User Guide for Reproducibility Assessment and Optimal Study Design using a Bayesian Hierarchical Model

Software to accompany Assessing the validity and reproducibility of genome-scale predictions Sugden et al. DOI: 10.1093/bioinformatics/btt508

In many genome scale studies, large numbers of predictions are made (e.g. sites of transcription factor binding) and a validation scheme (e.g. Sanger sequencing) is applied to a subset of predictions. When validations are carried out in a single replicate, there is no way to assess the effect of biological and sample preparation variability on the fraction of predictions found to be valid. Our software serves two distinct purposes:

1. Assessment of Reproducibility: Given validation experiments carried out on randomly drawn predictions in 2 or more biological and technical replicates, assess the reproducibility of the study by inferring the distribution of the fraction of predictions that would be found to be valid in an as-yet unseen replicate (called the predictive distribution).
2. Optimal Design of Validation Studies: Given a total number of experiments to be carried out, distribute the experiments into the optimal number of replicates that will return the most favorable reproducibility assessment.

Assessment of Reproducibility

To assess the reproducibility of results from *N* replicates, run `hierarchical_pred_inf_FAST.m` with the following arguments:

- `trialnum`: vector of length *N* with the number of predictions tested in each replicate (not counting those predictions that failed from the start, e.g. primers failed to bind).
- `trialsucc`: vector of length *N* with the number of positive validations in each replicate.
- `verbose=1` will print all of the relevant graphs and statistics.
- NUMSAMPLES: the number of samples drawn from the predictive distribution in order to infer statistics like mean, variance and percentiles. A good default is 10,000.

Three values are returned, the mean, standard deviation, and 10th percentile of the predictive distribution, and further statistics are saved in the current directory in a file called `Predictive_Distribution_Statistics.txt`.

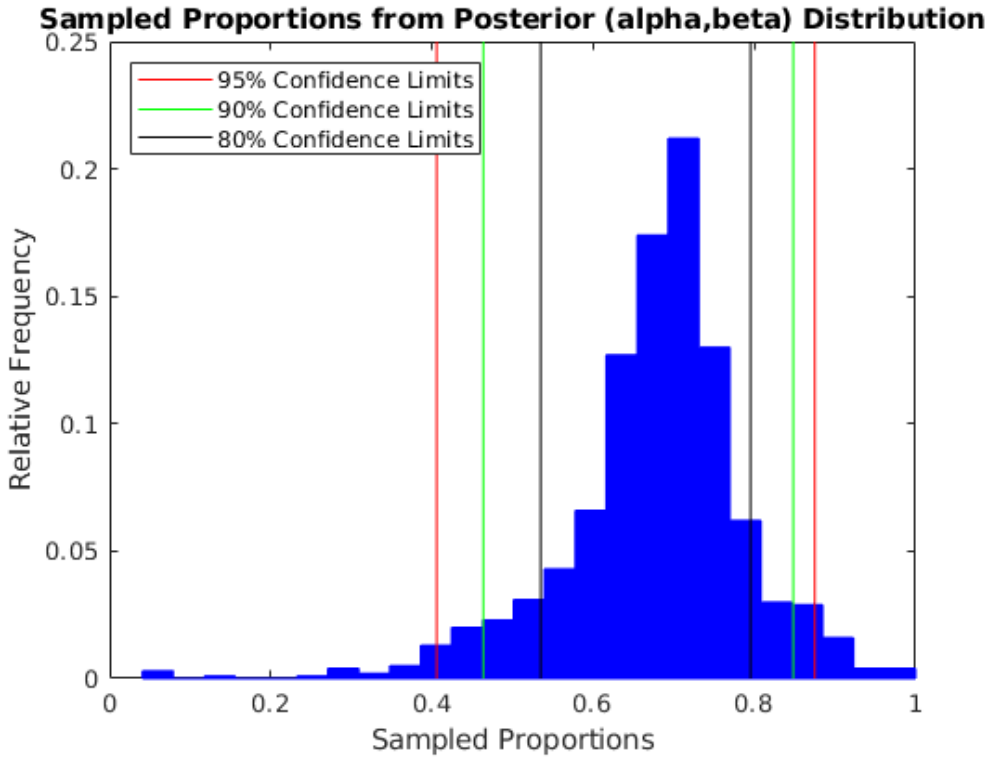
Also saved in the current directory are a histogram representing the predictive distribution, and a contour plot representing the posterior distribution of the hyperparameters of our hierarchical model (see paper for details), both in pdf and png format.

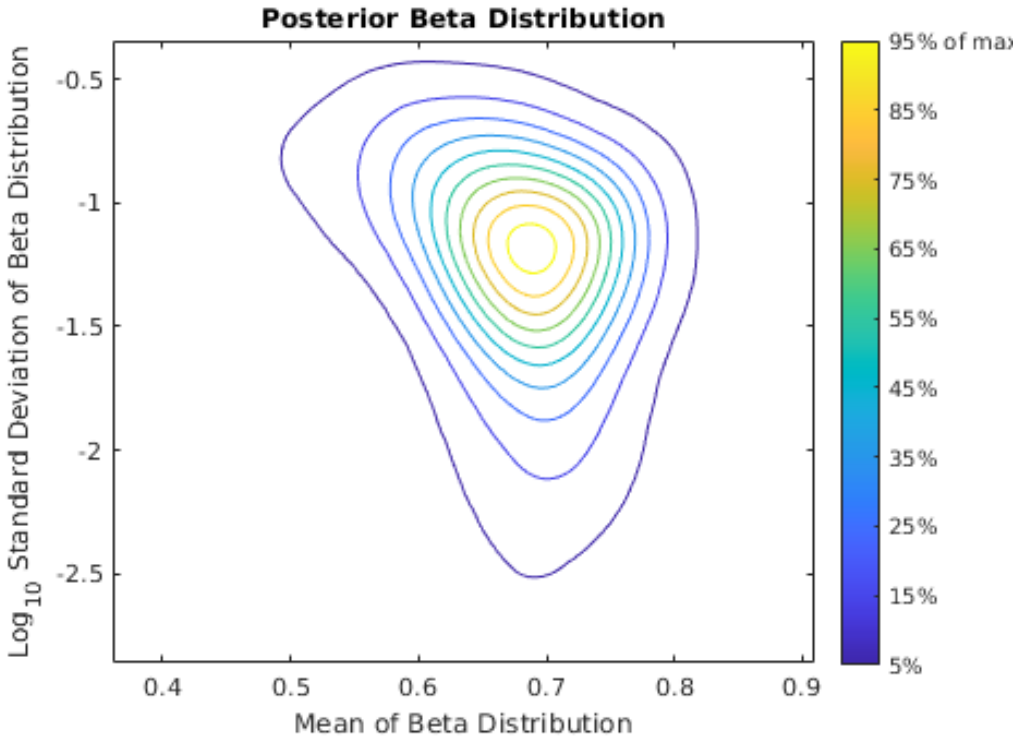
The ADAR dataset described in our paper is provided in ADAR.mat as a test set. Results can vary a little bit, since predictive distribution values are based on samples, but `hierarchical_pred_inf_FAST.m` should return something like the following in `Predictive_Distribution_Statistics.txt`:

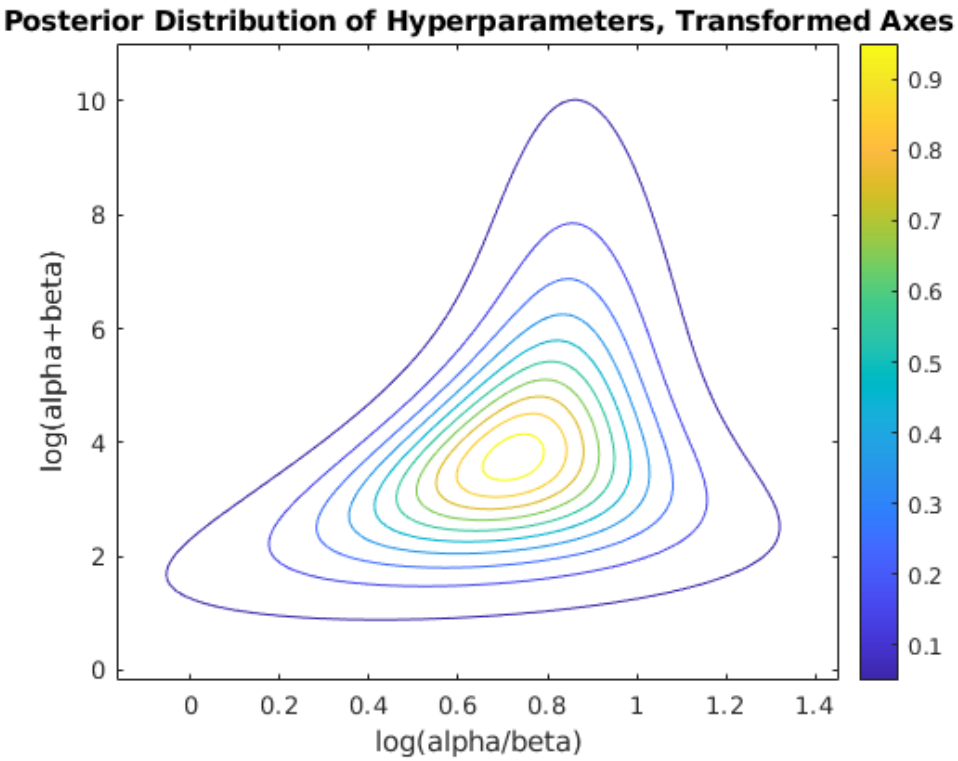
```
>> load('ADAR.mat')
>> [pavg,pstdev,ppercentile10] =
hierarchical_pred_inf_FAST(trialnum,trialsucc,1,10000);

Number of Experiments: 205   32   30   32   29
Number of Validations: 151   17   21   24   18
Mean of p: 0.67123
Standard Deviation of p: 0.11451
10th percentile of p: 0.53664
0.95 Confidence Interval: (0.39099, 0.87705)
0.90 Confidence Interval: (0.46635, 0.83692)
0.80 Confidence Interval: (0.53664, 0.78957)
```

With the following figures:


Sampled Proportions from Posterior (alpha,beta) Distribution
95% Confidence Limits
90% Confidence Limits
80% Confidence Limits
Relative Frequency
Sampled Proportions
0
0.05
0.1
0.15
0.2
0.25
0.2
0.4
0.6
0.8
1


Posterior Beta Distribution
Log10 Standard Deviation of Beta Distribution
-0.5
-1
-1.5
-2
-2.5
0.4
0.5
0.6
0.7
0.8
0.9
Mean of Beta Distribution
95% of max
85%
75%
65%
55%
45%
35%
25%
15%
5%



Optimal Design of Validation Studies

Before a validation study is carried out, investigators may want to design the study so that the predictive distribution computed above has the highest 10th percentile and narrowest standard deviation possible. The MATLAB script `draw_optimal_curves.m` simulates thousands of datasets for different numbers of replicates so that the optimal number may be chosen. The user must specify 5 arguments:

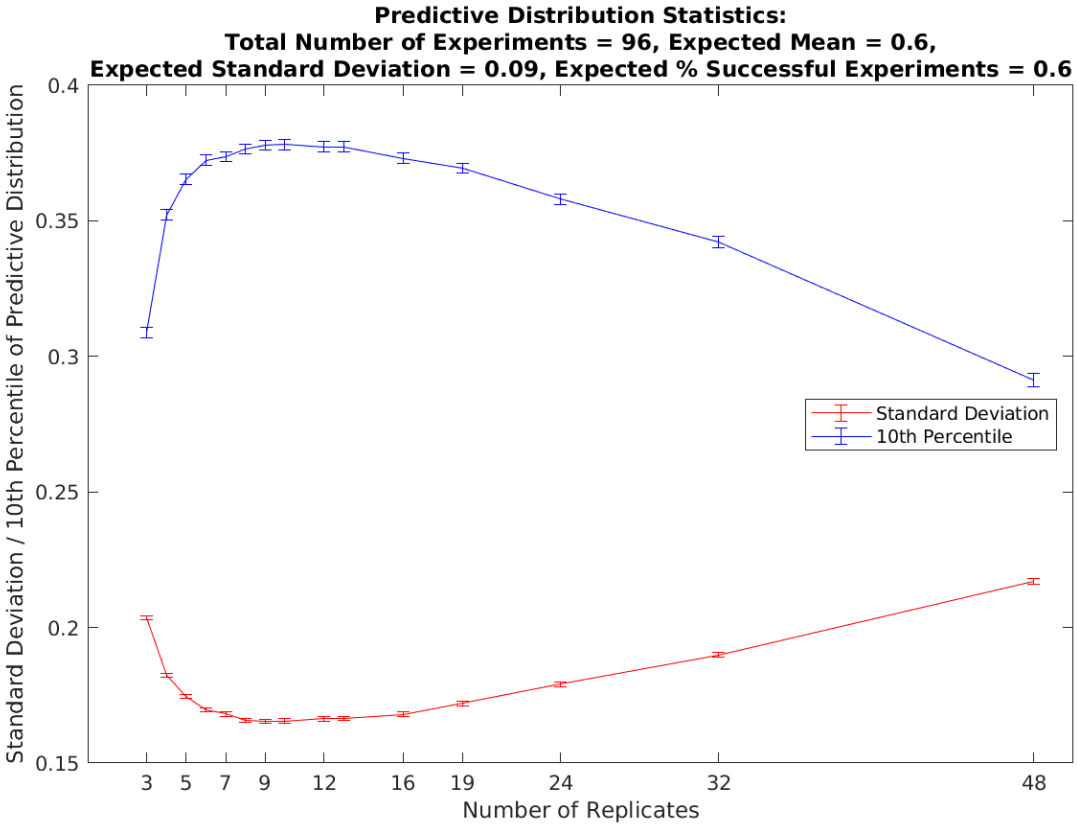
- `N`: The total number of validation experiments (i.e. the number of predictions) to be performed across all replicates.
- `inputmean`: The average proportion valid that the investigator expects to see across the replicate pools.
- `stdev`: The expected standard deviation of the proportions valid between different replicate pools.

- `ps`: The fraction of experiments expected to successfully yield either a positive or negative result (1−ps is the fraction of experiments expected to fail due to failure of primers, etc...)
- `Nper`: The number of simulations per replicate number. `Nper = 3,000` gives tight error bars, but is extremely slow, and only feasible if multiple processors are available to run simulations in parallel. To take advantage of parallel computing, uncomment lines 16, 28, 104, and 105, Comment line 29. See the `parpool` help file for more details.

The middle three "expected" values will typically be based on the previous experience of the investigators.

When the script is finished running, it outputs a text file with the means and standard errors for the predictive distribution standard deviations and 10th percentiles for each replicate number, and a plot of these values (in pdf and png format) so that an assignment of replicates with low standard deviation and high 10th percentile can be easily chosen. An example of the curves generated by this script is given below, for input values N= 96, inputmean= 0.6, stdev= 0.09, ps= 0.6, Nper= 3000. Here, we would choose somewhere between 8 and 12 replicates.

```
draw_optimal_curves(96, 0.6, 0.09, 0.6, 3000)
```


Predictive Distribution Statistics:
Total Number of Experiments = 96, Expected Mean = 0.6,
Expected Standard Deviation = 0.09, Expected % Successful Experiments = 0.6
Standard Deviation / 10th Percentile of Predictive Distribution
0.4
0.35
0.3
0.25
0.2
0.15
3
5
7
9
12
16
19
24
32
48
Number of Replicates
Standard Deviation
10th Percentile