ODD protocol for the simulation model

“Wisdom of Expert Panels”

This model description follows the ODD (Overview, Design concepts, Details) protocol for describing individual- and agent-based models (Grimm et al., 2006, 2010).

The simulation model “Wisdom of Expert Panels” is used in a research paper currently under review.

Unless noted otherwise, all variables and functions are named consistently between this ODD document, the code, and the accompanying paper (though with minor spelling differences, e.g. due to the use of *camelCase* in the code). However, when names do differ between these documents, cross-refences are helped by the color-codes:

* Black – naming convention for this ODD protocol;
* Orange – naming in the R scripts;
* Purple – naming in the accompanying paper.

# 1. Purpose

This model simulates a scientific peer review process, where panel members (i.e. reviewers) evaluate a set of alternatives (e.g. research grant proposals), and where their individual judgments are aggregated to a panel decision (i.e. a funding recommendation). Simulations allow the exploration of the panel design principles listed below, and their effect on the performance of the panel.

1. the number and level of competence of the reviewers;
2. the granularity of the grading language;
3. the magnitude of interpersonal differences in the interpretation of the grading language;
4. and the aggregation method, or rule, adopted for combining individual judgments into panel judgements.

Despite this documentation following the format of an ODD protocol, this model is technically not an agent-based model (ABM), for it lacks the inter-agent interdependencies and interactions typical of ABMs. Nevertheless, this model is designed to allow for ABM components (or submodels) to be plugged in. For example, the model can easily incorporate opinion dynamics between panel members, simulating peer review panels where reviewers jointly discuss the submission under review.

# 2. Entities, state variables, and scales

There are three main entities in the model: the peer review system (i.e. the environment and rules within review panels must wor), proposals, and reviewers. In this section all their attributes are listed.

## Peer review system

This list contains the global variables (or ‘parameters’) of the ABM, which are constant throughout the simulation run:

* Random seed. 32-bit integer. Seed for the random number generation.
* Number of proposals (nSubmissions). Integer > 1.
* Number of reviewers per proposal (nReviewersPerProp). Integer > 1. Defines how many reviewers review each proposal.
* Number of proposals per reviewer (nPropPerReviewer). Integer ≥ 1. Minimum number of proposals that each reviewer reviews.
* *α* and *β*. Continuous variable in [0, +∞]. These are the two parameters of a beta distribution, used to sample the true quality of the proposals (see next section, “Proposals”).
* L (scale). Integer > 1. Defines the granularity of the grading language. When L=5, for example, we are modelling a review panel where proposals are evaluated on a scale with 5 categories, e.g. a 5-point Likert scale.
* Grade language. Dichotomous: either “symmetric“ or “asymmetric“. This defines how the categories in the grading language are mapped to the imaginary continuous merit scale: at regular intervals (“symmetric”) or such that more categories are concentrated on the high end of the merit scale (“asymmetric”). See Section 5 “Initialization” for the details.
* H, grade language heterogeneity (glh or, In the accompanying paper, “D” for inter-reviewer *diversity* in the interpretation of the grading language). Continuous variable in [0,1]. When H=0 it means that all reviewers will map the grading language on the merit scale in the exact same way. In the accompanying paper, this is called “D”, for inter-reviewer *diversity* in the interpretation of the grading language.
* Reviewer error. Continuous variable in [0,1]. This captures the amount of noise that reviewers have on average when estimating a proposal’s true quality. Higher reviewer error means that reviewers will make noisier evaluations. The accompanying paper speaks of “reviewer competence”, denoted “C” and defined as 1 – reviewer error.
* Aggregation rule (aggrRule). Vector of character strings. This defines the procedure (or set of alternative procedures) used by the review panels to aggregate the individual evaluations of its reviewers. The valid strings / rules are listed in section 4 “Design concepts / Basic principles”.
* Acceptance/funding rate (nAccepted). Vector of integers greater than 0 and lower than the number of proposals. This indicates the alternative number of proposals that the panel must choose out of those submitted. Each of the alternative values is denoted “k” in the explanations below as well as in the accompanying paper.

## Proposals

Proposals have the following attributes.

* True quality. Continuous variable in [0,1], where higher values signify higher quality. This variable defines the quality of the most recent submission made by scholars and is updated every time a scholar’s role is set to ‘author’. In the accompanying paper, true quality is referred to as “objective merit”.
* True ranking. Integer. This is the ranking position in the weak ordering of proposals by their true quality. Proposals of relatively high true quality will occupy relatively low true ranking positions.
* Estimated quality. Continuous variable in [0,1], where higher values signify higher quality. This is the panel’s evaluation of a proposal – which is why it is called “panel judgment” in the accompanying paper.
* Estimated ranking. integer. The ranking position in the weak ordering of proposals by estimated quality. It is called “panel ranking” in the accompanying paper.
* Deserving funding (tqDeserved). Logical. Set to TRUE for proposals that are among the best k by true quality; FALSE otherwise. In case there are ties for the k-th best proposals, all proposals in the tie are set TRUE. Thus, there may be more than k proposals with deserving acceptance == TRUE.
* Found to be deserving funding (estimated). Logical. Set to TRUE for proposals that are among the k-best by estimated quality; FALSE otherwise. Ties for the k-th best position are treated in the same way as above - see “deserving acceptance”.

## Outcome variables

In addition to the model parameters under section “Peer review system” there is another set of global variables: the outcome variables outlined in the accompanying paper.

* Ranking performance: Spearman's ρ (spearman). This is one of the two main metrics of panel performance in the accompanying paper: it is the ranking correlation between proposals’ “true ranking” and “estimated ranking” (see next section: “Proposals”). The higher the correlation, the stronger the performance. For panels that do not make distinctions between proposals (i.e. who give all proposals the same grade and ranking position), a correlation coefficient cannot be calculated – for these panels, ranking performance is set to zero, as to indicate that the panel failed to rank proposals by their true quality.
* Ranking performance: Kendall τ-b (ktc). Similarly to Spearman's ρ, Kendall τ-b is a correlation coefficient that we calculate between proposals’ “true ranking” and “estimated ranking”. Just like for Spearman's ρ, when there is no variability in the estimated ranking, ranking performance is set to zero.
* Choice performance (rankingEfficacy). This is one of the two main metrics of panel performance in the accompanying paper. Choice performance is the average proportion of proposals deserving of funding (i.e. proposals for which “deserve funding” == TRUE) that are correctly recognized as such by the panel (“found to be deserving funding” == TRUE). This is an *average* proportion, because the panel’s “estimated ranking” is a weak ordering: thus, choice performance is averaged across all possible resolution of ties in the estimated ranking.
* Rank similarity (ktd). It is based on the Kendall distance (a.k.a. “bubble-sort distance”, an algorithm that repeatedly compare and order adjacent elements in a list) between the “true ranking” and the “estimated ranking”. Rank similarity is calculated as 1 minus the normalized Kendall distance. The Kendall distance is normalized to range in [0,1] by dividing the raw Kendall distance by its theoretical maximum (i.e. the theoretical maximum number of “bubble swaps”, given a ranking of that length).
* Peer review error (evaluationBias). This is the absolute difference between a proposal’s true quality and its estimated quality, averaged across all proposals.

# 3. Process overview and scheduling

This table shows the model’s pseudocode. The pseudocode follows strictly its code implementation, which can be seen in the script “simulation.r”. Each function is explained in detail in the next sections.

|  |  |
| --- | --- |
| initialization | Initialize proposals  Initialize reviewers |
| simulation | For each proposal *p* (     For each reviewer of *p* (        Calculate the opinion of *p*        Return grade given to *p*     )     For each *aggregation rule* (        Calculate estimated quality of *p*        Calculate estimated ranking of *p*        Calculate outcome measures     )  )  Return outcome variables |

After the initialization of the model entities (see section 5: “Initialization”), the model simulates the panel’s decision making by doing the following:

1. Determining the grade that each reviewer entity gives to the assigned proposals.
2. Aggregating reviewer’s grade into a panel judgment. This is done for each of the specified “aggregation rules”. For each aggregation rule, the simulation creates an entry in a list (called “r” for *results*) where it stores the resulting proposals’ attributes “estimated quality”, “estimated ranking”, “deserving funding” and “found to be deserving funding”.
3. Calculating the outcome variables. All outcome variables (and, when relevant, for all values of “funding rate”) are added to the results list “r”.

# 4. Design concepts

Elaborating on the eleven design concepts (only some of which are applicable to this simulation model):

## Basic principles

The model explores four principle that can inspire the design of expert (review) panels:

(1) their size (“number of reviewers per proposal”);

(2) the magnitude of reviewer error;

(3) the granularity of the grading language (L);

(4) reviewers’ interpretation of the grading language

(“grade language” and “grade language heterogeneity”),

and (5) the aggregation rule adopted.

Some of these design principles are inspired by the literature on wisdom of crowd and social simulation literature on peer review (1, 2) (REF); some are ingredients that are novel for simulation models of peer review, like the representation of realistic grading languages and heterogeneity in reviewers’ interpretation thereof (3, 4). Last, aggregation rules (5) were never systematically studied on the same benchmark, with the exception of few studied that only compared few at a time (REF).

Furthermore, to our knowledge there is no research on the *combined* effect of these design aspects even beyond the peer-review and social simulation literature.

To build this model, we relied on a standard set of assumption typical to simulation models of peer review, including the assumption that:

* reviewers act independently form one another (i.e. without social influence effects);
* proposals have a true quality;
* and the performance of a peer review panel can be measured (in the model) as some form of *similarity* between proposals’ true quality (e.g. its resulting ranking) or the panel’s decisions (REF).

Of particular interest in this study is the design aspect concerning the aggregation rules.

Drawing from the simulation literature on peer review (REF) and on expert panels (REF) we constructed and implemented in the model a list of alternatives, competing aggregation rules:

* Mean. A proposal’s estimated quality is the average of its reviewers’ scores.
* Hypermean. A proposal’s estimated quality is the weighted average of its reviewers’ scores. The weight of a reviewer r is calculated as follows. First, a ranking correlation coefficient (Spearman's ρ) is calculated between r’s grades and the average grades by the whole panel. Then, r’s weight is defined as the correlation coefficient normalized to range in [0,1] using the formula: (ρ + 1)/2. This procedure assigns lower weights (and thus gives less importance) to the evaluation by reviewers who tend not to agree with the panel: in other words, it exerts a dampening effect on the evaluation of reviewers who are particularly influential in the panel (i.e. outliers). A special case is when Spearman's ρ cannot be calculated, e.g. for reviewers who gave all proposals the same grade. These special cases are treated by assigning these reviewers a weight = 0: this reflects their null contribution to the panel in discriminating proposals by merit.
* Median. (self-explanatory)
* Lowest score. A proposal’s estimated quality is the grade given by the reviewer who gave the lowest grade.
* Majority judgment. A proposals’ estimated ranking is determined by the majority judgment rule, defined in Balinski & Laraki (2007) (FIX REF).
* Borda count. The Borda count (REF Pacuit on voting methods in the Stanford encyclopaedia) determines proposals’ estimated ranking. The Borda count is here generalized to allow weak orderings. Specifically, when there are ties in the estimated ranking, the Borda count is the result from averaging the Borda count that can be calculated for all possible resolution of the ties. Note this treatment of ties is different from how the proposal attributes “deserving funding” and “Found to be deserving funding” are calculated.
* Control condition. This represents a scenario where there is no aggregation and no wisdom of crowd: a proposal’s estimated quality is the grade given by one of the reviewers, picked at random (uniform).

Next to the 5 principles for the design of review panels, the model also incorporates some contextual constraints to the review process, such as the distribution of true quality among the submitted proposals (determined by “α and β”), the number of submitted proposals (“number of proposals”), and the number of proposals that a panel has to recommend for funding (“funding rate”).

Generally, our expectations are that noisier conditions (e.g. low number of reviewers, high reviewer error, high grade language heterogeneity) might be detrimental to panel performance. We also expect the funding rate to impact performance: lower funding rates might mean that the panel has a harder task.

## Emergence

The main emergent phenomenon in this model is the wisdom of crowd: the degree to which review panels perform better than their individual reviewers. Wisdom of crowd is observable by comparing the outcome measures in the control condition with the outcome measures from other aggregation rules. Simulation results show strong and robust wisdom of crowd, and allow to identify conditions and panel designs that boost (or reduce) the wisdom of crowd effect in the simulated review panels.

## Adaptation

This model includes no adaptation.

## Objectives

Although objectives do not explicitly drive agent behavior in this simulation model, it is implicitly assumed that:

* Proposals have the objective of being recommended for funding (found to be deserving funding == TRUE);
* Reviewers have the objective of making an accurate estimation of the proposals’ true quality;
* Review panels strive to maximize performance (as measured by the outcome measures).

## Learning

There is no learning implemented in the model.

## Prediction

Prediction is not implemented in the model.

## Sensing

Sensing is only implemented for reviewers, who “sense” the true quality of a proposal as a more-or-less noisy signal. Reviewers’ sensing of proposals’ true quality is described in section 7: “Submodels / rate”.

## Interaction

The lack of interaction between agents is why this model does not strictly qualify as an ABM.

## Stochasticity

Stochasticity plays a role in various instances, specifically in the:

* calculation of true quality. Randomness does model variability in proposals’ true quality.
* calculation of reviewer assessment (i.e. in the rating function). Randomness, is determined by “reviewer error”, and implicitly encapsulates all that affects a reviewer’s capacity to estimate the proposal’s true quality (e.g. competence, expertise, tiredness, cognitive load, etc.) Randomness captures the variability in reviewer cognitive process, other than interpretation of the grade language itself (see next point).
* creation of discretization thresholds constituting a reviewer’s interpretation of the grading language. These thresholds are drawn at random from a uniform distribution (truncated as not to exceed the range [0,1].

## Collectives

There are no collectives in this model.

## Observation

The model output consists of the outcome variables measured for each of the specified aggregation rules (and for each level of acceptance rate indicated). For the results reported in the accompanying paper, all outcome measures were averaged across a battery of independent simulation runs initialized with the same parameter configuration and a different random seed (see scripts “batteries.r” and “results.r”).

# 5. Initialization

The first initialization step consists of setting the random seed for the random number generation. For this step we rely on the defaults of the R function “set.seed”, which is the Mersenne-Twister method.

Then, proposals are created with two attributes: “true quality”, sampled from a beta distribution with parameters α and β, and the resulting “true ranking”. In the R scripts, these data are stored in a data.frame named submissions.

Then, a review network is created by creating a binary matrix (“rnw”) with proposals as columns and reviewers as rows. The matrix is built to ensure that all proposals are assigned exactly as many reviewers as specified by the “number of reviewers per proposal”, and to ensure that all reviewers are assigned *at least* as many proposals as given by the “number of proposals per reviewer”.

Then, each reviewer in the network is given an interpretation of the grading language (see section 7: subprocess “createGradeLanguage”), and a degree of error. For each reviewer r, errorr is sampled from a normal distribution truncated at (0,1), with mean “reviewer error” and standard deviation 0.2.

# 6. Input data

The model does not use input data to represent time-varying processes. However, the model is built on (and its parameters are calibrated after) empirical data from a research funding agency – the details are provided in Appendix A to the accompanying paper; the data itself cannot be distributed and are thus excluded from this repository.

# 7. Submodels

All submodels are implemented as functions in the script “util.r”, or as part of the general “simulation” function from “simulation.r”. Submodels are here introduced sequentially, following the order in which they are executed during a simulation run.

## allocationNetwork

This function creates the binary matrix that encapsulates the review network (i.e. which reviewer reviews which proposal). The network is created with the following constraints:

* A fixed “number of proposals”;
* A fixed “number of reviewers per proposal”;
* A minimum “number of proposals per reviewer”.

## createGradeLanguage

The grading language is a set of L-1 thresholds on the true quality scale [0,1] that thus discretize it into L partitions: the L categories in the grading language.

If “grade language” is set to “symmetric”, the thresholds are evenly distributed in the interval [0,1], which creates an even, regular partition of the true quality scale. However, if “grade language” is “asymmetric”, then the thresholds are concentrated on the high-end of the true quality scale. This causes reviewers to have higher standards for what deserves higher grades – hence why, in the accompanying paper, the “asymmetric” grade language is called “strict interpretation of the grading language”. The asymmetric grade language is created by positioning thresholds according to this equation:

Where xl is the point on the true quality scale where the l-th threshold is.

Furthermore, regardless of whether the scale is symmetric or asymmetric, each reviewer may have a unique interpretation of the grading language – and interpretations are more different if the parameter H is higher. So, the threshold xl for reviewer r is drawn from a (truncated) normal distribution with these parameters:

... where the mean xl is the l-th threshold as determined by the symmetric or asymmetric grade language parameter.

## Rate

The function rate() determines the grade that a reviewer will give to a proposal. First, a reviewer r makes an evaluation of a proposal p by “sensing” its true quality:

This equation implements Gaussian noise in the evaluation: the reviewer’s estimation sr,p is drawn from a normal distribution that has the proposal’s true quality as mean, and the error as standard deviation.

Then, this evaluation is translated into a grade in the correct grading language. Given the reviewer’s interpretation of the grading language (i.e. the reviewer’s unique set of discretization thresholds), the chosen grade is that of the interval on the true quality scale where sr,p falls.

For convenience, the R script “simulation.r” then normalizes the grade so that it ranges between 0 and 1; this is done by dividing the grade by the number of categories L (which can also be seen as the maximum grade available).

# References

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., Müller, B., Pe’er, G., Piou, C., Railsback, S. F., Robbins, A. M., … DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, *198*(1–2), 115–126. https://doi.org/10.1016/j.ecolmodel.2006.04.023

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., & Railsback, S. F. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, *221*(23), 2760–2768. https://doi.org/10.1016/j.ecolmodel.2010.08.019