**Summaries**

Reviewer: 1   
Summary: The author suggest a combination of semi-automated experimentation and machine learning to optimize RBS sequences in an iterative fashion (design-build-test-learn cycles).   
  
Reviewer: 2   
Summary: The authors provide an approach for ML-guided experimental design of bacterial RBSs, with the aim of improving protein expression. They used Gaussian Process Regression to map genotype (core RBS sequence) to phenotype (translation initiation rate, TIR, measured through GFP fluorescence) and batch Upper Confidence Bound-based approach for generating recommendations of sequence designs to be tested in vivo. Four DBTL cycles, which integrated the ML approach with automation and high-throughput data generation, were performed, with the final one achieving up to 34% higher TIR than the benchmark sequence.   
  
Reviewer: 3   
Summary: The manuscript from Zhang and colleagues describes the application of active learning in order to predict and recommend best RBS sequences for an iterative DBTL cycle. The authors showed that 4 experimental iterations involving a 10% of the full design space was enough in order to identify RBS combinations with high performance.

**Strengths**  
  
Reviewer: 1   
Strengths: The work is a combination of previously published tools and methods. It is difficult to point to unique assets and/or new findings/tools/methods.

**NOTE: Pieces are not unique, but put together they are, this plays into this trend (cite a few papers). We have also received two other positive reviews.**  
Reviewer: 2   
Strengths: This is an important demonstration of the potential that machine learning has as a tool for designing RBSs. The paper is of high quality, technically solid, well written and mostly clear. I believe that the contribution of this work is valuable and is worth being published in Synthetic Biology.   
  
Reviewer: 3   
Strengths: Reading this study is really enjoyable and shows both a machine learning and synthetic biology mind.   
  
Results are positive and convincing, i.e., better RBS were successfully designed through the described approach.   
  
The authors made use of several iterations of a full DBTL cycle in a biofoundry, involving LEARN stages.   
  
**Major Comments**

Reviewer: 1   
More review: - In general, the descriptions in the manuscript lack critical detail (both for experimental and computational works), which makes it impossible to reconstruct or repeat what the authors precisely did. The figure captions are not sufficient to understand the content of the figure (see for instance Fig. S7). The methods section is very much underexplained given the complexity of applied methods (automation etc.).   
  
- It is unclear why the authors choose very strong RBSs as a design goal. This challenge has been amply solved in prior studies and in most optimization cases (e.g. in metabolic pathway optimization) the goal is certainly not to have the strongest RBSs (here fine tuning is key to avoid metabolic burden). Also the claim that the authors have found RBSs “exceeding the currently known strong RBSs” (quote from abstract) is unjustified and also meaningless to a large degree. RBSs are know to be highly dependent on the context (CDS and upstream 5’-UTR). The authors have at most found RBSs that are very strong in their specific context. A comparison to other studies is questionable, there are simply no strong standard RBSs. A similar argument may be made for the claim that the authors have identified “an extensive and reliable library of novel RBSs“ (p.2, line 42-43): It is unclear what the authors mean by “extensive” and “reliable” here, and what the “novelty” of the RBS is. Designing novel RBS is by no means a challenge or an achievement given the extremely large sequence space (any RBS that is designed at random has an extremely high chance to never have been tested before).

**Due to the complex nature of this comment, we have tried to break down our answer into points addressing specific questions:**

**a) We have chosen maximising the TIR as our design goal to showcase how the exploitative nature of the Bandits algorithm can be used to optimise a specific quality of a genetic part, in our case that was the TIR of the RBS. We agree with the sentiment that optimisation will not usually mean maximisation in more complex metabolic engineering uses, but it is an appropriate function for a relatively simple case like ours. As mentioned throughout the text, our algorithm can be used to find a range of TIRs, we are reiterating this point now with the new paragraph in the discussion section:**

**“COPY FROM TEXT”**

**b) We also agree that the TIR of an RBS will be context dependent. In this study we strived to create easily reproducible conditions, which have been commonly used to study and characterizea number of types of genetic elements, including, but not limited to, as part of the iGEM competition. At no point in our work we make a suggestion that a standard, high strength RBS exists, we merely say that: “This [our benchmark RBS] sequence is known to have a very high TIR and is present in the pBb series of plasmids”. We have modified this sentence to say: “This sequence is known to have a very high TIR in a similar context (expressing fluorescent protein) and is present in the pBb series of plasmids” to further underline that point. We have also updated the abstract accordingly.**

**c) There is currently a very small selection of RBS libraries, including, but not limited to, the iGEM public collections. However, using them is often problematic, as the reported characteristics numbers for these RBSs have been acquired using different and sometimes unreliable methods. In contrast, our library has been created using standarised automated methods which contributed to relatively low coefficients of variation (‘reliable’). Additionally, compared to other widely known libraries which usually have up to 100 entries, ours has 445 RBSs (‘extensive’). We have updated the first paragraph of the ‘BUILD & TEST: Characteristics of the tested sequences’ section to reflect this.**

**d) The authors do not agree that designing novel RBS is a trivial task – as long as finding new RBSs is not hard, finding ones with the desired qualities is (see our answer to the comment below).**  
- the “benchmark RBS” that is used/treated as a standard for very strong RBSs is found also for instance in the pET plasmid series. This RBS may in many cases be quite strong, but its use as a benchmark for very strong RBSs is questionable due to context dependence. It is by no means a surprise or a significant achievement that the authors find stronger sequences than this one amongst 450 tested sequences.

**Out of 450 tested sequences almost 40% have been created by some kind of sequence randomisation. Out of these randomised sequences, only a few got close to benchmark sequences TIR and were still 20% weaker than it is (Figure 2). In fact, these 80% TIR ratio sequences were created by randomizing the sequence outside of the core RBS region, which were statistically shown not to be significantly impactful on the TIR. More representative would be sequences from other random groups, from which the best one achieved only about 65%** **of benchmark TIR. Contrasted with this, non-randomized, bandit driven design batch shown much better results, with RBSs getting close to benchmark performance and even exceeding it. The authors believe that this is evidence enough of finding RBS stronger than our benchmark (which, as stated before, is known to be strong in similar context) to be a significant achievement (contrasted with simply randomly mutating the sequence), but definitely not a surprise – the goal of this work was to shown that this is achievable using ML.**

**To point this fact to the readers, we have added this sentence at the end of the paragraph describing our results for randomly generated sequences:**

**“These results show that it is hard to generate a strong RBS sequence by random mutations, when the tested data set is relatively small.”**  
  
-  Testing “just” 450 different variants to obtain strong RBSs seems like a lot of effort, especially in view of the fact that there are many predictions algorithms around for RBSs that can be used to design RBSs with quite high confidence. If that goal was to simply increase/maximize the expression level, one could also simply swap the promoter from pLac to pT7, which is much stronger, or by changing the copy number of the used plasmid (unfortunately there is no reference for the plasmid “pBbB6c-GFP” the authors used in this study, but it seems to be a low copy plasmid originally from the Keasling lab).

**The authors, in their own and others experience, did not find the currently available algorithms to be reliable RBS designers. This is also exemplified by the fact that there is still no “state-of-the-art” algorithm for RBS design that is commonly used by the synthetic biology (and wider) community.**

**The authors do agree with the comment that if the goal of the work was simply to maximise the expression level, it would be easier to simply swap the promoter. However, this was not the goal of this work. Instead, this goal was to show how machine learning can be paired together with automated, high-throughput laboratory methods to design RBS, in this case by maximising it, but the exact design goal can be modified by subsequent users as described in previous answer.**

**Finally, the authors did reference the plasmid in the last paragraph of the “The experimental workflow” subsection.**  
  
- the two sequences that are stronger than the “benchmark” are only measured in a single replicate; given typical standard deviations in such experiments (compare also the experimental error for the “benchmark” in SI Fig. S2), it is at least questionable whether the small TIR increases for these two sequences (claimed 34% and 15%) are statistically solid.

**The referenced sequences have been tested in 6 replicates, not a single one, as described in the methods section. The raw results for the 134% RBS can be found at row 364 and for the 115% RBS at row 368 of the Results\_Masterfile.xlsx file found in our supplementary file Code.zip\synbio\_rbs\data . Coefficients of variation (mean/standard variation) for these two RBSs are 7.1% and 6.1% respectively, which we would describe as very low for biological samples .**

- In practical terms, it would have been more helpful to focus on how well the generated models predict the entire range of RBS strengths (from weak to strong). The R^2 values reached in this study are significantly worse than in comparable studies which reach 0.8 and beyond (as the authors correctly mention).

**The authors agree with this sentiment. We have addressed one of the main reasons for low reported R^2 values in the text (further modified in this revision to make the point clearer):**

“  
  
- directly related prior art is not appropriately introduced: for instance in Ref25 a combination of high-throughput data generation and machine learning is introduced as a means to predict RBS behavior with high accuracy; similarly, Salis and coworkers have used ML to improve a biophysical model for the prediction RBS strength (this latter reference is entirely missing: https://pubs.acs.org/doi/abs/10.1021/acssynbio.0c00394); given the very high degree of similarity to this study, it seems surprising that the authors forgot (or chose not to?) explicitely introduce these studies as prior art. Importantly, these studies reach much higher prediction accuracies than the underlying work.

**The authors did not choose to omit the work pointed out, rather they have decided to cite the original RBS calculator paper from Salis lab as the most relevant work from the said group. However, authors do recognize that the *Reis and Salis* paper does show the most recent advance in Salis RBS calculator and so did include it in the manuscript now.**

**We feel the work in question can’t be directly compared to ours in its goals. In the newly cited work Machine Learning is used to select model and parameters that would allow for efficient prediction, whereas in our work we present a new ML approach towards designing RBS, where ML is used directly used to learn from the data. In other words, we describe an ML model to evaluate RBSs, whereas in the cited work ML is only used to choose from non-ML models. Additionally, ML learning algorithm (bandits) is also used in our work for recommendation to balance the exploitation and exploration. We show how the increase in data volume (by dividing our data set querying into distinct rounds) improves the model, whereas in the cited work all the data is presented to the model at once.**

**Similarly, we feel that both ref 25 & 26 can’t be directly compared to ours. They both require data sets orders of magnitude bigger than ours and use neural networks, which is different approach to ours. They also do not use the cycling approach to teaching the algorithm that we have presented in our work.** **Finally, our approach allows for much targeted and rational design, rather than requiring generating hundreds of thousands of designs to ensure good predictability.**  
  
- It is unlikely that the designed RBSs will behave in a similar fashion in combination with coding sequences other than the used GFP gene or in other contexts. Therefore, any claim that the identified RBSs could serve as some kind of new standardized parts or would have utility in future studies is invalid.

**The authors agree with the fact that in different context the RBS would lose their ability to be seen as “standard”. However**, **standardisation studies should be done in keeping the context as close to the original one as possible.** **This is the reason why we have kept our system very simple – so that other researchers could easily replicate the conditions should they decide to** **test their RBSs against ours. One of our key contribution is to propose how to apply machine learning into DBTL design cycle and show Bayesian optimisation approach can help identify desired biological designs within a fixed (possible reletively small) budget. We have added the following sentence to the second paragraph of the “The experimental workflow” subsection to underline this point:**

**“Since standardisation and comparative studies should be done in as similar genetic context as possible, the design of this device has been deliberately kept simple to make such studies easier.”**  
  
Reviewer: 2   
More review: N/A   
  
Reviewer: 3   
More review: A major concern is that the authors have not described how they would deal with a multi-gene pathway with multiple RBS. Is this something doable within their workflow? What are the challenges for multigene pathways?

**The authors believe that their approach is appropriate for design of small genetic parts in isolation. For multigene pathways a more advanced deep learning techniques would be required. We are now added the following sentences to our method drawbacks analysis in the discussion:**

**“Additionally, our approach would not be appropriate for more complex cases involving multi-part devices/systems. In such cases, significantly higher number of data points and deep learning techniques would be required since the number of parameters to optimise would be also significantly higher”**  
  
Is this study really a ML-guided DBTL cycle? Experimental design of RBS might be seen only as one step between Design and Build. The impact of RBS on the final outcome is often only relative, as other factors might have a higher influence on the final performance of strains, such as those in the fermentation process.

**The authors agree that there is a multitude factors that play into the final performance of a strain. Having said that when we refer to the DBTL cycle in our paper we specifically focus on its use for the design of the RBS and so exclude other performance parameters by definition. We have updated our opening to the results section with the following sentence to better reflect that:**

**We present our RBS-optimising DBTL workflow that uses machine learning in Section 3.1.** **In this case, the DTBL paradigm is applied directly to the design of the RBS itself, rather than general strain optimisation.**  
  
LEARN: I don't see the point of showing R2 up to 0.546, this is far below reasonable values.   
I think that the valuable result is the one in Fig 3A or 3B, where high TIRs were achieved through different iterations.

**The reported 0.546 score is for Spearman correlation coefficient instead of R2, as shown in section 3.3. The authors find the reported R2 value to be reasonable for the presented conditions. However, since this has been also pointed out by other reviewers, we have modified the prediction explanation paragraph to better explain the values:**

**TEXT HERE**  
  
LEARN: Not clear why Hamming distance might impact the budget. As the authors are using Machine Learning, predicted sequences that are Hamming distant might be possible, not necessarily just those of single edit.

**The Hamming distance, as mentioned in the text, does not impact Machine Learning. However, it does pose a problem when an untargeted (random) approach like Adaptive Laboratory Evolution would be used as it might not be able to generate sequences distant enough which we show is required for positive change in RBS TIR. We explain this in more detail in third paragraph of subsection “BUILD & TEST: Characteristics of the tested sequences”.**  
  
**Minor Comments**

Reviewer: 1   
without review: Page 2:   
- line 15: DBTL (Not DTBL)   
- line 23: unclear meaning of “perceived prediction error”   
- line 30: spell check “Escherichia coli”   
- line 40: The argument that the workflow for RBSs can be later translated to more complicated systems is speculative and likely not true. For instance, promoters or proteins are governed by entirely different mechanisms/principles. Whether this studies has useful implications for these systems remains to be demonstrated. The authors should relativize this statement to that end.

**All fixed.** **Line 40 changed to: “As the RBS is one of the key genetic elements controlling protein expression and has a relatively short sequence, it is a perfect target for establishing a workflow that could be potentially translated to more complicated systems”.**  
  
Page 7:   
- the workflow description is very generic/inconcrete

**That is intentional – all the required details are included in the Methods section.**

- k is mentioned in the Figure caption, but not used in the figure (Fig. 2)

**k is used in the cell in the upper left corner: “n=k”.**

- line 54: the 20 bp sequence is only 19 bp long (T missing); the authors should specify what they mean by “core RBS”

**Added the missing T, added core of the RBS description:**

**“In our design, we focus on modification of the core at nucleotide positions -8 to -13 (relative to the start codon of the GFP; this is where the consensus Shine-Dalgarno AGGAGG sequence is usually found in wild type *E. coli* of the RBS and keep other positions the same as the benchmark sequence, i.e. TTTAAGA + NNNNNN + TATACAT, where N can be any nucleotide (A, C, G, T).”**  
  
Page 8:   
- line 30: what is the “original, consensus sequence”?

**Corrected from consensus to benchmark.**

- the authors should explain in more detail what the “Bandit-0” dataset is, how it was generated, and why they selected this

Page 9:   
- the authors should specify the SD sequence they use. Is that the core SD motif AGGAGG put into the context of the “benchmark”?

**Description of the Shine-Dalgarno sequence has now been put into text as specified by previous answer.**   
  
Page 13:   
- Lines 17 -19: The authors should specify how their approach compares favorably to Ref25. In this work, only one round is used in a very simple experimental setup, which none the less leads to about 1000-fold more datapoints.  Here, the authors use multiple rounds and complicated lab automation to generate only a small fraction of the data. The prediction accuracy reached in Ref25 is R^2 greater than 0.9 whereas the authors here reach only 0.27. Therefore, it is unjustified to claim that this work compares favourably to Ref25. Similar arguments may be made for other works in the context of RBSs.

**The authors agree that the way that comparison has been put into the discussion section is too simplistic.** **While it is true that the method presented by Hollerer *et al.*** **allows for acquisition of thousands of data points, it does rely on NGS, which in itself can be also seen as a complicated method.** **We would also argue that we did not used complicated methods, merely standard laboratory ones scaled up using automation.** **We would still argue that our method compares favourably, as it can be used with much smaller data sets and that is usually norm when it comes to biological research**. We have changed the relevant part in the text to better reflect this:

**“Compared to solutions like the one reported by Hollerer \cite{Hollerer2020}, our solution can be used when a high volume method for data-generation is not available, while still providing the required results (optimised part).”**  
  
Reviewer: 2   
without review: I have some comments/questions (see attached file) which, if addressed, I believe would improve the clarity of the manuscript:

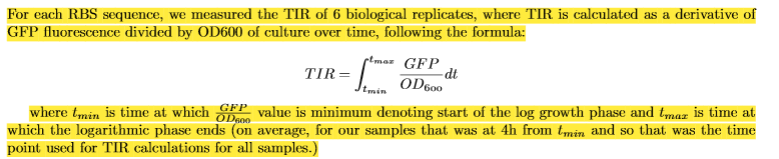
1. It is not completely clear from the beginning what do the authors refer to with the term Bandit and what with Round. It might be helpful to explain it somewhere in the beginning of Section2.2. It would also be advisable to clarify that Round refers to DBTL round and use either “round” or “Round” throughout the paper.

**The authors have clarified the terms in section 2.2 and unified the them throughout the text following this suggestion. The second paragraph of section 2.2 now reads:**

**“In round t, the prediction and design is based on the results obtained in all previous rounds. Our implementation of the machine learning algorithms was tested in Python 3.6 and used the scikit-learn library [29]. In the following paragraphs, we describe our machine learning pipeline(which is applicable to design rounds 1-3, denoted as Bandit1-3) in the following order: data pre-processing, prediction and kernels and finally recommendation. The pipeline for our 0th round of designs, denoted as Bandit-0 ,is reported in Supplementary A.4.”**

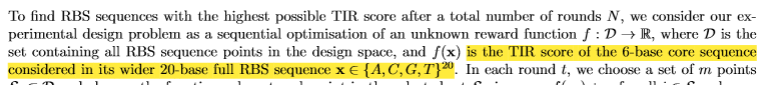
2. Section 2.2.1 – Please clarify how TIR is calculated, in particular the derivative part. An expression in a formula may be helpful.

**The authors have updated the relevant section following this suggestion and now it reads:**



3. Section 2.2.2 – The authors introduce their GPR approach and define the design space of 20-base RBS sequences. This may be misleading, as it turns out the design space for the algorithm is consisting of 6-base long sequences, which is explained at the end of Section 3.1. Only the BPS-NC, BPS-C, UNI, PPM methods consider 20-bp sequences, not the GPR approach that is employed for generating recommendations.

**The authors have updated the relevant section following this suggestion and now it reads:**



4. Page 4, line 32 – The assumption on the unknown mean and variance is in contrast to the one given in Supplementary, where a zero mean is assumed. It also turns out that α is the noise standard deviation, mentioned at the end of the section. Please clarify. I’d also suggest to clearly define yi (observations, labels).

5. Page 4, line 38 – The posterior mean and standard deviation, of what? Please say explicitly.

6. Page 5, last paragraph – I suggest to be more strict in using the term observation. For example,it is used both in “observed points (i.e. features)” and “true label, i.e. observation”.

7. Page 5, last paragraph – Given the batch approach for generating recommendations, it would be useful to clarify whether the training should be, or not, performed for each point in the batch, i.e. 90 times, so to calculate predicted standard deviation and update the UCB score.

8. Figure 1 – “...mean of±1.96 standard deviation”. If “of” is not a typo here it is confusing.

9. I strongly suggest to increase font size in all figures’ axes, titles, labels.

10. Section 3.1, Figure 2 – Please explain TIR(n)=TIR(n-1) in the caption or make a reference in the text. Also, in caption n, k –>n,k.

**We have added the following sentence to the caption:**

**TIR(n)=TIR(n-1) denotes "lack of TIR improvement following the used optimisation function".**

11. Section 3.1, last paragraph – The authors chose to modify the 6-bp core of the sequence with their approach, motivated by larger impact on TIR when compared to single nucleotide changes in the non-core region. However, although the impact is larger, in all cases it is towards lower values of TIR. Having in mind that the objective of the study is to increase protein expression, the question arises of why they decided not to consider non-core modifications.

**The authors agree with this observation. However, this is mostly because our benchmark sequence is already very strong and random changes will in most cases result in decreased TIR, as exemplified in our other results. Further, focusing on the 6 to 8bp long core sequence is a common RBS design approach (**[**https://www.nature.com/articles/ncomms11163**](https://www.nature.com/articles/ncomms11163)**). We have added the following sentence to make it clearer in our text:**

**“Notably, focusing on the 6 to 8bp long core sequence is a common RBS design approach (**[**https://www.nature.com/articles/ncomms11163**](https://www.nature.com/articles/ncomms11163)**). “**

12. Figure 3 caption, line 37 – I suggest to write in full “position probability matrix” instead of the second abbreviation.

**Text has been updated following this suggestion.**

13. Figure 3 caption, line 46 – Two consecutive “the”.

**Second “the” has been removed.**

14. Figure 3C – What exactly different exploration/exploitation colors for points within a round mean? This is not clear as exploration-exploitation parameter β is fixed for each round. For example, how is the color of a point, for which both predicted mean and standard deviation are high, determined?

15. Page 10, line 15 – Please give more details on “our embedding function”.

16. Page 10, line 26 – It is not clear what is the difference between Fig. 4 and Fig. S4 – the titles are the same, except of metrics’ values.

17. Page 11, lines 43-44 – I suggest to make the last sentence a bit more clear.

**The sentence has been updated to:**

**“This observation is in line with approaches seen in other disciplines, e.g. protein engineering, where more directed changes yield better results than wide random changes”**

18. Please introduce the STD and GFP abbreviations when mentioned for the first time.

Supplementary

1. Page 18 – Please define explicitly X,y.

2. After Eq. (4) – typo in covariance.

3. “...prior on the...” –> “covariance of the prior on the...”

4. The part “diagonal matrixα2Ion the kernel matrix” is not clear, Eq. (5) suggests it is a sum.

5. I suggest to write explicitly the expressions for the predictive distribution fory∗.

6. Eq. (8) – Seems like the left hand side should have x,x′ instead of X,X′. Please check.

7. It is not clear what exactly is φ specl(x).

8. Seems liked should be l just before Eq. (9) in Φ Specd (X). Please check.

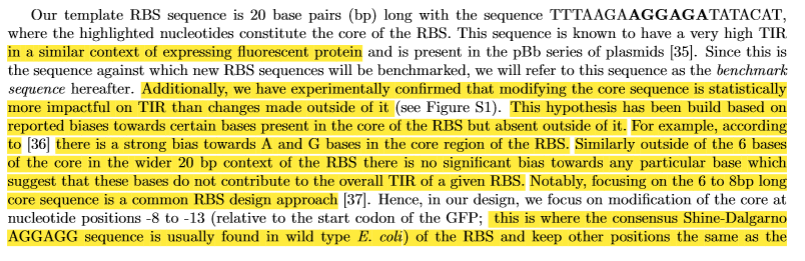
9. Section A.2.1 – It is not clear what actually is the φ function. The text suggests it is ad-dimensional function – what is d?

10. It is not clear how Eq. (17) follows from Eq. (16).

11. Unit norm definition – Please check the subscripts m,d.

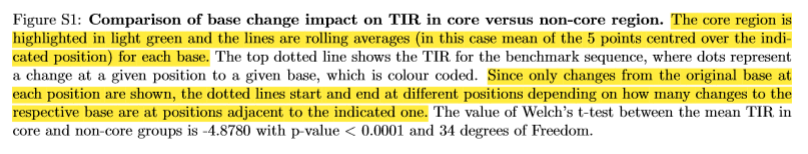
12. Figure S1:– What exactly are the rolling averages? Why they don’t start from the same position for all bases?– There is a self-reference to Fig. S1 within the caption.– What is “set X of sequences”?– It would be useful to mention in the caption that these are single nucleotide changes.

**Part of the main text relating to this plot has been mistakenly left in this caption. This has now been fixed and both main text:**





**and the caption in question:**



**have been corrected.**

13. Figure S2 – “Dash line is...” seems copied from another caption.

14. Figure S3 – It is not clear what are the projected values  
  
Reviewer: 3   
without review: Figure 3A. I am not sure if random selections should always led to such low TIRs compared with the benchmark because of the low dimension of the design space (4096). Have the authors tried to statistically validate the results? Comparison with a random sampling, etc.

**Our results for random sampling are in line with other results published in literature. For example, Hollerer *et al*. in their work (**[**https://www.nature.com/articles/s41467-020-17222-4**](https://www.nature.com/articles/s41467-020-17222-4)**) show relative TIRs for over 300,000 randomly sampled RBSs in Figure 3b. There, one can see that most randomly generated RBSs perform worse by at least 25% compared to the strongest sequences. In our case the benchmark sequence is very strong and our randomly sampled sequences also perform at least 25% worse than it does.**   
  
Figure 3C. Exploitation results are significantly higher than exploration results. I would expect exploration results higher but perhaps it makes sense. Could the authors comment on this?

**While focusing on exploitation, the algorithm recommends RBSs that have a high predicted mean, resulting in overall higher TIR RBSs. On the other hand, in exploration, algorithm focuses on RBSs that have high uncertainty and these sequences do not necessarily have high predicted mean which results in overall lower TIR RBSs. This is also the reason for relatively low R^2 value. However, like we mentioned in text, exploration allows for better understanding of the underlying function and, in result, better predictions in subsequent rounds.**  
  
I believe that t-SNE is not actually defined.

**We have updated the first sentence of the last paragraph of subsection 3.2 to read:**

**In 3E we show a t-distributed stochastic neighbour embedding (t-SNE) plot depicting the experimental space.**