I have read the submission by Khatami, Mendes, Wiebe, and Kim titled "Gate-based quantum computing for protein design". In this manuscript, the authors examine the optimization problem of finding an amino acid sequence corresponding to the minimum value of an energy or scoring function, a problem that is solved during the protein design process. This problem scales poorly with the number of amino acid positions being designed in the protein (which the authors represent as s), or with the number of amino acid types considered at each position (which the authors represent as A), and this represents a serious challenge for design algorithms on classical computers. The authors show that this problem can be solved on gate-based quantum computers using Grover's algorithm, a method for searching a large solution space using quadratically fewer samples than would be required classically. Using quantum simulators run on classical hardware, they show that they are able to produce solution states with high probability (quantum computing is probabilistic computing) for small problems, and that noise hinders but does not eliminate the functionality of their approach. They also show enrichment of the solution state for a very small problem on an actual, current-generation IBM gate-based quantum computer.

The authors have identified a sensible problem for which quantum computers could plausibly offer an advantage over classical computers one day. Their approach appears to be sound. The manuscript reads well, like one that has had quite a bit of thought go into revisions already. I am largely satisfied that this manuscript is suitable for publication. Below, I list one concern that I hope the authors can address or rebut, as well as a small number of minor points related to presentation which I think the authors will be able to address easily.

Major concern:

1. The PLoS author guidelines state: "PLOS expects researchers to share software and scripts needed for the work. If this cannot be made publicly available (e.g. due to licenses), the simulation method should be provided in sufficient detail so the results can, in principle, be reproduced using publicly available software." I think that the authors have made a good faith effort to describe their methods in a good deal of detail. I am *moderately* confident that a person with sufficient domain expertise could reproduce what was done here, though the complexity of the approach does mean that it would be hard to say for certain whether any differences that that person saw between their results and the authors' results was due to errors in reimplementing the method versus actual failure of the method to reproduce the result. I would be more comfortable if the authors were willing to make their Qiskit code for simulating these circuits (or, at least, for simulating *a particular* problem) publicly available. I won't insist on this, since the description in the supplement is quite detailed, but it would make me more comfortable for the sake of full reproducibility.

Minor points:

1. On lines 70-72, the statement of the main characteristic of *NP*-complete problems could be a bit more precise and a bit clearer. The main characteristic of these problems is that the time or resources needed to find a solution *scales* poorly, not that they are intrinsically costly across all problem sizes. (There are *NP*-complete problems, like the travelling salesman problem, that are solved routinely for small problem instances, such as getting delivery vehicles to customers.) Similarly, on line 73, the proposed answer need not be scored *easily*, but in *polynomial time*. (There are polynomial-time tasks, such as the two-body molecular docking problem, that are extremely costly.) I mention this mainly because this is often a misconception that I have to dispel with students, that *NP*-completeness intrinsically means computational intractability or that polynomial-time scaling intrinsically means computational tractability *for all problem sizes*. It's actually something that's kind of interesting about the protein design problem: it's an *NP*-complete problem with interesting real-world applications across a range of scales, ranging from extremely tractable (allowing classical and quantum solvers to be compared on these problems) to challenging (meaning that there may be problems for which there's a slight quantum advantage) to utterly intractable (meaning that a good quantum solver would have problems that it was uniquely suited to do).

2. In the caption to Figure 1, there's a sentence that seems to be incomplete. (As an aside, I like the detail in figure 1 – particularly panel B. Explanations like this in plan language make these algorithms much easier to understand.)

3. On lines 137-138, “is not advantageous” is a little bit confusing. Maybe it could be made clearer that the sign flip does not, at this point, alter the relative probabilities of states (something that will be altered by the subsequent diffuser step)?

4. Figure 2 is pretty clear. A suggestion that the authors might consider is adding a dashed line for each pairwise interaction, maybe in the same dark red colour as the listed interactions. (The lists currently refer to a feature that the reader can infer, but which is invisible. I wonder if it might be a little easier on the reader just to make it more visible.)

5. On lines 194 and 201, “number of residues” is a bit ambiguous: it could be the number of designable positions, *s*, or the number of amino acid possibilities at each position, *A*. I think the latter is meant. Perhaps using the nomenclature established earlier (*A*) or indicating that this is number of residue *types* might make this clearer.

6. A small singular/plural typo on line 228: “simulate circuits with this many qubit*s*.”

7. Figures 5 and 6 show good results. The one minor criticism I have is of the x-axis label in figure 5 and the x- and y-axis labels in figure 6: descriptive labels such as “Max number of iterations (R_{max})” are easier to follow than the symbol alone (especially when the axis label is otherwise an expression of several symbols, such as R/R_{max}. It’s not always intuitive to a reader why a particular expression is on an axis or what the expected relationship between an abstract expression on one axis and an abstract expression on another should be.)

8. It’s worth being aware that a circuit depth of 20 (line 415) sounds a little bit low as an upper limit. My experience has been that circuit depth limits of 40 or 50 seem a little more realistic, currently. This is very minor, though: either way, the authors’ point that only very short circuits are currently possible is a valid one.

9. On line 444, “Coulomb” should be capitalized.

10. I’d be a little bit careful with the claim on lines 446-448, that the distance reciprocal and decimal number approach used here could be used in lattice model folding studies. If the claim is that one could keep the lattice model, but use pairwise interaction energies represented as decimal numbers, possibly taken from classically precomputed lookup tables, this sounds reasonable. If the claim is that this provides a route to non-integer positions for the beads on the chain (i.e. a means of moving past lattice models to continuous-space models), this is harder to envisage, and would require means of dealing with very non-pairwise effects (such as the fact that the position of residue 3 depends on both residue 1 and residue 2). Maybe rephrasing this to make it clear that the former is meant would be a good idea.

11. In the first paragraph of the methods, it might be good to make it even more explicit what’s done classically and what’s done in the quantum circuit. For instance, “First, the *classically-computed* values in the energy tables are introduced to the oracle… Next the energy values for every pair-wise interaction in the structure are summed in the quantum circuit to find the total energy…”

12. The point made on lines 549-555, about the circuit depth needed for a real problem compared to what’s currently possible, is an important one, and one that many readers will be looking for in the conclusions. It’s likely worth including in the discussion in more detail, not just in the methods. I suspect that many readers will be looking for this as a bottom line to the story: they’ll be wondering, “at what point is the quantum hardware likely to be useful for real design problems”? (Note: I do NOT consider the fact that real problems will require larger, more robust hardware than is currently available to be a shortcoming of this work.)