This document contains responses to the comments raised by the reviewers. We would like to thank the reviewers and the AE for some insightful and highly relevant comments and we hope that the reviewed manuscript reflects the changes inspired by your comments. In the main manuscript major changes are shown in red for easy identification.

Reviewer 1

* Code for reproducing manuscript results is not available.

Unfortunately, we are not allowed to distribute the *data* used for the ovarian data example, but we have made the full set of R code for the simulations and the analysis of the Golub data are available on github under the address <https://github.com/tagteam/SuperRanker> which also houses the development version of the R package. XXX

* The practical use of SRA is not well-motivated. The real-data examples show the behavior of SRA curves for many different ranking procedures, but it is not obvious how to actually use this information. It would be helpful to describe a scenario in which a decision is made with the SRA curve. Consider using multiple rank agreement methods in your real-data scenario and show that SRA suggests a different decision than the others (if this is true).

We agree that that point may be hidden in the previous version of the manuscript. We have rephrased the start of the introduction emphasizing more specific and detailed scenarios in which the sra may come into use.

Also, we have put the Golub dataset into a framework resembling a real-data scenario 8section 2.1), have added a discussion about how to use the sra, and have stressed the conclusions that we obtain from the analysis in that section.

* Why not compare to TopK in the real-data example as well as the simulation? The Sampath and Verducci method is described in the introduction as “guaranteed to work for small samples”, so why is it not compared to as well?

The comparison study of the methods (Figure 4) shows that the two methods (TopK and sra) focus on different aspects of the ranked lists and are not really directly comparable. We hope that Figure 4 (and its updated explanation in Section 5) high-lights the differences between the methods and explains why we have not extended the comparison the other settings.

The “guaranteed to work …” comment was not that well worded. The method of Sampath and Verducci works only for exactly 2 lists and thus they do not consider the situation where the number of lists increases.(that was what was hinted to with our earlier wording). That sentence has been rephrased. For the same reason we cannot compare this approach in the real data application (Figures 2 and 3) where we measure the agreement of 1000 lists.

* It is difficult to understand what the simulation result (Figure 4) is telling us. SRA produces smaller agreement sets, but is this a good thing? Is this correct in light of the simulation setting? Is there a clearer simulation setting with some sort of gold standard within which the methods could be compared?

You are completely right. However, we wanted the simulation study to show two things: 1) how the sra is influenced by the choices of threshold q and number of lists L and 2) to show that the different methods capture different aspects. To be able to answer whether one method is better than another we should in reality compare to the truth: do we correctly identify the 15 predictors that actually had an impact on the mean. However, both sra and topK are concerned with agreement and not whether they agree on the correct predictors. A possibility would be to consider the average number of predictors that are part of the identified set, and here the sra seems to hit close to the right number whereas topK obtains too high a number. We have updated the text in the simulation study to stress these points.

* There is no discussion of what to do if your lists contain ties.

Note that we do discuss how to handle ties that occur at the end of the lists in form of missing/truncated rankings in section 2.2. We have not discussed ties otherwise because we thought that ties will rarely occur in practice – they do not occur in the real data applications of our manuscript. However, considering ties could be interesting to pursue at a later point and a possible approach would be to make a (restricted) permutation clause between the tied ranks in much the same manner as what is done for the truncated lists. Say, if items 3,4 and 5 were tied for positions 1,2, and 3 then we could permute the positions 1,2, and 3 among the 3 items and average over the resulting sra curves.

* The treatment of missing data seems somewhat unrealistic. How often are we dealing with lists in which the missing values begin occurring at a certain depth? What do we do if a particular feature is ranked very highly in a few lists and is unranked in the remaining lists? Assigning a poor random rank for these features seems improper in light of their high ranking in the few lists.

This is a great question and underlines a subtle point when comparing ranks. The situation with truncated data is – in our experience – more common than what would be expected. If information is sampled from scientific papers that only present, say, top-10 lists of rankings (e.g., the most important genes for a specific trait) then we have incomplete lists. Likewise, sparse regression methods such as the lasso inherently produces lists where several (if not most) predictors have a coefficient set to zero. Those predictors cannot be easily ranked and thus the list of rankings becomes incomplete. This is stressed in the first paragraph of section 2.2. The point about the poor random rank stresses the fact that what we are comparing is *not* whether or not the ranks are correct but whether or not the lists agree. If high rankings only occur in a few lists then that will diminish the overall rank agreement among the lists. We have tried to stress that even further in sections 2 and 3 to make it clearer.

* More description of hypothesis testing and how/when it should be used would be helpful. I created a simple example (code at the end of the document), and find that the p-value depends fairly heavily on the permuted null distribution.

The p-value in your example can indeed vary somewhat depending on the particular random seed (Monte-Carlo variation). However, the problem is due to the short length of the lists and the too low number of permutations.

For 3 random lists of length 5 there is – by chance – a 4% probability that all three lists have the same item at rank 1. This means that 100 permutations (random list rankings) is too small a number: the results will change quite substantially depending on the actual number of permutations. Also, you compare to a significance level of 0.05 but the actual level of most of your non-significant p-values are just over 0.05:

summary(output)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00000 0.02970 0.03960 0.04317 0.05941 0.09901

If we increase the number of samples to 400 we get the following

summary(output2)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.01746 0.03242 0.03990 0.04027 0.04738 0.06484

where it is clear that the variation is much smaller than before – just by increasing the number of comparisons. If we increase the number of permutations to 800 the difference between the extremes are even smaller: minimum 0.026 and maximum 0.0599.

Also, because there are only 5 items in each list, the number of different rank list configuration is small which results in only a few number of possible, different p-values, which makes comparisons against a small number of replication less than ideal. In summary: increase the number of samples and the result will be stable. We have emphasized this in the section on permutation inference.

* The example ranking probabilities is somewhat problematic. For one, since probabilities fall in such a small range, ranking may not be meaningful or helpful. Why not use an example where a continuous risk score or survival prediction is produced instead?

In many situations there exists a direct one-to-one link between a (continuous) risk score and a risk prediction or survival prediction. That means that the ranking based on the risk score is identical to the ranking based on the risk prediction. Although the risk predictions may appear closer together because they are restricted to the interval [0, 1] it will have no impact on the individual ordering. Also, note that due to right censored data survival predictions are usually given as probabilities, e.g., 5-year risks, and not as predicted survival times at least in the biomedical applications.

* Second, we should expect agreement at the top and bottom of the list of probabilities since both correspond to high-confidence predictions (close to one and close to zero). This is not reflected in the SRA curve and I am therefore not sure how to interpret or use the curve.

This is a great comment and a question we have been pondering ourselves. It is possible to reverse all the lists and consider items towards the bottom, but in some instances it would of interest to consider ranking at both ends simultaneously (notably for risk prediction where *both* ends of the lists are of interest). In some cases, depending on the underlying scale used to generate the rankings – it might be feasible to use the proposed sra method directly. If, say, we have risk predictions Pi for each individual i then we could use the transformed score

\tilde(Pi) = |Pi-0.5|

to produce the ranking. It would then put equal weight at both ends of the probability scale. This obviously require that the risk prediction span the same range on both sides of 0.5. We have added a comment about this to the discussion.

* When predictors are truncated, as in Figure 3 left panel, the resulting curve looks more as we expect (low, then increasing), but the absolute SRA values are higher. What does this mean, and is this desired behavior?

In some sense we can consider the sequential rank agreement obtained from truncated lists as a smoothed version of the sequential rank agreement from the full-length lists. If an item is not part of the observed part of all the lists then its ranking in the lists where it is unobserved is based on an average and consequently it will vary less, but have a higher sra because it is – in some sense – compared to the middle part of the lists where it was not observed. It is indeed the desired behavior, and we have added this to the discussion of Figure 3.

* Suppose that we trust one ranker more than others. Is it possible to weight these rankings higher than the others?

That would indeed be possible by constructing a *weighted sequential rank agreement* by multiplying a pre-specified list-specific weight, w\_i, to each ranking, when computing A(X\_p) in formulas (2.1) and (2.2), and normalizing with the total sum of the weights. However, that is not the focus of the current paper, and we haven’t looked into the statistical properties of the weighted ranking or implemented it. With the current proposal and implementation, however, it would be possible to increase the number of lists and clone specific lists to put extra weight on the ranking obtained by the lists that are cloned.

* Because it is sequential, each subsequent depth depends on all that has occurred previously in the list. Have you considered computing agreement in windows? i.e. 1 <= d <= 10, 11 <= d <= 20, etc. How would this piecewise curve compare to SRA?

That is an interesting suggestion. If we assume that the lists are (in some sense) likely to include the same of items towards the top of the lists then the current sra would represent a lower bound compared to the piecewise curve. Since the piecewise curve will be less influence by the start of the lists the piecewise rank agreement should be larger than what we see with the sequential approach. It is also likely to be more variable since the number of items in each window is likely to be more or less constant and not increased with d as is the case for sra.

* What is a “high” SRA value? What is a “low” value? Are they relative to their curves, or can they be compared cross-curve?

The sra measures (“average”) rank positions. So clearly its scale depends on the length of the lists. It also depends on the number of lists. In order to assess whether the value is high/low we propose to compare to what can be expected from permuted lists (same number of lists and same length as in the current application) and to a pre-specified threshold which clearly depends on what the investigator thinks is acceptable.

* Asymptotic properties for changepoints are derived, but changepoints are not really mentioned in the analyses/simulations, and it is not particularly obvious what a changepoint estimate should be used for.

That is indeed correct and it should have been clearer. The suggestion of using change point is to replace the pre-specified acceptable threshold for rank agreement and to use the data to identify where changes in rank agreement (and hence sets of items that are ranked similarly) occur. We have included this point in the analysis of the Golub data in Section 2.1 and have stressed it as part of Section 3.2.

* Figure 1 has no axes or labels.

All figures in the revised version have axes and labels on both axes.

Reviewer 2

* My main concern of the sequential rank agreement is its harsh threshold. In other words, even a slight mis-ranking of the top objects results in unexpected results. For example, suppose that under Q distribution (which is the collection of the ranking methods), for a fixed d, P(Ri<=d)>0 for all i's i.e., that is, no matter the object i is, there is a chance (even though slim) that it is ranked in the top d objects. In fact, this is a very common and plausible scenario. Now, S(d')={X1,...,XP}  for any d'>=d and sra(d')=sra(d)= A(X1)+...A(XP). Note this is the theoretical statistic not an empirical estimate. This situation for low number of rankers, i.e. small L's, is not a problem but if we increase the number of L, the empirical estimate of sra(d) converges to true sra(d) which is constant across d and there is no way that the algorithm detect any concordance beyond d while it could be very little discordance after d.  
  I recommend that the authors fix this harsh thresholding problem by modifying the definition, or study such cases more in depth under the simulation section and explain where SRA is powerful and where it has its own drawbacks and come up with

Thank you for this extremely relevant comment. The definition in the previous paper which used *all* items that had a non-vanishing probability of being ranked 1 would indeed be part of S(1) (asymptotically) and the sra would be a horizontal line. In practice, when the number of lists, *L*, is not too big this would not result in a problem. However, we have modified the definition of S(d) to accommodate that the probability must exceed a pre-specified threshold, , to represent the fact that the item should have a probability greater than  of having a given rank (ie, be part of more than \*100% of the lists) to be included. We have updated the proofs and the R implementation accordingly.

* In the second permutation test, it is unclear to me how you perform the permutations. Please elaborate more on it.

Thanks for pointing this out. We have rewritten parts of the section about the permutations and hope it is clearer now.

* I think mentioning and comparing SRA to CATPlot (implemented in matchBox package) is required. Also, other rank methods applied to RNA expression analysis such as kTSP (implemented in switchBox) used for phenotype prediction and EVA (implemented in GSReg) for gene set analysis worth mentioning as successful applications for cancer analysis

A correspondence at the top (CAT) plot has a similar aim as our SRA

plot. It takes exactly two lists and plots for each list depth the number of

items that the two lists have in common. This plot was introduced to

compare two procedures for detecting differentially expressed genes

(REF: Irizarry and others, 2005, vedhæftet). In fact, it seems that essentially a CAT plot is the special case of our setting where the agreement metric is the overlap proportion.