. Future work will investigate this fact, and aim to establish ‘crossover’ points where the scaling worsens.

We believe these results suggest that annealing approaches to quantum optimisation may be a powerful heuristic approach to solve the protein lattice problem, whether in specialised processors [38], near-term digital quantum computers [67] or otherwise. The difficulties of the quantum approach are shared by the

classical simulated annealing approach, while the scaling is better. Moreover, even in cases where the annealing approach fails, it can provide solutions that despite not being equivalent to the global minimum, are very close to it, and may be useful in a subsequent refinement procedure. These findings offer encouragement for further research in quantum protein lattice folding and other hybrid quantum–classical algorithms for protein structure prediction.

Acknowledgments

CO would like to thank F Hoffmann-La Roche, UCB and the UK National Quantum Technologies Programme for financial support through an EPSRC studentship (EP/M013243/1). GMM thanks the EPSRC and MRC for support via EP/L016044/1 and EP/S024093/1. SCB acknowledges support from the EU Flagship project AQTION, the NQIT Hub (EP/M013243/1) and the QCS Hub. The authors would like to thank Daniel Nissley and Niel de Beaudrap for useful discussions.

Data availability

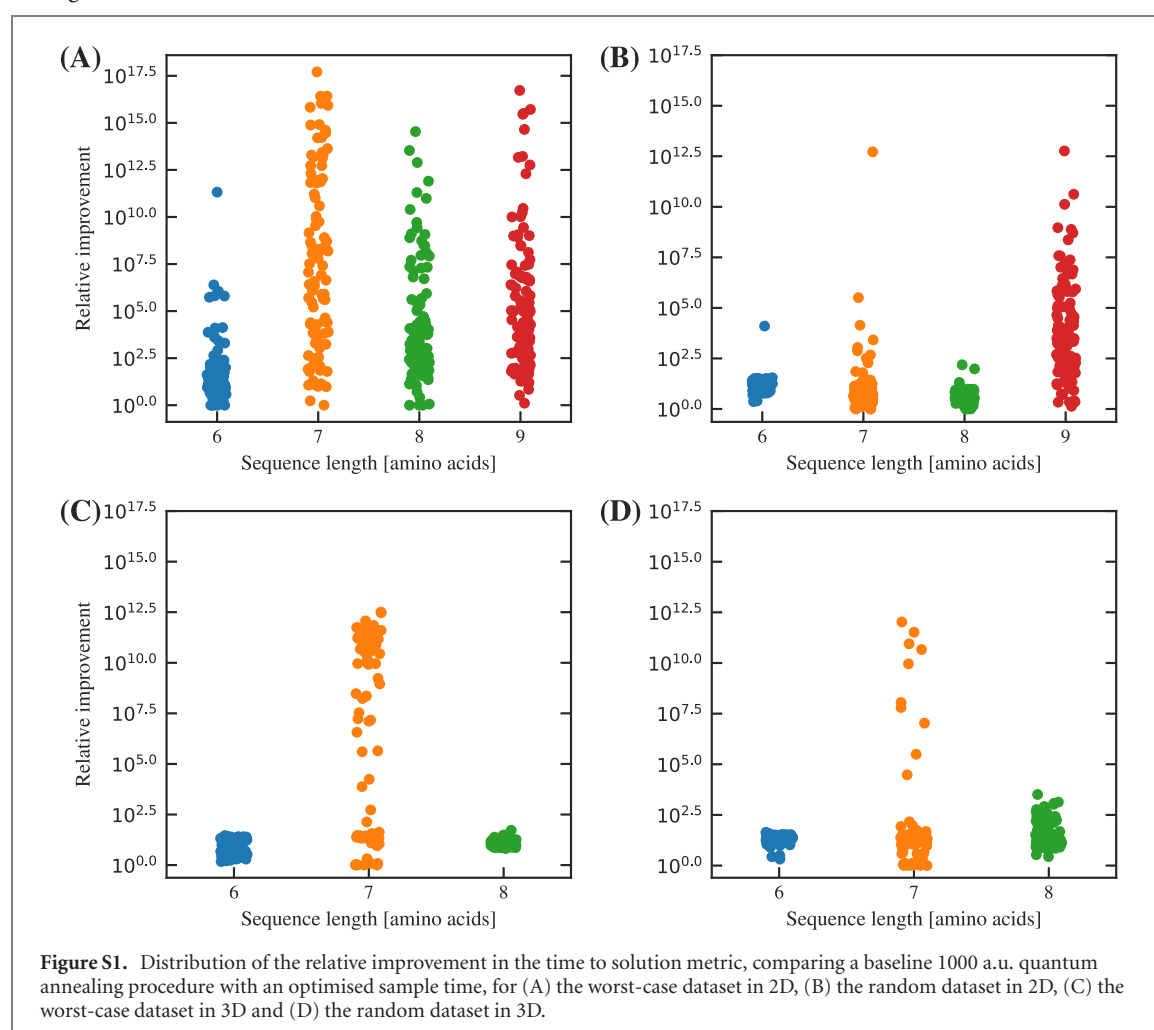
The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: <https://github.com/couteiral/proteinham>. Data will be available from 15 October 2021.

Code availability

The code that supports the findings of this study will be made available after acceptance at <https://github.com/couteiral/proteinham>.

Appendix A. Additional figures

See figures S1–S20 and table S1.



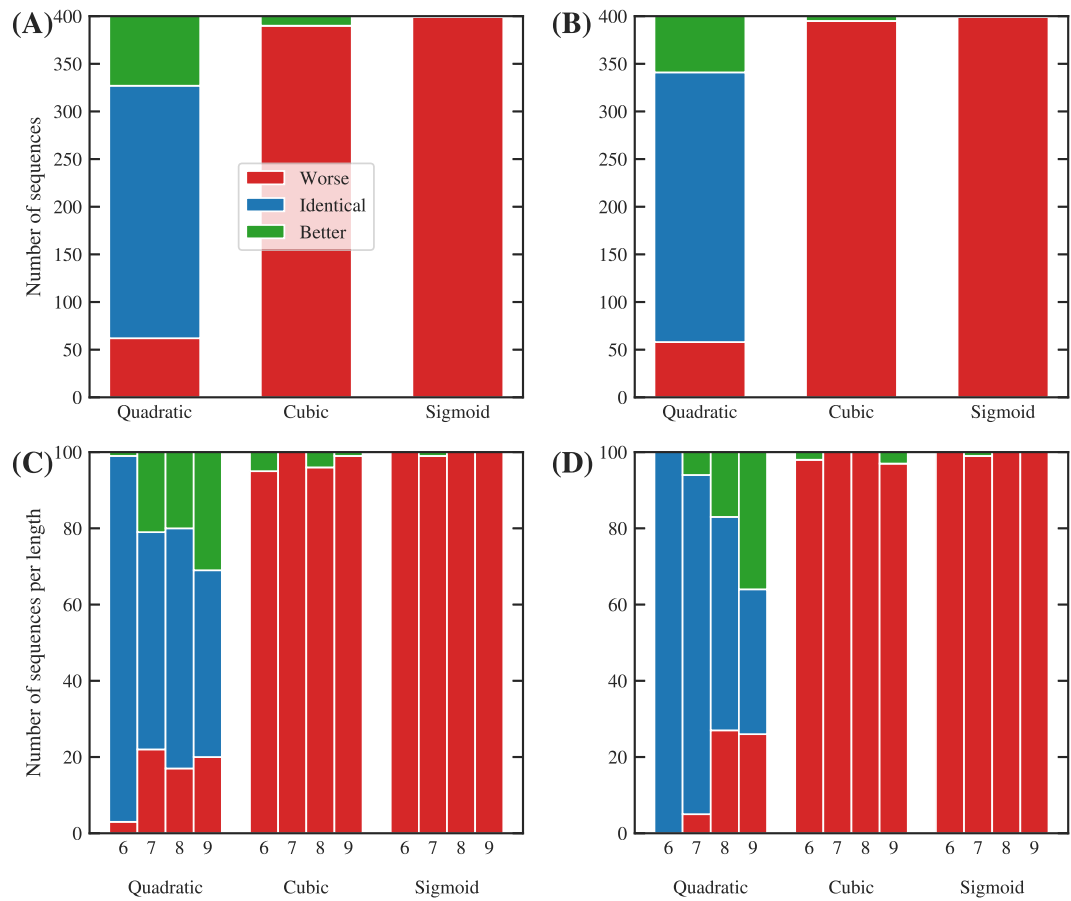


Figure S2. Variation of the annealing process using alternative interpolation functions with respect to the linear baseline. We consider quadratic (x^2), cubic (x^3) and sigmoid ($1/(1 + e^x)$) interpolations. (A) and (B) Proportion of solutions with a worse (red), identical (blue) or better (green) expected runtime than the baseline, for the worst-case (A) and random (B) datasets. (C) and (D) Proportions classified by length for the worst-case (C) and random (D) datasets. The quadratic function shows scarce deviation from the baseline, while both the cubic and sigmoid functions result in worse performances for all but a few cases.

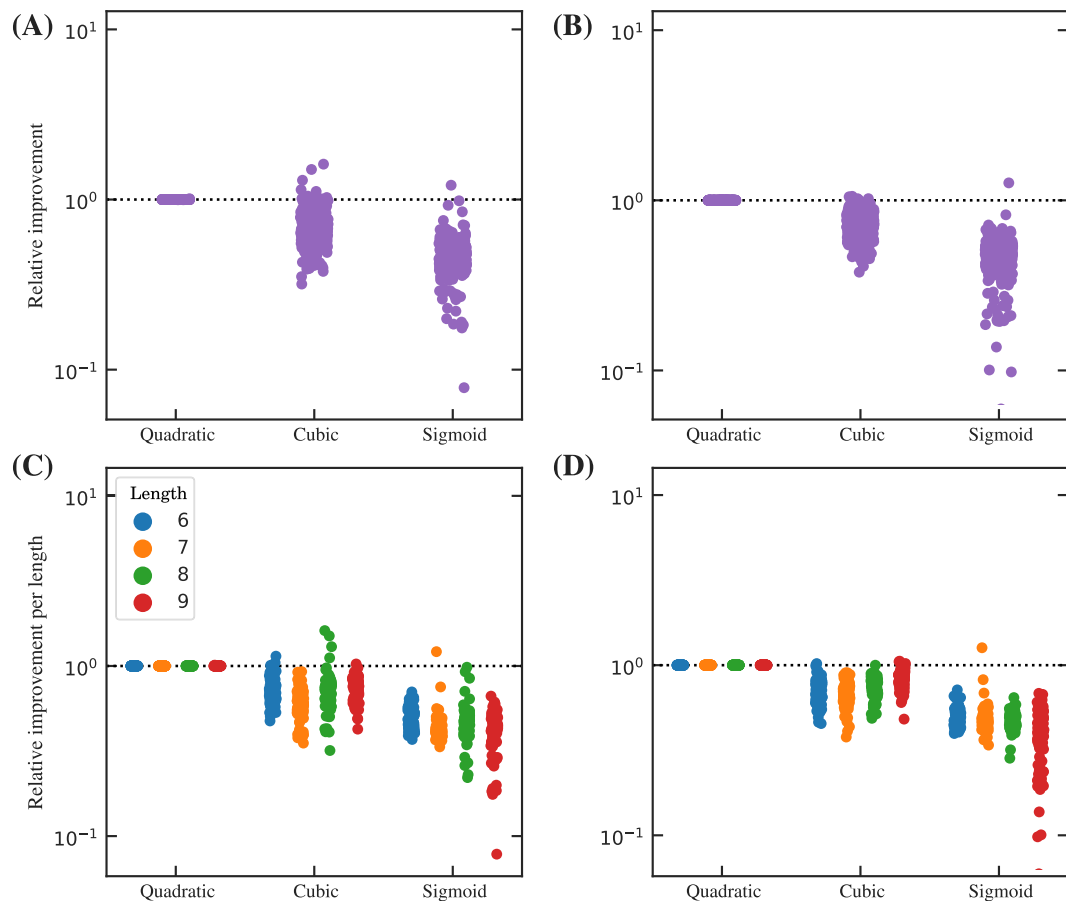


Figure S3. Relative improvement of the annealing process using alternative interpolation functions with respect to the linear baseline. We consider quadratic (x^2), cubic (x^3) and sigmoid ($1/(1 + e^x)$) interpolations. Relative improvement of the expected time to solution time for the worst-case (A) and random (B) datasets. (C) and (D) Relative improvement classified by length for the worst-case (C) and random (D) datasets. Relative improvement here is defined as the ratio between the expected time to solution of the optimised non-stochastic run and the stochastic baseline: values under 1.0 (dotted line) indicate worse performance, while values over 1.0 are improved by the choice of interpolation function. We observe that most choices of interpolation functions show worse performance than linear interpolation.

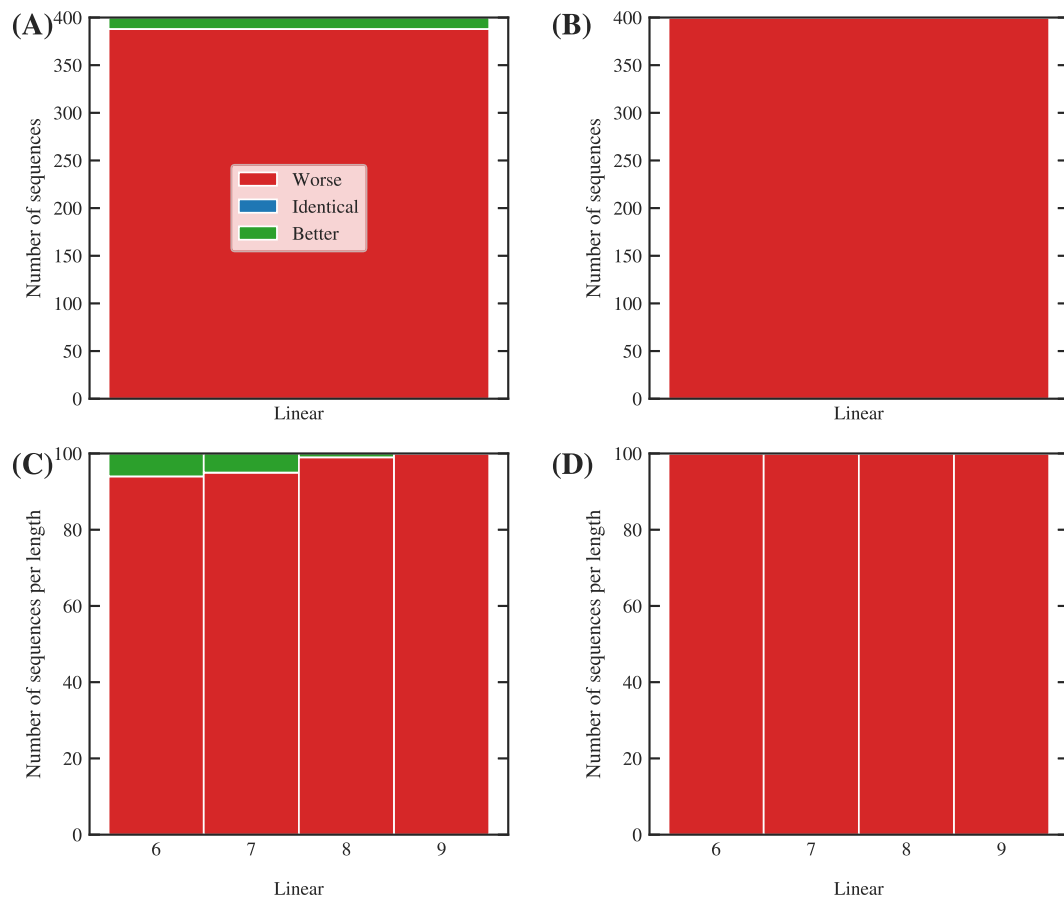


Figure S4. Variation of the annealing process with a non-stoquastic catalyst. (A) and (B) Proportion of solutions with a worse (red), identical (blue) or better (green) expected runtime than the baseline, for the worst-case (A) and random (B) datasets. (C) and (D) Proportions classified by length for the worst-case (C) and random (D) datasets. The non-stoquastic catalyst improves the annealing process in a reduced proportion of the worst-case examples, negatively affecting the performance of the rest.

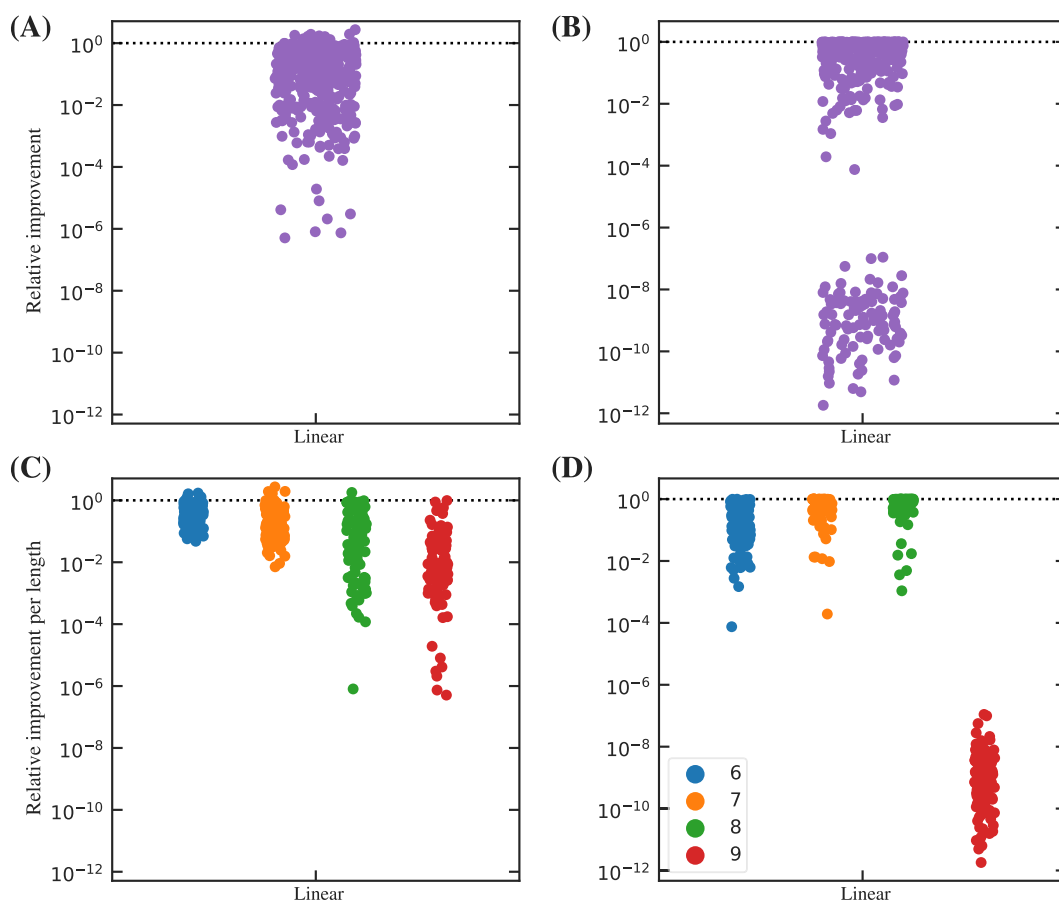


Figure S5. Relative improvement of the annealing process using a non-stoquastic catalyst, with respect to the stoquastic baseline. (A) and (B) Relative improvement of the expected time to solution time for the worst-case (A) and random (B) datasets. (C) and (D) Relative improvement classified by length for the worst-case (C) and random (D) datasets. Relative improvement here is defined as the ratio between the expected time to solution of the optimised non-stoquastic run and the stoquastic baseline: values under 1.0 (dotted line) indicate worse performance, while values over 1.0 are improved by the non-stoquastic catalyst. Our results suggest that the non-stoquastic catalyst notably worsens the performance of quantum annealing in virtually all cases, with a modest improvement in a fraction of the worst-case examples. Note that some of the results for size 9 peptides had to be restricted to a smaller set of iterations due to increased computational burden, which may explain why size 9 random peptides seem to perform significantly worse than the trend would suggest.

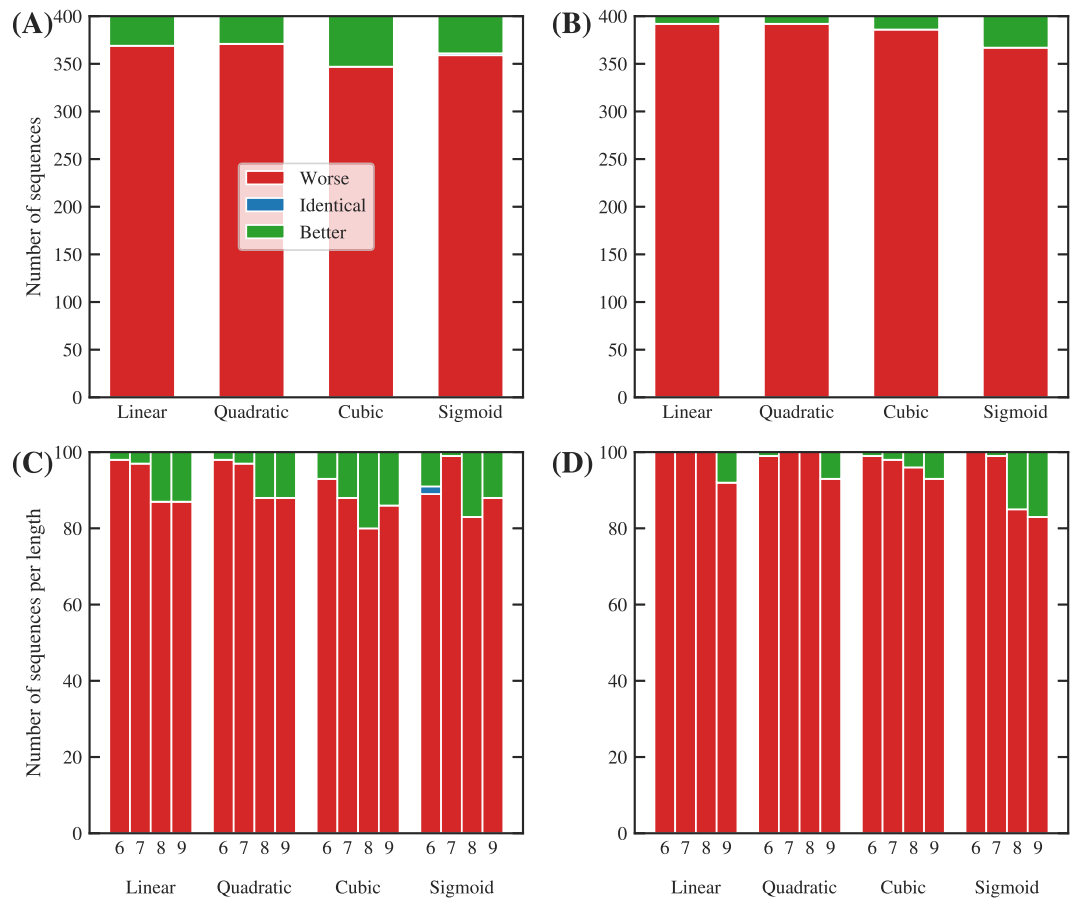


Figure S6. Variation of the annealing process with a stoquastic catalyst. We consider linear (x), quadratic (x^2), cubic (x^3) and sigmoid ($1/(1 + e^x)$) interpolation functions for the annealing schedule. (A) and (B) Proportion of solutions with a worse (red), identical (blue) or better (green) expected runtime than the baseline, for the worst-case (A) and random (B) datasets. (C) and (D) Proportions classified by length for the worst-case (C) and random (D) datasets. The non-stoquastic catalyst improves the annealing process in a reduced proportion of the cases, negatively affecting the performance of the rest.

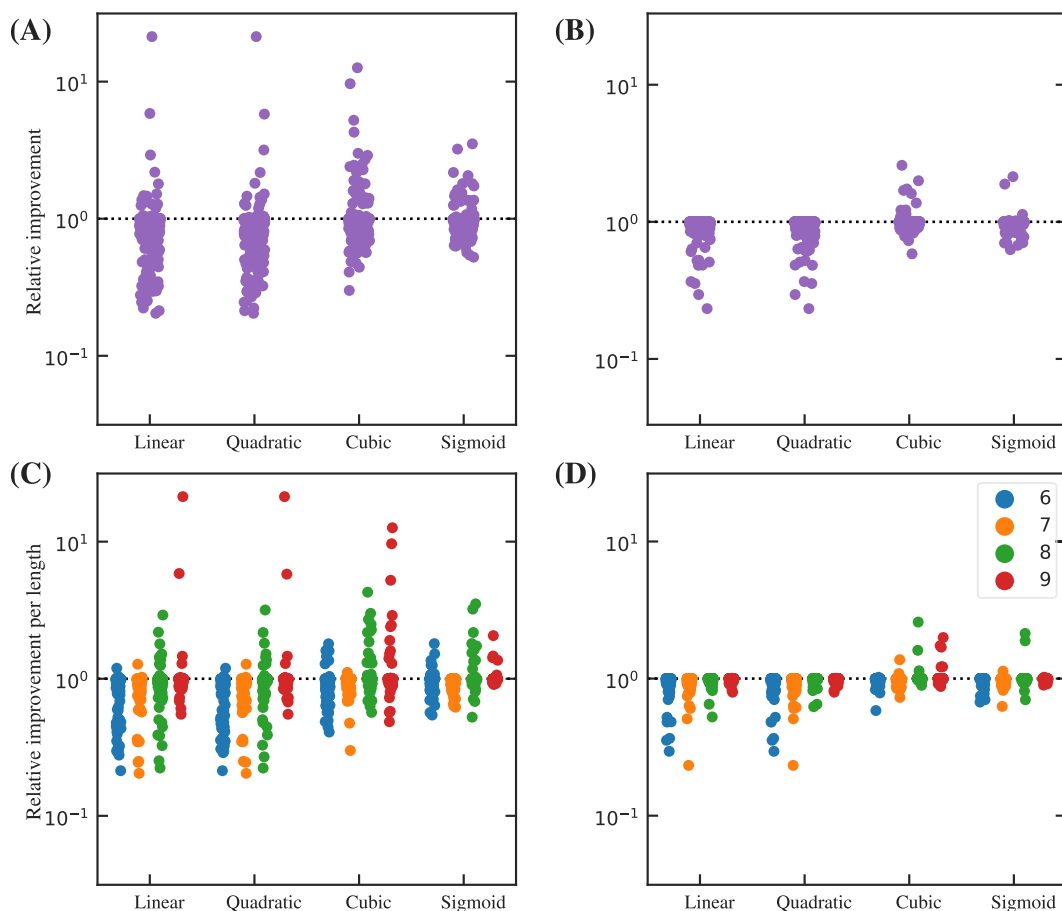


Figure S7. Relative improvement of the annealing process using a stoquastic catalyst, with respect to the non-catalysed baseline. We consider linear (x), quadratic (x^2), cubic (x^3) and sigmoid ($1/(1 + e^x)$) interpolation functions for the annealing schedule. (A) and (B) Relative improvement of the expected time to solution time for the worst-case (A) and random (B) datasets. (C) and (D) Relative improvement classified by length for the worst-case (C) and random (D) datasets. Relative improvement here is defined as the ratio between the expected time to solution of the optimised non-stoquastic run and the stoquastic baseline: values under 1.0 (dotted line) indicate worse performance, while values over 1.0 are improved by the non-stoquastic catalyst. Our results suggest that the magnitude of the improvement is much better in the worst-case dataset, with some examples reaching $20\times$ acceleration.

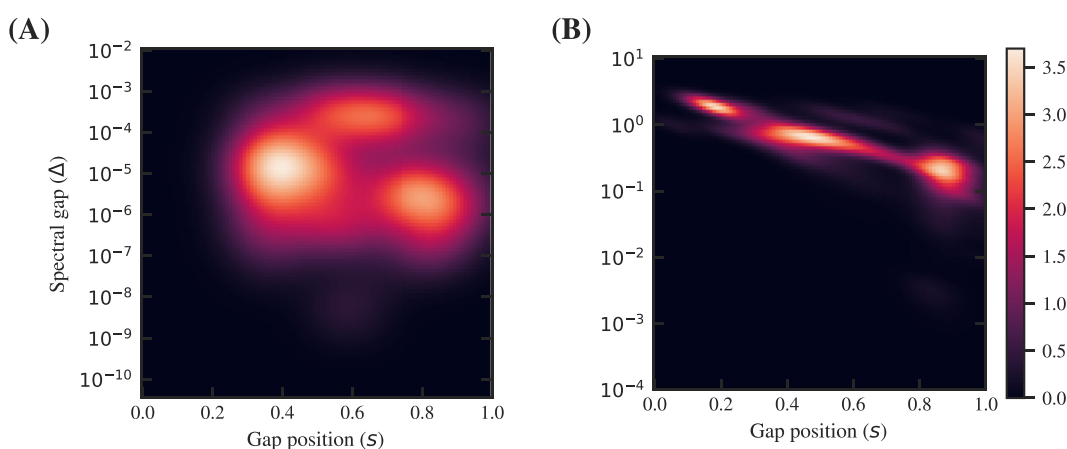


Figure S8. Distribution of spectral gap positions and magnitudes for 2D (A) and 3D (B) peptide sequences. The distributions have been estimated using Gaussian kernel density estimation (KDE). In the case of 2D peptides, we have removed the gaps greater than 10^{-3} to improve the visualization of smaller gaps; in the right plot we have used all the gaps in the dataset.

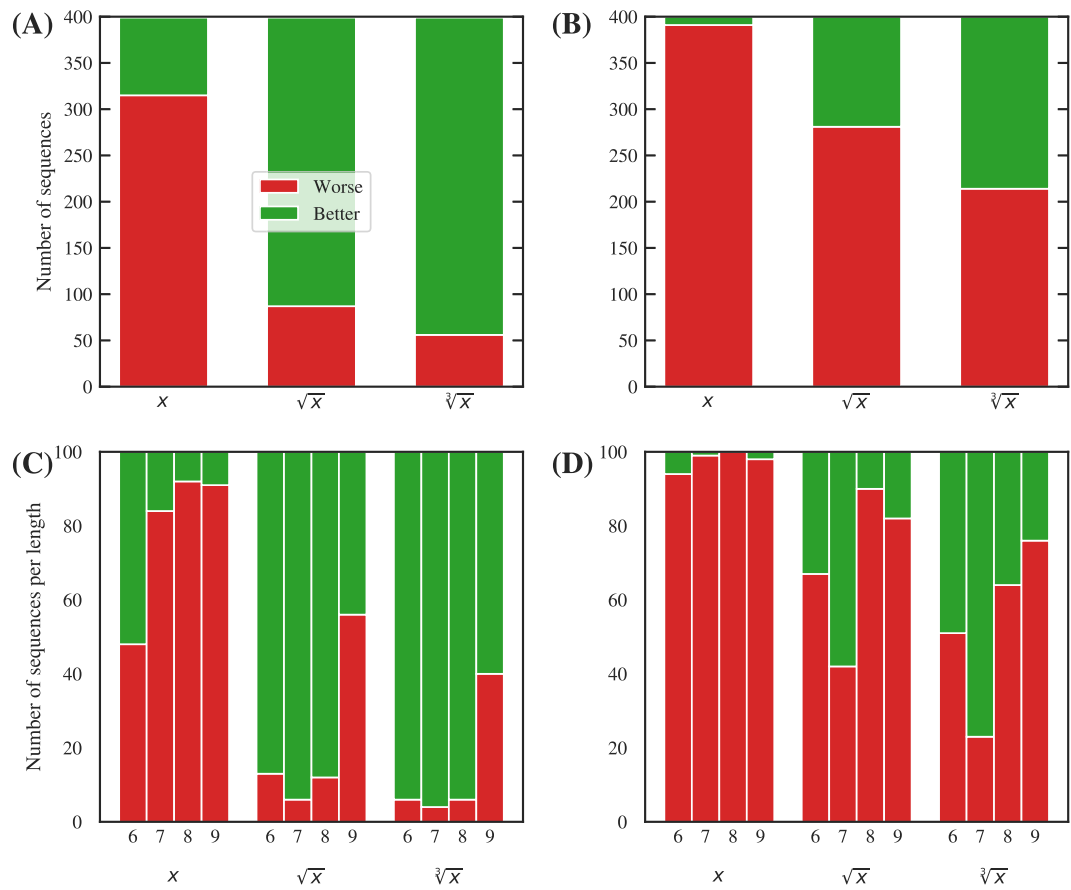


Figure S9. Variation of the annealing process with an annealing schedule built to slow down at regions most likely to display the minimal gap for any length. We consider linear (x), square root (\sqrt{x}) and cubic root ($\sqrt[3]{x}$) functions to compute the R -score (see section 2). (A) and (B) Proportion of solutions with a worse (red), identical (blue) or better (green) expected runtime than the baseline, for the worst-case (A) and random (B) datasets. (C) and (D) Proportions classified by length for the worst-case (C) and random (D) datasets.

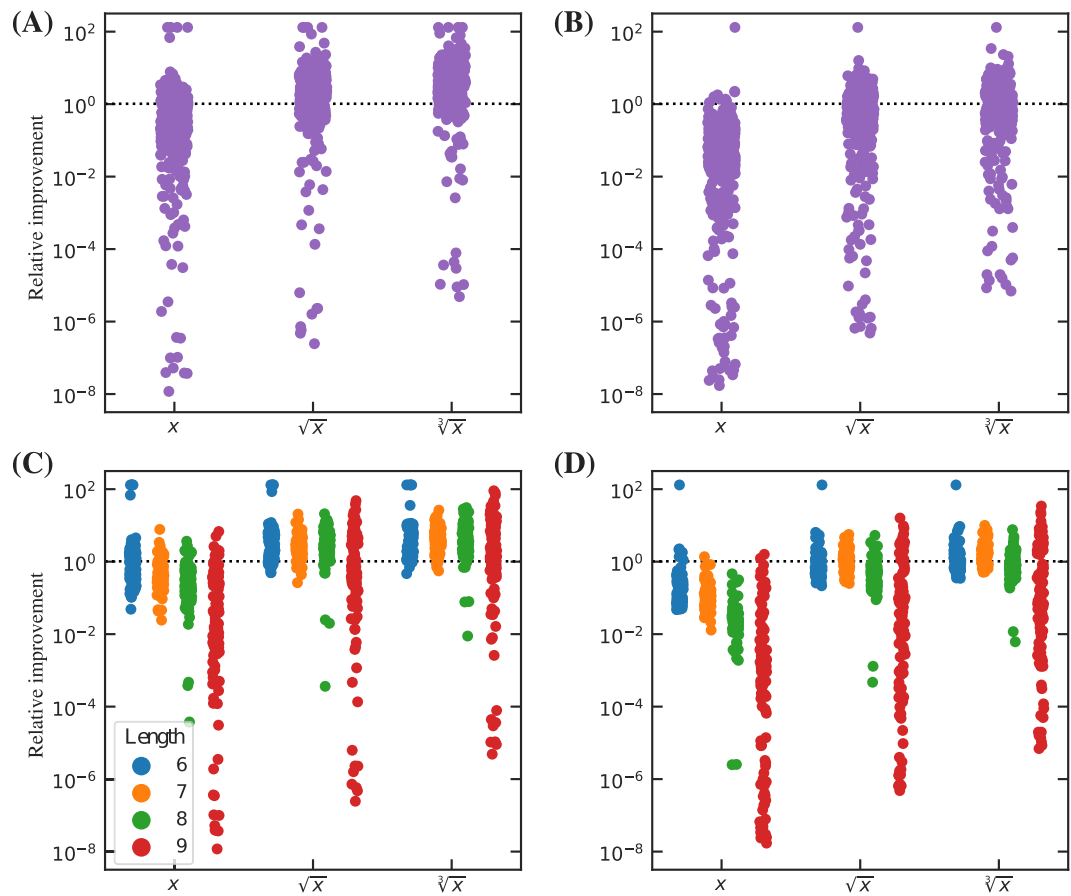


Figure S10. Relative improvement of the annealing process with an annealing schedule built to slow down at regions most likely to display the minimal gap for any length. We consider linear (x), square root (\sqrt{x}) and cubic root ($\sqrt[3]{x}$) functions to compute the R-score (see section 2). (A) and (B) Relative improvement of the expected time to solution time for the worst-case (A) and random (B) datasets. (C) and (D) Relative improvement classified by length for the worst-case (C) and random (D) datasets. Relative improvement here is defined as the ratio between the expected time to solution of the tailored trajectory over the baseline: values under 1.0 (dotted line) indicate worse performance, while values over 1.0 are improved by the tailored annealing schedule.

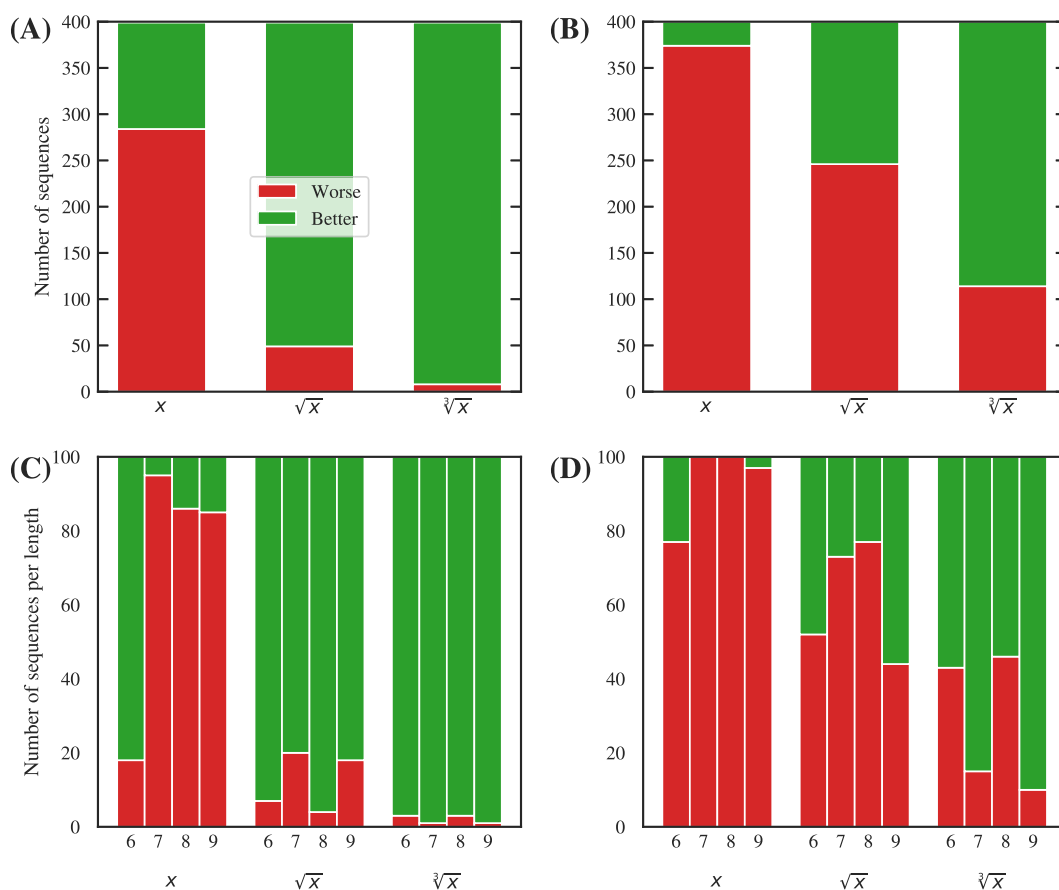


Figure S11. Variation of the annealing process with an annealing schedule built to slow down at regions most likely to display the minimal gap for the peptide's length. We consider linear (x), square root (\sqrt{x}) and cubic root ($\sqrt[3]{x}$) functions to compute the R-score (see section 2). (A) and (B) Proportion of solutions with a worse (red), identical (blue) or better (green) expected runtime than the baseline, for the worst-case (A) and random (B) datasets. (C) and (D) Proportions classified by length for the worst-case (C) and random (D) datasets.

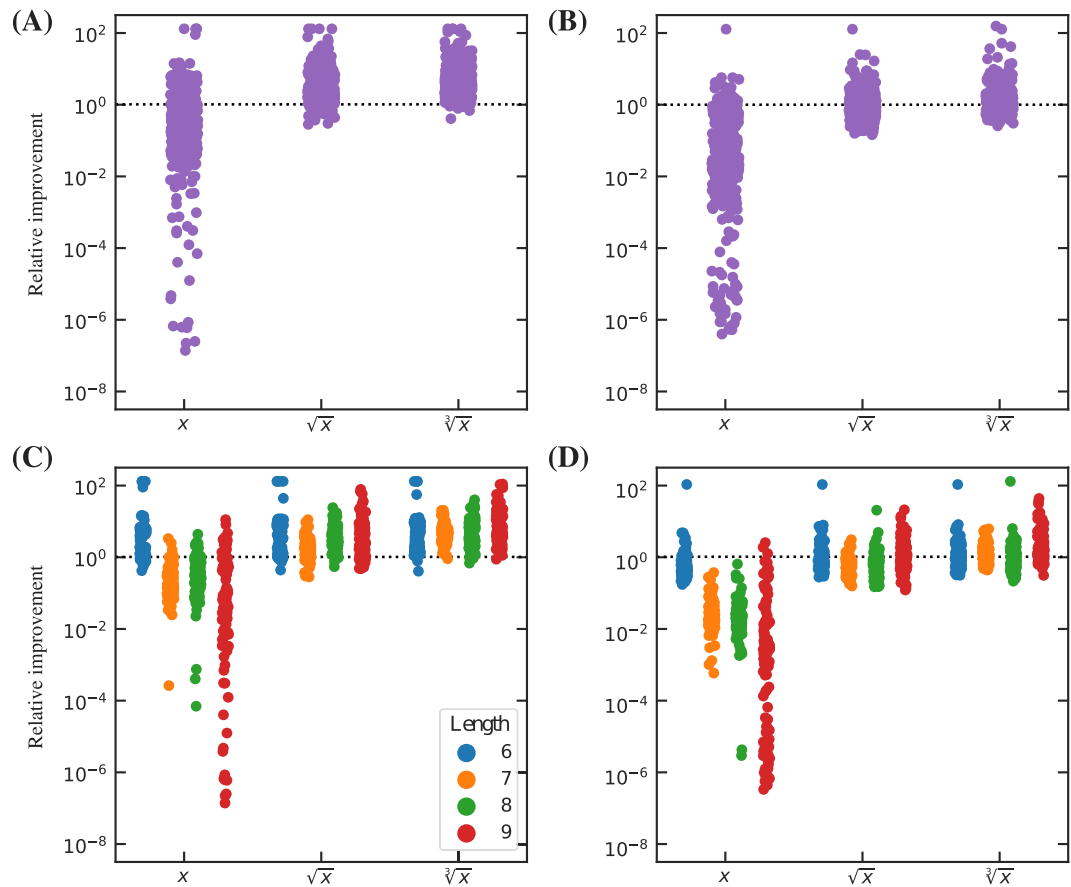


Figure S12. Relative improvement of the annealing process with an annealing schedule built to slow down at regions most likely to display the minimal gap for the peptide's length. We consider linear (x), square root (\sqrt{x}) and cubic root ($\sqrt[3]{x}$) functions to compute the R -score (see section 2). (A) and (B) Relative improvement of the expected time to solution time for the worst-case (A) and random (B) datasets. (C) and (D) Relative improvement classified by length for the worst-case (C) and random (D) datasets. Relative improvement here is defined as the ratio between the expected time to solution of the tailored trajectory over the baseline: values under 1.0 (dotted line) indicate worse performance, while values over 1.0 are improved by the tailored annealing schedule.

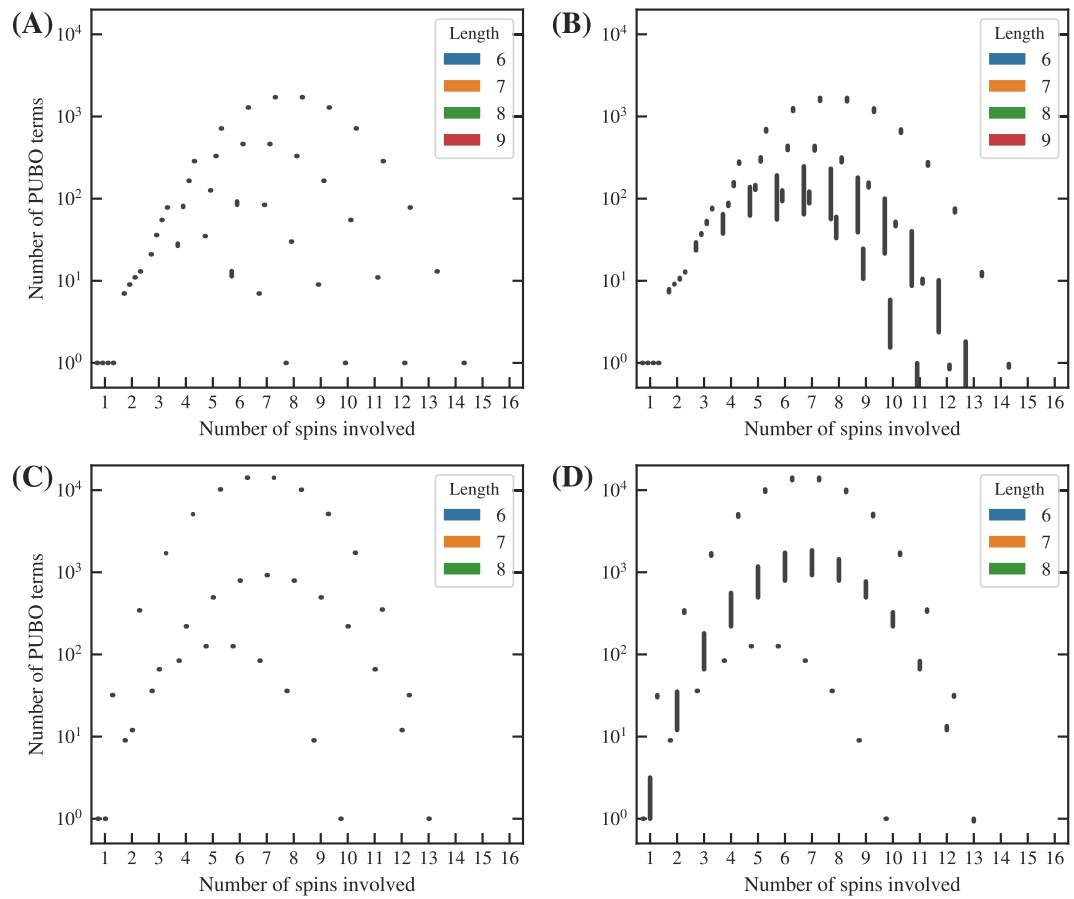


Figure S13. Distribution of PUBO terms in the problem Hamiltonians. We plot the number of PUBO terms that involve exactly N spins. (A) and (B) Distribution of terms for the worst-case (A) and random (B) two-dimensional datasets. (C) and (D) Distribution of terms for the worst-case (C) and random (D) three-dimensional datasets.

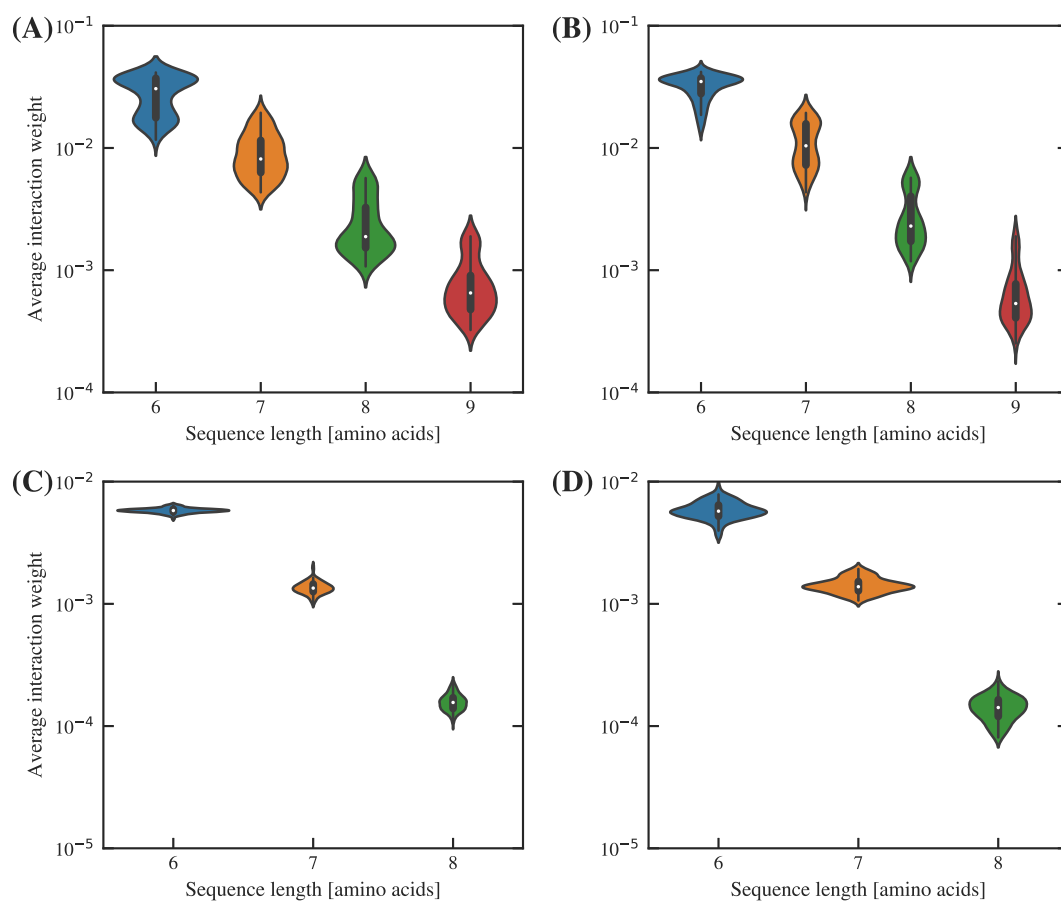


Figure S14. Distribution of the average weight of the terms in the polynomial unconstrained binary optimization (PUBO) problems, stratified by dimension, dataset and protein length. (A) and (B) Distribution of average weight for the worst-case (A) and random (B) two-dimensional datasets. (C) and (D) Distribution of average weight for the worst-case (C) and random (D) three-dimensional datasets.

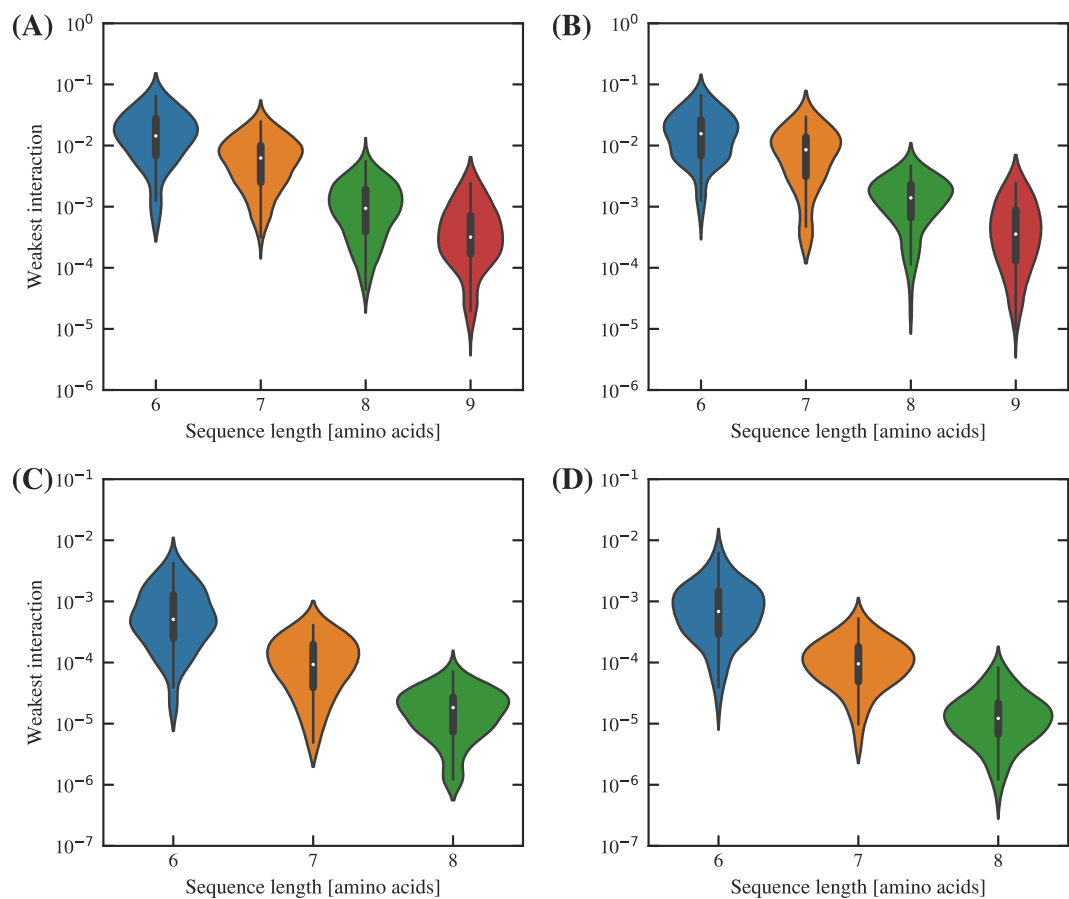


Figure S15. Distribution of the magnitude weakest terms in the polynomial unconstrained binary optimization (PUBO) problems, stratified by dimension, dataset and protein length. (A) and (B) Distributions for the worst-case (A) and random (B) two-dimensional datasets. (C) and (D) Distributions for the worst-case (C) and random (D) three-dimensional datasets.

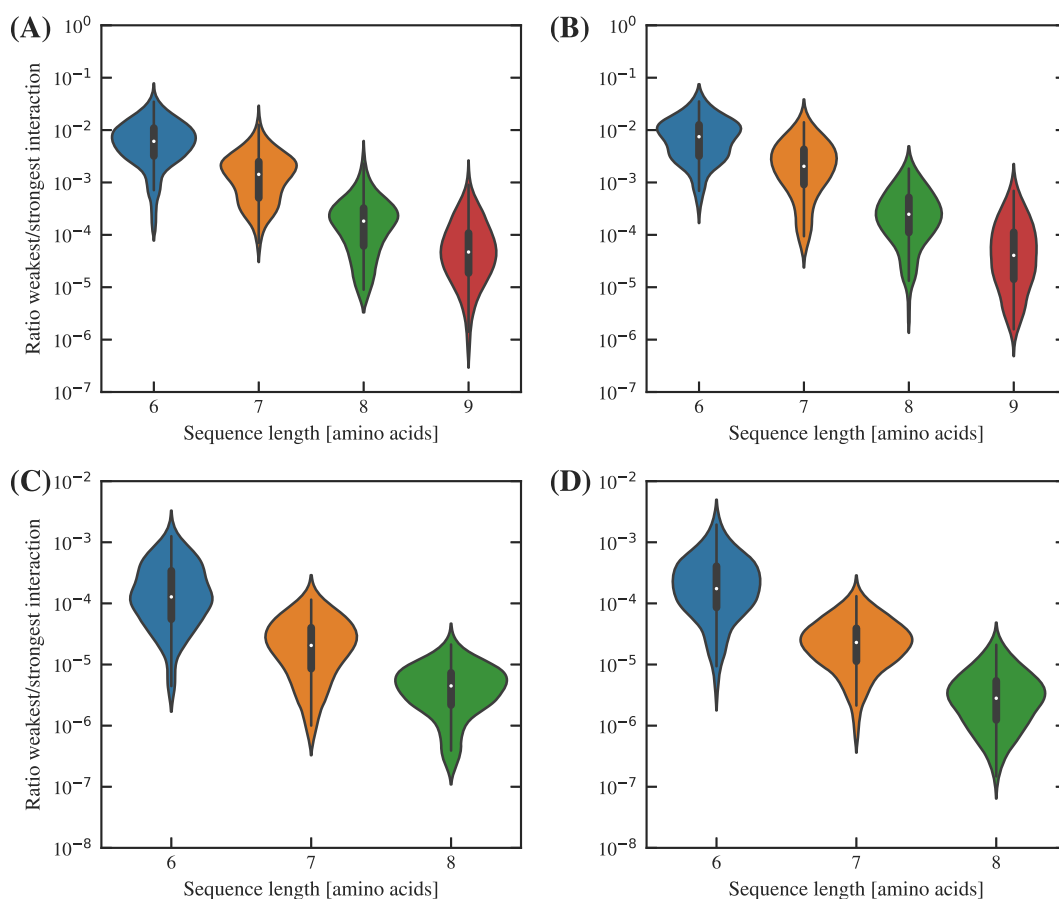


Figure S16. Distribution of the ratio of the weakest to the strongest terms in the polynomial unconstrained binary optimization (PUBO) problems, stratified by dimension, dataset and protein length. (A) and (B) Distribution of the ratio for the worst-case (A) and random (B) two-dimensional datasets. (C) and (D) Distribution of the ratio for the worst-case (C) and random (D) three-dimensional datasets.

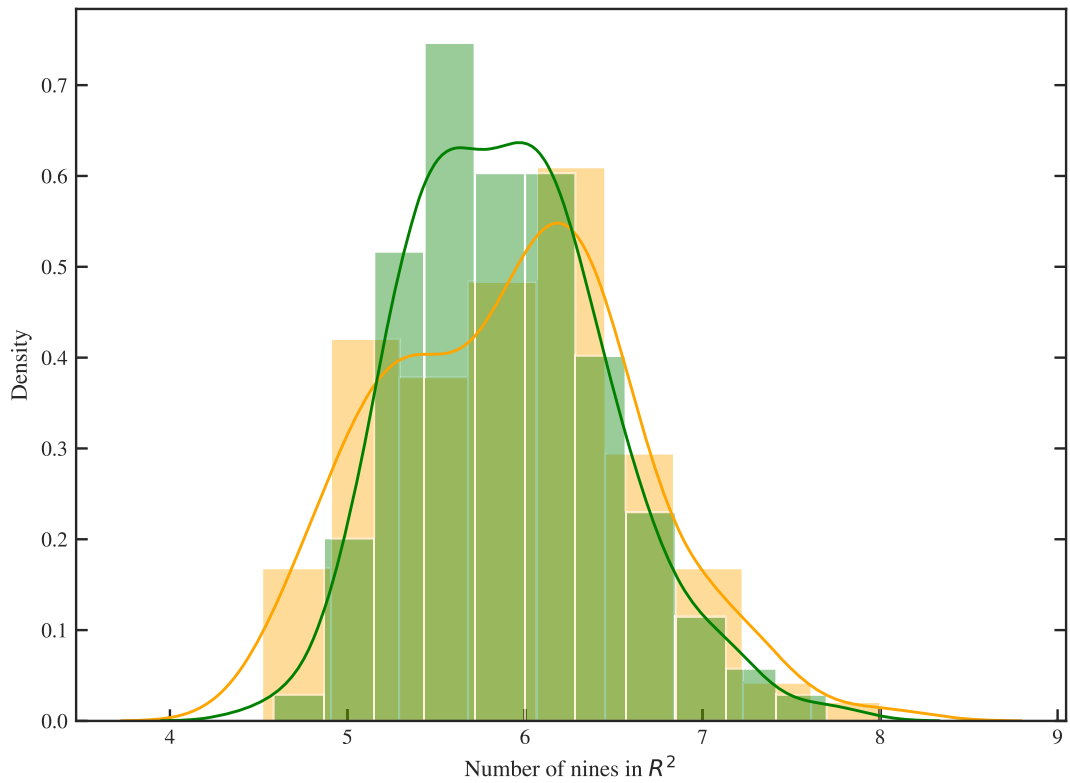


Figure S17. Validation of the cubic splines model used to analyse the gap. The eigenvalues of 120 two-dimensional Hamiltonians selected by stratified random sampling in both length and dataset (worst-case and average-case) were calculated with a grid spacing of $\Delta s = 0.001$. Cubic splines models were fit to a $\Delta s = 0.1$ grid identical to that used in section 3, and the predictions for the rest of the grid were compared with the real values. We display the distribution of R^2 coefficients for all fits for the ground state and first excited state. In every case, the value is over $R^2 = 0.9999$ (four nines), suggesting an excellent fit.

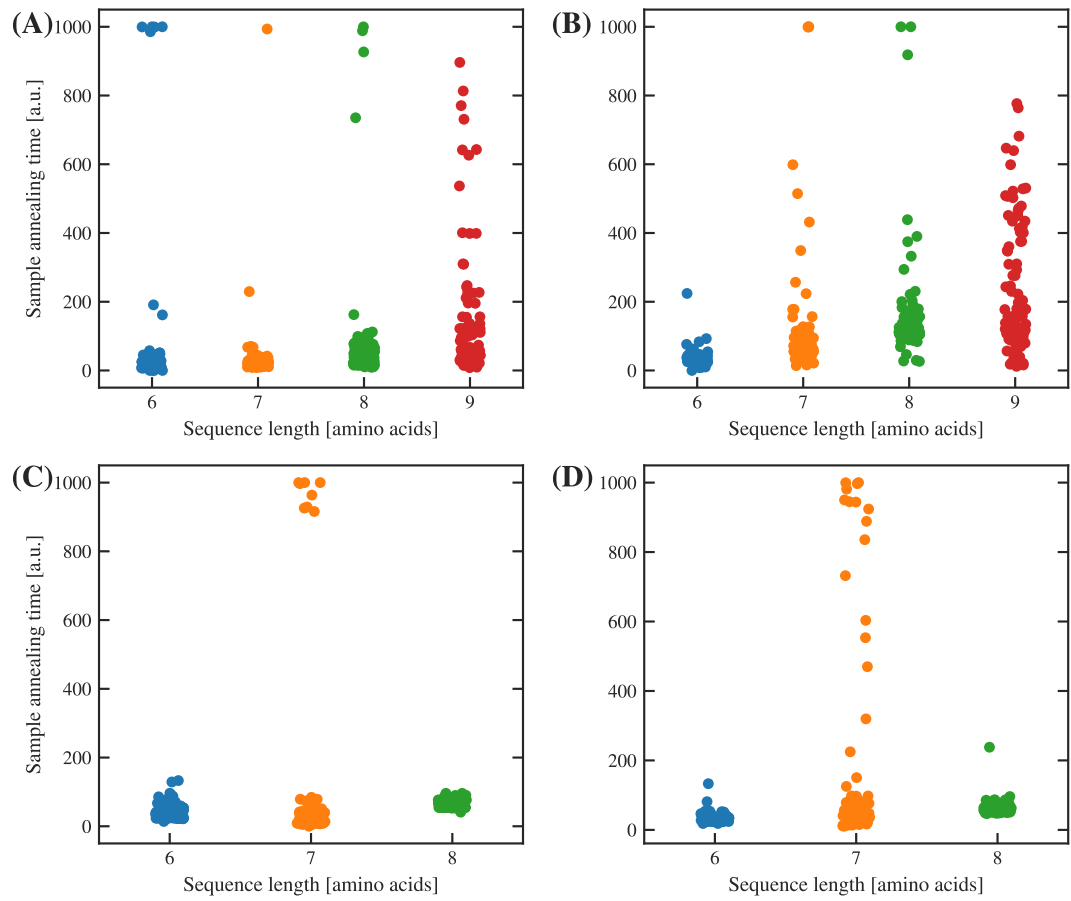


Figure S18. Distribution of the optimised sample annealing times for the Hamiltonians studied in section 4. (A) and (B) Distribution of the sample annealing time for the worst-case (A) and random (B) two-dimensional datasets. (C) and (D) Distribution of the sample annealing times for the worst-case (C) and random (D) three-dimensional datasets.

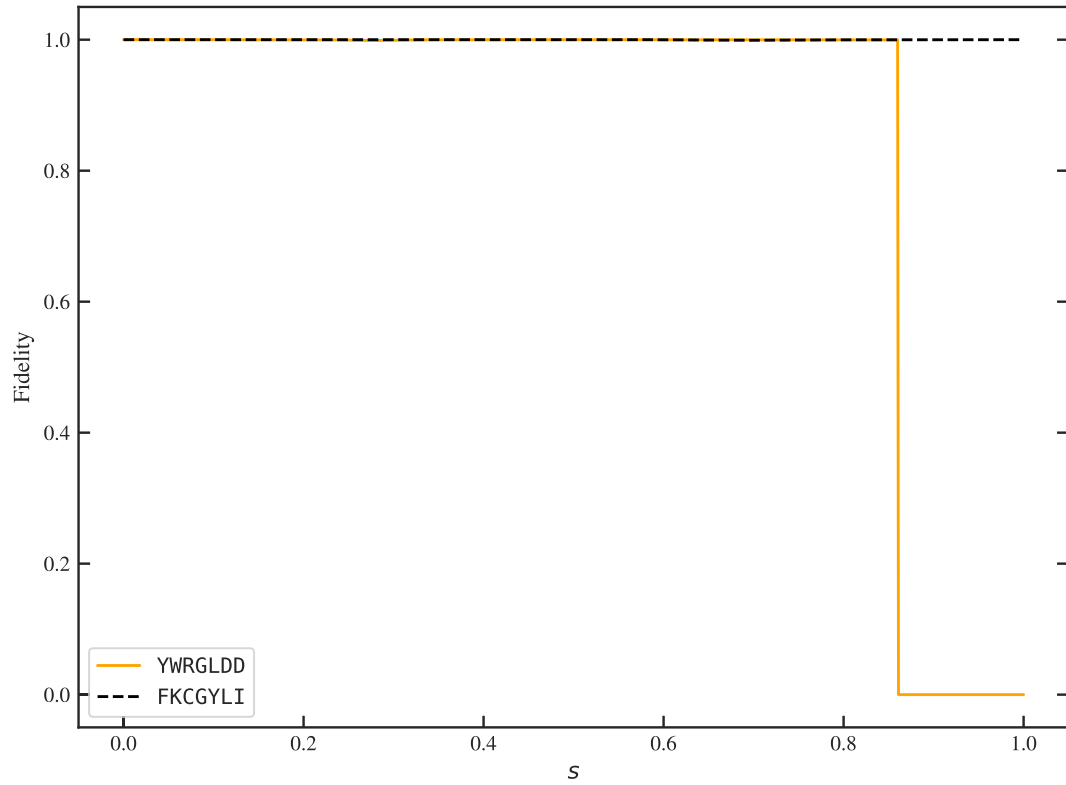


Figure S19. Fidelity to the instantaneous ground state for two annealing experiments ($T = 1000$ a.u.) with two-dimensional Hamiltonians of size 7: an average-case (FKCGYLI, $\Delta = 0.07$) and a worst-case (YWRGLDD, $\Delta = 7.56 \times 10^{-4}$). In the average-case, the evolution is fully adiabatic, remaining over 99.9% throughout the evolution. In the worst-case, the fidelity suddenly drops to 10^{-7} shortly after traversing the region with the minimum gap.

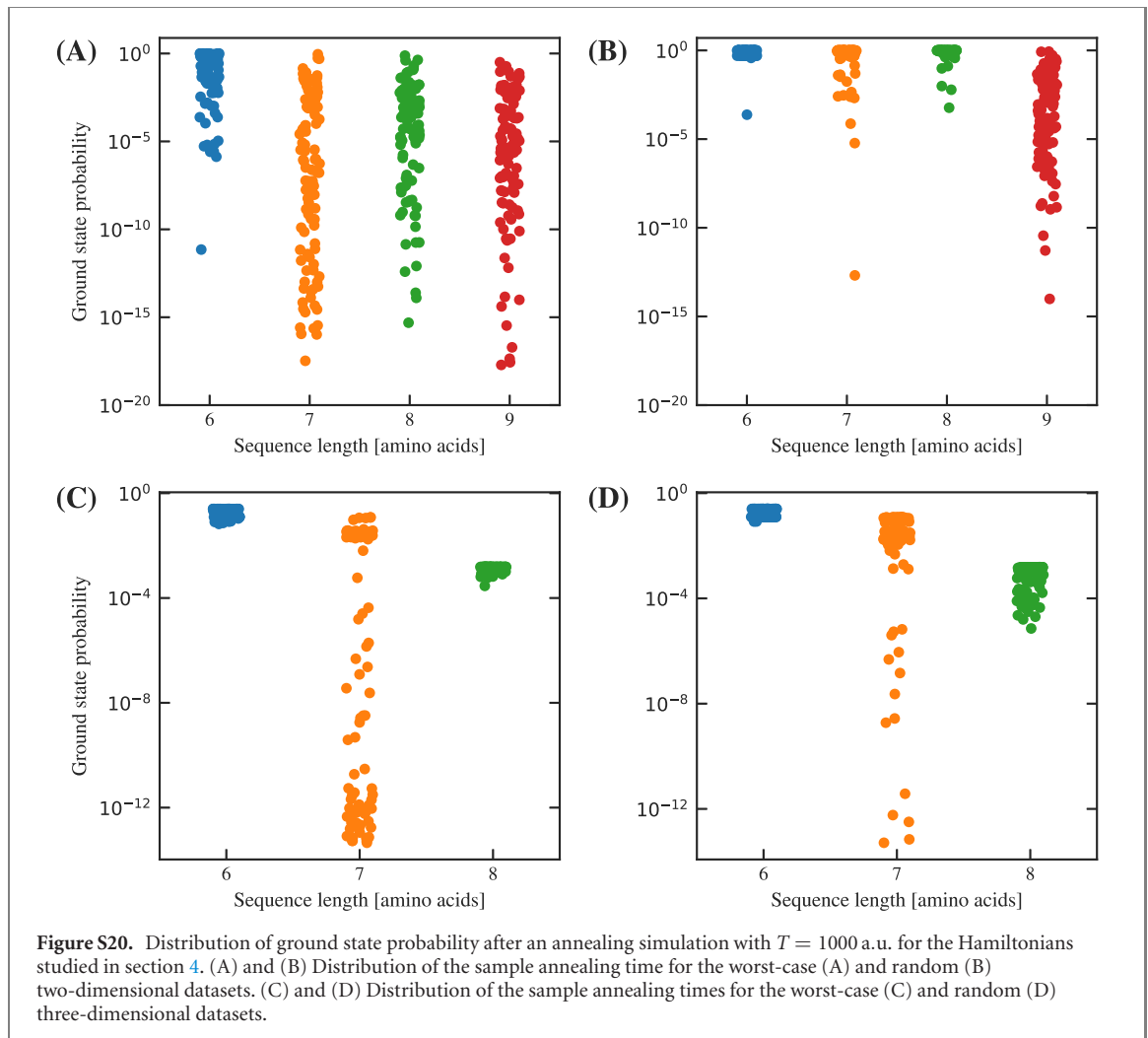


Table S1. Correspondence between the size of a protein lattice problem and the number of qubits of the corresponding polynomial unconstrained binary optimization (PUBO) problem. On the left-hand side we display the numbers for 2D proteins, and on the right-hand side the figures for 3D proteins. In this work we use the turn encoding [24], where the number of qubits is linearly related to the number of amino acids in the protein. In particular, for a 2D protein of length N we require $2N - 4$ fully connected qubits, while for a 3D protein we require $3N - 6$.

2D protein size	Number of qubits	3D protein size	Number of qubits
6	7	6	10
7	9	7	13
8	11	8	16
9	13	—	—

Appendix B. Model comparison raw data

As described in the methods section, we employ three independent statistical criteria to select the functional form that best fits the data. These criteria have been widely used in the statistical literature; we provide a brief introduction below:

- Akaike's information criterion (AIC), an information-theoretic metric that balances the complexity of the model (first term on the right-hand side) and the consistency with the probability model (second term on the right-hand side). Models with lower AIC are considered to be of higher quality

$$\text{AIC} = 2k - 2 \ln \mathcal{L}(\theta). \quad (\text{B.1})$$

Table S2. Table of statistical fits for the minimum spectral gap between the ground state and the first excited state, for the dataset of 2D protein problems. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	-3.286 ± 0.053	-1.605 ± 0.012	-0.779 ± 0.005	-0.184 ± 0.005	0.148 ± 0.008	0.361 ± 0.013	-0.752 ± 0.010
α	$e^{\alpha x}$	-1.598 ± 0.028	-0.781 ± 0.007	-0.379 ± 0.003	-0.088 ± 0.002	0.073 ± 0.004	0.173 ± 0.007	-0.366 ± 0.005
α	$e^{\alpha x^2}$	-0.552 ± 0.012	-0.269 ± 0.003	-0.130 ± 0.001	-0.029 ± 0.001	0.025 ± 0.001	0.057 ± 0.003	-0.126 ± 0.002
α	$e^{\alpha x^3}$	-0.181 ± 0.005	-0.088 ± 0.001	-0.042 ± 0.001	-0.009 ± 0.000	0.008 ± 0.001	0.018 ± 0.001	-0.041 ± 0.001
AIC	x^α	2210.30	-2566.22	-9641.00	-9715.92	-4488.14	-2076.36	7421.17
AIC	$e^{\alpha x}$	2381.08	-1849.16	-8455.74	-9587.50	-4474.26	-2016.89	7670.17
AIC	$e^{\alpha x^2}$	2771.33	-540.84	-6273.76	-9297.51	-4429.33	-1884.47	8348.67
AIC	$e^{\alpha x^3}$	3021.55	159.84	-5094.38	-9098.27	-4389.71	-1794.93	8876.53
BIC	x^α	2215.63	-2560.49	-9634.76	-9709.68	-4482.41	-2071.03	7428.80
BIC	$e^{\alpha x}$	2386.40	-1843.43	-8449.50	-9581.26	-4468.53	-2011.56	7677.80
BIC	$e^{\alpha x^2}$	2776.65	-535.11	-6267.52	-9291.27	-4423.60	-1879.14	8356.30
BIC	$e^{\alpha x^3}$	3026.87	165.57	-5088.14	-9092.03	-4383.98	-1789.60	8884.15
MSE	x^α	22.435 48	2.515 34	0.201 13	1.587 45	3.472 87	5.097 03	0.208 60
MSE	$e^{\alpha x}$	20.791 87	2.516 87	0.392 73	1.677 94	3.399 60	4.839 87	0.399 67
MSE	$e^{\alpha x^2}$	18.276 69	2.742 69	0.862 78	1.870 08	3.236 16	4.320 81	0.864 82
MSE	$e^{\alpha x^3}$	16.625 59	2.851 32	1.158 69	1.991 24	3.125 96	3.994 54	1.158 46

Table S3. Table of statistical fits for the minimum spectral gap between the ground state and the first excited state, for the dataset of 3D protein problems. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	-3.665 ± 0.051	-1.575 ± 0.011	-0.850 ± 0.004	0.391 ± 0.012	1.162 ± 0.010	1.397 ± 0.015	-0.403 ± 0.014
α	$e^{\alpha x}$	-2.131 ± 0.029	-0.911 ± 0.007	-0.491 ± 0.002	0.239 ± 0.007	0.680 ± 0.006	0.798 ± 0.010	-0.231 ± 0.008
α	$e^{\alpha x^2}$	-1.121 ± 0.017	-0.473 ± 0.005	-0.254 ± 0.002	0.139 ± 0.004	0.363 ± 0.003	0.403 ± 0.007	-0.117 ± 0.004
α	$e^{\alpha x^3}$	-0.553 ± 0.010	-0.231 ± 0.003	-0.124 ± 0.001	0.072 ± 0.002	0.180 ± 0.002	0.194 ± 0.004	-0.056 ± 0.002
AIC	x^α	1021.64	−4169.39	−12 803.81	−4379.94	−4437.00	−2481.66	5915.36
AIC	$e^{\alpha x}$	992.01	−4058.99	−12 412.42	−4490.25	−4677.59	−2250.96	5936.65
AIC	$e^{\alpha x^2}$	1249.22	−2938.47	−9875.84	−4675.12	−4325.99	−1550.74	6033.55
AIC	$e^{\alpha x^3}$	1510.35	−2213.93	−8464.09	−4729.63	−3775.90	−1173.71	6110.65
BIC	x^α	1026.91	−4163.71	−12 797.62	−4373.75	−4431.33	−2476.39	5922.93
BIC	$e^{\alpha x}$	997.28	−4053.32	−12 406.23	−4484.06	−4671.92	−2245.69	5944.22
BIC	$e^{\alpha x^2}$	1254.49	−2932.80	−9869.66	−4668.93	−4320.32	−1545.48	6041.12
BIC	$e^{\alpha x^3}$	1515.61	−2208.26	−8457.91	−4723.44	−3770.23	−1168.44	6118.22
MSE	x^α	17.942 29	2.314 03	0.337 12	1.068 90	4.141 15	5.477 80	0.001 91
MSE	$e^{\alpha x}$	18.053 14	2.313 51	0.344 74	1.115 57	4.165 63	5.313 65	0.009 12
MSE	$e^{\alpha x^2}$	17.159 02	2.186 61	0.358 62	1.160 65	3.964 42	4.651 49	0.042 26
MSE	$e^{\alpha x^3}$	16.057 52	2.051 04	0.364 10	1.149 90	3.716 57	4.146 21	0.068 87

Table S4. Table of statistical fits for the expected quantum annealing runtime for the worst-case dataset in 2D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	0.031 ± 0.058	0.265 ± 0.026	0.460 ± 0.010	0.795 ± 0.015	1.107 ± 0.027	1.510 ± 0.060	0.672 ± 0.025
α	$e^{\alpha x}$	0.009 ± 0.013	0.059 ± 0.006	0.101 ± 0.002	0.176 ± 0.003	0.245 ± 0.005	0.333 ± 0.013	0.149 ± 0.005
α	$e^{\alpha x^2}$	0.005 ± 0.005	0.022 ± 0.002	0.036 ± 0.002	0.064 ± 0.002	0.089 ± 0.004	0.120 ± 0.008	0.054 ± 0.002
α	$e^{\alpha x^3}$	0.002 ± 0.002	0.007 ± 0.001	0.012 ± 0.001	0.021 ± 0.001	0.030 ± 0.002	0.040 ± 0.003	0.018 ± 0.001
AIC	x^α	–149.70	–287.88	–612.57	–527.89	–276.94	–147.00	–1240.01
AIC	$e^{\alpha x}$	–149.96	–288.86	–592.28	–533.08	–287.01	–145.83	–1243.19
AIC	$e^{\alpha x^2}$	–150.45	–281.14	–491.06	–409.62	–213.34	–112.17	–1186.99
AIC	$e^{\alpha x^3}$	–150.55	–272.31	–442.33	–352.09	–178.27	–93.24	–1129.40
BIC	x^α	–148.01	–285.82	–610.01	–525.35	–274.91	–145.31	–1236.06
BIC	$e^{\alpha x}$	–148.27	–286.80	–589.71	–530.54	–284.98	–144.14	–1239.24
BIC	$e^{\alpha x^2}$	–148.76	–279.08	–488.49	–407.08	–211.31	–110.48	–1183.04
BIC	$e^{\alpha x^3}$	–148.86	–270.25	–439.76	–349.55	–176.24	–91.55	–1125.45
MSE	x^α	0.282 34	0.114 76	0.032 90	0.012 52	0.130 92	0.480 34	0.002 26
MSE	$e^{\alpha x}$	0.272 06	0.112 99	0.032 75	0.011 43	0.130 57	0.475 86	0.000 97
MSE	$e^{\alpha x^2}$	0.260 61	0.127 61	0.056 64	0.035 12	0.146 24	0.458 02	0.025 40
MSE	$e^{\alpha x^3}$	0.261 28	0.146 45	0.083 73	0.062 95	0.164 69	0.445 68	0.054 36

Table S5. Table of statistical fits for the expected quantum annealing runtime for the random dataset in 2D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	0.197 ± 0.034	0.313 ± 0.025	0.497 ± 0.017	0.728 ± 0.015	0.928 ± 0.033	1.294 ± 0.065	0.642 ± 0.020
α	$e^{\alpha x}$	0.046 ± 0.007	0.071 ± 0.005	0.111 ± 0.004	0.161 ± 0.004	0.203 ± 0.009	0.282 ± 0.017	0.142 ± 0.004
α	$e^{\alpha x^2}$	0.018 ± 0.003	0.027 ± 0.002	0.041 ± 0.002	0.058 ± 0.002	0.073 ± 0.005	0.099 ± 0.009	0.052 ± 0.002
α	$e^{\alpha x^3}$	0.006 ± 0.001	0.009 ± 0.001	0.014 ± 0.001	0.020 ± 0.001	0.024 ± 0.002	0.033 ± 0.004	0.017 ± 0.001
AIC	x^α	–186.48	–282.65	–491.03	–514.65	–248.51	–137.01	–1357.19
AIC	$e^{\alpha x}$	–190.18	–290.24	–499.15	–501.84	–231.11	–125.93	–1351.74
AIC	$e^{\alpha x^2}$	–193.61	–289.73	–456.11	–402.68	–186.83	–98.81	–1274.09
AIC	$e^{\alpha x^3}$	–192.53	–281.21	–424.12	–358.44	–166.78	–86.11	–1212.57
BIC	x^α	–184.84	–280.66	–488.53	–512.15	–246.54	–135.37	–1353.30
BIC	$e^{\alpha x}$	–188.54	–288.25	–496.65	–499.34	–229.14	–124.29	–1347.85
BIC	$e^{\alpha x^2}$	–191.97	–287.74	–453.61	–400.18	–184.86	–97.17	–1270.20
BIC	$e^{\alpha x^3}$	–190.89	–279.22	–421.62	–355.94	–164.81	–84.47	–1208.68
MSE	x^α	0.142 49	0.081 53	0.021 62	0.011 97	0.062 27	0.294 82	0.007 06
MSE	$e^{\alpha x}$	0.136 55	0.078 29	0.021 71	0.013 40	0.061 07	0.282 77	0.008 24
MSE	$e^{\alpha x^2}$	0.139 27	0.088 46	0.041 72	0.035 87	0.075 12	0.256 62	0.030 91
MSE	$e^{\alpha x^3}$	0.148 35	0.102 30	0.061 92	0.057 42	0.090 64	0.242 76	0.052 90

Table S6. Table of statistical fits for the expected quantum annealing runtime for the worst-case dataset in 3D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	1.241 ± 0.137	1.344 ± 0.096	1.491 ± 0.055	1.773 ± 0.021	2.002 ± 0.053	2.343 ± 0.171	1.676 ± 0.036
α	$e^{\alpha x}$	0.329 ± 0.029	0.351 ± 0.019	0.384 ± 0.010	0.445 ± 0.006	0.496 ± 0.019	0.571 ± 0.052	0.424 ± 0.009
α	$e^{\alpha x^2}$	0.191 ± 0.010	0.198 ± 0.006	0.211 ± 0.003	0.231 ± 0.008	0.250 ± 0.018	0.274 ± 0.039	0.224 ± 0.006
α	$e^{\alpha x^3}$	0.099 ± 0.004	0.102 ± 0.002	0.107 ± 0.002	0.113 ± 0.006	0.120 ± 0.011	0.127 ± 0.023	0.111 ± 0.003
AIC	x^α	−83.43	−137.51	−276.34	−412.16	−189.25	−70.28	−937.74
AIC	$e^{\alpha x}$	−93.59	−156.59	−324.99	−391.41	−158.26	−59.27	−961.33
AIC	$e^{\alpha x^2}$	−122.89	−212.15	−408.62	−258.19	−109.09	−40.07	−874.59
AIC	$e^{\alpha x^3}$	−138.75	−227.55	−343.24	−213.18	−90.80	−32.14	−785.15
BIC	x^α	−82.03	−135.73	−274.04	−409.87	−187.46	−68.88	−934.05
BIC	$e^{\alpha x}$	−92.19	−154.81	−322.69	−389.12	−156.47	−57.87	−957.64
BIC	$e^{\alpha x^2}$	−121.49	−210.37	−406.31	−255.90	−107.31	−38.67	−870.91
BIC	$e^{\alpha x^3}$	−137.35	−225.77	−340.93	−210.89	−89.02	−30.74	−781.46
MSE	x^α	0.068 99	0.044 07	0.020 23	0.013 00	0.044 34	0.152 89	0.009 75
MSE	$e^{\alpha x}$	0.045 70	0.026 94	0.008 00	0.002 21	0.025 62	0.107 06	0.000 02
MSE	$e^{\alpha x^2}$	0.060 41	0.052 59	0.043 58	0.041 14	0.050 80	0.081 94	0.040 43
MSE	$e^{\alpha x^3}$	0.106 48	0.102 66	0.098 07	0.097 14	0.101 78	0.114 01	0.096 81

Table S7. Table of statistical fits for the expected quantum annealing runtime for the random dataset in 3D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	1.644 ± 0.122	1.733 ± 0.091	1.816 ± 0.060	1.942 ± 0.054	2.200 ± 0.035	2.815 ± 0.175	1.978 ± 0.038
α	$e^{\alpha x}$	0.429 ± 0.023	0.448 ± 0.016	0.469 ± 0.010	0.499 ± 0.008	0.554 ± 0.007	0.688 ± 0.055	0.504 ± 0.008
α	$e^{\alpha x^2}$	0.241 ± 0.004	0.248 ± 0.002	0.258 ± 0.002	0.272 ± 0.003	0.290 ± 0.012	0.334 ± 0.043	0.271 ± 0.005
α	$e^{\alpha x^3}$	0.123 ± 0.002	0.126 ± 0.002	0.130 ± 0.003	0.136 ± 0.004	0.143 ± 0.009	0.156 ± 0.025	0.135 ± 0.003
AIC	x^α	−90.49	−142.04	−256.61	−272.30	−221.56	−69.03	−903.10
AIC	$e^{\alpha x}$	−107.31	−170.90	−319.80	−349.58	−236.72	−55.79	−975.92
AIC	$e^{\alpha x^2}$	−170.86	−283.27	−491.91	−395.21	−142.57	−33.77	−920.09
AIC	$e^{\alpha x^3}$	−166.46	−230.03	−325.45	−283.26	−112.39	−24.94	−803.86
BIC	x^α	−89.09	−140.26	−254.34	−270.02	−219.80	−67.63	−899.42
BIC	$e^{\alpha x}$	−105.91	−169.11	−317.52	−347.30	−234.96	−54.39	−972.24
BIC	$e^{\alpha x^2}$	−169.46	−281.48	−489.63	−392.93	−140.81	−32.37	−916.41
BIC	$e^{\alpha x^3}$	−165.05	−228.24	−323.18	−280.98	−110.63	−23.54	−800.19
MSE	x^α	0.070 51	0.054 30	0.043 81	0.036 17	0.052 08	0.260 86	0.035 83
MSE	$e^{\alpha x}$	0.034 39	0.021 62	0.012 38	0.006 23	0.018 76	0.175 19	0.006 09
MSE	$e^{\alpha x^2}$	0.043 46	0.036 99	0.031 32	0.028 34	0.034 76	0.095 25	0.028 34
MSE	$e^{\alpha x^3}$	0.100 02	0.096 02	0.092 51	0.091 05	0.094 99	0.119 41	0.090 93

Table S8. Table of statistical fits for the expected number of energy evaluations in classical simulated annealing for the worst-case dataset in 2D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	0.342 ± 0.024	0.370 ± 0.022	0.406 ± 0.017	0.441 ± 0.019	0.474 ± 0.027	0.582 ± 0.043	0.431 ± 0.010
α	$e^{\alpha x}$	0.079 ± 0.004	0.086 ± 0.003	0.094 ± 0.003	0.102 ± 0.003	0.110 ± 0.004	0.134 ± 0.007	0.100 ± 0.002
α	$e^{\alpha x^2}$	0.031 ± 0.000	0.034 ± 0.000	0.037 ± 0.000	0.040 ± 0.000	0.043 ± 0.000	0.053 ± 0.001	0.039 ± 0.000
α	$e^{\alpha x^3}$	0.011 ± 0.000	0.012 ± 0.000	0.013 ± 0.000	0.014 ± 0.000	0.015 ± 0.000	0.019 ± 0.000	0.014 ± 0.000
AIC	x^α	–221.09	–302.58	–502.23	–484.59	–281.00	–174.66	–1909.64
AIC	$e^{\alpha x}$	–249.38	–343.60	–575.55	–558.84	–325.25	–199.98	–2139.47
AIC	$e^{\alpha x^2}$	–368.49	–553.71	–956.21	–866.85	–525.90	–270.71	–2608.48
AIC	$e^{\alpha x^3}$	–320.65	–436.13	–692.79	–647.66	–380.99	–253.89	–2428.29
BIC	x^α	–219.40	–300.56	–499.68	–482.06	–278.97	–172.97	–1905.70
BIC	$e^{\alpha x}$	–247.69	–341.57	–573.00	–556.31	–323.22	–198.29	–2135.53
BIC	$e^{\alpha x^2}$	–366.80	–551.68	–953.67	–864.31	–523.87	–269.02	–2604.55
BIC	$e^{\alpha x^3}$	–318.96	–434.11	–690.25	–645.13	–378.96	–252.20	–2424.35
MSE	x^α	0.027 63	0.024 68	0.022 33	0.021 79	0.022 82	0.036 39	0.021 77
MSE	$e^{\alpha x}$	0.016 33	0.012 86	0.010 63	0.010 07	0.011 42	0.026 44	0.010 05
MSE	$e^{\alpha x^2}$	0.006 62	0.002 64	0.000 75	0.000 16	0.001 90	0.018 57	0.000 13
MSE	$e^{\alpha x^3}$	0.008 83	0.004 92	0.003 25	0.002 66	0.004 40	0.021 36	0.002 64

Table S9. Table of statistical fits for the expected number of energy evaluations in classical simulated annealing for the random dataset in 2D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	0.291 ± 0.022	0.330 ± 0.020	0.366 ± 0.017	0.404 ± 0.019	0.450 ± 0.027	0.585 ± 0.046	0.397 ± 0.010
α	$e^{\alpha x}$	0.068 ± 0.003	0.077 ± 0.003	0.085 ± 0.003	0.094 ± 0.003	0.105 ± 0.004	0.136 ± 0.008	0.092 ± 0.002
α	$e^{\alpha x^2}$	0.027 ± 0.000	0.030 ± 0.000	0.034 ± 0.000	0.037 ± 0.000	0.041 ± 0.000	0.054 ± 0.001	0.036 ± 0.000
α	$e^{\alpha x^3}$	0.009 ± 0.000	0.011 ± 0.000	0.012 ± 0.000	0.013 ± 0.000	0.014 ± 0.000	0.019 ± 0.001	0.013 ± 0.000
AIC	x^α	−220.91	−308.49	−481.97	−467.88	−261.99	−163.71	−1819.85
AIC	$e^{\alpha x}$	−247.33	−347.44	−544.89	−532.60	−299.81	−186.41	−2004.83
AIC	$e^{\alpha x^2}$	−349.17	−530.83	−832.72	−839.80	−481.49	−239.03	−2348.93
AIC	$e^{\alpha x^3}$	−312.87	−457.07	−720.89	−688.38	−383.29	−225.58	−2278.73
BIC	x^α	−219.27	−306.49	−479.48	−465.39	−260.04	−162.07	−1815.96
BIC	$e^{\alpha x}$	−245.69	−345.43	−542.41	−530.11	−297.86	−184.77	−2000.94
BIC	$e^{\alpha x^2}$	−347.53	−528.83	−830.23	−837.31	−479.54	−237.39	−2345.04
BIC	$e^{\alpha x^3}$	−311.24	−455.06	−718.40	−685.89	−381.34	−223.94	−2274.85
MSE	x^α	0.028 63	0.023 82	0.021 23	0.020 35	0.021 97	0.043 34	0.020 36
MSE	$e^{\alpha x}$	0.018 60	0.013 47	0.010 70	0.009 84	0.011 78	0.036 07	0.009 83
MSE	$e^{\alpha x^2}$	0.009 37	0.003 95	0.001 04	0.000 24	0.002 62	0.029 61	0.000 22
MSE	$e^{\alpha x^3}$	0.010 56	0.005 20	0.002 34	0.001 58	0.004 03	0.030 65	0.001 56

Table S10. Table of statistical fits for the expected number of energy evaluations in classical simulated annealing for the worst-case dataset in 3D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	0.331 ± 0.024	0.359 ± 0.020	0.395 ± 0.017	0.439 ± 0.019	0.464 ± 0.026	0.493 ± 0.032	0.414 ± 0.009
α	$e^{\alpha x}$	0.086 ± 0.004	0.093 ± 0.004	0.103 ± 0.003	0.114 ± 0.004	0.121 ± 0.005	0.128 ± 0.006	0.108 ± 0.002
α	$e^{\alpha x^2}$	0.048 ± 0.001	0.052 ± 0.000	0.057 ± 0.000	0.064 ± 0.000	0.068 ± 0.000	0.071 ± 0.000	0.060 ± 0.000
α	$e^{\alpha x^3}$	0.025 ± 0.000	0.027 ± 0.000	0.029 ± 0.000	0.032 ± 0.000	0.034 ± 0.000	0.036 ± 0.001	0.031 ± 0.000
AIC	x^α	–189.03	–266.65	–437.52	–414.11	–250.89	–164.34	–1697.35
AIC	$e^{\alpha x}$	–207.15	–294.11	–483.51	–459.26	–278.90	–184.10	–1856.48
AIC	$e^{\alpha x^2}$	–295.05	–454.12	–717.37	–714.97	–444.30	–310.68	–2316.48
AIC	$e^{\alpha x^3}$	–276.28	–384.31	–622.73	–613.55	–373.74	–230.00	–2217.73
BIC	x^α	–187.63	–264.89	–435.24	–411.84	–249.11	–162.98	–1693.68
BIC	$e^{\alpha x}$	–205.74	–292.35	–481.23	–457.00	–277.12	–182.73	–1852.81
BIC	$e^{\alpha x^2}$	–293.65	–452.36	–715.09	–712.71	–442.52	–309.31	–2312.81
BIC	$e^{\alpha x^3}$	–274.88	–382.54	–620.45	–611.29	–371.95	–228.64	–2214.06
MSE	x^α	0.009 43	0.008 24	0.007 40	0.007 52	0.008 13	0.009 33	0.007 30
MSE	$e^{\alpha x}$	0.006 07	0.004 84	0.003 96	0.004 05	0.004 73	0.005 95	0.003 84
MSE	$e^{\alpha x^2}$	0.002 39	0.001 13	0.000 22	0.000 27	0.001 02	0.002 21	0.000 07
MSE	$e^{\alpha x^3}$	0.002 74	0.001 52	0.000 62	0.000 66	0.001 42	0.002 53	0.000 46

Table S11. Table of statistical fits for the expected number of energy evaluations in classical simulated annealing for the random dataset in 3D. In bold, we indicate the model preferred by a particular model selection criterion (see section 2).

Parameter	Model	0%–10%	10%–25%	25%–50%	50%–75%	75%–90%	90%–100%	Full
α	x^α	0.307 ± 0.022	0.350 ± 0.020	0.393 ± 0.018	0.434 ± 0.018	0.471 ± 0.025	0.511 ± 0.031	0.412 ± 0.009
α	$e^{\alpha x}$	0.080 ± 0.004	0.091 ± 0.004	0.102 ± 0.003	0.113 ± 0.003	0.122 ± 0.005	0.132 ± 0.005	0.107 ± 0.002
α	$e^{\alpha x^2}$	0.045 ± 0.001	0.051 ± 0.000	0.057 ± 0.000	0.063 ± 0.000	0.068 ± 0.000	0.074 ± 0.000	0.060 ± 0.001
α	$e^{\alpha x^3}$	0.023 ± 0.000	0.026 ± 0.000	0.029 ± 0.000	0.032 ± 0.000	0.035 ± 0.001	0.037 ± 0.001	0.030 ± 0.000
AIC	x^α	−193.22	−279.36	−438.15	−432.59	−254.52	−172.20	−1732.82
AIC	$e^{\alpha x}$	−211.13	−306.42	−482.39	−480.03	−284.26	−193.96	−1887.22
AIC	$e^{\alpha x^2}$	−293.47	−436.46	−705.67	−746.36	−458.90	−326.59	−2286.23
AIC	$e^{\alpha x^3}$	−278.35	−410.57	−663.73	−621.05	−354.01	−228.19	−2204.57
BIC	x^α	−191.81	−277.55	−435.86	−430.30	−252.74	−170.80	−1729.13
BIC	$e^{\alpha x}$	−209.72	−304.61	−480.10	−477.74	−282.48	−192.56	−1883.53
BIC	$e^{\alpha x^2}$	−292.07	−434.65	−703.38	−744.07	−457.12	−325.18	−2282.54
BIC	$e^{\alpha x^3}$	−276.95	−408.77	−661.44	−618.76	−352.23	−226.79	−2200.88
MSE	x^α	0.010 53	0.008 30	0.007 23	0.007 32	0.008 30	0.010 37	0.007 14
MSE	$e^{\alpha x}$	0.007 31	0.004 95	0.003 84	0.003 91	0.004 95	0.007 05	0.003 74
MSE	$e^{\alpha x^2}$	0.003 78	0.001 30	0.000 17	0.000 21	0.001 28	0.003 33	0.000 06
MSE	$e^{\alpha x^3}$	0.004 13	0.001 68	0.000 57	0.000 62	0.001 65	0.003 59	0.000 47

- BIC, an information-theoretic metric similar to AIC, but that also considers the size of the dataset it has been parametrized on. Models with lower BIC are considered to be of higher quality

$$\text{BIC} = k \ln N - 2 \ln \mathcal{L}(\theta). \quad (\text{B.2})$$

- Mean-squared error (MSE), a classical metric that considers the average error of the model at predicting the points it has been fitted on. Models with lower MSE are better at predicting a dataset, and thus are considered to be of higher quality

$$\text{MSE} = \frac{1}{N} \sum_{r=0}^N [y_r - f(x_r)]^2. \quad (\text{B.3})$$

In these equations, N is the number of data points, $f(x)$ is the fitting function, θ is the vector of adjustable parameters, k is the dimension of θ and $\mathcal{L}(\theta)$ is the likelihood function. The calculation of the likelihood function requires a model of the underlying probability distribution that generates the data. In this work, as is common in statistical analysis, we have assumed that the data is generated by a deterministic process that is corrupted by Gaussian noise. In other words, a data point is generated as:

$$y = f(x) + \epsilon. \quad (\text{B.4})$$

Here, ϵ is a normally-distributed random variable with zero mean and a non-zero standard deviation σ that is obtained from the data (i.e. $\epsilon \sim \mathcal{N}(0, \sigma)$). Therefore, the likelihood function for a given model $f(x)$ is:

$$\mathcal{L}(\theta) = (2\pi\sigma^2)^{N/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{r=1}^N (y_r - f(x_r))^2\right). \quad (\text{B.5})$$

In this equation, $f(x_r)$ also depends on additional parameters which are include in θ alongside the standard deviation, e.g. the slope and the intercept if the model is linear. The parameters of the model are found by numerical maximization of the likelihood, and used to compute AIC, BIC and MSE.

ORCID iDs

Carlos Outeiral  <https://orcid.org/0000-0003-1408-5554>
 Garrett M Morris  <https://orcid.org/0000-0003-1731-8405>
 Jiye Shi  <https://orcid.org/0000-0002-9628-8680>
 Simon C Benjamin  <https://orcid.org/0000-0002-7766-5348>
 Charlotte M Deane  <https://orcid.org/0000-0003-1388-2252>

References

- [1] Nelson D L, Lehninger A L and Cox M M 2008 *Lehninger's Principles of Biochemistry* (London: Macmillan)
- [2] Jones S and Thornton J M 1996 Principles of protein–protein interactions *Proc. Natl Acad. Sci.* **93** 13–20
- [3] Jhoti H and Leach A R 2007 *Structure-Based Drug Discovery* 1st edn (Heidelberg: Springer)
- [4] Zeymer C and Hilvert D 2018 Directed evolution of protein catalysts *Annu. Rev. Biochem.* **87** 131–57
- [5] Blundell T L 2017 Protein crystallography and drug discovery: recollections of knowledge exchange between academia and industry *Int. Union Crystallogr. J.* **4** 308–21
- [6] Renaud J-P, Chari A, Ciferri C, Liu W-t, Rémigy H-W, Stark H and Wiesmann C 2018 Cryo-EM in drug discovery: achievements, limitations and prospects *Nat. Rev. Drug Discovery* **17** 471–92
- [7] UniProt Consortium 2019 UniProt: a worldwide hub of protein knowledge *Nucleic Acids Res.* **47** D506–15
- [8] Berman H M, Bourne E, Westbrook J and Zarddecki C 2003 The protein data bank *Protein Structure* (Boca Raton, FL: CRC Press) pp 394–410
- [9] Zvelebil M J and Baum J O 2007 *Understanding Bioinformatics* (New York: Garland Science)
- [10] Kendrew J C, Bodo G, Dintzis H M, Parrish R G, Wyckoff H and Phillips D C 1958 A three-dimensional model of the myoglobin molecule obtained by x-ray analysis *Nature* **181** 662–6
- [11] Dill K A and MacCallum J L 2012 The protein-folding problem, 50 years on *Science* **338** 1042–6
- [12] Abriata L A, Tamò G E and Dal Peraro M 2019 A further leap of improvement in tertiary structure prediction in casp13 prompts new routes for future assessments *Proteins* **87** 1100–12
- [13] Jumper J *et al* 2021 Highly accurate structure prediction with AlphaFold *Nature* **596** 583–9
- [14] Baek M *et al* 2021 Accurate prediction of protein structures and interactions using a three-track neural network *Science* **373** 871–6
- [15] Outeiral C, Strahm M, Shi J, Morris G M, Benjamin S C and Deane C M 2021 *The Prospects of Quantum Computing in Computational Molecular Biology* (New York: Wiley) p e1481
- [16] Farhi E, Goldstone J, Gutmann S and Sipser M 2000 Quantum computation by adiabatic evolution (arXiv:quant-ph/0001106)
- [17] Born M and Fock V 1928 Beweis des adiabatsatzes *Z. Phys.* **51** 165–80

- [18] Lau K F and Dill K A 1989 A lattice statistical mechanics model of the conformational and sequence spaces of proteins *Macromolecules* **22** 3986–97
- [19] Skolnick J, Kolinski A, Kihara D, Betancourt M, Rotkiewicz P and Boniecki M 2001 *Ab initio* protein structure prediction via a combination of threading, lattice folding, clustering, and structure refinement *Proteins* **45** 149–56
- [20] Hoque T, Chetty M and Sattar A 2009 Extended HP model for protein structure prediction *J. Comput. Biol.* **16** 85–103
- [21] Hart W E and Istrail S 1997 Robust proofs of NP-hardness for protein folding: general lattices and energy potentials *J. Comput. Biol.* **4** 1–22
- [22] Berger B and Leighton T 1998 Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete *J. Comput. Biol.* **5** 27–40
- [23] Perdomo A, Truncic C, Tubert-Brohman I, Rose G and Aspuru-Guzik A 2008 Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models *Phys. Rev. A* **78** 012320
- [24] Babbush R, Perdomo-Ortiz A, O’Gorman B, Macready W and Aspuru-Guzik A 2012 Construction of energy functions for lattice heteropolymer models: a case study in constraint satisfaction programming and adiabatic quantum optimization (arXiv:1211.3422)
- [25] Babej T, Ing C and Fingerhuth M 2018 Coarse-grained lattice protein folding on a quantum annealer (arXiv:1811.00713)
- [26] Perdomo-Ortiz A, Dickson N, Drew-Brook M, Rose G and Aspuru-Guzik A 2012 Finding low-energy conformations of lattice protein models by quantum annealing *Sci. Rep.* **2** 571
- [27] Robert A, Barkoutsos K, Woerner S and Tavernelli I 2019 Resource-efficient quantum algorithm for protein folding (arXiv:1908.02163)
- [28] Mulligan V K, Melo H, Merritt H I et al 2019 Designing peptides on a quantum computer *bioRxiv Preprint* <https://doi.org/10.1101/752485> (retrieved 18 May 2020)
- [29] Casares P A M, Campos R and Martin-Delgado M A 2021 QFold: quantum walks and deep learning to solve protein folding (arXiv:2101.10279)
- [30] Brocchieri L and Karlin S 2005 Protein length in eukaryotic and prokaryotic proteomes *Nucleic Acids Res.* **33** 3390–400
- [31] Wang Y et al 2020 Therapeutic target database 2020: enriched resource for facilitating research and early development of targeted therapeutics *Nucleic Acids Res.* **48** D1031–41
- [32] Jordan S P, Farhi E and Shor W 2006 Error-correcting codes for adiabatic quantum computation *Phys. Rev. A* **74** 052322
- [33] Lidar D A 2008 Towards fault tolerant adiabatic quantum computation *Phys. Rev. Lett.* **100** 160506
- [34] Pudenz K L, Albash T and Lidar D A 2014 Error-corrected quantum annealing with hundreds of qubits *Nat. Commun.* **5** 3243
- [35] Jiang Z and Rieffel E G 2017 Non-commuting two-local Hamiltonians for quantum error suppression *Quantum Inf. Process.* **16** 89
- [36] Marvian M and Lidar D A 2017 Error suppression for Hamiltonian-based quantum computation using subsystem codes *Phys. Rev. Lett.* **118** 030504
- [37] Lloyd S 1996 Universal quantum simulators *Science* **273** 1073–8
- [38] D-Wave Systems 2020 The D-wave advantage system: an overview https://dwavesys.com/sites/default/files/14-1049A-A_The_D-Wave_Advantage_System_An_Overview_0.pdf (retrieved 20 April 2021)
- [39] Dill K A, Bromberg S, Yue K, Chan H S, Ftebig K M, Yee D P and Thomas P D 1995 Principles of protein folding—a perspective from simple exact models *Protein Sci.* **4** 561–602
- [40] Albash T and Lidar D A 2018 Adiabatic quantum computation *Rev. Mod. Phys.* **90** 015002
- [41] van Dam W, Mosca M and Vazirani U 2001 How powerful is adiabatic quantum computation? *Proc. 42nd IEEE Symp. Foundations of Computer Science* (Piscataway, NJ: IEEE) pp 279–87
- [42] van Dam W and Vazirani U 2001 Limits on quantum adiabatic optimization (unpublished) <https://people.eecs.berkeley.edu/vazirani/pubs/qao.pdf> (retrieved 10 March 2020)
- [43] Reichardt B W 2004 The quantum adiabatic optimization algorithm and local minima *Proc. 36th Annual ACM Symp. Theory of Computing* (New York: ACM) pp 502–10
- [44] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem *Science* **292** 472–5
- [45] Young A P, Knysh S and Smelyanskiy V N 2008 Size dependence of the minimum excitation gap in the quantum adiabatic algorithm *Phys. Rev. Lett.* **101** 170503
- [46] Young A, Knysh S and Smelyanskiy V 2010 First-order phase transition in the quantum adiabatic algorithm *Phys. Rev. Lett.* **104** 020502
- [47] Ronnow T F, Wang Z, Job J, Boixo S, Isakov S V, Wecker D, Martinis J M, Lidar D A and Troyer M 2014 Defining and detecting quantum speedup *Science* **345** 420–4
- [48] Hormozi L, Brown E W, Carleo G and Troyer M 2017 Nonstoquastic Hamiltonians and quantum annealing of an Ising spin glass *Phys. Rev. B* **95** 184416
- [49] Albash T 2019 Role of nonstoquastic catalysts in quantum adiabatic optimization *Phys. Rev. A* **99** 042334
- [50] Boixo S et al 2016 Computational multiqubit tunnelling in programmable quantum annealers *Nat. Commun.* **7** 10327
- [51] Crosson E, Albash T, Hen I and Young A P 2020 De-signing Hamiltonians for quantum adiabatic optimization *Quantum* **4** 334
- [52] Roland J and Cerf N J 2002 Quantum search by local adiabatic evolution *Phys. Rev. A* **65** 042308
- [53] Rezakhanlou A T, Kuo W-J, Hama A, Lidar D A and Zanardi P 2009 Quantum adiabatic brachistochrone *Phys. Rev. Lett.* **103** 080502
- [54] Jarret M, Lackey B, Liu A and Wan K 2018 Quantum adiabatic optimization without heuristics (arXiv:1810.04686)
- [55] Matsuura S, Buck S, Senicourt V and Zaribafian A 2020 Variationally scheduled quantum simulation (arXiv:2003.09913)
- [56] Shahriari B, Swersky K, Wang Z, Adams R P and De Freitas N 2015 Taking the human out of the loop: a review of Bayesian optimization *Proc. IEEE* **104** 148–75
- [57] The GPyOpt Authors 2016 GPyOpt: a Bayesian optimization framework in Python <http://github.com/SheffieldML/GPyOpt>
- [58] Katsuda H and Nishimori H 2013 Nonadiabatic quantum annealing for one-dimensional transverse-field Ising model (arXiv:1303.6045)
- [59] Muthukrishnan S, Albash T and Lidar D A 2016 Tunneling and speedup in quantum optimization for permutation-symmetric problems *Phys. Rev. X* **6** 031010
- [60] Crosson E J and Lidar D A 2021 Prospects for quantum enhancement with diabatic quantum annealing *Nat. Rev. Phys.* **3** 466–89
- [61] Crosson E J, Farhi E, Lin C Y, Lin H and Shor P 2014 Different strategies for optimization using the quantum adiabatic algorithm (arXiv:1401.7320)
- [62] Das R and Baker D 2008 Macromolecular modeling with Rosetta *Annu. Rev. Biochem.* **77** 363–82

- [63] Galassi *et al* 2002 *GNU Scientific Library Reference Manual* 2nd edn
- [64] Albash T and Lidar D A 2018 Demonstration of a scaling advantage for a quantum annealer over simulated annealing *Phys. Rev. X* **8** 031016
- [65] Rajasekaran S 1990 On the convergence time of simulated annealing *Technical Reports* CIS p 356
- [66] Pearl F M G, Bennett C, Bray J E, Harrison A P, Martin N, Shepherd A, Sillitoe I, Thornton J and Orengo C A 2003 The CATH database: an extended protein family resource for structural and functional genomics *Nucleic Acids Res.* **31** 452–5
- [67] Chen M-C *et al* 2020 Demonstration of adiabatic variational quantum computing with a superconducting quantum coprocessor *Phys. Rev. Lett.* **125** 180501
- [68] Miyazawa S and Jernigan R L 1985 Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation *Macromolecules* **18** 534–52
- [69] D-Wave Systems 2019 D-wave problem-solving handbook https://docs.dwavesys.com/docs/latest/doc_handbook.html (retrieved 10 March 2020)
- [70] Čertík O 2012 SymEngine <https://github.com/symengine/symengine/> (retrieved 1 April 2019)
- [71] Hernandez V, Roman J E and Vidal V 2005 SLEPc *ACM Trans. Math. Softw.* **31** 351–62
- [72] Abhyankar S, Brown J, Constantinescu E M, Ghosh D, Smith B F and Zhang H 2018 PETSc/TS: a modern scalable ODE/DAE solver library (arXiv:1806.01437)
- [73] Balay S *et al* 2019 *PETSc Users Manual*
- [74] Farhi E, Goldstone J and Gutmann S 2000 A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability (arXiv:quant-ph/0007071)
- [75] Balay S *et al* 2016 PETSc users manual ANL-95/11 - Revision 3.7 (Argonne National Laboratory)
- [76] D-Wave Systems 2019 Technical description of the D-wave quantum processing unit https://docs.dwavesys.com/docs/latest/doc_qpu.html (retrieved 10 March 2020)
- [77] Schrödinger L L C 2015 *The PyMOL Molecular Graphics System* version 2.4.0a0 Zenodo <https://doi.org/10.5281/zenodo.11813>